

Mempelajari Dasar-Dasar Shell (Operasi File dan Direktori)

Mohammad Safii

karebet_asli@telkom.net

<http://sapitenk.cjb.net>

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Dunia Linux atau umumnya keluarga unix mempunyai perbedaan jauh di bandingkan dengan system operasi lain, misalnya Microsoft Windows. Dikarenakan terbiasa dengan windows maka akan sangat sulit jika harus langsung berhadapan dengan linux. Secara teoritis sebelum menggunakan linux hal dasar yang harus di ketahui ialah istilah-istilah yang ada di Linux. *Shell* salah satu istilah yang umum diperbincangkan tapi banyak juga yang sampai detik ini belum mempunyai gambaran jelas, padahal *shell* ialah tempat konfigurasi dan menjalankan / mengoperasikan Linux.

Distro (distribusi) linux sekarang ini telah dikembangkan oleh para developer untuk membuat Linux secantik mungkin. Dengan tujuan agar linux digemari dan masyarakat mulai beralih ke sebuah babak baru; *open source*. Dengan menyertakan berbagai kecantikan dari *window manager* seperti KDE, Gnome maka banyak pemula linux yang mengabaikan keberadaan *shell*. *Shell* tidak ditujukan bagi seorang admin tapi juga bagi pengguna linux tanpa kecuali. Dibawah ini diuraikan kegunaan *shell* yang saya petik dari artikel I Made Wiryana (Pengantar Belajar Unix)

Kegunaan shell

1. Pemakaian interaktif
2. Untuk mengontrol *session* UNIX
contohnya: Set home dir., mail directory, startup file
3. Pemrograman
Unix shell menyediakan sekumpulan instruksi khusus yang dapat digunakan untuk membuat program shell scripts

Shell secara singkat seperti MS Dos milik Microsoft. Memang benar bahwa dibutuhkan hafalan-hafalan perintah jika ingin melakukan sesuatu. Contohnya proses *copy* file, jika ingin *mengcopy* file maka perlu mengetikkan perintah tertentu dan mungkin anda berfikir terlalu lama. Jika dibandingkan dengan *window manager* seperti KDE anda hanya dengan memainkan mouse saja.

Kalaupun merasa nyaman dengan menggunakan *window manager* (KDE, Gnome) maka yakinlah bahwa sesungguhnya *window manager* merupakan 'kepanjangan tangan' dari shell. Window manager merupakan visualisasi dari *shell* untuk mempermudah pengoperasian.

Contoh sederhana ialah jika menggunakan KDE untuk melihat informasi prosessor maka bisa melihatnya di control center. Tapi sebenarnya perintah tersebut merupakan visualisasi dari perintah di shell yaitu :

```
[syafii@sapitenk syafii]$ cat /proc/cpuinfo
```

Kenapa Shell ?

Shell merupakan *command-line interpreter* artinya untuk melakukan apapun harus dengan mengetikkan baris perintah. Dennis Ritchie dan Ken thompson (Pembuat Unix) berpendapat media apa yang harus di buat agar orang dapat 'mencoba' unix mereka. Kemudian dibuatlah shell dengan nama *Bourne Shell (sh)* yang dibuat oleh SR Bourne. Semenjak itu bermuncullah jenis shell yang lain yaitu *C shell, Korn shell* dan milik GNU sendiri yaitu *Bourne Again Shell (bash), shell* versi gratis. Secara *default* dalam Linux menggunakan *bash shell*. Untuk melihat tipe shell dari user ketikkan :

sintaknya yaitu : `$grep nama_user /etc/passwd`

```
[syafii@sapitenk syafii]$ grep syafii /etc/passwd  
syafii:x:500:500::/home/syafii:/bin/bash
```

Panduan untuk mengetahui jenis shell

| Shell | Jenis |
|----------------------------|------------------------------------|
| /bin/csh atau /usr/bin/csh | C shell |
| /bin/sh | Bourne shell (bash) |
| /bin/rsh | Restricted Bourne Shell |
| /bin/jsh | Bourne shell termasuk jobs control |
| /bin/ksh | Korn shell |
| /bin/rksh | Restricted Korn shell |

Dari contoh diatas username syafii menggunakan *shell* jenis *bash*. *Shell* secara garis besar lebih power full dibandingkan dengan *window manager*. Yang perlu diingat karena linux ditulis dengan bahasa C dan bahasa C bersifat *case sensitif* yang artinya membedakan antara huruf kecil dan besar, maka perintah di shell ketik dalam huruf kecil jika diketik dalam huruf besar shell tidak akan mengenali.

