

Apa Itu LINUX ?

Linux pada awalnya dibuat oleh seorang mahasiswa Finlandia yang bernama Linus Torvalds. Dulunya Linux merupakan proyek hobi yang diinspirasi dari Minix, yaitu sistem UNIX kecil yang dikembangkan oleh Andrew Tanenbaum. Linux versi 0.01 dikerjakan sekitar bulan Agustus 1991. Kemudian pada tanggal 5 Oktober 1991, Linus mengumumkan versi resmi Linux, yaitu versi 0.02 yang hanya dapat menjalankan shell bash (GNU Bourne Again Shell) dan gcc (GNU C Compiler).

Saat ini Linux adalah sistem UNIX yang sangat lengkap, bisa digunakan untuk jaringan, pengembangan software dan bahkan untuk pekerjaan sehari-hari. Linux sekarang merupakan alternatif sistem operasi yang jauh lebih murah jika dibandingkan dengan sistem operasi komersial (misalnya Windows 9.x/NT/2000/ME). Linux mempunyai perkembangan yang sangat cepat. Hal ini dapat dimungkinkan karena Linux dikembangkan oleh beragam kelompok orang. Keragaman ini termasuk tingkat pengetahuan, pengalaman serta geografis. Agar kelompok ini dapat berkomunikasi dengan cepat dan efisien, internet menjadi pilihan yang sangat tepat.

Karena kernel Linux dikembangkan dengan usaha yang independent, banyak aplikasi yang tersedia, sebagai contoh, C Compiler menggunakan **gcc** dari **Free Software Foundation** GNU's Project. Compiler ini banyak digunakan pada lingkungan Hewlett-Packard dan Sun.

Sekarang ini, banyak aplikasi Linux yang dapat digunakan untuk keperluan kantor seperti untuk spreadsheet, word processor, database dan program editor grafis yang memiliki fungsi dan tampilan seperti Microsoft Office, yaitu Star Office. Selain itu, juga sudah tersedia versi Corel untuk Linux dan aplikasi seperti Matlab yang pada Linux dikenal sebagai Scilab.

Linux bisa didapatkan dalam berbagai distribusi (sering disebut Distro). Distro adalah bundel dari kernel Linux, beserta sistem dasar linux, program instalasi, tools basic, dan program-program lain yang bermanfaat sesuai dengan tujuan pembuatan distro. Ada banyak sekali distro Linux, diantaranya :

□ **RedHat**

Distribusi yang paling populer, minimal di Indonesia. RedHat merupakan distribusi pertama yang instalasi dan pengoperasiannya mudah.

□ **Debian**

Distribusi yang mengutamakan kestabilan dan kehandalan, meskipun mengorbankan aspek kemudahan dan kemutakhiran program. Debian menggunakan .deb dalam paket instalasi programnya.

□ **Slackware**

Merupakan distribusi yang pernah merajai di dunia Linux. Hampir semua dokumentasi Linux disusun berdasarkan Slackware. Dua hal penting dari Slackware adalah bahwa semua isinya (kernel, library ataupun aplikasinya) adalah yang sudah teruji. Sehingga mungkin agak tua tapi yang pasti stabil. Yang kedua karena dia menganjurkan untuk menginstall dari source sehingga setiap program yang kita install teroptimasi dengan sistem kita. Ini alasannya dia tidak mau untuk menggunakan binary RPM dan sampai Slackware 4.0, ia tetap menggunakan libc5 bukan glibc2 seperti yang lain.

□ **SuSE**

Distribusi yang sangat terkenal dengan YaST (Yet another Setup Tools) untuk mengkonfigurasi sistem. SuSE merupakan distribusi pertama dimana instalasinya dapat menggunakan bahasa Indonesia.

□ **Mandrake**

Merupakan varian distro RedHat yang dioptimasi untuk pentium. Kalau komputer kita menggunakan pentium ke atas, umumnya Linux bisa jalan lebih cepat dengan Mandrake.

□ **WinLinux**

Distro yang dirancang untuk diinstall di atas partisi DOS (WIndows). Jadi untuk menjalankannya bisa di-klik dari Windows. WinLinux dibuat seakan-akan merupakan suatu program aplikasi under Windows.

Masih banyak lagi distro linux yang lain, diatas hanyalah sebagian distro yang populer dan sering digunakan.

KEISTIMEWAAN LINUX

Hardware komputer pada hakekatnya tidak dapat berfungsi tanpa adanya Software yang mengaturnya. Software yang paling dekat ke Hardware adalah "Sistem Operasi". Suatu jenis prosessor tentu dapat saja di-"atur" oleh berbagai macam Sistem-Operasi pada saat yang berbeda, asalkan ada orang yang mau menulis program Sistem-Operasi tersebut.

Walaupun pada saat Linux ditulis prosessor Intel 80386 telah dipasarkan dengan Sistem-Operasi DOS, namun DOS belum menggunakan 80386 secara optimal. Intel 80386 sebenarnya dirancang sebagai prosessor yang mampu melakukan "multi-tasking" atau "multi-program" namun DOS hanya mampu melakukan "single-tasking". Linux ketika dirancang memanfaatkan 80386 seoptimal mungkin.

Di sini akan dijelaskan beberapa kelebihan dari sistem operasi Linux/UNIX dibandingkan dengan dengan sistem operasi yang lain. Dan berikut ini adalah beberapa fakta dari hal-hal yang menguntungkan dengan menggunakan program dan file-file Linux/UNIX :

- ❑ Pada dasarnya semua data tersimpan di dalam harddisk walau ada beberapa kondisi dimana data tersimpan di disket. Linux/UNIX memberikan beberapa proses spesial dimana terminal, printer dan device hardware lainnya dapat diakses seperti kita mengakses file yang tersimpan dalam harddisk atau disket.
- ❑ Ketika program dijalankan, program tersebut dijalankan dari harddisk ke dalam RAM dan setelah dijalankan akan dinamakan sebagai proses.
- ❑ Linux/UNIX menyediakan servis untuk membuat, memodifikasi program, proses dan file.
- ❑ Linux/UNIX mendukung struktur file yang bersifat hirarki.
- ❑ Linux/UNIX adalah salah satu sistem operasi yang termasuk ke dalam kelas sistem operasi yang dapat melakukan multitasking. Multitasking sendiri adalah

keadaan dimana suatu sistem operasi dapat melakukan banyak kerjaan pada saat yang bersamaan.

- Selain multitasking, Linux/UNIX juga dapat mendukung multiuser. Yaitu sistem operasi yang pada saat bersamaan dapat digunakan oleh lebih dari satu user yang masuk ke dalam sistem. Bahkan untuk Linux juga mendukung untuk multiconsole dimana pada saat bersamaan di depan komputer langsung tanpa harus melalui jaringan dan memungkinkan lebih dari satu user masuk ke dalam sistem.

APLIKASI LINUX

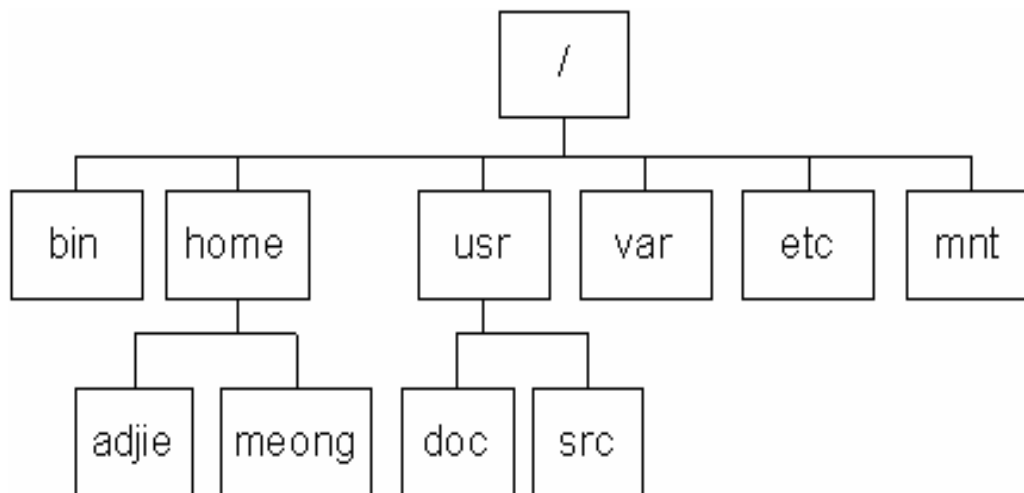
- Sebagai Server
web server, mail server, ftp server, dll.
- Sebagai Client / Workstation pengguna
Aplikasi perkantoran (Star office, yang mirip Microsoft office)
Gimp (image tool seperti Photoshop)
- Koleksi aplikasi di <http://freshmeat.net>, <http://www.sourceforge.net>, dll

DASAR-DASAR LINUX

FILE SYSTEM LINUX

Struktur filesystem Linux adalah berbentuk tree, dengan “pusat”-nya adalah root (akar) yang dilambangkan dengan tanda “/” (slash).

