

debian

Referensi Debian

Osamu Aoki

Hak Cipta © 2013-2024 Osamu Aoki

Referensi Debian ini (versi 2.152) (2026-10-10 14:56:25 UTC) ditujukan untuk memberikan ringkasan yang lebar atas sistem Debian sebagai panduan pengguna pasca instalasi. Ini mencakup banyak aspek administrasi sistem melalui contoh-contoh perintah shell bagi mereka yang bukan pengembang.

Daftar Isi

1 Tutorial GNU/Linux **1**

1.1 Dasar-dasar konsol . . . 1

1.1.1 Prompt shell . . . 1

1.1.2 Prompt shell di bawah GUI . . . 2

1.1.3 Akun root . . . 2

1.1.4 Prompt shell root . . . 3

1.1.5 Alat administrasi sistem GUI . . . 3

1.1.6 Konsol virtual . . . 3

1.1.7 Cara meninggalkan command prompt . . . 4

1.1.8 Bagaimana mematikan sistem . . . 4

1.1.9 Memulihkan suatu konsol yang waras . . . 4

1.1.10 Saran paket tambahan untuk newbie . . . 4

1.1.11 Akun pengguna tambahan . . . 5

1.1.12 konfigurasi sudo . . . 5

1.1.13 Waktu bermain . . . 6

1.2 Sistem berkas mirip Unix . . . 6

1.2.1 Dasar-dasar berkas Unix . . . 7

1.2.2 Internal sistem berkas . . . 8

1.2.3 Hak akses sistem berkas . . . 8

1.2.4 Kontrol izin untuk berkas yang baru dibuat: umask . . . 11

1.2.5 Izin untuk grup pengguna (grup) . . . 11

1.2.6 Stempel waktu . . . 13

1.2.7 Taut . . . 14

1.2.8 Pipa bernama (FIFO) . . . 15

1.2.9 Soket . . . 15

1.2.10 Berkas perangkat . . . 16

1.2.11 Berkas perangkat khusus . . . 17

1.2.12 procfs dan sysfs . . . 17

1.2.13 tmpfs . . . 17

1.3 Midnight Commander (MC) . . . 18

1.3.1 Penyesuaian MC . . . 18
1.3.2 Memulai MC . . . 18
1.3.3 Manajer berkas di MC . . . 19
1.3.4 Trik baris perintah di MC . . . 19
1.3.5 Penyunting internal di MC . . . 19
1.3.6 Penampil internal di MC . . . 20
1.3.7 Fitur mulai sendiri dari MC . . . 20
1.3.8 Sistem berkas virtual MC . . . 20
1.4 Lingkungan kerja dasar mirip Unix . . . 20
1.4.1 Shell log masuk . . . 21
1.4.2 Menyesuaikan bash . . . 21
1.4.3 Ketukan tombol khusus . . . 22
1.4.4 Operasi tetikus . . . 23
1.4.5 Pager . . . 23
1.4.6 Penyunting teks . . . 24
1.4.7 Menyiapkan penyunting teks default . . . 24
1.4.8 Menggunakan vim . . . 24
1.4.9 Merekam aktivitas shell . . . 26
1.4.10 Perintah Unix Dasar . . . 26
1.5 Perintah shell sederhana . . . 28
1.5.1 Eksekusi perintah dan variabel lingkungan . . . 28
1.5.2 Variabel ”$LANG” . . . 29
1.5.3 Variabel ”$PATH” . . . 30
1.5.4 Variabel ”$HOME” . . . 30
1.5.5 Opsi baris perintah . . . 30
1.5.6 Glob shell . . . 31
1.5.7 Nilai kembalian perintah . . . 32
1.5.8 Urutan perintah umum dan pengalihan shell . . . 32
1.5.9 Alias perintah . . . 34
1.6 Pemrosesan teks mirip Unix . . . 35
1.6.1 Alat teks Unix . . . 35
1.6.2 Ekspresi reguler . . . 36
1.6.3 Ekspresi penggantian . . . 37
1.6.4 Substitusi global dengan ekspresi reguler . . . 37
1.6.5 Mengekstrak data dari tabel berkas teks . . . 38
1.6.6 Cuplikan skrip untuk perintah perpipaan . . . 40

2 Manajemen paket Debian **41**

2.1 Prasyarat manajemen paket Debian . . . 41
2.1.1 Sistem manajemen paket Debian . . . 41
2.1.2 Konfigurasi paket . . . 41
2.1.3 Tindakan pencegahan dasar . . . 42
2.1.4 Hidup dengan peningkatan abadi . . . 43
2.1.5 Dasar-dasar arsip Debian . . . 44
2.1.6 Debian adalah perangkat lunak 100% bebas . . . 48
2.1.7 Dependensi paket . . . 49
2.1.8 Alur kejadian manajemen paket . . . 50
2.1.9 Tanggapan pertama terhadap masalah manajemen paket . . . 51
2.1.10 Cara memilih paket Debian . . . 51
2.1.11 Bagaimana menghadapi persyaratan yang bertentangan . . . 52
2.2 Operasi manajemen paket dasar . . . 52
2.2.1 `apt` vs. `apt-get` / `apt-cache` vs. `aptitude` . . . 53
2.2.2 Operasi manajemen paket dasar dengan barus perintah . . . 54
2.2.3 Penggunaan aptitude interaktif . . . 55
2.2.4 Pengikatan tombol dari aptitude . . . 55
2.2.5 Tampilan paket di bawah aptitude . . . 56
2.2.6 Opsi metode pencarian dengan aptitude . . . 57
2.2.7 Rumus regex aptitude . . . 58
2.2.8 Resolusi ketergantungan aptitude . . . 58
2.2.9 Log aktivitas paket . . . 58
2.3 Contoh operasi aptitude . . . 60
2.3.1 Mencari paket yang menarik . . . 60
2.3.2 Menampilkan daftar paket dengan pencocokan regex pada nama paket . . . 60
2.3.3 Meramban dengan pencocokan regex . . . 60
2.3.4 Membersihkan paket yang dihapus untuk selamanya . . . 60
2.3.5 Merapikan status pemasangan otomatis/manual . . . 61
2.3.6 Peningkatan seluruh sistem . . . 61
2.4 Operasi manajemen paket tingkat lanjut . . . 64
2.4.1 Operasi manajemen paket tingkat lanjut dengan baris perintah . . . 64
2.4.2 Verifikasi berkas-berkas paket yang terpasang . . . 64
2.4.3 Menjaga masalah paket . . . 65
2.4.4 Mencari pada data meta paket . . . 65
2.5 Internal manajemen paket Debian . . . 65
2.5.1 Meta data arsip . . . 65
2.5.2 Berkas ”Release” tingkat puncak dan keaslian . . . 66
2.5.3 Berkas ”Release” tingkat arsip . . . 67

2.5.4 Pengambilan data meta untuk paket . . . 67
2.5.5 Keadaan paket untuk APT . . . 68
2.5.6 Keadaan paket untuk aptitude . . . 68
2.5.7 Salinan lokal dari paket yang diambil . . . 68
2.5.8 Nama berkas paket Debian . . . 68
2.5.9 Perintah dpkg . . . 69
2.5.10 Perintah update-alternatives . . . 69
2.5.11 Perintah dpkg-statoverride . . . 70
2.5.12 Perintah dpkg-divert . . . 71
2.6 Pemulihan dari sistem yang rusak . . . 71
2.6.1 Ketidakcocokan dengan konfigurasi pengguna lama . . . 71
2.6.2 Kesalahan penyinggahan atas data paket . . . 71
2.6.3 Penyelamatan dengan perintah dpkg . . . 71
2.6.4 Instalasi gagal karena dependensi yang kurang . . . 72
2.6.5 Paket yang berbeda dengan berkas-berkas yang tumpang tindih . . . 72
2.6.6 Memperbaiki skrip paket yang rusak . . . 73
2.6.7 Memulihkan data pemilihan paket . . . 73
2.7 Tips untuk manajemen paket . . . 73
2.7.1 Siapa yang mengunggah paket tersebut? . . . 74
2.7.2 Membatasi bandwidth unduhan untuk APT . . . 74
2.7.3 Pengunduhan dan peningkatan paket secara otomatis . . . 74
2.7.4 Pembaruan dan Backport . . . 74
2.7.5 Arsip paket eksternal . . . 75
2.7.6 Paket dari campuran sumber arsip tanpa **apt-pinning** . . . 75
2.7.7 Menyetel halus versi kandidat dengan **apt-pinning** . . . 76
2.7.8 Memblokir paket yang dipasang oleh ”Recommends” . . . 78
2.7.9 Melacak `testing` dengan beberapa paket dari `unstable` . . . 78
2.7.10 Pelacakan `unstable` dengan beberapa paket dari `experimental` . . . 80
2.7.11 Penurunan tingkat darurat . . . 80
2.7.12 Paket equivs . . . 81
2.7.13 Mem-port paket ke sistem stable . . . 81
2.7.14 Server proksi untuk APT . . . 82
2.7.15 Lebih banyak bacaan untuk manajemen paket . . . 82

3 Inisialisasi sistem 84
3.1 Ringkasan proses boot strap . . . 84
3.1.1 Tahap 1: UEFI . . . 84
3.1.2 Tahap 2: boot loader . . . 85
3.1.3 Tahap 3: sistem mini-Debian . . . 86
3.1.4 Tahap 4: sistem Debian normal . . . 87
3.2 Sistem penyelamat . . . 87
3.2.1 Sistem penyelamat GRUB UEFI pada USB . . . 88
3.2.2 Sistem penyelamat Linux live pada USB . . . 89
3.2.3 Sistem penyelamat Linux live dari GRUB . . . 90
3.3 Systemd . . . 90
3.3.1 Init systemd . . . 90
3.3.2 Login systemd . . . 91
3.4 Pesan kernel . . . 92
3.5 Pesan sistem . . . 92
3.6 Manajemen sistem . . . 92
3.7 Pemantau sistem lainnya . . . 94
3.8 Konfigurasi sistem . . . 94
3.8.1 Nama host . . . 94
3.8.2 Sistem berkas . . . 94
3.8.3 Inisialisasi antarmuka jaringan . . . 95
3.8.4 Inisialisasi sistem cloud . . . 95
3.8.5 Contoh penyesuaian untuk mengubah layanan sshd . . . 95
3.9 Sistem udev . . . 96
3.10 Inisialisasi modul kernel . . . 96

4 Kontrol akses dan autentikasi 98
4.1 Autentikasi Unix normal . . . 98
4.2 Mengelola informasi akun dan kata sandi . . . 100
4.3 Kata sandi yang baik . . . 100
4.4 Membuat kata sandi terenkripsi . . . 101
4.5 PAM dan NSS . . . 101
4.5.1 Berkas konfigurasi yang diakses oleh PAM dan NSS . . . 102
4.5.2 Manajemen sistem terpusat modern . . . 103
4.5.3 ”Mengapa GNU su tidak mendukung kelompok wheel” . . . 103
4.5.4 Aturan kata sandi yang lebih ketat . . . 103
4.6 Keamanan autentikasi . . . 104
4.6.1 Kata sandi aman di Internet . . . 104
4.6.2 Secure Shell . . . 105

4.6.3 Langkah-langkah keamanan tambahan untuk Internet . . . 105
4.6.4 Mengamankan kata sandi root . . . 105
4.7 Kontrol akses lainnya . . . 106
4.7.1 Access control lists (ACL) . . . 106
4.7.2 sudo . . . 107
4.7.3 PolicyKit . . . 107
4.7.4 Membatasi akses ke beberapa layanan server . . . 107
4.7.5 Fitur keamanan Linux . . . 108

5 Penyiapan jaringan **109**
5.1 Infrastruktur jaringan dasar . . . 109
5.1.1 Resolusi nama host . . . 109
5.1.2 Nama antarmuka jaringan . . . 111
5.1.3 Rentang alamat jaringan untuk LAN . . . 112
5.1.4 Dukungan perangkat jaringan . . . 112
5.2 Konfigurasi jaringan modern untuk desktop . . . 112
5.2.1 Alat konfigurasi jaringan GUI . . . 113
5.3 Konfigurasi jaringan modern tanpa GUI . . . 113
5.4 Konfigurasi jaringan modern untuk cloud . . . 114
5.4.1 Konfigurasi jaringan modern untuk cloud dengan DHCP . . . 114
5.4.2 Konfigurasi jaringan modern untuk cloud dengan IP statik . . . 114
5.4.3 Konfigurasi jaringan modern untuk cloud dengan Network Manager . . . 114
5.5 Konfigurasi jaringan tingkat rendah . . . 115
5.5.1 Perintah Iproute2 . . . 115
5.5.2 Operasi jaringan tingkat rendah yang aman . . . 115
5.6 Optimalisasi jaringan . . . 116
5.6.1 Mencari MTU yang optimal . . . 116
5.6.2 Optimasi TCP WAN . . . 117
5.7 Infrastruktur netfilter . . . 117

6 Aplikasi jaringan **120**
6.1 Peramban Web . . . 120
6.1.1 Memalsu string User-Agent . . . 121
6.1.2 Ekstensi peramban . . . 121
6.2 Sistem surat . . . 121
6.2.1 Dasar-dasar surel . . . 121
6.2.2 Batasan layanan surat modern . . . 122
6.2.3 Harapan layanan surat bersejarah . . . 122
6.2.4 Agen transportasi surat (mail transport agent/MTA) . . . 123

6.2.4.1 Konfigurasi exim4 . . . 124
6.2.4.2 Konfigurasi postfix dengan SASL . . . 125
6.2.4.3 Konfigurasi alamat surel . . . 126
6.2.4.4 Operasi dasar MTA . . . 127
6.3 Server dan utilitas akses jarak jauh (SSH) . . . 127
6.3.1 Dasar-dasar SSH . . . 128
6.3.2 Nama pengguna di host jarak jauh . . . 128
6.3.3 Menyambungkan tanpa kata sandi jarak jauh . . . 129
6.3.4 Berurusan dengan klien SSH alien . . . 129
6.3.5 Menyiapkan ssh-agent . . . 129
6.3.6 Mengirim surat dari host jarak jauh . . . 130
6.3.7 Penerusan port untuk tunneling SMTP/POP3 . . . 130
6.3.8 Cara mematikan sistem jarak jauh di SSH . . . 130
6.3.9 Pemecahan masalah SSH . . . 130
6.4 Server cetak dan utilitas . . . 131
6.5 Server aplikasi jaringan lainnya . . . 131
6.6 Klien aplikasi jaringan lainnya . . . 132
6.7 Diagnosis daemon sistem . . . 132

7 Sistem GUI 134

7.1 Lingkungan desktop GUI . . . 134
7.2 Protokol komunikasi GUI . . . 135
7.3 Infrastruktur GUI . . . 136
7.4 Aplikasi GUI . . . 137
7.5 Direktori pengguna . . . 137
7.6 Fonta . . . 137
7.6.1 Fonta dasar . . . 137
7.6.2 Rasterisasi fonta . . . 140
7.7 Sandbox . . . 141
7.8 Desktop jarak jauh . . . 142
7.9 Sambungan server X . . . 142
7.9.1 koneksi lokal server X . . . 142
7.9.2 Sambungan jarak jauh server X . . . 143
7.9.3 Koneksi chroot server X . . . 143
7.10 Papanklip . . . 143

8 I18N dan L10N 145

8.1 Lokal . . . 145
8.1.1 Alasan untuk lokal UTF-8 . . . 145
8.1.2 Konfigurasi ulang lokal . . . 146
8.1.3 Pengodean nama berkas . . . 147
8.1.4 Pesan terlokalkan dan dokumentasi yang diterjemahkan . . . 147
8.1.5 Efek dari lokal . . . 148
8.2 Masukan papan ketik . . . 148
8.2.1 Masukan papan ketik untuk konsol Linux dan X Window . . . 149
8.2.2 Masukan papan ketik untuk Wayland . . . 149
8.2.3 Dukungan metode masukan dengan IBus . . . 149
8.2.4 Dukungan metode masukan dengan Fcitx . . . 152
8.2.5 Contoh untuk bahasa Jepang . . . 152
8.3 Keluaran tampilan . . . 152
8.3.1 Konfigurasi terminal . . . 153
8.3.2 Karakter Lebar Karakter Ambigu Asia Timur . . . 153

9 Tips sistem 154

9.1 Tips konsol . . . 154
9.1.1 Merekam aktivitas shell secara bersih . . . 154
9.1.2 Program screen . . . 155
9.1.3 Menavigasi di sekitar direktori . . . 156
9.1.4 Pembungkus readline . . . 156
9.1.5 Memindai pohon kode sumber . . . 156
9.2 Menyesuaikan vim . . . 157
9.2.1 Menyesuaikan vim dengan fitur internal . . . 157
9.2.2 Menyesuaikan vim dengan paket eksternal . . . 159
9.3 Perekaman dan presentasi data . . . 160
9.3.1 Daemon log . . . 160
9.3.2 Penganalisis log . . . 160
9.3.3 Tampilan data teks yang dikustomisasi . . . 161
9.3.4 Tampilan waktu dan tanggal yang disesuaikan . . . 161
9.3.5 Echo shell berwarna . . . 162
9.3.6 Perintah berwarna . . . 162
9.3.7 Merekam aktivitas penyunting untuk pengulangan yang kompleks . . . 163
9.3.8 Merekam gambar grafis dari aplikasi X . . . 163
9.3.9 Merekam perubahan dalam berkas konfigurasi . . . 163
9.4 Memantau, mengendalikan, dan memulai aktivitas program . . . 164
9.4.1 Mencatat waktu eksekusi proses . . . 164

9.4.2 Prioritas penjadwalan . . . 164
9.4.3 Perintah ps . . . 165
9.4.4 Perintah top . . . 165
9.4.5 Daftar berkas yang dibuka oleh suatu proses . . . 165
9.4.6 Menelusuri aktivitas program . . . 165
9.4.7 Identifikasi proses menggunakan berkas atau soket . . . 166
9.4.8 Mengulangi perintah dengan interval konstan . . . 166
9.4.9 Mengulangi perintah atas berkas . . . 166
9.4.10 Memulai program dari GUI . . . 167
9.4.11 Menyesuaikan program yang akan dimulai . . . 168
9.4.12 Membunuh sebuah proses . . . 169
9.4.13 Menjadwalkan tugas sekali . . . 169
9.4.14 Menjadwalkan tugas secara teratur . . . 170
9.4.15 Menjadwalkan tugas pada acara . . . 170
9.4.16 Tombol Alt-SysRq . . . 170
9.5 Tips pemeliharaan sistem . . . 171
9.5.1 Siapa yang ada di sistem? . . . 171
9.5.2 Memperingatkan semua orang . . . 172
9.5.3 Identifikasi perangkat keras . . . 172
9.5.4 Konfigurasi perangkat keras . . . 172
9.5.5 Waktu sistem dan perangkat keras . . . 172
9.5.6 Infrastruktur suara . . . 173
9.5.7 Menonaktifkan screen saver . . . 174
9.5.8 Menonaktifkan suara bip . . . 174
9.5.9 Penggunaan memori . . . 175
9.5.10 Pemeriksaan keamanan dan integritas sistem . . . 175
9.6 Tips penyimpanan data . . . 176
9.6.1 Penggunaan ruang disk . . . 176
9.6.2 Konfigurasi partisi disk . . . 177
9.6.3 Mengakses partisi menggunakan UUID . . . 177
9.6.4 LVM2 . . . 178
9.6.5 Konfigurasi sistem berkas . . . 178
9.6.6 Pembuatan sistem berkas dan pemeriksaan integritas . . . 179
9.6.7 Optimalisasi sistem berkas dengan opsi mount . . . 179
9.6.8 Optimasi sistem berkas melalui superblok . . . 180
9.6.9 Optimalisasi hard disk . . . 180
9.6.10 Optimasi solid state drive . . . 180
9.6.11 Menggunakan SMART untuk memprediksi kegagalan hard disk . . . 181
9.6.12 Menentukan direktori penyimpanan sementara melalui $TMPDIR . . . 181

9.6.13 Perluasan ruang penyimpanan yang dapat digunakan melalui LVM . . . 181
9.6.14 Perluasan ruang penyimpanan yang dapat digunakan dengan memasang partisi lain . . . 181
9.6.15 Perluasan ruang penyimpanan yang dapat digunakan dengan mengait-bind direktori lain . . . 182
9.6.16 Perluasan ruang penyimpanan yang dapat digunakan dengan mengait overlay direktori lain . . . 182
9.6.17 Perluasan ruang penyimpanan yang dapat digunakan menggunakan symlink . . . 182
9.7 Image disk . . . 182
9.7.1 Membuat berkas image disk . . . 183
9.7.2 Menulis secara langsung ke disk . . . 183
9.7.3 Mengait berkas image disk . . . 183
9.7.4 Membersihkan berkas image disk . . . 185
9.7.5 Membuat berkas image disk kosong . . . 185
9.7.6 Membuat berkas image ISO9660 . . . 186
9.7.7 Menulis secara langsung ke CD/DVD-R/RW . . . 187
9.7.8 Mengait berkas image ISO9660 . . . 187
9.8 Data biner . . . 187
9.8.1 Melihat dan menyunting data biner . . . 187
9.8.2 Memanipulasi berkas tanpa mengait disk . . . 188
9.8.3 Redundansi data . . . 188
9.8.4 Pemulihan berkas data dan analisis forensik . . . 188
9.8.5 Memecah sebuah berkas besar menjadi berkas-berkas kecil . . . 189
9.8.6 Menghapus konten berkas . . . 189
9.8.7 Berkas dummy . . . 189
9.8.8 Menghapus seluruh hard disk . . . 190
9.8.9 Menghapus area hard disk yang tidak terpakai . . . 190
9.8.10 Membatalkan penghapusan berkas yang dihapus tapi masih terbuka . . . 191
9.8.11 Mencari semua hardlink . . . 191
9.8.12 Konsumsi ruang disk yang tak terlihat . . . 191
9.9 Tips enkripsi data . . . 192
9.9.1 Enkripsi disk lepasan dengan dm-crypt/LUKS . . . 192
9.9.2 Mengait disk terenkripsi dengan dm-crypt/LUKS . . . 193
9.10 Kernel . . . 193
9.10.1 Parameter kernel . . . 193
9.10.2 Header kernel . . . 193
9.10.3 Mengompail kernel dan modul terkait . . . 194
9.10.4 Mengompail sumber kernel: Rekomendasi Tim Kernel Debian . . . 194
9.10.5 Driver perangkat keras dan firmware . . . 195
9.11 Sistem tervirtualisasi . . . 196
9.11.1 Alat virtualisasi dan emulasi . . . 196
9.11.2 Alur kerja virtualisasi . . . 198
9.11.3 Mengait berkas image disk virtual . . . 198
9.11.4 Sistem chroot . . . 199
9.11.5 Beberapa sistem desktop . . . 200

10 Manajemen data **201**

10.1 Berbagi, menyalin, dan mengarsipkan . . . 201
10.1.1 Alat arsip dan kompresi . . . 202
10.1.2 Alat salin dan sinkronisasi . . . 202
10.1.3 Idiom untuk arsip . . . 202
10.1.4 Idiom untuk menyalin . . . 204
10.1.5 Idiom untuk pemilihan berkas . . . 205
10.1.6 Media arsip . . . 206
10.1.7 Perangkat penyimpanan lepasan . . . 207
10.1.8 Pilihan sistem berkas untuk berbagi data . . . 208
10.1.9 Berbagi data melalui jaringan . . . 210
10.2 Pencadangan dan pemulihan . . . 210
10.2.1 Kebijakan pencadangan dan pemulihan . . . 210
10.2.2 Keluarga utilitas pencadangan . . . 212
10.2.3 Tips pencadangan . . . 213
10.2.3.1 Pencadangan GUI . . . 213
10.2.3.2 Pencadangan yang dipicu kejadian mount . . . 214
10.2.3.3 Pencadangan yang dipicu oleh kejadian pewaktu . . . 214
10.3 Infrastruktur keamanan data . . . 215
10.3.1 Manajemen kunci untuk GnuPG . . . 216
10.3.2 Menggunakan GnuPG pada berkas . . . 217
10.3.3 Menggunakan GnuPG dengan Mutt . . . 217
10.3.4 Menggunakan GnuPG dengan Vim . . . 217
10.3.5 Sidikjari MD5 . . . 217
10.3.6 Ring kunci kata sandi . . . 219
10.4 Alat penggabungan kode sumber . . . 219
10.4.1 Mengekstrak perbedaan untuk berkas sumber . . . 219
10.4.2 Menggabungkan pembaruan untuk berkas sumber . . . 221
10.4.3 Penggabungan interaktif . . . 221
10.5 Git . . . 221
10.5.1 Konfigurasi klien Git . . . 221
10.5.2 Perintah Git dasar . . . 222
10.5.3 Tips Git . . . 223
10.5.4 Referensi Git . . . 225
10.5.5 Sistem kontrol versi lainnya . . . 225

11 Konversi data **226**
11.1 Alat konversi data teks . . . 226
11.1.1 Mengonversi berkas teks dengan iconv . . . 226
11.1.2 Memeriksa berkas apakah UTF-8 dengan iconv . . . 228
11.1.3 Mengonversi nama berkas dengan iconv . . . 228
11.1.4 Konversi EOL . . . 228
11.1.5 Konversi TAB . . . 229
11.1.6 Penyunting dengan konversi otomatis . . . 229
11.1.7 Ekstraksi teks polos . . . 230
11.1.8 Menyoroti dan memformat data teks polos . . . 230
11.2 Data XML . . . 232
11.2.1 Petunjuk dasar untuk XML . . . 232
11.2.2 Pemrosesan XML . . . 233
11.2.3 Ekstraksi data XML . . . 234
11.2.4 Lint data XML . . . 234
11.3 Tata cetak . . . 234
11.3.1 typesetting roff . . . 235
11.3.2 TeX/LaTeX . . . 235
11.3.3 Mencetak cantik halaman manual . . . 236
11.3.4 Membuat halaman manual . . . 236
11.4 Data yang dapat dicetak . . . 236
11.4.1 Ghostscript . . . 237
11.4.2 Menggabungkan dua berkas PS atau PDF . . . 237
11.4.3 Utilitas data yang dapat dicetak . . . 237
11.4.4 Mencetak dengan CUPS . . . 237
11.5 Konversi data surat . . . 239
11.5.1 Dasar-dasar data surel . . . 239
11.6 Alat data grafis . . . 240
11.6.1 Alat data grafis (paket-meta) . . . 240
11.6.2 Alat data grafis (GUI) . . . 240
11.6.3 Alat data grafis (CLI) . . . 243
11.7 Konversi data lain-lain . . . 243

12 Pemrograman **244**
12.1 Skrip shell . . . 244
12.1.1 Kompatibilitas shell POSIX . . . 245
12.1.2 Parameter shell . . . 245
12.1.3 Kondisional Shell . . . 246
12.1.4 Loop shell . . . 247

12.1.5 Variabel lingkungan shell . . . 247
12.1.6 Urutan pemrosesan baris perintah shell . . . 248
12.1.7 Program utilitas untuk skrip shell . . . 249
12.2 Scripting dalam bahasa yang diinterpretasi . . . 249
12.2.1 Debugging kode bahasa yang diinterpretasi . . . 250
12.2.2 Program GUI dengan skrip shell . . . 250
12.2.3 Tindakan ubahan untuk filer GUI . . . 251
12.2.4 Kegilaan skrip pendek Perl . . . 251
12.3 Menuis kode dalam bahasa yang dikompilasi . . . 252
12.3.1 C . . . 252
12.3.2 Program C Sederhana (gcc) . . . 253
12.3.3 Flex - Lex yang lebih baik . . . 253
12.3.4 Bison - Yacc yang lebih baik . . . 254
12.4 Alat analisis kode statis . . . 255
12.5 Awakutu . . . 257
12.5.1 Eksekusi gdb dasar . . . 257
12.5.2 Debugging paket Debian . . . 257
12.5.3 Mendapatkan backtrace . . . 258
12.5.4 Perintah gdb tingkat lanjut . . . 259
12.5.5 Periksa ketergantungan pada pustaka . . . 259
12.5.6 Alat pelacakan panggilan dinamis . . . 259
12.5.7 Men-debug Galat X . . . 259
12.5.8 Alat deteksi kebocoran memori . . . 260
12.5.9 Disassembly biner . . . 260
12.6 Alat build . . . 260
12.6.1 Make . . . 260
12.6.2 Autotools . . . 261
12.6.2.1 Mengkompilasi dan menginstall program . . . 261
12.6.2.2 Menghapus instalasi program . . . 262
12.6.3 Meson . . . 262
12.7 Web . . . 262
12.8 Terjemahan kode sumber . . . 263
12.9 Membuat paket Debian . . . 263

A Lampiran **264**

A.1 Labirin Debian . . . 264
A.2 Riwayat hak cipta . . . 264
A.3 Format dokumen . . . 265

Daftar Tabel

1.1 Daftar paket program mode teks yang menarik . . . 5
1.2 Daftar paket dokumentasi informatif . . . 5
1.3 Daftar penggunaan direktori kunci . . . 8
1.4 Daftar karakter pertama dari keluaran "`ls -l`" . . . 9
1.5 Mode numerik untuk izin berkas dalam perintah chmod(1) . . . 10
1.6 Contoh-contoh nilai **umask** . . . 11
1.7 Daftar grup yang disediakan sistem terkenal untuk akses berkas . . . 12
1.8 Daftar grup penting yang disediakan untuk eksekusi perintah tertentu . . . 13
1.9 Daftar jenis stempel waktu . . . 13
1.10 Daftar berkas perangkat khusus . . . 17
1.11 Pengikatan tombol MC . . . 19
1.12 Reaksi terhadap tombol enter di MC . . . 20
1.13 Daftar program shell . . . 21
1.14 Daftar pengikatan kunci untuk bash . . . 22
1.15 Daftar operasi tetikus dan tindakan tombol terkait pada Debian . . . 23
1.16 Daftar ketukan tombol Vim dasar . . . 25
1.17 Daftar perintah dasar Unix . . . 27
1.18 3 bagian dari nilai lokal . . . 29
1.19 Daftar rekomendasi lokal . . . 29
1.20 Daftar nilai "`$HOME`" . . . 30
1.21 Pola glob shell . . . 31
1.22 Kode keluar perintah . . . 32
1.23 Idiom perintah Shell . . . 33
1.24 Deskriptor berkas yang telah ditentukan . . . 34
1.25 Karakter meta untuk BRE dan ERE . . . 36
1.26 Ekspresi penggantian . . . 37
1.27 Daftar cuplikan skrip untuk perintah perpipaan . . . 40

2.1 Daftar alat manajemen paket Debian . . . 42
2.2 Daftar situs arsip Debian . . . 45

2.3 Daftar area arsip Debian . . . 46
2.4 Hubungan antara keluarga dan nama kode . . . 47
2.5 Daftar situs web kunci untuk menyelesaikan masalah dengan paket tertentu . . . 51
2.6 Operasi manajemen paket dasar dengan baris perintah menggunakan apt(8), aptitude(8), dan apt-get(8) /apt-cache(8) . . . 54
2.7 Opsi perintah penting untuk aptitude(8) . . . 55
2.8 Daftar pengikatan tombol untuk aptitude . . . 56
2.9 Daftar tampilan untuk aptitude . . . 57
2.10 Kategorisasi tampilan paket standar . . . 57
2.11 Daftar rumus regex aptitude . . . 59
2.12 Berkas-berkas log untuk aktivitas paket . . . 60
2.13 Daftar operasi manajemen paket tingkat lanjut . . . 63
2.14 Isi data meta arsip Debian . . . 65
2.15 Struktur nama paket-paket Debian . . . 68
2.16 Karakter yang dapat digunakan untuk setiap komponen dalam nama paket Debian . . . 69
2.17 Berkas-berkas penting yang dibuat oleh `dpkg` . . . 70
2.18 Daftar nilai Pin-Priority yang terkenal untuk teknik **apt-pinning** . . . 77
2.19 Daftar alat proksi khusus untuk arsip Debian . . . 82

3.1 Daftar boot loader . . . 85
3.2 Arti dari entri menu dari bagian di atas dari `/boot/grub/grub.cfg` . . . 86
3.3 Daftar utilitas boot untuk sistem Debian . . . 88
3.4 Daftar tingkat kesalahan kernel . . . 92
3.5 Daftar cuplikan perintah `journalctl` yang umum . . . 92
3.6 Daftar cuplikan perintah `systemctl` umum . . . 93
3.7 Daftar cuplikan perintah pemantauan lainnya di bawah `systemd` . . . 94

4.1 3 berkas konfigurasi penting untuk pam_unix(8) . . . 98
4.2 Konten entri kedua dari "`/etc/passwd`" . . . 99
4.3 Daftar perintah untuk mengelola informasi akun . . . 100
4.4 Daftar alat untuk menghasilkan kata sandi . . . 101
4.5 Daftar sistem PAM dan NSS yang terkenal . . . 101
4.6 Daftar berkas konfigurasi yang diakses oleh PAM dan NSS . . . 102
4.7 Daftar layanan dan port yang tidak aman dan aman . . . 104
4.8 Daftar alat untuk memberikan langkah-langkah keamanan tambahan . . . 105

5.1 Daftar alat konfigurasi jaringan . . . 110
5.2 Daftar rentang alamat jaringan . . . 112
5.3 Tabel terjemahan dari perintah `net-tools` yang usang ke perintah-perintah baru `iproute2` . . . 115
5.4 Daftar perintah jaringan tingkat rendah . . . 115

5.5 Daftar alat optimalisasi jaringan . . . 116
5.6 Panduan dasar dari nilai MTU yang optimal . . . 117
5.7 Daftar alat firewall . . . 118

6.1 Daftar peramban web . . . 120
6.2 Daftar agen pengguna surat (mail user agent/MUA) . . . 122
6.3 Daftar paket terkait mail transport agent dasar . . . 123
6.4 Daftar halaman-halaman penting manual postfix . . . 125
6.5 Daftar berkas konfigurasi terkait alamat surel . . . 126
6.6 Daftar operasi dasar MTA . . . 127
6.7 Daftar server dan utilitas akses jarak jauh . . . 128
6.8 Daftar berkas konfigurasi SSH . . . 128
6.9 Daftar contoh awal mula klien SSH . . . 129
6.10 Daftar klien SSH gratis untuk platform lain . . . 129
6.11 Daftar server cetak dan utilitas . . . 131
6.12 Daftar server aplikasi jaringan lainnya . . . 132
6.13 Daftar klien aplikasi jaringan . . . 133
6.14 Daftar RFC populer . . . 133

7.1 Daftar lingkungan desktop . . . 134
7.2 Daftar paket infrastruktur GUI yang terkenal . . . 136
7.3 Daftar aplikasi GUI yang terkenal . . . 138
7.4 Daftar fonta TrueType dan OpenType yang terkenal . . . 139
7.5 Daftar lingkungan fonta terkenal dan paket-paket terkait . . . 140
7.6 Daftar lingkungan sandbox terkenal dan paket terkait . . . 141
7.7 Daftar server akses jarak jauh yang terkenal . . . 142
7.8 Daftar metode koneksi ke server X . . . 142
7.9 Daftar program yang terkait dengan memanipulasi papan klip karakter . . . 144

8.1 Daftar IBus dan paket-paket pengayanya . . . 150
8.2 Daftar paket Fcitx5 dan pengayanya . . . 151

9.1 Daftar program untuk mendukung aktivitas konsol . . . 154
9.2 Daftar pengikatan tombol untuk screen . . . 156
9.3 Informasi tentang inisialisasi `vim` . . . 160
9.4 Daftar penganalisis log sistem . . . 161
9.5 Menampilkan contoh waktu dan tanggal untuk perintah "`ls -l`" dengan **nilai gaya waktu** . . . 162
9.6 Daftar alat manipulasi gambar grafis . . . 163
9.7 Daftar paket yang dapat merekam riwayat konfigurasi . . . 163
9.8 Daftar alat untuk memantau dan mengendalikan aktivitas program . . . 164

9.9 Daftar nilai nice untuk prioritas penjadwalan . . . 165
9.10 Daftar gaya perintah ps . . . 165
9.11 Daftar sinyal yang sering digunakan untuk perintah bunuh . . . 169
9.12 Daftar tombol perintah SAK yang terkenal . . . 171
9.13 Daftar alat identifikasi perangkat keras . . . 172
9.14 Daftar alat konfigurasi perangkat keras . . . 173
9.15 Daftar paket suara . . . 174
9.16 Daftar perintah untuk menonaktifkan screen saver . . . 174
9.17 Daftar ukuran memori yang dilaporkan . . . 175
9.18 Daftar alat untuk pemeriksaan keamanan dan integritas sistem . . . 176
9.19 Daftar paket manajemen partisi disk . . . 177
9.20 Daftar paket manajemen sistem berkas . . . 179
9.21 Daftar paket yang menampilkan dan menyunting data biner . . . 188
9.22 Daftar paket untuk memanipulasi berkas tanpa mengait disk . . . 188
9.23 Daftar alat untuk menambahkan redundansi data ke berkas . . . 188
9.24 Daftar paket untuk pemulihan berkas data dan analisis forensik . . . 189
9.25 Daftar utilitas enkripsi data . . . 192
9.26 Daftar paket kunci yang akan dipasang untuk rekompilasi kernel pada sistem Debian . . . 194
9.27 Daftar alat virtualisasi . . . 197

10.1 Daftar alat arsip dan kompresi . . . 203
10.2 Daftar alat salin dan sinkronisasi . . . 204
10.3 Daftar pilihan sistem berkas untuk perangkat penyimpanan lepasan dengan skenario penggunaan yang khas . . . 209
10.4 Daftar layanan jaringan yang akan dipilih dengan skenario penggunaan umum . . . 210
10.5 Daftar utilitas keluarga pencadangan . . . 212
10.6 Daftar alat infrastruktur keamanan data . . . 215
10.7 Daftar perintah GNU Privacy Guard untuk manajemen kunci . . . 216
10.8 Daftar arti kode kepercayaan . . . 216
10.9 Daftar perintah GNU Privacy Guard pada berkas . . . 218
10.10 Daftar alat penggabungan kode sumber . . . 220
10.11 Daftar paket dan perintah terkait git . . . 222
10.12 Perintah Git Utama . . . 223
10.13 Tips Git . . . 224
10.14 Daftar alat sistem kontrol versi lainnya . . . 225

11.1 Daftar alat konversi data teks . . . 226
11.2 Daftar nilai pengodean dan penggunaannya . . . 227
11.3 Daftar gaya EOL untuk platform yang berbeda . . . 229
11.4 Daftar perintah konversi TAB dari paket `bsdmainutils` dan `coreutils` . . . 229

11.5 Daftar alat untuk mengekstrak data teks polos . . . 230
11.6 Daftar alat untuk menyoroti data teks polos . . . 231
11.7 Daftar entitas terpradefinisi untuk XML . . . 232
11.8 Daftar alat XML . . . 233
11.9 Daftar alat DSSSL . . . 233
11.10 Daftar alat ekstraksi data XML . . . 234
11.11 Daftar alat cetak cantik XML . . . 234
11.12 Daftar alat tata cetak . . . 235
11.13 Daftar paket untuk membantu membuat manpage . . . 236
11.14 Daftar penerjemah PostScript Ghostscript . . . 237
11.15 Daftar utilitas data yang dapat dicetak . . . 238
11.16 Daftar paket untuk membantu konversi data surel . . . 239
11.17 Daftar alat data grafis (paket-meta) . . . 240
11.18 Daftar alat data grafis (GUI) . . . 241
11.19 Daftar alat data grafis (CLI) . . . 242
11.20 Daftar alat konversi data lain-lain . . . 243

12.1 Daftar bashisme khas . . . 245
12.2 Daftar parameter shell . . . 245
12.3 Daftar ekspansi parameter shell . . . 246
12.4 Daftar substitusi parameter shell utama . . . 246
12.5 Daftar operator perbandingan berkas dalam ekspresi bersyarat . . . 247
12.6 Daftar operator perbandingan string dalam ekspresi bersyarat . . . 247
12.7 Daftar paket yang berisi program utilitas kecil untuk skrip shell . . . 249
12.8 Daftar paket terkait interpreter . . . 250
12.9 Daftar program dialog . . . 251
12.10 Daftar paket terkait kompiler . . . 252
12.11 Daftar generator parser LALR yang kompatibel dengan Yacc . . . 254
12.12 Daftar alat untuk analisis kode statis . . . 256
12.13 Daftar paket debug . . . 257
12.14 Daftar perintah gdb tingkat lanjut . . . 259
12.15 Daftar alat deteksi kebocoran memori . . . 260
12.16 Daftar paket alat build . . . 260
12.17 Daftar variabel otomatis make . . . 261
12.18 Daftar ekspansi variabel make . . . 261
12.19 Daftar alat terjemahan kode sumber . . . 263

Ringkasan

Buku ini bebas; Anda dapat mendistribusikan ulang dan/atau mengubahnya di bawah persyaratan sebarang versi GNU General Public License yang memnuhi Debian Free Software Guidelines (DFSG).

Kata Pengantar

Referensi Debian (versi 2.152) ini (2026-10-10 14:56:25 UTC) ditujukan untuk menyediakan ringkasan yang lebar atas sistem Debian sebagai panduan pengguna pasca instalasi.

Target pembaca adalah mereka yang mau belajar shell script tapi tidak siap untuk membaca semua sumber C untuk memahami bagaimana sistem GNU/Linux bekerja.

Untuk petunjuk instalasi, lihat:

- Panduan Instalasi Debian GNU/Linux untuk sistem stabil saat ini
- Panduan Instalasi Debian GNU/Linux untuk sistem testing saat ini

Sangkalan

Semua garansi disangkal. Semua merek dagang adalah milik masing-masing pemegang merek dagang.

Sistem Debian itu sendiri adalah suatu target yang bergerak. Ini membuat dokumentasi sulit untuk terkini dan benar. Walaupun versi `testing` dari sistem Debian saat ini dipakai sebagai dasar untuk menulis ini, beberapa konten mungkin sudah kedaluwarsa saat Anda membaca ini.

Harap perlakukan dokumen ini sebagai acuan sekunder. Dokumen ini tidak menggantikan panduan otoritatif manapun. Penulis dan para kontributor tidak bertanggung jawab atas konsekuensi kesalahan-kesalahan, penghilangan, maupun ketidakjelasan dalam dokumen ini.

Apa itu Debian

Proyek Debian adalah asosiasi para individu yang telah membuat prinsip umum untuk menciptakan suatu sistem operasi yang bebas. Distribusinya dicirikan oleh hal-hal berikut.

- Komitmen ke kebebasan perangkat lunak: Kontrak Sosial Debian dan Panduan Perangkat Lunak Bebas Debian (Debian Free Software Guidelines, DFSG)
- Upaya sukarelawan tak berbayar yang tersebar dan berbasis Internet: https://www.debian.org
- Sejumlah besar paket perangkat lunak kualitas tinggi terprakompilasi
- Fokus pada stabilitas dan keamanan dengan akses mudah ke pembaruan keamanan
- Fokus pada peningkatan ke paket-paket perangkat lunak terbaru secara mulus dalam arsip `testing`
- Sejumlah besar arsitektur perangkat keras yang didukung

Potongan-potongan Perangkat Lunak Bebas dalam Debian berasal dari GNU, Linux, BSD, X, ISC, Apache, Ghostscript, Common Unix Printing System , Samba, GNOME, KDE, Mozilla, LibreOffice, Vim, TeX, LaTeX, DocBook, Perl, Python, Tcl, Java, Ruby, PHP, Berkeley DB, MariaDB, PostgreSQL, SQLite, Exim, Postfix, Mutt, FreeBSD, OpenBSD, Plan 9 dan banyak lagi proyek perangkat lunak bebas yang independen. Debian mengintegrasikan keragaman Perangkat Lunak Bebas ini ke dalam satu sistem.

Tentang dokumen ini

Aturan pemandu

Aturan pemandu berikut diikuti ketika menyusun dokumen ini.

- Menyediakan ikhtisar dan mengabaikan kasus-kasus khusus. **Gambar Kasar**
- Singkat dan Sederhana.(**KISS** Keep It Short and Simple)
- Jangan reinvent the wheel. (Gunakan petunjuk ke **referensi yang sudah ada**)
- Fokus pada alat bukan GUI dan konsol. (Gunakan **contoh-contoh shell**)
- Berlaku obyektif. (Gunakan popcon etc.)

Tip
Saya mencoba menjelaskan aspek-aspek hirarkis dan tingkat yang lebih rendah dari sistem.

Prasyarat



Awas
Anda diharapkan berupaya keras untuk mencari jawaban sendiri lebih jauh dari dokumentasi ini. Dokumen ini hanya memberikan titik awal yang efisien.

Anda harus mencari solusi sendiri dari sumber-sumber primer.

- Situs Debian di https://www.debian.org untuk informasi umum
- Dokumentasi di bawah direktori "`/usr/share/doc`*`nama_paket`*"
- **manpage** gaya Unix: "`dpkg -L` *`nama_paket`* `|grep '/man/man.*/'`"
- **halaman info** gaya GNU: "`dpkg -L` *`nama_paket`* `|grep '/info/'`"
- Laporan bug: https://bugs.debian.org/*nama_paket*
- Wiki Debian di https://wiki.debian.org/ untuk topik-topik yang berubah dan spesifik
- Spesifikasi UNIX Tunggal (Single UNIX Specification) dari Laman Web The UNIX System milik Open Group
- Ensiklopedia bebas dari Wikipedia di https://www.wikipedia.org/
- Buku Pegangan Administrator Debian
- HOWTO dari Proyek Dokumentasi Linux (TLDP)

Catatan
Untuk dokumentasi rinci, Anda mungkin perlu memasang paket dokumentasi yang sesuai dengan akhiran "`-doc`".

Konvensi

Dokumen ini menyediakan informasi melalui gaya presentasi sederhana dengan contoh perintah shell bash(1) berikut ini.

```
# command-in-root-account
$ command-in-user-account
```

Sapaan shell ini membedakan akun yang dipakai dan sesuai dengan variabel lingkungan yang ditata sebagai: ”PS1='\$' ” dan ”PS2=' ' ”. Nilai-nilai ini dipilih agar dokumen ini mudah terbaca dan tidak umum dipakai pada sistem terpasang yang sebenarnya.

Semua contoh perintah dijalankan di bawah locale bahasa Inggris ”LANG=en_US.UTF8”. Jangan berharap string placeholder seperti *command-in-root-account* dan *command-in-user-account* diterjemahkan dalam contoh-contoh perintah. Ini pilihan yang disengaja untuk menjaga agar semua contoh yang diterjemahkan mutakhir.

Catatan
Lihat arti variabel lingkungan ”$PS1” dan ”$PS2” dalam bash(1).

Aksi yang diperlukan oleh administrator sistem ditulis dalam kalimat imperatif, mis. ”Ketikkan tombol Enter setelah mengetikkan setiap string perintah ke shell.”

Kolom **deskripsi** dan yang serupa dalam tabel mungkin berisi **frasa kata benda** mengikuti konvensi deskripsi singkat paket yang membuang artikel pembuka seperti ”a” dan ”the”. Mereka mungkin berisi suatu frasa infinitif seperti **frasa kata benda** tanpa pembuka ”to” mengikuti konvensi deskripsi perintah singkat dalam manpages. Ini mungkin terlihat aneh bagi beberapa orang tapi merupakan pilihan gaya yang disengaja untuk mempertahankan dokumentasi ini sesederhana mungkin. **Frasa kata benda** ini tidak mengkapitalkan awalannya maupun akhirannya dengan titik mengikuti konvensi deskripsi pendek ini.

Catatan
Kata benda yang tepat termasuk nama perintah mempertahankan besar kecilnya huruf tanpa tergantung lokasi mereka.

Suatu **cuplikan perintah** dikutip dalam suatu paragraf teks yang diacu oleh fonta mesin ketik di antara tanda kutip ganda, seperti misalnya ”`aptitude safe-upgrade`”.

Suatu **data teks** dari sebuah berkas konfigurasi yang dikutip dalam sebuah paragraf teks diacu dengan fonta mesin ketik di antara tanda kutip ganda, seperti misalnya ”`deb-src`”.

Suatu **perintah** diacu oleh namanya dalam fonta mesin ketik, dan mungkin juga diikuti oleh nomor seksi halaman man di dalam kurung, seperti misalnya bash(1). Anda disarankan mendapatkan informasi dengan mengetikkan yang berikut.

```
$ man 1 bash
```

Suatu **halaman man** diacu oleh namanya dalam fonta mesin ketik dan diikuti oleh nomor seksi halaman man di dalam kurung, seperti misalnya sources.list(5). Anda disarankan mendapatkan informasi dengan mengetikkan yang berikut.

```
$ man 5 sources.list
```

Suatu **halaman info** diacu oleh cuplikan perintahnya dalam fonta mesin ketik di antara tanda kutip ganda, seperti misalnya ”`info make`”. Anda disarankan untuk mendapatkan informasi dengan mengetikkan perintah berikut.

```
$ info make
```

Suatu **nama berkas** diacu oleh fonta mesin ketik di antara tanda kutip ganda, seperti misalnya ”`/etc/passwd`”. Untuk berkas konfigurasi, Anda disarankan untuk mendapatkan informasi dengan mengetikkan yang berikut.

```
$ sensible-pager "/etc/passwd"
```

Suatu **nama direktori** diacu oleh fonta mesin ketik di antara tanda kutip ganda, seperti misalnya "`/etc/apt/`". Anda disarankan untuk mengeksplorasi isinya dengan mengetikkan yang berikut.

```
$ mc "/etc/apt/"
```

Suatu **nama paket** diacu dengan namanya dalam fonta mesin ketik, seperti misalnya `vim`. Anda disarankan untuk mendapatkan informasi dengan mengetikkan yang berikut.

```
$ dpkg -L vim
$ apt-cache show vim
$ aptitude show vim
```

Suatu **dokumentasi** dapat menunjukkan lokasinya melalui nama berkas dalam fonta mesin tik antara tanda kutip ganda, seperti "`/usr/share/doc/base-passwd/users-and-groups.txt.gz`" dan "`/usr/share/doc/base-passwd/`"; atau URLnya, seperti https://www.debian.org. Anda disarankan untuk membaca dokumentasi dengan mengetik yang berikut.

```
$ zcat "/usr/share/doc/base-passwd/users-and-groups.txt.gz" | sensible-pager
$ sensible-browser "/usr/share/doc/base-passwd/users-and-groups.html"
$ sensible-browser "https://www.debian.org"
```

Sebuah **variabel lingkungan** diacu dengan namanya dengan awalan "$ " dalam fonta mesin tik antara tanda kutip ganda, seperti "`$TERM`". Anda disarankan untuk memperoleh nilai saat ini dengan mengetik yang berikut.

```
$ echo "$TERM"
```

popcon

Data popcon disajikan sebagai ukuran objektif bagi popularitas dari setiap paket. Itu diunduh pada 2026-10-07 05:10:21 UTC dan memuat total kiriman 291550 laporan atas 224298 paket biner dan 25 arsitektur.

Catatan
Harap perhatikan bahwa arsip `amd64 unstable` saat ini hanya memuat 78072 paket. Data popcon memuat laporan dari banyak instalasi sistem lama.

Angka popcon yang diawali dengan "V:" untuk "suara" (votes) dihitung dengan "1000 * (kiriman popcon bagi paket yang baru-baru ini dieksekusi pada PC)/(total kiriman popcon)".

Angka popcon yang diawali dengan "I:" untuk "instalasi" dihitung dengan "1000 * (kiriman popcon bagi paket yang dipasang pada PC)/(total kiriman popcon)".

Catatan
Angka-angka popcon tidak boleh dianggap sebagai ukuran mutlak tentang pentingnya paket. Ada banyak faktor yang dapat mencondongkan statistik. Misalnya, beberapa sistem yang berpartisipasi popcon mungkin mengait direktori seperti "`/usr/bin`" dengan opsi "`noatime`" untuk peningkatan kinerja sistem dan secara efektif menonaktifkan "pemberian suara" dari sistem tersebut.

Ukuran paket

Data ukuran paket juga disajikan sebagai ukuran objektif untuk setiap paket. Hal ini didasarkan pada "`Installed-Size:`" yang dilaporkan oleh perintah "`apt-cache show`" atau "`aptitude show`" (saat ini pada arsitektur `amd64` untuk rilis `unstable`). Ukuran yang dilaporkan dalam KiB (Kibibyte = unit untuk 1024 byte).

Catatan
Sebuah paket dengan ukuran paket numerik kecil mungkin menunjukkan bahwa paket di rilis `unstable` adalah sebuah paket dummy yang memasang paket-paket lain dengan isi signifikan dari ketergantungan. Paket dummy memungkinkan transisi yang mulus atau pemecahan paket.

Catatan
Ukuran paket yang diikuti oleh "(*)" menunjukkan bahwa paket di rilis `unstable` hilang dan ukuran paket untuk rilis `experimental` digunakan sebagai gantinya.

Laporan bug pada dokumen ini

Harap laporkan bug pada paket `debian-reference` menggunakan reportbug(1) jika Anda menemukan masalah di dokumen ini. Harap sertakan saran koreksi memakai "`diff -u`" ke versi teks polos atau sumber.

Pengingat untuk para pengguna baru

Berikut adalah beberapa pengingat untuk para pengguna baru:

- Membuat cadangan data Anda
  - Lihat Bagian 10.2.
- Amankan kata sandi dan kunci keamanan Anda
- KISS (keep it simple stupid)
  - Jangan merekayasa sistem Anda secara berlebihan
- Membaca berkas log Anda
  - Kesalahan **PERTAMA** adalah yang penting
- RTFM (read the fine manual, baca manualnya)
- Cari di Internet sebelum bertanya
- Jangan menjadi root ketika Anda tidak perlu
- Jangan mengacau sistem manajemen paket
- Jangan ketikkan apapun yang tidak Anda pahami
- Jangan mengubah izin berkas (sebelum peninjauan keamanan secara penuh)
- Jangan tinggalkan shell root Anda sampai Anda **MENGUJI** perubahan Anda
- Selalu miliki media boot alternatif (USB flash drive, CD-ROM, …)

Beberapa kutipan untuk pengguna baru

Berikut adalah beberapa kutipan menarik dari milis Debian yang dapat membantu mencerahkan pengguna baru.

- ”Ini adalah Unix. Ini memberi Anda cukup tali untuk menggantung diri Anda sendiri.” ---Miquel van Smoorenburg `<miquels at cistron.nl>`
- ”Unix itu bersahabat... Hanya saja dia selektif tentang siapa yang menjadi temannya. ” ---Tollef kabut Heen `<tollef at add.no>`

Wikipedia memiliki artikel ”Filosofi Unix” yang mencantumkan kutipan-kutipan menarik.

Bab 1

Tutorial GNU/Linux

Saya rasa belajar sistem komputer seperti belajar bahasa asing baru. Meskipun buku dan dokumentasi tutorial sangat membantu, Anda harus mempraktikkannya sendiri. Untuk membantu Anda memulai dengan lancar, saya menguraikan beberapa hal mendasar.

Desain kuat dari Debian GNU/Linux berasal dari sistem operasi Unix, yaitu, suatu sistem operasi yang multiuser, multitasking. Anda harus belajar untuk mengambil keuntungan dari kekuatan fitur-fitur ini dan kemiripan antara Unix dan GNU/Linux.

Jangan menghindar dari teks yang berorientasi Unix dan jangan hanya mengandalkan teks GNU/Linux, karena ini merampas banyak informasi berguna.

Catatan
Jika Anda telah menggunakan salah satu sistem mirip Unix untuk sementara waktu dengan baris perintah, Anda mungkin tahu segala sesuatu yang saya jelaskan di sini. Harap gunakan ini sebagai uji realita dan penyegaran.

1.1 Dasar-dasar konsol

1.1.1 Prompt shell

Saat memulai sistem, Anda disajikan layar login berbasis karakter bila Anda tidak memasang lingkungan GUI seperti misalnya sistem desktop GNOME atau KDE. Misalnya nama host Anda adalah `foo`, sapaan login tampak sebagai berikut.

Bila Anda memasang suatu lingkungan GUI , maka Anda masih dapat menuju ke sapaan login berbasis karakter dengan Ctrl-Alt-F3, dan Anda dapat kembali ke lingkungan GUI melalui Ctrl-Alt-F2 (lihat Bagian 1.1.6 di bawah untuk lebih lanjut).

```
foo login:
```

Pada sapaan login, Anda mengetikkan nama pengguna Anda, mis. `penguin`, dan menekan tombol Enter, lalu ketikkan kata sandi Anda dan tekan tombol Enter lagi.

Catatan
Mengikuti tradisi Unix, nama pengguna dan kata sandi sistem Debian membedakan huruf besar kecil. Nama pengguna biasanya dipilih hanya dari huruf kecil. Akun pengguna pertama biasanya dibuat saat instalasi. Akun pengguna tambahan dapat dibuat dengan adduser(8) oleh root.

Sistem mengawali dengan pesan sapaan yang disimpan dalam ”/etc/motd” (Message Of The Day, Pesan Hari Ini) dan menyajikan suatu sapaan perintah.

```
Debian GNU/Linux 12 foo tty3

foo login: penguin
Password:

Linux foo 6.5.0-0.deb12.4-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.5.10-1~bpo12+1 (2023-11-23) ↩
     x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

Last login: Wed Dec 20 09:39:00 JST 2023 on tty3
foo:~$
```

Sekarang Anda berada di shell. Shell menafsirkan perintah-perintah Anda.

1.1.2 Prompt shell di bawah GUI

Jika Anda menginstall lingkungan GUI selama instalasi, Anda akan disajikan dengan layar login grafis saat memulai sistem Anda. Anda mengetikkan nama pengguna dan kata sandi Anda untuk log masuk ke akun pengguna biasa. Gunakan tab untuk menavigasi antara nama pengguna dan kata sandi, atau gunakan klik primer tetikus.

Anda bisa mendapatkan prompt shell di bawah lingkungan GUI dengan memulai program `x-terminal-emulator` seperti gnome-terminal(1), rxvt(1), atau xterm(1). Di bawah lingkungan Desktop GNOME, menekan tombol SUPER (tombol Windows) dan mengetikkan ”terminal” ke prompt pencarian akan melakukan itu.

Di bawah beberapa sistem Desktop lain (seperti `fluxbox`), mungkin ada titik awal yang jelas untuk menu. Jika ini terjadi, cobalah mengklik (kanan) latar belakang layar desktop dan berharap untuk munculnya suatu menu.

1.1.3 Akun root

Akun root juga disebut superuser atau pengguna istimewa. Dari akun ini, Anda dapat melakukan tugas-tugas administrasi sistem berikut.

- Baca, tulis, dan hapus berkas apapun di sistem terlepas dari hak akses berkas mereka
- Setel kepemilikan berkas dan hak akses berkas apapun pada sistem
- Tetapkan kata sandi pengguna non-istimewa di sistem
- Login ke akun manapun tanpa kata sandi mereka

Kekuatan akun root tak terbatas ini mengharuskan Anda untuk mempertimbangkan dan bertanggung jawab saat menggunakannya.



Awas
Jangan pernah berbagi kata sandi root dengan orang lain.

Catatan
Izin dalam sebuah berkas (termasuk perangkat keras seperti dll. CD-ROM yang sekedar suatu berkas lain untuk sistem Debian) dapat membuatnya tidak dapat digunakan atau tidak dapat diakses oleh pengguna non-root. Meskipun penggunaan akun root adalah cara cepat untuk menguji situasi semacam ini, resolusi harus dilakukan melalui pengaturan hak akses berkas dan keanggotaan grup pengguna yang tepat (Lihat Bagian 1.2.3).

1.1.4 Prompt shell root

Berikut adalah beberapa metode dasar untuk mendapatkan prompt shell root dengan menggunakan kata sandi root.

- Ketikkan `root` pada prompt login mode teks.
- Ketik "`su -l`" dari sebarang prompt shell pengguna.
  - Ini tidak mempertahankan lingkungan pengguna saat ini.
- Ketik "`su`" dari sebarang prompt shell pengguna.
  - Ini mempertahankan sebagian lingkungan pengguna saat ini.

1.1.5 Alat administrasi sistem GUI

Ketika menu desktop Anda tidak memulai alat administrasi sistem GUI secara otomatis dengan hak istimewa yang sesuai, Anda dapat memulai mereka dari prompt shell root dari emulator terminal, seperti gnome-terminal(1), rxvt(1), atau xterm(1). Lihat Bagian 1.1.4 dan Bagian 7.9.



Awas
Jangan pernah memulai pengelola tampilan/sesi GUI di bawah akun root dengan mengetikkan `root` ke prompt manajer tampilan seperti gdm3(1).
Jangan pernah menjalankan program GUI remote tidak terpercaya di bawah X Window ketika informasi penting ditampilkan karena itu dapat menguping X layar Anda.

1.1.6 Konsol virtual

Dalam sistem Debian default, ada enam konsol mode teks mirip VT100 yang dapat ditukar, tersedia untuk memulai shell perintah langsung pada host Linux. Kecuali Anda berada di lingkungan GUI, Anda dapat beralih antara konsol virtual dengan menekan `tombol Alt kiri` dan salah satu tombol `F1`-`F6` secara bersamaan. Setiap konsol mode teks memungkinkan login yang independen ke akun dan menawarkan lingkungan multiuser. Lingkungan multiuser ini adalah fitur Unix yang hebat, dan sangat adiktif.

Jika Anda berada dalam lingkungan GUI, Anda mendapatkan akses ke konsol mode teks 3 dengan menekan tombol `Ctrl-Alt-F3`, yaitu `tombol Ctrl kiri`, `tombol Alt kiri`, dan `tombol F3` ditekan bersamaan. Anda dapat bisa kembali ke lingkungan GUI, biasanya berjalan pada konsol virtual 2, dengan menekan `Alt-F2`.

Anda juga dapat berpindah ke konsol virtual lain, mis. ke konsol 3, dari baris perintah.

```
# chvt 3
```

1.1.7 Cara meninggalkan command prompt

Anda mengetik `Ctrl-D`, yaitu `tombol Ctrl kiri` dan `tombol d` ditekan bersama-sama, pada prompt perintah untuk menutup kegiatan shell. Jika Anda berada pada konsol mode teks, Anda kembali ke prompt login dengan ini. Meskipun karakter kontrol ini disebut sebagai ”kontrol D” dengan huruf besar, Anda tidak perlu menekan tombol Shift. Ekspresi singkat, `^D`, juga digunakan untuk `Ctrl-D`. Sebagai alternatif, Anda dapat mengetikkan ”exit”.

Jika Anda di x-terminal-emulator(1), Anda bisa menutup jendela `x-terminal-emulator` dengan ini.

1.1.8 Bagaimana mematikan sistem

Sama seperti OS moden lain manapun dimana operasi berkas melibatkan penyinggahan data dalam memori untuk peningkatan kinerja, sistem Debian memerlukan prosedur shutdown yang tepat sebelum daya aman dimatikan. Ini adalah untuk mempertahankan integritas berkas, dengan memaksa semua perubahan dalam memori harus ditulis ke disk. Jika perangkat lunak kendali daya tersedia, prosedur shutdown secara otomatis mematikan daya sistem. (Jika tidak, Anda mungkin harus menekan tombol power selama beberapa detik setelah prosedur shutdown.)

Anda bisa mematikan sistem di bawah mode multi user normal dari command line.

```
# shutdown -h now
```

Anda bisa mematikan sistem di bawah mode single-user dari commandline.

```
# poweroff -i -f
```

Lihat Bagian 6.3.8.

1.1.9 Memulihkan suatu konsol yang waras

Ketika layar mengamuk setelah melakukan beberapa hal lucu seperti ”`cat` *`suatu-berkas-biner`*”, ketikkan ”`reset`” pada prompt perintah. Anda mungkin tidak dapat melihat perintah dipantulkan saat Anda mengetik. Anda dapat juga memerintahkan ”`clear`” untuk membersihkan layar.

1.1.10 Saran paket tambahan untuk newbie

Meskipun instalasi minimal sistem Debian tanpa task lingkungan desktop apa pun telah menyediakan fungsionalitas dasar Unix, ide yang baik untuk menginstall beberapa perintah tambahan dan paket terminal karakter berbasis curses seperti `mc` dan `vim` dengan apt-get(8) bagi pemula untuk mengawali dengan yang berikut ini.

```
# apt-get update
 ...
# apt-get install mc vim sudo aptitude
 ...
```

Jika Anda sudah menginstall paket ini, tidak ada paket baru yang diinstal.

Mungkin ada baiknya membaca beberapa dokumentasi informatif.

Anda dapat menginstall beberapa paket berikut ini.

```
# apt-get install package_name
```

paket	popcon	ukuran	deskripsi
mc	V:40, I:176	1590	Manajer berkas layar penuh mode teks
sudo	V:761, I:869	6814	Sebuah program untuk mengizinkan hak istimewa root terbatas kepada pengguna
vim	V:83, I:338	4164	Editor teks Unix Vi IMproved, editor teks pemrogram (versi standar)
vim-tiny	V:59, I:980	1867	Editor teks Unix, Vi IMproved, editor teks programmer (versi ringkas)
emacs-nox	V:3, I:12	46564	Proyek GNU Emacs, penyunting teks yang dapat diperluas berbasis Lisp
w3m	V:10, I:129	2853	Peramban WWW mode teks
gpm	V:8.3, I:9.0	524	Potong dan tempel gaya Unix pada konsol mode teks (daemon)

Tabel 1.1: Daftar paket program mode teks yang menarik

paket	popcon	ukuran	deskripsi
doc-debian	I:886	185	Dokumentasi Proyek Debian, (FAQ Debian) dan dokumen lainnya
debian-policy	I:6.8	5147	Manual Kebijakan Debian dan dokumen terkait
developers-reference	V:0.2, I:2.2	2657	Panduan dan informasi untuk pengembang Debian
debmake-doc	I:0.33	11803	Panduan untuk Pengelola Debian
debian-history	I:0.50	6513	Sejarah Proyek Debian
debian-faq	I:885	791	FAQ Debian

Tabel 1.2: Daftar paket dokumentasi informatif

1.1.11 Akun pengguna tambahan

Jika Anda tidak ingin menggunakan akun pengguna utama untuk kegiatan pelatihan berikut, Anda dapat membuat akun pengguna pelatihan, misalnya `fish` dengan cara berikut.

```
# adduser fish
```

Jawablah semua pertanyaan.

Ini membuat akun baru bernama `fish`. Setelah latihan Anda, Anda dapat menghapus akun pengguna ini dan direktori home-nya dengan cara berikut.

```
# deluser --remove-home fish
```

Pada sistem Debian terspesialisasi dan non-Debian, kegiatan di atas perlu menggunakan utilitas level bawah useradd(8) dan userdel(8).

1.1.12 konfigurasi sudo

Untuk workstation pengguna tunggal biasa seperti sistem Debian desktop pada PC laptop, biasanya menggunakan konfigurasi sederhana sudo(8) sebagai berikut untuk membiarkan pengguna yang tidak memiliki hak istimewa, mis. `penguin`, untuk mendapatkan hak administratif hanya dengan kata sandi penggunanya tetapi tanpa kata sandi root.

```
# echo "penguin  ALL=(ALL) ALL" >> /etc/sudoers
```

Sebagai alternatif, juga biasa dilakukan sebagai berikut untuk membiarkan pengguna biasa, mis. `penguin`, untuk mendapatkan hak administratif tanpa kata sandi.

```
# echo "penguin  ALL=(ALL) NOPASSWD:ALL" >> /etc/sudoers
```

Trik ini hanya boleh digunakan untuk workstation pengguna tunggal yang Anda kelola dan di mana Anda adalah satu-satunya pengguna.



Awas
Jangan membuat akun pengguna biasa di workstation multiuser seperti ini karena akan sangat buruk bagi keamanan sistem.



Perhatian
Kata sandi dan akun `penguin` dalam contoh di atas membutuhkan perlindungan seperti kata sandi root dan akun root.
Hak administratif dalam konteks ini adalah milik seseorang yang berwenang untuk melakukan tugas administrasi sistem pada workstation. Jangan pernah memberikan hak semacam itu kepada manajer di departemen Admin di perusahaan Anda atau bos Anda, kecuali mereka memiliki wewenang dan kemampuan.

Catatan
Untuk memberikan hak akses ke perangkat terbatas dan berkas terbatas, Anda mesti mempertimbangkan untuk menggunakan **group** untuk memberikan akses terbatas daripada menggunakan hak istimewa `root` melalui sudo(8) .
Dengan konfigurasi yang lebih bijaksana dan hati-hati, sudo(8) dapat memberikan hak administratif terbatas kepada pengguna lain pada sistem bersama tanpa membagikan kata sandi root. Ini dapat membantu akuntabilitas dengan host dengan banyak administrator sehingga Anda dapat mengetahui siapa yang melakukan apa. Di sisi lain, Anda mungkin tidak ingin orang lain memiliki hak istimewa seperti itu.

1.1.13 Waktu bermain

Sekarang Anda siap untuk bermain dengan sistem Debian tanpa risiko selama Anda menggunakan akun pengguna yang tidak memiliki hak istimewa.

Ini karena sistem Debian, bahkan setelah instalasi baku, dikonfigurasi dengan izin berkas yang tepat yang mencegah pengguna yang tidak memiliki hak istimewa merusak sistem. Tentu saja, mungkin masih ada beberapa lubang yang dapat dieksploitasi tetapi mereka yang khawatir tentang masalah ini tidak boleh membaca bagian ini tetapi harus membaca Manual Mengamankan Debian.

Kami mempelajari sistem Debian sebagai sebuah sistem mirip Unix dengan yang berikut ini.

- Bagian 1.2 (konsep dasar)
- Bagian 1.3 (metode survival)
- Bagian 1.4 (metode dasar)
- Bagian 1.5 (mekanisme shell)
- Bagian 1.6 (metode pengolahan teks)

1.2 Sistem berkas mirip Unix

Di GNU/Linux dan sistem operasi mirip Unix lainnya, berkas diatur ke dalam direktori. Semua berkas dan direktori diatur dalam satu pohon besar yang berakar pada "`/`". Disebut pohon karena jika Anda menggambar sistem berkas, itu terlihat seperti pohon tetapi terbalik.

Berkas dan direktori ini dapat tersebar di beberapa perangkat. mount(8) berfungsi untuk mencantolkan sistem berkas yang ditemukan pada beberapa perangkat ke pohon berkas besar. Sebaliknya, umount(8) melepaskannya lagi. Pada kernel Linux terbaru, mount(8) dengan beberapa opsi dapat mengikat bagian dari pohon berkas di tempat lain atau dapat mengait sistem berkas sebagai shared, private, slave, atau unbindable. Opsi pemasangan yang didukung untuk setiap sistem berkas tersedia di "`/usr/share/doc/linux-doc-*/Documentation/filesystems/`".

Direktori pada sistem Unix disebut **folder** pada beberapa sistem lain. Harap perhatikan juga bahwa tidak ada konsep untuk **drive** seperti "`A:`" pada sistem Unix mana pun. Ada satu sistem berkas, dan semuanya disertakan. Ini adalah keuntungan besar dibandingkan dengan Windows.

1.2.1 Dasar-dasar berkas Unix

Berikut adalah beberapa dasar berkas Unix.

- Nama berkas **peka huruf besar/kecil**. Artinya, "`MYFILE`" dan "`MyFile`" adalah berkas yang berbeda.
- **Direktori root** berarti akar dari sistem berkas yang disebut sebagai "`/`". Jangan bingung dengan direktori home untuk pengguna root: "`/root`".
- Setiap direktori memiliki nama yang dapat berisi huruf atau simbol **kecuali "`/`"**. Direktori root adalah pengecualian; namanya adalah "`/`" (diucapkan "slash" atau "direktori root") dan tidak dapat diganti namanya.
- Setiap berkas atau direktori ditunjuk oleh **fully-qualified filename**, **nama berkas absolut**, atau **path**, memberikan urutan direktori yang harus dilalui untuk mencapainya. Ketiga istilah tersebut sinonim.
- Semua **fully-qualified filenames** dimulai dengan direktori "`/`", dan ada "`/`" di antara setiap direktori atau berkas dalam nama berkas. "`/`" pertama adalah direktori tingkat puncak, dan subdirektori terpisah "`/`" lainnya, sampai kita mencapai entri terakhir yang merupakan nama berkas sebenarnya. Kata-kata yang digunakan di sini bisa membingungkan. Ambil **fully-qualified filenames** berikut sebagai contoh: "`/usr/share/keytables/us.map.gz`". Namun, orang juga merujuk ke nama dasarnya "`us.map.gz`" saja sebagai nama berkas.
- Direktori root memiliki sejumlah cabang, seperti "`/etc/`" dan "`/usr/`". Subdirektori ini pada gilirannya bercabang menjadi lebih banyak subdirektori, seperti "`/etc/systemd/`" dan "`/usr/local/`". Semua itu dilihat secara kolektif disebut sebagai **pohon direktori**. Anda dapat menganggap nama berkas absolut sebagai rute dari dasar pohon ("`/`") ke akhir beberapa cabang (berkas). Anda juga mendengar orang berbicara tentang pohon direktori seolah-olah itu adalah pohon **keluarga** yang mencakup semua keturunan langsung dari satu figur yang disebut direktori root ("`/`"): jadi subdirektori memiliki **induk**, dan sebuah path menunjukkan keturunan lengkap dari sebuah berkas. Ada juga jalur relatif yang dimulai di suatu tempat selain direktori root. Anda harus ingat bahwa direktori "`../`" merujuk ke direktori induk. Terminologi ini juga berlaku untuk direktori lain seperti struktur, seperti struktur data hierarkis.
- Tidak ada komponen nama path direktori khusus yang sesuai dengan perangkat fisik, seperti hard disk Anda. Ini berbeda dari RT-11, CP/M, OpenVMS, MS-DOS, AmigaOS, dan Microsoft Windows, dengan path berisi nama perangkat seperti "`C:\`". (Namun, entri direktori memang ada yang merujuk ke perangkat fisik sebagai bagian dari sistem berkas normal. Lihat Bagian 1.2.2.)

Catatan
Meskipun Anda **dapat** menggunakan hampir semua huruf atau simbol dalam nama berkas, dalam praktiknya adalah ide yang buruk untuk melakukannya. Sebaiknya hindari karakter yang sering memiliki arti khusus pada baris perintah, termasuk spasi, tab, baris baru, dan karakter khusus lainnya: `{ } ( ) [ ] ' ` " \ / > < | ; ! # & ^ * % @ $`. Jika Anda ingin memisahkan kata dalam sebuah nama, pilihan yang baik adalah titik, tanda hubung, dan garis bawah. Anda juga dapat menggunakan huruf besar untuk setiap kata, "`SepertiIni`". Pengguna Linux yang berpengalaman cenderung menghindari spasi dalam nama berkas.

Catatan
Kata "root" dapat berarti "pengguna root" atau "direktori root". Konteks penggunaannya harus membuatnya jelas.

Catatan
Kata **path** digunakan tidak hanya untuk **nama berkas lengkap** seperti di atas tetapi juga untuk **path pencarian perintah**. Makna yang dimaksud biasanya jelas dari konteksnya.

Praktik terbaik terperinci untuk hierarki berkas dijelaskan dalam Standar Hierarki Sistem Berkas/Filesystem Hierarchy Standard (”`/usr/share/doc/debian-policy/fhs/fhs-2.3.txt.gz`” dan hier(7)). Anda harus mengingat fakta-fakta berikut di awal.

direktori	penggunaan direktori
`/`	direktori root
`/etc/`	berkas konfigurasi seluruh sistem
`/var/log/`	berkas log sistem
`/home/`	semua direktori rumah bagi semua pengguna tanpa hak istimewa

Tabel 1.3: Daftar penggunaan direktori kunci

1.2.2 Internal sistem berkas

Mengikuti **tradisi Unix**, sistem GNU/Linux Debian menyediakan sistem berkas di mana data fisik pada hard disk dan perangkat penyimpanan lainnya berada, dan interaksi dengan perangkat keras seperti layar konsol dan konsol serial jarak jauh diwakili secara terpadu di bawah ”`/dev/`”.

Setiap berkas, direktori, pipa bernama (cara dua program dapat berbagi data), atau perangkat fisik pada sistem Debian GNU/Linux memiliki struktur data yang disebut inode yang menjelaskan atribut terkait seperti pengguna yang memilikinya (pemilik), grup yang diikutinya, waktu terakhir diakses, dll. Gagasan untuk mewakili hampir semua hal dalam sistem berkas adalah inovasi Unix, dan kernel Linux modern telah mengembangkan gagasan ini lebih jauh. Sekarang, bahkan informasi tentang proses yang berjalan di komputer dapat ditemukan di sistem berkas.

Representasi abstrak dan kesatuan entitas fisik dan proses internal ini sangat kuat karena ini memungkinkan kita untuk menggunakan perintah yang sama untuk jenis operasi yang sama pada banyak perangkat yang sama sekali berbeda. Bahkan dimungkinkan untuk mengubah cara kerja kernel dengan menulis data ke berkas khusus yang terkait dengan proses yang berjalan.

Tip
Jika Anda perlu mengidentifikasi korespondensi antara pohon berkas dan entitas fisik, jalankan mount(8) tanpa argumen.

1.2.3 Hak akses sistem berkas

Izin sistem berkas dari sistem seperti Unix ditentukan untuk tiga kategori pengguna yang terpengaruh.

- **pengguna** yang memiliki berkas (**u**)
- Pengguna lain di **grup** tempat berkas tersebut termasuk (**g**)
- Semua pengguna **lainnya** (**o**) juga disebut sebagai ”dunia” dan ”semua orang”

Untuk berkas tersebut, setiap izin yang sesuai memungkinkan tindakan berikut.

- Izin **baca** (**r**) memungkinkan pemilik untuk memeriksa konten berkas.

- Izin **tulis** (**w**) memungkinkan pemilik untuk memodifikasi berkas.
- Izin **eksekusi** (**x**) memungkinkan pemilik untuk menjalankan berkas sebagai perintah.

Untuk direktori, setiap izin yang sesuai memungkinkan tindakan berikut.

- Izin **baca** (**r**) memungkinkan pemilik untuk membuat daftar isi direktori.
- Izin **tulis** (**w**) memungkinkan pemilik untuk menambah atau menghapus berkas dalam direktori.
- Izin **eksekusi** (**x**) memungkinkan pemilik untuk mengakses berkas dalam direktori.

Di sini, izin **eksekusi** pada direktori berarti tidak hanya mengizinkan pembacaan berkas di direktori itu tetapi juga untuk memungkinkan melihat atribut mereka, seperti ukuran dan waktu modifikasi.

ls(1) digunakan untuk menampilkan informasi izin (dan lebih banyak lagi) untuk berkas dan direktori. Ketika dipanggil dengan opsi "`-l`", ini menampilkan informasi berikut dalam urutan yang diberikan.

- **Tipe berkas** (karakter pertama)
- **Izin** akses berkas (sembilan karakter, masing-masing terdiri dari tiga karakter untuk pengguna, grup, dan lainnya dalam urutan ini)
- **Banyaknya hard link** ke berkas
- Nama dari **pengguna** yang memiliki berkas tersebut
- Nama dari **grup** pemilik berkas tersebut
- **Ukuran** berkas dalam karakter (byte)
- **Tanggal dan waktu** berkas (mtime)
- **Nama** berkas

karakter	**arti**
`-`	berkas biasa
d	direktori
`l`	symlink
c	simpul perangkat karakter
b	simpul perangkat blok
p	pipa bernama
s	soket

Tabel 1.4: Daftar karakter pertama dari keluaran "`ls -l`"

chown(1) digunakan dari akun root untuk mengubah pemilik berkas. chgrp(1) digunakan dari pemilik berkas atau akun root untuk mengubah grup berkas. chmod(1) digunakan dari pemilik berkas atau akun root untuk mengubah izin akses berkas dan direktori. Sintaks dasar untuk memanipulasi berkas `foo` adalah sebagai berikut.

```
# chown newowner foo
# chgrp newgroup foo
# chmod  [ugoa][+-=][rwxXst][,...] foo
```

Misalnya, Anda dapat membuat pohon direktori agar dimiliki oleh pengguna `foo` dan dipakai bersama oleh grup `bar` dengan cara berikut ini.

```
# cd /some/location/
# chown -R foo:bar .
# chmod -R ug+rwX,o=rX .
```

Ada tiga bit izin khusus lagi.

- Bit **set user ID** (**s** atau **S** alih-alih **x** pengguna)
- Bit **set group ID** (**s** atau **S** alih-alih **x** grup)
- Bit **sticky** (**t** atau **T** alih-alih **x** lainnya)

Di sini keluaran dari "`ls -l`" untuk bit ini adalah **dalam kapital** jika bit eksekusi yang disembunyikan oleh keluaran ini **tak ditata**.

Menyetel **set ID pengguna** pada berkas executable memungkinkan pengguna untuk menjalankan berkas executable dengan ID pemilik berkas (misalnya **root**). Demikian pula, pengaturan **set ID grup** pada berkas executable memungkinkan pengguna untuk menjalankan berkas executable dengan ID grup berkas (misalnya **root**). Karena pengaturan ini dapat menyebabkan risiko keamanan, mengaktifkannya memerlukan kehati-hatian ekstra.

Mengatur **set ID grup** pada direktori akan mengaktifkan skema pembuatan berkas seperti BSD di mana semua berkas yang dibuat dalam direktori menjadi milik **grup** dari direktori.

Menyetel **sticky bit** pada direktori mencegah berkas dalam direktori dihapus oleh pengguna yang bukan pemilik berkas. Untuk mengamankan konten berkas di direktori yang dapat ditulis dunia seperti "`/tmp`" atau dalam direktori yang dapat ditulis grup, seseorang tidak hanya harus mengatur ulang izin **tulis** untuk berkas tersebut, tetapi juga mengatur **sticky bit** pada direktori. Jika tidak, berkas dapat dihapus dan berkas baru dapat dibuat dengan nama yang sama oleh setiap pengguna yang memiliki akses tulis ke direktori.

Berikut adalah beberapa contoh menarik dari izin berkas.

```
$ ls -l /etc/passwd /etc/shadow /dev/ppp /usr/sbin/exim4
crw------T 1 root root   108, 0 Oct 16 20:57 /dev/ppp
-rw-r--r-- 1 root root     2761 Aug 30 10:38 /etc/passwd
-rw-r----- 1 root shadow   1695 Aug 30 10:38 /etc/shadow
-rwsr-xr-x 1 root root   973824 Sep 23 20:04 /usr/sbin/exim4
$ ls -ld /tmp /var/tmp /usr/local /var/mail /usr/src
drwxrwxrwt 14 root root  20480 Oct 16 21:25 /tmp
drwxrwsr-x 10 root staff  4096 Sep 29 22:50 /usr/local
drwxr-xr-x 10 root root   4096 Oct 11 00:28 /usr/src
drwxrwsr-x  2 root mail   4096 Oct 15 21:40 /var/mail
drwxrwxrwt  3 root root   4096 Oct 16 21:20 /var/tmp
```

Ada mode numerik alternatif untuk menjelaskan izin berkas dengan chmod(1). Mode numerik ini menggunakan 3 hingga 4 digit angka oktal lebar (radix = 8).

nomor	**arti**
digit opsional pertama	jumlahdari **set ID pengguna** (=4), **set ID grup** (=2), dan **sticky bit** (=1)
angka ke-2	jumlah dari izin **baca** (=4), **tulis** (=2), dan **eksekusi** (=1) untuk **pengguna**
angka ke-3	begitu juga untuk **grup**
angka ke-4	begitu juga untuk **lainnya**

Tabel 1.5: Mode numerik untuk izin berkas dalam perintah chmod(1)

Ini terdengar rumit tetapi sebenarnya cukup sederhana. Jika Anda melihat beberapa (2-10) kolom pertama dari keluaran perintah "`ls -l`" dan membacanya sebagai representasi biner (basis=2) dari izin berkas ("-" menjadi "0" dan "rwx" menjadi "1"), 3 digit terakhir dari nilai mode numerik harus masuk akal sebagai representasi oktal (basis=8) dari izin berkas bagi Anda.

Misalnya, coba yang berikut ini

```
$ touch foo bar
$ chmod u=rw,go=r foo
$ chmod 644 bar
```

```
$ ls -l foo bar
-rw-r--r-- 1 penguin penguin 0 Oct 16 21:39 bar
-rw-r--r-- 1 penguin penguin 0 Oct 16 21:35 foo
```

Tip
Jika Anda perlu mengakses informasi yang ditampilkan oleh "`ls -l`" dalam skrip shell, Anda harus menggunakan perintah terkait seperti test(1), `stat(1)`, dan readlink(1). Bawaan shell seperti "`[`" atau "`test`" dapat digunakan juga.

1.2.4 Kontrol izin untuk berkas yang baru dibuat: umask

Izin apa yang diterapkan ke berkas atau direktori yang baru dibuat dibatasi oleh perintah bawaan shell `umask`. Lihat dash(1), bash(1), dan builtins(7).

```
(file permissions) = (requested file permissions) & ~(umask value)
```

umask	izin berkas dibuat	izin direktori dibuat	pengunaan
`0022`	`-rw-r--r--`	`-rwxr-xr-x`	hanya dapat ditulis oleh pengguna
`0002`	`-rw-rw-r--`	`-rwxrwxr-x`	dapat ditulis oleh grup

Tabel 1.6: Contoh-contoh nilai **umask**

Sistem Debian menggunakan skema grup privat pengguna (user private group/UPG) sebagai bakunya. UPG dibuat setiap kali pengguna baru ditambahkan ke sistem. UPG memiliki nama yang sama dengan pengguna yang dibuat dan pengguna itu adalah satu-satunya anggota UPG. Skema UPG membuatnya aman untuk mengatur umask ke `0002` karena setiap pengguna memiliki grup pribadi mereka sendiri. (Dalam beberapa varian Unix, sangat umum untuk mengatur semua pengguna normal yang termasuk dalam satu grup **`users`** dan merupakan ide yang baik untuk mengatur umask ke `0022` untuk keamanan dalam kasus seperti itu.)

Tip
Aktifkan UPG dengan meletakkan "`umask 002`" di berkas `~/.bashrc`.

1.2.5 Izin untuk grup pengguna (grup)



Awas
Pastikan untuk menyimpan perubahan yang belum tersimpan sebelum melakukan reboot atau tindakan serupa.

Anda dapat menambahkan pengguna `penguin` ke grup `bird` dalam dua langkah:

- Ubah konfigurasi grup menggunakan salah satu dari yang berikut:
  - Jalankan "`sudo usermod -aG bird penguin`".
  - Jalankan "`sudo adduser penguin bird`". (hanya pada sistem Debian umum)
  - Jalankan "`sudo vigr`" bagi `/etc/group` dan "`sudo vigr -s`" bagi `/etc/gshadow` untuk mengimbuhkan `penguin` dalam baris bagi `bird`.

- Terapkan konfigurasi menggunakan salah satu yang berikut:
  - Boot ulang cold (matikan daya lalu nyalakan lagi) dan log masuk. (Pilihan terbaik)
  - Log keluar melalui menu GUI dan log masuk. (Ini mungkin tidak berjalan di bawah lingkungan Desktop modern.)

Anda dapat menghapus pengguna `penguin` dari grup `bird` dalam dua langkah:

- Ubah konfigurasi grup menggunakan salah satu dari yang berikut:
  - Jalankan "`sudo usermod -rG bird penguin`".
  - Jalankan "`sudo deluser penguin bird`". (hanya pada sistem Debian umum)
  - Jalankan "`sudo vigr`" bagi `/etc/group` dan "`sudo vigr -s`" bagi `/etc/gshadow` untuk membuang `penguin` dalam baris bagi `bird`.
- Terapkan konfigurasi menggunakan salah satu yang berikut:
  - Boot ulang cold (matikan daya lalu nyalakan lagi) dan log masuk. (Pilihan terbaik)
  - Jalankan "`kill -TERM -1`" dan lakukan beberapa tindakan perbaikan seperti "`systemctl restart NetworkMana`
  - Log keluar melalui menu GUI bukanlah suatu opsi bagi Desktop Gnome.

Sebarang upaya boot ulang warm adalah pengganti yang rentan dari boot ulang cold di bawah sistem desktop modern.

Catatan
Atau, Anda dapat secara dinamis menambahkan pengguna ke grup selama proses otentikasi dengan menambahkan baris "`auth optional pam_group.so`" ke "`/etc/pam.d/common-auth`" dan pengaturan "`/etc/security/group.conf`". (Lihat Bab 4.)

Perangkat keras hanyalah jenis berkas lain pada sistem Debian. Jika Anda mengalami masalah dalam mengakses perangkat seperti USB flash drive dan CD-ROM dari akun pengguna, Anda harus menjadikan pengguna tersebut anggota grup yang relevan.

Beberapa grup terkenal yang disediakan sistem mengizinkan anggotanya mengakses berkas dan perangkat tertentu tanpa hak istimewa `root`.

kelompok	**deskripsi untuk berkas dan perangkat yang dapat diakses**
`dialout`	akses penuh dan langsung ke port serial ("`/dev/ttyS[0-3]`")
`dip`	akses terbatas ke port serial untuk koneksi **IP Dialup** ke rekan tepercaya
`cdrom`	Drive CD-ROM, DVD+/-RW
`audio`	perangkat audio
`video`	perangkat video
`scanner`	pemindai
`adm`	log pemantauan sistem
`staff`	beberapa direktori untuk pekerjaan administratif junior: "`/usr/local`", "`/home`"

Tabel 1.7: Daftar grup yang disediakan sistem terkenal untuk akses berkas

---

Tip
Anda harus tergabung dalam grup `dialout` untuk mengkonfigurasi ulang modem, dial ke mana pun, dll. Tetapi jika `root` membuat berkas konfigurasi yang telah ditentukan sebelumnya untuk rekan tepercaya di "`/etc/ppp/peers/`", Anda hanya perlu menjadi bagian dari grup `dip` untuk membuat koneksi **IP Dialup** ke rekan tepercaya tersebut menggunakan perintah pppd(8), pon(1), dan poff(1).

---

kelompok	perintah yang dapat diakses
`sudo`	menjalankan sebarang perintah dengan hak superuser
`lpadmin`	menjalankan perintah untuk menambah, memodifikasi, dan menghapus printer dari database printer

Tabel 1.8: Daftar grup penting yang disediakan untuk eksekusi perintah tertentu

Beberapa grup terkenal yang disediakan sistem mengizinkan anggotanya untuk menjalankan perintah tertentu tanpa hak istimewa `root`.

Untuk daftar lengkap pengguna dan grup yang disediakan sistem, lihat versi terbaru dari dokumen ”Pengguna dan Grup” di ”`/usr/share/doc/base-passwd/users-and-groups.html`” yang disediakan oleh paket `base-passwd`.

Lihat passwd(5), group(5), shadow(5), newgrp(1), vipw(8), vigr(8), dan pam_group(8) untuk perintah manajemen pengguna dan sistem grup.

1.2.6 Stempel waktu

Ada tiga jenis stempel waktu untuk berkas GNU/Linux.

jenis	artinya (definisi Unix historis)
mtime	waktu modifikasi berkas (`ls -l`)
ctime	waktu perubahan status berkas (`ls -lc`)
atime	waktu akses berkas terakhir (`ls -lu`)

Tabel 1.9: Daftar jenis stempel waktu

Catatan
ctime bukanlah waktu pembuatan berkas.

Catatan
Nilai sebenarnya dari **atime** pada sistem GNU/Linux mungkin berbeda dari definisi Unix historis.

- Menimpa berkas mengubah semua atribut **mtime**, **ctime**, dan **atime** dari berkas.
- Mengubah kepemilikan atau izin berkas mengubah atribut **ctime** dan **atime** berkas.
- Membaca berkas mengubah atribut **atime** berkas pada sistem Unix bersejarah.
- Membaca berkas mengubah atribut **atime** berkas pada sistem GNU/Linux jika sistem berkasnya dikait dengan ”`strictatime`”.
- Membaca berkas untuk pertama kalinya atau setelah satu hari mengubah atribut **atime** berkas pada sistem GNU/Linux jika sistem berkasnya dikait dengan ”`relatime`”. (perilaku baku sejak Linux 2.6.30)
- Membaca berkas tidak mengubah atribut **atime** berkas pada sistem GNU/Linux jika sistem berkasnya dikait dengan ”`noatime`”.

Catatan
Opsi kait ”`noatime`” dan ”`relatime`” diperkenalkan untuk meningkatkan kinerja baca sistem berkas di bawah kasus penggunaan normal. Operasi baca berkas sederhana di bawah opsi ”`strictatime`” menyertai operasi tulis yang memakan waktu untuk memperbarui atribut **atime**. Tetapi atribut **atime** jarang digunakan kecuali untuk berkas mbox(5). Lihat mount(8).

Gunakan perintah touch(1) untuk mengubah stempel waktu berkas yang ada.

Untuk stempel waktu, perintah `ls` menghasilkan string terlokalkan di bawah lokal non-Inggris (”`fr_FR.UTF-8`”).

```
$ LANG=C  ls -l foo
-rw-rw-r-- 1 penguin penguin 0 Oct 16 21:35 foo
$ LANG=en_US.UTF-8  ls -l foo
-rw-rw-r-- 1 penguin penguin 0 Oct 16 21:35 foo
$ LANG=fr_FR.UTF-8  ls -l foo
-rw-rw-r-- 1 penguin penguin 0 oct. 16 21:35 foo
```

Tip
Lihat Bagian 9.3.4 untuk menyesuaikan keluaran ”`ls -l`”.

1.2.7 Taut

Ada dua metode untuk menghubungkan berkas ”`foo`” dengan nama berkas yang berbeda ”`bar`”.

- Hard link
  - Nama duplikat untuk berkas yang sudah ada
  - ”`ln foo bar`”
- Tautan simbolik atau symlink
  - Berkas khusus yang menunjuk ke berkas lain berdasarkan nama
  - ”`ln -s foo bar`”

Lihat contoh berikut untuk perubahan cacah tautan dan perbedaan halus dalam hasil perintah `rm`.

```
$ umask 002
$ echo "Original Content" > foo
$ ls -li foo
1449840 -rw-rw-r-- 1 penguin penguin 17 Oct 16 21:42 foo
$ ln foo bar     # hard link
$ ln -s foo baz  # symlink
$ ls -li foo bar baz
1449840 -rw-rw-r-- 2 penguin penguin 17 Oct 16 21:42 bar
1450180 lrwxrwxrwx 1 penguin penguin  3 Oct 16 21:47 baz -> foo
1449840 -rw-rw-r-- 2 penguin penguin 17 Oct 16 21:42 foo
$ rm foo
$ echo "New Content" > foo
$ ls -li foo bar baz
1449840 -rw-rw-r-- 1 penguin penguin 17 Oct 16 21:42 bar
1450180 lrwxrwxrwx 1 penguin penguin  3 Oct 16 21:47 baz -> foo
1450183 -rw-rw-r-- 1 penguin penguin 12 Oct 16 21:48 foo
$ cat bar
Original Content
$ cat baz
New Content
```

Hardlink dapat dibuat dalam sistem berkas yang sama dan memiliki nomor inode yang sama yang diungkap oleh opsi ”`-i`” ls(1).

