Praktikum 5

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

- Editing

  Editor vi mengintepretasikan 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 k ursor)	
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

TUGAS PENDAHULUAN :

Jawablah pertanyaan-pertanyaan di bawah ini :

1. Apa yang dimaksud dengan shell dan sebuatkan shell yang ada di system operasi Linux.
2. Apa yang dimaksud dengan profile pada Bash Shell.
3. Apa yang Anda ketahui mengenai file `.bashrc`.
4. Apa yang dimaksud dengan history pada Bash Shell. Apa kegunaan perintah history, sebutkan cara-cara untuk mengetahui history perintah-perintah yang pernah digunakan oleh user!
5. Cobalah menggunakan editor vi untuk mengetik dan pahami perintah-perintah yang ada seperti yang terdapat pada dasar teori (untuk dilakukan, tidak perlu dijawab sebagai tugas pendahuluan). Perintah-perintah yang penting : insert huruf(kalimat), delete (per huruf, per kata dan per baris), simpan file dan keluar dari editori vi.

PERCOBAAN:

1. Login sebagai user.
2. Bukalah Console Terminal dan lakukan percobaan-percobaan di bawah ini kemudian analisa hasil percobaan.
3. Selesaikan soal-soal latihan.

Percobaan 1 : Profile

1. File .bash_profile dijalankan pada home direktori pemakai yang login. File .bash_profile adalah *hidden file*, sehingga untuk melihatnya gunakan opsi a pada instruksi ls.

```
$ ls –a
$ more .bash_profile
```

2. File .bash_logout akan diekseksi sesaat sebelum logout, berfungsi sebagai *house clearing jobs*, artinya membersihkan semuanya, misalnya menghapus temporary file atau job lainnya. Melihat file .bash_logout dengan instruksi

```
$ cat .bash_logout
```