Contoh struktur direktori pada Linux dapat digambarkan sebagai berikut:



PERINTAH - PERINTAH DASAR LINUX

Sintaks

Secara umum perintah pada Linux memiliki sintaks sebagai berikut :

perintah [option....] [argumen....]

- *option* adalah pilihan untuk mendapatkan hasil tertentu
- *argumen* adalah sesuatu yang akan diproses, misalnya file atau direktori

CATATAN :

Perintah dalam Linux adalah Case Sensitive yang berbeda dengan DOS Command. Pada DOS Command perintah 'dir' sama dengan 'DIR', tetapi pada linux 'ls' tidak sama dengan 'LS' atau 'Ls'.

Perintah **ls** (list directory)

Perintah ini akan menampilkan isi sebuah direktori.

Bila Anda ketikkan :

ls

maka akan ditampilkan isi direktori di mana Anda berada (working directory).

Perintah :

ls /home/adjie

akan menampilkan isi direktori /home/adjie

option yang dapat digunakan antara lain

- a menampilkan semua file termasuk yang beratribut hidden, yaitu file atau direktori yang berawalan tanda titik (.)
- l menampilkan file dan direktori dalam tampilan yang lengkap, termasuk nama file, ukuran, tanggal modifikasi, owner, group dan atributnya.

Perintah **cd** (Change Directory)

Perintah ini untuk berpindah direktori.

Perintah :

cd

akan membawa ke home directory Anda.

Perintah :

cd /home/meong

akan membawa Anda ke direktori /home/meong.

Bila terdapat susunan direktori berikut : /usr/src/linux/arch

dan Anda sedang berada di direktori /usr/src/linux, maka bila Anda hendak masuk ke direktori arch, maka ketikkan :

cd arch

sedangkan jika Anda ingin naik satu tingkat lebih atas maka ketikkan :

```
# cd ..
```

Perintah pwd (Print Working Directory)

Perintah ini digunakan untuk melihat di direktori mana Anda sekarang berada.

Perintahnya :

```
# pwd
```

Perintah more

Perintah more dapat Anda gunakan untuk melihat isi suatu file teks dengan layar perlayar, Artinya jika file Anda tampilkan ukurannya lebih dari suatu layar, maka more akan menghentikan tampilannya saat isi file telah mencapai satu layar. Anda dapat melanjutkan ke tampilan berikutnya dengan menekan tombol [SPACEBAR], melihat baris berikutnya dengan tombol [enter], melihat baris sebelumnya dengan tombol b, atau keluar dari tampilan more dengan tombol q. Misalnya:

```
# more /etc/httpd/httpd.conf
```

akan menampilkan isi file /etc/httpd/httpd.conf layar per layar. Indikator (15%) di baris paling bawah manandai posisi (dalam persen) dari seluruh isi file (yaitu file httpd.conf) yang sekarang sedang ditampilkan.

Perintah cat

Perintah ini digunakan untuk menampilkan isi file ke layar tanpa fasilitas tampilan layar per layar. Biasanya digunakan bersamaan dengan pipeline atau redirection.

Misalnya, untuk melihat dan menampilkan isi file /etc/passwd dan /etc/group, gunakan perintah:

```
# cat /etc/passwd /etc/group
```

Perintah rm

Perintah ini digunakan untuk menghapus direktori atau file. Misalnya:

```
# rm data.txt
```

akan menghapus file data1.txt yang terletak pada direktori tempat Anda berada sekarang, asalkan Anda memiliki hak untuk itu.

Perintah berikut akan menghapus file data.txt yang terletak pada direktori /usr/data, asalkan Anda memiliki hak untuk itu.

```
# rm /usr/data/data2.txt
```

Perintah berikut akan menghapus direktori data, yang terletak pada direktori /usr, beserta seluruh isinya.

```
# rm -r /usr/data
```

Perintah **mkdir**

Perintah ini digunakan untuk membuat direktori. Misalnya:

```
# mkdir data
```

akan membuat direktori data pada current directory. Perintah berikut akan membuat direktori januari pada direktori /usr/data, asal direktori data telah terdapat pada direktori /usr.

```
# mkdir /usr/data/januari
```

Jika direktori data belum terdapat pada direktori /usr, Anda dapat menggunakan perintah berikut yang akan secara otomatis membuat semua direktori yang diperlukan dalam rangka membuat direktori januari.

```
# mkdir -p /usr/data/januari
```

Perintah berikut akan secara otomatis membuat direktori januari, februari, dan maret secara sekaligus pada current directory.

```
# mkdir januari februari maret
```

Pipeline

Pada UNIX dan Linux, hasil keluaran suatu proses program dapat diberikan sebagai input pada proses lainnya. Misalnya, jika Anda menampilkan isi direktori /etc dengan `ls -l`, maka hasil tampilannya akan sangat banyak dan Anda tidak sempat membaca nama file yang paling atas. Sementara itu, Anda telah mengenal perintah `more`, yang dapat digunakan untuk menampilkan sesuatu layar per layar.

Dengan pipeline, Anda dapat memberikan keluaran perintah `ls -l` sebagai masukan perintah `more`. Caranya adalah sebagai berikut:

ls -l /etc | more

Tanda vertical bar (|) adalah tanda yang digunakan untuk pipeline. Penggunaan pipeline pada perintah-perintah Linux tidak terbatas. Perhatikan lagi contoh berikut:

ls /etc | sort | more

Program sort adalah program yang akan mensortir inputnya dan menampilkan hasilnya ke layar monitor.

Filter

Dengan menggunakan pipeline, Anda dapat melakukan filter, atau penyaringan hasil proses suatu program untuk ditampilkan sesuai dengan kriteria yang Anda tentukan. Misalnya tampilan layar per layar (dengan pipeline ke more) atau tampilan tersortir (dengan pipeline ke sort).

Ada banyak perintah Linux dan Unix lainnya yang dapat Anda gunakan untuk melakukan penyaringan ini. Beberapa diantaranya adalah:

Perintah grep

Perintah grep digunakan untuk menyaring masukannya dan menampilkan baris-baris yang hanya mengandung pola yang Anda tentukan. Pola ini disebut *regular expression*.

Misalnya, pada Linux terdapat perintah `w` yang digunakan untuk mengetahui siapa saja yang sedang login pada komputer Anda.

Untuk mengetahui siapa saja yang telah login sejak malam dan sore hari, Anda dapat menyaring tampilan perintah `w` sehingga hanya menampilkan baris-baris yang mengandung karakter "PM"

w -h | grep PM

Catatan: Option `-h` akan menghilangkan header tampilan perintah `w`

Perintah `grep` dapat juga digunakan tanpa pipeline, yaitu mengambil inputnya langsung dari file. Misalnya, pada UNIX dan Linux terdapat file `/etc/passwd` yang berisi database user account yang terdapat pada komputer Anda. Jika Anda hendak melihat baris-baris pada file `/etc/passwd` yang mengandung kata `dan` (artinya Anda hendak mengetahui user account di komputer Anda yang namanya `dan`)

mengandung kata dan, misalnya daniel, dani, buldan, dan sebagainya), gunakan perintah:

```
# grep dan /etc/passwd
```

Perintah **wc**

Perintah **wc** dapat Anda gunakan untuk menghitung jumlah baris, kata, dan karakter dari baris-baris masukan yang diberikan kepadanya. Untuk mengetahui berapa baris, gunakan option **-l**, untuk mengetahui berapa kata, digunakan option **-w**, dan untuk mengetahui berapa karakter, gunakan option **-c**. Jika salah satu option itu tidak Anda gunakan, maka tampilannya adalah jumlah baris, jumlah kata, dan jumlah karakter.

