

Nama : Kinanthi Putri Ariyani
NPM : 21083010047
Mata Kuliah : Sistem Operasi
Kelas : B

Pemrograman Shell ??

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

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 special seperti !, *, \$, #, -, dll karena karakter tersebut punya makna khusus untuk shell
- Bersifat case sensitive (membedakan huruf besar dan kecil)
- Pemberian nilai variabel 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

1. + Penjumlahan
2. - Pengurangan
3. * Perkalian
4. / Pembagian
5. % Modulus (sisa 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

A. Penggunaan String (single-quoted) & (double-quoted)

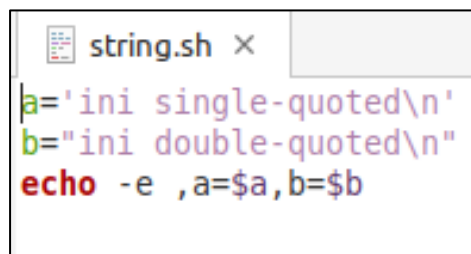
1. Non-nano version (terminal)

```
File Edit View Search Terminal Help
kinanthi@Kinanthi-Linux:~$ a='ini single-quoted\n'
kinanthi@Kinanthi-Linux:~$ b='ini double-quoted\n'
kinanthi@Kinanthi-Linux:~$ echo -e ,a=$a,b=$b
,a=ini single-quoted
,b=ini double-quoted
kinanthi@Kinanthi-Linux:~$
```

2. Nano version

```
kinanthi@Kinanthi-Linux:~$ nano string.sh
```

- Membuat file menggunakan perintah nano string.sh



```
string.sh x
a='ini single-quoted\n'
b="ini double-quoted\n"
echo -e ,a=$a,b=$b
```

- Membuat isi di dalam file string.sh, lalu tekan Ctrl + S untuk menyimpan dan Ctrl + X untuk mengembalikan tampilan terminal

```
kinanthi@Kinanthi-Linux:~$ bash string.sh
,a=ini single-quoted
,b=ini double-quoted
```

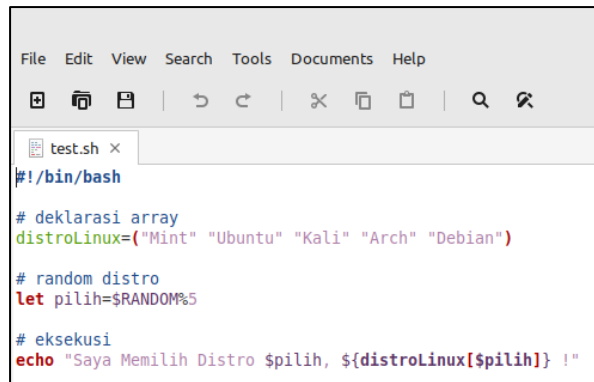
- Memanggil isi file menggunakan perintah bash string.sh

B. Penggunaan Array

1. Nano version

```
File Edit View Search Terminal Tabs Help
kinanthi@Kinanthi-Linux:~$ nano test.sh
```

- Membuat file menggunakan perintah nano test.sh



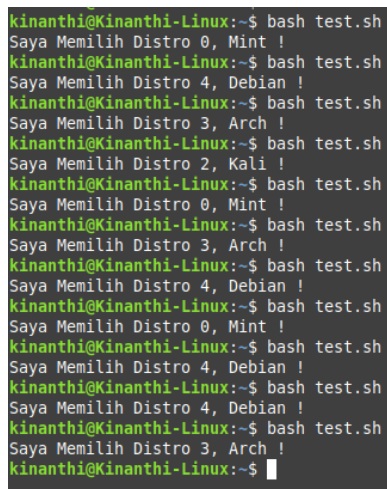
```
File Edit View Search Tools Documents Help
test.sh x
#!/bin/bash

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

# random distro
let pilih=$RANDOM%5

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

- Membuat isi di dalam file test.sh, lalu tekan Ctrl + S untuk menyimpan dan Ctrl + X untuk mengembalikan tampilan terminal