Percobaan 2 : Menggunakan Feature History Bash

1. Bash shell menyimpan ”history” perintah yang digunakan sebelumnya. Anda dapat mengaksis 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 menggunkaan 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 tidka ditemukan pada perintha 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
   ```

9. Anda bisa menggantikant string pada perintah history, terutama pada perintah yang panjang. Misalnya ketik **cat /bin/bash | strings | grep shell | less** dan tekan **Enter**. Maka akan menampilkan semua string pada file `/bin/bash` yang berisi kata ”shell”. Untuk keluar tekan q. Jika ingin menampilkan kata ”alias”, maka Anda tidak perlu mengetik perintah yang panjang lagi, tetapi cukup ketik **^shell^alias^** dan tekan **Enter** maka akan menggantikan kata ”shell” dengan ”alias”.

   ```
   $ cat /bin/bash | strings | grep shell | less
   $ ^shell^alias^
   ```

Percobaan 3 : Mengubah Feature History Bash

1. Bash shell akan menyimpan perintah history meskipun telah log out dan log in kembali. File `.bash_history` menyimpan file history yang terdapat pada home directory.

   ```
   $ cd
   ```

2. Lihat beberapa baris pada file .bash_history dengan ketik **tail .bash_history** dan tekan **Enter**. File ini bukan file yang up to date.

   ```
   $ tail .bash_history
   ```

3. Ketik **history** dan tekan **Enter**. Maka akan terlihat baris terakhir adalah perintah history dan baris sebelumnya adalah `tail .bash_history`. Perintah history bersifat up to date, karena disimpan pada memory sistem.

   ```
   $ history
   ```

4. Ketik perintah berikut

   ```
   $ echo 'Ini perintah saya'
   ```

5. Log out dan log in kembali sebagai user yang sama. Ketik **history** dan tekan **Enter**. Maka perintah `echo 'Ini perintah saya'` akan berada pada baris terakhir. Lihat file `.bash_history`, maka perintah tsb akan terdapat pada file `.bash_history`.

   ```
   $ history
   $ tail .bash_history
   ```

6. Ketik **history|less** untuk melihat perintah history terakhir pada screen. Tekan spacebar untuk melihat file lebih banyak. Untuk keluar tekan q

   ```
   $ history|less
   ```

7. Untuk melihat berapa banyak perintah history yang ada pada file ketik berikut dan output yang keluar serupa di bawah ini

   ```
   $ wc -l .bash_history
   1000       .bash_history
   ```

8. Output menunjukkan bahwa 1000 perintah history disimpan pada file history. Untuk melihat jangkauan (limit) perintah history digunakan variabel **HISTSIZE**. Untuk melihat jangkauan history ketik sebagai berikut

   ```
   $ set|grep HISTSIZE
   ```

9. Bila ingin memperbesar jangkauan file history, maka ubahlah variabel **HISTSIZE** pada skrip startup yang disebut .bashrc pada home directory.

   ```
   $ echo 'HISTSIZE=5000' >> .bashrc
   ```

10. Log out dan log in kembali sebagai user yang sama. Lihat perubahan variabel **HISTSIZE**.

    ```
    $ set|grep HISTSIZE
    ```

11. Ketikkan perintah history beberapa kali, maka perintah ini akan disimpan pada BASH history meskipun yang diketikkan perintahnya sama.

12. Anda dapat melakukan konfigurasi BASH agar tidak menambah perintah ke history jika perintah yang diketikkan sama dengan sebelumnya. Hal ini dilakukan dengan menambahkan variabel **HISTCONTROL** dan diberikan nilai **ignoredups** pada file `.bashrc`

    ```
    $ echo 'HISTCONTROL=ignoredups' >> .bashrc
    ```

13. Log out dan log in kembali sebagai user yang sama . Ketikkan history beberapa kali dan perhatikan berapa kali history muncul.

Percobaan 4 : Mengubah Prompt Shell

1. Prompt Bash shell dikonfigurasi dengan men-setting nilai variabel PS1. Selain menampilkan string statik sebagai prompt, Anda dapat menampilkan menjadi dinamis. Contohnya, apabila ingin menunjukkan *current directory* atau *current time.* Ketik **PS1='\t:'** dan tekan **Enter** untuk menampilkan waktu sistem da lam format 24 jam sebagai prompt Bash. Format dalam HH:MM:SS

   ```
   $ PS1='\t:'
   ```

3. Untuk menampilkan format 12 jam dengan indikator am dan pm ketik sebagai berikut :

   ```
   $ PS1='\t:'
   ```

4. Kebanyakan orang menginginkan prompt Bash menampilkan *current working directory*. Direktory dapat ditampilkan dalam bentuk keseluruhan path atau hanya nama direktory. Karakter `\w` menampilkan hanya nama direktory. Jika *current directory* adalah home directory, maka tampil prompt ~:

   ```
   $ PS1='\w:'
   ```

5. Ketik **cd /usr/sbin** untuk melihat prompt `/usr/sbin`:

   ```
   $ cd /usr/sbin
   ```

5. Ketik **PS1='\W:'** untuk melihat prompt `sbin`:

   ```
   $ PS1='\W:'
   ```

6. Ada beberapa prompt BASH lain yang dapat diubah, yaitu PS2, PS3 dan PS4. Prompt PS2 digunakan sebagai prompt sekunder. Untuk melihat bagaimana penggunaannya, ketik **echo 'Hello** (tanpa diakhiri penutup quote) dan tekan **Enter**. Simbol lebih besar dari (>) akan muncul. Hal ini memberitahukan bahwa BASH menunggu Anda menyelesaikan perintah. Ketik penutup quote (') dan tekan **Enter**. Perintah ini akan menyelesaikan prompt PS2, kata **"Hello, "** muncul diikuti dengan prompt PS1 pada baris baru.

   ```
   $ echo 'Hello
   >'
   ```

7. Anda dapat mengubah prompt PS2 seperti mengubah prompt PS1. Ketik perintah berikut :

   ```
   $ PS2='Selesai memasukkan perintah Anda:'
   ```

8. Kemudian ketik echo ’Hello (tanpa diakhiri penutup quote) dan tekan Enter. Pada baris berikutnya akan muncul **Selesai memasukkan perintah Anda:**. Kemudian ketikkan penutup quote (**’**) dan tekan **Enter**. Jika perintah selesai, maka kata **Hello** akan muncul diikuti prompt PS1 pada baris baru.

