

Praktikum 3

Bash Shell

POKOK BAHASAN:

- ✓ Shell
- ✓ Pengenalan Bash
- ✓ Penggunaan Baris Perintah dengan Cepat
- ✓ Lebih Lanjut Tentang Baris Perintah

TUJUAN BELAJAR:

Setelah mempelajari materi dalam bab ini, mahasiswa diharapkan mampu:

- ✓ Memahami bagaimana menggunakan baris perintah (*command line*)
- ✓ Mengerti penggunaan baris perintah lebih mendalam
- ✓ Mengetahui bagaimana menggunakan history dan cara pengeditannya

DASAR TEORI:

1 SHELL

Shell adalah *Command executive*, artinya program yang menunggu instruksi dari pemakai, memeriksa sintak dari instruksi yang diberikan, kemudian mengeksekusi perintah tersebut. Shell ditandai dengan prompt. Untuk pemakai menggunakan prompt \$ dan untuk superuser menggunakan prompt #.

Beberapa macam shell :

- /bin/sh
Bourne shell, dirancang oleh Steve Bourne dari AT&T
- /bin/csh
Dikembangkan oleh UNIX Berkeley yang dikenal dengan C-Shell

- `/bin/bash`

Kompatibel dengan Bourne Shell dan juga mengadaptasi kemampuan Korn-Shell.

Perbedaan mendasar antara Shell di atas hampir tidak ada, kecuali pada fasilitas pemrograman dan editing.

2 PENGENALAN BASH

Bash merupakan kepanjangan dari "*Bourne Again Shell*" yang merupakan kelanjutan dari sh (shell dari original UNIX). Dikembangkan untuk proyek GNU dan secara de facto merupakan shell standar Linux.

Bourne Again Shell (**bash**) mengimplementasikan beberapa fitur dari shell sebelumnya yaitu **sh**, **csh**, **ksh**, **tcsh**.

3 PENGGUNAAN BARIS PERINTAH DENGAN CEPAT

Terdapat beberapa cara untuk menggunakan baris perintah dengan cepat (*command line shortcut*) yaitu menggunakan wildcard, kunci Tab dan history

- **Wildcard**

Ketika mengetikkan perintah, biasanya perlu memanggil/menggunakan perintah yang sama untuk beberapa file pada waktu yang sama. Penggunaan wildcard atau meta karakter, memungkinkan untuk memetakan beberapa nama file dengan proses yang disebut *globbing*. Beberapa wildcard antara lain:

- * : satu atau lebih karakter yang sesuai
- ? : single karakter yang sesuai
- [a-z] : satu jangkauan karakter yang sesuai
- [^a-z] : semua karakter yang sesuai kecuali jangkauan tersebut

Sebagai contoh, jika satu direktori terdiri dari beberapa file: `joshua.txt alex.txt angelo.txt gong.mp3 zonk.mp3`.

Bila diketikkan perintah :

```
$ rm *.mp3 sama dengan mengetikkan
```

```
$ rm gong.mp3 zonk.mp3
```

- **Kunci Tab**

Kunci Tab digunakan untuk meneruskan perintah baris dengan lengkap. Kunci ini dapat digunakan untuk melengkapi nama perintah atau nama file. Untuk melengkapi nama file, ketik perintah dan dilanjutkan dengan ketik nama file. Jika yang diketikkan sudah merupakan karakter unik, tekan kunci tab. Jika nama file sudah berupa karakter unik, kunci tab akan melengkapi nama file. Sebagai contoh :

```
$ ls myf<Tab>
```

Jika hanya ada satu file yang dimulai dengan `myf` yaitu `myfile.txt` maka nama file ini akan muncul di layar sebagai berikut

```
$ ls myfile.txt
```

- **History**

History diadaptasi dari C-Shell, yaitu catatan dari semua instruksi yang sejauh ini telah dilakukan. Catatan ini dapat dilihat sebagai history, kemudian dapat dipilih kembali, diedit dan dieksekusi. History memudahkan pemakai untuk mengedit kembali instruksi kompleks dan panjang, terutama bila terjadi kesalahan pada penulisan instruksi maupun parameter.

Navigasi pada daftar history menggunakan karakter kontrol sebagai berikut :

<code>^P (Ctrl-P)</code>	melihat instruksi sebelumnya
<code>^N (Ctrl-N)</code>	melihat instruksi berikutnya
<code>!!</code>	eksekusi kembali instruksi sebelumnya
<code>!!-3</code>	3 instruksi sebelumnya akan diulang
<code>!!88</code>	ulangi instruksi no 88

Beberapa trik tentang history :

- Gunakan kunci panah atas dan bawah untuk melihat perintah sebelumnya.
- Gunakan `Ctrl-r` untuk mencari perintah pada history
- Untuk memanggil kembali argumen terakhir dari perintah sebelumnya gunakan kunci `Esc` (kunci escape dilanjutkan dengan periode) dan kunci `Alt-` (tekan kunci `Alt` dilanjutkan periode)

4 LEBIH LANJUT TENTANG PERINTAH BARIS

Fitur *tilde* (~) direferensi dari C shell digunakan untuk mempermudah mereferensi file dan direktori pada *home directory*. Sebagai contoh:

```
$ cat ~/.bash_profile
```

Tanda ~ mereferensi ke *home directory*.

