



SISTEM MANAJEMEN PAKET
UBUNTU
EBOOK PANDUAN UNTUK PEMULA

ADE MALSASA AKBAR

Kata Pengantar

Bismillahirrahmanirrahim.

Alhamdulillah, setelah menulis 12 artikel manajemen paket Ubuntu di Linuxku.com selama periode 2016-2017, akhirnya saya bisa menghasilkan buku elektronik darinya. Buku ini dibuat dalam Bahasa Indonesia sebagai pegangan pengguna mengelola software di sistem operasi Ubuntu GNU/Linux. Materi di dalamnya mencakup bagaimana instal, hapus, perbarui, unduh paket-paket software sampai pengguna mampu mengoperasikan Ubuntu sehari-hari selancar Windows. Materi ini dinamakan Sistem Manajemen Paket (SMP) yang merupakan 50% materi yang pasti dipelajari pengguna di sistem operasi Ubuntu. Target pembaca buku ini ialah pemerintahan, UKM, pengguna rumahan, pelajar & mahasiswa, dan masyarakat secara umum. Buku ini berlisensi CC BY-SA 3.0 agar pembaca bebas menyebarluaskan buku ini ke seluruh Indonesia.

Ebook ini sejatinya adalah kumpulan dari 12 tulisan yang telah terbit di Linuxku.com:

- [SMP Ubuntu 1: Perkenalan SMP](#)
- [SMP Ubuntu 2: Mengenal Repositori](#)
- [SMP Ubuntu 3: Mengenal Dpkg](#)
- [SMP Ubuntu 4: Mengenal Apt](#)
- [SMP Ubuntu 5: Sources.list](#)
- [SMP Ubuntu 6: GUI Front-End](#)
- [SMP Ubuntu 7: Mengenal Apt-cache](#)
- [SMP Ubuntu 8: Synaptic](#)
- [SMP Ubuntu 9: Memperoleh Dependensi Tanpa Internet](#)
- [SMP Ubuntu 10: Memperoleh Dependensi via Synaptic Script](#)
- [SMP Ubuntu 11: Repositori Pihak Ketiga/PPA](#)
- [SMP Ubuntu 12: Repositori CDROM](#)

Buku ini tidaklah sempurna. Karena itu saya mengharapkan gotong royong, pertanyaan, saran, dan kritik pembaca dialamatkan ke email teknoloid@gmail.com. Semoga hadirnya buku ini bermanfaat sebesar-besarnya bagi seluruh masyarakat Indonesia dan menginspirasi hadirnya buku-buku serupa.

Wassalamu'alaykum,

Ade Malsasa Akbar

Detail Buku

- Judul: Sistem Manajemen Paket Ubuntu untuk Pemula
- Deskripsi: ebook kumpulan 12 artikel manajemen paket Ubuntu dari <http://www.linuxku.com>
- Penulis: Ade Malsasa Akbar <teknoloid@gmail.com>
- Tebal: 80 halaman
- Bab: 12
- Tanggal terbit: 7 April 2017
- Tingkat: pemula
- Format: buku dipublikasikan digital dalam PDF dan ODT
- Produksi: buku ditulis dengan LibreOffice, kover dibuat dengan Inkscape
- Lisensi: Creative Commons Attribution-ShareAlike 3.0
<https://creativecommons.org/licenses/by-sa/3.0/>
- Penerbit: <http://www.linuxku.com>

Daftar Isi

| | |
|---|----|
| Daftar Isi..... | 2 |
| Bab 1: Perkenalan..... | 5 |
| Ringkasan..... | 5 |
| Menenal Distribusi Software..... | 5 |
| Menenal Format Paket Software..... | 6 |
| Menenal Dependensi..... | 6 |
| Menenal Sistem Manajemen Paket..... | 7 |
| Menenal Tugas-Tugas Manajer Paket Ubuntu..... | 8 |
| Menenal Manajer Paket GUI..... | 8 |
| Menenal Berkas Konfigurasi..... | 8 |
| Menenal Aktivitas yang Sering Dilakukan..... | 9 |
| Kamus Istilah..... | 9 |
| Pranala Penting..... | 10 |
| Bab 2: Menenal Repositori..... | 11 |
| Ringkasan..... | 11 |
| Perbandingan Platform..... | 11 |
| Definisi Repositori..... | 12 |
| Definisi Mirror..... | 12 |
| Contoh Mirror Repositori Ubuntu..... | 12 |
| Klasifikasi Repositori Ubuntu | 13 |
| Berkas Konfigurasi Lokal..... | 14 |
| Contoh Isi sources.list | 14 |
| Metode Mengonfigurasi Pemakaian Repositori..... | 15 |
| Distribusi Repositori..... | 15 |
| Referensi | 15 |
| Bab 3: Menenal Dpkg..... | 17 |
| Ringkasan..... | 17 |
| Tentang Dpkg..... | 17 |
| Melihat Daftar Paket..... | 17 |
| Instal Paket..... | 18 |
| Hapus Paket | 18 |
| Contoh Penggunaan..... | 18 |
| Singkatan-Singkatan..... | 20 |
| Kamus Istilah | 20 |
| Bab 4: Menenal Apt..... | 22 |
| Ringkasan..... | 22 |
| Perkenalan..... | 22 |
| Program-Program APT..... | 23 |
| Reload Repositori..... | 23 |
| Instal Paket..... | 25 |
| Hapus Paket..... | 26 |
| Upgrade Seluruh Paket..... | 27 |
| Upgrade Versi Rilis Ubuntu..... | 28 |
| Hapus APT Cache..... | 28 |
| Kamus Istilah..... | 29 |

| | |
|---|----|
| Memperoleh Bantuan..... | 29 |
| Bab 5: sources.list..... | 31 |
| Perkenalan..... | 31 |
| Klasifikasi Repositori Ubuntu..... | 31 |
| Klasifikasi Repositori Debian..... | 32 |
| Format Setelan Sumber Repositori..... | 33 |
| Contoh sources.list Ubuntu 16.04..... | 33 |
| Cara Menyunting sources.list..... | 35 |
| Yang Tidak Boleh Dilakukan..... | 36 |
| Kamus Istilah..... | 36 |
| Referensi..... | 36 |
| Bab 6: GUI Front-End..... | 37 |
| Perkenalan Istilah GUI Front-End..... | 37 |
| Contoh GUI Front-End untuk Manajemen Paket..... | 38 |
| Catatan Penulis..... | 42 |
| Referensi | 42 |
| Bab 7: Mengenal Apt-Cache..... | 43 |
| Sekilas Mengingat APT..... | 43 |
| 1. Output Normal..... | 43 |
| 2. Mencari Paket..... | 45 |
| 3. Mengetahui Status Repositori..... | 46 |
| 4. Mengetahui Beda Versi Paket pada Sejumlah Repositori yang Berbeda..... | 47 |
| 5. Mengetahui Metadata Paket..... | 48 |
| Kamus Istilah..... | 50 |
| Bab 8: Penggunaan Synaptic di Ubuntu..... | 52 |
| Perkenalan Synaptic..... | 52 |
| Menginstal Synaptic..... | 52 |
| Membuka Synaptic | 53 |
| 1. Penggunaan Pertama..... | 53 |
| 2. Reload..... | 54 |
| 3. Instal..... | 55 |
| 4. Hapus..... | 56 |
| 5. Upgrade..... | 57 |
| Bab 9: Memperoleh Dependensi Secara Offline..... | 59 |
| Syarat..... | 59 |
| Langkah 1: Memperoleh URL Seluruh Dependensi..... | 60 |
| Langkah 2: Mengunduh Seluruh Dependensi..... | 61 |
| Langkah 3: Menginstal Seluruh Dependensi..... | 62 |
| Penjelasan..... | 62 |
| Catatan Penting..... | 62 |
| Bab 10: Memperoleh Daftar Dependensi via Synaptic Script..... | 64 |
| Syarat..... | 64 |
| Langkah 1: Memilih Paket..... | 65 |
| Langkah 2: Memodifikasi Berkas Script..... | 65 |
| Langkah 3: Mengunduh Seluruh Dependensi..... | 66 |
| Langkah 4: Menginstal Seluruh Dependensi..... | 67 |
| Bab 11: Repositori Pihak Ketiga (PPA)..... | 68 |
| 1. Mengenal Repositori..... | 68 |
| 2. Mengenal Repositori Pihak Ketiga..... | 68 |
| 3. Mengenal Launchpad..... | 69 |

| | |
|---|----|
| 4. Contoh PPA..... | 70 |
| 5. Cara Mencari PPA..... | 70 |
| 6. Yang Perlu Diperhatikan..... | 71 |
| 7. Memasang Sebuah PPA..... | 71 |
| 8. Mengecek Versi Software dari PPA..... | 72 |
| 9. Menginstal Software dari PPA..... | 73 |
| 10. Menghapus Software yang Terinstal dari PPA..... | 73 |
| 11. Menghapus Sebuah PPA..... | 74 |
| 12. Catatan Penting..... | 74 |
| Bab 12: Repositori CDRom..... | 75 |
| Tentang CD..... | 75 |
| Peringatan..... | 75 |
| Percontohan..... | 75 |
| Langkah 1..... | 76 |
| Langkah 2..... | 76 |
| Langkah 3..... | 76 |
| Langkah 4..... | 77 |
| Selamat Menginstal..... | 79 |

Bab 1: Perkenalan

Tulisan ini ditujukan sebagai pengantar umum sistem manajemen paket untuk setiap pengguna GNU/Linux terkhusus Ubuntu. Di dalam tulisan ini dijelaskan mengapa ada sistem manajemen paket dan bagaimana secara umum penggunaannya. Di bagian akhir tulisan, terdapat tambahan referensi luar bagi pembaca yang ingin belajar lebih lanjut. Tulisan ini dilengkapi dengan kamus istilah demi memudahkan pembaca. Tulisan ini adalah perwujudan dari [pengantar GNU/Linux untuk pemula](#). Semoga tulisan ini bermanfaat.

Ringkasan

1. Mengetahui Distribusi Software
2. Mengetahui Format Paket Software
3. Mengetahui Dependensi
4. Mengetahui Sistem Manajemen Paket
5. Mengetahui Tugas-Tugas Manajer Paket
6. Mengetahui Manajer Paket GUI
7. Mengetahui Berkas Konfigurasi
8. Mengetahui Aktivitas yang Sering Dilakukan
9. Kamus Istilah

Mengenal Distribusi Software

Di dalam komunitas GNU/Linux, ketika disebut kata software, maksudnya adalah [free software](#) (sebagian orang menamakannya [open source](#)). Sebuah free software itu didistribusikan oleh developernya (upstream) dalam bentuk **source code**. Upstream biasanya mengemas setiap source code dalam berkas arsip tarball seperti .tar.gz atau .tar.bz2 (mirip dengan .zip). Para developer distribusi GNU/Linux (downstream) mengambil source code-source code itu secara terpisah-pisah dari upstream-upstream di seluruh dunia kemudian memrosesnya menjadi software-software dalam bentuk **binary**. Setiap software diproses dan dikemas menjadi satu buah **package** (paket). Proses perubahan dari source code jadi binary ini dilakukan sebagai layanan kepada pengguna suatu distribusi GNU/Linux, terkadang berbayar (seperti Red Hat Enterprise Linux) terkadang gratis (seperti Ubuntu). Paket-paket yang sudah jadi disimpan di dalam server yang disebut dengan nama **repositori**. Setiap [distribusi GNU/Linux](#) adalah downstream, dan setiap downstream memiliki repositorinya masing-masing. Setiap repositori umumnya berisi puluhan ribu paket software.

Pengguna akhir memperoleh software bagi distribusi yang dipakainya dari repositori distribusi tersebut. Misalnya, pengguna Ubuntu akan mengambil software dari repositori Ubuntu pula. Yang diambil adalah software dalam bentuk paket. Cara mengambilnya adalah dengan memakai sistem manajemen paket.

Mengenal Format Paket Software

Di dunia GNU/Linux, terdapat dua jenis format paket (package) yang paling populer. Mereka adalah **DEB** dan **RPM** (.deb dan .rpm). Format DEB digunakan di sistem operasi keluarga Debian, misalnya Ubuntu dan Linux Mint. Format RPM digunakan di sistem operasi keluarga Red Hat Enterprise Linux, misalnya Fedora dan CentOS. Maka di Ubuntu paket-paket software dikemas dalam format .deb, sedangkan di Fedora berformat .rpm.

Sebagai perbandingan, di Microsoft Windows format paket software resmi adalah **MSI** (.msi). Sedangkan format dari berkas software dalam bentuk binary adalah **EXE** (.exe). MSI bisa saling dibandingkan dengan DEB atau RPM. MSI memuat software dan metadata, sebagaimana DEB atau RPM juga memuat hal yang sama.

Mengenal Dependensi

Inggris: *dependency*, Indonesia: *dependensi*; berarti *ketergantungan*. Di dalam komunitas free software, dependensi adalah istilah yang merujuk kepada ketergantuan suatu software kepada software lain. Jika A tergantung pada B, maka B harus diinstal dahulu sebelum A. Karena software memiliki dua bentuk, maka dependensi juga ada dalam dua bentuk. Software dalam bentuk source code bisa memiliki dependensi, software dalam bentuk binary juga bisa memiliki dependensi. Pembaca yang belum mengenal dependensi bisa mengingat kejadian ".Net Framework" yang harus diinstal terlebih dahulu sebelum menginstal software A ketika di Microsoft Windows. Maka dalam hal ini .Net Framework adalah dependensi bagi software A.

Pada praktiknya di lapangan, dependensi itu sangat banyak terjadi antara software-software yang tersedia di komunitas free software. Hal ini karena free software dibangun atas prinsip kooperasi (gotong royong) terutama di GNU/Linux dengan prinsip warisan UNIX *do one thing and do it well*. Dependensi yang terjadi di suatu sistem GNU/Linux adalah pada paket software (software package). Untuk menginstal suatu paket software, sering kali pengguna harus menginstal dahulu paket software yang lain. Dan paket yang lain itu memiliki dependensi lagi kepada paket lain pula. Dan seterusnya. Dalam banyak kejadian, untuk menginstal satu software (misalnya KDE) dibutuhkan puluhan atau ratusan paket berbeda-beda karena ada dependensi dan bercabang pula. Inilah praktik di lapangan dan ini adalah **masalah**.

Maka untuk dapat menyelesaikan masalah itu komunitas membuat suatu sistem di dalam sistem manajemen paket yang disebut *dependency resolver* (pencari dependensi). Suatu *dependency*

resolver adalah program yang memiliki kemampuan mencari seluruh dependensi secara otomatis terhadap suatu paket di sistem GNU/Linux yang digunakan. Dependency resolver sangat mempermudah pengguna akhir dalam menginstal suatu software di GNU/Linux karena pengguna tidak perlu mencarinya secara manual lagi. Contoh *dependency resolver* di Ubuntu adalah **apt**, di Arch Linux adalah **pacman**, di CentOS dan Fedora adalah **yum**, di openSUSE adalah **zypper**, dan seterusnya.

Mengenal Sistem Manajemen Paket

Sistem manajemen paket (disingkat *smp*) adalah mekanisme pengelolaan paket software di suatu sistem operasi. Pengelolaan yang dimaksud adalah menginstal, menghapus, dan memutakhirkan paket. Istilah aslinya adalah *package management system*.

Yang dimaksud sebuah **paket** adalah sebuah berkas arsip yang memuat suatu software dan suatu metadata. Paket yang memuat selain software biasanya memuat gambar, atau audio, atau video, atau apa saja, selalu disertai pula dengan metadata. Paket adalah metode setiap developer distribusi GNU/Linux untuk mendistribusikan software melalui repositori mereka masing-masing. Pengguna menginstal suatu software adalah dengan menginstal paket tersebut dari repositori. Paket di Ubuntu berformat DEB (.deb).

Yang dimaksud **metadata** adalah informasi penting terkait dengan isi paket. Isi metadata di antaranya nama paket, daftar dependensi paket, deskripsi, ukuran, dan sebagainya. Setiap paket selalu berisi metadata.

Komponen suatu sistem manajemen paket setidaknya dua program: **package manager** (manajer paket) dan **update manager** (manajer pemutakhiran). Hampir semua distribusi GNU/Linux menyertakan kedua komponen utama ini. Kedua jenis program ini selalu berwujud command line dan dijalankan dari shell (Terminal). Hanya saja, biasanya, distribusi seperti Ubuntu menambahkan versi GUI dari setiap program.

Package manager adalah program inti di dalam *smp*, tugas pokoknya pasang dan hapus paket di sistem. Tugas untuk pemasangan adalah membuka, mengeluarkan isi, memeriksa metadata, lalu meletakkan setiap berkas di dalam paket ke posisinya masing-masing di dalam sistem. Tugas untuk penghapusan adalah membaca kembali metadata, menemukan skrip untuk penghapusan, melakukan penghapusan satu demi satu secara otomatis sampai paket itu terhapus dari sistem. **Update manager** adalah program yang bekerja di atas package manager, tugas utamanya adalah *dependency resolving* (mencari dependensi), mengunduh paket melalui jaringan, dan menemukan pemutakhiran versi paket (sesuai namanya). Update manager biasanya hanya mencari dependensi dan mengunduh, sedangkan pemasangan dan penghapusan paket tetap dilakukan oleh package manager.

Catatan: di Arch Linux, package manager dan update manager menyatu dalam satu program bernama **pacman**. Di Ubuntu, package manager terpisah dari update manager, masing-masing bernama **dpkg** dan **apt**.

Mengenal Tugas-Tugas Manajer Paket Ubuntu

- Package manager: **dpkg**
- Update manager: **apt**

Ubuntu adalah keturunan Debian, maka manajer paket di Ubuntu sama dengan di Debian. **Tugas dpkg** sebagaimana dijelaskan di atas, pasang dan hapus paket. Maka jika pengguna Ubuntu melakukan install/uninstall terhadap suatu software aplikasi dari repositori, sesungguhnya pengguna sedang memakai jasa dpkg di situ. **Tugas apt** sebagaimana disebut, pokoknya adalah untuk *dependency resolving*. Maka jika pengguna Ubuntu me-reload informasi repositori, mengunduh suatu paket, melakukan upgrade, sebenarnya dia memakai jasa apt di situ. Bisa diasosiasikan bahwa dpkg itu *internal*, apt itu *eksternal*. Karena dpkg urusannya dengan paket di sistem, sedangkan apt urusannya dengan repositori.

Mengenal Manajer Paket GUI

- Package manager: **GDebi**
- Update manager: **Synaptic, Ubuntu Software Center**

Karena dpkg dan apt adalah program konsol, developer membuat software versi GUI yang fungsinya sama atau menggunakan kemampuan mereka. **Untuk perkara pasang/hapus paket**, pengguna Ubuntu bisa menggunakan GDebi. Praktis pemakaiannya hanya dengan klik kanan secara GUI. **Untuk perkara update management**, pengguna bisa menggunakan Synaptic atau Ubuntu Software Center. Keduanya menayangkan daftar seluruh paket yang dapat diinstal secara GUI dan menginstal/menghapus/memutakhirkan secara GUI pula. Praktis, semua versi GUI ini sangat memudahkan pengguna akhir yang memilih tidak menggunakan program konsol.

Mengenal Berkas Konfigurasi

Terdapat banyak berkas penting di dalam sistem manajemen paket Ubuntu. Namun yang paling penting yang layak diketahui ada beberapa sebagai berikut.

- Database dpkg: **/var/lib/dpkg/status**: berkas teks yang berisi daftar paket yang sudah terinstal di dalam sistem. Berkas ini dibangun dari metadata dari setiap paket terinstal.

- Database apt: **/var/lib/apt/lists/**: direktori yang berisi berkas-berkas teks daftar paket yang berada pada repositori. Berkas ini diperoleh dari *reloading* yaitu perintah apt-get update.
- Konfigurasi dpkg: **/etc/dpkg/dpkg.conf**: berkas teks berisi konfigurasi dpkg.
- Konfigurasi apt: **/etc/apt/apt.conf**: berkas teks berisi konfigurasi apt.
- Konfigurasi sumber repositori: **/etc/apt/sources.list**: berkas teks berisi daftar sumber paket (repositori) yang dituju oleh sistem.
- Konfigurasi sumber repositori pihak ketiga: **/etc/apt/sources.list.d/**: direktori yang berisi berkas-berkas teks (.list) yang berisi daftar sumber paket (repositori) pihak ketiga yang dituju oleh sistem.
- Tembolok paket: **/var/cache/apt/archives/**: direktori yang berisi seluruh paket DEB yang diunduh oleh apt. Dari sinilah apt menyuruh dpkg untuk menginstal mereka satu per satu di setiap sesi instalasi.
- Riwayat apt: **/var/log/apt/**: direktori yang berisi sejarah pasang/hapus/mutakhir dari aktivitas apt.

Mengenal Aktivitas yang Sering Dilakukan

Di antara sekian banyak komponen program, berkas konfigurasi, direktori konfigurasi, dari sistem manajemen paket di Ubuntu, terdapat sejumlah aktivitas umum yang paling sering dilakukan oleh pengguna sebagai berikut. Aktivitas ini tergolong favorit dan biasanya kebutuhan pengguna akhir hanya berkisar padanya.

- Menginstal paket dari repositori dengan apt-get.
- Menghapus paket terinstal dengan apt-get.
- Memutakhirkan paket terinstal dengan apt-get.
- Menginstal paket satu per satu dengan dpkg.
- Menyunting sources.list secara manual.
- Mencari nama paket di dalam database apt melalui GUI atau CLI.

Kamus Istilah

Upstream: di dalam komunitas free software, artinya adalah developer asli dari suatu software. Upstream juga berarti setiap software-nya. Contoh upstream adalah Proyek GNU, Proyek KDE, Proyek X11, Proyek LibreOffice, Proyek Apache Web Server, dan seterusnya.

Downstream: di dalam komunitas free software, artinya adalah developer di luar upstream. Downstream adalah developer yang mendistribusikan ulang upstream-upstream dalam bentuk baru.

Downstream inilah yang membuat distribusi GNU/Linux seperti Ubuntu. Distribusi = bentuk baru dan komposisi dari upstream-upstream = sistem operasi.

Command line: lawan dari graphical, metode komunikasi manusia-komputer dengan menggunakan perintah-perintah teks. Command line bersinonim dengan *console* atau konsol. Contoh command line adalah antarmuka MS-DOS.

GUI: *graphical user interface* (antarmuka pengguna grafikal), lawan dari command line; metode komunikasi manusia-komputer dengan menggunakan gambar. Contoh GUI adalah antarmuka Kubuntu.