Symlink selalu memiliki izin akses berkas nominal ”`rwxrwxrwx`”, seperti yang ditunjukkan pada contoh di atas, dengan izin akses efektif ditentukan oleh izin berkas yang diacunya.



Perhatian
Hal ini umumnya ide yang baik untuk tidak membuat taut simbolis yang rumit atau hardlink sama sekali kecuali Anda memiliki alasan yang sangat baik. Hal ini dapat menyebabkan mimpi buruk di mana kombinasi logis dari taut simbolis menghasilkan loop dalam sistem berkas.

Catatan
Umumnya lebih baik untuk menggunakan taut simbolis daripada hardlink kecuali Anda memiliki alasan yang baik untuk menggunakan hardlink.

Direktori ”`.`” menaut ke direktori tempat itu muncul, sehingga cacah taut dari setiap direktori baru dimulai pada 2. Direktori ”`..`” menaut ke direktori induk, sehingga cacah taut dari direktori meningkat dengan penambahan subdirektori baru.

Jika Anda baru saja pindah ke Linux dari Windows, segera menjadi jelas seberapa baik dirancang penautan nama berkas Unix, dibandingkan dengan yang setara pada Windows terdekat yaitu ”pintasan”. Karena diimplementasikan dalam sistem berkas, aplikasi tidak dapat melihat perbedaan antara berkas yang ditautkan dan yang asli. Dalam kasus hardlink, benar-benar tidak ada perbedaan.

1.2.8 Pipa bernama (FIFO)

Suatu pipa bernama (named pipe) adalah berkas yang bertindak seperti pipa. Anda memasukkan sesuatu ke dalam berkas, dan itu keluar di ujung yang lain. Jadi itu disebut FIFO, atau First-In-First-Out: hal pertama yang Anda masukkan ke dalam pipa adalah hal pertama yang keluar dari ujung yang lain.

Jika Anda menulis ke pipa bernama, proses yang menulis ke pipa tidak berakhir sampai informasi yang ditulis dibaca dari pipa. Jika Anda membaca dari pipa bernama, proses membaca menunggu sampai tidak ada yang bisa dibaca sebelum mengakhiri. Ukuran pipa selalu nol --- itu tidak menyimpan data, itu hanya menghubungkan dua proses seperti fungsi yang ditawarkan oleh sintaks shell ”`|`”. Namun, karena pipa ini memiliki nama, kedua proses tidak harus berada di baris perintah yang sama atau bahkan dijalankan oleh pengguna yang sama. Pipa adalah inovasi Unix yang sangat berpengaruh.

Misalnya, coba yang berikut ini

```
$ cd; mkfifo mypipe
$ echo "hello" >mypipe & # put into background
[1] 8022
$ ls -l mypipe
prw-rw-r-- 1 penguin penguin 0 Oct 16 21:49 mypipe
$ cat mypipe
hello
[1]+  Done                    echo "hello" >mypipe
$ ls mypipe
mypipe
$ rm mypipe
```

1.2.9 Soket

Soket digunakan secara luas oleh semua komunikasi Internet, basis data, dan sistem operasi itu sendiri. Hal ini mirip dengan pipa bernama (FIFO) dan memungkinkan proses untuk bertukar informasi bahkan antara komputer yang berbeda. Untuk soket, proses tersebut tidak perlu berjalan pada saat yang sama atau berjalan sebagai anak dari proses leluhur yang sama. Ini adalah titik akhir untuk komunikasi antar proses (inter process communication/IPC). Pertukaran informasi dapat terjadi melalui jaringan antara host yang berbeda. Dua yang paling umum adalah soket Internet dan soket domain Unix.

Tip
"`ss -t -u -a`" (dari paket `iproute2`) memberikan gambaran yang sangat berguna dari soket yang terbuka pada sistem tertentu.

1.2.10 Berkas perangkat

Berkas perangkat mengacu pada perangkat fisik atau virtual pada sistem Anda, seperti hard disk, kartu video, layar, atau papan ketik Anda. Contoh perangkat virtual adalah konsol, diwakili oleh "`/dev/console`".

Ada dua jenis berkas perangkat.

- **Perangkat karakter**
  - Mengakses satu karakter pada satu waktu
  - 1 karakter = 1 byte
  - Mis. perangkat papan ketik, port serial, …
- **Perangkat blok**
  - diakses dalam unit yang lebih besar yang disebut blok
  - 1 blok > 1 byte
  - Mis. hard disk, …

Anda dapat membaca dan menulis berkas perangkat, meskipun berkas mungkin berisi data biner yang mungkin tidak dapat dimengerti oleh manusia. Menulis data langsung ke berkas-berkas ini kadang-kadang berguna untuk pemecahan masalah koneksi perangkat keras. Misalnya, Anda dapat mencurahkan berkas teks ke perangkat pencetak "`/dev/lp0`" atau mengirim perintah modem ke port serial yang sesuai "`/dev/ttyS0`". Tapi, kecuali ini dilakukan dengan hati-hati, itu dapat menyebabkan bencana besar. Jadi berhati-hatilah.

Catatan
Untuk akses normal ke pencetak, gunakan lp(1).

Nomor simpul perangkat ditampilkan dengan mengeksekusi ls(1) sebagai berikut.

```
$ ls -l /dev/sda /dev/sr0 /dev/ttyS0 /dev/zero
brw-rw---T  1 root disk     8,  0 Oct 16 20:57 /dev/sda
brw-rw---T+ 1 root cdrom   11,  0 Oct 16 21:53 /dev/sr0
crw-rw---T  1 root dialout  4, 64 Oct 16 20:57 /dev/ttyS0
crw-rw-rw-  1 root root     1,  5 Oct 16 20:57 /dev/zero
```

- "`/dev/sda`" memiliki nomor perangkat mayor 8 dan nomor perangkat minor 0. Ini dapat dibaca/tulis oleh pengguna yang termasuk dalam grup `disk`.
- "`/dev/sr0`" memiliki nomor perangkat mayor 11 dan nomor perangkat minor 0. Ini dapat dibaca/tulis oleh pengguna yang tergabung dalam grup `cdrom`.
- "`/dev/ttyS0`" memiliki nomor perangkat mayor 4 dan nomor perangkat minor 64. Ini dapat dibaca/tulis oleh pengguna yang termasuk dalam grup `dialout`.
- "`/dev/zero`" memiliki nomor perangkat mayor 1 dan perangkat minor nomor 5. Ini dapat dibaca/ditulis oleh siapa saja.

Pada sistem Linux modern, sistem berkas di bawah "`/dev/`" secara otomatis dihuni oleh mekanisme udev(7).

berkas perangkat	aksi	deskripsi respon
`/dev/null`	baca	mengembalikan ”karakter end-of-file (EOF)”
`/dev/null`	tulis	tidak mengembalikan apa-apa (lubang pembuangan data tanpa dasar)
`/dev/zero`	baca	mengembalikan ”karakter `\0` (NUL)” (tidak sama dengan angka nol ASCII)
`/dev/random`	baca	mengembalikan karakter acak dari pembangkit angka acak sejati, memberikan entropi nyata (lambat)
`/dev/urandom`	baca	mengembalikan karakter acak dari pembangkit angka pseudorandom yang aman secara kriptografis
`/dev/full`	tulis	mengembalikan kesalahan disk-full (ENOSPC)

Tabel 1.10: Daftar berkas perangkat khusus

1.2.11 Berkas perangkat khusus

Ada beberapa berkas perangkat khusus.

Ini sering digunakan bersamaan dengan pengalihan shell (lihat Bagian 1.5.8).

1.2.12 procfs dan sysfs

procfs dan sysfs yang dipasang pada ”`/proc`” dan ”`/sys`” adalah pseudo-filesystem dan mengekspos struktur data internal kernel ke userspace. Dengan kata lain, entri ini bersifat virtual, yang berarti bahwa mereka bertindak sebagai jendela yang nyaman ke dalam pengoperasian sistem operasi.

Direktori ”`/proc`” berisi (antara lain) satu subdirektori untuk setiap proses yang berjalan pada sistem, yang dinamai dengan id proses (PID). Utilitas sistem yang mengakses informasi proses, seperti ps(1), mendapatkan informasi mereka dari struktur direktori ini.

Direktori di bawah ”`/proc/sys/`” berisi antarmuka untuk mengubah parameter kernel tertentu pada saat berjalan. (Anda dapat melakukan hal yang sama melalui perintah khusus sysctl(8) atau berkas preload/konfigurasinya ”`/etc/sysctl.d/*.conf`”.)

Orang sering panik ketika mereka melihat satu berkas khususnya - ”`/proc/kcore`” - yang umumnya sangat besar. Ini (kurang lebih) salinan konten memori komputer Anda. Ini digunakan untuk men-debug kernel. Ini adalah berkas virtual yang menunjuk ke memori komputer, jadi jangan khawatir tentang ukurannya.

Direktori di bawah ”`/sys`” berisi struktur data kernel yang diekspor, atributnya, dan hubungan di antara mereka. Ini juga berisi antarmuka untuk mengubah parameter kernel tertentu pada saat berjalan.

Lihat ”`proc.txt(.gz)`”, ”`sysfs.txt(.gz)`”, dan dokumen terkait lainnya dalam dokumentasi kernel Linux (”`/usr/share`
yang disediakan oleh paket `linux-doc-*`.

1.2.13 tmpfs

tmpfs adalah sistem berkas sementara yang menyimpan semua berkas dalam memori virtual. Data tmpfs dalam cache halaman pada memori dapat ditukar ke ruang swap pada disk sesuai kebutuhan.

Direktori ”`/run`” dikait sebagai tmpfs dalam proses boot awal. Hal ini memungkinkan menulis ke sana bahkan ketika direktori ”/” dikait sebagai hanya-baca. Ini adalah lokasi baru untuk penyimpanan berkas keadaan sementara dan menggantikan beberapa lokasi yang dijelaskan dalam Filesystem Hierarchy Standard versi 2.3:

- ”`/var/run`” → ”`/run`”
- ”`/var/lock`” → ”`/run/lock`”

- ”/dev/shm” → ”/run/shm”

Lihat ”tmpfs.txt(.gz)” dalam dokumentasi kernel Linux (”/usr/share/doc/linux-doc-*/Documentation/files yang disediakan oleh paket linux-doc-*.

1.3 Midnight Commander (MC)

Midnight Commander (MC) adalah GNU ”pisau tentara Swiss” untuk konsol Linux dan lingkungan terminal lainnya. Ini memberi pemula pengalaman konsol berbasis menu yang jauh lebih mudah dipelajari daripada perintah Unix standar.

Anda mungkin perlu menginstall paket Midnight Commander yang berjudul ”mc” dengan yang berikut ini.

```
$ sudo apt-get install mc
```

Gunakan perintah mc(1) untuk menjelajahi sistem Debian. Ini adalah cara terbaik untuk belajar. Silakan jelajahi beberapa lokasi menarik hanya dengan menggunakan tombol kursor dan tombol Enter.

- ”/etc” dan subdirektorinya
- ”/var/log” dan subdirektorinya
- ”/usr/share/doc” dan subdirektorinya
- ”/usr/sbin” dan ”/usr/bin”

1.3.1 Penyesuaian MC

Untuk membuat MC mengubah direktori kerja saat keluar dan cd ke direktori, saya sarankan untuk memodifikasi ”~/.bashrc” untuk menyertakan skrip yang disediakan oleh paket mc.

```
. /usr/lib/mc/mc.sh
```

Lihat mc(1) (di bawah opsi ”-P” karena alasan tersebut. (Jika Anda tidak mengerti apa sebenarnya yang saya bicarakan di sini, Anda dapat melakukan ini nanti.)

1.3.2 Memulai MC

MC dapat dimulai dengan yang berikut.

```
$ mc
```

MC menangani semua operasi berkas melalui menunya, membutuhkan upaya pengguna minimal. Cukup tekan F1 untuk mendapatkan layar bantuan. Anda dapat bermain dengan MC hanya dengan menekan tombol kursor dan tombol fungsi.

Catatan
Di beberapa konsol seperti gnome-terminal(1), ketukan tombol fungsi dapat dicuri oleh program konsol. Anda dapat menonaktifkan fitur-fitur ini di menu ”Preferensi” → ”Umum” dan ”Pintasan” untuk gnome-terminal.

Jika Anda menemui masalah pengkodean karakter yang menampilkan karakter sampah, menambahkan ”-a” ke baris perintah MC dapat membantu mencegah masalah.

Jika ini tidak menjernihkan masalah tampilan Anda dengan MC, lihat Bagian 8.3.1.

1.3.3 Manajer berkas di MC

Default adalah dua panel direktori yang berisi daftar berkas. Mode lain yang berguna adalah mengatur jendela yang tepat ke ”informasi” untuk melihat informasi hak istimewa akses berkas, dll. Berikut ini adalah beberapa penekanan tombol penting. Dengan daemon gpm(8) berjalan, seseorang dapat menggunakan tetikus pada konsol karakter Linux juga. (Pastikan untuk menekan tombol shift untuk mendapatkan perilaku normal potong and tempel di MC.)

tombol	**pengikatan tombol**
`F1`	menu bantuan
`F3`	penampil berkas internal
`F4`	penyunting internal
`F9`	mengaktifkan menu tarik turun
`F10`	keluar dari Midnight Commander
`Tab`	berpindah di antara dua jendela
`Insert` atau `Ctrl-T`	menandai berkas untuk operasi beberapa berkas seperti menyalin
`Del`	menghapus berkas (hati-hati---atur MC ke mode hapus yang aman)
Tombol kursor	sudah jelas

Tabel 1.11: Pengikatan tombol MC

1.3.4 Trik baris perintah di MC

- Perintah `cd` mengubah direktori yang ditampilkan pada layar yang dipilih.
- `Ctrl-Enter` atau `Alt-Enter` menyalin nama berkas ke baris perintah. Gunakan ini dengan perintah cp(1) dan mv(1) bersama dengan penyuntingan baris perintah.
- `Alt-Tab` memperlihatkan pilihan ekspansi nama berkas shell.
- Kita dapat menentukan direktori awal untuk kedua jendela sebagai argumen untuk MC; misalnya, ”`mc /etc /root`”.
- `Esc` + `n-key` → `Fn` (yaitu, `Esc` + `1` → `F1`, dll.; `Esc` + `0` → `F10`)
- Menekan `Esc` sebelum tombol memiliki efek yang sama seperti menekan `Alt` dan tombol bersama-sama.; yaitu, ketik `Esc` + `c` untuk `Alt-C`. `Esc` disebut tombol meta dan kadang-kadang ditulis sebagai ”`M-`”.

1.3.5 Penyunting internal di MC

Penyunting internal memiliki skema potong dan tempel yang menarik. Menekan `F3` menandai dimulainya pilihan, `F3` kedua menandai akhir seleksi dan menyoroti pilihan. Kemudian Anda dapat memindahkan kursor Anda. Jika Anda menekan F6, area yang dipilih dipindahkan ke lokasi kursor. Jika Anda menekan F5, area yang dipilih disalin dan disisipkan di lokasi kursor. `F2` menyimpan berkas. `F10` mengeluarkanmu. Sebagian besar tombol kursor bekerja secara intuitif.

Penyunting ini dapat langsung dimulai pada sebuah berkas menggunakan salah satu perintah berikut.

```
$ mc -e filename_to_edit
```

```
$ mcedit filename_to_edit
```

Ini bukan penyunting multi-jendela, tetapi seseorang dapat menggunakan beberapa konsol Linux untuk mencapai efek yang sama. Untuk menyalin antar jendela, gunakan tombol Alt-F*n* untuk beralih konsol virtual dan gunakan ”Berkas→Sisipkan berkas” atau ”Berkas→Salin ke berkas” untuk memindahkan sebagian berkas ke berkas lain.

Penyunting internal ini dapat diganti dengan penyunting pilihan eksternal.

Juga, banyak program menggunakan variabel lingkungan "`$EDITOR`" atau "`$VISUAL`" untuk memutuskan penyunting mana yang akan digunakan. Jika Anda tidak nyaman dengan vim(1) atau nano(1) pada awalnya, Anda dapat mengatur ini ke "`mcedit`" dengan menambahkan baris berikut ke "`~/.bashrc`".

```
export EDITOR=mcedit
export VISUAL=mcedit
```

Saya sarankan untuk mengatur ini ke "`vim`" jika memungkinkan.

Jika Anda merasa tidak nyaman dengan vim(1), Anda dapat terus menggunakan mcedit(1) untuk sebagian besar tugas pemeliharaan sistem.

1.3.6 Penampil internal di MC

MC adalah penampil yang sangat cerdas. Ini adalah alat yang hebat untuk mencari kata-kata dalam dokumen. Saya selalu menggunakan ini untuk berkas-berkas di direktori "`/usr/share/doc`". Ini adalah cara tercepat untuk menelusuri banyak informasi Linux. Penampil ini dapat langsung dimulai dengan menggunakan salah satu perintah berikut.

```
$ mc -v path/to/filename_to_view
```

```
$ mcview path/to/filename_to_view
```

1.3.7 Fitur mulai sendiri dari MC

Tekan Enter pada berkas, dan program yang sesuai menangani konten berkas (lihat Bagian 9.4.11). Ini adalah fitur MC yang sangat nyaman.

jenis berkas	**reaksi untuk tombol enter**
berkas yang dapat dieksekusi	menjalankan perintah
berkas man	menyalurkan konten ke perangkat lunak penampil
berkas html	menyalurkan konten ke peramban web
berkas "`*.tar.gz`" dan "`*.deb`"	menelusuri isinya seolah-olah subdirektori

Tabel 1.12: Reaksi terhadap tombol enter di MC

Untuk memungkinkan fitur penampil dan berkas virtual ini berfungsi, berkas-berkas yang dapat dilihat tidak boleh diatur sebagai executable. Ubah status mereka menggunakan chmod(1) atau melalui menu berkas MC.

1.3.8 Sistem berkas virtual MC

MC dapat digunakan untuk mengakses berkas-berkas melalui Internet. Buka menu dengan menekan `F9`, lalu ketik "`Enter`" dan "h" untuk mengaktifkan sistem berkas Shell. Masukkan URL dalam bentuk "`sh://[pengguna@]mesin[:ops`
yang mengambil direktori jarak jauh dan muncul seperti direktori lokal memakai `ssh`.

1.4 Lingkungan kerja dasar mirip Unix

Meskipun MC memungkinkan Anda untuk melakukan hampir segalanya, sangat penting bagi Anda untuk belajar bagaimana menggunakan alat baris perintah yang dipanggil dari prompt shell dan menjadi akrab dengan lingkungan kerja mirip Unix.

1.4.1 Shell log masuk

Karena shell log masuk mungkin dipakai oleh beberapa program inisialisasi sistem, bijaksana untuk mempertahankannya sebagai bash(1) dan menghindari beralih shell log masuk dengan chsh(1).

Bila Anda ingin memakai prompt shell interaktif lain, atur itu dari konfigurasi emulator terminal GUI atau mulailah itu dari `~/.bashrc`. mis., dengan menempatkan "`exec /usr/bin/zsh -i -l`" atau "`exec /usr/bin/fish -i -l`" di dalamnya.

paket	popcon	ukuran	Shell POSIX	deskripsi
`bash`	V:886, I:999	7308	Ya	Bash: GNU Bourne Again SHell (standar de facto)
`bash-completion`	V:37, I:962	1952	T/T	penyelesaian yang dapat diprogram untuk shell bash
`dash`	V:929, I:999	207	Ya	Debian Almquist Shell, bagus untuk skrip shell
`zsh`	V:39, I:68	2571	Ya	Z shell: shell standar dengan banyak tambahan
`tcsh`	V:4, I:14	1366	Tidak	TENEX C Shell: versi yang disempurnakan dari Berkeley csh
`mksh`	V:4.7, I:7.5	7713	Ya	Versi dari shell Korn
`csh`	V:1.0, I:5.2	348	Tidak	C Shell OpenBSD, suatu versi dari Berkeley csh
`sash`	V:0.3, I:4.7	1245	Ya	Stand-alone dengan perintah-perintah bawaan (Tidak dimaksudkan untuk standar "`/usr/bin/sh`")
`ksh`	V:0.3, I:7.9	65	Ya	versi AT&T yang sebenarnya dari shell Korn
`rc`	V:0.07, I:0.70	182	Tidak	implementasi dari AT&T Plan 9 rc shell
`posh`	V:0.02, I:0.22	195	Ya	Policy-compliant Ordinary SHell (turunan `pdksh`)

Tabel 1.13: Daftar program shell

Tip
Meskipun shell mirip POSIX berbagi sintaks dasar, mereka dapat berbeda dalam perilaku untuk hal-hal yang mendasar seperti variabel shell dan ekspansi glob. Silakan periksa dokumentasi mereka untuk rinciannya.

Dalam bab tutorial ini, shell interaktif selalu berarti `bash`.

1.4.2 Menyesuaikan bash

Anda dapat menyesuaikan perilaku bash(1) dengan "`~/.bashrc`".

Misalnya, coba yang berikut.

```
# enable bash-completion
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

# CD upon exiting MC
```

```
. /usr/lib/mc/mc.sh

# set CDPATH to a good one
CDPATH=.:/usr/share/doc:~:~/Desktop:~
export CDPATH

PATH="${PATH+$PATH:}/usr/sbin:/sbin"
# set PATH so it includes user's private bin if it exists
if [ -d ~/bin ] ; then
  PATH="~/bin${PATH+:$PATH}"
fi
export PATH

EDITOR=vim
export EDITOR
```

Tip
Anda dapat menemukan lebih banyak tips penyesuaian `bash`, seperti Bagian 9.3.6, dalam Bab 9.

Tip
Paket `bash-completion` memungkinkan penyelesaian yang dapat diprogram untuk `bash`.

1.4.3 Ketukan tombol khusus

Dalam lingkungan mirip Unix, ada beberapa ketukan tombol yang memiliki arti khusus. Harap dicatat bahwa pada konsol karakter Linux normal, hanya tombol `Ctrl` dan `Alt` kiri yang berfungsi seperti yang diharapkan. Berikut adalah beberapa ketukan kunci yang penting untuk diingat.

tombol	**deskripsi pengikatan kunci**
`Ctrl-U`	menghapus baris sebelum kursor
`Ctrl-H`	menghapus satu karakter sebelum kursor
`Ctrl-D`	mengakhiri masukan (keluar shell jika Anda menggunakan shell)
`Ctrl-C`	mengakhiri program yang sedang berjalan
`Ctrl-Z`	menghentikan sementara program dengan memindahkannya ke pekerjaan latar belakang
`Ctrl-S`	menghentikan keluaran ke layar
`Ctrl-Q`	mengaktifkan kembali keluaran ke layar
`Ctrl-Alt-Del`	reboot/halt sistem, lihat inittab(5)
`tombol Alt Kiri` (opsional, `tombol Windows`)	tombol meta untuk Emacs dan UI serupa
`Up-arrow`	mulai pencarian riwayat perintah di bawah `bash`
`Ctrl-R`	memulai pencarian riwayat perintah inkremental di bawah `bash`
`Tab`	melengkapi masukan nama berkas ke baris perintah di bawah `bash`
`Ctrl-V Tab`	memasukkan `Tab` tanpa ekspansi ke baris perintah di bawah `bash`

Tabel 1.14: Daftar pengikatan kunci untuk bash

Tip
Fitur terminal `Ctrl-S` dapat dinonaktifkan menggunakan stty(1).

1.4.4 Operasi tetikus

Operasi tetikus untuk teks pada sistem Debian mencampur 2 gaya dengan beberapa keunikan:

- Operasi tetikus gaya Unix tradisional:
  - gunakan tombol 3 (klik)
  - gunakan PRIMER
  - digunakan oleh aplikasi X seperti `xterm` dan aplikasi teks di konsol Linux
- Operasi tetikus gaya GUI modern:
  - gunakan tombol 2 (seret + klik)
  - gunakan PRIMER dan PAPANKLIP
  - digunakan dalam aplikasi GUI modern seperti `gnome-terminal`

aksi	**respon**
Kiri-klik-dan-seret tetikus	pilih rentang sebagai pilihan PRIMER
Klik kiri	pilih awal rentang untuk pilihan PRIMER
Klik kanan (tradisional)	pilih akhir rentang untuk pilihan PRIMER
Klik kanan (modern)	menu bergantung konteks (potong/salin/tempel)
Klik tengah atau Shift-`Ins`	menyisipkan pilihan PRIMER di kursor
`Ctrl-X`	memotong pilihan PRIMER ke PAPANKLIP
`Ctrl-C` (`Shift-Ctrl-C` dalam terminal)	menyalin pilihan PRIMER ke PAPANKLIP
`Ctrl-V`	menempelkan PAPANKLIP pada kursor

Tabel 1.15: Daftar operasi tetikus dan tindakan tombol terkait pada Debian

Di sini, pilihan PRIMER adalah rentang teks yang disorot. Dalam program terminal, `Shift-Ctrl-C` digunakan sebagai gantinya untuk menghindari menghentikan program yang sedang berjalan.

Roda tengah pada tetikus roda modern dianggap tombol tetikus tengah dan dapat digunakan untuk klik tengah. Mengklik tombol kiri dan kanan tetikus bersama-sama berfungsi sebagai klik tengah di bawah situasi sistem tetikus 2 tombol.

Untuk menggunakan tetikus di konsol karakter Linux, Anda harus menjalankan gpm(8) sebagai daemon.

1.4.5 Pager

Perintah less(1) adalah pager yang disempurnakan (peramban konten berkas). Ini membaca berkas yang ditentukan oleh argumen perintah atau masukan standarnya. Tekan ”h” jika Anda memerlukan bantuan saat menjelajah dengan perintah `less`. Ini dapat melakukan jauh lebih banyak daripada more(1) dan dapat didongkrak dengan mengeksekusi ”`eval $(lesspipe)`” atau ”`eval $(lessfile)`” dalam skrip awal mula shell. Lihat lebih lanjut di ”`/usr/share/doc/less/LESSOPEN`”. Opsi ”`-R`” memungkinkan keluaran karakter mentah dan memfungsikan escape sequence warna ANSI. Lihat less(1).

Tip
Dalam perintah `less`, ketik ”h” untuk melihat layar bantuan, ketik ”`/`” atau ”?” untuk mencari string, dan ketik ”`-i`” untuk mengubah sensitivitas atas huruf besar kecil.

1.4.6 Penyunting teks

Anda harus menjadi mahir dalam salah satu varian program Vim atau Emacs yang populer di sistem mirip Unix.

Saya pikir membiasakan diri dengan perintah Vim adalah hal yang benar untuk dilakukan, karena Vi-editor selalu ada di dunia Linux/Unix. (Sebenarnya, `vi` asli atau `nvi` baru adalah program yang Anda temukan di mana-mana. Saya memilih Vim sebagai gantinya untuk pemula karena menawarkan bantuan melalui tombol `F1` sementara itu cukup mirip dan lebih kuat.)

Jika Anda memilih Emacs atau XEmacs sebagai pilihan penyunting Anda, itu adalah pilihan lain yang baik memang, terutama untuk pemrograman. Emacs memiliki sejumlah besar fitur lain juga, termasuk berfungsi sebagai pembaca berita, penyunting direktori, program surat, dll. Ketika digunakan untuk pemrograman atau menyunting skrip shell, itu secara cerdas mengenali format dari apa yang sedang Anda kerjakan, dan mencoba memberikan bantuan. Beberapa orang berpendapat bahwa satu-satunya program yang mereka butuhkan di Linux adalah Emacs. Sepuluh menit belajar Emacs sekarang dapat menghemat berjam-jam kemudian. Memiliki manual GNU Emacs untuk referensi ketika belajar Emacs sangat dianjurkan.

Semua program ini biasanya datang dengan program bimbingan belajar bagi Anda untuk mempelajarinya dengan latihan. Mulailah Vim dengan mengetik "`vim`" dan tekan tombol F1. Anda setidaknya harus membaca 35 baris pertama. Kemudian lakukan kursus pelatihan daring dengan memindahkan kursor ke "`|tutor|`" dan menekan `Ctrl-]`.

Catatan
Penyunting yang baik, seperti Vim dan Emacs, dapat menangani UTF-8 dan pengodean teks eksotis lainnya dengan benar. Ide yang baik untuk menggunakan lingkungan GUI di lokal UTF-8 dan untuk memasang program yang diperlukan dan fonta untuk itu. Penyunting memiliki opsi untuk mengatur pengodean berkas independen dari lingkungan GUI. Silakan lihat dokumentasi mereka tentang teks multi-byte.

1.4.7 Menyiapkan penyunting teks default

Debian hadir dengan sejumlah penyunting yang berbeda. Kami sarankan untuk memasang paket `vim`, seperti yang disebutkan di atas.

Debian menyediakan akses terpadu ke penyunting baku sistem melalui perintah "`/usr/bin/editor`" sehingga program lain (misalnya, reportbug(1)) dapat memanggilnya. Anda dapat mengubahnya dengan yang berikut.

```
$ sudo update-alternatives --config editor
```

Pilihan "`/usr/bin/vim.basic`" daripada "`/usr/bin/vim.tiny`" adalah rekomendasi saya untuk pemula karena mendukung penyorotan sintaksis.

Tip
Banyak program menggunakan variabel lingkungan "`$EDITOR`" atau "`$VISUAL`" untuk memutuskan penyunting mana yang akan digunakan (lihat Bagian 1.3.5 dan Bagian 9.4.11). Untuk konsistensi pada sistem Debian, atur ini ke "`/usr/bin/editor`". (Secara historis, "`$EDITOR`" adalah "`ed`" dan "`$VISUAL`" adalah "`vi`".)

1.4.8 Menggunakan vim

vim(1) baru-baru ini memulai dirinya dalam opsi "`nocompatible`" waras dan masuk ke mode `NORMAL`.[1]

Harap gunakan program "`vimtutor`" untuk belajar `vim` melalui suatu kursus tutorial interaktif.

Program `vim` mengubah perilakunya ke tombol yang diketik berdasarkan **mode**. Mengetikkan tombol ke buffer sebagian besar dilakukan dalam mode `INSERT` dan mode `REPLACE`. Menggerakkan kursor sebagian besar dilakukan dalam mode `NORMAL`. Pemilihan interaktif dilakukan dalam mode `VISUAL`. Mengetik "`:`" dalam mode `NORMAL` mengubah **mode** menjadi mode `Ex`. Mode `Ex` menerima perintah.

[1] Bahkan `vim` yang lebih tua dapat dimulai dalam mode "`nocompatible`" yang waras dengan memulainya dengan opsi "`-N`".

mode	ketikan tombol	aksi
NORMAL	:help|only	menampilkan berkas bantuan
NORMAL	:e filename.ext	membuka penyangga baru untuk menyunting namaberkas.ext
NORMAL	:w	menimpa penyangga saat ini ke berkas asli
NORMAL	:w filename.ext	menulis penyangga saat ini ke namaberkas.ext
NORMAL	:q	keluar vim
NORMAL	:q!	memaksa keluar vim
NORMAL	:only	tutup semua jendela belah terbuka lainnya
NORMAL	:set nocompatible?	periksa apakah vim berada dalam mode nocompatible
NORMAL	:set nocompatible	atur vim ke mode nocompatible yang waras
NORMAL	i	masuk ke mode SISIP
NORMAL	R	masuk ke mode TIMPA
NORMAL	v	masuk ke mode VISUAL
NORMAL	V	masuk ke mode VISUAL baris
NORMAL	Ctrl-V	masuk ke mode VISUAL blok
selain TERMINAL-JOB	ESC-key	masuk mode NORMAL
NORMAL	:term	masuk ke mode TERMINAL-JOB
TERMINAL-NORMAL	i	masuk ke mode TERMINAL-JOB
TERMINAL-JOB	Ctrl-W N (atau Ctrl-\ Ctrl-N)	masuk ke mode TERMINAL-NORMAL
TERMINAL-JOB	Ctrl-W :	masuk mode Ex dalam mode TERMINAL-NORMAL

Tabel 1.16: Daftar ketukan tombol Vim dasar

Tip
Vim hadir dengan paket **Netrw**. Netrw mendukung membaca berkas, menulis berkas, meramban direktori melalui jaringan, dan meramban lokal! Cobalah Netrw dengan "`vim .`" (sebuah titik sebagai argumen) dan membaca manualnya di "`:help netrw`".

Untuk konfigurasi tingkat lanjut `vim`, lihat Bagian 9.2.

1.4.9 Merekam aktivitas shell

Keluaran dari perintah shell dapat bergulir keluar layar Anda dan mungkin hilang selamanya. Adalah praktik yang baik untuk mencatat aktivitas shell ke dalam berkas bagi Anda untuk meninjaunya nanti. Catatan semacam ini sangat penting ketika Anda melakukan tugas-tugas manajemen sistem apa pun.

Tip
Vim baru (versi>=8.2) dapat digunakan untuk merekam aktivitas shell secara bersih menggunakan mode `TERMINAL-JOB`. Lihat Bagian 1.4.8.

Metode dasar untuk merekam aktivitas shell adalah menjalankannya di bawah script(1).

Misalnya, coba yang berikut ini

```
$ script
Script started, file is typescript
```

Lakukan perintah shell apa pun di bawah `script`.

Tekan `Ctrl-D` untuk keluar dari `script`.

```
$ vim typescript
```

Lihat Bagian 9.1.1 .

1.4.10 Perintah Unix Dasar

Mari kita pelajari perintah dasar Unix. Di sini saya menggunakan "Unix" dalam arti generiknya. Setiap OS klon Unix biasanya menawarkan perintah yang setara. Sistem Debian tidak terkecuali. Jangan khawatir jika beberapa perintah tidak berfungsi seperti yang Anda inginkan sekarang. Jika `alias` digunakan dalam shell, keluaran perintah yang sesuai akan berbeda. Contoh-contoh ini tidak dimaksudkan untuk dieksekusi dalam urutan ini.

Cobalah semua perintah berikut dari akun pengguna yang tidak memiliki hak istimewa.

Catatan
Unix memiliki tradisi untuk menyembunyikan nama berkas yang dimulai dengan "`.`". Mereka secara tradisional adalah berkas-berkas yang berisi informasi konfigurasi dan preferensi pengguna.
Untuk perintah `cd`, lihat builtins(7).
Pager baku dari sistem Debian polos adalah more(1) yang tidak dapat menggulir balik. Dengan memasang paket `less` menggunakan baris perintah "`apt-get install less`", less(1) menjadi pager baku dan Anda dapat menggulir balik dengan tombol-tombol kursor.
"`[`" dan "`]`" dalam ekspresi reguler dari perintah "`ps aux | grep -e "[e]xim4*"`" di atas memungkinkan `grep` untuk menghindari pencocokan dirinya sendiri. "`4*`" dalam ekspresi reguler berarti 0 atau lebih pengulangan karakter "`4`" sehingga memungkinkan `grep` untuk mencocokkan "`exim`" dan "`exim4`". Meskipun "`*`" digunakan dalam glob namaberkas shell dan ekspresi reguler, maknanya berbeda. Pelajari ekspresi reguler dari grep(1).

perintah	deskripsi
`pwd`	menampilkan nama direktori kerja/saat ini
`whoami`	menampilkan nama pengguna saat ini
`id`	menampilkan identitas pengguna saat ini (nama, uid, gid, dan grup terkait)
`file` *foo*	menampilkan jenis berkas untuk berkas "*foo*"
`type -p` *commandname*	menampilkan lokasi berkas perintah "*nama_perintah*"
`which` *commandname*	, ,
`type` *commandname*	menampilkan informasi tentang perintah "*nama_perintah*"
`apropos` *key-word*	temukan perintah yang terkait dengan "*kata-kunci*"
`man -k` *kata-kunci*	, ,
`whatis` *namaperintah*	tampilkan penjelasan satu baris pada perintah "*namaperintah*"
`man -a` *commandname*	menampilkan penjelasan pada perintah "*namaperintah*" (gaya Unix)
`info` *commandname*	menampilkan penjelasan yang agak panjang pada perintah "*namaperintah*" (gaya GNU)
`ls`	menampilkan isi direktori (berkas non-dot dan direktori)
`ls -a`	menampilkan isi direktori (semua berkas dan direktori)
`ls -A`	menampilkan daftar isi direktori (hampir semua berkas dan direktori, yaitu, melewati "`..`" dan "`.`")
`ls -la`	menampilkan semua isi direktori dengan informasi detail
`ls -lai`	menampilkan semua isi direktori dengan nomor inode dan informasi detail
`ls -d`	menampilkan semua direktori di bawah direktori saat ini
`tree`	menampilkan konten pohon berkas
`lsof` *foo*	menampilkan daftar status terbuka dari berkas "*foo*"
`lsof -p` *pid*	menampilkan daftar berkas yang dibuka oleh ID proses: "*pid*"
`mkdir` *foo*	membuat direktori baru "*foo*" di direktori saat ini
`rmdir` *foo*	menghapus direktori "*foo*" di direktori saat ini
`cd` *foo*	berpindah direktori ke direktori "*foo*" di direktori saat ini atau di direktori yang tercantum dalam variabel "`$CDPATH`"
`cd /`	berpindah direktori ke direktori root
`cd`	berpindah direktori ke direktori beranda pengguna saat ini
`cd` */foo*	berpindah direktori ke direktori path absolut "*/foo*"
`cd ..`	berpindah direktori ke direktori induk
`cd ~`*foo*	berpindah direktori ke direktori rumah pengguna "*foo*"
`cd -`	berpindah direktori ke direktori sebelumnya
`</etc/motd pager`	menampilkan isi "`/etc/motd`" menggunakan pager baku
`touch` *junkfile*	membuat berkas kosong "*junkfile*"
`cp` *foo bar*	menyalin berkas yang ada "*foo*" ke berkas baru "*bar*"
`rm` *junkfile*	menghapus berkas "*junkfile*"
`mv` *foo bar*	mengganti nama berkas yang ada "*foo*" menjadi nama baru "*bar*" ("*bar*" tidak boleh ada)
`mv` *foo bar*	memindahkan berkas yang ada "*foo*" ke lokasi baru "*bar/foo*" (direktori "*bar*" harus ada)
`mv` *foo bar/baz*	memindahkan berkas yang ada "*foo*" ke lokasi baru dengan nama baru "*bar/baz*" (direktori "*bar*" harus ada tetapi direktori "*bar/baz*" tidak boleh ada)
`chmod 600` *foo*	membuat berkas yang ada "*foo*" menjadi tidak dapat dibaca dan tidak dapat ditulis oleh orang lain (tidak dapat dieksekusi untuk semua)
`chmod 644` *foo*	membuat berkas yang ada "*foo*" menjadi bisa dibaca tetapi tidak dapat ditulis oleh orang lain (tidak dapat dieksekusi untuk semua)
`chmod 755` *foo*	membuat berkas yang ada "*foo*" menjadi bisa dibaca tetapi tidak dapat ditulis oleh orang lain (dapat dieksekusi untuk semua)
`find . -name` *pattern*	mencari nama berkas yang cocok menggunakan "*pola*" shell (lebih lambat)
`locate -d . `*pattern*	mencari nama berkas yang cocok menggunakan "*pola*" shell (lebih cepat menggunakan basis data yang dihasilkan secara teratur)
`grep -e "`*pattern*`" *.html`	mencari suatu "*pola*" di semua berkas yang berakhiran "`.html`" di direktori saat ini dan tampilkan semuanya
.	menampilkan informasi proses menggunakan layar penuh, ketik

Silakan melintasi direktori dan mengintip ke dalam sistem menggunakan perintah di atas sebagai pelatihan. Jika Anda memiliki pertanyaan tentang perintah konsol apa pun, pastikan untuk membaca halaman manual.

Misalnya, coba yang berikut ini

```
$ man man
$ man bash
$ man builtins
$ man grep
$ man ls
```

Gaya halaman man mungkin sedikit sulit untuk dibiasakan, karena mereka agak singkat, terutama yang sangat tradisional, yang lebih tua. Tetapi begitu Anda terbiasa, Anda akan menghargai ringkasnya.

Harap dicatat bahwa banyak perintah mirip Unix termasuk yang dari GNU dan BSD menampilkan informasi bantuan singkat jika Anda memanggil mereka dalam salah satu cara berikut (atau tanpa argumen dalam beberapa kasus).

```
$ commandname --help
$ commandname -h
```

1.5 Perintah shell sederhana

Sekarang Anda mulai merasakan tentang cara menggunakan sistem Debian. Mari kita lihat jauh ke dalam mekanisme eksekusi perintah dalam sistem Debian. Di sini, saya telah menyederhanakan realitas untuk pemula. Lihat bash(1) untuk penjelasan yang tepat.

Perintah sederhana adalah urutan komponen.

1. Penugasan variabel (opsional)
2. Nama perintah
3. Argumen (opsional)
4. Pengalihan (opsional: > , >> , < , << , dll.)
5. Operator kontrol (opsional: && , | | , *newline* , ; , & , (,))

1.5.1 Eksekusi perintah dan variabel lingkungan

Nilai-nilai dari beberapa variabel lingkungan mengubah perilaku beberapa perintah Unix.

Nilai baku variabel lingkungan pada awalnya ditetapkan oleh sistem PAM dan kemudian beberapa di antaranya dapat diatur ulang oleh beberapa program aplikasi.

- Sistem PAM seperti `pam_env` dapat mengatur variabel lingkungan dengan `/etc/pam.conf`", "`/etc/environment`", dan "`/etc/default/locale`".
- Lingkungan GUI modern yang dijalankan dari unit systemd mode pengguna (lihat systemd.unit(5)) menata variabel lingkungannya dengan suatu berkas dalam direktori "`~/.config/environment.d/`".
- Inisialisasi program spesifik pengguna dapat mengatur ulang variabel lingkungan dengan "`~/.profile`", "`~/.bash_prof`
  dan "`~/.bashrc`".

1.5.2 Variabel ”$LANG”

Lokal baku didefinisikan dalam variabel lingkungan ”$LANG” dan dikonfigurasi sebagai ”LANG=xx_YY.UTF-8” oleh installer atau dengan konfigurasi GUI berikutnya, mis., ”Pengaturan” → ”Wilayah & Bahasa” → ”Bahasa” / ”Format” untuk GNOME.

Catatan
Saya sarankan Anda untuk mengkonfigurasi lingkungan sistem hanya dengan variabel ”$LANG” untuk saat ini dan untuk menjauh dari variabel ”$LC_*” kecuali itu benar-benar diperlukan.

Nilai lokal lengkap yang diberikan kepada variabel ”$LANG” terdiri dari 3 bagian: ”xx_YY.ZZZZ”.

nilai lokal	arti
xx	Kode bahasa ISO 639 (huruf kecil) seperti ”en”
YY	Kode negara ISO 3166 (huruf besar) seperti ”US”
ZZZZ	codeset, selalu diatur ke ”UTF-8”

Tabel 1.18: 3 bagian dari nilai lokal

rekomendasi lokal	Bahasa (area)
en_US.UTF-8	Inggris (Amerika Serikat)
en_GB.UTF-8	Inggris (Inggris Raya)
fr_FR.UTF-8	Prancis (Prancis)
de_DE.UTF-8	Jerman (Jerman)
it_IT.UTF-8	Italia (Italia)
es_ES.UTF-8	Spanyol (Spanyol)
ca_ES.UTF-8	Katala (Spanyol)
sv_SE.UTF-8	Swedia (Swedia)
pt_BR.UTF-8	Portugis (Brasil)
ru_RU.UTF-8	Rusia (Rusia)
zh_CN.UTF-8	Cina (Republik Rakyat Cina)
zh_TW.UTF-8	Cina (Taiwan)
ja_JP.UTF-8	Jepang (Jepang)
ko_KR.UTF-8	Korea (Republik Korea)
vi_VN.UTF-8	Vietnam (Vietnam)

Tabel 1.19: Daftar rekomendasi lokal

Eksekusi perintah umum menggunakan urutan baris shell sebagai berikut.

```
$ echo $LANG
en_US.UTF-8
$ date -u
Wed 19 May 2021 03:18:43 PM UTC
$ LANG=fr_FR.UTF-8 date -u
mer. 19 mai 2021 15:19:02 UTC
```

Di sini, program date(1) dijalankan dengan nilai lokal yang berbeda.

- Untuk perintah pertama, ”$LANG” diatur ke nilai lokal baku sistem ”en_US.UTF-8”.
- Untuk perintah kedua, ”$LANG” diatur ke nilai lokal UTF-8 Prancis ”fr_FR.UTF-8”.

Sebagian besar eksekusi perintah biasanya tidak memiliki definisi variabel lingkungan sebelumnya. Untuk contoh di atas, Anda dapat mengeksekusi sebagai berikut.

```
$ LANG=fr_FR.UTF-8
$ date -u
mer. 19 mai 2021 15:19:24 UTC
```

Tip
Saat mengajukan laporan bug, menjalankan dan memeriksa perintah di bawah lokal "en_US.UTF-8" adalah ide yang baik, jika Anda menggunakan lingkungan non-Inggris.

Untuk detail lengkap konfigurasi lokal, lihat Bagian 8.1.

1.5.3 Variabel "$PATH"

Ketika Anda mengetik perintah ke dalam shell, shell mencari perintah dalam daftar direktori yang terkandung dalam variabel lingkungan "$PATH". Nilai variabel lingkungan "$PATH" juga disebut path pencarian shell.

Dalam instalasi Debian baku, variabel lingkungan "$PATH" dari akun pengguna mungkin tidak termasuk "/usr/sbin" dan "/usr/sbin". Misalnya, perintah ifconfig perlu dijalankan dengan path lengkap sebagai "/usr/sbin/ifconfig". (Perintah ip yang serupa terletak di "/usr/bin".)

Anda dapat mengubah variabel lingkungan "$PATH" dari shell Bash dengan berkas "~/.bash_profile" atau "~/.bashrc"

1.5.4 Variabel "$HOME"

Banyak perintah menyimpan konfigurasi spesifik pengguna di direktori rumah dan mengubah perilaku mereka dengan isinya. Direktori rumah diidentifikasi oleh variabel lingkungan "$HOME".

nilai "$HOME"	situasi eksekusi program
/	program yang dijalankan oleh proses init (daemon)
/root	program yang dijalankan dari shell root normal
/home/*normal_user*	program yang dijalankan dari shell pengguna normal
/home/*normal_user*	program yang dijalankan dari menu desktop GUI pengguna normal
/home/*normal_user*	program dijalankan sebagai root dengan "sudo program"
/root	program dijalankan sebagai root dengan "sudo -H program"

Tabel 1.20: Daftar nilai "$HOME"

Tip
Shell mengekspansi "~/" ke direktori rumah pengguna saat ini, yaitu, "$HOME/". Shell mengekspansi "~foo/" ke direktori rumah foo yaitu "/home/foo/".

Lihat Bagian 12.1.5 bila $HOME tidak tersedia bagi program Anda.

1.5.5 Opsi baris perintah

Beberapa perintah menerima argumen. Argumen yang dimulai dengan "-" atau "--" disebut opsi dan mengontrol perilaku perintah.

```
$ date
Thu 20 May 2021 01:08:08 AM JST
$ date -R
Thu, 20 May 2021 01:08:12 +0900
```

Di sini argumen baris perintah "`-R`" mengubah perilaku date(1) untuk menghasilkan string tanggal yang patuh RFC2822.

1.5.6 Glob shell

Seringkali Anda ingin perintah untuk bekerja dengan sekelompok berkas tanpa mengetik semuanya. Pola ekspansi nama berkas menggunakan **glob** shell, (kadang-kadang disebut sebagai **wildcard**), memfasilitasi kebutuhan ini.

pola glob shell	deskripsi aturan kecocokan
*	nama berkas tidak dimulai dengan "`.`"
`.*`	nama berkas yang diawali dengan "`.`" (dalam Bash 5.2 dan setelahnya, "`.`" dan "`..`" dikecualikan secara baku)
?	tepat satu karakter
`[…]`	tepat satu karakter dengan karakter apa pun dalam tanda kurung
`[a-z]`	tepat satu karakter dengan karakter apapun antara "a" dan "z"
`[!…]`	tepat satu karakter selain karakter apa pun yang dilampirkan dalam tanda kurung (tidak termasuk "`!`")

Tabel 1.21: Pola glob shell

Catatan
Suatu "`/`" dalam sebuah nama path tidak bisa dicocokkan dengan pola shell glob yang dicantumkan dalam tabel di atas.

Misalnya, coba yang berikut ini

```
$ mkdir foo; cd foo; touch b1.txt b2.txt b3.c b4.h .b5.txt ..b6.txt
$ mkdir bar; cd bar; touch b1.txt b2.txt b3.c b4.h .b5.txt ..b6.txt ; cd -
$ echo b*.txt
b1.txt b2.txt
$ echo b*
b1.txt b2.txt b3.c b4.h bar
$ echo */*.txt
bar/b1.txt bar/b2.txt
$ echo *.[hc]
b3.c b4.h
$ echo .*
.b5.txt ..b6.txt
$ echo b[!1-3]*
b4.h bar
$ echo b*/b[!1-3]*
bar/b4.h
$ cd ..; rm -rf junk
```

Lihat glob(7).

Catatan
Tidak seperti ekspansi nama berkas normal oleh shell, pola shell "`*`" yang diuji dalam find(1) dengan uji "`-nama`" dll., cocok dengan awal "`.`" dari nama berkas. (Fitur POSIX baru)

Catatan
BASH dapat disetel untuk mengubah perilaku glob dengan opsi bawaan shopt builtin ”`dotglob`”, ”`noglob`”, ”`nocaseglob`”, ”`nullglob`”, ”`extglob`”, dll. Lihat bash(1).

1.5.7 Nilai kembalian perintah

Setiap perintah mengembalikan status keluarnya (variabel: ”$?”) sebagai nilai kembalian.

status keluar perintah	nilai kembalian numerik	nilai kembalian logis
sukses	nol, 0	**TRUE**
galat	bukan-nol, -1	**FALSE**

Tabel 1.22: Kode keluar perintah

Misalnya, coba yang berikut.

```
$ [ 1 = 1 ] ; echo $?
0
$ [ 1 = 2 ] ; echo $?
1
```

Catatan
Harap dicatat bahwa, dalam konteks logis untuk shell, **sukses** diperlakukan sebagai **TRUE** logis yang memiliki 0 (nol) sebagai nilainya. Ini agak tidak intuitif dan perlu diingatkan di sini.

1.5.8 Urutan perintah umum dan pengalihan shell

Mari kita coba mengingat idiom perintah shell berikut yang diketik dalam satu baris sebagai bagian dari perintah shell.

Sistem Debian adalah sistem multi-tasking. Pekerjaan latar belakang memungkinkan pengguna untuk menjalankan beberapa program dalam satu shell. Manajemen proses latar belakang melibatkan bawaan shell: `jobs`, `fg`, `bg`, dan `kill`. Silakan baca bagian bash(1) di bawah ”SIGNALS”, dan ”JOB CONTROL”, dan builtins(1).

Misalnya, coba yang berikut ini

```
$ </etc/motd pager
```

```
$ pager </etc/motd
```

```
$ pager /etc/motd
```

```
$ cat /etc/motd | pager
```

Meskipun semua 4 contoh pengalihan shell menampilkan hal yang sama, contoh terakhir menjalankan perintah `cat` tambahan dan menghamburkan sumber daya tanpa alasan.

Shell memungkinkan Anda untuk membuka berkas menggunakan `exec` bawaan dengan sembarang deskriptor berkas.

idiom perintah	deskripsi
`command &`	eksekusi **latar belakang** `perintah` dalam subshell
`command1 | command2`	Menyalurkan lewat **pipa** keluaran standar `perintah1` ke masukan standar `perintah2` (eksekusi**bersamaan**)
`command1 2>&1 | command2`	Menyalurkan lewat **pipa** keluaran standar dan kesalahan standar `command1` ke masukan standar `command2` (eksekusi**bersamaan**)
`command1 ; command2`	menjalankan `command1` dan `command2` **secara berurutan**
`command1 && command2`	menjalankan `command1`; jika berhasil, menjalankan `command2` **secara berurutan** (mengembalikan sukses jika `command1` **dan** `command2` berhasil)
`command1 || command2`	menjalankan `command1`; jika tidak berhasil, menjalankan `command2` **secara berurutan** (mengembalikan sukses jika `command1` **atau** `command2` berhasil)
`command > foo`	mengalihkan keluaran standar dari `command` ke berkas `foo` (menimpa)
`command 2> foo`	mengalihkan kesalahan standar `command>` ke berkas `foo` (menimpa)
`command >> foo`	mengalihkan keluaran standar `command` ke berkas `foo` (menambah)
`command 2>> foo`	mengalihkan kesalahan standar `command` ke berkas `foo` (menambah)
`command > foo 2>&1`	mengalihkan keluaran dan kesalahan standar `command` ke berkas `foo`
`command < foo`	mengalihkan masukan standar `command` ke berkas `foo`
`command << delimiter`	mengalihkan masukan standar `command` ke baris berikut sampai ”`delimiter`” dijumpai (di sini dokumen)
`command <<- delimiter`	mengalihkan masukan standar `command` ke baris berikut sampai ”`delimiter`” dijumpai (di sini dokumen, karakter tab di awal dibuang dari baris masukan)

Tabel 1.23: Idiom perintah Shell

```
$ echo Hello >foo
$ exec 3<foo 4>bar  # open files
$ cat <&3 >&4       # redirect stdin to 3, stdout to 4
$ exec 3<&- 4>&-    # close files
$ cat bar
Hello
```

Deskriptor berkas 0-2 sudah ditentukan.

perangkat	deskripsi	deskriptor berkas
`stdin`	masukan standar	0
`stdout`	keluaran standar	1
`stderr`	galat standar	2

Tabel 1.24: Deskriptor berkas yang telah ditentukan

1.5.9 Alias perintah

Anda dapat mengatur alias untuk perintah yang sering digunakan.

Misalnya, coba yang berikut ini

```
$ alias la='ls -la'
```

Sekarang, "`la`" bekerja sebagai singkatan untuk "`ls -la`" yang mencantumkan semua berkas dalam format daftar panjang.

Anda dapat menampilkan daftar alias yang ada dengan `alias` (lihat bash(1) di bawah "PERINTAH BAWAAN SHELL").

```
$ alias
...
alias la='ls -la'
```

Anda dapat mengidentifikasi path atau identitas yang tepat dari perintah berdasarkan `type` (lihat bash(1) di bawah "PERINTAH BAWAAN SHELL").

Misalnya, coba yang berikut ini

```
$ type ls
ls is hashed (/bin/ls)
$ type la
la is aliased to ls -la
$ type echo
echo is a shell builtin
$ type file
file is /usr/bin/file
```

Di sini `ls` baru-baru ini dicari sementara "`file`" tidak, sehingga "`ls`" adalah "di-hash", yaitu, shell memiliki catatan internal untuk akses cepat ke lokasi perintah "`ls`".

Tip
Lihat Bagian 9.3.6.

1.6 Pemrosesan teks mirip Unix

Dalam lingkungan kerja mirip Unix, pemrosesan teks dilakukan dengan menyalurkan teks melalui rantai alat pemrosesan teks standar. Ini adalah inovasi Unix penting lainnya.

1.6.1 Alat teks Unix

Ada beberapa alat pemrosesan teks standar yang sangat sering digunakan pada sistem mirip Unix.

- Tidak ada ekspresi reguler yang digunakan:
  - cat(1) menyambungkan berkas-berkas dan mengeluarkan seluruh konten.
  - tac(1) menyambungkan berkas-berkas dan mengeluarkan secara terbalik.
  - cut(1) memilih bagian dari baris dan keluaran.
  - head(1) mengeluarkan bagian pertama dari berkas.
  - tail(1) mengeluarkan bagian terakhir dari berkas.
  - sort(1) mengurutkan baris-baris berkas teks.
  - uniq(1) menghapus baris duplikat dari suatu berkas terurut.
  - tr(1) menerjemahkan atau menghapus karakter.
  - diff(1) membandingkan berkas-berkas baris demi baris.
- Ekspresi reguler dasar (**BRE**) dipakai sebagai baku:
  - ed(1) adalah penyunting baris primitif.
  - sed(1) adalah penyunting stream.
  - grep(1) mencocokkan teks dengan pola.
  - vim(1) adalah penyunting layar.
  - emacs(1) adalah penyunting layar. (**BRE** yang agak diperluas)
- Extended regular expression (**ERE**) digunakan untuk:
  - awk(1) melakukan pemrosesan teks sederhana.
  - egrep(1) mencocokkan teks dengan pola.
  - `tcl`(3tcl) dapat melakukan setiap pemrosesan teks yang mungkin: Lihat re_syntax(3). Sering digunakan dengan `tk`(3tk).
  - perl(1) dapat melakukan setiap pemrosesan teks yang mungkin. Lihat perlre(1).
  - pcre2grep(1) dari paket `pcre2-util` mencocokkan teks dengan pola Perl Compatible Regular Expressions (PCRE).
  - python(1) dengan modul `re` dapat melakukan setiap pemrosesan teks yang mungkin. Lihat "`/usr/share/doc/python`

Jika Anda tidak yakin apa sebenarnya perintah ini, silakan gunakan "`man command`" untuk mencari tahu sendiri.

Catatan
Pengurutan dan ekspresi rentang tergantung lokal. Jika Anda ingin mendapatkan perilaku tradisional untuk perintah, gunakan lokal **C** atau **C.UTF-8** bukan **UTF-8** normal (lihat Bagian 8.1).

Catatan
Ekspresi reguler Perl (perlre(1)), Perl Compatible Regular Expressions (PCRE), dan ekspresi reguler Python yang ditawarkan oleh modul `re` memiliki banyak ekstensi yang sama ke **ERE** normal.

1.6.2 Ekspresi reguler

Ekspresi reguler digunakan di banyak alat pemrosesan teks. Mereka serupa dengan glob shell, tetapi mereka lebih rumit dan kuat.

Ekspresi reguler menggambarkan pola pencocokan dan terdiri dari karakter teks dan **karakter meta**.

Suatu **karakter meta** hanyalah karakter dengan makna khusus. Ada 2 gaya utama, **BRE** dan **ERE**, tergantung pada alat teks seperti yang dijelaskan di atas.

BRE	ERE	deskripsi ekspresi reguler
`\ . [ ] ^ $ *`	`\ . [ ] ^ $ *`	**karakter meta** umum
`\+ \? \( \) \{ \} |`		**karakter meta** yang di-escape "\" hanya oleh BRE
	`+ ? ( ) { } |`	**karakter meta** yang tidak di-escape "\" hanya oleh ERE
c	c	cocok **bukan karakter meta** "c"
`\c`	`\c`	cocok karakter literal "c" bahkan jika "c" adalah **karakter meta** dengan sendirinya
`.`	`.`	cocok dengan karakter apa pun termasuk baris baru
`^`	`^`	posisi di awal string
`$`	`$`	posisi di akhir string
`\<`	`\<`	posisi di awal sebuah kata
`\>`	`\>`	posisi di akhir sebuah kata
`[abc…]`	`[abc…]`	cocok dengan karakter apa pun dalam "`abc…`"
`[^abc…]`	`[^abc…]`	cocok dengan karakter apa pun kecuali dalam "`abc…`"
`r*`	`r*`	cocok dengan nol atau lebih ekspresi reguler yang diidentifikasi oleh "`r`"
`r\+`	`r+`	cocok dengan satu atau lebih ekspresi reguler yang diidentifikasi oleh "`r`"
`r\?`	`r?`	cocok dengan nol atau satu ekspresi reguler yang diidentifikasi oleh "`r`"
`r1|r2`	`r1|r2`	cocok dengan salah satu ekspresi reguler yang diidentifikasi oleh "`r1`" atau "`r2`"
`\(r1|r2\)`	`(r1|r2)`	cocok dengan salah satu ekspresi reguler yang diidentifikasi oleh "`r1`" atau "`r2`" dan memperlakukannya sebagai ekspresi reguler **yang dikurung**

Tabel 1.25: Karakter meta untuk BRE dan ERE

Ekspresi reguler **`emacs`** pada dasarnya adalah **BRE** tetapi telah diperluas untuk memperlakukan "+" dan "?" sebagai **karakter meta** seperti dalam **ERE**. Dengan demikian, tidak perlu meng-escape mereka dengan "\" dalam ekspresi reguler `emacs`.

grep(1) dapat digunakan untuk melakukan pencarian teks menggunakan ekspresi reguler.

Misalnya, coba yang berikut ini

```
$ egrep 'GNU.*LICENSE|Yoyodyne' /usr/share/common-licenses/GPL
GNU GENERAL PUBLIC LICENSE
GNU GENERAL PUBLIC LICENSE
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
```

Tip
Lihat Bagian 9.3.6.

1.6.3 Ekspresi penggantian

Untuk ekspresi pengganti, beberapa karakter memiliki arti khusus.

ekspresi penggantian	deskripsi teks untuk menggantikan ekspresi pengganti
&	apa yang cocok dengan ekspresi reguler (gunakan \& dalam emacs)
\n	apa yang cocok dengan ekspresi reguler **dikurung** ke-n (”n” adalah angka)

Tabel 1.26: Ekspresi penggantian

Untuk string pengganti Perl, ”$&” digunakan sebagai pengganti ”&” dan ”$n” digunakan sebagai pengganti ”\n”.

Misalnya, coba yang berikut ini

```
$ echo zzz1abc2efg3hij4 | \
sed -e 's/\(1[a-z]*\)[0-9]*\(.*\)$/=&=/'
zzz=1abc2efg3hij4=
$ echo zzz1abc2efg3hij4 | \
sed -E -e 's/(1[a-z]*)[0-9]*(.*)$/=&=/'
zzz=1abc2efg3hij4=
$ echo zzz1abc2efg3hij4 | \
perl -pe 's/(1[a-z]*)[0-9]*(.*)$/=$&=/'
zzz=1abc2efg3hij4=
$ echo zzz1abc2efg3hij4 | \
sed -e 's/\(1[a-z]*\)[0-9]*\(.*\)$/\2===\1/'
zzzefg3hij4===1abc
$ echo zzz1abc2efg3hij4 | \
sed -E -e 's/(1[a-z]*)[0-9]*(.*)$/\2===\1/'
zzzefg3hij4===1abc
$ echo zzz1abc2efg3hij4 | \
perl -pe 's/(1[a-z]*)[0-9]*(.*)$/$2===$1/'
zzzefg3hij4===1abc
```

Di sini harap perhatikan lebih lanjut gaya ekspresi reguler **yang dikurung** dan bagaimana string yang cocok digunakan dalam proses penggantian teks pada alat yang berbeda.

Ekspresi reguler ini juga dapat digunakan untuk gerakan kursor dan tindakan penggantian teks di beberapa penyunting.

Garis miring balik ”\” di akhir baris di baris perintah shell meng-escape baris baru sebagai karakter white space dan melanjutkan masukan baris perintah shell ke baris berikutnya.

Silakan baca semua halaman manual terkait untuk mempelajari perintah ini.

1.6.4 Substitusi global dengan ekspresi reguler

Perintah ed(1) dapat menggantikan semua kemunculan ”FROM_REGEX” dengan ”TO_TEXT” dalam ”file”.

```
$ ed file <<EOF
,s/FROM_REGEX/TO_TEXT/g
w
q
EOF
```

Perintah sed(1) dapat menggantikan semua kemunculan "FROM_REGEX" dengan "TO_TEXT" dalam "file".

```
$ sed -i -e 's/FROM_REGEX/TO_TEXT/g' file
```

Perintah vim(1) dapat menggantikan semua kemunculan "FROM_REGEX" dengan "TO_TEXT" dalam "file" dengan menggunakan perintah ex(1).

```
$ vim '+%s/FROM_REGEX/TO_TEXT/gc' '+update' '+q' file
```

Tip
Bendera "c" di atas memastikan konfirmasi interaktif untuk setiap substitusi.

Beberapa berkas ("file1", "file2", dan "file3") dapat diproses dengan ekspresi reguler secara serupa dengan vim(1) atau perl(1).

```
$ vim '+argdo %s/FROM_REGEX/TO_TEXT/gce|update' '+q' file1 file2 file3
```

Tip
Bendera "e" di atas mencegah kesalahan "Tidak ada yang cocok" dari merusak pemetaan.

```
$ perl -i -p -e 's/FROM_REGEX/TO_TEXT/g;' file1 file2 file3
```

Dalam contoh perl(1), "-i" adalah untuk penyuntingan di tempat dari setiap berkas target, dan "-p" adalah untuk pengulangan implisit atas semua berkas yang diberikan.

Tip
Penggunaan argumen "-i.bak" bukan "-i" menyimpan setiap berkas asli dengan menambahkan ".bak" ke nama berkasnya. Hal ini membuat pemulihan dari kesalahan lebih mudah untuk substitusi yang kompleks.

Catatan
ed(1) dan vim(1) adalah **BRE**; perl(1) adalah **ERE**.

1.6.5 Mengekstrak data dari tabel berkas teks

Mari kita pertimbangkan berkas teks yang disebut "DPL" dimana beberapa nama pemimpin proyek Debian pra-2004 dan tanggal inisiasi mereka tercantum dalam format yang dipisah spasi.

```
Ian     Murdock   August  1993
Bruce   Perens    April   1996
Ian     Jackson   January 1998
Wichert Akkerman  January 1999
Ben     Collins   April   2001
Bdale   Garbee    April   2002
Martin  Michlmayr March   2003
```

Tip
Lihat "Sejarah Singkat Debian" untuk sejarah kepemimpinan Debian terbaru.

Awk sering digunakan untuk mengekstrak data dari jenis berkas ini.

Misalnya, coba yang berikut ini

```
$ awk '{ print $3 }' <DPL                   # month started
August
April
January
January
April
April
March
$ awk '($1=="Ian") { print }' <DPL          # DPL called Ian
Ian     Murdock   August  1993
Ian     Jackson   January 1998
$ awk '($2=="Perens") { print $3,$4 }' <DPL # When Perens started
April 1996
```

Shell seperti Bash juga dapat digunakan untuk mengurai berkas semacam ini.

Misalnya, coba yang berikut ini

```
$ while read first last month year; do
    echo $month
  done <DPL
... same output as the first Awk example
```

Di sini, perintah `read` bawaan menggunakan karakter dalam ”`$IFS`” (internal field separators/pemisah ruas internal) untuk memecah baris menjadi kata-kata.

Jika Anda mengubah ”`$IFS`” menjadi ”`:`”, Anda dapat mengurai ”`/etc/passwd`” memakai shell dengan baik.

```
$ oldIFS="$IFS"   # save old value
$ IFS=':'
$ while read user password uid gid rest_of_line; do
    if [ "$user" = "bozo" ]; then
      echo "$user's ID is $uid"
    fi
  done < /etc/passwd
bozo's ID is 1000
$ IFS="$oldIFS"   # restore old value
```

(Jika Awk digunakan untuk melakukan yang setara, gunakan ”`FS=':'`” untuk mengatur pemisah ruas.)

IFS juga digunakan oleh shell untuk memecah hasil ekspansi parameter, substitusi perintah, dan ekspansi aritmatika. Ini tidak terjadi dalam kata-kata yang dikutip ganda atau tunggal. Nilai baku IFS adalah *spasi*, *tab*, dan *baris baru* digabungkan.

Berhati-hatilah dalam menggunakan trik IFS shell ini. Hal-hal aneh dapat terjadi, ketika shell menafsirkan beberapa bagian dari skrip sebagai **masukan**.

```
$ IFS=":,"                          # use ":" and "," as IFS
$ echo IFS=$IFS,   IFS="$IFS"       # echo is a Bash builtin
IFS=  , IFS=:,
$ date -R                           # just a command output
Sat, 23 Aug 2003 08:30:15 +0200
$ echo $(date -R)                   # sub shell --> input to main shell
Sat  23 Aug 2003 08 30 36 +0200
$ unset IFS                         # reset IFS to the default
$ echo $(date -R)
Sat, 23 Aug 2003 08:30:50 +0200
```

1.6.6 Cuplikan skrip untuk perintah perpipaan

Skrip berikut melakukan hal-hal baik sebagai bagian dari pipa.

cuplikan skrip (ketik dalam satu baris)	**efek perintah**
`find /usr -print`	cari semua berkas di bawah "`/usr`"
`seq 1 100`	cetak 1 hingga 100
`| xargs -n 1 command`	jalankan perintah berulang kali dengan setiap butir dari pipa sebagai argumennya
`| xargs -n 1 echo`	memecah butir yang dipisah white-space dari pipa ke dalam baris-baris
`| xargs echo`	menggabungkan semua baris dari pipa ke dalam satu baris
`| grep -e regex_pattern`	mengekstrak baris-baris dari pipa yang mengandung *pola_regex*
`| grep -v -e regex_pattern`	mengekstrak baris-baris dari pipa yang tidak mengandung *pola_regex*
`| cut -d: -f3 -`	ekstrak ruas ketiga dari pipa yang dipisahkan oleh "`:`" (berkas passwd dll.)
`| awk '{ print $3 }'`	mengekstrak ruas ketiga dari pipa yang dipisah oleh whitespace
`| awk -F'\t' '{ print $3 }'`	mengekstrak ruas ketiga dari pipa yang dipisah oleh tab
`| col -bx`	menghapus backspace dan mengekspansi tab ke spasi
`| expand -`	mengekspansi tab
`| sort| uniq`	mengurutkan dan menghapus duplikat
`| tr 'A-Z' 'a-z'`	mengonversi huruf besar menjadi huruf kecil
`| tr -d '\n'`	menyambung baris-baris menjadi satu
`| tr -d '\r'`	menghapus CR
`| sed 's/^/# /'`	menambahkan "#" ke awal setiap baris
`| sed 's/\.ext//g'`	menghapus "`.ext`"
`| sed -n -e 2p`	mencetak baris kedua
`| head -n 2 -`	mencetak 2 baris pertama
`| tail -n 2 -`	mencetak 2 baris terakhir

Tabel 1.27: Daftar cuplikan skrip untuk perintah perpipaan

Skrip shell satu baris dapat mengulang atas banyak berkas menggunakan find(1) dan xargs(1) untuk melakukan tugas yang cukup rumit. Lihat Bagian 10.1.5 dan Bagian 9.4.9.

Saat menggunakan mode interaktif shell menjadi terlalu rumit, silakan pertimbangkan untuk menulis skrip shell (lihat Bagian 12.1).

Bab 2

Manajemen paket Debian

Catatan
Bab ini ditulis dengan asumsi rilis stabil terbaru adalah nama kode: `trixie`.
Sumber data dari sistem APT secara kolektif diacu sebagai **daftar sumber** dalam dokumen ini. Ini dapat didefinisikan di mana saja dalam berkas "`/etc/apt/sources.list`", berkas-berkas "`/etc/apt/sources.list.d/*.list`", atau berkas-berkas "`/etc/apt/sources.list.d/*.sources`".

2.1 Prasyarat manajemen paket Debian

2.1.1 Sistem manajemen paket Debian

Debian adalah organisasi sukarelawan yang membangun distribusi **yang konsisten** dari paket biner perangkat lunak bebas yang diprakompilasi dan mendistribusikan mereka dari arsipnya.

Arsip Debian ditawarkan oleh banyak situs cermin jarak jauh untuk akses melalui metode HTTP dan FTP. Ini juga tersedia sebagai CD-ROM/DVD.

Sistem manajemen paket Debian saat ini yang dapat memanfaatkan semua sumber daya ini adalah Advanced Packaging Tool (APT).

Sistem manajemen paket Debian, **bila digunakan dengan benar**, menawarkan pengguna untuk memasang **set paket biner yang konsisten** ke sistem dari arsip. Saat ini, ada 78072 paket yang tersedia untuk arsitektur amd64.

Sistem manajemen paket Debian memiliki sejarah yang kaya dan banyak pilihan untuk front end program pengguna dan metode akses arsip back end yang akan digunakan. Saat ini, kami merekomendasikan hal-hal berikut.

- apt(8) untuk semua operasi baris perintah interaktif, termasuk instalasi paket, penghapusan, dan dist-upgrade.
- apt-get(8) untuk memanggil sistem manajemen paket Debian dari skrip. Ini juga merupakan pilihan fallback ketika `apt` tidak tersedia (sering dengan sistem Debian yang lebih tua).
- aptitude(8) bagi antarmuka teks interaktif untuk mengelola paket yang dipasang dan untuk mencari paket yang tersedia.

2.1.2 Konfigurasi paket

Berikut adalah beberapa poin penting untuk konfigurasi paket pada sistem Debian.

paket	popcon	ukuran	deskripsi
dpkg	V:906, I:1000	4944	sistem manajemen paket tingkat rendah untuk Debian (berbasis berkas)
apt	V:895, I:1000	4956	Front-end APT untuk mengelola paket dengan CLI: apt/apt-get/apt-cache
aptitude	V:31, I:160	4621	Front-end APT untuk mengelola paket secara interaktif dengan konsol layar penuh: aptitude(8)
tasksel	V:38, I:985	351	Front-end APT untuk memasang tugas yang dipilih: tasksel(8)
unattended-upgrades	V:130, I:182	320	paket peningkatan bagi APT untuk mengaktifkan instalasi otomatis peningkatan keamanan
gnome-software	V:158, I:263	4732	Pusat Perangkat Lunak untuk GNOME (front-end APT GUI)
synaptic	V:31, I:269	7264	manajer paket grafis (front-end APT GTK)
apt-utils	V:420, I:998	1181	Program utilitas APT: apt-extracttemplates(1), apt-ftparchive(1), dan apt-sortpkgs(1)
apt-listchanges	V:403, I:894	562	alat pemberitahuan riwayat perubahan paket
apt-listbugs	V:4.7, I:6.5	512	menampilkan daftar bug kritis sebelum setiap instalasi APT
apt-file	V:14, I:53	76	Utilitas pencarian paket APT — antarmuka baris perintah
apt-rdepends	V:0.4, I:4.2	39	mencantumkan dependensi paket secara rekursif

Tabel 2.1: Daftar alat manajemen paket Debian

- Untuk lingkungan Desktop modern, mem-boot ulang sistem setelah perubahan konfigurasi paket dan peningkatan paket adalah ide yang baik untuk memastikan bahwa sistem berfungsi secara benar.
- Konfigurasi manual oleh administrator sistem dihormati. Dengan kata lain, sistem konfigurasi paket tidak membuat konfigurasi yang mengganggu demi kenyamanan.
- Setiap paket dilengkapi dengan skrip konfigurasi sendiri dengan antarmuka pengguna standar yang disebut debconf(7) untuk membantu proses instalasi awal paket.
- Para Pengembang Debian mencoba yang terbaik untuk membuat pengalaman peningkatan Anda sempurna dengan skrip konfigurasi paket.
- Fungsionalitas penuh perangkat lunak yang dikemas tersedia untuk administrator sistem. Tetapi yang memiliki risiko keamanan dinonaktifkan dalam instalasi baku.
- Jika Anda mengaktifkan layanan secara manual dengan beberapa risiko keamanan, Anda bertanggung jawab atas pengurungan (containment) risiko.
- Konfigurasi esoterik dapat diaktifkan secara manual oleh administrator sistem. Hal ini dapat membuat gangguan dengan program pembantu generik populer untuk konfigurasi sistem.

2.1.3 Tindakan pencegahan dasar



Awas
Jangan menginstall paket dari campuran acak keluarga. Ini mungkin merusak konsistensi paket yang membutuhkan pengetahuan manajemen sistem yang mendalam, seperti ABI kompiler, versi pustaka, fitur interpreter, dll.

Administrator sistem Debian pemula harus tetap memakai rilis Debian **`stable`** sambil hanya menerapkan pembaruan keamanan. Sampai Anda memahami sistem Debian dengan sangat baik, Anda mesti mengikuti pencegahan berikut.

- Jangan sertakan **`testing`** atau **`unstable`** dalam **daftar sumber**.

- Jangan mencampur Debian standar dengan arsip non-Debian lainnya seperti Ubuntu di **daftar sumber**.
- Jangan membuat "`/etc/apt/preferences`".
- Jangan mengubah perilaku baku alat manajemen paket melalui berkas konfigurasi tanpa mengetahui dampak penuhnya.
- Jangan memasang paket acak dengan "`dpkg -i paket_acak`".
- Jangan pernah memasang paket acak dengan "`dpkg --force-all -i random_package`".
- Jangan menghapus atau mengubah berkas di "`/var/lib/dpkg/`".
- Jangan menimpa sistem berkas dengan memasang program perangkat lunak yang langsung dikompilasi dari sumber.
  - Pasang mereka ke dalam "`/usr/local`" atau "`/opt`", jika diperlukan.

Efek tidak kompatibel yang disebabkan oleh tindakan pencegahan di atas pada sistem manajemen paket Debian dapat menyebabkan sistem Anda tidak bisa dipakai.

Administrator sistem Debian yang serius yang menjalankan server-server dengan misi penting, harus menerapkan tindakan pencegahan ekstra.

- Jangan memasang paket apa pun termasuk pembaruan keamanan dari Debian tanpa mengujinya secara menyeluruh dengan konfigurasi khusus Anda dalam kondisi aman.
  - Anda sebagai administrator sistem bertanggung jawab atas sistem Anda pada akhirnya.
  - Sejarah stabilitas panjang sistem Debian bukanlah jaminan dengan sendirinya.

2.1.4 Hidup dengan peningkatan abadi



Perhatian
Untuk **server produksi** Anda, disarankan keluarga `stable` dengan pembaruan keamanan. Hal yang sama dapat dikatakan untuk PC desktop di mana Anda dapat menghabiskan upaya administrasi yang terbatas.

Terlepas dari peringatan saya di atas, saya tahu banyak pembaca dokumen ini mungkin ingin menjalankan keluarga `testing` atau `unstable` yang lebih baru.

Enlightenment dengan yang berikut menyelamatkan seseorang dari perjuangan karma abadi atas neraka peningkatan dan membiarkan dia mencapai sorga Debian.

Daftar ini ditargetkan untuk lingkungan Desktop **yang dikelola sendiri**.

- Gunakan keluarga `testing` karena praktis itu adalah rilis bergulir yang secara otomatis dikelola oleh infrastruktur QA arsip Debian seperti integrasi berkelanjutan Debian, praktik hanya mengunggah sumber, dan pelacakan transisi pustaka. Paket-paket di keluarga `testing` diperbarui cukup sering untuk menawarkan semua fitur terbaru.
- Mengatur nama kode yang sesuai dengan keluarga `testing` ("`forky`" selama `trixie` sebagai siklus rilis `stable`) dalam **daftar sumber**.
- Perbarui nama kode ini secara manual di "**daftar sumber**" ke yang baru hanya setelah menilai situasi sendiri selama sekitar satu bulan setelah rilis keluarga mayor. Milis pengguna Debian dan pengembang adalah sumber informasi yang baik untuk ini juga.

Penggunaan keluarga `unstable` tidak dianjurkan. Keluarga `unstable` **baik untuk debugging paket** sebagai pengembang tetapi cenderung mengekspos Anda untuk risiko yang tidak perlu untuk penggunaan Desktop normal. Meskipun keluarga sistem Debian `unstable` terlihat sangat stabil untuk sebagian besar waktu, ada beberapa masalah paket dan beberapa di antaranya tidak begitu sepele untuk diselesaikan.

Berikut adalah beberapa ide tindakan pencegahan dasar untuk memastikan pemulihan cepat dan mudah dari bug dalam paket Debian.

- Jadikan sistem penyelamat tersedia dengan mengikuti Bagian 3.2.
- Buat sistem **boot ganda** dengan memasang sistem Debian keluarga `stable` ke partisi lain
- Pertimbangkan untuk memasang `apt-listbugs` untuk memeriksa informasi Debian Bug Tracking System (BTS) sebelum peningkatan
- Pelajari infrastruktur sistem paket yang cukup untuk mengatasi masalah



Perhatian
Jika Anda tidak dapat melakukan salah satu dari tindakan pencegahan ini, Anda mungkin tidak siap untuk keluarga `testing` dan `unstable`.

2.1.5 Dasar-dasar arsip Debian

Tip
Kebijakan resmi arsip Debian didefinisikan di Manual Kebijakan Debian, Bab 2 - Arsip Debian.

Mari kita lihat arsip Debian dari perspektif pengguna sistem.

Untuk pengguna sistem, arsip Debian diakses menggunakan sistem APT.

Sistem APT menentukan sumber datanya sebagai **daftar sumber** dan itu dijelaskan dalam sources.list(5).

Untuk sistem `trixie` dengan akses HTTP biasa, **daftar sumber** disediakan dalam gaya deb822 modern di "`/etc/apt/sou`
sebagai berikut:

```
Types: deb deb-src
URIs: http://deb.debian.org/debian/
Suites: trixie trixie-updates
Components: main non-free-firmware contrib non-free

Types: deb deb-src
URIs: http://security.debian.org/debian-security/
Suites: trixie-security
Components: main non-free-firmware contrib non-free
```

Tip
Jika **daftar sumber** disediakan dalam gaya satu baris lama yang sudah usang di berkas "`/etc/apt/sources.list`" atau "`/etc/apt/sources.list.d/*.list`", perbarui dengan:

```
$ sudo apt modernize-sources
```

Poin-poin utama dari **daftar sumber** dalam gaya deb822 adalah sebagai berikut.

- Berkas definisinya ada dalam berkas-berkas "`/etc/apt/sources.list.d/*.sources`".
- Setiap blok baris dipisahkan oleh satu baris kosong mendefinisikan sumber data untuk sistem APT.
- Bait "`Types:`" mendefinisikan daftar jenis seperti "`deb`" dan "`deb-src`".
- Bait "`URIs:`" mendefinisikan daftar akar URIs dari arsip Debian.
- Bait "`Suites:`" mendefinisikan daftar nama distribusi menggunakan nama keluarga atau nama kode.
- Bait "`Components:`" mendefinisikan daftar nama wilayah arsip yang valid dari arsip Debian.

Definisi untuk "`deb-src`" dapat dengan aman dihilangkan jika itu hanya untuk `aptitude` yang tidak mengakses sumber data meta terkait. Ini mempercepat pembaruan data meta arsip.

URL bisa berupa "`https://`", "`http://`", "`ftp://`", "`file://`", ...

Baris yang diawali dengan "#" adalah komentar dan diabaikan.

Di sini, saya cenderung menggunakan nama kode "`trixie`" atau "`forky`" bukan nama keluarga "`stable`" atau "`testing`" untuk menghindari kejutan ketika `stable` dirilis.

Tip
Jika "`sid`" digunakan dalam contoh di atas, bukan "`trixie`", baris "`deb: http://security.debian.org/ ...`" atau konten deb822 yang setara untuk pembaruan keamanan dalam **daftar sumber** tidak diperlukan. Ini karena tidak ada arsip pembaruan keamanan untuk "`sid`" (`unstable`).

Berikut adalah daftar URL situs arsip Debian dan nama keluarga atau nama kode yang digunakan dalam berkas konfigurasi setelah rilis `trixie`.

URL arsip	nama keluarga	nama kode	tujuan repositori
http://deb.debian.org/debian/	`stable`	`trixie`	Rilis `stable` kuasi-statik setelah pemeriksaan yang ekstensif
http://deb.debian.org/debian/	`stable-updates`	`trixie-updates`	Pembaruan mendesak bagi keluarga `stable` (penting)
http://security.debian.org/debian-security/	`stable-security`	`trixie-security`	Pembaruan keamanan bagi keluarga `stable` (penting)
http://deb.debian.org/debian/	`testing`	`forky`	Rilis `testing` dinamis setelah pemeriksaan yang layak dan menunggu singkat
http://deb.debian.org/debian/	`unstable`	`sid`	Rilis `unstable` dinamis setelah pemeriksaan minimum dan tanpa menunggu
http://deb.debian.org/debian/	`experimental`	`T/T`	Eksperimen pra-rilis oleh para pengembang (opsional, hanya untuk pengembang)
http://deb.debian.org/debian/	`stable-proposed-updates`	`trixie-proposed-updates`	Perbaikan untuk titik `stable` berikutnya (opsional)
http://deb.debian.org/debian/	`stable-backports`	`trixie-backports`	Koleksi acak dari paket yang dikompilasi ulang kebanyakan dari rilis `testing` (opsional)
http://security.debian.org/debian-security/	`testing-security`	`forky-security`	Ini tidak didukung secara aktif maupun dipakai oleh tim keamanan

Tabel 2.2: Daftar situs arsip Debian



Perhatian
Hanya rilis **`stable`** murni dengan pembaruan keamanan yang memberikan stabilitas terbaik. Menjalankan sebagian besar rilis **`stable`** dicampur dengan beberapa paket dari **`testing`** atau rilis **`unstable`** lebih berrisiko daripada menjalankan rilis murni **`unstable`** untuk ketidakcocokan versi pustaka dll. Jika Anda benar-benar memerlukan versi terbaru dari beberapa program di bawah rilis **`stable`** silakan gunakan paket dari layanan stable-updates dan backports (lihat Bagian 2.7.4). Layanan ini harus digunakan dengan ekstra hati-hati.



Perhatian
Anda pada dasarnya harus mencantumkan hanya satu keluarga `stable`, `testing`, atau `unstable` di baris "`deb`". Jika Anda mencantumkan kombinasi keluarga `stable`, `testing`, dan `unstable` di baris "`deb`", program APT melambat sementara hanya arsip terbaru yang efektif. Beberapa daftar masuk akal untuk ini ketika berkas "`/etc/apt/preferences`" digunakan dengan tujuan yang jelas (lihat Bagian 2.7.7).

Tip
Untuk sistem Debian dengan keluarga `stable`, adalah ide yang baik untuk memasukkan baris dengan "`http://security.debian.org/`" dalam "**daftar sumber**" untuk mengaktifkan pembaruan keamanan seperti pada contoh di atas.

Catatan
Bug keamanan untuk arsip `stable` diperbaiki oleh tim keamanan Debian. Kegiatan ini cukup ketat dan dapat diandalkan. Untuk arsip `testing` mungkin diperbaiki oleh tim keamanan testing Debian. Untuk beberapa alasan, kegiatan ini tidak seketat itu untuk `stable` dan Anda mungkin perlu menunggu migrasi paket `unstable` yang diperbaiki ke arsip `testing`. Untuk arsip `unstable` diperbaiki oleh pengelola individu. Paket-paket `unstable` yang dipelihara secara aktif biasanya dalam kondisi yang cukup baik dengan memanfaatkan perbaikan keamanan hulu terbaru. Lihat FAQ keamanan Debian untuk cara Debian menangani bug-bug keamanan.

area	banyaknya paket	kriteria komponen paket
`main`	76587	Patuh DFSG dan tidak ada ketergantungan terhadap `non-free`
`non-free-firmware`	78	tidak patuh DFSG, firmware diperlukan untuk pengalaman instalasi sistem yang cukup beralasan
`contrib`	362	Patuh DFSG tetapi memiliki ketergantungan ke `non-free`
`non-free`	1045	tidak patuh DFSG dan tidak dalam `non-free-firmware`

Tabel 2.3: Daftar area arsip Debian

Di sini jumlah paket di atas adalah untuk arsitektur amd64. Area `main` menyediakan sistem Debian (lihat Bagian 2.1.6).

Organisasi arsip Debian dapat dipelajari paling baik dengan mengarahkan peramban Anda ke setiap URL arsip ditambah dengan `dists` atau `pool`.

Distribusi disebut dengan dua cara, keluarga atau nama kode. Kata distribusi secara alternatif digunakan sebagai sinonim untuk keluarga dalam banyak dokumentasi. Hubungan antara keluarga dan nama kode dapat diringkas sebagai berikut.

Riwayat nama kode dijelaskan dalam DEBIAN FAQ: 6.2.1 Nama kode lain mana yang telah digunakan di masa lalu?

Dalam terminologi arsip Debian yang lebih ketat, kata "section" secara khusus digunakan untuk kategorisasi paket oleh area aplikasi. (Meskipun, kata "main section" kadang-kadang dapat digunakan untuk menggambarkan area arsip Debian bernama "main".)

Kapan	suite = stable	suite = testing	suite = unstable
setelah rilis trixie	codename = trixie	codename = forky	codename = sid
setelah rilis forky	codename = forky	codename = duke	codename = sid

Tabel 2.4: Hubungan antara keluarga dan nama kode

Setiap kali pengunggahan baru, dilakukan oleh pengembang Debian (DD) ke arsip unstable (melalui pemrosesan incoming), DD perlu memastikan paket yang diunggah kompatibel dengan set paket terbaru dalam arsip unstable terakhir.

Jika DD merusak kompatibilitas ini dengan sengaja untuk peningkatan pustaka penting dll, biasanya ada pengumuman ke milis debian-devel dll.

Sebelum satu set paket dipindahkan oleh skrip pemeliharaan arsip Debian dari arsip unstable ke arsip testing, skrip pemeliharaan arsip tidak hanya memeriksa kematangan (sekitar 2-10 hari) dan status laporan bug RC untuk paket tetapi juga mencoba untuk memastikan mereka kompatibel dengan set paket terbaru dalam arsip testing. Proses ini membuat arsip testing sangat terkini dan dapat digunakan.

Melalui proses pembekuan arsip bertahap yang dipimpin oleh tim rilis, arsip testing matang untuk membuatnya benar-benar konsisten dan bebas bug dengan beberapa intervensi manual. Kemudian rilis stable baru dibuat dengan menetapkan nama kode untuk arsip testing lama ke arsip stable baru dan membuat nama kode baru untuk arsip testing baru. Isi awal dari arsip testing baru persis sama dengan arsip stable yang baru saja dirilis.

Baik arsip unstable dan testing mungkin menderita gangguan sementara karena beberapa faktor.

- Pengunggahan paket rusak ke arsip (kebanyakan untuk unstable)
- Penundaan menerima paket baru ke arsip (sebagian besar untuk unstable)
- Masalah waktu sinkronisasi arsip (baik untuk testing dan unstable)
- Intervensi manual ke arsip seperti penghapusan paket (lebih untuk testing) dll.

Jadi jika Anda pernah memutuskan untuk menggunakan arsip ini, Anda harus dapat memperbaiki atau mengatasi gangguan semacam ini.



Perhatian
Selama sekitar beberapa bulan setelah rilis stable baru, sebagian besar pengguna desktop harus menggunakan arsip stable dengan pembaruan keamanannya bahkan jika mereka biasanya menggunakan arsip unstable atau testing. Untuk masa transisi ini, arsip unstable dan testing tidak baik bagi kebanyakan orang. Sistem Anda sulit untuk tetap dalam kondisi kerja yang baik dengan arsip unstable karena menderita lonjakan peningkatan mayor untuk paket-paket inti. Arsip testing juga tidak berguna karena berisi sebagian besar konten yang sama dengan arsip stable tanpa dukungan keamanannya (Debian testing-security-announce 2008-12). Setelah satu bulan atau lebih, arsip unstable atau testing mungkin menjadi berguna jika Anda berhati-hati.

Tip
Saat melacak arsip testing, masalah yang disebabkan oleh paket yang dihapus biasanya diatasi dengan memasang paket yang sesuai dari arsip unstable yang diunggah untuk perbaikan bug.

Lihat Manual Kebijakan Debian untuk definisi arsip.

- "Bagian"
- "Prioritas"
- "Sistem dasar"
- "Paket esensial"

2.1.6 Debian adalah perangkat lunak 100% bebas

Debian adalah perangkat lunak 100% bebas karena yang berikut ini:

- Debian hanya memasang perangkat lunak bebas secara baku untuk menghormati kebebasan pengguna.
- Debian hanya menyediakan perangkat lunak bebas di `main`.
- Debian merekomendasikan untuk hanya menjalankan perangkat lunak bebas dari `main`.
- Tidak ada paket `main` yang bergantung atau merekomendasikan paket dalam `non-free`, `non-free-firmware`, maupun `contrib`.

Beberapa orang bertanya-tanya apakah 2 fakta berikut bertentangan atau tidak.

- ”Debian akan tetap 100% bebas”. (Istilah pertama dari Kontrak Sosial Debian)
- Server Debian mewadahi beberapa paket `non-free-firmware`, `non-free`, dan `contrib`.

Ini tidak bertentangan, karena hal-hal berikut.

- Sistem Debian adalah 100% bebas dan paketnya diwadahi oleh server Debian di area `main`.
- Paket di luar sistem Debian diwadahi oleh server Debian di area `non-free`, `non-free-firmware`, dan `contrib`.

Ini dijelaskan secara tepat dalam ketentuan ke-4 dan ke-5 dari Kontrak Sosial Debian:

- Prioritas kami adalah pengguna kami dan perangkat lunak bebas
  - Kami akan dipandu oleh kebutuhan pengguna kami dan komunitas perangkat lunak bebas. Kami akan menempatkan kepentingan mereka terlebih dahulu dalam prioritas kami. Kami akan mendukung kebutuhan pengguna kami untuk beroperasi di berbagai jenis lingkungan komputasi. Kami tidak akan keberatan dengan karya non-bebas yang dimaksudkan untuk digunakan pada sistem Debian, atau mencoba untuk membebankan biaya kepada orang-orang yang membuat atau menggunakan karya tersebut. Kami akan mengizinkan orang lain untuk membuat distribusi yang berisi sistem Debian dan karya lainnya, tanpa biaya apa pun dari kami. Sebagai kelanjutan dari tujuan ini, kami akan menyediakan sistem terpadu dengan bahan-bahan berkualitas tinggi tanpa batasan hukum yang akan mencegah penggunaan sistem tersebut.
- Karya yang tidak memenuhi standar perangkat lunak bebas kami
  - Kami mengakui bahwa beberapa pengguna kami memerlukan penggunaan karya yang tidak sesuai dengan Pedoman Perangkat Lunak Bebas Debian. Kami telah menciptakan area ”`non-free`”, ”`non-free-firmware`”, dan ”`contrib`” dalam arsip kami untuk karya-karya ini. Paket-paket di area ini bukan bagian dari sistem Debian, meskipun telah dikonfigurasi untuk digunakan dengan Debian. Kami mendorong produsen CD untuk membaca lisensi paket di area ini dan menentukan apakah mereka dapat mendistribusikan paket pada CD mereka. Jadi, meskipun karya non-free bukan bagian dari Debian, kami mendukung penggunaannya dan menyediakan infrastruktur untuk paket-paket non-free (seperti sistem pelacakan bug dan milis kami). Media resmi Debian dapat menyertakan firmware yang sebenarnya bukan dari sistem Debian untuk memfungsikan penggunaan Debian dengan perangkat keras yang memerlukan firmware seperti itu.

Catatan
Teks sebenarnya dari syarat ke-5 dalam Kontrak Sosial Debian 1.2 saat ini sedikit berbeda dari teks di atas. Deviasi editorial ini disengaja untuk membuat dokumen pengguna ini konsisten tanpa mengubah konten sebenarnya dari Kontrak Sosial.

Pengguna harus menyadari risiko menggunakan paket-paket di area `non-free`, `non-free-firmware`, dan `contrib`:

- ketiadaan kebebasan untuk paket perangkat lunak tersebut
- ketiadaan dukungan dari Debian pada paket perangkat lunak tersebut (Debian tidak dapat mendukung perangkat lunak dengan benar tanpa memiliki akses ke kode sumbernya.)
- kontaminasi sistem Debian 100% bebas Anda

Debian Free Software Guidelines (Panduan Perangkat Lunak Bebas Debian) adalah standar perangkat lunak bebas untuk Debian. Debian menafsirkan ”perangkat lunak” dalam lingkup terluas termasuk dokumen, firmware, logo, dan data karya seni dalam paket. Hal ini membuat standar perangkat lunak bebas Debian sangat ketat.

