

Perintah Dasar UNIX

Ver 0.1*

Tedy Tirtawidjaja

15 Juli 2007



Tulisan ini berisi beberapa perintah dasar dalam lingkungan Unix (berlaku pula untuk Linux). Apa yang dijelaskan dalam tulisan ini sangat mendasar. Saya sangat menganjurkan Anda untuk membaca juga dokumentasi dari setiap perintah dengan menggunakan perintah `man`.

Beberapa perintah dasar dalam lingkungan Unix dapat digunakan pula dalam sistem operasi Linux. Perintah yang dibahas berikut ini adalah semua perintah yang dapat digunakan dalam terminal/konsol. Memang distribusi Linux saat ini sudah dilengkapi dengan banyak fitur GUI (*graphical user interface*), tapi sangat disarankan pengguna sistem operasi berbasis Unix memahami juga perintah-perintah berbasis teks.

Semua perintah dalam lingkungan Unix sifatnya *case-sensitive* (memperhatikan besar kecilnya huruf). Jadi, perintah `ls` tidak akan berfungsi jika kita salah mengetikkannya menjadi `LS`. Ini berbeda dengan lingkungan Windows, di lingkungan Windows perintah dalam konsol tidak memperhatikan besar kecilnya penulisan hurufnya.

Beberapa perintah dasar yang penting diketahui adalah sebagai berikut :

1. `pwd` : perintah ini digunakan untuk mengetahui di direktori mana kita sedang berada. Lihat contoh berikut ini :

```
# pwd
/usr/local
```

Pada contoh di atas, perintah `pwd` menampilkan hasil `/usr/local`, artinya saat ini *user* sedang berada dalam direktori `/usr/local`.

2. `cd` : perintah ini artinya "*change directory*"; digunakan untuk berganti direktori. Perhatikan contoh berikut ini untuk mengetahui bagaimana menggunakan perintah `cd` :

```
# pwd
/home/tedy
# cd /usr/local
# pwd
/usr/local
```

*Tulisan ini masih terus diperbaiki.

Pada contoh di atas, mula-mula kita berada di direktori `/home/tesy`. Perintah `cd /usr/local`, membuat kita berpindah ke direktori `/usr/local`.

Perintah `cd ..` dapat kita gunakan untuk berpindah 1 direktori di atas direktori dimana kita berada. Lihat contoh berikut ini :

```
# pwd
/usr/local
# cd ..
# pwd
/usr
```

Pada contoh di atas mula-mula kita berada dalam direktori `/usr/local`, dengan menggunakan perintah `cd ..` kita berpindah ke direktori `/usr`.

3. `ls` : perintah ini digunakan untuk mengetahui isi dari sebuah direktori. Perintah ini sama dengan perintah `dir` dalam lingkungan DOS. Lihat contoh berikut ini :

```
# pwd
/home/tesy/tes
# ls
BAB III edit.doc*  blog.txt
```

Pada contoh di atas, direktori yang sedang aktif adalah `/home/tesy/tes`. Saat kita menjalankan perintah `ls`, sistem akan menampilkan isi dari direktori yang sedang aktif tersebut. Untuk melihat isi dari direktori lain (tanpa harus berada pada direktori tersebut), kita harus menuliskan nama direktori yang akan dilihat. Lihat contoh berikut :

```
# pwd
/home/tesy/tes
# ls
BAB III edit.doc*  data/  latihan.txt
# ls /home/tesy/Project/
BAB III edit.doc*  DATA/          skripsi-times.pdf*
BAB IV edit.doc*  skripsi-arial.pdf*
```

Pada contoh di atas, direktori yang sedang aktif adalah `/home/tesy/tes`. Walau demikian, kita masih tetap bisa melihat isi dari direktori `/home/tesy/Project` tanpa harus pindah ke dalam direktori itu terlebih dulu.

Beberapa opsi yang dapat ditambahkan pada perintah `ls` antara lain adalah :

- (a) `-l` : opsi `-l` digunakan untuk menampilkan semua *file* dalam bentuk daftar, seperti pada contoh berikut ini :

```
# ls -l
total 64
-rwxr-xr-x  1 tedy tedy 53760 Jun 24 17:57 BAB III edit.doc*
-rw-r--r--  1 tedy tedy  2806 Jun 24 17:57 blog.txt
```

- (b) `-a` : opsi `-a` digunakan untuk menampilkan semua *file* yang ada termasuk *file-file* yang tersembunyi (*hidden files*). Lihat contoh berikut ini :

```
# ls -al
total 72
drwxr-xr-x  2 tedy tedy  4096 Jun 24 17:57 ./
drwxr-xr-x 43 tedy tedy  4096 Jun 24 17:56 ../
-rwxr-xr-x  1 tedy tedy 53760 Jun 24 17:57 BAB III edit.doc*
-rw-r--r--  1 tedy tedy  2806 Jun 24 17:57 blog.txt
```