Source code: bentuk primer dari software; berkas teks yang berisi kode-kode yang membangun suatu software; setiap software pasti dibangun dari source code, source code bagi software seperti resep bagi masakan.

Binary: bentuk sekunder dari software; berkas yang berisi angka-angka nol dan satu yang hanya dimengerti oleh komputer. Contoh binary misalnya .exe di Windows.

Pranala Penting

Bagi pemula yang hendak belajar lebih lanjut, berikut ini pranala (link) yang sangat bagus dijadikan referensi.

- <http://distrowatch.com/dwres.php?resource=package-management> "**Package Management Cheatsheet**": tabel perbandingan perintah-perintah sistem manajemen paket di antara semua distro GNU/Linux yang ada. Sangat direkomendasikan.
- <https://wiki.debian.org/DebianPackageManagement> "**DebianPackageManagement**": selang pandang yang mendetail mengenai sistem manajemen paket Debian.
- <https://wiki.debian.org/Synaptic> "**Synaptic**": pengenalan singkat Synaptic.
- <https://help.ubuntu.com/community/AptGet/Howto?action=show&redirect=AptGetHowto> "**Package management with APT**": artikel komprehensif mengenai apt-get dari dokumentasi resmi Ubuntu.

Bab 2: Mengenal Repositori

Tulisan ini memberikan pengetahuan umum mengenal repositori suatu distribusi GNU/Linux terkhusus Ubuntu. Dijelaskan mulai perbandingan budaya pengguna, pengertian repositori, contoh repositori, klasifikasi repositori, sampai metode umum mengonfigurasinya di Ubuntu pada komputer lokal. Tulisan ini adalah lanjutan dari artikel [Perkenalan Sistem Manajemen Paket](#). Semoga tulisan ini bermanfaat.

Ringkasan

1. Perbandingan Platform
2. Definisi Repositori
3. Definisi Mirror
4. Contoh Mirror Repositori Ubuntu
5. Klasifikasi Repositori Ubuntu
6. Berkas Konfigurasi Lokal
7. Contoh Isi *sources.list*
8. Metode Mengonfigurasi Pemakaian Repositori
9. Distribusi Repositori
10. Referensi

Perbandingan Platform

Di platform sistem operasi Microsoft Windows atau Apple OS X, pengguna komputer terbiasa menginstal paket software dengan mengambilnya satu demi satu dari masing-masing situs software tersebut. Jika ada 10.000 software secara teori ada 10.000 situs web yang dikunjungi. Umumnya pula di sana, paket software didistribusikan dalam boks CD (contohnya Adobe Photoshop).

Di platform GNU/Linux, terkhusus di Ubuntu, yang terjadi adalah kebalikannya. Mayoritas paket software yang dibutuhkan pengguna akhir telah disediakan oleh developer distribusi **di dalam repositori**. Anda mau software untuk desain grafis dan audio? Ada di repositori. Anda mau wallpaper dan web browser? Ada di sana. Total jumlah paket software untuk satu repositori Ubuntu bisa lebih dari 50.000 paket (lihat contoh [16.04](#)). Di Ubuntu, pihak yang membuat repositori Ubuntu adalah para developer Ubuntu. Distribusi GNU/Linux yang lain memiliki repositorinya sendiri-sendiri. Pengguna akhir cukup menginstal paket software dari satu repositori distribusinya sendiri, tidak mengunduh manual dari satu per satu situs. Setiap developer distribusi GNU/Linux

(termasuk Ubuntu) menyediakan repositori sebagai layanan untuk para penggunanya.

Menyediakan suatu repositori adalah *pengorbanan*. Setiap developer melakukan pengambilan upstream-upstream, mengolahnya, lalu meletakkannya di repositori. Penyediaan ini adalah proses yang sangat berat. Sebagian distribusi memberikan layanan ini berbayar (seperti Red Hat Enterprise Linux) dan sebagian gratis (seperti Ubuntu). Paket-paket di sebuah repositori untuk satu rilis saja bisa berukuran puluhan GB. Contohnya, total ukuran untuk repositori Ubuntu 10.10 Maverick Meerkat tahun 2010 adalah 30 GB. Lebih-lebih lagi server yang berisi seluruh repositori untuk seluruh rilis Ubuntu bisa berukuran sekitar 1 TB pada saat ini. Bacalah artikel [Perkenalan Sistem Manajemen Paket](#) untuk hal ini.

Definisi Repositori

Repositori (Inggris: repository) adalah direktori tempat paket-paket software disimpan. Pengguna akhir mengambil paket-paket software dari repositori. Repositori milik suatu distribusi GNU/Linux disimpan di dalam suatu server yang tersambung ke jaringan internet. Pengguna mengakses repositori (instal & pemutakhiran paket software) melalui jaringan internet. Repositori adalah layanan dari pihak developer distribusi GNU/Linux kepada pengguna akhir.

Definisi Mirror

Mirror (Indonesia: cermin) adalah istilah di dunia internet untuk menyebut server yang merupakan salinan persis dari server lain. Di dalam konteks server repositori, mirror dari suatu server repositori berarti server salinan yang sama persis isinya dengan server repositori lain. Mirror repositori adalah layanan dari komunitas atau pihak tertentu terhadap pengguna akhir.

Di komunitas GNU/Linux, server repositori dari suatu distribusi bisa memiliki mirror-mirror yang tersebar di seluruh dunia. Umumnya, pihak yang menyediakan mirror adalah universitas, perusahaan tertentu, atau pemerintahan yang meletakkan mirror itu pada server-server mereka. Tujuan penyediaan mirror di suatu negara umumnya untuk memperpendek jarak transfer jaringan antara pengguna akhir dengan server repositori. Akhirnya pengguna akhir bisa mengambil paket software dari mirror yang tersedia paling dekat di negaranya sendiri, tidak perlu mengambil dari server di luar negeri. Artinya, pengguna akhir bisa memilih mirror repositori mana pun yang diinginkan selama itu tersedia karena seluruh server mirror isinya sama persis satu sama lain.

Contoh Mirror Repositori Ubuntu

Repositori Ubuntu yang resmi dari pengembang bisa dilihat di situsnya yaitu <http://archive.ubuntu.com>. Semua mirror repositori di seluruh dunia menyalin repositori resmi ini. Sedangkan repositori Ubuntu yang merupakan mirror ada banyak sekali. Contohnya salah satu

mirror repositori Ubuntu di Indonesia adalah <http://kambing.ui.ac.id/ubuntu> dari Universitas Indonesia. Salah satu mirror repositori Ubuntu di negara Jepang misalnya <http://ftp.tsukuba.wide.ad.jp/Linux/ubuntu/> dari Universitas Tsukuba. Pembaca dapat memerhatikan lebih dekat bahwa struktur direktori repositori Ubuntu di dalam *kambing* atau *tsukuba* adalah sama dengan di dalam *archive*. Karena itulah mereka dinamakan *mirror*.

- Baca daftar mirror repositori seluruh dunia: <https://launchpad.net/ubuntu/+archivemirrors>
- Untuk sysadmin: baca panduan pembuatan mirror: <https://wiki.ubuntu.com/Mirrors>

Klasifikasi Repositori Ubuntu

(1) Dibagi berdasarkan komponennya, maka repositori Ubuntu terbagi menjadi empat jenis untuk setiap rilisnya:

- **main**: dikelola resmi oleh developer Ubuntu, free software.
- **restricted**: dikelola resmi oleh developer Ubuntu, nonfree software.
- **universe**: dikelola oleh selain developer Ubuntu, free software.
- **multiverse**: dikelola oleh selain developer Ubuntu, nonfree software.

Keterangan: yang disebut free software dan nonfree software adalah klasifikasi software yang ditetapkan mencocoki atau tidak mencocoki dokumen *Licensing* Ubuntu <http://www.ubuntu.com/about/about-ubuntu/licensing>. Prinsip free software yang dimaksud oleh Ubuntu adalah *free as in freedom* (bukan free = gratis) yang dijelaskan di <http://www.ubuntu.com/about/about-ubuntu/our-philosophy>.

(2) Dibagi berdasarkan wujud paketnya:

- **deb**: paket DEB biner.
- **deb-src**: paket DEB source code.

Keterangan: seluruh pengguna secara umum hanya membutuhkan repositori paket biner. Repositori paket source code biasanya dibutuhkan oleh developer. Repositori yang biasanya diaktifkan oleh pengguna akhir adalah deb, sedangkan deb-src biasanya dinonaktifkan.

(3) Dibagi berdasarkan pasca-rilis (post-release) yang disediakan untuk setiap rilis, maka repositori pasca-rilis Ubuntu terbagi menjadi empat jenis:

- **\$release-security**: pemutakhiran keamanan.
- **\$release-updates**: pemutakhiran tambalan bug yang tidak berhubungan dengan keamanan.
- **\$release-proposed**: pemutakhiran tambalan bug khusus untuk keperluan *testing* yang nantinya akan masuk ke \$release-updates.

- **\$release-backports**: pemutakhiran paket yang dibuat dari rilis setelahnya untuk rilis sebelumnya.

Berkas Konfigurasi Lokal

Berkas konfigurasi repositori yang ada di dalam setiap sistem Ubuntu terinstal terletak di

- **/etc/apt/sources.list**: berkas yang berisi daftar sumber repositori resmi.
- **/etc/apt/sources.list.d/**: direktori yang berisi berkas-berkas .list berisi daftar sumber repositori pihak ketiga.

Contoh Isi sources.list

(1) Contoh isi sebuah berkas sources.list berdasarkan penjelasan yang telah berlalu adalah sebagai berikut:

```
deb http://id.archive.ubuntu.com/ubuntu/ wily main restricted
universe multiverse
deb http://id.archive.ubuntu.com/ubuntu/ wily-security main
deb http://id.archive.ubuntu.com/ubuntu/ wily-updates main
deb http://id.archive.ubuntu.com/ubuntu/ wily-backports main
deb http://id.archive.ubuntu.com/ubuntu/ wily-proposed main
```

Keterangan: contoh di atas berasal dari Ubuntu 15.10 (Wily Werewolf). Karena itu, nama \$release di situ diganti **wily** (dari Wily Werewolf) dan \$release-updates diganti **wily-updates**. Pada baris pertama, empat komponen repositori semua diaktifkan. Pada tiga baris terakhir, semua repositori pasca-rilis diaktifkan. Untuk mengonfigurasi sources.list, pengguna harus mengetahui nama rilis dari rilis Ubuntu yang digunakannya. Baca daftar nama-nama rilis di <https://wiki.ubuntu.com/Releases>.

(2) Contoh isi sources.list yang umum digunakan lebih pendek seperti ini:

```
deb http://id.archive.ubuntu.com/ubuntu/ wily main restricted
universe multiverse
deb http://id.archive.ubuntu.com/ubuntu/ wily-security main
deb http://id.archive.ubuntu.com/ubuntu/ wily-updates main
```


Keterangan: baris untuk repositori \$release-backports dan \$release-proposed dihilangkan. Konsekuensinya adalah perintah apt-get update akan mengunduh lebih sedikit berkas indeks dan sistem tidak memperoleh paket dari backports maupun proposed. Tiga baris contoh di atas adalah yang paling umum dan wajar digunakan pengguna akhir.

Metode Mengonfigurasi Pemakaian Repositori

Di komputer lokal, untuk bisa mempergunakan layanan repositori Ubuntu, pengguna harus mengatur konfigurasi repositorinya. Pengaturannya tersedia baik secara CLI maupun GUI, secara manual atau otomatis. Adapun metode secara umum langkah-langkahnya adalah sebagai berikut.

1. Sunting sources.list: tetapkan sumber repositori apa saja yang hendak dituju oleh Ubuntu.
2. Lakukan *reloading*: perintahkan **sudo apt-get update**. Perintah *reloading* ini tujuannya untuk mengambil daftar isi dari repositori yang dituju. Tanpa daftar isi, maka Ubuntu tidak mengetahui di mana harus mengambil paket. Reloading cukup dilakukan satu kali setiap pengguna mengubah sources.list. Umumnya kebutuhan data setiap satu *reload* adalah sekitar 15 MB.
3. Lakukan manajemen paket: silakan instal paket atau update Ubuntu sesuka Anda.

Catatan: reloading secara CLI dilakukan dengan apt-get update, tetapi secara GUI bisa dilakukan otomatis dengan klik tombol **Reload** di Synaptic.

Distribusi Repositori

Repositori dari suatu GNU/Linux seperti Ubuntu, didistribusikan setidaknya dengan dua jalur. Jalur pertama seperti dijelaskan di atas yakni dengan menggunakan jaringan internet. Jalur kedua adalah dengan media fisik yang portabel. Contoh jalur kedua yang paling umum adalah DVD repositori. Karena repositori bisa disalin ke sesama server, maka bisa pula disalin ke DVD atau media lain. DVD repositori atau mungkin HDD repositori sangat mempermudah pengguna akhir yang kesulitan mengakses internet supaya tetap bisa menginstal paket software. Pengguna bisa memperolehnya dari toko online seperti [Juragan Kambing](#).

Referensi

- <https://fajran.web.id/2013/05/ubuntu-repository-total-packages-and-sizes.html>
- <https://wiki.ubuntu.com/Mirrors>
- <http://bethesignal.org/blog/2008/03/31/understanding-the-ubuntu-package-repositories/>
- <https://help.ubuntu.com/community/Repositories/Ubuntu>
- <https://help.ubuntu.com/community/Repositories>

Bab 3: Mengenal Dpkg

Tulisan ini memperkenalkan mengenai dpkg, perintah-perintahnya, dan bagaimana secara umum mengelola paket program di keluarga sistem operasi Debian (khususnya Ubuntu) dengannya. Panduan ini bisa diterapkan di Linux Mint, BlankOn, gNewsense, dan distro keluarga Debian lainnya. Diharapkan dengan ini pembaca memperoleh gambaran dasar sekaligus mampu melakukan manajemen paket dasar (install & remove) dengan dpkg. Aturan baca tulisan ini adalah perintah bertanda # dilakukan sebagai root sedangkan \$ sebagai user biasa. Tulisan ini lanjutan dari [Seri Manajemen Paket 2: Repositori](#). Semoga tulisan ini bermanfaat.

Ringkasan

1. Tentang Dpkg
2. Melihat Daftar Paket
3. Instal Paket
4. Hapus Paket
5. Contoh Penggunaan
6. Singkatan-Singkatan

Tentang Dpkg

Dpkg (nama program: dpkg) adalah program manajer paket (package manager) yang berasal dari sistem operasi Debian. Dpkg asalnya dibuat dalam bahasa Perl oleh Ian Murdock (bapak Debian) beserta rekan-rekannya. Di sebuah distribusi GNU/Linux, dpkg yang bertugas menginstal paket sementara apt yang bertugas mencari depedensi. Format paket yang dikenal oleh dpkg adalah .deb. Dpkg digunakan secara internal, yakni hanya di dalam sistem operasi dan tidak berhubungan dengan repositori di luarnya. Berbeda dengan apt yang bekerja secara eksternal, berhubungan dengan repositori di luar sistem. Tulisan ini membahas tentang dpkg.

Melihat Daftar Paket

Dpkg bisa menayangkan seluruh paket terinstal di dalam sistem dengan perintah:

```
$ dpkg --list
```

Instal Paket

Dpkg tugas utamanya adalah menginstal paket dan perintahnya adalah:

```
# dpkg --install <nama_paket>
```

Hapus Paket

Dpkg selain menginstal, harus bisa menghapus paket yang sudah terinstal. Perintahnya adalah:

```
# dpkg --remove <nama_paket>
```

Contoh Penggunaan

Berikut ini contoh penggunaan paling sering dilakukan atau paling esensial sehari-hari bagi pengguna akhir. Disebutkan contoh langsung agar pembaca mudah memahami.

1. Instal Paket

Intisari tugas dpkg adalah menginstal paket satu per satu. Misalnya pembaca ingin menginstal program fping yang paket itu bisa diperoleh dari packages.ubuntu.com, maka perintah untuk menginstal paket fping tersebut ialah

```
# dpkg --install fping_3.13-1_i386.deb
```

Contoh output:

```
master@master:~$ sudo dpkg --install fping_3.13-1_i386.deb
[sudo] password for master:
Selecting previously unselected package fping.
(Reading database ... 265493 files and directories currently installed.)
Preparing to unpack fping_3.13-1_i386.deb ...
Unpacking fping (3.13-1) ...
Setting up fping (3.13-1) ...
Processing triggers for man-db (2.7.5-1) ...
master@master:~$
```

Keterangan: pembaca bisa mengunduh paket fping sesuai versi Ubuntu yang digunakan dari alamat <http://packages.ubuntu.com/search?>

[keywords=fping&searchon=names&suite=all§ion=all](#). Jangan menginstal paket dari versi Ubuntu berbeda.

2. Hapus Paket

Setelah menginstal, pengguna mungkin ingin menghapus paket terinstal. Contoh di atas paket fping sudah diinstal maka bisa dihapus. Perintahnya seperti ini:

```
# dpkg --remove fping_3.13-1_i386.deb
```

Contoh output:

```
master@master:~$ sudo dpkg -r fping
(Reading database ... 265502 files and directories currently installed.)
Removing fping (3.13-1) ...
Processing triggers for man-db (2.7.5-1) ...
master@master:~$
```

3. Instal Banyak Paket Bersamaan

Dpkg mampu menginstal banyak paket bersamaan di dalam sebuah direktori. Artinya, pengguna tidak perlu mengetik satu per satu nama paket yang mau diinstal (bayangkan jika ada 1000 nama paket) melainkan cukup ketik 1 karakter seperti ini:

```
# dpkg -i *.deb
```

Keterangan: perintah di atas maksudnya menyuruh dpkg untuk menginstal seluruh paket .deb di satu direktori yang sama. Selalu pastikan Anda tahu di mana direktori Anda saat ini dengan perintah **pwd**.

4. Memeriksa Terinstal Tidaknya Suatu Paket

Opsi --list bisa digunakan di sini digabung dengan grep seperti:

```
dpkg --get-selections | grep -i fping
```

Contoh output:

```
master@master:~$ dpkg --get-selections | grep -i fping
```

```

ii fping                      3.13-1                      i386      sends ICMP ECHO_REQUEST packets to
network hosts
master@master:~$

```

Keterangan: baris perintah di atas menyambungkan perintah pertama kepada perintah kedua. Perintah pertama meminta seluruh daftar paket terinstal di dalam sistem, di tengah-tengahnya ada pipeline (|) penyambung perintah, perintah kedua menyaring output perintah pertama supaya hanya menampilkan baris yang berisi nama "fping". Opsi -i membuat grep tidak melihat huruf kecil/besar. Output di atas memberikan informasi bahwa paket bernama fping terinstal (ii) dengan versi 3.13-1. Aktivitas perintah ini sama persis dengan tampilan daftar paket di Synaptic Package Manager dan semisalnya.

5. Memeriksa Versi Paket Terinstal

Sama dengan poin 4 di atas.

```
dpkg --get-architecture | grep -i fping
```

Singkatan-Singkatan

Opsi pada perintah dpkg memiliki singkatan-singkatan sebagai berikut:

- --install bisa ditulis -i
- --list bisa ditulis -l
- --remove bisa ditulis -r

Contohnya perintah **dpkg --install paket.deb** bisa ditulis **dpkg -i paket.deb**.

Kamus Istilah

- **Paket:** package; istilah untuk menyebut berkas yang memuat suatu software dalam bentuk binary atau source code. Di GNU/Linux, tidak semua paket berisi software tetapi bisa juga berisi berkas lain seperti wallpaper (disebut paket *wallpaper*), icons (paket *icon*), dokumentasi (paket *docs*), dan lain-lain. Masing-masing bisa diinstal ke sistem. Format paket di keluarga sistem operasi Debian adalah .deb, sedangkan di Windows .msi. Umumnya sebuah distribusi GNU/Linux memiliki 10000 paket lebih di dalam repositorinya.
- **Manajer paket:** package manager; software yang bertugas me-manage paket di dalam sistem. Di keluarga sistem operasi Debian, manajer pakatnya adalah dpkg. Di sistem operasi Arch Linux, pacman. Di Fedora, rpm.

- **Sistem manajemen paket:** package management system; sistem yang mengelola input, process, dan output dari suatu paket di dalam sistem. Nama sistem manajemen paket lazimnya digunakan di GNU/Linux, sedangkan di Windows lazim dinamakan *add/remove programs*.
- **root:** pengguna dengan kuasa paling tinggi di sistem operasi keluarga UNIX. Cara untuk berubah menjadi root adalah dengan perintah *su*. Lebih disarankan menggunakan perintah *sudo* supaya tidak berubah permanen.
- **user biasa:** user yang selain root. Kuasa user biasa terbatas pada direktorinya sendiri, tidak kepada seluruh sistem.
- **Remove:** hapus, uninstall; istilah untuk kegiatan mencabut software dari komputer.

Bab 4: Mengenal Apt

Tulisan ini menjelaskan penggunaan APT (Advanced Package Tool) untuk instal, hapus, dan upgrade paket software di Ubuntu. Dengan tulisan ini, Anda belajar bagaimana perintah-perintah **apt-get** dijalankan dan apa fungsi masing-masing. Tulisan ini dirancang sesederhana mungkin untuk pemula. Pada bagian akhir, tulisan ini dilengkapi kamus istilah dan referensi lanjutan. Tulisan ini adalah bagian ke-4 dari serial manajemen paket Ubuntu. Semoga bermanfaat.

Serial Manajemen Paket Ubuntu

Berikut serial panduan manajemen paket dari kami.

- Seri 1 [Perkenalan, Dasar-Dasar, Source Code & Binary](#)
- Seri 2 [Repositori, Asal Muasal, Contoh](#)
- Seri 3 [Dpkg, List, Install, Remove](#)

Ringkasan

1. Perkenalan
2. Program-Program APT
3. Reload Repositori
4. Instal Paket
5. Hapus Paket
6. Upgrade Seluruh Paket
7. Upgrade Versi Rilis Ubuntu
8. Hapus APT Cache
9. Kamus Istilah
10. Memperoleh Bantuan

Perkenalan

APT (Advanced Package Tool) adalah sistem yang sangat canggih untuk mengelola paket di sistem operasi keluarga Debian. APT terdiri dari sejumlah program (bisa dibaca di bawah) yang punya tugas sendiri-sendiri untuk mengelola. Tugas utama APT adalah menemukan dependensi, yang diserahkan kepada program apt-get. Sedangkan yang lainnya adalah program-program yang

menunjang penemuan dependensi tersebut baik dari segi security (terdapat dua program untuk itu) maupun segi manfaat teknis (seperti apt-cdrom, apt-config). APT adalah free software berlisensi GNU GPL.