Misalnya jika Anda ingin mengetahui berapa jumlah file yang ada pada direktori **/etc**, maka ketikkan perintah:

```
# ls /etc | wc -l
```

Hasilnya adalah :

```
84 753 4908
```

Artinya bahwa hasil tampilan isi direktori **/etc** terdiri dari 84 baris (file dan direktori), 753 kata, dan 4908 karakter. Perhatikan bahwa perintah **ls** umumnya menampilkan hasilnya dalam bentuk beberapa kolom, namun jika Anda menggunakan pipeline, **ls** secara otomatis mengubah tampilannya menjadi satu kolom.

Perintah **wc** dapat juga digunakan tanpa pipeline, yaitu mengambil inputnya langsung dari file.

Jika Anda jalankan perintah:

```
# wc data.txt
```

maka hasilnya adalah

```
3 15 104 data
```

yang artinya adalah bahwa file **data.txt** terdiri atas tiga baris, 15 kata, dan 104 karakter.

Perintah **sort**

Perintah **sort** digunakan untuk mensortir masukannya berdasarkan urutan nomor ASCII dari karakter. Misalnya Anda memiliki file **kelas1.txt** yang isinya sebagai berikut:

Badu

Zulkifli

Yulizir

Yudi

Ade

Maka untuk menampilkan file itu dengan urutan yang beraturan, gunakan perintah

sort kelas1.txt

Misalnya Anda memiliki lagi file kelas2.txt yang isinya sebagai berikut:

Budi

Gama

Asep

Mukhlis

Maka untuk mencetak di printer kedua file yang Anda miliki itu dengan urutan berdasarkan abjad, gunakan perintah:

cat kelas1.txt kelas2.txt | sort | lpr

Catatan: Perintah lpr adalah perintah untuk mencetak inputnya ke langsung printer.

Perintah **cut**

Perintah **cut** digunakan untuk mengambil kolom tertentu dari baris-baris masukannya, yang ditentukan pada option **-c**. Misalnya Anda ingin memproses hasil perintah **who** yang menampilkan informasi user yang sedang login suatu saat.

Seperti terlihat, nama user ditampilkan pada kolom 1-8, dan untuk mengambil kolom ini saja, Anda dapat menggunakan:

who | cut -c1-8

Hasilnya adalah

Daniel

train-01

train-02

Catatan: Istilah kolom yang dimaksud di sini adalah kolom-kolom karakter yang terdapat pada layar monitor atau terminal Anda, biasanya maksimal 80 kolom.

Lihat pada manual (man cut) untuk cara penggunaan perintah ini secara lebih detail.

Perintah **uniq**

Perintah `uniq` digunakan untuk menghilangkan baris-baris berurutan yang mengalami duplikasi, biasanya digabungkan dalam pipeline dengan `sort`. Misalnya jika Anda memiliki file `kelas.txt` dengan isi:

Bambang

Badu

Budi

Ade

Bambang

Ade

Maka untuk menghilangkan baris-baris yang mengalami duplikasi, gunakan perintah:

```
# cat kelas.txt |sort |
```

```
uniq
```

Catatan : Lihat pada manual (man `uniq`) untuk cara penggunaan perintah ini secara lebih detail.

Reguler expression

Reguler expression adalah cara untuk menentukan sebuah pola karakter untuk pencarian dan pemfilteran. Dengan reguler expression, Anda tidak harus menentukan pola karakter yang eksak, misalnya pada perintah `grep`, untuk mencari baris-baris yang mengandung kata `dan`, Anda tinggal menggunakan `grep`. Namun bagaimana jika Anda ingin mencari baris-baris yang mengandung kata yang dimulai dengan karakter `H`, diikuti dengan sembarang karakter, lalu diakhiri dengan karakter `n`? Disinilah dibutuhkan reguler expression.

Untuk menggunakan reguler expression, Anda harus mengenal karakter-karakter khusus yang memiliki arti tersendiri jika digunakan didalam pola reguler expression.

Akses Floppy, Harddisk dan CDROM

Membaca isi disket

Untuk dapat membuka dan mengakses disket, terlebih dahulu dilakukan proses mounting terhadap device file ke sebuah direktori di bawah root direktori.

Misalnya :

```
# mount /dev/fd0 /mnt/floppy
```

(catatan : direktori /mnt/floppy harus sudah ada, bila belum buatlah dengan mkdir)

Setelah itu jika kita akan mengakses floppy tersebut, kita melakukannya lewat direktori /mnt/floppy

Untuk melihat isi disket kita tuliskan :

```
#ls /mnt/floppy
```

Untuk mengkopi file meong.txt ke direktori /home/adjie, ketikkan :

```
#cp /mnt/floppy/meong.txt /home/adjie
```

Untuk disket yang diformat dalam lingkungan DOS atau Windows (FAT12), maka perlu disertakan option jenis filessistem yang digunakan disket, yaitu -t msdos.

```
#mount -t msdos /dev/fd0 /mnt/floppy
```

Jika hendak mengganti disket Anda harus meng-unmount disket yang lama dengan :

```
#umount /mnt/floppy
```

Keluarkan disket lama, masukkan disket baru dan mount lagi.

Membaca isi CDROM

Sama halnya dengan disket untuk membaca CDROM, terlebih dahulu harus dilakukan proses mount.

```
# mount /dev/cdrom /mnt/cdrom
```

Membaca isi partisi DOS dan Windows

Bila harddisk Anda memiliki partisi DOS atau Windows, maka untuk membacanya harus dilakukan juga proses mounting.

Misal partisi itu adalah hda3, dan jenis partisi DOS, maka kita ketikkan :

```
#mount -t msdos /dev/hda3 /mnt/harddisk
```

(asumsi sebelumnya sudah dibuta direktori /mnt/harddisk)

Bila partisi itu berformat Windows maka option yang digunakan adalah -t vfat, jadi kita ketikkan :

```
# mount -t vfat /dev/hda3 /mnt/harddisk
```

Bila partisi itu berformat Windows NT (NTFS=NT File System) maka option yang digunakan adalah -t vfat, jadi kita ketikkan :

```
# mount -t vfat /dev/hda3 /mnt/harddisk
```

File System Linux

File system

Untuk mengorganisasi file-file pada *device* diperlukan suatu metode yang disebut dengan **filesystem**. Jika Anda mengenal **FAT** selama ini di sistem operasi Windows maka Anda akan mengenal beberapa metode *filesystem* di Linux, seperti **ext fs**, **ext2 fs** atau **xia fs** dll. Saat ini *ext2 fs* adalah *filesystem* yang banyak digunakan untuk Linux karena terkenal sangat **efisien**. Meskipun demikian Red Hat Linux tetap menyediakan [dukungan terhadap filesystem lain](#) seperti *msdos* yang sudah *built in* di kernel atau dalam bentuk modul seperti *vfat* (Windows95 *native fs*), *ext*, *umsdos* dan sebagainya.

Mount & Umount Command

Untuk menggunakan *filesystem* tersebut kita lebih dahulu harus me-*mount* sebuah *block device* yang memiliki *filesystem*. Perintahnya adalah sebagai berikut:

```
# mount [-t] [-o] device mount_point
```

device berupa *block device*, *mount_point* berupa sebuah direktori untuk menampilkan *filesystem*. *t* adalah *type* atau jenis *filesystem* dan *o* adalah *option*, keduanya boleh saja tidak disertakan bila Anda sudah mengkonfigurasi file [/etc/fstab](#) yang berisi keterangan detail mengenai *device*, jenis *filesystem*, *mount point* yang digunakan dan sebagainya.

Misalnya saya akan mengakses sebuah file di disket di drive A: maka pertama kali saya harus me-*mount* dulu disk tersebut ke sebuah direktori yang sudah saya buat misalnya */mnt/floppy*:

```
# mount /dev/fd1 -t vfat /mnt/floppy
```

<enter>

```
mount: block device /dev/fd1 is write-protected, mounting read-only
```

Setelah perintah itu barulah saya bisa membaca disket di drive A: tersebut di direktori */mnt/floppy*. Misalnya dengan mengetikkan perintah *ls* maka akan ditampilkan isi disket di drive A:

```
# ls /mnt/floppy
```

Untuk membatalkan perintah *mount* digunakan perintah **umount**

```
# umount /mnt/floppy
```

Setelah perintah tersebut dieksekusi, otomatis drive A: tidak bisa digunakan, cobalah dengan mengetikkan perintah `ls` lagi. Pesan kesalahan akan ditampilkan seperti dibawah ini.

```
# ls /mnt/floppy
```

```
filesystem not mounted
```

mtools

Bila Anda memiliki *filesystem DOS* dan ingin menggunakannya tanpa harus melakukan *mount* maka gunakan **mtools**. Dengan cara ini Anda tidak perlu lagi melakukan mount bila ingin mengakses *filesystem DOS* dan partisi yang belum diformat sekalipun. (Syafrudin, haydin@arjuna.csc.ui.ac.id). Setelah menginstalasi *mtools* tersebut, Anda cukup menjalankan perintah-perintah seperti di DOS untuk mengakses *filesystem DOS* tersebut, misalnya **mdir**, **mcopy** dan sebagainya.

