

VERSION 3.0  
FEBRUARI , 2019



# [SISTEM OPERASI]

## MODUL 5 – SHELL PROGRAMMING

TIM PENYUSUN: - DENAR REGATA AKBI S.KOM M.KOM  
- MOHAMMAD ZIDAN

PRESENTED BY: LAB. TEKNIK INFORMATIKA  
UNIVERSITAS MUHAMMADIYAH MALANG

## [SISTEM OPERASI]

---

### CAPAIAN PEMBELAJARAN MATA KULIAH

#### SUB CAPAIAN PEMBELAJARAN MATA KULIAH

1. Mahasiswa mampu Memahami shell pada sistem operasi Linux.
2. Mahasiswa mampu Melakukan konfigurasi Bash Shell untuk menjalankan skrip secara otomatis.
3. Mahasiswa mampu Membuat dan mengeksekusi shell script sederhana.

---

### KEBUTUHAN HARDWARE & SOFTWARE

- PC/Laptop
- Sistem Operasi Linux

---

### MATERI POKOK

#### SHELL

Apa itu shell ? shell adalah program (penterjemah perintah) yang menjembatani user dengan sistem operasi dalam hal ini kernel (inti sistem operasi), umumnya shell menyediakan prompt sebagai user interface, tempat dimana user mengetikkan perintah-perintah yang diinginkan baik berupa perintah internal shell (internal command), ataupun perintah eksekusi suatu file program (eksternal command), selain itu shell memungkinkan user menyusun sekumpulan perintah pada sebuah atau beberapa file untuk dieksekusi sebagai program.

##### I. Macam - macam shell

Tidak seperti sistem operasi lain yang hanya menyediakan satu atau 2 shell, sistem operasi dari keluarga unix misalnya linux sampai saat ini dilengkapi oleh banyak shell dengan kumpulan perintah yang sangat banyak, sehingga memungkinkan pemakai memilih shell mana yang paling baik untuk membantu menyelesaikan pekerjaannya, atau dapat pula berpindah-pindah dari shell yang satu ke shell yang lain dengan mudah, beberapa shell yang ada di linux antara lain:

- Bourne shell(sh),
- C shell(csh),
- Korn shell(ksh),
- Bourne again shell(bash),
- dsb.

Masing - masing shell mempunyai kelebihan dan kekurangan yang mungkin lebih didasarkan pada kebutuhan pemakai yang makin hari makin meningkat, untuk dokumentasi ini shell yang digunakan adalah bash shell dari GNU, yang merupakan pengembangan dari Bourne shell dan mengambil beberapa feature (keistimewaan) dari C shell serta Korn shell, Bash shell merupakan shell yang cukup banyak digunakan pemakai linux karena kemudahan serta banyaknya fasilitas perintah yang disediakan.

#### cara meihat versi shell

```
root@kali:~# echo $BASH_VERSION
5.0.0(1)-release
root@kali:~#
```

## II. Pemrograman Shell

Yaitu menyusun atau mengelompokkan beberapa perintah shell (internal ataupun eksternal command) menjadi kumpulan perintah yang melakukan tugas tertentu sesuai tujuan penyusunnya. Kelebihan shell di linux dibanding sistem operasi lain adalah bahwa shell di linux memungkinkan kita untuk menyusun serangkaian perintah seperti halnya bahasa pemrograman (interpreter language), melakukan proses I/O, menyeleksi kondisi, looping, membuat fungsi, dsb. adalah proses - proses yang umumnya dilakukan oleh suatu bahasa pemrograman, jadi dengan shell di linux kita dapat membuat program seperti halnya bahasa pemrograman, untuk pemrograman shell pemakai unix atau linux menyebutnya sebagai script shell.

## III. Kebutuhan Dasar

Sebelum mempelajari pemrograman Bash shell di linux sebaiknya anda telah mengetahui dan menggunakan perintah - perintah dasar shell baik itu internal command yang telah disediakan shell maupun eksternal command atau utility, seperti

- cd, pwd, exit, logout, fg, bg, ls, mkdir, rmdir, mv, cp, rm, clear, ...
- utilitas seperti cat, cut, paste, chmod
- redirection (cara mengirim output ke file atau menerima input dari file), menggunakan operator redirect >, >>, <, <<, contohnya:

#### **ls > data**

hasil ls dikirim ke file data, jika file belum ada akan dibuat tetapi jika sudah ada isinya akan ditimpa.