- Baca artikel APT di Wikipedia https://en.wikipedia.org/wiki/Advanced_Packaging_Tool
- Kunjungi situs resmi APT <https://wiki.debian.org/Apt>
- Kunjungi laman developer APT <https://wiki.debian.org/Teams/Apt>
- Source code dari APT https://alioth.debian.org/scm/browser.php?group_id=31152

Program-Program APT

Advanced Package Tool memiliki sejumlah perintah yang ditugaskan untuk mengelola paket di suatu sistem Debian atau keturunan Debian (seperti Ubuntu). Masing-masing perintah merupakan program tersendiri. Di antara mereka adalah:

- apt-get
- apt-cache
- apt-cdrom
- apt-config
- apt-key
- apt-mark
- apt-secure
- apt-sortpkgs

Di Ubuntu (atau Debian) versi terbaru, misalnya 16.04 Xenial Xerus (atau 8 Jessie), APT memiliki perintah **apt** tersendiri yang bisa melakukan manajemen paket dan konfigurasi. Adapun program yang paling umum digunakan pengguna akhir adalah **apt-get**. Tulisan ini berbicara mengenai apt-get.

Reload Repositori

```
# apt-get update
```

Contoh output:


```
Terminal File Edit View Search Terminal Tabs Help
master@master: ~ x master@master: ~ x +
master@master:~$ sudo apt-get update
Hit:1 http://security.ubuntu.com/ubuntu xenial-security InRelease
Hit:2 http://archive.canonical.com/ubuntu xenial InRelease
Hit:4 http://id.archive.ubuntu.com/ubuntu xenial-updates InRelease
Hit:5 http://id.archive.ubuntu.com/ubuntu xenial-backports InRelease
Get:3 http://id.archive.ubuntu.com/ubuntu xenial InRelease [247 kB]
Get:6 http://id.archive.ubuntu.com/ubuntu xenial/main i386 Packages [1,196 kB]
Get:7 http://id.archive.ubuntu.com/ubuntu xenial/main Translation-en [568 kB]
Get:8 http://id.archive.ubuntu.com/ubuntu xenial/main i386 DEP-11 Metadata [733
kB]
Get:9 http://id.archive.ubuntu.com/ubuntu xenial/main DEP-11 64x64 Icons [409 kB
]
Get:10 http://id.archive.ubuntu.com/ubuntu xenial/universe i386 Packages [7,514
kB]
Get:11 http://id.archive.ubuntu.com/ubuntu xenial/universe Translation-en [4,355
kB]
Get:12 http://id.archive.ubuntu.com/ubuntu xenial/universe i386 DEP-11 Metadata
[3,407 kB]
Get:13 http://id.archive.ubuntu.com/ubuntu xenial/universe DEP-11 64x64 Icons [7
,448 kB]
Fetched 22.0 MB in 3min 49s (96.1 kB/s)
Reading package lists... Done
master@master:~$
```

Penjelasan:

Perintah ini berfungsi untuk me-reload atau mengunduh daftar isi dari repositori yang dipilih oleh sistem. Perintah ini tidak akan meng-upgrade atau menginstal paket apa pun. Jika berkas **/etc/apt/sources.list** telah disetel sesuai kebutuhan Anda, maka perintah ini adalah wajib dijalankan satu kali. Perintah ini sesungguhnya akan mengunduh daftar isi berupa berkas-berkas teks yang banyak (lihat output di atas dari baris ke baris) ke dalam direktori **/var/lib/apt/lists/** dari repositori yang ditunjuk oleh sources.list. Satu kali saja sistem Anda sudah memiliki daftar isi, maka perintah ini sudah tidak dibutuhkan lagi kecuali Anda mengubah setelah sources.list. Perintah apt-get install baru bisa dilakukan jika sistem Anda sudah memiliki daftar isi dari repositori yang disetel dalam sources.list.

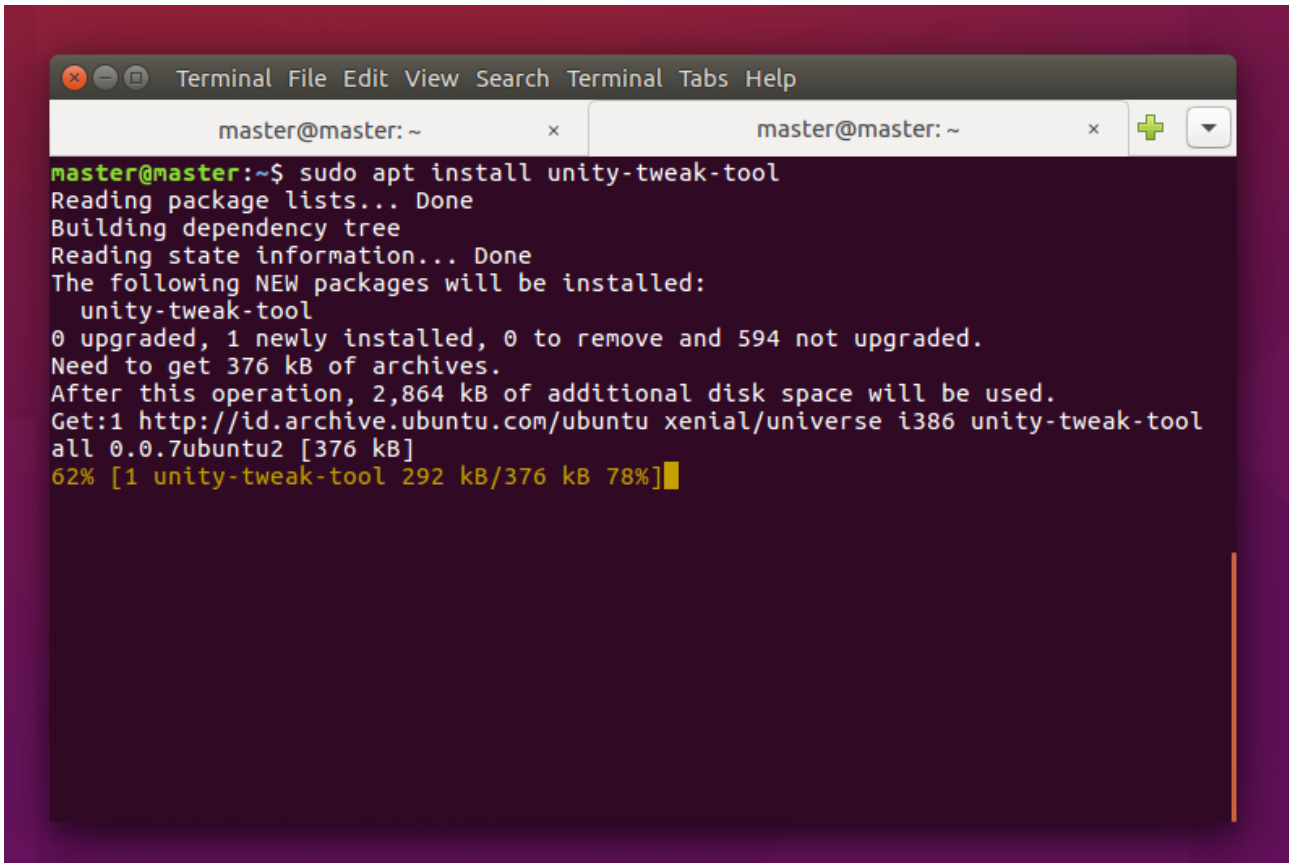
Catatan:

- Ukuran unduhan seluruh paket selalu ditayangkan pada saat instalasi akan dilaksanakan. Ukuran paket bergantung pada paket-paket yang Anda minta.
- Selalu periksa berapa ukuran paket yang akan diunduh terutama jika Anda menggunakan koneksi jaringan yang terbatas.
- Biasanya paket-paket program GUI semacam KDE atau GNOME akan berukuran besar, sedangkan paket-paket program CLI semacam nmap atau fping akan berukuran kecil.

Instal Paket

```
# apt-get install <nama_paket>
```

Contoh output:

A screenshot of a terminal window with a dark background and light-colored text. The window title bar shows 'Terminal' and several menu options: File, Edit, View, Search, Terminal, Tabs, Help. There are two tabs open, both labeled 'master@master: ~'. The terminal output shows the command 'sudo apt install unity-tweak-tool' being executed. The output includes status messages like 'Reading package lists... Done', 'Building dependency tree', and 'Reading state information... Done'. It lists the packages to be installed, disk space requirements, and the progress of downloading the package. The progress bar shows 62% completion for the 292 kB of the 376 kB package.

```
master@master:~$ sudo apt install unity-tweak-tool
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  unity-tweak-tool
0 upgraded, 1 newly installed, 0 to remove and 594 not upgraded.
Need to get 376 kB of archives.
After this operation, 2,864 kB of additional disk space will be used.
Get:1 http://id.archive.ubuntu.com/ubuntu xenial/universe i386 unity-tweak-tool
all 0.0.7ubuntu2 [376 kB]
62% [1 unity-tweak-tool 292 kB/376 kB 78%]
```

Penjelasan:

Perintah apt-get install bekerja dengan cara membaca alamat repositori yang sudah tersimpan di dalam sistem. Apt-get akan membaca nama paket yang Anda minta, mencocokkannya dengan database dpkg (/var/lib/dpkg/status) untuk mencari dependensi dari paket tersebut, lalu mencocokkannya dengan database APT di dalam sistem Anda, lalu berdasarkan daftar isi repositori yang sudah dimiliki sistem Anda, apt-get akan mengunduh seluruh paket beserta dependensinya. Di bagian akhir kerja, apt-get akan memanggil dpkg untuk melakukan instalasi paket-paket yang diperoleh satu per satu. Jika dpkg selesai kerja dengan sukses, apt-get akan mengembalikan command line Anda ke prompt. Demikian secara singkat bagaimana apt-get install bekerja.

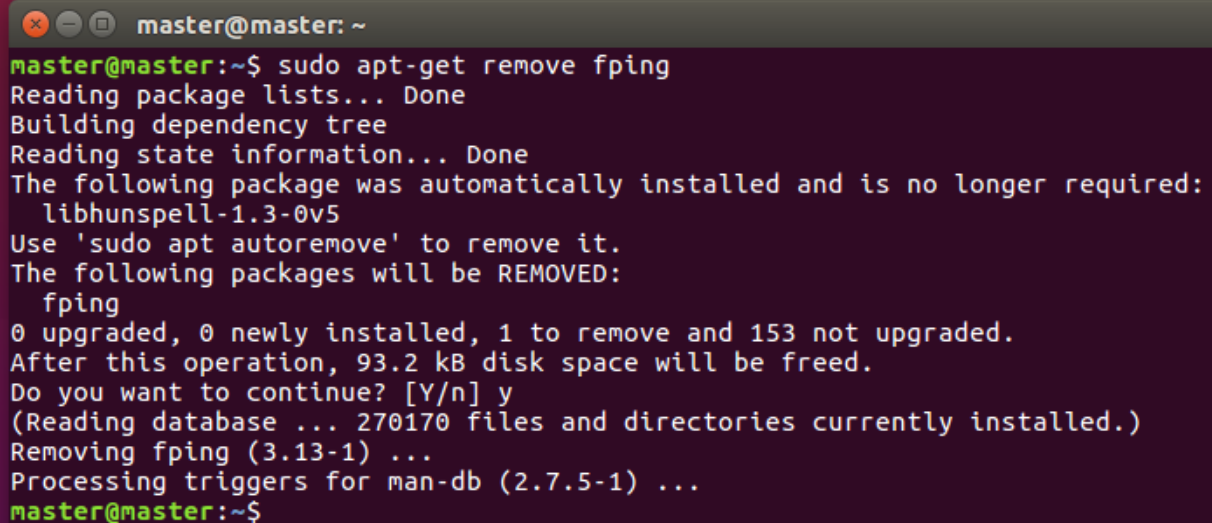
Catatan:

- apt-get tidak menginstal paket, apt-get hanya mencari dependensi, sedangkan dpkg-lah yang menginstal.
- Oleh karena apt-get bekerja dengan daftar isi repositori, perintah apt-get update menjadi penting.
- Paket-paket yang diunduh oleh apt-get disimpan secara campur di satu direktori `/var/cache/apt/archives/`, sedangkan paket yang tidak terunduh sebagian (mis. ketika jaringan putus) disimpan di `/var/cache/apt/archives/partial/`. Direktori ini disebut dengan nama APT cache.

Hapus Paket

```
# apt-get remove <nama_paket>
```

Contoh output:



```

master@master: ~
master@master:~$ sudo apt-get remove fping
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libhunspell-1.3-0v5
Use 'sudo apt autoremove' to remove it.
The following packages will be REMOVED:
  fping
0 upgraded, 0 newly installed, 1 to remove and 153 not upgraded.
After this operation, 93.2 kB disk space will be freed.
Do you want to continue? [Y/n] y
(Reading database ... 270170 files and directories currently installed.)
Removing fping (3.13-1) ...
Processing triggers for man-db (2.7.5-1) ...
master@master:~$
  
```

Penjelasan:

Perintah apt-get remove bertugas membaca permintaan paket Anda, mencocokkannya dengan database, lalu akhirnya apt-get memanggil dpkg untuk menghapus tepat paket yang Anda minta

saja. Perintah `apt-get remove` tidak menghapus dependensi yang datang bersama diinstalnya paket yang Anda minta.

Catatan:

- Meski demikian, kadang `apt-get` bisa memberitahu akan menghapus banyak paket bersamaan jika paket yang Anda minta untuk dihapus adalah paket yang dibutuhkan oleh paket-paket yang lainnya.
- Jika itu yang terjadi, selalu baca dengan hati-hati satu per satu nama paket yang ditampilkan oleh `apt-get` sebelum Anda benar-benar menyetujui penghapusannya. Karena bisa saja Anda menghapus seluruh GUI di Ubuntu Anda sendiri tanpa disadari.

Upgrade Seluruh Paket

```
# apt-get update
# apt-get upgrade
# apt-get dist-upgrade
```

Penjelasan:

Upgrade di sini adalah upgrade terhadap seluruh software yang terinstal secara otomatis ke versi terbarunya masing-masing. Bayangkan Anda punya Firefox, LibreOffice, Thunderbird, Nautilus, dan lain-lain versi lawas. Dengan upgrade, Anda otomatis memperoleh versi baru dari semua software tanpa perlu upgrade satu-satu. Upgrade di sini adalah upgrade versi seluruh paket terinstal. Itulah yang dimaksud dengan upgrade di sini.

Perintah pertama memastikan sistem Anda memiliki seluruh daftar isi dari repositori yang disetel di `source.list`, kemudian perintah kedua melakukan upgrading terhadap tiap-tiap paket yang terinstal ke versi paket-paket terbaru yang tersedia di repositori, kemudian perintah ketiga akan melakukan clean up (pembersihan) otomatis jika terdapat sejumlah error atau kekurangan tertentu. Perintah-perintah ini **tidak** meng-upgrade versi Ubuntu Anda menjadi versi baru (mis. 15.10 ke 16.04) tetapi hanya mengupgrade seluruh paket-paket terinstal ke paket-paket versi baru yang tersedia di repositori (mis. Firefox 40 ke Firefox 41).

Catatan:

- Jangan meninggalkan salah satu dari ketiga perintah di atas. Lakukan semuanya jika Anda memang ingin upgrade paket-paket.
- Siapkan jaringan internet yang besar dan stabil. Pada Ubuntu desktop, sering terjadi sekali upgrade membutuhkan data kurang lebih 400 MB. Kestabilan koneksi internet juga harus.

- Pastikan listrik tidak mati ketika upgrade berlangsung.
- Hati-hati terhadap kasus-kasus gagal upgrade yang berakhir dengan tidak bisa login atau rusak sistem. Selalu baca informasi dari sumber resmi dan komunitas sebelum melakukan upgrade.

Upgrade Versi Rilis Ubuntu

```
# do-release-upgrade
```

Penjelasan:

Perintah ini **bukan** bagian dari APT melainkan bagian dari paket [ubuntu-release-upgrader-core](#) tambahan dari Canonical. Pembuat perintah ini adalah Michael Vogt dan lain-lain. Perintah ini akan membaca kesediaan versi Ubuntu Anda apakah bisa melakukan upgrade versi atau tidak, lalu akan melakukan sejumlah aksi upgrade ke versi Ubuntu terbaru tersebut.

Catatan:

Upgrade versi Ubuntu membutuhkan koneksi jaringan yang besar dan stabil. Jika Anda sudah siap mengeluarkan beberapa ratus MB data dan tidak mematikan komputer selama upgrading berlangsung, hal ini boleh Anda lakukan. Namun perlu diingat bahwa terdapat kasus-kasus tertentu yang membuat sistem menjadi rusak (broken) setelah upgrading. Hendaknya Anda berhati-hati dan selalu mencari informasi sebelum melakukannya.

Hapus APT Cache

```
# apt-get clean
```

Penjelasan:

Perintah ini menghapus semua berkas paket .deb di dalam **/var/cache/apt/archives** dan **/apt/cache/apt/archives/partial** (direktori APT cache). Pada sistem yang ruang partisinya sempit, umumnya perintah ini berguna untuk membebaskan banyak ruang partisi (karena semakin banyak .deb di dalam APT cache = semakin banyak ruang termakan).

Catatan:

Bagi pengguna yang menomorsatukan instalasi paket secara offline (mis. pengguna yang koneksi jaringannya terbatas), perintah ini umumnya tidak dijalankan atau jika dijalankan maka seluruh paket harus di-backup dahulu. Sekali perintah dijalankan, semua paket terhapus tidak bisa dikembalikan lagi.

Kamus Istilah

- **Update:** mutakhir; istilah ini di dalam konteks apt berarti pengambilan daftar isi dari repositori yang disetel dalam sources.list. Istilah ini tidak mengandung makna instalasi paket sama sekali.
- **Upgrade:** naik tingkat; istilah ini di dalam konteks apt berarti penginstalan paket versi baru untuk menggantikan paket yang sama versi lawas. Di dalam beberapa jenis software, istilah upgrade diterjemahkan sebagai *pemutakhiran*.
- **Reload:** muat ulang; istilah ini digunakan di GUI front-end seperti Synaptic untuk menyebut makna yang sama dengan apt-get update. Reload itu sama dengan fungsi apt-get update.
- **Paket:** package; berkas yang memuat software dalam bentuk binary atau source code.
- **.deb:** format paket yang memuat software binary di Ubuntu.
- **Database:** basis data; dalam konteks manajemen paket yang dimaksud adalah basis data dpkg (utamanya /var/lib/dpkg/status) dan basis data apt (utamanya /var/lib/apt/lists/).
- **Automatically Installed:** ketika instalasi paket selesai dilakukan (mis. # apt-get install scribus), paket yang diminta disebut *manually installed* dan dependensi yang diinstal bersamanya disebut *automatically installed*. Istilah ini merupakan istilah *penandaan* (marking). Istilah ini penting di dalam pemakaian dpkg atau apt-mark tingkat lanjut.
- **Manually Installed:** telah jelas.

Memperoleh Bantuan

(1) Sistem Ubuntu Anda telah memuat bantuan-bantuan dalam bentuk dokumentasi manual yang cukup. Di antaranya adalah:

- Perintah `man apt`
- Perintah `man apt-get`
- Perintah `man apt-cache`
- Perintah `man apt-<ganti_sendiri_namanya>`
- Direktori `/usr/share/doc/apt/`
- Direktori `/usr/share/doc/apt/examples/`

(2) Jika tidak bisa mengakses bantuan internal, Anda bisa mengakses bantuan eksternal di internet seperti

- <https://wiki.debian.org/apt-get>
- <http://manpages.ubuntu.com/manpages/wily/man8/apt-get.8.html>
- <https://help.ubuntu.com/community/AptGet/Howto>

Bab 5: sources.list

Tulisan ini menjelaskan mengenai berkas konfigurasi `sources.list` yang merupakan fondasi bagi APT untuk bekerja. Tulisan ini dibagi menjadi dua bagian, pertama penjelasan `sources.list` itu sendiri dan kedua bagaimana contoh konfigurasi di dalam `sources.list`. Tulisan ini dilengkapi kamus istilah dan referensi lanjutan di bagian akhir. Tulisan ini adalah seri ke-5 dari serial manajemen paket Ubuntu. Semoga tulisan ini bermanfaat.

Serial Manajemen Paket Ubuntu

Berikut serial panduan manajemen paket dari kami.

- Seri 1 [Perkenalan, Dasar-Dasar, Source Code & Binary](#)
- Seri 2 [Repositori, Asal Muasal, Contoh](#)
- Seri 3 [Dpkg, List, Install, Remove](#)

Perkenalan

`sources.list` adalah sebuah berkas teks biasa yang memuat daftar sumber-sumber repositori yang APT mengambil paket darinya. Letak `sources.list` di dalam sistem ada di dua tempat. Tempat pertama tempat yang paling umum adalah **`/etc/apt/sources.list`**. Tempat kedua adalah tempat yang kurang umum, biasanya digunakan untuk sumber-sumber repositori pihak ketiga (repositori yang diasuh pihak di luar Ubuntu), yang posisinya ada di **`/etc/apt/sources.list.d/*.list`**.

Isi dari sebuah `sources.list` adalah baris-baris setelan sumber repositori. Setiap baris umumnya mengarah kepada satu repositori. Isi `sources.list` tidak dibatasi berapa pun jumlah barisnya.

Catatan:

Di keluarga UNIX, berkas teks tidak harus memiliki nama akhir `.txt` seperti di Windows. Berkas teks walaupun diberi nama akhir `.list` atau apa saja akan selalu dikenali sebagai berkas teks karena UNIX mampu membaca berkas tanpa melihat namanya. Itulah kenapa `sources.list` bernama akhir `.list` bukan `.txt`.

Klasifikasi Repositori Ubuntu

Distribusi GNU/Linux Ubuntu membagi repositorinya di setiap rilis berdasarkan 2 kelas besar, dengan masing-masing kelas memiliki pengkategorian sendiri-sendiri. Klasifikasi inilah yang

membedakan Ubuntu dari Debian.

Berdasarkan **komponennya**:

1. main
2. restricted
3. universe
4. multiverse

Berdasarkan **layanannya**:

1. <nama_rilis> contoh: xenial
2. <nama_rilis>-updates contoh: xenial-updates
3. <nama_rilis>-security contoh: xenial-security
4. <nama_rilis>-proposed contoh: xenial-proposed
5. <nama_rilis>-backports contoh: xenial-backports

Klasifikasi Repositori Debian

Debian sebagai induk Ubuntu memiliki klasifikasi repositori tersendiri pula. Klasifikasinya dibagi menjadi 3 kelas besar, setiap kelas dengan pengkategorian sendiri-sendiri. Data ini diberikan sebatas untuk pembandingan karena perbedaan antara Ubuntu dan Debian sesungguhnya ada di sini.