Filesystem Manager

Pada saat *boot*, kernel akan me-*mount root filesystem* dari *device* yang telah ditentukan dalam LILO. Selama proses *boot* tersebut semua *filesystem* yang ada dalam tabel *filesystem* atau dalam file `/etc/fstab` juga akan diperiksa apakah perlu di *mount* atau tidak.

Red Hat telah menyediakan program yang sangat mudah digunakan untuk mengkonfigurasi *filesystem* tersebut. Program ini berjalan di X dan dapat dipanggil melalui **control panel** atau cukup mengeksekusinya dengan perintah **fstool**. Selanjutnya hasil konfigurasinya juga akan disimpan dalam file `/etc/fstab`.

Bila Anda tidak menginstalasi X dan tidak bisa menjalankan *fstool*, jangan khawatir file `/etc/fstab` juga dapat di [edit secara manual](#) dengan program teks editor favorit Anda. *Filesystem manager* menampilkan informasi mengenai nama *device*, *mount point*, jenis *filesystem*, ukuran dan sisa spasi yang masih tersedia. (Informasi ini seperti ini juga bisa diperoleh dengan perintah **df**).

Filesystem dapat di *mount* atau di *unmount* dengan tombol **Mount** dan **Unmount**. Tanda asterik (*) didepan jenis *filesystem* menandakan *filesystem* tersebut sudah atau sedang di *mount*.

Tombol **Format** hanya berfungsi untuk *partisi hard disk*. Tombol ini akan menjalankan program **mkfs** yaitu program untuk membuat *filesystem*.

Tombol **Check** hanya berfungsi untuk *filesystem ext2* dan *minix*. Tombol ini akan menjalankan program **fsck** dan membutuhkan waktu beberapa saat untuk menampilkan hasil pemeriksaan. Bila *filesystem* digunakan secara normal dan *fsck* selalu dijalankan pada saat *boot* maka hasil *check* seringkali tidak menunjukkan kesalahan apapun.

Mengedit file /etc/fstab

File */etc/fstab* berupa file teks biasa, bisa dibaca dan mudah di edit dengan program editor teks favorit kita. Lakukan secara hati-hati karena kesalahan dalam memasukkan atau menghapus sesuatu menyebabkan sistim tidak bisa *me-mount filesystem* saat *boot* nanti.

Mirip dengan tampilan program *fstool* diatas, saat dibuka file */etc/fstab* menampilkan kolom-kolom yang berisi informasi nama *device*, *mount point*, jenis *filesystem*, option, dump dan *fsckorder*. Kolom-kolom tersebut masing-masing dipisahkan dengan tabs atau spasi.

Kolom dump berhubungan dengan program **dump**, lihat manual page program *dumb*. Kolom terakhir *fsckorder* berhubungan dengan program **fsck**. Urutan pengecekan *filesystem* saat *boot* ditunjukkan oleh nomor dikolom ini. Pengecekan *filesystem* dapat dilakukan secara paralel dengan memberikan nomor urut yang sama tapi *filesystem root* selalu harus nomor 1. *Filesystem* dengan *fsckorder* 0 tidak akan dicek saat *boot*.

Filesystem Support

Linux memiliki dukungan terhadap beberapa *filesystem* lain sehingga kita dapat menggunakan atau mengakses *filesystem* yang berbeda tanpa harus melakukan konversi lebih dulu. Berikut ini adalah beberapa *filesystem* yang bisa di dukung Linux sejak kernel 2.0.30 di keluarkan: (Anda dapat mengkonfigurasi dukungan *filesystem* ini saat [konfigurasi kernel](#))

Minix	Merupakan <i>filesystem</i> Linux yang pertama dan saat ini masih banyak digunakan untuk <i>boot disk</i> dan beberapa format disket.
Extended fs	Tidak banyak yang menggunakannya lagi dan sebaiknya tidak perlu di kompilasi dalam kernel.
Second Extended fs	Saat ini merupakan <i>filesystem</i> default untuk Linux dan seharusnya di kompilasi dalam kernel karena <i>filesystem root</i> tidak bisa bekerja bila berupa modul.
xiafs filesystem	Diperkenalkan bersamaan dengan second extended fs dan dimaksudkan untuk menggantikan extended fs. Saat ini jarang digunakan dan sebaiknya tidak di kompilasi dalam kernel kecuali Anda membutuhkannya.
DOS FAT fs	Pada dasarnya bukan merupakan sebuah <i>filesystem</i> tapi merupakan dasar bagi <i>filesystem</i> berbasis FAT lainnya seperti MS-DOS FAT, VFAT (Windows95) atau umsdos.
MS-DOS FAT fs	Jika Anda menginginkan Linux dapat mengakses sistem berbasis DOS maka Anda dapat mengkompilasi dalam kernel. Sangat berguna untuk komputer yang memiliki sistem <i>dual-boot</i> .
VFAT (Windows95) fs	Merupakan peningkatan dari MS-DOS FAT fs dan mendukung format <i>long filename</i> . Ini juga berguna untuk komputer yang memiliki sistem <i>dual-boot</i> .
umsdos	Dukungan ini dibutuhkan bila ingin menjalankan Linux diatas partisi DOS tapi sepertinya SuSE Linux tidak akan bisa berjalan pada sebuah <i>filesystem</i> umsdos.
/proc	<i>Filesystem</i> ini dipakai oleh kernel untuk menyediakan informasi mengenai sistem kepada <i>user</i> program, seperti ps , top , xload , free atau netstat . <i>Filesystem</i> ini memang seharusnya selalu ada dan walaupun ada isinya tapi tidak akan mengisi spasi hard disk.
NFS	Dibutuhkan bila Anda akan mengakses <i>remote filesystem</i> dan untuk sebuah <i>server</i> dukungan ini merupakan suatu keharusan.
SMB	Sangat berguna bila Anda menginginkan memiliki akses langsung ke Windows95 atau NT.

NCP	Dukungan terhadap NetWare melalui protokol NCP.
ISO9660	Bila Anda memiliki CD-ROM sebaiknya memanfaatkan dukungan ini karena banyak sekali CD-ROM yang ada dipasaran menggunakan format ISO9660 ini.
OS/2 HPFS	Mendukung <i>filesystem</i> OS/2 dan HPFS. Linux hanya dapat membaca <i>filesystem</i> tersebut (<i>read-only</i>).
System V & Coherent	Merupakan <i>filesystem</i> UNIX generasi lama dan dukungan ini hanya diperlukan bila Anda memiliki data-data lama dan ingin memindahkannya ke <i>filesystem</i> yang baru.
Amiga FFS	Masih berupa <i>experimental code</i> dan sebaiknya gunakan dengan hati-hati.
UFS	<i>Filesystem</i> ini digunakan oleh beberapa sistem UNIX, Solaris dan SunOS(4.2). Linux hanya dapat membaca <i>filesystem</i> ini.

Manajemen Sistem

Organisasi File

Disamping memiliki program instalasi yang baik, distribusi SuSE juga mengorganisasi file-file yang diinstalasi dengan baik pula. SuSE mengikuti standar pengorganisasian *filesystem* Linux atau **FSSTND** yang tersedia di *website* <http://www.pathname.com/fhs/>

Keterangan lebih lengkap tentu saja dapat diperoleh di *website* tersebut tapi secara sederhana organisasi file di SuSE dapat dijelaskan sebagai berikut:

Pada tingkat tertinggi adalah **root direktori**, /, yang hanya mengandung sejumlah file penting seperti:

bin/, boot/, dev/, etc/, home/, lib/, lost+found/, mnt/, proc/, sbin/, tmp/, usr/, dan var/.

/bin & /sbin

Direktori ini menyimpan program-program penting yang digunakan untuk pemeliharaan sistem. Sesuai dengan nama direktorinya, file-file program yang tersimpan didalamnya berupa file *binary* yang dapat dieksekusi (*executable*).