- (c) `-h` : opsi ini artinya "*human-readable*"; maksudnya adalah sistem akan menampilkan besar ukuran *file* dalam format yang mudah dibaca. Format yang mudah dibaca maksudnya seperti 29K, 3M, 1G.
- (d) `-r` : opsi ini membalik urutan daftar (tergantung dari aturan pengurutan). Misalnya daftar diurutkan menurut abjad (a-z) maka dengan tambahan opsi ini urutannya akan dibalik (menjadi z-a).
- (e) `-R` : opsi ini artinya "*recursive*"; dengan menggunakan opsi ini kita bisa melihat isi dari direktori dan juga isi dari subdirektori yang ada di dalamnya. Lihat contoh berikut ini :

```
# pwd
/home/tesy/tes
# ls -R
.:
BAB III edit.doc* data/ latihan.txt

./data:
BAB III edit.doc*
```

Pada contoh di atas, direktori `/home/tesy/tes` memiliki subdirektori `data`. Dengan menggunakan perintah `ls -R`, sistem akan menampilkan isi dari direktori `/home/tesy/tes` serta subdirektori `data` sekaligus.

- (f) `-s` : opsi ini menampilkan ukuran *file* dalam satuan byte.
- (g) `-S` : opsi ini mengurutkan *file* menurut dari yang terbesar sampai yang terkecil ukuran *filenya*.
- (h) `-t` : opsi ini mengurutkan *file* menurut waktu modifikasinya.

Opsi-opsi di atas dapat digunakan bersamaan (lebih dari 1 opsi sekaligus).

4. `cp` : perintah ini digunakan untuk membuat salinan dari sebuah *file*. Format perintah untuk menyalin sebuah *file* adalah seperti berikut ini :

```
# cp /home/tesy/contoh.txt /home/tesy/tes/
```

Pada contoh di atas *file* `contoh.txt` yang ada dalam direktori `/home/tesy` disalin ke dalam direktori `/home/tesy/tes`. Kita juga dapat membuat salinan dari *file* yang sama dengan nama yang berbeda, seperti contoh berikut :

```
# cp /home/tesy/contoh.txt /home/tesy/tes/latihan.txt
```

Perintah pada contoh di atas akan menyalin *file* `contoh.txt` ke dalam direktori `/home/tesy/tes` tapi dengan nama baru yaitu `latihan.txt`.

Untuk menyalin sebuah direktori, perintah yang digunakan adalah `cp -ir`. Lihat contoh berikut ini :

```
# ls -l
total 12
drwxr-xr-x  2 root root 4096 Jun 29 15:13 folder_1/
drwxr-xr-x  2 root root 4096 Jun 29 15:14 folder_2/
drwxr-xr-x  3 root root 4096 Jun 29 15:38 folder_baru/
# cp -ir folder_1 folder_baru
# cd folder_baru
# ls -l
total 4
drwxr-xr-x  2 root root 4096 Jun 29 15:38 folder_1/
```

5. `mv` : perintah ini sama digunakan untuk memindahkan suatu *file* dari suatu direktori ke direktori lainnya. Perintah ini juga bisa dipakai untuk mengganti nama sebuah *file*.

```
# mv /home/tesy/contoh.txt /home/tesy/tes
```

Pada contoh di atas, *file* contoh.txt dipindahkan dari direktori /home/tesy ke direktori /home/tesy/tes. Sama seperti pada perintah cp, kita juga bisa mengubah nama *file* setelah dipindahkan. Lihat contoh berikut ini :

```
# mv /home/tesy/contoh.txt /home/tesy/tes/latihan.txt
```

Dengan perintah di atas, *file* contoh.txt dipindahkan ke dalam direktori /home/tesy/tes sekaligus diganti namanya menjadi latihan.txt. Melihat sifat perintah mv ini, kita dapat menggunakannya untuk mengganti nama *file* (*rename*). Misalnya kita hanya ingin mengganti nama *file* contoh.txt menjadi latihan.txt (tanpa memindahkannya), maka perintahnya seperti berikut ini :

```
# mv /home/tesy/contoh.txt /home/tesy/latihan.txt
```

6. **rm** : perintah ini digunakan untuk menghapus *file*. Lihat contoh berikut ini :

```
# pwd
/home/tesy/tes
# ls -l
total 68
-rwxr-xr-x  1 tedy tedy 53760 Jun 24 17:57 BAB III edit.doc*
-rw-r--r--  1 tedy tedy  2806 Jun 24 17:57 blog.txt
-rw-r--r--  1 tedy tedy   127 Jun 24 18:02 latihan.txt
# rm blog.txt
rm: remove regular file 'blog.txt'? yes
# ls -l
total 64
-rwxr-xr-x  1 tedy tedy 53760 Jun 24 17:57 BAB III edit.doc*
-rw-r--r--  1 tedy tedy   127 Jun 24 18:02 latihan.txt
```

Beberapa opsi yang dapat ditambahkan pada perintah rm ini antara lain adalah :