Berdasarkan **rilisnya**:

1. stable
2. testing
3. unstable

Berdasarkan **komponennya**:

1. main
2. contrib
3. non-free

Berdasarkan **layanannya**:

1. <nama_rilis>contoh: jessie
2. <nama_rilis>-updates contoh: jessie-updates
3. <nama_rilis>/updates contoh: jessie/updates

Format Setelan Sumber Repositori

Format setelan sumber repositori secara umum ada 2 jenis:

```
[deb]      [url] [nama_rilis] [golongan_repositori]
[deb-src]  [url] [nama_rilis] [golongan_repositori]
```

Sedangkan contoh penulisan setelan bagi Ubuntu 16.04 misalnya seperti ini:

```
deb http://kambing.ui.ac.id/ubuntu xenial main
deb http://kambing.ui.ac.id/ubuntu xenial restricted
deb http://kambing.ui.ac.id/ubuntu xenial universe
deb http://kambing.ui.ac.id/ubuntu xenial multiverse

deb http://kambing.ui.ac.id/ubuntu xenial-updates main
deb http://kambing.ui.ac.id/ubuntu xenial-security main
deb http://kambing.ui.ac.id/ubuntu xenial-proposed main
deb http://kambing.ui.ac.id/ubuntu xenial-backports main
```

Penjelasan:

- Kode **deb** dan kode **deb-src** menandakan paket binary dan paket source code berurutan. Bacalah [seri pertama manajemen paket Ubuntu](#) pada bagian *Mengenai Distribusi Software*.
- Kode **url** menandakan alamat URL dari repositori yang hendak dituju. URL ini umumnya adalah alamat URL server di internet, tetapi karena APT sangat fleksibel URL di sini bisa diganti alamat penyimpanan lokal seperti DVD atau HDD. Untuk mengetahui alamat URL ini, Anda harus merujuk kepada dokumentasi dari distribusi GNU/Linux yang Anda gunakan.
- Kode **nama_repositori** menandakan nama repositori dari rilis distribusi GNU/Linux yang hendak dituju. Contohnya, distribusi Ubuntu 16.04 memiliki 4 jenis nama_repositori dimulai dari xenial, xenial-updates, xenial-security, xenial-backports, dan xenial-proposed.
- Kode **golongan_repositori** menandakan komponen repositori yang hendak dituju. Di Ubuntu, ada 4 komponen: main restricted universe multiverse; sedangkan di Debian, ada 3 komponen main contrib non-free.
- Pengguna memiliki kebebasan menyetel repositorinya ke arah repositori mana pun, nama repositori yang mana pun, dan komponen mana pun dari repositori yang hendak dipakai.

Contoh sources.list Ubuntu 16.04

Berikut ini diberikan sejumlah contoh setelan yang siap ditempel di sources.list Anda pengguna

16.04 diklasifikasikan berdasarkan jumlah setelan. Setelan ini diarahkan ke server mirror repositori Kambing milik Universitas Indonesia. Anda bisa memilih mirror-mirror yang lain jika Anda suka. Semua setelan ini berpusat pada binary, tanpa ada yang source code.

Setelan Penuh

Setelan penuh ini menyalakan seluruh nama repositori dan golongan repositori yang ada. Setelan ini cocok untuk developer dan tester Ubuntu. Ini tidak direkomendasikan untuk pengguna akhir.

```
deb http://kambing.ui.ac.id/ubuntu xenial main
deb http://kambing.ui.ac.id/ubuntu xenial restricted
deb http://kambing.ui.ac.id/ubuntu xenial universe
deb http://kambing.ui.ac.id/ubuntu xenial multiverse

deb http://kambing.ui.ac.id/ubuntu xenial-updates main
deb http://kambing.ui.ac.id/ubuntu xenial-security main
deb http://kambing.ui.ac.id/ubuntu xenial-proposed main
deb http://kambing.ui.ac.id/ubuntu xenial-backports main
```

Setelan Umum

Setelan umum ini menyalakan hampir semua nama repositori dan golongan repositori kecuali proposed dan backports. Setelan umum ini cocok untuk pengguna akhir. Catatan: repositori restricted dan multiverse mengandung nonfree software.

```
deb http://kambing.ui.ac.id/ubuntu xenial main
deb http://kambing.ui.ac.id/ubuntu xenial restricted
deb http://kambing.ui.ac.id/ubuntu xenial universe
deb http://kambing.ui.ac.id/ubuntu xenial multiverse

deb http://kambing.ui.ac.id/ubuntu xenial-updates main
deb http://kambing.ui.ac.id/ubuntu xenial-security main
```

Setelan Minimalis

Setelan minimalis ini mematikan nama repositori updates dan security, hanya menyisakan empat komponen dari repositori rilis. Pengguna yang jaringan internetnya terbatas mungkin akan suka setelan ini.

```
deb http://kambing.ui.ac.id/ubuntu xenial main
deb http://kambing.ui.ac.id/ubuntu xenial restricted
```

```
deb http://kambing.ui.ac.id/ubuntu xenial universe
deb http://kambing.ui.ac.id/ubuntu xenial multiverse
```

Setelan Free Software

Setelan Free Software ini ditujukan bagi pengguna yang hanya mau menggunakan free software semata tanpa proprietary. Yang ada di sini hanyalah komponen main dan komponen universe.

```
deb http://kambing.ui.ac.id/ubuntu xenial main
deb http://kambing.ui.ac.id/ubuntu xenial universe
deb http://kambing.ui.ac.id/ubuntu xenial-updates main
deb http://kambing.ui.ac.id/ubuntu xenial-security main
```

Cara Menyunting sources.list

Cara paling mudah menyunting sources.list adalah secara GUI dengan program **software-properties-gtk** atau Software Sources di menu di Unity. Adapun pengguna yang mau menyunting seperti penjelasan di dalam tulisan ini, gunakanlah penyunting teks. Di Ubuntu, penyunting teks yang tersedia adalah **gedit** (GUI), **nano** (CLI), dan **vim** (CLI). Silakan pilih perintah yang Anda suka untuk menyunting:

- `sudo gedit /etc/apt/sources.list`
- `sudo nano /etc/apt/sources.list`
- `sudo vi /etc/apt/sources.list`

Pengguna pemula umumnya lebih suka menyunting dengan gedit karena ia memiliki tampilan GUI yang mudah digunakan.

Yang Tidak Boleh Dilakukan

Terdapat larangan umum yang harus diperhatikan pengguna Ubuntu atau Debian yang harusnya tidak dilakukan. Jika dilanggar maka risiko terburuknya sistem akan rusak dan tidak dapat diperbaiki kecuali instal ulang.

1. Jangan menggunakan sumber repositori yang tidak terpercaya. Selalu gunakan sumber repositori yang berasal dari developer aslinya.
2. Jangan mencampur repositori berbeda versi bersamaan, misalnya di Debian menggunakan repositori experimental, sid, dan testing bersamaan.

Kamus Istilah

Sources: sumber-sumber; di dalam konteks manajemen paket sources maksudnya alamat URL dari repositori yang hendak dituju. APT bekerja dengan sistem alamat URL untuk mengetahui di mana letak paket-paket yang hendak diambil. Jika tidak ada alamat URL (baca: sources) maka APT tidak bisa mengambil apa-apa.

Keluarga UNIX: sistem operasi baik yang merupakan turunan UNIX langsung, atau tersertifikasi UNIX(R) oleh The Open Group, atau UNIX-like. Contoh keluarga UNIX adalah BSD dan GNU/Linux. Windows bukan keluarga UNIX.

xenial: Xenial Xerus; nama rilis untuk Ubuntu 16.04. Sama seperti nama Jessie bagi Debian 8.

Referensi

- <https://help.ubuntu.com/community/Repositories/CommandLine>
- <https://help.ubuntu.com/community/Repositories/Ubuntu>
- <https://help.ubuntu.com/community/Repositories>
- <https://debgen.simplylinux.ch>
- <https://repogen.simplylinux.ch>

Bab 6: GUI Front-End

Tulisan ini memperkenalkan program-program GUI front-end untuk sistem manajemen paket di Ubuntu. Contohnya adalah Synaptic, GNOME PackageKit, GDebi, Ubuntu Software Center, dan GNOME Software. Tulisan ini adalah bagian ke-6 dari serial manajemen paket Ubuntu.

Serial Manajemen Paket Ubuntu

Berikut serial panduan manajemen paket dari kami.

- Seri 1 [Perkenalan, Dasar-Dasar, Source Code & Binary](#)
- Seri 2 [Repositori, Asal Muasal, Contoh](#)
- Seri 3 [Dpkg, List, Install, Remove](#)
- Seri 4 [APT & apt-get, Install, Remove, Upgrade](#)
- Seri 5 [Setelan sources.list & Contoh](#)

Perkenalan Istilah GUI Front-End

Di dunia software, terkhusus di sistem operasi keluarga UNIX, terdapat istilah back-end dan istilah front-end. Kedua istilah ini adalah penyebutan untuk konsep desain “tumpukan” (stack), di mana sebuah software bisa berperilaku sebagai fondasi bagi software yang dibangun di atasnya. Software yang berada di **lapisan bawah** disebut **back-end** (back = belakang), software yang dibangun di **lapisan atasnya** disebut **front-end** (front = depan). Biasanya, sebuah front-end sengaja dibuat untuk menyediakan user interface (antarmuka pengguna) yang mudah digunakan. Biasanya pula, sebuah program front-end hanya bertugas memberikan tampilan luar sementara proses-proses riilnya tetaplah dikerjakan oleh program back-end. Front-end dilihat dari wujudnya ada dua macam, ada yang **CLI** (command line interface) ada juga yang **GUI** (graphical user interface).

Contoh hubungan keduanya di Ubuntu adalah **Synaptic** (front-end) terhadap **APT** dan **dpkg** (back-end) secara berurutan. Contoh front-end yang GUI misalnya Synaptic, sedangkan front-end yang CLI misalnya APT, keduanya adalah front-end terhadap dpkg. Selain APT, ada front-end yang CLI yang dibuat untuk dpkg, bernama **aptitude**. Pengguna akhir ketika mengelola paket di Debian atau Ubuntu, bisa memilih front-end CLI mana yang lebih disukai antara APT atau aptitude. Jika tidak menggunakan CLI, pengguna juga bisa memilih front-end GUI seperti Synaptic.

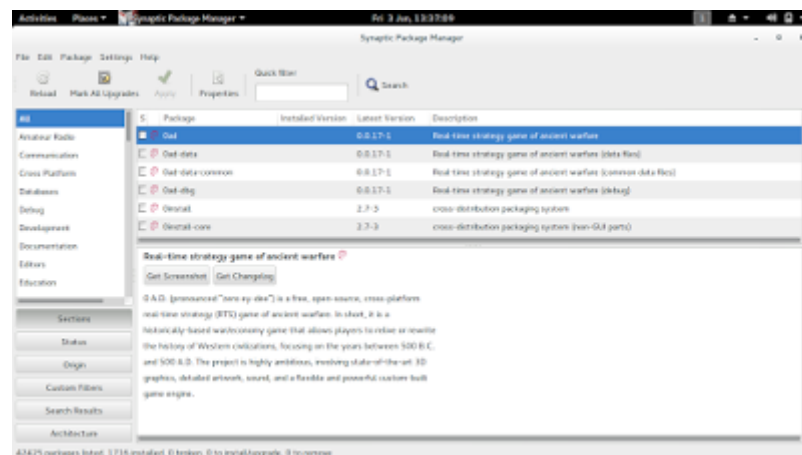
Di sini secara umum bisa dikatakan bahwa tugas Synaptic (dan front-end semisalnya) hanyalah

memberikan tampilan GUI yang mudah dipakai untuk pengguna akhir, sementara ketika proses berlangsung sesungguhnya yang bekerja adalah APT dan dpkg. Pembaca diharapkan tidak lagi bingung dengan istilah front-end dan back-end nantinya.

Contoh GUI Front-End untuk Manajemen Paket

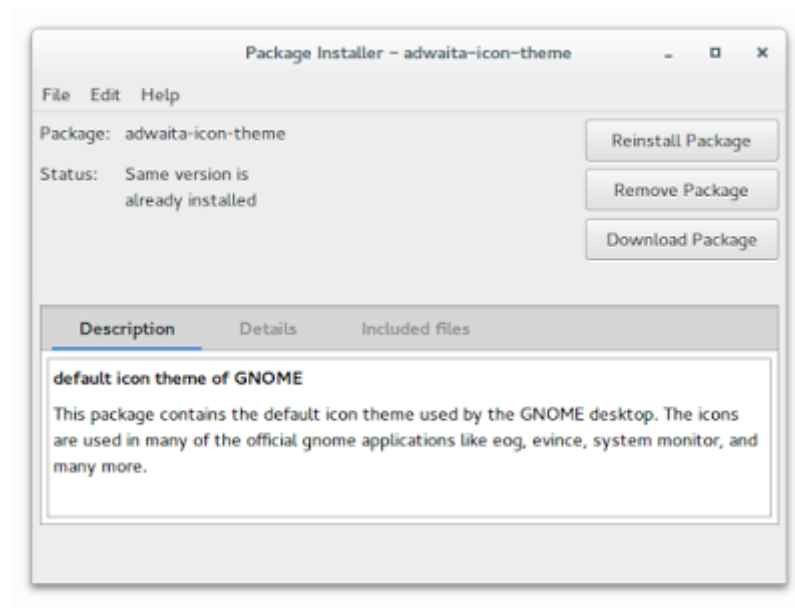
Program manajer paket asli sistem operasi keluarga Debian adalah dpkg. Selain dpkg, software semacam APT atau Synaptic adalah front-end terhadap dpkg. Tanpa APT atau Synaptic, sebuah sistem Debian tetap bisa mengelola paket dengan dpkg saja. Tetapi sebaliknya, menggunakan APT atau Synaptic tanpa dpkg adalah mustahil. Hanya saja, software seperti APT adalah CLI front-end dan Synaptic adalah GUI front-end terhadap dpkg. Tulisan ini akan memperkenalkan GUI front-end bagi sistem manajemen paket Debian (termasuk Ubuntu) yang Anda bisa pilih mana yang paling nyaman digunakan. Tulisan ini tidak membahas CLI front-end seperti APT atau aptitude. Menggunakan GUI front-end di sini sama seperti menggunakan back-end hanya saja dalam banyak sisi lebih mudah, lebih efisien, dan lebih hemat waktu.

1. Synaptic



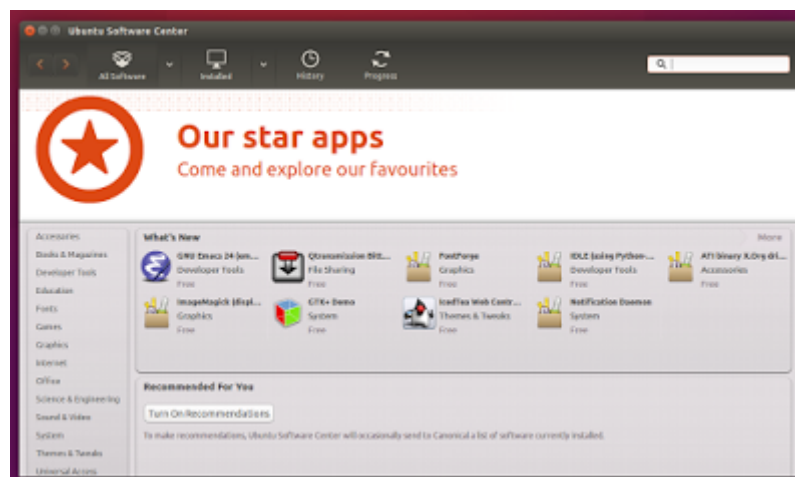
Synaptic Package Manager adalah program GUI yang menampilkan antarmuka pencarian paket yang mudah. Jika di konsol pengguna mengetik perintah untuk mencari paket maupun menginstal, di GUI di Synaptic pengguna bisa mencari paket-paket secara live kemudian cukup pilih dan klik tombol untuk menginstal. Synaptic terkenal dengan pola pemakaiannya yang sederhana: cari, centang, instal. Synaptic mendukung jaringan dengan proxy yang berautentikasi, mendukung history, pembuatan script untuk instalasi offline, dan lain-lain. Di antara GUI front-end yang lain, dihitung dari segi teknis manipulasi paket di keluarga Debian, Synaptic adalah yang paling fleksibel.

2. GDebi



GDebi adalah GUI front-end untuk dpkg. GDebi bisa Anda bayangkan menginstal paket atau banyak paket seperti menggunakan dpkg tetapi dilakukan dengan sekali klik. Selain itu, GDebi memiliki kelebihan yakni mampu mencari dependensi dari paket yang diminta kemudian menginstalnya. Pengguna yang suka menginstal paket-paket pihak ketiga akan senang menggunakan GDebi.

3. Ubuntu Software Center



Ubuntu Software Center adalah GUI front-end buatan Canonical khusus untuk Ubuntu. Dengan Ubuntu Software Center (USC) pengguna akhir dapat melakukan instalasi paket-paket software

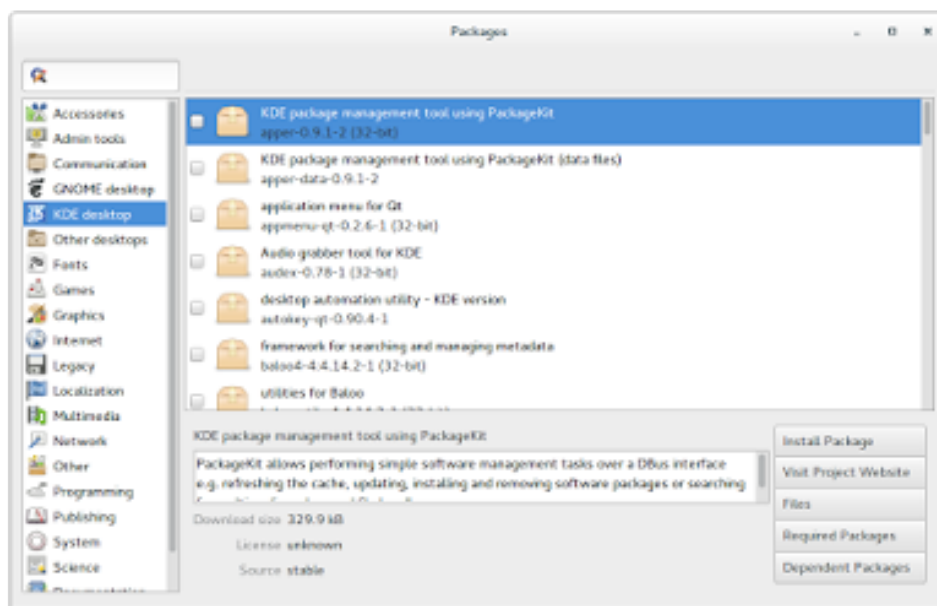
secara iconic (bergambar, ber-icon) dengan klik saja. USC dilengkapi fitur dasar ditambah bisa menginstal paket lokal seperti halnya GDebi.

4. GNOME Software



GNOME Software adalah sebuah GUI front-end buatan developer GNOME sebagai bagian dari GNOME desktop environment. GNOME Software relatif baru, fiturnya kira-kira setara dengan USC. GNOME Software adalah GUI front-end untuk PackageKit (dijelaskan di bawah).

5. GNOME PackageKit



PackageKit adalah sistem universal yang memberikan kemudahan bagi pengguna distro apa pun dengan sistem manajemen paket apa pun untuk menginstal dan meng-upgrade paket-paket sistem. PackageKit memiliki GUI front-end yang bagus yang bernama **GNOME PackageKit**. Dalam konteks tulisan ini, PackageKit (daemon: `packagekitd`) adalah front-end untuk **APT** sedangkan GNOME PackageKit adalah GUI front-end untuk PackageKit. Di luar konteks, PackageKit adalah front-end pula untuk manajer paket distro lain seperti zypp (openSUSE), yum (Red Hat Enterprise Linux), emerge (Gentoo), dan lain-lain. GNOME PackageKit merupakan satu GUI front-end untuk semua distro dalam hal ini. PackageKit dan GNOME PackageKit adalah karya Richard Hughes, seorang maintainer Proyek GNOME.

Catatan: jika di lingkungan GNOME ada **GNOME PackageKit** sebagai GUI untuk PackageKit, maka di lingkungan KDE ada juga **KPackageKit** (sekarang bernama **Apper**) yang setara dengan GNOME PackageKit.

6. Muon Discover



Muon adalah GUI front-end untuk APT dari Proyek KDE. Muon Discover adalah nama program yang cenderung lebih mirip seperti USC atau GNOME Software. Fitur mereka kurang lebih sama. Intinya, Muon Discover menitikberatkan pada kemudahan untuk pengguna akhir. Muon Discover bisa ditemukan di distribusi yang memakai KDE sebagai bawaan, contohnya Kubuntu.

Catatan Penulis

Terima kasih untuk Kang Abu Fakhri Ali atas koreksi mengenai aptitude dan PackageKit untuk tulisan ini pada 5 Juni 2016.

Referensi

- <https://www.freedesktop.org/software/PackageKit/index.html>
- <http://www.hughsie.com/public/introduction-to-packagekit.pdf>
- <https://www.freedesktop.org/software/PackageKit/pk-intro.html>
- <https://sources.debian.net/src/gdebi/0.9.5.7/README>
- <https://github.com/GNOME/gnome-packagekit/blob/master/AUTHORS>
- <https://userbase.kde.org/KPackageKit>
- <https://www.freedesktop.org/software/PackageKit/pk-intro.html>

Bab 7: Mengenal Apt-Cache

Tutorial ini menjelaskan betapa bergunanya perintah **apt-cache** selaku bagian dari program APT di Ubuntu. Setelah pada [seri ke-3 menjelaskan dpkg](#) dan [seri ke-4 menjelaskan apt](#) secara umum, kini saatnya artikel ini lebih spesifik menjelaskan bagaimana database dari APT itu bisa dilihat isinya (di-query) oleh pengguna. Keperluannya bisa sangat luas. Dengan apt-cache, pengguna bisa melihat dependensi dari suatu paket, dependensi-terbalik dari paket yang sama, status repositori yang sedang digunakan, kemungkinan upgrade dan downgrade suatu paket, dan lain-lain. Perintah ini meskipun jarang dikenal pengguna akhir, tetapi menyimpan fitur yang sangat bagus. Tulisan ini membahas apt-cache secara ringkas hanya pada perintah yang diprioritaskan agar mudah untuk pemula. Tulisan ini dilengkapi kamus istilah di bagian akhir. Semoga tulisan ini bermanfaat.

