



## Apa itu Pemrograman Shell?

Pemrograman Shell atau disebut Shell Scripting ialah menyusun beberapa perintah shell (internal maupun eksternal) menjadi serangkaian perintah untuk melakukan tugas tertentu.

- Shell sendiri mempunyai beberapa macam 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.

## Variabel

Beberapa hal yang perlu diperhatikan dalam mendefinisikan variabel:

- Angka (0-9)
- Huruf (a-z dan A-Z)
- Karakter underscore (\_)
- Nama variabel tidak boleh dimulai dengan angka

- Tidak boleh menggunakan karakter spesial seperti !, \*, \$, #, -, dll karena karakter tersebut punya makna khusus untuk shell
- Bersifat case sensitive (membedakan huruf besar dan kecil)
- Pemberian nilai variable tidak boleh dipisahkan dengan spasi, karena shell akan menganggap pemisahan tersebut sebagai parameter

## Syntax Variabel

- Mendefinisikan variabel

```
nama_var=nilai
```

- Mengakses variabel

```
$nama_var
```

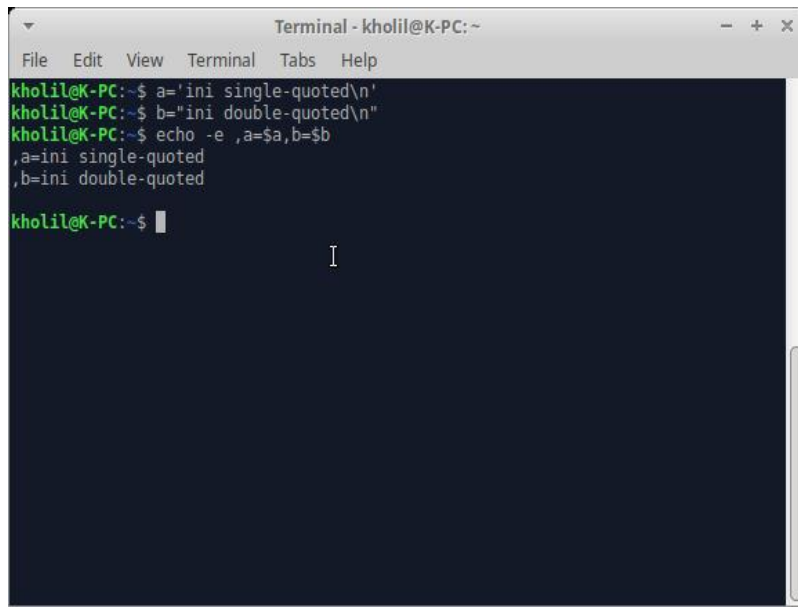
## Operator Aritmatika

### No Operator Deskripsi

1	+	Penjumlahan
2	-	Pengurangan
3	*	Perkalian
4	/	Pembagian
5	%	Modulus (sisanya pembagian)
6	=	Menempatkan nilai di sisi kanan ke variabel di sisi kiri
7	==	Membandingkan 2 nilai yang sama
8	!=	Membandingkan 2 nilai yang tidak sama