Paket `non-free`, `non-free-firmware`, dan `contrib` umumnya termasuk paket yang dapat didistribusikan secara bebas dari jenis berikut:

- Paket dokumen di bawah GNU Free Documentation License (Lisensi Dokumentasi Bebas GNU) dengan bagian invarian seperti yang untuk GCC dan Make. (sebagian besar ditemukan di bagian `non-free/doc`.)
- Paket-paket firmware yang berisi data biner tanpa sumber seperti yang terdaftar dalam Bagian 9.10.5 sebagai `non-free-firmware`. (sebagian besar ditemukan di bagian `non-free-firmware/kernel`.)
- Paket=paket permainan dan fonta dengan pembatasan penggunaan komersial dan/atau modifikasi konten.

Harap dicatat bahwa cacah paket `non-free`, `non-free-firmware`, dan `contrib` kurang dari 2% dari paket-paket `main`. Memfungsikan akses ke area `non-free`, `non-free-firmware`, dan `contrib` tidak mengaburkan sumber paket. Penggunaan aptitude(8) layar penuh interaktif memberi Anda visibilitas penuh dan kontrol atas paket apa yang dipasang dari area mana untuk menjaga sistem Anda sebebas yang Anda inginkan.

2.1.7 Dependensi paket

Sistem Debian menawarkan serangkaian paket biner yang konsisten melalui mekanisme deklarasi ketergantungan biner versinya di ruas berkas control. Berikut adalah sedikit definisi yang disederhanakan untuk mereka.

- ”Depends”
  - Ini menyatakan ketergantungan mutlak dan semua paket yang tercantum dalam ruas ini harus dipasang pada saat yang sama atau sebelumnya.
- ”Pre-Depends”
  - Ini seperti Depends, kecuali bahwa itu memerlukan instalasi lengkap dari paket yang terdaftar sebelumnya.
- ”Recommends”
  - Ini menyatakan ketergantungan yang kuat, tetapi tidak mutlak. Sebagian besar pengguna tidak menginginkan paket kecuali semua paket yang tercantum di ruas ini dipasang.
- ”Suggests”
  - Ini menyatakan ketergantungan yang lemah. Banyak pengguna paket ini mungkin mendapat manfaat dari memasang paket yang tercantum di ruas ini tetapi dapat memiliki fungsi yang wajar tanpa mereka.
- ”Enhances”
  - Ini menyatakan ketergantungan lemah seperti Suggests tetapi bekerja ke arah yang berlawanan.
- ”Breaks”
  - Ini menyatakan ketidakcocokan paket biasanya dengan beberapa spesifikasi versi. Umumnya resolusinya adalah untuk meningkatkan semua paket yang tercantum dalam ruas ini.
- ”Conflicts”

  - Ini menyatakan ketidakcocokan mutlak. Semua paket yang tercantum di ruas ini harus dihapus untuk memasang paket ini.

- ”Replaces”
  - Ini dinyatakan ketika berkas yang dipasang oleh paket ini menggantikan berkas dalam paket yang terdaftar.
- ”Provides”
  - Ini dinyatakan ketika paket ini menyediakan semua berkas dan fungsionalitas dalam paket yang terdaftar.

Catatan
Harap dicatat bahwa mendefinisikan ”Provides”, ”Conflicts”, dan ”Replaces” secara bersamaan ke paket virtual adalah konfigurasi yang waras. Ini memastikan bahwa hanya satu paket nyata yang menyediakan paket virtual ini yang dapat dipasang pada satu waktu.

Definisi resmi termasuk ketergantungan sumber dapat ditemukan dalam Manual Kebijakan: Bab 7 - Mendeklarasikan hubungan antar paket.

2.1.8 Alur kejadian manajemen paket

Berikut adalah ringkasan dari alur kejadian yang disederhanakan dari manajemen paket oleh APT.

- **Pembaruan** (”`apt update`”, ”`aptitude update`”, atau ”`apt-get update`”):
  1. Mengambil metadata arsip dari arsip jarak jauh
  2. Merekonstruksi dan memperbarui metadata lokal untuk digunakan oleh APT
- **Peningkatan** (”`apt upgrade`”dan”`apt full-upgrade`”, atau”`aptitude safe-upgrade`” dan ”`aptitude full-up`
  atau ”`apt-get upgrade`” dan ”`apt-get dist-upgrade`”):
  1. Memilih versi kandidat yang biasanya merupakan versi terbaru yang tersedia untuk semua paket yang dipasang (lihat Bagian 2.7.7 untuk pengecualian)
  2. Membuat resolusi ketergantungan paket
  3. Mengambil paket biner yang dipilih dari arsip jarak jauh jika versi kandidat berbeda dari versi terpasang
  4. Membuka kemasan paket biner yang diambil
  5. Jalankan skrip **preinst**
  6. Memasang berkas biner
  7. Jalankan skrip **postinst**
- **Memasang** (”`apt install …`”, `aptitude install …`”, atau ”`apt-get install …`”):
  1. Memilih paket yang tercantum di baris perintah
  2. Membuat resolusi ketergantungan paket
  3. Mengambil paket biner yang dipilih dari arsip jarak jauh
  4. Membuka kemasan paket biner yang diambil
  5. Jalankan skrip **preinst**
  6. Memasang berkas biner
  7. Jalankan skrip **postinst**
- **Menghapus** (”`apt remove …`”, ”`aptitude remove …`”, atau ”`apt-get remove …`”):
  1. Memilih paket yang tercantum di baris perintah

2. Membuat resolusi ketergantungan paket
3. Jalankan skrip **prerm**
4. Menghapus berkas yang dipasang **kecuali** berkas konfigurasi
5. Jalankan skrip **postrm**

- **Pembersihan** (”`apt purge`”, ”`aptitude purge` …”, atau ”`apt-get purge` …”):
  1. Memilih paket yang tercantum di baris perintah
  2. Membuat resolusi ketergantungan paket
  3. Jalankan skrip **prerm**
  4. Menghapus berkas yang dipasang **termasuk** berkas konfigurasi
  5. Jalankan skrip **postrm**

Di sini, saya sengaja melewatkan detail teknis demi gambaran besar.

2.1.9 Tanggapan pertama terhadap masalah manajemen paket

Anda harus membaca dokumentasi resmi yang bagus. Dokumen pertama yang dibaca adalah spesifik Debian ”`/usr/share/doc/`*`nama_paket`*`/README.Debian`”. Dokumentasi lain dalam ”`/usr/share/doc/`*`nama_paket`*`/`” harus dikonsultasikan juga. Jika Anda mengatur shell sebagai Bagian 1.4.2, ketikkan yang berikut ini.

```
$ cd package_name
$ pager README.Debian
$ mc
```

Anda mungkin perlu memasang paket dokumentasi yang sesuai yang memiliki nama dengan akhiran ”`-doc`” untuk informasi terperinci.

Jika Anda mengalami masalah dengan paket tertentu, pastikan untuk terlebih dahulu memeriksa situs Debian bug tracking system (BTS).

situs web	**perintah**
Halaman beranda sistem pelacakan bug Debian (Bug Tracking System/BTS)	`sensible-browser "https://bugs.debian.org/"`
Laporan bug dari nama paket yang dikenal	`sensible-browser "https://bugs.debian.org/`*`package_name`*`"`
Laporan bug dari nomor bug yang diketahui	`sensible-browser "https://bugs.debian.org/`*`bug_number`*`"`

Tabel 2.5: Daftar situs web kunci untuk menyelesaikan masalah dengan paket tertentu

Cari di Google dengan kata-kata pencarian termasuk ”`site:debian.org`”, ”`site:wiki.debian.org`”, ”`site:lists.d`
dll.

Saat Anda mengajukan laporan bug, silakan gunakan perintah reportbug(1).

2.1.10 Cara memilih paket Debian

Ketika Anda menemukan lebih dari 2 paket serupa dan bertanya-tanya mana yang harus dipasang tanpa upaya ”coba-coba”, Anda harus menggunakan **akal sehat**. Saya menganggap poin-poin berikut adalah indikasi yang baik dari paket pilihan.

- Esensial: yes > no

- Wilayah: main > contrib > non-free
- Priority: required > important > standard > optional > extra
- Tugas: paket yang tercantum dalam tugas-tugas seperti ”Lingkungan desktop”
- Paket yang dipilih oleh paket dependensi (mis., `gcc-16` oleh `gcc`)
- Popcon: lebih tinggi dalam perolehan suara dan cacah instalasi
- Changelog: pembaruan rutin oleh pengelola
- BTS: Tidak ada bug RC (tidak ada bug kritis, grave, dan serius)
- BTS: pengelola responsif terhadap laporan bug
- BTS: cacah bug yang lebih tinggi baru-baru ini diperbaiki
- BTS: cacah bug non-wishlist yang tersisa

Debian yang berupa proyek sukarela dengan model pengembangan terdistribusi, arsipnya berisi banyak paket dengan fokus dan kualitas yang berbeda. Anda harus membuat keputusan sendiri apa yang harus dilakukan dengan mereka.

2.1.11 Bagaimana menghadapi persyaratan yang bertentangan

Apa pun keluarga sistem Debian yang mungkin Anda putuskan untuk digunakan, Anda mungkin masih ingin menjalankan versi program yang tidak tersedia di keluarga tersebut. Bahkan jika Anda menemukan paket biner program tersebut di keluarga Debian lain atau dalam sumber non-Debian lainnya, persyaratan mereka dapat bertentangan dengan sistem Debian saat ini.

Walaupun Anda bisa men-tweak sistem manajemen paket dengan teknik **apt-pinning** dsb. seperti yang diuraikan dalam Bagian 2.7.7 untuk memasang paket biner tak selaras seperti itu, pendekatan tweaking seperti itu hanya memiliki use case terbatas karena mereka mungkin merusak program itu dan sistem Anda.

Sebelum secara brutal memasang paket tak selaras seperti itu, Anda harus mencari semua alternatif solusi teknis yang lebih aman yang kompatibel dengan sistem Debian Anda saat ini.

- Pasang program-program seperti itu memakai paket biner hulu ter-sandbox yang sesuai (lihat Bagian 7.7).
  - Kebanyakan program GUI seperti aplikasi LibreOffice dan GNOME tersedia sebagai paket Flatpak, Snap, atau AppImage.
- Buat chroot atau lingkungan serupa dan jalankan program-program seperti itu di dalamnya (lihat Bagian 9.11).
  - Perintah CLI dapat dengan mudah dieksekusi di bawah chroot-nya yang kompatibel (lihat Bagian 9.11.4).
  - Beberapa lingkungan desktop lengkap dapat dicoba dengan mudah tanpa reboot (lihat Bagian 9.11.5).
- Bangun versi yang diinginkan dari paket biner yang kompatibel dengan sistem Debian Anda saat ini oleh Anda sendiri.
  - Ini adalah suatu tugas yang tidak mudah (lihat Bagian 2.7.13).

2.2 Operasi manajemen paket dasar

Operasi manajemen paket berbasis repositori pada sistem Debian dapat dilakukan oleh banyak alat manajemen paket berbasis APT yang tersedia pada sistem Debian. Di sini, kami menjelaskan 3 alat manajemen paket dasar: `apt`, `apt-get` / `apt-cache`, dan `aptitude`.

Untuk operasi manajemen paket yang melibatkan instalasi paket atau pembaruan metadata paket, Anda harus memiliki hak istimewa root.

2.2.1 `apt` vs. `apt-get` / `apt-cache` vs. `aptitude`

Meskipun `aptitude` adalah alat interaktif yang sangat bagus yang terutama digunakan penulis, Anda harus mengetahui beberapa fakta peringatan:

- Perintah `aptitude` tidak dianjurkan untuk meningkatkan sistem rilis ke rilis pada sistem Debian `stable` setelah rilis baru.
  - Penggunaan "`apt full-upgrade`" atau "`apt-get dist-upgrade`" direkomendasikan untuk itu. Lihat Bug #411280.
- Perintah `aptitude` kadang menyarankan penghapusan paket massal untuk peningkatan sistem pada sistem Debian `testing` atau `unstable`.
  - Situasi ini telah membuat takut banyak administrator sistem. Jangan panik.
  - Hal ini tampaknya sebagian besar disebabkan oleh ketidaksepakatan versi di antara paket-paket yang tergantung atau direkomendasikan oleh paket meta seperti `gnome-core`.
  - Hal ini dapat diselesaikan dengan memilih "Batalkan tindakan yang tertunda" di menu perintah `aptitude`, keluar dari `aptitude`, dan menggunakan "`apt full-upgrade`".

Perintah `apt-get` dan `apt-cache` adalah alat manajemen paket berbasis APT yang paling **dasar**.

- `apt-get` dan `apt-cache` hanya menawarkan antarmuka pengguna baris perintah.
- `apt-get` paling cocok untuk **peningkatan sistem mayor** antara rilis, dll.
- `apt-get` menawarkan pengurai ketergantungan paket yang **kuat**.
- `apt-get` kurang menuntut sumber daya perangkat keras. Ini mengkonsumsi lebih sedikit memori dan berjalan lebih cepat.
- `apt-cache` menawarkan pencarian berbasis regex **standar** pada nama dan deskripsi paket.
- `apt-get` dan `apt-cache` dapat mengelola beberapa versi paket menggunakan `/etc/apt/preferences` tetapi cukup rumit.

Perintah `apt` adalah antarmuka baris perintah tingkat tinggi untuk manajemen paket. Ini pada dasarnya adalah pembungkus `apt-get`, `apt-cache`, dan perintah-perintah serupa, awalnya dimaksudkan sebagai antarmuka pengguna akhir dan memungkinkan beberapa opsi yang lebih cocok untuk penggunaan interaktif secara default.

- `apt` menyediakan bilah kemajuan yang ramah saat memasang paket menggunakan `apt install`.
- `apt` akan **menghapus** paket `.deb` yang disinggahkan secara baku setelah instalasi paket yang diunduh dengan sukses.

Tip
Pengguna disarankan untuk menggunakan perintah apt(8) baru untuk penggunaan **interaktif** dan menggunakan perintah apt-get(8) dan apt-cache(8) dalam skrip shell.

Perintah `aptitude` adalah alat manajemen paket berbasis APT yang paling **serbaguna**.

- `aptitude` menawarkan antarmuka pengguna teks interaktif layar penuh.
- `aptitude` menawarkan antarmuka pengguna baris perintah juga.
- `aptitude` paling cocok untuk **manajemen paket interaktif harian** seperti memeriksa paket yang dipasang dan mencari paket yang tersedia.
- `aptitude` lebih menuntut pada sumber daya perangkat keras. Ini mengkonsumsi lebih banyak memori dan berjalan lebih lambat.
- `aptitude` menawarkan pencarian berbasis regex yang **disempurnakan** pada semua metadata paket.
- `aptitude` dapat mengelola beberapa versi paket tanpa menggunakan `/etc/apt/preferences` dan itu cukup intuitif.

2.2.2 Operasi manajemen paket dasar dengan barus perintah

Berikut adalah operasi manajemen paket dasar dengan baris perintah menggunakan apt(8), aptitude(8), dan apt-get(8) /apt-cache(8).

sintaks `apt`	`aptitude` syntax	sintaks `apt-get`/`apt-cache`	deskripsi
`apt update`	`aptitude update`	`apt-get update`	memperbarui metadata arsip paket
`apt install foo`	`aptitude install foo`	`apt-get install foo`	memasang versi kandidat paket ”`foo`” dengan dependensinya
`apt upgrade`	`aptitude safe-upgrade`	`apt-get upgrade`	memasang versi kandidat dari paket yang terpasang tanpa menghapus paket lain
`apt full-upgrade`	`aptitude full-upgrade`	`apt-get dist-upgrade`	memasang versi kandidat paket yang terpasang sambil menghapus paket lain jika diperlukan
`apt remove foo`	`aptitude remove foo`	`apt-get remove foo`	menghapus paket ”`foo`” sambil membiarkan berkas konfigurasinya
`apt autoremove`	T/T	`apt-get autoremove`	menghapus paket yang terpasang secara otomatis yang tidak lagi diperlukan
`apt purge foo`	`aptitude purge foo`	`apt-get purge foo`	membersihkan paket ”`foo`” dengan berkas-berkas konfigurasinya
`apt clean`	`aptitude clean`	`apt-get clean`	membersihkan sepenuhnya repositori lokal dari berkas-berkas paket yang diambil
`apt autoclean`	`aptitude autoclean`	`apt-get autoclean`	membersihkan repositori lokal dari berkas-berkas paket yang diambil untuk paket-paket usang
`apt show foo`	`aptitude show foo`	`apt-cache show foo`	menampilkan informasi terperinci tentang paket ”`foo`”
`apt search` *`regex`*	`aptitude search` *`regex`*	`apt-cache search` *`regex`*	mencari paket-paket yang cocok dengan *regex*
T/T	`aptitude why` *`regex`*	T/T	menjelaskan alasan mengapa paket-paket yang cocok *regex* harus dipasang
T/T	`aptitude why-not` *`regex`*	T/T	menjelaskan alasan mengapa paket-paket yang cocok *regex* tidak dapat dipasang
`apt list --manual-installed`	`aptitude search '~i!~M'`	`apt-mark showmanual`	menampilkan daftar paket yang dipasang secara manual

Tabel 2.6: Operasi manajemen paket dasar dengan baris perintah menggunakan apt(8), aptitude(8), dan apt-get(8) /apt-cache(8)

`apt` / `apt-get` dan `aptitude` dapat dicampur tanpa masalah besar.

”`aptitude why` *`regex`*” dapat mencantumkan lebih banyak informasi dengan ”`aptitude -v why` *`regex`*”. Informasi serupa dapat diperoleh dengan ”`apt rdepends` *`paket`*” atau ”`apt-cache rdepends` *`paket`*”.

Ketika perintah `aptitude` dimulai dalam mode baris perintah dan menghadapi beberapa masalah seperti konflik paket, Anda dapat beralih ke mode interaktif layar penuh dengan menekan tombol ”e” nanti di prompt.

Catatan
Meskipun perintah `aptitude` dilengkapi dengan fitur yang kaya seperti resolver paket yang disempurnakan, kompleksitas ini telah menyebabkan (atau mungkin masih menyebabkan) beberapa regresi seperti Bug # 411123, Bug # 514930, dan Bug # 570377. Jika terjadi keraguan, silakan gunakan perintah `apt`, `apt-get`, dan `apt-cache` atas perintah `aptitude`.

Anda dapat memberikan opsi perintah tepat setelah "`aptitude`".

opsi perintah	deskripsi
`-s`	mensimulasikan hasil perintah
`-d`	unduh saja tetapi jangan pasang/tingkatkan
`-D`	menampilkan penjelasan singkat sebelum instalasi dan penghapusan otomatis

Tabel 2.7: Opsi perintah penting untuk aptitude(8)

Lihat aptitude(8) dan "manual pengguna aptitude" di "`/usr/share/doc/aptitude/README`" untuk lebih lanjut.

2.2.3 Penggunaan aptitude interaktif

Untuk manajemen paket interaktif, Anda memulai `aptitude` dalam mode interaktif dari prompt shell konsol sebagai berikut.

```
$ sudo aptitude -u
Password:
```

Ini memperbarui salinan lokal dari informasi arsip dan menampilkan daftar paket di layar penuh dengan menu. Aptitude menempatkan konfigurasinya di "`~/.aptitude/config`".

Tip
Jika Anda ingin menggunakan konfigurasi root sebagai pengganti milik pengguna, gunakan "`sudo -H aptitude` …" bukan "`sudo aptitude` …" dalam ekspresi di atas.

Tip
`Aptitude` secara otomatis menetapkan **tindakan yang tertunda** karena dimulai secara interaktif. Jika Anda tidak menyukainya, Anda dapat mengatur ulang dari menu: "Tindakan" → "Batalkan tindakan yang tertunda".

2.2.4 Pengikatan tombol dari aptitude

Ketukan kunci penting untuk menelusuri status paket dan untuk mengatur "tindakan yang direncanakan" pada mereka dalam mode layar penuh ini adalah sebagai berikut.

Spesifikasi nama berkas dari baris perintah dan prompt menu setelah menekan "`l`" dan "`//`" menerima regex aptitude seperti yang dijelaskan di bawah ini. Regex aptitude dapat secara eksplisit mencocokkan nama paket menggunakan string yang dimulai dengan "~n" dan diikuti dengan nama paket.

Tip
Anda perlu menekan "U" agar semua paket yang terpasang ditingkatkan ke **versi kandidat** di antarmuka visual. Jika tidak, hanya paket yang dipilih dan paket tertentu dengan ketergantungan versi kepada mereka yang ditingkatkan ke **versi kandidat**.

tombol	pengikatan tombol
F10 atau Ctrl-t	menu
?	menampilkan **bantuan** untuk ketukan tombol (daftar yang lebih lengkap)
F10 → Bantuan → Manual Pengguna	menampilkan Manual Pengguna
u	memperbarui informasi arsip paket
+	menandai paket untuk **peningkatan** atau **instalasi**
-	menandai paket untuk **dihapus** (menyimpan berkas konfigurasi)
_	menandai paket untuk **pembersihan** (menghapus berkas-berkas konfigurasi)
=	menempatkan paket ke keadaan **ditahan**
U	menandai semua paket yang dapat ditingkatkan (berfungsi sebagai **full-upgrade**)
g	mulai **mengunduh** dan **memasang** paket-paket yang dipilih
q	keluar dari layar saat ini dan menyimpan perubahan
x	keluar dari layar saat ini dan membuang perubahan
Enter	melihat informasi tentang paket
C	melihat changelog paket
l	mengubah batas untuk paket yang ditampilkan
/	mencari kecocokan pertama
\	mengulangi pencarian terakhir

Tabel 2.8: Daftar pengikatan tombol untuk aptitude

2.2.5 Tampilan paket di bawah aptitude

Dalam mode aptitude(8) layar penuh interaktif, paket-paket dalam daftar paket ditampilkan seperti contoh berikutnya.

```
idA   libsmbclient                          -2220kB 3.0.25a-1  3.0.25a-2
```

Di sini, baris ini berarti dari kiri sebagai berikut.

- Bendera ”keadaan saat ini” (huruf pertama)
- Bendera ”tindakan yang direncanakan” (huruf kedua)
- Bendera ”otomatis” (huruf ketiga)
- Nama paket
- Perubahan penggunaan ruang disk yang dikaitkan dengan ”tindakan yang direncanakan”
- Versi paket saat ini
- Versi kandidat dari paket

Tip
Daftar lengkap bendera diberikan di bagian bawah layar **Bantuan** yang ditampilkan dengan menekan ”?”.

Versi kandidat dipilih sesuai dengan preferensi lokal saat ini (lihat apt_preferences(5) dan Bagian 2.7.7).

Beberapa jenis tampilan paket tersedia di bawah menu ”Tampilan”.

Catatan
Tolong bantu kami memperbaiki penandaan paket dengan debtags!

tampilan	deskripsi tampilan
`Package View`	lihat Tabel 2.10 (baku)
`Audit Recommendations`	menampilkan daftar paket yang direkomendasikan oleh beberapa paket yang terpasang tetapi belum dipasang
`Flat Package List`	menampilkan daftar paket tanpa kategorisasi (untuk digunakan dengan regex)
`Debtags Browser`	menampilkan daftar paket yang dikategorikan menurut entri debtags mereka
`Source Package View`	menmpilkan daftar paket yang dikelompokkan menurut paket sumber

Tabel 2.9: Daftar tampilan untuk aptitude

kategori	deskripsi tampilan
`Upgradable Packages`	menampilkan daftar paket yang disusun sebagai `bagian` → `area` → `paket`
`New Packages`	,,
`Installed Packages`	,,
`Not Installed Packages`	,,
`Obsolete and Locally Created Packages`	,,
`Virtual Packages`	daftar paket dengan fungsi yang sama
`Tasks`	menampilkan daftar paket dengan fungsi berbeda yang umumnya diperlukan untuk suatu tugas

Tabel 2.10: Kategorisasi tampilan paket standar

"`Tampilan Paket`" standar mengkategorikan paket kurang lebih seperti `dselect` dengan beberapa fitur tambahan.

Tip
Tampilan `Tugas` dapat digunakan untuk memilih paket untuk tugas Anda.

2.2.6 Opsi metode pencarian dengan aptitude

Aptitude menawarkan beberapa pilihan bagi Anda untuk mencari paket menggunakan rumus regex-nya.

- Baris perintah shell:
  - "`aptitude search '`*`regex_aptitude`*`'`" untuk daftar status instalasi, nama paket, dan deskripsi singkat dari paket yang cocok
  - "`aptitude show '`*`nama_paket`*`'`" untuk daftar deskripsi rinci dari paket
- Mode layar penuh interaktif:
  - "`l`" untuk membatasi tampilan paket ke paket-paket yang cocok
  - "`/`" untuk pencarian ke paket yang cocok
  - "`\`" untuk pencarian mundur ke paket yang cocok
  - "`n`" untuk cari-berikutnya
  - "`N`" untuk cari-berikutnya (mundur)

Tip
String untuk *nama_paket* diperlakukan sebagai pencocokan string yang tepat dengan nama paket kecuali dimulai secara eksplisit dengan ”~” untuk menjadi rumus regex.

2.2.7 Rumus regex aptitude

Rumus regex aptitude adalah **ERE** diperluas mirip mutt (lihat Bagian 1.6.2) dan arti dari ekstensi aturan kecocokan khusus spesifik `aptitude` adalah sebagai berikut.

- Bagian regex adalah **ERE** yang sama dengan yang biasa digunakan dalam alat teks mirip Unix memakai ”^”, ”`.*`”, ”$” dll. seperti pada egrep(1), awk(1), dan perl(1).
- *Tipe* ketergantungan adalah satu dari (depends, predepends, recommends, suggests, conflicts, replaces, provides) yang menyatakan keterkaitan paket.
- *Tipe* kebergantungan baku adalah ”depends”.

Tip
Ketika *pola_regex* adalah string null, tempatkan ”~T” segera setelah perintah.

Berikut adalah beberapa pintasan.

- ”`~P`*`istilah`*” == ”`~Dprovides:`*`istilah`*”
- ”`~C`*`istilah`*” == ”`~Dconflicts:`*`istilah`*”
- ”…`~W istilah`” == ”`(…|istilah)`”

Pengguna yang akrab dengan `mutt` belajar dengan cepat, karena mutt adalah inspirasi untuk sintaks ekspresi. Lihat ”MENCARI, MEMBATASI, DAN EKSPRESI” di ”Manual Pengguna” ”`/usr/share/doc/aptitude/README`”.

Catatan
Dengan versi `lenny` aptitude(8), sintaks **bentuk panjang** baru seperti ”`?broken`” dapat digunakan untuk pencocokan regex di tempat untuk **bentuk pendek** lama yang setara ”~ b”. Sekarang karakter spasi ” ” dianggap sebagai salah satu karakter regex pengakhiran selain karakter tilde ”~”. Lihat ”Manual Pengguna” untuk sintaks **bentuk panjang** yang baru.

2.2.8 Resolusi ketergantungan aptitude

Pemilihan paket dalam `aptitude` tidak hanya menarik paket yang didefinisikan dalam daftar ”`Depends:`” tetapi juga didefinisikan dalam daftar ”`Recommends:`” jika menu ”Opsi →`F10` → Preferensi → Penanganan Ketergantungan” diatur sesuai dengan itu. Paket yang dipasang otomatis ini dihapus secara otomatis jika tidak lagi diperlukan di bawah `aptitude`.

Bendera yang mengendalikan perilaku ”pemasangan otomatis” dari perintah `aptitude` juga dapat dimanipulasi menggunakan perintah apt-mark(8) dari paket `apt`.

2.2.9 Log aktivitas paket

Anda dapat memeriksa riwayat aktivitas paket di berkas-berkas log.

Pada kenyataannya, tidak mudah untuk mendapatkan pemahaman yang berarti dengan cepat dari log ini. Lihat Bagian 9.3.9 untuk cara yang lebih mudah.

deskripsi aturan pencocokan yang diperluas	**rumus regex**
cocok pada nama paket	`~n`*`regex_name`*
cocok pada deskripsi	`~d`*`regex_description`*
cocokkan pada nama task	`~t`*`regex_task`*
cocok dengan debtag	`~G`*`regex_debtag`*
cocok pada maintainer	`~m`*`regex_maintainer`*
cocok pada bagian paket	`~s`*`regex_section`*
cocok pada versi paket	`~V`*`regex_version`*
cocok arsip	`~A{trixie,forky,sid}`
cocok asal	`~O{debian,…}`
cocok prioritas	`~p{extra,important,optional,required,standard}`
cocok paket esensial	`~E`
cocok paket virtual	`~v`
cocok paket baru	`~N`
cocok dengan tindakan yang tertunda	`~a{install,upgrade,downgrade,remove,purge,hold,keep}`
cocok paket terpasang	`~i`
cocok dengan paket yang dipasang dengan tanda **A** (paket yang dipasang otomatis)	`~M`
cocok dengan paket terpasang tanpa tanda **M** (paket yang dipilih administrator)	`~i!~M`
cocok dengan paket-paket yang dipasang dan dapat ditingkatkan	`~U`
cocok dengan paket-paket yang dihapus tetapi belum dibersihkan	`~c`
cocok dengan paket-paket yang dihapus, dibersihkan, atau dapat-dihapus	`~g`
cocok dengan paket-paket yang menyatakan ketergantungan yang rusak	`~b`
cocok dengan paket yang menyatakan ketergantungan *tipe* yang rusak	`~B`*`type`*
cocok dengan *pola* paket yang menyatakan ketergantungan *tipe*	`~D[`*`type`*`:]`*`pattern`*
cocok dengan *pola* paket yang menyatakan ketergantungan *tipe* yang rusak	`~DB[`*`type`*`:]`*`pattern`*
cocok dengan paket yang memiliki *pola* sesuai dengan paket yang menyatakan ketergantungan *tipe*	`~R[`*`type`*`:]`*`pattern`*
cocok dengan paket yang memiliki *pola* sesuai dengan paket yang menyatakan ketergantungan *tipe* yang rusak	`~RB[`*`type`*`:]`*`pattern`*
cocok dengan paket yang beberapa paket terpasang lainnya bergantung pada mereka	`~R~i`
cocok dengan paket yang tidak bergantung pada paket terpasang lainnya	`!~R~i`
cocok dengan paket yang beberapa paket terpasang lainnya bergantung atau direkomendasikan pada mereka	`~R~i|~Rrecommends:~i`
cocok dengan *pola* paket dengan versi yang difilter	`~S filter `*`pattern`*
cocok dengan semua paket (benar)	`~T`
tidak cocok dengan paket apa pun (salah)	`~F`

Tabel 2.11: Daftar rumus regex aptitude

berkas	isi
`/var/log/dpkg.log`	Log kegiatan tingkat `dpkg` untuk semua kegiatan paket
`/var/log/apt/term.log`	Log aktivitas APT generik
`/var/log/aptitude`	Log aktivitas perintah `aptitude`

Tabel 2.12: Berkas-berkas log untuk aktivitas paket

2.3 Contoh operasi aptitude

Berikut adalah beberapa contoh operasi aptitude(8).

2.3.1 Mencari paket yang menarik

Anda dapat mencari paket yang memenuhi kebutuhan Anda dengan `aptitude` dari deskripsi paket atau dari daftar di bawah ”Tugas”.

2.3.2 Menampilkan daftar paket dengan pencocokan regex pada nama paket

Perintah berikut mencantumkan paket dengan pencocokan regex pada nama paket.

```
$ aptitude search '~n(pam|nss).*ldap'
p libnss-ldap - NSS module for using LDAP as a naming service
p libpam-ldap - Pluggable Authentication Module allowing LDAP interfaces
```

Ini sangat berguna bagi Anda untuk menemukan nama yang tepat dari sebuah paket.

2.3.3 Meramban dengan pencocokan regex

Regex ”`~dipv6`” dalam tampilan ”Daftar Paket Datar baru” dengan prompt ”`l`”, membatasi tampilan ke paket dengan deskripsi yang cocok dan memungkinkan Anda menelusuri informasinya secara interaktif.

2.3.4 Membersihkan paket yang dihapus untuk selamanya

Anda dapat membersihkan semua berkas konfigurasi yang tersisa dari paket yang dihapus.

Periksa hasil perintah berikut.

```
# aptitude search '~c'
```

Jika Anda berpikir paket yang terdaftar baik-baik saja untuk dibersihkan, jalankan perintah berikut.

```
# aptitude purge '~c'
```

Anda mungkin ingin melakukan hal yang sama dalam mode interaktif untuk kontrol yang lebih halus.

Anda memberikan regex ”`~c`” dalam tampilan ”Tampilan Paket Baru” dengan prompt ”`l`. Ini membatasi tampilan paket hanya untuk paket yang cocok regex, yaitu, ”dihapus tetapi tidak dibersihkan”. Semua paket yang cocok regex ini dapat ditunjukkan dengan menekan ”`[`” pada judul tingkat puncak.

Kemudian Anda menekan ”_” pada judul tingkat puncak seperti ”Pakat Tidak Terpasang”. Hanya paket yang cocok regex di bawah judul yang ditandai untuk dibersihkan oleh ini. Anda dapat mengecualikan beberapa paket untuk dibersihkan dengan menekan ”=” secara interaktif untuk masing-masing.

Teknik ini cukup berguna dan bekerja untuk banyak tombol perintah lainnya.

2.3.5 Merapikan status pemasangan otomatis/manual

Berikut adalah cara saya merapikan status pemasangan otomatis/manual untuk paket (setelah menggunakan pemasang paket non-aptitude dll.).

1. Mulai `aptitude` dalam mode interaktif sebagai root.
2. Ketik ”u”, ”U”, ”f”, dan ”g” untuk memperbarui dan meningkatkan daftar paket dan paket-paket.
3. Ketik ”l” untuk memasukkan batas tampilan paket sebagai ”`~i(~R~i|~Rrecommends:~i)`” dan ketik ”M” di atas ”`Paket Terpasang`” sebagai terpasang otomatis.
4. Ketik ”l” untuk memasukkan batas tampilan paket sebagai ”`~required|~pimportant|~pstandard|~E`” dan ketik ”m” di atas ”`Paket Terpasang`” sebagai terpasang manual.
5. Ketik ”l” untuk memasukkan batas tampilan paket sebagai ”`~i!~M`” dan hapus paket yang tidak terpakai dengan mengetik ”-” di atas masing-masing setelah mengekspos mereka dengan mengetik ”[” di atas ”`Paket Terpasang`”.
6. Ketik ”l”, untuk memasukkan batas tampilan paket sebagai ”`~i`”; kemudian ketik ”m” di atas ”`Tugas`” untuk menandai paket itu sebagai terpasang manual.
7. Keluar dari `aptitude`.
8. Mulai ”`apt-get -s autoremove|less`” sebagai root untuk memeriksa apa yang tidak digunakan.
9. Mulai ulang `aptitude` dalam mode interaktif dan tandai paket yang dibutuhkan sebagai ”m”.
10. Jalankan ulang ”`apt-get -s autoremove|less`” sebagai root untuk memeriksa ulang bahwa DIHAPUS hanya berisi paket yang diharapkan.
11. Mulai ”`apt-get autoremove|less`” sebagai root untuk menghapus otomatis paket yang tidak terpakai.

Tindakan ”m” atas ”`Tugas`” adalah tindakan opsional untuk mencegah situasi penghapusan paket massal di masa depan.

2.3.6 Peningkatan seluruh sistem

Catatan
Saat pindah ke rilis baru dll, Anda harus mempertimbangkan untuk melakukan instalasi yang bersih dari sistem baru meskipun Debian dapat ditingkatkan seperti yang dijelaskan di bawah ini. Ini memberi Anda kesempatan untuk membuang sampah yang dikumpulkan dan menghadapkan Anda ke kombinasi terbaik dari paket terbaru. Tentu saja, Anda harus membuat cadangan penuh sistem ke tempat yang aman (lihat Bagian 10.2) sebelum melakukan ini. Saya sarankan untuk membuat konfigurasi boot ganda menggunakan partisi yang berbeda untuk mendapatkan transisi yang paling mulus.

Anda dapat melakukan peningkatan seluruh sistem ke rilis yang lebih baru dengan mengubah isi **daftar sumber** yang menunjuk ke rilis baru dan menjalankan perintah ”`apt update; apt dist-upgrade`”.

Untuk meningkatkan dari `stable` ke `testing` atau `unstable`, selama siklus rilis `trixie`-sebagai-`stable`, Anda mengganti ”`trixie`” dalam contoh **daftar sumber** dari Bagian 2.1.5 dengan ”`forky`” atau ”`sid`”.

Kenyataannya, Anda mungkin menghadapi beberapa komplikasi karena beberapa masalah transisi paket, sebagian besar karena ketergantungan paket. Semakin besar perbedaan peningkatan, semakin besar kemungkinan Anda menghadapi masalah yang lebih besar. Untuk transisi dari `stable` lama ke `stable` baru setelah rilis, Anda dapat membaca Catatan Rilis baru dan mengikuti prosedur yang tepat dijelaskan di dalamnya untuk meminimalkan masalah.

Ketika Anda memutuskan untuk berpindah dari `stable` ke `testing` sebelum rilis resminya, tidak ada Catatan Rilis untuk membantu Anda. Perbedaan antara `stable` dan `testing` bisa tumbuh cukup besar setelah rilis `stable` sebelumnya dan membuat situasi peningkatan rumit.

Anda harus membuat langkah pencegahan untuk peningkatan penuh sambil mengumpulkan informasi terbaru dari milis dan menggunakan akal sehat.

1. Membaca ”Catatan Rilis” sebelumnya.
2. Cadangkan seluruh sistem (terutama data dan informasi konfigurasi).
3. Memiliki media yang dapat di-boot yang siap untuk bootloader rusak.
4. Informasikan ke para pengguna sistem jauh sebelumnya.
5. Rekam aktivitas peningkatan dengan script(1).
6. Terapkan ”unmarkauto” ke paket yang diperlukan, mis., ”`aptitude unmarkauto vim`”, untuk mencegah penghapusan.
7. Minimalkan paket yang dipasang untuk mengurangi kemungkinan konflik paket, mis., menghapus paket tugas desktop.
8. Hapus berkas ”`/etc/apt/preferences`” (nonaktifkan **apt-pinning**).
9. Cobalah untuk meningkatkan secara bertahap: `oldstable` → `stable` → `testing` → `unstable`.
10. Perbarui **daftar sumber** agar menunjuk hanya ke arsip baru dan jalankan ”`aptitude update`”.
11. Pasang terlebih dahulu, secara opsional, **paket-paket inti** baru, mis.,”`aptitude install perl`”.
12. Jalankan perintah ”`apt-get -s dist-upgrade`” untuk menilai dampak.
13. Pada akhirknya jalankan perintah ”`apt-get dist-upgrade`”.



Perhatian
Tidak bijaksana untuk melewatkan rilis Debian mayor ketika meningkatkan antar rilis `stable`.



Perhatian
Dalam ”Catatan Rilis” sebelumnya, GCC, Kernel Linux, initrd-tools, Glibc, Perl, rantai alat APT, dll. membutuhkan perhatian khusus untuk peningkatan seluruh sistem.



Perhatian
”Catatan Rilis” mungkin tidak mencakup semua kasus yang mungkin. Bila Anda mengubah konfigurasi tingkat rendah, peningkatan berikutnya mungkin gagal dengan buruk sebagai ”... segfault setelah peningkatan ...”.

Untuk peningkatan harian dalam `unstable`, lihat Bagian 2.4.3.

perintah	aksi
`COLUMNS=120 dpkg -l` *`package_name_pattern`*	menampilkan daftar status paket yang dipasang untuk laporan bug
`dpkg -L` *`package_name`*	menampilkan daftar isi paket yang dipasang
`dpkg -L` *`package_name`* `| egrep '/usr/share/man/man.*/.+'`	menampilkan daftar halaman man untuk paket yang dipasang
`dpkg -S` *`file_name_pattern`*	daftar paket terpasang yang memiliki nama berkas yang cocok
`apt-file search` *`file_name_pattern`*	tampilkan daftar paket dalam arsip yang memiliki nama berkas yang cocok
`apt-file list` *`package_name_pattern`*	menampilkan daftar isi paket yang cocok dalam arsip
`dpkg-reconfigure` *`package_name`*	mengonfigurasi ulang paket yang tepat
`dpkg-reconfigure -plow` *`package_name`*	mengonfigurasi ulang paket yang tepat dengan pertanyaan paling rinci
`configure-debian`	mengonfigurasi ulang paket dari menu layar penuh
`dpkg --audit`	mengaudit sistem untuk paket-paket yang terpasang sebagian
`dpkg --configure -a`	mengonfigurasi semua paket yang terpasang sebagian
`apt-cache policy` *`binary_package_name`*	menampilkan versi yang tersedia, prioritas, dan informasi arsip dari suatu paket biner
`apt-cache madison` *`package_name`*	menampilkan versi yang tersedia, informasi arsip paket
`apt-cache showsrc` *`binary_package_name`*	menampilkan informasi paket sumber dari suatu paket biner
`apt-get build-dep` *`package_name`*	memasang paket-paket yang diperlukan untuk membangun paket
`aptitude build-dep` *`package_name`*	memasang paket-paket yang diperlukan untuk membangun paket
`apt-get source` *`nama_paket`*	mengunduh suatu sumber (dari arsip standar)
`dget` *`URL for dsc file`*	mengunduh suatu paket sumber (dari arsip lainnya)
`dpkg-source -x` *`package_name_version-debian.revision`*`.dsc`	membangun pohon sumber dari satu set paket sumber (”`*.orig.tar.gz`” dan ”`*.debian.tar.gz`”/”`*.diff.gz`”)
`debuild binary`	membangun paket dari pohon sumber lokal
`make-kpkg kernel_image`	membangun paket kernel dari pohon sumber kernel
`make-kpkg --initrd kernel_image`	membangun paket kernel dari pohon sumber kernel dengan initramfs diaktifkan
`dpkg -i` *`package_name_version-debian.revision_arch`*`.deb`	memasang sebuah paket lokal ke sistem
`apt install /path/to/`*`package_filename`*`.deb`	memasang paket lokal ke sistem, sambil mencoba untuk menyelesaikan ketergantungan secara otomatis
`debi` *`package_name_version-debian.revision_arch`*`.dsc`	memasang paket lokal ke sistem
`dpkg --get-selections '*' >selection.txt`	menyimpan informasi keadaan pemilihan paket tingkat `dpkg`
`dpkg --set-selections <selection.txt`	mengatur informasi keadaan pemilihan paket tingkat `dpkg`
`echo` *`nama_paket`* `hold | dpkg --set-selections`	menata status pemilihan paket tingkat `dpkg` untuk paket yang akan **ditahan** (setara dengan ”`aptitude hold` *`nama_paket`*”)

Tabel 2.13: Daftar operasi manajemen paket tingkat lanjut

2.4 Operasi manajemen paket tingkat lanjut

2.4.1 Operasi manajemen paket tingkat lanjut dengan baris perintah

Berikut adalah daftar operasi manajemen paket lain dimana `aptitude` tingkatnya terlalu tinggi atau tidak memiliki fungsi yang diperlukan.

Catatan
Untuk paket dengan fitur multi-arch, Anda mungkin perlu menyatakan nama arsitektur untuk beberapa perintah. Misalnya, gunakan "`dpkg -L libglib2.0-0:amd64`" untuk mencantumkan isi paket `libglib2.0-0` bagi arsitektur `amd64`.



Perhatian
Alat paket tingkat rendah seperti "`dpkg -i …`" dan "`debi …`" harus hati-hati digunakan oleh administrator sistem. Itu tidak secara otomatis mengurus dependensi paket yang diperlukan. Opsi baris perintah dpkg "`--force-all`" dan yang serupa (lihat dpkg(1)) dimaksudkan untuk digunakan oleh para ahli saja. Menggunakannya tanpa sepenuhnya memahami efeknya dapat merusak seluruh sistem Anda.

Harap catat hal-hal berikut.

- Semua konfigurasi sistem dan perintah instalasi harus dijalankan dari root.
- Tidak seperti `aptitude` yang menggunakan regex (lihat Bagian 1.6.2), perintah manajemen paket lainnya menggunakan pola seperti glob shell (lihat Bagian 1.5.6).
- apt-file(1) yang disediakan oleh paket `apt-file` harus menjalankan "`apt-file update`" terlebih dahulu.
- configure-debian(8) yang disediakan oleh paket `configure-debian` menjalankan dpkg-reconfigure(8) sebagai backend-nya.
- dpkg-reconfigure(8) menjalankan skrip paket menggunakan debconf(1) sebagai backend-nya.
- Perintah "`apt-get build-dep`", "`apt-get source`", dan "`apt-cache showsrc`" memerlukan entri "`deb-src`" di **daftar sumber**.
- dget(1), debuild(1), dan debi(1) memerlukan paket `devscripts`.
- Lihat prosedur pengemasan (ulang) menggunakan "`apt-get source`" dalam Bagian 2.7.13.
- Perintah `make-kpkg` memerlukan paket `kernel-package` (lihat Bagian 9.10).
- Lihat Bagian 12.9 untuk pengemasan umum.

2.4.2 Verifikasi berkas-berkas paket yang terpasang

Pemasangan `debsum` memungkinkan verifikasi berkas paket yang terpasang terhadap nilai MD5sum dalam berkas "`/var/lib/dpkg/info/*.md5sums`" dengan debsum(1). Lihat Bagian 10.3.5 untuk cara kerja MD5sum.

Catatan
Karena database MD5sum dapat dirusak oleh penyusup, penggunaan debsum(1) terbatas sebagai alat keamanan. Ini hanya baik untuk memeriksa modifikasi lokal oleh administrator atau kerusakan karena kesalahan media.

2.4.3 Menjaga masalah paket

Banyak pengguna lebih memilih untuk mengikuti rilis sistem Debian **testing** (atau **unstable**) untuk fitur dan paket barunya. Hal ini membuat sistem lebih rentan terkena bug paket kritis.

Instalasi paket `apt-listbugs` melindungi sistem Anda dari bug kritis dengan memeriksa Debian BTS secara otomatis untuk bug kritis saat memutakhirkan dengan sistem APT.

Pemasangan paket `apt-listchanges` memberikan berita penting dalam "`NEWS.Debian`" saat memutakhirkan dengan sistem APT.

2.4.4 Mencari pada data meta paket

Meskipun mengunjungi situs Debian https://packages.debian.org/ memfasilitasi cara mudah untuk mencari data meta paket akhir-akhir ini, mari kita lihat cara yang lebih tradisional.

Perintah grep-dctrl(1), grep-status(1), dan grep-available(1) dapat digunakan untuk mencari berkas apa pun yang memiliki format umum berkas kontrol paket Debian.

"`dpkg -S` *`pola_nama_berkas`*" dapat digunakan untuk mencari nama paket yang berisi berkas dengan nama yang cocok yang dipasang oleh `dpkg`. Tapi ini mengabaikan berkas-berkas yang dibuat oleh skrip pengelola.

Jika Anda perlu melakukan pencarian yang lebih rumit pada data meta dpkg, Anda perlu menjalankan perintah "`grep -e pola_regex *`" di direktori "`/var/lib/dpkg/info/`". Hal ini membuat Anda mencari kata-kata yang disebutkan dalam skrip paket dan teks kueri instalasi.

Jika Anda ingin mencari ketergantungan paket secara rekursif, Anda harus menggunakan apt-rdepends(8).

2.5 Internal manajemen paket Debian

Mari kita pelajari cara kerja sistem manajemen paket Debian secara internal. Ini akan membantu Anda membuat solusi sendiri untuk beberapa masalah paket.

2.5.1 Meta data arsip

Berkas meta data untuk setiap distribusi disimpan di bawah "`dist/`*`namakode`*" pada setiap situs cermin Debian, mis., "`http://deb.debian.org/debian/`". Struktur arsipnya dapat ditelusuri oleh peramban web. Ada 6 jenis data meta kunci.

berkas	**lokasi**	**isi**
`Release`	puncak distribusi	deskripsi arsip dan informasi integritas
`Release.gpg`	puncak distribusi	berkas tanda tangan untuk berkas "`Release`" yang ditandatangani dengan kunci arsip
`Contents-`*`architecture`*	puncak distribusi	daftar semua berkas untuk semua paket dalam arsip yang bersangkutan
`Release`	puncak dari setiap kombinasi distribusi/area/arsitektur	deskripsi arsip yang digunakan untuk aturan apt_preferences(5)
`Packages`	puncak dari setiap kombinasi distribusi/area/arsitektur biner	`debian/control` gabungan untuk paket biner
`Sources`	puncak dari setiap kombinasi distribusi/area/sumber	`debian/control` gabungan untuk paket sumber

Tabel 2.14: Isi data meta arsip Debian

Dalam arsip baru-baru ini, data meta ini disimpan sebagai berkas terkompresi dan diferensial untuk mengurangi lalu lintas jaringan.

2.5.2 Berkas ”Release” tingkat puncak dan keaslian

Tip
Berkas ”Release” tingkat puncak digunakan untuk menandatangani arsip di bawah sistem **APT yang aman**.

Setiap keluarga arsip Debian memiliki berkas ”Release” tingkat puncak, mis., ”http://deb.debian.org/debian/dist sebagai berikut.

```
Origin: Debian
Label: Debian
Suite: unstable
Codename: sid
Date: Sat, 14 May 2011 08:20:50 UTC
Valid-Until: Sat, 21 May 2011 08:20:50 UTC
Architectures: alpha amd64 armel hppa hurd-i386 i386 ia64 kfreebsd-amd64 kfreebsd-i386 mips ↩
     mipsel powerpc s390 sparc
Components: main contrib non-free
Description: Debian x.y Unstable - Not Released
MD5Sum:
 bdc8fa4b3f5e4a715dd0d56d176fc789 18876880 Contents-alpha.gz
 9469a03c94b85e010d116aeeab9614c0 19441880 Contents-amd64.gz
 3d68e206d7faa3aded660dc0996054fe 19203165 Contents-armel.gz
...
```

Catatan
Di sini, Anda dapat menemukan alasan saya untuk menggunakan ”keluarga”, dan ”nama kode” di Bagian 2.1.5. ”Distribusi” digunakan ketika mengacu pada ”keluarga” dan ”nama kode”. Semua nama ”area” arsip yang ditawarkan oleh arsip tercantum di bawah ”Komponen”.

Integritas berkas ”Release” tingkat puncak diverifikasi oleh infrastruktur kriptografi yang disebut apt aman sebagaimana diuraikan dalam apt-secure(8).

- Berkas tanda tangan kriptografis ”Release.gpg” dibuat dari berkas ”Release” tingkat puncak yang otentik dan kunci arsip Debian rahasia.
- Kunci arsip Debian publik dipasang secara lokal oleh paket debian-archive-keyring terbaru.
- Sistem **APT aman** memverifikasi secara otomatis integritas berkas ”Release” tingkat puncak yang diunduh secara kriptografis memakai berkas ”Release.gpg” ini dan kunci arsip Debian publik yang terpasang secara lokal.
- Integritas semua berkas ”Packages” dan ”Sources” diverifikasi dengan menggunakan nilai MD5sum dalam berkas ”Release” tingkat puncak. Integritas semua berkas paket diverifikasi dengan menggunakan nilai MD5sum dalam berkas ”Packages” dan ”Sources”. Lihat debsum(1) dan Bagian 2.4.2.
- Karena verifikasi tanda tangan kriptografis adalah proses yang jauh lebih intensif CPU daripada perhitungan nilai MD5sum, penggunaan nilai MD5sum untuk setiap paket saat menggunakan tanda tangan kriptografis untuk berkas ”Releases” tingkat puncak memberikan keamanan yang baik dengan kinerja (lihat Bagian 10.3).

Bila entri **daftar sumber** menyatakan opsi ”signed-by”, integritas berkas ”Release” tingkat puncaknya yang diunduh diverifikasi memakai kunci publik yang dinyatakan. Ini berguna ketika **daftar sumber** menmuat arsip-arsip non-Debian.

Tip
Penggunaan perintah apt-key(8) untuk manajemen kunci APT tidak berlaku lagi.

Anda juga dapat memverifikasi secara manual integritas dari berkas ”Release” dengan berkas ”Release.gpg” dan kunci arsip Debian publik yang diumumkan di ftp-master.debian.org memakai gpg.

2.5.3 Berkas ”Release” tingkat arsip

Tip
Berkas ”`Release`” tingkat arsip digunakan untuk aturan apt_preferences(5).

Ada berkas ”`Release`” tingkat arsip untuk semua lokasi arsip yang ditentukan oleh **daftar sumber**, seperti ”`http://deb.de` atau ”`http://deb.debian.org/debian/dists/sid/main/binary-amd64/Release`” sebagai berikut.

```
Archive: unstable
Origin: Debian
Label: Debian
Component: main
Architecture: amd64
```



Perhatian
Untuk stanza ”`Archive:`” , nama keluarga (”`stable`”, ”`testing`”, ”`unstable`”, ...) digunakan dalam arsip Debian sementara nama kode (”`trusty`”, ”`xenial`”, ”`artful`”, ...) digunakan dalam arsip Ubuntu.

Untuk beberapa arsip, seperti `experimental`, dan `trixie-backports`, yang berisi paket yang tidak boleh dipasang secara otomatis, ada baris tambahan, misalnya, ”`http://deb.debian.org/debian/dists/experimental/mai` sebagai berikut.

```
Archive: experimental
Origin: Debian
Label: Debian
NotAutomatic: yes
Component: main
Architecture: amd64
```

Harap dicatat bahwa untuk arsip normal tanpa ”`NotAutomatic: yes`”, nilai Pin-Priority baku adalah 500, sedangkan untuk arsip khusus dengan ”`NotAutomatic: yes`”, nilai Pin-Priority baku adalah 1 (lihat apt_preferences(5) dan Bagian 2.7.7).

2.5.4 Pengambilan data meta untuk paket

Ketika alat APT, seperti `aptitude`, `apt-get`, `synaptic`, `apt-file`, `auto-apt`, ... digunakan, kita perlu memperbarui salinan lokal dari data meta yang berisi informasi arsip Debian. Salinan lokal ini memiliki nama berkas berikut yang sesuai dengan `distribusi`, `area`, dan nama `arsitektur` dalam **daftar sumber** (lihat Bagian 2.1.5).

- ”`/var/lib/apt/lists/deb.debian.org_debian_dists_`*`distribusi`*`_Release`”
- ”`/var/lib/apt/lists/deb.debian.org_debian_dists_`*`distribusi`*`_Release.gpg`”
- ”`/var/lib/apt/lists/deb.debian.org_debian_dists_`*`distribusi_area`*`_binary-`*`arsitektur`*`_Package`
- ”`/var/lib/apt/lists/deb.debian.org_debian_dists_`*`distribusi_area`*`_source_Sources`”
- ”`/var/cache/apt/apt-file/deb.debian.org_debian_dists_`*`distribusi`*`_Contents-`*`arsitektur`*`.gz`” (untuk `apt-file`)

4 jenis berkas pertama dipakai bersama oleh semua perintah APT yang bersangkutan dan diperbarui dari baris perintah dengan "`apt-get update`" atau "`aptitude update`". Data meta "`Packages`" diperbarui jika ada baris "`deb`" di **daftar sumber**. Data meta "`Sources`" diperbarui jika ada baris "`deb-src`" di **daftar sumber**.

Data meta "`Packages`" dan "`Sources`" berisi stanza "`Filename:`" yang menunjuk ke lokasi berkas paket biner dan sumber. Saat ini, paket-paket ini terletak di bawah pohon direktori "`pool/`" untuk transisi antar rilis yang lebih baik.

Salinan lokal dari meta data "`Packages`" dapat secara interaktif dicari dengan bantuan `aptitude`. Perintah pencarian khusus grep-dctrl(1) dapat mencari salinan lokal dari meta data "`Packages`" dan "`Sources`".

Salinan lokal dari data meta "`Contents-`*`arsitektur`*" dapat diperbarui dengan "`apt-file update`" dan lokasinya berbeda dari 4 yang lainnya. Lihat apt-file(1). `auto-apt` menggunakan lokasi yang berbeda untuk salinan lokal "`Contents-`*`arsitektur`*`.gz`" sebagai baku.)

2.5.5 Keadaan paket untuk APT

Selain data meta yang diambil dari jarak jauh, alat APT setelah `lenny` menyimpan informasi keadaan instalasi yang dihasilkan secara lokal di "`/var/lib/apt/extended_states`" yang digunakan oleh semua alat APT untuk melacak semua paket yang dipasang secara otomatis.

2.5.6 Keadaan paket untuk aptitude

Selain data meta yang diambil dari jarak jauh, perintah `aptitude` menyimpan informasi keadaan instalasi yang dihasilkan secara lokal di "`/var/lib/aptitude/pkgstates`" yang hanya digunakan olehnya.

2.5.7 Salinan lokal dari paket yang diambil

Semua paket yang diambil dari jarak jauh melalui mekanisme APT disimpan dalam "`/var/cache/apt/archives`" sampai dibersihkan.

Kebijakan pembersihan berkas singgahan untuk `aptitude` ini dapat diatur di bawah "`Opsi`" → "`Preferensi`" dan dapat dipaksa oleh menunya "`Bersihkan singgahan paket`" atau "`Bersihkan berkas usang`" di bawah "`Aksi`".

2.5.8 Nama berkas paket Debian

Berkas-berkas paket Debian memiliki struktur nama tertentu.

tipe paket	**struktur nama**
Paket biner (a.k.a `deb`)	*`package-name_upstream-version-debian.revision_architec`*
Paket biner untuk debian-installer (a.k.a `udeb`)	*`package-name_upstream-version-debian.revision_architec`*
Paket sumber (sumber hulu)	*`package-name_upstream-version-debian.revision`*`.orig.tar`
Paket sumber `1.0` (Debian changes)	*`package-name_upstream-version-debian.revision`*`.diff.gz`
Paket sumber `3.0 (quilt)` (Debian changes)	*`package-name_upstream-version-debian.revision`*`.debian.ta`
Paket sumber (deskripsi)	*`package-name_upstream-version-debian.revision`*`.dsc`

Tabel 2.15: Struktur nama paket-paket Debian

Tip
Di sini hanya format paket sumber dasar yang dijelaskan. Lihat selengkapnya di dpkg-source(1).

komponen nama	karakter yang dapat digunakan (regex ERE)	eksistensi
package-name	`[a-z0-9][-a-z0-9.+]+`	diperlukan
epoch`:`	`[0-9]+:`	opsional
upstream-version	`[-a-zA-Z0-9.+:]+`	diperlukan
debian.revision	`[a-zA-Z0-9.+~]+`	opsional

Tabel 2.16: Karakter yang dapat digunakan untuk setiap komponen dalam nama paket Debian

Catatan
Anda dapat memeriksa urutan versi paket dengan dpkg(1), misalnya, "`dpkg --compare-versions 7.0 gt 7.~pre1 ; echo $?`".

Catatan
debian-installer (d-i) menggunakan `udeb` sebagai ekstensi berkas untuk paket binernya, bukan `deb`normal. Paket `udeb` adalah paket `deb` yang dirampingkan dengan menghilangkan beberapa konten yang tidak penting seperti dokumentasi untuk menghemat ruang sambil melonggarkan persyaratan kebijakan paket. Paket `deb` dan `udeb` berbagi struktur paket yang sama. "u" singkatan dari mikro.

2.5.9 Perintah dpkg

dpkg(1) adalah alat tingkat terendah untuk manajemen paket Debian. Ini sangat kuat dan perlu digunakan dengan hati-hati.

Saat memasang paket yang disebut "*nama_paket*", `dpkg` mengolahnya dalam urutan berikut.

1. Membuka kemasan berkas deb ("setara `ar -x`")
2. Menjalankan "*nama_paket*`.preinst`" menggunakan debconf(1)
3. Memasang konten paket ke sistem ("setara `tar -x`")
4. Menjalankan "*nama_paket*`.postinst`" menggunakan debconf(1)

Sistem `debconf` menyediakan interaksi pengguna standar dengan dukungan I18N dan L10N (Bab 8).

Berkas "`status`" juga digunakan oleh alat-alat seperti dpkg(1), "`dselect update`", dan "`apt-get -u dselect-upgrad`

Perintah pencarian khusus grep-dctrl(1) dapat mencari salinan lokal dari meta data "`status`" dan "`available`".

Tip
Di lingkungan debian-installer, perintah `udpkg` digunakan untuk membuka paket `udeb`. Perintah `udpkg` adalah versi ramping dari perintah `dpkg`.

2.5.10 Perintah update-alternatives

Sistem Debian memiliki mekanisme untuk memasang program yang agak tumpang tindih secara damai menggunakan update-alternatives(1). Misalnya, Anda dapat membuat perintah `vi` memilih untuk menjalankan `vim` saat memasang paket `vim` dan `nvi`.

berkas	deskripsi konten
`/var/lib/dpkg/info/`*`package_name`*`.conffiles`	daftar berkas konfigurasi. (bisa diubah oleh pengguna)
`/var/lib/dpkg/info/`*`package_name`*`.list`	daftar berkas dan direktori yang dipasang oleh paket
`/var/lib/dpkg/info/`*`package_name`*`.md5sums`	daftar nilai hash MD5 untuk berkas-berkas yang dipasang oleh paket
`/var/lib/dpkg/info/`*`package_name`*`.preinst`	skrip paket yang akan dijalankan sebelum pemasangan paket
`/var/lib/dpkg/info/`*`package_name`*`.postinst`	skrip paket yang akan dijalankan setelah instalasi paket
`/var/lib/dpkg/info/`*`package_name`*`.prerm`	skrip paket yang akan dijalankan sebelum penghapusan paket
`/var/lib/dpkg/info/`*`package_name`*`.postrm`	skrip paket yang akan dijalankan setelah penghapusan paket
`/var/lib/dpkg/info/`*`package_name`*`.config`	skrip paket untuk sistem `debconf`
`/var/lib/dpkg/alternatives/`*`package_name`*	informasi alternatif yang digunakan oleh perintah `update-alternatives`
`/var/lib/dpkg/available`	informasi ketersediaan untuk semua paket
`/var/lib/dpkg/diversions`	informasi diversi yang digunakan oleh dpkg(1) dan ditata oleh dpkg-divert(8)
`/var/lib/dpkg/statoverride`	informasi penimpaan stat yang digunakan oleh dpkg(1) dan ditata oleh dpkg-statoverride(8)
`/var/lib/dpkg/status`	informasi status untuk semua paket
`/var/lib/dpkg/status-old`	cadangan generasi pertama dari berkas "`/var/lib/dpkg/status`"
`/var/backups/dpkg.status*`	cadangan generasi kedua dan yang lebih tua dari berkas "`/var/lib/dpkg/status`"

Tabel 2.17: Berkas-berkas penting yang dibuat oleh `dpkg`

```
$ ls -l $(type -p vi)
lrwxrwxrwx 1 root root 20 2007-03-24 19:05 /usr/bin/vi -> /etc/alternatives/vi
$ sudo update-alternatives --display vi
...
$ sudo update-alternatives --config vi
  Selection    Command
 -----------------------------------------------
      1        /usr/bin/vim
*+    2        /usr/bin/nvi

Enter to keep the default[*], or type selection number: 1
```

Sistem alternatif Debian menjaga pemilihannya sebagai symlink dalam "`/etc/alternatives/`". Proses seleksi menggunakan berkas yang sesuai dalam "`/var/lib/dpkg/alternatives/`".

2.5.11 Perintah dpkg-statoverride

Penimpaan stat yang disediakan oleh perintah dpkg-statoverride(8) adalah cara untuk memberi tahu dpkg(1) agar menggunakan pemilik atau mode yang berbeda bagi suatu **berkas** saat paket dipasang. Jika "`--update`" dinyatakan dan berkas ada, itu seketika diatur ke pemilik dan mode baru.



Perhatian
Perubahan langsung pemilik atau mode untuk **berkas** yang dimiliki oleh paket menggunakan perintah `chmod` atau `chown` oleh administrator sistem diatur ulang dengan peningkatan paket berikutnya.

Catatan
Saya menggunakan kata **berkas** di sini, tetapi pada kenyataannya ini bisa menjadi sebarang objek sistem berkas yang ditangani `dpkg`, termasuk direktori, perangkat, dll.

2.5.12 Perintah dpkg-divert

Pengalihan berkas yang disediakan oleh perintah dpkg-divert(8) adalah cara memaksa dpkg(1) untuk tidak meng-install berkas ke lokasi baku, tetapi ke lokasi yang **dialihkan**. Penggunaan `dpkg-divert` dimaksudkan bagi skrip pemeliharaan paket. Penggunaannya yang santai oleh administrator sistem sudah usang.

2.6 Pemulihan dari sistem yang rusak

Ketika menjalankan sistem `testing` atau `unstable`, administrator diharapkan memulihkan dari situasi manajemen paket yang rusak.



Perhatian
Beberapa metode yang dijelaskan di sini adalah tindakan berisiko tinggi. Anda telah diperingatkan!

2.6.1 Ketidakcocokan dengan konfigurasi pengguna lama

Jika program GUI desktop mengalami ketidakstabilan setelah peningkatan versi hulu yang signifikan, Anda harus mencurigai gangguan atas berkas konfigurasi lokal lama yang dibuat olehnya. Jika itu stabil di bawah akun pengguna yang baru dibuat, hipotesis ini dikonfirmasi. (Ini adalah bug pemaketan dan biasanya dihindari oleh pemaket.)

Untuk memulihkan stabilitas, Anda harus memindahkan berkas konfigurasi lokal yang sesuai dan memulai ulang program GUI. Anda mungkin perlu membaca konten berkas konfigurasi lama untuk memulihkan informasi konfigurasi nanti. (Jangan menghapusnya terlalu cepat.)

2.6.2 Kesalahan penyinggahan atas data paket

Kesalahan penyinggahan data paket menyebabkan kesalahan yang menarik, seperti "Kesalahan GPG: ... tidak valid: BADSIG ..." dengan APT.

Anda harus menghapus semua data yang disinggahkan dengan "`sudo rm -rf /var/lib/apt/*` " dan coba lagi. (Jika `apt-cacher-ng` digunakan, Anda juga harus menjalankan "`sudo rm -rf /var/cache/apt-cacher-ng/*` ".)

2.6.3 Penyelamatan dengan perintah dpkg

Karena `dpkg` adalah alat paket tingkat rendah, alat ini dapat berfungsi tanpa koneksi jaringan.

Asumsikan paket `foo` rusak dan perlu diperbaiki.

Anda mungkin masih menemukan salinan singgahan dari versi bebas bug yang lebih lama dari paket `foo` di direktori singgahan paket: "`/var/cache/apt/archives/`". (Jika tidak, Anda dapat mengunduhnya dari arsip https://snapshot.debi atau menyalinnya dari singgahan paket mesin yang berfungsi.)

Jika Anda dapat mem-boot sistem, Anda dapat memasangnya dengan perintah berikut.

```
# dpkg -i /path/to/foo_old_version_arch.deb
```

Jika mencoba memasang paket dengan cara ini gagal karena beberapa pelanggaran ketergantungan dan Anda benar-benar perlu melakukan ini sebagai upaya terakhir, Anda dapat mengesampingkan ketergantungan menggunakan `dpkg` "`--ignore-depends`", "`--force-depends`", dan opsi lainnya. Jika Anda melakukan ini, Anda perlu melakukan upaya serius untuk mengembalikan ketergantungan yang tepat nanti. Lihat dpkg(8) untuk detailnya.

Catatan
Jika sistem Anda rusak parah, Anda harus membuat cadangan penuh sistem ke tempat yang aman (lihat Bagian 10.2) dan harus melakukan instalasi bersih. Ini lebih singkat dan memberikan hasil yang lebih baik pada akhirnya.

Tip
Jika kerusakan sistem kecil, Anda dapat menuruntingkatkan seluruh sistem seperyi dalam Bagian 2.7.11 menggunakan sistem APT tingkat yang lebih tinggi.

2.6.4 Instalasi gagal karena dependensi yang kurang

Jika Anda memaksa memasang paket dengan "`sudo dpkg -i ...`" ke sistem tanpa semua paket ketergantungan dipasang, instalasi paket akan gagal sebagai terpasang sebagian.

Anda harus menginstall semua paket dependensi secara berulang menggunakan "`sudo dpkg -i ...`" atau dengan menggunakan:

```
# apt --fix-broken install
```

Lalu konfigurasikan semua paket yang terpasang sebagian dengan perintah berikut.

```
# dpkg --configure -a
```

2.6.5 Paket yang berbeda dengan berkas-berkas yang tumpang tindih

Sistem manajemen paket tingkat arsip, seperti aptitude(8) atau apt-get(1), bahkan tidak mencoba memasang paket dengan berkas yang tumpang tindih menggunakan dependensi paket (lihat Bagian 2.1.7).

Kesalahan oleh pengelola paket atau penyebaran sumber arsip campuran yang tidak konsisten (lihat Bagian 2.7.6) oleh administrator sistem dapat menciptakan situasi dengan dependensi paket yang salah didefinisikan. Ketika Anda memasang paket dengan berkas yang tumpang tindih menggunakan aptitude(8) atau apt-get(1) dalam situasi seperti itu, dpkg(1) yang membuka kemasan paket memastikan untuk mengembalikan kesalahan ke program pemanggil tanpa menimpa berkas yang ada.

Perhatian
Penggunaan paket pihak ketiga memperkenalkan risiko sistem yang signifikan melalui skrip pengelola yang dijalankan dengan hak istimewa root dan dapat melakukan apa saja pada sistem Anda. Perintah dpkg(1) hanya melindungi terhadap penimpaan oleh pembukaan kemasan.

Anda dapat mengatasi instalasi yang rusak tersebut dengan pertama kali menghapus paket lama yang bermasalah, *`paket-lama`*.

```
$ sudo dpkg -P old-package
```

2.6.6 Memperbaiki skrip paket yang rusak

Ketika perintah dalam skrip paket mengembalikan kesalahan karena beberapa alasan dan skrip keluar dengan kesalahan, sistem manajemen paket membatalkan tindakan mereka dan berakhir dengan paket yang terpasang sebagian. Ketika sebuah paket berisi bug dalam skrip penghapusannya, paket bisa saja menjadi tidak mungkin dihapus dan cukup jahat.

Untuk masalah skrip paket ”*nama_paket*”, Anda harus melihat ke dalam skrip-skrip paket berikut.

- ”`/var/lib/dpkg/info/`*`nama_paket`*`.preinst`”
- ”`/var/lib/dpkg/info/`*`nama_paket`*`.postinst`”
- ”`/var/lib/dpkg/info/`*`nama_paket`*`.prerm`”
- ”`/var/lib/dpkg/info/`*`nama_paket`*`.postrm`”

Sunting skrip paket yang bermasalah dari root menggunakan teknik-teknik berikut.

- nonaktifkan baris yang bermasalah dengan menyisipkan ”#” di awal
- paksa untuk mengembalikan sukses dengan menambahi baris bermasalah dengan ”`|| true`”

Lalu konfigurasikan semua paket yang terpasang sebagian dengan perintah berikut.

```
# dpkg --configure -a
```

2.6.7 Memulihkan data pemilihan paket

Jika ”`/var/lib/dpkg/status`” terkorupsi karena alasan apapun, sistem Debian kehilangan data pemilihan paket dan sangat menderita. Cari berkas lama ”`/var/lib/dpkg/status`” di ”`/var/lib/dpkg/status-old`”, atau ”`/var/backups/dpkg.status.*`”.

Menjaga ”`/var/backups/`” dalam partisi terpisah mungkin merupakan ide yang baik karena direktori ini berisi banyak data sistem yang penting.

Untuk kerusakan serius, saya sarankan untuk melakukan instalasi ulang dari awal setelah membuat cadangan sistem. Bahkan jika semua dalam ”`/var/`” hilang, Anda masih dapat memulihkan sebagian informasi dari direktori di ”`/usr/share/doc/`” untuk memandu instalasi baru Anda.

Memasang ulang sistem minimal (desktop).

```
# mkdir -p /path/to/old/system
```

Kait sistem lama di ”`/path/ke/sistem/lama/`”.

```
# cd /path/to/old/system/usr/share/doc
# ls -1 >~/ls1.txt
# cd /usr/share/doc
# ls -1 >>~/ls1.txt
# cd
# sort ls1.txt | uniq | less
```

Kemudian Anda disajikan dengan nama paket untuk yang akan dipasang. (Mungkin ada beberapa nama non-paket seperti ”`texmf`”.)

2.7 Tips untuk manajemen paket

Untuk kesederhanaan, contoh-contoh **daftar sumber** dalam bagian ini disajikan sebagai ”`/etc/apt/sources.list`” dalam gaya satu baris setelah rilis `trixie`.

2.7.1 Siapa yang mengunggah paket tersebut?

Meskipun nama pengelola yang tercantum dalam "`/var/lib/dpkg/available`" dan "`/usr/share/doc/nama_paket/`
memberikan beberapa informasi tentang "siapa yang berada di balik aktivitas pengemasan", pengunggah sebenarnya dari paket ini agak tidak jelas. who-uploads(1) dalam paket `devscripts` mengidentifikasi pengunggah sebenarnya dari paket sumber Debian.

2.7.2 Membatasi bandwidth unduhan untuk APT

Jika Anda ingin membatasi bandwidth unduhan untuk APT mis. 800Kib/detik (=100kiB/detik), Anda harus mengonfigurasi APT dengan parameter konfigurasinya sebagai berikut.

```
APT::Acquire::http::Dl-Limit "800";
```

2.7.3 Pengunduhan dan peningkatan paket secara otomatis

Paket `apt` dilengkapi dengan skrip cron sendiri "`/etc/cron.daily/apt`"untuk mendukung pengunduhan paket secara otomatis. Skrip ini dapat dilengkapkan untuk melakukan peningkatan otomatis paket dengan memasang paket `unattended-upgrades`. Ini dapat disesuaikan dengan parameter dalam "`/etc/apt/apt.conf.d/02backup`" dan "`/etc/apt/apt.conf.d/50unattended-upgrades`" seperti yang dijelaskan dalam "`/usr/share/doc/unatten`

Paket `unattended-upgrade` terutama ditujukan untuk peningkatan keamanan untuk sistem `stable`. Jika risiko merusak sistem `stable` yang ada dengan peningkatan otomatis lebih kecil dari sistem yang rusak oleh penyusup menggunakan lubang keamanan yang telah ditutup oleh pembaruan keamanan, Anda harus mempertimbangkan untuk menggunakan peningkatan otomatis ini dengan parameter konfigurasi sebagai berikut.

```
APT::Periodic::Update-Package-Lists "1";
APT::Periodic::Download-Upgradeable-Packages "1";
APT::Periodic::Unattended-Upgrade "1";
```

Jika Anda menjalankan sistem `testing` atau `unstable`, Anda tidak ingin menggunakan peningkatan otomatis karena pasti merusak sistem suatu hari nanti. Bahkan untuk kasus `testing` atau `unstable` seperti itu, Anda mungkin masih ingin mengunduh paket terlebih dahulu untuk menghemat waktu untuk peningkatan interaktif dengan parameter konfigurasi sebagai berikut.

```
APT::Periodic::Update-Package-Lists "1";
APT::Periodic::Download-Upgradeable-Packages "1";
APT::Periodic::Unattended-Upgrade "0";
```

2.7.4 Pembaruan dan Backport

Ada arsip stable-updates ("`trixie`-updates" selama siklus rilis `trixie`-as-`stable`) dan backports.debian.org yang menyediakan paket-paket peningkatan untuk `stable`.

Untuk menggunakan arsip ini, Anda mencantumkan semua arsip yang diperlukan dalam berkas "`/etc/apt/sources.list`" seperti berikut ini.

```
deb http://deb.debian.org/debian/ trixie main non-free-firmware contrib non-free
deb http://security.debian.org/debian-security trixie-security main non-free-firmware ↩
    contrib non-free
deb http://deb.debian.org/debian/ trixie-updates main non-free-firmware contrib non-free
deb http://deb.debian.org/debian/ trixie-backports main non-free-firmware contrib non-free
```

Tidak perlu mengatur nilai Pin-Priority secara eksplisit dalam berkas "`/etc/apt/preferences`". Ketika paket yang lebih baru tersedia, konfigurasi baku menyediakan peningkatan yang paling masuk akal (lihat Bagian 2.5.3).

- Semua paket lama yang terpasang ditingkatkan ke yang lebih baru dari `trixie-updates`.
- Hanya paket lama yang dipasang secara manual dari `trixie-backports` yang ditingkatkan ke yang lebih baru dari `trixie-backports`.

Setiap kali Anda ingin memasang paket bernama "*`nama-paket`*" dengan ketergantungannya dari arsip `trixie-backports` secara manual, Anda menggunakan perintah berikut saat beralih rilis target dengan opsi "`-t`".

```
$ sudo apt-get install -t trixie-backports package-name
```



Awas
Jangan memasang terlalu banyak paket dari arsip backports.debian.org. Itu bisa menyebabkan komplikasi kebergantungan paket. Lihat Bagian 2.1.11 untuk solusi-solusi alternatif.

2.7.5 Arsip paket eksternal



Awas
Anda harus menyadari bahwa paket eksternal mendapatkan hak root ke sistem Anda. Anda hanya boleh menggunakan arsip paket eksternal terpercaya. Lihat Bagian 2.1.11 untuk solusi alternatif.

Anda dapat menggunakan APT yang aman dengan arsip paket eksternal yang kompatibel dengan Debian dengan menambahkannya ke **daftar sumber** dan berkas kunci arsip ke direktori "`/etc/apt/trusted.gpg.d/`". Lihat sources.list(5), apt-secure(8), dan apt-key(8).

2.7.6 Paket dari campuran sumber arsip tanpa apt-pinning



Perhatian
Memasangl paket dari sumber campuran arsip tidak didukung oleh distribusi Debian resmi kecuali untuk kombinasi tertentu dari arsip yang secara resmi didukung seperti `stable` dengan pemutakhiran keamanan dan stable-updates.

Berikut adalah contoh operasi untuk memasukkan paket versi hulu baru tertentu yang ditemukan di `unstable` sambil melacak `testing` untuk satu kesempatan.

1. Ubah berkas "`/etc/apt/sources.list`" untuk sementara menjadi entri tunggal "`unstable`".
2. Jalankan "`aptitude update`".
3. Jalankan "`aptitude install nama-paket`".
4. Pulihkan berkas asli "`/etc/apt/sources.list`" untuk `testing`.
5. Jalankan "`aptitude update`".

Anda tidak membuat berkas "`/etc/apt/preferences`" atau khawatir tentang **apt-pinning** dengan pendekatan manual ini. Tapi ini sangat rumit.



Perhatian
Saat menggunakan sumber arsip campuran, Anda harus memastikan sendiri kompatibilitas paket karena Debian tidak menjaminnya. Jika ketidakcocokan paket ada, Anda dapat merusak sistem. Anda harus dapat menilai persyaratan teknis ini. Penggunaan sumber campuran arsip acak benar-benar merupakan operasi opsional dan penggunaannya bukan sesuatu yang saya dorong Anda untuk lakukan.

Aturan umum untuk memasang paket dari arsip yang berbeda adalah sebagai berikut.

- Paket non-biner (”`Architecture: all`”) **lebih aman** untuk dipasang.
  - paket dokumentasi: tidak ada persyaratan khusus
  - paket program interpreter: interpreter yang kompatibel harus tersedia
- Paket biner (non ”`Architecture: all`”) biasanya menghadapi banyak hambatan dan **tidak aman** untuk dipasang.
  - kompatibilitas versi pustaka (termasuk ”`libc`”)
  - kompatibilitas versi program utilitas terkait
  - Kompatibilitas ABI kernel
  - Kompatibilitas ABI C++
  - …

Catatan
Untuk membuat paket menjadi **lebih aman** untuk dipasang, beberapa paket program biner komersial non-free dapat disediakan dengan pustaka yang sepenuhnya di-link secara statis. Anda masih harus memeriksa masalah kompatibilitas ABI dsb. untuk mereka.

Catatan
Kecuali untuk menghindari paket rusak dalam jangka pendek, memasang paket biner dari arsip non-Debian umumnya ide buruk. Anda harus mencari semua alternatif solusi teknis yang lebih aman yang kompatibel dengan sistem Debian Anda saat ini (lihat Bagian 2.1.11).

2.7.7 Menyetel halus versi kandidat dengan apt-pinning



Awas
Penggunaan teknik **apt-pinning** oleh pengguna pemula itu mengundang masalah besar. Anda harus menghindari menggunakan teknik ini kecuali ketika Anda benar-benar membutuhkannya. Lihat Bagian 2.1.11 untuk solusi alternatif.

Tanpa berkas ”`/etc/apt/preferences`”, sistem APT memilih versi terbaru yang tersedia sebagai **versi kandidat** menggunakan string versi. Ini adalah keadaan normal dan penggunaan sistem APT yang paling direkomendasikan. Semua kombinasi arsip yang didukung secara resmi tidak memerlukan berkas ”`/etc/apt/preferences`” karena beberapa arsip yang tidak boleh digunakan sebagai sumber otomatis peningkatan ditandai sebagai **NotAutomatic** dan ditangani dengan benar.

Tip
Aturan perbandingan string versi dapat diverifikasi dengan, misalnya, ”`dpkg --compare-versions ver1.1 gt ver1.1~1; echo $?`” (lihat dpkg(1)).

Ketika Anda memasang paket dari sumber campuran arsip (lihat Bagian 2.7.6) secara teratur, Anda dapat mengotomatisasi operasi rumit ini dengan membuat berkas "`/etc/apt/preferences`" dengan entri yang tepat dan mengutak-atik aturan pemilihan paket untuk **versi kandidat** seperti yang dijelaskan dalam apt_preferences(5). Ini disebut **apt-pinning**.

Saat menggunakan **apt-pinning**, Anda harus memastikan kompatibilitas paket sendiri karena Debian tidak menjaminnya. **Apt-pinning** adalah operasi yang benar-benar opsional dan penggunaannya bukanlah sesuatu yang saya anjurkan untuk Anda gunakan.

Berkas Release tingkat arsip (lihat Bagian 2.5.3) digunakan untuk aturan apt_preferences(5). Jadi **apt-pinning** bekerja hanya dengan nama "suite" untuk arsip Debian normal dan arsip Debian keamanan. (Ini berbeda dari arsip Ubuntu.) Misalnya, Anda dapat melakukan "`Pin: release a=unstable`" tetapi tidak dapat melakukan "`Pin: release a=sid`" di berkas "`/etc/apt/preferences`".

Ketika Anda menggunakan arsip non-Debian sebagai bagian dari **apt-pinning**, Anda harus memeriksa apa yang mereka maksudkan untuk dan juga memeriksa kredibilitas mereka. Misalnya, Ubuntu dan Debian tidak dimaksudkan untuk dicampur.

Catatan
Bahkan jika Anda tidak membuat berkas "`/etc/apt/preferences`", Anda dapat melakukan operasi sistem yang cukup kompleks (lihat Bagian 2.6.3 dan Bagian 2.7.6) tanpa **apt-pinning**.

Berikut adalah penjelasan sederhana tentang teknik **apt-pinning**.

Sistem APT memilih paket **peningkatan** dengan Pin-Priority tertinggi dari sumber paket yang tersedia yang didefinisikan dalam berkas "`/etc/apt/sources.list`" sebagai paket **versi kandidat**. Jika Pin-Priority paket lebih besar dari 1000, pembatasan versi ini untuk **peningkatan** dihapus untuk mengaktifkan penurunan tingkat (lihat Bagian 2.7.11).

Nilai Pin-Priority dari setiap paket didefinisikan oleh entri "Pin-Priority" dalam berkas "`/etc/apt/preferences`" atau menggunakan nilai defaultnya.

Pin-Priority	**efek apt-pinning ke paket**
1001	pasang paket bahkan jika ini merupakan penurunan tingkat paket
990	digunakan sebagai baku untuk arsip **rilis target**
500	digunakan sebagai baku untuk arsip **normal**
100	digunakan sebagai baku untuk arsip **NotAutomatic** dan **ButAutomaticUpgrades**
100	digunakan untuk paket **terpasang**
1	digunakan sebagai baku untuk arsip **NotAutomatic**
-1	**jangan pernah pasang** paket bahkan jika direkomendasikan

Tabel 2.18: Daftar nilai Pin-Priority yang terkenal untuk teknik **apt-pinning**.

Arsip **rilis target** dapat ditata dengan opsi baris perintah, mis., "`apt-get install -t testing paket-anu`"

Arsip **NotAutomatic** dan **ButAutomaticUpgrades** diatur oleh server arsip yang memiliki berkas Release tingkat arsip (lihat Bagian 2.5.3) yang berisi "`NotAutomatic: yes`" dan "`ButAutomaticUpgrades: yes`". Arsip **NotAutomatic** diatur oleh server arsip yang memiliki berkas Release tingkat arsip yang hanya berisi "`NotAutomatic: yes`".

Situasi apt-pinning *paket* dari beberapa sumber arsip ditampilkan oleh "`apt-cache policy paket`".

- Baris yang diawali dengan "`Package pin:`" mencantumkan daftar versi paket **pin** jika asosiasi hanya dengan *paket* didefinisikan, mis., "`Package pin: 0.190`".
- Tidak ada baris dengan "`Package pin:`" yang ada jika tidak ada asosiasi hanya dengan *paket* yang didefinisikan.
- Nilai Pin-Priority yang terkait hanya dengan *paket* tercantum di sisi kanan semua string versi, mis., "`0.181 700`".

- ”0” terdaftar di sisi kanan semua string versi jika tidak ada asosiasi hanya dengan *paket* yang didefinisikan, mis., ”0.181 0”.
- Nilai Pin-Priority arsip (didefinisikan sebagai ”Package: *” dalam berkas ”/etc/apt/preferences”) terdaftar di sisi kiri semua jalur arsip, mis., ”100 http://deb.debian.org/debian/ trixie-backports/main Packages”.

2.7.8 Memblokir paket yang dipasang oleh ”Recommends”



Awas
Penggunaan teknik **apt-pinning** oleh pengguna pemula itu mengundang masalah besar. Anda harus menghindari menggunakan teknik ini kecuali ketika Anda benar-benar membutuhkannya. Lihat Bagian 2.1.11 untuk solusi alternatif.

Jika Anda ingin tidak menarik paket tertentu secara otomatis dengan ”Recommends”, Anda harus membuat berkas ”/etc/apt/preferences” dan secara eksplisit mencantumkan semua paket tersebut di bagian atasnya sebagai berikut ini.

```
Package: package-1
Pin: version *
Pin-Priority: -1

Package: package-2
Pin: version *
Pin-Priority: -1
```

2.7.9 Melacak `testing` dengan beberapa paket dari `unstable`



Awas
Penggunaan teknik **apt-pinning** oleh pengguna pemula itu mengundang masalah besar. Anda harus menghindari menggunakan teknik ini kecuali ketika Anda benar-benar membutuhkannya. Lihat Bagian 2.1.11 untuk solusi alternatif.

Berikut adalah contoh teknik **apt-pinning** untuk menyertakan paket versi hulu baru tertentu yang ditemukan dalam `unstable` secara teratur ditingkatkan saat melacak `testing`. Anda mencantumkan semua arsip yang diperlukan dalam berkas ”/etc/apt/sources.list” dengan cara berikut ini.

```
deb http://deb.debian.org/debian/ testing main contrib non-free
deb http://deb.debian.org/debian/ unstable main contrib non-free
deb http://security.debian.org/debian-security testing-security main contrib
```

Atur berkas ”/etc/apt/preferences” dengan cara berikut ini.

```
Package: *
Pin: release a=unstable
Pin-Priority: 100
```

Ketika Anda ingin memasang paket bernama ”*nama-paket*” dengan dependensinya dari arsip `unstable` di bawah konfigurasi ini, Anda mengeluarkan perintah berikut yang mengalihkan rilis target dengan opsi ”-t” (Pin-Priority `unstable` menjadi 990).

```
$ sudo apt-get install -t unstable package-name
```

Dengan konfigurasi ini, eksekusi biasa "`apt-get upgrade`" dan "`apt-get dist-upgrade`" (atau "`aptitude safe-upgrade`" dan "`aptitude full-upgrade`") meningkatkan paket-paket yang dipasang dari arsip `testing` menggunakan arsip `testing` saat ini dan paket-paket yang dipasang dari arsip `unstable` menggunakan arsip `unstable` saat ini.



Perhatian
Berhati-hatilah untuk tidak menghapus entri "`testing`" entri dari "`/etc/apt/sources.list`". Tanpa entri "`testing` di dalamnya, sistem APT meningkatkan paket menggunakan arsip `unstable` yang lebih baru.

Tip
Saya biasanya menyunting berkas "`/etc/apt/sources.list`" untuk mengomentari entri arsip "`unstabil`" tepat setelah operasi di atas. Ini menghindari proses pembaruan yang lambat karena memiliki terlalu banyak entri dalam berkas "`/etc/apt/sources.list`" meskipun ini mencegah peningkatan paket yang dipasang dari arsip `unstable` menggunakan arsip `unstable` saat ini.

Tip
Jika "`Pin-Priority: 1`" digunakan sebagai pengganti "`Pin-Priority: 100`" dalam berkas "`/etc/apt/preferences`", paket yang sudah diinstall dengan nilai Pin-Priority 100 tidak ditingkatkan oleh arsip `unstable` bahkan jika entri "`testing`" di "`/etc/apt/sources.list`" dihapus.

Jika Anda ingin melacak paket tertentu dalam `unstable` secara otomatis tanpa instalasi awal "`-t unstable`", Anda harus membuat berkas "`/etc/apt/preferences`" dan secara eksplisit mencantumkan semua paket tersebut di bagian atasnya sebagai berikut.

```
Package: package-1
Pin: release a=unstable
Pin-Priority: 700

Package: package-2
Pin: release a=unstable
Pin-Priority: 700
```

Ini menetapkan nilai Pin-Priority untuk setiap paket tertentu. Misalnya, untuk melacak versi `unstable` terbaru dari "Referensi Debian" ini dalam bahasa Inggris, Anda harus memiliki entri berikut dalam berkas "`/etc/apt/preferences`".

```
Package: debian-reference-en
Pin: release a=unstable
Pin-Priority: 700

Package: debian-reference-common
Pin: release a=unstable
Pin-Priority: 700
```

Tip
Teknik **apt-pinning** ini berlaku bahkan ketika Anda melacak arsip `stable`. Sejauh ini, paket dokumentasi selalu aman untuk dipasang dari arsip `unstable` dalam pengalaman saya.

2.7.10 Pelacakan `unstable` dengan beberapa paket dari `experimental`



Awas
Penggunaan teknik **apt-pinning** oleh pengguna pemula itu mengundang masalah besar. Anda harus menghindari menggunakan teknik ini kecuali ketika Anda benar-benar membutuhkannya. Lihat Bagian 2.1.11 untuk solusi alternatif.

Berikut adalah contoh lain dari teknik **apt-pinning** untuk memasukkan paket versi hulu baru tertentu yang ditemukan dalam `experimental` sambil melacak `unstable`. Anda mencantumkan semua arsip yang diperlukan dalam berkas "`/etc/apt/sources.list`" sebagai berikut ini.

```
deb http://deb.debian.org/debian/ unstable main contrib non-free
deb http://deb.debian.org/debian/ experimental main contrib non-free
deb http://security.debian.org/ testing-security main contrib
```

Nilai Pin-Priority baku untuk arsip `experimental` selalu 1 (<<100) karena merupakan arsip **NotAutomatic** (lihat Bagian 2.5.3). Tidak perlu mengatur nilai Pin-Priority secara eksplisit dalam berkas "`/etc/apt/preferences`" hanya untuk menggunakan arsip `experimental` kecuali Anda ingin melacak paket tertentu di dalamnya secara otomatis untuk peningkatan berikutnya.

2.7.11 Penurunan tingkat darurat



Awas
Penggunaan teknik **apt-pinning** oleh pengguna pemula itu mengundang masalah besar. Anda harus menghindari menggunakan teknik ini kecuali ketika Anda benar-benar membutuhkannya. Lihat Bagian 2.1.11 untuk solusi alternatif.



Perhatian
Penurunan tingkat tidak secara resmi didukung oleh Debian secara desain. Ini harus dilakukan hanya sebagai bagian dari proses pemulihan darurat. Terlepas dari situasi ini, telah diketahui bekerja dengan baik dalam banyak insiden. Untuk sistem kritis, Anda harus mencadangkan semua data penting pada sistem setelah operasi pemulihan dan memasang ulang sistem baru dari awal.

Anda mungkin beruntung untuk menurun-tingkatkan dari arsip yang lebih baru ke arsip lama untuk memulihkan dari peningkatan sistem yang rusak dengan memanipulasi **versi kandidat** (lihat Bagian 2.7.7). Ini adalah alternatif malas untuk tindakan membosankan dari banyak perintah "`dpkg -i paket-rusak_versi-lama.deb`" perintah (lihat Bagian 2.6.3).

Cari baris-baris di berkas "`/etc/apt/sources.list`" yang melacak `unstable` sebagai berikut.

```
deb http://deb.debian.org/debian/ sid main contrib non-free
```

Ganti dengan yang berikut untuk melacak `testing`.

```
deb http://deb.debian.org/debian/ forky main contrib non-free
```

Atur berkas "`/etc/apt/preferences`" dengan cara berikut ini.

```
Package: *
Pin: release a=testing
Pin-Priority: 1010
```

Jalankan "`apt-get update; apt-get dist-upgrade`" untuk memaksa penurunan tingkat paket di seluruh sistem.

Hapus berkas khusus "`/etc/apt/preferences`" ini setelah penurunan tingkat darurat ini.

Tip
Ide yang baik untuk menghapus (tidak membersihkan!) sebanyak mungkin paket untuk meminimalkan masalah ketergantungan. Anda mungkin perlu menghapus dan memasang beberapa paket secara manual untuk menurun-tingkatkan sistem secara manual. Kernel Linux, bootloader, udev, PAM, APT, dan paket terkait jaringan serta berkas-berkas konfigurasi mereka membutuhkan perhatian khusus.

2.7.12 Paket equivs

Jika Anda ingin mengkompilasi program dari sumber untuk menggantikan paket Debian, yang terbaik adalah membuatnya menjadi paket debianized lokal nyata (`*.deb`) dan menggunakan arsip pribadi.

Jika Anda memilih untuk mengkompilasi program dari sumber dan menginstalnya di bawah "`/usr/local`" sebagai gantinya, Anda mungkin perlu menggunakan `equivs` sebagai upaya terakhir untuk memenuhi ketergantungan paket yang hilang.

```
Package: equivs
Priority: optional
Section: admin
Description: Circumventing Debian package dependencies
 This package provides a tool to create trivial Debian packages.
 Typically these packages contain only dependency information, but they
 can also include normal installed files like other packages do.
 .
 One use for this is to create a metapackage: a package whose sole
 purpose is to declare dependencies and conflicts on other packages so
 that these will be automatically installed, upgraded, or removed.
 .
 Another use is to circumvent dependency checking: by letting dpkg
 think a particular package name and version is installed when it
 isn't, you can work around bugs in other packages' dependencies.
 (Please do still file such bugs, though.)
```

2.7.13 Mem-port paket ke sistem stable



Perhatian
Tidak ada garansi bahwa prosedur yang diuraikan di sini bekerja tanpa upaya manual ekstra bagi perbedaan sistem.

