Praktikum 5-B

Bekerja Dengan Bash Shell

POKOK BAHASAN:

- ✓ History pada Bash Shell
- ✓ Membuat Bash Shell Script

TUJUAN BELAJAR:

Setelah mempelajari materi dalam bab ini, mahasiswa diharapkan mampu:

- ✓ Memahami shell pada sistem operasi Linux.
- ✓ Menggunakan feature history pada Bash Shell.
- ✓ Mengubah feature history pada Bash Shell.
- ✓ Mengubah prompt shell.
- ✓ Melakukan konfigurasi Bash Shell untuk menjalankan skrip secara otomatis.
- ✓ Membuat dan mengeksekusi shell script sederhana melalui editor vi.
- ✓ Memahami job control.
- ✓ Memahami stack.
- ✓ Menggunakan alias.

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 promp **#**.

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 diatasi hampir tidak ada, kecuali pada fasilitas pemrograman dan editing.

2 PROFILE

Pada saat login, program akan menjalankan beberapa program yaitu :

1. **/etc/profile**

   Berisi shell script yang berlaku untuk seluruh pengguna Linux.

2. **Profil untuk setiap pemakai**

   Pada home directory, login pertama kali akan memeriksa file **.bash_profile** . Bila tidak ada, maka file **.bash_login** akan dicari. Bila **.bash_login** tidak ada, maka dicari file bernama **.profile** .

3. **.bashrc**

   File ini akan dieksekusi untuk perpindahan dari satu shell ke shell yang lain melalui instruksi su.

4. **.bash_logout**

   Pada saat logout, maka bash akan mencari file **.bash_logout**. Bila ada, file tersebut akan dieksekusi sebelum logout

Isi dari /etc/profile:

```
# System wide environment and startup programs
# Functions and aliases go in /etc/bashrc

PATH="$PATH:/usr/X11R6/bin"
PS1="[\u@\h \W]\\$ "
umask 022

USER='id –un'
LOGNAME=$USER
MAIL="/var/spool/mail/$USER"

HOSTNAME='/bin/hostname'
HISTSIZE=1000
HISTFILESIZE=1000
```

```
Export PATH PS1 HOSTNAME HISTSIZE HISTFILESIZE USER LOGNAME MAIL
```

PATH merupakan daftar nama direktori. Bila sebuah instruksi diberikan dari prompt shell, maka instruksi tersebut akan dicari pada daftar tersebut.

PS1 adalah prompt dimana

\u = Nama User

\h = Nama Host

\W = Nama working direktory

3 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 kontrool sebagai berikut :

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

4 BASH-SCRIPT

Bash-script adalah file yang berisi koleksi program yang dapat dieksekusi. Untuk eksekusi bash script gunakan . sebelum file bash-script yang berarti eksekusi shell dan tanda ./ berarti file bash-script berada pada direktori actual.

5 JOB CONTROL

Job adalah sebuah eksekusi program yang diberikan kepada kernel. Sebuah Job dianggap selesai, bila eksekusi program tersebut berakhir. Eksekusi Job adalah sama dengan eksekusi program, baik proses *Background* maupun proses *Foreground*.

6 EDITOR vi

Vi adalah full screen editor, artinya editor tersebut dapat memanfaatkan fasilitas satu layar penuh. Vi mempunyai 2 buah modus, yaitu :

- Command line

  Editor vi menginteprestasikan input sebagai instruksi untuk dieksekusi oleh editor, contoh seperti mencari teks, mengganti teks secara otomatis dan lainnya.

- Editing

  Editor vi menginteprestasikan input sebagai teks yang akan dimasukkan ke dalam buffer editor. Pada bagian bawah layar akan tampil teks "INSERTING".

Pada awal vi dijalankan, maka program memasuki command mode. Dengan menekan tombol "i" maka akan memasuki editing. Untuk kembali ke command mode, tekan tombol Esc.

Kunci-kunci teks editor vi dapat dilihat pada tabel sebagai berikut :

Kunci	Keterangan	
H	Pindah kursor ke kiri satu karakter	
J	Pindah kursor ke kanan satu karakter	
K	Pindah kursor ke atas	
L	Pindah kursor ke bawah	
O	Menyisipkan teks (satu baris setelah posisi kursor)	Untuk keluar dari 5 model kunci *insert* di samping ini dan mengaktifkan kunci-kunci lain, maka kita harus menekan tombol Esc terlebih dahulu.
I	Menyisipkan teks (di sebelah kiri posisi kursor)	
A	Menyisipkan teks (di sebelah kanan posisi kursor)	
I (shift i)	Menyisipkan teks (di posisi awal baris)	
A (shift a)	Menyisipkan teks (di posisi akhir baris)	