## Tipe Data

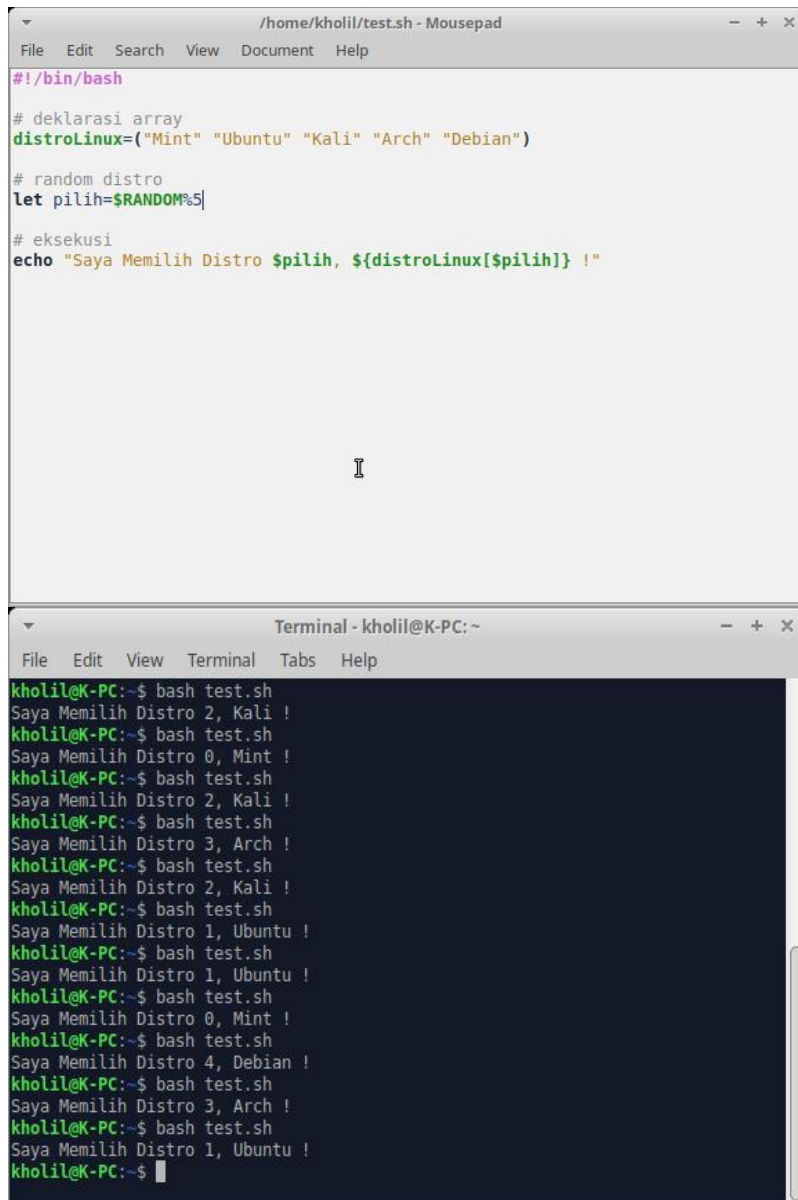
- Beberapa tipe data di pemrograman shell antara lain :
  - String
  - Array
  - Integer
  - dsb.
- Contoh penggunaan String (single-quoted) & (double-quoted) :

A terminal window titled "Terminal - kholil@K-PC: ~" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal shows three commands and their outputs:

```
kholil@K-PC:~$ a='ini single-quoted\n'  
kholil@K-PC:~$ b="ini double-quoted\n"  
kholil@K-PC:~$ echo -e ,a=$a,b=$b  
,a=ini single-quoted  
,b=ini double-quoted  
kholil@K-PC:~$
```

The output of the echo command is displayed on two lines: ",a=ini single-quoted" and ",b=ini double-quoted". A cursor is visible on the line following the last command.