Untuk peningkatan parsial dari sistem `stable`, membangun kembali paket dalam lingkungannya menggunakan paket sumber itu diinginkan. Ini menghindari peningkatan paket besar-besaran karena ketergantungan mereka.

Tambahkan entri berikut ke "`/etc/apt/sources.list`" dari sistem `stable`.

```
deb-src http://deb.debian.org/debian unstable  main contrib non-free
```

Pasang paket yang diperlukan untuk kompilasi dan unduh paket sumber sebagai berikut.

```
# apt-get update
# apt-get dist-upgrade
# apt-get install fakeroot devscripts build-essential
# apt-get build-dep foo
$ apt-get source foo
$ cd foo*
```

Perbarui beberapa paket rantai alat seperti `dpkg` dan `debhelper` dari paket backport jika mereka diperlukan untuk mem-backport.

Jalankan hal-hal berikut.

```
$ dch -i
```

Naikkan versi paket, mis. dengan menambahkan "`+bp1`" di "`debian/changelog`"

Bangun paket dan pasang mereka ke sistem sebagai berikut.

```
$ debuild
$ cd ..
# debi foo*.changes
```

2.7.14 Server proksi untuk APT

Karena mencerminkan seluruh sub bagian arsip Debian membuang ruang disk dan bandwidth jaringan, penggelaran server proksi lokal untuk APT adalah pertimbangan yang diinginkan ketika Anda mengelola banyak sistem pada LAN. APT dapat dikonfigurasi untuk menggunakan server proxy web generik (http) seperti `squid` (lihat Bagian 6.5) seperti yang dijelaskan dalam apt.conf(5) dan dalam "`/usr/share/doc/apt/examples/configure-index.gz`". Variabel lingkungan "`$http_proxy`" dapat digunakan untuk menimpa pengaturan server proksi dalam berkas "`/etc/apt/apt.c`

Ada alat proksi khusus untuk arsip Debian. Anda harus memeriksa BTS sebelum menggunakannya.

paket	popcon	ukuran	deskripsi
`apt-cacher`	V:0.28, I:0.33	265	Proksi penyinggahan untuk berkas sumber dan paket Debian (program Perl)
`apt-cacher-ng`	V:3.9, I:4.1	2063	Proksi penyinggahan untuk distribusi paket perangkat lunak (program C++ yang dikompilasi)

Tabel 2.19: Daftar alat proksi khusus untuk arsip Debian



Perhatian
Ketika Debian mereorganisasi struktur arsipnya, alat proksi khusus ini cenderung memerlukan penulisan ulang kode oleh pengelola paket dan mungkin tidak berfungsi untuk sementara waktu. Di sisi lain, server proksi web generik (http) lebih kuat dan lebih mudah untuk mengatasi perubahan tersebut.

2.7.15 Lebih banyak bacaan untuk manajemen paket

Anda dapat mempelajari lebih lanjut tentang manajemen paket dari dokumentasi berikut.

- Dokumentasi primer tentang manajemen paket:
  - aptitude(8), dpkg(1), tasksel(8), apt(8), apt-get(8), apt-config(8), apt-secure(8), sources.list(5), apt.conf(5), dan apt_preferences(5);

  - "`/usr/share/doc/apt-doc/guide.html/index.html`" dan "`/usr/share/doc/apt-doc/offline.html/in`
    dari paket `apt-doc`; dan
  - "`/usr/share/doc/aptitude/html/en/index.html`" dari paket `aptitude-doc-en`.
- Dokumentasi resmi dan terperinci pada arsip Debian:
  - "Manual Kebijakan Debian Bab 2 - Arsip Debian",
  - "Acuan Pengembang Debian, Bab 4 - Sumber Daya bagi Para Pengembang Debian 4.6 Arsip Debian", dan
  - "FAQ Debian GNU/Linux, Bab 6 - Arsip FTP Debian".
- Tutorial untuk membangun paket Debian untuk pengguna Debian:
  - "Panduan bagi Para Pemelihara Debian".

Bab 3

Inisialisasi sistem

Adalah bijaksana bagi Anda sebagai administrator sistem untuk mengetahui kira-kira bagaimana sistem Debian dimulai dan dikonfigurasi. Meskipun rincian yang tepat ada di berkas sumber paket yang dipasang dan dokumentasinya, itu agak membuat kewalahan bagi kebanyakan dari kita.

Berikut adalah gambaran kasar dari poin-poin penting dari inisialisasi sistem Debian. Karena sistem Debian adalah target yang bergerak, Anda harus merujuk ke dokumentasi terbaru.

- Buku Pegangan Kernel Linux Debian adalah sumber informasi primer atas kernel Debian.
- bootup(7) menggambarkan proses boot sistem yang berbasis `systemd`. (Debian terbaru)
- boot(7) menjelaskan proses boot sistem berbasis UNIX System V Release 4. (Debian Yang Lebih Tua)

3.1 Ringkasan proses boot strap

Sistem komputer mengalami beberapa fase proses boot strap dari peristiwa daya dinyalakan sampai menawarkan sistem operasi (OS) yang berfungsi penuh kepada pengguna.

Untuk kesederhanaan, saya membatasi diskusi ke platform PC umum dengan instalasi baku.

Proses boot strap yang umum itu seperti roket empat tahap. Setiap tahap roket menyerahkan kontrol sistem ke tahap berikutnya.

- Bagian 3.1.1
- Bagian 3.1.2
- Bagian 3.1.3
- Bagian 3.1.4

Tentu saja, ini dapat dikonfigurasi secara berbeda. Misalnya, jika Anda menyusun kernel Anda sendiri, Anda mungkin melewatkan langkah dengan sistem mini-Debian. Jadi tolong jangan berasumsi ini adalah kasus untuk sistem Anda sampai Anda memeriksanya sendiri.

3.1.1 Tahap 1: UEFI

Unified Extensible Firmware Interface (UEFI) mendefinisikan boot manager sebagai bagian dari spesifikasi UEFI. Ketika komputer dinyalakan, boot manager adalah tahap pertama dari proses boot yang memeriksa konfigurasi boot dan berdasarkan pengaturannya, kemudian mengeksekusi boot loader OS atau kernel sistem operasi yang ditentukan

(biasanya boot loader). Konfigurasi boot didefinisikan oleh variabel yang disimpan dalam NVRAM, termasuk variabel yang menunjukkan path sistem berkas ke loader OS atau kernel OS.

EFI system partition (ESP) adalah partisi perangkat penyimpanan data yang digunakan dalam komputer yang mengikuti spesifikasi UEFI. Diakses oleh firmware UEFI ketika komputer dinyalakan, ia menyimpan aplikasi UEFI dan berkas-berkas yang perlu dijalankan aplikasi ini, termasuk boot loader sistem operasi. (Pada sistem PC warisan, BIOS yang disimpan dalam MBR dapat digunakan sebagai gantinya.)

3.1.2 Tahap 2: boot loader

Boot loader adalah tahap ke-2 dari proses boot yang dimulai oleh UEFI. Ini memuat image kernel sistem dan image initrd ke memori dan memindahkan kontrol kepada mereka. Image initrd ini adalah image sistem berkas root dan dukungannya tergantung pada bootloader yang digunakan.

Sistem Debian biasanya menggunakan kernel Linux sebagai kernel sistem baku. Image initrd untuk kernel Linux 5.x saat ini secara teknis adalah image initramfs (sistem berkas RAM awal).

Ada banyak boot loader dan opsi konfigurasi yang tersedia.

paket	popcon	ukuran	initrd	bootloader	deskripsi
grub-efi-amd64	I:466	142	Didukung	GRUB UEFI	Ini cukup pintar untuk memahami partisi disk dan sistem berkas seperti vfat, ext4, ... (UEFI)
grub-pc	V:16, I:512	479	Didukung	GRUB 2	Ini cukup pintar untuk memahami partisi disk dan sistem berkas seperti vfat, ext4, ... (BIOS)
grub-rescue-pc	V:0.05, I:0.58	7269	Didukung	GRUB 2	Ini adalah image penyelamatan GRUB 2 yang dapat di-boot (CD dan floppy) (versi PC/BIOS)
grml-rescueboot	V:0.1, I:1.0	39	T/T	Plugin GRUB 2	grml-rescueboot menambahkan image ISO Linux live ke menu boot grub2
syslinux	V:3, I:31	325	Didukung	Isolinux	Ini memahami sistem berkas ISO9660. Ini digunakan oleh CD boot.
syslinux	V:3, I:31	325	Didukung	Syslinux	Ini memahami sistem berkas MSDOS (FAT). Ini digunakan oleh boot floppy.
loadlin	V:0.03, I:0.39	87	Didukung	Loadlin	Sistem baru dimulai dari sistem FreeDOS/MSDOS.
mbr	V:0.2, I:2.7	47	Tidak didukung	MBR oleh Neil Turton	Ini adalah perangkat lunak bebas yang menggantikan MBR MSDOS. Ini hanya memahami partisi disk.

Tabel 3.1: Daftar boot loader

Untuk sistem UEFI, GRUB2 terlebih dahulu membaca partisi ESP dan menggunakan UUID yang ditentukan untuk `search.fs_uuid` di "`/boot/efi/EFI/debian/grub.cfg`" untuk menentukan partisi berkas konfigurasi menu GRUB2 "`/boot/grub/grub.cfg`".

Bagian penting dari berkas konfigurasi menu GRUB2 terlihat seperti:

```
menuentry 'Debian GNU/Linux' ... {
        load_video
        insmod gzio
        insmod part_gpt
        insmod ext2
        search --no-floppy --fs-uuid --set=root fe3e1db5-6454-46d6-a14c-071208ebe4b1
```

```
        echo    'Loading Linux 5.10.0-6-amd64 ...'
        linux   /boot/vmlinuz-5.10.0-6-amd64 root=UUID=fe3e1db5-6454-46d6-a14c-071208ebe4b1 ↩
            ro quiet
        echo    'Loading initial ramdisk ...'
        initrd  /boot/initrd.img-5.10.0-6-amd64
}
```

Untuk bagian `/boot/grub/grub.cfg` ini, entri menu ini berarti sebagai berikut.

setelan	**nilai**
Modul GRUB2 dimuat	`gzio`, `part_gpt`, `ext2`
partisi sistem berkas root yang digunakan	partisi yang diidentifikasi oleh `UUID=fe3e1db5-6454-46d6-a14c-071208ebe4b1`
path image kernel dalam sistem berkas root	`/boot/vmlinuz-5.10.0-6-amd64`
parameter boot kernel yang digunakan	”`root=UUID=fe3e1db5-6454-46d6-a14c-071208ebe4b1 ro quiet`”
path image initrd dalam sistem berkas root	`/boot/initrd.img-5.10.0-6-amd64`

Tabel 3.2: Arti dari entri menu dari bagian di atas dari `/boot/grub/grub.cfg`

Pada sistem Debian, ”`/boot/grub/grub.cfg`” dikelola oleh paket GRUB yang terinstall (misalnya `grub-efi-amd64`) dan modifikasi langsung oleh pengguna pada berkas ini sudah tidak dianjurkan. Anda sebaiknya menyesuaikan berkas konfigurasi di ”`/etc/grub.d/`” dan ”`/etc/default/grub`”. Kemudian konfigurasikan berkas konfigurasi GRUB dan perbarui variabel NVRAM untuk boot otomatis ke Debian dengan:

```
# dpkg-reconfigure grub-efi-amd64
```

Tip
Anda dapat menampilkan pesan log boot kernel dengan menghapus ”`quiet`” dari nilai ”`GRUB_CMDLINE_LINUX_DEFAULT`” di ”`/etc/default/grub`”.

Tip
Anda dapat menambahkan latar belakang splash GRUB dengan menempatkan berkas gambarnya di ”`/boot/grub/`”.

Lihat ”`info grub`”, grub-install(8), grub-mkconfig(8).

3.1.3 Tahap 3: sistem mini-Debian

Sistem mini-Debian adalah tahap ke-3 dari proses boot yang dimulai oleh boot loader. Ini menjalankan kernel sistem dengan sistem berkas root pada memori. Ini adalah tahap persiapan opsional dari proses boot.

Catatan
Istilah ”sistem mini-Debian” diciptakan oleh penulis untuk menggambarkan proses boot tahap ke-3 ini untuk dokumen ini. Sistem ini sering disebut sebagai sistem initrd atau initramfs. Sistem serupa pada memori digunakan oleh Debian Installer.

Program ”`/init`” dijalankan sebagai program pertama dalam sistem berkas root ini pada memori. Ini adalah program yang menginisialisasi kernel di ruang pengguna dan mewariskan kendali ke tahap berikutnya. Sistem mini-Debian ini menawarkan fleksibilitas untuk proses boot seperti menambahkan modul kernel sebelum proses boot utama atau mengait sistem berkas root sebagai yang dienkripsi.

- Program "`/init`" adalah program skrip shell jika initramfs dibuat oleh `initramfs-tools`.
  - Anda dapat menginterupsi bagian dari proses boot ini untuk mendapatkan shell root dengan memberikan "`break=init`" dll. bagi parameter boot kernel. Lihat skrip "`/init`" untuk kondisi istirahat lainnya. Lingkungan shell ini cukup canggih untuk melakukan inspeksi yang baik terhadap perangkat keras mesin Anda.
  - Perintah yang tersedia dalam sistem mini-Debian ini adalah yang dirampingkan dan terutama disediakan oleh alat GNU yang disebut busybox(1).
- Program "`/init`" adalah program biner `systemd` jika initramfs diciptakan oleh `dracut`.
  - Perintah yang tersedia dalam sistem mini-Debian adalah lingkungan systemd(1) yangn dirampingkan.



Perhatian
Anda perlu menggunakan opsi "`-n`" untuk perintah `mount` saat Anda berada di sistem berkas root yang hanya-baca.

3.1.4 Tahap 4: sistem Debian normal

Sistem Debian normal adalah tahap ke-4 dari proses boot yang dimulai oleh sistem mini-Debian. Kernel sistem untuk sistem mini-Debian terus berjalan di lingkungan ini. Sistem berkas root dialihkan dari yang ada di memori ke yang ada di perangkat penyimpanan fisik.

Program init dijalankan sebagai program pertama dengan PID=1 untuk melakukan proses boot utama yang memulai banyak program. Path berkas baku untuk program init adalah "`/usr/sbin/init`" tetapi dapat diubah oleh parameter boot kernel sebagai "`init=/path/ke/program_init`".

"`/usr/sbin/init`" di-symlink ke "`/lib/systemd/systemd`" setelah Debian 8 Jessie (dirilis pada tahun 2015).

Tip
Perintah init yang sebenarnya pada sistem Anda dapat diverifikasi oleh perintah "`ps --pid 1 -f`".

Tip
Lihat Wiki Debian: BootProcessSpeedup bagi tips terbaru untuk mempercepat proses boot.

3.2 Sistem penyelamat



Awas
Jangan lakukan tugas administrasi sistem di sekitar proses boot strap tanpa memiliki sistem penyelamat.

Ketersediaan sistem penyelamat memungkinkan kita melakukan tugas-tugas menantang seperti:

- Mem-boot sistem dari instalasi boot loader yang rusak
- Memperbaiki instalasi boot loader yang rusak
- Mengekstrak data dari sistem yang rusak dan tidak dapat di-boot

paket	popcon	ukuran	deskripsi
systemd	V:920, I:980	11386	daemon init(8) berbasis kejadian untuk konkurensi (alternatif dari sysvinit)
cloud-init	V:4.1, I:6.3	3267	sistem inisialisasi bagi instansi cloud infrastruktur
systemd-sysv	V:915, I:983	76	halaman manual dan tautan yang diperlukan bagi systemd untuk menggantikan sysvinit
init-system-helpers	V:723, I:988	132	alat-alat pembantu untuk beralih antara sysvinit dan systemd
initscripts	V:15, I:54	197	skrip untuk menginisialisasi dan mematikan sistem
sysvinit-core	V:3.1, I:3.6	365	Utilitas init(8) mirip System-V
sysv-rc	V:29, I:57	85	Mekanisme perubahan runlevel mirip System-V
sysvinit-utils	V:725, I:1000	100	Utilitas mirip System-V (startpar(8), bootlogd(8), …)
lsb-base	V:205, I:300	12	Fungsionalitas skrip init Linux Standard Base 3.2
insserv	V:31, I:57	132	alat untuk mengatur urutan boot menggunakan dependensi skrip init.d LSB
kexec-tools	V:1.4, I:5.3	320	alat kexec untuk mem-boot ulang kexec(8) (boot ulang warm)
systemd-bootchart	V:0.17, I:0.52	131	penganalisis kinerja proses boot
mingetty	V:0.2, I:2.3	36	getty(8) hanya konsol
mgetty	V:0.13, I:0.47	318	pengganti getty(8) smart modem

Tabel 3.3: Daftar utilitas boot untuk sistem Debian

- Mengedit sistem berkas, partisi disk, dan volume LVM yang melibatkan sistem berkas root

Biasanya, sistem penyelamat disediakan sebagai berkas image ISO dan ditulis ke media penyimpanan yang dapat dilepas seperti:

- USB flash drive yang disiapkan sebagai Bagian 9.7.2
- CD / DVD yang disiapkan sebagai Bagian 9.7.7

Demi kesederhanaan, kasus USB flash drive disebutkan sebagai contoh di bawah ini, tetapi CD atau DVD juga dapat digunakan.

Tip
Anda mungkin perlu mengubah beberapa variabel UEFI NVRAM untuk mem-boot boot loader sembarang pada media penyimpanan yang dapat dilepas.

3.2.1 Sistem penyelamat GRUB UEFI pada USB

Sistem penyelamat GRUB UEFI dengan menu dapat dimulai dengan menyalakan daya sistem saat ”Sistem penyelamat GRUB UEFI pada USB” terpasang.

”Sistem penyelamat GRUB UEFI pada USB” ini disiapkan dengan menulis image ISO Super Grub2 Disk ke USB flash drive terlebih dahulu.

Jika konfigurasi boot loader rusak akibat instalasi sistem operasi lain, dll., Anda dapat memperbaikinya dengan:

- Mulai sistem penyelamat GRUB UEFI untuk menemukan sistem yang terinstall dan dapat di-boot secara otomatis.
- Mulai sistem Debian yang terinstall dari menu GRUB.

- Pada prompt shell root Linux:

```
# dpkg-reconfigure grub-efi-amd64
```

Catatan
Image ISO penyelamat GRUB yang dapat di-boot juga dapat dibuat dengan mengikuti ”`info grub-mkrescue`”. Ini menawarkan prompt shell GRUB CLI tetapi tidak menawarkan penemuan otomatis sistem yang terinstall dan dapat di-boot.

3.2.2 Sistem penyelamat Linux live pada USB

Sistem penyelamat Linux live dapat dimulai dengan menyalakan daya sistem saat ”Sistem penyelamat Linux live pada USB” terpasang.

”Sistem penyelamat Linux live pada USB” ini dapat disiapkan dengan menulis salah satu image ISO Linux live berbasis Debian ke USB flash drive terlebih dahulu. Berikut beberapa contoh image Linux live tersebut.

- Image Debian Live
- Kali Linux Live
- Grml Live Linux

Berikut beberapa kasus penggunaan sistem penyelamat Linux live ini:

- Memperbaiki konfigurasi boot loader yang rusak akibat instalasi sistem operasi lain, dll.:
  - Mulai sistem penyelamat Linux live.
  - Pasang partisi yang berisi sistem berkas root dari sistem Debian yang terinstall dan tidak dapat di-boot ke ”`/mnt`”.
  - Pada prompt shell root Linux:

```
# chroot /mnt; dpkg-reconfigure grub-efi-amd64
```

- Memperbaiki paket `dpkg` yang rusak:
  - Mulai sistem penyelamat Linux live.
  - Pasang partisi yang berisi sistem berkas root dari sistem Debian yang terinstall dengan paket `dpkg` yang rusak ke ”`/mnt`”.
  - Pada prompt shell root Linux:

```
# dpkg --root /mnt -i /mnt/var/cache/apt/archives/dpkg_old_version_amd64.deb
```

- Lakukan perubahan yang biasanya dilarang seperti operasi sistem berkas pada sistem yang terinstall (lihat Bagian 9.6).



Awas
Layar GUI sistem Linux live Anda mungkin terkunci setelah tidak aktif.

Tip

- Sebaiknya atur kata sandi Anda segera setelah memulai sistem Linux live.
- Anda dapat mengatur kata sandi misalnya dengan "`sudo passwd user`" dari prompt shell pengguna di konsol virtual Linux (lihat Bagian 1.1.6).
- Beberapa sistem Linux live mungkin menetapkan "*user*/*password*" bawaan: "Debian Live" = "user/live", "Kali Linux Live" = "kali/kali"

3.2.3 Sistem penyelamat Linux live dari GRUB

Sistem penyelamat Linux live dapat dimulai dari entri menu GRUB. Konfigurasi GRUB untuk ini disiapkan dengan langkah berikut:

- Instal paket `grml-rescueboot`.
- Temukan image ISO Linux live yang memiliki "`/boot/grub/loopback.cfg`" di dalam imagenya.
  - Image ISO yang disebutkan di Bagian 3.2.2 memiliki "`/boot/grub/loopback.cfg`" di dalam imagenya.
- Salin beberapa image ISO Linux live ini ke direktori "`/boot/grml/`".
- Perbarui menu GRUB dengan:

```
# dpkg-reconfigure grub-efi-amd64
```

Saat menyalakan daya sistem, GRUB menampilkan menu dengan kandidat sistem penyelamat Linux live.

3.3 Systemd

3.3.1 Init systemd

Ketika sistem Debian mulai berjalan, `/usr/sbin/init` yang di-symlink ke `/usr/sbin/init` dimulai sebagai proses init sistem (`PID=1`) yang dimiliki oleh root (`UID=0`). Lihat systemd(1).

Proses init `systemd` men-spawn proses secara paralel berdasarkan berkas konfigurasi unit (lihat systemd.unit(5)) yang ditulis dalam gaya deklaratif, bukan gaya prosedural seperti SysV.

Proses yang di-spawn ditempatkan dalam kelompok kontrol Linux individu yang dinamai sesuai dengan unit yang mereka miliki dalam hierarki sistem pribadi (lihat cgroup dan Bagian 4.7.5).

Unit-unit untuk mode sistem dimuat dari "Path Pencarian Unit Sistem" yang diuraikan dalam systemd.unit(5). Yang utama adalah sebagai berikut dalam urutan prioritas:

- "`/etc/systemd/system/*`": Unit sistem yang dibuat oleh administrator
- "`/run/systemd/system/*`": Unit runtime
- "`/lib/systemd/system/*`": Unit sistem yang dipasang oleh manajer paket distribusi

Inter-dependensi mereka ditentukan oleh arahan "`Wants=`", "`Requires=`", "`Before=`", "`After=`", ... (lihat "PEMETAAN PROPERTI UNIT UNTUK INVERSI MEREKA" dalam systemd.unit(5)). Kontrol sumber daya juga didefinisikan (lihat systemd.resource-control(5)).

Akhiran berkas konfigurasi unit mengodekan jenisnya sebagai:

- ***.service** menjelaskan proses yang dikendalikan dan diawasi oleh `systemd`. Lihat systemd.service(5).
- ***.device** menggambarkan perangkat yang terpapar dalam sysfs(5) sebagai pohon perangkat udev(7). Lihat systemd.device(5).
- ***.mount** menggambarkan titik kait sistem berkas yang dikendalikan dan diawasi oleh `systemd`. Lihat systemd.mount(5).
- ***.automount** menggambarkan titik kait otomatis sistem berkas yang dikendalikan dan diawasi oleh `systemd`. Lihat systemd.automount(5).
- ***.swap** menggambarkan perangkat atau berkas swap yang dikontrol dan diawasi oleh `systemd`. Lihat systemd.swap(5).
- ***.path** menggambarkan path yang dipantau oleh `systemd` untuk aktivasi berbasis-path. Lihat systemd.path(5).
- ***.socket** menjelaskan soket yang dikendalikan dan diawasi oleh `systemd` untuk aktivasi berbasis soket. Lihat systemd.socket(5).
- ***.timer** menjelaskan timer yang dikendalikan dan diawasi oleh `systemd` untuk aktivasi berbasis timer. Lihat systemd.timer(5).
- ***.slice** mengelola sumber daya dengan cgroup(7). Lihat systemd.slice(5).
- ***.scope** dibuat secara pemrograman menggunakan antarmuka bus `systemd` untuk mengelola serangkaian proses sistem. Lihat systemd.scope(5).
- ***.target** kelompok berkas konfigurasi unit lainnya untuk membuat titik sinkronisasi selama start-up. Lihat systemd.target(5).

Saat sistem mulai dijalankan (yaitu, init), proses `systemd` mencoba untuk memulai "`/lib/systemd/system/default.ta`
(biasanya di-symlink ke "`graphical.target`"). Pertama, beberapa unit target khusus (lihat systemd.special(7)) seperti "`local-fs.target`", "`swap.target`", dan "`cryptsetup.target`" ditarik untuk mengait sistems berkas. Kemudian, unit target lainnya juga ditarik oleh dependensi unit target. Untuk detailnya, baca bootup(7).

`systemd` menawarkan fitur kompatibilitas mundur. Skrip boot gaya SysV dalam "`/etc/init.d/rc[0123456S].d/[KS]n`
masih diurai dan telinit(8) diterjemahkan ke dalam permintaan aktivasi unit systemd.



Perhatian
Runlevel 2 hingga 4 yang diemulasi semuanya di-symlink ke "`multi-user.target`" yang sama.

3.3.2 Login systemd

Ketika seorang pengguna log masuk ke sistem Debian melalui gdm3(8), sshd(8), dsb., `/lib/systemd/system --user` mulai dijalankan sebagai proses manajer layanan pengguna yang dimiliki oleh pengguna yang bersangkitan. Lihat systemd(1).

Proses manajer layanan pengguna `systemd` men-spawn proses secara paralel berdasarkan berkas konfigurasi unit (lihat systemd.unit(5)) dan `user@.service`(5)).

Unit untuk mode pengguna dimuat dari "Path Pencarian Unit Pengguna" yang dijelaskan dalam systemd.unit(5). Yang utama adalah sebagai berikut dalam urutan prioritas:

- "`~/.config/systemd/user/*`": Unit konfigurasi pengguna
- "`/etc/systemd/user/*`": Unit pengguna yang dibuat oleh administrator
- "`/run/systemd/user/*`": Unit runtime
- "`/lib/systemd/user/*`": Unit pengguna yang dipasang oleh manajer paket distribusi

Ini dikelola dengan cara yang sama dengan Bagian 3.3.1.

3.4 Pesan kernel

Pesan kesalahan kernel yang ditampilkan ke konsol dapat dikonfigurasi dengan mengatur tingkat ambang batasnya.

```
# dmesg -n3
```

nilai tingkat kesalahan	nama tingkat kesalahan	arti
0	KERN_EMERG	sistem tidak dapat digunakan
1	KERN_ALERT	tindakan harus segera diambil
2	KERN_CRIT	kondisi kritis
3	KERN_ERR	kondisi galat
4	KERN_WARNING	kondisi peringatan
5	KERN_NOTICE	kondisi normal namun signifikan
6	KERN_INFO	informasi
7	KERN_DEBUG	pesan tingkat debug

Tabel 3.4: Daftar tingkat kesalahan kernel

3.5 Pesan sistem

Di bawah `systemd`, baik kernel dan pesan sistem dicatat oleh layanan jurnal `systemd-journald.service` (alias `journald`) baik ke dalam data biner persisten di bawah "`/var/log/journal`" atau ke dalam data biner volatile di bawah "`/run/log/journal/`". Data log biner ini diakses oleh perintah journalctl(1). Misalnya, Anda dapat menampilkan log dari boot terakhir sebagai:

```
$ journalctl -b
```

Operasi	Cuplikan perintah
Melihat log untuk layanan-layanan sistem dan kernel dari boot terakhir	"`journalctl -b --system`"
Melihat log untuk layanan pengguna saat ini dari boot terakhir	"`journalctl -b --user`"
Melihat log pekerjaan "`$unit`" dari boot terakhir	"`journalctl -b -u $unit`"
Melihat log pekerjaan "`$unit`" (gaya "`tail -f`") dari boot terakhir	"`journalctl -b -u $unit -f`"

Tabel 3.5: Daftar cuplikan perintah `journalctl` yang umum

Di bawah `systemd`, utilitas log sistem rsyslogd(8) dapat dihapus. Jika terpasang, ia mengubah perilakunya untuk membaca data log biner volatil (bukan "`/dev/log`" baku pra-systemd) dan untuk membuat data log sistem ASCII permanen tradisional. Ini dapat disesuaikan dengan "`/etc/default/rsyslog`" dan "`/etc/rsyslog.conf`" baik untuk berkas log maupun tampilan layar. Lihat rsyslogd(8) dan rsyslog.conf(5). Lihat juga Bagian 9.3.2.

3.6 Manajemen sistem

`systemd` ini tidak hanya menawarkan sistem init tetapi juga operasi manajemen sistem generik dengan perintah systemctl(1).

Operasi	Cuplikan perintah
Daftar semua tipe unit yang tersedia	"`systemctl list-units --type=help`"
Daftar semua unit target dalam memori	"`systemctl list-units --type=target`"
Daftar semua unit layanan dalam memori	"`systemctl list-units --type=service`"
Daftar semua unit perangkat dalam memori	"`systemctl list-units --type=device`"
Cantumkan semua unit mount dalam memori	"`systemctl list-units --type=mount`"
Daftar semua unit soket dalam memori	"`systemctl list-sockets`"
Daftar semua unit timer dalam memori	"`systemctl list-timers`"
Start "`$unit`"	"`systemctl start $unit`"
Stop "`$unit`"	"`systemctl stop $unit`"
Memuat ulang konfigurasi spesifik layanan	"`systemctl reload $unit`"
Menghentikan dan memulai semua "`$unit`"	"`systemctl restart $unit`"
Memulai "`$unit`" dan menghentikan yang lainnya	"`systemctl isolate $unit`"
Beralih ke "`grafis`" (sistem GUI)	"`systemctl isolate graphical`"
Beralih ke "`multi-user`" (sistem CLI)	"`systemctl isolate multi-user`"
Beralih ke "`penyelamatan`" (sistem CLI pengguna tunggal)	"`systemctl isolate rescue`"
Mengirim sinyal kill ke "`$unit`"	"`systemctl kill $unit`"
Memeriksa apakah layanan "`$unit`" aktif	"`systemctl is-active $unit`"
Memeriksa apakah layanan "`$unit`" gagal	"`systemctl is-failed $unit`"
Memeriksa status "`$unit\|$PID|device`"	"`systemctl status $unit\|$PID|$device`"
Menampilkan properti "`$unit\|$job`"	"`systemctl show $unit\|$job`"
Me-reset "`$unit`" yang gagal	"`systemctl reset-failed $unit`"
Menampilkan daftar dependensi semua layanan unit	"`systemctl list-dependencies --all`"
Menampilkan daftar berkas unit yang dipasang pada sistem	"`systemctl list-unit-files`"
Memfungsikan "`$unit`" (menambahkan symlink)	"`systemctl enable $unit`"
Menonaktifkan "`$unit`" (menghapus symlink)	"`systemctl disable $unit`"
Membuka mask "`$unit`" (menghapus symlink ke "`/dev/null`")	"`systemctl unmask $unit`"
Me-mask "`$unit`" (menambahkan symlink ke "`/dev/null`")	"`systemctl mask $unit`"
Mendapatkan pengaturan target baku	"`systemctl get-default`"
Mengatur target baku ke "`grafis`" (sistem GUI)	"`systemctl set-default graphical`"
Mengatur target baku ke "`multi-user`" (sistem CLI)	"`systemctl set-default multi-user`"
Tampilkan lingkungan kerja	"`systemctl show-environment`"
Menata "`variabel`" lingkungan pekerjaan ke "`nilai`"	"`systemctl set-environment variable=value`"
Menghapus tatanan "`variabel`" lingkungan pekerjaan	"`systemctl unset-environment variable`"
Memuat ulang semua berkas unit dan daemon	"`systemctl daemon-reload`"
Mematikan sistem	"`systemctl poweroff`"
Mematikan dan reboot sistem	"`systemctl reboot`"
Suspensikan sistem	"`systemctl suspend`"
Hibernasikan sistem	"`systemctl hibernate`"

Tabel 3.6: Daftar cuplikan perintah `systemctl` umum

Di sini, "`$unit`" dalam contoh di atas mungkin nama unit tunggal (akhiran seperti `.service` dan `.target` adalah opsional) atau, dalam banyak kasus, beberapa spesifikasi unit (glob gaya shell "`*`", "?", "`[]`" menggunakan fnmatch(3) yang akan dicocokkan dengan nama-nama utama dari semua unit yang saat ini dalam memori).

Perintah yang mengubah keadaan sistem dalam contoh-contoh di atas biasanya didahului oleh "`sudo`" untuk mendapatkan hak administratif yang diperlukan.

Keluaran dari "`systemctl status $unit|$PID|$device`" menggunakan warna titik ("●") untuk meringkas keadaan unit secara sekilas.

- "●" putih mengindikasikan keadaan "tidak aktif" atau "menonaktifkan".
- "●" merah mengindikasikan keadaan "gagal" atau "galat".
- "●" hijau mengindikasikan keadaan "aktif", "memuat ulang", atau "mengaktifkan".

3.7 Pemantau sistem lainnya

Berikut adalah daftar cuplikan perintah pemantauan lain di bawah `systemd`. Silakan baca halaman man yang bersangkutan termasuk cgroups(7).

Operasi	Cuplikan perintah
Menampilkan waktu yang dihabiskan untuk setiap langkah inisialisasi	"`systemd-analyze time`"
Daftar semua unit pada saat menginisialisasi	"`systemd-analyze blame`"
Memuat dan mendeteksi kesalahan dalam berkas "`$unit`"	"`systemd-analyze verify $unit`"
Tampilkan informasi status runtime singkat dari pengguna sesi pemanggil	"`loginctl user-status`"
Tampilkan informasi status runtime singkat dari sesi pemanggil	"`loginctl session-status`"
Melacak proses boot oleh cgroups	"`systemd-cgls`"
Melacak proses boot oleh cgroups	"`ps xawf -eo pid,user,cgroup,args`"
Melacak proses boot oleh cgroups	Membaca sysfs di bawah "`/sys/fs/cgroup/`"

Tabel 3.7: Daftar cuplikan perintah pemantauan lainnya di bawah `systemd`

3.8 Konfigurasi sistem

3.8.1 Nama host

Kernel mempertahankan **nama host** sistem. Unit sistem yang dimulai oleh `systemd-hostnamed.service` menetapkan nama host sistem pada waktu boot ke nama yang disimpan dalam "`/etc/hostname`". Berkas ini harus berisi **hanya** nama host sistem, bukan suatu FQDN (fully qualified domain name).

Untuk mencetak nama host saat ini jalankan hostname(1) tanpa argumen.

3.8.2 Sistem berkas

Opsi kait disk normal dan sistem berkas jaringan diatur dalam "`/etc/fstab`". Lihat fstab(5) dan Bagian 9.6.7.

Konfigurasi sistem berkas terenkripsi diatur dalam "`/etc/crypttab`". Lihat crypttab(5)

Konfigurasi perangkat lunak RAID dengan mdadm(8) diatur dalam "`/etc/mdadm/mdadm.conf`". Lihat mdadm.conf(5).



Awas
Setelah mengait semua sistem berkas, berkas-berkas sementara di "`/tmp`", "`/var/lock`", dan "`/var/run`" dibersihkan setiap kali boot.

3.8.3 Inisialisasi antarmuka jaringan

Antarmuka jaringan biasanya diinisialisasi dalam "`networking.service`" untuk antarmuka `lo` dan "`NetworkManager.se` untuk antarmuka lain pada sistem desktop Debian modern di bawah `systemd`.

Lihat Bab 5 untuk cara mengonfigurasi mereka.

3.8.4 Inisialisasi sistem cloud

Instansi sistem cloud dapat diluncurkan sebagai klon dari "Image Cloud Resmi Debian" atau image serupa. Untuk instansi sistem seperti itu, kepribadian seperti nama host, sistem berkas, jaringan, lokal, kunci SSH, pengguna, dan grup dapat dikonfigurasikan menggunakan fungsionalitas yang disediakan oleh paket `cloud-init` dan `netplan.io` dengan berbagai sumber data seperti berkas yang diletakkan di dalam image sistem asli dan data eksternal yang disediakan selama peluncuran. Paket-paket ini memungkinkan konfigurasi sistem deklaratif menggunakan data YAML.

Lihat lebih lanjut di "Cloud Computing dengan Debian dan turunannya", "Dokumentasi Cloud-init", dan Bagian 5.4.

3.8.5 Contoh penyesuaian untuk mengubah layanan sshd

Dengan instalasi baku, banyak layanan jaringan (lihat Bab 6) dimulai sebagai proses daemon setelah `network.target` pada waktu boot oleh `systemd`. "`sshd`" tidak terkecuali. Mari kita ubah ini menjadi memulai "`sshd`" saat dibutuhkan sebagai contoh kustomisasi.

Pertama, nonaktifkan unit layanan yang dipasang sistem.

```
$ sudo systemctl stop sshd.service
$ sudo systemctl mask sshd.service
```

Sistem aktivasi soket on-demand dari layanan Unix klasik adalah melalui superserver `inetd` (atau `xinetd`). Di bawah `systemd`, yang setara dapat diaktifkan dengan menambahkan berkas konfigurasi unit ***.socket** dan ***.service**.

`sshd.socket` untuk menentukan soket tempat mendengarkan

```
[Unit]
Description=SSH Socket for Per-Connection Servers

[Socket]
ListenStream=22
Accept=yes

[Install]
WantedBy=sockets.target
```

`sshd@.service` sebagai berkas layanan pencocokan `sshd.socket`

```
[Unit]
Description=SSH Per-Connection Server

[Service]
ExecStart=-/usr/sbin/sshd -i
StandardInput=socket
```

Lalu muat ulang.

```
$ sudo systemctl daemon-reload
```

3.9 Sistem udev

Sistem udev menyediakan mekanisme untuk penemuan dan inisialisasi perangkat keras otomatis (lihat udev(7)) sejak kernel Linux 2.6. Setelah penemuan setiap perangkat oleh kernel, sistem udev memulai proses pengguna yang menggunakan informasi dari sistem berkas sysfs (lihat Bagian 1.2.12), memuat modul kernel yang diperlukan untuk mendukungnya menggunakan program modprobe(8) (lihat Bagian 3.10), dan menciptakan simpul perangkat yang sesuai.

Tip
Jika "`/lib/modules/versi-kernel/modules.dep`" tidak dihasilkan dengan benar oleh depmod(8) karena suatu alasan, modul mungkin tidak dimuat seperti yang diharapkan oleh sistem udev. Jalankan "`depmod -a`" untuk memperbaikinya.
Untuk aturan pengaitan di "`/etc/fstab`", simpul perangkat tidak perlu statis. Anda dapat menggunakan UUID untuk mengait perangkat menggantikan nama perangkat seperti "`/dev/sda`". Lihat Bagian 9.6.3.

Karena sistem udev agak merupakan target bergerak, saya meninggalkan rincian ke dokumentasi lain dan menjelaskan informasi minimum di sini.

Awas
Jangan mencoba menjalankan program yang berjalan lama seperti skrip backup dengan `RUN` dalam aturan udev seperti yang disebutkan dalam udev(7). Buatlah berkas systemd.service(5) yang tepat dan aktifkan berkas tersebut sebagai pengganti. Lihat Bagian 10.2.3.2.

3.10 Inisialisasi modul kernel

Program modprobe(8) memungkinkan kita untuk mengonfigurasi kernel Linux yang berjalan dari proses pengguna dengan menambahkan dan menghapus modul kernel. Sistem udev (lihat Bagian 3.9) mengotomatiskan pemanggilannya untuk membantu inisialisasi modul kernel.

Ada modul non-perangkat keras dan modul driver perangkat keras khusus sebagai berikut yang perlu dimuat sebelumnya dengan mencantumkannya dalam berkas "`/etc/modules`" (lihat modules(5)).

- Modul TUN/TAP yang menyediakan perangkat jaringan Point-to-Point (TUN) virtual dan perangkat jaringan Ethernet virtual (TAP),
- Modul netfilter yang menyediakan kemampuan firewall netfilter (iptables(8), Bagian 5.7), dan
- modul driver timer watchdog.

Berkas konfigurasi untuk program modprobe(8) terletak di bawah direktori "`/etc/modprobes.d/`" seperti yang dijelaskan dalam modprobe.conf(5). (Jika Anda ingin menghindari beberapa modul kernel dimuat secara otomatis, pertimbangkan untuk memasukkannya ke dalam berkas "`/etc/modprobes.d/blacklist`".)

Berkas "`/lib/modules/versi/modules.dep`" yang dihasilkan oleh program depmod(8) menjelaskan dependensi modul yang digunakan oleh program modprobe(8).

Catatan
Jika Anda mengalami masalah pemuatan modul dengan pemuatan modul waktu boot atau dengan modprobe(8), ”`depmod -a`” dapat menyelesaikan masalah ini dengan merekonstruksi ”`modules.dep`”.

Program modinfo(8) menunjukkan informasi tentang suatu modul kernel Linux.

Program lsmod(8) dengan baik memformat isi ”`/proc/modules`”, menunjukkan modul kernel apa yang saat ini dimuat.

Tip
Anda dapat mengidentifikasi perangkat keras yang tepat pada sistem Anda. Lihat Bagian 9.5.3.
Anda dapat mengonfigurasi perangkat keras pada waktu boot untuk mengaktifkan fitur perangkat keras yang diharapkan. Lihat Bagian 9.5.4.
Anda mungkin dapat menambahkan dukungan untuk perangkat khusus Anda dengan mengkompilasi ulang kernel. Lihat Bagian 9.10.

Bab 4

Kontrol akses dan autentikasi

Ketika seseorang (atau program) meminta akses ke sistem, otentikasi menegaskan identitas menjadi yang tepercaya.



Awas
Kesalahan konfigurasi PAM dapat mengunci Anda keluar dari sistem Anda sendiri. Anda harus memiliki CD penyelamatan yang berguna atau mengatur partisi boot alternatif. Untuk memulihkan, boot sistem dengan mereka dan memperbaiki hal-hal dari sana.

4.1 Autentikasi Unix normal

Otentikasi Unix normal disediakan oleh modul pam_unix(8) di bawah PAM (Pluggable Authentication Modules). 3 berkas konfigurasi penting, dengan ”`:`” yang memisah entri, adalah sebagai berikut.

berkas	izin	pengguna	kelompok	deskripsi
`/etc/passwd`	`-rw-r--r--`	`root`	`root`	(tersanitasi) informasi akun pengguna
`/etc/shadow`	`-rw-r-----`	`root`	`shadow`	informasi akun pengguna yang aman
`/etc/group`	`-rw-r--r--`	`root`	`root`	informasi grup

Tabel 4.1: 3 berkas konfigurasi penting untuk pam_unix(8)

”`/etc/passwd`” berisi yang berikut ini.

```
 ...
user1:x:1000:1000:User1 Name,,,:/home/user1:/bin/bash
user2:x:1001:1001:User2 Name,,,:/home/user2:/bin/bash
 ...
```

Seperti yang dijelaskan dalam passwd(5), masing-masing entri yang dipisah ”`:`” dari berkas ini berarti sebagai berikut.

- Nama log masuk
- Entri spesifikasi kata sandi
- ID pengguna numerik
- ID grup numerik
- Nama pengguna atau kolom komentar

- Direktori home milik pengguna
- Interpreter perintah pengguna opsional

Entri kedua ”/etc/passwd” digunakan untuk entri kata sandi terenkripsi. Setelah pengenalan ”/etc/shadow”, entri ini digunakan untuk entri spesifikasi kata sandi.

isi	arti
(kosong)	akun tanpa kata sandi
x	kata sandi terenkripsi ada di ”/etc/shadow”

Tabel 4.2: Konten entri kedua dari ”/etc/passwd”

”/etc/shadow” berisi yang berikut.

```
 ...
user1:$1$Xop0FYH9$IfxyQwBe9b8tiyIkt2P4F/:13262:0:99999:7:::
user2:$1$vXGZLVbS$ElyErNf/agUDsm1DehJMS/:13261:0:99999:7:::
 ...
```

Seperti yang dijelaskan dalam shadow(5), masing-masing entri yang dipisah ”:” dari berkas ini berarti sebagai berikut.

- Nama log masuk
- Kata sandi terenkripsi (”1” di awal menunjukkan penggunaan enkripsi MD5. ”*” menunjukkan tidak diizinkan login.)
- Tanggal perubahan kata sandi terakhir, dinyatakan sebagai cacah hari sejak 1 Januari 1970
- Berapa hari pengguna harus menunggu sebelum dia diizinkan untuk mengubah kata sandinya lagi
- Cacah hari yang setelah itu pengguna harus mengubah kata sandinya
- Cacah hari sebelum kata sandi akan kedaluwarsa di mana pengguna harus diperingatkan
- Cacah hari setelah kata sandi kedaluwarsa di mana kata sandi masih harus diterima
- Tanggal kedaluwarsa akun, dinyatakan sebagai cacah hari sejak 1 Januari 1970
- …

”/etc/group” berisi yang berikut ini.

```
group1:x:20:user1,user2
```

Seperti yang dijelaskan dalam group(5), masing-masing entri yang dipisah ”:” dari berkas ini berarti sebagai berikut.

- Nama grup
- Kata sandi terenkripsi (tidak benar-benar digunakan)
- ID grup numerik
- daftar nama pengguna yang dipisah ”,”

Catatan
”/etc/gshadow” menyediakan fungsi yang serupa dengan ”/etc/shadow” untuk ”/etc/group” namun tidak benar-benar digunakan.

Catatan
Keanggotaan grup pengguna yang sebenarnya dapat ditambahkan secara dinamis jika baris "`auth optional pam_group.so`" ditambahkan ke "`/etc/pam.d/common-auth`" dan menatanya dalam "`/etc/security/group.conf`". Lihat pam_group(8).

Catatan
Paket `base-passwd` berisi daftar otoritatif pengguna dan grup: "`/usr/share/doc/base-passwd/users-and-group`

4.2 Mengelola informasi akun dan kata sandi

Berikut adalah beberapa perintah penting untuk mengelola informasi akun.

perintah	fungsi
`getent passwd` *`user_name`*	menelusuri informasi akun "*`user_name`*"
`getent shadow` *`user_name`*	menelusuri informasi akun ter-shadow "*`nama_pengguna`*"
`getent group` *`group_name`*	menelusuri informasi grup "*`nama_grup_`*"
`passwd`	mengelola kata sandi untuk akun
`passwd -e`	mengatur kata sandi satu kali untuk aktivasi akun
`chage`	mengelola informasi pembatasan umur kata sandi

Tabel 4.3: Daftar perintah untuk mengelola informasi akun

Anda mungkin perlu memiliki hak istimewa root agar beberapa fungsi dapat bekerja. Lihat crypt(3) untuk enkripsi data dan kata sandi.

Catatan
Pada sistem yang disiapkan dengan PAM dan NSS sebagai mesin salsa Debian, konten lokal "`/etc/passwd`", "`/etc/group`", dan "`/etc/shadow`" tidak dapat digunakan secara aktif oleh sistem. Perintah di atas berlaku bahkan di bawah lingkungan tersebut.

4.3 Kata sandi yang baik

Saat membuat akun selama instalasi sistem Anda atau dengan perintah passwd(1), Anda harus memilih kata sandi yang baik yang terdiri dari setidaknya 6 hingga 8 karakter termasuk satu atau lebih karakter dari masing-masing set berikut sesuai dengan passwd(1).

- Alfabet huruf kecil
- Digit 0 sampai 9
- Tanda baca



Awas
Jangan memilih kata-kata yang bisa ditebak untuk kata sandi. Nama akun, nomor jaminan sosial, nomor telepon, alamat, ulang tahun, nama anggota keluarga atau hewan peliharaan Anda, kata-kata kamus, urutan karakter sederhana seperti "12345" atau "qwerty", ... Ini adalah pilihan yang buruk untuk kata sandi.

4.4 Membuat kata sandi terenkripsi

Ada alat independen untuk menghasilkan kata sandi terenkripsi dengan salt.

paket	popcon	ukuran	perintah	fungsi
whois	V:23, I:203	384	mkpasswd	front end dengan fitur berlebih untuk pustaka crypt(3)
openssl	V:846, I:996	2564	openssl passwd	menghitung hash kata sandi (OpenSSL). passwd(1ssl)

Tabel 4.4: Daftar alat untuk menghasilkan kata sandi

4.5 PAM dan NSS

Sistem mirip Unix modern seperti sistem Debian menyediakan mekanisme PAM (Pluggable Authentication Modules) dan NSS (Name Service Switch) ke administrator sistem lokal untuk mengonfigurasi sistemnya. Peran ini dapat diringkas sebagai berikut.

- PAM menawarkan mekanisme otentikasi fleksibel yang digunakan oleh perangkat lunak aplikasi sehingga melibatkan pertukaran data kata sandi.
- NSS menawarkan mekanisme layanan nama fleksibel yang sering digunakan oleh pustaka standar C untuk mendapatkan nama pengguna dan grup untuk program seperti ls(1) dan id(1).

Sistem PAM dan NSS ini perlu dikonfigurasi secara konsisten.

Paket penting dari sistem PAM dan NSS adalah sebagai berikut.

paket	popcon	ukuran	deskripsi
libpam-modules	V:942, I:1000	892	Pluggable Authentication Modules (layanan dasar)
libpam-ldapd	V:7, I:13	79	Pluggable Authentication Modules yang memungkinkan antarmuka LDAP
libpam-systemd	V:764, I:969	848	Pluggable Authentication Modules untuk mendaftarkan sesi pengguna bagi logind
libpam-doc	I:7.5	1492	Pluggable Authentication Modules (dokumentasi dalam html dan teks)
libc6	V:936, I:1000	5483	Pustaka C GNU: Pustaka bersama yang juga menyediakan layanan ”Name Service Switch”
glibc-doc	I:5.1	3927	Pustaka C GNU: Halaman man
glibc-doc-reference	I:3.1	14764	Pustaka C GNU: Manual referensi dalam format info, pdf, dan html (non-free)
libnss-mdns	V:299, I:507	142	Modul NSS untuk resolusi nama DNS Multicast
libnss-ldapd	V:8, I:16	131	Modul NSS untuk menggunakan LDAP sebagai layanan penamaan

Tabel 4.5: Daftar sistem PAM dan NSS yang terkenal

- ”The Linux-PAM System Administrators’ Guide” (Panduan Administrator Sistem Linux-PAM) dalam libpam-doc sangat penting untuk mempelajari konfigurasi PAM.

- Bagian ”System Databases and Name Service Switch” (Basis Data Sistem dan Sakelar Layanan Nama) di `glibc-doc-re` sangat penting untuk mempelajari konfigurasi NSS.

Catatan
Anda dapat melihat daftar yang lebih luas dan terkini dengan perintah ”`aptitude search 'libpam-|libnss-'`”. Akronim NSS juga bisa berarti ”Network Security Service” yang berbeda dari ”Name Service Switch”.

Catatan
PAM adalah cara paling dasar untuk menginisialisasi variabel lingkungan bagi setiap program dengan nilai baku seluruh sistem.

Di bawah systemd, paket `libpam-systemd` dipasang untuk mengelola login pengguna dengan mendaftarkan sesi pengguna dalam hirarki grup kontrol `systemd` untuk logind. Lihat systemd-logind(8), logind.conf(5), dan pam_systemd(8).

4.5.1 Berkas konfigurasi yang diakses oleh PAM dan NSS

Berikut adalah beberapa berkas konfigurasi penting yang diakses oleh PAM dan NSS.

berkas konfigurasi	fungsi
`/etc/pam.d/`*`program_name`*	menyiapkan konfigurasi PAM untuk program ”*`nama_program`*”; lihat pam(7) dan pam.d(5)
`/etc/nsswitch.conf`	menyiapkan konfigurasi NSS dengan entri untuk setiap layanan. Lihat nsswitch.conf(5)
`/etc/nologin`	membatasi login pengguna dengan modul pam_nologin(8)
`/etc/securetty`	membatasi tty untuk akses root oleh modul pam_securetty(8)
`/etc/security/access.conf`	menetapkan batas akses dengan modul pam_access(8)
`/etc/security/group.conf`	mengatur pengekangan berbasis grup oleh modul pam_group(8)
`/etc/security/pam_env.conf`	mengatur variabel lingkungan dengan modul pam_env(8)
`/etc/environment`	mengatur variabel lingkungan tambahan dengan modul pam_env(8) dengan argumen ”`readenv=1`”
`/etc/default/locale`	mengatur lokal dengan modul pam_env(8) memakai argumen ”`readenv=1 envfile=/etc/default/locale`” (Debian)
`/etc/security/limits.conf`	mengatur pengekangan sumber daya (ulimit, core, …) oleh modul pam_linits(8)
`/etc/security/time.conf`	mengatur pengekangan waktu oleh modul pam_time(8)
`/etc/systemd/logind.conf`	mengatur konfigurasi manajer login `systemd` (lihat logind.conf(5) dan systemd-logind.service(8))

Tabel 4.6: Daftar berkas konfigurasi yang diakses oleh PAM dan NSS

Pembatasan pemilihan kata sandi dilaksanakan oleh modul PAM, pam_unix(8) dan pam_cracklib(8). Mereka dapat dikonfigurasi oleh argumen mereka.

Tip
Modul PAM menggunakan akhiran ”`.so`” untuk nama berkas mereka.

4.5.2 Manajemen sistem terpusat modern

Manajemen sistem terpusat modern dapat digelar memakai server Lightweight Directory Access Protocol (LDAP) terpusat untuk mengelola banyak sistem mirip Unix dan non-Unix di jaringan. Implementasi open source dari Lightweight Directory Access Protocol adalah Perangkat Lunak OpenLDAP.

Server LDAP menyediakan informasi akun melalui penggunaan PAM dan NSS dengan paket `libpam-ldapd` dan `libnss-ldapd` untuk sistem Debian. Beberapa tindakan diperlukan untuk mengaktifkan ini (Saya belum menggunakan penyiapan ini dan berikut ini adalah informasi sekunder murni. Silakan baca ini dalam konteks ini).

- Anda menyiapkan server LDAP terpusat dengan menjalankan program seperti daemon LDAP yang berdiri sendiri, slapd(8).
- Anda mengubah berkas konfigurasi PAM di direktori "`/etc/pam.d/`" untuk menggunakan "`pam_ldap.so`" alih-alih baku "`pam_unix.so`".
- Anda mengubah konfigurasi NSS dalam berkas "`/etc/nsswitch.conf`" untuk menggunakan "`ldap`" alih-alih baku ("`compat`" atau "`file`").
- Anda harus membuat `libpam-ldapd` untuk menggunakan koneksi SSL (atau TLS) bagi keamanan kata sandi.
- Anda dapat membuat `libnss-ldapd` menggunakan koneksi SSL (atau TLS) untuk memastikan integritas data dengan beban overhead jaringan LDAP.
- Anda harus menjalankan nscd(8) secara lokal untuk menyinggahkan hasil pencarian LDAP agar mengurangi lalu lintas jaringan LDAP.

Lihat dokumentasi di nsswitch.conf(5), pam.conf(5), ldap.conf(5), dan "`/usr/share/doc/libpam-doc/html/`" dari paket `libpam-doc` dan "`info libc 'Name Service Switch'`" dari paket `glibc-doc` package.

Demikian pula, Anda dapat mengatur sistem terpusat alternatif dengan metode lain.

- Integrasi pengguna dan grup dengan sistem Windows.
  - Mengakses layanan domain Windows dengan paket `winbind` dan `libpam_winbind`.
  - Lihat winbindd(8) dan Mengintegrasikan Jaringan MS Windows dengan Samba.
- Integrasi pengguna dan grup dengan sistem warisan mirip Unix.
  - Mengakses NIS (awalnya disebut YP) atau NIS+ dengan paket `nis`.
  - Lihat Linux NIS(YP)/NYS/NIS+ HOWTO.

4.5.3 "Mengapa GNU su tidak mendukung kelompok wheel"

Ini adalah frasa terkenal di bagian bawah halaman "`info su`" lama oleh Richard M. Stallman. Tidak perlu khawatir: perintah `su` saat ini di Debian menggunakan PAM, sehingga seseorang dapat membatasi kemampuan untuk menggunakan `su` ke grup `root` dengan mengaktifkan baris dengan "`pam_wheel.so`" dalam "`/etc/pam.d/su`".

4.5.4 Aturan kata sandi yang lebih ketat

Memasang paket `libpam-cracklib` memungkinkan Anda untuk memaksa aturan kata sandi yang lebih ketat.

Pada sistem GNOME umum yang secara otomatis memasang `libpam-gnome-keyring`, "`/etc/pam.d/common-passwo`
terlihat seperti:

```
# here are the per-package modules (the "Primary" block)
password  requisite     pam_cracklib.so retry=3 minlen=8 difok=3
password  [success=1 default=ignore]  pam_unix.so obscure use_authtok try_first_pass  ↩
    yescrypt
# here's the fallback if no module succeeds
password  requisite     pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
password  required      pam_permit.so
# and here are more per-package modules (the "Additional" block)
password  optional  pam_gnome_keyring.so
# end of pam-auth-update config
```

4.6 Keamanan autentikasi

Catatan
Informasi di sini **mungkin tidak cukup** untuk kebutuhan keamanan Anda tetapi itu mestinya menjadi **awal yang baik**.

4.6.1 Kata sandi aman di Internet

Banyak layanan lapisan transportasi populer mengkomunikasikan pesan termasuk otentikasi kata sandi dalam teks polos. Adalah ide yang sangat buruk untuk mengirimkan kata sandi dalam teks polos melalui Internet liar di mana ia dapat disadap. Anda dapat menjalankan layanan ini melalui ”Transport Layer Security” (TLS) atau pendahulunya, ”Secure Sockets Layer” (SSL) untuk mengamankan seluruh komunikasi termasuk kata sandi dengan enkripsi.

nama layanan tidak aman	port	nama layanan aman	port
`www (http)`	80	`https`	443
`smtp (mail)`	25	`ssmtp (smtps)`	465
`ftp-data`	20	`ftps-data`	989
`ftp`	21	`ftps`	990
`telnet`	23	`telnets`	992
`imap2`	143	`imaps`	993
`pop3`	110	`pop3s`	995
`ldap`	389	`ldaps`	636

Tabel 4.7: Daftar layanan dan port yang tidak aman dan aman

Enkripsi memakan waktu CPU. Sebagai alternatif yang ramah CPU, Anda dapat menjaga komunikasi dalam teks polos sambil mengamankan hanya kata sandi dengan protokol otentikasi aman seperti ”Authenticated Post Office Protocol” (APOP) untuk POP dan ”Challenge-Response Authentication Mechanism MD5” (CRAM-MD5) untuk SMTP dan IMAP. (Untuk mengirim pesan surel melalui Internet ke server surel Anda dari klien surel Anda, baru-baru ini populer untuk menggunakan port message submission baru 587 alih-alih port SMTP tradisional 25 untuk menghindari pemblokiran port 25 oleh penyedia jaringan sambil mengautentikasi diri Anda dengan CRAM-MD5.)

autentikasi berbasis kata sandi tradisional masih memiliki masalah-masalah keamanan inheren seperti misalnya penggunaan ulang antar berbagai situs web. Lihat metode autentikasi lain untuk solusi aman: Kata sandi sekali pakai berbasis waktu (Time-based one-time password, TOTP), Universal 2nd Factor (U2F), Web Authentication (WebAuthn, biasanya disebut ”passkey”)

4.6.2 Secure Shell

Program Secure Shell (SSH) menyediakan komunikasi terenkripsi yang aman antara dua host yang tidak tepercaya melalui jaringan yang tidak aman dengan otentikasi yang aman. Ini terdiri dari klien OpenSSH, ssh(1), dan daemon OpenSSH, sshd(8). SSH ini dapat digunakan untuk membungkus komunikasi protokol yang tidak aman seperti POP dan X dengan aman melalui Internet dengan fitur penerusan port.

Klien mencoba mengautentikasi dirinya sendiri menggunakan autentikasi berbasis host, otentikasi kunci publik, otentikasi challenge-response, atau autentikasi kata sandi. Penggunaan otentikasi kunci publik memungkinkan login tanpa kata sandi jarak jauh. Lihat Bagian 6.3.

4.6.3 Langkah-langkah keamanan tambahan untuk Internet

Bahkan ketika Anda menjalankan layanan aman seperti server Secure Shell (SSH) dan Point-to-point tunneling protocol (PPTP), masih ada kemungkinan untuk pembobolan menggunakan serangan brute force menebak kata sandi dari Internet. Penggunaan kebijakan firewall (lihat Bagian 5.7) bersama dengan alat-alat keamanan berikut dapat meningkatkan situasi keamanan.

paket	popcon	ukuran	deskripsi
`knockd`	V:0.7, I:1.6	110	daemon port-knock kecil knockd(1) dan klien knock(1)
`fail2ban`	V:101, I:111	2275	memblokir IP-IP yang menyebabkan beberapa kesalahan otentikasi
`libpam-shield`	V:0.04, I:0.05	115	mengunci penyerang jarak jauh yang mencoba menebak kata sandi

Tabel 4.8: Daftar alat untuk memberikan langkah-langkah keamanan tambahan

4.6.4 Mengamankan kata sandi root

Untuk mencegah orang mengakses mesin Anda dengan hak istimewa root, Anda perlu membuat tindakan berikut.

- Cegah akses fisik ke perangkat penyimpanan sistem (HDD / SSD / ...)
- Mengunci UEFI/BIOS dan mencegah boot dari media lepasan
- Mengatur kata sandi untuk sesi interaktif GRUB
- Mengunci menu GRUB dari penyuntingan

Dengan akses fisik ke perangkat penyimpanan sistem, pengaturan ulang kata sandi relatif mudah dilakukan dengan langkah-langkah berikut.

1. Pindahkan perangkat penyimpanan sistem ke PC dengan UEFI/BIOS yang dapat di-boot melalui USB.
2. Boot sistem dengan media penyelamat (lihat Bagian 3.2.2).
3. Kait partisi root dengan akses baca/tulis.
4. Sunting "`/etc/passwd`" di partisi root dan jadikan entri kedua untuk akun `root` kosong.

Jika Anda memiliki akses sunting ke entri menu GRUB (lihat Bagian 3.1.2) untuk `grub-rescue-pc` pada waktu boot, bahkan lebih mudah dengan langkah-langkah berikut.

1. Boot sistem dengan parameter kernel berubah menjadi sesuatu seperti "`root=/dev/sda6 rw init=/bin/sh`".

2. Sunting "`/etc/passwd`" dan jadikan entri kedua untuk akun `root` kosong.
3. Boot ulang sistem.

Shell root sistem sekarang dapat diakses tanpa kata sandi.

Catatan
Setelah seseorang memiliki akses root shell, ia dapat mengakses semuanya di sistem dan mengatur ulang kata sandi apa pun pada sistem. Lebih lanjut, ia dapat mengkompromikan kata sandi untuk semua akun pengguna menggunakan alat cracking kata sandi brute force seperti paket `john` dan `crack` (lihat Bagian 9.5.10). Kata sandi yang tertebak ini dapat menyebabkan kompromi sistem lain.

Satu-satunya solusi perangkat lunak yang masuk akal untuk menghindari semua masalah ini adalah dengan menggunakan partisi root terenkripsi perangkat lunak (atau partisi "`/etc`") menggunakan dm-crypt dan initramfs (lihat Bagian 9.9). Namun Anda selalu membutuhkan kata sandi untuk mem-boot sistem.

4.7 Kontrol akses lainnya

Ada kontrol akses ke sistem selain otentikasi berbasis kata sandi dan izin berkas.

Catatan
Lihat Bagian 9.4.16 untuk membatasi fitur secure attention key (SAK) kernel.

4.7.1 Access control lists (ACL)

ACL adalah superset dari izin reguler seperti yang dijelaskan di Bagian 1.2.3.

Anda menemui ACL beraksi di lingkungan desktop modern. Ketika perangkat penyimpanan USB yang terformat dipasang secara otomatis sebagai, mis., "`/media/penguin/USBSTICK`", pengguna normal `penguin` dapat menjalankan:

```
 $ cd /media/penguin
 $ ls -la
total 16
drwxr-x---+ 1 root    root    16 Jan 17 22:55 .
drwxr-xr-x  1 root    root    28 Sep 17 19:03 ..
drwxr-xr-x  1 penguin penguin 18 Jan  6 07:05 USBSTICK
```

"+" pada kolom ke-11 menunjukkan ACL sedang bekerja. Tanpa ACL, pengguna normal `penguin` tidak akan dapat membuat daftar seperti ini karena `penguin` tidak berada di grup `root`. Anda dapat melihat ACL sebagai:

```
 $ getfacl .
# file: .
# owner: root
# group: root
user::rwx
user:penguin:r-x
group::---
mask::r-x
other::---
```

Sini:

- ”`user::rwx`”, ”`group::---`”, dan ”`other::---`” berkorespondensi ke izin pemilik reguler, grup, dan yang lain.
- ACL ”`user:penguin:r-x`” mengizinkan seorang pengguna normal `penguin` untuk memiliki izin ”`r-x`”. Ini memungkinkan ”`ls -la`” mencantumkan daftar konten direktori.
- ACL ”`mask::r-x`” menata batas atas izin.

Lihat ”POSIX Access Control Lists on Linux”, acl(5), getfacl(1), dan setfacl(1) untuk lebih banyak lagi.

4.7.2 sudo

sudo(8) adalah program yang dirancang untuk memungkinkan sysadmin untuk memberikan hak istimewa root terbatas kepada pengguna dan mencatat log aktivitas root. `sudo` hanya membutuhkan kata sandi pengguna biasa. Pasang paket `sudo` dan aktifkan dengan mengatur opsi di ”`/etc/sudoers`”. Lihat contoh konfigurasi di ”`/usr/share/doc/sudo/`” dan Bagian 1.1.12.

Penggunaan `sudo` saya untuk sistem pengguna tunggal (lihat Bagian 1.1.12) bertujuan untuk melindungi diri dari kebodohan saya sendiri. Secara pribadi, saya menganggap menggunakan `sudo` alternatif yang lebih baik daripada menggunakan sistem dari akun root sepanjang waktu. Misalnya, berikut ini mengubah pemilik ”*`suatu_berkas`*” menjadi ”*`namaku`*”.

```
$ sudo chown my_name some_file
```

Tentu saja jika Anda tahu kata sandi root (seperti yang dilakukan pengguna Debian yang dipasang sendiri), perintah apa pun dapat dijalankan di bawah root dari akun pengguna mana pun menggunakan ”`su -c`”.

4.7.3 PolicyKit

PolicyKit adalah komponen sistem operasi untuk mengendalikan hak istimewa di seluruh sistem dalam sistem operasi mirip Unix.

Aplikasi GUI yang lebih baru tidak dirancang untuk berjalan sebagai proses istimewa. Mereka berbicara dengan proses istimewa melalui PolicyKit untuk melakukan operasi administratif.

PolicyKit membatasi operasi tersebut ke akun pengguna milik grup `sudo` pada sistem Debian.

Lihat polkit(8).

4.7.4 Membatasi akses ke beberapa layanan server

Untuk keamanan sistem, adalah ide yang baik untuk menonaktifkan sebanyak mungkin program server. Ini menjadi penting untuk server jaringan. Memiliki server yang tidak digunakan, diaktifkan baik secara langsung sebagai daemon atau melalui program super-server, dianggap sebagai risiko keamanan.

Banyak program, seperti sshd(8), menggunakan kontrol akses berbasis PAM. Ada banyak cara untuk membatasi akses ke beberapa layanan server.

- berkas konfigurasi: ”`/etc/default/`*`nama_program`*”
- Konfigurasi unit layanan systemd untuk daemon
- PAM (Pluggable Authentication Modules)
- ”`/etc/inetd.conf`” untuk super-server
- ”`/etc/hosts.deny`” dan ”`/etc/hosts.allow`” untuk TCP wrapper, tcpd(8)
- ”`/etc/rpc.conf`” untuk Sun RPC

- "`/etc/at.allow`" dan "`/etc/at.deny`" untuk atd(8)
- "`/etc/cron.allow`" dan "`/etc/cron.deny`" untuk crontab(1)
- Firewall jaringan dari infrastruktur netfilter

Lihat Bagian 3.6, Bagian 4.5.1, dan Bagian 5.7.

Tip
Layanan Sun RPC harus aktif untuk NFS dan program berbasis RPC lainnya.

Tip
Jika Anda mengalami masalah akses jarak jauh dalam sistem Debian baru-baru ini, jadikan komentar saja baris konfigurasi yang menyinggung seperti "`ALL: PARANOID`" di "`/etc/hosts.deny`" jika ada. (Tetapi Anda harus berhati-hati terhadap risiko keamanan yang terlibat dengan tindakan semacam ini.)

4.7.5 Fitur keamanan Linux

Kernel Linux telah berevolusi dan mendukung fitur keamanan yang tidak ditemukan dalam implementasi UNIX tradisional.

Linux mendukung atribut extended yang memperluas atribut UNIX tradisional (lihat xattr(7)).

Linux membagi hak istimewa yang secara tradisional terkait dengan superuser menjadi unit-unit yang berbeda, yang dikenal sebagai capabilities(7), yang dapat diaktifkan dan dinonaktifkan secara independen. Kapabilitas adalah atribut per-thread sejak kernel versi 2.2.

Kerangka Kerja Linux Security Module (LSM) menyediakan mekanisme untuk berbagai pemeriksaan keamanan yang akan di-hook oleh ekstensi kernel baru. Misalnya:

- AppArmor
- Security-Enhanced Linux (SELinux)
- Smack (Simplified Mandatory Access Control Kernel)
- Tomoyo Linux

Karena ekstensi ini dapat memperketat model hak istimewa lebih ketat daripada kebijakan model keamanan mirip-Unix biasa, bahkan kekuatan root dapat dibatasi. Anda disarankan untuk membaca dokumen kerangka kerja Linux Security Module (LSM) di kernel.org.

namespaces Linux membungkus sumber daya sistem global dalam abstraksi yang membuatnya tampak pada proses dalam namespace bahwa mereka memiliki instansi terisolasi mereka sendiri dari sumber daya global. Perubahan pada sumber daya global terlihat oleh proses lain yang merupakan anggota ruang nama, tetapi tidak terlihat oleh proses lain. Sejak kernel versi 5.6, ada 8 jenis namespace (lihat namespaces(7), unshare(1), nsenter(1)).

Pada Debian 11 Bullseye (2021), Debian menggunakan hirarki cgroup terpadu (alias cgroups-v2).

Contoh penggunaan namespaces dengan cgroup untuk mengisolasi proses mereka dan untuk memungkinkan kontrol sumber daya adalah:

- Systemd. Lihat Bagian 3.3.1.
- Lingkungan sandbox. Lihat Bagian 7.7.
- Kontainer Linux seperti Docker, LXC. Lihat Bagian 9.11.

Fungsi-fungsi ini tidak dapat direalisasikan oleh Bagian 4.1. Topik-topik tingkat lanjut ini sebagian besar di luar cakupan untuk dokumen pengantar ini.

Bab 5

Penyiapan jaringan

Tip
Untuk panduan khusus Debian modern tentang jaringan, baca Buku Pegangan Administrator Debian — Mengonfigurasi Jaringan.

Tip
Di bawah systemd, networkd dapat digunakan untuk mengelola jaringan. Lihat systemd-networkd(8).

5.1 Infrastruktur jaringan dasar

Mari kita tinjau infrastruktur jaringan dasar pada sistem Debian modern.

5.1.1 Resolusi nama host

Resolusi nama host saat ini juga didukung oleh mekanisme NSS (Name Service Switch). Aliran resolusi ini adalah sebagai berikut.

1. Berkas "`/etc/nsswitch.conf`" dengan stanza seperti "`host: files dns`" menentukan urutan resolusi nama host. (Ini menggantikan fungsi lama dari stanza "`order`" dalam "`/etc/host.conf`".)
2. Metode `files` dipanggil terlebih dahulu. Jika nama host ditemukan di berkas "`/etc/hosts`", ia mengembalikan semua alamat yang valid untuk itu dan keluar. (Berkas "`/etc/host.conf`" berisi "`multi on`".)
3. Metode `dns` dipanggil. Jika nama host ditemukan oleh kueri ke Internet Domain Name System (DNS) yang diidentifikasi oleh berkas "`/etc/resolv.conf`", ia mengembalikan semua alamat yang valid untuk itu dan keluar.

Workstation biasa mungkin dipasang dengan nama host yang disetel ke, mis., *host_name*" dan nama domain opsional disetel ke string kosong. Kemudian, "`/etc/hosts`" terlihat seperti berikut ini.

```
127.0.0.1 localhost
127.0.1.1 host_name

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

paket	popcon	ukuran	jenis	deskripsi
network-manager	V:427, I:474	7365	config::NM	NetworkManager (daemon): mengelola jaringan secara otomatis
network-manager-gnome	V:36, I:147	19	config::NM	NetworkManager (frontend GNOME)
netplan.io	V:2.1, I:8.4	340	config::NM+networkd	Netplan (generator): Antarmuka deklaratif tempatku untuk NetworkManager dan backend systemd-networkd
ifupdown	V:649, I:973	204	config::ifupdown	alat standar untuk menghidupkan dan mematikan jaringan (spesifik Debian)
pppoeconf	V:0.2, I:3.9	176	config::helper	pembantu konfigurasi untuk koneksi PPPoE
wpasupplicant	V:421, I:521	3896	config::helper	dukungan klien untuk WPA dan WPA2 (IEEE 802.11i)
wpagui	V:0.2, I:1.2	779	config::helper	Klien GUI Qt untuk wpa_supplicant
wireless-tools	V:193, I:262	232	config::helper	alat untuk memanipulasi Ekstensi Nirkabel Linux
iw	V:40, I:478	332	config::helper	alat untuk mengonfigurasi perangkat nirkabel Linux
iproute2	V:781, I:987	4268	config::iproute2	iproute2, IPv6 dan konfigurasi jaringan tingkat lanjut lainnya: ip(8), tc(8), dll
iptables	V:381, I:635	2408	config::Netfilter	alat administrasi untuk penyaringan paket dan NAT (Netfilter)
nftables	V:266, I:880	189	config::Netfilter	alat administrasi untuk penyaringan paket dan NAT (Netfilter) (penerus {ip,ip6,arp,eb}tables)
iputils-ping	V:206, I:998	188	ujian	menguji keterjangkauan jaringan host jarak jauh berdasarkan nama host atau alamat IP (iproute2)
iputils-arping	V:2, I:17	53	ujian	menguji keterjangkauan jaringan host jarak jauh yang ditentukan oleh alamat ARP
iputils-tracepath	V:2, I:19	50	ujian	melacak jalur jaringan ke host jarak jauh
ethtool	V:98, I:251	1093	ujian	menampilkan atau mengubah pengaturan perangkat Ethernet
mtr-tiny	V:4, I:38	182	test::low-level	melacak jalur jaringan ke host jarak jauh (curses)
mtr	V:4, I:40	231	test::low-level	melacak jalur jaringan ke host jarak jauh (curses dan GTK)
gnome-nettool	V:0.5, I:8.4	2480	test::low-level	alat untuk operasi informasi jaringan umum (GNOME)
nmap	V:25, I:182	4934	test::low-level	pemetaan jaringan / pemindai port (Nmap, konsol)
tcpdump	V:20, I:177	1346	test::low-level	penganalisis lalu lintas jaringan (Tcpdump, konsol)
wireshark	V:4, I:39	17068	test::low-level	penganalisis lalu lintas jaringan (Wireshark, GTK)
tshark	V:2, I:23	433	test::low-level	penganalisis lalu lintas jaringan (konsol)
tcptrace	V:0.2, I:1.5	407	test::low-level	menghasilkan ringkasan koneksi dari keluaran `tcpdump`
dnsutils	V:5, I:146	23	test::low-level	klien jaringan yang disediakan dengan BIND: nslookup(8), nsupdate(8), dig(8)
dlint	V:0.1, I:2.0	52	test::low-level	memeriksa informasi zona DNS menggunakan pencarian server nama
dnstracer	V:0.1, I:1.1	63	test::low-level	melacak rantai server DNS ke sumber

Tabel 5.1: Daftar alat konfigurasi jaringan

Setiap baris dimulai dengan alamat IP dan diikuti oleh nama host terkait.

Alamat IP `127.0.1.1` di baris kedua dari contoh ini mungkin tidak ditemukan pada beberapa sistem mirip-Unix lainnya. Debian Installer membuat entri ini untuk sistem tanpa alamat IP permanen sebagai solusi untuk beberapa perangkat lunak (misalnya, GNOME) seperti yang didokumentasikan dalam bug #719621.

nama_host cocok dengan nama host yang didefinisikan dalam "`/etc/hostname`" (lihat Bagian 3.8.1).

Untuk sistem dengan alamat IP permanen, alamat IP permanen itu harus digunakan di sini, bukan `127.0.1.1`.

Untuk sistem dengan alamat IP permanen dan fully qualified domain name (FQDN) yang disediakan oleh Domain Name System (DNS), *nama_host.nama_domain* kanonik itu harus digunakan, bukan hanya *nama_host*.

"`/etc/resolv.conf`" adalah berkas statis jika paket `resolvconf` tidak dipasang. Jika dipasang, itu adalah taut simbolis. Apapun kasusnya, itu berisi informasi yang menginisialisasi rutinitas resolver. Jika DNS ditemukan di IP="`192.168.11.1`", itu berisi yang berikut.

```
nameserver 192.168.11.1
```

Paket `resolvconf` membuat "`/etc/resolv.conf`" ini menjadi taut simbolis dan mengelola isinya dengan skrip hook secara otomatis.

Untuk workstation PC pada lingkungan LAN adhoc umum, nama host dapat diurai melalui DNS Multicast (mDNS) selain ke metode dasar `files` dan `dns`.

- Avahi menyediakan kerangka kerja untuk Multicast DNS Service Discovery di Debian.
- Ini setara dengan Apple Bonjour / Apple Rendezvous.
- Paket plugin `libnss-mdns` menyediakan resolusi nama host melalui mDNS untuk fungsi GNU Name Service Switch (NSS) dari GNU C Library (glibc).
- Berkas "`/etc/nsswitch.conf`" harus memiliki stanza seperti "`hosts: files mdns4_minimal [NOTFOUND=retur dns`" (lihat `/usr/share/doc/libnss-mdns/README.Debian` untuk konfigurasi lain).
- Nama host yang diakhiri dengan ".local" domain pseudo-top-level diuraikan dengan mengirim suatu pesan query mDNS dalam suatu paket UDP multicast memakai alamat IPv "`224.0.0.251`" atau alamat IPv6 "`FF02::FB`".

Catatan
Perluasan Domain Tingkat Atas generik (gTLD) di Sistem Nama Domain sedang berlangsung. Hati-hati dengan tabrakan nama saat memilih nama domain yang hanya digunakan dalam LAN.

Catatan
Penggunaan paket seperti `libnss-resolve` bersama dengan `systemd-resolved`, atau `libnss-myhostname`, atau `libnss-mymachine`, dengan daftar yang sesuai pada baris "`hosts`" di berkas "`/etc/nsswitch.conf`" dapat menggantikan konfigurasi jaringan tradisional yang dibahas di atas. Lihat nss-resolve(8), systemd-resolved(8), nss-myhostname(8), dan nss-mymachines(8) untuk informasi lebih lanjut.

5.1.2 Nama antarmuka jaringan

systemd menggunakan "Nama Antarmuka Jaringan yang Dapat Diprediksi" seperti "`enp0s25`".

Kelas	alamat jaringan	net mask	net mask /bit	banyaknya subnet
A	10.x.x.x	255.0.0.0	/8	1
B	172.16.x.x — 172.31.x.x	255.255.0.0	/16	16
C	192.168.0.x — 192.168.255.x	255.255.255.0	/24	256

Tabel 5.2: Daftar rentang alamat jaringan

5.1.3 Rentang alamat jaringan untuk LAN

Mari kita diingatkan tentang rentang alamat IPv4 32 bit di setiap kelas yang disediakan untuk digunakan pada jaringan area lokal (LAN) oleh rfc1918. Alamat-alamat ini dijamin tidak konflik dengan alamat apa pun di Internet yang benar.

Catatan
Alamat IP yang ditulis dengan titik dua adalah alamat IPv6, misalnya, "`::1`" untuk `localhost`.

Catatan
Jika salah satu alamat ini ditugaskan ke host, maka host tersebut tidak boleh mengakses Internet secara langsung tetapi harus mengaksesnya melalui gateway yang bertindak sebagai proksi untuk layanan individual atau network address translation (NAT). Router broadband biasanya melakukan NAT untuk lingkungan LAN konsumen.

5.1.4 Dukungan perangkat jaringan

Meskipun sebagian besar perangkat keras didukung oleh sistem Debian, ada beberapa perangkat jaringan yang memerlukan firmware non-free DFSG untuk mendukungnya. Silakan lihat Bagian 9.10.5.

5.2 Konfigurasi jaringan modern untuk desktop

Antarmuka jaringan biasanya diinisialisasi dalam "`networking.service`" untuk antarmuka `lo` dan "`NetworkManager.se`
untuk antarmuka lain pada sistem desktop Debian modern di bawah `systemd`.

Debian dapat mengelola koneksi jaringan melalui perangkat lunak daemon manajemen seperti NetworkManager (NM) (network-manager dan paket terkait).

- Mereka datang dengan GUI dan program baris perintah mereka sendiri sebagai antarmuka pengguna mereka.
- Mereka datang dengan daemon mereka sendiri sebagai sistem backend mereka.
- Mereka memungkinkan koneksi mudah dari sistem Anda ke Internet.
- Mereka memungkinkan manajemen yang mudah dari konfigurasi jaringan kabel dan nirkabel.
- Mereka memungkinkan kita untuk mengonfigurasi jaringan independen dari paket `ifupdown` warisan.

Catatan
Jangan gunakan alat konfigurasi jaringan otomatis ini untuk server. Ini ditujukan terutama untuk pengguna desktop yang berpindah-pindah di laptop.

Alat konfigurasi jaringan modern ini perlu dikonfigurasi dengan benar untuk menghindari konflik dengan paket `ifupdown` warisan dan berkas konfigurasinya "`/etc/network/interfaces`".

5.2.1 Alat konfigurasi jaringan GUI

Dokumentasi resmi untuk NM pada Debian disediakan di "`/usr/share/doc/network-manager/README.Debian`".

Pada dasarnya, konfigurasi jaringan untuk desktop dilakukan sebagai berikut.

1. Jadikan pengguna desktop, misalnya `foo`, masuk grup "`netdev`" dengan yang berikut (Atau, lakukan secara otomatis melalui D-bus di bawah lingkungan desktop modern seperti GNOME dan KDE).

```
$ sudo usermod -a -G netdev foo
```

2. Mempertahankan konfigurasi "`/etc/network/interfaces`" sesederhana yang berikut ini.

```
auto lo
iface lo inet loopback
```

3. Jalankan ulang NM dengan yang berikut.

```
$ sudo systemctl restart NetworkManager
```

4. Konfigurasikan jaringan Anda melalui GUI.

Catatan
Hanya antarmuka yang **tidak** tercantum dalam "`/etc/network/interfaces`" yang dikelola oleh NM untuk menghindari konflik dengan `ifupdown`.

Tip
Jika Anda ingin memperluas kemampuan konfigurasi jaringan NM, silakan mencari modul plug-in yang sesuai dan paket tambahan seperti `network-manager-openconnect`, `network-manager-openvpn-gnome`, `network-manager-pptp-gnome`, `mobile-broadband-provider-info`, `gnome-bluetooth`, dll.

5.3 Konfigurasi jaringan modern tanpa GUI

Di bawah systemd, jaringan dapat dikonfigurasi dalam `/etc/systemd/network/` sebagai gantinya. Lihat systemd-resolved(8), resolved.conf(5), dan systemd-networkd(8).

Ini memungkinkan konfigurasi jaringan modern tanpa GUI.

Konfigurasi klien DHCP dapat diatur dengan membuat "`/etc/systemd/network/dhcp.network`". Mis.:

```
[Match]
Name=en*

[Network]
DHCP=yes
```

Konfigurasi jaringan statis dapat diatur dengan membuat "`/etc/systemd/network/static.network`". Mis.:

```
[Match]
Name=en*

[Network]
Address=192.168.0.15/24
Gateway=192.168.0.1
```

5.4 Konfigurasi jaringan modern untuk cloud

Konfigurasi jaringan modern untuk cloud mungkin memakai paket `cloud-init` dan `netplan.io` (lihat Bagian 3.8.4).

Paket `netplan.io` mendukung `systemd-networkd` dan `NetworkManager` sebagai backend konfigurasi jaringannya, dan memungkinkan konfigurasi jaringan deklaratif menggunakan data YAML. Ketika Anda mengubah YAML:

- Jalankan perintah "`netplan generate`" untuk menghasilkan semua konfigurasi backend yang diperlukan dari YAML.
- Jalankan perintah "`netplan apply`" untuk menerapkan konfigurasi yang dibuat ke backend.

Lihat "dokumentasi Netplan", netplan(5), netplan-generate(8), dan netplan-apply(8).

Lihat juga "Dokumentasi Cloud-init" (terutama di sekitar "Sumber konfigurasi" dan "Netplan Passthrough") untuk mengetahui bagaimana `cloud-init` dapat mengintegrasikan konfigurasi `netplan.io` dengan sumber data alternatif.

5.4.1 Konfigurasi jaringan modern untuk cloud dengan DHCP

Konfigurasi klien DHCP dapat disiapkan dengan membuat berkas sumber data "`/etc/netplan/50-dhcp.yaml`":

```
network:
  version: 2
  ethernets:
    all-en:
      match:
        name: "en*"
      dhcp4: true
      dhcp6: true
```

5.4.2 Konfigurasi jaringan modern untuk cloud dengan IP statik

Konfigurasi jaringan statis dapat disiapkan dengan membuat berkas sumber data "`/etc/netplan/50-static.yaml`":

```
network:
  version: 2
  ethernets:
    eth0:
      addresses:
        - 192.168.0.15/24
      routes:
        - to: default
          via: 192.168.0.1
```

5.4.3 Konfigurasi jaringan modern untuk cloud dengan Network Manager

Konfigurasi jaringan klien memakai infrastruktur NetworkManager dapat disiapkan dengan membuat berkas sumber data "`/etc/netplan/00-network-manager.yaml`".:

```
network:
  version: 2
  renderer: NetworkManager
```

5.5 Konfigurasi jaringan tingkat rendah

Untuk konfigurasi jaringan tingkat rendah di Linux, gunakan program iproute2 (ip(8), …) .

5.5.1 Perintah Iproute2

Perintah Iproute2 menawarkan kemampuan konfigurasi jaringan tingkat rendah yang lengkap. Berikut adalah tabel terjemahan dari perintah net-tools yang usang ke perintah-perintah baru iproute2 dll.

net-tools usang	**iproute2 baru dll.**	**manipulasi**
ifconfig(8)	`ip addr`	alamat protokol (IP atau IPv6) pada perangkat
route(8)	`ip route`	entri tabel routing
arp(8)	`ip neigh`	Entri singgahan ARP atau NDISC
`ipmaddr`	`ip maddr`	alamat multicast
`iptunnel`	`ip tunnel`	tunnel di atas IP
nameif(8)	ifrename(8)	menamai antarmuka jaringan berdasarkan alamat MAC
mii-tool(8)	ethtool(8)	Pengaturan perangkat Ethernet

Tabel 5.3: Tabel terjemahan dari perintah `net-tools` yang usang ke perintah-perintah baru `iproute2`

Lihat ip(8) dan Linux Advanced Routing & Traffic Control.

5.5.2 Operasi jaringan tingkat rendah yang aman

Anda dapat menggunakan perintah jaringan tingkat rendah sebagai berikut dengan aman karena mereka tidak mengubah konfigurasi jaringan.

perintah	**deskripsi**
`ip addr show`	menampilkan status link dan alamat antarmuka aktif
`route -n`	menampilkan semua tabel routing dalam alamat numerik
`ip route show`	menampilkan semua tabel routing dalam alamat numerik
`arp`	menampilkan konten saat ini dari tabel singgahan ARP
`ip neigh`	menampilkan konten saat ini dari tabel singgahan ARP
`plog`	menampilkan log daemon ppp
`ping yahoo.com`	memeriksa koneksi internet ke ”`yahoo.com`”
`whois yahoo.com`	memeriksa siapa yang mendaftarkan ”`yahoo.com`” dalam basis data domain
`traceroute yahoo.com`	melacak koneksi Internet ke ”`yahoo.com`”
`tracepath yahoo.com`	melacak koneksi Internet ke ”`yahoo.com`”
`mtr yahoo.com`	melacak koneksi Internet ke ”`yahoo.com`” (berulang kali)
`dig [@dns-server.com] example.com [{a|mx|any}]`	memeriksa catatan DNS ”`example.com`” oleh ”`dns-server.com`” untuk record ”`a`”, ”`mx`”, atau ”`any`
`iptables -L -n`	memeriksa penyaringan paket
`ss -a`	cari semua port yang terbuka
`ss -l`	cari port yang mendengarkan
`ss -l -t -n`	menemukan port TCP yang mendengarkan (numerik)
`dlint example.com`	memeriksa informasi zona DNS dari ”`example.com`”

Tabel 5.4: Daftar perintah jaringan tingkat rendah

Tip
Beberapa alat konfigurasi jaringan tingkat rendah ini berada di "`/usr/sbin/`". Anda mungkin perlu mengeluarkan path perintah lengkap seperti "`/usr/sbin/ifconfig`" atau menambahkan "`/usr/sbin`" ke daftar "`$PATH`" di "`~/.bashrc`" Anda.

5.6 Optimalisasi jaringan

Optimasi jaringan generik berada di luar lingkup dokumentasi ini. Saya hanya menyentuh subjek yang berkaitan dengan koneksi kelas konsumen.

paket	popcon	ukuran	deskripsi
`iftop`	V:6, I:85	93	menampilkan informasi penggunaan bandwidth pada antarmuka jaringan
`iperf`	V:2, I:32	431	Alat pengukur bandwidth Protokol Internet
`ifstat`	V:0.5, I:5.3	52	InterFace STATistics Monitoring
`bmon`	V:2, I:19	154	pemantau bandwidth portabel dan estimator laju
`ethstatus`	V:0.2, I:2.3	41	skrip yang dengan cepat mengukur throughput perangkat jaringan
`bing`	V:0.08, I:0.47	80	penguji bandwidth stokastik empiris
`bwm-ng`	V:1.0, I:9.1	95	pemantau bandwidth berbasis konsol kecil dan sederhana
`ethstats`	V:0.05, I:0.36	21	pemantau statistik Ethernet berbasis konsol
`ipfm`	V:0.04, I:0.09	78	alat analisis bandwidth

Tabel 5.5: Daftar alat optimalisasi jaringan

5.6.1 Mencari MTU yang optimal

NM biasanya menetapkan Maximum Transmission Unit (MTU) yang optimal secara otomatis.

Dalam beberapa kesempatan, Anda mungkin ingin mengatur MTU secara manual setelah percobaan dengan ping(8) dengan opsi "`-M do`" untuk mengirim paket ICMP dengan berbagai ukuran paket data. MTU adalah ukuran paket data maksimum yang berhasil tanpa fragmentasi IP ditambah 28 byte untuk IPv4 dan ditambah 48 byte untuk IPv6. Misalnya berikut menemukan MTU untuk koneksi IPv4 menjadi 1460 dan MTU untuk koneksi IPv6 menjadi 1500.