```
kinanthi@Kinanthi-Linux:~$ bash test.sh
Saya Memilih Distro 0, Mint !
kinanthi@Kinanthi-Linux:~$ bash test.sh
Saya Memilih Distro 4, Debian !
kinanthi@Kinanthi-Linux:~$ bash test.sh
Saya Memilih Distro 3, Arch !
kinanthi@Kinanthi-Linux:~$ bash test.sh
Saya Memilih Distro 2, Kali !
kinanthi@Kinanthi-Linux:~$ bash test.sh
Saya Memilih Distro 0, Mint !
kinanthi@Kinanthi-Linux:~$ bash test.sh
Saya Memilih Distro 3, Arch !
kinanthi@Kinanthi-Linux:~$ bash test.sh
Saya Memilih Distro 4, Debian !
kinanthi@Kinanthi-Linux:~$ bash test.sh
Saya Memilih Distro 0, Mint !
kinanthi@Kinanthi-Linux:~$ bash test.sh
Saya Memilih Distro 4, Debian !
kinanthi@Kinanthi-Linux:~$ bash test.sh
Saya Memilih Distro 4, Debian !
kinanthi@Kinanthi-Linux:~$ bash test.sh
Saya Memilih Distro 3, Arch !
kinanthi@Kinanthi-Linux:~$
```

- Memanggil isi file menggunakan perintah bash test.sh dan melakukannya sebanyak 11x dan bisa dilihat bahwa output dari tiap-tiap perintah bash akan memunculkan output yang berbeda

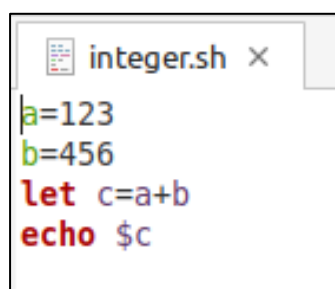
C. Penggunaan Integer

1. Nano version



```
kinanthi@Kinanthi-Linux:~$ nano integer.sh
```

- Membuat file menggunakan perintah nano integer.sh



```
integer.sh x
a=123
b=456
let c=a+b
echo $c
```

- Membuat isi di dalam file integer.sh, lalu tekan Ctrl + S untuk menyimpan dan Ctrl + X untuk mengembalikan tampilan terminal

```
kinanthi@Kinanthi-Linux:~$ bash integer.sh
579
kinanthi@Kinanthi-Linux:~$
```

- Memanggil isi file menggunakan perintah bash integer.sh

2. Non-nano version

```
kinanthi@Kinanthi-Linux:~$ a=123
kinanthi@Kinanthi-Linux:~$ b=456
kinanthi@Kinanthi-Linux:~$ let c=a+b
kinanthi@Kinanthi-Linux:~$ echo $c
579
kinanthi@Kinanthi-Linux:~$
```

Catatan : let = digunakan untuk melakukan operasi perhitungan aritmatika

- Ada 3 cara yang dapat digunakan untuk melakukan operasi aritmatika, yaitu :
 1. Menggunakan perintah built-in let
 2. Menggunakan perintah eksternal expr atau awk
 3. Menggunakan perintah substitusi \$((ekspresi))

D. Penggunaan Operasi Aritmatika

1. Nano version

```
kinanthi@Kinanthi-Linux:~$ nano operasiaritmatika.sh
```

- Membuat file menggunakan perintah nano operasiaritmatika.sh

```
File Edit View Search Terminal Tabs Help
kinanthi@Kinanthi-Linux: ~
GNU nano 6.2
#!/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"
```

- Membuat isi di dalam file operasiaritmatika.sh, lalu tekan Ctrl + S untuk menyimpan dan Ctrl + X untuk mengembalikan tampilan terminal

```
kinanthi@Kinanthi-Linux:~$ bash operasiaritmatika.sh
a + b = 22
a - b = 8
a * b = 105
a / b = 2
a % b = 1
a = 15
b = 15
kinanthi@Kinanthi-Linux:~$
```

- Memanggil isi file menggunakan perintah bash operasiaritmatika.sh

E. Penggunaan Input

1. Nano version

```
kinanthi@Kinanthi-Linux:~$ nano input.sh
```

- Membuat file menggunakan perintah nano input.sh

```
kinanthi@Kinanthi-Linux: ~
GNU nano 6.2
#!/bin/bash
clear

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

- Membuat isi di dalam file input.sh, lalu tekan Ctrl + S untuk menyimpan dan Ctrl + X untuk mengembalikan tampilan terminal
- Untuk membaca inputan dari user gunakanlah “read” dengan format penulisan “read nama_var”

```
kinanthi@Kinanthi-Linux:~$ bash input.sh
File Edit View Search Terminal Tabs Help
kinanthi@Kinanthi-Linux: ~
Hallo, masukkan nama anda : Kinanthi Putri Ariyani
Selamat datang Kinanthi Putri Ariyani
kinanthi@Kinanthi-Linux:~$
```

- Memanggil isi file menggunakan perintah bash input.sh dan menampilkan output nama yang wajib kita isi lalu setelah di enter akan muncul kalimat “Selamat datang ...”