Serial Manajemen Paket Ubuntu

Berikut serial panduan manajemen paket dari kami.

- Seri 1 [Perkenalan, Dasar-Dasar, Source Code & Binary](#)
- Seri 2 [Repositori, Asal Muasal, Contoh](#)
- Seri 3 [Dpkg, List, Install, Remove](#)
- Seri 4 [APT & apt-get, Install, Remove, Upgrade](#)
- Seri 5 [Setelan sources.list & Contoh](#)
- Seri 6 [GUI Front-End](#)

Sekilas Mengingat APT

APT adalah program yang besar. APT merupakan kumpulan dari program-program yang memiliki satu fungsi masing-masing. Di antara program-program itu adalah apt-cache. Program apt-cache ini (perintah: **apt-cache**) bertugas melakukan query terhadap cache atau database milik APT. Database yang dimaksud adalah segala berkas di dalam direktori **/var/lib/apt/lists/**. Anda bisa melihat sendiri bahwa output-output dari apt-cache nanti serupa dengan isi dari database tersebut.

1. Output Normal

Perintah:

```
$ apt-cache
```

Contoh Output:

```
master@master:~$ apt-cache
apt 1.2.10 (i386)
Usage: apt-cache [options] command
       apt-cache [options] show pkg1 [pkg2 ...]

apt-cache queries and displays available information about
installed
and installable packages. It works exclusively on the data
acquired
into the local cache via the 'update' command of e.g. apt-get. The
displayed information may therefore be outdated if the last update
was
too long ago, but in exchange apt-cache works independently of the
availability of the configured sources (e.g. offline).

Most used commands:
  showsrc - Show source records
  search  - Search the package list for a regex pattern
  depends - Show raw dependency information for a package
  rdepends - Show reverse dependency information for a package
  show    - Show a readable record for the package
  pkgnames - List the names of all packages in the system
  policy  - Show policy settings

See apt-cache(8) for more information about the available
commands.
Configuration options and syntax is detailed in apt.conf(5).
Information about how to configure sources can be found in
sources.list(5).
Package and version choices can be expressed via
apt_preferences(5).
Security details are available in apt-secure(8).
master@master:~$
```

Penjelasan:

Jika perintah apt-cache dikerjakan begitu saja tanpa imbuhan argumen, maka output yang Anda lihat seperti di atas. Ingat, output di atas berasal dari versi APT 1.2.10. Output pada versi APT Anda mungkin berbeda tetapi intinya kurang lebih akan sama.

2. Mencari Paket

Perintah:

```
$ apt-cache search <kata_kunci>
```

Contoh perintah:

```
$ apt-cache search pdf reader
$ apt-cache search office suite
$ apt-cache search video editor
$ apt-cache search audio editor
$ apt-cache search image editor
```

Contoh output:

```
master@master:~$ apt-cache search pdf reader
antiword - Converts MS Word files to text, PS, PDF and XML
c++-annotations - Extensive tutorial and documentation about C++
c++-annotations-pdf - Extensive tutorial and documentation about C++ - PDF output
calibre - e-book converter and library management
calibre-bin - e-book converter and library management
ebook-speaker - eBook reader that reads aloud in a synthetic voice
gimagereader - Graphical GTK+ front-end to tesseract-ocr
gpdftext - GTK+ text editor for ebook PDF files
ruby-pdf-inspector - Ruby library for analyzing PDF output
ruby-pdf-reader - Ruby library for accessing the content of PDF files
texlive-latex-extra - TeX Live: LaTeX additional packages
```

```
wv - Programs for accessing Microsoft Word documents
xpdf - Portable Document Format (PDF) reader
master@master:~$
```

Penjelasan:

Apakah Anda pernah bertanya mengapa Synaptic atau Ubuntu Software Center bisa menayangkan daftar lengkap paket-paket software yang tersedia di repositori, bahkan dalam keadaan offline? Jika ya, jawabannya adalah karena program GUI front-end tersebut membaca database APT yang sudah tersimpan secara offline di dalam **/var/lib/apt/lists/**. Database ini diperoleh oleh Anda sendiri dengan perintah **# apt-get update** atau dengan klik tombol **Reload**. Pertanyaan selanjutnya adalah bagaimana melakukan pencarian yang sama seperti GUI front-end itu tetapi di Terminal? Jawabannya adalah dengan perintah **\$ apt-cache search**.

Perintah ini akan melakukan hal yang sama ketika Anda mencari kata kunci suatu paket di suatu GUI front-end. Dia akan menampilkan di Terminal Anda daftar nama paket yang mencocoki kata kunci Anda. Nama-nama paket inilah yang Anda akan gunakan untuk **# apt-get install** nanti atau perintah lainnya.

Ubuntu 14.04 Trusty Tahr saja telah memiliki sekitar 50.000 paket lebih di dalam repositorinya. Jadi Anda tidak perlu khawatir kekurangan paket software ketika mencari. Coba biasakan mencari dengan perintah ini dan nikmati fasilitasnya.

3. Mengetahui Status Repositori

Perintah:

```
$ apt-cache policy
```

Contoh output:

```
master@master:~$ apt-cache policy
Package files:
 100 /var/lib/dpkg/status
      release a=now
 500 http://archive.ubuntu.com/ubuntu xenial-updates/main i386
Packages release v=16.04,o=Ubuntu,a=xenial-
```

```

updates,n=xenial,l=Ubuntu,c=main,b=i386
    origin archive.ubuntu.com
500 http://archive.ubuntu.com/ubuntu xenial/multiverse i386
Packages
    release
v=16.04,o=Ubuntu,a=xenial,n=xenial,l=Ubuntu,c=multiverse,b=i386
    origin archive.ubuntu.com
500 http://archive.ubuntu.com/ubuntu xenial/universe i386
Packages
    release
v=16.04,o=Ubuntu,a=xenial,n=xenial,l=Ubuntu,c=universe,b=i386
    origin archive.ubuntu.com
500 http://archive.ubuntu.com/ubuntu xenial/restricted i386
Packages
    release
v=16.04,o=Ubuntu,a=xenial,n=xenial,l=Ubuntu,c=restricted,b=i386
    origin archive.ubuntu.com
500 http://archive.ubuntu.com/ubuntu xenial/main i386 Packages
    release
v=16.04,o=Ubuntu,a=xenial,n=xenial,l=Ubuntu,c=main,b=i386
    origin archive.ubuntu.com
Pinned packages:
master@master:~$

```

Penjelasan:

Perhatikan bagian-bagian yang **ditebalkan** di atas.

Dengan perintah `policy`, tanpa imbuhan argumen, kita bisa melihat repositori apa saja yang aktif di dalam Ubuntu kita. Tidak hanya itu, kita juga bisa melihat komponen dan jenis apa saja yang aktif. Di dalam contoh di atas, Ubuntu 16.04 Xenial Xerus saya tampak menggunakan dua jenis repositori yaitu `$release: xenial` dan repositori `$release-updates: xenial-updates`. Pada repositori `xenial`, Ubuntu saya mengaktifkan keempat komponen `main restricted universe multiverse` bersamaan. Pada repositori `xenial-updates`, Ubuntu saya mengaktifkan hanya repositori `main`. Artinya, ketika saya menginstal program dengan `apt-get install` nanti, `apt` akan mencari kepada repositori-repositori yang tersebut di atas.

Ketika Anda mengurangi atau menambah jumlah repositori pada `sources.list`, setelah `Reload`, maka hasil dari `apt-cache policy` ini akan berubah pula. Hal ini dikarenakan `apt-cache` membaca perubahan dari database APT.

4. Mengetahui Beda Versi Paket pada Sejumlah Repositori yang Berbeda

Perintah:

```
$ apt-cache policy <nama_paket>
```

Contoh perintah:

```
$ apt-cache policy linux-image-generic
$ apt-cache policy bash
$ apt-cache policy inkscape
```

Contoh output:

```
master@master:~$ apt-cache policy linux-image-generic
linux-image-generic:
  Installed: 4.4.0.21.22
  Candidate: 4.4.0.24.25
Version table:
   4.4.0.24.25 500
                 500 http://archive.ubuntu.com/ubuntu xenial-updates/main
i386 Packages
                 500 http://security.ubuntu.com/ubuntu xenial-security/main
i386 Packages
*** 4.4.0.21.22 500
                 500 http://archive.ubuntu.com/ubuntu xenial/main i386
Packages
   100 /var/lib/dpkg/status
master@master:~$
```

Penjelasan:

Perhatikan bagian-bagian yang **ditebalkan** di atas.

Dengan menambah imbuhan argumen nama paket pada perintah policy, maka kita bisa mengetahui paket tersebut beradanya di repositori mana (saja) dan versi berapa (saja) yang tersedia pada setiap repositori. Kita juga bisa tahu versi terbaru berapa yang bisa diinstal sekarang. Hal ini penting untuk upgrade atau downgrade paket. Penting juga untuk mengetahui versi terbaru yang ada di repositori updates atau security.

Pada contoh di atas, output dari proses paket linux-image-generic menunjukkan bahwa yang terinstal di dalam Ubuntu saya adalah versi 4.4.0.21 (lihat tanda `***`). Output itu menunjukkan bahwa versi 4.4.0.21 itu diinstal dari repositori xenial dari komponen main.

Adapun versi terbaru dari paket linux-image-generic berdasarkan output di atas, adalah 4.4.0.24. Versi terbaru ini tersedianya di repositori xenial-updates dan di xenial-security. Ketika saya menjalankan `# apt-get install linux-image-generic` pada kondisi ini, maka APT akan memberi tahu saya bahwa paket ini bisa di-upgrade ke versi 4.4.0.24. Sekali lagi, apt-cache bisa mengetahui versi-versi ini karena membaca database `/var/lib/apt/lists/`.

5. Mengetahui Metadata Paket

Perintah:

```
$ apt-cache show <nama_paket>
```

Contoh perintah:

```
$ apt-cache show gimp
$ apt-cache show nautilus
$ apt-cache show firefox
$ apt-cache show wvdial
```

Contoh output:

```
master@master:~$ apt-cache show wvdial
Package: wvdial
Priority: optional
Section: comm
```

Installed-Size: 263

Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>

Original-Maintainer: Thierry Randrianiriana <thierry@debian.org>

Architecture: i386

Version: 1.61-4.1

Depends: ppp (>= 2.3.0), debconf (>= 0.5.00) | cdebconf, libc6 (>= 2.4), libstdc++6 (>= 4.1.1), libuniconf4.6, libwvstreams4.6-base, libwvstreams4.6-extras, debconf (>= 0.5) | debconf-2.0

Filename: pool/main/w/wvdial/wvdial_1.61-4.1_i386.deb

Size: 76616

MD5sum: 4bd62b8b5afedb610ce5aac78ac4d522

SHA1: 0c234445497bd7c7ad9ac67dbac0a07cd6109ffb

SHA256:

cfecc0074759aac409ce636eb0389026dcec81dc27c6b80288300f7f2f115700

Description-en: intelligent Point-to-Point Protocol dialer

WvDial sacrifices some of the flexibility of programs like "chat" in order

to make dialup configuration easier. With WvDial, modems are detected automatically and only three additional parameters

are required: the telephone number, username, and password.

WvDial knows

enough to dial with most modems and log in to most servers without any other help.

.

In particular, a "chat script" is not required to handle the most common situations.

Description-md5: b8bf30c8dfd4d09e02af74bf497505d6

Homepage: <http://alumnit.ca/wiki/index.php?page=WvDial>

Bugs: <https://bugs.launchpad.net/ubuntu/+filebug>

Origin: Ubuntu

Supported: 5y

Task: lubuntu-desktop

master@master:~\$

Penjelasan:

Perhatikan bagian-bagian yang **ditebalkan** di atas.

Pernahkah Anda bertanya dari mana GUI front-end bisa mengetahui nama paket, total ukuran, total ukuran ketika selesai diinstal, teks deskripsi, dan terpenting daftar dependensi paketnya? Tidak peduli apakah itu Synaptic atau apt-get, bagaimana mereka bisa menayangkannya? Jawabannya adalah dari database APT. Perintah apt-cache show menayangkan metadata itu dengan sempurna.

Menganalisis output dari sebuah query dari apt-cache bisa panjang sekali. Sebagai contohnya diambil satu kasus. Lihat bagian yang ditebalkan di atas. Lihat pada kolom **Depends:** yang di sana disebutkan nama-nama paket lain. Itulah daftar dependensi dari paket **wvdial**.

Ketika perintah apt-get install membaca metadata dari paket wvdial, dia akan membaca kolom dependensinya. Ketika membaca dependensi-dependensi tersebut, otomatis apt-get akan juga membaca dependensi dari masing-masing dependensi. Terus dilakukan itu sembari membandingkannya dengan database dpkg status (**/var/lib/dpkg/status**) sampai ujung akhirnya. Lalu daftar dependensinya ditayangkan ke layar. Itu tugas apt-get. Namun jika kita ingin mencari sendiri apa dependensi dari suatu paket, kita cukup menggunakan apt-cache show.

Keterangan di atas berlaku juga untuk kolom-kolom lain. Misalnya kenapa apt-get bisa mengetahui persis alamat unduhan setiap paket .deb di setiap proses instalasi? Jawabannya adalah kolom **Filename:** di atas. apt-get akan menggabungkan nilai metadata itu dengan alamat yang Anda setel di dalam sources.list. Jadilah nanti URL lengkap. Semua perbuatan apt-get yang sangat panjang ini terjadi dalam orde milidetik sehingga kita tidak menyadarinya.

Kamus Istilah

- **Repositori:** sumber pengambilan paket. Bentuk fisiknya biasanya sebuah server di internet. Di keluarga Debian, sumber-sumber itu (alamat-alamat URL itu) disetel di dalam berkas **/etc/apt/sources.list**. Baca kembali artikel [perkenalan repositori](#).
- **Metadata:** data yang memuat informasi tentang data lain. Di dalam konteks kita, metadata yang dimaksud adalah informasi yang terdapat di dalam database APT mengenai suatu paket. Informasi ini diambil dari berkas bernama **control** yang pasti ada di setiap paket.
- **Argumen:** teks yang dituliskan setelah nama perintah. Contohnya perintah **mkdir rumahku** berarti **rmdir** perintahnya sedangkan **rumahku** argumennya.
- **Dependensi:** dependency, ketergantungan; dalam konteks manajemen paket artinya kondisi ketika sebuah paket dibutuhkan oleh paket lain, tanpa paket itu maka paket lain itu tidak bisa diinstal.
- **Dependensi-terbalik:** reverse-dependency, ketergantungan terbalik; dalam konteks manajemen paket artinya daftar paket yang membutuhkan paket yang bersangkutan, tanpa paket yang bersangkutan paket yang dependensi-terbaliknya mengarah kepadanya tidak bisa diinstal.

Bab 8: Penggunaan Synaptic di Ubuntu

Tulisan seri ke-8 ini menjelaskan bagaimana penggunaan dasar **Synaptic Package Manager** di Ubuntu. Synaptic adalah GUI untuk menginstal, menghapus, dan meng-upgrade paket-paket dengan mudah bagi sistem operasi Ubuntu. Secara fungsi, Synaptic itu bagaikan *Add/Remove Programs* bagi Windows, *Play Store* bagi Android, atau *App Store* bagi Apple OS X. Anda akan mengenal bagaimana menggunakan Synaptic untuk pertama kali dan menggunakannya untuk kebutuhan sehari-hari. Tulisan ini adalah kelanjutan spesifik dari [Mengetahui GUI Front-End](#).

Serial Manajemen Paket Ubuntu

Berikut serial panduan manajemen paket dari kami.

- Seri 1 [Perkenalan, Dasar-Dasar, Source Code & Binary](#)
- Seri 2 [Repositori, Asal Muasal, Contoh](#)
- Seri 3 [Dpkg, List, Install, Remove](#)
- Seri 4 [APT & apt-get, Install, Remove, Upgrade](#)
- Seri 5 [Setelan sources.list & Contoh](#)
- Seri 6 [GUI Front-End](#)
- Seri 7 [Mengetahui Apt-Cache](#)

Perkenalan Synaptic

Synaptic Package Manager (sering disebut *Synaptic* saja) adalah GUI front-end untuk manajemen paket di Debian. Synaptic adalah perwakilan terhadap program apt dan dpkg. Dengan menggunakan Synaptic, Anda sebenarnya menggunakan apt dan dpkg juga, hanya saja dengan Synaptic lebih mudah karena aksi dilakukan dengan klik tombol. Dengan Synaptic Anda bisa melakukan yang bisa dilakukan dengan apt dan dpkg secara manual, seperti menyetel sumber repositori, mencari, menginstal, menghapus, dan meng-upgrade.

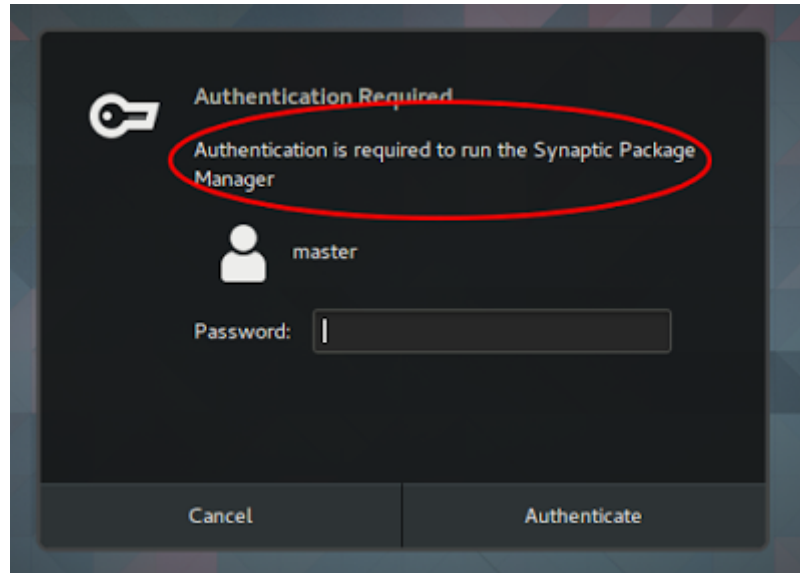
Menginstal Synaptic

Apabila Synaptic Package Manager belum terinstal di distribusi GNU/Linux Anda, maka Anda bisa menginstalnya dengan baris perintah:

```
sudo apt-get install synaptic
```

Membuka Synaptic

Synaptic bisa ditemukan di menu utama desktop Anda setelah dia diinstal. Misalnya di GNOME, buka menu (Win) lalu ketik "synaptic" lalu klik ikon Synaptic yang muncul. Anda akan diminta memasukkan password untuk membukanya.



Jika Anda ingin membuka Synaptic dari baris perintah, maka perintahnya:

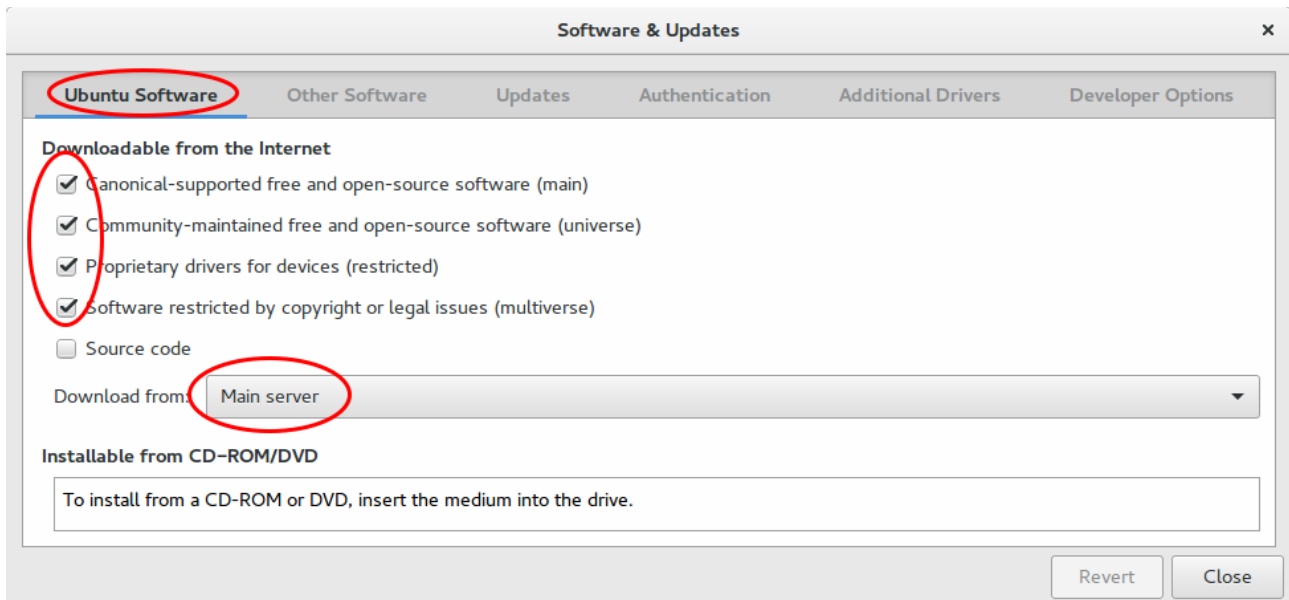
```
$ synaptic-pkexec
```

1. Penggunaan Pertama

Anggap Anda baru pertama kali menggunakan Synaptic. Maka hal pertama yang harus Anda lakukan adalah sebagai berikut:

1. Buka menu Tools > Repositories > tab Ubuntu Software.
2. **Centang** komponen repositori yang Anda inginkan dan **pilih** server cermin mana yang Anda inginkan. Contohnya: centang **main restricted multiverse universe** maka mengaktifkan keempat komponen repositori; pilih server cermin **Kambing UI** mengarahkan segala unduhan ke Indonesia (tidak ke luar negeri).
3. Close.
4. Klik tombol **Reload** di tool bar Synaptic.
5. Biarkan Synaptic mengunduh "peta repositori" dari setiap centangan dan pilihan yang baru dilakukan.