```
$ ping -4 -c 1 -s $((1500-28)) -M do www.debian.org
PING  (149.20.4.15) 1472(1500) bytes of data.
ping: local error: message too long, mtu=1460

---  ping statistics ---
1 packets transmitted, 0 received, +1 errors, 100% packet loss, time 0ms

$ ping -4 -c 1 -s $((1460-28)) -M do www.debian.org
PING  (130.89.148.77) 1432(1460) bytes of data.
1440 bytes from klecker-misc.debian.org (130.89.148.77): icmp_seq=1 ttl=50 time=325 ms

---  ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 325.318/325.318/325.318/0.000 ms
$ ping -6 -c 1 -s $((1500-48)) -M do www.debian.org
PING www.debian.org(mirror-csail.debian.org (2603:400a:ffff:bb8::801f:3e)) 1452 data bytes
```

```
1460 bytes from mirror-csail.debian.org (2603:400a:ffff:bb8::801f:3e): icmp_seq=1 ttl=47 ↩
    time=191 ms

--- www.debian.org ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 191.332/191.332/191.332/0.000 ms
```

Proses ini adalah penemuan Path MTU (PMTU) (RFC1191) dan perintah tracepath(8) dapat mengotomatisasi ini.

lingkungan jaringan	MTU	alasan
Link dial-up (IP: PPP)	576	standar
Link Ethernet (IP: DHCP atau tetap)	1500	standar dan baku

Tabel 5.6: Panduan dasar dari nilai MTU yang optimal

Selain panduan dasar ini, Anda harus tahu hal berikut.

- Setiap penggunaan metode tunneling (VPN dll.) dapat mengurangi MTU optimal lebih lanjut dengan overhead mereka.
- Nilai MTU tidak boleh melebihi nilai PMTU yang ditentukan secara eksperimental.
- Nilai MTU yang lebih besar umumnya lebih baik ketika keterbatasan lain terpenuhi.

Ukuran segmen maksimum (maximum segment size/MSS) digunakan sebagai ukuran alternatif ukuran paket. Hubungan antara MSS dan MTU adalah sebagai berikut.

- MSS = MTU - 40 untuk IPv4
- MSS = MTU - 60 untuk IPv6

Catatan
Iptables(8) (lihat Bagian 5.7) optimasi berbasis dapat menjepit ukuran paket oleh MSS dan berguna untuk router. Lihat ”TCPMSS” di iptables(8).

5.6.2 Optimasi TCP WAN

Throughput TCP dapat dimaksimalkan dengan menyesuaikan parameter ukuran penyangga TCP seperti yang dijelaskan dalam ”Penalaan TCP” untuk WAN bandwidth tinggi dan latensi tinggi modern. Sejauh ini, pengaturan baku Debian saat ini berfungsi dengan baik bahkan untuk LAN saya yang terhubung oleh layanan FTTP 1G bps cepat.

5.7 Infrastruktur netfilter

Netfilter menyediakan infrastruktur untuk stateful firewall dan network address translation (NAT) dengan modul kernel Linux (lihat Bagian 3.10).

Lihat Proyek ”nftables” netfilter.org untuk perintah ruang pengguna nft(8) yang lebih baru dalam paket `nftables`. (Linux kernel >= 3.13)

Lihat Proyek ”iptables” netfilter.org untuk perintah ruang pengguna iptables(8) dan yang serupa, yang lebih lama, dalam paket `iptables` dan yang serupa. (Linux kernel >= 2.4.x)

paket	popcon	ukuran	deskripsi
nftables	V:266, I:880	189	alat administrasi untuk penyaringan paket dan NAT (Netfilter) (penerus {ip,ip6,arp,eb}tables)
iptables	V:381, I:635	2408	alat administrasi untuk netfilter (iptables(8) untuk IPv4, ip6tables(8) untuk IPv6)
arptables	V:0.1, I:1.6	102	alat administrasi untuk netfilter (arptables(8) untuk ARP)
ebtables	V:14, I:23	276	alat administrasi untuk netfilter (ebtables(8) untuk Ethernet bridging)
iptstate	V:0.1, I:1.6	121	terus-menerus memantau keadaan netfilter (mirip dengan top(1))
ufw	V:83, I:109	862	Uncomplicated Firewall (UFW) adalah program untuk mengelola firewall netfilter
gufw	V:6, I:11	3682	antarmuka pengguna grafis untuk Uncomplicated Firewall (UFW)
firewalld	V:15, I:22	2581	firewalld adalah program firewall yang dikelola secara dinamis dengan dukungan untuk zona jaringan
firewall-config	V:0.8, I:3.2	1096	antarmuka pengguna grafis untuk firewalld
shorewall-init	V:0.17, I:0.33	88	Inisialisasi Firewall Shoreline
shorewall	V:2.1, I:4.7	3090	Firewall Shoreline, generator berkas konfigurasi netfilter
shorewall-lite	V:0.03, I:0.06	71	Firewall Shoreline, generator berkas konfigurasi netfilter (versi ringan)
shorewall6	V:0.6, I:1.2	1334	Firewall Shoreline, generator berkas konfigurasi netfilter (versi IPv6)
shorewall6-lite	V:0.01, I:0.02	71	Firewall Shoreline, generator berkas konfigurasi netfilter (IPv6, versi ringan)

Tabel 5.7: Daftar alat firewall

Anda dapat mengonfigurasi secara manual netfilter secara interaktif dari shell, menyimpan statusnya dengan "`nft list ruleset > file`" atau iptables-save(8), dan mengembalikannya melalui skrip init dengan "`nft -f file`" atau iptables-restore(8) saat boot ulang sistem.

Skrip pembantu konfigurasi seperti shorewall memudahkan proses ini.

Lihat dokumentasi di Dokumentasi Netfilter (atau di "`/usr/share/doc/iptables/html/`").

- Linux Networking-concepts HOWTO
- Linux 2.4 Packet Filtering HOWTO
- Linux 2.4 NAT HOWTO
- nftables HOWTO

Bab 6

Aplikasi jaringan

Setelah menjalin konektivitas jaringan (lihat Bab 5), Anda dapat menjalankan berbagai aplikasi jaringan.

Tip
Untuk panduan khusus Debian modern bagi infrastruktur jaringan, baca Buku Pegangan Administrator Debian — Infrastruktur Jaringan.

Tip
Jika Anda mengaktifkan ”Verifikasi 2 Langkah” dengan beberapa ISP, Anda perlu mendapatkan kata sandi aplikasi untuk mengakses layanan POP dan SMTP dari program Anda. Anda mungkin perlu menyetujui IP host Anda terlebih dahulu.

6.1 Peramban Web

Ada banyak paket peramban web untuk mengakses konten jarak jauh dengan Hypertext Transfer Protocol (HTTP).

paket	popcon	ukuran	jenis	deskripsi peramban web
chromium	V:31, I:100	320285	X	Chromium, (peramban open-source dari Google)
firefox	V:15, I:21	308351	, ,	Firefox, (peramban open-source dari Mozilla, hanya tersedia di Debian Unstable)
firefox-esr	V:186, I:425	299827	, ,	Firefox ESR, (Firefox Extended Support Release)
epiphany-browser	V:2, I:10	6735	, ,	GNOME, patuh HIG, Epiphany
konqueror	V:26, I:111	8179	, ,	KDE, Konqueror
dillo	V:0.4, I:3.9	2203	, ,	Dillo, (peramban ringan, berbasis FLTK)
w3m	V:10, I:129	2853	teks	w3m
lynx	V:29, I:455	2031	, ,	Lynx
elinks	V:2, I:15	1791	, ,	ELinks
links	V:2, I:20	2321	, ,	Links (hanya teks)
links2	V:0.8, I:9.4	5466	grafis	Links (konsol grafis tanpa X)

Tabel 6.1: Daftar peramban web

6.1.1 Memalsu string User-Agent

Untuk mengakses beberapa situs web yang terlalu membatasi, Anda mungkin perlu menipu string User-Agent yang dikembalikan oleh program browser web. Lihat:

- MDN Web Docs: userAgent
- Pengembang Chrome: Menimpa string user agent
- Cara mengubah user agent Anda
- Cara Mengubah User-Agent di Chrome, Firefox, Safari, dan banyak lagi
- Cara Mengubah User Agent Peramban Anda Tanpa Memasang Ekstensi Apa Pun
- Cara mengubah User Agent di Gnome Web (epiphany)

6.1.2 Ekstensi peramban

Semua peramban GUI modern mendukung ekstensi browser berbasis kode sumber dan menjadi standar sebagai ekstensi web.

6.2 Sistem surat

Bagian ini berfokus pada workstation umum yang dapat berpindah pada koneksi Internet kelas konsumen.



Perhatian
Jika Anda ingin mengatur server surel untuk bertukar surat langsung dengan Internet, Anda harus lebih baik daripada membaca dokumen dasar ini.

6.2.1 Dasar-dasar surel

Pesan surel terdiri dari tiga komponen, amplop pesan, header pesan, dan badan pesan.

- Informasi ”To” dan ”From” dalam amplop pesan digunakan oleh SMTP untuk mengirimkan surel. (Informasi ”From” dalam amplop pesan juga disebut alamat bounce, From_, dll.).
- Informasi ”To” dan ”From” di header pesan ditampilkan oleh klien surel. (Meskipun paling umum bagi ini untuk menjadi sama dengan yang ada di amplop pesan, seperti itu tidak selalu terjadi.)
- Format pesan surel yang mencakup header dan data tubuh diperluas oleh Multipurpose Internet Mail Extensions (MIME) dari teks ASCII polos ke pengodean karakter lainnya, serta lampiran audio, video, gambar, dan program aplikasi.

Klien surel berbasis GUI berfitur lengkap menawarkan semua fungsi berikut menggunakan konfigurasi intuitif berbasis GUI.

- Itu membuat dan menginterpretasi header pesan dan data tubuh memakai Multipurpose Internet Mail Extensions (MIME) untuk menangani pengkodean dan tipe data konten.
- Ini mengautentikasi dirinya ke server SMTP dan IMAP ISP menggunakan otentikasi akses dasar warisan atau OAuth 2.0modern. (Untuk OAuth 2.0, atur itu melalui pengaturan lingkungan Desktop. Mis., ”Pengaturan” -> ”Akun Daring”.)

- Itu mengirimkan pesan ke server SMTP smarthost ISP yang mendengarkan port message submission (587).
- Itu menerima pesan yang tersimpan di server ISP dari port TLS/IMAP4 (993).
- Ini dapat menyaring surat dengan atribut mereka.
- Ini mungkin menawarkan fungsionalitas tambahan: Kontak, Kalender, Tugas, Memo.

paket	popcon	ukuran	jenis
`evolution`	V:27, I:229	489	Program GUI X (GNOME, suite groupware)
`thunderbird`	V:39, I:102	303329	Program X GUI (GTK, Mozilla Thunderbird)
`kmail`	V:43, I:103	25265	Program X GUI (KDE)
`mutt`	V:10, I:82	7334	program terminal karakter yang mungkin digunakan dengan `vim`
`mew`	V:0.01, I:0.14	2319	program terminal karakter di bawah `(x)emacs`

Tabel 6.2: Daftar agen pengguna surat (mail user agent/MUA)

6.2.2 Batasan layanan surat modern

Layanan surat modern berada mengalami beberapa keterbatasan untuk meminimalkan paparan masalah spam (surel yang tidak diinginkan dan tidak diminta).

- Tidak realistis untuk menjalankan server SMTP di jaringan kelas konsumen untuk mengirim surel langsung ke host jarak jauh secara andal.
- Sebuah surat dapat ditolak oleh setiap host dalam perjalanan ke tujuan diam-diam kecuali tampak seotentik mungkin.
- Tidak realistis untuk mengharapkan satu smarthost untuk mengirim surel dari alamat surat sumber yang tidak terkait ke host jarak jauh secara andal.

Ini karena:

- Koneksi port SMTP (25) dari host yang dilayani oleh jaringan kelas konsumen ke Internet diblokir.
- Koneksi port SMTP (25) ke host yang dilayani oleh jaringan kelas konsumen dari Internet diblokir.
- Pesan keluar dari host yang dilayani oleh jaringan kelas konsumen ke Internet hanya dapat dikirim melalui port message submission (587).
- Teknik anti-spam seperti DomainKeys Identified Mail (DKIM), Sender_Policy_Framework (SPF), dan Domain-based Message Authentication, Reporting and Conformance (DMARC) banyak digunakan untuk penyaringan surel.
- Layanan DomainKeys Identified Mail dapat disediakan untuk surel Anda yang dikirim melalui smarthost.
- Smarthost dapat menulis ulang alamat surel sumber di header pesan ke akun surel Anda di smarthost untuk mencegah spoofing alamat surel.

6.2.3 Harapan layanan surat bersejarah

Beberapa program di Debian berharap untuk mengakses perintah `/usr/sbin/sendmail` untuk mengirim surel sebagai pengaturan baku atau yang disesuaikan karena layanan surel pada sistem UNIX berfungsi secara historis sebagai:

- Surel dibuat sebagai suatu berkas teks.
- Surel diserahkan ke perintah `/usr/sbin/sendmail`.
- Untuk alamat tujuan pada host yang sama, perintah `/usr/sbin/sendmail` melakukan pengiriman surel lokal dengan menambahkannya ke berkas `/var/mail/$username`.
  - Perintah yang mengharapkan fitur ini: `apt-listchanges`, `cron`, `at`, ...
- Untuk alamat tujuan pada host jarak jauh, perintah `/usr/sbin/sendmail` melakukan transfer jarak jauh surel ke host tujuan yang ditemukan oleh catatan DNS MX menggunakan SMTP.
  - Perintah yang mengharapkan fitur ini: `popcon`, `reportbug`, `bts`, ...

6.2.4 Agen transportasi surat (mail transport agent/MTA)

Workstation mobile Debian dapat dikonfigurasi dengan klien surel berbasis GUI berfitur lengkap tanpa program mail transfer agent (MTA) setelah Debian 12 Bookworm.

Debian secara tradisional memasang beberapa program MTA untuk mendukung program yang mengharapkan perintah `/usr/sbin/sendmail`. MTA tersebut pada workstation mobile harus mengatasi Bagian 6.2.2 dan Bagian 6.2.3.

Untuk workstation mobile, pilihan khas MTA adalah `exim4-daemon-light` atau `postfix` dengan opsi instalasinya seperti ”Mail sent by smarthost; received via SMTP or fetchmail” dipilih. Ini adalah MTA ringan yang menghormati ”`/etc/aliases`”.

Tip
Mengkonfigurasi `exim4` untuk mengirim surel Internet melalui beberapa smarthost yang sesuai untuk beberapa alamat surel sumber tidak sepele. Jika Anda memerlukan kemampuan seperti itu untuk beberapa program, siapkan menggunakan `msmtp` yang mudah diatur untuk beberapa alamat surel sumber. Kemudian tinggalkan MTA utama hanya untuk satu alamat surel.

paket	popcon	ukuran	deskripsi
`exim4-daemon-light`	V:200, I:209	1776	Mail transport agent Exim4 (MTA: baku Debian)
`exim4-daemon-heavy`	V:4.8, I:4.9	1954	Mail transport agent Exim4 (MTA: alternatif fleksibel)
`exim4-base`	V:205, I:214	1685	Dokumentasi (teks) dan berkas umum Exim4
`exim4-doc-html`	I:0.98	3809	Dokumentasi Exim4 (html)
`exim4-doc-info`	I:0.55	649	Dokumentasi Exim4 (info)
`postfix`	V:102, I:107	4274	Mail transport agent Postfix (MTA: alternatif aman)
`postfix-doc`	I:4.3	5028	Dokumentasi Postfix (html+text)
`sasl2-bin`	V:4.5, I:9.9	381	Implementasi API Cyrus SASL (suplemen postfix untuk SMTP AUTH)
`cyrus-sasl2-doc`	I:0.68	2140	Cyrus SASL - dokumentasi
`msmtp`	V:9, I:15	822	MTA ringan
`msmtp-mta`	V:7.2, I:9.1	139	MTA ringan (ekstensi kompatibilitas sendmail ke `msmtp`)
`nullmailer`	V:7.4, I:7.7	483	MTA fungsi minimal, tidak ada surat lokal
`ssmtp`	V:3.9, I:5.9	134	MTA fungsi minimal, tidak ada surat lokal
`sendmail-bin`	V:10, I:10	1954	MTA berfitur lengkap (hanya jika Anda sudah akrab)
`git-email`	V:0.3, I:9.3	1229	program git-send-email(1) untuk mengirim seri surel patch

Tabel 6.3: Daftar paket terkait mail transport agent dasar

6.2.4.1 Konfigurasi exim4

Untuk surel Internet melalui smarthost, Anda mengonfigurasi (ulang) paket `exim4-*` sebagai berikut.

```
$ sudo systemctl stop exim4
$ sudo dpkg-reconfigure exim4-config
```

Pilih ”surat dikirim oleh smarthost; diterima melalui SMTP atau fetchmail” untuk ”Konfigurasi surat tipe umum”.

Atur ”Nama surat sistem:” ke bakunya sebagai FQDN (lihat Bagian 5.1.1).

Atur ”Alamat IP untuk didengarkan bagi koneksi SMTP masuk:” ke bakunya sebagai ”127.0.0.1 ; ::1”.

Hapus isi dari ”Tujuan lain yang surat akan diterima:”.

Hapus isi dari ”Mesin untuk me-relay surat:”.

Atur ”Alamat IP atau nama host smarthost keluar:” ke ”smtp.hostname.dom:587”.

Pilih ”Tidak” untuk ”Sembunyikan nama surat lokal di surat keluar?”. (Gunakan ”`/etc/email-addresses`” seperti dalam Bagian 6.2.4.3, sebagai gantinya.)

Jawab ke ”Pertahankan banyaknya DNS-query minimal (Dial-on-Demand)?” sebagai salah satu dari berikut ini.

- ”Tidak” jika sistem terhubung ke Internet saat boot.
- ”Ya” jika sistem **tidak** terhubung ke Internet saat boot.

Atur ”Metode pengiriman untuk surat lokal:” ke ”format mbox di /var/mail/”.

Pilih ”Ya” untuk ”Pisahkan konfigurasi menjadi berkas-berkas kecil?:”.

Buat entri kata sandi untuk smarthost dengan menyunting ”`/etc/exim4/passwd.client`”.

```
$ sudo vim /etc/exim4/passwd.client
 ...
$ cat /etc/exim4/passwd.client
^smtp.*\.hostname\.dom:username@hostname.dom:password
```

Konfigurasikan exim4(8) dengan ”`QUEUERUNNER='queueonly'`”, ”`QUEUERUNNER='nodaemon'`”, dll. dalam”`/etc/defau` untuk meminimalkan penggunaan sumber daya sistem. (opsional)

Jalankan `exim4` dengan yang berikut.

```
$ sudo systemctl start exim4
```

Nama host di ”`/etc/exim4/passwd.client`” tidak boleh alias. Anda memeriksa nama host asli dengan yang berikut.

```
$ host smtp.hostname.dom
smtp.hostname.dom is an alias for smtp99.hostname.dom.
smtp99.hostname.dom has address 123.234.123.89
```

Saya menggunakan regex di ”`/etc/exim4/passwd.client`” untuk mengatasi masalah alias. SMTP AUTH mungkin bekerja bahkan jika ISP memindahkan host yang ditunjuk oleh alias.

Anda dapat memperbarui konfigurasi `exim4` secara manual dengan yang berikut:

- Perbarui berkas konfigurasi `exim4` di ”`/etc/exim4/`”.
  - membuat ”`/etc/exim4/exim4.conf.localmacros`” untuk mengatur MACRO dan menyunting ”`/etc/exim4/exi` (konfigurasi non-split)
  - membuat berkas baru atau menyunting berkas yang ada di subdirektori ”`/etc/exim4/exim4.conf.d`”. (konfigurasi terpisah)

- Jalankan "`systemctl reload exim4`".



Perhatian
Memulai `exim4` membutuhkan waktu lama jika "Tidak" (nilai baku) dipilih untuk kueri debconf "Pertahankan cacah kueri DNS minimal (Dial-on-Demand)?" dan sistem **tidak** terhubung ke Internet saat boot.

Silakan baca panduan resmi di: "`/usr/share/doc/exim4-base/README.Debian.gz`" dan update-exim4.conf(8).



Awas
Untuk semua pertimbangan praktis, gunakan SMTP dengan STARTTLS di port 587 atau SMTPS SSL (SMTP di atas SSL) di port 465, bukan SMTP polos di port 25.

6.2.4.2 Konfigurasi postfix dengan SASL

Untuk surel Internet melalui smarthost, Anda harus terlebih dahulu membaca dokumentasi postfix dan halaman manual utama.

perintah	fungsi
postfix(1)	Program kontrol postfix
postconf(1)	Utilitas konfigurasi postfix
postconf(5)	Parameter konfigurasi postfix
postmap(1)	Pemeliharaan tabel pencarian postfix
postalias(1)	Pemeliharaan basis data alias postfix

Tabel 6.4: Daftar halaman-halaman penting manual postfix

Anda mengonfigurasi (ulang) paket `postfix` dan `sasl2-bin` sebagai berikut.

```
$ sudo systemctl stop postfix
$ sudo dpkg-reconfigure postfix
```

Pilih "Internet dengan smarthost".

Atur "Host relay SMTP (kosongkan untuk tidak ada):" ke "`[smtp.namahost.dom]:587`" dan konfigurasikan dengan yang berikut.

```
$ sudo postconf -e 'smtp_sender_dependent_authentication = yes'
$ sudo postconf -e 'smtp_sasl_auth_enable = yes'
$ sudo postconf -e 'smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd'
$ sudo postconf -e 'smtp_sasl_type = cyrus'
$ sudo vim /etc/postfix/sasl_passwd
```

Buat entri kata sandi untuk smarthost.

```
$ cat /etc/postfix/sasl_passwd
[smtp.hostname.dom]:587     username:password
$ sudo postmap hush:/etc/postfix/sasl_passwd
```

Jalankan `postfix` dengan yang berikut ini.

```
$ sudo systemctl start postfix
```

Di sini penggunaan "`[`" dan "`]`" dalam dialog `dpkg-reconfigure` dan "`/etc/postfix/sasl_passwd`" memastikan untuk tidak memeriksa catatan MX tetapi langsung menggunakan nama host yang tepat yang ditentukan. Lihat "Mengaktifkan autentikasi SASL di klien Postfix SMTP" di "`/usr/share/doc/postfix/html/SASL_README.html`".

6.2.4.3 Konfigurasi alamat surel

Ada beberapa berkas konfigurasi alamat surat untuk transportasi surat, pengiriman. dan agen pengguna.

berkas	fungsi	aplikasi
`/etc/mailname`	nama host baku untuk surat (keluar)	Spesifik Debian, mailname(5)
`/etc/email-addresses`	spoofing nama host untuk surat keluar	spesifik exim(8), exim4-config_files(5)
`/etc/postfix/generic`	spoofing nama host untuk surat keluar	spesifik postfix(1), diaktifkan setelah eksekusi perintah postmap(1).
`/etc/aliases`	alias nama akun untuk surat masuk	umum, diaktifkan setelah eksekusi perintah newaliases(1).

Tabel 6.5: Daftar berkas konfigurasi terkait alamat surel

mailname dalam berkas "`/etc/mailname`" biasanya merupakan fully qualified domain name (FQDN) yang diurai ke salah satu alamat IP host. Untuk workstation mobile yang tidak memiliki nama host dengan alamat IP yang dapat diuraikan, atur **mailname** ini ke nilai "`hostname -f`". (Ini adalah pilihan yang aman dan bekerja untuk `exim4-*` dan `postfix`.)

Tip
Isi "`/etc/mailname`" digunakan oleh banyak program non-MTA untuk perilaku baku mereka. Untuk `mutt`, atur variabel "`hostname`" dan "`from`" dalam berkas `~/muttrc` untuk menimpa nilai **mailname**. Untuk program dalam paket `devscripts`, seperti bts(1) dan dch(1), ekspor variabel lingkungan "`$DEBFULLNAME`" dan "`$DEBEMAIL`" untuk menimpanya.

Tip
Paket `popularity-contest` biasanya mengirim surat dari akun root dengan FQDN. Anda perlu mengatur `MAILFROM` di `/etc/popularity-contest.conf` seperti yang dijelaskan dalam berkas `/usr/share/popularity-contest/default.conf`. Jika tidak, surat Anda akan ditolak oleh server SMTP smarthost. Meskipun ini membosankan, pendekatan ini lebih aman daripada menulis ulang alamat sumber untuk semua surat dari root oleh MTA dan harus digunakan untuk daemon dan skrip cron lainnya.

Saat mengatur **mailname** ke "`hostname -f`", spoofing alamat surat sumber melalui MTA dapat direalisasikan dengan yang berikut.

- berkas "`/etc/email-addresses`" untuk exim4(8) seperti dijelaskan dalam exim4-config_files(5)
- berkas "`/etc/postfix/generic`" untuk postfix(1) seperti yang dijelaskan dalam generic(5)

Untuk `postfix`, diperlukan langkah-langkah tambahan berikut.

```
# postmap hash:/etc/postfix/generic
# postconf -e 'smtp_generic_maps = hash:/etc/postfix/generic'
# postfix reload
```

Anda dapat menguji konfigurasi alamat surat menggunakan yang berikut ini.

- exim(8) dengan opsi-opsi `-brw, -bf, -bF, -bV, …`
- postmap(1) dengan opsi `-q`.

Tip
Exim hadir dengan beberapa program utilitas seperti exiqgrep(8) dan exipick(8). Lihat "`dpkg -L exim4-base|grep man8/`" untuk perintah yang tersedia.

6.2.4.4 Operasi dasar MTA

Ada beberapa operasi dasar MTA. Beberapa dapat dilakukan melalui antarmuka kompatibilitas sendmail(1).

perintah exim	perintah postfix	deskripsi
`sendmail`	`sendmail`	membaca surat dari masukan standar dan mengatur untuk pengiriman (`-bm`)
`mailq`	`mailq`	daftar antrian surat dengan status dan ID antrian (`-bp`)
`newaliases`	`newaliases`	menginisialisasi basis data alias `-I`
`exim4 -q`	`postqueue -f`	mendorong surat-surat yang menunggu (`-q`)
`exim4 -qf`	`postsuper -r ALL deferred; postqueue -f`	mendorong semua surat
`exim4 -qff`	`postsuper -r ALL; postqueue -f`	mendorong bahkan surat-surat yang dibekukan
`exim4 -Mg queue_id`	`postsuper -h queue_id`	bekukan satu pesan berdasarkan ID antriannya
`exim4 -Mrm queue_id`	`postsuper -d queue_id`	menghapus satu pesan berdasarkan ID antriannya
T/T	`postsuper -d ALL`	menghapus semua pesan

Tabel 6.6: Daftar operasi dasar MTA

Tip
Mungkin ide yang baik untuk mendorong (flush) semua surat dengan suatu skrip di "`/etc/ppp/ip-up.d/*`".

6.3 Server dan utilitas akses jarak jauh (SSH)

Secure SHell (SSH) adalah cara **aman** untuk terhubung melalui Internet. Versi bebas SSH yang disebut OpenSSH tersedia sebagai paket `openssh-client` dan `openssh-server` di Debian.

Untuk pengguna, ssh(1) berfungsi sebagai telnet(1) yang lebih cerdas dan lebih aman. Tidak seperti perintah `telnet`, perintah `ssh` tidak berhenti pada karakter escape `telnet` (baku awal CTRL-]).

Meskipun `shellinabox` bukan program SSH, itu terdaftar di sini sebagai alternatif yang menarik untuk akses terminal jarak jauh.

Lihat juga Bagian 7.9 untuk menyambung ke program klien X jarak jauh.



Perhatian
Lihat Bagian 4.6.3 bila SSH Anda dapat diakses dari Internet.

Tip
Silakan gunakan screen(1) untuk mengaktifkan proses shell jarak jauh yang tetap bertahan saat koneksi terputus (lihat Bagian 9.1.2).

paket	popcon	ukuran	perkakas	deskripsi
openssh-client	V:682, I:997	3611	ssh(1)	Klien shell aman
openssh-server	V:774, I:820	3615	sshd(8)	Server shell aman
ssh-askpass	I:15	103	ssh-askpass(1)	meminta pengguna untuk frasa sandi bagi ssh-add (X polos)
ssh-askpass-gnome	V:0.5, I:2.9	46	ssh-askpass-gnome(1)	meminta pengguna untuk frase sandi bagi ssh-add (GNOME)
ssh-askpass-fullscreen	V:0.09, I:0.40	41	ssh-askpass-fullscreen(1)	meminta pengguna untuk frase sandi bagi ssh-add (GNOME) dengan pernik ekstra
shellinabox	V:0.61, I:0.94	528	shellinaboxd(1)	server web untuk emulator terminal VT100 yang dapat diakses peramban

Tabel 6.7: Daftar server dan utilitas akses jarak jauh

6.3.1 Dasar-dasar SSH

Daemon OpenSSH SSH hanya mendukung protokol SSH 2.

Silakan baca "`/usr/share/doc/openssh-client/README.Debian.gz`", ssh(1), sshd(8), ssh-keygen(1), ssh-add(1), dan ssh-agent(1).

Awas
"`/etc/ssh/sshd_not_to_be_run`" tidak boleh ada jika seseorang ingin menjalankan server OpenSSH. Jangan aktifkan autentikasi berbasis rhost (`HostbasedAuthentication` dalam `/etc/ssh/sshd_config`).

berkas konfigurasi	deskripsi berkas konfigurasi
`/etc/ssh/ssh_config`	Baku klien SSH, lihat ssh_config(5)
`/etc/ssh/sshd_config`	Baku server SSH, lihat sshd_config(5)
`~/.ssh/authorized_keys`	kunci SSH publik baku yang digunakan klien untuk menyambung ke akun ini pada server SSH ini
`~/.ssh/id_rsa`	kunci rahasia SSH-2 RSA dari pengguna
`~/.ssh/id_`*`key-type-name`*	kunci *nama-tipe-kunci* SSH-2 rahasia seperti `ecdsa`, `ed25519`, ... dari pengguna

Tabel 6.8: Daftar berkas konfigurasi SSH

Berikut ini memulai koneksi ssh(1) dari klien.

6.3.2 Nama pengguna di host jarak jauh

Jika Anda menggunakan nama pengguna yang sama pada host lokal dan jarak jauh, Anda dapat menghilangkan mengetik "`namapengguna@`".

Bahkan jika Anda menggunakan nama pengguna yang berbeda pada host lokal dan jarak jauh, Anda dapat menghilangkannya menggunakan "`~/.ssh/config`". Untuk layanan Debian Salsa dengan nama akun "`foo-guest`", Anda mengatur "`~/.ssh/config`" untuk memuat hal-hal berikut.

```
Host salsa.debian.org people.debian.org
User foo-guest
```

perintah	deskripsi
`ssh username@hostname.domain.ext`	terhubung dengan mode baku
`ssh -v username@hostname.domain.ext`	terhubung dengan mode baku dengan pesan debugging
`ssh -o PreferredAuthentications=password username@hostname.domain.ext`	memaksa untuk menggunakan kata sandi dengan SSH versi 2
`ssh -t username@hostname.domain.ext passwd`	menjalankan program `passwd` untuk memperbarui kata sandi pada host jarak jauh

Tabel 6.9: Daftar contoh awal mula klien SSH

6.3.3 Menyambungkan tanpa kata sandi jarak jauh

Seseorang dapat menghindari harus mengingat kata sandi untuk sistem jarak jauh dengan menggunakan "`PubkeyAuthent`
(protokol SSH-2).

Pada sistem jarak jauh, atur entri yang sesuai, "`PubkeyAuthentication yes`", dalam "`/etc/ssh/sshd_config`".

Hasilkan kunci otentikasi secara lokal dan install kunci publik pada sistem jarak jauh dengan yang berikut ini.

```
$ ssh-keygen -t rsa
$ cat .ssh/id_rsa.pub | ssh user1@remote "cat - >>.ssh/authorized_keys"
```

Anda dapat menambahkan opsi ke entri dalam "`~/.ssh/authorized_keys`" untuk membatasi host dan menjalankan perintah tertentu. Lihat sshd(8) "BERKAS FORMAT AUTHORIZED_KEYS".

6.3.4 Berurusan dengan klien SSH alien

Ada beberapa klien SSH gratis yang tersedia untuk platform lain.

lingkungan	program SSH bebas
Windows	puTTY (PuTTY: klien SSH dan Telnet gratis) (GPL)
Windows (cygwin)	SSH di cygwin (Cygwin: Dapatkan sensasi Linux - di Windows) (GPL)
Mac OS X	OpenSSH; menggunakan `ssh` di aplikasi Terminal (GPL)

Tabel 6.10: Daftar klien SSH gratis untuk platform lain

6.3.5 Menyiapkan ssh-agent

Lebih aman untuk melindungi kunci rahasia otentikasi SSH Anda dengan frasa sandi. Jika frasa sandi tidak diatur, gunakan "`ssh-keygen -p`" untuk mengaturnya.

Tempatkan kunci SSH publik Anda (misalnya. "`~/.ssh/id_rsa.pub`") ke dalam "`~/.ssh/authorized_keys`" pada host jarak jauh menggunakan koneksi berbasis kata sandi ke host jarak jauh seperti yang dijelaskan di atas.

```
$ ssh-agent bash
$ ssh-add ~/.ssh/id_rsa
Enter passphrase for /home/username/.ssh/id_rsa:
Identity added: /home/username/.ssh/id_rsa (/home/username/.ssh/id_rsa)
```

Tidak ada kata sandi jarak jauh yang diperlukan dari sini untuk perintah berikutnya.

```
$ scp foo username@remote.host:foo
```

Tekan ^D untuk mengakhiri sesi ssh-agent.

Untuk server X, skrip awal mula Debian normal mengeksekusi `ssh-agent` sebagai proses induk. Jadi Anda hanya perlu mengeksekusi `ssh-add` sekali. Untuk lebih lanjut, baca ssh-agent(1) dan ssh-add(1).

6.3.6 Mengirim surat dari host jarak jauh

Jika Anda memiliki akun shell SSH di server dengan pengaturan DNS yang tepat, Anda dapat mengirim surat yang dihasilkan di workstation Anda sebagai surel yang benar-benar dikirim dari server jarak jauh.

```
$ ssh username@example.org /usr/sbin/sendmail -bm -ti -f "username@example.org" < mail_data ↩
    .txt
```

6.3.7 Penerusan port untuk tunneling SMTP/POP3

Untuk menjalin pipa untuk menyambung ke port 25 `remote-server` dari port 4025 dari `localhost`, dan ke port 110 `remote-server` dari port 4110 `localhost` melalui `ssh`, jalankan pada host lokal sebagai berikut.

```
# ssh -q -L 4025:remote-server:25 4110:remote-server:110 username@remote-server
```

Ini adalah cara yang aman untuk membuat koneksi ke server SMTP/POP3 melalui Internet. Atur entri "`AllowTcpForwardin` menjadi "`yes`" di "`/etc/ssh/sshd_config`" pada host jarak jauh.

6.3.8 Cara mematikan sistem jarak jauh di SSH

Anda perlu melindungi proses melakukan "`shutdown -h now`" (lihat Bagian 1.1.8) dari penghentian SSH menggunakan perintah at(1) (lihat Bagian 9.4.13) dengan berikut ini.

```
# echo "shutdown -h now" | at now
```

Menjalankan sesi "`shutdown-h now`" dalam screen(1) (lihat Bagian 9.1.2) adalah cara lain untuk melakukan hal yang sama.

6.3.9 Pemecahan masalah SSH

Jika Anda memiliki masalah, periksa izin berkas konfigurasi dan jalankan `ssh` dengan opsi "`-v`".

Gunakan opsi "`-p`" jika Anda adalah root dan memiliki masalah dengan firewall; ini menghindari penggunaan port server 1 - 1023.

Jika koneksi `ssh` ke situs jarak jauh tiba-tiba berhenti bekerja, itu mungkin hasil dari coba-coba oleh sysadmin, kemungkinan besar perubahan dalam "`host_key`" selama pemeliharaan sistem. Setelah memastikan ini adalah kasusnya dan tidak ada yang mencoba memalsukan host jarak jauh dengan beberapa peretasan pintar, seseorang dapat memperoleh kembali koneksi dengan menghapus entri "`host_key`" dari "`~/.ssh/known_hosts`" pada host lokal.

6.4 Server cetak dan utilitas

Dalam sistem mirip-Unix lama, Line printer daemon (lpd) BSD adalah standar dan format cetak standar dari perangkat lunak bebas klasik adalah PostScript (PS). Beberapa sistem filter digunakan bersama dengan Ghostscript untuk memungkinkan pencetakan ke printer non-PostScript. Lihat Bagian 11.4.1.

Dalam sistem Debian modern, Common UNIX Printing System (CUPS) adalah standar de facto dan format cetak standar dari perangkat lunak bebas modern adalah Portable Document Format (PDF).

CUPS menggunakan Internet Printing Protocol (IPP). IPP adalah standar de facto lintas platform untuk pencetakan jarak jauh dengan kemampuan komunikasi dua arah.

Berkat fitur konversi otomatis yang bergantung pada format berkas dari sistem CUPS, cukup mengumpan data apa pun ke perintah `lpr` harus menghasilkan keluaran cetak yang diharapkan. (Dalam CUPS, `lpr` dapat diaktifkan dengan memasang paket `cups-bsd`.)

Sistem Debian memiliki beberapa paket penting untuk server dan utilitas cetak.

paket	popcon	ukuran	port	deskripsi
`lpr`	V:1.9, I:2.2	378	printer (515)	BSD lpr/lpd (Daemon line printer)
`cups`	V:144, I:447	1088	IPP (631)	Server Pencetakan Internet CUPS
`cups-client`	V:152, I:459	429	, ,	Perintah printer Sistem V untuk CUPS: lp(1), lpstat(1), lpoptions(1), cancel(1), lpmove(8), lpinfo(8), lpadmin(8), …
`cups-bsd`	V:34, I:182	131	, ,	Perintah printer BSD untuk CUPS: lpr(1), lpq(1), lprm(1), lpc(8)
`printer-driver-gutenprint`	V:11, I:52	1117	Tidak berlaku	pengandar pencetak untuk CUPS

Tabel 6.11: Daftar server cetak dan utilitas

Tip
Anda dapat mengkonfigurasi sistem CUPS dengan mengarahkan peramban web Anda ke ”http://localhost:631/” .

6.5 Server aplikasi jaringan lainnya

Berikut adalah server aplikasi jaringan lainnya.

Common Internet File System Protocol (CIFS) adalah protokol yang sama dengan Server Message Block (SMB) dan digunakan secara luas oleh Microsoft Windows.

Tip
Lihat Bagian 4.5.2 untuk integrasi sistem server.

Tip
Resolusi nama host biasanya disediakan oleh server DNS. Untuk alamat IP host yang ditetapkan secara dinamis oleh DHCP, DNS Dinamis dapat diatur untuk resolusi nama host menggunakan `bind9` dan `kea` seperti yang dijelaskan di halaman DDNS di wiki Debian.

paket	popcon	ukuran	protokol	deskripsi
telnetd	V:0.3, I:1.6	56	TELNET	Server TELNET
nfs-kernel-server	V:47, I:54	874	NFS	Berbagi berkas Unix
samba	V:110, I:123	5182	SMB	Berbagi berkas dan pencetak Windows
netatalk	V:0.62, I:0.85	933	ATP	Berbagi berkas dan pencetak Apple/Mac (AppleTalk)
proftpd-basic	V:3.3, I:8.4	452	FTP	Pengunduhan berkas umum
apache2	V:175, I:211	587	HTTP	Server web umum
squid	V:9.0, I:9.4	9583	, ,	Server proksi web umum
bind9	V:33, I:35	904	DNS	Alamat IP untuk host lain
kea	I:0.51	245	DHCP	Alamat IP klien itu sendiri

Tabel 6.12: Daftar server aplikasi jaringan lainnya

Tip
Penggunaan server proksi seperti `squid` jauh lebih efisien untuk menghemat bandwidth daripada penggunaan server cermin lokal dengan isi arsip Debian lengkap.

6.6 Klien aplikasi jaringan lainnya

Berikut adalah klien aplikasi jaringan lainnya.

6.7 Diagnosis daemon sistem

Program `telnet` memungkinkan koneksi manual ke daemon sistem dan diagnosisnya.

Untuk menguji layanan POP3 polos, coba yang berikut ini

```
$ telnet mail.ispname.net pop3
```

Untuk menguji layanan POP3 yang memakai TLS/SSL oleh beberapa ISP, Anda memerlukan klien `telnet` yang mendukung TLS/SSL oleh paket `telnet-ssl` atau `openssl`.

```
$ telnet -z ssl pop.gmail.com 995
```

```
$ openssl s_client -connect pop.gmail.com:995
```

RFC berikut memberikan pengetahuan yang diperlukan untuk setiap daemon sistem.

Penggunaan port dijelaskan dalam ”`/etc/services`”.

paket	popcon	ukuran	protokol	deskripsi
netcat-traditional	V:55, I:906	139	TCP/IP	pisau tentara Swiss untuk TCP/IP
netcat-openbsd	V:22, I:119	105	TCP/IP	Perkakas TCP/IP serba guna dengan dukungan IPv6, proksi, dan soket Unix
openssl	V:846, I:996	2564	SSL	biner dan alat kriptografi terkait Secure Socket Layer (SSL)
stunnel4	V:6.6, I:9.9	24	, ,	pembungkus SSL universal
telnet	V:10, I:190	56	TELNET	Klien TELNET
nfs-common	V:147, I:191	1158	NFS	Berbagi berkas Unix
smbclient	V:25, I:206	2117	SMB	klien berbagi berkas dan pencetak MS Windows
cifs-utils	V:32, I:118	368	, ,	perintah mount dan umount untuk berkas MS Windows jarak jauh
wget	V:181, I:982	3784	HTTP dan FTP	pengunduh web
curl	V:260, I:727	520	, ,	, ,
transmission-gtk	V:9, I:143	6259	BitTorrent	Klien BitTorrent (GTK)
transmission-qt	V:0.6, I:2.6	6213	, ,	Klien BitTorrent (Qt)
ktorrent	V:1.3, I:5.0	5031	, ,	Klien BitTorrent (Qt)
qbittorrent	V:8, I:23	15378	, ,	Klien BitTorrent (Qt)
bind9-host	V:124, I:940	136	DNS	host(1) dari bind9, ”`Priority: standard`”
dnsutils	V:5, I:146	23	, ,	dig(1) dari bind, ”`Priority: standard`”
ldap-utils	V:11, I:57	790	LDAP	mendapatkan data dari server LDAP

Tabel 6.13: Daftar klien aplikasi jaringan

RFC	deskripsi
rfc1939 dan rfc2449	Layanan POP3
rfc3501	Layanan IMAP4
rfc2821 (rfc821)	Layanan SMTP
rfc2822 (rfc822)	Format berkas surel
rfc2045	Multipurpose Internet Mail Extensions (MIME)
rfc819	Layanan DNS
rfc2616	Layanan HTTP
rfc2396	Definisi URI

Tabel 6.14: Daftar RFC populer

Bab 7

Sistem GUI

7.1 Lingkungan desktop GUI

Ada beberapa pilihan untuk lingkungan desktop GUI berfitur lengkap pada sistem Debian.

paket tugas	popcon	ukuran	deskripsi
task-gnome-desktop	I:192	9	Lingkungan desktop GNOME
task-xfce-desktop	I:88	9	Lingkungan desktop Xfce
task-kde-desktop	I:94	6	Lingkungan desktop Plasma KDE
task-mate-desktop	I:30	9	Lingkungan desktop MATE
task-cinnamon-desktop	I:35	9	Lingkungan desktop Cinnamon
task-lxde-desktop	I:19	9	Lingkungan desktop LXDE
task-lxqt-desktop	I:16	9	Lingkungan desktop LXQt
task-gnome-flashback-desktop	I:9.6	6	Lingkungan desktop GNOME Flashback

Tabel 7.1: Daftar lingkungan desktop

Tip
Paket dependensi yang dipilih oleh metapackage tugas mungkin tidak sinkron dengan status transisi paket terbaru di bawah lingkungan Debian `unstable`/`testing`. Untuk `task-gnome-desktop`, Anda mungkin perlu menyesuaikan pilihan paket sebagai berikut:

- Mulailah aptitude(8) sebagai `sudo aptitude -u`.
- Pindahkan kursor ke ”Tugas” dan tekan ”Enter”.
- Pindahkan kursor ke ”Pengguna akhir” tekan ”Enter”.
- Pindahkan kursor ke ”GNOME” tekan ”Enter”.
- Pindahkan kursor ke `task-gnome-desktop` dan tekan ”Enter”.
- Pindahkan kursor ke ”Depends” dan tekan ”m” (dipilih secara manual).
- Pindahkan kursor ke ”Recommends” dan tekan ”m” (dipilih secara manual).
- Pindahkan kursor ke ”`task-gnome-desktop` dan tekan ”-”. (keluarkan)
- Sesuaikan paket yang dipilih sambil mengeluarkan yang bermasalah yang menyebabkan konflik paket.
- Tekan ”g” untuk mulai memasang.

Bab ini akan fokus terutama pada lingkungan desktop default Debian: `task-gnome-desktop` yang menawarkan GNOME pada wayland.

7.2 Protokol komunikasi GUI

Protokol komunikasi GUI yang digunakan pada desktop GNOME bisa berupa:

- Wayland (protokol server tampilan) (native)
- Protokol inti Sistem X Window (melalui `xwayland`)

Silakan periksa situs freedesktop.org untuk bagaimana arsitektur Wayland berbeda dari arsitektur X Window.

Dari perspektif pengguna, perbedaan dapat diringkas secara bahasa sehari-hari sebagai:

- Wayland adalah protokol komunikasi GUI host-yang-sama: baru, lebih sederhana, lebih cepat, tidak ada biner setuid root
- X Window adalah protokol komunikasi GUI berkemampuan jaringan: tradisional, kompleks, lebih lambat, biner setuid root

Untuk aplikasi yang menggunakan protokol Wayland, akses ke konten tampilan mereka dari host jarak jauh didukung oleh VNC atau RDP. Lihat Bagian 7.8

Server X modern memiliki Ekstensi Memori Bersama MIT dan berkomunikasi dengan klien X lokal mereka menggunakan memori bersama lokal. Ini melewati jaringan transparan Xlib saluran komunikasi inter proses dan meningkatkan kinerja. Situasi ini adalah latar belakang menciptakan Wayland sebagai protokol komunikasi GUI lokal saja.

Dengan menggunakan program `xeyes` yang dimulai dari terminal GNOME, Anda dapat memeriksa protokol komunikasi GUI yang digunakan oleh setiap aplikasi GUI.

```
$ xeyes
```

- Jika kursor tetikus ada pada aplikasi seperti ”terminal GNOME” yang menggunakan protokol server tampilan Wayland, mata tidak bergerak dengan kursor tetikus.
- Jika kursor tetikus berada pada aplikasi seperti ”xterm” yang menggunakan protokol inti X Window System, mata bergerak dengan kursor tetikus mengekspos sifat arsitektur X Window yang tidak begitu terisolasi.

Pada April 2021, banyak aplikasi GUI populer seperti gnome dan aplikasi LibreOffice (LO) telah bermigrasi ke protokol server tampilan Wayland. Saya melihat `xterm`, `gitk`, `chromium`, `firefox`, `gimp`, `dia`, dan aplikasi-aplikasi KDE masih menggunakan protokol inti X Window System.

Catatan
Untuk xwayland di Wayland atau X Window System asli, berkas konfigurasi server X lama ”`/etc/X11/xorg.conf`” seharusnya tidak ada pada sistem. Perangkat grafis dan masukan sekarang dikonfigurasi oleh kernel dengan DRM, KMS, dan udev. Server X asli telah ditulis ulang untuk menggunakannya. Lihat ”dukungan mode video baku modedb” dalam dokumentasi kernel Linux.

7.3 Infrastruktur GUI

Berikut adalah paket infrastruktur GUI yang terkenal untuk GNOME di lingkungan Wayland.

paket	popcon	ukuran paket	deskripsi
`mutter`	V:1, I:22	216	Manajer jendela mutter GNOME **[auto]**
`xwayland`	V:256, I:335	2652	Server X yang berjalan di atas wayland **[auto]**
`gnome-remote-desktop`	V:147, I:241	2743	Daemon desktop jarak jauh untuk GNOME menggunakan PipeWire **[auto]**
`gnome-tweaks`	V:17, I:232	1145	Setelan konfigurasi tingkat lanjut untuk GNOME
`gnome-shell-extension-prefs`	V:8, I:117	73	Perkakas untuk memfungsikan / menonaktifkan ekstensi GNOME Shell

Tabel 7.2: Daftar paket infrastruktur GUI yang terkenal

Di sini, ”**[auto]**” berarti bahwa paket-paket ini secara otomatis dipasang ketika `task-gnome-desktop` dipasang.

Tip
`gnome-tweak` adalah utilitas konfigurasi yang sangat diperlukan. Misalnya:

- Anda dapat memaksa ”Penguatan-Berlebih” volume suara dari ”Umum”.
- Anda dapat memaksa ”Caps” untuk menjadi ”Esc” dari ”Papan Ketik & Tetikus” -> ”Papan Ketik” -> ”Opsi Tata Letak Tambahan”.

Tip
Fitur rinci dari lingkungan desktop GNOME dapat dikonfigurasi dengan utilitas yang dimulai dengan mengetikkan ”`settings`”, ”`tweaks`”, atau ”`extensions`” setelah menekan tombol `Super`.

7.4 Aplikasi GUI

Banyak aplikasi GUI yang berguna tersedia di Debian sekarang. Memasang paket perangkat lunak seperti `scribus` (KDE) di lingkungan desktop GNOME cukup dapat diterima karena fungsi yang sesuai tidak tersedia di bawah lingkungan desktop GNOME. Tetapi memasang terlalu banyak paket dengan fungsi duplikat dapat mengacaukan sistem Anda.

Berikut adalah daftar aplikasi GUI yang menarik perhatian saya.

7.5 Direktori pengguna

Nama baku untuk direktori pengguna seperti "`~/Desktop`", "`~/Dokumen`", ..., yang digunakan oleh lingkungan Desktop tergantung pada lokal yang digunakan untuk instalasi sistem. Anda dapat mengatur ulang nama-nama tersebut ke dalam bahasa Inggris:

```
$ LANGUAGE=C xdg-user-dirs-update --force
```

Kemudian Anda secara manual memindahkan semua data ke direktori yang lebih baru. Lihat xdg-user-dirs-update(1).

Anda juga dapat mengaturnya ke nama apa pun dengan menyunting "`~/.config/user-dirs.dirs`". Lihat user-dirs.dirs(5).

7.6 Fonta

Banyak fonta yang dapat diskalakan yang berguna tersedia untuk pengguna di Debian. Kekhawatiran pengguna adalah bagaimana menghindari redundansi dan cara mengkonfigurasi bagian dari fonta yang dipasang untuk dinonaktifkan. Jika tidak, pilihan fonta yang tidak berguna dapat mengacaukan menu aplikasi GUI Anda.

Sistem Debian menggunakan pustaka FreeType 2.0 untuk merasterkan banyak format fonta yang dapat diskalakan untuk layar dan cetak:

- Fonta Tipe 1 (PostScript) yang menggunakan kurva Bézier kubik (format hampir usang)
- Fonta TrueType yang menggunakan kurva Bézier kuadrat (format pilihan yang baik)
- Fonta OpenType yang menggunakan kurva Bézier kubik (format pilihan terbaik)

7.6.1 Fonta dasar

Tabel berikut disusun dengan harapan dapat membantu pengguna untuk memilih fonta yang dapat diskalakan yang sesuai dengan pemahaman yang jelas tentang kompatibilitas metrik dan cakupan glyph. Sebagian besar fonta mencakup semua fonta glyph karakter Latin, Yunani, dan Sirilik. Pilihan akhir fonta yang diaktifkan juga dapat dipengaruhi oleh estetika Anda. Fonta ini dapat digunakan untuk layar atau untuk pencetakan kertas.

Sini:

- "MCM" adalah singkatan dari "metric compatible with fonts provided by Microsoft (metrik yang kompatibel dengan fonta yang disediakan oleh Microsoft)"
- "MCMATC" adalah singkatan dari "metric compatible with fonts provided by Microsoft: (metrik yang kompatibel dengan fonta yang disediakan oleh Microsoft:) Arial, Times New Roman, Courier New"
- "MCAHTC" adalah singkatan dari "metric compatible with fonts provided by (metrik yang kompatibel dengan fonta yang disediakan oleh) Adobe: Helvetica, Times, Courier"

paket	popcon	ukuran paket	jenis	deskripsi
evolution	V:27, I:229	489	GNOME	Manajemen informasi pribadi (groupware dan surel)
thunderbird	V:39, I:102	303329	GTK	Klien surel (Mozilla Thunderbird)
kontact	V:1, I:10	2360	KDE	Manajemen informasi pribadi (groupware dan surel)
libreoffice-writer	V:105, I:422	34232	LO	pengolah kata
abiword	V:0.8, I:4.7	3596	GNOME	pengolah kata
calligrawords	V:0.2, I:2.7	7144	KDE	pengolah kata
scribus	V:1, I:12	32942	KDE	penyunting penerbitan desktop untuk menyunting berkas PDF
glabels	V:0.3, I:2.5	1283	GNOME	editor label
libreoffice-calc	V:101, I:419	29620	LO	lembar kerja
gnumeric	V:3, I:10	9945	GNOME	lembar kerja
calligrasheets	V:0.1, I:1.7	14014	KDE	lembar kerja
libreoffice-impress	V:88, I:417	2499	LO	presentasi
calligrastage	V:0.1, I:1.6	6237	KDE	presentasi
libreoffice-base	V:21, I:67	5130	LO	manajemen basis data
kexi	V:0.05, I:0.80	7565	KDE	manajemen basis data
libreoffice-draw	V:89, I:418	11418	LO	penyunting grafik vektor (draw)
inkscape	V:12, I:76	113353	GNOME	penyunting grafik vektor (draw)
karbon	V:0.1, I:2.1	4029	KDE	penyunting grafik vektor (draw)
dia	V:1, I:16	3846	GTK	editor diagram dan diagram alur
gimp	V:34, I:211	33315	GTK	penyunting grafik bitmap (paint)
shotwell	V:14, I:246	6334	GTK	pengorganisasi foto digital
digikam	V:1.5, I:8.3	333	KDE	pengorganisasi foto digital
darktable	V:4, I:11	36092	GTK	lighttable dan darkroom untuk para fotografer
gnome-video-trimmer	V:0.1, I:1.0	1665	GNOME	pemotongan tanpa kehilangan atas berkas video
kdenlive	V:3, I:21	22134	KDE	penyunting video non linier
shotcut	V:1.0, I:7.5	8270	Qt-app	penyunting video non linier
openshot-qt	V:0.9, I:8.8	196007	Qt-app	penyunting video non linier
flowblade	V:0.2, I:2.2	42213	GTK	penyunting video non linier
planner	V:0.2, I:4.2	1400	GNOME	manajemen proyek
calligraplan	V:0.2, I:2.9	23545	KDE	manajemen proyek
gnucash	V:2.0, I:6.7	29434	GNOME	akuntansi pribadi
homebank	V:0.3, I:1.5	3194	GTK	akuntansi pribadi
kmymoney	V:0.4, I:1.9	19297	KDE	akuntansi pribadi
frescobaldi	V:0.3, I:1.5	11102	Qt-app	LilyPond penyunting teks lembar musik
librecad	V:1, I:13	9164	Qt-app	sistem computer-aided design (CAD) (2D)
freecad	V:1, I:20	112	Qt-app	sistem computer-aided design (CAD) (3D)
kicad	V:3, I:14	224155	GTK	perangkat lunak desain skematik dan PCB elektronik
xsane	V:9, I:128	1512	GTK	frontend pemindai
libreoffice-math	V:84, I:420	1982	LO	penyunting persamaan/rumus matematika
calibre	V:7, I:24	71216	KDE	konverter e-book dan manajemen perpustakaan
fbreader	V:0.7, I:5.4	3827	GTK	pembaca e-book
evince	V:76, I:284	953	GNOME	penampil dokumen(pdf)
okular	V:42, I:130	4544	KDE	penampil dokumen(pdf)
x11-apps	V:31, I:443	2461	app-X murni	xeyes(1), dsb.
x11-utils	V:228, I:555	651	app-X murni	xev(1), xwininfo(1), dsb.

Tabel 7.3: Daftar aplikasi GUI yang terkenal

paket	popcon	ukuran	sans	serif	mono	catatan pada fonta
fonts-cantarell	V:174, I:288	224	59	-	-	Cantarell (GNOME 3, tampilan)
fonts-noto	I:150	30	61	63	40	Fonta Noto (Google, multibahasa dengan CJK)
fonts-dejavu	I:385	34	58	68	40	DejaVu (GNOME 2, MCM:Verdana, Bitstream Vera yang diperloas)
fonts-liberation2	V:40, I:156	15	56	60	40	Fonta Liberation bagi LibreOffice (Red Hat, MCMATC)
fonts-croscore	V:20, I:36	5260	56	60	40	Chrome OS: Arimo, Tinos, dan Cousine (Google, MCMATC)
fonts-crosextra-carlito	V:20, I:90	2696	57	-	-	Chrome OS: Carlito (Google, MCM:Calibri)
fonts-crosextra-caladea	V:12, I:83	347	-	55	-	Chrome OS: Caladea (Google, MCM:Cambria) (hanya bahasa Latin)
fonts-freefont-ttf	V:78, I:203	14460	57	59	40	GNU FreeFont (URW Nimbus yang diperluas)
fonts-quicksand	V:210, I:452	392	56	-	-	Debian task-desktop, Quicksand (tampilan, hanya Latin)
fonts-hack	V:34, I:136	2509	-	-	40 P	Jenis huruf yang dirancang untuk kode sumber Hack (Facebook)
fonts-sil-gentiumplus	V:0, I:28	14345	-	54	-	Gentium SIL
fonts-sil-charis	V:1, I:28	6704	-	59	-	Charis SIL
fonts-urw-base35	V:187, I:534	15558	56	60	40	URW Nimbus (Nimbus Sans, Roman No. 9 L, Mono L, MCAHTC)
fonts-ubuntu	V:2.2, I:4.7	4339	58	-	33 P	Fonta Ubuntu (tampilan)
fonts-terminus	V:0.3, I:3.9	451	-	-	33	Cool retro terminal fonts
ttf-mscorefonts-installer	V:1, I:40	85	56?	60	40	Pengunduh fonta non-bebas Microsoft (lihat di bawah)

Tabel 7.4: Daftar fonta TrueType dan OpenType yang terkenal

- Angka dalam kolom tipe fonta adalah singkatan dari lebar ”M” relatif kasar untuk fonta dengan ukuran titik yang sama.
- ”P” dalam kolom tipe fonta mono adalah singkatan dari kegunaannya untuk pemrograman yang jelas dapat membedakan ”0”/”O” dan ”1”/”I”/”l”.
- Paket `ttf-mscorefonts-installer` mengunduh ”Fonta Inti untuk Web” Microsoft dan memasang Arial, Times New Roman, Courier New, Verdana, ... Data fonta yang dipasang ini adalah data yang tidak bebas.

Banyak fonta Latin gratis memiliki garis keturunan mereka ditelusuri ke keluarga URW Nimbus atau Bitstream Vera.

Tip
Jika lokal Anda membutuhkan fonta-fonta yang tidak tercakup dengan baik oleh fonta-fonta di atas, silakan gunakan aptitude untuk memeriksa di bawah paket tugas yang tercantum di bawah ”Tugas” -> ”Pelokalan”. Paket fonta yang terdaftar sebagai ”Depends:” atau ”Recommends:” dalam paket-paket tugas pelokalan adalah kandidat utama.

7.6.2 Rasterisasi fonta

Debian menggunakan FreeType untuk merasterkan fonta. Infrastruktur pilihan fontanya disediakan oleh pustaka konfigurasi fonta Fontconfig.

paket	popcon	ukuran	deskripsi
`libfreetype6`	V:589, I:997	1050	Pustaka rasterisasi fonta FreeType
`libfontconfig1`	V:577, I:808	344	Pustaka konfigurasi fonta Fontconfig
`fontconfig`	V:471, I:693	415	`fc-*`: perintah-perintah CLI untuk Fontconfig
`font-manager`	V:2.3, I:6.6	1116	Manajer Fonta: perintah GUI untuk Fontconfig
`nautilus-font-manager`	V:0.07, I:0.38	38	Ekstensi Nautilus untuk Manajer Fonta

Tabel 7.5: Daftar lingkungan fonta terkenal dan paket-paket terkait

Tip
Beberapa paket fonta seperti `fonts-noto*` memasang terlalu banyak fonta. Anda mungkin juga ingin mempertahankan beberapa paket fonta terpasang tetapi dinonaktifkan di bawah situasi penggunaan normal. Beberapa gliph diharapkan bagi beberapa titik kode Unicode karena penyatuan Han dan gliph yang tidak diinginkan dapat dipilih oleh pustaka Fontconfig yang tidak dikonfigurasi. Salah satu kasus yang paling menjengkelkan adalah ”U+3001 IDEOGRAPHIC COMMA” dan ”U+3002 IDEOGRAPHIC FULL STOP” di antara negara-negara CJK. Anda dapat menghindari situasi bermasalah ini dengan mudah dengan mengonfigurasi ketersediaan fonta menggunakan GUI Manajer Fonta (font-manager).

Anda juga dapat melihat daftar status konfigurasi fonta dari baris perintah.

- ”`fc-match(1)`” untuk baku fonta fontconfig
- ”`fc-list(1)`” untuk fonta fontconfig yang tersedia

Anda dapat mengonfigurasi status konfigurasi fonta dari penyunting teks tetapi ini tidak sepele. Lihat fonts.conf(5).

7.7 Sandbox

Banyak sebagian besar aplikasi GUI di Linux tersedia dalam format biner dari sumber non-Debian.

- AppImage -- Aplikasi Linux yang berjalan di mana saja
- FLATHUB -- Aplikasi untuk Linux, di sini
- snapcraft -- Toko aplikasi untuk Linux



Awas
Biner dari situs-situs ini mungkin termasuk paket perangkat lunak non-bebas proprietari.

Ada beberapa raison d'être untuk distribusi format biner ini bagi penggemar Perangkat Lunak Bebas yang menggunakan Debian karena ini dapat mengakomodasi kumpulan pustaka bersih yang digunakan untuk setiap aplikasi oleh pengembang hulu masing-masing independen dari yang disediakan oleh Debian.

Risiko melekat saat menjalankan biner eksternal dapat dikurangi dengan menggunakan lingkungan sandbox yang memanfaatkan fitur keamanan Linux modern (lihat Bagian 4.7.5).

- Untuk biner dari AppImage dan beberapa situs hulu, jalankan di firejail dengan konfigurasi manual.
- Untuk biner dari FLATHUB, jalankan di Flatpak . (Tidak diperlukan konfigurasi manual.)
- Untuk biner dari snapcraft, jalankan di Snap . (Tidak diperlukan konfigurasi manual. Kompatibel dengan program daemon.)

Paket `xdg-desktop-portal` menyediakan API standar untuk fitur desktop umum. Lihat xdg-desktop-portal (flatpak) dan xdg-desktop-portal (snap) .

paket	popcon	ukuran	deskripsi
`flatpak`	V:104, I:115	9293	Kerangka kerja penggelaran aplikasi Flatpak untuk aplikasi desktop
`gnome-software-plugin-flatpak`	V:29, I:41	291	Dukungan Flatpak untuk GNOME Perangkat Lunak
`snapd`	V:58, I:62	62738	Daemon dan perkakas yang memfungsikan paket snap
`gnome-software-plugin-snap`	V:1.4, I:2.2	144	Dukungan Snap untuk GNOME Perangkat Lunak
`xdg-desktop-portal`	V:373, I:439	2291	portal integrasi desktop untuk Flatpak dan Snap
`xdg-desktop-portal-gtk`	V:337, I:438	715	backend xdg-desktop-portal untuk gtk (GNOME)
`xdg-desktop-portal-kde`	V:81, I:112	3333	backend xdg-desktop-portal untuk Qt (KDE)
`xdg-desktop-portal-wlr`	V:2.4, I:6.6	159	backend xdg-desktop-portal untuk wlroots (Wayland)
`firejail`	V:1.2, I:4.4	1827	sebuah program sandbox keamanan SUID firejail untuk digunakan dengan AppImage

Tabel 7.6: Daftar lingkungan sandbox terkenal dan paket terkait

Teknologi lingkungan sandbox ini sangat mirip dengan aplikasi di OS ponsel pintar tempat aplikasi dijalankan di bawah akses sumber daya yang dikendalikan.

Beberapa aplikasi GUI besar seperti peramban web pada Debian juga menggunakan teknologi lingkungan sandbox secara internal untuk membuatnya lebih aman.

7.8 Desktop jarak jauh

paket	popcon	ukuran	protokol	deskripsi
gnome-remote-desktop	V:14, I:241	2743	RDP	Server Desktop Jarak Jauh GNOME
xrdp	V:28, I:31	4757	RDP	xrdp, server Remote Desktop Protocol (RDP)
x11vnc	V:9, I:46	1863	RFB (VNC)	x11vnc, server Remote Framebuffer Protocol (VNC)
tigervnc-standalone-server	V:5, I:15	3124	RFB (VNC)	TigerVNC, server Remote Framebuffer Protocol (VNC)
gnome-connections	V:6, I:149	1726	RDP, RFB (VNC)	Klien desktop jarak jauh GNOME
vinagre	V:1, I:21	4249	RDP, RFB (VNC), SPICE, SSH	Vinagre: Klien desktop jarak jauh GNOME
remmina	V:16, I:59	983	RDP, RFB (VNC), SPICE, SSH, ...	Remmina: Klien desktop jarak jauh GTK
krdc	V:2, I:15	4144	RDP, RFB (VNC)	KRDC: Klien desktop jarak jauh KDE
virt-viewer	V:4, I:37	1278	RFB (VNC), SPICE	GUI Virtual Machine Manager menampilkan klien OS tamu

Tabel 7.7: Daftar server akses jarak jauh yang terkenal

7.9 Sambungan server X

Ada beberapa cara untuk menyambung dari aplikasi pada host jarak jauh ke server X termasuk `xwayland` pada host lokal.

paket	popcon	ukuran	perintah	deskripsi
openssh-server	V:774, I:820	3615	`sshd` dengan opsi `X11-forwarding`	Server SSH (aman)
openssh-client	V:682, I:997	3611	`ssh -X`	Klien SSH (aman)
xauth	V:192, I:974	81	`xauth`	Utilitas berkas otoritas X
x11-xserver-utils	V:373, I:533	559	`xhost`	kontrol akses server untuk X

Tabel 7.8: Daftar metode koneksi ke server X

7.9.1 koneksi lokal server X

Akses ke server X lokal oleh aplikasi lokal yang menggunakan protokol inti X dapat terhubung secara lokal melalui soket domain UNIX lokal. Ini dapat disahkan oleh berkas otoritas yang memegang cookie akses. Lokasi berkas

otoritas diidentifikasi oleh variabel lingkungan ”$XAUTHORITY” dan tampilan X diidentifikasi oleh variabel lingkungan ”$DISPLAY”. Karena ini biasanya diatur secara otomatis, tidak ada tindakan khusus yang diperlukan, misalnya ”gitk” sebagai berikut.

```
username $ gitk
```

Catatan
Untuk xwayland, XAUTHORITY memegang nilai seperti ”/run/user/1000/.mutter-Xwaylandauth.YVSU30”.

7.9.2 Sambungan jarak jauh server X

Akses ke tampilan server X lokal dari aplikasi jarak jauh yang menggunakan protokol inti X didukung dengan menggunakan fitur penerusan X11.

- Buka gnome-terminal pada host lokal.
- Jalankan ssh(1) dengan opsi -X untuk membuat koneksi dengan situs jarak jauh sebagai berikut.

```
localname @ localhost $ ssh -q -X loginname@remotehost.domain
Password:
```

- Menjalankan suatu perintah aplikasi X, mis. ”gitk”, pada situs remote sebagai berikut.

```
loginname @ remotehost $ gitk
```

Metode ini dapat menampilkan keluaran dari klien X jarak jauh seolah-olah terhubung secara lokal melalui soket domain UNIX lokal.

Lihat Bagian 6.3 untuk SSH/SSHD.



Awas
Koneksi TCP/IP jarak jauh ke server X dinonaktifkan secara baku pada sistem Debian karena alasan keamanan. Jangan mengaktifkannya hanya dengan mengatur ”xhost +” atau dengan mengaktifkan koneksi XDMCP, jika Anda dapat menghindarinya.

7.9.3 Koneksi chroot server X

Akses ke server X oleh aplikasi yang menggunakan protokol inti X dan berjalan pada host yang sama tetapi dalam lingkungan seperti chroot di mana berkas otoritas tidak dapat diakses, dapat disahkan dengan aman dengan xhost dengan menggunakan akses berbasis pengguna, misalnya ”gitk” sebagai berikut.

```
username $ xhost + si:localuser:root ; sudo chroot /path/to
# cd /src
# gitk
# exit
username $ xhost -
```

7.10 Papanklip

Untuk menyalin teks ke papan klip, lihat Bagian 1.4.4.

Untuk menyalin grafis ke papan klip, lihat Bagian 11.6.

Beberapa perintah CLI juga dapat memanipulasi papan klip karakter (PRIMARY dan CLIPBOARD).

paket	popcon	ukuran paket	target	deskripsi
xsel	V:8, I:42	55	X	antarmuka baris perintah ke pilihan X (papan klip)
xclip	V:18, I:78	62	X	antarmuka baris perintah ke pilihan X (papan klip)
wl-clipboard	V:10, I:27	174	Wayland	`wl-copy` `wl-paste`: antarmuka baris perintah ke papan klip Wayland
gpm	V:8.3, I:9.0	524	Konsol Linux	daemon yang menangkap peristiwa tetikus di konsol Linux

Tabel 7.9: Daftar program yang terkait dengan memanipulasi papan klip karakter

Bab 8

I18N dan L10N

Multilingualisasi (M17N) atau Dukungan Bahasa Asli untuk perangkat lunak aplikasi dilakukan dalam 2 langkah.

- Internasionalisasi (I18N): Untuk membuat perangkat lunak berpotensi menangani beberapa lokal.
- Pelokalan (L10N): Untuk membuat perangkat lunak menangani lokal tertentu.

Tip
Ada 17, 18, atau 10 huruf antara ”m” dan ”n”, ”i” dan ”n”, atau ”l” dan ”n” dalam multilingualization (multibahasa), internationalization (internasionalisasi), dan localization (pelokalan) yang sesuai dengan M17N, I18N, dan L10N. Lihat Internasionalisasi dan pelokalan untuk detailnya.

8.1 Lokal

Perilaku program yang mendukung internasionalisasi dikonfigurasi oleh variabel lingkungan ”`$LANG`” untuk mendukung pelokalan. Dukungan aktual dari fitur dependen lokal oleh pustaka `libc` memerlukan pemasangan paket `locales` atau `locales-all`. Paket `locales` harus diinisialisasi dengan benar.

Jika paket `locales` dan `locales-all` tidak terpasang, dukungan fitur lokal hilang dan sistem menggunakan pesan bahasa Inggris AS dan menangani data sebagai **ASCII**. Perilaku ini adalah cara yang sama seperti ”`$LANG`” diatur oleh ”`LANG=`”, ”`LANG=C`”, atau ”`LANG=POSIX`”.

Perangkat lunak modern seperti GNOME dan KDE telah menjadi multibahasa. Mereka diinternasionalisasi dengan membuat mereka menangani data UTF-8 dan dilokalkan dengan menyediakan pesan terjemahan mereka melalui infrastruktur gettext(1). Pesan yang diterjemahkan dapat disediakan sebagai paket pelokalan terpisah.

Sistem GUI desktop Debian saat ini biasanya menetapkan lokal di bawah lingkungan GUI sebagai ”`LANG=xx_YY.UTF-8`”. Di sini, ”`xx`” adalah kode bahasa ISO 639 dan ”YY” adalah kode negara ISO 3166. Nilai-nilai ini diatur oleh dialog GUI konfigurasi desktop dan mengubah perilaku program. Lihat Bagian 1.5.2

8.1.1 Alasan untuk lokal UTF-8

Representasi paling sederhana dari data teks adalah **ASCII** yang cukup untuk bahasa Inggris dan menggunakan kurang dari 127 karakter (diwakili dengan 7 bit).

Bahkan teks bahasa Inggris polos mungkin berisi karakter non-ASCII, misalnya tanda kutip kiri dan kanan yang sedikit keriting tidak tersedia dalam ASCII.

```
b'‘b'double quoted textb'’b'' is not "double quoted ASCII"
b'‘b'single quoted textb'’b'' is not 'single quoted ASCII'
```

Untuk mendukung lebih banyak karakter, banyak set karakter dan sistem pengodean telah digunakan untuk mendukung banyak bahasa (lihat Tabel 11.2).

Kumpulan karakter Unicode dapat mewakili hampir semua karakter yang dikenal manusia dengan rentang titik kode 21 bit (yaitu, 0 hingga 10FFFF dalam notasi heksadesimal).

Sistem pengodean teks UTF-8 mempaskan titik kode Unicode ke dalam aliran data 8 bit yang masuk akal yang sebagian besar kompatibel dengan sistem pemrosesan data ASCII. Hal ini membuat **UTF-8** pilihan modern yang disukai. **UTF** adalah singkatan dari Unicode Transformation Format. Ketika data teks polos ASCII dikonversi ke UTF-8, ia memiliki konten dan ukuran yang persis sama dengan yang asli ASCII. Jadi Anda tidak kehilangan apa pun dengan menggelar lokal UTF-8.

Di bawah lokal UTF-8 dengan program aplikasi yang kompatibel, Anda dapat menampilkan dan menyunting data teks bahasa asing selama fonta yang diperlukan dan metode masukan dipasang dan diaktifkan. Misalnya di bawah lokal ”`LANG=fr_FR.UTF-8`”, gedit(1) (penyunting teks untuk Desktop GNOME) dapat menampilkan dan menyunting data teks karakter Cina sambil menyajikan menu dalam bahasa Prancis.

Tip
Lokal standar baru ”`en_US.UTF-8`” dan lokal standar lama ”`C`”/”`POSIX`” menggunakan pesan bahasa Inggris standar AS, mereka memiliki perbedaan halus dalam urutan penyortiran dll. Jika Anda ingin menangani tidak hanya karakter ASCII tetapi juga menangani semua karakter yang dikodekan UTF-8 dengan anggun sambil mempertahankan perilaku lokal ”`C`” lama, gunakan lokal ”`C.UTF-8`” non-standar pada Debian.

Catatan
Beberapa program mengkonsumsi lebih banyak memori setelah mendukung I18N. Ini karena mereka dikodekan untuk menggunakan UTF-32(UCS4) secara internal untuk mendukung Unicode bagi optimasi kecepatan dan mengkonsumsi 4 byte per setiap data karakter ASCII independen dari lokal yang dipilih. Sekali lagi, Anda tidak kehilangan apa pun dengan menggelar lokal UTF-8.

8.1.2 Konfigurasi ulang lokal

Agar sistem dapat mengakses lokal tertentu, data lokal harus dikompilasi dari basis data lokal.

Paket `locales` **tidak** dilengkapi dengan data lokal yang telah dikompilasi sebelumnya. Anda harus mengkonfigurasinya sebagai:

```
# dpkg-reconfigure locales
```

Proses ini melibatkan 2 langkah.

1. Pilih semua data lokal yang diperlukan untuk dikompilasi ke dalam bentuk biner. (Pastikan untuk menyertakan setidaknya satu lokal UTF-8)
2. Atur nilai lokal baku seluruh sistem dengan membuat ”`/etc/default/locale`” untuk digunakan oleh PAM (lihat Bagian 4.5).

Nilai lokal baku seluruh sistem yang diatur dalam ”`/etc/default/locale`” dapat ditimpa oleh konfigurasi GUI bagi aplikasi GUI.

Catatan
Sistem pengodean tradisional sebenarnya dapat diidentifikasi dengan ”`/usr/share/i18n/SUPPORTED`”. Jadi, ”`LANG=en_US`” adalah ”`LANG=en_US.ISO-8859-1`”.

Paket `locales-all` dilengkapi dengan data lokal yang telah diprakompilasi untuk semua data lokal. Karena itu tidak membuat "`/etc/default/locale`", Anda mungkin masih perlu memasang paket `locales` juga.

Tip
Paket `locales` dari beberapa distribusi turunan Debian datang dengan data locale terprakompilasi untuk semua data locale. Anda perlu memasang paket `locales` dan `locales-all` pada Debian untuk mengemulasi lingkungan sistem seperti itu.

8.1.3 Pengodean nama berkas

Untuk pertukaran data lintas platform (lihat Bagian 10.1.7), Anda mungkin perlu mengait beberapa sistem berkas dengan pengodean tertentu. Misalnya, mount(8) untuk sistem berkas vfat mengasumsikan CP437 jika digunakan tanpa opsi. Anda perlu memberikan opsi mount eksplisit untuk menggunakan UTF-8 atau CP932 untuk nama berkas.

Catatan
Saat memasang otomatis USB flash drive yang dapat dipasang panas di lingkungan desktop modern seperti GNOME, Anda dapat memberikan opsi pasang tersebut dengan mengklik kanan ikon di desktop, klik tab "Drive", klik untuk memperluas "Setting", dan masukkan "utf8" ke "Mount options:". Lain kali USB flash drive ini dipasang, pemasangan dengan UTF-8 akan diaktifkan.

Catatan
Jika Anda memutakhirkan sistem atau memindahkan disk drive dari sistem non-UTF-8 yang lebih lama, nama berkas dengan karakter non-ASCII dapat dikodekan dalam pengodean bersejarah dan usang seperti ISO-8859-1 atau eucJP. Silakan cari bantuan alat konversi teks untuk mengubahnya menjadi UTF-8. Lihat Bagian 11.1.

Samba menggunakan Unicode untuk klien yang lebih baru (Windows NT, 200x, XP) tetapi menggunakan CP850 untuk klien yang lebih lama (DOS dan Windows 9x/Me) sebagai baku. Baku untuk klien yang lebih tua ini dapat diubah menggunakan "`dos charset`" dalam berkas "`/etc/samba/smb.conf`", misalnya, menjadi CP932 untuk bahasa Jepang.

8.1.4 Pesan terlokalkan dan dokumentasi yang diterjemahkan

Terjemahan ada untuk banyak pesan teks dan dokumen yang ditampilkan dalam sistem Debian, seperti pesan kesalahan, keluaran program standar, menu, dan halaman manual. rantai alat perintah GNU gettext (1) digunakan sebagai alat backend untuk sebagian besar kegiatan terjemahan.

Di bawah "Tugas" → "Pelokalan" aptitude(8) menyediakan daftar panjang paket biner yang berguna yang menambahkan pesan terlokalkan ke aplikasi dan menyediakan dokumentasi yang diterjemahkan.

Misalnya, Anda dapat memperoleh pesan lokal untuk manpage dengan memasang paket `manpages-`*`LANG`*. Untuk membaca manpage berbahasa Italia bagi *nama program* dari "`/usr/share/man/it/`", jalankan sebagai berikut.

```
LANG=it_IT.UTF-8 man programname
```

GNU gettext dapat mengakomodasi daftar prioritas bahasa terjemahan dengan variabel lingkungan `$LANGUAGE`. Misalnya:

```
$ export LANGUAGE="pt:pt_BR:es:it:fr"
```

Untuk informasi selengkapnya, lihat `info gettext` dan baca bagian "Variabel LANGUAGE".

8.1.5 Efek dari lokal

Pengurutan karakter dengan sort(1) dan ls(1) dipengaruhi oleh lokal. Mengekspor `LANG=en_US.UTF-8` mengurutkan dalam urutan kamus `A->a->B->b...->Z->z`, sementara mengekspor `LANG=C.UTF-8` mengurutkan dalam urutan biner ASCII `A->B->...->Z->a->b.....`

Format tanggal ls(1) dipengaruhi oleh lokal (lihat Bagian 9.3.4).

Format tanggal dari date(1) dipengaruhi oleh lokal. Misalnya:

```
 $ unset LC_ALL
 $ LANG=en_US.UTF-8 date
Thu Dec 24 08:30:00 PM JST 2023
 $ LANG=en_GB.UTF-8 date
Thu 24 Dec 20:30:10 JST 2023
 $ LANG=es_ES.UTF-8 date
jue 24 dic 2023 20:30:20 JST
 $ LC_TIME=en_DK.UTF-8 date
2023-12-24T20:30:30 JST
```

Tanda baca angka berbeda untuk lokal. Misalnya, di lokal Inggris, seribu satu titik satu ditampilkan sebagai "`1.000,1`" sementara di lokal Jerman, ditampilkan sebagai "`1.000,1`". Anda mungkin melihat perbedaan ini dalam program lembar kerja.

Setiap fitur detail variabel lingkungan "`$LANG`" dapat ditimpa dengan menetapkan variabel "`$LC_*`". Variabel lingkungan ini dapat ditimpa lagi dengan menetapkan variabel "`$LC_ALL`". Lihat manpage locale(7) untuk rinciannya. Kecuali Anda memiliki alasan kuat untuk membuat konfigurasi yang rumit, silakan menjauh dari mereka dan hanya menggunakan variabel "`$LANG`" yang diatur ke salah satu lokal UTF-8.

8.2 Masukan papan ketik

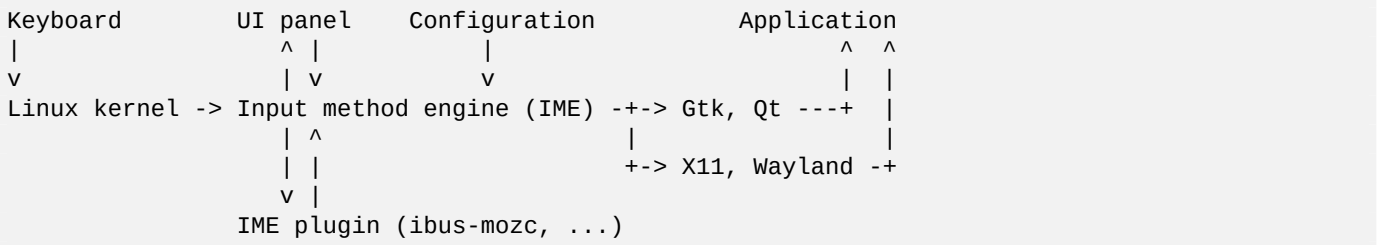
Sistem papan ketik dapat dikonfigurasi pada lapisan yang berbeda dari sistem.

- Kernel Linux: keyboard(5)
- X server: setxkbmap(1), xkeyboard-config(5), variabel lingkungan `XMODIFIERS`
- Lingkungan desktop GUI: Kerangka kerja Metode Masukan" `ibus`, `fcitx5`
- Aplikasi: variabel lingkungan untuk menata sumber masukannya: `GTK_IM_MODULE`, `QT_IM_MODULE`, `QT_IM_MODULES`, ...

Kerangka kerja metode masukan (input method, IM) terdiri dari:

- Mesin metode masukan (Input method engine, IME): Metode masukan sebenarnya
- Konfigurasi: Menangani konfigurasi bagi IBus dan layanan lain seperti pengaya IME
- Panel: Antarmuka pengguna seperti misalnya bilah bahasa dan tabel pemilihan kandidat

Masukan multibahasa ke aplikasi diproses kurang lebih sebagai:



8.2.1 Masukan papan ketik untuk konsol Linux dan X Window

Sistem Debian dapat dikonfigurasi untuk bekerja dengan banyak pengaturan papan ketik internasional menggunakan paket `keyboard-configuration`.

```
# dpkg-reconfigure keyboard-configuration
```

Untuk konsol Linux dan sistem X Window, ini memperbarui parameter konfigurasi di "`/etc/default/keyboard`". Banyak karakter non-ASCII termasuk karakter beraksen yang digunakan oleh banyak bahasa Eropa dapat dibuat tersedia dengan tombol mati, tombol AltGr, dan tombol compose.

Catatan
Jika ibus aktif, konfigurasi papan ketik X klasik Anda dengan `setxkbmap` mungkin ditimpa oleh `ibus` bahkan di bawah lingkungan desktop berbasis X klasik. Anda dapat menonaktifkan `ibus` yang dipasang menggunakan `im-config` untuk mengatur metode masukan ke "None". Untuk informasi selengkapnya, lihat Wiki Debian tentang papan ketik.

8.2.2 Masukan papan ketik untuk Wayland

Tidak seperti protokol X Window, protokol inti Wayland bahkan tidak mendukung masukan dari karakter beraksen. Kompositor Wayland populer, seperti misalnya Mutter untuk GNOME atau KWin untuk KDE, mengimplementasikan protokol ekstensi seperti `text-input-unstable-v3` bagi masukan teks (lihat protokol Wayland saat ini dan status dukungan mereka").

Protokol `text-input-unstable-v3` bekerja baik dengan metode masukan bagi Wayland (lihat Post mortem proyek metode masukan Wayland").

- Kebanyakan aplikasi GUI dibangun dengan pustaka GUI seperti GTK atau Qt yang mendukung `text-input-unstable-` ini.
- Mesin Metode Masukan (Input Method Engine, IME) populer, seperti IBus atau Fcitx (versi 5), dapat bekerja dengan `text-input-unstable-v3` ini.
- IME mendukung masukan teks bagi banyak bahasa dengan pengaya.
- IME terkini mengintegrasikan fungsionalitas X Keyboard Extension (XKB) seperti `setxkbmap` yang sebelumnya disediakan oleh X Window untuk mendukung masukan teks karakter beraksen bagi bahasa-bahasa Eropa untuk Wayland.

8.2.3 Dukungan metode masukan dengan IBus

Untuk GNOME, "`ibus`" adalah IME baku yang secara otomatis dipasang melalui ketergantungan paketnya.

- Kebanyakan fitur masukan papan ketik multi-bahasa dapat dikonfigurasi dari "Pengaturan GNOME" atau "Tweak GNOME".
- Beberapa fitur masukan papan ketik multi-bahasa mungkin perlu dikonfigurasi dari perintah ibus-setup(1).
- Masukan papan ketik emoji tersedia dengan mengetikkan `SUPER-.` (Secara simultan menekan tombol Windows dan titik) diikuti dengan suatu kata kunci bagi setiap emoji dan tombol SPASI.

Daftar paket-paket IBus dan pengayanya adalah sebagai berikut.

paket	popcon	ukuran	lokal yang didukung
ibus	V:212, I:253	1884	kerangka kerja metode masukan menggunakan dbus
ibus-mozc	V:2.1, I:3.8	1054	Jepang
ibus-anthy	V:0.5, I:1.1	8740	Jepang
ibus-skk	V:0.03, I:0.14	242	Jepang
ibus-kkc	V:0.04, I:0.19	206	Jepang
ibus-libpinyin	V:1.1, I:5.1	123	Cina (untuk zh_CN)
ibus-chewing	V:0.16, I:0.84	298	Tionghoa (untuk zh_TW)
ibus-libzhuyin	I:0.10	41021	Tionghoa (untuk zh_TW)
ibus-rime	V:0.29, I:0.51	76	Tionghoa (untuk zh_CN/zh_TW)
ibus-cangjie	V:0.02, I:0.10	235	Tionghoa (untuk zh_HK)
ibus-hangul	V:0.3, I:1.9	261	Korea
ibus-libthai	V:0.00, I:0.05	84	Bahasa Thai
ibus-table-thai	I:0.04	59	Bahasa Thai
ibus-unikey	V:0.22, I:0.40	286	Vietnam
keyman	V:0.01, I:0.10	507	Multibahasa: Pengaya Keyman untuk lebih dari 2000 bahasa
ibus-table	V:0.04, I:0.92	2364	tabel pengaya untuk IBus
ibus-m17n	V:0.1, I:1.8	373	Multibahasa: Indic, Arab, dan lain-lain

Tabel 8.1: Daftar IBus dan paket-paket pengayanya

paket	popcon	ukuran	lokal yang didukung
fcitx5	V:8, I:13	712	kerangka metode input yang kompatibel dengan ”`ibus`”
fcitx5-mozc	V:1.1, I:1.8	1328	Jepang
fcitx5-anthy	V:0.07, I:0.25	853	Jepang
fcitx5-skk	V:0.05, I:0.15	370	Jepang
fcitx5-kkc	V:0.01, I:0.06	418	Jepang
fcitx5-chinese-addons	I:9.2	17	Tionghoa (metapaket untuk zh_*)
fcitx5-pinyin	V:4.0, I:9.7	1000	Cina (untuk zh_CN)
fcitx5-chewing	V:0.29, I:0.98	228	Tionghoa (untuk zh_TW)
fcitx5-zhuyin	I:0.08	108535	Tionghoa (untuk zh_TW)
fcitx5-rime	V:0.46, I:0.93	393	Tionghoa (untuk zh_CN/zh_TW)
fcitx5-table-cangjie-large	I:0.13	1292	Tionghoa (untuk zh_HK)
fcitx5-hangul	V:0.12, I:0.27	239	Korea
fcitx5-libthai	V:0.00, I:0.04	119	Bahasa Thai
fcitx5-table-thai	I:0.09	34	Bahasa Thai
fcitx5-unikey	V:0.13, I:0.23	624	Vietnam
fcitx5-m17n	V:0.16, I:0.61	171	Multibahasa: Indic, Arab, dan lain-lain
fcitx5-table	V:0.3, I:9.6	428	pengaya tabel untuk fcitx5
fcitx5-keyman	V:0.04, I:0.05	241	Multibahasa: Pengaya Keyman untuk lebih dari 2000 bahasa

Tabel 8.2: Daftar paket Fcitx5 dan pengayanya

8.2.4 Dukungan metode masukan dengan Fcitx

Kerangka metode input Fcitx (versi 5) populer di kalangan pengguna Tionghoa dan kompatibel dengan "`ibus`".

Daftar paket "`fcitx5`" dan pengayanya adalah sebagai berikut.

8.2.5 Contoh untuk bahasa Jepang

Saya menemui bahwa metode masukan Jepang yang dimulai di bawah lingkungan bahasa Inggris ("`en_US.UTF-8`" sangat berguna. Berikut adalah bagaimana saya melakukan ini dengan IBus:

1. Pasang paket alat masukan Jepang `ibus-mozc` (atau `ibus-anthy`).
2. • Generuk: Jalankan ibus-setup(1) → pilih "Metode Masukan" → klik "`Tambahkan`" → "Jepang" → "Mozc (atau Anthy)" → klik "Tambahkan"
   • GNOME: Pilih "Pengaturan" → "Papan Ketik" → "Sumber Masukan" → klik "+" di "Sumber Masukan" → "Jepang" → Mozc (atau Anthy)" → klik "Tambahkan"
3. Anda dapat memilih sebanyak mungkin sumber masukan.
4. Login ulang ke akun pengguna.
5. Siapkan setiap sumber masukan dengan mengklik kanan ikon bilah alat GUI.
6. Beralih di antara sumber masukan yang dipasang dengan SUPER-SPACE. (SUPER biasanya adalah kunci Windows.)

Tip
Jika Anda ingin memiliki akses ke lingkungan papan ketik alfabet saja dengan papan ketik Jepang fisik di mana shift-2 memiliki " (tanda kutip ganda) terukir, Anda memilih "Jepang" dalam prosedur di atas. Anda dapat memasukkan bahasa Jepang menggunakan "mozc Jepang (atau anthy)" dengan papan ketik fisik "AS" di mana shift-2 memiliki @ yang terukir.

- Untuk Wayland:
  - Paket `im-config` tidak melakukan apa pun aman untuk dihapus.
  - Anda mungkin tidak perlu menata variabel lingkungan kecuali bagi kompatibilitas mundur dsb.
  - Bila Anda perlu menata suatu variabel lingkungan, buat suatu berkas misalnya "`~/.config/environment.d/50-inp` untuk menata mereka.
- Untuk X Window:
  - Pasang paket `im-config`.
  - Entri menu GUI untuk im-config(8) adalah "Metode masukan".
  - Atau, jalankan "`im-config`" dari shell pengguna.
  - im-config(8) berperilaku berbeda jika perintah dijalankan dari root atau bukan.
  - im-config(8) memungkinkan metode masukan terbaik pada sistem sebagai baku tanpa tindakan pengguna.

8.3 Keluaran tampilan

Konsol Linux hanya dapat menampilkan karakter terbatas. (Anda perlu menggunakan program terminal khusus seperti jfbterm(1) untuk menampilkan bahasa non-Eropa pada konsol non-GUI.)

Lingkungan GUI (Bab 7) dapat menampilkan karakter apa pun di UTF-8 selama fonta yang diperlukan terpasang dan diaktifkan. (Pengodean data fonta asli diurus dan transparan kepada pengguna.)

8.3.1 Konfigurasi terminal

Sistem Debian dapat dikonfigurasi untuk bekerja dengan banyak pengaturan konsol internasional menggunakan paket `console-setup`.

```
# dpkg-reconfigure console-setup
```

Untuk konsol Linux dan sistem X Window, ini memperbarui parameter konfigurasi di "`/etc/default/console-setup`" untuk menampilkan banyak karakter non-ASCII termasuk karakter beraksen yang dipakai oleh banyak bahasa Eropa dapat menjadi tersedia.

Ada beberapa komponen untuk mengkonfigurasi konsol karakter dan fitur sistem ncurses(3).

- Berkas "`/etc/terminfo/*/*`" (terminfo(5))
- Variabel lingkungan "`$TERM`" (term(7))
- setterm(1), stty(1), tic(1), dan toe(1)

Jika entri `terminfo` untuk `xterm` tidak berfungsi dengan `xterm` non-Debian, ubah tipe terminal Anda, "`$TERM`", dari "`xterm`" ke salah satu versi terbatas fitur seperti "`xterm-r6`" ketika Anda masuk ke sistem Debian dari jarak jauh. Lihat "`/usr/share/doc/libncurses5/FAQ`" untuk informasi lebih lanjut. "`dumb`" adalah penyebut umum terrendah untuk "`$TERM`".

8.3.2 Karakter Lebar Karakter Ambigu Asia Timur

Di bawah lokal Asia Timur, karakter box drawing, Yunani, dan Sirilik mungkin ditampilkan lebih lebar dari lebar yang diinginkan sehingga menyebabkan keluaran terminal yang tidak sejajar (lihat Unicode Standard Annex #11, 4.2 Ambiguous Characters).

Anda dapat mengatasi masalah ini:

- `gnome-terminal`: Preferensi → Profil → *Nama profil* → Kompatibilitas → Karakter lebar-ambigu → Sempit
- `ncurses`: Atur lingkungan `export NCURSES_NO_UTF8_ACS=0`.

Bab 9

Tips sistem

Di sini, saya menjelaskan tips dasar untuk mengonfigurasi dan mengelola sistem, sebagian besar dari konsol.

9.1 Tips konsol

Ada beberapa program utilitas untuk membantu aktivitas konsol Anda.

paket	popcon	ukuran	deskripsi
mc	V:40, I:176	1590	Lihat Bagian 1.3
bsdutils	V:421, I:999	338	perintah script(1) untuk membuat rekaman sesi terminal
screen	V:50, I:191	1007	pemultipleks terminal dengan emulasi terminal VT100/ANSI
tmux	V:106, I:167	1423	pemultipleks terminal alternatif (Gunakan ”Control-B” sebagai gantinya)
ripgrep	V:17, I:48	5284	pencarian string rekursif yang cepat dalam pohon kode sumber dengan penyaringan otomatis
fzf	V:10, I:36	5677	pencari teks fuzzy
fzy	V:0.07, I:0.34	59	pencari teks fuzzy
rlwrap	V:1, I:11	328	pembungkus baris perintah fitur readline
ledit	V:0.5, I:7.3	375	pembungkus baris perintah fitur readline
rlfe	V:0.02, I:0.48	45	pembungkus baris perintah fitur readline

Tabel 9.1: Daftar program untuk mendukung aktivitas konsol

9.1.1 Merekam aktivitas shell secara bersih

Penggunaan sederhana script(1) (lihat Bagian 1.4.9) untuk merekam aktivitas shell menghasilkan berkas dengan karakter kontrol. Hal ini dapat dihindari dengan menggunakan col(1) sebagai berikut.

```
$ script
Script started, file is typescript
```

Lakukan apapun … dan tekan `Ctrl-D` untuk keluar `dari skrip`.