Struktur Prompt di shell

```
[username@hostname pwd]$
```

contoh :

```
[syafii@sapitenk /home]$
```

Rumus diatas merupakan struktur *prompt shell* dimana *username* ialah nama user, *hostname* ialah nama host (default-nya *localhost*), sedangkan *pwd* ialah direktory sekarang (letak kursor berada), tanda \$ berarti sebagai user saja, untuk *root* (sebutan administrator untuk keluarga unix) menggunakan tanda #.

Perintah dasar Shell

pwd

pwd (present working direktory) ialah perintah untuk melihat direktory kerja sekarang atau mudahnya untuk melihat posisi kursor sekarang.

```
$ pwd
```

```
/home
```

cd (change direktory)

cd digunakan untuk berpindah direktory. Sintaknya : `cd nama_direktory.`

```
$ cd syafii
```

ada trik mengenai perintah cd, ketika anda dihadapkan pada suatu kasus dimana nama direktori tersebut panjang dan jika di ketik terlalu lama dan rentang kesalahan ketik, maka cukup ketik satu atau beberapa huruf dan tekan tombol tab.

1. `$ cd s` # tekan tab hasilnya seperti dibawah ini
2. `$ cd syafii/`

Contoh diatas ingin berpindah ke direktory syafii, cukup dengan mengetikkan 'cd s' dan tekan tombol tab maka hasilnya akan seperti pada contoh kedua. Anda yang mulai menekuni dunia linux dan menginstall aplikasi under *shell* gunakan trik diatas.

Perintah cd digunakan untuk berpindah direktori sedangkan untuk berpindah direktori diatas nya

```
$ pwd
/home/syafii/mail
$ cd ..      #naik satu tingkat direktiry diatasnya
$ pwd
/home/syafii
```

su (subtitute user)

su digunakan untuk berpindah user (switch).

Sintaknya : `$su nama_user`

Namun jika anda mengetikkan 'su' saja maka defaultnya *switch* ke *root* (administrator).

```
[syafii@sapitenk syafii]$ su
```

```
Password:
```

```
[root@sapitenk syafii]#
```

Ingat password di linux tidak ditampilkan (*no-echo*).

ls (List)

Perintah 'dir' pada Dos digunakan untuk melihat isi directory begitu juga dengan di shell digunakan 'ls'. Namun di shell juga di kenal perintah 'dir' yang mempunyai output yang sama dengan 'ls'.

Sintaknya : `$ls`

```
$ ls
Mail home mail mbox
```

Perintah di atas digunakan untuk menampilkan isi directory, seperti halnya pada windows di linux juga terdapat *file hidden* (nama file yang diawali dengan tanda titik berarti file hidden). Untuk menampilkan file hidden maka tambahkan atribut '-a' pada ls. Huruf 'a' berarti menampilkan semua (all) file tanpa kecuali

Sintaknya : `$ls -a`

```
$ ls -a
.          .addressbook.lu .bash_profile .mh_profile Mail mbox
..         .bash_history .bashrc   .pinerc   home
.addressbook .bash_logout .emacs    .screenrc mail
```

Untuk menampilkan file lengkap dengan ukuran, waktu di-*create* dan hak kepemilikan tambahkan atribut '-l' (long).

```
$ ls -l
total 16    # nilai total dicari dari jumlah file keseluruhan termasuk file hidden (file yang tersembunyi)
drwx----- 2 syafii syafii  4096 Aug 12 22:46 Mail
drwxrwxr-x 2 syafii syafii  4096 Aug 15 08:37 home
drwx----- 2 syafii syafii  4096 Aug 12 22:44 mail
-rw----- 1 syafii syafii  1797 Aug 12 22:45 mbox
atribut 'a' dan 'l' juga dapat digabung dengan sintak $ls -al
```

cat (Concatenate)

cat digunakan untuk melihat isi file. Layaknya perintah 'type' di dos.

Sintak : `$ cat nama_file`

```
$ cat test.txt
ilmu
komputer
```

mkdir (Make directory)

Perintah yang satu ini mempunyai fungsi yang sama dengan perintah 'md' pada Dos. Untuk membuat directory ketik saja mkdir.

Sintak : \$mkdir nama_directory

\$ mkdir ilmu_komputer

\$ ls

Mail home mail mbox ilmu_komputer test.txt

rmdir (remove directory)

Kebalikan dari mkdir, rmdir berfungsi menghapus directory.

Sintak : \$rmdir nama_directory

\$rmdir pcplus

rm (remove)

Jika rmdir digunakan untuk menghapus directory maka untuk menghapus (remove) file digunakan perintah rm.