Isi direktori /bin umumnya adalah user program seperti:

- login
- Shell (bash, ksh, csh)
- File utility (cp, mv, rm, ln, tar)
- Editor (ed, vi)
- Filesystem utility (dd, df, mount, umount, sync)
- System utility (uname, hostname, arch)
- GNU utility (gzip, gunzip)

Isi direktori /sbin umumnya adalah program pemeliharaan atau sistem program. Program-program yang disimpan di direktori /sbin ini hanya dapat dieksekusi oleh **root**. Contohnya adalah sebagai berikut:

- fsck
- fdisk
- mkfs
- shutdown
- lilo
- init

/etc

Direktori ini menyimpan file-file konfigurasi *systemwide* yang dibutuhkan oleh program-program lainnya. Beberapa file penting di direktori ini misalnya:

- passwd
- shadow
- fstab
- hosts
- motd
- profile
- shells
- services
- lilo.conf

/home

Direktori ini menyimpan direktori masing-masing user yang ada di sistem termasuk HTTPD. Beberapa sistem Linux menyimpan home user root di direktori ini sebagai /home/root tapi ada juga yang menyimpan direktori home user root di direktori / atau root direktori.

/mnt

Umumnya direktori ini didalamnya mengandung subdirektori-subdirektori yang merupakan sebuah *mount point* untuk jenis *device* tertentu. Misalnya cdrom/, floppy/, atau zip/.

/tmp & /var

Direktori /tmp untuk menampung file-file sementara(temporary) dan /var menampung varying content atau macam-macam file.

Direktori /tmp biasanya juga dimanfaatkan oleh program instalasi saat kita menginstalasi program atau aplikasi. Isi direktori ini bisa saja dihapus setiap saat tanpa menimbulkan penagaruh apapun.

Direktori /var memiliki isi yang lebih banyak dari /tmp dan biasanya isinya adalah seperti ini:

catman/, lib/, local/, lock/, log/, nis/, preserve/, run/, spool/ dan tmp/.

Direktori /var/log merupakan direktori yang sangat familiar bagi setiap user. Di dalamnya disimpan pesan-pesan yang dihasilkan oleh sistem. Berikut ini isi direktori /var/log di komputer saya:

httpd/, boot.log, cron, dmesg, htmlaccess.log, lastlog, maillog, messages, netconf.log, secure, sendmail.st, spooler, wtmp. File-file pesan tersebut sangat berguna bagi kita untuk mendiagnosis masalah atau kesalahan sistem. Bagi yang tidak berpengalaman dalam menangani masalah sistem, isi file-file pesan tersebut bisa disertakan bila kita minta bantuan orang lain atau di forum milis.

/usr

Berisi semua program dan file yang secara langsung berhubungan dengan setiap user dalam sistem. Misalnya di komputer saya isinya adalah sebagai berikut:

~X11, X11R6/, ~X386, bin/, dict/, doc/, etc/, games/, i486-linux-libc5/, i486-linuxaout/, include/, info/, lib/, libexec/, local/, man/, sbin/, share/, src/, ~tmp.

Direktori `/usr/bin` dan `/usr/sbin` menyimpan banyak sekali file *executable*. File-file *executable* yang disimpan di direktori ini memiliki kesamaan fungsi dan jenis dengan file-file di direktori `/bin` dan `/sbin`.

Direktori `/usr/X11` dan `/usr/X11R6` dan subdirektori dibawahnya berisi semua file yang berhubungan dengan X Window seperti *man page*, *library* dan file *executable*.

Menganalisis Sistem

Menganalisis sistem biasanya dilakukan oleh seorang administrator sistem untuk mendeteksi adanya gejala-gejala yang dapat menyebabkan kerusakan atau masalah pada sistem kelak. SuSE sudah menyediakan *tool-tool* untuk melakukan hal itu, diantaranya adalah dua buah program *tool* yang sangat berguna yaitu **vmstat** dan **top**, dengan menggunakan kedua program tersebut administrator memperoleh informasi mengenai sistem seperti penggunaan CPU, memori atau proses-proses yang sedang berjalan dan dengan bantuan informasi ini administrator sistem bisa mendeteksi secara cepat apakah sistemnya mengalami gangguan atau tidak.

Dengan menggunakan kedua program tersebut, Anda sendiri bisa mencoba dan melihat apakah sistem yang dipakai saat ini berjalan dengan baik atau tidak. Lihatlah informasi yang ditampilkan program-program tersebut dan bila Anda memperkirakan ada masalah, silakan menghubungi administrator sistem untuk mendapatkan bantuan.

vmstat

Program atau perintah ini akan melakukan test dan menampilkan informasi penggunaan CPU, memori, proses-proses yang sedang berjalan serta operasi I/O. Sintaks penulisan perintahnya adalah sebagai berikut:

```
$ vmstat [interval] [count]
```

Interval adalah waktu jeda test dalam detik dan *count* adalah jumlah test yang kita kehendaki. Misalnya akan dilakukan test sebanyak 5 kali dengan waktu jeda setiap 5 detik maka perintah dan hasilnya adalah seperti dibawah ini:

```
[zakaria@linux source]$ vmstat 5 5
```

```
procs          memory swap      io  system      cpu
r  b  w  swpd free  buff cache si so  bi  bo  in  cs  us  sy  id
0  0  0  37924  824  840 13280 54 14  30  9 522  271 21  6 73
3  0  0  37924 1080  816 13060  0  0   1  0 110  329 20 16 64
1  0  0  37924 1036  824 13084  0  0   2  1 130  331 18 18 64
1  0  0  37924 1036  824 13084  0  0   0  0 107  329 19 16 65
2  0  0  37924 1036  828 13084  0  0   0  1 112  326 21 14 65
```

Tiga kolom pertama menunjukkan proses-proses yang sedang dikerjakan, yaitu *waiting for runtime*(r), *uninterrupted sleep*(b) dan *swapped out*(w). Banyaknya proses yang mengalami *waiting for runtime*(r) dapat merupakan indikasi terjadinya suatu masalah misalnya telah terjadi *bottleneck* yaitu penumpukan proses-proses disuatu tempat.

Kolom *memory* dan *swap* digunakan untuk mendeteksi adanya kesalahan yang ditimbulkan oleh manajemen memori. Kolom-kolom **swpd**, **free**, **buff** dan **cache** berturut-turut menjelaskan jumlah memori *virtual* yang digunakan, jumlah memori *idle*, jumlah memori yang dipakai sebagai *buffer* dan jumlah memori yang tersisa dalam *cache*, semuanya dalam KB. Perhatikan kolom *swap in*(si), merupakan jumlah memori yang di-*paging* dari disk dalam satuan KB/detik sedangkan *swap out*(so) adalah kebalikannya.

Tiga kolom terakhir adalah persentasi penggunaan CPU yaitu persentasi penggunaan CPU untuk tugas-tugas *user*(us), persentasi penggunaan CPU untuk tugas-tugas sistem termasuk waktu tunggu I/O, pelaksanaan fungsi-fungsi sistem operasi secara umum(sy) dan persentasi CPU *idle* atau saat tidak digunakan(id). Untuk mendeteksi suatu kesalahan yang diakibatkan oleh masalah CPU sebaiknya dengan melihat persentasinya secara signifikan dalam satu periode waktu. Misalnya pada saat komputer *idle* dalam satu periode waktu, perhatikan kolom **us** dan **sy** seharusnya menunjukkan angka yang rendah sedangkan kolom **id** menunjukkan angka yang tinggi. Bila tidak seperti itu maka dipastikan ada masalah yang berkaitan dengan CPU.

top

Menampilkan informasi proses yang sedang dilakukan CPU secara real time. Kelebihan dari top ini adalah kita dapat memanipulasi proses-proses tersebut secara interaktif dengan menekan tombol-tombol menu. Misalnya untuk meng-kill sebuah proses, kita cukup menekan tombol K dan mengisi nomor proses(PID) yang ingin di-kill.

