

Praktikum 10

Manajemen Paket Linux

POKOK BAHASAN:

- ✓ Manajemen Paket dengan apt dan dpkg

TUJUAN BELAJAR:

Setelah mempelajari materi dalam bab ini, mahasiswa diharapkan mampu:

- ✓ Memahami tentang manajemen paket dengan apt dan dpkg

DASAR TEORI:

1 Manajemen Paket GNU/Linux

Sebuah sistem manajemen paket bagi suatu distribusi sistem operasi GNU/Linux adalah seperangkat tool untuk mengotomatisasi proses penginstallan, pengkonfigurasi dan pembuangan suatu paket software dari sistem GNU/Linux. Biasanya sistem ini merupakan suatu bagian dari sistem operasi yang datang dalam distribusi resmi suatu distro GNU/Linux. Tools sistem manajemen paket ini biasa lebih dikenal dengan paket manager. Paket manager di GNU/Linux biasanya punya kemampuan memeriksa dan mengatur semua paket di dalam sistem linux.

2 Manajemen Paket dengan DPKG

Untuk bekerja dengan file .deb , di Ubuntu digunakan dpkg (debian package manager), sebuah tool manajemen paket yang bisa digunakan untuk untuk mengakses file .deb secara langsung, menginstall file deb, meremove paket. Dpkg merupakan tool dasar dalam manajemen paket berbasis debian. Perintah dpkg ini akan banyak digunakan ketika menginstall paket secara offline dimana paket telah didownload secara manual dari internet.

Beberapa perintah yang umum (perintah dijalankan di terminal) :

- Menginstall suatu paket .deb

```
$ sudo dpkg -i namapaket.deb
```

- Meremove paket yang sudah diinstall (tanpa meremove file konfigurasi)

```
$ sudo dpkg -r namapaket
```

- Mem-purge paket (meremove paket dan file konfigurasi) yang sudah diinstall

```
$ sudo dpkg --purge namapaket
```

- Melihat informasi daftar paket software yang terinstall

```
$ dpkg -l
```

- Melihat informasi daftar paket software yang terinstall dan memasukan daftar tersebut ke sebuah file bernama listpaket.txt

```
$ dpkg -l > listpaket.txt
```

- Melihat informasi apakah suatu paket sudah terinstall

```
$ dpkg -l namapaket
```

Kalau paket yang dimaksud belum terinstall, maka ada info “No packages found matching namapaket”

- Melihat file apa saja yang masuk ke dalam paket tertentu

```
$ dpkg -L namapaket
```

- Melihat suatu file termasuk ke paket mana

```
$ dpkg -S /path/ke/file/system
```

Namun dibalik kemudahan dan kekuatan yang datang bersama dpkg ini, terdapat juga beberapa kekurangan fitur diantaranya yang melatar belakangi dikembangkannya APT yakni:

1. Dpkg tidak menangani dependensi atau ketergantungan suatu paket akan paket lain secara otomatis, sehingga jika menginstall paket dan paket tersebut memerlukan paket lain untuk berfungsi, maka user harus secara manual menginstall paket yang dibutuhkan tersebut.
2. Dpkg tidak dapat mengambil paket dari lokasi lain di network, seperti yang APT lakukan. Jadi saat bekerja dengan dpkg, paket-paket tersebut sudah di download ke dalam sistem lokal.

3 Manajemen Paket dengan APT

APT (Advanced Packaging Tool) diciptakan karena keterbatasan dpkg. Sistem APT mengembangkan kemampuan dari dpkg sehingga mampu menangani masalah ketergantungan antar paket (dalam dunia GNU/Linux disebut “dependency hell”). APT juga mampu menjalankan tugas menginstall paket dengan mengambil secara online suatu paket dari “gudang paket” yang disebut repository.

APT bekerja dengan cara tetap menjaga daftar paket apa saja yang tersedia di dalam sistem dengan membaca sebuah daftar paket yang dapat didownload dari sebuah repository. Sumber-sumber repository dimana APT dapat mengambil paket-paketnya dikonfigurasi melalui sebuah file di `/etc/apt/sources.list`.