```
$ col -bx < typescript > cleanedfile
$ vim cleanedfile
```

Ada metode alternatif untuk merekam aktivitas shell:

- Menggunakan `tee` (dapat digunakan selama proses boot di initramfs):

```
$ sh -i 2>&1 | tee typescript
```

- Menggunakan `gnome-terminal` dengan penyangga baris yang dinaikkan untuk gulir balik.
- Menggunakan `screen` dengan "`^A H`" (lihat Bagian 9.1.2) untuk melakukan perekaman konsol.
- Menggunakan `vim` dengan "`:terminal`" untuk memasuki mode terminal. Gunakan "`Ctrl-W N`" untuk keluar dari mode terminal ke mode normal. Gunakan "`:w typescript`" untuk menulis penyangga ke berkas.
- Menggunakan `emacs` dengan "`M-x shell`", "`M-x eshell`", atau "`M-x term`" untuk memasuki konsol rekaman. Gunakan "`C-x C-w`" untuk menulis penyangga ke berkas.

9.1.2 Program screen

screen(1) tidak hanya memungkinkan satu jendela terminal bekerja dengan beberapa proses, tetapi juga memungkinkan **proses shell jarak jauh untuk bertahan dari koneksi yang terputus**. Berikut adalah skenario penggunaan screen(1) yang biasa.

1. Anda masuk ke mesin remote.
2. Anda memulai `screen` pada satu konsol.
3. Anda menjalankan beberapa program di jendela `screen` yang dibuat dengan `^A c` ("Control-A" diikuti oleh "c").
4. Anda beralih di antara beberapa jendela `screen` dengan `^A n` ("Control-A" diikuti oleh "n").
5. Tiba-tiba Anda harus meninggalkan terminal Anda, tetapi Anda tidak ingin kehilangan pekerjaan aktif Anda dengan mempertahankan koneksi.
6. Anda dapat **melepaskan** sesi `layar` dengan metode apa pun.
   - Secara brutal mencabut koneksi jaringan Anda
   - Mengetik `^A d` ("Control-A" diikuti oleh "d") dan keluar secara manual dari sambungan jarak jauh
   - Ketik `^A DD` ("Control-A" diikuti oleh "DD") agar `layar` terlepas dan log Anda keluar
7. Anda masuk lagi ke mesin remote yang sama (bahkan dari terminal yang berbeda).
8. Anda memulai `screen` sebagai "`screen -r`".
9. `screen` secara ajaib **memasang kembali** semua jendela `screen` sebelumnya dengan semua program aktif berjalan.

Tip
Anda dapat menghemat biaya koneksi dengan `screen` untuk koneksi jaringan terukur seperti dial-up dan paket, karena Anda dapat membiarkan proses aktif saat terputus, lalu mencantolkan kembali nanti ketika Anda terhubung lagi.

Dalam sesi `screen`, semua masukan papan ketik dikirim ke jendela Anda saat ini kecuali untuk penekanan tombol perintah. Semua penekanan tombol perintah `screen` dimasukkan dengan mengetik `^A` ("Control-A") ditambah satu tombol [ditambah parameter apa pun]. Berikut adalah hal-hal penting untuk diingat.

Lihat screen(1) untuk detailnya.

Lihat tmux(1)untuk fungsionalitas dari perintah alternatif.

pengikatan tombol	arti
^A ?	memperlihatkan layar bantuan (menampilkan pengikatan tombol)
^A c	membuat jendela baru dan beralih ke jendela itu
^A n	pergi ke jendela berikutnya
^A p	pergi ke jendela sebelumnya
^A 0	masuk ke jendela nomor 0
^A 1	masuk ke jendela nomor 1
^A w	memperlihatkan daftar jendela
^A a	mengirim Ctrl-A ke jendela saat ini sebagai masukan papan ketik
^A h	menulis hardcopy jendela saat ini ke berkas
^A H	memulai/mengakhiri mencatat jendela saat ini ke berkas
^A ^X	mengunci terminal (dilindungi kata sandi)
^A d	melepas sesi screen dari terminal
^A DD	melepas sesi screen dan log keluar

Tabel 9.2: Daftar pengikatan tombol untuk screen

9.1.3 Menavigasi di sekitar direktori

Pada Bagian 1.4.2, 2 tips untuk memungkinkan navigasi cepat di sekitar direktori dijelaskan: `$CDPATH` dan `mc`.

Jika Anda menggunakan program filter teks fuzzy, Anda dapat melakukannya tanpa mengetik path yang tepat. Untuk `fzf`, sertakan yang berikut dalam `~/.bashrc`.

```
FZF_KEYBINDINGS_PATH=/usr/share/doc/fzf/examples/key-bindings.bash
if [ -f $FZF_KEYBINDINGS_PATH ]; then
  . $FZF_KEYBINDINGS_PATH
fi
```

Sebagai contoh:

- Anda dapat melompat ke subdirektori yang sangat dalam dengan upaya minimal. Anda pertama mengetik ”`cd **`” dan menekan `Tab`. Kemudian Anda akan diberi tampilan path kandidat. Mengetikkan string path parsial, misalnya, `s/d/b foo`, akan mempersempit path kandidat. Anda memilih path yang akan digunakan oleh `cd` dengan kursor dan tombol return.
- Anda dapat memilih perintah dari riwayat perintah dengan lebih efisien dengan upaya minimal. Anda menekan `Ctrl-R` di sapaan perintah. Kemudian Anda akan dimintai perintah kandidat. Mengetikkan string perintah parsial, misalnya, `vim d`, akan mempersempit kandidat. Anda memilih yang akan digunakan dengan kursor dan tombol return.

9.1.4 Pembungkus readline

Beberapa perintah seperti `/usr/bin/dash` yang tidak memiliki kemampuan penyuntingan riwayat baris perintah dapat menambahkan fungsionalitas tersebut secara transparan dengan berjalan di bawah `rlwrap` atau yang setara.

```
$ rlwrap dash -i
```

Ini menyediakan platform yang nyaman untuk menguji titik-titik halus untuk `dash` dengan lingkungan mirip `bash` yang ramah.

9.1.5 Memindai pohon kode sumber

Perintah rg(1) dalam paket `ripgrep` menawarkan alternatif yang lebih cepat dari perintah grep(1) untuk memindai pohon kode sumber bagi situasi tipikal. Ini mengambil keuntungan dari CPU multi-core modern dan secara otomatis menerapkan filter yang masuk akal untuk melewati beberapa berkas.

9.2 Menyesuaikan vim

Setelah Anda mempelajari dasar-dasar vim(1) melalui Bagian 1.4.8, silakan baca "Seven habits of effective text editing (2000)" dari Bram Moolenaar untuk memahami bagaimana `vim` harus digunakan.

9.2.1 Menyesuaikan vim dengan fitur internal

Perilaku `vim` dapat diubah secara signifikan dengan mengaktifkan fitur internalnya melalui perintah mode `Ex` seperti "`set ...`" untuk mengatur opsi vim.

Perintah mode `Ex` ini dapat dimasukkan dalam berkas vimrc pengguna, "`~/.vimrc`" tradisional, atau"`~/.vim/vimrc`" yang ramah git. Berikut adalah contoh yang sangat sederhana 1:

```
""" Generic baseline Vim and Neovim configuration (~/.vimrc)
"""   - For NeoVim, use "nvim -u ~/.vimrc [filename]"
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
let mapleader = ' '             " :h mapleader
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
set nocompatible                " :h 'cp -- sensible (n)vim mode
syntax on                       " :h :syn-on
filetype plugin indent on       " :h :filetype-overview
set encoding=utf-8              " :h 'enc (default: latin1) -- sensible encoding
""" current vim option value can be verified by :set encoding?
set backspace=indent,eol,start  " :h 'bs (default: nobs) -- sensible BS
set statusline=%<%f%m%r%h%w%=%y[U+%04B]%2l/%2L=%P,%2c%V
set listchars=eol:¶,tab:b''▫b''\ ,extends:b''▫b'',precedes:b''▫b'',nbsp:b''▫b''
set viminfo=!,'100,<5000,s100,h " :h 'vi -- bigger copy buffer etc.
""" Pick "colorscheme" from blue darkblue default delek desert elflord evening
""" habamax industry koehler lunaperche morning murphy pablo peachpuff quiet ron
""" shine slate torte zellner
colorscheme industry
""" don't pick "colorscheme" as "default" which may kill SpellUnderline settings
set scrolloff=5                 " :h 'scr -- show 5 lines around cursor
set laststatus=2                " :h 'ls (default 1)  k
""" boolean options can be unset by prefixing "no"
set ignorecase                  " :h 'ic
set smartcase                   " :h 'scs
set autoindent                  " :h 'ai
set smartindent                 " :h 'si
set nowrap                      " :h 'wrap
"set list                        " :h 'list (default nolist)
set noerrorbells                " :h 'eb
set novisualbell                " :h 'vb
set t_vb=                       " :h 't_vb -- termcap visual bell
set spell                       " :h 'spell
set spelllang=en_us,cjk         " :h 'spl -- english spell, ignore CJK
set clipboard=unnamedplus       " :h 'cb -- cut/copy/paste with other app
set hidden                      " :h 'hid
set autowrite                   " :h 'aw
set timeoutlen=300              " :h 'tm
```

Keymap dari `vim` dapat diubah dalam berkas vimrc pengguna. Misalnya:



Perhatian
Jangan mencoba mengubah pengikatan tombol baku tanpa alasan yang sangat bagus.

1Contoh penyesuaian yang lebih rumit: "Vim Galore", "sensible.vim", ...

```
""" Popular mappings (imitating LazyVim etc.)
""" Window moves without using CTRL-W which is dangerous in INSERT mode
nnoremap <C-H> <C-W>h
nnoremap <C-J> <C-W>j
nnoremap <C-K> <C-W>k
silent! nnoremap <C-L> <C-W>l
""" Window resize
nnoremap <C-LEFT> <CMD>vertical resize -2<CR>
nnoremap <C-DOWN> <CMD>resize -2<CR>
nnoremap <C-UP> <CMD>resize +2<CR>
nnoremap <C-RIGHT> <CMD>vertical resize +2<CR>
""" Clear hlsearch with <ESC> (<C-L> is mapped as above)
nnoremap <ESC> <CMD>noh<CR><ESC>
inoremap <ESC> <CMD>noh<CR><ESC>
""" center after jump next
nnoremap n nzz
nnoremap N Nzz
""" fast "jk" to get out of INSERT mode (<ESC>)
inoremap  jk <CMD>noh<CR><ESC>
""" fast "<ESC><ESC>" to get out of TERM mode (CTRL-\ CTRL-N)
tnoremap <ESC><ESC> <C-\><C-N>
""" fast "jk" to get out of TERM mode (CTRL-\ CTRL-N)
tnoremap jk <C-\><C-N>
""" previous/next trouble/quickfix item
nnoremap [q <CMD>cprevious<CR>
nnoremap ]q <CMD>cnext<CR>
""" buffers
nnoremap <S-H> <CMD>bprevious<CR>
nnoremap <S-L> <CMD>bnext<CR>
nnoremap [b <CMD>bprevious<CR>
nnoremap ]b <CMD>bnext<CR>
""" Add undo break-points
inoremap  , ,<C-G>u
inoremap  . .<C-G>u
inoremap  ; ;<C-G>u
""" save file
inoremap <C-S> <CMD>w<CR><ESC>
xnoremap <C-S> <CMD>w<CR><ESC>
nnoremap <C-S> <CMD>w<CR><ESC>
snoremap <C-S> <CMD>w<CR><ESC>
""" better indenting
vnoremap < <gv
vnoremap > >gv
""" terminal (Somehow under Linux, <C-/> becomes <C-_> in Vim)
nnoremap <C-_> <CMD>terminal<CR>
"nnoremap <C-/> <CMD>terminal<CR>
""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
if ! has('nvim')
""" Toggle paste mode with <SPACE>p for Vim (no need for Nvim)
set pastetoggle=<leader>p
""" nvim default mappings for Vim.  See :h default-mappings in nvim
""" copy to EOL (no delete) like D for d
noremap Y y$
""" sets a new undo point before deleting
inoremap <C-U> <C-G>u<C-U>
inoremap <C-W> <C-G>u<C-W>
""" <C-L> is re-purposed as above
""" execute the previous macro recorded with Q
nnoremap Q @@
""" repeat last substitute and *KEEP* flags
nnoremap & :&&<CR>
```

```
""" search visual selected string for visual mode
xnoremap * y/\V<C-R>"<CR>
xnoremap # y?\V<C-R>"<CR>
endif
```

Agar pengikatan tombol di atas berfungsi dengan baik, program terminal perlu dikonfigurasi untuk menghasilkan ”ASCII DEL” bagi tombol `Backspace` dan ”Escape sequence” untuk tombol `Delete`.

Konfigurasi lain-lain dapat diubah dalam berkas vimrc pengguna. Mis.:

```
""" Use faster 'rg' (ripgrep package) for :grep
if executable("rg")
  set grepprg=rg\ --vimgrep\ --smart-case
  set grepformat=%f:%l:%c:%m
endif
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
""" Retain last cursor position :h '"
augroup RetainLastCursorPosition
  autocmd!
  autocmd BufReadPost *
    \ if line("'\"") > 0 && line ("'\"") <= line("$") |
    \   exe "normal! g'\"" |
    \ endif
augroup END
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
""" Force to use underline for spell check results
augroup SpellUnderline
  autocmd!
  autocmd ColorScheme * highlight SpellBad term=Underline gui=Undercurl
  autocmd ColorScheme * highlight SpellCap term=Underline gui=Undercurl
  autocmd ColorScheme * highlight SpellLocal term=Underline gui=Undercurl
  autocmd ColorScheme * highlight SpellRare term=Underline gui=Undercurl
augroup END
"""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""""
""" highlight tailing spaces except when typing as red (set after colorscheme)
highlight TailingWhitespaces ctermbg=red guibg=red
""" \s\+     1 or more whitespace character: <Space> and <Tab>
""" \%#\@<!  Matches with zero width if the cursor position does NOT match.
match TailingWhitespaces /\s\+\%#\@<!$/
```

9.2.2 Menyesuaikan vim dengan paket eksternal

Paket pengaya eksternal yang menarik dapat ditemukan:

- Vim - editor teks di mana-mana -- Situs hulu resmi Vim dan skrip vim
- VimAwsome -- Daftar pengaya Vim
- vim-scripts -- Paket Debian: kumpulan skrip vim

Paket pengaya dalam paket vim-scripts dapat difungsikan memakai berkas vimrc pengguna. Mis.:

```
packadd! secure-modelines
packadd! winmanager
" IDE-like UI for files and buffers with <space>w
nnoremap <leader>w         :WMToggle<CR>
```

Sistem paket Vim asli yang baru, bekerja dengan baik dengan ”`git`” dan ”`git submodule`”. Salah satu contoh konfigurasi tersebut dapat ditemukan di repositori git saya: dot-vim. Ini pada dasarnya:

- Dengan menggunakan ”`git`” dan ”`git submodule`”, paket eksternal terbaru, seperti ”*nama*”, ditempatkan ke dalam `~/.vim/pack/*/opt/`*nama* dan sejenisnya.
- Dengan menambahkan baris `:packadd!` *nama* ke berkas vimrc pengguna, paket-paket ini ditempatkan pada `runtimepath`.
- Vim memuat paket-paket ini pada `runtimepath` selama inisialisasinya.
- Pada akhir inisialisasinya, tag untuk dokumen yang dipasang diperbarui dengan ”`helptags ALL`”.

Untuk lebih lanjut, silakan mulai `vim` dengan ”`vim --startuptime vimstart.log`” untuk memeriksa urutan eksekusi aktual dan waktu yang dihabiskan untuk setiap langkah.

Cukup membingungkan untuk melihat terlalu banyak cara[2] untuk mengelola dan memuat paket-paket eksternal ini bagi `vim`. Memeriksa informasi asli adalah obat terbaik.

ketikan tombol	**informasi**
`:help package`	penjelasan tentang mekanisme paket vim
`:help runtimepath`	penjelasan tentang mekanisme `runtimepath`
`:version`	keadaan internal termasuk kandidat untuk berkas vimrc
`:echo $VIM`	variabel lingkungan ”`$VIM`” yang digunakan untuk menemukan berkas vimrc
`:set runtimepath?`	daftar direktori yang akan dicari untuk semua berkas dukungan runtime
`:echo $VIMRUNTIME`	variabel lingkungan ”`$VIMRUNTIME`” digunakan untuk menemukan berbagai berkas dukungan runtime yang disediakan oleh sistem

Tabel 9.3: Informasi tentang inisialisasi `vim`

9.3 Perekaman dan presentasi data

9.3.1 Daemon log

Banyak program tradisional merekam aktivitas mereka dalam format berkas teks di bawah direktori ”`/var/log/`”.

logrotate(8) digunakan untuk menyederhanakan administrasi berkas log pada sistem yang menghasilkan banyak berkas log.

Banyak program baru merekam aktivitasnya dalam format berkas biner menggunakan layanan Jurnal systemd-journald(8) di bawah direktori ”`/var/log/journal`”.

Anda dapat mencatat data ke Jurnal systemd-journald(8) dari skrip shell dengan menggunakan perintah systemd-cat(1).

Lihat Bagian 3.5 dan Bagian 3.4.

9.3.2 Penganalisis log

Berikut adalah penganalisis log terkenal (”`~Gsecurity::log-analyzer`” dalam aptitude(8)).

Catatan
CRM114 menyediakan infrastruktur bahasa untuk menulis filter **fuzzy** dengan perpustakaan regex TRE. Penggunaannya yang populer adalah filter surel spam tetapi dapat digunakan sebagai penganalisis log.

2vim-pathogen populer.

paket	popcon	ukuran	deskripsi
fail2ban	V:101, I:111	2275	memblokir IP-IP yang menyebabkan beberapa kesalahan otentikasi
logwatch	V:9, I:10	2440	penganalisis log dengan keluaran bagus yang ditulis dalam Perl
awstats	V:5.1, I:8.0	6934	penganalisis log server web yang kuat dan penuh fitur
analog	V:3, I:83	3739	penganalisis log server web
sarg	V:0.65, I:0.71	863	generator laporan analisis squid
pflogsumm	V:1.5, I:3.5	173	Perangkum entri log Postfix
fwlogwatch	V:0.10, I:0.16	487	penganalisis log firewall
squidview	V:0.04, I:0.41	224	memantau dan menganalisis berkas access.log squid
swatch	V:0.07, I:0.30	99	penampil berkas log dengan pencocokan regex, penyorotan, dan kait
crm114	V:0.06, I:0.32	1371	Mutilator Regex dan Filter Spam yang Dapat Dikontrol (CRM114)
icmpinfo	V:0.01, I:0.35	42	menafsirkan pesan ICMP

Tabel 9.4: Daftar penganalisis log sistem

9.3.3 Tampilan data teks yang dikustomisasi

Meskipun alat pager seperti more(1) dan less(1) (lihat Bagian 1.4.5) dan alat ubahan untuk menyoroti dan memformat (lihat Bagian 11.1.8) dapat menampilkan data teks dengan baik, penyunting tujuan umum (lihat Bagian 1.4.6) paling serbaguna dan dapat disesuaikan.

Tip
Untuk vim(1) dan alias mode pager-nya view(1), "`:set hls`" memungkinkan pencarian yang disorot.

9.3.4 Tampilan waktu dan tanggal yang disesuaikan

Format tampilan baku waktu dan tanggal oleh perintah "`ls -l`" tergantung pada **lokal** (lihat Bagian 1.2.6 untuk nilainya). Variabel "`$LANG`" dirujuk terlebih dahulu dan dapat ditimpa oleh variabel lingkungan "`$LC_TIME`" atau "`$LC_ALL`" yang diekspor.

Format tampilan baku aktual untuk setiap lokal tergantung pada versi pustaka C standar (paket `libc6`) yang digunakan. Yaitu, rilis Debian yang berbeda memiliki baku yang berbeda. Untuk format iso, lihat ISO 8601.

Jika Anda benar-benar ingin menyesuaikan format tampilan waktu dan tanggal ini di luar **lokal**, Anda harus mengatur **nilai gaya waktu** dengan argumen "`--time-style`" atau dengan nilai "`$TIME_STYLE`" (lihat ls(1), date(1), "`info coreutils 'ls invocation'`").

Tip
Anda dapat menghilangkan mengetik opsi panjang pada baris perintah menggunakan alias perintah (lihat Bagian 1.5.9):

```
alias ls='ls --time-style=+%d.%m.%y %H:%M'
```

nilai gaya waktu	lokal	tampilan waktu dan tanggal
iso	apa pun	01-19 00:15
long-iso	apa pun	2009-01-19 00:15
full-iso	apa pun	2009-01-19 00:15:16.000000000 +0900
locale	C	Jan 19 00:15
locale	en_US.UTF-8	Jan 19 00:15
locale	es_ES.UTF-8	ene 19 00:15
+%d.%m.%y %H:%M	apa pun	19.01.09 00:15
+%d.%b.%y %H:%M	C atau en_US.UTF-8	19.Jan.09 00:15
+%d.%b.%y %H:%M	es_ES.UTF-8	19.ene.09 00:15

Tabel 9.5: Menampilkan contoh waktu dan tanggal untuk perintah "`ls -l`" dengan **nilai gaya waktu**

9.3.5 Echo shell berwarna

Shell echo ke sebagian besar terminal modern dapat diwarnai menggunakan kode escape ANSI (lihat "`/usr/share/doc/x`

Misalnya, coba yang berikut ini

```
$ RED=$(printf "\x1b[31m")
$ NORMAL=$(printf "\x1b[0m")
$ REVERSE=$(printf "\x1b[7m")
$ echo "${RED}RED-TEXT${NORMAL} ${REVERSE}REVERSE-TEXT${NORMAL}"
```

9.3.6 Perintah berwarna

Perintah berwarna berguna untuk memeriksa keluaran mereka di lingkungan interaktif. Saya memasukkan yang berikut dalam "`~/.bashrc`".

```
if [ "$TERM" != "dumb" ]; then
    eval "`dircolors -b`"
    alias ls='ls --color=always'
    alias ll='ls --color=always -l'
    alias la='ls --color=always -A'
    alias less='less -R'
    alias ls='ls --color=always'
    alias grep='grep --color=always'
    alias egrep='egrep --color=always'
    alias fgrep='fgrep --color=always'
    alias zgrep='zgrep --color=always'
else
    alias ll='ls -l'
    alias la='ls -A'
fi
```

Penggunaan alias membatasi efek warna pada penggunaan perintah interaktif. Ini memiliki keuntungan dibandingkan mengekspor variabel lingkungan "`export GREP_OPTIONS='--color=auto'`" karena warna dapat dilihat di bawah program pager seperti less(1). Jika Anda ingin meniadakan warna saat menyalurkan ke program lain, gunakan "`--color=auto`" sebagai gantinya dalam contoh di atas untuk "`~/.bashrc`".

Tip
Anda dapat mematikan alias mewarnai ini di lingkungan interaktif dengan memanggil shell memakai "`TERM=dumb bash`".

9.3.7 Merekam aktivitas penyunting untuk pengulangan yang kompleks

Anda dapat merekam aktivitas penyunting untuk pengulangan yang kompleks.

Untuk Vim, sebagai berikut.

- ”qa”: mulai merekam karakter yang diketik ke dalam register bernama ”a”.
- … aktivitas penyunting
- ”q”: mengakhiri rekaman karakter yang diketik.
- ”@a”: jalankan isi register ”a”.

Untuk Emacs, sebagai berikut.

- ”`C-x (`”: mulai mendefinisikan makro papan ketik.
- … aktivitas penyunting
- ”`C-x )`”: akhir mendefinisikan makro papan ketik.
- ”`C-x e`”: jalankan makro papan ketik.

9.3.8 Merekam gambar grafis dari aplikasi X

Ada beberapa cara untuk merekam gambar grafis dari aplikasi X, termasuk tampilan `xterm`.

paket	popcon	ukuran	layar
`gnome-screenshot`	V:12, I:94	1115	Wayland
`flameshot`	V:7, I:17	3542	Wayland
`gimp`	V:34, I:211	33315	Wayland + X
`x11-apps`	V:31, I:443	2461	X
`imagemagick`	V:7, I:272	80	X
`scrot`	V:4, I:48	144	X

Tabel 9.6: Daftar alat manipulasi gambar grafis

9.3.9 Merekam perubahan dalam berkas konfigurasi

Ada alat khusus untuk merekam perubahan dalam berkas konfigurasi dengan bantuan DVCS dan untuk membuat snapshot sistem di Btrfs.

paket	popcon	ukuran	deskripsi
`etckeeper`	V:24, I:27	157	menyimpan berkas konfigurasi dan metadatanya dengan Git (baku), Mercurial, atau GNU Bazaar
`timeshift`	V:7, I:13	4481	utilitas pemulihan sistem menggunakan snapshot rsync atau BTRFS
`snapper`	V:6.1, I:8.3	2396	Alat manajemen snapshot sistem berkas Linux

Tabel 9.7: Daftar paket yang dapat merekam riwayat konfigurasi

Anda juga dapat mempertimbangkan pendekatan skrip lokal Bagian 10.2.3.

9.4 Memantau, mengendalikan, dan memulai aktivitas program

Aktivitas program dapat dipantau dan dikendalikan menggunakan alat khusus.

paket	popcon	ukuran	deskripsi
coreutils	V:912, I:1000	17994	nice(1): menjalankan program dengan prioritas penjadwalan yang dimodifikasi
bsdutils	V:421, I:999	338	renice(1): memodifikasi prioritas penjadwalan dari proses yang berjalan
procps	V:831, I:997	2691	Utilitas sistem berkas "/proc": ps(1), top(1), kill(1), watch(1), …
psmisc	V:400, I:715	950	Utilitas sistem berkas "/proc": killall(1), fuser(1), peekfd(1), pstree(1)
time	V:6, I:76	148	time(1): menjalankan program untuk melaporkan penggunaan sumber daya sistem sehubungan dengan waktu
sysstat	V:115, I:164	1904	sar(1), iostat(1), mpstat(1), …: alat kinerja sistem untuk Linux
isag	V:0.1, I:3.1	109	Grafer Aktivitas Sistem Interaktif untuk sysstat
lsof	V:456, I:950	492	lsof(8): daftar berkas yang dibuka oleh proses berjalan menggunakan opsi "-p"
strace	V:10, I:100	3209	strace(1): melacak panggilan sistem dan sinyal
ltrace	V:1, I:11	420	ltrace(1): melacak panggilan pustaka
xtrace	V:0.03, I:0.69	342	xtrace(1): melacak komunikasi antara klien dan server X11
powertop	V:36, I:230	696	powertop(1): informasi tentang penggunaan daya sistem
cron	V:923, I:997	248	menjalankan proses sesuai dengan jadwal di latar belakang dari daemon cron(8)
anacron	V:423, I:481	110	penjadwal perintah seperti cron untuk sistem yang tidak berjalan 24 jam sehari
at	V:69, I:91	158	at(1) atau batch(1): menjalankan pekerjaan pada waktu tertentu atau di bawah tingkat beban tertentu

Tabel 9.8: Daftar alat untuk memantau dan mengendalikan aktivitas program

Tip
Paket procps memberikan dasar-dasar pemantauan, pengendalian, dan memulai kegiatan program. Anda harus mempelajari semuanya.

9.4.1 Mencatat waktu eksekusi proses

Menampilkan waktu yang digunakan oleh proses yang dipanggil oleh perintah.

```
# time some_command >/dev/null
real    0m0.035s        # time on wall clock (elapsed real time)
user    0m0.000s        # time in user mode
sys     0m0.020s        # time in kernel mode
```

9.4.2 Prioritas penjadwalan

Suatu nilai nice digunakan untuk mengontrol prioritas penjadwalan untuk proses tersebut.

```
# nice  -19 top                                      # very nice
# nice --20 wodim -v -eject speed=2 dev=0,0 disk.img # very fast
```

nilai nice	prioritas penjadwalan
19	proses prioritas terendah (nice)
0	proses prioritas sangat tinggi untuk pengguna
-20	proses prioritas yang sangat tinggi untuk root (tidak-nice)

Tabel 9.9: Daftar nilai nice untuk prioritas penjadwalan

Terkadang nilai nice yang ekstrem lebih berbahaya daripada baik untuk sistem. Gunakan perintah ini dengan hati-hati.

9.4.3 Perintah ps

Perintah ps(1) pada sistem Debian mendukung fitur BSD dan SystemV dan membantu mengidentifikasi aktivitas proses secara statis.

gaya	perintah tipikal	fitur
BSD	`ps aux`	display %CPU %MEM
System V	`ps -efH`	tampilkan PPID

Tabel 9.10: Daftar gaya perintah ps

Untuk proses anak zombie (mati), Anda dapat membunuh mereka dengan id proses induk yang diidentifikasi di bidang "`PPID`".

Perintah pstree(1) menampilkan pohon proses.

9.4.4 Perintah top

top(1) pada sistem Debian memiliki fitur yang kaya dan membantu mengidentifikasi proses apa yang bertindak aneh secara dinamis.

Ini adalah program layar penuh interaktif. Anda bisa mendapatkan bantuan penggunaannya dengan menekan tombol "h" dan menghentikannya dengan menekan tombol "q".

9.4.5 Daftar berkas yang dibuka oleh suatu proses

Anda dapat mencantumkan semua berkas yang dibuka oleh proses dengan ID proses (PID), misalnya 1, dengan yang berikut ini.

```
$ sudo lsof -p 1
```

PID=1 biasanya adalah program `init`.

9.4.6 Menelusuri aktivitas program

Anda dapat melacak aktivitas program dengan strace(1), ltrace(1), atau xtrace(1) untuk panggilan dan sinyal sistem, panggilan pustaka, atau komunikasi antara klien dan server X11.

Anda dapat melacak panggilan sistem dari perintah `ls` sebagai berikut.

```
$ sudo strace ls
```

Tip
Gunakan skrip **strace-graph** yang ditemukan di **/usr/share/doc/strace/examples/** untuk membuat tampilan pohon yang bagus

9.4.7 Identifikasi proses menggunakan berkas atau soket

Anda juga dapat mengidentifikasi proses yang menggunakan berkas dengan fuser(1), misalnya untuk "`/var/log/mail.lo` dengan berikut ini.

```
$ sudo fuser -v /var/log/mail.log
                     USER        PID ACCESS COMMAND
/var/log/mail.log:   root       2946 F.... rsyslogd
```

Anda melihat bahwa berkas "`/var/log/mail.log`" dibuka untuk ditulis oleh perintah rsyslogd(8).

Anda juga dapat mengidentifikasi proses menggunakan soket dengan fuser(1), misalnya untuk "`smtp/tcp`" dengan yang berikut.

```
$ sudo fuser -v smtp/tcp
                     USER        PID ACCESS COMMAND
smtp/tcp:            Debian-exim   3379 F.... exim4
```

Sekarang Anda tahu sistem Anda menjalankan exim4(8) untuk menangani koneksi TCP ke port SMTP (25).

9.4.8 Mengulangi perintah dengan interval konstan

watch(1) mengeksekusi program berulang kali dengan interval konstan sambil menunjukkan keluarannya di layar penuh.

```
$ watch w
```

Ini menampilkan siapa yang masuk ke sistem, diperbarui setiap 2 detik.

9.4.9 Mengulangi perintah atas berkas

Ada beberapa cara untuk mengulangi perintah atas berkas yang cocok dengan beberapa kondisi, misalnya pencocokan pola glob "`*.ext`".

- Metode for-loop shell (lihat Bagian 12.1.4):

```
for x in *.ext; do if [ -f "$x"]; then command "$x" ; fi; done
```

- Kombinasi find(1) dan xargs(1):

```
find . -type f -maxdepth 1 -name '*.ext' -print0 | xargs -0 -n 1 command
```

- find(1) dengan opsi "`-exec`" dengan suatu perintah:

```
find . -type f -maxdepth 1 -name '*.ext' -exec command '{}' \;
```

- find(1) dengan opsi "`-exec`" dengan skrip shell pendek:

```
find . -type f -maxdepth 1 -name '*.ext' -exec sh -c "command '{}' && echo 'successful'" \;
```

Contoh di atas ditulis untuk memastikan penanganan yang tepat dari nama berkas aneh seperti yang berisi spasi. Lihat Bagian 10.1.5 untuk penggunaan `find` (1) pada tingkat lanjut.

9.4.10 Memulai program dari GUI

Untuk antarmuka baris perintah (CLI), program pertama dengan nama yang cocok yang ditemukan di direktori yang ditentukan dalam variabel lingkungan $PATH dijalankan. Lihat Bagian 1.5.3.

Untuk antarmuka pengguna grafis (GUI) yang sesuai dengan standar freedesktop.org, berkas `*.desktop` di direktori `/usr/share/applications/` memberikan atribut yang diperlukan untuk tampilan menu GUI dari setiap program. Setiap paket yang sesuai dengan sistem menu xdg Freedesktop.org memasang data menu yang disediakan oleh "*.desktop" di bawah "/usr/share/applications/". Lingkungan desktop modern yang sesuai dengan standar Freedesktop.org menggunakan data ini untuk menghasilkan menu mereka menggunakan paket xdg-utils. Lihat "/usr/share/doc/xdg-utils/README".

Misalnya, berkas `chromium.desktop` mendefinisikan atribut untuk "Peramban Web Chromium" seperti "Name" untuk nama program, "Exec" untuk path eksekusi program dan argumen, "Icon" untuk ikon yang digunakan, dll. (lihat Spesifikasi Entri Desktop) sebagai berikut:

```
[Desktop Entry]
Version=1.0
Name=Chromium Web Browser
GenericName=Web Browser
Comment=Access the Internet
Comment[fr]=Explorer le Web
Exec=/usr/bin/chromium %U
Terminal=false
X-MultipleArgs=false
Type=Application
Icon=chromium
Categories=Network;WebBrowser;
MimeType=text/html;text/xml;application/xhtml_xml;x-scheme-handler/http;x-scheme-handler/ ↩
    https;
StartupWMClass=Chromium
StartupNotify=true
```

Ini adalah deskripsi yang terlalu disederhanakan. Berkas `*.desktop` dipindai sebagai berikut.

Lingkungan desktop menetapkan variabel lingkungan $XDG_DATA_HOME dan $XDG_DATA_DIR. Misalnya, di bawah GNOME:

- $XDG_DATA_HOME tidak ditata. (Nilai baku `$HOME/.local/share` digunakan.)
- $XDG_DATA_DIRS diatur ke `/usr/share/gnome:/usr/local/share/:/usr/share/`.

Jadi direktori dasar (lihat Spesifikasi Direktori Dasar XDG) dan direktori `applications` adalah sebagai berikut.

- `$HOME/.local/share/` → `$HOME/.local/share/applications/`
- `/usr/share/gnome/` → `/usr/share/gnome/applications/`
- `/usr/local/share/` → `/usr/local/share/applications/`
- `/usr/share/` → `/usr/share/applications/`

Berkas `*.desktop` dipindai dalam direktori `applications` ini dalam urutan ini.

Tip
Entri menu GUI ubahan pengguna dapat dibuat dengan menambahkan berkas `*.desktop` di direktori `$HOME/.local/share/applications/`.

Tip
Baris "`Exec=...`" tidak diuraikan oleh shell. Gunakan perintah env(1) jika variabel lingkungan perlu diatur.

Tip
Demikian pula, jika berkas `*.desktop` dibuat di direktori `autostart` di bawah direktori dasar ini, program yang ditentukan dalam berkas `*.desktop` dijalankan secara otomatis ketika lingkungan desktop dimulai. Lihat Spesifikasi Autostart Aplikasi Desktop.

Tip
Demikian pula, jika berkas `*.desktop` dibuat di direktori `$HOME/Desktop` dan lingkungan Desktop dikonfigurasi untuk mendukung fitur peluncur ikon desktop, program yang ditentukan di dalamnya dijalankan setelah mengklik ikon. Harap dicatat bahwa nama sebenarnya dari direktori `$HOME/Desktop` tergantung pada lokal. Lihat xdg-user-dirs-update(1).

9.4.11 Menyesuaikan program yang akan dimulai

Beberapa program memulai program lain secara otomatis. Berikut adalah poin-poin pemeriksaan untuk menyesuaikan proses ini.

- Menu konfigurasi aplikasi:
  - Desktop GNOME: "Settings" → "Apps" → "Default Apps"
  - Desktop KDE: "Application Launcher" → "System Settings" → "Default Applications"
  - Peramban Iceweasel: "Sunting" → "Preferensi" → "Aplikasi"
  - mc(1): "`/etc/mc/mc.ext`"
- Variabel lingkungan seperti "`$BROWSER`", "`$EDITOR`", "`$VISUAL`", dan "`$PAGER`" (lihat environ(7))
- Sistem update-alternatives(1) untuk program seperti "`editor`", "`view`", "`x-www-browser`", "`gnome-www-browser`", dan"`www-browser`" (lihat Bagian 1.4.7)
- isi berkas "`~/.mailcap`" dan "`/etc/mailcap`" yang mengaitkan jenis MIME dengan program (lihat mailcap(5))
- Konten berkas "`~/.mime.types`" dan "`/etc/mime.types`" yang mengaitkan ekstensi nama berkas dengan tipe MIME (lihat run-mailcap(1))

Tip
update-mime(8) memperbarui berkas "`/etc/mailcap`" menggunakan berkas "`/etc/mailcap.order`" (lihat mailcap.order(5)).

Tip
Paket `debianutils` menyediakan sensible-browser(1), sensible-editor(1), dan sensible-pager(1) yang membuat keputusan yang masuk akal atas penyunting, pager, dan peramban web yang akan dipanggil. Saya sarankan Anda untuk membaca skrip shell ini.

Tip
Untuk menjalankan aplikasi konsol seperti `mutt` di bawah X sebagai aplikasi pilihan Anda, Anda mesti mencipta aplikasi GUI sebagai berikut dan mengatur "`/usr/local/bin/mutt-term`" sebagai aplikasi pilihan Anda yang akan dimulai seperti yang dijelaskan.

```
# cat /usr/local/bin/mutt-term <<EOF
#!/bin/sh
gnome-terminal -e "mutt \$@"
EOF
# chmod 755 /usr/local/bin/mutt-term
```

9.4.12 Membunuh sebuah proses

Gunakan kill(1) untuk membunuh (atau mengirim sinyal ke) suatu proses dengan ID proses.

Gunakan killall(1) atau pkill(1) untuk melakukan hal yang sama dengan nama perintah proses dan atribut lainnya.

nilai sinyal	nama sinyal	aksi	catatan
0	---	tidak ada sinyal yang dikirim (lihat kill(2))	memeriksa apakah proses sedang berjalan
1	SIGHUP	mengakhiri proses	terminal terputus (sinyal terputus)
2	SIGINT	mengakhiri proses	interupsi dari papan ketik (`CTRL-C`)
3	SIGQUIT	mengakhiri proses dan dump core	berhenti dari papan ketik (`CTRL-\`)
9	SIGKILL	mengakhiri proses	sinyal kill yang tidak dapat diblokir
15	SIGTERM	mengakhiri proses	sinyal pengakhiran yang dapat diblokir

Tabel 9.11: Daftar sinyal yang sering digunakan untuk perintah bunuh

9.4.13 Menjadwalkan tugas sekali

Jalankan perintah at(1) untuk menjadwalkan pekerjaan satu kali dengan yang berikut.

```
$ echo 'command -args'| at 3:40 monday
```

9.4.14 Menjadwalkan tugas secara teratur

Gunakan cron(8) untuk menjadwalkan tugas secara teratur. Lihat crontab(1) dan crontab(5).

Anda dapat menjadwalkan untuk menjalankan proses sebagai pengguna normal, misalnya `foo` dengan membuat berkas crontab(5) sebagai "`/var/spool/cron/crontabs/foo`" dengan perintah "`crontab -e`".

Berikut adalah contoh berkas crontab(5).

```
# use /usr/bin/sh to run commands, no matter what /etc/passwd says
SHELL=/bin/sh
# mail any output to paul, no matter whose crontab this is
MAILTO=paul
# Min Hour DayOfMonth Month DayOfWeek command (Day... are OR'ed)
# run at 00:05, every day
5  0  *  * *   $HOME/bin/daily.job >> $HOME/tmp/out 2>&1
# run at 14:15 on the first of every month -- output mailed to paul
15 14 1  * *   $HOME/bin/monthly
# run at 22:00 on weekdays(1-5), annoy Joe. % for newline, last % for cc:
0 22 *   * 1-5 mail -s "It's 10pm" joe%Joe,%%Where are your kids?%.%%
23 */2 1 2 *   echo "run 23 minutes after 0am, 2am, 4am ..., on Feb 1"
5  4 *   * sun echo "run at 04:05 every Sunday"
# run at 03:40 on the first Monday of each month
40 3 1-7 * *   [ "$(date +%a)" == "Mon" ] && command -args
```

Tip

Untuk sistem yang tidak berjalan terus menerus, pasang paket `anacron` untuk menjadwalkan perintah berkala pada interval yang ditentukan sedekat mungkin dengan waktu aktif mesin mengizinkan. Lihat anacron(8) dan anacrontab(5).

Tip

Untuk skrip pemeliharaan sistem terjadwal, Anda dapat menjalankannya secara berkala dari akun root dengan menempatkan skrip tersebut di "`/etc/cron.hourly/`", "/etc/cron.daily/","`/etc/cron.weekly/`", atau "`/etc/cron.monthly/`". Waktu eksekusi skrip ini dapat disesuaikan dengan "`/etc/crontab`" dan "`/etc/anacrontab`".

Systemd memiliki kemampuan tingkat rendah untuk menjadwalkan program untuk berjalan tanpa daemon `cron`. Misalnya, `/lib/systemd/system/apt-daily.timer` dan `/lib/systemd/system/apt-daily.service` mengatur kegiatan pengunduhan apt harian. Lihat systemd.timer(5).

9.4.15 Menjadwalkan tugas pada acara

Systemd dapat menjadwalkan program tidak hanya pada kejadian timer tetapi juga pada kejadian mount. Lihat Bagian 10.2.3.3 dan Bagian 10.2.3.2 untuk contoh.

9.4.16 Tombol Alt-SysRq

Menekan Alt-SysRq (PrtScr) diikuti oleh satu tombol melakukan keajaiban menyelamatkan kontrol sistem.

Lihat lebih lanjut pada Linux kernel user's and administrator's guide » Linux Magic System Request Key Hacks

tombol setelah Alt-SysRq	deskripsi tindakan
k	mati**k**an semua proses pada **konsol virtual saat ini** (SAK)
s	**s**elaraskan semua sistem berkas yang dikait untuk menghindari korupsi data
u	kait ulang ke hanya baca semua sistem berkas yang dikait (**u**mount)
r	memulihkan papan ketik dari mode **r**aw setelah X crash

Tabel 9.12: Daftar tombol perintah SAK yang terkenal

Tip
Dari terminal SSH dll., Anda dapat menggunakan fitur Alt-SysRq dengan menulis ke "`/proc/sysrq-trigger`". Misalnya, "`echo s > /proc/sysrq-trigger; echo u > /proc/sysrq-trigger`" dari prompt shell root menyelara**s**kan dan **u** melepas kait semua sistem berkas yang dikait.

Kernel Linux Debian amd64 saat ini (2021) memiliki `/proc/sys/kernel/sysrq=438=0b110110110`:

- 2 = 0x2 - memfungsikan kontrol tingkat log konsol (ON)
- 4 = 0x4 - memfungsikan kontrol papan ketik (SAK, unraw) (ON)
- 8 = 0x8 - memfungsikan dump debug proses dll (OFF)
- 16 = 0x10 - memfungsikan perintah sinkronisasi (ON)
- 32 = 0x20 - memfungsikan remount hanya baca (ON)
- 64 = 0x40 - memfungsikan pensinyalan proses (term, kill, oom-kill) (OFF)
- 128 = 0x80 - izinkan reboot/poweroff (ON)
- 256 = 0x100 - memungkinkan me-nice semua tugas RT (ON)

9.5 Tips pemeliharaan sistem

9.5.1 Siapa yang ada di sistem?

Anda dapat memeriksa siapa yang ada di sistem dengan berikut ini.

- who(1) menunjukkan siapa yang sedang log masuk.
- w(1) menunjukkan siapa yang sedang log masuk dan apa yang mereka lakukan.
- last(1) menampilkan daftar pengguna terakhir yang masuk.
- lastb(1) menunjukkan daftar pengguna buruk terakhir yang masuk.

Tip
"`/var/run/utmp`", dan"`/var/log/wtmp`" menyimpan informasi pengguna tersebut. Lihat login(1) dan utmp(5).

9.5.2 Memperingatkan semua orang

Anda dapat mengirim pesan ke semua orang yang masuk ke sistem dengan wall(1) dengan berikut ini.

```
$ echo "We are shutting down in 1 hour" | wall
```

9.5.3 Identifikasi perangkat keras

Untuk perangkat mirip PCI (AGP, PCI-Express, CardBus, ExpressCard, dll), lspci(8) (mungkin dengan opsi "`-nn`") adalah awal yang baik untuk identifikasi perangkat keras.

Atau, Anda dapat mengidentifikasi perangkat keras dengan membaca isi "`/proc/bus/pci/devices`" atau meramban pohon direktori di bawah "`/sys/bus/pci`" (lihat Bagian 1.2.12).

paket	popcon	ukuran	deskripsi
pciutils	V:264, I:994	289	Utilitas PCI Linux: lspci(8)
usbutils	V:90, I:892	322	Utilitas USB Linux: lsusb(8)
nvme-cli	V:27, I:36	2222	Utilitas NVMe untuk Linux: nvme(1)
pcmciautils	V:4.1, I:6.1	92	Utilitas PCMCIA untuk Linux: pccardctl(8)
scsitools	V:0.2, I:2.1	261	kumpulan alat untuk manajemen perangkat keras SCSI: lsscsi(8)
procinfo	V:0.4, I:5.8	149	informasi sistem yang diperoleh dari "`/proc`": lsdev(8)
lshw	V:12, I:88	971	informasi tentang konfigurasi perangkat keras: lshw(1)
discover	V:24, I:561	81	sistem identifikasi perangkat keras: discover(8)

Tabel 9.13: Daftar alat identifikasi perangkat keras

9.5.4 Konfigurasi perangkat keras

Meskipun sebagian besar konfigurasi perangkat keras pada sistem desktop GUI modern seperti GNOME dan KDE dapat dikelola melalui alat konfigurasi GUI yang menyertainya, adalah ide yang baik untuk mengetahui beberapa metode dasar untuk mengkonfigurasinya.

Untuk konfigurasi papan ketik dan terminal, lihat Bagian 8.2 dan Bagian 8.3.

Di sini, ACPI adalah kerangka kerja yang lebih baru untuk sistem manajemen daya daripada APM.

Tip
Penskalaan frekuensi CPU pada sistem modern diatur oleh modul kernel seperti `acpi_cpufreq`.

9.5.5 Waktu sistem dan perangkat keras

Berikut ini mengatur waktu sistem dan perangkat keras ke BB/TT jj:mm, AAYY.

```
# date MMDDhhmmCCYY
# hwclock --utc --systohc
# hwclock --show
```

Waktu biasanya ditampilkan pada waktu lokal pada sistem Debian tetapi waktu perangkat keras dan sistem biasanya menggunakan UTC (GMT).

Jika waktu perangkat keras diatur ke UTC, ubah pengaturan menjadi "`UTC=yes`" di "`/etc/default/rcS`".

Berikut konfigurasi ulang zona waktu yang digunakan oleh sistem Debian.

paket	popcon	ukuran	deskripsi
console-setup	V:81, I:976	422	Fonta konsol Linux dan utilitas keytable
keyd	V:1.5, I:1.7	1437	daemon pemetaan ulang tombol papan ketik memakai primitif masukan tingkat kernel (evdev, uinput)
keyd-application-mapper	V:0.1, I:0.08	34	daemon pemetaan ulang tombol papan ketik - pemeta ulang spesifik aplikasi
x11-xserver-utils	V:373, I:533	559	Utilitas server X: xset(1), xmodmap(1)
acpid	V:52, I:78	158	daemon untuk mengelola kejadian yang disampaikan oleh Advanced Configuration and Power Interface (ACPI)
acpi	V:6, I:73	49	utilitas untuk menampilkan informasi pada perangkat ACPI
sleepd	V:0.07, I:0.10	84	daemon untuk membuat laptop tidur selama tidak aktif
hdparm	V:110, I:200	246	optimasi akses hard disk (lihat Bagian 9.6.9)
smartmontools	V:242, I:270	2455	mengontrol dan memantau sistem penyimpanan menggunakan S.M.A.R.T.
setserial	V:3.0, I:4.9	104	koleksi alat untuk manajemen port serial
memtest86+	V:1, I:19	12483	koleksi alat untuk manajemen perangkat keras memori
scsitools	V:0.2, I:2.1	261	koleksi alat untuk manajemen perangkat keras SCSI
setcd	V:0.05, I:0.61	33	optimasi akses drive compact disc
big-cursor	I:0.67	26	kursor tetikus yang lebih besar untuk X

Tabel 9.14: Daftar alat konfigurasi perangkat keras

```
# dpkg-reconfigure tzdata
```

Jika Anda ingin memperbarui waktu sistem melalui jaringan, pertimbangkan untuk menggunakan layanan NTP dengan paket seperti `ntp`, `ntpdate`, dan `chrony`.

Tip
Di bawah systemd, gunakan `systemd-timesyncd` untuk sinkronisasi waktu jaringan sebagai gantinya. Lihat systemd-timesyncd(8).

Lihat yang berikut.

- HOWTO Mengelola Tanggal dan Waktu yang Akurat
- Proyek Layanan Publik NTP
- Paket `ntp-doc`

Tip
ntptrace(8) dalam paket `ntp` dapat melacak rantai server NTP kembali ke sumber utama.

9.5.6 Infrastruktur suara

Driver perangkat untuk kartu suara bagi Linux saat ini disediakan oleh Advanced Linux Sound Architecture (ALSA). ALSA menyediakan mode emulasi untuk Open Sound System (OSS) sebelumnya untuk kompatibilitas.

Perangkat lunak aplikasi dapat dikonfigurasi tidak hanya untuk mengakses perangkat suara secara langsung tetapi juga untuk mengaksesnya melalui beberapa sistem server suara standar. Saat ini, PulseAudio, JACK, dan PipeWire digunakan sebagai sistem server suara. Lihat halaman wiki Debian tentang Suara untuk situasi terbaru.

Biasanya ada mesin suara umum untuk setiap lingkungan desktop yang populer. Setiap mesin suara yang digunakan oleh aplikasi dapat memilih untuk terhubung ke server suara yang berbeda.

Tip
Gunakan "`cat /dev/urandom > /dev/audio`" atau speaker-test(1) untuk menguji speaker (^C untuk berhenti).

Tip
Jika Anda tidak bisa mendapatkan suara, speaker Anda mungkin terhubung ke keluaran yang dibisukan. Sistem suara modern memiliki banyak keluaran. alsamixer(1) dalam paket `alsa-utils` berguna untuk mengkonfigurasi pengaturan volume dan bisu.

paket	popcon	ukuran	deskripsi
`alsa-utils`	V:342, I:460	2700	utilitas untuk mengonfigurasi dan menggunakan ALSA
`oss-compat`	V:0.6, I:9.2	21	Kompatibilitas OSS di bawah ALSA mencegah kesalahan "`/dev/dsp tidak ditemukan`"
`pipewire`	V:326, I:370	136	server multimedia mesin pemrosesan audio dan video - paket meta
`pipewire-bin`	V:333, I:370	2251	server multimedia mesin pemrosesan audio dan video - program server dan CLI audio
`pipewire-alsa`	V:172, I:241	194	server multimedia mesin pemrosesan audio dan video - server audio untuk menggantikan ALSA
`pipewire-pulse`	V:298, I:347	53	server multimedia mesin pemrosesan audio dan video - server audio untuk menggantikan PulseAudio
`pulseaudio`	V:147, I:171	6616	server PulseAudio
`libpulse0`	V:443, I:579	984	pustaka klien PulseAudio
`jackd`	V:2, I:14	8	server JACK Audio Connection Kit. (JACK) (latensi rendah)
`libjack0`	V:1.8, I:8.6	328	pustaka JACK Audio Connection Kit. (JACK) (latensi rendah)
`libphonon4qt5-4`	V:18, I:42	572	Phonon: Mesin suara KDE

Tabel 9.15: Daftar paket suara

9.5.7 Menonaktifkan screen saver

Untuk menonaktifkan screen saver, gunakan perintah berikut.

lingkungan	perintah
Konsol Linux	`setterm -powersave off`
X Window (mematikan screensaver)	`xset s off`
X Window (menonaktifkan DPMS)	`xset -dpms`
X Window (konfigurasi GUI screen saver)	`xscreensaver-command -prefs`

Tabel 9.16: Daftar perintah untuk menonaktifkan screen saver

9.5.8 Menonaktifkan suara bip

Seseorang selalu dapat mencabut speaker PC untuk menonaktifkan suara bip. Menghapus modul kernel `pcspkr` melakukan ini untuk Anda.

Berikut ini mencegah program readline(3) yang digunakan oleh bash(1) untuk berbunyi bip saat menemui karakter peringatan (ASCII=7).

```
$ echo "set bell-style none">> ~/.inputrc
```

9.5.9 Penggunaan memori

Ada 2 sumber daya yang tersedia bagi Anda untuk mendapatkan situasi penggunaan memori.

- Pesan boot kernel dalam "`/var/log/dmesg`" berisi ukuran total yang tepat dari memori yang tersedia.
- free(1) dan top(1) menampilkan informasi tentang sumber daya memori pada sistem yang sedang berjalan.

Berikut adalah contohnya.

```
# grep '\] Memory' /var/log/dmesg
[    0.004000] Memory: 990528k/1016784k available (1975k kernel code, 25868k reserved, 931k ↩
    data, 296k init)
$ free -k
             total       used       free     shared    buffers     cached
Mem:        997184     976928      20256          0     129592     171932
-/+ buffers/cache:     675404     321780
Swap:      4545576          4    4545572
```

Anda mungkin bertanya-tanya "dmesg memberi tahu saya bebas 990 MB, dan free -k mengatakan 320 MB bebas. Lebih dari 600 MB hilang …".

Jangan khawatir tentang ukuran besar "`digunakan`" dan ukuran kecil "`bebas`" dalam baris "`Mem:`", tetapi baca yang di bawahnya (675404 dan 321780 dalam contoh di atas) dan rileks.

Untuk MacBook saya dengan DRAM 1GB=1048576k (sistem video mencuri sebagian dari ini), saya melihat yang berikut.

laporan	**ukuran**
Ukuran total dalam dmesg	1016784k = 1GB - 31792k
Bebas di dmesg	990528k
Total di bawah shell	997184k
Bebas di bawah shell	20256k (tetapi efektifnya 321780k)

Tabel 9.17: Daftar ukuran memori yang dilaporkan

9.5.10 Pemeriksaan keamanan dan integritas sistem

Pemeliharaan sistem yang buruk dapat mengekspos sistem Anda ke eksploitasi eksternal.

Untuk pemeriksaan keamanan dan integritas sistem, Anda harus mulai dengan yang berikut.

- Paket `debsums`, lihat debsum(1) dan Bagian 2.5.2.
- Paket `chkrootkit`, lihat chkrootkit(1).
- Keluarga paket `clamav`, lihat clamscan(1) dan freshclam(1).
- FAQ keamanan Debian.
- Manual Mengamankan Debian.

paket	popcon	ukuran	deskripsi
logcheck	V:4.4, I:5.4	120	daemon untuk mengirimkan anomali dalam berkas log sistem ke administrator
debsums	V:4, I:29	108	utilitas untuk memverifikasi berkas-berkas paket yang dipasang terhadap checksum MD5
chkrootkit	V:10, I:14	987	detektor rootkit
clamav	V:8, I:38	19877	utilitas anti-virus untuk Unix - antarmuka baris perintah
tiger	V:1.0, I:1.3	7800	melaporkan kerentanan keamanan sistem
tripwire	V:1.1, I:1.4	5060	pemeriksa integritas berkas dan direktori
john	V:1.1, I:7.3	469	alat pemecah kata sandi aktif
aide	V:2.2, I:2.6	328	Advanced Intrusion Detection Environment - biner statis
integrit	V:0.09, I:0.17	2759	program verifikasi integritas berkas
crack	V:0.06, I:0.55	151	program menebak kata sandi

Tabel 9.18: Daftar alat untuk pemeriksaan keamanan dan integritas sistem

Berikut adalah skrip sederhana untuk memeriksa salah izin berkas sehingga dapat ditulis oleh siapapun.

```
# find / -perm 777 -a \! -type s -a \! -type l -a \! \( -type d -a -perm 1777 \)
```



Perhatian
Karena paket debsums menggunakan checksum MD5 yang disimpan secara lokal, itu tidak dapat sepenuhnya dipercaya sebagai alat audit keamanan sistem terhadap serangan berbahaya.

9.6 Tips penyimpanan data

Mem-boot sistem Anda sebagai sistem Linux live (lihat Bagian 3.2.2 dan Bagian 3.2.3) memudahkan Anda mengonfigurasi ulang penyimpanan data pada sistem yang terinstal.

Catatan
Pernyataan tentang hard disk (HDD) berlaku untuk perangkat penyimpanan lain seperti SSD / USB flash drive / Kartu memori / ... . Ganti nama perangkat dalam contoh seperti /dev/sda dengan nama perangkat yang sesuai /dev/nvme0, /dev/mmcblk0, ...

Anda mungkin perlu umount(8) beberapa perangkat secara manual dari baris perintah sebelum beroperasi pada mereka jika mereka secara otomatis dipasang oleh sistem desktop GUI.

9.6.1 Penggunaan ruang disk

Penggunaan ruang disk dapat dievaluasi oleh program yang disediakan oleh paket mount, coreutils, dan xdu:

- mount(8) melaporkan semua sistem berkas yang dikait (= disk).
- df(1) melaporkan penggunaan ruang disk untuk sistem berkas.
- du(1) melaporkan penggunaan ruang disk untuk pohon direktori.

Tip
Anda dapat mengumpan keluaran dari du(8) ke xdu(1) untuk menghasilkan presentasi grafis dan interaktif dengan ”`du -k . |xdu`”, ”`sudo du -k -x / |xdu`”, dll.

9.6.2 Konfigurasi partisi disk

Untuk konfigurasi partisi disk, meskipun fdisk(8) telah dianggap standar, parted(8) layak mendapat perhatian. ”Data partisi disk”, ”tabel partisi”, ”peta partisi”, dan ”label disk” semuanya sinonim.

PC yang lebih tua menggunakan skema Master Boot Record (MBR) klasik untuk menyimpan data partisi disk di sektor pertama, yaitu, sektor LBA 0 (512 byte).

PC terbaru dengan Unified Extensible Firmware Interface (UEFI), termasuk Mac berbasis Intel, menggunakan skema GUID Partition Table (GPT) untuk menyimpan data partisi disk yang tidak terletak di sektor pertama.

Meskipun fdisk(8) telah menjadi standar untuk alat partisi disk, parted(8) menggantikannya.

paket	popcon	ukuran	deskripsi
`util-linux`	V:913, I:1000	4826	utilitas sistem lain-lain termasuk fdisk(8) dan cfdisk(8)
`parted`	V:449, I:569	126	GNU Parted program pengubahan ukuran partisi disk
`gparted`	V:12, I:83	2313	GNOME penyunting partisi berdasarkan `libparted`
`gdisk`	V:19, I:253	967	penyunting partisi untuk disk hibrid GPT/MBR
`kpartx`	V:15, I:24	78	program untuk membuat pemetaan perangkat bagi partisi

Tabel 9.19: Daftar paket manajemen partisi disk



Perhatian
Meskipun parted(8) mengklaim untuk membuat dan mengubah ukuran sistem berkas juga, lebih aman untuk melakukan hal-hal seperti itu menggunakan alat khusus terbaik yang dipelihara seperti mkfs(8) (mkfs.msdos(8), mkfs.ext2(8), mkfs.ext3(8), mkfs.ext4(8), …) dan resize2fs(8).

Catatan
Untuk beralih antara GPT dan MBR, Anda perlu menghapus beberapa blok pertama isi disk secara langsung (lihat Bagian 9.8.6) dan menggunakan ”`parted /dev/sdx mklabel gpt`” atau ”`parted /dev/sdx mklabel msdos`” untuk mengaturnya. Harap dicatat ”`msdos`” digunakan di sini untuk MBR.

9.6.3 Mengakses partisi menggunakan UUID

Meskipun konfigurasi ulang partisi atau urutan aktivasi media penyimpanan lepasan dapat menghasilkan nama yang berbeda untuk partisi, Anda dapat mengaksesnya secara konsisten. Ini juga membantu jika Anda memiliki beberapa disk dan BIOS/UEFI Anda tidak memberi mereka nama perangkat yang konsisten.

- mount(8) dengan opsi ”`-U`” dapat mengait perangkat blok menggunakan UUID, bukan menggunakan nama berkas seperti ”`/dev/sda3`”.
- ”`/etc/fstab`” (lihat fstab(5)) dapat menggunakan UUID.
- Boot loader (Bagian 3.1.2) dapat menggunakan UUID juga.

Tip
Anda dapat menjajaki UUID dari perangkat khusus blok dengan blkid(8).
Anda juga dapat menjajaki UUID dan informasi lainnya dengan "`lsblk -f`".

9.6.4 LVM2

LVM2 adalah manajer volume logis untuk kernel Linux. Dengan LVM2, partisi disk dapat dibuat pada volume logis bukan harddisk fisik.

LVM membutuhkan hal-hal berikut.

- dukungan device-mapper di kernel Linux (baku untuk kernel Debian)
- pustaka dukungan device-mapper ruang pengguna (paket `libdevmapper*`)
- perkakas LVM2 ruang pengguna (paket `lvm2`)

Silakan mulai belajar LVM2 dari manpages berikut.

- lvm(8): Dasar-dasar mekanisme LVM2 (daftar semua perintah LVM2)
- lvm.conf(5): Berkas konfigurasi untuk LVM2
- lvs(8): Melaporkan informasi tentang volume logis
- vgs(8): Melaporkan informasi tentang grup volume
- pvs(8): Melaporkan informasi tentang volume fisik

9.6.5 Konfigurasi sistem berkas

Untuk sistem berkas ext4, paket `e2fsprogs` menyediakan yang berikut.

- mkfs.ext4(8) untuk membuat sistem berkas ext4 baru
- fsck.ext4(8) untuk memeriksa dan memperbaiki sistem berkas ext4 yang ada
- tune2fs(8) untuk mengkonfigurasi superblok dari sistem berkas ext4
- debugfs(8) untuk debug sistem berkas ext4 interaktif. (Ini memiliki perintah `undel` untuk memulihkan berkas yang dihapus.)

Perintah mkfs(8) dan fsck(8) disediakan oleh paket `e2fsprogs` sebagai front-end untuk berbagai program yang bergantung kepada sistem berkas (`mkfs.fstype` dan `fsck.fstype`). Untuk sistem berkas ext4, mereka adalah mkfs.ext4(8) dan fsck.ext4(8) (mereka di-symlinked mke2fs(8) dan e2fsck(8)).

Perintah serupa tersedia untuk setiap sistem berkas yang didukung oleh Linux.

Tip
Sistem berkas Ext4 adalah sistem berkas baku untuk sistem Linux dan sangat disarankan untuk menggunakannya kecuali Anda memiliki beberapa alasan khusus untuk tidak melakukannya.
Status Btrfs dapat ditemukan di wiki Debian tentang btrfs dan wiki kernel.org tentang btrfs. Hal ini diharapkan menjadi sistem berkas baku berikutnya setelah sistem berkas ext4.
Beberapa alat memungkinkan akses ke sistem berkas tanpa dukungan kernel Linux (lihat Bagian 9.8.2).

paket	popcon	ukuran	deskripsi
e2fsprogs	V:821, I:997	1560	utilitas untuk sistem berkas ext2/ext3/ext4
btrfs-progs	V:52, I:79	5255	utilitas untuk sistem berkas Btrfs
reiserfsprogs	V:11, I:21	473	utilitas untuk sistem berkas Reiserfs
zfsutils-linux	V:32, I:33	1905	utilitas untuk sistem berkas OpenZFS
dosfstools	V:265, I:568	310	utilitas untuk sistem berkas FAT. (Microsoft: MS-DOS, Windows)
exfatprogs	V:36, I:466	580	utilitas untuk sistem berkas exFAT yang dikelola oleh Samsung.
exfat-fuse	V:2, I:40	73	driver sistem berkas exFAT (Microsoft) baca/tulis untuk FUSE.
xfsprogs	V:42, I:84	4651	utilitas untuk sistem berkas XFS. (SGI: IRIX)
ntfs-3g	V:208, I:514	1507	baca/tulis sistem berkas NTFS (Microsoft: Windows NT, …) driver untuk FUSE.
jfsutils	V:0.4, I:6.7	1104	utilitas untuk sistem berkas JFS. (IBM: AIX, OS/2)
xorriso	V:15, I:65	378	utilitas untuk sistem berkas ISO-9660 dan penulisan CD/DVD dari libburnia
wodim	V:7, I:83	898	alat penulisan CD/DVD baris perintah dari cdrkit
genisoimage	V:17, I:153	1567	alat sistem berkas ISO-9660 baris perintah dari cdrkit
reiser4progs	V:0.1, I:1.1	1367	utilitas untuk sistem berkas Reiser4
hfsprogs	V:0.3, I:3.1	394	utilitas untuk sistem berkas HFS dan HFS Plus. (Apple: Mac OS)
zerofree	V:7, I:116	30	program mengisi dengan nol blok bebas dari sistem berkas ext2/3/4

Tabel 9.20: Daftar paket manajemen sistem berkas

9.6.6 Pembuatan sistem berkas dan pemeriksaan integritas

Perintah mkfs(8) membuat sistem berkas pada sistem Linux. Perintah fsck(8) menyediakan pemeriksaan integritas dan perbaikan sistem berkas pada sistem Linux.

Debian sekarang membakukan untuk tidak ada `fsck` periodik setelah pembuatan sistem berkas.



Perhatian
Umumnya tidak aman untuk menjalankan `fsck` pada **sistem berkas yang dikait**.

Tip
Anda dapat menjalankan perintah fsck(8) dengan aman pada semua sistem berkas termasuk sistem berkas root pada saat reboot dengan mengatur "`enable_periodic_fsck`" di "`/etc/mke2fs.conf`" dan cacah mount maks ke 0 menggunakan "`tune2fs -c0 /dev/`*`nama_partisi`*". Lihat mke2fs.conf(5) dan tune2fs(8).
Periksa berkas di "`/var/log/fsck/`" untuk hasil perintah fsck(8) yang dijalankan dari skrip boot.

9.6.7 Optimalisasi sistem berkas dengan opsi mount

Konfigurasi sistem berkas statis dasar diberikan oleh "`/etc/fstab`". Misalnya,

```
«file system»                    «mount point» «type» «options»    «dump» «pass»
proc                                      /proc proc   defaults          0 0
UUID=709cbe4c-80c1-56db-8ab1-dbce3146d2f7 /     ext4   errors=remount-ro 0 1
UUID=817bae6b-45d2-5aca-4d2a-1267ab46ac23 none  swap   sw                0 0
/dev/scd0                          /media/cdrom0  udf,iso9660 user,noauto  0 0
```

Tip
UUID (lihat Bagian 9.6.3) dapat digunakan untuk mengidentifikasi perangkat blok alih-alih nama perangkat blok normal seperti "`/dev/sda1`", "`/dev/sda2`", ...

Sejak Linux 2.6.30, baku kernel adalah perilaku yang disediakan oleh opsi "`relatime`".

Lihat fstab(5) dan mount(8).

9.6.8 Optimasi sistem berkas melalui superblok

Karakteristik sistem berkas dapat dioptimalkan melalui superblok menggunakan perintah tune2fs(8).

- Eksekusi "`sudo tune2fs -l /dev/sda1`" menampilkan isi superblok sistem berkas pada "`/dev/sda1`".
- Eksekusi "`sudo tune2fs -c 50 /dev/sda1`" mengubah frekuensi pemeriksaan sistem bermas (eksekusi `fsck` selama boot-up) ke setiap 50 boot pada "`/dev/sda1`".
- Eksekusi "`sudo tune2fs -j /dev/sda1`" menambahkan kemampuan jurnal ke sistem berkas, yaitu konversi sistem berkas dari ext2 ke ext3 pada "`/dev/sda1`". (Lakukan ini pada sistem berkas yang tidak dikait.)
- Eksekusi "`sudo tune2fs -O extents,uninit_bg,dir_index /dev/sda1 && fsck -pf /dev/sda1`" mengubahnya dari ext3 menjadi ext4 pada "`/dev/sda1`". (Lakukan ini pada sistem berkas yang tidak dikait.)

Tip
Meskipun bernama seperti itu, tune2fs(8) bekerja tidak hanya pada sistem berkas ext2 tetapi juga pada sistem berkas ext3 dan ext4.

9.6.9 Optimalisasi hard disk



Awas
Silakan periksa perangkat keras Anda dan baca halaman man hdparm(8) sebelum bermain dengan konfigurasi hard disk karena ini mungkin cukup berbahaya bagi integritas data.

Anda dapat menguji kecepatan akses disk dari hard disk, misalnya "`/dev/sda`", dengan "`hdparm -tT /dev/sda`". Untuk beberapa hard disk yang terhubung dengan (E)IDE, Anda dapat mempercepatnya dengan "`hdparm -q -c3 -d1 -u1 -m16 /dev/sda`" dengan mengaktifkan "(E)IDE 32-bit I/O support", memfungsikan "using_dma flag", pengaturan "interrupt-unmask flag", dan pengaturan "multiple 16 sector I/O" (berbahaya!).

Anda dapat menguji fitur singgahan tulis dari hard disk, misalnya "`/dev/sda`", dengan "`hdparm -W /dev/sda`". Anda dapat menonaktifkan fitur singgahan tulisnya dengan "`hdparm -W 0 /dev/sda`".

Anda mungkin dapat membaca CDROM yang dicetak dengan buruk pada drive CD-ROM berkecepatan tinggi modern dengan memperlambatnya memakai "`setcd -x 2`".

9.6.10 Optimasi solid state drive

Solid state drive (SSD) terdeteksi secara otomatis sekarang.

Mengurangi akses disk yang tidak perlu untuk mencegah disk aus dengan mengait "`tmpfs`" pada path data volatil di `/etc/fstab`.

9.6.11 Menggunakan SMART untuk memprediksi kegagalan hard disk

Anda dapat memantau dan mencatat hard disk Anda yang patuh ke SMART dengan daemon smartd(8).

1. Aktifkan fitur SMART di BIOS.
2. Instal paket `smartmontools`.
3. Identifikasi hard disk drive Anda dengan menampilkan daftar mereka memakai df(1).
   - Mari kita asumsikan hard disk drive yang akan dipantau sebagai "`/dev/sda`".
4. Periksa keluaran "`smartctl -a /dev/sda`" untuk melihat apakah fitur SMART benar-benar diaktifkan.
   - Jika tidak, aktifkan dengan "`smartctl -s on -a /dev/sda`".
5. Fungsikan daemon smartd(8) agar berjalan dengan cara berikut.
   - jadikan bukan komentar "`start_smartd=yes`" dalam berkas "`/etc/default/smartmontools`".
   - jalankan ulang daemon smartd(8) dengan "`sudo systemctl restart smartmontools`".

Tip
Daemon smartd(8) dapat disesuaikan dengan berkas `/etc/smartd.conf` termasuk cara diberitahu tentang peringatan.

9.6.12 Menentukan direktori penyimpanan sementara melalui $TMPDIR

Aplikasi biasanya membuat berkas sementara di bawah direktori penyimpanan sementara "`/tmp`". Jika "`/tmp`" tidak menyediakan ruang yang cukup, Anda dapat menentukan direktori penyimpanan sementara tersebut melalui variabel `$TMPDIR` untuk program yang berperilaku baik.

9.6.13 Perluasan ruang penyimpanan yang dapat digunakan melalui LVM

Untuk partisi yang dibuat pada Logical Volume Manager (LVM) (fitur Linux) pada waktu instalasi, mereka dapat diubah ukurannya dengan mudah dengan menggabungkan extent ke mereka atau memotong extent dari mereka atas beberapa perangkat penyimpanan tanpa konfigurasi ulang sistem utama.

9.6.14 Perluasan ruang penyimpanan yang dapat digunakan dengan memasang partisi lain

Jika Anda memiliki partisi kosong (misalnya, "`/dev/sdx`"), Anda dapat memformatnya dengan mkfs.ext4(1) dan mount(8) ke direktori di mana Anda membutuhkan lebih banyak ruang. (Anda perlu menyalin konten data asli.)

```
$ sudo mv work-dir old-dir
$ sudo mkfs.ext4 /dev/sdx
$ sudo mount -t ext4 /dev/sdx work-dir
$ sudo cp -a old-dir/* work-dir
$ sudo rm -rf old-dir
```

Tip
Anda dapat mengait berkas image disk kosong (lihat Bagian 9.7.5) sebagai perangkat loop (lihat Bagian 9.7.3). Penggunaan disk yang sebenarnya tumbuh dengan data aktual yang disimpan.

9.6.15 Perluasan ruang penyimpanan yang dapat digunakan dengan mengait-bind direktori lain

Jika Anda memiliki direktori kosong (misalnya,"`/path/ke/dir-kosong`") pada partisi lain dengan ruang yang dapat digunakan, Anda dapat me-mount(8) itu dengan opsi "`--bind`" ke direktori (misalnya, "`dir-kerja`") di mana Anda membutuhkan lebih banyak ruang.

```
$ sudo mount --bind /path/to/emp-dir work-dir
```

9.6.16 Perluasan ruang penyimpanan yang dapat digunakan dengan mengait overlay direktori lain

Jika Anda memiliki ruang yang dapat digunakan di partisi lain (misalnya, "`/path/ke/kosong`" dan "`/path/ke/kerja`"), Anda dapat membuat direktori di dalamnya dan menumpuknya ke direktori lama (misalnya, "`/path/ke/lama`") di mana Anda memerlukan ruang menggunakan OverlayFS untuk kernel Linux 3.18 atau yang lebih baru (Debian Stretch 9.0 atau yang lebih baru).

```
$ sudo mount -t overlay overlay \
  -olowerdir=/path/to/old-dir,upperdir=/path/to/empty,workdir=/path/to/work
```

Di sini, "`/path/ke/kosong`" dan "`/path/ke/kerja`" harus berada di partisi yang mendukung RW untuk menulis di "`/path/ke/lama`".

9.6.17 Perluasan ruang penyimpanan yang dapat digunakan menggunakan symlink



Perhatian
Ini adalah metode yang usang. Beberapa perangkat lunak mungkin tidak berfungsi dengan baik dengan "symlink ke direktori". Sebaliknya, gunakan pendekatan "mounting" yang dijelaskan di atas.

Jika Anda memiliki direktori kosong (misalnya, "`/path/ke/dir-kosong`") di partisi lain dengan ruang yang dapat digunakan, Anda dapat membuat symlink ke direktori dengan ln(8).

```
$ sudo mv work-dir old-dir
$ sudo mkdir -p /path/to/emp-dir
$ sudo ln -sf /path/to/emp-dir work-dir
$ sudo cp -a old-dir/* work-dir
$ sudo rm -rf old-dir
```



Awas
Jangan gunakan "symlink ke direktori" untuk direktori yang dikelola oleh sistem seperti "`/opt`". Symlink semacam itu mungkin ditimpa ketika sistem ditingkatkan.

9.7 Image disk

Di sini, kita membahas manipulasi image disk.

9.7.1 Membuat berkas image disk

Berkas image disk, "`disk.img`", dari perangkat yang tidak dipasang, misalnya drive SCSI atau serial ATA kedua "`/dev/sdb`", dapat dibuat dengan salah satu cara berikut.

```
# dd if=/dev/sdb of=disk.img; sync
```

```
# cp /dev/sdb disk.img ; sync
```

```
# cat /dev/sdb > disk.img ; sync
```

Image disk dari master boot record (MBR) PC tradisional (lihat Bagian 9.6.2) yang berada di sektor pertama pada disk IDE utama dapat dibuat menggunakan dd(1) dengan yang berikut.

```
# dd if=/dev/sda of=mbr.img bs=512 count=1
# dd if=/dev/sda of=mbr-nopart.img bs=446 count=1
# dd if=/dev/sda of=mbr-part.img skip=446 bs=1 count=66
```

- "`mbr.img`": MBR dengan tabel partisi
- "`mbr-nopart.img`": MBR tanpa tabel partisi
- "`mbr-part.img`": Tabel partisi MBR saja

Jika Anda membuat image partisi disk dari disk asli, ganti "`/dev/sda`" dengan "`/dev/sda1`" dll.

9.7.2 Menulis secara langsung ke disk

Berkas image disk, "`disk.img`" dapat ditulis ke perangkat yang tidak dipasang, misalnya drive SCSI kedua "`/dev/sdb`" dengan ukuran yang sesuai, dengan salah satu cara berikut.

```
# dd if=disk.img of=/dev/sdb ; sync
```

```
# cp disk.img /dev/sdb ; sync
```

```
# cat disk.img >/dev/sdb ; sync
```

Demikian pula, berkas image partisi disk, "`partition.img`" dapat ditulis ke partisi yang tidak dikait, misalnya, partisi pertama dari drive SCSI kedua "`/dev/sdb1`" dengan ukuran yang cocok, memakai yang berikut ini.

```
# dd if=partition.img of=/dev/sdb1 ; sync
```

9.7.3 Mengait berkas image disk

Image disk "`partition.img`" yang berisi image partisi tunggal dapat dikait dan dilepas kait dengan menggunakan perangkat loop sebagai berikut.

```
# losetup --show -f partition.img
/dev/loop0
# mkdir -p /mnt/loop0
# mount -t auto /dev/loop0 /mnt/loop0
...hack...hack...hack
# umount /dev/loop0
# losetup -d /dev/loop0
```

Ini dapat disederhanakan sebagai berikut.

```
# mkdir -p /mnt/loop0
# mount -t auto -o loop partition.img /mnt/loop0
...hack...hack...hack
# umount partition.img
```

Setiap partisi image disk "`disk.img`" yang berisi beberapa partisi dapat dikait dengan menggunakan perangkat loop.

```
# losetup --show -f -P disk.img
/dev/loop0
# ls -l /dev/loop0*
brw-rw---- 1 root disk   7,  0 Apr  2 22:51 /dev/loop0
brw-rw---- 1 root disk 259, 12 Apr  2 22:51 /dev/loop0p1
brw-rw---- 1 root disk 259, 13 Apr  2 22:51 /dev/loop0p14
brw-rw---- 1 root disk 259, 14 Apr  2 22:51 /dev/loop0p15
# fdisk -l /dev/loop0
Disk /dev/loop0: 2 GiB, 2147483648 bytes, 4194304 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 6A1D9E28-C48C-2144-91F7-968B3CBC9BD1

Device         Start     End Sectors  Size Type
/dev/loop0p1  262144 4192255 3930112  1.9G Linux root (x86-64)
/dev/loop0p14   2048    8191    6144    3M BIOS boot
/dev/loop0p15   8192  262143  253952  124M EFI System

Partition table entries are not in disk order.
# mkdir -p /mnt/loop0p1
# mkdir -p /mnt/loop0p15
# mount -t auto /dev/loop0p1 /mnt/loop0p1
# mount -t auto /dev/loop0p15 /mnt/loop0p15
# mount |grep loop
/dev/loop0p1 on /mnt/loop0p1 type ext4 (rw,relatime)
/dev/loop0p15 on /mnt/loop0p15 type vfat (rw,relatime,fmask=0002,dmask=0002,allow_utime ↩
    =0020,codepage=437,iocharset=ascii,shortname=mixed,utf8,errors=remount-ro)
...hack...hack...hack
# umount /dev/loop0p1
# umount /dev/loop0p15
# losetup -d /dev/loop0
```

Atau, efek serupa dapat dilakukan dengan menggunakan device mapper yang dibuat oleh kpartx(8) dari paket `kpartx` sebagai berikut.

```
# kpartx -a -v disk.img
add map loop0p1 (253:0): 0 3930112 linear 7:0 262144
add map loop0p14 (253:1): 0 6144 linear 7:0 2048
add map loop0p15 (253:2): 0 253952 linear 7:0 8192
# fdisk -l /dev/loop0
Disk /dev/loop0: 2 GiB, 2147483648 bytes, 4194304 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 6A1D9E28-C48C-2144-91F7-968B3CBC9BD1

Device         Start     End Sectors  Size Type
/dev/loop0p1  262144 4192255 3930112  1.9G Linux root (x86-64)
/dev/loop0p14   2048    8191    6144    3M BIOS boot
/dev/loop0p15   8192  262143  253952  124M EFI System
```

```
Partition table entries are not in disk order.
# ls -l /dev/mapper/
total 0
crw------- 1 root root 10, 236 Apr  2 22:45 control
lrwxrwxrwx 1 root root       7 Apr  2 23:19 loop0p1 -> ../dm-0
lrwxrwxrwx 1 root root       7 Apr  2 23:19 loop0p14 -> ../dm-1
lrwxrwxrwx 1 root root       7 Apr  2 23:19 loop0p15 -> ../dm-2
# mkdir -p /mnt/loop0p1
# mkdir -p /mnt/loop0p15
# mount -t auto /dev/mapper/loop0p1 /mnt/loop0p1
# mount -t auto /dev/mapper/loop0p15 /mnt/loop0p15
# mount |grep loop
/dev/loop0p1 on /mnt/loop0p1 type ext4 (rw,relatime)
/dev/loop0p15 on /mnt/loop0p15 type vfat (rw,relatime,fmask=0002,dmask=0002,allow_utime ↩
    =0020,codepage=437,iocharset=ascii,shortname=mixed,utf8,errors=remount-ro)
...hack...hack...hack
# umount /dev/mapper/loop0p1
# umount /dev/mapper/loop0p15
# kpartx -d disk.img
```

9.7.4 Membersihkan berkas image disk

Sebuah berkas image disk, ”`disk.img`” dapat dibersihkan dari semua berkas yang dihapus menjadi image sparse bersih ”`new.img`” dengan berikut.

```
# mkdir old; mkdir new
# mount -t auto -o loop disk.img old
# dd bs=1 count=0 if=/dev/zero of=new.img seek=5G
# mount -t auto -o loop new.img new
# cd old
# cp -a --sparse=always ./ ../new/
# cd ..
# umount new.img
# umount disk.img
```

Jika ”`disk.img`” berada dalam ext2, ext3, atau ext4, Anda juga dapat menggunakan zerofree(8) dari paket `zerofree` sebagai berikut.

```
# losetup --show -f disk.img
/dev/loop0
# zerofree /dev/loop0
# cp --sparse=always disk.img new.img
# losetup -d /dev/loop0
```

9.7.5 Membuat berkas image disk kosong

Image disk kosong ”`disk.img`” yang dapat tumbuh hingga 5GiB dapat dibuat menggunakan dd(1) sebagai berikut.

```
$ dd bs=1 count=0 if=/dev/zero of=disk.img seek=5G
```

Alih-alih menggunakan dd(1), fallocate(8) yang khusus dapat digunakan di sini.

Anda dapat membuat sistem berkas ext4 pada image disk ”`disk.img`” ini menggunakan perangkat loop sebagai berikut.

```
# losetup --show -f disk.img
/dev/loop0
# mkfs.ext4 /dev/loop0
...hack...hack...hack
# losetup -d /dev/loop0
$ du  --apparent-size -h disk.img
5.0G  disk.img
$ du -h disk.img
83M disk.img
```

Untuk "`disk.img`", ukuran berkasnya adalah 5,0 GiB dan penggunaan disk yang sebenarnya hanya 83MiB. Perbedaan ini dimungkinkan karena ext4 dapat menyimpan berkas sparse.

Tip
Penggunaan disk sebenarnya dari berkas sparse tumbuh dengan data yang ditulis ke sana.

Menggunakan operasi serupa pada perangkat yang dibuat oleh perangkat loop atau device mapper sebagai Bagian 9.7.3, Anda dapat mempartisi image disk "`disk.img`" ini menggunakan parted(8) atau fdisk(8), dan dapat membuat sistem berkas di atasnya menggunakan mkfs.ext4(8), mkswap(8), dll.

9.7.6 Membuat berkas image ISO9660

Tip
Baik genisoimage(1) yang disediakan oleh cdrkit maupun xorrisofs(1) yang disediakan oleh Libburnia berbagi sintaks perintah yang sama kecuali nama perintahnya.

Berkas image ISO9660, "`cd.iso`", dari pohon direktori sumber di "`direktori_sumber`" dapat dibuat menggunakan genisoimage(1) yang disediakan oleh cdrkit dengan berikut.

```
#  genisoimage -r -J -T -V volume_id -o cd.iso source_directory
```

Demikian pula, berkas image ISO9660 yang dapat di-boot, "`cdboot.iso`", dapat dibuat dari pohon direktori mirip `debian-installer` di "`direktori_sumber`" dengan berikut.

```
#  genisoimage -r -o cdboot.iso -V volume_id \
   -b isolinux/isolinux.bin -c isolinux/boot.cat \
   -no-emul-boot -boot-load-size 4 -boot-info-table source_directory
```

Di sini boot loader Isolinux (lihat Bagian 3.1.2) digunakan untuk boot.

Anda dapat menghitung nilai md5sum dan membuat image ISO9660 langsung dari perangkat CD-ROM sebagai berikut.

```
$ isoinfo -d -i /dev/cdrom
CD-ROM is in ISO 9660 format
...
Logical block size is: 2048
Volume size is: 23150592
...
# dd if=/dev/cdrom bs=2048 count=23150592 conv=notrunc,noerror | md5sum
# dd if=/dev/cdrom bs=2048 count=23150592 conv=notrunc,noerror > cd.iso
```



Awas
Anda harus hati-hati menghindari kutu read ahead sistem berkas ISO9660 Linux seperti di atas untuk mendapatkan hasil yang tepat.

9.7.7 Menulis secara langsung ke CD/DVD-R/RW

Tip
DVD hanyalah CD berukuran besar bagi wodim(1) yang disediakan oleh cdrkit dan xorrecord(1) yang disediakan oleh Libburnia. Perintah-perintah ini berbagi sintaks perintah yang sama kecuali nama perintahnya.

Anda dapat menemukan perangkat yang dapat digunakan dengan berikut ini.

```
# wodim --devices
```

Kemudian CD-R kosong dimasukkan ke drive CD, dan berkas image ISO9660, "`cd.iso`" ditulis ke perangkat ini, misalnya, "`/dev/sda`", menggunakan wodim(1) dengan berikut ini.

```
# wodim -v -eject dev=/dev/sda cd.iso
```

Jika CD-RW digunakan alih-alih CD-R, lakukan ini sebagai gantinya dengan yang berikut ini.

```
# wodim -v -eject blank=fast dev=/dev/sda cd.iso
```

Tip
Jika sistem desktop Anda mengait CD secara otomatis, lepas kaitnya dengan "`sudo umount /dev/sda`" dari konsol sebelum menggunakan wodim(1).

9.7.8 Mengait berkas image ISO9660

Jika "`cd.iso`" berisi image ISO9660, maka berikut ini secara manual mengaitnya ke "`/cdrom`".

```
# mount -t iso9660 -o ro,loop cd.iso /cdrom
```

Tip
Sistem desktop modern dapat mengait media lepasan seperti CD terformat ISO9660 secara otomatis (lihat Bagian 10.1.7).

9.8 Data biner

Di sini, kita membahas manipulasi langsung data biner pada media penyimpanan.

9.8.1 Melihat dan menyunting data biner

Metode tampilan data biner yang paling dasar adalah dengan menggunakan perintah "`od -t x1`".

Tip
HEX digunakan sebagai akronim untuk format heksadesimal dengan basis 16. OCTAL adalah untuk format oktal dengan basis 8. ASCII adalah untuk American Standard Code for Information Interchange, yaitu kode teks bahasa Inggris normal. EBCDIC adalah untuk Extended Binary Coded Decimal Interchange Code yang digunakan pada sistem operasi mainframe IBM.

paket	popcon	ukuran	deskripsi
coreutils	V:912, I:1000	17994	paket dasar yang memiliki od(1) untuk mencurah berkas (HEX, ASCII, OCTAL, …)
bsdmainutils	V:4, I:121	17	paket utilitas yang memiliki hd(1) untuk mencurah berkas (HEX, ASCII, OCTAL, …)
hexedit	V:0.8, I:7.5	70	editor dan penampil biner (HEKSA, ASCII)
bless	V:0.1, I:1.2	924	editor heksadesimal berfitur lengkap (GNOME)
okteta	V:1, I:11	1591	editor heksadesimal berfitur lengkap (KDE4)
ncurses-hexedit	V:0.1, I:1.1	130	editor dan penampil biner (HEKSA, ASCII, EBCDIC)
beav	V:0.04, I:0.26	137	editor dan penampil biner (HEKSA, ASCII, EBCDIC, OKTAL, …)

Tabel 9.21: Daftar paket yang menampilkan dan menyunting data biner

9.8.2 Memanipulasi berkas tanpa mengait disk

Ada alat untuk membaca dan menulis berkas tanpa mengait disk.

paket	popcon	ukuran	deskripsi
mtools	V:7, I:52	400	utilitas untuk berkas MSDOS tanpa mengaitnya
hfsutils	V:0.1, I:2.5	178	utilitas untuk berkas HFS dan HFS+ tanpa mengait mereka

Tabel 9.22: Daftar paket untuk memanipulasi berkas tanpa mengait disk

9.8.3 Redundansi data

Sistem RAID perangkat lunak yang ditawarkan oleh kernel Linux menyediakan redundansi data di tingkat sistem berkas kernel untuk mencapai tingkat keandalan penyimpanan yang tinggi.

Ada alat untuk menambahkan redundansi data ke berkas di tingkat program aplikasi untuk mencapai tingkat keandalan penyimpanan yang tinggi juga.

paket	popcon	ukuran	deskripsi
par2	V:9, I:116	330	Parity Archive Volume Set, untuk memeriksa dan memperbaiki berkas
dvdisaster	V:0.1, I:1.1	1422	perlindungan kehilangan data/goresan/penuaan untuk media CD/DVD
dvbackup	V:0.01, I:0.03	413	alat cadangan menggunakan camcorder MiniDV (menyediakan rsbep(1))

Tabel 9.23: Daftar alat untuk menambahkan redundansi data ke berkas

9.8.4 Pemulihan berkas data dan analisis forensik

Ada alat untuk pemulihan berkas data dan analisis forensik.

Tip
Anda dapat membatalkan penghapusan berkas pada sistem berkas ext2 menggunakan perintah `list_deleted_inodes` dan `undel` dari debugfs(8) dalam paket `e2fsprogs`.

paket	popcon	ukuran	deskripsi
testdisk	V:2, I:25	528	utilitas untuk pemindaian partisi dan pemulihan disk
magicrescue	V:0.2, I:1.7	257	utilitas untuk memulihkan berkas dengan mencari byte ajaib
scalpel	V:0.2, I:2.2	89	frugal, pemahat berkas berkinerja tinggi
myrescue	V:0.2, I:1.9	81	menyelamatkan data dari harddisk yang rusak
extundelete	V:0.5, I:7.3	152	utilitas untuk membatalkan penghapusan berkas pada sistem berkas ext3/4
ext4magic	V:0.2, I:3.5	235	utilitas untuk membatalkan penghapusan berkas pada sistem berkas ext3/4
ext3grep	V:0.2, I:1.8	299	alat untuk membantu memulihkan berkas yang dihapus pada sistem berkas ext3
scrounge-ntfs	V:0.1, I:1.6	49	program pemulihan data untuk sistem berkas NTFS
gzrt	V:0.02, I:0.43	34	toolkit pemulihan gzip
sleuthkit	V:3, I:25	1119	alat untuk analisis forensik. (Sleuthkit)
autopsy	V:0.1, I:1.1	1026	antarmuka grafis ke SleuthKit
foremost	V:0.3, I:3.8	101	aplikasi forensik untuk memulihkan data
guymager	V:0.08, I:0.71	1049	alat pembuatan citra forensik berbasis Qt
dcfldd	V:0.3, I:2.6	113	versi dd yang disempurnakan untuk forensik dan keamanan

Tabel 9.24: Daftar paket untuk pemulihan berkas data dan analisis forensik

9.8.5 Memecah sebuah berkas besar menjadi berkas-berkas kecil

Ketika data terlalu besar untuk dicadangkan sebagai satu berkas, Anda dapat membuat cadangan kontennya setelah membaginya menjadi, misalnya potongan 2000MiB dan menggabungkan potongan-potongan itu kembali ke berkas asli nanti.

```
$ split -b 2000m large_file
$ cat x* >large_file
```



Perhatian
Pastikan Anda tidak memiliki berkas yang dimulai dengan ”x” untuk menghindari crash nama.

9.8.6 Menghapus konten berkas

Untuk menghapus isi berkas seperti berkas log, jangan gunakan rm(1) untuk menghapus berkas dan kemudian membuat berkas kosong baru, karena berkas mungkin masih diakses dalam interval antara perintah. Berikut ini adalah cara yang aman untuk menghapus isi berkas.

```
$ :>file_to_be_cleared
```

9.8.7 Berkas dummy

Perintah berikut membuat berkas dummy atau kosong.

```
$ dd if=/dev/zero    of=5kb.file bs=1k count=5
$ dd if=/dev/urandom of=7mb.file bs=1M count=7
$ touch zero.file
$ : > alwayszero.file
```

Anda harus menemukan berkas berikut.

- ”5kb.file” adalah nol sebanyak 5KB.
- ”7mb.file” adalah 7MB data acak.
- ”zero.file” mungkin suatu berkas 0 byte. Jika ada, mtime-nya diperbarui sedangkan konten dan panjangnya dipertahankan.
- ”alwayszero.file” selalu merupakan berkas 0 byte. Jika ada, mtime-nya diperbarui dan kontennya diatur ulang.

9.8.8 Menghapus seluruh hard disk

Ada beberapa cara untuk menghapus data sepenuhnya dari seluruh perangkat seperti hard disk, misalnya USB flash drive di ”/dev/sda”.



Perhatian
Periksa lokasi USB flash drive Anda dengan mount(8) terlebih dahulu sebelum menjalankan perintah di sini. Perangkat yang ditunjuk oleh ”/dev/sda” mungkin adalah hard disk SCSI atau hard disk serial-ATA tempat seluruh sistem Anda berada.

Hapus semua konten disk dengan mengatur ulang data ke 0 dengan yang berikut ini.

```
# dd if=/dev/zero of=/dev/sda
```

Hapus semuanya dengan menimpa dengan data acak sebagai berikut.

```
# dd if=/dev/urandom of=/dev/sda
```

Hapus semuanya dengan menimpa dengan data acak dengan sangat efisien sebagai berikut.

```
# shred -v -n 1 /dev/sda
```

Anda dapat menggunakan badblocks (8) dengan opsi -t random.

Karena dd(1) tersedia dari shell dari banyak CD Linux bootable seperti Debian installer CD, Anda dapat menghapus sistem terpasang Anda sepenuhnya dengan menjalankan perintah menghapus dari media tersebut pada sistem hard disk, misalnya, ”/dev/sda”, ”/dev/sda”, dll.

9.8.9 Menghapus area hard disk yang tidak terpakai

Area yang tidak digunakan pada hard disk (atau USB flash drive), misalnya ”/dev/sdb1” mungkin masih mengandung data yang telah dihapus karena data tersebut hanya dihapus tautannya dari sistem berkas. Data-data ini dapat dibersihkan dengan menimpanya.

```
# mount -t auto /dev/sdb1 /mnt/foo
# cd /mnt/foo
# dd if=/dev/zero of=junk
dd: writing to `junk': No space left on device
...
# sync
# umount /dev/sdb1
```



Awas
Ini biasanya sudah cukup untuk USB flash drive Anda. Namun ini tidak sempurna. Sebagian besar nama berkas yang dihapus dan atributnya mungkin tersembunyi dan tetap ada di dalam sistem berkas.

9.8.10 Membatalkan penghapusan berkas yang dihapus tapi masih terbuka

Bahkan jika Anda secara tidak sengaja menghapus suatu berkas, selama berkas itu masih digunakan oleh beberapa aplikasi (mode baca atau tulis), adalah mungkin untuk memulihkan berkas tersebut.

Misalnya, coba yang berikut ini

```
$ echo foo > bar
$ less bar
$ ps aux | grep ' less[ ]'
bozo    4775  0.0  0.0  92200   884 pts/8    S+   00:18   0:00 less bar
$ rm bar
$ ls -l /proc/4775/fd | grep bar
lr-x------ 1 bozo bozo 64 2008-05-09 00:19 4 -> /home/bozo/bar (deleted)
$ cat /proc/4775/fd/4 >bar
$ ls -l
-rw-r--r-- 1 bozo bozo 4 2008-05-09 00:25 bar
$ cat bar
foo
```

Jalankan pada terminal lain (ketika Anda memiliki paket `lsof`) sebagai berikut.

```
$ ls -li bar
2228329 -rw-r--r-- 1 bozo bozo 4 2008-05-11 11:02 bar
$ lsof |grep bar|grep less
less 4775 bozo 4r REG 8,3 4 2228329 /home/bozo/bar
$ rm bar
$ lsof |grep bar|grep less
less 4775 bozo 4r REG 8,3 4 2228329 /home/bozo/bar (deleted)
$ cat /proc/4775/fd/4 >bar
$ ls -li bar
2228302 -rw-r--r-- 1 bozo bozo 4 2008-05-11 11:05 bar
$ cat bar
foo
```

9.8.11 Mencari semua hardlink

Berkas dengan hardlink dapat diidentifikasi dengan "`ls -li`".