- Contoh penggunaan Array :



```
#!/bin/bash

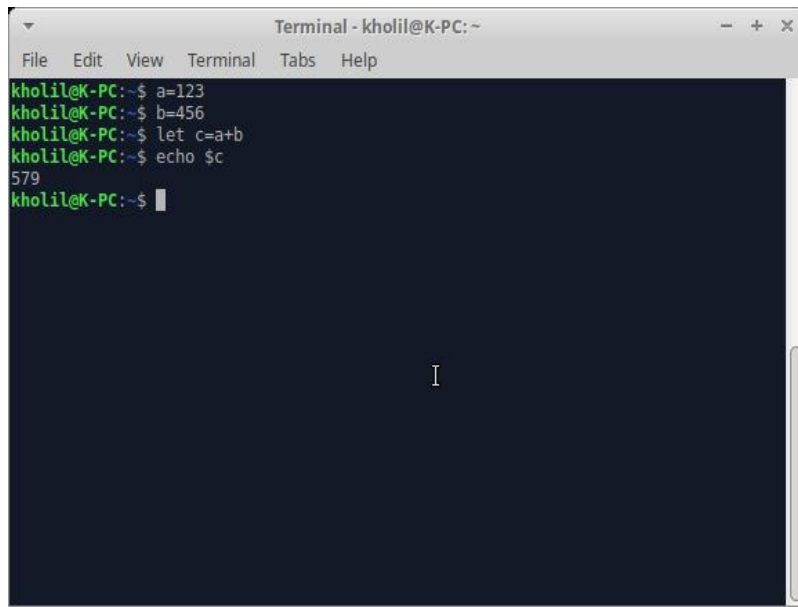
# deklarasi array
distroLinux=("Mint" "Ubuntu" "Kali" "Arch" "Debian")

# random distro
let pilih=$RANDOM%5

# eksekusi
echo "Saya Memilih Distro $pilih, ${distroLinux[$pilih]} !"

kholil@K-PC:~$ bash test.sh
Saya Memilih Distro 2, Kali !
kholil@K-PC:~$ bash test.sh
Saya Memilih Distro 0, Mint !
kholil@K-PC:~$ bash test.sh
Saya Memilih Distro 2, Kali !
kholil@K-PC:~$ bash test.sh
Saya Memilih Distro 3, Arch !
kholil@K-PC:~$ bash test.sh
Saya Memilih Distro 2, Kali !
kholil@K-PC:~$ bash test.sh
Saya Memilih Distro 1, Ubuntu !
kholil@K-PC:~$ bash test.sh
Saya Memilih Distro 1, Ubuntu !
kholil@K-PC:~$ bash test.sh
Saya Memilih Distro 0, Mint !
kholil@K-PC:~$ bash test.sh
Saya Memilih Distro 4, Debian !
kholil@K-PC:~$ bash test.sh
Saya Memilih Distro 3, Arch !
kholil@K-PC:~$ bash test.sh
Saya Memilih Distro 1, Ubuntu !
kholil@K-PC:~$
```

- Contoh penggunaan Integer :

A terminal window titled "Terminal - kholil@K-PC: ~" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal shows a series of commands and their outputs: 

```
kholil@K-PC:~$ a=123
kholil@K-PC:~$ b=456
kholil@K-PC:~$ let c=a+b
kholil@K-PC:~$ echo $c
579
kholil@K-PC:~$
```

Catatan : let = digunakan untuk melakukan operasi perhitungan aritmatika

- Ada 3 cara yang dapat digunakan untuk melakukan operasi matematika, yaitu:
  - Menggunakan perintah built-in `let`
  - Menggunakan perintah eksternal `expr` atau `awk`
  - Menggunakan perintah substitusi `${(ekspresi)}`
- Contoh operasi "Let, Expr, Ekspresi" :

```
#!/bin/bash

a=15
b=7

#memakai let
let jumlah=$a+$b
let kurang=$a-$b
let kali=$a*$b

#memakai expr
bagi=`expr $a / $b`

#memakai perintah substitusi ${(ekspresi)}
mod=$(( $a % $b ))

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

b=$a

echo "a = $a"
echo "b = $b"
```

- Output :

```
a + b = 22
a - b = 8
a * b = 105
a / b = 2
a % b = 1
a = 15
b = 15
```