Parameter atau variabel \$ menggantikan nilai variabel dalam baris perintah. Shell menggantikan string dengan nilai variabel sebelum memanggil. Sebagai contoh:

```
$ cd $HOME/public_html
```

Tanda \$HOME mereferensi pada *home directory*.

Tanda kurung kurawal ({ }) membuat string untuk setiap pola di dalam kurung kurawal jika terdapat file. Sebagai contoh :

```
$ echo {a,b}
```

akan menghasilkan string a dan b.

Perintah output `` atau \$() disebut perintah pengganti. Pada perintah pengganti, perintah pada *backquote* dieksekusi dan output perintah diberikan pada baris perintah, sebagai contoh:

```
$ echo "Hostname: `hostname`"
```

Perintah aritmatika \$ [] menggantikan hasil dari ekspresi matematika ke dalam baris perintah. Penggunaan perintah expr diikuti dengan *backslash* sebelum tanda asterik untuk perkalian harus diperhatikan. Alternatif lain menggunakan fungsi matematika dengan sintak \$ []. Penggunaan sintak tidak perlu menggunakan *backslash* sebelum asterik untuk perkalian. Aritmatika dasar yang dikenali terdiri dari :

- + penambahan
- pengurangan
- * perkalian
- / pembagian
- ** pangkat
- % modulo

Tanda backslash (\) menyebabkan karakter berikutnya dikenal, sebagai contoh :

```
$ echo Your cost: \$5.00
```

Selain itu tanda backslash juga digunakan untuk melanjutkan perintah pada baris berikutnya, sebagai contoh :

```
$ echo "This long string will be echoed \  
> back as one long line
```

Tanda petik satu (') membatasi semua ekspansi, tanda petik dua (") membatasi semua ekspansi kecuali:

- \$ ekspansi variabel
- - mengganti perintah
- \ ekspansi satu karakter
- ! pengganti history

TUGAS PENDAHULUAN:

Jawablah pertanyaan-pertanyaan di bawah ini :

1. Apa yang dimaksud dengan shell dan sebutkan shell yang ada di sistem operasi Linux.
2. Apa kegunaan wildcard, kunci Tab dan history ?
3. Apa yang dimaksud fitur tilde (~), parameter \$, kurung kurawal { }, perintah ` ` atau \$(), \$[], tanda \, tanda petik satu (') dan petik dua (")

PERCOBAAN:

1. Login sebagai user.
2. Masuk ke TTY1 dengan menekan *Ctrl-Alt-F1* dan lakukan percobaan-percobaan di bawah ini. Perhatikan hasilnya.
3. Selesaikan soal-soal latihan.

Percobaan 1 : Wildcard

1. Membuat beberapa file

```
$ touch joshua.txt James.txt alex.txt Angelo.txt  
gong.mp3 zonk.mp3
```

2. Melihat daftar file yang mempunyai *extension* sama

```
$ ls *.txt  
$ ls *.mp3
```

3. Melihat daftar file dengan jangkauan tertentu

```
$ ls [a-j]*.txt
```

4. Mencari nama file dengan single karakter tertentu

```
$ echo ?o*
```

5. Menghapus file yang mempunyai *extension* .mp3

```
$ rm *.mp3
```

Percobaan 2 : Kunci Tab

Gunakan kunci Tab dan lihat hasilnya

```
$ ls  
$ ls a<Tab>  
$ cat j<Tab>  
$ cat jo<Tab>  
$ echo ?o*
```

Percobaan 3 : Perintah History

1. Bash shell menyimpan "history" perintah yang digunakan sebelumnya. Anda dapat mengakses history dalam beberapa cara. Cara paling mudah adalah menggunakan **Panah Atas**. Maka perintah sebelumnya akan ditampilkan.
2. Berikutnya, berikan Bash shell beberapa perintah untuk diingat. Masukkan perintah berikut dan tekan **Enter** pada setiap baris.

```
$ cd  
$ ls -l /etc  
$ ls -l  
$ whoami  
$ who
```

3. Untuk memeriksa apakah perintah ini ditambahkan pada history, dapat menggunakan perintah `history` untuk melihat semua perintah yang pernah dimasukkan.

```
$ history
```

4. Anda dapat memilih perintah sebelumnya dengan menggunakan **Panah Atas**, tetapi hal ini tidak efisien untuk perintah yang semakin bertambah banyak. Cara yang mudah menggunakan nomor pada perintah history atau mencarinya. Untuk memilih dan mengeksekusi perintah dengan nomor, masukkan kunci ! diikuti nomor perintah.

```
$ !<Nomor Perintah>          Contoh : !780
```