- (a) **-f** : opsi ini artinya "*force*"; dengan menggunakan opsi ini sistem akan langsung menghapus *file* tanpa menampilkan konfirmasi terlebih dulu (seperti pada contoh sebelumnya).
- (b) **-i** : opsi ini artinya "*interaktif*"; dengan menggunakan opsi ini sistem akan menampilkan konfirmasi terlebih dulu sebelum menghapus *file*. Secara *default*, perintah rm menggunakan opsi ini.
- (c) **-rf** : opsi ini dapat dipakai untuk menghapus sebuah direktori yang memiliki isi, kita dapat menggunakan perintah rm -rf. Lihat contoh berikut ini :

```
# ls -l
total 68
-rwxr-xr-x  1 tedy tedy 53760 Jun 24 17:57 BAB III edit.doc*
drwxr-xr-x  2 tedy tedy  4096 Jun 24 19:30 data/
-rw-r--r--  1 tedy tedy   127 Jun 24 18:02 latihan.txt
# rm -rf data
# ls -l
total 64
-rwxr-xr-x  1 tedy tedy 53760 Jun 24 17:57 BAB III edit.doc*
-rw-r--r--  1 tedy tedy   127 Jun 24 18:02 latihan.txt
```

Pada contoh di atas, di dalam direktori /home/tesy/tes ada sebuah direktori dengan nama data. Direktori /data tidak kosong. Untuk menghapus direktori tersebut, kita harus menggunakan perintah rm -rf.

7. **mkdir** : perintah ini digunakan untuk membuat direktori (folder) baru. Lihat contoh berikut ini :

```

# pwd
/home/tesy/tes
# ls -l
total 8
drwxr-xr-x  2 root root 4096 Jun 29 15:13 folder_1/
drwxr-xr-x  2 root root 4096 Jun 29 15:14 folder_2/
# mkdir folder_baru
# ls -l
total 12
drwxr-xr-x  2 root root 4096 Jun 29 15:13 folder_1/
drwxr-xr-x  2 root root 4096 Jun 29 15:14 folder_2/
drwxr-xr-x  2 root root 4096 Jun 29 15:14 folder_baru/

```

Pada contoh di atas, di dalam direktori /home/tesy/tes mula-mula ada 2 direktori (folder_1 dan folder_2). Sebuah direktori baru dengan nama folder_baru dibuat dengan menggunakan perintah mkdir.

8. **tar** : perintah ini digunakan untuk menggabungkan beberapa *file* menjadi sebuah *file* tunggal. Selain itu perintah ini digunakan juga untuk memecah *file* tar yang ada menjadi *file-file* aslinya. Format perintah untuk membuat sebuah *file* dari beberapa *file* seperti berikut ini :

```
# tar -cvf <hasil.tar> <file1> <file2>
```

Perhatikan contoh berikut ini :

```

# ls -l
total 68
-rwxr-xr-x  1 tedy tedy 53760 Jun 24 17:57 BAB III.doc*
drwxr-xr-x  2 tedy tedy  4096 Jun 24 19:33 data/
-rw-r--r--  1 tedy tedy   127 Jun 24 18:02 latihan.txt
# tar -cvf sample.tar latihan.txt BAB\ III.doc
latihan.txt
BAB III.doc
# ls -l
total 132
-rwxr-xr-x  1 tedy tedy 53760 Jun 24 17:57 BAB III.doc*
drwxr-xr-x  2 tedy tedy  4096 Jun 24 19:33 data/
-rw-r--r--  1 tedy tedy   127 Jun 24 18:02 latihan.txt
-rw-r--r--  1 tedy tedy 61440 Jun 27 17:21 sample.tar

```

Pada contoh di atas, dua buah *file*; latihan.txt dan BAB III.doc; digabungkan menjadi sebuah *file* tunggal dengan nama sample.tar.

Perintah untuk membuka sebuah *file* *.tar adalah sebagai berikut :

```
# tar -xvf file.tar
```

Dengan menggunakan perintah di atas, semua isi file.tar akan diekstrak dan disimpan dalam sebuah folder baru dengan nama file. Lihat contoh berikut ini :

```

# ls -l
total 64
-rw-r--r--  1 tedy tedy 61440 Jun 27 17:21 sample.tar
# tar -xvf sample.tar
latihan.txt
BAB III edit.doc
# ls -l
total 128

```

```

-rwxr-xr-x  1 tedy tedy 53760 Jun 24 17:57 BAB III edit.doc*
-rw-r--r--  1 tedy tedy   127 Jun 24 18:02 latihan.txt
-rw-r--r--  1 tedy tedy 61440 Jun 27 17:21 sample.tar

```

9. **gzip** : perintah ini digunakan untuk mengkompresi sebuah *file* sehingga memiliki ukuran lebih kecil. Perintah ini juga dipakai untuk mendekomresi sebuah *file* yang sudah dikompres sebelumnya. Untuk mengkompres sebuah *file* perintah yang digunakan adalah seperti ini :

```
# gzip <nama_file>
```

Lihat contoh berikut ini :

```

# ls -l
total 64
-rw-r--r--  1 tedy tedy 61440 Jun 27 17:21 sample.tar
# gzip sample.tar
# ls -l
total 16
-rw-r--r--  1 tedy tedy 12878 Jun 27 17:21 sample.tar.gz

```

Pada contoh di atas sebuah *file* bernama *sample.tar* dikompres dengan menggunakan perintah **gzip**. Hasil dari kompresinya dinamai *sample.tar.gz*. Perhatikan bahwa ukuran *file* hasil kompresi menjadi jauh lebih kecil daripada ketika belum dikompres (dari 61440 byte menjadi 12878 byte).