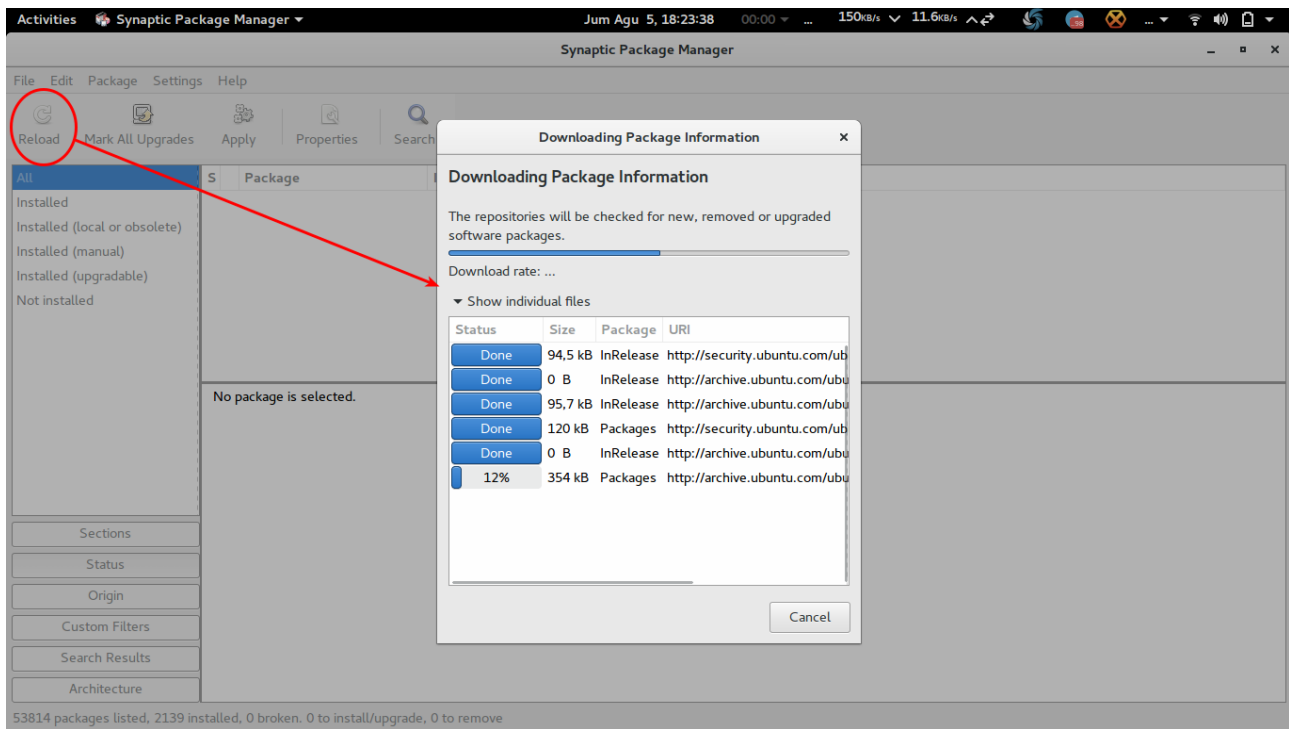
Penjelasan: yang Anda lakukan di atas sebetulnya adalah menyetel `/etc/apt/sources.list` (langkah 1 sampai 3) kemudian melakukan baris perintah `sudo apt-get update` (langkah 4). Hanya saja, di Synaptic semua aksi dilakukan secara GUI. Penggunaan pertama ini mengharuskan OS Anda tersambung ke jaringan internet karena Synaptic harus memperoleh peta repositori itu dari server di internet.



2. Reload

Tombol **Reload** yang ada di tool bar itu fungsinya untuk mengunduh "peta repositori" dari setelan repositori-repositori yang Anda telah tentukan. Pengunduhan peta repositori ini bertujuan ganda, pertama memberi tahu APT ke mana harus mengunduh paket dan kedua memberi tahu APT apabila terdapat pembaruan dari repositori yang dituju. Aksi **Reload** ini tidak akan menginstal paket. Aksi Reload ini sama dengan baris perintah **`sudo apt-get update`**.

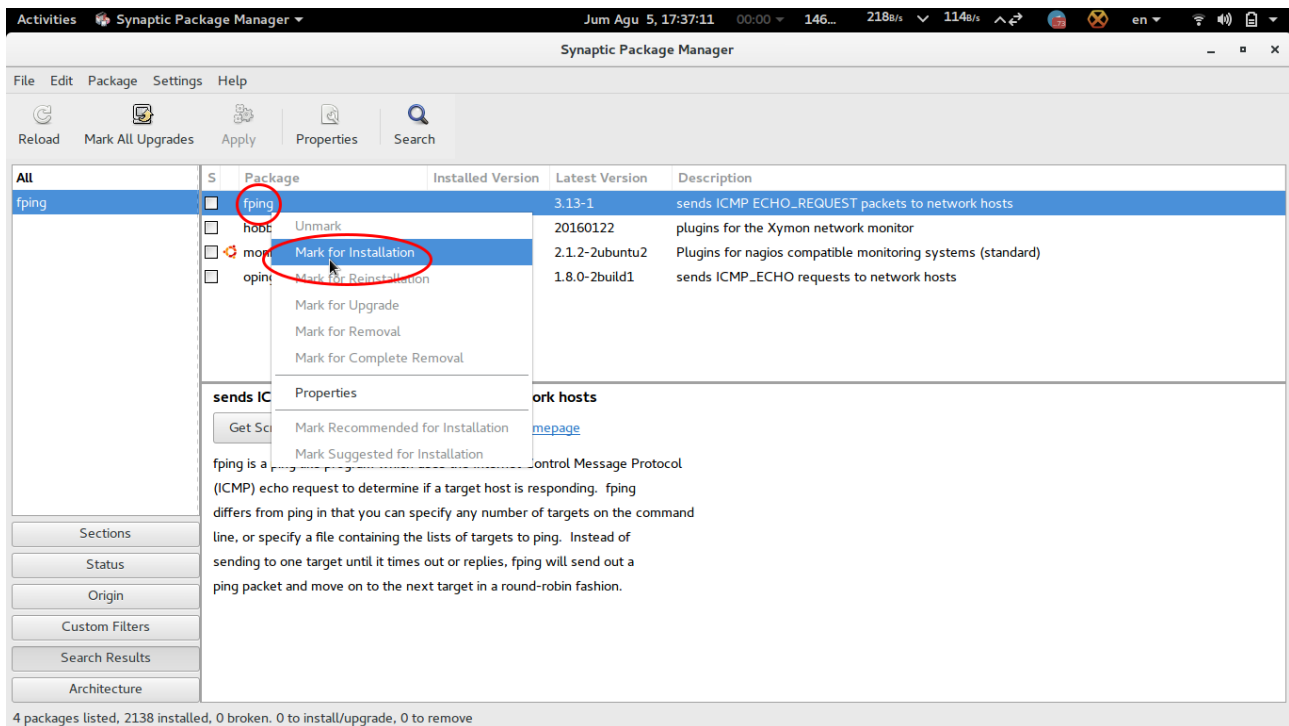
Penggunaan: lakukan Reload di setiap kali Anda melakukan perubahan terhadap `sources.list` atau melakukan bagian **1. Penggunaan Pertama** di atas.



3. Instal

Cara menginstal paket dengan Synaptic adalah sebagai berikut:

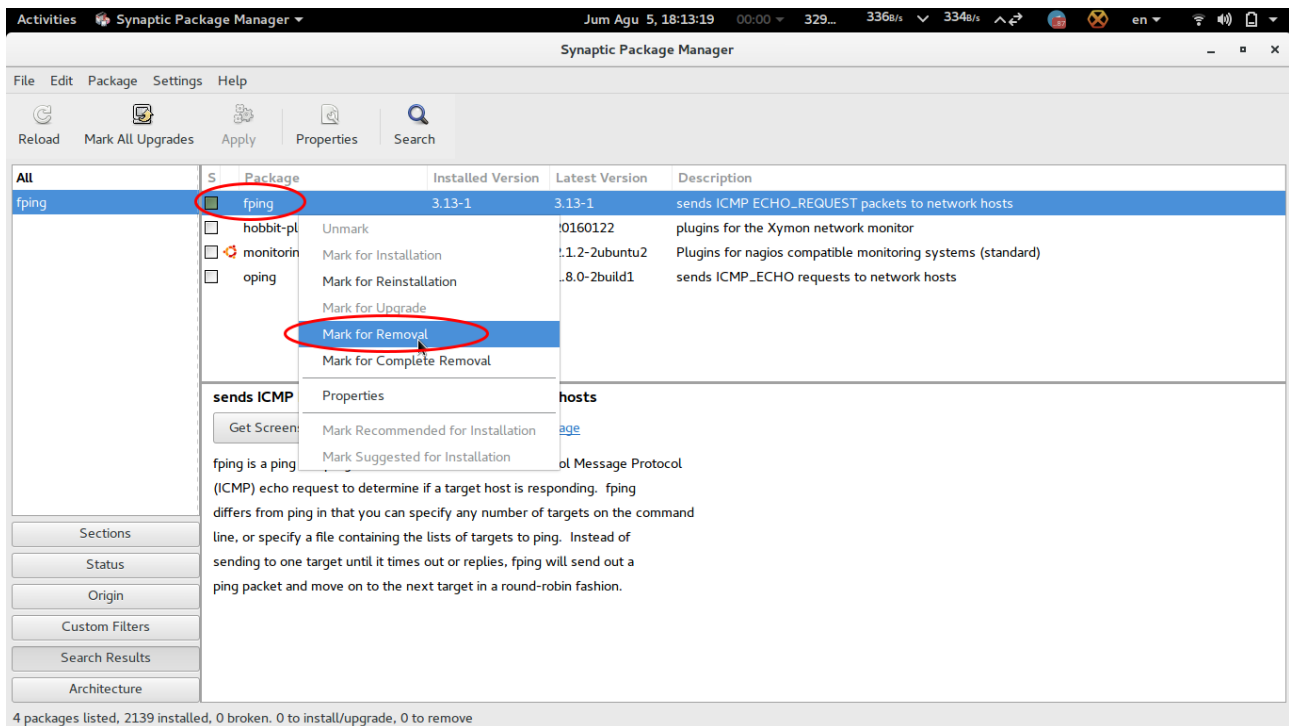
1. Cari nama paket yang hendak diinstal.
2. Klik kanal pada nama paket > **Mark for installation**.
3. Bila Synaptic menemukan lebih dari satu paket akan diinstal, sebuah jendela akan muncul memberi tahu nama paket yang hendak diinstal dan nama-nama paket dependensinya. Setujui saja.
4. Nama paket yang akan diinstal akan tampak bertanda panah kuning di jendela utama.
5. Klik tombol **Apply** di tool bar.
6. Jendela dialog baru akan menayangkan daftar nama-nama paket yang mau diinstal dan total ukurannya. Untuk menginstal, tekan tombol **Apply**.
7. Jendela baru akan menayangkan prosesi pengunduhan paket beserta kecepatan transfer jaringan di setiap paket yang diunduh.
8. Jendela baru akan muncul setelah seluruh unduhan selesai menayangkan progress bar dengan jendela hitam (konsol) menayangkan prosesi kerja dpkg menginstali satu per satu paket yang telah diunduh.
9. Jika jendela tersebut menyatakan instalasi selesai, maka instalasi selesai.



4. Hapus

Untuk menghapus (uninstall) suatu paket, ikuti langkah berikut:

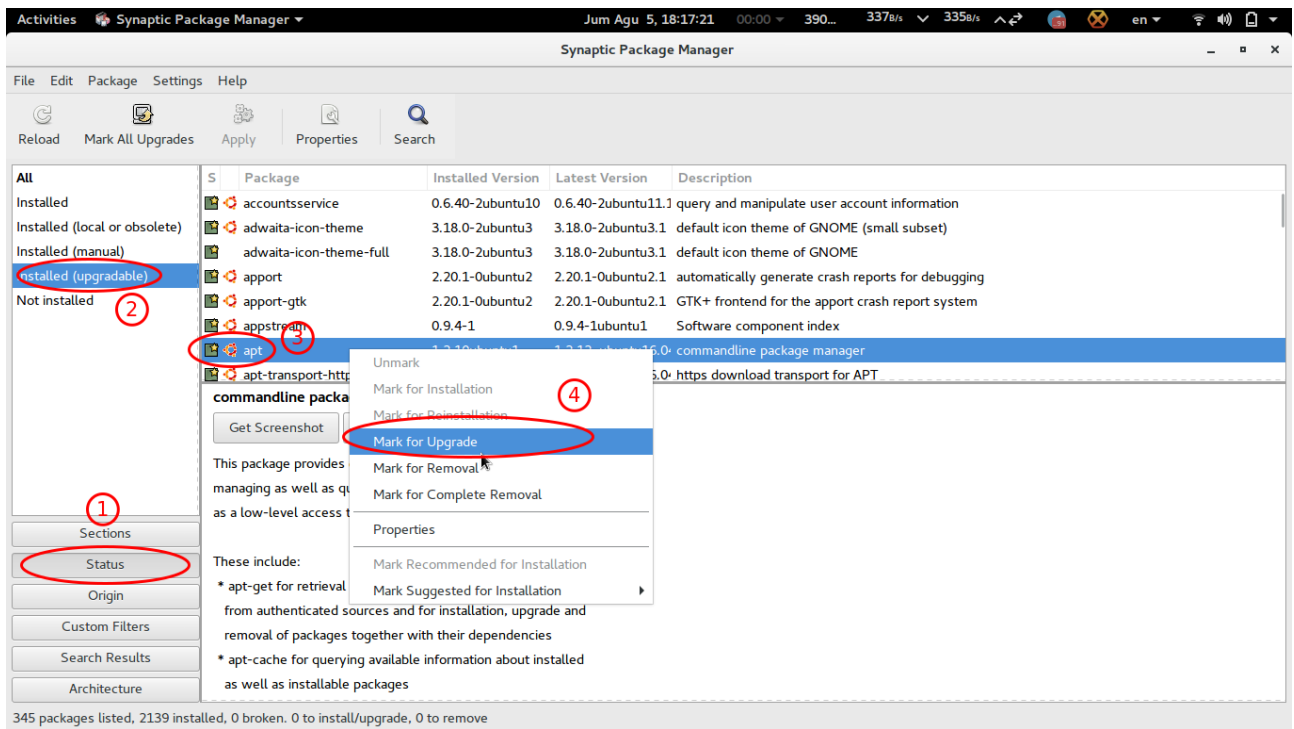
1. Cari nama paket yang hendak dihapus.
2. Klik kanan nama paket > **Mark for removal**.
3. Tekan tombol **Apply** di tool bar.
4. Jendela baru muncul menayangkan daftar nama paket yang hendak dihapus. Perhatikan bahwa pada beberapa jenis paket, jumlah yang dihapus bisa lebih dari satu. Setujui saja.
5. Jendela baru muncul menayangkan proses penghapusan.



5. Upgrade

Untuk meng-upgrade suatu paket tertentu (bukan keseluruhan paket di dalam sistem) ikuti langkah berikut:

1. Cari nama paket yang mau di-upgrade. Caranya klik tombol Status (kiri bawah) lalu klik pilihan "**Installed (upgradable)**" (kiri atas) lalu cari nama paket yang ada di jendela tengah.
2. Klik kanan nama paket > **Mark for Upgrade**.
3. Tekan tombol **Apply** di tool bar.



Bab 9: Memperoleh Dependensi Secara Offline

Tulisan ini adalah seri ke-9 dari serial panduan sistem manajemen paket Ubuntu. Tulisan ini memberikan contoh dalam 3 langkah bagaimana pengguna Ubuntu yang komputernya **offline** bisa menginstal paket-paket software tanpa komputernya tersambung ke internet. Pengunduhan paket-paket software bisa dilakukan di komputer lain dengan sistem operasi apa pun secara instan. Tulisan ini tidak terikat pada repositori Ubuntu sehingga bisa diterapkan pada semua keluarga Debian (Linux Mint, BlankOn, dll.).

Serial Manajemen Paket Ubuntu

Berikut serial panduan manajemen paket dari kami.

- Seri 1 [Perkenalan, Dasar-Dasar, Source Code & Binary](#)
- Seri 2 [Repositori, Asal Muasal, Contoh](#)
- Seri 3 [Dpkg, List, Install, Remove](#)
- Seri 4 [APT & apt-get, Install, Remove, Upgrade](#)
- Seri 5 [Setelan sources.list & Contoh](#)
- Seri 6 [GUI Front-End](#)
- Seri 7 [Mengetahui Apt-Cache](#)
- Seri 8 [Penggunaan Synaptic Package Manager](#)

Syarat

Untuk menerapkan contoh ini ada syarat yang harus dipenuhi pengguna:

- Ubuntu yang berada di komputer offline harus sudah pernah mengalami **Reload** (sudo apt-get update) yakni sudah memiliki "[peta repositori](#)".
- Paket dependensi untuk satu versi Ubuntu **tidak bisa diinstal** di Ubuntu versi lain. Misalnya paket dependensi di 14.04 hanya bisa diinstal di 14.04 pula, tidak di versi lain.

Langkah 1: Memperoleh URL Seluruh Dependensi

Dilakukan di: komputer offline.

Jalankan contoh perintah-perintah di bawah ini di Ubuntu yang sedang offline. Setiap perintah ini menghasilkan sebuah berkas TXT yang isinya adalah baris-baris URL paket-paket dependensi. Sesuaikan saja nama paket yang dicontohkan dengan nama paket yang Anda mau instal. Perintah-perintah ini tidak membutuhkan jaringan internet.

Contoh 1: URL Dependensi Inkscape

```
sudo apt-get install --print-uris --yes inkscape | sed -n "/p" | cut --delim=" " --field=2 > dependensi-inkscape.txt
```

Contoh 2: URL Dependensi GIMP

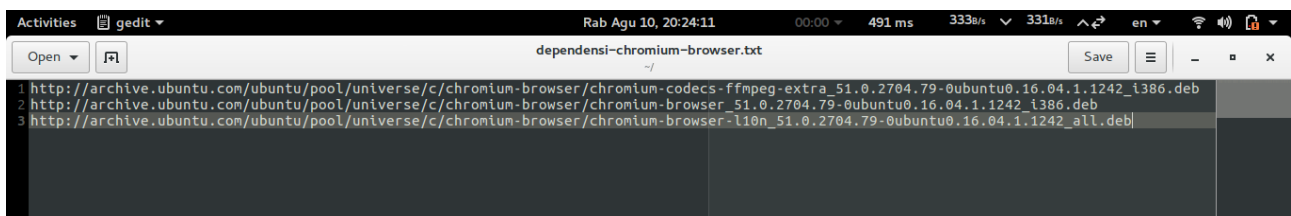
```
sudo apt-get install --print-uris --yes gimp | sed -n "/p" | cut --delim=" " --field=2 > dependensi-gimp.txt
```

Contoh 3: URL Dependensi Chromium Browser

```
sudo apt-get install --print-uris --yes chromium-browser | sed -n "/p" | cut --delim=" " --field=2 > dependensi-chromium-browser.txt
```

Contoh Output

Contoh isi dari berkas TXT dari contoh 3 di atas adalah seperti gambar berikut. Di dalam TXT ini tercantum 3 buah URL untuk 3 paket .deb yang merupakan dependensi untuk menginstal aplikasi Chromium Browser.



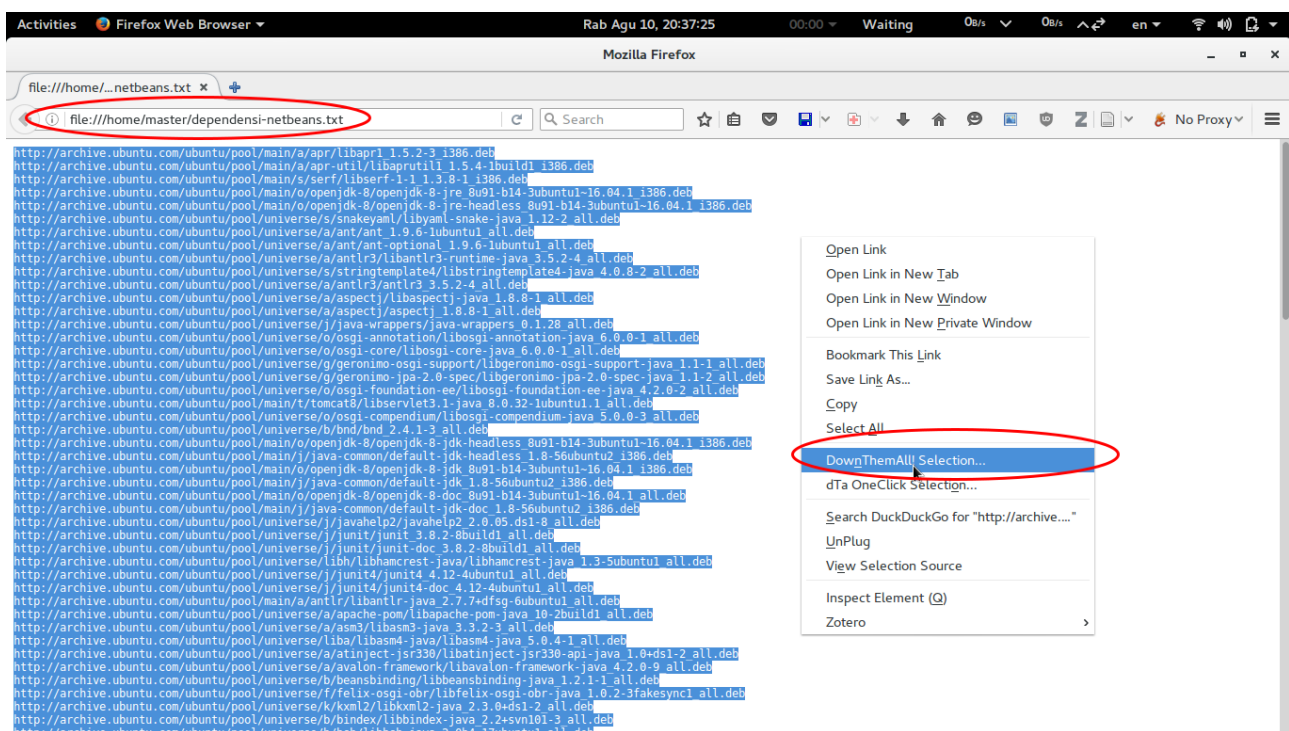
Langkah 2: Mengunduh Seluruh Dependensi

Dilakukan di: komputer online.

Bawalah berkas TXT yang sudah berisi URL paket-paket dependensi tadi ke **komputer lain** yang tersambung ke internet. Sistem operasi pada komputer lain ini bebas, terserah apakah itu Windows, OS X, BSD, GNU/Linux, atau sama-sama Ubuntu.

Unduh semua URL yang tercantum dalam berkas TXT tadi ke sebuah folder di media penyimpanan Anda. Setelah semua paket .deb dari URL-URL tersebut selesai diunduh, bawa pulang media penyimpanan itu ke komputer offline Anda.