F. Penggunaan Output

Output 1

1. Nano version

```
kinanthi@Kinanthi-Linux: ~
kinanthi@Kinanthi-Linux:~$ nano output1.sh
```

- Membuat file menggunakan perintah nano output1.sh

```

output1.sh x
#!/bin/bash
matakuliah="Sistem Operasi"

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

```

- Membuat isi di dalam file output1.sh, lalu tekan Ctrl + S untuk menyimpan dan Ctrl + X untuk mengembalikan tampilan terminal
- Echo digunakan untuk menampilkan output dengan syntax sebagai berikut
 1. Menampilkan teks biasa echo "teks"
 2. Menampilkan isi dari sebuah variabel echo \$nama_var

```

kinanthi@Kinanthi-Linux:~$ bash output1.sh
Siapa namamu?
Kinanthi Putri Ariyani

Hai Kinanthi Putri Ariyani!
Selamat datang di praktikum Sistem Operasi:)))))
kinanthi@Kinanthi-Linux:~$

```

- Memanggil isi file menggunakan perintah bash output1.sh dan menampilkan output nama yang wajib kita isi lalu setelah di enter akan muncul kalimat "Hai ..."

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

Output 2

1. Nano version

```

kinanthi@Kinanthi-Linux:~$ nano output.sh

```

- Membuat file menggunakan perintah nano output.sh

```

kinanthi@Kinanthi-Linux: ~
GNU nano 6.2
#!/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;

```

- Membuat isi di dalam file output.sh, lalu tekan Ctrl + S untuk menyimpan dan Ctrl + X untuk mengembalikan tampilan terminal

```
kinanthi@Kinanthi-Linux:~$ bash output.sh
OS : Ubuntu 19.04 LTS
3
55,00 float
55,0 float
kinanthi@Kinanthi-Linux:~$
```

- Memanggil isi file menggunakan perintah bash output.sh

G. Penggunaan Percabangan

Operator Percabangan

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 nilai operan kiri lebih kecil atau sama dengan operan kanan (<=)

Percabangan 1

1. Nano version

```
kinanthi@Kinanthi-Linux:~$ nano percabangan1.sh
```

- Membuat file menggunakan perintah nano percabangan1.sh

```
kinanthi@Kinanthi-Linux: ~
GNU nano 6.2
#!/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
```

- Membuat isi di dalam file percabangan1.sh, lalu tekan Ctrl + S untuk menyimpan dan Ctrl + X untuk mengembalikan tampilan terminal

```
kinanthi@Kinanthi-Linux:~$ bash percabangan1.sh
a lebih besar dari b
kinanthi@Kinanthi-Linux:~$
```

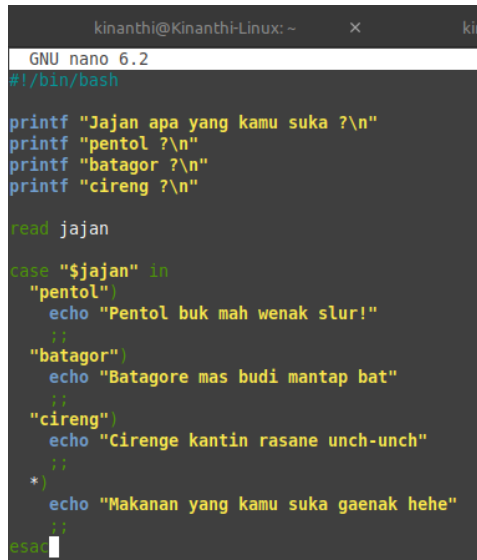
- Memanggil isi file menggunakan perintah bash percabangan1.sh