Untuk mengembalikan *file* yang sudah terkompres ke dalam bentuk semula, perintahnya adalah seperti berikut :

```
# gzip -d <nama_file>
```

Lihat contoh berikut ini :

```

# ls -l
total 16
-rw-r--r--  1 tedy tedy 12878 Jun 27 17:21 sample.tar.gz
# gzip -d sample.tar.gz
# ls -l
total 64
-rw-r--r--  1 tedy tedy 61440 Jun 27 17:21 sample.tar

```

Contoh di atas memperlihatkan bagaimana mengembalikan *file* yang sudah dikompres menjadi *file* aslinya. Ukuran hasil dekompresi pasti menjadi lebih besar daripada sebelumnya.

10. **clear** : perintah ini dipakai untuk menghapus semua tampilan yang ada pada layar terminal. Perintah ini serupa dengan perintah **cls** pada lingkungan Windows.
11. **touch** : perintah ini dipakai untuk membuat sebuah *file* kosong, seperti pada contoh berikut ini :

```

# touch contoh.txt
# ls -l
total 0
-rw-r--r--  1 tedy tedy 0 2007-06-26 09:11 contoh.txt

```

Pada contoh di atas, kita membuat *file* *contoh.txt* dengan perintah **touch**. Oleh karenanya, besar dari *file* yang terbentuk hanya 0 byte.

12. **whatis** : perintah ini digunakan untuk mengetahui apa fungsi dari suatu perintah. Perintah ini akan menghasilkan sebaris deskripsi singkat dari sebuah perintah. Lihat contoh berikut ini :

```

# whatis cp
cp                (1) - copy files and directories
# whatis rm
rm                (1) - remove files or directories
# whatis mkdir
mkdir             (1) - make directories
mkdir             (2) - create a directory

```

Tidak semua perintah dapat dijelaskan oleh perintah `whatis`.

13. **man** : perintah ini dipakai untuk membaca dokumentasi dari aplikasi atau perintah. Penggunaannya adalah dengan menuliskan nama aplikasi atau perintah yang ingin diketahui dokumentasinya. Dengan menggunakan perintah `man` kita bisa mempelajari sintaks dari perintah-perintah yang ada. Contoh tampilan yang muncul ketika menggunakan perintah `man` adalah seperti berikut :

```

LS(1)                                User Commands                                LS(1)

NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILEs (the current directory by default).
    Sort entries alphabetically if none of -cftuSUX nor --sort.

    Mandatory arguments to long options are mandatory for short options
    too.

    -a, --all
        do not hide entries starting with .

    -A, --almost-all
        do not list implied . and ..

```

Contoh di atas hanya sepotong dari tampilan manual dari perintah `ls`. Inilah salah satu kelebihan sistem operasi Unix, tiap perintah memiliki manual/dokumentasi yang terbuka, terstruktur, dan memudahkan *user* untuk mengenali perintah yang dapat digunakan.

14. **cat** : perintah ini dipakai untuk menampilkan isi dari suatu *file* yang tersusun dari kode-kode ASCII. Misalnya kita memiliki *file* teks bernama `latihan.txt` yang ada di direktori `/home/tesy/tes`. Jika kita hanya ingin melihat isi dari *file* tersebut kita bisa melihatnya dengan menggunakan konsol, seperti ditunjukkan pada contoh berikut ini :

```

# ls
BAB III edit.doc*  blog.txt  latihan.txt
# cat latihan.txt
Ini adalah contoh \emph{file} teks.

```

Kita dapat membuat *file* teks dalam lingkungan Unix dengan menggunakan program yang bernama `vi`.

15. **more** : perintah ini dipakai untuk menampilkan isi dari sebuah *file*, tahap demi tahap sehingga mudah untuk dibaca. Lihat contoh berikut ini :

```

# more ntp.conf
# Permit time synchronization with our time source, but do not
# permit the source to query or modify the service on this system.

restrict default nomodify notrap noquery

# Permit all access over the loopback interface. This could
# be tightened as well, but to do so would effect some of
# the administrative functions.
restrict 127.0.0.1

# -- CLIENT NETWORK -----
# Permit systems on this network to synchronize with this
# time service. Do not permit those systems to modify the
# configuration of this service. Also, do not use those
--More--(22%)

```

Jika sebuah *file* teks yang ingin dilihat terlalu panjang isinya (melampaui ukuran terminal yang sedang digunakan), perintah `more` akan membuat sistem menampilkan isi teks separuh-separuh. Kita bisa menekan *Enter* untuk melihat baris berikutnya, atau menekan *Space* untuk melihat tampilan berikutnya.

16. **grep** : perintah ini dipakai untuk mencari suatu kata/*string* dalam sebuah *file* berbasis teks. Misalnya kita ingin mencari kata "Linux" dalam sebuah *file* teks dengan nama "tes.txt", lihat contoh berikut ini :

```

# cat tes.txt
Ada banyak sistem operasi yang berbasis Unix di antaranya adalah :
1. Linux, memiliki banyak distro, seperti misalnya :
  - Red Hat
  - Fedore Core
  - Suse
  - Open Suse
  - Ubuntu
  - Debian
2. FreeBSD
3. Solaris
4. Open Solaris

# grep "Linux" tes.txt
1. Linux, memiliki banyak distro, seperti misalnya :

```

Pada contoh di atas, dengan menggunakan perintah `grep` kita bisa dengan mudah menemukan baris yang mengandung kata "Linux".