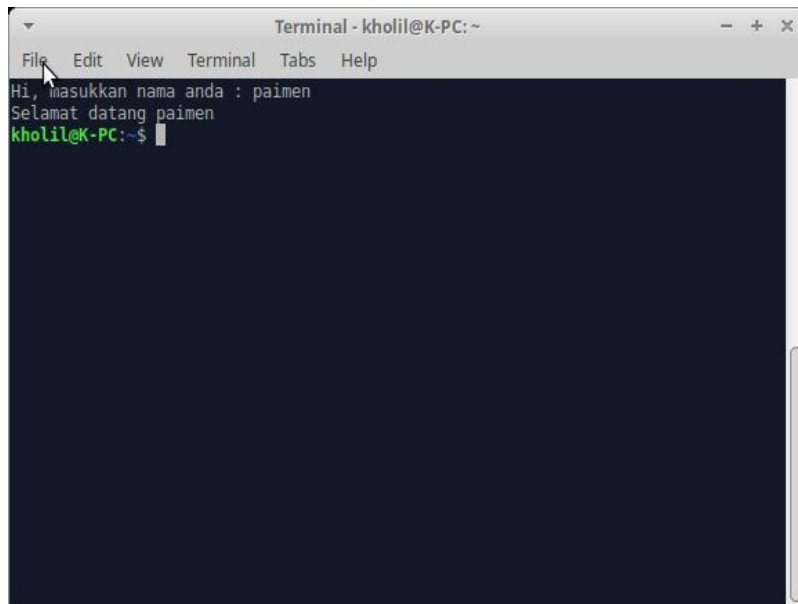
## Input

### Input pada shell

- Untuk membaca inputan dari user gunakanlah "read" dengan format penulisan "read nama\_var"

```
#!/bin/bash
clear
```

```
echo -n "Hallo, masukkan nama anda : ";
read nama;
echo "Selamat datang $nama";
```



## Ouput

### Output pada shell

- echo digunakan untuk menampilkan output dengan syntax sebagai berikut:
  - Menampilkan teks biasa echo "teks"

- o Menampilkan isi dari sebuah variabel `echo $nama_var`

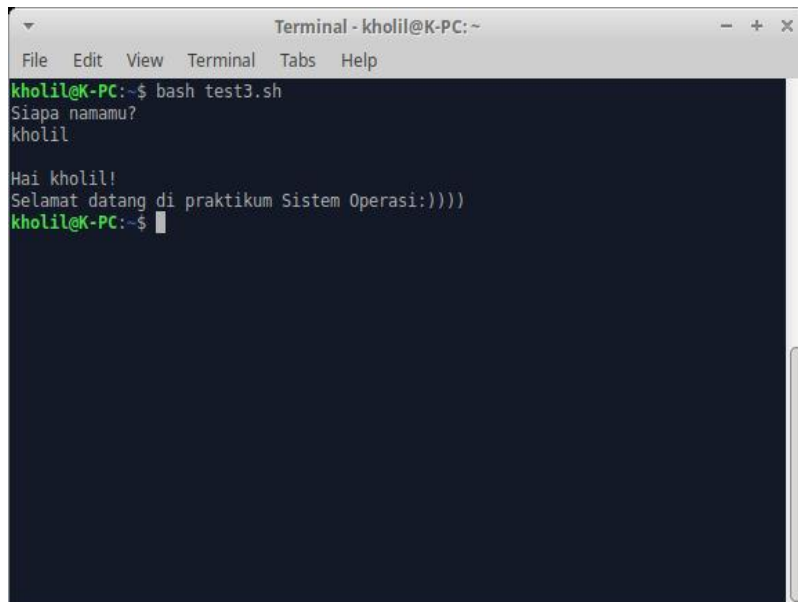
#### Catatan:

Jika ingin menggunakan new line character (`\n`) pada `echo`, gunakan `echo -e "teks\n teks"`

- Contoh

```
#!/bin/bash
matakuliah="Sistem Operasi"

echo "Siapa namamu?"
read nama
echo -e "\nHai $nama!\nSelamat datang di praktikum $matakuliah:))))"
```



```
Terminal - kholil@K-PC: ~
File Edit View Terminal Tabs Help
kholil@K-PC:~$ bash test3.sh
Siapa namamu?
kholil

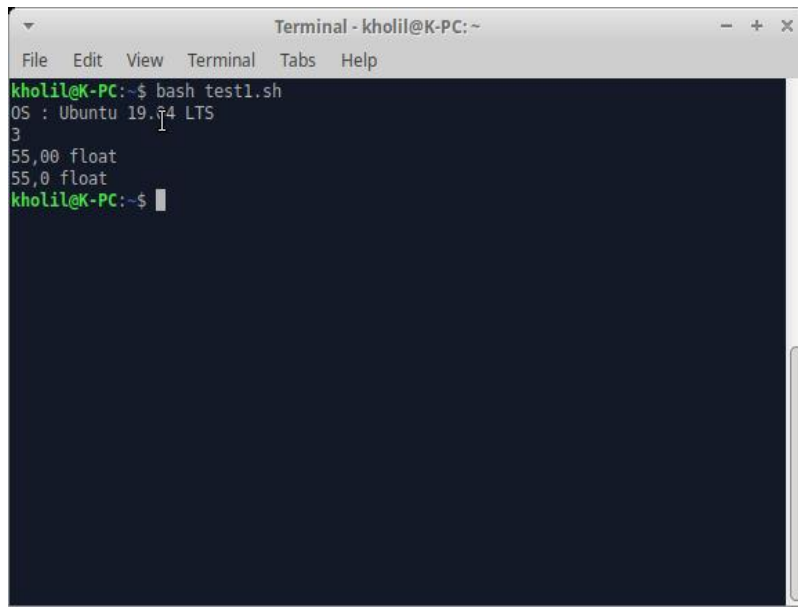
Hai kholil!
Selamat datang di praktikum Sistem Operasi:))))
kholil@K-PC:~$
```