#### **ls >> data**

hampir sama, bedanya jika file sudah ada maka isinya akan ditambah di akhir file.

#### **cat < data**

file **data** dijadikan input oleh perintah **cat**

- pipa (output suatu perintah menjadi input perintah lain), operatornya : | ,  
contoh:  
**ls -l | sort -s**  
output perintah ls -l (long) menjadi input perintah sort -s (urutkan secara descending).  
**ls -l | sort -s | more**  
**cat <data | sort > databaru**
- Wildcard dengan karakter \*, ?, [ ], contohnya:  
**ls i\***  
tampilkan semua file yang dimulai dengan i  
**ls i?i**  
tampilkan file yang dimulai dengan i, kemudian sembarang karakter tunggal, dan diakhiri dengan i  
**ls [ab]\***  
tampilkan file yang dimulai dengan salah satu karakter a atau b

## MATERI PRAKTIKUM (PERCOBAAN)

### PERCOBAAN PRAKTIKUM

#### I. Membuat Bash Script sederhana

1. Masuk ke text editor (nano, pico, vi), lalu isikan perintah dibawah ini

```
#!/bin/bash
echo "Ini File Bash saya, bash mu endi??"
```

2. Simpan dengan nama **coba**
3. Ubah Permissionnya

```
root@kali:~/modulSO# chmod 755 coba
root@kali:~/modulSO#
```

4. Untuk menjalankan file bash ada dua cara
  - menggunakan perintah ./nama\_file

```
root@kali:~/modulSO# ./coba
Ini File Bash saya, bash mu endi??
root@kali:~/modulSO#
```

- menggunakan perintah bash [spasi] nama\_file

```
root@kali:~/modulSO# bash coba
Ini File Bash saya, bash mu endi??
root@kali:~/modulSO#
```

tanda **#!** pada **/bin/bash** dalam script coba adalah perintah yang diterjemahkan ke kernel linux untuk mengeksekusi path yang disertakan dalam hal ini program **bash** pada direktory **/bin**.

#### II. Membuat Script bash yang berisi command line

1. Buka text editor (vi, nano, pico,dll) lalu ketikan perintah dibawah ini

```
#!/bin/bash
#myinfo

#membersihkan tampilan layar
clear

#menampilkan informasi
echo -n "Waktu system      :"; date
echo -n "Versi Kernel      :"; uname -r
echo -n "Directory          :"; pwd
```

2. simpan dengan nama **coba\_info**

## 3. ubah permission

```
root@kali:~/modulSO# chmod 755 coba_info
root@kali:~/modulSO#
```

## 4. Jalankan file tersebut

```
Waktu system      :Fri Feb  1 09:47:20 WIB 2019
Versi Kernel      :4.19.0-kali1-amd64
Directory         :/root/modulSO
root@kali:~/modulSO#
```

## III. Variable dalam script bash

Dalam Membuat variable ada beberapa hal yang perlu di perhatikan, yaitu

- dimulai dengan huruf atau underscore.
- hindari pemakaian spesial karakter seperti \*,\$,#,dll.
- bash bersifat case sensitive.
- dilarang di awali dengan angka

Script 1 :

```
#!/bin/bash
nama="mas bro"
kalimat1="Hi $nama, kamu mantul!!" #menyisipkan variabel nama
kalimat2="aku rapopo :)"
echo $kalimat1;
echo $kalimat2;
```

Hasil :

```
root@kali:~/modulSO# ./coba_var
Hi mas bro, kamu mantul!!
aku rapopo :)
root@kali:~/modulSO#
```

Script 2 :

```
#!/bin/bash
kalimat="anda berada pada direktori :"
lokasi=`pwd` ;
echo $kalimat;
echo $lokasi;
```

Hasil 2 :

```
root@kali:~/modulSO# ./coba_var2
anda berada pada direktori :
/root/modulSO
root@kali:~/modulSO#
```