17. **vi** : perintah ini dipakai untuk menjalankan sebuah editor teks Vi. Aplikasi Vi adalah editor berbasis teks yang sangat populer di lingkungan Linux. Vi memiliki 2 mode operasi : mode editing teks dan mode kontrol. Ketika pertama kali kita masuk ke dalam Vi, kita otomatis masuk ke dalam mode kontrol. Untuk masuk ke dalam mode editing teks, tekan tombol *i*. Untuk pindah dari mode editing teks ke mode kontrol kita menggunakan tombol *Esc*.

Bagi orang yang pertama kali menggunakan Vi, mungkin akan banyak mengalami kesulitan sebab semua dikontrol dengan menggunakan tombol *keyboard*. Berikut ini beberapa tombol yang digunakan dalam mode kontrol :

- (a) *i* - tombol ini dipakai untuk masuk ke dalam mode editing teks. Lihat contoh tampilan ketika sedang berada pada mode editing :


```

tes menggunakan teks editor Vi
~
~
~
~
-- INSERT --

```

Saat berada pada mode editing, di bagian bawah ditandai dengan tulisan INSERT.

- (b) :w - kombinasi tombol ini dipakai untuk menyimpan *file*. Perintah ini dapat dilengkapi dengan tanda seru (!) - tanda seru dipakai untuk memaksa Vi menyimpan perubahan yang dilakukan pada *file* teks tersebut, walaupun *file* teks tersebut awalnya adalah *read only file*.

```

tes menggunakan teks editor Vi
~
~
~
~

```

```
:w tes.txt
```

Setelah mengetikkan perintah :w tes.txt tekan *Enter*, maka akan muncul tampilan seperti ini :

```

tes menggunakan teks editor Vi
~
~
~
~

```

```
"tes.txt" 1L, 31C written
```

Contoh berikut ini menunjukkan tampilan yang muncul ketika kita membuka sebuah *read only file* (*file* yang hanya bisa dibaca).

```

tes menggunakan teks editor Vi
~
~
~
~

```

```
"tes.txt" [readonly] 1L, 31C
```

Jika kita mencoba menambahkan teks ke dalam *file* di atas akan muncul peringatan seperti berikut ini :

```
-- INSERT -- W10: Warning: Changing a readonly file
```

Jika kita tetap menambahkan teks dan mencoba menjalankan perintah :w maka akan muncul peringatan berikut ini :

```
E45: 'readonly' option is set (add ! to override)
```

Oleh karena itu, kita perlu menggunakan perintah :w! jika tetap ingin menambahkan teks ke dalam *file* tersebut.

- (c) :q - kombinasi ini dipakai untuk keluar dari Vi. Perintah ini juga dapat diikuti dengan tanda seru, artinya memerintahkan Vi untuk mengabaikan peringatan yang muncul. Misalnya saat kita belum menyimpan teks yang sudah diketik dan kita menggunakan perintah :q maka Vi akan memberitahu kita untuk menyimpan terlebih dahulu teks tersebut; seperti contoh berikut ini :

```
E37: No write since last change (add ! to override)
```

Tanpa tanda seru kita tidak bisa keluar dari Vi. Dengan menggunakan perintah :q! kita memaksa Vi untuk berhenti tanpa perlu menyimpan teks tersebut.

- (d) :d - kombinasi tombol ini dipakai untuk menghapus semua karakter yang berada dalam satu baris. Pindahkan kursor terlebih dahulu ke dalam baris yang ingin dihapus lalu jalankan kombinasi tombol tersebut.

- (e) x - tombol ini dipakai untuk menghapus karakter yang ada di sebelah kanan kursor.
- (f) r - tombol ini dipakai untuk mengganti karakter yang ada di sebelah kanan kursor.
- (g) a - tombol ini akan membuat kursor bergeser 1 karakter dan Vi sudah siap menerima teks lagi (sudah langsung masuk ke dalam mode editing).

Saat kita menggunakan Vi untuk mengedit sebuah teks, kita sulit untuk mundur ke karakter sebelumnya dengan menggunakan tanda panah kiri. Jika kita ingin memindahkan kursor ke tempat tertentu, maka kita harus masuk dulu ke dalam mode kontrol.