X	Menghapus 1 huruf (di sebelah kanan posisi kursor)
Dw	Manghapus 1 kata (di sebelah kanan posisi kursor)
Dd	Menghapus 1 baris (di sebelah kanan posisi kursor)
Yy	Mengkopi 1 baris
2yy	Mengkopi 2 baris
P	(*Paste*) Menampilkan baris kalimat yang sudah dikopi dengan kunci yy
Cw	Mengganti 1 kata yang telah ditulis di sebelah kanan posisi kursor dengan kata lain
Cc	Mengganti 1 baris kalimat yang telah ditulis di sebelah kanan posisi kursor dengan kalimat lain
ctrl-b	Mundur satu layar
ctrl-f	Maju satu layar
ctrl-d	Maju setengah layar
B	Menggerakkan kursor ke kiri satu kata
W	Manggerakkan kursor ke kanan satu kata
^	Pergi ke awal baris
$	Pergi ke akhir baris
U	Membatalkan perintah yang terakhir kali
U	Membatalkan seluruh perubahan teks pada baris tempat kursor berada
:!	Keluar untuk sementara dari editor vi dan menjalankan perintah yang lain
:wq	Write dan quite, simpan berkas dan keluar
:q!	Keluar vi tanpa menyimpan

:se all	Menampilkan semua pilihan set status
:se nu	Menampilkan nomor baris pada kiri layar
/string	Mencari string ke arah depan
?string	Mencari string ke arah belakang
N	Meneruskan pencarian untuk arah yang sama
N	Meneruskan pencarian untuk arah yang berbeda

Percobaan 6 : Membuat Bash-script dan menjalankannya

1. Membuat file p1.sh

```
$ vi p1.sh
echo "Program bash Script"
```

2. Mengubah program menjadi executable

```
$ ls –l p1.sh
$ chmod +x p1.sh
$ ls –l p1.sh
```

3. Menjalankan script

```
$ bash p1.sh
$ sh p1.sh
$ . p1.sh
$ ./p1.sh
```

4. Konvensi dalam pembuatan script shell dinyatakan sebagai `#!/bin/bash`. Tambahkan pada file p1.sh konvensi tersebut

```
$ vi p1.sh
#!/bin/bash
echo "Program bash script"
```

5. Buatlah file p2.sh

```
$ vi p2.sh
#!/bin/bash
echo "Program 2 bash script"
```

6. Menjalankan beberapa program shell dalam satu baris instruksi yang dipisahkan dengan tanda ;

```
$ cat p1.sh ; cat p2.sh
$ ./p1.sh ; ./p2.sh
```

5. Menjalankan script sebagai prosees background, sehingga prompt tidak

Percobaan 7 : Job Control

1. Proses foreground

   ```
   $ ps x
   ```

2. Proses background

   ```
   $ ps x > hasil &
   ```

3. Setiap job mempunyai PID yang tunggal (unique). Untuk melihat jobs yang aktif

   ```
   $ jobs
   ```

4. Buatlah file `ploop.sh`. File ini tidak akan pernah berhenti kecuali ditekan Ctrl-C

   ```
   $ vi ploop.sh
   #!/bin/bash
   while [ true ]
   do
        sleep 10
        echo "Hallo"
   done
   ```

5. Buatlah file ploop.sh menjadi executable. Jalankan program, akan ditampilkan kata Hallo setiap 10 detik. Untuk keluar program, tekan Ctrl-C (^C)

   ```
   $ chmod +x ploop.sh
   $ ./ploop.sh
   ```

Percobaan 8 : Manipulasi stack untuk Direktori

1. Instruksi `dirs` digunakan untuk melihat stack direktori, pada output h ditampilkan direktori home ~

   ```
   $ dirs
   ```

2. Membuat 3 buah direktori

   ```
   $ mkdir marketing sales support
   ```

3. Instruksi `dirs` digunakan untuk melihat stack direktori, pada output h ditampilkan direktori home ~

   ```
   $ dirs
   ```

4. Membuat 3 buah direktori

Percobaan 9 : Alias

1. Alias adalah mekanisme untuk memberi nama alias pada satu atau sekelompok instruksi. Untuk melihat alias yang sudah terdaftar pada system :

   ```
   $ alias
   ```

2. Membuat beberapa alias

   ```
   $ alias del='rm –i'
   $ alias h='history'
   ```

3. Gunakan instruksi hasil alias

   ```
   $ ls
   $ del hasil
   $ h | more
   ```

4. Untuk menghapus alias gunakan instruksi unalias

   ```
   $ unalias del
   $ del files
   ```
   (Terdapat Pesan Kesalahan, mengapa ?)