Sintak : \$rm nama_file

\$rm test.txt

cp (copy)

Untuk mengcopy file dalam satu directory atau antar directory secara umum mempunyai sintak :

\$cp nama_file tujuan

\$ cp test.txt pcplus/

keterangan perintah diatas yaitu file test.txt akan di copy ke directory pcplus.

mv (move)

Membuat, menghapus file sudah anda pelajari. Sekarang perintah mv mempunyai dua makna artian yaitu

1. 'mv' untuk memindahkan file ke directory lain

sintak : \$mv nama_file directory_tujuan

\$mv test.txt pcplus

file test.txt akan dipindahkan di directory pcplus.

2. 'mv' digunakan untuk merubah ke nama file lain (rename).

Sintak : \$mv nama_file_lama nama_file-baru

\$mv test.txt tess.txt

file test.txt akan di rename menjadi tess.txt

more

Perintah terakhir ini sama fungsinya dengan cat yaitu melihat isi file. Tapi sekarang akan ditunjukkan bagaimana menggabungkan isi dari 2 file menjadi satu tanpa menghapus masing-masing isi.

Sekarang buat dua file yaitu : coba.txt dan test.txt

\$vi coba.txt

Tekan 'i' , ketikkan kalimat 'kalimat ini saya tulis dari coba.txt' , tekan tombol esc (escape) → titik dua (:)

→ lanjutkan dengan ketik wq

\$ cat coba.txt

kalimat ini saya tulis dari coba.txt

buat lagi file yang kedua dengan nama test.txt yang isinya 'kalimat ini saya tulis dari test.txt'

\$ cat test.txt

kalimat ini saya tulis dari test.txt

Nah, sekarang anda sudah mempunyai dua file coba.txt dan test.txt sekarang akan dicopy isi file dari coba.txt ke dalam test.txt

\$more coba.txt >> test.txt

\$ cat test.txt

kalimat ini saya tulis dari test.txt

kalimat ini saya tulis dari coba.txt

tanda >> berarti memasukkan suatu kejadian. Misalnya dari contoh diatas , perintah \$more coba.txt >> test.txt berarti pertama kali yang dilakukan dengan perintah diatas ialah melihat isi file (more) dari coba.txt setelah itu hasil eksekusinya di masukkan (>>) ke test.txt. Tanda >> berarti tidak meng-overwrite isi file yang terdapat pada test.txt . Tetapi jika anda hanya menggunakan tanda > saja, maka isi file test.txt akan di hapus diganti dengan isi file coba.txt

```
$ more coba.txt > test.txt
$ cat test.txt
kalimat ini saya tulis dari coba.txt
```

Trik Shell :

- Untuk mencari file perintah yang digunakan *locate*

```
$locate nama_file
```

- jika ingin reboot (harus menjadi root dulu) lewat shell ketikkan perintah

```
#!/sbin/reboot
```

Untuk shutdown

```
#!/sbin/shutdown -h now
```

atribut -h berarti halt dan shutdown dilaksanakan sekarang (now).

- Untuk menghapus layer ketikkan 'clear'
\$clear # atau juga dapat dengan menekan ctrl+l

Kesimpulan

Perintah pada shell sebenarnya sangat mudah. Semua daftar perintah mudah dihafal. Contohnya ls berasal dari kata list, cp berasal dari kata copy. Namun untuk melihat argument (atribut) yang lengkap dapat anda ketikkan :

```
$man nama_perintah
```

```
$man cp
```

Untuk keluar dari man ketik q

Biografi Penulis



Moh. Safii lahir di Kota Malang , tanggal 12 Oktober 1982. Jenjang SMU di lewatinya di SMU N 1 Malang dan sekarang kuliah di D-3 Manajemen Informatika Semester V Brawijaya Fakultas MIPA. Kebanggaan terbesarnya ialah pernah menjadi anggota Paskibraka 1999 sebagai pengibar bendera di Balaikota Kodya Malang . Mengenal linux sejak tahun 2002. Sebagai penulis artikel di majalah komputer. Artikelnya pernah di terbitkan di berbagai majalah komputer diantaranya PcPlus dan neotek. Selain aktif menjadi penulis artikel, juga telah menulis buku pertamanya seputar Linux yang di terbitkan oleh Andi Offset. Cita-cita terbesarnya ialah ingin kuliah di Luar Negeri.

Informasi lebih lanjut tentang penulis dapat dihubungi lewat email di :

karebet_asli@telkom.net

<http://sapitenk.cjb.net>