```
$ ls -li
total 0
2738405 -rw-r--r-- 1 root root 0 2008-09-15 20:21 bar
2738404 -rw-r--r-- 2 root root 0 2008-09-15 20:21 baz
2738404 -rw-r--r-- 2 root root 0 2008-09-15 20:21 foo
```

Baik "`baz`" dan "`foo`" memiliki cacah tautan "2" (>1) yang menunjukkan bahwa mereka memiliki hardlink. Nomor inoda mereka adalah sama-sama "2738404". Ini berarti mereka adalah berkas yang sama yang di-hardlink. Jika Anda tidak kebetulan menemukan semua berkas yang di-hardlink secara kebetulan, Anda dapat mencarinya dengan inode, misalnya, "2738404" sebagai berikut.

```
# find /path/to/mount/point -xdev -inum 2738404
```

9.8.12 Konsumsi ruang disk yang tak terlihat

Semua berkas yang dihapus tetapi terbuka mengkonsumsi ruang disk meskipun tidak terlihat dari du(1) normal. Mereka dapat ditampilkan daftarnya dengan ukuran mereka dengan berikut.

```
# lsof -s -X / |grep deleted
```

9.9 Tips enkripsi data

Dengan akses fisik ke PC Anda, siapa pun dapat dengan mudah mendapatkan hak istimewa root dan mengakses semua berkas di PC Anda (lihat Bagian 4.6.4). Ini berarti bahwa sistem kata sandi masuk tidak dapat mengamankan data pribadi dan sensitif Anda terhadap kemungkinan pencurian PC Anda. Anda harus menerapkan teknologi enkripsi data untuk melakukannya. Meskipun GNU privacy guard (lihat Bagian 10.3) dapat mengenkripsi berkas, dibutuhkan beberapa upaya pengguna.

Dm-crypt memfasilitasi enkripsi data otomatis melalui modul kernel Linux asli dengan upaya pengguna minimal menggunakan device-mapper.

paket	popcon	ukuran	deskripsi
`cryptsetup`	V:34, I:77	469	utilitas untuk perangkat blok terenkripsi (dm-crypt / LUKS)
`cryptmount`	V:1.8, I:2.3	236	utilitas untuk perangkat blok terenkripsi (dm-crypt / LUKS) dengan fokus pada mount/unmount oleh pengguna normal
`fscrypt`	V:0.4, I:1.1	6844	utilitas untuk enkripsi sistem berkas Linux (fscrypt)
`libpam-fscrypt`	V:0.31, I:0.34	6941	Modul PAM untuk enkripsi sistem berkas Linux (fscrypt)

Tabel 9.25: Daftar utilitas enkripsi data



Perhatian
Enkripsi memakan waktu CPU dll. Data terenkripsi menjadi tidak dapat diakses jika kata sandinya hilang. Tolong pertimbangkan manfaat dan biayanya.

Catatan
Seluruh sistem Debian dapat dipasang pada disk terenkripsi oleh debian-installer (lenny atau yang lebih baru) menggunakan dm-crypt/LUKS dan initramfs.

Tip
Lihat Bagian 10.3 untuk utilitas enkripsi ruang pengguna: GNU Privacy Guard.

9.9.1 Enkripsi disk lepasan dengan dm-crypt/LUKS

Anda dapat mengenkripsi isi perangkat massal yang dapat dilepas, misalnya USB flash drive di "`/dev/sdx`", menggunakan dm-crypt/LUKS. Cukup format dengan cara berikut.

```
# fdisk /dev/sdx
... "n" "p" "1" "return" "return" "w"
# cryptsetup luksFormat /dev/sdx1
...
# cryptsetup open /dev/sdx1 secret
...
# ls -l /dev/mapper/
total 0
crw-rw---- 1 root root  10, 60 2021-10-04 18:44 control
lrwxrwxrwx 1 root root       7 2021-10-04 23:55 secret -> ../dm-0
# mkfs.vfat /dev/mapper/secret
...
# cryptsetup close secret
```

Kemudian, dapat dikait seperti biasa pada ”/media/*nama_pengguna/label_disk*”, kecuali untuk menanyakan kata sandi (lihat Bagian 10.1.7) di bawah lingkungan desktop modern menggunakan paket udisks2. Perbedaannya adalah bahwa setiap data yang ditulis padanya dienkripsi. Entri kata sandi dapat diotomatisasi menggunakan keyring (lihat Bagian 10.3.6).

Anda dapat juga memformat media dalam sistem berkas yang berbeda, misalnya, ext4 dengan ”mkfs.ext4 /dev/mapper Jika btrfs digunakan sebagai gantinya, paket udisks2-btrfs perlu dipasang. Untuk sistem berkas ini, kepemilikan dan izin berkas mungkin perlu dikonfigurasi.

9.9.2 Mengait disk terenkripsi dengan dm-crypt/LUKS

Misalnya, partisi disk terenkripsi yang dibuat dengan dm-crypt/LUKS pada ”/dev/sdc5” oleh Debian Installer dapat dikait ke ”/mnt” sebagai berikut:

```
$ sudo cryptsetup open /dev/sdc5 ninja --type luks
Enter passphrase for /dev/sdc5: ****
$ sudo lvm
lvm> lvscan
  inactive          '/dev/ninja-vg/root' [13.52 GiB] inherit
  inactive          '/dev/ninja-vg/swap_1' [640.00 MiB] inherit
  ACTIVE            '/dev/goofy/root' [180.00 GiB] inherit
  ACTIVE            '/dev/goofy/swap' [9.70 GiB] inherit
lvm> lvchange -a y /dev/ninja-vg/root
lvm> exit
  Exiting.
$ sudo mount /dev/ninja-vg/root /mnt
```

9.10 Kernel

Debian mendistribusikan kernel Linux modular sebagai paket untuk arsitektur yang didukung.

Jika Anda membaca dokumentasi ini, Anda mungkin tidak perlu mengkompilasi kernel Linux sendiri.

9.10.1 Parameter kernel

Banyak fitur Linux dapat dikonfigurasi melalui parameter kernel sebagai berikut.

- Parameter kernel yang diinisialisasi oleh bootloader (lihat Bagian 3.1.2)
- Parameter kernel diubah oleh sysctl(8) saat runtime untuk yang dapat diakses melalui sysfs (lihat Bagian 1.2.12)
- Parameter modul yang ditetapkan oleh argumen modprobe(8) ketika modul diaktifkan (lihat Bagian 9.7.3)

Lihat ”Panduan pengguna dan administrator kernel Linux » Parameter baris perintah kernel” untuk detailnya.

9.10.2 Header kernel

Sebagian besar **program normal** tidak memerlukan header kernel dan pada kenyataannya dapat rusak jika Anda menggunakannya secara langsung untuk kompilasi. Mereka harus dikompilasi terhadap header di ”/usr/include/linux” dan ”/usr/include/asm” yang disediakan oleh paket libc6-dev (dibuat dari paket sumber glibc) pada sistem Debian.

Catatan
Untuk mengompail beberapa program kernel-spesifik seperti modul kernel dari sumber eksternal dan daemon automounter (amd), Anda harus menyertakan path ke header kernel yang sesuai, misalnya "`-I/usr/src/linux-particular-version/include/`", ke baris perintah Anda.

9.10.3 Mengompail kernel dan modul terkait

Debian memiliki metode sendiri untuk mengompail kernel dan modul terkait.

paket	popcon	ukuran	deskripsi
`build-essential`	V:20, I:510	17	paket penting untuk membangun paket Debian: `make`, `gcc`, ...
`bzip2`	V:163, I:973	113	utilitas kompresi dan dekompresi untuk berkas bz2
`libncurses5-dev`	I:31	6	pustaka dan dokumen pengembang untuk ncurses
`git`	V:422, I:625	52512	git: sistem kontrol revisi terdistribusi yang digunakan oleh kernel Linux
`fakeroot`	V:32, I:511	225	menyediakan lingkungan fakeroot untuk membangun paket sebagai non-root
`initramfs-tools`	V:495, I:989	49	alat untuk membangun initramfs (spesifik Debian)
`dkms`	V:73, I:147	235	dynamic kernel module support (DKMS) (generik)
`module-assistant`	V:1, I:12	395	alat pembantu untuk membuat paket modul (spesifik Debian)
`devscripts`	V:5, I:31	2783	skrip pembantu untuk pengelola Paket Debian (spesifik Debian)

Tabel 9.26: Daftar paket kunci yang akan dipasang untuk rekompilasi kernel pada sistem Debian

Jika Anda menggunakan `initrd` dalam Bagian 3.1.2, pastikan untuk membaca informasi terkait dalam initramfs-tools(8), update-initramfs(8), mkinitramfs(8), dan initramfs.conf(5).



Awas
Jangan menempatkan symlink ke direktori di pohon sumber (misalnya "`/usr/src/linux*`") dari "`/usr/include/linux`" dan "`/usr/include/asm`" saat mengompail sumber kernel Linux. (Beberapa dokumen usang menyarankan ini.)

Catatan
Ketika mengompail kernel Linux terbaru pada sistem `stable` Debian, penggunaan alat terbaru yang di-backport dari Debian `unstable` mungkin diperlukan.
module-assistant(8) (atau bentuk pendeknya `m-a`) membantu pengguna untuk membangun dan menginstall paket modul secara mudah untuk satu atau lebih kernel ubahan.
dynamic kernel module support (DKMS) adalah kerangka kerja independen distribusi baru yang dirancang untuk memungkinkan modul kernel individu ditingkatkan tanpa mengubah seluruh kernel. Ini digunakan untuk pemeliharaan modul di luar pohon. Ini juga membuatnya sangat mudah untuk membangun kembali modul saat Anda meningkatkan kernel.

9.10.4 Mengompail sumber kernel: Rekomendasi Tim Kernel Debian

Untuk membangun paket biner kernel ubahan dari sumber kernel hulu, Anda mesti memakai target "`deb-pkg`" yang disediakan olehnya.

```
$ sudo apt-get build-dep linux
$ cd /usr/src
$ wget https://mirrors.edge.kernel.org/pub/linux/kernel/v7.x/linux-7.2.8.tar.xz
$ tar --xz -xvf linux-7.2.8.tar.xz
$ cd linux-7.2.8
$ cp /boot/config-7.2.8.config
$ make menuconfig
 ...
$ make deb-pkg
```

Tip
Paket linux-source-*versi* menyediakan sumber kernel Linux dengan patch Debian sebagai "`/usr/src/linux-versi.tar.bz2`".

Untuk membangun paket biner tertentu dari paket sumber kernel Debian, Anda harus menggunakan target "`binary-arch_a` dalam "`debian/rules.gen`".

```
$ sudo apt-get build-dep linux
$ apt-get source linux
$ cd linux-7.*
$ fakeroot make -f debian/rules.gen binary-arch_i386_none_686
```

Lihat informasi lebih lanjut:

- Wiki Debian: KernelFAQ
- Wiki Debian: DebianKernel
- Buku Pegangan Kernel Debian Linux: https://kernel-handbook.debian.net

9.10.5 Driver perangkat keras dan firmware

Driver perangkat keras adalah kode yang berjalan pada CPU utama dari sistem target. Sebagian besar driver perangkat keras tersedia sebagai perangkat lunak bebas sekarang dan termasuk dalam paket kernel Debian normal di area `main`.

- Driver GPU
  - Driver GPU Intel (`main`)
  - Driver GPU AMD/ATI (`main`)
  - Driver GPU NVIDIA (`main` untuk driver nouveau, dan `non-free` untuk driver biner saja yang didukung oleh vendor.)

Firmware adalah kode atau data yang dimuat pada perangkat yang melekat pada sistem target (misalnya, microcode CPU, kode rendering yang berjalan pada GPU, atau data FPGA / CPLD, ...). Beberapa paket firmware tersedia sebagai perangkat lunak bebas tetapi banyak paket firmware tidak tersedia sebagai perangkat lunak bebas karena mengandung data biner tanpa sumber. Menginstall data firmware ini sangat penting bagi perangkat untuk berfungsi seperti yang diharapkan.

- Paket data firmware yang berisi data yang dimuat ke memori volatile pada perangkat target.
  - `firmware-linux-free` (`main`)
  - `firmware-linux-nonfree` (`non-free-firmware`)

  - `firmware-linux-*` (`non-free-firmware`)
  - `*-firmware` (`non-free-firmware`)
  - `intel-microcode` (`non-free-firmware`)
  - `amd64-microcode` (`non-free-firmware`)
- Paket program pembaruan firmware yang memperbarui data pada memori non-volatile pada perangkat target.
  - fwupd (`main`): Daemon pembaruan firmware yang mengunduh data firmware dari Linux Vendor Firmware Service.
  - gnome-firmware (`main`): Front end GTK untuk fwupd
  - plasma-discover-backend-fwupd (`main`): Front end Qt untuk fwupd

Harap diperhatikan bahwa akses ke paket `non-free-firmware` disediakan oleh media instalasi resmi untuk memberikan pengalaman instalasi fungsional kepada pengguna sejak Debian 12 Bookworm. Area `non-free-firmware` dijelaskan di Bagian 2.1.5.

Harap perhatikan juga bahwa data firmware yang diunduh oleh fwupd dari Linux Vendor Firmware Service dan dimuat ke kernel Linux yang berjalan mungkin `non-free`.

9.11 Sistem tervirtualisasi

Penggunaan sistem tervirtualisasi memungkinkan kita untuk menjalankan beberapa instansi sistem secara simultan pada satu perangkat keras.

Tip
Lihat wiki Debian tentang SystemVirtualization.

9.11.1 Alat virtualisasi dan emulasi

Ada beberapa platform alat virtualisasi dan emulasi.

- Paket emulasi perangkat keras lengkap seperti yang dipasang oleh metapackage game-emulator
- Sebagian besar emulasi tingkat CPU dengan beberapa emulasi perangkat I/O seperti QEMU
- Sebagian besar virtualisasi tingkat CPU dengan beberapa emulasi perangkat I/O seperti Kernel-based Virtual Machine (KVM)
- Virtualisasi kontainer tingkat OS dengan dukungan tingkat kernel seperti LXC (Linux Containers), Docker, systemd-nspawn(1), ...
- Virtualisasi akses sistem berkas tingkat OS dengan penimpaan pemanggilan pustaka sistem pada path berkas seperti chroot
- Virtualisasi akses sistem berkas tingkat OS dengan penimpaan panggilan pustaka sistem pada kepemilikan berkas seperti fakeroot
- Emulasi API OS seperti Wine
- Virtualisasi tingkat interpreter dengan pilihan yang dapat dieksekusi dan penimpaan pustaka run-time seperti virtualenv dan venv untuk Python

Virtualisasi kontainer menggunakan Bagian 4.7.5 dan teknologi backend dari Bagian 7.7.

Berikut adalah beberapa paket untuk membantu Anda menyiapkan sistem tervirtualisasi.

Lihat artikel Wikipedia Perbandingan mesin virtual platform untuk perbandingan detail solusi virtualisasi platform yang berbeda.

paket	popcon	ukuran	deskripsi
coreutils	V:912, I:1000	17994	Utilitas inti GNU yang memuat chroot(8)
util-linux	V:913, I:1000	4826	berbagai utilitas sistem yang memuat unshare(1)
systemd-container	V:74, I:77	2906	perkakas container/nspawn systemd yang memuat systemd-nspawn(1)
schroot	V:7.2, I:9.1	2222	alat khusus untuk mengeksekusi paket biner Debian di chroot
sbuild	V:3.9, I:6.9	155	alat untuk membangun paket biner Debian dari sumber Debian
debootstrap	V:4, I:44	332	mem-bootstrap sistem Debian dasar (ditulis dalam sh)
mmdebstrap	V:8, I:12	592	mem-bootstrap sistem Debian (ditulis dalam Perl)
cdebootstrap	V:0.1, I:1.3	114	mem-bootstrap sistem Debian (ditulis dalam C)
cloud-image-utils	V:1, I:14	52	utilitas manajemen image cloud
cloud-guest-utils	V:5, I:23	67	utilitas guest cloud
virt-manager	V:12, I:48	2309	Virtual Machine Manager: aplikasi desktop untuk mengelola mesin virtual
libvirt-clients	V:45, I:71	1168	program untuk pustaka libvirt
docker.io	V:50, I:53	80705	docker: Runtime kontainer Linux
podman	V:34, I:38	85864	podman: mesin untuk menjalankan kontainer berbasis OCI di dalam Pod
podman-docker	V:2.8, I:3.4	271	mesin untuk menjalankan kontainer berbasis OCI di Pods - pembungkus untuk docker
incus	V:0.8, I:3.4	24	Incus: kontainer sistem dan manajer mesin virtual
games-emulator	I:0.19	21	game-emulator: Emulator Debian untuk permainan
bochs	V:0.04, I:0.59	8469	Bochs: Emulator PC IA-32
qemu-system	I:20	69	QEMU: biner emulasi sistem penuh
qemu-user	V:5.6, I:9.3	472213	QEMU: biner emulasi mode pengguna
qemu-utils	V:14, I:114	12643	QEMU: utilitas
qemu-system-x86	V:52, I:98	69257	KVM: virtualisasi penuh pada perangkat keras x86 dengan virtualisasi yang dibantu perangkat keras
virtualbox	V:2.9, I:3.7	151079	VirtualBox: solusi virtualisasi x86 pada i386 dan amd64
gnome-boxes	V:1.2, I:6.1	7074	Boxes: Aplikasi GNOME sederhana untuk mengakses sistem virtual
xen-tools	V:0.1, I:1.4	719	alat untuk mengelola server virtual XEN Debian
wine	V:13, I:54	204	Wine: Implementasi API Windows (keluarga standar)
dosbox	V:1, I:11	2697	DOSBox: emulator x86 dengan grafis Tandy/Herc/CGA/EGA/VGA/SVGA, suara, dan DOS
lxc	V:10, I:13	1681	Alat ruang pengguna kontainer Linux
python3-venv	V:8, I:159	6	venv untuk membuat lingkungan python virtual (pustaka sistem)
python3-virtualenv	V:7, I:37	374	virtualenv untuk membuat lingkungan python virtual yang terisolasi
pipx	V:8, I:50	4412	pipx untuk memasang aplikasi python di lingkungan yang terisolasi

Tabel 9.27: Daftar alat virtualisasi

9.11.2 Alur kerja virtualisasi

Catatan
Kernel Debian baku mendukung KVM sejak `lenny`.

Aliran kerja umum untuk virtualisasi melibatkan beberapa langkah.

- Buat sistem berkas kosong (pohon berkas atau image disk).
  - Pohon berkas dapat dibuat oleh "`mkdir -p /path/ke/chroot`".
  - Berkas image disk mentah dapat dibuat dengan dd(1) (lihat Bagian 9.7.1 dan Bagian 9.7.5).
  - qemu-img(1) dapat digunakan untuk membuat dan mengkonversi berkas image disk yang didukung oleh QEMU.
  - Format berkas mentah dan VMDK dapat digunakan sebagai format umum di antara alat virtualisasi.
- Mount image disk dengan mount(8) ke sistem berkas (opsional).
  - Untuk berkas image disk mentah, kait itu sebagai perangkat loop atau perangkat device mapper (lihat Bagian 9.7.3).
  - Untuk image disk yang didukung oleh QEMU, kait mereka sebagai perangkat blok jaringan (lihat Bagian 9.11.3).
- Mengisi sistem berkas target dengan data sistem yang diperlukan.
  - Penggunaan program seperti `debootstrap` dan `cdebootstrap` membantu proses ini (lihat Bagian 9.11.4).
  - Gunakan installer OS di bawah emulasi sistem penuh.
- Menjalankan program di bawah lingkungan yang tervirtualisasi.
  - chroot menyediakan lingkungan virtual dasar yang cukup untuk mengkompilasi program, menjalankan aplikasi konsol, dan menjalankan daemon di dalamnya.
  - QEMU menyediakan emulasi CPU lintas platform.
  - QEMU dengan KVM menyediakan emulasi sistem penuh dengan virtualisasi yang dibantu perangkat keras.
  - VirtualBox menyediakan emulasi sistem penuh pada i386 dan amd64 dengan atau tanpa virtualisasi yang dibantu perangkat keras.

9.11.3 Mengait berkas image disk virtual

Untuk berkas image disk mentah, lihat Bagian 9.7.

Untuk berkas image disk virtual lainnya, Anda dapat menggunakan qemu-nbd(8) untuk mengekspornya menggunakan protokol perangkat blok jaringan dan memasangnya menggunakan modul kernel `nbd`.

qemu-nbd(8) mendukung format disk yang didukung oleh QEMU: mentah, qcow2, qcow, vmdk, vdi, bochs, cow (user-mode Linux copy-on-write), parallels, dmg, cloop, vpc, vvfat (virtual VFAT), dan host_device.

Perangkat blok jaringan dapat mendukung partisi dengan cara yang sama seperti perangkat loop (lihat Bagian 9.7.3). Anda dapat mengait partisi pertama dari "`disk.img`" sebagai berikut.

```
# modprobe nbd max_part=16
# qemu-nbd -v -c /dev/nbd0 disk.img
...
# mkdir /mnt/part1
# mount /dev/nbd0p1 /mnt/part1
```

Tip
Anda dapat mengekspor hanya partisi pertama dari "`disk.img`" menggunakan opsi "`-P 1`" ke qemu-nbd(8).

9.11.4 Sistem chroot

Jika Anda ingin mencoba lingkungan Debian baru dari konsol terminal, saya sarankan Anda untuk menggunakan chroot. Hal ini memungkinkan Anda untuk menjalankan aplikasi konsol Debian `unstable` dan `testing` tanpa risiko yang biasanya terkait dan tanpa reboot. chroot(8) adalah cara yang paling dasar.



Perhatian
Contoh di bawah ini mengasumsikan sistem induk dan sistem chroot berbagi arsitektur CPU `amd64` yang sama.

Meskipun Anda dapat secara manual membuat lingkungan chroot(8) menggunakan debootstrap(1), ini membutuhkan upaya yang tidak sepele.

Paket sbuild untuk membangun paket Debian dari sumber menggunakan lingkungan chroot yang dikelola oleh paket schroot. Ini datang dengan skrip pembantu sbuild-createchroot(1). Mari kita pelajari cara kerjanya dengan menjalankannya sebagai berikut.

```
$ sudo mkdir -p /srv/chroot
$ sudo sbuild-createchroot -v --include=eatmydata,ccache unstable /srv/chroot/unstable- ↩
    amd64-sbuild http://deb.debian.org/debian
 ...
```

Anda melihat bagaimana debootstrap(8) mengisi data sistem untuk lingkungan `unstable` di bawah "`/srv/chroot/unstab` untuk sistem build minimal.

Anda dapat masuk ke lingkungan ini menggunakan schroot(1).

```
$ sudo schroot -v -c chroot:unstable-amd64-sbuild
```

Anda melihat bagaimana shell sistem berjalan di bawah lingkungan `unstable` dibuat.

Catatan
Berkas "`/usr/sbin/policy-rc.d`" yang selalu keluar dengan 101 mencegah program daemon dimulai secara otomatis pada sistem Debian. Lihat "`/usr/share/doc/init-system-helpers/README.policy-rc.d.gz`".

Catatan
Beberapa program di bawah chroot mungkin memerlukan akses ke lebih banyak berkas dari sistem induk untuk berfungsi daripada yang disediakan `sbuild-createchroot` seperti di atas. Misalnya, "`/sys`", "`/etc/passwd`", "`/etc/group`", "`/var/run/utmp`", "`/var/log/wtmp`", dll mungkin perlu dikait-bind atau disalin.

Tip
Paket `sbuild` membantu membangun sistem chroot dan membangun paket di dalam chroot menggunakan `schroot` sebagai backend-nya. Ini adalah sistem yang ideal untuk memeriksa ketergantungan build. Lihat lebih jauh tentang sbuild di Debian wiki dan contoh konfigurasi sbuild di "Panduan untuk Pemelihara Debian".

Tip
Perintah systemd-nspawn(1) membantu menjalankan suatu perintah atau OS dalam suatu kontainer ringan dengan cara yang mirip dengan `chroot`. Itu lebih bertenaga karena itu memakai namespace untuk memvirtualisasi secara penuh pohon proses, IPC, nama host, nama domain, dan secara opsional jejaring dan basis data pengguna. Lihat systemd-nspawn.

9.11.5 Beberapa sistem desktop

Jika Anda ingin mencoba lingkungan Desktop GUI baru dari OS apa pun, saya sarankan Anda untuk menggunakan QEMU atau KVM pada suatu sistem Debian `stable` untuk menjalankan beberapa sistem desktop dengan aman menggunakan virtualisasi. Ini memungkinkan Anda untuk menjalankan aplikasi desktop apa pun termasuk Debian `unstable` dan `testing` tanpa risiko biasa yang terkait dengan mereka dan tanpa reboot.

Karena QEMU murni sangat lambat, disarankan untuk mempercepatnya dengan KVM ketika sistem host mendukungnya.

Virtual Machine Manager juga dikenal sebagai `virt-manager` adalah alat GUI yang nyaman untuk mengelola mesin virtual KVM melalui libvirt.

Image disk virtual "`virtdisk.qcow2`" yang berisi sistem Debian untuk QEMU dapat dibuat menggunakan debian-installer: CD kecil sebagai berikut.

```
$ wget https://cdimage.debian.org/debian-cd/5.0.3/amd64/iso-cd/debian-503-amd64-netinst.iso
$ qemu-img create -f qcow2 virtdisk.qcow2 5G
$ qemu -hda virtdisk.qcow2 -cdrom debian-503-amd64-netinst.iso -boot d -m 256
...
```

Tip
Menjalankan distribusi GNU/Linux lainnya seperti Ubuntu dan Fedora di bawah virtualisasi adalah cara yang bagus untuk mempelajari tips konfigurasi. OS proprietari lainnya dapat dijalankan dengan baik di bawah virtualisasiGNU/Linux ini juga.

Lihat tips lainnya di Wiki Debian: SystemVirtualization.

Bab 10

Manajemen data

Alat dan tips untuk mengelola data biner dan teks pada sistem Debian dijelaskan.

10.1 Berbagi, menyalin, dan mengarsipkan



Awas
Akses tulis yang tidak terkoordinasi ke perangkat dan berkas yang diakses secara aktif dari berbagai proses tidak boleh dilakukan untuk menghindari kondisi race. Mekanisme penguncian berkas menggunakan flock(1) dapat digunakan untuk menghindarinya.

Keamanan data dan berbagi yang terkontrol memiliki beberapa aspek.

- Pembuatan arsip data
- Akses penyimpanan jarak jauh
- Duplikasi
- Pelacakan riwayat modifikasi
- Memfasilitasi berbagi data
- Pencegahan akses berkas tanpa otorisasi
- Deteksi modifikasi berkas tanpa otorisasi

Ini dapat direalisasikan dengan menggunakan beberapa kombinasi alat.

- Alat arsip dan kompresi
- Alat salin dan sinkronisasi
- Sistem berkas jaringan
- Media penyimpanan lepasan
- Secure shell
- Sistem autentikasi
- Alat sistem kontrol versi
- Alat enkripsi hash dan kriptografi

10.1.1 Alat arsip dan kompresi

Berikut adalah ringkasan alat kompresi dan arsip yang tersedia pada sistem Debian.



Awas
Jangan mengatur variabel "$TAPE" kecuali Anda tahu apa yang diharapkan. Itu mengubah perilaku tar(1).

- Arsip tar(1) yang di-gzip menggunakan ekstensi berkas "`.tgz`" atau "`.tar.gz`".
- Arsip tar(1) terkompresi xz menggunakan ekstensi berkas "`.txz`" atau "`.tar.xz`".
- Metode kompresi populer dalam alat FOSS seperti tar(1) telah bergerak sebagai berikut: `gzip` → `bzip2` → `xz`
- cp(1), scp(1), dan tar(1) mungkin memiliki beberapa batasan untuk berkas khusus. cpio(1) adalah yang paling serbaguna.
- cpio(1) dirancang untuk digunakan dengan find(1) dan perintah lainnya dan cocok untuk membuat skrip cadangan karena bagian pemilihan berkas dari skrip dapat diuji secara independen.
- Struktur internal berkas data Libreoffice adalah berkas "`.jar`" yang dapat dibuka juga oleh `unzip`.
- Alat arsip lintas platform de-facto adalah `zip`. Gunakan sebagai "`zip -rX`" untuk mencapai kompatibilitas maksimum. Gunakan juga opsi "`-s`", jika ukuran berkas maksimum penting.

10.1.2 Alat salin dan sinkronisasi

Berikut adalah ringkasan alat cadangan dan salin sederhana yang tersedia di sistem Debian.

Menyalin berkas dengan rsync(8) menawarkan fitur yang lebih kaya daripada yang lain.

- algoritma delta-transfer yang hanya mengirimkan perbedaan antara berkas sumber dan berkas yang ada di tujuan
- algoritma pemeriksaan cepat (secara baku) yang mencari berkas yang telah berubah dalam ukuran atau dalam waktu terakhir dimodifikasi
- opsi "`--exclude`" dan "`--exclude-from`" yang mirip dengan tar(1)
- Sintaks "garis miring di akhir pada direktori sumber" yang menghindari pembuatan tingkat direktori tambahan di tujuan.

Tip
Alat sistem kontrol versi (VCS) Tabel 10.14 dapat berfungsi sebagai alat salinan dan sinkronisasi multi-arah.

10.1.3 Idiom untuk arsip

Berikut adalah beberapa cara untuk mengarsipkan dan membuka arsip seluruh isi direktori "`./source`" menggunakan alat yang berbeda.

GNU tar(1):

```
$ tar -cvJf archive.tar.xz ./source
$ tar -xvJf archive.tar.xz
```

Atau, dengan berikut.

paket	popcon	ukuran	ekstensi	perintah	kommen
tar	V:907, I:1000	3101	.tar	tar(1)	pengarsip standar (standar de facto)
cpio	V:352, I:999	1201	.cpio	cpio(1)	Pengarsipan gaya Unix System V, gunakan dengan find(1)
binutils	V:191, I:631	1131	.ar	ar(1)	pengarsip untuk pembuatan pustaka statis
fastjar	V:1.0, I:9.5	182	.jar	fastjar(1)	pengarsip untuk Java (mirip zip)
pax	V:5.1, I:9.0	167	.pax	pax(1)	pengarsip standar POSIX baru, kompromi antara tar dan cpio
gzip	V:895, I:1000	262	.gz	gzip(1), zcat(1), …	Utilitas kompresi GNU LZ77 (standar de facto)
bzip2	V:163, I:973	113	.bz2	bzip2(1), bzcat(1), …	Utilitas kompresi pengurut blok Burrows-Wheeler dengan rasio kompresi yang lebih tinggi daripada gzip(1) (lebih lambat dari gzip dengan sintaks serupa)
lzma	V:0.6, I:9.6	349	.lzma	lzma(1)	Utilitas kompresi LZMA dengan rasio kompresi yang lebih tinggi daripada gzip(1) (usang)
xz-utils	V:310, I:981	1551	.xz	xz(1), xzdec(1), …	Utilitas kompresi XZ dengan rasio kompresi yang lebih tinggi daripada bzip2(1) (lebih lambat dari gzip tetapi lebih cepat dari bzip2; pengganti utilitas kompresi LZMA)
zstd	V:334, I:830	2308	.zstd	zstd(1), zstdcat(1), …	Zstandard utilitas kompresi lossless cepat
p7zip	V:5, I:172	8	.7z	7zr(1), p7zip(1)	Pengarsip berkas 7-Zip dengan rasio kompresi tinggi (kompresi LZMA)
p7zip-full	V:17, I:199	12	.7z	7z(1), 7za(1)	Pengarsip berkas 7-Zip dengan rasio kompresi tinggi (kompresi LZMA dan lain-lain)
lzop	V:14, I:129	164	.lzo	lzop(1)	Utilitas kompresi LZO dengan kompresi dan kecepatan dekompresi yang lebih tinggi daripada gzip(1) (rasio kompresi yang lebih rendah daripada gzip dengan sintaks serupa)
zip	V:48, I:355	635	.zip	zip(1)	InfoZIP: Alat arsip dan kompresi DOS
unzip	V:123, I:744	399	.zip	unzip(1)	InfoZIP: Alat pembuka arsip dan dekompresi DOS

Tabel 10.1: Daftar alat arsip dan kompresi

paket	popcon	ukuran	perkakas	fungsi
coreutils	V:912, I:1000	17994	GNU cp	menyalin berkas dan direktori secara lokal (”-a” untuk rekursif)
openssh-client	V:682, I:997	3611	scp	dari jarak jauh menyalin berkas dan direktori (klien, ”-r” untuk rekursif)
openssh-server	V:774, I:820	3615	sshd	menyalin berkas dan direktori dari jarak jauh (server jarak jauh)
rsync	V:180, I:539	967		Sinkronisasi dan pencadangan jarak jauh 1 arah
unison	V:3, I:12	14		Sinkronisasi dan pencadangan jarak jauh 2 arah

Tabel 10.2: Daftar alat salin dan sinkronisasi

```
$ find ./source -xdev -print0 | tar -cvJf archive.tar.xz --null -T -
```

cpio(1):

```
$ find ./source -xdev -print0 | cpio -ov --null > archive.cpio; xz archive.cpio
$ zcat archive.cpio.xz | cpio -i
```

10.1.4 Idiom untuk menyalin

Berikut adalah beberapa cara untuk menyalin seluruh konten direktori ”./source” menggunakan alat-alat yang berbeda.

- Salinan lokal: direktori ”./source” → direktori ”/dest”
- Salinan jarak jauh: direktori ”./source” di host lokal → direktori ”/dest” di host ”user@host.dom”

rsync(8):

```
# cd ./source; rsync -aHAXSv . /dest
# cd ./source; rsync -aHAXSv . user@host.dom:/dest
```

Anda dapat menggunakan sintaks ”garis miring di akhir pada direktori sumber”.

```
# rsync -aHAXSv ./source/ /dest
# rsync -aHAXSv ./source/ user@host.dom:/dest
```

Atau, dengan berikut.

```
# cd ./source; find . -print0 | rsync -aHAXSv0 --files-from=- . /dest
# cd ./source; find . -print0 | rsync -aHAXSv0 --files-from=- . user@host.dom:/dest
```

GNU cp(1) dan openSSH scp(1):

```
# cd ./source; cp -a . /dest
# cd ./source; scp -pr . user@host.dom:/dest
```

GNU tar(1):

```
# (cd ./source && tar cf - . ) | (cd /dest && tar xvfp - )
# (cd ./source && tar cf - . ) | ssh user@host.dom '(cd /dest && tar xvfp - )'
```

cpio(1):

```
# cd ./source; find . -print0 | cpio -pvdm --null --sparse /dest
```

Anda dapat mengganti "`.`" dengan "`foo`" untuk semua contoh yang mengandung "`.`" untuk menyalin berkas dari direktori "`./source/foo`" ke direktori "`/dest/foo`".

Anda dapat mengganti "`.`" dengan path absolut "`/path/to/source/foo`" untuk semua contoh yang mengandung "`.`" untuk membuang "`cd ./source;`". Ini menyalin berkas ke lokasi yang berbeda tergantung pada alat yang digunakan sebagai berikut.

- "`/dest/foo`": rsync(8), GNU cp(1), dan scp(1)
- "`/dest/path/to/source/foo`": GNU tar(1), dan cpio(1)

Tip
rsync(8) dan GNU cp(1) memiliki opsi "`-u`" untuk melewatkan berkas yang lebih baru pada penerima.

10.1.5 Idiom untuk pemilihan berkas

find(1) digunakan untuk memilih berkas untuk perintah arsip dan salin (lihat Bagian 10.1.3 dan Bagian 10.1.4) atau untuk xargs(1) (lihat Bagian 9.4.9). Ini dapat ditingkatkan dengan memakai argumen-argumen perintahnya.

Sintaks dasar find(1) dapat diringkas sebagai berikut.

- Argumen bersyaratnya dievaluasi dari kiri ke kanan.
- Evaluasi ini berhenti setelah hasilnya ditentukan.
- "**OR** logis" (ditentukan oleh "`-o`" antara syarat) memiliki prioritas yang lebih rendah daripada "**AND** logis" (ditentukan oleh "`-a`" atau nihil antara syarat).
- "**NOT** logis" (ditentukan oleh "`!`" sebelum syarat) memiliki prioritas yang lebih tinggi daripada "**AND** logis".
- "`-prune`" selalu mengembalikan **TRUE** logis dan, jika itu adalah direktori, pencarian berkas dihentikan di luar titik ini.
- "`-name`" mencocokkan basis nama berkas dengan shell glob (lihat Bagian 1.5.6) tetapi juga mencocokkan "`.`" awal dengan karakter-karakter meta seperti "`*`" dan "?". (Fitur POSIX baru)
- "`-regex`" mencocokkan path lengkap dengan **BRE** gaya emacs (lihat Bagian 1.6.2) secara baku.
- "`-size`" cocok dengan berkas berdasarkan ukuran berkas (nilai didahului dengan "+" untuk lebih besar, didahului dengan "`-`" untuk lebih kecil)
- "`-newer`" cocok dengan berkas yang lebih baru dari yang ditentukan dalam argumennya.
- "`-print0`" selalu mengembalikan **TRUE** logis dan mencetak nama berkas lengkap (diakhiri null) pada keluaran standar.

find(1) sering digunakan dengan gaya idiomatik sebagai berikut.

```
# find /path/to \
    -xdev -regextype posix-extended \
    -type f -regex ".*\.cpio|.*~" -prune -o \
    -type d -regex ".*/\.git" -prune -o \
    -type f -size +99M -prune -o \
    -type f -newer /path/to/timestamp -print0
```

Ini berarti melakukan tindakan berikut.

1. Cari semua berkas mulai dari "`/path/to`"
2. Secara global membatasi pencariannya dalam sistem berkas awalnya dan menggunakan **ERE** (lihat Bagian 1.6.2) sebagai gantinya
3. Kecualikan berkas yang cocok regex dari "`.*\.cpio`" atau "`.*~`" dari pencarian dengan berhenti memroses
4. Kecualikan direktori yang cocok dengan regex "`.*/\.git`" dari pencarian dengan berhenti memroses
5. Kecualikan berkas yang lebih dari 99 Megabyte (satuan 1048576 byte) dari pencarian dengan berhenti memroses
6. Cetak nama berkas yang memenuhi kondisi pencarian di atas dan lebih baru dari "`/path/to/timestamp`"

Harap dicatat penggunaan idiomatik "`-prune -o`" untuk mengecualikan berkas dalam contoh di atas.

Catatan
Untuk sistem mirip Unix non-Debian, beberapa opsi mungkin tidak didukung oleh find(1). Dalam kasus seperti itu, silakan pertimbangkan untuk menyesuaikan metode yang cocok dan mengganti "`-print0`" dengan "`-print`". Anda mungkin perlu menyesuaikan perintah terkait juga.

10.1.6 Media arsip

Ketika memilih media penyimpanan data komputer untuk arsip data penting, Anda harus berhati-hati tentang keterbatasan mereka. Untuk cadangan data pribadi kecil, saya menggunakan CD-R dan DVD-R oleh perusahaan dengan merek terkenal dan menyimpan di lingkungan yang sejuk, teduh, kering, dan bersih. (Media arsip tape tampaknya populer untuk penggunaan profesional.)

Catatan
Brankas tahan api dimaksudkan untuk dokumen kertas. Sebagian besar media penyimpanan data komputer memiliki toleransi suhu yang lebih rendah daripada kertas. Saya biasanya mengandalkan beberapa salinan terenkripsi aman yang disimpan di beberapa lokasi aman.

Umur penyimpanan optimis dari media arsip yang terlihat di internet (sebagian besar dari info vendor).

- 100+ tahun : Kertas bebas asam dengan tinta
- 100 tahun : Penyimpanan optik (CD/DVD, CD/DVD-R)
- 30 tahun : Penyimpanan magnetik (tape, floppy)
- 20 tahun : Penyimpanan optik perubahan fase (CD-RW)

Ini tidak memperhitungkan kegagalan mekanis karena penanganan dll.

Siklus tulis optimis dari media arsip yang terlihat di internet (sebagian besar dari informasi vendor).

- 250.000+ siklus : Harddisk drive
- 10.000+ siklus : Memori flash
- 1.000 siklus : CD/DVD-RW
- 1 siklus : CD/DVD-R, kertas



Perhatian
Angka masa penyimpanan dan siklus penulisan di sini tidak boleh digunakan untuk keputusan tentang penyimpanan data penting apa pun. Silakan berkonsultasi dengan informasi produk spesifik yang disediakan oleh produsen.

Tip
Karena CD/DVD-R dan kertas hanya memiliki 1 siklus penulisan, mereka secara inheren mencegah kehilangan data yang tidak disengaja dengan menimpa. Ini adalah keuntungan!

Tip
Jika Anda memerlukan cadangan data dalam jumlah besar dan sering, hard disk pada host jarak jauh yang dihubungkan oleh koneksi jaringan cepat, mungkin satu-satunya pilihan yang realistis.

Tip
Jika Anda menggunakan media yang dapat ditulis ulang untuk cadangan Anda, penggunaan sistem berkas seperti btrfs atau zfs yang mendukung snapshot hanya baca mungkin merupakan ide yang baik.

10.1.7 Perangkat penyimpanan lepasan

Perangkat penyimpanan lepasan mungkin salah satu dari yang berikut ini.

- USB flash drive
- Hard disk drive
- Drive cakram optik
- Kamera digital
- Pemutar musik digital

Mereka dapat dihubungkan melalui salah satu dari yang berikut.

- USB
- IEEE 1394 / FireWire

Lingkungan desktop modern seperti GNOME dan KDE dapat mengait perangkat lepasan ini secara otomatis tanpa entri ”`/etc/fstab`” yang cocok.

- Paket `udisks2` menyediakan daemon dan utilitas terkait untuk mengait dan melepas kait perangkat-perangkat ini.
- D-bus membuat kejadian-kejadian untuk memulai proses otomatis.
- PolicyKit memberikan hak istimewa yang diperlukan.

Tip
Perangkat yang dikait otomatis mungkin memiliki opsi mount ”`uhelper=`” yang digunakan oleh umount(8).

Tip
Mengait otomatis di bawah lingkungan desktop modern terjadi hanya ketika perangkat media lepasan tidak terdaftar di "`/etc/fstab`".

Titik kait di bawah lingkungan desktop modern dipilih sebagai "`/media/username/disk_label`" yang dapat disesuaikan dengan hal-hal berikut.

- mlabel(1) untuk sistem berkas FAT
- genisoimage(1) dengan opsi "`-V`" untuk sistem berkas ISO9660
- tune2fs(1) dengan opsi "`-L`" untuk sistem berkas ext2/ext3/ext4

Tip
Pilihan pengodean mungkin perlu disediakan sebagai opsi kait (lihat Bagian 8.1.3).

Tip
Penggunaan menu GUI untuk melepas kait sistem berkas dapat menghapus node perangkat yang dihasilkan secara dinamis seperti "`/dev/sdc`". Jika Anda ingin menyimpan node perangkatnya, lepas kait dengan perintah umount(8) dari prompt shell.

10.1.8 Pilihan sistem berkas untuk berbagi data

Saat berbagi data dengan sistem lain melalui perangkat penyimpanan lepasan, Anda harus memformatnya dengan sistem berkas umum yang didukung oleh kedua sistem. Berikut adalah daftar pilihan sistem berkas.

Catatan
Pernyataan tentang hard disk (HDD) berlaku untuk perangkat penyimpanan lain seperti SSD / USB flash drive / Kartu memori / ... . Ganti nama perangkat dalam contoh seperti `/dev/sda` dengan nama perangkat yang sesuai `/dev/nvme0`, `/dev/mmcblk0`, ...

Tip
Lihat Bagian 9.9.1 untuk berbagi data lintas platform menggunakan enkripsi tingkat perangkat.

Sistem berkas FAT didukung oleh hampir semua sistem operasi modern dan cukup berguna untuk tujuan pertukaran data melalui media lepasan seperti hard disk.

Saat memformat perangkat lepasan seperti hard disk untuk berbagi data lintas platform dengan sistem berkas FAT, berikut ini harus menjadi pilihan yang aman.

- Memartisi mereka dengan fdisk(8), cfdisk(8), atau parted(8) (lihat Bagian 9.6.2) menjadi partisi primer tunggal dan untuk menandainya sebagai berikut.
  - Ketik "6" untuk FAT16 bagi media yang kurang dari 2GB.
  - Ketik "c" untuk FAT32 (LBA) untuk media yang lebih besar.
- Memformat partisi utama dengan mkfs.vfat(8) dengan yang berikut.
  - Hanya nama perangkatnya, misalnya "`/dev/sda1`" untuk FAT16

nama sistem berkas	skenario penggunaan tipikal
FAT12	berbagi data lintas platform pada floppy disk (<32MiB)
FAT16	berbagi data lintas platform pada perangkat seperti hard disk kecil (<2GiB)
FAT32	berbagi data lintas platform pada perangkat seperti hard disk besar (<8TiB, didukung oleh yang lebih baru daripada MS Windows95 OSR2)
exFAT	berbagi data lintas platform pada perangkat seperti hard disk besar (<512TiB, didukung oleh WindowsXP, Mac OS X Snow Leopard 10.6.5, dan kernel Linux sejak rilis 5.4)
NTFS	berbagi data lintas platform pada perangkat seperti hard disk besar (didukung secara native pada MS Windows NT dan versi yang lebih baru, dan didukung oleh NTFS-3G melalui FUSE di Linux)
ISO9660	berbagi lintas platform data statis pada CD-R dan DVD+/-R
UDF	penulisan data inkremental pada CD-R dan DVD+/-R (baru)
MINIX	penyimpanan data berkas unix yang efisien pada floppy disk
ext2	berbagi data pada perangkat seperti hard disk dengan sistem Linux yang lebih lama
ext3	berbagi data pada perangkat seperti hard disk dengan sistem Linux yang lebih lama
ext4	berbagi data pada perangkat seperti hard disk dengan sistem Linux saat ini
btrfs	berbagi data pada perangkat seperti hard disk dengan sistem Linux saat ini dengan snapshot hanya-baca

Tabel 10.3: Daftar pilihan sistem berkas untuk perangkat penyimpanan lepasan dengan skenario penggunaan yang khas

– Opsi eksplisit dan nama perangkatnya, misalnya "`-F 32 /dev/sda1`" untuk FAT32

Saat menggunakan sistem berkas FAT atau ISO9660 untuk berbagi data, berikut ini harus menjadi pertimbangan yang aman.

- Pengarsipan berkas ke dalam berkas arsip terlebih dahulu menggunakan tar(1), atau cpio(1) untuk mempertahankan nama berkas yang panjang, taut simbolis, izin berkas Unix asli dan informasi pemilik.
- Memecah berkas arsip menjadi potongan kurang dari 2 GiB dengan perintah split(1) untuk melindunginya dari batasan ukuran berkas.
- Mengenkripsi berkas arsip untuk mengamankan isinya dari akses yang tidak sah.

Catatan
Untuk sistem berkas FAT dengan desainnya, ukuran berkas maksimum adalah `(2^32 - 1) byte = (4GiB - 1 byte)`. Untuk beberapa aplikasi pada OS 32 bit yang lebih tua, ukuran berkas maksimum bahkan lebih kecil `(2^31 - 1) byte = (2GiB - 1 byte)`. Debian tidak mengalami masalah yang terakhir.

Catatan
Microsoft sendiri tidak merekomendasikan untuk menggunakan FAT bagi drive atau partisi lebih dari 200 MB. Microsoft menyoroti kekurangannya seperti penggunaan ruang disk yang tidak efisien dalam "Overview of FAT, HPFS, and NTFS File Systems" mereka. Tentu saja, kita biasanya harus menggunakan sistem berkas ext4 untuk Linux.

Tip
Untuk informasi lebih lanjut tentang sistem berkas dan mengakses sistem berkas, silakan baca Filesystems HOWTO".

10.1.9 Berbagi data melalui jaringan

Saat berbagi data dengan sistem lain melalui jaringan, Anda harus menggunakan layanan umum. Berikut adalah beberapa petunjuk.

layanan jaringan	deskripsi skenario penggunaan umum
Sistem berkas dikait jaringan SMB/CIFS dengan Samba	berbagi berkas melalui "Microsoft Windows Network", lihat smb.conf(5) dan HOWTO dan Panduan Acuan Resmi Samba 3.x.x atau paket `samba-doc`
Sistem berkas dikait jaringan NFS dengan kernel Linux	berbagi berkas melalui "Unix /Linux Network", lihat exports(5) dan Linux NFS-HOWTO
Layanan HTTP	berbagi berkas antara server web/klien
Layanan HTTPS	berbagi berkas antara server web/klien dengan Secure Sockets Layer (SSL) terenkripsi atau Transport Layer Security (TLS)
Layanan FTP	berbagi berkas antara server/klien FTP

Tabel 10.4: Daftar layanan jaringan yang akan dipilih dengan skenario penggunaan umum

Meskipun sistem berkas ini dikait melalui jaringan dan metode transfer berkas melalui jaringan cukup nyaman untuk berbagi data, ini mungkin tidak aman. Koneksi jaringan mereka harus diamankan oleh yang berikut.

- Enkripsi dengan SSL/TLS
- Tembuskan itu melalui SSH
- Tembuskan itu melalui VPN
- Batasi di belakang firewall yang aman

Lihat juga Bagian 6.5 dan Bagian 6.6.

10.2 Pencadangan dan pemulihan

Kita semua tahu bahwa komputer gagal suatu saat atau kesalahan manusia menyebabkan kerusakan sistem dan data. Operasi pencadangan dan pemulihan adalah bagian penting dari administrasi sistem yang sukses. Semua mode kegagalan yang mungkin menimpa Anda suatu hari nanti.

Tip
Jaga sistem cadangan Anda sederhana dan seringlah mencadangkan sistem Andag. Memiliki data cadangan lebih penting daripada seberapa baik metode cadangan Anda secara teknis.

10.2.1 Kebijakan pencadangan dan pemulihan

Ada 3 faktor kunci yang menentukan kebijakan pencadangan dan pemulihan yang sebenarnya.

1. Mengetahui apa yang harus dicadangkan dan dipulihkan.
   - Berkas data yang langsung dibuat oleh Anda: data dalam "`~/`"
   - Berkas data yang dibuat oleh aplikasi yang digunakan oleh Anda: data dalam "`/var/`" (kecuali "`/var/cache/`", "`/var/run/`", dan "`/var/tmp/`")
   - Berkas konfigurasi sistem: data dalam "`/etc/`"

- Program lokal: data dalam "`/usr/local/`" atau "`/opt/`"
- Informasi instalasi sistem: memo dalam teks polos pada langkah-langkah kunci (partisi, …)
- Kumpulan data yang terbukti: dikonfirmasi oleh operasi pemulihan eksperimental sebelumnya
  - Pekerjaan cron sebagai proses pengguna: berkas di direktori "`/var/spool/cron/crontabs`" dan memulai ulang cron(8). Lihat Bagian 9.4.14 untuk cron(8) dan crontab(1).
  - Pekerjaan pengatur waktu sistem sebagai proses pengguna: berkas di direktori "`~/.config/systemd/user`". Lihat systemd.timer(5) dan systemd.service(5).
  - Memulai otomatis pekerjaan sebagai proses pengguna: berkas di direktori "`~/.config/autostart`". Lihat Spesifikasi Autostart Aplikasi Desktop.

2. Mengetahui cara membuat cadangan dan memulihkan.
   - Penyimpanan data yang aman: perlindungan dari penimpaan dan kegagalan sistem
   - Pencadangan yang sering: cadangan terjadwal
   - Cadangan redundan: mirroring data
   - Proses fool proof: pencadangan dengan perintah tunggal yang mudah
3. Menilai risiko dan biaya yang terlibat.
   - Risiko data saat hilang
     - Data harus setidaknya pada partisi disk yang berbeda, sebaiknya pada disk dan mesin yang berbeda agar bertahan atas korupsi sistem berkas. Data penting paling baik disimpan pada sistem berkas hanya baca. [1]
   - Risiko data saat bocor
     - Data identitas sensitif seperti "`/etc/ssh/ssh_host_*_key`", "`~/.gnupg/*`", "`~/.ssh/*`", "`~/.local/share/`", "`/etc/passwd`", "`/etc/shadow`", "`popularity-contest.conf`", "`/etc/ppp/pap-secrets`", and "`/etc/ex`" harus dicadangkan maupun dienkripsi. [2] (Lihat Bagian 9.9.)
     - Jangan pernah menanam kata sandi login sistem atau frasa sandi dekripsi dalam skrip apa pun bahkan pada sistem terpercaya apa pun. (Lihat Bagian 10.3.6.)
   - Mode kegagalan dan kemungkinannya
     - Perangkat keras (terutama HDD) akan rusak
     - Sistem berkas mungkin terkorupsi dan data di dalamnya mungkin hilang
     - Sistem penyimpanan jarak jauh tidak dapat dipercaya untuk pelanggaran keamanan
     - Perlindungan kata sandi yang lemah dapat dengan mudah dikompromikan
     - Sistem izin berkas mungkin terkompromi
   - Sumber daya yang diperlukan untuk membuat cadangan: manusia, perangkat keras, perangkat lunak, …
     - Pencadangan terjadwal otomatis dengan pekerjaan cron atau pekerjaan timer systemd

---

Tip
Anda dapat memulihkan data konfigurasi debconf dengan "`debconf-set-selections debconf-selections`" dan data seleksi dokg dengan "`dpkg --set-selection <dpkg-selections.list`".

Catatan
Jangan mencadangkan konten sistem berkas pseudo yang ditemukan pada `/proc`, `/sys`, `/tmp`, dan `/run` (lihat Bagian 1.2.12 dan Bagian 1.2.13). Kecuali Anda tahu persis apa yang Anda lakukan, mereka adalah data besar yang tidak berguna.

---

1 Media write-once seperti CD/DVD-R dapat mencegah kecelakaan penimpaan. (Lihat Bagian 9.8 tentang cara menulis ke media penyimpanan dari baris perintah shell. Lingkungan GUI desktop GNOME memberi Anda akses mudah melalui menu: "Tempat → Pencipta CD/DVD".)

2 Sebagian data ini tidak dapat diregenerasi dengan memasukkan string masukan yang sama ke sistem.

Catatan
Anda mungkin ingin menghentikan beberapa daemon aplikasi seperti MTA (lihat Bagian 6.2.4) saat mencadangkan data.

10.2.2 Keluarga utilitas pencadangan

Berikut adalah daftar keluarga utilitas pencadangan terkemuka yang tersedia di sistem Debian.

paket	popcon	ukuran	deskripsi
bacula-common	V:6.0, I:6.7	2501	Bacula: pencadangan jaringan, pemulihan, dan verifikasi - berkas dukungan umum
bacula-client	V:0.2, I:1.8	199	Bacula: pencadangan jaringan, pemulihan, dan verifikasi - meta-paket klien
bacula-console	V:0.6, I:2.1	112	Bacula: pencadangan jaringan, pemulihan, dan verifikasi - konsol teks
bacula-server	I:0.59	199	Bacula: pencadangan jaringan, pemulihan, dan verifikasi - meta-paket server
amanda-common	V:0.6, I:2.1	9951	Amanda: Advanced Maryland Automatic Network Disk Archiver (Pustaka)
amanda-client	V:0.5, I:2.1	1109	Amanda: Advanced Maryland Automatic Network Disk Archiver (Klien)
amanda-server	V:0.11, I:0.22	1095	Amanda: Advanced Maryland Automatic Network Disk Archiver (Server)
backuppc	V:1.4, I:1.6	3088	BackupPC adalah sistem kinerja tinggi, tingkat enterprise untuk mencadangkan PC (berbasis disk)
duplicity	V:5, I:44	2662	cadangan (jarak jauh) inkremental
deja-dup	V:28, I:41	5274	Antarmuka GUI untuk duplicity
borgbackup	V:13, I:28	3532	deduplikasi cadangan (jarak jauh)
borgmatic	V:3.2, I:4.4	946	pembantu borgbackup
rdiff-backup	V:2.1, I:6.1	1169	cadangan (jarak jauh) inkremental
restic	V:8, I:15	41617	cadangan (jarak jauh) inkremental
backupninja	V:2.0, I:2.4	360	sistem **meta-backup** ringan dan dapat diperluas
slbackup	V:0.10, I:0.13	147	cadangan (jarak jauh) inkremental
backup-manager	V:0.41, I:0.70	573	alat pencadangan baris perintah
backup2l	V:0.35, I:0.46	110	alat cadangan/pemulihan pemeliharaan rendah untuk media yang dapat dikait (berbasis disk)

Tabel 10.5: Daftar utilitas keluarga pencadangan

Alat cadangan memiliki fokus khusus mereka.

- Mondo Rescue adalah sistem pencadangan untuk memfasilitasi pemulihan sistem lengkap dengan cepat dari CD/DVD cadangan dll. tanpa melalui proses instalasi sistem normal.
- Bacula, Amanda, dan BackupPC adalah utilitas keluarga pencadangan berfitur lengkap yang berfokus pada cadangan reguler melalui jaringan.
- Duplicity, dan Borg adalah utilitas cadangan yang lebih sederhana untuk workstation biasa.

10.2.3 Tips pencadangan

Untuk workstation pribadi, utilitas paket pencadangan berfitur lengkap yang dirancang untuk lingkungan server mungkin tidak berfungsi dengan baik. Pada saat yang sama, utilitas pencadangan yang ada untuk workstation mungkin memiliki beberapa kekurangan.

Berikut ini beberapa tips untuk membuat pencadangan lebih mudah dengan upaya minimal dari pengguna. Teknik-teknik ini dapat digunakan dengan utilitas pencadangan apa pun.

Untuk tujuan demonstrasi, mari kita asumsikan pengguna utama dan nama grup sebagai `penguin` dan membuat contoh skrip cadangan dan snapshot "`/usr/local/bin/bkss.sh`" sebagai:

```
#!/bin/sh -e
SRC="$1" # source data path
DSTFS="$2" # backup destination filesystem path
DSTSV="$3" # backup destination subvolume name
DSTSS="${DSTFS}/${DSTSV}-snapshot" # snapshot destination path
if [ "$(stat -f -c %T "$DSTFS")" != "btrfs" ]; then
  echo "E: $DESTFS needs to be formatted to btrfs" >&2 ; exit 1
fi
MSGID=$(notify-send -p "bkup.sh $DSTSV" "in progress ...")
if [ ! -d "$DSTFS/$DSTSV" ]; then
  btrfs subvolume create "$DSTFS/$DSTSV"
  mkdir -p "$DSTSS"
fi
rsync -aHxS --delete --mkpath "${SRC}/" "${DSTFS}/${DSTSV}"
btrfs subvolume snapshot -r "${DSTFS}/${DSTSV}" ${DSTSS}/$(date -u --iso=min)
notify-send -r "$MSGID" "bkup.sh $DSTSV" "finished!"
```

Di sini, hanya alat dasar rsync(1) yang digunakan untuk memfasilitasi pencadangan sistem dan ruang penyimpanan digunakan secara efisien oleh Btrfs.

Tip
FYI: Penulis ini menggunakan skrip shell serupa miliknya sendiri "bss: Utilitas Snapshot Sub Volume Btrfs" untuk workstation-nya.

10.2.3.1 Pencadangan GUI

Berikut ini adalah contoh untuk menyiapkan pencadangan klik GUI tunggal.

- Siapkan perangkat penyimpanan USB yang akan digunakan untuk cadangan.
  - Format perangkat penyimpanan USB dengan satu partisi di btrfs dengan nama labelnya sebagai "BKUP". Ini dapat dienkripsi (lihat Bagian 9.9.1).
  - Tancapkan ini ke sistem Anda. Sistem desktop akan secara otomatis memasangnya sebagai "`/media/penguin/BKUP`"
  - Jalankan "`sudo chown penguin:penguin /media/penguin/BKUP`" untuk membuatnya dapat ditulis oleh pengguna.
- Buatlah "`~/.local/share/applications/BKUP.desktop`" dengan mengikuti teknik yang tertulis di Bagian 9.4.10 sebagai:

```
[Desktop Entry]
Name=bkss
Comment=Backup and snapshot of ~/Documents
Exec=/usr/local/bin/bkss.sh /home/penguin/Documents /media/penguin/BKUP Documents
Type=Application
```

Untuk setiap klik GUI, data Anda dicadangkan dari "`~/Documents`" ke perangkat penyimpanan USB dan snapshot hanya-baca dibuat.

10.2.3.2 Pencadangan yang dipicu kejadian mount

Berikut ini adalah contoh untuk menyiapkan pencadangan otomatis yang dipicu oleh kejadian mount.

- Siapkan perangkat penyimpanan USB yang akan digunakan untuk pencadangan seperti pada Bagian 10.2.3.1.
- Buat berkas unit layanan sistemd "`~/.config/systemd/user/back-BKUP.service`" sebagai:

```
[Unit]
Description=USB Disk backup
Requires=media-%u-BKUP.mount
After=media-%u-BKUP.mount

[Service]
ExecStart=/usr/local/bin/bkss.sh %h/Documents /media/%u/BKUP Documents
StandardOutput=append:%h/.cache/systemd-snap.log
StandardError=append:%h/.cache/systemd-snap.log

[Install]
WantedBy=media-%u-BKUP.mount
```

- Fungsikan konfigurasi unit systemd ini dengan yang berikut:

```
 $ systemctl --user enable bkup-BKUP.service
```

Untuk setiap kejadian mount, data Anda dicadangkan dari "`~/Documents`" ke perangkat penyimpanan USB dan snapshot hanya-baca dibuat.

Di sini, nama-nama unit mount systemd yang saat ini dimiliki systemd dalam memori dapat ditanyakan kepada manajer layanan dari pengguna yang memanggil dengan "`systemctl --user list-units --type=mount`".

10.2.3.3 Pencadangan yang dipicu oleh kejadian pewaktu

Berikut ini adalah contoh untuk menyiapkan pencadangan otomatis yang dipicu oleh kejadian pewaktu.

- Siapkan perangkat penyimpanan USB yang akan digunakan untuk pencadangan seperti pada Bagian 10.2.3.1.
- Buat berkas unit pewaktu systemd "`~/.config/systemd/user/snap-Documents.timer`" sebagai:

```
[Unit]
Description=Run btrfs subvolume snapshot on timer
Documentation=man:btrfs(1)

[Timer]
OnStartupSec=30
OnUnitInactiveSec=900

[Install]
WantedBy=timers.target
```

- Buat berkas unit layanan systemd "`~/.config/systemd/user/snap-Documents.service`" sebagai:

```
[Unit]
Description=Run btrfs subvolume snapshot
Documentation=man:btrfs(1)

[Service]
Type=oneshot
Nice=15
```

```
ExecStart=/usr/local/bin/bkss.sh %h/Documents /media/%u/BKUP Documents
IOSchedulingClass=idle
CPUSchedulingPolicy=idle
StandardOutput=append:%h/.cache/systemd-snap.log
StandardError=append:%h/.cache/systemd-snap.log
```

- Fungsikan konfigurasi unit systemd ini dengan yang berikut:

```
$ systemctl --user enable snap-Documents.timer
```

Untuk setiap kejadian pewaktu, data Anda dicadangkan dari "`~/Documents`" ke perangkat penyimpanan USB dan snapshot hanya-baca dibuat.

Di sini, nama-nama unit pengguna pewaktu systemd yang saat ini dimiliki systemd di memori dapat ditanyakan kepada manajer layanan dari pengguna yang memanggil dengan "`systemctl --user list-units --type=timer`".

Untuk sistem desktop modern, pendekatan systemd ini dapat menawarkan kontrol yang lebih rinci daripada Unix tradisional yang menggunakan at(1), cron(8), atau anacron(8).

10.3 Infrastruktur keamanan data

Infrastruktur keamanan data disediakan oleh kombinasi alat enkripsi data, alat message digest, dan alat tanda tangan.

paket	popcon	ukuran	perintah	deskripsi
gnupg	V:418, I:860	465	gpg(1)	GNU Privacy Guard - Alat enkripsi dan penandatanganan OpenPGP
gpgv	V:203, I:923	577	gpgv(1)	GNU Privacy Guard - Alat verifikasi tanda tangan OpenPGP
sq	V:1, I:33	24447	sq(1)	Sequoia-PGP - Alat enkripsi dan penandatanganan OpenPGP
sqv	V:479, I:554	1919	sqv(1)	Sequoia-PGP - Alat verifikasi tanda tangan OpenPGP
paperkey	V:1, I:12	58	paperkey(1)	mengekstrak hanya informasi rahasia dari kunci rahasia OpenPGP
cryptsetup	V:34, I:77	469	cryptsetup(8), …	utilitas untuk enkripsi perangkat blok dm-crypt yang mendukung LUKS
coreutils	V:912, I:1000	17994	md5sum(1)	menghitung dan memeriksa digest pesan MD5
coreutils	V:912, I:1000	17994	sha1sum(1)	menghitung dan memeriksa digest pesan SHA1
openssl	V:846, I:996	2564	openssl(1ssl)	menghitung pesan digest dengan "`openssl dgst`" (OpenSSL)
libsecret-tools	V:2, I:13	44	secret-tool(1)	menyimpan dan mengambil kata sandi (CLI)
seahorse	V:76, I:260	7999	seahorse(1)	alat manajemen kunci (GNOME)

Tabel 10.6: Daftar alat infrastruktur keamanan data

Lihat Bagian 9.9 pada dm-crypt dan fscrypt yang menerapkan infrastruktur enkripsi data otomatis melalui modul kernel Linux.

perintah	deskripsi
`gpg --gen-key`	membuat suatu kunci baru
`gpg --gen-revoke my_user_ID`	menghasilkan kunci pencabutan untuk ID_pengguna_saya
`gpg --edit-key user_ID`	menyunting kunci secara interaktif, ”help” untuk bantuan
`gpg -o file --export`	mengekspor semua kunci publik ke berkas
`gpg -o berkas --export-secret-keys`	mengekspor semua kunci privat ke berkas
`gpg --import file`	mengimpor semua kunci dari berkas
`gpg --send-keys user_ID`	mengirim kunci user_ID ke keyserver
`gpg --recv-keys user_ID`	terima kunci user_ID dari keyserver
`gpg --list-keys user_ID`	daftar kunci dari ID_pengguna
`gpg --list-sigs user_ID`	daftar ttd. dari ID_pengguna
`gpg --check-sigs user_ID`	periksa ttd. dari ID_pengguna
`gpg --fingerprint user_ID`	periksa sidik jari ID_pengguna
`gpg --refresh-keys`	memperbarui keyring lokal

Tabel 10.7: Daftar perintah GNU Privacy Guard untuk manajemen kunci

10.3.1 Manajemen kunci untuk GnuPG

Berikut adalah perintah GNU Privacy Guard untuk manajemen kunci dasar.

Berikut adalah arti dari kode kepercayaan.

kode	deskripsi kepercayaan
`-`	tidak ada kepercayaan pemilik yang ditugaskan / belum dihitung
`e`	perhitungan kepercayaan gagal
`q`	tidak cukup informasi untuk perhitungan
`n`	jangan pernah mempercayai kunci ini
`m`	dipercaya secara terbatas
`f`	dipercaya sepenuhnya
`u`	dipercaya setinggi-tingginya

Tabel 10.8: Daftar arti kode kepercayaan

Yang berikut ini membuat kunci saya ”`9563FC29932C409F1A77F9C1AB8A1522D8234C6A`”.

```
$ gpg --gen-key
gpg (GnuPG) 2.4.7; Copyright (C) 2024 g10 Code GmbH
...
GnuPG needs to construct a user ID to identify your key.

Real name: Osamu Aoki
Email address: osamu@debian.org
You selected this USER-ID:
    "Osamu Aoki <osamu@debian.org>"

Change (N)ame, (E)mail, or (O)kay/(Q)uit? o
...
public and secret key created and signed.

pub   ed25519 2026-02-14 [SC] [expires: 2029-02-13]
      9563FC29932C409F1A77F9C1AB8A1522D8234C6A
uid                      Osamu Aoki <osamu@debian.org>
sub   cv25519 2026-02-14 [E] [expires: 2029-02-13]
```

Yang berikut ini mengunggah kunci saya ”`9563FC29932C409F1A77F9C1AB8A1522D8234C6A`” ke keyserver populer ”`hkp://keys.gnupg.net`”.

```
$ gpg --keyserver hkp://keys.gnupg.net --send-keys 9563FC29932C409F1A77F9C1AB8A1522D8234C6A
```

Keyserver baku yang baik diatur dalam "`~/.gnupg/gpg.conf`" (atau lokasi lama "`~/.gnupg/options`") berisi hal berikut.

```
keyserver hkp://keys.gnupg.net
```

Yang berikut ini mendapatkan kunci yang tidak diketahui dari keyserver.

```
$ gpg --list-sigs --with-colons | grep '^sig.*\[User ID not found\]' |\
          cut -d ':' -f 5| sort | uniq | xargs gpg --recv-keys
```

Ada bug di OpenPGP Public Key Server (pra-versi 0.9.6) yang merusak kunci dengan lebih dari 2 sub-kunci. Paket `gnupg` (>1.2.1-2) yang lebih baru dapat menangani subkunci yang rusak ini. Lihat gpg(1) di bawah opsi "`--repair-pks-su`

Penggunaan SHA-1 untuk has telah usang. Bila kunci openPGP RSA lama Anda memakai SHA-1 untuk hash, perbaiki itu memakai FixKeyWithSha1.

Tip
Perintah sq(1) dan sqv(1) adalah set alternatif dari perintah openPGP. Lihat dokumentasi pengguna sq dan A Practical Introduction to using sq, Sequoia PGP's CLI.

10.3.2 Menggunakan GnuPG pada berkas

Berikut adalah contoh untuk menggunakan perintah GNU Privacy Guard pada berkas.

10.3.3 Menggunakan GnuPG dengan Mutt

Tambahkan yang berikut ke "`~/.muttrc`" untuk menjaga GnuPG yang lambat agar tidak dimulai secara otomatis, sambil membiarkannya digunakan dengan mengetik "S" di menu indeks.

```
macro index S ":toggle pgp_verify_sig\n"
set pgp_verify_sig=no
```

10.3.4 Menggunakan GnuPG dengan Vim

Plugin `gnupg` memungkinkan Anda menjalankan GnuPG secara transparan untuk berkas dengan ekstensi "`.gpg`", "`.asc`", dan "`.pgp`".[3]

```
$ sudo aptitude install vim-scripts
$ echo "packadd! gnupg" >> ~/.vim/vimrc
```

10.3.5 Sidikjari MD5

md5sum(1) menyediakan utilitas untuk membuat berkas digest menggunakan metode dalam rfc1321 dan memverifikasi setiap berkas dengan itu.

3Jika Anda menggunakan "`~/.vimrc`" bukan "`~/.vim/vimrc`", silakan ganti sesuai dengan itu.

perintah	deskripsi
`gpg -a -s file`	menandatangani berkas ke file.asc berlapis ASCII
`gpg --armor --sign file`	,,
`gpg --clearsign file`	tandatangani pesan dengan clear-sign
`gpg --clearsign file|mail foo@example.org`	menyurel pesan yang ditandatangani secara clear-sign ke `foo@example.org`
`gpg --clearsign --not-dash-escaped patchfile`	menandatangani berkas patch dengan cara clear-sign
`gpg --verify file`	memverifikasi berkas yang ditandatangani dengan cara clear-sign
`gpg -o file.sig -b file`	membuat tanda tangan terpisah
`gpg -o file.sig --detach-sign file`	,,
`gpg --verify berkas.sig berkas`	verifikasi berkas dengan berkas.sig
`gpg -o crypt_file.gpg -r name -e file`	enkripsi kunci publik yang ditujukan untuk nama dari berkas ke berkas_crypt biner
`gpg -o crypt_file.gpg --recipient name --encrypt file`	,,
`gpg -o crypt_file.asc -a -r name -e file`	enkripsi kunci publik yang ditujukan untuk nama dari berkas ke berkas_crypt.asc terbungkus ASCII
`gpg -o crypt_file.gpg -c file`	enkripsi simetris dari berkas ke berkas_crypt.gpg
`gpg -o crypt_file.gpg --symmetric file`	,,
`gpg -o crypt_file.asc -a -c file`	enkripsi simetris yang ditujukan untuk nama dari berkas ke berkas_crypt.asc terbungkus ASCII
`gpg -o file -d crypt_file.gpg -r name`	dekripsi
`gpg -o file --decrypt crypt_file.gpg`	,,

Tabel 10.9: Daftar perintah GNU Privacy Guard pada berkas

```
$ md5sum foo bar >baz.md5
$ cat baz.md5
d3b07384d113edec49eaa6238ad5ff00  foo
c157a79031e1c40f85931829bc5fc552  bar
$ md5sum -c baz.md5
foo: OK
bar: OK
```

Catatan
Perhitungan untuk jumlah MD5 kurang CPU intensif daripada yang untuk tanda tangan kriptografi oleh GNU Privacy Guard (GnuPG). Biasanya, hanya berkas digest tingkat atas yang ditandatangani secara kriptografis untuk memastikan integritas data.

10.3.6 Ring kunci kata sandi

Pada sistem GNOME, alat GUI seahorse(1) mengelola kata sandi dan menyimpannya dengan aman di ring kunci `~/.local/share/keyrings/*`.

secret-tool(1) dapat menyimpan kata sandi ke ring kunci dari baris perintah.

Mari kita simpan frasa sandi yang dipakai untuk image disk terenkripsi LUKS/dm-crypt

```
$ secret-tool store --label='LUKS passphrase for disk.img' LUKS my_disk.img
Password: ********
```

Kata sandi yang tersimpan ini dapat diambil dan diumpankan ke program lain, misalnya, cryptsetup(8).

```
$ secret-tool lookup LUKS my_disk.img | \
  cryptsetup open disk.img disk_img --type luks --keyring -
$ sudo mount /dev/mapper/disk_img /mnt
```

Tip
Setiap kali Anda perlu memberikan kata sandi dalam skrip, gunakan `secret-tool` dan hindari langsung menanam frasa sandi di dalamnya.

10.4 Alat penggabungan kode sumber

Ada banyak alat gabungan untuk kode sumber. Perintah berikut menarik perhatian saya.

10.4.1 Mengekstrak perbedaan untuk berkas sumber

Prosedur berikut mengekstrak perbedaan antara dua berkas sumber dan membuat berkas diff terpadu "`file.patch0`" atau "`file.patch1`" tergantung pada lokasi berkas.

```
$ diff -u file.old file.new > file.patch0
$ diff -u old/file new/file > file.patch1
```

paket	popcon	ukuran	perintah	deskripsi
patch	V:102, I:727	242	patch(1)	menerapkan berkas diff ke yang asli
vim	V:83, I:338	4164	vimdiff(1)	membandingkan 2 berkas berdampingan di vim
imediff	V:0.02, I:0.25	329	imediff(1)	alat gabungan 2/3 arah layar penuh interaktif
meld	V:5, I:23	3996	meld(1)	membandingkan dan menggabungkan berkas (GTK)
wiggle	V:0.02, I:0.12	175	wiggle(1)	menerapkan patch yang ditolak
diffutils	V:896, I:998	1768	diff(1)	membandingkan berkas baris per baris
diffutils	V:896, I:998	1768	diff3(1)	membandingkan dan menggabungkan tiga berkas baris demi baris
quilt	V:2, I:16	880	quilt(1)	mengelola serangkaian patch
wdiff	V:5, I:38	580	wdiff(1)	menampilkan perbedaan kata antara berkas-berkas teks
diffstat	V:9, I:93	79	diffstat(1)	menghasilkan histogram perubahan oleh diff
patchutils	V:11, I:92	269	combinediff(1)	membuat patch kumulatif dari dua patch inkremental
patchutils	V:11, I:92	269	dehtmldiff(1)	mengekstrak diff dari halaman HTML
patchutils	V:11, I:92	269	filterdiff(1)	mengekstrak atau mengecualikan diff dari berkas diff
patchutils	V:11, I:92	269	fixcvsdiff(1)	memperbaiki berkas diff yang dibuat oleh CVS yang oleh patch(1) disalahtafsir
patchutils	V:11, I:92	269	flipdiff(1)	menukar urutan dua patch
patchutils	V:11, I:92	269	grepdiff(1)	menunjukkan berkas mana yang dimodifikasi oleh patch yang cocok dengan regex
patchutils	V:11, I:92	269	interdiff(1)	menunjukkan perbedaan antara dua berkas diff unified
patchutils	V:11, I:92	269	lsdiff(1)	menunjukkan berkas mana yang dimodifikasi oleh patch
patchutils	V:11, I:92	269	recountdiff(1)	hitung ulang cacah dan ofset dalam diff konteks terpadu
patchutils	V:11, I:92	269	rediff(1)	memperbaiki ofset dan cacah diff yang disunting dengan tangan
patchutils	V:11, I:92	269	splitdiff(1)	memisahkan patch inkremental
patchutils	V:11, I:92	269	unwrapdiff(1)	demangle patch yang telah berganti baris
dirdiff	V:0.1, I:1.3	167	dirdiff(1)	menampilkan perbedaan dan menggabungkan perubahan antara pohon direktori
docdiff	V:0.01, I:0.16	554	docdiff(1)	membandingkan dua berkas kata demi kata / huruf demi huruf
makepatch	V:0.01, I:0.14	99	makepatch(1)	menghasilkan berkas patch yang diperluas
makepatch	V:0.01, I:0.14	99	applypatch(1)	menerapkan berkas patch yang diperluas

Tabel 10.10: Daftar alat penggabungan kode sumber

10.4.2 Menggabungkan pembaruan untuk berkas sumber

Berkas diff (atau disebut berkas patch) digunakan untuk mengirim pembaruan program. Pihak penerima menerapkan pembaruan ini ke berkas lain dengan yang berikut ini.

```
$ patch -p0 file < file.patch0
$ patch -p1 file < file.patch1
```

10.4.3 Penggabungan interaktif

Jika Anda memiliki dua versi kode sumber, Anda dapat melakukan penggabungan 2 arah secara interaktif menggunakan imediff(1) dengan yang berikut.

```
$ imediff -o file.merged file.old file.new
```

Jika Anda memiliki tiga versi kode sumber, Anda dapat melakukan penggabungan 3 arah secara interaktif menggunakan imediff(1) dengan yang berikut.

```
$ imediff -o file.merged file.yours file.base file.theirs
```

10.5 Git

Git adalah alat pilihan hari ini untuk sistem kontrol versi (version control system/VCS) karena Git dapat melakukan segalanya untuk manajemen kode sumber lokal dan jarak jauh.

Debian menyediakan layanan Git bebas melalui layanan Debian Salsa. Dokumentasinya dapat ditemukan di https://wiki.debian.org/Salsa.

Berikut adalah beberapa paket terkait Git.

10.5.1 Konfigurasi klien Git

Anda mungkin ingin mengatur beberapa konfigurasi global di "`~/.gitconfig`" seperti nama dan alamat surel Anda yang digunakan oleh Git dengan yang berikut ini.

```
$ git config --global user.name "Name Surname"
$ git config --global user.email yourname@example.com
```

Anda juga dapat menyesuaikan perilaku baku Git dengan yang berikut ini.

```
$ git config --global init.defaultBranch main
$ git config --global pull.rebase true
$ git config --global push.default current
```

Jika Anda terlalu terbiasa dengan perintah CVS atau Subversion, Anda mungkin ingin menetapkan beberapa alias perintah dengan yang berikut.

```
$ git config --global alias.ci "commit -a"
$ git config --global alias.co checkout
```

Anda dapat memeriksa konfigurasi global Anda dengan berikut ini.

```
$ git config --global --list
```

paket	popcon	ukuran	perintah	deskripsi
git	V:422, I:625	52512	git(7)	Git, sistem kontrol revisi yang cepat, mampu menangani beban skala besar, dan terdistribusi
gitk	V:3, I:26	2045	gitk(1)	Peramban repositori GUI Git dengan riwayat
qgit	V:0.2, I:1.9	1431	qgit(1)	Peramban repositori GUI Git dengan riwayat
git-cola	V:0.8, I:4.5	5129	git-cola(1)	Peramban repositori GUI Git dengan riwayat
tig	V:2, I:11	1245	tig(1)	peramban repositori Git konsol dengan riwayat
lazygit	V:1.0, I:4.1	24167	lazygit(1)	peramban repositori Git konsol dengan riwayat
git-gui	V:1, I:16	2549	git-gui(1)	GUI untuk Git (Tanpa riwayat)
git-email	V:0.3, I:9.3	1229	git-send-email(1)	mengirim kumpulan patch sebagai surel dari Git
git-buildpackage	V:1.2, I:6.8	2027	git-buildpackage(1)	mengotomatiskan pemaketan Debian dengan Git
dgit	V:0.12, I:0.97	757	dgit(1)	interoperabilitas git dengan arsip Debian
imediff	V:0.02, I:0.25	329	git-ime(1)	alat pembantu pemecah commit git interaktif
stgit	V:0.08, I:0.50	6253	stg(1)	quilt di atas git (Python)
git-doc	I:10	15192	T/T	dokumentasi resmi untuk Git
gitmagic	I:0.48	721	T/T	”Git Magic”, panduan yang lebih mudah dipahami untuk Git

Tabel 10.11: Daftar paket dan perintah terkait git

10.5.2 Perintah Git dasar

Operasi Git melibatkan beberapa data.

- Pohon kerja yang memuat berkas yang menghadap pengguna dan tempat Anda membuat perubahan padanya.
  - Perubahan yang akan dicatat harus dipilih secara eksplisit dan dipentaskan ke indeks. Ini adalah perintah `git add` dan `git rm`.
- Indeks penyimpan berkas yang di-stage.
  - Berkas yang di-stage akan di-commit ke repositori lokal atas permintaan berikutnya. Ini adalah perintah `git commit`.
- Repositori lokal yang menyimpan berkas yang di-commit.
  - Git mencatat riwayat tertaut dari data yang di-commit dan mengaturnya sebagai branch dalam repositori.
  - Repositori lokal dapat mengirim data ke repositori jarak jauh dengan perintah `git push`.
  - Repositori lokal dapat menerima data dari repositori jarak jauh dengan perintah `git fetch` dan `git pull`.
    - Perintah `git pull` melakukan perintah `git merge` atau `git rebase` setelah perintah `git fetch`.
    - Di sini, `git merge` menggabungkan dua branch sejarah yang terpisah di akhir hingga satu titik. (Ini adalah baku dari `git pull` tanpa penyesuaian dan mungkin baik untuk orang hulu yang mempublikasikan cabang kepada banyak orang.)
    - Di sini, `git rebase` menciptakan satu cabang tunggal sejarah berurutan dari cabang remote yang diikuti oleh cabang lokal. (Ini adalah kasus penyesuaian `pull.rebase true` dan mungkin baik untuk kita semua.)
- Repositori jarak jauh yang menyimpan berkas yang di-commit.
  - Komunikasi ke repositori jarak jauh menggunakan protokol komunikasi yang aman seperti SSH atau HTTPS.

Pohon kerja adalah berkas-berkas di luar direktori `.git/`. Berkas di dalam direktori `.git/` menyimpan indeks, data repositori lokal, dan beberapa berkas teks konfigurasi git.

Berikut adalah ikhtisar perintah Git utama.

Perintah Git	fungsi
`git init`	membuat repositori (lokal)
`git clone URL`	mengklon repositori jarak jauh ke repositori lokal dengan pohon kerja
`git pull origin main`	memperbarui brach `main` lokal berdasarkan repositori `origin` jarak jauh
`git add .`	menambahkan berkas di pohon kerja ke indeks untuk berkas yang sudah ada sebelumnya dalam indeks saja
`git add -A .`	menambahkan berkas di pohon kerja ke indeks untuk semua berkas termasuk penghapusan
`git rm filename`	menghapus berkas dari pohon kerja dan indeks
`git commit`	meng-commit perubahan yang di-stage dalam indeks ke repositori lokal
`git commit -a`	menambahkan semua perubahan pada pohon kerja ke indeks dan meng-commit mereka ke repositori lokal (menambah + commit)
`git push -u origin branch_name`	memperbarui repositori `origin` jarak jauh oleh branch `branch_name` lokal (pemanggilan awal)
`git push origin branch_name`	memperbarui repositori `origin` jarak jauh oleh branch `branch_name` lokal (pemanggilan berikutnya)
`git diff treeish1 treeish2`	menunjukkan perbedaan antara commit *treeish1* dan commit *treeish2*
`gitk`	Tampilan GUI dari pohon sejarah branch repositori VCS

Tabel 10.12: Perintah Git Utama

10.5.3 Tips Git

Berikut adalah tips Git.



Awas
Jangan gunakan string tag dengan spasi di dalamnya bahkan jika beberapa alat seperti gitk(1) mengizinkan Anda untuk menggunakannya. Ini mungkin mencekik beberapa perintah `git` lainnya.



Perhatian
Jika branch lokal yang telah di-push ke repositori jarak jauh di-rebase atau di-squash, mem-push branch ini memiliki risiko dan membutuhkan opsi `--force`. Ini biasanya tidak dapat diterima untuk branch `main` tetapi mungkin dapat diterima untuk branch topik sebelum merge ke branch `main`.



Perhatian
Menjalankan suatu sub perintah `git` langsung sebagai ”`git-xyz`” dari baris perintah telah usang sejak awal 2006.

Tip
Jika ada berkas yang dapat dieksekusi `git-foo` di path yang ditentukan oleh `$PATH`, memasukkan ”`git foo`” tanpa tanda hubung ke baris perintah memanggil `git-foo` ini. Ini adalah fitur dari perintah `git`.

Baris perintah Git	**fungsi**
`gitk --all`	melihat sejarah Git lengkap dan beroperasi pada mereka seperti mengatur ulang HEAD ke commit lain, cherry-pick patch, membuat tag dan branch ...
`git stash`	mengambil pohon kerja bersih tanpa kehilangan data
`git remote -v`	memeriksa pengaturan untuk remote
`git branch -vv`	memeriksa pengaturan untuk branch
`git status`	menampilkan status pohon kerja
`git config -l`	daftar pengaturan git
`git reset --hard HEAD; git clean -x -d -f`	mengembalikan semua perubahan pohon kerja dan membersihkannya sepenuhnya
`git rm --cached filename`	memulihkan indeks ter-stage yang diubah oleh `git add nama_berkas`
`git reflog`	dapatkan log referensi (berguna untuk memulihkan commit dari branch yang dihapus)
`git branch new_branch_name HEAD@{6}`	membuat branch baru dari informasi reflog
`git remote add new_remote URL`	menambahkan repositori jarak jauh `new_remote` yang diacu oleh URL
`git remote rename origin upstream`	mengganti nama repositori jarak jauh dari `origin` ke `upstream`
`git branch -u upstream/branch_name`	mengatur pelacakan jarak jauh ke repositori jarak jauh di `upstream` dan nama branch-nya `branch_name`.
`git remote set-url origin https://foo/bar.git`	mengubah URL `origin`
`git remote set-url --push upstream DISABLED`	nonaktifkan push ke `upstream` (Sunting `.git/config` untuk mengaktifkan kembali)
`git remote update upstream`	mengambil pembaruan dari semua branch remote dalam repositori `upstream`
`git fetch upstream foo:upstream-foo`	membuat branch `upstream-foo` lokal (mungkin orphan) sebagai salinan branch `foo` dalam repositori `upstream`
`git checkout -b topic_branch ; git push -u topic_branch origin`	membuat `topic_branch` baru dan mem-push ke `origin`
`git branch -m oldname newname`	mengganti nama nama cabang lokal
`git push -d origin branch_to_be_removed`	menghapus branch remote (metode baru)
`git push origin :branch_to_be_removed`	menghapus branch remote (metode lama)
`git checkout --orphan unconnected`	membuat branch baru `yang tidak terhubung`
`git rebase -i origin/main`	reorder/drop/squish commit dari `origin/main` untuk membersihkan riwayat branch
`git reset HEAD^; git commit --amend`	squash 2 commit terakhir menjadi satu
`git checkout topic_branch ; git merge --squash topic_branch`	squash seluruh `topic_branch` menjadi satu commit
`git fetch --unshallow --update-head-ok origin '+refs/heads/*:refs/heads/*'`	mengubah klon shallow menjadi klon penuh dari semua branch
`git ime`	membagi commit terakhir menjadi serangkaian commit berkas demi berkas yang lebih kecil dll. (paket `imediff` diperlukan)
`git repack -a -d; git prune`	mengemas ulang repositori lokal ke dalam satu pack (ini dapat membatasi kemungkinan pemulihan data yang hilang dari branch yang terhapus, dll.)

Tabel 10.13: Tips Git

10.5.4 Referensi Git

Lihat yang berikut.

- halaman man: git(1) (`/usr/share/doc/git-doc/git.html`)
- Manual Pengguna Git (`/usr/share/doc/git-doc/user-manual.html`)
- Pengantar tutorial untuk git (`/usr/share/doc/git-doc/gittutorial.html`)
- Pengantar tutorial untuk git: bagian dua (`/usr/share/doc/git-doc/gittutorial-2.html`)
- GIT Sehari-hari Dengan Sekitar 20 Perintah (`/usr/share/doc/git-doc/giteveryday.html`)
- Git Magic (`/usr/share/doc/gitmagic/html/index.html`)

10.5.5 Sistem kontrol versi lainnya

Sistem kontrol versi (VCS) kadang-kadang dikenal sebagai sistem kontrol revisi (RCS), atau manajemen konfigurasi perangkat lunak (SCM).

Berikut adalah ringkasan VCS non-Git lainnya yang terkenal pada sistem Debian.

paket	popcon	ukuran	perkakas	Tipe VCS	kommen
`mercurial`	V:3, I:25	1375	Mercurial	terdistribusi	DVCS dalam Python dan sebagian C
`darcs`	V:0.2, I:3.1	38856	Darcs	terdistribusi	DVCS dengan aljabar patch cerdas (lambat)
`tla`	V:0.03, I:0.67	1022	GNU arch	terdistribusi	DVCS terutama oleh Tom Lord (historis)
`bzr`	V:0.0, I:3.9	28	GNU Bazaar	terdistribusi	DVCS yang dipengaruhi oleh `tla` ditulis dalam Python (historis)
`subversion`	V:9, I:54	4856	Subversion	remote	”CVS dilakukan dengan benar”, VCS jarak jauh standar yang lebih baru (bersejarah)
`cvs`	V:3, I:23	4835	CVS	remote	VCS jarak jauh standar sebelumnya (bersejarah)
`tkcvs`	V:0.11, I:0.83	34	CVS, …	remote	Tampilan GUI dari pohon repositori VCS (CVS, Subversion, RCS)
`rcs`	V:1.6, I:9.0	578	RCS	lokal	”Unix SCCS dilakukan dengan benar” (historis)
`cssc`	V:0.01, I:0.28	2044	CSSC	lokal	klon dari Unix SCCS (historis)

Tabel 10.14: Daftar alat sistem kontrol versi lainnya

Bab 11

Konversi data

Alat dan tips untuk mengonversi format data pada sistem Debian dijelaskan.

Alat berbasis standar dalam kondisi yang sangat baik tetapi dukungan untuk format data proprietari terbatas.

11.1 Alat konversi data teks

Paket-paket berikut untuk konversi data teks menarik perhatian saya.

paket	popcon	ukuran	kata kunci	deskripsi
libc6	V:936, I:1000	5483	set karakter	konverter pengodean teks antar lokal dengan iconv(1) (fundamental)
recode	V:2, I:12	528	charset+eol	konverter pengodean teks antar lokal (serbaguna, lebih banyak alias dan fitur)
konwert	V:2, I:42	137	set karakter	konverter pengodean teks antar lokal (mewah)
nkf	V:0.4, I:8.6	360	set karakter	penerjemah set karakter untuk bahasa Jepang
tcs	V:0.01, I:0.14	518	set karakter	penerjemah set karakter
unaccent	V:0.03, I:0.27	35	set karakter	mengganti huruf beraksen dengan ekuivalennya yang tidak berakses
tofrodos	V:1, I:12	50	eol	pengubah format teks antara DOS dan Unix: fromdos(1) dan todos(1)
macutils	V:0.04, I:0.39	319	eol	pengubah format teks antara Macintosh dan Unix: frommac(1) dan tomac(1)

Tabel 11.1: Daftar alat konversi data teks

11.1.1 Mengonversi berkas teks dengan iconv

Tip
iconv(1) disediakan sebagai bagian dari paket `libc6` dan selalu tersedia di hampir semua sistem mirip Unix untuk mengonversi pengodean karakter.

Anda dapat mengonversi pengodean berkas teks dengan iconv(1) dengan yang berikut ini.

```
$ iconv -f encoding1 -t encoding2 input.txt >output.txt
```

Nilai pengodean tidak membedakan huruf besar kecil dan mengabaikan "-" dan "_" untuk pencocokkan. Pengodean yang didukung dapat diperiksa oleh perintah "`iconv -l`".

nilai pengodean	pengunaan
ASCII	American Standard Code for Information Interchange, kode 7 bit tanpa karakter beraksen
UTF-8	standar multibahasa saat ini untuk semua OS modern
ISO-8859-1	standar lama untuk bahasa Eropa Barat, ASCII + karakter beraksen
ISO-8859-2	standar lama untuk bahasa Eropa Timur, karakter beraksen ASCII +
ISO-8859-15	standar lama untuk bahasa Eropa Barat, ISO-8859-1 dengan tanda euro
CP850	code page 850, karakter Microsoft DOS dengan grafis untuk bahasa Eropa Barat, varian ISO-8859-1
CP932	code page 932, varian Shift-JIS gaya Microsoft Windows untuk bahasa Jepang
CP936	code page 936, varian GB2312, GBK, atau GB18030 gaya Microsoft Windows untuk bahasa Cina yang Disederhanakan
CP949	code page 949, varian EUC-KR atau Unified Hangul Code gaya Microsoft Windows untuk bahasa Korea
CP950	code page 950, varian Big5 gaya Microsoft Windows untuk bahasa Cina Tradisional
CP1251	code page 1251, pengodean gaya Microsoft Windows untuk alfabet Cyrillic
CP1252	code page 1252, varian ISO-8859-15 gaya Microsoft Windows untuk bahasa-bahasa Eropa Barat
KOI8-R	standar UNIX bahasa Rusia lama untuk alfabet Cyrillic
ISO-2022-JP	pengodean standar untuk surel berbahasa Jepang yang hanya menggunakan kode 7 bit
eucJP	kode 8 bit standar UNIX Jepang lama dan benar-benar berbeda dari Shift-JIS
Shift-JIS	JIS X 0208 Lampiran 1 standar untuk bahasa Jepang (lihat CP932)

Tabel 11.2: Daftar nilai pengodean dan penggunaannya

Catatan
Beberapa pengodean hanya didukung untuk konversi data dan tidak digunakan sebagai nilai lokal (Bagian 8.1).

Untuk set karakter yang masuk dalam byte tunggal seperti set karakter ASCII dan ISO-8859, pengodean karakter berarti hampir sama dengan set karakter.

Untuk set karakter dengan banyak karakter seperti JIS X 0213 untuk Jepang atau Universal Character Set (UCS, Unicode, ISO-10646-1) untuk hampir semua bahasa, ada banyak skema pengodean yang sesuai dengan mereka ke dalam urutan data byte.

- EUC dan ISO/IEC 2022 (juga dikenal sebagai JIS X 0202) untuk bahasa Jepang
- UTF-8, UTF-16/UCS-2, dan UTF-32/UCS-4 untuk Unicode

Untuk ini, ada diferensiasi yang jelas antara set karakter dan pengodean karakter.

Code page digunakan sebagai sinonim untuk tabel pengodean karakter untuk beberapa vendor tertentu.

Catatan
Harap dicatat sebagian besar sistem pengodean berbagi kode yang sama dengan ASCII untuk karakter 7 bit. Tapi ada beberapa pengecualian. Jika Anda mengonversi program C Jepang lama dan data URL dari format pengodean yang disebut santai shift-JIS ke format UTF-8, gunakan "`CP932`" sebagai nama pengodean alih-alih "`shift-JIS`" untuk mendapatkan hasil yang diharapkan: `0x5C` → "\" dan `0x7E` → "~". Jika tidak, ini diubah menjadi karakter yang salah.

Tip
recode(1) dapat digunakan juga dan menawarkan lebih dari fungsi gabungan iconv(1), fromdos(1), todos(1), frommac(1), dan tomac(1). Untuk informasi lebih lanjut, lihat "`info recode`".

11.1.2 Memeriksa berkas apakah UTF-8 dengan iconv

Anda dapat memeriksa apakah berkas teks dikodekan dalam UTF-8 dengan iconv(1) dengan yang berikut ini.