LATIHAN:

1. Eksekusi seluruh profile yang ada :
   a. Edit file profile `/etc/profile` dan tampilkan pesan sebagai berikut :

      ```
      echo 'Profile dari /etc/profile'
      ```

   b. Asumsi nama anda `student`, maka edit semua profile yang ada yaitu :

      ```
      /home/student/.bash_profile
      /home/. student/.bash_login
      /home/student/.profile
      /home/student/.bashrc
      ```

   c. Ganti nama `/home/student` dengan nama anda sendiri. Pada setiap file tersebut, cantumkan instruksi echo, misalnya pada `/home/student/.bash_profile`:

      ```
      echo "Profile dari .bash_profile"
      ```

   d. Lakukan hal yang sama untuk file lainnya, sesuaikan tampilan dengan nama file yang bersangkutan.
2. Jalankan instruksi subtitute user, kemudian keluar dengan perintah exit sebagai berikut :

   ```
   $ su student
   ```

```
$ exit
```

kemudian gunakan opsi – sebagai berikut :

```
$ su – student
$ exit
```

Jelaskan perbedaan kedua utilitas tersebut.

3. Logout

   a. Edit file `.bash_logout`, tampilkan pesan dan tahan selama 5 detik, sebelum eksekusi logout

   ```
   Echo “Terima kasih atas sesi yang diberikan”
   Sleep 5
   Clear
   ```

   b. Edit file `.bash_logout`, tampilkan pesan dan tahan selama 4 detik, sebelum eksekusi logout

4. History

   a. Ganti nilai HISTSIZE dari 1000 menjadi 20

   ```
   $ HISTSIZE=20
   $ h
   ```

   b. Gunakan fasilitas history dengan mengedit instruksi baris ke 5 dari instruksi yang terakhir dilakukan.

   ```
   $ !-5
   ```

   c. Ulangi instruksi yang terakhir. Gunakan juga ^P dan ^N untuk bernavigasi pada history buffer

   ```
   $ !!
   ```

   d. Ulaingi instruksi pada history buffer nomor tertentu, misalnya nomor 150

   ```
   $ !150
   ```

   e. Ulangi instruksi dengan prefix “ls”

   ```
   $ !ls
   $ !?ls?
   ```

   Jelaskan perbedaan instruksi diatas

5. Prompt String (PS)

a. Edit file `.bash_profile`, ganti prompt PS1 dengan '>'. Instruksi export diperlukan dengan parameter nama variable tersebut, agar perubahan variable PS1 dikenal oleh semua shell

```
PS1='> '
export PS1
```

Eksperimen hasil PS1 :

```
$ PS1="\! > "
69 > PS1="\d > "
Mon Sep 23 > PS1="\t > "
10:10:20 > PS1="Saya=\u > "
Saya=stD02001 > PS1="\w >"
~ > PS1=\h >"
```

b. Ubahlah warna shell prompt dengan warna biru dan berkedip.

6. Bash script

a. Buat 3 buah script p1.sh, p2.sh, p3.sh dengan isi masing-masing :

p1.sh

```
#! /bin/bash
echo "Program p1"
ls –l
```

p2.sh

```
#! /bin/bash
echo "Program p2"
who
```

p3.sh

```
#! /bin/bash
echo "Program p3"
ps x
```

b. Jalankan script tersebut sebagai berikut dan perhatikan hasilnya :

```
$  ./p1.sh ; ./p3.sh ; ./p2.sh
$  ./p1.sh &
$  ./p1.sh $ ./p2.sh & ./p3.sh &
$  ( ./p1.sh ; ./p3.sh ) &
```

7. Jobs

a. Buat shell-script yang melakukan loop dengan nama `pwaktu.sh`, setiap 10 detik, kemudian menyimpan tanggal dan jam pada file hasil.

```
#!/bin/bash
while [ true ]
```

```
do
   date >> hasil
   sleep 10
done
```

b. Jalankan sebagai background; kemudian jalankan satu program (utilitas find) di background sebagai berikut :

```
$ jobs
$ find / -print > files 2>/dev/null &
$ jobs
```

c. Jadikan program ke 1 sebagai foreground, tekan ^Z dan kembalikan program tersebut ke background

```
$ fg %1
$ bg
```

d. Stop program background dengan utilitas kill

```
$ ps x
$ kill [Nomor PID]
```

LAPORAN RESMI:

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