Sistem APT bukanlah tool tunggal. APT datang dengan beberapa tools yang sangat berguna untuk manajemen paket, di antaranya :

- `apt-mark` : digunakan secara internal oleh APT untuk memberi tanda dan menghilangkan tanda suatu paket terinstall secara otomatis.
- `apt-cdrom` : digunakan untuk menambahkan cd/dvd repository ke dalam `sources.list`
- `apt-cache` : `apt-cache` berguna untuk berbagai keperluan yang berkaitan dengan cache database APT, seperti pencarian paket, status sebuah paket, informasi tentang paket dan lainnya.

PRAKTIKUM 9 MANAJEMEN PAKET LINUX

- apt-config : program internal yang digunakan oleh berbagai bagian sistem APT untuk menjamin konsistensi pengkonfigurasiannya.
- apt-get : merupakan tool command line untuk berurusan dengan paket, dan ini yang paling umum digunakan user untuk keperluan menginstall dan remove paket.
- apt-key : digunakan untuk mengatur daftar key yang digunakan APT untuk mengautentifikasi paket. Paket yang telah diautentifikasi menggunakan key tersebut dapat dinyatakan terpercaya, berasal dari sumber yang dapat dipercaya.

Berikut beberapa perintah-perintah yang umum dan akan sering digunakan saat bekerja dengan sistem berbasis APT. Mulai dari penginstallan paket, penghapusan paket, pencarian paket, pengupdatean dan mengupgrade versi Ubuntu yang digunakan.

- Instalasi Paket Ubuntu

- Instalasi Paket Tunggal

Menginstall sebuah paket kedalam sistem, dengan perintah sebagai berikut.

```
$ sudo apt-get install namapaket
```

Atau jika menggunakan Aptitude gunakan perintah dibawah ini.

```
$ sudo aptitude install namapaket
```

- Instalasi beberapa paket secara bersamaan

```
$ sudo apt-get install paket-pertama paket-kedua
```

atau

```
$ sudo aptitude install paket-pertama paket-kedua
```

- Menghapus (remove) paket
 - Menghapus (remove) paket dari sistem tanpa menghapus file konfigurasinya

```
$ sudo apt-get remove namapaket
```

Atau dengan aptitude,

```
$ sudo aptitude remove namapaket
```

PRAKTIKUM 9 MANAJEMEN PAKET LINUX

- Menghapus paket dari sistem termasuk semua file konfigurasinya

\$ sudo apt-get remove --purge namapaket

Atau dengan aptitude,

\$ sudo aptitude purge namapaket

- Mengupgrade system

Adakalanya beberapa paket software yang ada di sistem Ubuntu sudah terlalu kuno, sementara di repository sudah ada yang lebih baru, maka perlu segera melakukan proses upgrade, baik paket individual maupun mengupgrade sistem Ubuntu secara keseluruhan.

- Mengupgrade suatu paket

Mengupgrade suatu paket software sama artinya mengganti paket software yang lama dengan versi terbaru. Hal ini cukup dilakukan dengan apt-get install seperti biasa, dengan catatan sistem APT sudah disinkronkan terlebih dahulu dengan server repository agar selalu up to date.

\$ sudo apt-get update

- Mengupgrade paket-paket system secara keseluruhan

\$ sudo apt-get upgrade

Apt-get upgrade digunakan untuk menginstall ke versi terbaru dari semua paket yang terinstall di dalam sistem Ubuntu dari repository yang terdapat di /etc/apt/sources.list.

TUGAS PENDAHULUAN :

Jawablah pertanyaan-pertanyaan di bawah ini :

1. Apa yang dimaksud dengan dpkg dan apt?

PERCOBAAN:

1. Login sebagai user.
2. Bukalah Console Terminal dan lakukan percobaan-percobaan di bawah ini kemudian analisa hasil percobaan.
3. Selesaikan soal-soal latihan.