```
$ iconv -f utf8 -t utf8 input.txt >/dev/null || echo "non-UTF-8 found"
```

Tip
Gunakan opsi "`--verbose`" dalam contoh di atas untuk menemukan karakter non-UTF-8 pertama.

11.1.3 Mengonversi nama berkas dengan iconv

Berikut adalah contoh skrip untuk mengonversi pengkodean nama berkas dari yang dibuat di bawah OS lama ke UTF-8 modern dalam satu direktori.

```
#!/bin/sh
ENCDN=iso-8859-1
for x in *;
 do
 mv "$x" "$(echo "$x" | iconv -f $ENCDN -t utf-8)"
done
```

Variabel "`$ENCDN`" menentukan pengodean asli yang digunakan untuk nama berkas di bawah OS yang lebih tua seperti dalam Tabel 11.2.

Untuk kasus yang lebih rumit, silakan kait suatu sistem berkas (misalnya partisi pada disk drive) yang berisi nama berkas tersebut dengan pengodean yang tepat sebagai opsi mount(8) (lihat Bagian 8.1.3) dan menyalin seluruh isinya ke sistem berkas lain yang dikait sebagai UTF-8 dengan perintah "`cp -a`".

11.1.4 Konversi EOL

Format berkas teks, khususnya kode end-of-line (EOL/akhir baris), tergantung pada platform.

Program konversi format EOL, fromdos(1), todos(1), frommac(1), dan tomac(1), cukup berguna. recode(1) juga berguna.

Catatan
Sebagian data pada sistem Debian, seperti data halaman wiki untuk paket `python-moinmoin`, menggunakan CR-LF gaya MSDOS sebagai kode EOL. Jadi aturan di atas hanyalah aturan umum.

platform	Kode EOL	kontrol	desimal	heksadesimal
Debian (unix)	LF	^J	10	0A
MSDOS dan Windows	CR-LF	^M^J	13 10	0D 0A
Apple Macintosh	CR	^M	13	0D

Tabel 11.3: Daftar gaya EOL untuk platform yang berbeda

Catatan
Sebagian besar penyunting (misalnya `vim`, `emacs`, `gedit`, ...) dapat menangani berkas dalam EOL gaya MSDOS secara transparan.

Tip
Penggunaan "`sed -e '/\r$/!s/$/\r/'`" daripada todos(1) lebih baik ketika Anda ingin menyatukan gaya EOL ke gaya MSDOS dari campuran gaya MSDOS dan Unix. (misalnya, setelah menggabungkan 2 berkas gaya MSDOS dengan diff3(1).) Ini karena `todos` menambahkan CR ke semua baris.

11.1.5 Konversi TAB

Ada beberapa program khusus yang populer untuk mengonversi kode-kode tab.

fungsi	`bsdmainutils`	`coreutils`
mengekspansi tab ke spasi	"`col -x`"	`expand`
membatalkan ekspansi tab dari spasi	"`col -h`"	`unexpand`

Tabel 11.4: Daftar perintah konversi TAB dari paket `bsdmainutils` dan `coreutils`

indent(1) dari paket `indent` benar-benar memformat ulang spasi dalam program C.

Program penyunting seperti `vim` dan `emacs` dapat digunakan untuk konversi TAB juga. Misalnya dengan `vim`, Anda dapat memperluas TAB dengan urutan perintah "`:set expandtab`" dan "`:%retab`". Anda dapat mengembalikan ini dengan urutan perintah "`:set noexpandtab`" dan "`:%retab!`".

11.1.6 Penyunting dengan konversi otomatis

Penyunting modern cerdas seperti program `vim` cukup cerdas dan mengatasi dengan baik dengan sistem pengodean dan format berkas apa pun. Anda harus menggunakan penyunting ini di bawah lokal UTF-8 di konsol yang mampu UTF-8 untuk kompatibilitas terbaik.

Sebuah berkas teks Unix Eropa barat lama, "`u-file.txt`", disimpan dalam pengodean latin1 (iso-8859-1) dapat disunting memakai `vim` dengan yang berikut.

```
$ vim u-file.txt
```

Hal ini dimungkinkan karena mekanisme deteksi otomatis dari pengodean berkas di `vim` mengasumsikan pengodean UTF-8 terlebih dahulu dan, jika gagal, mengasumsikannya sebagai latin1.

Berkas teks Unix Polandia lama, "`pu-file.txt`", disimpan dalam pengodean latin2 (iso-8859-2) dapat disunting memakai `vim` dengan yang berikut.

```
$ vim '+e ++enc=latin2 pu-file.txt'
```

Berkas teks unix Jepang lama, "`ju-file.txt`", disimpan dalam pengodean eucJP dapat disunting memakai `vim` dengan yang berikut.

```
$ vim '+e ++enc=eucJP ju-file.txt'
```

Berkas teks MS-Windows Jepang lama, "`jw-file.txt`", disimpan dalam apa yang disebut pengodean shift-JIS (lebih tepatnya: CP932) dapat disunting memakai `vim` dengan yang berikut.

```
$ vim '+e ++enc=CP932 ++ff=dos jw-file.txt'
```

Ketika sebuah berkas dibuka dengan opsi "`++enc`" dan "`++ff`", "`:w`" di baris perintah Vim menyimpannya dalam format asli dan menimpa berkas asli. Anda juga dapat menentukan format penyimpanan dan nama berkas di baris perintah Vim, misalnya, "`:w ++enc=utf8 new.txt`".

Silakan lihat mbyte.txt "dukungan teks multi-byte" dalam bantuan daring `vim` dan Tabel 11.2 untuk nilai lokal yang digunakan dengan "`++enc`".

Program keluarga `emacs` dapat melakukan fungsi yang setara.

11.1.7 Ekstraksi teks polos

Yang berikut ini membaca halaman web ke dalam berkas teks. Ini sangat berguna saat menyalin konfigurasi dari Web atau menerapkan alat teks Unix dasar seperti grep(1) di halaman web.

```
$ w3m -dump https://www.remote-site.com/help-info.html >textfile
```

Demikian pula, Anda dapat mengekstrak data teks polos dari format lain menggunakan yang berikut.

paket	popcon	ukuran	kata kunci	fungsi
`w3m`	V:10, I:129	2853	`html→text`	Konverter HTML ke teks dengan perintah "`w3m -dump`"
`html2text`	V:3, I:66	298	`html→text`	konverter tingkat lanjut HTML ke teks (ISO 8859-1)
`lynx`	V:29, I:455	2031	`html→text`	Pengubah HTML ke teks dengan perintah "`lynx -dump`"
`elinks`	V:2, I:15	1791	`html→text`	Pengubah HTML ke teks dengan perintah "`elinks -dump`"
`links`	V:2, I:20	2321	`html→text`	Pengubah HTML ke teks dengan perintah "`links -dump`"
`links2`	V:0.8, I:9.4	5466	`html→text`	Pengubah HTML ke teks dengan perintah "`links2 -dump`"
`catdoc`	V:16, I:170	682	`MSWord→text,TeX`	mengonversi berkas MSWord menjadi teks polos atau TeX
`antiword`	V:0.9, I:6.3	596	`MSWord→text,ps`	mengonversi berkas MSWord menjadi teks polos atau ps
`unhtml`	V:0.04, I:0.49	40	`html→text`	menghapus tag markup dari berkas HTML
`odt2txt`	V:1, I:19	60	`odt→text`	konverter dari OpenDocument Text ke teks

Tabel 11.5: Daftar alat untuk mengekstrak data teks polos

11.1.8 Menyoroti dan memformat data teks polos

Anda dapat menyorot dan memformat data teks polos dengan berikut ini.

paket	popcon	ukuran	kata kunci	deskripsi
vim-runtime	V:17, I:355	38849	penyorotan	MAKRO Vim untuk mengonversi kode sumber ke HTML dengan ":source $VIMRUNTIME/syntax/html.vim"
cxref	V:0.01, I:0.21	1191	c → html	pengubah bagi program C ke latex dan HTML (bahasa C)
src2tex	V:0.02, I:0.16	1799	penyorotan	mengonversi banyak kode sumber ke TeX (bahasa C)
source-highlight	V:0.4, I:2.9	2131	penyorotan	mengonversi banyak kode sumber ke berkas HTML, XHTML, LaTeX, Texinfo, escape sequence warna ANSI, dan DocBook dengan penyorotan (C++)
highlight	V:0.4, I:2.8	1412	penyorotan	mengonversi banyak kode sumber ke berkas HTML, XHTML, RTF, LaTeX, TeX, atau XSL-FO dengan penyorotan (C++)
grc	V:0.9, I:5.7	208	text → color	pewarna generik untuk semuanya (Python)
pandoc	V:11, I:47	209559	text → any	pengubah markup umum (Haskell)
python3-docutils	V:13, I:51	2013	text → any	Pemformat dokumen ReStructured Text ke XML (Python)
markdown	V:0.4, I:5.7	56	text → html	Pemformat dokumen teks markdown ke (X)HTML (Perl)
asciidoctor	V:0.5, I:4.6	101	text → any	Pemformat dokumen teks AsciiDoc ke XML/HTML (Python)
python3-sphinx	V:7, I:27	3234	text → any	Sistem publikasi dokumen berbasis ReStructured Text (Python)
hugo	V:0.8, I:5.0	66708	text → html	Sistem publikasi situs statis berbasis markdown (Go)

Tabel 11.6: Daftar alat untuk menyoroti data teks polos

11.2 Data XML

Extensible Markup Language (XML) adalah bahasa markup untuk dokumen yang berisi informasi terstruktur.

Lihat informasi pengantar di XML.COM.

- ”Apa itu XML?”
- ”Apa itu XSLT?”
- ”Apa itu XSL-FO?”
- ”Apa itu XLink?”

11.2.1 Petunjuk dasar untuk XML

Teks XML terlihat agak mirip HTML. Ini memungkinkan kita untuk mengelola beberapa format keluaran untuk dokumen. Salah satu sistem XML yang mudah adalah paket `docbook-xsl`, yang digunakan di sini.

Setiap berkas XML dimulai dengan deklarasi XML standar sebagai berikut.

```
<?xml version="1.0" encoding="UTF-8"?>
```

Sintaks dasar untuk satu elemen XML ditandai sebagai berikut.

```
<name attribute="value">content</name>
```

Elemen XML dengan konten kosong ditandai dalam bentuk pendek berikut.

```
<name attribute="value" />
```

”`attribute="nilai"`” dalam contoh di atas adalah opsional.

Bagian komentar dalam XML ditandai sebagai berikut.

```
<!-- comment -->
```

Selain menambahkan markup, XML memerlukan konversi kecil ke konten menggunakan entitas terpradefinisi untuk karakter-karakter berikut.

entitas terpradefinisi	**tujuan konversi karakter**
`"`	`"` : kutip
`'`	`'` : apostrof
`<`	`<` : kurang dari
`>`	`>:` : lebih dari
`&`	`&` : ampersand

Tabel 11.7: Daftar entitas terpradefinisi untuk XML



Perhatian
”<” atau ”&” tidak dapat digunakan dalam atribut atau elemen.

Catatan
Ketika entitas tentuan pengguna gaya SGML, misalnya "`&tag-anu;`", digunakan, definisi pertama menang atas yang lain. Definisi entitas dinyatakan dalam "`<!ENTITY tag-anu "entity value">`".

Catatan
Selama markup XML dilakukan secara konsisten dengan set tertentu dari nama tag (baik beberapa data sebagai konten atau nilai atribut), konversi ke XML lain adalah tugas sepele menggunakan Extensible Stylesheet Language Transformations (XSLT).

11.2.2 Pemrosesan XML

Ada banyak alat yang tersedia untuk memproses berkas XML seperti Extensible Stylesheet Language (XSL).

Pada dasarnya, setelah Anda membuat berkas XML yang terbentuk dengan baik, Anda dapat mengonversinya ke format apa pun menggunakan Extensible Stylesheet Language Transformations (XSLT).

Extensible Stylesheet Language for Formatting Objects (XSL-FO) seharusnya menjadi solusi untuk pemformatan. Paket `fop` baru di arsip `main` Debian karena ketergantungannya pada bahasa pemrograman Java. Jadi kode LaTeX biasanya dihasilkan dari XML menggunakan XSLT dan sistem LaTeX digunakan untuk membuat berkas yang dapat dicetak seperti DVI, PostScript, dan PDF.

paket	popcon	ukuran	kata kunci	deskripsi
`docbook-xml`	V:15, I:407	2126	xml	Definisi tipe dokumen XML (DTD) untuk DocBook
`docbook-xsl`	V:15, I:144	14823	xml/xslt	Stylesheet XSL untuk memroses DocBook XML ke berbagai format keluaran dengan XSLT
`xsltproc`	V:15, I:71	83	xslt	Prosesor baris perintah XSLT (XML → XML, HTML, teks polos, dll.)
`xmlto`	V:0.5, I:8.0	124	xml/xslt	Konverter XML-ke-apa-pun dengan XSLT
`fop`	V:0.7, I:8.0	281	xml/xsl-fo	mengonversi berkas XML Docbook ke PDF
`dblatex`	V:0.9, I:5.4	4635	xml/xslt	mengonversi berkas Docbook ke dokumen DVI, PostScript, PDF dengan XSLT
`dbtoepub`	V:0.04, I:0.54	37	xml/xslt	konverter DocBook XML ke .epub

Tabel 11.8: Daftar alat XML

Karena XML adalah bagian dari Standard Generalized Markup Language (SGML), itu dapat diproses oleh alat yang banyak tersedia untuk SGML, seperti Document Style Semantics and Specification Language (DSSSL).

paket	popcon	ukuran	kata kunci	deskripsi
`openjade`	V:1, I:21	1066	`dsssl`	ISO /IEC 10179:1996 standar prosesor DSSSL (terbaru)
`docbook-dsssl`	V:0.5, I:7.4	2594	`xml/dsssl`	Stylesheet DSSSL untuk memroses DocBook XML ke berbagai format keluaran dengan DSSSL
`docbook-utils`	V:0.3, I:5.2	287	`xml/dsssl`	utilitas untuk berkas DocBook termasuk konversi ke format lain (HTML, RTF, PS, man, PDF) dengan perintah `docbook2*` dengan DSSSL

Tabel 11.9: Daftar alat DSSSL

Tip
GNOME `yelp` kadang-kadang berguna untuk membaca berkas XML Docbook secara langsung karena itu merender secara layak pada X.

11.2.3 Ekstraksi data XML

Anda dapat mengekstrak data HTML atau XML dari format lain menggunakan berikut.

paket	popcon	ukuran	kata kunci	deskripsi
`man2html`	V:0.1, I:1.2	142	`manpage→html`	konverter dari manpage ke HTML (dukungan CGI)
`doclifter`	V:0.00, I:0.05	487	`troff→xml`	konverter dari troff ke DocBook XML
`texi2html`	V:0.2, I:2.8	1847	`texi→html`	konverter dari Texinfo ke HTML
`info2www`	V:0.8, I:1.4	76	`info→html`	konverter dari info GNU ke HTML (dukungan CGI)
`wv`	V:0.2, I:2.3	733	`MSWord→any`	konverter dokumen dari Microsoft Word ke HTML, LaTeX, dll.
`unrtf`	V:0.3, I:2.8	159	`rtf→html`	konverter dokumen dari RTF ke HTML, dll
`wp2x`	V:0.00, I:0.08	200	`WordPerfect→any`	berkas WordPerfect 5.0 dan 5.1 ke TeX, LaTeX, troff, GML, dan HTML

Tabel 11.10: Daftar alat ekstraksi data XML

11.2.4 Lint data XML

Untuk berkas HTML non-XML, Anda dapat mengonversinya ke XHTML yang merupakan contoh XML yang terbentuk dengan baik. XHTML dapat diproses dengan alat XML.

Sintaks berkas XML dan kebaikan URL yang ditemukan di dalamnya dapat diperiksa.

paket	popcon	ukuran	fungsi	deskripsi
`libxml2-utils`	V:60, I:207	213	`xml↔html↔xhtml`	alat XML baris perintah dengan xmllint(1) (pemeriksaan sintaks, memformat ulang, lint, …)
`tidy`	V:0.9, I:6.9	79	`xml↔html↔xhtml`	Pemeriksa sintaks dan pemformat ulang HTML
`weblint-perl`	V:0.05, I:0.90	32	lint	sintaks dan pemeriksa gaya minimal untuk HTML
`linklint`	V:0.04, I:0.44	343	pemeriksaan tautan	alat pemeriksa tautan cepat dan pemeliharaan situs web

Tabel 11.11: Daftar alat cetak cantik XML

Setelah XML yang tepat dihasilkan, Anda dapat menggunakan teknologi XSLT untuk mengekstrak data berdasarkan konteks mark-up dll.

11.3 Tata cetak

Program troff Unix yang awalnya dikembangkan oleh AT&T dapat digunakan untuk tata cetak sederhana. Ini biasanya digunakan untuk membuat manpage.

TeX yang dibuat oleh Donald Knuth adalah alat tata cetak yang sangat kuat dan merupakan standar de facto. LaTeX awalnya ditulis oleh Leslie Lamport memungkinkan akses tingkat tinggi ke kekuatan TeX.

paket	popcon	ukuran	kata kunci	deskripsi
texlive	V:1, I:27	55	(La)TeX	Sistem TeX untuk tata cetak, pratinjau, dan pencetakan
lilypond	V:0.7, I:5.4	16924	(La)TeX	penata cetak musik
groff	V:2, I:23	16536	troff	Sistem pemformatan teks GNU troff

Tabel 11.12: Daftar alat tata cetak

11.3.1 typesetting roff

Secara tradisional, roff adalah sistem pemrosesan teks Unix utama. Lihat roff(7), groff(7), groff(1), grotty(1), troff(1), groff_mdoc(7), groff_man(7), groff_ms(7), groff_me(7), groff_mm(7), dan "`info groff`".

Anda dapat membaca atau mencetak tutorial dan referensi yang bagus tentang "`-me`" makro di "`/usr/share/doc/groff/`" dengan memasang paket `groff`.

Tip
"`groff -Tascii -me -`" menghasilkan keluaran teks polos dengan kode escape ANSI. Jika Anda ingin mendapatkan keluaran mirip manpage dengan banyak "^H" dan "_", gunakan "`GROFF_NO_SGR=1 groff -Tascii -me -`" sebagai gantinya.

Tip
Untuk menghapus "^H" dan "_" dari berkas teks yang dihasilkan oleh `groff`, filter dengan "`col -b -x`".

11.3.2 TeX/LaTeX

Distribusi perangkat lunak TeX Live menawarkan sistem TeX yang lengkap. Metapackage `texlive` menyediakan pilihan yang layak dari paket TeX Live yang seharusnya cukup untuk tugas yang paling umum.

Ada banyak referensi yang tersedia untuk TeX dan LaTeX.

- The teTeX HOWTO: The Linux-teTeX Local Guide
- tex(1)
- latex(1)
- texdoc(1)
- texdoctk(1)
- "The TeXbook", oleh Donald E. Knuth (Addison-Wesley)
- "LaTeX - A Document Preparation System", oleh Leslie Lamport, (Addison-Wesley)
- "The LaTeX Companion", oleh Goossens, Mittelbach, Samarin, (Addison-Wesley)

Ini adalah lingkungan tata cetak yang paling kuat. Banyak prosesor SGML menggunakan ini sebagai prosesor teks back end mereka. Lyx yang disediakan oleh paket `lyx` dan GNU TeXmacs yang disediakan oleh paket `texmacs` menawarkan lingkungan penyuntingan WYSIWYG yang bagus untuk LaTeX sementara banyak yang menggunakan Emacs dan Vim sebagai pilihan untuk penyunting sumber.

Ada banyak sumber daya daring yang tersedia.

- The TEX Live Guide - TEX Live 2007 ("`/usr/share/doc/texlive-doc-base/english/texlive-en/live.html`") (paket `texlive-doc-base`)

- Panduan Sederhana untuk Lateks/Lyx
- Pengolah Kata Menggunakan LaTeX

Ketika dokumen menjadi lebih besar, terkadang TeX dapat menyebabkan kesalahan. Anda harus meningkatkan ukuran pool di "`/etc/texmf/texmf.cnf`" (atau lebih tepat menyunting "`/etc/texmf/texmf.d/95NonPath`" dan menjalankan update-texmf(8)) untuk memperbaiki ini.

Catatan
Sumber TeX "The TeXbook" tersedia di situs arsip text www.ctan.org bagi texbook.tex. Berkas ini berisi sebagian besar makro yang diperlukan. Saya mendengar bahwa Anda dapat memroses dokumen ini dengan tex(1) setelah mengomentari baris 7 hingga 10 dan menambahkan "`\input manmac \proofmodefalse`". Sangat disarankan untuk membeli buku ini (dan semua buku lain dari Donald E. Knuth) daripada menggunakan versi daring tetapi sumbernya adalah contoh yang bagus dari masukan TeX!

11.3.3 Mencetak cantik halaman manual

Anda dapat mencetak halaman manual dalam PostScript dengan baik dengan salah satu perintah berikut.

```
$ man -Tps some_manpage | lpr
```

11.3.4 Membuat halaman manual

Meskipun menulis halaman manual (manpage) dalam format troff polos adalah mungkin, ada beberapa paket pembantu untuk membuatnya.

paket	popcon	ukuran	kata kunci	deskripsi
`docbook-to-man`	V:0.6, I:5.3	189	SGML → `manpage`	konverter dari DocBook SGML ke makro man roff
`help2man`	V:0.5, I:6.1	542	`text` → `manpage`	generator manpage otomatis dari --help
`info2man`	V:0.02, I:0.18	134	`info` → `manpage`	konverter dari info GNU ke halaman POD atau man
`txt2man`	V:0.04, I:0.61	112	`text` → `manpage`	mengonversi teks ASCII datar ke format halaman man

Tabel 11.13: Daftar paket untuk membantu membuat manpage

11.4 Data yang dapat dicetak

Data yang dapat dicetak dinyatakan dalam format PostScript pada sistem Debian. Common Unix Printing System (CUPS) menggunakan Ghostscript sebagai program backend peraster untuk pencetak non-PostScript.

Data yang dapat dicetak juga dapat dinyatakan dalam format PDF pada sistem Debian terbaru.

Berkas PDF dapat ditampilkan dan entri formulirnya dapat diisi menggunakan alat penampil GUI seperti Evince dan Okular (lihat Bagian 7.4); dan peramban modern seperti Chromium.

Berkas PDF dapat disunting menggunakan beberapa alat bantu grafis seperti LibreOffice, Scribus, dan Inkscape (lihat Bagian 11.6).

Tip
Anda dapat membaca berkas PDF dengan GIMP dan mengonversinya ke dalam format PNG dengan resolusi lebih tinggi dari 300 dpi. Ini dapat digunakan sebagai gambar latar bagi LibreOffice untuk menghasilkan pengubahan cetakan yang diinginkan dengan upaya minimum.

11.4.1 Ghostscript

Inti dari manipulasi data yang dapat dicetak adalah interpreter PostScript (PS) Ghostscript yang menghasilkan gambar raster.

paket	popcon	ukuran	deskripsi
ghostscript	V:165, I:559	177	Interpreter GPL Ghostscript PostScript/PDF
ghostscript-x	V:0, I:13	88	Interpreter GPL Ghostscript PostScript/PDF - Dukungan tampilan X
libpoppler162	V:14, I:22	5313	Pustaka perenderan PDF yang di-fork dari penampil PDF xpdf
libpoppler-glib8t64	V:77, I:323	612	Pustaka perenderan PDF (pustaka bersama berbasis GLib)
poppler-data	V:161, I:582	13086	CMaps untuk pustaka perenderan PDF (untuk dukungan CJK: Adobe-*)

Tabel 11.14: Daftar penerjemah PostScript Ghostscript

Tip
”`gs -h`” dapat menampilkan konfigurasi Ghostscript.

11.4.2 Menggabungkan dua berkas PS atau PDF

Anda dapat menggabungkan dua berkas PostScript (PS) atau Portable Document Format (PDF) menggunakan gs(1) dari Ghostscript.

```
$ gs -q -dNOPAUSE -dBATCH -sDEVICE=pswrite -sOutputFile=bla.ps -f foo1.ps foo2.ps
$ gs -q -dNOPAUSE -dBATCH -sDEVICE=pdfwrite -sOutputFile=bla.pdf -f foo1.pdf foo2.pdf
```

Catatan
PDF, yang merupakan format data cetak lintas platform yang banyak digunakan, pada dasarnya adalah format PS terkompresi dengan beberapa fitur dan ekstensi tambahan.

Tip
Untuk baris perintah, psmerge(1) dan perintah lainnya dari paket `psutils` berguna untuk memanipulasi dokumen PostScript. pdftk(1) dari paket `pdftk` berguna untuk memanipulasi dokumen PDF juga.

11.4.3 Utilitas data yang dapat dicetak

Paket berikut untuk utilitas data yang dapat dicetak menarik perhatian saya.

11.4.4 Mencetak dengan CUPS

Perintah lp(1) dan lpr(1) yang ditawarkan oleh Common Unix Printing System (CUPS) menyediakan opsi untuk pencetakan data yang dapat dicetak secara disesuaikan.

Anda dapat mencetak 3 salinan berkas yang dikolasi menggunakan salah satu perintah berikut.

```
$ lp -n 3 -o Collate=True filename
```

paket	popcon	ukuran	kata kunci	deskripsi
poppler-utils	V:169, I:476	804	pdf → ps, text, ...	Utilitas PDF: pdftops, pdfinfo, pdfimages, pdftotext, pdffonts
psutils	V:4, I:48	34	ps → ps	Alat konversi dokumen PostScript
poster	V:0.1, I:1.6	57	ps → ps	membuat poster besar dari halaman PostScript
enscript	V:1, I:10	2138	text → ps, html, rtf	mengonversi teks ASCII ke PostScript, HTML, RTF, atau Pretty-Print
a2ps	V:0.7, I:6.7	4109	text → ps	Konverter 'Apa pun ke PostScript' dan cetak cantik
pdftk	V:1, I:20	28	pdf → pdf	Alat konversi dokumen PDF: pdftk
html2ps	V:0.2, I:1.7	256	html → ps	konverter dari HTML ke PostScript
gnuhtml2latex	V:0.03, I:0.57	26	html → latex	konverter dari html ke latex
latex2rtf	V:0.1, I:1.9	495	latex → rtf	mengonversi dokumen dari LaTeX ke RTF yang dapat dibaca oleh MS Word
ps2eps	V:2, I:31	95	ps → eps	konverter dari PostScript ke EPS (Encapsulated PostScript)
e2ps	V:0.01, I:0.11	104	text → ps	Konverter Teks ke PostScript dengan dukungan pengodean Jepang
impose+	V:0.1, I:1.6	118	ps → ps	Utilitas PostScript
trueprint	V:0.00, I:0.07	148	text → ps	cetak cantik banyak kode sumber (C, C ++, Java, Pascal, Perl, Pike, Sh, dan Verilog) ke PostScript. (Bahasa C)
pdf2svg	V:0.2, I:2.9	33	pdf → svg	konverter dari PDF ke format Scalable vector graphics
pdftoipe	V:0.01, I:0.43	69	pdf → ipe	konverter dari PDF ke format XML IPE

Tabel 11.15: Daftar utilitas data yang dapat dicetak

```
$ lpr -#3 -o Collate=True filename
```

Anda dapat lebih jauh menyesuaikan operasi pencetak dengan menggunakan opsi pencetak seperti "`-o number-up=2`", "`-o page-set=even`", "`-o page-set=odd`","`-o scaling=200`","`-o natural-scaling=200`",dll., yang didokumentasikan di Command-Line Printing and Options.

11.5 Konversi data surat

Paket berikut untuk konversi data surel menarik perhatian saya.

paket	popcon	ukuran	kata kunci	deskripsi
`sharutils`	V:2, I:27	1436	mail	shar(1), unshar(1), uuencode(1), uudecode(1)
`mpack`	V:0.7, I:7.4	109	`MIME`	enkode dan dekode pesan MIME: mpack(1) dan munpack(1)
`tnef`	V:0.3, I:3.7	103	`ms-tnef`	membongkar lampiran MIME bertipe "application/ms-tnef" yang merupakan format hanya Microsoft
`uudeview`	V:0.2, I:1.7	105	`mail`	enkoder dan dekoder untuk format berikut: uuencode, xxencode, BASE64, quoted printable, dan BinHex

Tabel 11.16: Daftar paket untuk membantu konversi data surel

Tip
Internet Message Access Protocol versi 4 (IMAP4) dapat digunakan untuk memindahkan surel keluar dari sistem surat proprietari jika perangkat lunak klien surel dapat dikonfigurasi untuk menggunakan server IMAP4 juga.

11.5.1 Dasar-dasar data surel

Data surat (SMTP) harus dibatasi pada serangkaian data 7 bit. Jadi data biner dan data teks 8 bit dikodekan ke dalam format 7 bit dengan Multipurpose Internet Mail Extensions (MIME) dan pemilihan charset (lihat Tabel 11.2).

Format penyimpanan surat standar diformat mbox sesuai dengan RFC2822 (memperbarui RFC822). Lihat mbox(5) (disediakan oleh paket `mutt`).

Untuk bahasa Eropa, "`Content-Transfer-Encoding: quoted-printable`" dengan charset ISO-8859-1 biasanya digunakan untuk surat karena tidak ada banyak karakter 8 bit. Jika teks Eropa dikodekan dalam UTF-8, "`Content-Transfer-Encoding: quoted-printable`" kemungkinan akan digunakan karena sebagian besar data 7 bit.

Untuk bahasa Jepang, secara tradisional "`Content-Type: text/plain; charset=ISO-2022-JP`" biasanya digunakan untuk surat untuk menyimpan teks dalam 7 bit. Tetapi sistem Microsoft yang lebih lama dapat mengirim data surel dalam Shift-JIS tanpa deklarasi yang tepat. Jika teks Jepang dikodekan dalam UTF-8, Base64 kemungkinan akan digunakan karena berisi banyak data 8 bit. Situasi bahasa Asia lainnya serupa.

Catatan
Jika data surel non-Unix Anda dapat diakses oleh perangkat lunak klien non-Debian yang dapat berbicara dengan server IMAP4, Anda mungkin dapat memindahkannya dengan menjalankan server IMAP4 Anda sendiri.

Catatan
Jika Anda menggunakan format penyimpanan surel lainnya, memindahkannya ke format mbox adalah langkah pertama yang baik. Program klien serbaguna seperti mutt(1) mungkin berguna untuk ini.

Anda dapat memecah konten kotak surat ke setiap pesan menggunakan procmail(1) dan formail(1).

Setiap pesan surat dapat dibongkar menggunakan munpack(1) dari paket `mpack` (atau alat khusus lainnya) untuk mendapatkan konten yang dikodekan MIME.

11.6 Alat data grafis

Meskipun program GUI seperti gimp(1) sangat kuat, alat baris perintah seperti imagemagick(1) cukup berguna untuk mengotomatisasi manipulasi gambar melalui skrip.

Format berkas gambar de facto dari kamera digital adalah Exchangeable Image File Format (EXIF) yang merupakan format berkas gambar JPEG dengan tag metadata tambahan. Ini dapat menyimpan informasi seperti pengaturan tanggal, waktu, dan kamera.

Paten kompresi data lossless Lempel-Ziv-Welch (LZW) telah kedaluwarsa. Utilitas Graphics Interchange Format (GIF) yang menggunakan metode kompresi LZW sekarang tersedia secara bebas pada sistem Debian.

Tip
Setiap kamera digital atau pemindai dengan media rekam lepasan bekerja dengan Linux melalui pembaca penyimpanan USB karena mengikuti Aturan Desain untuk Sistem Berkas Kamera dan menggunakan sistem berkas FAT. Lihat Bagian 10.1.7.

11.6.1 Alat data grafis (paket-meta)

Paket meta berikut ini adalah titik awal yang baik untuk mencari alat data grafis menggunakan aptitude(8). "Ikhtisar paket untuk Pemelihara Alat Foto Debian" dapat menjadi titik awal lainnya.

paket	popcon	ukuran	kata kunci	deskripsi
`education-graphics`	I:0.34	26	`svg,jpeg,…`	paket meta untuk mengajar seni grafis dan seni gambar.
`open-font-design-toolkit`	I:0.04	9	`ttf,ps,…`	metapackage untuk desain fonta terbuka

Tabel 11.17: Daftar alat data grafis (paket-meta)

Tip
Cari lebih banyak alat gambar menggunakan regex "`~Gworks-with::image`" dalam aptitude(8) (lihat Bagian 2.2.6).

11.6.2 Alat data grafis (GUI)

Paket berikut untuk konversi data grafis, penyuntingan, dan alat organisasi GUI menarik perhatian saya.

paket	popcon	ukuran	kata kunci	deskripsi
gimp	V:34, I:211	33315	image(bitmap)	GNU Image Manipulation Program
xsane	V:9, I:128	1512	image(bitmap)	Frontend X11 berbasis GTK untuk SANE (Scanner Access Now Easy)
scribus	V:1, I:12	32942	ps/pdf/SVG/...	Penyunting DTP Scribus
libreoffice-draw	V:89, I:418	11418	image(vector)	keluarga perkantoran LibreOffice - menggambar
inkscape	V:12, I:76	113353	image(vector)	Penyunting SVG (Scalable Vector Graphics)
dia	V:1, I:16	3846	image(vector)	penyunting diagram (Gtk)
xfig	V:0.5, I:8.5	7951	image(vector)	Fasilitas untuk Pembangkitan Interaktif figur di bawah X11
gocr	V:0.5, I:3.6	549	image→text	perangkat lunak OCR bebas
eog	V:23, I:129	10535	image(Exif)	Program penampil grafis Eye of GNOME
gthumb	V:3, I:11	5162	image(Exif)	penampil dan peramban citra (GNOME)
geeqie	V:3, I:11	3367	image(Exif)	penampil citra menggunakan GTK
shotwell	V:14, I:246	6334	image(Exif)	pengorganisasi foto digital (GNOME)
gwenview	V:41, I:114	6057	image(Exif)	penampil citra (KDE)
kamera	I:114	992	image(Exif)	dukungan kamera digital untuk aplikasi KDE
digikam	V:1.5, I:8.3	333	image(Exif)	aplikasi manajemen foto digital untuk KDE
darktable	V:4, I:11	36092	image(Exif)	kamar gelap dan meja cahaya virtual untuk para fotografer
hugin	V:0.5, I:5.3	6524	image(Exif)	penjahit foto panorama
librecad	V:1, I:13	9164	DXF,...	Penyunting data CAD 2D
freecad	V:1, I:20	112	DXF,...	Penyunting data CAD 3D
blender	V:2, I:20	171893	blend,TIFF,VRML,...	Penyunting konten 3D untuk animasi dll
mm3d	V:0.02, I:0.22	4123	ms3d,obj,dxf,...	Penyunting model 3D berbasis OpenGL
fontforge	V:0.6, I:5.3	4736	ttf,ps,...	penyunting fonta untuk fonta PS, TrueType, dan OpenType
xgridfit	V:0.02, I:0.08	878	ttf	program untuk gridfitting dan hinting fonta TrueType

Tabel 11.18: Daftar alat data grafis (GUI)

paket	popcon	ukuran	kata kunci	deskripsi
imagemagick	V:7, I:272	80	image(bitmap)	program manipulasi gambar
graphicsmagick	V:1.0, I:8.3	5945	image(bitmap)	program manipulasi citra (fork dari imagemagick)
netpbm	V:27, I:282	8435	image(bitmap)	alat konversi grafis
libheif-examples	V:0.3, I:3.6	528	heif→jpeg(bitmap)	mengonversi High Efficiency Image File Format (HEIF) ke format JPEG, PNG, atau Y4M dengan perintah heif-convert(1)
icoutils	V:3, I:33	221	png↔ico(bitmap)	mengonversi ikon dan kursor MS Windows ke dan dari format PNG (favicon.ico)
pstoedit	V:2, I:37	1075	ps/pdf→image(vector)	Konverter berkas PostScript dan PDF ke grafis vektor yang dapat disunting (SVG)
libwmf-bin	V:5, I:80	149	Windows/gambar(vector)	Alat konversi metafile Windows (data grafik vektor)
fig2sxd	V:0.01, I:0.16	158	fig→sxd(vector)	mengonversi berkas XFig ke format OpenOffice.org Draw
unpaper	V:2, I:16	417	image→image	alat pasca-pemrosesan bagi halaman yang dipindai untuk OCR
tesseract-ocr	V:8, I:34	2209	image→text	perangkat lunak OCR bebas berdasarkan mesin OCR komersial HP
tesseract-ocr-eng	V:9, I:35	4032	image→text	Data mesin OCR: berkas bahasa tesseract-ocr untuk teks bahasa Inggris
ocrad	V:0.3, I:2.2	608	image→text	perangkat lunak OCR bebas
exif	V:3, I:49	335	image(Exif)	utilitas baris perintah untuk menampilkan informasi EXIF dalam berkas JPEG
exiv2	V:1, I:18	436	image(Exif)	alat manipulasi metadata EXIF/IPTC
exiftran	V:1, I:11	81	image(Exif)	mentransformasi citra jpeg kamera digital
exiftags	V:0.2, I:2.5	309	image(Exif)	utilitas untuk membaca tag Exif dari berkas JPEG kamera digital
exifprobe	V:0.2, I:2.1	506	image(Exif)	membaca metadata dari gambar digital
dcraw	V:0.7, I:7.2	428	image(Raw)→ppm	dekode gambar kamera digital mentah
findimagedupes	V:0.11, I:0.96	76	image→fingerprint	menemukan gambar yang serupa secara visual atau duplikat
ale	V:0.00, I:0.14	850	image→image	menggabung gambar untuk meningkatkan mutu atau membuat mosaik
imageindex	V:0.1, I:1.1	143	image(Exif)→html	menghasilkan galeri HTML statis dari gambar
outguess	V:0.1, I:1.0	230	jpeg,png	alat Steganografi universal
jpegoptim	V:0.6, I:5.8	59	jpeg	mengoptimalkan berkas JPEG
optipng	V:2, I:39	187	png	mengoptimalkan berkas PNG, kompresi lossless
pngquant	V:1, I:11	62	png	mengoptimalkan berkas PNG, kompresi lossy

Tabel 11.19: Daftar alat data grafis (CLI)

11.6.3 Alat data grafis (CLI)

Paket berikut untuk konversi data grafis, penyuntingan, dan alat organisasi CLI menarik perhatian saya.

11.7 Konversi data lain-lain

Ada banyak program lain untuk mengonversi data. Paket-paket berikut menarik perhatian saya menggunakan regex "`~Guse::converting`" dalam aptitude(8) (lihat Bagian 2.2.6).

paket	popcon	ukuran	kata kunci	deskripsi
`alien`	V:1, I:12	150	`rpm/tgz→deb`	konverter untuk paket asing ke dalam paket Debian
`freepwing`	V:0.00, I:0.01	447	`EB→EPWING`	konverter dari "Electric Book" (populer di Jepang) ke format JIS X 4081 tunggal (subset dari EPWING V1)
`calibre`	V:7, I:24	71216	`any→EPUB`	konverter e-book dan manajemen perpustakaan

Tabel 11.20: Daftar alat konversi data lain-lain

Anda juga dapat mengekstrak data dari format RPM dengan yang berikut.

```
$ rpm2cpio file.src.rpm | cpio --extract
```

Bab 12

Pemrograman

Saya memberikan beberapa petunjuk bagi orang untuk belajar pemrograman pada sistem Debian yang cukup untuk melacak kode sumber yang dikemas. Berikut adalah paket penting dan paket dokumentasi yang sesuai untuk pemrograman.

Referensi daring tersedia dengan mengetik ”`man nama`” setelah memasang paket `manpages` dan `manpages-dev`. Referensi daring untuk alat GNU tersedia dengan mengetik ”`info nama_program`” setelah memasang paket dokumentasi yang bersangkutan. Anda mungkin perlu memasukkan arsip `contrib` dan `non-free` selain arsip `main` karena beberapa dokumentasi GFDL tidak dianggap sesuai dengan DFSG.

Harap pertimbangkan untuk menggunakan alat sistem kontrol versi. Lihat Bagian 10.5.



Awas
Jangan gunakan ”`test`” sebagai nama berkas uji yang dapat dieksekusi. ”`test`” adalah shell builtin.



Perhatian
Anda harus memasang program perangkat lunak yang langsung dikompilasi dari sumber ke ”`/usr/local`” atau ”`/opt`” untuk menghindari tabrakan dengan program sistem.

Tip
Contoh kode untuk membuat ”Song 99 Bottles of Beer” mestinya memberi Anda ide bagus tentang hampir semua bahasa pemrograman.

12.1 Skrip shell

Skrip shell adalah berkas teks dengan bit eksekusi yang dihidupkan dan berisi perintah dalam format berikut.

```
#!/bin/sh
 ... command lines
```

Baris pertama menentukan interpreter shell yang membaca dan mengeksekusi isi berkas ini.

Membaca skrip shell adalah cara **terbaik** untuk memahami cara kerja sistem seperti Unix. Di sini, saya memberikan beberapa petunjuk dan pengingat untuk pemrograman shell. Lihat ”Kesalahan Shell” (https://www.greenend.org.uk/rjk/2001/04/shell.html) untuk belajar dari kesalahan.

Tidak seperti mode interaktif shell (lihat Bagian 1.5 dan Bagian 1.6), skrip shell sering menggunakan parameter, kondisional, dan loop.

12.1.1 Kompatibilitas shell POSIX

Banyak skrip sistem dapat diinterpretasi oleh salah satu shell POSIX (lihat Tabel 1.13).

- Shell POSIX non-interaktif baku "`/usr/bin/sh`" adalah symlink yang menunjuk ke `/usr/bin/dash` dan digunakan oleh banyak program sistem.
- Shell POSIX interaktif baku adalah `/usr/bin/bash`.

Hindari menulis skrip shell dengan **bashisme** atau **zshisme** agar membuatnya portabel di antara semua shell POSIX. Anda dapat memeriksanya menggunakan checkbashisms(1).

Baik: POSIX	**Hindari: bashisme**
`if [ "$foo" = "$bar" ] ; then …`	`if [ "$foo" == "$bar" ] ; then …`
`diff -u file.c.orig file.c`	`diff -u file.c{.orig,}`
`mkdir /foobar /foobaz`	`mkdir /foo{bar,baz}`
`funcname() { … }`	`function funcname() { … }`
format oktal: "`\377`"	format heksadesimal: "`\xff`"

Tabel 12.1: Daftar bashisme khas

Perintah "`echo`" harus digunakan dengan kehati-hatian berikut karena implementasinya berbeda di antara bawaan shell dan perintah eksternal.

- Hindari menggunakan opsi perintah apa pun kecuali "`-n`".
- Hindari menggunakan urutan escape dalam string karena penanganannya bervariasi.

Catatan
Meskipun opsi "`-n`" **bukan** benar-benar sintaks POSIX, itu umumnya diterima.

Tip
Gunakan perintah "`printf`" alih-alih perintah "`echo`" jika Anda perlu menanamkan urutan escape dalam string keluaran.

12.1.2 Parameter shell

Parameter shell khusus sering digunakan dalam skrip shell.

parameter shell	**nilai**
`$0`	nama shell atau skrip shell
`$1`	argumen shell pertama
`$9`	argumen shell kesembilan
`$#`	cacah parameter posisional
`"$*"`	`"$1 $2 $3 $4 … "`
`"$@"`	`"$1" "$2" "$3" "$4" …`
`$?`	status keluar dari perintah terkini
`$$`	PID dari skrip shell ini
`$!`	PID dari pekerjaan latar belakang yang baru-baru ini dimulai

Tabel 12.2: Daftar parameter shell

Ekspansi parameter dasar yang perlu diingat adalah sebagai berikut.

Di sini, titik dua "`:`" di semua operator ini sebenarnya opsional.

bentuk ekspresi parameter	nilai jika `var` diatur	nilai jika `var` tidak diatur
`${var:-string}`	"`$var`"	"`string`"
`${var:+string}`	"`string`"	"`null`"
`${var:=string}`	"`$var`"	"`string`" (dan jalankan "`var=string`")
`${var:?string}`	"`$var`"	echo "`string`" ke **stderr** (dan keluar dengan kesalahan)

Tabel 12.3: Daftar ekspansi parameter shell

- **dengan** "`:`" operator = menguji **keberadaan** dan **bukan null**
- **tanpa** "`:`" operator = hanya menguji **keberadaan**

bentuk substitusi parameter	hasil
`${var%suffix}`	menghapus pola akhiran terkecil
`${var%%suffix}`	menghapus pola akhiran terbesar
`${var#prefix}`	menghapus pola awalan terkecil
`${var##prefix}`	menghapus pola awalan terbesar

Tabel 12.4: Daftar substitusi parameter shell utama

12.1.3 Kondisional Shell

Setiap perintah mengembalikan **status keluar** yang dapat digunakan untuk ekspresi bersyarat.

- Sukses: 0 ("Benar/True")
- Galat: bukan 0 ("Salah/False")

Catatan
"0" dalam konteks kondisional shell berarti "Benar", sedangkan "0" dalam konteks kondisional C berarti "Salah".

Catatan
"`[`" adalah setara dengan perintah `test`, yang mengevaluasi argumennya sampai "`]`" sebagai ekspresi bersyarat.

Idiom kondisionak dasar untuk diingat adalah sebagai berikut.

- "*`perintah`* `&&` *`bila_sukses_jalankan_perintah_ini_juga`* `||` `true`"
- "*`perintah`* `||` *`bila_tidak_sukses_jalankan_perintah_ini_juga`* `||` `true`"
- Cuplikan skrip multi-baris sebagai berikut

```
if [ conditional_expression ]; then
 if_success_run_this_command
else
 if_not_success_run_this_command
fi
```

Di sini "`|| true`" di akhir diperlukan untuk memastikan skrip shell ini tidak keluar pada baris ini secara tidak sengaja ketika shell dipanggil dengan flag"`-e`".

Operator perbandingan integer **aritmatika** dalam ekspresi bersyarat adalah "`-eq`", "`-ne`", "`-lt`", "`-le`", "`-gt`", dan "`-ge`".

persamaan	kondisi yang mengembalikan logis benar (true)
`-e` *`file`*	*berkas* ada
`-d` *`file`*	*berkas* ada dan berupa direktori
`-f` *`berkas`*	*berkas* ada dan berupa berkas biasa
`-w` *`file`*	*berkas* ada dan dapat ditulis
`-x` *`file`*	*berkas* ada dan dapat dieksekusi
`file1` `-nt` *`file2`*	*berkas1* lebih baru dari *berkas2* (modifikasi)
`file1` `-ot` *`file2`*	*berkas1* lebih tua dari *berkas2* (modifikasi)
`file1` `-ef` *`file2`*	*berkas1* dan *berkas2* berada di perangkat yang sama dan nomor inode yang sama

Tabel 12.5: Daftar operator perbandingan berkas dalam ekspresi bersyarat

persamaan	kondisi yang mengembalikan logis benar (true)
`-z` *`str`*	panjang *str* adalah nol
`-n` *`str`*	panjang *str* adalah bukan nol
`str1` `=` *`str2`*	*str1* dan *str2* adalah sama
`str1` `!=` *`str2`*	*str1* dan *str2* tidak sama
`str1` `<` *`str2`*	*str1* diurut sebelum *str2* (tergantung lokal)
`str1` `>` *`str2`*	*str1* diurut setelah *str2* (tergantung lokal)

Tabel 12.6: Daftar operator perbandingan string dalam ekspresi bersyarat

12.1.4 Loop shell

Ada beberapa idiom pengulangan untuk digunakan dalam shell POSIX.

- "`for x in foo1 foo2 … ; do perintah ; done`" mengulang dengan menetapkan butir dari daftar "`foo1 foo2 …`" ke variabel "x" dan mengeksekusi "`perintah`".
- "`while condition ; do perintah ; done`" mengulangi "`perintah`" saat "`kondisi`" adalah benar.
- "`until condition ; do perintah ; done`" mengulangi "`perintah`" saat "`kondisi`" tidak benar.
- "`break`" memungkinkan untuk keluar dari pengulangan.
- "`continue`" memungkinkan untuk melanjutkan iterasi pengulangan berikutnya.

Tip
Iterasi numerik bahasa mirip C dapat direalisasikan dengan menggunakan seq(1) sebagai generator "`foo1 foo2 ...`".

Tip
Lihat Bagian 9.4.9.

12.1.5 Variabel lingkungan shell

Beberapa variabel lingkungan populer untuk command prompt shell normal mungkin tidak tersedia di bawah lingkungan eksekusi skrip Anda.

- Untuk "`$USER`", gunakan "`$(id -un)`"
- Untuk "`$UID`", gunakan "`$(id -u)`"
- Untuk "`$HOME`", gunakan "`$(getent passwd "$(id -u)"|cut -d ":" -f 6)`" (ini juga berfungsi pada Bagian 4.5.2)

12.1.6 Urutan pemrosesan baris perintah shell

Shell memproses skrip kira-kira sebagai urutan berikut.

- Shell membaca satu baris.
- Shell mengelompokkan bagian dari baris sebagai **satu token** jika berada di dalam "…" atau '…'.
- Shell memecah bagian lain dari suatu baris menjadi **token** dengan yang berikut.
  - Ruang kosong: *spasi tab ganti baris*
  - Karakter meta: < > | ; & ( )
- Shell memeriksa **kata tetapan** untuk setiap token untuk menyesuaikan perilakunya jika tidak dalam "…" atau '…'.
  - **kata tetapan**: if then elif else fi for in while unless do done case esac
- Shell mengekspansikan **alias** jika tidak dalam "…" atau '…'.
- Shell mengekspansikan **tilde** jika tidak dalam "…" atau '…'.
  - "~" → direktori rumah pengguna saat ini
  - "~~*pengguna*" → direktori rumah *pengguna*
- Shell mengekspansikan **parameter** ke nilainya jika tidak dalam '…'.
  - **parameter**: "$PARAMETER" atau "${PARAMETER} "
- Shell mengekspansikan **substitusi perintah** jika tidak dalam '…'.
  - "$(command)" → keluaran dari "perintah"
  - "` perintah `" → keluaran dari "perintah"
- Shell mengekspansikan **pathname glob** untuk mencocokkan nama berkas jika tidak dalam "…"" atau '…'.
  - * → sebarang karakter
  - ? → satu karakter
  - [...] → salah satu karakter dalam "..."
- Shell mencari **perintah** dari yang berikut dan mengeksekusinya.
  - definisi **fungsi**
  - perintah **builtin**
  - **berkas yang dapat dieksekusi** dalam "$PATH"
- Shell pergi ke baris berikutnya dan mengulangi proses ini lagi dari puncak urutan ini.

Kutip tunggal dalam kutip ganda tidak berpengaruh.

Mengeksekusi "set -x" di shell atau menjalankan shell dengan opsi "-x" membuat shell mencetak semua perintah yang dieksekusi. Ini sangat berguna untuk debugging.

paket	popcon	ukuran	deskripsi
dash	V:929, I:999	207	shell kecil dan cepat patuh POSIX untuk sh
coreutils	V:912, I:1000	17994	Utilitas inti GNU
grep	V:782, I:1000	1297	GNU grep, egrep, dan fgrep
sed	V:769, I:1000	987	GNU sed
mawk	V:495, I:998	299	awk yang kecil dan cepat
debianutils	V:935, I:996	228	utilitas lain-lain khusus untuk Debian
bsdutils	V:421, I:999	338	utilitas dasar dari 4.4BSD-Lite
bsdextrautils	V:761, I:870	382	utilitas tambahan dari 4.4BSD-Lite
moreutils	V:18, I:39	232	utilitas Unix tambahan

Tabel 12.7: Daftar paket yang berisi program utilitas kecil untuk skrip shell

12.1.7 Program utilitas untuk skrip shell

Untuk membuat program shell Anda seportabel mungkin antar sistem Debian, adalah ide yang baik untuk membatasi program utilitas hanya ke yang disediakan oleh paket-paket **esensial**.

- "`aptitude search ~E`" menampilkan daftar paket-paket **penting**.
- "`dpkg -L` *`nama_paket`* `|grep '/man/man.*/'`" mencantumkan daftar halaman man untuk perintah yang ditawarkan oleh paket *`nama_paket`*.

Tip
Meskipun `moreutils` mungkin tidak ada di luar Debian, ia menawarkan program kecil yang menarik. Yang paling menonjol adalah sponge(8) yang sangat berguna ketika Anda ingin menimpa berkas asli.

Lihat Bagian 1.6 misalnya.

12.2 Scripting dalam bahasa yang diinterpretasi

Ketika Anda ingin mengotomatiskan tugas di Debian, Anda harus menulisnya dengan bahasa yang diintepretasi terlebih dahulu. Panduan untuk pilihan bahasa yang diinterpretasi adalah:

- Gunakan `dash`, jika tugas sederhana yang menggabungkan program CLI dengan program shell.
- Gunakan `python3`, jika tugas tidak sederhana dan Anda menulisnya dari awal.
- Gunakan `perl`, `tcl`, `ruby`, ... jika sudah ada kode yang menggunakan salah satu bahasa ini di Debian yang perlu disentuh untuk melakukan tugas tersebut.

Jika kode yang dihasilkan terlalu lambat, Anda dapat menulis ulang hanya bagian penting untuk kecepatan eksekusi dalam bahasa yang dikompilasi dan memanggilnya dari bahasa yang ditafsirkan.

paket	popcon	ukuran	dokumentasi
dash	V:929, I:999	207	sh: shell kecil dan cepat patuh POSIX untuk sh
bash	V:886, I:999	7308	sh: "info bash" disediakan oleh bash-doc
mawk	V:495, I:998	299	AWK: awk yang kecil dan cepat
gawk	V:247, I:294	3289	AWK: "info gawk" disediakan oleh gawk-doc
perl	V:663, I:990	840	Perl: perl(1) dan halaman html disediakan oleh perl-doc dan perl-doc-html
libterm-readline-gnu-perl	V:2, I:27	439	Ekstensi Perl untuk GNU ReadLine/History Library: perlsh(1)
libreply-perl	V:0.01, I:0.07	171	REPL untuk Perl: reply(1)
libdevel-repl-perl	V:0.02, I:0.50	237	REPL untuk Perl: re.pl(1)
python3	V:720, I:975	81	Python: python3(1) dan halaman html yang disediakan oleh python3-doc
tcl	V:24, I:173	20	Tcl: tcl(3) dan detail halaman manual yang disediakan oleh tcl-doc
tk	V:18, I:167	20	Tk: tk(3) dan detail halaman manual yang disediakan oleh tk-doc
ruby	V:72, I:154	32	Ruby: ruby(1), erb(1), irb(1), rdoc(1), ri(1)

Tabel 12.8: Daftar paket terkait interpreter

12.2.1 Debugging kode bahasa yang diinterpretasi

Sebagian besar interpreter menawarkan pemeriksaan sintaks dasar dan fungsi pelacakan kode.

- "**dash -n** *script.sh*" - Pemeriksaan sintaks dari suatu skrip Shell
- "**dash -x** *script.sh*" - Melacak suatu skrip Shell
- "**python -m py_compile** *script.py*" - Pemeriksaan sintaks skrip Python
- "**python -mtrace --trace** *script.py*" - Melacak suatu skrip Python
- "**perl -I .. /libpath -c** *script.pl*" - Pemeriksaan sintaks dari suatu skrip Perl
- "**perl -d:Trace** *script.pl*" - Melacak suatu skrip Perl

Untuk menguji kode bagi `dash`, cobalah Bagian 9.1.4 yang mengakomodasi lingkungan interaktif mirip `bash`.

Untuk menguji kode bagi `perl`, cobalah lingkungan REPL untuk Perl yang mengakomodasi lingkungan mirip Python REPL (= READ + EVAL + PRINT + LOOP) untuk Perl.

12.2.2 Program GUI dengan skrip shell

Skrip shell dapat diperbaiki untuk membuat program GUI yang menarik. Caranya adalah dengan menggunakan salah satu program dialog yang disebut alih-alih interaksi membosankan menggunakan perintah `echo` dan `read`.

Berikut adalah contoh program GUI untuk menunjukkan betapa mudahnya itu hanya dengan suatu skrip shell.

Skrip ini menggunakan `zenity` untuk memilih berkas (baku `/etc/motd`) dan menampilkannya.

Peluncur GUI untuk skrip ini dapat dibuat berikut Bagian 9.4.10.

```
#!/bin/sh -e
# Copyright (C) 2021 Osamu Aoki <osamu@debian.org>, Public Domain
# vim:set sw=2 sts=2 et:
```

paket	popcon	ukuran	deskripsi
x11-utils	V:228, I:555	651	xmessage(1): menampilkan pesan atau kueri di jendela (X)
whiptail	V:307, I:997	61	menampilkan kotak dialog yang mudah digunakan dari skrip shell (newt)
dialog	V:8, I:77	520	menampilkan kotak dialog yang mudah digunakan dari skrip shell (ncurses)
zenity	V:68, I:341	193	menampilkan kotak dialog grafis dari skrip shell (GTK)
ssft	V:0.01, I:0.16	80	Shell Scripts Frontend Tool (pembungkus untuk zenity, kdialog, dan dialog dengan gettext)
gettext	V:49, I:210	20549	"/usr/bin/gettext.sh": terjemahkan pesan

Tabel 12.9: Daftar program dialog

```
DATA_FILE=$(zenity --file-selection --filename="/etc/motd" --title="Select a file to check ↩
    ") || \
  ( echo "E: File selection error" >&2 ; exit 1 )
# Check size of archive
if ( file -ib "$DATA_FILE" | grep -qe '^text/' ) ; then
  zenity --info --title="Check file: $DATA_FILE" --width 640  --height 400 \
    --text="$(head -n 20 "$DATA_FILE")"
else
  zenity --info --title="Check file: $DATA_FILE" --width 640  --height 400 \
    --text="The data is MIME=$(file -ib "$DATA_FILE")"
fi
```

Pendekatan semacam ini untuk program GUI dengan skrip shell hanya berguna untuk kasus pilihan sederhana. Jika Anda ingin menulis program apa pun dengan kompleksitas, silakan pertimbangkan untuk menulisnya di platform yang lebih mampu.

12.2.3 Tindakan ubahan untuk filer GUI

Program filer GUI dapat diperluas untuk melakukan beberapa tindakan populer pada berkas yang dipilih menggunakan paket ekstensi tambahan. Mereka juga dapat dibuat untuk melakukan tindakan ubahan yang sangat spesifik dengan menambahkan skrip spesifik Anda.

- Untuk GNOME, lihat NautilusScriptsHowto.
- Untuk KDE, lihat Membuat Menu Layanan Dolphin.
- Untuk Xfce, lihat Thunar - Tindakan Ubahan dan https://help.ubuntu.com/community/ThunarCustomActions.
- Untuk LXDE, lihat Tindakan Ubahan.

12.2.4 Kegilaan skrip pendek Perl

Untuk memproses data, `sh` perlu men-spawn sub-proses yang menjalankan `cut`, `grep`, `sed`, dll, dan lambat. Di sisi lain, `perl` memiliki kemampuan internal untuk memroses data, dan cepat. Begitu banyak skrip pemeliharaan sistem pada Debian menggunakan `perl`.

Mari kita pikirkan untuk mengikuti cuplikan skrip AWK satu-baris dan yang setara di Perl.

```
awk '($2=="1957") { print $3 }' |
```

Ini setara dengan salah satu dari baris berikut.

```
perl -ne '@f=split; if ($f[1] eq "1957") { print "$f[2]\n"}' |
```

```
perl -ne 'if ((@f=split)[1] eq "1957") { print "$f[2]\n"}' |
```

```
perl -ne '@f=split; print $f[2] if ( $f[1]==1957 )' |
```

```
perl -lane 'print $F[2] if $F[1] eq "1957"' |
```

```
perl -lane 'print$F[2]if$F[1]eq+1957' |
```

Yang terakhir adalah teka-teki. Itu mengambil keuntungan dari mengikuti fitur Perl.

- Ruang kosong (whitespace) adalah opsional.
- Konversi otomatis ada dari angka ke string.
- Trik eksekusi perl melalui opsi baris perintah: perlrun(1)
- Variabel khusus Perl: perlvar(1)

Fleksibilitas ini adalah kekuatan Perl. Pada saat yang sama, ini memungkinkan kita untuk membuat kode sulit dibaca dan kusut. Jadi berhati-hatilah.

12.3 Menuis kode dalam bahasa yang dikompilasi

paket	popcon	ukuran	deskripsi
gcc	V:156, I:561	36	Kompiler GNU C
libc6-dev	V:258, I:580	12863	GNU C Library: Pustaka Pengembangan dan Berkas Header
g++	V:61, I:529	13	Kompiler GNU C++
libstdc++-16-dev	V:3, I:19	24398	GNU Standard C++ Library v3 (berkas pengembangan)
cpp	V:337, I:720	18	Preprocessor C GNU
gettext	V:49, I:210	20549	Utilitas internasionalisasi GNU
glade	V:0.2, I:2.6	1613	Pembangun Antarmuka Pengguna GTK
valac	V:0.2, I:3.0	533	Bahasa mirip C# untuk sistem GObject
flex	V:6, I:67	1247	Kompatibel LEX fast lexical analyzer generator
bison	V:7, I:71	3122	Kompatibel YACC parser generator
susv2	I:0.03	16	ambil ”The Single UNIX Specifications v2”
susv3	I:0.05	16	ambil ”The Single UNIX Specifications v3”
susv4	I:0.02	16	ambil ”The Single UNIX Specifications v4”
golang	I:21	11	Kompiler bahasa pemrograman Go
rustc	V:5, I:20	13195	Bahasa pemrograman sistem Rust
gfortran	V:4, I:49	15	Kompiler GNU Fortran 95
fpc	I:2.2	101	Free Pascal

Tabel 12.10: Daftar paket terkait kompiler

Di sini, Bagian 12.3.3 dan Bagian 12.3.4 disertakan untuk menunjukkan bagaimana program seperti kompiler dapat ditulis dalam bahasa C dengan menyusun deskripsi tingkat yang lebih tinggi ke dalam bahasa C.

12.3.1 C

Anda dapat mengatur lingkungan yang tepat untuk mengkompilasi program yang ditulis dalam bahasa pemrograman C dengan yang berikut.

```
# apt-get install glibc-doc manpages-dev libc6-dev gcc build-essential
```

Paket `libc6-dev`, yaitu Pustaka GNU C, menyediakan pustaka standar C yang merupakan kumpulan berkas header dan pustaka fungsi yang digunakan oleh bahasa pemrograman C.

Lihat referensi untuk C dengan cara berikut.

- "`info libc`" (referensi fungsi pustaka C)
- gcc(1) dan "`info gcc`"
- each_C_library_function_name(3)
- Kernighan & Ritchie, "The C Programming Language", edisi 2 (Prentice Hall)

12.3.2 Program C Sederhana (gcc)

Contoh sederhana "`example.c`" dapat dikompilasi dengan pustaka "`libm`" menjadi "`run_example`" yang dapat dieksekusi dengan berikut.

```
$ cat > example.c << EOF
#include <stdio.h>
#include <math.h>
#include <string.h>

int main(int argc, char **argv, char **envp){
        double x;
        char y[11];
        x=sqrt(argc+7.5);
        strncpy(y, argv[0], 10); /* prevent buffer overflow */
        y[10] = '\0'; /* fill to make sure string ends with '\0' */
        printf("%5i, %5.3f, %10s, %10s\n", argc, x, y, argv[1]);
        return 0;
}
EOF
$ gcc -Wall -g -o run_example example.c -lm
$ ./run_example
        1, 2.915, ./run_exam,     (null)
$ ./run_example 1234567890qwerty
        2, 3.082, ./run_exam, 1234567890qwerty
```

Di sini, "`-lm`" diperlukan untuk me-link pustaka "`/usr/lib/libm.so`" dari paket `libc6` untuk sqrt(3). Pustaka yang sebenarnya adalah dalam "`/lib/`" dengan nama berkas "`libm.so.6`", yang merupakan symlink untuk "`libm-2.7.so`".

Lihatlah parameter terakhir dalam teks keluaran. Ada lebih dari 10 karakter meskipun "`%10s`" ditentukan.

Penggunaan fungsi operasi memori pointer tanpa pemeriksaan batas, seperti sprintf(3) dan strcpy(3), tidak berlaku lagi untuk mencegah eksploitasi buffer overflow yang memanfaatkan efek overrun di atas. Sebagai gantinya, gunakan snprintf(3) dan strncpy(3).

12.3.3 Flex - Lex yang lebih baik

Flex adalah generator analisis leksikal cepat yang kompatibel dengan Lex.

Tutorial untuk flex(1) dapat ditemukan di "`info flex`".

Banyak contoh sederhana dapat ditemukan di bawah "`/usr/share/doc/flex/examples/`". 1

1Beberapa penyesuaian mungkin diperlukan untuk membuatnya bekerja di bawah sistem saat ini.

12.3.4 Bison - Yacc yang lebih baik

Beberapa paket menyediakan parser LR lookahead yang kompatibel dengan Yacc atau generator parser LALR di Debian.

paket	popcon	ukuran	deskripsi
`bison`	V:7, I:71	3122	Generator parser GNU LALR
`byacc`	V:0.1, I:2.9	263	Generator parser Berkeley LALR
`btyacc`	V:0.01, I:0.04	247	backtracking parser generator berdasarkan `byacc`

Tabel 12.11: Daftar generator parser LALR yang kompatibel dengan Yacc

Tutorial untuk bison(1) dapat ditemukan di"`info bison`".

Anda perlu menyediakan "`main()`" dan "`yyerror()`" Anda sendiri. "`main()`" memanggil "`yyparse()`" yang memanggil "`yylex()`", biasanya dibuat dengan Flex.

Berikut ini adalah contoh untuk membuat program kalkulator terminal sederhana.

Mari kita buat `example.y`:

```
/* calculator source for bison */
%{
#include <stdio.h>
extern int yylex(void);
extern int yyerror(char *);
%}

/* declare tokens */
%token NUMBER
%token OP_ADD OP_SUB OP_MUL OP_RGT OP_LFT OP_EQU

%%
calc:
 | calc exp OP_EQU    { printf("Y: RESULT = %d\n", $2); }
 ;

exp: factor
 | exp OP_ADD factor  { $$ = $1 + $3; }
 | exp OP_SUB factor  { $$ = $1 - $3; }
 ;

factor: term
 | factor OP_MUL term { $$ = $1 * $3; }
 ;

term: NUMBER
 | OP_LFT exp OP_RGT  { $$ = $2; }
  ;
%%

int main(int argc, char **argv)
{
  yyparse();
}

int yyerror(char *s)
{
  fprintf(stderr, "error: '%s'\n", s);
}
```

Mari kita buat, `example.l`:

```
/* calculator source for flex */
%{
#include "example.tab.h"
%}

%%
[0-9]+ { printf("L: NUMBER = %s\n", yytext); yylval = atoi(yytext); return NUMBER; }
"+"    { printf("L: OP_ADD\n"); return OP_ADD; }
"-"    { printf("L: OP_SUB\n"); return OP_SUB; }
"*"    { printf("L: OP_MUL\n"); return OP_MUL; }
"("    { printf("L: OP_LFT\n"); return OP_LFT; }
")"    { printf("L: OP_RGT\n"); return OP_RGT; }
"="    { printf("L: OP_EQU\n"); return OP_EQU; }
"exit" { printf("L: exit\n");   return YYEOF; } /* YYEOF = 0 */
.      { /* ignore all other */ }
%%
```

Kemudian jalankan perintah berikut ini dari prompt shell untuk mencobanya:

```
$ bison -d example.y
$ flex example.l
$ gcc -lfl example.tab.c lex.yy.c -o example
$ ./example
1 + 2 * ( 3 + 1 ) =
L: NUMBER = 1
L: OP_ADD
L: NUMBER = 2
L: OP_MUL
L: OP_LFT
L: NUMBER = 3
L: OP_ADD
L: NUMBER = 1
L: OP_RGT
L: OP_EQU
Y: RESULT = 9

exit
L: exit
```

12.4 Alat analisis kode statis

Alat seperti Lint dapat membantu analisis kode statis otomatis.

Alat seperti indent dapat membantu peninjauan kode dengan memformat ulang kode sumber secara konsisten.

Alat seperti Ctags dapat membantu peninjauan kode dengan menghasilkan berkas indeks (atau tag) nama yang ditemukan dalam kode sumber.

Tip
Mengonfigurasi penyunting favorit Anda (`emacs` atau `vim`) untuk menggunakan plugin mesin lint asinkron membantu penulisan kode Anda. Plugin ini menjadi sangat kuat dengan memanfaatkan Language Server Protocol. Karena mereka bergerak cepat, menggunakan kode hulu mereka, bukan paket Debian mungkin merupakan pilihan yang baik.

paket	popcon	ukuran	deskripsi
vim-ale	I:0.74	2833	Asynchronous Lint Engine untuk Vim 8 dan NeoVim
vim-syntastic	I:1.9	1379	Hack pemeriksaan sintaks untuk vim
elpa-flycheck	V:0.1, I:1.6	1130	pemeriksaan sintaks sambil jalan modern untuk Emacs
elpa-relint	V:0.00, I:0.04	150	Pencari kesalahan regex Emacs Lisp
cppcheck-gui	V:0.11, I:0.97	7973	alat untuk analisis kode C/C++ statis (GUI)
shellcheck	V:4, I:17	22860	alat lint untuk skrip shell
pyflakes3	V:2, I:14	20	pemeriksa pasif dari program Python 3
pylint	V:4, I:19	2104	Pemeriksa statis kode python
perl	V:663, I:990	840	interpreter dengan pemeriksa kode statis internal: B::Lint(3perl)
rubocop	V:0.1, I:1.0	4502	Penganalisis kode statis Ruby
clang-tidy	V:2, I:11	20	alat linter C++ berbasis clang
splint	V:0.05, I:0.79	2325	alat untuk memeriksa secara statis program C untuk bug
flawfinder	V:0.05, I:0.43	187	alat untuk memeriksa kode sumber C/C++ dan mencari kelemahan keamanan
black	V:3, I:15	5868	pemformat kode Python tanpa kompromi
perltidy	V:0.4, I:2.9	3086	Pengindentasi dan pemformat ulang skrip Perl
indent	V:0.3, I:4.7	438	program pemformatan kode sumber bahasa C
astyle	V:0.2, I:2.5	770	Pengindentasi kode sumber untuk C, C++, Objective-C, C#, dan Java
bcpp	V:0.02, I:0.22	114	pemercantik C(++)
xmlindent	V:0.07, I:0.73	52	Pemformat ulang stream XML
global	V:0.2, I:1.5	1923	Alat pencarian dan penelusuran kode sumber
exuberant-ctags	V:1, I:12	341	membangun indeks berkas tag atas definisi kode sumber
universal-ctags	V:1, I:12	4238	membangun indeks berkas tag atas definisi kode sumber

Tabel 12.12: Daftar alat untuk analisis kode statis

12.5 Awakutu

Debug adalah bagian penting dari kegiatan pemrograman. Mengetahui cara men-debug program membuat Anda pengguna Debian yang baik yang dapat menghasilkan laporan bug yang berarti.

paket	popcon	ukuran	dokumentasi
gdb	V:92, I:162	13472	"`info gdb`" disediakan oleh `gdb-doc`
ddd	V:0.3, I:4.9	5094	"`info ddd`" disediakan oleh `ddd-doc`

Tabel 12.13: Daftar paket debug

12.5.1 Eksekusi gdb dasar

Debugger utama pada Debian adalah gdb(1) yang memungkinkan Anda untuk menginspeksi program saat dijalankan.

Mari kita pasang `gdb` dan program terkait dengan yang berikut ini.

```
# apt-get install gdb gdb-doc build-essential devscripts
```

Tutorial yang baik dari `gdb` dapat ditemukan:

- "`info gdb`"
- "Debugging dengan GDB" di `/usr/share/doc/gdb-doc/html/gdb/index.html`
- "tutorial di web"

Berikut adalah contoh sederhana menggunakan gdb(1) pada suatu "`program`" yang dikompilasi dengan opsi "`-g`" untuk menghasilkan informasi debugging.

```
$ gdb program
(gdb) b 1               # set break point at line 1
(gdb) run args          # run program with args
(gdb) next              # next line
...
(gdb) step              # step forward
...
(gdb) p parm            # print parm
...
(gdb) p parm=12         # set value to 12
...
(gdb) quit
```

Tip
Banyak perintah gdb(1) dapat disingkat. Ekspansi tab bekerja seperti di shell.

12.5.2 Debugging paket Debian

Karena semua biner yang dipasang harus di-strip pada sistem Debian secara baku, sebagian besar simbol debug dihapus dalam paket normal. Untuk men-debug paket Debian dengan gdb(1), paket `*-dbgsym` perlu dipasang (misalnya `coreutils-dbgsym` dalam kasus `coreutils`). Paket sumber menghasilkan paket `*-dbgsym` secara otomatis bersama dengan paket biner normal dan paket debug tersebut ditempatkan secara terpisah dalam arsip debian-debug. Silakan merujuk ke artikel di Debian Wiki untuk informasi lebih lanjut.

Jika paket yang akan di-debug tidak menyediakan paket `*-dbgsym`, Anda perlu memasangnya setelah membangun kembali dengan yang berikut.

```
$ mkdir /path/new ; cd /path/new
$ sudo apt-get update
$ sudo apt-get dist-upgrade
$ sudo apt-get install fakeroot devscripts build-essential
$ apt-get source package_name
$ cd package_name*
$ sudo apt-get build-dep ./
```

Perbaiki bug jika diperlukan.

Bump versi paket ke yang tidak bertabrakan dengan versi Debian resmi, misalnya yang ditambahi "`+debug1`" ketika meng-compile ulang versi paket yang ada, atau yang ditambahi "`~ pre1`" ketika meng-compile versi paket yang belum pernah dirilis dengan yang berikut.

```
$ dch -i
```

Kompilasi dan install paket beserta simbol debug dengan yang berikut ini.

```
$ export DEB_BUILD_OPTIONS="nostrip noopt"
$ debuild
$ cd ..
$ sudo debi package_name*.changes
```

Anda perlu memeriksa skrip build paket dan memastikan menggunakan "`CFLAGS = -g -Wall`" untuk mengkompilasi biner.

12.5.3 Mendapatkan backtrace

Ketika Anda mengalami crash program, melaporkan laporan kutu dengan informasi backtrace yang dipotong-dan-tempel adalah ide yang baik.

Backtrace dapat diperoleh dengan gdb(1) menggunakan salah satu pendekatan berikut:

- Pendekatan Crash-in-GDB:
  - Jalankan program dari GDB.
  - Menjadikan program crash.
  - Ketik "`bt`" pada prompt GDB.
- Pendekatan crash-first:
  - Perbarui berkas "**/etc/security/limits.conf**" untuk menyertakan hal-hal berikut:

    ```
    * soft core unlimited
    ```

  - Ketik "`ulimit -c unlimited`" ke prompt shell.
  - Menjalankan program dari prompt shell ini.
  - Crash-kan program untuk menghasilkan berkas core dump.
  - Muat berkas core dump ke GDB sebagai "`gdb gdb ./biner_program core`".
  - Ketik "`bt`" pada prompt GDB.

Untuk pengulangan tak hingga atau situasi papan ketik beku, Anda dapat memaksa untuk program crash dengan menekan `Ctrl-\` atau `Ctrl-C` atau mengeksekusi"`kill -ABRT PID`". (Lihat Bagian 9.4.12)

Tip
Seringkali, Anda melihat backtrace dimana satu atau lebih dari baris atas berada dalam ”`malloc()`” atau”g_`malloc()`”. Ketika ini terjadi, kemungkinan backtrace Anda tidak terlalu berguna. Cara termudah untuk menemukan beberapa informasi yang berguna adalah dengan mengatur variabel lingkungan ”`$MALLOC_CHECK_`” ke nilai 2 (malloc(3)). Anda dapat melakukan ini saat sedang menjalankan `gdb` dengan melakukan hal berikut.

```
$ MALLOC_CHECK_=2 gdb hello
```

12.5.4 Perintah gdb tingkat lanjut

perintah	deskripsi untuk tujuan perintah
`(gdb) thread apply all bt`	mendapatkan backtrace untuk semua thread bagi program multi-thread
`(gdb) bt full`	mendapatkan parameter yang ada pada stack pemanggilan fungsi
`(gdb) thread apply all bt full`	mendapatkan backtrace dan parameter sebagai kombinasi dari opsi sebelumnya
`(gdb) thread apply all bt full 10`	mendapatkan backtrace dan parameter untuk 10 panggilan teratas untuk memotong keluaran yang tidak relevan
`(gdb) set logging on`	menulis log keluaran `gdb` ke berkas (bakunya adalah ”`gdb.txt`”)

Tabel 12.14: Daftar perintah gdb tingkat lanjut

12.5.5 Periksa ketergantungan pada pustaka

Gunakan ldd(1) untuk mengetahui ketergantungan program pada pustaka dengan yang berikut ini.

```
$ ldd /usr/bin/ls
        librt.so.1 => /lib/librt.so.1 (0x4001e000)
        libc.so.6 => /lib/libc.so.6 (0x40030000)
        libpthread.so.0 => /lib/libpthread.so.0 (0x40153000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

Agar ls(1) bekerja di lingkungan ’chroot’, pustaka di atas harus tersedia di lingkungan ’chroot’ Anda.

Lihat Bagian 9.4.6.

12.5.6 Alat pelacakan panggilan dinamis

Ada beberapa alat pelacakan panggilan dinamis yang tersedia di Debian. Lihat Bagian 9.4.

12.5.7 Men-debug Galat X

Jika sebuah program GNOME `preview1` telah menerima kesalahan X, Anda akan melihat pesan sebagai berikut.

```
The program 'preview1' received an X Window System error.
```

Jika ini masalahnya, Anda dapat mencoba menjalankan program dengan ”`--sync`”, dan memutus pada fungsi ”`gdk_x_error`”untuk mendapatkan backtrace.

12.5.8 Alat deteksi kebocoran memori

Ada beberapa alat deteksi kebocoran memori yang tersedia di Debian.

paket	popcon	ukuran	deskripsi
libc6-dev	V:258, I:580	12863	mtrace(1): fungsi debugging malloc di glibc
valgrind	V:4, I:32	89318	debugger dan profiler memori
electric-fence	V:0.2, I:2.2	69	malloc(3) debugger
libdmalloc5	V:0.01, I:0.57	373	debug pustaka alokasi memori
duma	V:0.00, I:0.05	297	pustaka untuk mendeteksi buffer overrun dan under-run dalam program C dan C++
leaktracer	V:0.01, I:0.27	56	pelacak kebocoran memori untuk program C++

Tabel 12.15: Daftar alat deteksi kebocoran memori

12.5.9 Disassembly biner

Anda dapat men-disassembly kode biner memakai objdump(1) dengan yang berikut ini.

```
$  objdump -m i386 -b binary -D /usr/lib/grub/x86_64-pc/stage1
```

Catatan
gdb(1) dapat digunakan untuk men-disassembly kode secara interaktif.

12.6 Alat build

paket	popcon	ukuran	dokumentasi
make	V:145, I:562	1762	"info make" disediakan oleh make-doc
autoconf	V:27, I:193	2244	"info autoconf" disediakan oleh autoconf-doc
automake	V:26, I:192	1958	"info automake" disediakan oleh automake1.10-doc
libtool	V:23, I:176	1276	"info libtool" disediakan oleh libtool-doc
cmake	V:22, I:123	59735	cmake(1) sistem make open-source yang lintas platform
ninja-build	V:8, I:54	456	ninja(1) sistem build kecil yang paling dekat dalam semangat ke Make
meson	V:6, I:29	4402	meson(1) sistem build produktivitas tinggi di atas ninja
xutils-dev	V:0.6, I:6.7	1495	imake(1), xmkmf(1), etc.

Tabel 12.16: Daftar paket alat build

12.6.1 Make

Make adalah utilitas untuk memelihara kelompok program. Saat eksekusi make(1), make membaca berkas aturan, "Makefile", dan memperbarui target jika itu tergantung pada berkas prasyarat yang telah dimodifikasi sejak target terakhir dimodifikasi, atau jika target tidak ada. Pelaksanaan pembaruan ini dapat terjadi secara bersamaan.

Sintaks berkas aturan adalah sebagai berikut.

```
target: [ prerequisites ... ]
 [TAB]  command1
 [TAB]  -command2 # ignore errors
 [TAB]  @command3 # suppress echoing
```

Di sini "`[TAB]`" adalah kode TAB. Setiap baris ditafsirkan oleh shell setelah membuat substitusi variabel. Gunakan "\" di akhir baris untuk melanjutkan skrip. Gunakan "$$" untuk memasukkan "$" bagi nilai lingkungan untuk skrip shell.

Aturan implisit untuk target dan prasyarat dapat ditulis, misalnya, dengan yang berikut.

```
%.o: %.c header.h
```

Di sini, target berisi karakter "%" (tepat satu dari mereka). "%" bisa cocok dengan sebarang sub string tidak kosong dalam nama berkas target yang sebenarnya. Prasyarat juga menggunakan "%" untuk menunjukkan bagaimana nama mereka berhubungan dengan nama target yang sebenarnya.

variabel otomatis	nilai
$@	target
$<	prasyarat pertama
$?	semua prasyarat yang lebih baru
$^	semua prasyarat
$*	"%" cocok dengan stem dalam pola target

Tabel 12.17: Daftar variabel otomatis make

ekspansi variabel	deskripsi
`foo1 := bar`	ekspansi satu kali
`foo2 = bar`	ekspansi rekursif
`foo3 += bar`	tambah

Tabel 12.18: Daftar ekspansi variabel make

Jalankan "`make -p -f/dev/null`" untuk melihat aturan internal otomatis.

12.6.2 Autotools

Autotools adalah keluarga alat pemrograman yang dirancang untuk membantu dalam membuat paket kode sumber portabel ke banyak sistem mirip Unix.

- Autoconf adalah alat untuk menghasilkan skrip shell "`configure`" dari "`configure.ac`".
  - "`configure`" digunakan kemudian untuk menghasilkan "`Makefile`" dari templat "`Makefile.in`".
- Automake adalah alat untuk menghasilkan"`Makefile.in`" dari "`Makefile.am`".
- Libtool adalah skrip shell untuk mengatasi masalah portabilitas perangkat lunak saat menyusun pustaka bersama dari kode sumber.

12.6.2.1 Mengkompilasi dan menginstall program



Awas
Jangan menimpa berkas sistem dengan program yang dikompilasi saat menginstalnya.

Debian tidak menyentuh berkas di "`/usr/local/`" atau "`/opt`". Jadi jika Anda mengkompilasi program dari sumber, pasang saja ke "`/usr/local/`" sehingga tidak mengganggu Debian.

```
$ cd src
$ ./configure --prefix=/usr/local
$ make # this compiles program
$ sudo make install # this installs the files in the system
```

12.6.2.2 Menghapus instalasi program

Jika Anda memiliki sumber asli dan jika itu menggunakan autoconf(1)/automake(1) dan jika Anda dapat mengingat bagaimana Anda mengonfigurasinya, jalankan sebagai berikut untuk menghapus instalasi program.

```
$ ./configure all-of-the-options-you-gave-it
$ sudo make uninstall
```

Atau, jika Anda benar-benar yakin bahwa proses instalasi menempatkan berkas hanya di bawah "`/usr/local/`" dan tidak ada yang penting di sana, Anda dapat menghapus semua isinya dengan berikut ini.

```
# find /usr/local -type f -print0 | xargs -0 rm -f
```

Jika Anda tidak yakin di mana berkas dipasang, Anda harus mempertimbangkan untuk menggunakan checkinstall(8) dari paket `checkinstall`, yang menyediakan path bersih untuk pembongkaran. Itu sekarang mendukung untuk membuat paket Debian dengan opsi "`-D`".

12.6.3 Meson

Sistem build perangkat lunak telah berkembang:

- Autotools di bagian atas Make telah menjadi standar de facto untuk infrastruktur bangunan portabel sejak 1990-an. Ini sangat lambat.
- CMake awalnya dirilis pada tahun 2000 meningkatkan kecepatan secara signifikan tetapi aslinya dibangun di atas Make yang inheren lambat. Kini Ninja bisa menjadi backend-nya.)
- Ninja awalnya dirilis pada tahun 2012 dimaksudkan untuk menggantikan Make untuk kecepatan build yang lebih ditingkatkan tetapi juga dirancang untuk memiliki berkas masukan yang dihasilkan oleh sistem build tingkat yang lebih tinggi.
- Meson awalnya dirilis pada tahun 2013 adalah sistem build tingkat tinggi yang baru, populer, dan cepat yang menggunakan Ninja sebagai backend-nya.

Lihat dokumen yang ditemukan di "The Meson Build system" dan "The Ninja build system".

12.7 Web

Halaman web dinamis interaktif dasar dapat dibuat sebagai berikut.

- Kueri disajikan kepada pengguna peramban menggunakan formulir HTML.
- Mengisi dan mengklik entri formulir mengirimkan salah satu string URL berikut dengan parameter yang dikodekan dari peramban ke server web.
  - "`https://www.foo.dom/cgi-bin/program.pl?VAR1=VAL1&VAR2=VAL2&VAR3=VAL3`"

  - ”`https://www.foo.dom/cgi-bin/program.py?VAR1=VAL1&VAR2=VAL2&VAR3=VAL3`”
  - ”`https://www.foo.dom/program.php?VAR1=VAL1&VAR2=VAL2&VAR3=VAL3`”
- ” `%nn`” dalam URL diganti dengan sebuah karakter dengan nilai heksadesimal `nn`.
- Variabel lingkungan diatur sebagai: ”`QUERY_STRING="VAR1=VAL1 VAR2=VAL2 VAR3=VAL3"`”.
- Program CGI (salah satu dari ”`program.*`”) di server web mengeksekusi dirinya sendiri dengan variabel lingkungan ”`$QUERY_STRING`”.
- `stdout` program CGI dikirim ke peramban web dan disajikan sebagai halaman web dinamis interaktif.

Untuk alasan keamanan lebih baik tidak membuat hack baru untuk mengurai parameter CGI. Ada modul yang telah mapan untuk mereka di Perl dan Python. PHP hadir dengan fungsi-fungsi ini. Ketika penyimpanan data klien diperlukan, cookie HTTP digunakan. Ketika pemrosesan data sisi klien diperlukan, Javascript sering digunakan.

Untuk informasi lebih lanjut, lihat Common Gateway Interface, The Apache Software Foundation, dan JavaScript.

Mencari ”tutorial CGI” di Google dengan mengetik URL yang dikodekan https://www.google.com/search?hl=en&ie=UTF-8&q=CGI+tutorial langsung ke alamat peramban adalah cara yang baik untuk melihat skrip CGI beraksi di server Google.

12.8 Terjemahan kode sumber

Ada program untuk mengonversi kode sumber.

paket	popcon	ukuran	kata kunci	deskripsi
`perl`	V:663, I:990	840	AWK → PERL	mengonversi kode sumber dari AWK ke PERL: a2p(1)
`f2c`	V:0.1, I:1.9	443	FORTRAN → C	mengonversi kode sumber dari FORTRAN 77 ke C/C++: f2c(1)
`intel2gas`	V:0.00, I:0.16	178	NASM → GAS	konverter dari NASM (format Intel) ke GNU Assembler (GAS)

Tabel 12.19: Daftar alat terjemahan kode sumber

12.9 Membuat paket Debian

Jika Anda ingin membuat paket Debian, baca berikut ini.

- Bab 2 untuk memahami sistem paket dasar
- Bagian 2.7.13 untuk memahami proses porting dasar
- Bagian 9.11.4 untuk memahami teknik chroot dasar
- debuild(1), dan sbuild(1)
- Bagian 12.5.2 untuk kompilasi ulang untuk pengawakutuan (debugging)
- Panduan untuk Pemelihara Debian (paket `debmake-doc`)
- Referensi Pengembang Debian (paket `developers-reference`)
- Manual Kebijakan Debian (paket `debian-policy`)

Ada paket seperti `debmake`, `dh-make`, `dh-make-perl`, dll., yang membantu pengemasan.

Lampiran A

Lampiran

Berikut adalah latar belakang dokumen ini.

A.1 Labirin Debian

Sistem Linux adalah platform komputasi yang sangat kuat untuk komputer jaringan. Namun, belajar bagaimana menggunakan semua kemampuannya tidaklah mudah. Menyiapkan antrian pencetak LPR dengan pencetak non-PostScript adalah contoh yang baik dari titik sandung. (Tidak ada masalah lagi karena instalasi yang lebih baru menggunakan sistem CUPS baru.)

Ada peta lengkap dan terperinci yang disebut ”KODE SUMBER”. Ini sangat akurat tetapi sangat sulit untuk dipahami. Ada juga referensi yang disebut HOWTO dan mini-HOWTO. Mereka lebih mudah dimengerti tetapi cenderung memberikan terlalu banyak detail dan kehilangan gambaran besarnya. Saya kadang-kadang memiliki masalah menemukan bagian yang tepat dalam HOWTO panjang ketika saya membutuhkan beberapa perintah untuk dipanggil.

Saya berharap ”Debian Reference (versi 2.152)” (2026-10-10 14:56:25 UTC) memberikan arahan awal yang baik bagi orang-orang di labirin Debian.

A.2 Riwayat hak cipta

Referensi Debian diprakarsai oleh saya, Osamu Aoki <osamu di debian dot org>, sebagai memo administrasi sistem pribadi. Banyak konten berasal dari pengetahuan yang saya peroleh dari milis debian-user dan sumber daya Debian lainnya.

Mengikuti saran dari Josip Rodin, yang sangat aktif dengan Debian Documentation Project (DDP), ”Debian Reference (versi 1, 2001-2007)” diciptakan sebagai bagian dari dokumen DDP.

Setelah 6 tahun, saya menyadari bahwa ”Debian Reference (versi 1)” yang asli sudah ketinggalan zaman dan mulai menulis ulang banyak konten. ”Debian Reference (versi 2)” yang baru, dirilis pada tahun 2008.

Saya telah memperbarui ”Debian Reference (versi 2)” untuk membahas topik baru (Systemd, Wayland, IMAP, PipeWire, kernel Linux 5.10) dan menghapus topik usang (SysV init, CVS, Subversion, SSH protokol 1, kernel Linux sebelum 2.5). Referensi ke situasi rilis Jessie 8 (2015-2020) atau yang lebih tua sebagian besar dihapus.

”Debian Reference (versi 2.152)” (2026-10-10 14:56:25 UTC) ini mencakup sebagian besar rilis Debian Trixie (`=stable`) dan Forky (`=testing`).

Isi tutorial dapat dilacak asal-usulnya dan inspirasinya dalam yang berikut.

- ”Panduan Pengguna Linux” by Larry Greenfield (December 1996)

  - usang oleh ”Debian Tutorial”
- ”Debian Tutorial” oleh Havoc Pennington (11 Desember 1998)
  - ditulis sebagian oleh Oliver Elphick, Ole Tetlie, James Treacy, Craig Sawyer, dan Ivan E. Moore II
  - usang oleh ”Debian GNU / Linux: Panduan untuk Instalasi dan Penggunaan”
- ”Debian GNU / Linux: Panduan untuk Instalasi dan Penggunaan” by John Goerzen and Ossama Othman (1999)
  - usang oleh ”Debian Reference (versi 1)”

Paket dan deskripsi arsip dapat dilacak beberapa asal-usul mereka dan inspirasi mereka dalam yang berikut.

- ”Debian FAQ” (versi Maret 2002, ketika ini dikelola oleh Josip Rodin)

Konten lainnya dapat dilacak beberapa asal-usul mereka dan inspirasi mereka dalam yang berikut.

- ”Debian Reference (versi 1)” oleh Osamu Aoki (2001–2007)
  - usang oleh ”Debian Reference (versi 2)” yang lebih baru pada tahun 2008.

”Debian Reference (versi 1)” yang sebelumnya, diciptakan dengan banyak kontributor.

- isi utama kontribusi pada topik konfigurasi jaringan oleh Thomas Hood
- kontribusi konten yang signifikan pada topik terkait X dan VCS oleh Brian Nelson
- bantuan pada pembuatan skrip dan banyak koreksi konten oleh Jens Seidel
- proofreading ekstensif oleh David Sewell
- banyak kontribusi oleh para penerjemah, kontributor, dan pelapor bug

Banyak halaman manual dan halaman info pada sistem Debian serta halaman web hulu dan dokumen Wikipedia digunakan sebagai referensi utama untuk menulis dokumen ini. Sejauh Osamu Aoki pertimbangkan dalam penggunaan yang adil, banyak bagian dari mereka, terutama definisi perintah, digunakan sebagai potongan frase setelah upaya editorial yang cermat untuk memasukkan mereka ke dalam gaya dan tujuan dari dokumen ini.

Deskripsi debugger gdb diperluas menggunakan konten wiki Debian di backtrace dengan persetujuan oleh Ari Pollak, Loïc Minier, dan Dafydd Harries.

Isi dari ”Debian Reference (version 2.152)” saat ini (2026-10-10 14:56:25 UTC) sebagian besar pekerjaan saya sendiri kecuali seperti yang disebutkan di atas. Ini telah diperbarui oleh para kontributor juga.

Penulis, Osamu Aoki, berterima kasih kepada semua orang yang membantu membuat dokumen ini mungkin.

A.3 Format dokumen

Sumber dokumen asli bahasa Inggris saat ini ditulis dalam berkas XML DocBook. Sumber XML DocBook ini dikonversi ke HTML, teks polos, PostScript, dan PDF. (Beberapa format mungkin dilewati untuk distribusi.)