

# Bab 11

## Bash Script

Program D3 PJJ PENS ITS

## Tujuan

- Setelah mempelajari materi dalam bab ini, mahasiswa diharapkan mampu:
  - Mempelajari elemen dasar shell script
  - Membuat program shell interaktif
  - Menggunakan parameter dalam program
  - Mempelajari test kondisi serta operator logic yang terkait dengan instruksi test
  - Mengenal variable built-in dari shell
  - Membuat aplikasi dengan shell menggunakan konstruksi if-then-else
  - Menggunakan struktur case - esac.
  - Loop dengan while, for, do while.

# Materi

- Parameter
- Test kondisi dan operator logika
- Variable built-in
- Konstruksi if-then-else
- Struktur case - esac.
- Loop dengan while, for, do while.

# Shell Script

- Shell script dibuat dengan editor teks (ASCII editor) dan umumnya diberikan ekstensi “.sh”.
- Script selalu diawali dengan komentar, yang dimulai dengan tanda #, disambung dengan ! dan nama shell yang digunakan.

```
#!/bin/sh             
# Program shell      
#  
var1=x               
var2=8
```

## Variabel (1)

- Variable shell adalah variable yang dapat mempunyai nilai berupa nilai String.

```
nama_var = nilai_var
```

- Variable harus dimulai dengan alfabet, disusul dengan alfanumerik dan karakter lain.
- Shell membedakan huruf besar dan huruf kecil (*case sensitive*)

```
VPT=poltek
```

```
i=5
```

- Pemberian nilai variable tidak boleh dipisahkan dengan spasi

```
VPT =poltek          ##error
```

```
VPT= poltek         ##error
```

## Variabel (2)

- Untuk melihat nilai/isi dari sebuah variable, gunakan tanda \$ di depan nama variable tersebut. VPT=poltek

```
echo $VPT
```

```
Gaji=450000
```

```
echo $Gaji
```

```
echo $VPT $Gaji
```

- Bila menggunakan string yang terdiri dari lebih dari satu kata, maka string tersebut harus berada dalam tanda kutip atau apostrof

```
VPT=poltek
```

```
VPT2="poltek elektronika ITS"
```

## Membaca Keyboard

- Nilai variable dapat diisi melalui keyboard (stdin) dengan instruksi `read`.

```
read nilai
```

## Parameter

- Sebuah program shell dapat mempunyai parameter sebanyak 9 buah dan direpresentasikan melalui variable khusus yaitu variable `$_`, `$2`, `$3`, `$4`, `$5`, `$6`, `$7`, `$8` dan `$9`.
- Nama program shell (nama script) direpresentasikan melalui variable `$0`.
- Jumlah parameter dinyatakan sebagai `$#`. Bila tidak memberikan parameter, maka nilai `$#` adalah 0.
- Shell variable `$*` menyatakan seluruh string yang menjadi parameter / argumen sebuah script (`$@` mempunyai arti yang sama).
- `$$` menyatakan nomor proses id (pid) dari script yang dijalankan. Pid ini akan terus berubah (umumnya) menaik, setiap kali proses berjalan.

## Status Exit

- Setiap program setelah selesai dieksekusi akan memberikan informasi melalui variable spesial `$?`. Indikasi yang diberikan adalah :
  - Bila program berakhir dengan sukses, `$? = 0`
  - Bila program berakhir dengan error, `$? != 0`
- Nilai dari status exit dapat dilihat melalui instruksi `echo $?`

## Konstruksi if

```
if instruksi-awal
then
    instruksi1
    instruksi2
    .....
fi
```

## Konstruksi if - then - else

```
if instruksi1
then
    instruksi1.1
    instruksi1.2
    .....
else
    instruksi2.1
    instruksi2.2
    .....
fi
```

## Instruksi test

- Instruksi test digunakan untuk memeriksa kondisi dari sebuah ekspresi.
- Ekspresi terdiri dari factor dan operator yang dipisahkan oleh spasi.
- Hasil test akan memberikan nilai berupa status exit, yaitu 0 bila ekspresi sesuai, bila tidak maka hasil adalah  $\neq 0$ .

## Operator pada test

Operator	0 atau TRUE, jika
<code>string1 = string2</code>	Identical
<code>string1 != string2</code>	Not identical
<code>-n string</code>	String is not null
<code>-z string</code>	String is null

## test untuk File dan Direktori

Operator	0 atau TRUE, jika
<code>-f namafile</code>	File ada, file biasa
<code>-d namafile</code>	File ada, file adalah direktori
<code>-r namafile</code>	File dapat dibaca
<code>-w namafile</code>	File dapat ditulis
<code>-x namafile</code>	File adalah executable
<code>-s namafile</code>	File ada dan tidak kosong
<code>-w namafile</code>	File dapat ditulis

## Penulisan test

- Untuk memudahkan pembacaan (readability), test dapat ditulis dengan  
[ ekspresi ]
- [ sebenarnya adalah nama lain dari test, bedanya [ akan mencari kurung penutup ] pada akhir ekspresi yang harus dipisahkan oleh spasi.

## Logical && dan ||

- Notasi && dan || digunakan untuk menggabungkan instruksi shell sebagai alternatif untuk if then else.
- instruksi1 && instruksi2
  - shell akan mengeksekusi instruksi1, dan bila exit status instruksi1 adalah FALSE, maka hasil dari AND tersebut sudah pasti sama dengan FALSE, sehingga instruksi2 tidak mempunyai pengaruh lagi.
- instruksi1 || instruksi2
  - shell akan mengeksekusi instruksi1, bila exit status adalah TRUE(0), hasil dari operasi OR tersebut sudah pasti menghasilkan TRUE, terlepas dari hasil eksekusi instruksi2.



## Operator Bilangan Bulat untuk test

Operator	0 atau TRUE, jika
<code>i1 -eq i2</code>	Bilangan sama
<code>i1 -ge i2</code>	Lebih besar atau sama dengan
<code>i1 -gt i2</code>	Lebih besar
<code>i1 -le i2</code>	Lebih kecil atau sama dengan
<code>i1 -lt i2</code>	Lebih kecil
<code>i1 -ne i2</code>	Bilangan tidak sama

## Operator Logika

- NOT : symbol !
- AND : symbol -a
- OR : symbol -o

## Konstruksi if - then - else - if

```
if instruksi1
then
    instruksi1.1
    instruksi1.2
    .....
elif instruksi2
then
    instruksi2.1
    instruksi2.2
    .....
else
    instruksi3.1
    instruksi3.2
    .....
fi
```

## Hitungan Aritmatika

- Tipe dari variable SHELL hanya satu yaitu STRING.
- Tidak ada tipe lain seperti Numerik, Floating, Boolean atau lainnya.
- Akibatnya variable ini tidak dapat membuat perhitungan aritmetika, misalnya :  
A=5  
B=\$A +1      ## error
- UNIX menyediakan utilitas yang bernama **expr** yaitu suatu utilitas yang melakukan aritmetika sederhana.

## Instruksi exit

- Program dapat dihentikan (terminated/selesai) dengan instruksi exit.
- Sebagai nilai default program tersebut akan memberikan status exit 0.

## Konstruksi case

```
case variable in
match1)
    instruksi1.1
    instruksi1.2
    .....
    ;;
match2)
    instruksi2.1
    instruksi2.2
    .....
    ;;
*)
    instruksi3.1
    instruksi3.2
    .....
    ;;
esac
```

## Konstruksi for

```
for var in str1 str2 ....strn
do
    instruksi1
    instruksi2
    .....
done
```

## Konstruksi while

```
while kondisi
do
    instruksi1
    instruksi2
    .....
done
```