Untuk menjalankannya cukup dengan mengetikkan perintah top dan segera ditampilkan daftar proses-proses yang sedang berjalan saat itu berikut informasi-informasi lainnya. Informasi-informasi yang ditampilkan tersebut secara default akan diperbarui atau diupdate setiap 5 detik. Berikut ini tampilan sebagian dari daftar proses-proses yang ditunjukkan oleh top:

```
2:10pm up 21 min, 2 users, load average: 1.02, 0.50, 0.26
35 processes: 30 sleeping, 4 running, 1 zombie, 0 stopped
CPU states: 7.8% user, 6.4% system, 0.0% nice, 86.2% idle
Mem: 14580K av, 14176K used, 404K free, 8664K shrd, 324K buff
Swap: 34268K av, 17600K used, 16668K free 5268K cached

```

PID	USER	PRI	NI	SIZE	RSS	SHARE	STAT	LIB	%CPU	%MEM	TIME	COMMAND
321	root	16	0	4584	3184	960	R	0	7.0	21.8	1:00	X
4	root	1	0	0	0	0	SW	0	0.1	0.0	0:00	kswapd
1	root	0	0	64	0	0	SW	0	0.0	0.0	0:04	init
2	root	0	0	0	0	0	SW	0	0.0	0.0	0:00	kflushd
3	root	0	0	0	0	0	SW	0	0.0	0.0	0:00	kpid
141	root	0	0	252	200	168	S	0	0.0	1.3	0:00	syslogd
152	root	0	0	372	0	0	SW	0	0.0	0.0	0:01	klogd
180	root	0	0	152	92	76	S	0	0.0	0.6	0:00	crond
194	root	0	0	100	0	0	SW	0	0.0	0.0	0:00	inetd
210	root	0	0	96	0	0	SW	0	0.0	0.0	0:00	lpd
239	root	0	0	464	268	224	S	0	0.0	1.8	0:00	sendmail
254	root	0	0	92	52	32	S	0	0.0	0.3	0:00	gpm
268	root	0	0	396	60	48	S	0	0.0	0.4	0:00	xf
293	root	0	0	252	0	0	SW	0	0.0	0.0	0:00	login
294	root	0	0	60	0	0	SW	0	0.0	0.0	0:00	mingetty

Baris-baris *header* menampilkan informasi mengenai waktu, jumlah *user*, banyaknya proses, penggunaan CPU serta memori secara jelas. Misalnya saat itu sistem memiliki 35 buah proses, dimana 4 proses sedang berjalan, 30 proses *sleeping* dan ada sebuah proses menjadi *zombie*. Proses *zombie* ini adalah sebuah proses yang *hang* atau hidup segan, mati tak mau dan apabila Anda tidak bisa mem-kill proses *zombie* ini maka **parent process**-nya harus di-kill lebih dahulu. Gunakan perintah **pstree** atau **ps ufx** untuk mengurut proses mana yang menjadi *parent* dari *zombie* tersebut.

Daftar menu untuk mengoperasikan top dapat dilihat dengan menekan tombol **?**, atau **h**. Dengan menggunakan tombol-tombol dalam daftar menu tersebut, kita dapat melakukan manipulasi semua proses dalam sistem. Misalnya, tombol **k** untuk

meng-kill sebuah proses, tombol **r** untuk me-*renice* proses dan **q** untuk keluar atau mengakhiri program top.

Multi User

Bagaimana Tidak Menggunakan Root?

Bila Anda membaca beberapa bab yang lalu, pernah disinggung mengenai kerugian-kerugian apabila kita selalu bekerja sebagai **root**. Tapi kemudian saya memperoleh beberapa pertanyaan seperti ini: Apakah saya masih bisa menggunakan hak-hak khusus *root* setelah saya tidak memakainya? Mengapa saya tidak bisa lagi melakukan koneksi *dial-up* setelah saya *login* bukan sebagai *root*?

Seperti yang telah kita ketahui, Linux termasuk sistem *multiuser* dimana suatu *resource* bisa digunakan oleh banyak *user*. Setiap *user* biasanya diberi ruangan atau *space* yang di simpan rapi dibawah direktori **/home**. Setiap *user* di *home* masing-masing memiliki hak mengakses, membaca atau menulis file-file di dalam *home* mereka sendiri tetapi mereka belum tentu bisa melakukan hal yang sama di *home* milik *user* lain atau direktori milik *root*. Masing-masing *user* bisa diberi hak-hak khusus yang berlainan untuk mengakses, membaca atau menulis ke sebuah file atau direktori oleh *root*. Oleh karena itu kita bisa saja meninggalkan *root* atau tidak lagi login sebagai *root* sepanjang hari dengan cara membuat *home* sendiri, *login* sebagai *user* biasa serta memberikan hak akses seperlunya saja agar tidak membahayakan sistem bila suatu saat kita melakukan kesalahan.

Membuat rumah untuk saya, mama, keponakan atau teman saya

Mudah saja, saat Anda login sebagai *root*, jalankan perintah sebagai berikut:

```
# adduser meong
```

```
# passwd meong
```

Setelah mengetik perintah terakhir Anda akan ditanya *password* untuk membuka *home* Anda. Ketikkan saja *password*nya sebanyak dua kali dengan kata yang sama. Selanjutnya Anda telah memiliki 'rumah' baru dan siap untuk digunakan. Bila Anda menginginkan, Anda juga bisa memberikan *home* lain kepada mama, keponakan

yang masih duduk TK atau kepada teman kuliah Anda yang sedang menumpang mengetik skripsinya di komputer Anda.

Group

Setiap *user* paling sedikit bergabung dengan sebuah **group**. *Group* bisa berisi kumpulan *user* lain atau program yang mempunyai kesamaan tugas. *Group* memungkinkan sebuah file bisa dipergunakan secara bersama hanya oleh *user-user* yang tergabung didalamnya. Oleh karena itu cara mengelompokkan *user-user* dalam group ini adalah salah satu cara yang mudah bagi *root* untuk memberikan hak akses file-file miliknya kepada sekelompok *user*.

Untuk membuat group baru, Anda bisa menggunakan perintah **groupadd**. Misalnya, Anda ingin membuat group baru yang namanya **konek** maka perintahnya adalah sbb:

```
# groupadd konek
```

Untuk parameter yang lain seperti menentukan *gid*, *password* dan lain-lain silakan lihat manualnya. Berikutnya adalah menambahkan *user-user* yang akan bergabung ke dalam group konek ini. Informasi group disimpan dalam file **/etc/group**, bukalah dengan menggunakan editor kesayangan Anda, kemudian tambahkan nama-nama *user* yang akan bergabung dalam group konek.

Setiap baris dalam file **/etc/group** terdiri dari empat segmen yang dipisahkan oleh tanda titik dua,

```
nama group : password : group id(gid) : user
```

Carilah baris group konek dan cukup tambahkan nama *user* yang akan bergabung dengan group konek ini di segmen terakhir. Pisahkan nama *user* dengan tanda koma bila *user* yang bergabung lebih dari satu, misalnya:

```
konek : : 501 : meong, fryda
```

Password biasanya kosong atau * atau biarkan saja bila Anda tidak membuat *password* untuk group ini. Setelah file **/etc/group** ini di simpan maka tugas berikutnya adalah merubah *permission* dan *ownership* file-file yang bisa diakses oleh group konek.

Permission dan Ownership

Setiap file memiliki perijinan dan kepemilikan yang menentukan siapa saja yang boleh mengaksesnya. Lihatlah kembali bab mengenai [command line](#) khususnya perintah **chgrp**, **chmod** dan **chown** atau lihat manual perintah-perintah tersebut untuk lebih memahami fungsi perijinan dan kepemilikan ini.

Sebagai contoh, bila Anda menginginkan group konek yang telah dibuat beberapa saat lalu itu, semua anggotanya bisa melakukan *dial-up* sendiri maka Anda harus merubah perijinan atau kepemilikan dari beberapa file yang berhubungan dengan *dial-up* supaya bisa diakses oleh group konek.

Bila Anda memakai kppp dari KDE sebagai *dial-up* maka Anda bisa merubah kepemilikan file kppp menjadi milik group konek seperti ini:

```
# ls -l /usr/bin/kppp
-rwx----- 1 root root 365660 Apr 19 00:33 /usr/bin/kppp
# chown .konek /usr/bin/kppp
# ls -l /usr/bin/kppp
-rwsrws--- 1 root konek 365660 Apr 19 00:33 /usr/bin/kppp
```

Lakukanlah hal yang sama pada file-file yang berhubungan dengan *dial-up* seperti */dev/modem*, file-file di */etc/ppp/option* dan sebagainya.

Setelah itu *user-user* yang tergabung dalam group konek diharuskan *login* ke group konek dengan mengetikkan perintah **newgrp** apabila ingin dapat melakukan *dial-up* dengan kppp:

```
$ newgrp konek
```

Sekarang *user fryda* sudah dapat melakukan *dial-up* sendiri dengan kppp tanpa bantuan *root*.