Untuk mempercepat proses pengunduhan di komputer lain tersebut, gunakan program *download manager* yang mampu menerima berkas TXT sebagai input. Contohnya dengan Mozilla Firefox gunakan DownThemAll! untuk mengunduh seluruh URL sekaligus. Caranya adalah buka terlebih dahulu TXT itu di jendela baru lalu dengan DownThemAll! pilih dan unduh semuanya. **Ini cara instan dan salah satu yang tercepat.**



Ini sangat penting jika berkas TXT Anda memuat ratusan URL. Karena tentu Anda tidak akan suka mengunduh ratusan URL satu per satu.

Langkah 3: Menginstal Seluruh Dependensi

Dilakukan di: komputer offline.

Salin folder yang sudah berisi paket-paket dependensi tadi ke Ubuntu di komputer offline Anda. Agar Anda mudah, salin ke \$HOME Anda.

Anggaplah nama folder itu **dependensi**. Buka Terminal dan laksanakan perintah:

```
cd ~/dependensi  
sudo dpkg -i *.deb
```

Maka perintah ini akan memindahkan konsol Anda dahulu ke direktori dependensi lalu akan memerintahkan dpkg untuk menginstal paket-paket .deb di dalamnya satu per satu sampai selesai. Dengan cara ini Anda telah menginstal sebuah software secara offline di Ubuntu. Perintah ini telah dijelaskan di [Seri 3 Mengetahui Dpkg](#).

Penjelasan

Perintah panjang untuk memperoleh URL dependensi pada *Langkah 1* di atas sejatinya adalah **3 perintah** yang disambung jadi satu. **Tujuan** dari perintah ini adalah mengeluarkan daftar URL dependensi yang lengkap untuk suatu nama paket. **Output** dari perintah ini adalah sebuah berkas TXT yang berisi baris-baris alamat URL dependensi suatu paket.

1. **Perintah pertama** adalah perintah apt-get khusus untuk mencetak URL dependensi ke layar. Opsi yang membuat apt-get bisa mencetak URL dependensi adalah **`--print-uris`**. Perintah pertama ini adalah yang paling esensial.
2. **Perintah kedua** adalah perintah sed khusus untuk mengambil hanya URL-URL dari output perintah pertama. Perintah ini ditambahkan karena output dari perintah pertama mengandung teks-teks yang bukan URL. Namun output dari perintah kedua ini masih menyisakan satu bagian teks yang bukan URL.
3. **Perintah ketiga** adalah perintah cut khusus untuk membuang teks sisa bukan URL dari daftar URL yang telah terbentuk. Output dari perintah ketiga ini adalah hanya benar-benar daftar URL murni yang dicetak ke layar.
4. Redireksi (tanda `>`) menyimpan output dari gabungan 3 perintah di atas menjadi sebuah berkas TXT.

Catatan Penting

Apabila pembaca tidak mengetahui nama paket dari program yang hendak diinstal, pembaca dapat

mencari nama paket tersebut di <http://packages.ubuntu.com>. Situs ini adalah layanan resmi Ubuntu yang mampu mencari nama-nama paket di dalam seluruh repositori resmi Ubuntu pada seluruh versi Ubuntu. Lihat contoh berikut ini.

The screenshot shows the Ubuntu Package Search Results page in a Firefox browser. The search term 'gnome' is entered in the search bar. The results are categorized into 'Exact hits' and 'Other hits'. The 'Exact hits' section lists several packages, with 'Package gnome' highlighted. Red arrows and text annotations are present: 'CARI DI SINI' points to the search bar, and 'INI NAMA PAKETNYA' points to the highlighted 'Package gnome' entry.

ubuntupackages

gnome

package names Search status

» Ubuntu » Packages » Package Search Results

Search in specific suite: [precise] [precise-updates] [precise-backports] [trusty] [trusty-updates] [trusty-backports] [wily] [wily-updates] [wily-backports] [xenial] [xenial-updates] [xenial-backports] [yakkety]

Limit search to a specific architecture: [i386] [amd64] [powerpc] [arm64] [armhf] [ppc64el] [s390x]

You have searched for packages that names contain gnome in all suites, all sections, and all architectures. Found 100 matching packages.

Your keyword was too generic, for optimizing reasons some results might have been suppressed. Please consider using a longer keyword or more keywords.

Exact hits

Package **gnome**

- **precise (12.04 LTS)** (gnome): The GNOME Desktop Environment, with extra components [universe]
13.0+6ubuntu3: amd64 armhf i386 powerpc
- **trusty (14.04 LTS)** (gnome): Full GNOME Desktop Environment, with extra components [universe]
13.8+4ubuntu3: amd64 arm64 armhf i386 powerpc ppc64el
- **wily** (gnome): Full GNOME Desktop Environment, with extra components [universe]
13.8+4ubuntu3: amd64 arm64 armhf i386 powerpc ppc64el
- **xenial** (gnome): Full GNOME Desktop Environment, with extra components [universe]
13.14+3ubuntu1: amd64 arm64 armhf i386 powerpc ppc64el s390x
- **yakkety** (gnome): Full GNOME Desktop Environment, with extra components [universe]
13.14+3ubuntu1: amd64 arm64 armhf i386 powerpc ppc64el s390x

Other hits

CARI DI SINI

INI NAMA PAKETNYA

Bab 10: Memperoleh Daftar Dependensi via Synaptic Script

Tulisan ini adalah seri ke-10 dari serial sistem manajemen paket Ubuntu kami. Tulisan ini melengkapi seri ke-9 dengan menunjukkan bagaimana mengeluarkan daftar URL dependensi secara GUI dengan Synaptic. Hasil yang diperoleh sama dengan seri ke-9 hanya saja dilakukan dengan cara berbeda. Semoga tulisan ini bermanfaat.

Serial Manajemen Paket Ubuntu

Berikut serial panduan manajemen paket dari kami.

- Seri 1 [Perkenalan, Dasar-Dasar, Source Code & Binary](#)
- Seri 2 [Repositori, Asal Muasal, Contoh](#)
- Seri 3 [Dpkg, List, Install, Remove](#)
- Seri 4 [APT & apt-get, Install, Remove, Upgrade](#)
- Seri 5 [Setelan sources.list & Contoh](#)
- Seri 6 [GUI Front-End](#)
- Seri 7 [Mengenal Apt-Cache](#)
- Seri 8 [Penggunaan Synaptic Package Manager](#)
- Seri 9 [Memperoleh Dependensi Secara Offline](#)

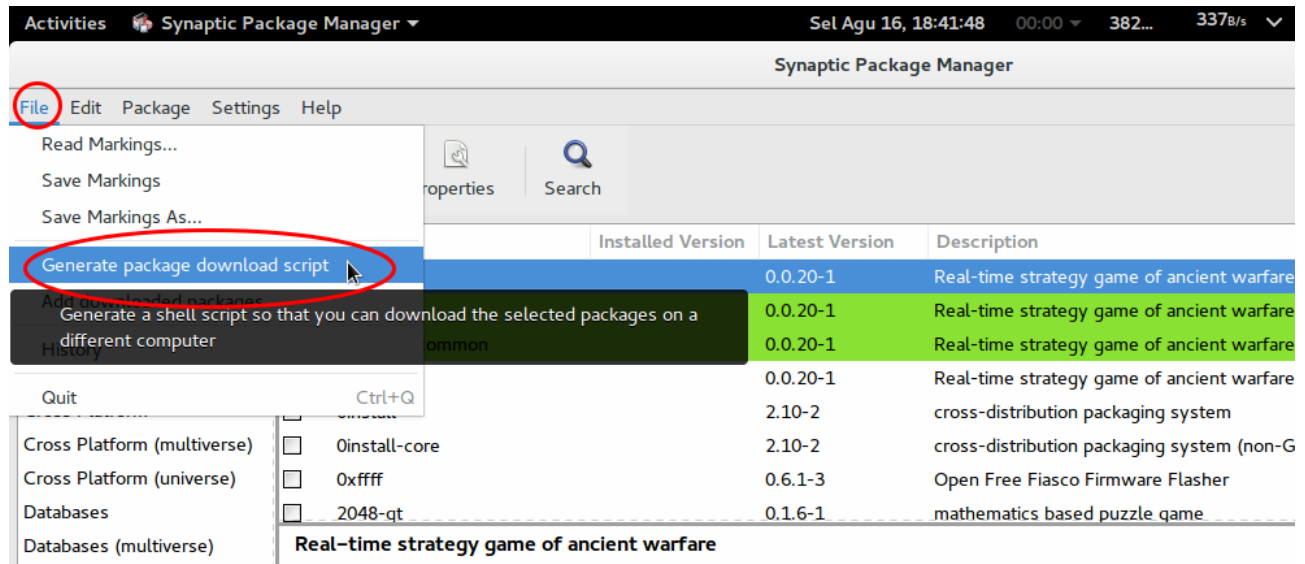
Syarat

Untuk menerapkan contoh ini ada syarat yang harus dipenuhi pengguna:

- Ubuntu yang berada di komputer offline harus sudah pernah mengalami **Reload** (sudo apt-get update) yakni sudah memiliki "[peta repositori](#)".
- Paket dependensi untuk satu versi Ubuntu **tidak bisa diinstal** di Ubuntu versi lain. Misalnya paket dependensi di 14.04 hanya bisa diinstal di 14.04 pula, tidak di versi lain.

Langkah 1: Memilih Paket

Buka Synaptic lalu centang paket mana yang mau diinstal lalu pilih Mark for installation. Buka menu File > Generate file download script > pilih tempat penyimpanan (letakkan di \$HOME user Anda) > beri nama file > OK. Anggaplah nama file yang Anda tentukan **dependensi.txt**.

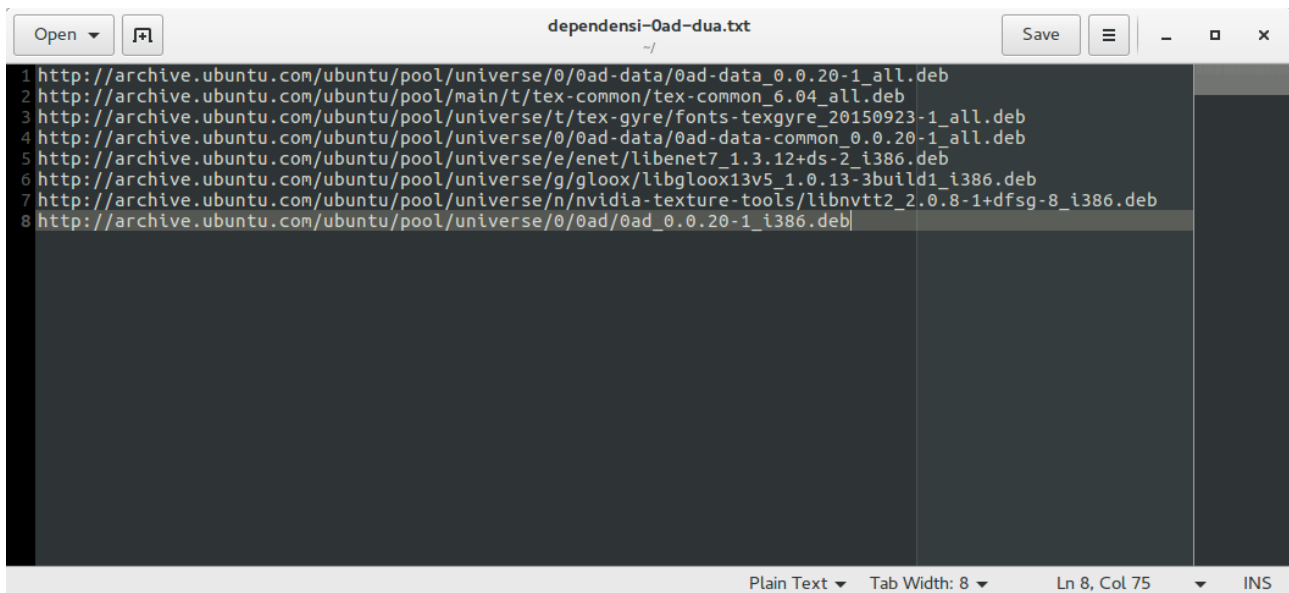


Langkah 2: Memodifikasi Berkas Script

Isi dari berkas yang dihasilkan oleh Synaptic sebetulnya berbentuk source code dari bash script. Panduan ini ditujukan untuk diterapkan di semua OS. Maka kita harus memodifikasi sedikit isi script ini supaya isinya murni daftar URL saja. Gunakan perintah berikut.

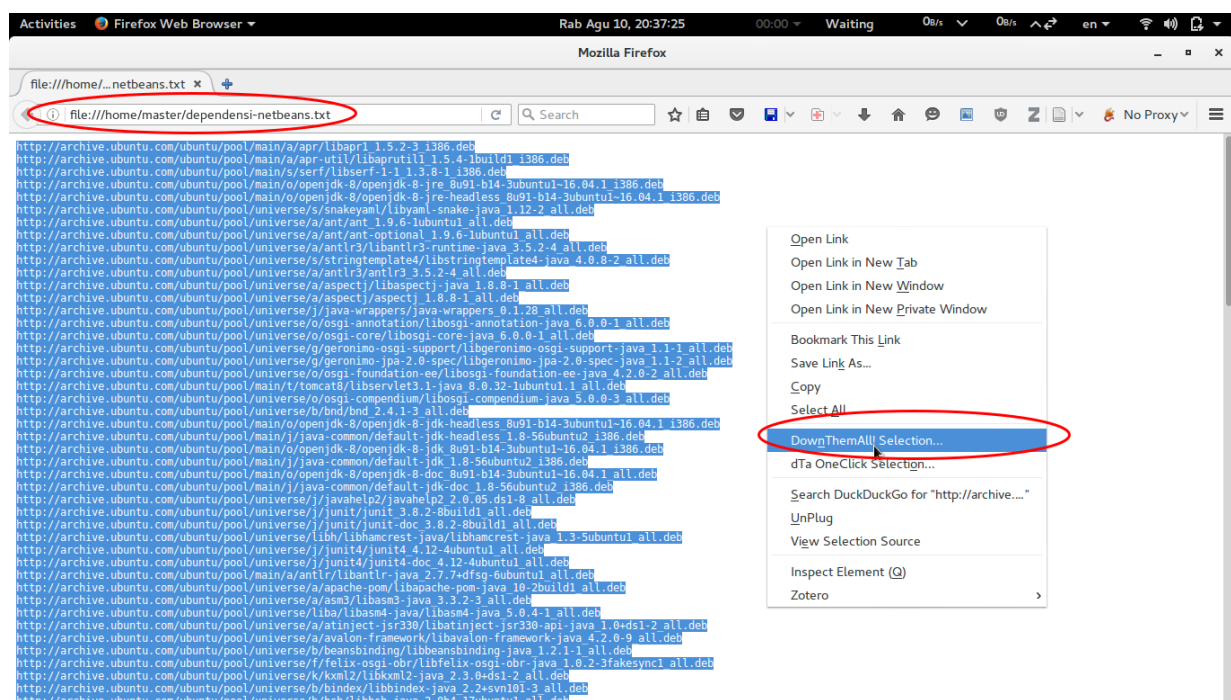
```
$ awk '{ print $3 }' dependensi.txt | sed 'ld' > namafile.txt
```

Ganti namafile.txt dengan nama berkas script yang Anda hasilkan. Maka hasilnya script tersebut akan murni berisi daftar URL saja. Contoh hasil yang diinginkan adalah seperti ini:



Langkah 3: Mengunduh Seluruh Dependensi

Bawalah berkas daftar URL tadi ke komputer yang tersambung internet. Gunakan browser untuk mengunduh seluruh dependensinya. Mau lebih cepat lagi, gunakan download manager seperti DownThemAll! (pengaya Firefox) untuk mengunduh semua URL sekaligus dalam sekali klik. Simpan semua dependensi (berformat .deb) dalam sebuah folder dan bawalah ke komputer yang offline. Contoh demonstrasi pengunduhannya seperti ini:



Langkah 4: Menginstal Seluruh Dependensi

Pindahkan seluruh dependensi yang telah diperoleh ke satu folder di Ubuntu yang offline. Pindahkan dia ke \$HOME saja biar mudah. Anggaplah nama folder itu **dependensi**. Buka Terminal dan laksanakan perintah:

```
cd ~/dependensi  
sudo dpkg -i *.deb
```

dan tunggu sampai selesai. Maka program ini akan terinstal di Ubuntu yang offline.

Bab 11: Repositori Pihak Ketiga (PPA)

Tulisan ini memperkenalkan penggunaan repositori pihak ketiga di Ubuntu. Di dalamnya termasuk bagaimana mencari PPA, mengetahui kecocokan PPA dengan versi sistem sekarang, memasang sebuah PPA, sampai menginstal paket dari PPA, dan terakhir menghapus sebuah PPA. Tulisan ini adalah seri ke-11 dari serial sistem manajemen paket Ubuntu kami.

Serial Manajemen Paket Ubuntu

Berikut serial panduan manajemen paket dari kami.

- Seri 1 [Perkenalan, Dasar-Dasar, Source Code & Binary](#)
- Seri 2 [Repositori, Asal Muasal, Contoh](#)
- Seri 3 [Dpkg, List, Install, Remove](#)
- Seri 4 [APT & apt-get, Install, Remove, Upgrade](#)
- Seri 5 [Setelan sources.list & Contoh](#)
- Seri 6 [GUI Front-End](#)
- Seri 7 [Mengenal Apt-Cache](#)
- Seri 8 [Penggunaan Synaptic Package Manager](#)
- Seri 9 [Memperoleh Dependensi Secara Offline](#)
- Seri 10: [Memperoleh Dependensi dengan Synaptic Script](#)

1. Mengenal Repositori

Repositori telah dijelaskan panjang lebar pada tulisan ini [seri ke-2 Mengenal Repositori](#).

2. Mengenal Repositori Pihak Ketiga

Di luar repositori pihak pertama (pihak resmi yang membuat sebuah distribusi GNU/Linux), pengguna akhir bisa memperoleh suatu software dari pihak ketiga atau dari luar repositori resmi, yang disebut **repositori pihak ketiga**. Di dunia keluarga Ubuntu, repositori semacam ini sering dinamakan **Personal Package Archive** atau **PPA**. Sebuah PPA adalah satu direktori di internet yang mengandung sebuah atau beberapa paket software yang dibangun khusus untuk Ubuntu versi tertentu.

- **Apa definisi PPA?** PPA adalah sebuah repositori di luar repositori resmi yang berisi paket versi tertentu untuk Ubuntu versi tertentu yang dibuat oleh suatu pihak untuk publik.
- **Apa tujuan dibuatnya sebuah PPA?** Di sisi developer, untuk mendistribusikan suatu software langsung kepada pengguna akhir tanpa harus memasuki repositori resmi Ubuntu.
Catatan: ada persyaratan ketat dan proses panjang supaya suatu software dapat masuk ke repositori resmi. Dengan PPA, prosedur itu tidak dibutuhkan.
- **Di mana saya bisa menemukan PPA-PPA?** Platform Ubuntu memiliki layanan PPA gratis paling besar di dunia, yang diletakkan di server Launchpad, dinamakan **Launchpad PPA**. Lihat <https://launchpad.net>.
- **Siapa yang membuat sebuah PPA?** Umumnya pihak perorangan di luar Canonical, atau pihak *upstream* yang ingin versi terbaru programnya dapat diakses semua pengguna Ubuntu.
- **Kenapa saya butuh sebuah PPA?** Karena mungkin versi paling baru dari software kesayangan Anda tidak tersedia di repositori resmi, mungkin suatu software tidak tersedia di repositori resmi, atau mungkin suatu software tertentu tidak ditemukan kecuali di dalam sebuah PPA. Anda bisa menginstal software-software itu dari PPA tertentu.
- **Apa saja isi sebuah PPA?** Ada PPA yang berisi software aplikasi, PPA berisi software baris perintah, kernel, tema desktop, driver printer, driver VGA, program editor teks, paket fonta, software library, paket wallpaper, atau apa pun lainnya yang dapat dibayangkan untuk dipasang di sebuah OS. Lihat <https://launchpad.net>.
- **Bagaimana menggunakan PPA?** Anda akan selalu menginstal dahulu alamat sebuah PPA sebelum bisa menginstal software yang terdapat di dalamnya.
- **Apakah PPA aman?** Pada dasarnya, **tidak**. PPA adalah repositori pihak ketiga (bukan repositori resmi) sehingga Anda harus yakin PPA yang Anda pilih aman terlebih dulu sebelum menggunakannya. Jangan instal apa pun dari sumber yang tidak dipercaya. Menggunakan PPA dari Launchpad cenderung lebih aman (dan cara ini disukai mayoritas pengguna akhir) karena adanya mekanisme-mekanisme keamanan untuk menjamin keabsahan setiap paket dan servernya.
- **Apa segala software tersedia PPA-nya?** Mayoritas, ya. Namun tetap ada software yang tidak tersedia PPA-nya karena tidak ada orang yang membuatkan PPA untuknya.

3. Mengenal Launchpad

Launchpad adalah layanan yang disediakan oleh Canonical Ltd. kepada semua orang untuk meng-hosting proyek-proyek free software. Istilah untuk Launchpad adalah *code hosting service*. Launchpad itu setara dengan layanan lain seperti Savannah, SourceForge, GitHub, BitBucket, dan seterusnya. Hanya saja, Launchpad punya fasilitas khusus yakni hosting PPA untuk Ubuntu. Situs resmi Launchpad adalah <https://launchpad.net>.

Isi Launchpad secara umum dapat dibagi menjadi dua: source code dan binary. Artinya, di dalam Launchpad Anda bisa menemukan dua jenis halaman berbeda untuk satu software yang sama, yakni **halaman pengembangannya** (isinya paket source code) dan **halaman PPA-nya** (isinya paket binary). Yang Anda butuhkan dalam hal PPA ini adalah **halaman PPA**. Anda akan mencari PPA di dalam Launchpad, yang semuanya adalah berisi paket berbentuk binary (.deb).

Catatan: walaupun mayoritas software telah memiliki PPA sendiri-sendiri di Launchpad, tidak tertutup kemungkinan ada PPA-PPA lain di luar Launchpad yang bisa Anda gunakan. Launchpad dijelaskan di sini sebagai contoh utama.