#### IV. Membuat inputan dengan bash script

Script :

```
#!/bin/bash
#rd1

echo -n "Nama mu :";
read nama;

echo "Hi $nama, bagi IG nya dong ";
echo "Jawaban :";
read jawab;
echo "kata $nama, $jawab";
```

Hasil :

```
root@kali:~/modulSO# ./coba_input
Nama mu :inisial *
Hi inisial *, bagi IG nya dong
Jawaban :
gah
kata inisial *, gah
root@kali:~/modulSO#
```

## V. Operasi bilangan dengan bash script

Script :

```
#!/bin/bash
#mat1

a=10
b=5
#memakai let
let jumlah=$a+$b
let kurang=$a-$b
let kali=$a*$b
#memakai expr
bagi=`expr $a / $b`

echo "$a+$b=$jumlah"
echo "$a-$b=$kurang"
echo "$a*$b=$kali"
echo "$a/$b=$bagi"
```

Hasil :

```
root@kali:~/modul50# ./coba_bil
10+5=15
10-5=5
10*5=50
10/5=2
root@kali:~/modul50#
```

## VI. Percabangan dengan bash script

Script :

```
#!/bin/bash
echo "silahkan input kata kunci :"
read pil;
key="bash";
key2="bash2";
if [ $pil == $key ];
then
    echo "anda berhasil menebak kata ke-1";
elif [ $pil == $key2 ];
then
    echo "anda berhasil menebak kata ke-2";
else
    echo "Ssalah"
    exit 1
fi
```



Hasil : `root@kali:~/modul50# ./coba_if`  
 silahkan input kata kunci :  
 bash  
 anda berhasil menebak kata ke-1  
`root@kali:~/modul50# ./coba_if`  
 silahkan input kata kunci :  
 bash2  
 anda berhasil menebak kata ke-2  
`root@kali:~/modul50# ./coba_if`  
 silahkan input kata kunci :  
 lol  
 Ssalah  
`root@kali:~/modul50#`

## VII. While dengan bash script

Script : `#!/bin/bash`  
`lagi='y'`  
`while [ $lagi == 'y' ] || [ $lagi == 'Y' ];`  
`do`  
   `clear`  
   `echo "1. Bakso ";`  
   `echo "2. Gado-Gado ";`  
   `echo "3. Exit ";`  
   `read -p "Pilihan anda [1-3] :" pil;`  
  
   `if [ $pil == 1 ];`  
   `then`  
     `echo "silahkan bayar 1500"`  
   `elif [ $pil == 2 ];`  
   `then`  
     `echo "silahkan ambil sendiri :P"`  
   `elif [ $pil == 3 ];`  
   `then`  
     `exit 0`  
   `else`  
     `echo "Sorry, tidak tersedia"`  
     `exit 1`  
   `fi`  
  
   `echo -n "beli lagi ? (y/t) :";`  
   `read lagi;`  
`done`

Hasil :

```
1. Bakso
2. Gado-Gado
3. Exit
Pilihan anda [1-3] :1
silahkan bayar 1500
beli lagi ? (y/t) :☐
```

```
1. Bakso
2. Gado-Gado
3. Exit
Pilihan anda [1-3] :2
silahkan ambil sendiri :P
beli lagi ? (y/t) :☐
```

```
1. Bakso
2. Gado-Gado
3. Exit
Pilihan anda [1-3] :3
root@kali:~/modul50# ☐
```

---

## KERJA

### KEGIATAN 1

1. Silahkan membuat kalkulator menggunakan bash script, dengan ketentuan :
  - a. Angka merupakan inputan dari user
  - b. Minimal dapat melakukan penjumlahan, pengurangan, perkalian dan pembagian
2. Buat inputan tentang identitas diri anda
  - a. Dapat menerima input dari user
  - b. Identitas diri berupa nama, nim, tempat lahir, jenis kelamin, alamat
3. Rekam kegiatan anda menggunakan TTYREC

---

**RUBRIK PENILAIAN**

Jelaskan secara rinci kriteria penilaian dan poin penilaian

KRITERIA	POIN
Dapat menerima input angka dari user	20
Terdapat fitur penjumlahan, pengurangan, perkalian, pembagian	40
Dapat menerima input biodata dari user	20
Menampilkan hasil biodata pada terminal	10
Mengerjakan salah satu percobaan praktikum	10