18. **uname** : perintah ini digunakan untuk menampilkan informasi dari sistem yang sedang berjalan. Lihat contoh berikut ini :

```
# uname
Linux
```

Ada beberapa opsi yang dapat ditambahkan pada perintah ini adalah :

- (a) -a : opsi ini akan menampilkan semua informasi tentang sistem operasi dan *hardware* yang digunakan. Lihat contoh berikut ini :

```
# uname -a
Linux localhost 2.6.8.1-12mdk #1 Fri Oct 1 12:53:41 CEST 2004 i686 Pentium III
(Coppermine) unknown GNU/Linux
```

- (b) -s : opsi ini menampilkan nama kernel yang digunakan dalam sistem operasi. Lihat contoh berikut ini :

```
# uname -s
Linux
```

- (c) -r : opsi ini akan menampilkan versi kernel yang digunakan. Lihat contoh berikut ini :

```
# uname -r
2.6.8.1-12mdk
```

- (d) -o : opsi ini akan menampilkan nama sistem operasi yang digunakan. Lihat contoh berikut ini :

```
# uname -o
GNU/Linux
```

- (e) -p : opsi ini akan menampilkan jenis prosesor yang digunakan. Lihat contoh berikut ini :

```
# uname -p
Pentium III (Coppermine)
```

19. **ps** : perintah ini digunakan untuk melihat proses-proses yang sedang berjalan di dalam komputer kita. Lihat contoh berikut ini :

```
# ps -ef | more
UID          PID  PPID  C  STIME TTY          TIME CMD
root          1      0  0  23:35 ?           00:00:00 init [5]

root          2      1  0  23:35 ?           00:00:00 [ksoftirqd/0]
root          3      1  0  23:35 ?           00:00:00 [events/0]
root          4      3  0  23:35 ?           00:00:00 [khelper]
root          5      3  0  23:35 ?           00:00:00 [kacpid]
root         21      3  0  23:35 ?           00:00:00 [kblockd/0]
root         22      1  0  23:35 ?           00:00:00 [khubd]
--More--
```

Untuk melihat proses-proses yang dijalankan oleh *user* tertentu kita dapat menggunakan opsi -U <user>. Lihat contoh berikut ini :

```
# ps -U tedy | more
  PID TTY          TIME CMD
 3392 ?            00:00:01 gnome-session
 3417 ?            00:00:00 ssh-agent
 3444 ?            00:00:00 dbus-launch
 3445 ?            00:00:00 dbus-daemon-1
 3449 ?            00:00:04 gconfd-2
 3452 ?            00:00:00 gnome-keyring-d
--More--
```

Pada contoh di atas, dengan menggunakan perintah `ps -U` kita bisa mengetahui proses yang sedang digunakan/dijalankan oleh *user* tedy.

20. **su** : perintah ini digunakan untuk mengubah hak akses menjadi root. Dalam sistem operasi Windows, root setara dengan Administrator. Hak akses root paling tinggi dalam tingkatan user. Root dapat memodifikasi *file-file* yang ada di dalam sistem. Sebenarnya perintah `su` tidak hanya digunakan untuk mengubah hak akses menjadi root, tetapi juga bisa untuk mengubah hak akses menjadi *user* tertentu. Misalnya di dalam sistem operasi ada 3 *user* : root, admin, guest. Untuk dapat mengakses *file-file*/konfigurasi dari *user* lain, kita dapat menggunakan perintah `su - <user>`.
21. **chmod** : perintah ini digunakan untuk mengubah hak akses *user* pada sebuah *file*. Ada 3 tipe hak akses dapat diterapkan pada sebuah *file*, yaitu :
 - *Read* (r) : *file* hanya bisa dibaca.
 - *Write* (w) : *file* dengan tipe ini bisa dibaca sekaligus bisa dimodifikasi.
 - *Execute* (x) : *file* dengan tipe ini bisa dieksekusi/dijalankan.

Lihat contoh berikut ini :

```
# ls -l
total 300
-rw-rw-r-- 1 tedy tedy  1996 Jul  2 17:00 blog0207.txt
-rw-rw-r-- 1 tedy tedy  3193 Jul  7 17:18 blog.txt
-rw-rw-r-- 1 tedy tedy   162 Jul 15 10:14 command.aux
-rw-rw-r-- 1 tedy tedy  8690 Jul 15 10:14 command.log
-rw-rw-r-- 1 tedy tedy 143680 Jul 15 10:14 command.pdf
-rwxrwxrwx 1 tedy tedy  38634 Jul 15 02:59 command.tex
-rwxr-xr-x 1 root root  53773 Jun 15 19:19 unix.jpg
```

Pada contoh di atas, tiap *file* memiliki aturan hak akses sendiri-sendiri bagi *owner*, *group*, dan *other* (ditandai misalnya dengan `-rw-rw-r--`). Masing-masing hak akses suatu *file* terbagi menjadi 3 bagian, tiap bagian tersusun dari 3 karakter dan dibatasi dengan tanda `-`. Perhatikan contoh berikut :

```
-rw-rw-r--
```

Pada contoh di atas, *owner* memiliki hak akses `rw`, *group* memiliki hak akses `rw`, dan *other* memiliki hak akses `r`.