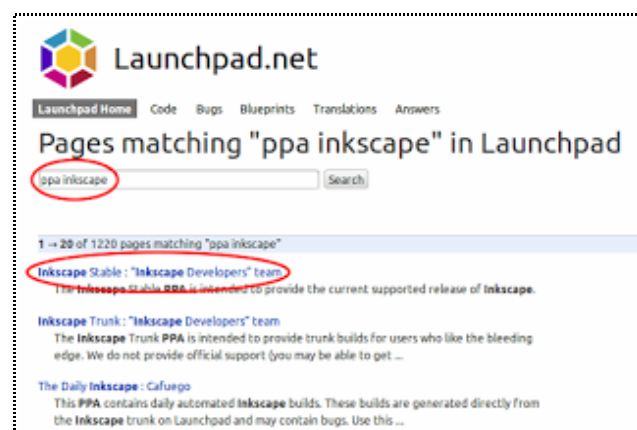
4. Contoh PPA

- PPA untuk Inkscape: <https://launchpad.net/~inkscape.dev/+archive/ubuntu/stable>
- PPA untuk GIMP: <https://launchpad.net/~otto-kesselgulasch/+archive/ubuntu/gimp>
- PPA untuk Blender: <https://launchpad.net/~thomas-schiex/+archive/ubuntu/blender>
- PPA untuk Scribus: <https://launchpad.net/~scribus/+archive/ubuntu/ppa>

Catatan: nama-nama otto.kesselgulasch atau thomas-schiex di atas adalah nama-nama pihak yang membuat masing-masing PPA. Mereka tidak mesti developer asli dari software di dalam PPA tersebut, melainkan kebanyakannya adalah distributor saja.

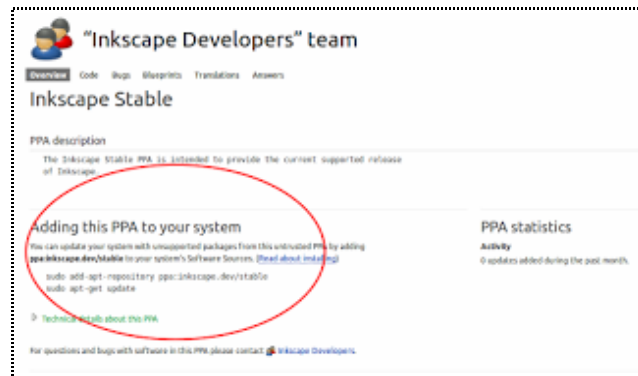
5. Cara Mencari PPA

Cara tercepat mencari sebuah PPA adalah dengan mengunjungi <https://launchpad.net> lalu mencari dengan kata kunci "**ppa <nama_software>**". Contohnya untuk mencari PPA Inkscape "**ppa inkscape**":



Nama PPA "**Inkscape Stable: Inkscape Developers Team**" dari hasil pencarian tersebut jika

dibuka maka seharusnya tampak seperti ini, yakni terdapat instruksi cara menambahkan PPA ke dalam sistem Anda:



6. Yang Perlu Diperhatikan

Hal penting yang perlu diperhatikan sebelum memasang sebuah PPA adalah sebagai berikut:

1. **Ketahui versi Ubuntu Anda dan *codename*-nya.** Misalnya 12.04 itu Precise Pangolin, 16.04 itu Xenial Xerus, dan seterusnya.
2. **Ketahui arsitektur Anda.** Anda harus tahu bahwa paket berlabel **i386** mewakili komputer 32 bit dan label **amd64** mewakili komputer 64 bit. Jangan salah arsitektur.
3. **Ketahui kecocokan versi paket di PPA dengan versi Ubuntu Anda.** Jika sebuah PPA tidak menyediakan versi untuk Xenial Xerus yang sedang Anda pakai, maka carilah PPA lain yang menyediakan paket untuk Xenial Xerus, atau tunggu empunya PPA memaketkan untuk Xenial Xerus.
4. **Ketahui alamat PPA yang Anda tuju.** Format alamat PPA dari Launchpad seperti ini:
ppa:nama-developer/nama-ppa dengan contohnya
ppa:inkscape.dev/stable.

7. Memasang Sebuah PPA

Terdapat sekian banyak cara untuk menginstal PPA. Dimulai dari cara normal (melalui baris perintah apt) sampai cara manual (melalui editing langsung sources.list). Pilihlah satu cara yang Anda suka. Dijelaskan sebagai berikut.

(1) **Cara baris perintah** adalah dengan menjalankan perintah:

```
$ sudo apt-add-repository ppa:nama-developer/nama-ppa
```


diikuti dengan satu kali [Reloading](#):

```
$ sudo apt-get update
```

sebagai contoh langsungnya begini untuk memasang PPA Inkscape:

```
$ sudo apt-add-repository ppa:inkscape.dev/stable  
$ sudo apt-get update
```

Catatan: cara pertama ini lebih direkomendasikan dan lebih aman bagi semua pengguna.

(2) Cara **sources.list** adalah dengan edit berkas **sources.list**:

- Buka **/etc/apt/sources.list** dengan editor kesukaan Anda dalam mode root.
- Masukkan alamat URL PPA ke dalam baris terakhir **sources.list**, contohnya

```
deb http://ppa.launchpad.net/inkscape.dev/stable/ubuntu  
YOUR_UBUNTU_VERSION_HERE main.
```

- Ganti "YOUR_UBUNTU_VERSION_HERE" dengan codename dari Ubuntu Anda. Misalnya **precise** untuk 12.04, **xenial** untuk 16.04, dan seterusnya.
- Contoh baris yang benar untuk 16.04 adalah

```
deb http://ppa.launchpad.net/inkscape.dev/stable/ubuntu xenial main.
```

- Lakukan [Reloading](#): **sudo apt-get update**.

Sampai di sini saja Anda sudah berhasil memasang sebuah PPA (tetapi belum menginstal software darinya).

8. Mengecek Versi Software dari PPA

Bukti bahwa pemasangan PPA berhasil adalah Anda bisa melihat dua versi berbeda, dari satu software yang sama di sistem Anda. Perintahkan **apt-cache policy <nama_program>** maka seharusnya Anda bisa melihat output seperti berikut:

```
master@master: ~
File Edit View Search Terminal Help
master@master:~$ apt-cache policy inkscape
inkscape:
  Installed: 0.91-7ubuntu2
  Candidate: 0.91.0+52~ubuntu16.04.1
  Version table:
     0.91.0+52~ubuntu16.04.1 500
        500 http://ppa.launchpad.net/inkscape.dev/stable/ubuntu xenial/main i386
 Packages
 *** 0.91-7ubuntu2 500
        500 http://archive.ubuntu.com/ubuntu xenial/main i386 Packages
        100 /var/lib/dpkg/status
master@master:~$
```

Output di atas menunjukkan dua versi Inkscape, yakni **versi telah terinstal** (berasal dari repositori resmi) dan **versi siap diinstal** (berasal dari PPA). Tampak di situ versi dari PPA sedikit lebih baru dibanding versi repositori resmi, oleh karena itu posisinya ada di atas baris repositori resmi. Dengan cara ini Anda bisa mengecek versi setiap kali Anda menginstal PPA.

9. Menginstal Software dari PPA

Untuk menginstal software yang PPA-nya telah Anda pasang, cukup perintahkan:

```
$ sudo apt-get install <nama_paket>
```

Contohnya sesuai konteks tulisan ini:

```
$ sudo apt-get install inkscape
```

Catatan: jika Inkscape telah terinstal di sistem Anda, maka perintah di atas akan meng-upgrade Inkscape tersebut ke versi terbaru dari PPA.

10. Menghapus Software yang Terinstal dari PPA

Untuk menghapus software yang terinstal dari PPA, sama saja dengan paket pada umumnya:

```
$ sudo apt-get remove <nama_paket>
```

11. Menghapus Sebuah PPA

Untuk menghapus sebuah PPA (bukan software dari PPA), gunakan perintah:

```
$ sudo apt-add-repository --remove ppa:nama-developer/nama-ppa
```

diikuti dengan satu kali [Reloading](#):

```
$ sudo apt-get update
```

contohnya:

```
$ sudo apt-add-repository --remove ppa:nama-developer/nama-ppa  
$ sudo apt-get update
```

12. Catatan Penting

Perintah `apt-add-repository` bekerja dengan cara membuat sebuah berkas `sources.list` spesial di dalam direktori `/etc/apt/sources.list.d/` dengan satu berkas per satu PPA. Semakin banyak Anda menginstal PPA, maka semakin banyak pula berkas di dalam direktori ini. Jika Anda menemukan suatu error, maka periksa terlebih dahulu berkas-berkas di dalam direktori ini.

Bab 12: Repositori CDROM

Ubuntu bisa menginstal software dari DVD [repositori](#) secara offline. Kemampuan ini telah tertanam di dalam [sistem APT](#) jadi pengguna hanya perlu "mengkenalkan" DVD tersebut ke sistem Ubuntu. APT akan menganggap DVD ini sama seperti server di internet sehingga Anda bisa menginstal software darinya dan bisa melihat daftar paketnya di [Synaptic](#). Bagi pengguna offline khususnya di Indonesia, metode DVD repositori sangat memudahkan. Semoga ini bermanfaat.

Tentang CD

Ubuntu mampu menerima media CD dan DVD yang berisi paket-paket binary .deb sebagai repositori. Ini berarti:

1. Ubuntu bisa menginstal software dari **CD Ubuntu itu sendiri** karena di dalamnya selalu ada repositori mini dari ruangan "main".
2. Ubuntu juga bisa menginstal software dari **CD repositori** yang bisa dibeli di [Juragan Kambing](#) atau [Baliwae](#) itu.
3. Ubuntu juga bisa menginstal software dari **CD apa pun yang memuat paket-paket** binary .deb.

Peringatan

Waspadai 2 hal:

1. **Jangan sembarangan instal** paket .deb dari sumber yang tidak terpercaya. Ubuntu Anda bisa dirusak oleh skrip-skrip shell berbahaya.
2. **Jangan salah versi OS**. Jangan instal paket dari versi OS Ubuntu yang berbeda. Misalnya, Ubuntu Anda 16.04 tetapi Anda pasang [repositori](#) Ubuntu 17.04 maka ini bisa merusak sistem. Ubuntu 16.04 32 bit harus dipasangi repositori Ubuntu 16.04 32 bit pula (harus sama persis).

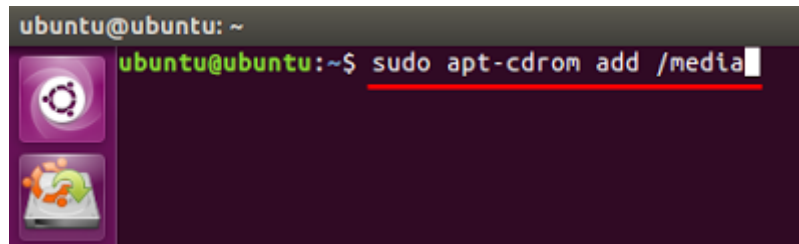
Percontohan

Artikel seri ke-12 ini mencontohkan pemasangan DVD instalasi Ubuntu 16.04 64 bit sebagai [repositori](#) untuk Ubuntu 16.04 64 bit. Adapun jenis CD lain caranya sama saja.

Langkah 1

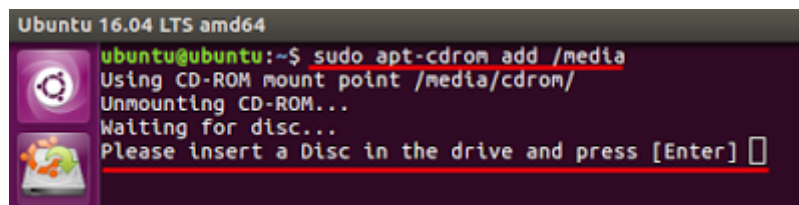
Jalankan perintah **apt-cdrom** dengan memakai alamat "mount point" dari Ubuntu Anda:

```
$ sudo apt-cdrom add /media
```



Langkah 2

Terminal akan meminta Anda memasukkan CD repositori. Masukkan CD ke perangkat CDROM lalu tekan Enter. Jika ada lebih dari 1 CD, masukkan keping urutan pertama sebelum urutan berikutnya.



Langkah 3

Biarkan Terminal menayangkan aksi apt-cdrom memindai (scanning) CD repositori Anda. Jika pemindaian selesai, apt-cdrom akan mengatakan "**Repeat this process for the rest of CDs in your set**" artinya Anda diminta memindai juga keping urutan berikutnya apabila ada.

```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ sudo apt-cdrom add /media  
Using CD-ROM mount point /media/cdrom/  
Unmounting CD-ROM...  
Waiting for disc...  
Please insert a Disc in the drive and press [Enter]  
Mounting CD-ROM...  
Identifying... [1a235e539972579c22a0bec9a077f7c3-2]  
Scanning disc for index files...  
Found 2 package indexes, 0 source indexes, 0 translation indexes and 1 signatures  
This disc is called:  
'Ubuntu 16.04 LTS _Xenial Xerus_ - Release amd64 (20160420.1)'  
Copying package lists...gpgv: Signature made Wed 20 Apr 2016 10:29:18 PM UTC using DSA key ID FBB75451  
gpgv: Good signature from "Ubuntu CD Image Automatic Signing Key <cdimage@ubuntu.com>"  
Reading Package Indexes... Done  
Writing new source list  
Source list entries for this disc are:  
deb cdrom:[Ubuntu 16.04 LTS _Xenial Xerus_ - Release amd64 (20160420.1)]/ xenial main restricted  
Unmounting CD-ROM...  
Repeat this process for the rest of the CDs in your set.  
ubuntu@ubuntu:~$
```

Langkah 4

Periksa kebenaran pemasangan sumber CD repositori di sistem Anda dengan 3 hal: perintah apt-cache policy, cek isi [sources.list](#), dan cek /var/lib/apt/lists/ Anda.

Dengan apt-cache:

Karena di setiap CD OS Ubuntu Desktop pasti ada paket **wvdial**, maka perintah **apt-cache policy wvdial** pasti menampilkan alamat CDROM berikut apabila benar sumber CD OS Ubuntu telah terpasang sebagai repo.

```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ apt-cache policy wvdial  
wvdial:  
  Installed: (none)  
  Candidate: 1.61-4.1  
  Version table:  
   1.61-4.1 500  
   500 cdrom://Ubuntu 16.04 LTS _Xenial Xerus_ - Release amd64 (20160420.1) xenial/main amd64 Packages  
   500 http://archive.ubuntu.com/ubuntu xenial/main amd64 Packages  
ubuntu@ubuntu:~$
```

Cara lebih pasti untuk melihat kebenaran pemasangan CD repositori adalah perintah **apt-cache policy** saja:

```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ apt-cache policy  
Package files:  
100 /var/lib/dpkg/status  
    release a=now  
500 http://archive.ubuntu.com/ubuntu xenial-updates/restricted amd64 Packages  
    release v=16.04,o=Ubuntu,a=xenial-updates,n=xenial,l=Ubuntu,c=restricted,b=amd64  
    origin archive.ubuntu.com  
500 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 Packages  
    release v=16.04,o=Ubuntu,a=xenial-updates,n=xenial,l=Ubuntu,c=main,b=amd64  
    origin archive.ubuntu.com  
500 http://security.ubuntu.com/ubuntu xenial-security/restricted amd64 Packages  
    release v=16.04,o=Ubuntu,a=xenial-security,n=xenial,l=Ubuntu,c=restricted,b=amd64  
    origin security.ubuntu.com  
500 http://security.ubuntu.com/ubuntu xenial-security/main amd64 Packages  
    release v=16.04,o=Ubuntu,a=xenial-security,n=xenial,l=Ubuntu,c=main,b=amd64  
    origin security.ubuntu.com  
500 http://archive.ubuntu.com/ubuntu xenial/restricted amd64 Packages  
    release v=16.04,o=Ubuntu,a=xenial,n=xenial,l=Ubuntu,c=restricted,b=amd64  
    origin archive.ubuntu.com  
500 http://archive.ubuntu.com/ubuntu xenial/main amd64 Packages  
    release v=16.04,o=Ubuntu,a=xenial,n=xenial,l=Ubuntu,c=main,b=amd64  
    origin archive.ubuntu.com  
500 cdrom://Ubuntu 16.04 LTS _Xenial Xerus_ - Release amd64 (20160420.1) xenial/restricted amd64 Packages  
    release v=16.04,o=Ubuntu,a=xenial,n=xenial,l=Ubuntu,c=restricted,b=amd64  
    origin Ubuntu 16.04 LTS _Xenial Xerus_ - Release amd64 (20160420.1)  
500 cdrom://Ubuntu 16.04 LTS _Xenial Xerus_ - Release amd64 (20160420.1) xenial/main amd64 Packages  
    release v=16.04,o=Ubuntu,a=xenial,n=xenial,l=Ubuntu,c=main,b=amd64  
    origin Ubuntu 16.04 LTS _Xenial Xerus_ - Release amd64 (20160420.1)  
Pinned packages:  
ubuntu@ubuntu:~$
```

INI ARTINYA SUMBER DVD REPO
BERHASIL DITAMBAHKAN

Cek sources.list:

Melihat [sources.list](#) bisa memastikan apakah CD repositori sudah dikenali atau tidak:

```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ cat /etc/apt/sources.list  
deb cdrom:[Ubuntu 16.04 LTS _Xenial Xerus_ - Release amd64 (20160420.1)]/ xenial main restricted  
deb http://archive.ubuntu.com/ubuntu/ xenial main restricted  
deb http://security.ubuntu.com/ubuntu/ xenial-security main restricted  
deb http://archive.ubuntu.com/ubuntu/ xenial-updates main restricted  
ubuntu@ubuntu:~$
```

INI SUMBER DVD REPO

Cek /var/lib/apt/lists/:

Melihat isi direktori ini memastikan apakah peta repositori dari CDROM sudah dibuat atau belum:

```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ cat /var/lib/apt/lists/  
archive.ubuntu.com_ubuntu_dists_xenial_InRelease  
archive.ubuntu.com_ubuntu_dists_xenial_main_binary-amd64_Packages  
archive.ubuntu.com_ubuntu_dists_xenial_main_dep11_Components-amd64.yml.gz  
archive.ubuntu.com_ubuntu_dists_xenial_main_dep11_icons-64x64.tar.gz  
archive.ubuntu.com_ubuntu_dists_xenial_main_i18n_Translation-en  
archive.ubuntu.com_ubuntu_dists_xenial_restricted_binary-amd64_Packages  
archive.ubuntu.com_ubuntu_dists_xenial_restricted_dep11_Components-amd64.yml.gz  
archive.ubuntu.com_ubuntu_dists_xenial_restricted_i18n_Translation-en  
archive.ubuntu.com_ubuntu_dists_xenial-updates_InRelease  
archive.ubuntu.com_ubuntu_dists_xenial-updates_main_binary-amd64_Packages  
archive.ubuntu.com_ubuntu_dists_xenial-updates_main_i18n_Translation-en  
archive.ubuntu.com_ubuntu_dists_xenial-updates_restricted_binary-amd64_Packages  
archive.ubuntu.com_ubuntu_dists_xenial-updates_restricted_i18n_Translation-en  
lock  
partial/  
security.ubuntu.com_ubuntu_dists_xenial-security_InRelease  
security.ubuntu.com_ubuntu_dists_xenial-security_main_binary-amd64_Packages  
security.ubuntu.com_ubuntu_dists_xenial-security_main_i18n_Translation-en  
security.ubuntu.com_ubuntu_dists_xenial-security_restricted_binary-amd64_Packages  
security.ubuntu.com_ubuntu_dists_xenial-security_restricted_i18n_Translation-en  
Ubuntu%2016.04%20LTS%20%5fXenial%20Xerus%5f%20-%20Release%20amd64%20(20160420.1)_dists_xenial_main_binary-amd64_Packages  
Ubuntu%2016.04%20LTS%20%5fXenial%20Xerus%5f%20-%20Release%20amd64%20(20160420.1)_dists_xenial_Release  
Ubuntu%2016.04%20LTS%20%5fXenial%20Xerus%5f%20-%20Release%20amd64%20(20160420.1)_dists_xenial_Release.gpg  
Ubuntu%2016.04%20LTS%20%5fXenial%20Xerus%5f%20-%20Release%20amd64%20(20160420.1)_dists_xenial_restricted_binary-amd64_Packages  
ubuntu@ubuntu:~$ cat /var/lib/apt/lists/
```

INILAH PETA DVD REPO
HASIL SCAN APT-CDROM

Selamat Menginstal

Sekarang Anda bisa menginstal software yang dimuat di dalam CD repositori tersebut dengan perintah **apt-get install** seperti biasa. Bedanya, Anda tidak akan diminta sambungan internet, melainkan hanya diminta memasukkan CD repositori ke CDROM. Saya contohkan instalasi program **wvdial** yang diambil dari CD instalasi OS Ubuntu 16.04 saya (bukan DVD repositori).

```
ubuntu@ubuntu: ~  
ubuntu@ubuntu:~$ apt-cache policy wvdial  
wvdial:  
  Installed: (none)  
  Candidate: 1.61-4.1  
  Version table:  
   1.61-4.1 500  
      500 cdrom://Ubuntu 16.04 LTS _Xenial Xerus_ - Release amd64 (20160420.1) xenial/main amd64 Packages  
      500 http://archive.ubuntu.com/ubuntu xenial/main amd64 Packages  
ubuntu@ubuntu:~$ sudo apt-get install wvdial  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  libunconf4.6 libwvstreams4.6-base libwvstreams4.6-extras  
The following NEW packages will be installed:  
  libunconf4.6 libwvstreams4.6-base libwvstreams4.6-extras wvdial  
0 upgraded, 4 newly installed, 0 to remove and 0 not upgraded.  
Need to get 0 B/858 kB of archives.  
After this operation, 2,723 kB of additional disk space will be used.  
Do you want to continue? [Y/n] y  
Get:1 cdrom://Ubuntu 16.04 LTS _Xenial Xerus_ - Release amd64 (20160420.1) xenial/main amd64 libwvstreams4.6-base amd64 4.6.1-7 [203 kB]  
Get:2 cdrom://Ubuntu 16.04 LTS _Xenial Xerus_ - Release amd64 (20160420.1) xenial/main amd64 libwvstreams4.6-extras amd64 4.6.1-7 [442 kB]  
Get:3 cdrom://Ubuntu 16.04 LTS _Xenial Xerus_ - Release amd64 (20160420.1) xenial/main amd64 libunconf4.6 amd64 4.6.1-7 [129 kB]  
Get:4 cdrom://Ubuntu 16.04 LTS _Xenial Xerus_ - Release amd64 (20160420.1) xenial/main amd64 wvdial amd64 1.61-4.1 [84.2 kB]  
Preconfiguring packages ...  
Selecting previously unselected package libwvstreams4.6-base.  
(Reading database ...
```

SOFTWARE DIINSTAL DARI CDROM