- Selain menggunakan output "echo", output bisa juga dilakukan dengan menggunakan `printf` seperti pada bahasa pemrograman C.
- Contoh

```
#!/bin/bash

# Inisialisasi Var
a=55;
b=4;
distroLinux="Ubuntu 19.04 LTS";
let c=a%b;

# Output Printf
printf "OS : $distroLinux \n";
printf "$c \n";
printf "%.2f float \n" $a;
printf "%.1f float \n" $a;
```



```
Terminal - kholil@K-PC: ~
File Edit View Terminal Tabs Help
kholil@K-PC:~$ bash test1.sh
OS : Ubuntu 19.04 LTS
3
55,00 float
55,0 float
kholil@K-PC:~$
```

## Percabangan

Sebelum melangkah ke percabangan alangkah baiknya terlebih dahulu mengetahui dasar operasi relasional yang mana biasanya digunakan bersama dengan conditional statements.

### No Operator Deskripsi

- 1    -eq    Memeriksa apakah nilai kedua operan sama (==)
- 2    -ne    Memeriksa apakah nilai kedua operan tidak sama (!=)
- 3    -gt    Memeriksa apakah nilai operan kiri lebih besar daripada operan kanan (>)
- 4    -lt    Memeriksa apakah nilai operan kiri lebih kecil daripada operan kanan (<)
- 5    -ge    Memeriksa apakah nilai operan kiri lebih besar atau sama dengan operan kanan (>=)
- 6    -le    Memeriksa apakah nilai operan kiri lebih kecil atau sama dengan operan kanan (<=)

- If ..... Else
  - Syntax

```
if [ kondisi1 ]
then
    perintah1
elif [ kondisi2 ]
then
    perintah2
else
    alternatif_perintah
fi
```

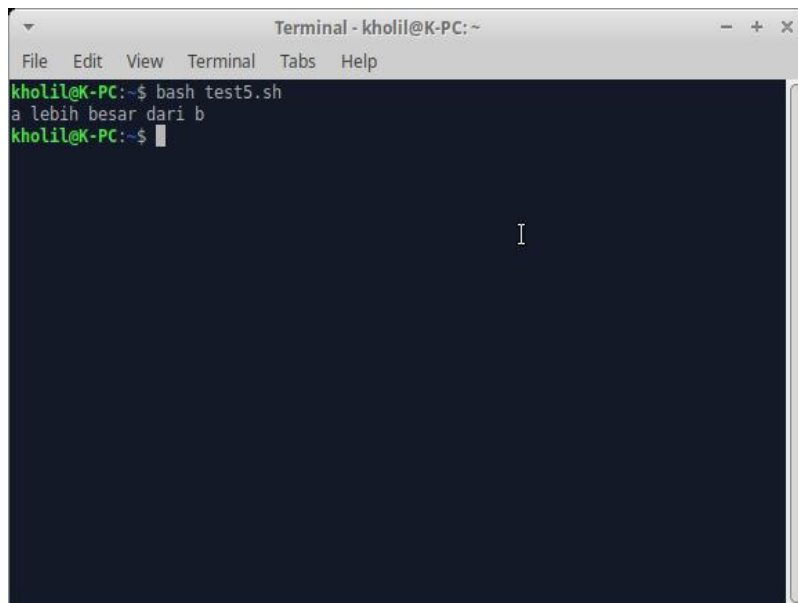
- Contoh:



```
#!/bin/bash

a=15
b=7

if [ $a == $b ]
then
    echo "a sama dengan b"
elif [ $a -gt $b ]
then
    echo "a lebih besar dari b"
elif [ $a -lt $b ]
then
    echo "a lebih kecil dari b"
else
    echo "Tidak ada kondisi yang memenuhi"
fi
```

