

A. Penggunaan Array

```
calviendanny@calviendanny-VirtualBox:~$ bash array.sh
saya memilih distro 4, !
calviendanny@calviendanny-VirtualBox:~$
```

```
#!/bin/bash

#deklarasi array
distroLinux=("mint" "ubuntu" "kali" "arch" "debian")

#random distro
let pilih=$RANDOM%5

#eksekusi
echo "saya memilih distro $pilih, ${distroLinux[pilih]} !"
```

Arrays adalah jenis struktur data yang digunakan untuk menyimpan nilai dari jenis tertentu. Anda juga dapat menganggap array sebagai variabel tetapi variabel hanya dapat menyimpan satu nilai di mana array dapat menyimpan beberapa nilai di dalamnya. Konsep array tidak hanya terikat pada bash. Bahasa pemrograman apa pun yang Anda gunakan akan memiliki array di dalamnya tetapi dengan perbedaan implementasi.

B. Penggunaan Interger (bilangan bulat)

```
calviendanny@calviendanny-VirtualBox:~$ nano integer.sh
calviendanny@calviendanny-VirtualBox:~$ a=123
calviendanny@calviendanny-VirtualBox:~$ b=456
calviendanny@calviendanny-VirtualBox:~$ let c=a+b
calviendanny@calviendanny-VirtualBox:~$ echo $c
579
calviendanny@calviendanny-VirtualBox:~$
```

C. Penggunaan operasi matematika

```
calviendanny@calviendanny-VirtualBox:~$ bash op_mtk.sh
a + b = 22
a - b = 8
a * b = 105
a / b = 2
a % b = 1
a = 15
b = 15
calviendanny@calviendanny-VirtualBox:~$
```

```
#!/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"
```

[ Read 26 lines ]

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute	^C Location
^X Exit	^R Read File	^_ Replace	^U Paste	^J Justify	^_ Go To Line

Catatan : let = digunakan untuk melakukan operasi perhitungan aritmatika

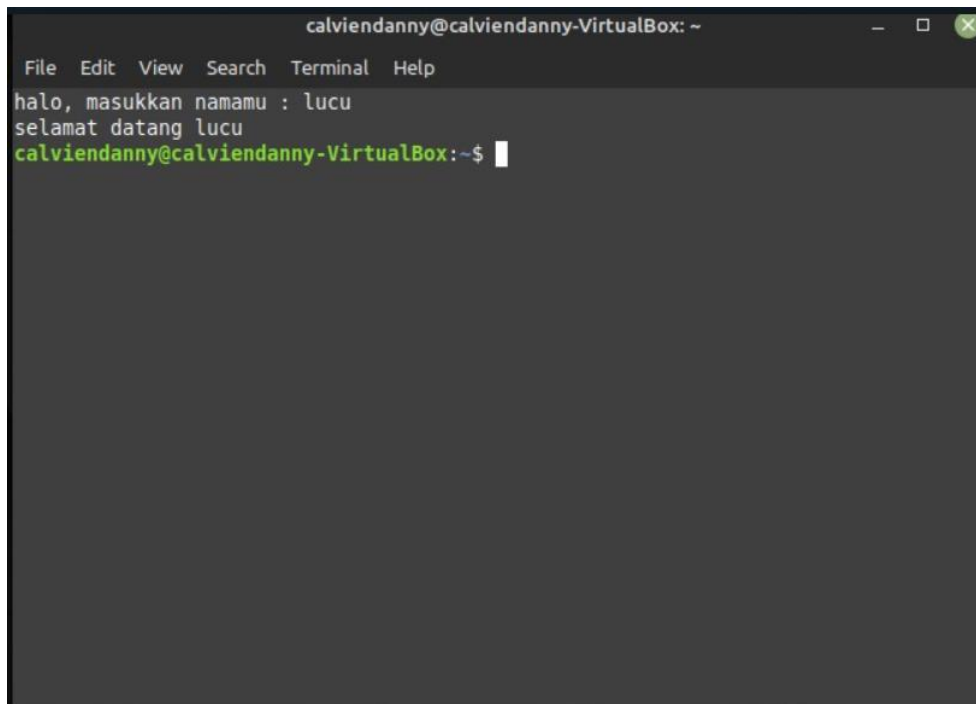
- Ada 3 cara yang dapat digunakan untuk melakukan operasi matematika, yaitu:
  - o Menggunakan perintah built-in let
  - o Menggunakan perintah eksternal expr atau awk
  - o Menggunakan perintah substitusi \$((ekspresi))