Manajemen Paket dengan APT

Percobaan 1 : Reload Repository

1. Perintah yang digunakan untuk mengunduh daftar isi yang berupa berkas dan disimpan pada direktori `/var/lib/apt/list/` dari repositori yang telah ditunjuk oleh `source.list`. Perintah `apt-get install` bisa dilakukan jika anda sudah memiliki daftar isi repositori yang sudah dikonfigurasi di `source.list`.

```
$ sudo apt update
```

2. Perintah dibawah harus di inputkan satu persatu ke terminal. Memasukkan kedua perintah tersebut berarti anda meng-upgrade seluruh aplikasi yang terinstal pada Ubuntu anda ke versi terbarunya.

```
$ sudo apt upgrade  
$ sudo apt dist-upgrade
```

Percobaan 2 : Menginstal apache

1. Berikut perintah untuk instal apache, server HTTP Apache adalah server web yang paling banyak digunakan.

```
$ sudo apt install apache2
```

Percobaan 3 : Menghapus paket

1. Berikut perintah untuk melakukan proses penghapusan paket.

```
$ sudo apt remove apache2
```

atau

```
$ sudo apt purge apache2
```

Manajemen Paket dengan DPKG

Percobaan 4 : Install aplikasi gedit

1. Untuk menginstal aplikasi menggunakan dpkg, kita harus download file .deb terlebih dahulu baru diinstall. Untuk download file menggunakan perintah **wget**.

```
$ wget  
http://archive.ubuntu.com/ubuntu/pool/main/g/gedit/gedit-dev_3.28.1-1ubuntu1_amd64.deb
```
2. Anda dapat melihat file yang telah terdownload dengan perintah.

```
$ ls
```
3. Kemudian install aplikasi gedit tersebut. Pastikan Anda menjalankan perintah di atas dari direktori yang sama tempat paket.deb berada. Atau, jika Anda menjalankan perintah di atas dari beberapa direktori lain, sebutkan path lengkap ke file.deb

```
$ sudo dpkg -i gedit-dev_3.28.1-1ubuntu1_amd64.deb
```
4. Jika ada error saat install, maka jalankan Perintah ini akan memperbaiki paket yang rusak dan menginstal dependensi yang diperlukan dengan asumsi mereka tersedia di repositori sistem. Baru diinstall lagi aplikasinya.

```
$ sudo apt-f install
```

Percobaan 5 : Verifikasi Instalasi Paket Menggunakan Dpkg

1. Dpkg juga memungkinkan untuk memverifikasi instalasi paket. Cukup ketik dpkg dengan opsi `-status` atau `-s` untuk memverifikasi instalasi paket:

```
$ dpkg --status gedit
```

Percobaan 6 : Melihat aplikasi yang sudah terinstall menggunakan Dpkg

2. Untuk mengetahui paket apasaja yang sudah terinstall di ubuntu, gunakan perintah dpkg diikuti opsi -l. Perintah di bawah ini akan menampilkan paket aplikasi apa saja yang sudah terinstall di komputer. Gunakan tanpa pipe (|) diikuti dengan perintah less, untuk melihat output perbagian.:

```
$ dpkg -l
```

Percobaan 7 : Remove software dengan perintah dpkg

1. Untuk meremove software yang sudah di install dengan menggunakan perintah dpkg, gunakan opsi -r di ikuti nama softwarenya.

```
$ sudo dpkg -r gedit
```

LATIHAN:

1. Silahkan praktekkan perintah-perintah pada percobaan untuk contoh aplikasi yang lain baik dengan menggunakan apt dan dpkg.
2. Silahkan mempraktekkan perintah-perintah yang lain baik pada apt dan juga dpkg yang ada pada dasar teori diatas. Perintah dan juga outputya

LAPORAN RESMI:

1. Analisa hasil percobaan yang Anda lakukan.
2. Kerjakan latihan diatas dan analisa hasil tampilannya.
3. Berikan kesimpulan dari praktikum ini.