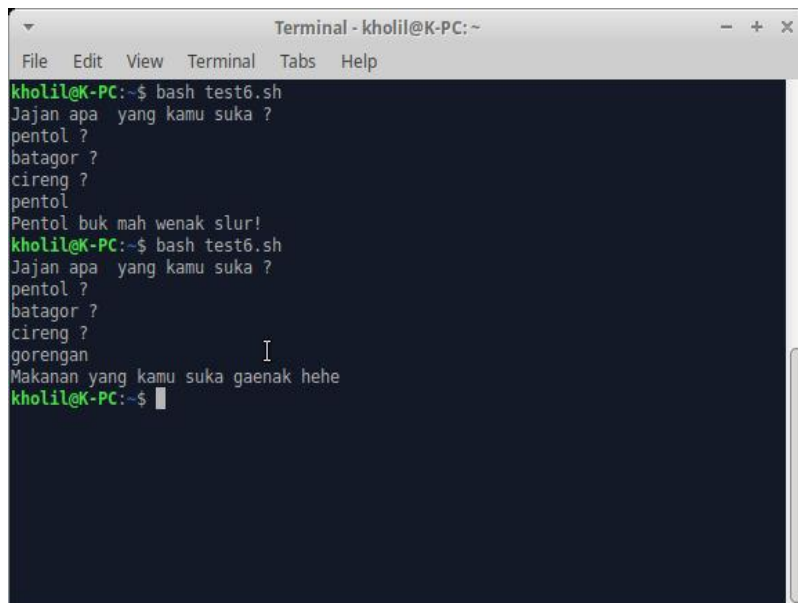


The screenshot shows a terminal window titled "Terminal - kholil@K-PC: ~". The window has a menu bar with "File", "Edit", "View", "Terminal", "Tabs", and "Help". The terminal content shows the user running the command "bash test5.sh", which outputs "a lebih besar dari b". The prompt "kholil@K-PC:~\$" is visible at the end of the line.

```
Terminal - kholil@K-PC: ~
File Edit View Terminal Tabs Help
kholil@K-PC:~$ bash test5.sh
a lebih besar dari b
kholil@K-PC:~$
```

- Case .... Esac
  - Syntax
- pola1)
- perintah1
- ;;
- pola2)
- perintah2
- ;;
- \*)
- alternatif\_perintah
- ;;
- esac
- Contoh:
- #!/bin/bash
-

- `printf "Jajan apa yang kamu suka ?\n"`
- `printf "pentol ?\n"`
- `printf "batagor ?\n"`
- `printf "cireng ?\n"`
- 
- `read jajan`
- 
- `case "$jajan" in`
- `"pentol")`
- `echo "Pentol buk mah wenak slur!"`
- `;;`
- `"batagor")`
- `echo "Batagore mas budi mantap bat"`
- `;;`
- `"cireng")`
- `echo "Cireng kantin rasane unch-unch"`
- `;;`
- `*)`
- `echo "Makanan yang kamu suka gaenak hehe"`
- `;;`
- `esac`



```

Terminal - kholil@K-PC: ~
File Edit View Terminal Tabs Help
kholil@K-PC:~$ bash test6.sh
Jajan apa yang kamu suka ?
pentol ?
batagor ?
cireng ?
pentol
Pentol buk mah wenak slur!
kholil@K-PC:~$ bash test6.sh
Jajan apa yang kamu suka ?
pentol ?
batagor ?
cireng ?
gorengan
Makanan yang kamu suka gaenak hehe
kholil@K-PC:~$

```

## Soal Latihan

- Buatlah program percabangan sederhana aritmatika yang menerapkan beberapa konsep pemrograman bash seperti diatas !