Untuk mengganti hak akses *file*, format perintahnya adalah seperti berikut ini :

```
# chmod ugo=rwx <nama_file>
```

Misalnya pada contoh di sebelumnya, sebuah *file* dengan nama `blog.txt` memiliki hak akses `-rw-rw-r--`. Artinya *user* selain pemilik *file* hanya bisa membaca dan tidak bisa memodifikasi *file* tersebut. Misalnya kita akan mengubah hak akses bagi *user* lainnya sehingga semua *user* dapat memodifikasi file tersebut, maka contoh perintahnya adalah :

```
# ls -l blog.txt
-rw-rw-r-- 1 tedy tedy 3193 Jul  7 17:18 blog.txt
# chmod o=rw blog.txt
# ls -l blog.txt
-rw-rw-rw- 1 tedy tedy 3193 Jul  7 17:18 blog.txt
```

22. **mount** : perintah ini dipakai untuk memasang sebuah perangkat yang memiliki *file system* ke dalam sistem operasi. Misalnya, untuk melihat sebuah partisi lain dalam *harddisk*, untuk dapat menggunakan USB Flash Drive, untuk dapat menggunakan CD (ingat bahwa sebuah CD/*image* CD pun mempunyai sebuah *file system*), dst. Semua perangkat yang ada di dalam sistem operasi Unix dianggap sebagai sebuah *file/directory*. Format penggunaan dari perintah ini adalah seperti berikut ini :

```
# mount -t <type> /dev/<perangkat> /mnt/<mount point>
```

Mula-mula kita perlu mengetahui dimana perangkat/*file system* yang akan digunakan itu berada. Kemudian kita perlu mendefinisikan dimana letak *mount point* untuk meletakkan perangkat/*file system* yang akan digunakan. Misalnya kita ingin menggunakan sebuah USB *flash disk*, maka contoh perintahnya adalah sebagai berikut :

```
# cd /media
# mkdir usbflashdisk
# mount /dev/sda1 /media/usbflashdisk/
# cd /media/usbflashdisk/
# ls
total 51954
-rwxr-xr-x  1 root root   111550 Jun 20 12:44 1showthread.php.html
drwxr-xr-x  3 root root    6656 Jun 20 12:44 1showthread.php_files
-rwxr-xr-x  1 root root  129669 Jun 20 12:45 2showthread.php.html
drwxr-xr-x  3 root root    6656 Jun 20 12:45 2showthread.php_files
-rwxr-xr-x  1 root root  113424 Jun 20 12:46 3showthread.php.html
drwxr-xr-x  4 root root    6656 Jun 20 12:46 3showthread.php_files
```

Contoh lain : misalkan di dalam komputer kita terdapat sebuah partisi yang digunakan oleh instalasi Linux lainnya. Untuk dapat membaca partisi dan *file-file* yang ada di dalamnya kita harus mount terlebih dulu partisi tersebut. Yang perlu diketahui adalah nama partisi tersebut, anggap partisi tersebut adalah *hda2*. Contoh perintahnya adalah :

```
# pwd
/media
# ls
cdrom floppy usbflashdisk
# mkdir linux
# mount /dev/hda2 /media/linux/
# cd linux/
# pwd
/media/linux
# ls
bin  dev  home  lib          media mnt  proc  sbin  srv  tmp  var
boot etc  initrd lost+found  misc  opt  root  selinux sys  usr
```

Beberapa distro Linux saat ini sudah menambahkan fitur *auto mount*, sehingga semua perangkat yang dipasangkan ke dalam komputer langsung dikenali (*plug and play*).

23. **umount** : perintah ini dipakai untuk melepaskan sebuah perangkat dari sistem operasi. Misalnya untuk melepaskan USB *flash disk* dari sistem kita dapat menggunakan perintah ini :

```
# umount /mnt/usb
```

24. **date** : perintah ini digunakan untuk menampilkan tanggal dan jam pada saat ini. Tampilan yang muncul ketika kita menggunakan perintah `date` adalah seperti berikut ini :

```
# date
Tue Jun 26 22:05:01 EDT 2007
```

25. **cal** : perintah ini digunakan untuk menampilkan kalender dalam terminal.

```
# cal
      June 2007
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
```

26. **exit** : perintah ini dipakai untuk keluar dari konsol/terminal.

27. **bash** : perintah ini dipakai untuk menggunakan konsol bash (GNU Bourne-Again Shell).

28. **ifconfig** : perintah ini digunakan untuk melihat konfigurasi IP yang sudah ada pada *network interface* yang ada dalam PC kita. Perintah untuk melihat konfigurasi IP yang sudah ada di dalam komputer adalah seperti berikut :

```
# ifconfig -a
```

Lihat contoh berikut ini :

```
# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:E0:00:3B:6E:CA
          inet addr:222.124.79.24  Bcast:222.124.79.255  Mask:255.255.255.0
          inet6 addr: fe80::2e0:ff:fe3b:6eca/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2782 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2903 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2040964 (1.9 MiB)  TX bytes:517219 (505.0 KiB)
          Interrupt:9 Base address:0x2000
```

Perintah `ifconfig` juga dapat dipakai untuk menentukan alamat IP dan *netmask* pada suatu *interface*. Untuk menkonfigurasi sebuah IP pada suatu *interface*, perintah yang digunakan adalah sebagai berikut :

```
# ifconfig eth0 192.168.1.1 netmask 255.255.255.0
```

Pada contoh di atas *interface* yang akan diberi IP misalnya adalah Ethernet 0 (`eth0`).

