

# BAB12: Manajemen Sistem

Oleh: Flory Katriena

## ::: Organisasi File

Disamping memiliki program instalasi yang baik, distribusi RedHat juga mengorganisasi file-file yang diinstalasi dengan baik pula. RedHat 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 RedHat 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

penengaruh 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/, i386-redhat-linux/, 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. RedHat 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:

```
$ 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.

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 (Ronny Haryanto, giant@canada.com) 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.

## ::: 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 flory
# passwd flory
```

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 : flory, 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:

```
[root@cybernetics /root]# ls -l /usr/bin/kppp
-rwx----- 1 root root 365660 Apr 19 00:33
/usr/bin/kppp

[root@cybernetics /root]# chown .konek /usr/bin/kppp

[root@cybernetics /root]# 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 menyetikkan perintah **newgrp** apabila ingin dapat melakukan *dial-up* dengan kppp:

```
[fryda@cybernetics fryda]$ newgrp konek
```

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

### ::: Command Line

Seperti halnya bila kita menyetikkan 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 menyetikkan perintah **man**:

```
$ man ls
```

Manual tersebut akan menampilkan bagaimana cara penggunaan perintah **ls** itu secara lengkap.

---

### &

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 dijalankan dengan 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 penggunaannya 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 expresion 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.

```
[flory@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-pointnya* 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 mengisikan *password* lama dan selanjutnya akan diminta mengisikan *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 ttyp0 May 22 11:44  
flory ttyp2 May 22 11:59  
pooh ttyp3 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*.

Sumber dari [www.linux.or.id](http://www.linux.or.id)