Command Line

Seperti halnya bila kita mengetikkan perintah di DOS, *command line* atau baris perintah di Linux juga diketikkan di *prompt* dan diakhiri *enter* untuk mengeksekusi perintah tersebut. Baris perintah merupakan cara yang lebih efisien untuk melakukan sesuatu pekerjaan oleh karena itu pemakai Linux tetap mengandalkan cara ini untuk bekerja. Sebaiknya pemula juga harus mengetahui dan sedikitnya

pernah menggunakan perintah baris ini karena suatu saat pengetahuan akan perintah-perintah ini bisa sangat diperlukan.

Saya mengumpulkan beberapa perintah dasar yang mungkin kelak akan sering digunakan terutama oleh para pemula. Perhatian: pengetahuan akan perintah-perintah yang lain akan segera bertambah seiring dengan kemajuan Anda menguasai sistem operasi Linux ini.

Penjelasan masing-masing perintah akan dipersingkat saja dan untuk mengetahui lebih detail lagi fungsi-fungsi suatu perintah, Anda dapat melihat manualnya, misalnya dengan mengetikkan perintah [man](#):

```
$ man ls
```

Manual tersebut akan menampilkan bagaimana cara penggunaan perintah [ls](#) itu secara lengkap.

Daftar Perintah Menurut Alfabet

&	adduser	alias	bg	cat	cd	chgrp	chmod	chown
cp	fg	find	grep	gzip	halt	hostname	kill	less
login	logout	ls	man	msg	mkdir	more	mount	mv
passwd	pwd	rm	rmdir	shutdown	su	tail	talk	tar
umount	unalias	unzip	wall	who	xhost +	xset	zip	

&

Perintah **&** dipakai dibelakang perintah lain dan menjalankannya di *background*. Tujuannya adalah untuk membebaskan *shell* agar bisa dipergunakan menjalankan proses-proses yang lain. Lihat juga perintah [bg](#) dan [fg](#).

adduser

Biasanya hanya dilakukan oleh root untuk menambahkan *user* atau *account* yg baru. Setelah perintah ini bisa dilanjutkan dengan perintah [passwd](#), yaitu perintah untuk membuat *password* bagi user tersebut.

```
# adduser udin
```

```
# passwd udin
```

Selanjutnya Anda akan diminta memasukkan *password* untuk *user* udin. Isikan

password untuk udin dua kali dengan kata yang sama.

alias

Digunakan untuk memberi nama lain dari sebuah perintah. Misalnya bila Anda ingin perintah **ls** dapat juga dijalankandengan mengetikkan perintah **dir**, maka buatlah aliasnya sbb:

```
$ alias dir=ls
```

Untuk melihat perintah-perintah apa saja yang mempunyai nama lain saat itu, cukup ketikkan alias. Lihat juga perintah **unalias**.

bg

Untuk memaksa sebuah proses yang dihentikan sementara(*suspend*) agar berjalan di *background*. Misalnya Anda sedang menjalankan sebuah perintah di *foreground* (tanpa diakhiri perintah **&**) dan suatu saat Anda membutuhkan *shell* tersebut maka Anda dapat memberhentikan sementara perintah tersebut dengan **Ctrl-Z** kemudian ketikkan perintah **bg** untuk menjalakkannya di *background*. Dengan cara ini Anda telah membebaskan*shell* tapi tetap mempertahankan perintah lama berjalan di *background*. Lihat juga perintah **fg**.

cat

Menampilkan isi dari sebuah file di layar.

```
$ cat namafile
```

cd

Change Directory atau untuk berpindah direktori dan saya kira Anda tidak akan menemui kesulitan menggunakan perintah ini karena cara penggunaanya mirip dengan perintah **cd** di DOS.

chgrp

Perintah ini digunakan untuk merubah kepemilikan kelompok file atau direktori. Misalnya untuk memberi ijin pada kelompok atau grup agar dapat mengakses suatu file. Sintaks penulisannya adalah sbb:

```
# chgrp <grup baru> <file>
```

chmod

Digunakan untuk menambah dan mengurangi ijin pemakai untuk mengakses file

atau direktori. Anda dapat menggunakan sistem **numeric coding** atau sistem **letter coding**. Ada tiga jenis *permission*/perijinan yang dapat dirubah yaitu **r** untuk **read**, **w** untuk **write** dan **x** untuk **execute**.

Dengan menggunakan *letter coding*, Anda dapat merubah *permission* diatas untuk masing-masing **u (user)**, **g (group)**, **o (other)** dan **a (all)** dengan hanya memberi tanda **plus (+)** untuk menambah ijin dan tanda **minus (-)** untuk mencabut ijin.

Misalnya untuk memberikan ijin baca dan eksekusi file coba1 kepada *owner* dan *group*, perintahnya adalah:

```
$ chmod ug+rx coba1
```

Untuk mencabut ijin-ijin tersebut:

```
$ chmod ug-rx coba1
```

Dengan menggunakan sitem *numeric coding*, *permission* untuk *user*, *group* dan *other* ditentukan dengan menggunakan kombinasi angka-angka, 4, 2 dan 1 dimana **4 (read)**, **2 (write)** dan **1 (execute)**.

Misalnya untuk memberikan ijin baca(4), tulis(2) dan eksekusi(1) file coba2 kepada *owner*, perintahnya adalah:

```
$ chmod 700 coba2
```

Contoh lain, untuk memberi ijin baca(4) dan tulis(2) file coba3 kepada *user*, baca(4) saja kepada *group* dan *other*, perintahnya adalah:

```
$ chmod 644 coba3
```

chown

Merubah user ID (*owner*) sebuah file atau direktori

```
$ chown <user id> <file>
```

cp

Untuk menyalin file atau *copy*. Misalnya untuk menyalin file1 menjadi file2:

```
$ cp <file1> <file2>
```

fg

Mengembalikan suatu proses yang dihentikan sementara(*suspend*) agar berjalan kembali di *foreground*. Lihat juga perintah [bg](#) diatas.

find

Untuk menemukan dimana letak sebuah file. Perintah ini akan mencari file sesuai

dengan kriteria yang Anda tentukan. Sintaksnya adalah perintah itu sendiri diikuti dengan nama direktori awal pencarian, kemudian nama file (bisa menggunakan *wildcard*, *metacharacters*) dan terakhir menentukan bagaimana hasil pencarian itu akan ditampilkan. Misalnya akan dicari semua file yang berakhiran *.doc* di *current* direktori serta tampilkan hasilnya di layar:

```
$ find . -name *.doc -print
./public/docs/account.doc
./public/docs/balance.doc
./public/docs/statistik/prospek.doc
./public/docs/statistik/presconf.doc
```

grep

Global regular expression parse atau *grep* adalah perintah untuk mencari file-file yang mengandung teks dengan kriteria yang telah Anda tentukan.

```
$ grep <teks> <file>
```

Misalnya akan dicari file-file yang mengandung teks **marginal** di *current* direktori:

```
$ grep marginal <file>
```

diferent.doc: Catatan: perkataan marginal luas dipergunakan di dalam ilmu ekonomi
prob.rtf: oleh fungsi hasil marginal dan fungsi biaya marginal jika fungsi
prob.rtf: jika biaya marginal dan hasil marginal diketahui maka biaya total

gzip

Ini adalah software kompresi [zip](#) versi GNU, fungsinya untuk mengkompresi sebuah file. Sintaksnya sangat sederhana:

```
$ gzip <namafile>
```

Walaupun demikian Anda bisa memberikan parameter tertentu bila memerlukan kompresi file yang lebih baik, silakan melihat *manual page*-nya. Lihat juga file [tar](#), [unzip](#) dan [zip](#).

halt

Perintah ini hanya bisa dijalankan oleh [super user](#) atau Anda harus [login](#) sebagai *root*. Perintah ini untuk memberitahu kernel supaya mematikan sistem atau [shutdown](#).

hostname

Untuk menampilkan host atau domain name sistem dan bisa pula digunakan untuk mengeset nama host sistem.

```
[meong@localhost docs] $ hostname
```

```
localhost.localdomain
```

kill

Perintah ini akan mengirimkan sinyal ke sebuah proses yang kita tentukan. Tujuannya adalah menghentikan proses.

```
$ kill <sinyal> <pid>
```

PID adalah nomor proses yang akan di hentikan.

less

Fungsinya seperti perintah [more](#).

login

Untuk masuk ke sistem dengan memasukkan *login* ID atau dapat juga digunakan untuk berpindah dari *user* satu ke *user* lainnya.

logout

Untuk keluar dari sistem.

ls

Menampilkan isi dari sebuah direktori seperti perintah **dir** di DOS. Anda dapat menggunakan beberapa option yang disediakan untuk mengatur tampilannya di layar. Bila Anda menjalankan perintah ini tanpa option maka akan ditampilkan seluruh file *nonhidden*(file tanpa awalan tanda titik) secara alfabet dan secara melebar mengisi kolom layar. Option *-la* artinya menampilkan seluruh file/all termasuk file *hidden*(file dengan awalan tanda titik) dengan format panjang.

man

Untuk menampilkan *manual page* atau teks yang menjelaskan secara detail bagaimana cara penggunaan sebuah perintah. Perintah ini berguna sekali bila sewaktu-waktu Anda lupa atau tidak mengetahui fungsi dan cara menggunakan sebuah perintah.

```
$ man <perintah>
```


mesg

Perintah ini digunakan oleh user untuk memberikan ijin user lain menampilkan pesan dilayar terminal. Misalnya mesg Anda dalam posisi y maka user lain bisa menampilkan pesan di layar Anda dengan **write** atau **talk**.