Percabangan 2

1. Nano version

```
kinanthi@Kinanthi-Linux:~$ nano percabangan2.sh
```

- Membuat file menggunakan perintah nano percabangan2.sh



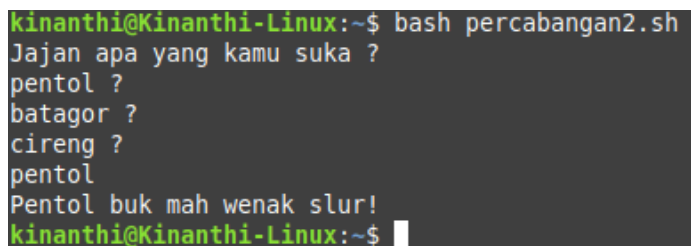
```
kinanthi@Kinanthi-Linux: ~
GNU nano 6.2
#!/bin/bash

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
```

- Membuat isi di dalam file percabangan2.sh, lalu tekan Ctrl + S untuk menyimpan dan Ctrl + X untuk mengembalikan tampilan terminal



```
kinanthi@Kinanthi-Linux:~$ bash percabangan2.sh
Jajan apa yang kamu suka ?
pentol ?
batagor ?
cireng ?
pentol
Pentol buk mah wenak slur!
kinanthi@Kinanthi-Linux:~$
```

- Memanggil isi file menggunakan perintah bash percabangan2.sh

TUGAS 2

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

Langkah – langkah :

1. Membuat file dengan nama nano soallatihan2.sh dengan menggunakan perintah **nano** **namafile.sh**
2. Membuat isi dari file soallatihan2.sh seperti pada gambar dibawah ini

```
File Edit View Search Tools Documents Help
soallatihan2.sh x
echo "-----SOAL LATIHAN ARITMATIKA-----"

echo "nilai a : "
read a
echo "nilai b : "
read b

echo "Berapakah hasil dari a+b ?"

let penjumlahan=a+b
echo "a + b = $penjumlahan"

# menggunakan percabangan
sleep 1.5
echo "-----Lanjut Yuk!!-----"

echo "Operasi Aritmatika"
echo "1. Penjumlahan"
echo "2. Pengurangan"
echo "3. Perkalian"
echo "4. Pembagian"
echo "Pilih 1-4"
read pilih

case "$pilih" in
    "1")
        echo "=== Operasi Penjumlahan ==="
        echo "Masukkan nilai a : "
        read a
        echo "Masukkan nilai b : "
        read b
        let hasil=$a+$b
        echo "a + b = $hasil"
        ;;
    "2")
        echo "=== Operasi Pengurangan ==="
        echo "Masukkan nilai a : "
        read a
        echo "Masukkan nilai b : "
        read b
        let hasil=$a-$b
        echo "a - b = $hasil"
        ;;
    "3")
        echo "=== Operasi Perkalian ==="
        echo "Masukkan nilai a : "
        read a
        echo "Masukkan nilai b : "
        read b
        let hasil=$a*$b
        echo "a * b = $hasil"
        ;;
    "4")
        echo "=== Operasi Pembagian ==="
        echo "Masukkan nilai a : "
        read a
        echo "Masukkan nilai b : "
        read b
        let hasil=$a/$b
        echo "a / b = $hasil"
        ;;
esac
```

3. Dapat dilihat pada gambar di atas pertama saya mencoba menggunakan non percabangan dan selanjutnya diikuti dengan menggunakan percabangan
4. Pada script di atas terdapat perintah **sleep 1.5** yang artinya script yang ada dibawah tulisan **sleep 1.5** akan muncul setelah 1,5detik
5. Setelah membuat script di atas, selanjutnya tekan tombol Ctrl + S dan Ctrl + X
6. Selanjutnya, masukkan perintah **bash soallatihan.sh** untuk menampilkan isi dari script yang sudah kita buat. Dibawah ini adalah tampilan dari output tersebut

```
kinanthi@Kinanthi-Linux:~$ nano soallatihan2.sh
kinanthi@Kinanthi-Linux:~$ bash soallatihan2.sh
-----SOAL LATIHAN ARITMATIKA-----
nilai a :
4
nilai b :
4
Berapakah hasil dari a+b ?
a + b = 8
-----Lanjut Yuk!!-----
Operasi Aritmatika
1. Penjumlahan
2. Pengurangan
3. Perkalian
4. Pembagian
Pilih 1-4
4
=== Operasi Pembagian ===
Masukkan nilai a :
4
Masukkan nilai b :
4
a / b = 1
```

7. Dapat dilihat bahwa jika kita memasukkan angka-angka ke dalam tempat yang sudah disediakan, sistem akan otomatis menghitung hasil dari operasi yang kita pilih.