5. Anda dapat mencari perintah dengan menyertakan perintah yang diinginkan. Misalnya **!*etc*?** akan menjalankan perintah `ls -l /etc` yang sebelumnya digunakan.

```
$ !?etc?
```

6. Kemudian gunakan perintah history, maka akan terlihat perintah `ls -l /etc` yang kedua dan bukan **!*etc*?**

```
$ history
```

7. Apabila string tidak ditemukan pada perintah history maka akan terdapat pesan error.

```
$ !?wombat99?
```

8. Jika diketikkan **!who** maka yang dijalankan adalah perintah `who`. Tetapi bila Anda ketikkan **!whoa** maka yang dijalankan adalah perintah `whoami`.

```
$ !who
```

```
$ !whoa
```

Percobaan 4 : Lebih Lanjut Tentang Perintah Baris

1. Penggunaan tanda `~` yang merefensi ke *home directory*

```
$ cat ~/.bash_profile
```

2. Penggunaan tanda `~` yang merefensi ke *home directory* user lain

```
$ ls ~teacher/public_html
```

3. Penggunaan parameter/variabel `$`

```
$ cd $HOME/public_html
```

```
$ cd /home/teacher/public_html
```

4. Penggunaan tanda kurung kurawal { }

```
$ echo {a,b}
$ echo x{a,b}
$ touch file.txt
$ cp file.txt(-,save)
$ mkdir -p work/{inbox,outbox,pending}/{normal,urgent,important}
$ ls work
$ ls work/inbox
```

5. Perintah output `` atau \$() dan aritmatika -\${ }

```
$ echo "Hostname: `hostname`"
$ echo "Hostname: $(hostname)"
$ echo Area: `expr $X \* $Y`
$ echo Area: ${ $X * $Y }
```

6. Parameter backslash (\)

```
$ echo Your cost: \$5.00
$ echo "This long string will be echoed \
> back as one long line"
$ find / -name foo*
$ find / -name foo\*
```

7. Membatasi ekspansi dengan tanda petik

```
$ echo \*\*\* SALE \*\*\*
$ echo '*** SALE ***'
$ echo ?*** SALE ***?
$ echo "The current date is `date`"
$ echo 'The current date is `date`'
```

LATIHAN:

1. Login sebagai student di TTY 1. Lihatlah direktori yang dimasuki dengan perintah **pwd**.
2. Buatlah beberapa file dengan perintah **touch**. Gunakan tanda kurung { } untuk membuat beberapa file dan lihat hasilnya dengan perintah **ls**.

```
$ touch {report,memo,graph}_{sep,oct,nov,dec}_{a,b,c}_{1,2,3}
```
3. Buatlah beberapa direktori a_report september october november december dengan perintah **mkdir**. Gunakan **ls** untuk melihat hasilnya,

4. Buatlah sub direktori di dalam `a_reports` yang bernama `one two three` dan tampilkan hasilnya dengan perintah `ls`. Pastikan direktori aktif (*current directory*) berada di sub direktori `a_reports`.
5. Pindahkan semua file laporan “b” pada *home directory* dan kelompokkan berdasarkan bulan (`september`, `october`, `november` dan `december`), misalnya `graph_dec_b_1` dipindah ke sub direktori `december`. Tuliskan langkah-langkah perintah yang digunakan. Gunakan perintah baris yang tepat sehingga perintah yang digunakan lebih sederhana (lebih sedikit).
6. Kelompokkan file laporan “a” pada *home directory* berdasarkan nomor direktori (`one`, `two` dan `three`), misalnya `graph_dec_a_1` dipindah ke sub direktori `a_reports/one`. Gunakan perintah baris yang tepat.
7. Hapus file bulan September (yang terdapat string “`sep`”). Gunakan perintah baris yang tepat.
8. Pindahkan semua file laporan yang mengandung string “`a_2`” dan “`a_3`” ke sub direktori `a_reports/two` dan `a_reports/three`. Lakukan dari direktori yang aktif (*current directory*).
9. Kembali ke *home directory* dan lihatlah file yang tersisa. Apabila direktori ini berisi file laporan “c” berarti proses Anda sejauh ini sudah benar. Copy-kan laporan yang mengandung string “`c_1`” dan “`c_2`” ke sub direktori baru bernama `archive` di dalam sub direktori `tmp` (direktori standar Linux). Copy-kan juga laporan bulan Desember ke direktori `/tmp/archive` dengan pilihan `-i` pada perintah `cp`.
10. Tampilkan semua file laporan “c” pada *home directory* dengan menggunakan wildcard `*c*`. Kemudian hapus semua file laporan “c” tersebut. Pastikan tidak ada file yang dibuat pada latihan berada di *home directory* (sudah masuk ke sub direktori yang ada)

LAPORAN RESMI:

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