```
$ echo ’Hello
Selesai memasukkan perintah Anda:’
```

9. Prompt BASH dapat ditampilkan berwar na dengan melakukan setting *color-setting string* . Sebagai contoh, prompt BASH di-set dengan `\w\$`, akan menampilkan *current working directory* yang diikuti $ (atau # jika anda login sebagai root). Untuk setting warna menjadi biru ketikkan berikut :

```
$ PS1=’\033[0;34m\w\$ \033[0;37m’
```

10. Untuk mendapatkan prompt warna merah ketikkan berikut :

```
$ PS1=’\033[0;31m\w\$ \033[0;37m’
```

30=hitam, 31=merah, 32=hijau, 34=biru, 35=ungu, 36=cyan, 37=putih.

11. Bila menginginkan beberapa warna, ketikkan perintah berikut :

```
$ PS1=’\033[0;31m\w\033[0;32m\$ \033[0;37m’
```

12. Anda bisa menampilkan atribut visual seperti lebih terang, berkedip dan warna kebalikannya. Untuk menampilkan prompt yang lebih terang, atribut control diganti 1, seperti perintah berikut :

```
$ PS1=’\033[1;34m\w\033[1;32m\$ \033[0;37m’
```

13. U ntuk menampilkan prompt dengan warna berkebalikan, atribut control diganti 7, seperti perintah berikut :

```
$ PS1=’\033[7;34m\w\033[7;32m\$ \033[0;37m’
```

14. Untuk menampilkan prompt berkedip, atribut control diganti 5, seperti perintah berikut :

```
$ PS1=’\033[5;34m\w\033[5;32m\$ \033[0;37m’
```

Percobaan 5 : Menambahkan otomatisasi ke Prompt Shell

1. Pastikan Anda berada di home directory

   ```
   $ cd ~
   ```

2. Buatlah skrip sederhana untuk mengurut daftar file. Anda dapat menggunakan teks editor, tetapi karena hanya satu baris, gunakan perintah echo untuk membuat file.

   ```
   $ echo 'sort ~/list > ~/r13; mv ~/r13 ~/list' > ~/sorter
   ```

3. Buatlah file skrip diatas menjadi file executable

   ```
   $ chmod +x sorter
   ```

4. Jalankan program sorter diatas setiap shell Bash menampilkan prompt PS1. Untuk melakukannya, buatlah variable **PROMPT_COMMAND** dimana nilainya adalah nama dari program `sorter`.

   ```
   $ PROMPT_COMMAND=~/sorter
   ```

5. Ketikkan **echo 'John Smith:13001'>>list** dan tekan **Enter**. Jika file `list` tidak ada, akan dibuat secara otomatis, tetapi jika sudah ada, string 'John Smith:13001' akan ditambahkan.

   ```
   $ echo 'John Smith:13001'>>list
   ```

6. Ketik **cat list** dan tekan **Enter.** Maka Anda akan melihat isi file `list`. Pada saat ini, file mungkin mempunyai hanya satu baris sehingga tidak dapat dilihat apakah file sudah terurut.

   ```
   $ cat list
   ```

7. Masukkan beberapa perintah serupa dengan point 5 tetapi dengan nama dan nomor yang berbeda. Kemudian ketik **cat list** dan tekan **Enter.**

```
$ echo 'Anita:13002'>>list
$ echo 'Samantha:13003'>>list
$ echo 'Patrik:13004'>>list
$ echo 'Sponse Bob:13005'>>list
$ echo 'Lisa:13006'>>list
$ echo 'Squid:13007'>>list
```

8. Apabila Anda tidak menginginkan Shell Bash menampilkan file terurut sepanjang waktu, Anda tidak perlu menambahkan variable `PROMPT_COMMAND=~/sorter` pada file konfigurasi seperti .bashrc. Bila Anda ingin BASH berhenti menjalankan program sorter, maka ketikkan variable **PROMPT_COMMAND=** dan tekan **Enter** atau log out dan login kembali.

```
$ PROMPT_COMMAND=
```

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
```

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
```

Perc obaan 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 variab le 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.