29. **shutdown** : perintah ini dipakai untuk me-*restart* atau mematikan komputer. Dengan menggunakan perintah ini kita dapat me-*restart* komputer, bisa juga men-*shutdown* komputer. Perintah untuk mematikan komputer adalah seperti berikut ini :

```
# shutdown -y -i0 -g0
```

Perintah untuk merestart komputer adalah seperti berikut ini :

```
# shutdown -y -i6 -g0
```

Kedua perintah di atas dilengkapi dengan beberapa opsi, arti dari masing-masing opsi tersebut adalah :

- (a) `-y` - opsi ini menjawab “*yes*” untuk semua pertanyaan yang muncul selama proses shutdown.
 - (b) `-g` - opsi ini adalah singkatan dari “*go*”. Dengan menggunakan opsi ini kita bisa mengatur kapan proses shutdown dijalankan. Misalnya kita menggunakan opsi `-g0`, artinya kita mengatur supaya proses shutdown langsung dijalankan nol detik setelah kita memberikan perintah. Satuan waktu yang digunakan adalah detik.
 - (c) `-i` - opsi ini adalah singkatan dari “*init*”. Ada 6 tingkatan mode init dalam lingkungan Unix. Yang sering digunakan adalah `init 0` dan `init 6`. `init 0` dipakai untuk mematikan komputer sementara `init 6` dipakai untuk me-*restart* komputer.
30. **reboot** : perintah ini digunakan untuk me-*restart* komputer. Perintah `reboot` ini mirip dengan perintah `shutdown -i6`. Perintah ini membutuhkan hak akses `root` untuk dapat dijalankan.
31. **useradd** : perintah ini dipakai untuk menambah pengguna (*user*) yang berhak menggunakan sistem operasi dalam komputer kita. Format perintahnya adalah seperti berikut ini :

```
# useradd <username>
```

Perintah penambahan *user account* ini memiliki beberapa opsi, di antaranya adalah :

- (a) `-u` : opsi ini dipakai untuk menentukan *user ID* (UID). Opsi ini diikuti dengan nilai yang diinginkan. Misalnya UID yang akan dipakai 100, maka format perintahnya adalah `-u 100`. Cara penulisan opsi dan nilainya ini berlaku sama untuk semua opsi.
 - (b) `-g` : opsi ini dipakai untuk menentukan *group ID*.
 - (c) `-c` : opsi ini dipakai untuk menambahkan *comment* pada sebuah *user account*.
 - (d) `-s` : opsi ini dipakai untuk menentukan jenis *shell* (*default shell*) yang digunakan oleh *user* ketika pertama kali *login*.
 - (e) `-d` : membuat *home directory*.
 - (f) `-m` : memaksa sebuah *home directory* yang belum ada untuk ditambahkan. Opsi ini tidak membutuhkan nilai apapun.
 - (g) `-p` : menentukan *password* untuk *user* tertentu.
32. **usermod** : perintah ini digunakan untuk mengubah sebuah *user account*. Perintah ini dapat menggunakan opsi-opsi yang dimiliki perintah `useradd`. Kita dapat memodifikasi *user ID*, *grup ID*, *comment*, *shell*, *home directory*, *password* dari sebuah *username*. Format perintahnya sama dengan perintah `useradd`, yaitu :

```
# usermod <opsi> <username>
```

33. **userdel** : perintah ini dipakai untuk menghapus sebuah *user account*. Format perintah untuk menghapus sebuah *username* adalah sederhana berikut ini :

```
# userdel <username>
```

34. **passwd** : perintah ini kita gunakan apabila kita ingin mengubah *password login* kita. Tampilan yang muncul ketika kita menggunakan perintah ini adalah seperti berikut ini :

```
# passwd
Changing password for user tedy.
Changing password for tedy
(current) UNIX password:
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

Ketika kita ingin mengganti *password*, mula-mula kita diminta memasukkan *password* lama kita. Kemudian kita diminta memasukkan *password* baru yang kita inginkan sebanyak 2 kali (untuk pemeriksaan). Perlu diketahui, tidak ada karakter yang muncul ketika kita mengetikkan *password*.

35. **df** : perintah ini dipakai untuk mengetahui berapa besar kapasitas *harddisk* yang terpasang dalam komputer, berapa yang sudah terpakai dan berapa sisanya. Contoh tampilan saat menggunakan perintah **df** adalah seperti ini :

```
# df
Filesystem      Size  Used Avail Use% Mounted on
/dev/hda2       7.7G  4.1G  3.3G  56% /
/dev/hda1       20G   8.6G   12G  44% /mnt/windows
```

Perintah **df** memiliki beberapa opsi yang dapat digunakan, beberapa di antaranya adalah :

- (a) **-k** : dengan menggunakan opsi ini kapasitas *harddisk* ditampilkan dalam satuan *block*.

```
# df -k
Filesystem      1K-blocks      Used Available Use% Mounted on
/dev/hda2       8064304    4257372   3397276  56% /
/dev/hda1       20482840   8922636  11560204  44% /mnt/windows
```

- (b) **-h** : dengan menggunakan opsi ini data kapasitas *harddisk* ditampilkan dalam format yang mudah dibaca; misalnya 130K, 432M, 4G.

- (c) **-t** : dengan menggunakan opsi ini data akan diurutkan menurut tipe *file system*nya.