\$ mesg y atau mesg n

Gunakan mesg n bila Anda tidak ingin diganggu dengan tampilan pesan-pesan dari *user* lain.

mkdir

Membuat direktori baru, sama dengan perintah **md** di DOS.

more

Mempaging halaman, seperti halnya [less](#)

mount

Perintah ini akan me-*mount filesystem* ke suatu direktori atau *mount-point* yang telah ditentukan. Hanya *superuser* yang bisa menjalankan perintah ini. Untuk melihat *filesystem* apa saja beserta *mount-point*nya saat itu, ketikkan perintah *mount*. Perintah ini dapat Anda pelajari di bab mengenai [filesystem](#). Lihat juga perintah [umount](#).

\$ mount

/dev/hda3 on / type ext2 (rw)

none on /proc type proc (rw)

/dev/hda1 on /dos type vfat (rw)

/dev/hda4 on /usr type ext2 (rw)

none on /dev/pts type devpts (rw,mode=0622)

mv

Untuk memindahkan file dari satu lokasi ke lokasi yang lain. Bila argumen yang kedua berupa sebuah direktori maka mv akan memindahkan file ke direktori tersebut. Bila kedua argumen berupa file maka nama file pertama akan menimpa file kedua. Akan terjadi kesalahan bila Anda memasukkan lebih dari dua argumen kecuali argumen terakhir berupa sebuah direktori.

passwd

Digunakan untuk mengganti *password*. Anda akan selalu diminta mengisi

password lama dan selanjutnya akan diminta mengisi *password* baru sebanyak dua kali. *Password* sedikitnya terdiri dari enam karakter dan sedikitnya mengandung sebuah karakter.

pwd

Menampilkan nama direktori dimana Anda saat itu sedang berada.

rm

Untuk menghapus file dan secara default **rm** tidak menghapus direktori. Gunakan secara hati-hati perintah ini terutama dengan option **-r** yang secara rekursif dapat menghapus seluruh file.

rmdir

Untuk menghapus direktori kosong.

shutdown

Perintah ini untuk mematikan sistem, seperti perintah [halt](#). Pada beberapa sistem anda bisa menghentikan komputer dengan perintah **shutdown -h now** dan merestart sistem dengan perintah **shutdown -r now** atau dengan kombinasi tombol **Ctrl-Alt-Del**.

su

Untuk *login* sementara sebagai *user* lain. Bila *user* ID tidak disertakan maka komputer menganggap Anda ingin *login* sementara sebagai *super user* atau *root*. Bila Anda bukan *root* dan *user* lain itu memiliki *password* maka Anda harus memasukkan *passwordnya* dengan benar. Tapi bila Anda adalah *root* maka Anda dapat *login* sebagai *user* lain tanpa perlu mengetahui *password user* tersebut.

tail

Menampilkan 10 baris terakhir dari suatu file. Default baris yang ditampilkan adalah 10 tapi Anda bisa menentukan sendiri berapa baris yang ingin ditampilkan:

```
$ tail <jumlah baris> <file file ....>
```

talk

Untuk mengadakan percakapan melalui terminal. Input dari terminal Anda akan disalin di terminal *user* lain, begitu sebaliknya.

tar

Menyimpan dan mengekstrak file dari media seperti tape drive atau hard disk. File

arsip tersebut sering disebut sebagai file tar. Sintaknya sebagai berikut:

```
$ tar <aksi> <option> <file atau direktori>
```

umount

Adalah kebalikan dari perintah [mount](#), yaitu untuk meng-*umount filesystem* dari *mount-point*nya. Setelah perintah ini dijalankan direktori yang menjadi *mount-point* tidak lagi bisa digunakan.

```
# umount <filesystem>
```

unalias

Kebalikan dari perintah [alias](#), perintah ini akan membatalkan sebuah alias. Jadi untuk membatalkan alias dir seperti telah dicontohkan [diatas](#), gunakan perintah:

```
$ unalias dir
```

unzip

Digunakan untuk mengekstrak atau menguraikan file yang dikompres dengan [zip](#). Sintaknya sederhana dan akan mengekstrak file yang anda tentukan:

```
$ unzip <namafile>
```

Lihat juga perintah-perintah [gzip](#) dan [unzip](#).

wall

Mengirimkan pesan dan menampilkannya di terminal tiap user yang sedang login. Perintah ini berguna bagi *superuser* atau *root* untuk memberikan peringatan ke seluruh *user*, misalnya pemberitahuan bahwa *server* sesaat lagi akan dimatikan.

```
# who Dear, everyone..... segera simpan pekerjaan kalian, server akan saya matikan 10 menit lagi.
```

who

Untuk menampilkan siapa saja yang sedang *login*. Perintah ini akan menampilkan informasi mengenai *login name*, jenis terminal, waktu *login* dan *remote hostname* untuk setiap *user* yang saat itu sedang *login*. Misalnya:

```
$ who
```

```
root tty0 May 22 11:44
```

```
meong tty2 May 22 11:59
```

```
pooh tty3 May 22 12:08
```

xhost +

Perintah ini digunakan untuk memberi akses atau menghapus akses(xhost -) host atau user ke sebuah *server X*.

xset

Perintah ini untuk mengeset beberapa option di X Window seperti bunyi bel, kecepatan *mouse*, *font*, parameter *screen saver* dan sebagainya. Misalnya bunyi bel dan kecepatan *mouse* dapat Anda set menggunakan perintah ini:

```
$ xset b <volume> <frekuensi> <durasi dalam milidetik>
```

```
$ xset m <akselerasi> <threshold>
```

zip

Perintah ini akan membuat dan menambahkan file ke dalam file arsip zip. Lihat juga perintah [gzip](#) dan [unzip](#).

MAILING LIST LINUX

Milis id-linux

Milis 'resmi' yang memakai bahasa Indonesia adalah [id-linux](#) yang beberapa saat lalu terpecah-pecah menjadi beberapa sub topik:

Linux Admin - Membicarakan administrasi sistem dan *networking* Linux.

Subscribe: linux-admin-subscribe@linux.or.id

Linux Aktivist - Membicarakan pengembangan dan pemasyarakatan Linux di Indonesia.

Subscribe: linux-aktivis@linux.or.id

Linux Berita - Memberikan informasi tentang berita dan pengumuman kegiatan Linux di Indonesia khususnya.

Subscribe: linux-berita-subscribe@linux.or.id

Linux Bursa - Jual-beli barang/jasa Linux, penawaran dan pencarian kerja di bidang Linux.

Subscribe: linux-bursa-subscribe@linux.or.id

Linux Desktop - Diskusi cara penggunaan program aplikasi seperti KDE, Gnome, StarOffice dsb.

Subscribe: linux-desktop-subscribe@linux.or.id

Linux Policy - Non teknis Linux misalnya GPL, Open Source.

Subscribe: linux-policy-subscribe@linux.or.id

Linux Programming - Diskusi mengenai pemrograman Linux, bahasa, teknik, trik dan tips.

Subscribe: linux-programming-subscribe@linux.or.id

Linux Setup - Tempat yang menarik bagi *newbie* menanyakan masalah instalasi, cara *setup* program dsb.

Subscribe: linux-setup-subscribe@linux.or.id