#### D. Penggunaan string

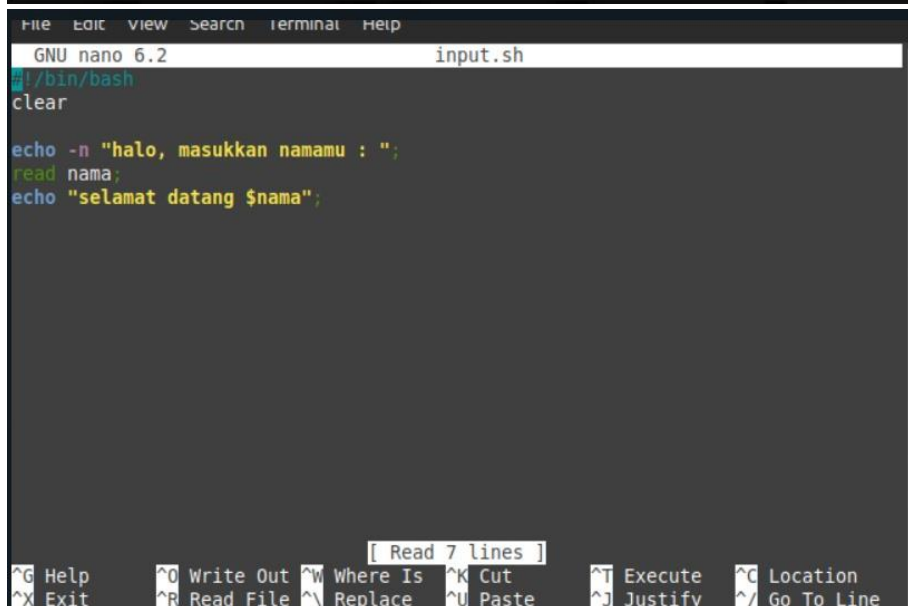
```
calviendanny@calviendanny-VirtualBox:~$ a='ini single-quoted\n'
calviendanny@calviendanny-VirtualBox:~$ b="ini double-quoted\n"
calviendanny@calviendanny-VirtualBox:~$ echo -e , a=$a,b=$b
, a=ini single-quoted
,b=ini double-quoted
```

string yang diapit tanda kutip ganda (double quoted) akan ditampilkan pada layar anda, echo adalah statement (perintah) built-in bash yang berfungsi menampilkan informasi ke standard output yang defaultnya adalah layar. jika diinginkan mengulangi proses tersebut, anda akan mengetikkan kembali perintah tadi, tapi dengan fasilitas history cukup menggunakan tombol panah kita sudah dapat mengulangi perintah tersebut, bagaimana jika berupa kumpulan perintah yang cukup banyak, tentunya dengan fasilitas hirtory kita akan kerepotan juga mengulangi perintah yang diinginkan apalagi jika selang beberapa waktu mungkin perintah-perintah tadi sudah tertimpa oleh perintah lain karena history mempunyai kapasitas penyimpanan yang ditentukan. untuk itulah sebaiknya perintah-perintah tsb disimpan ke sebuah file yang dapat kita panggil kapanpun diinginkan.

## E. Penggunaan Input



```
calviendanny@calviendanny-VirtualBox: ~  
File Edit View Search Terminal Help  
halo, masukkan namamu : lucu  
selamat datang lucu  
calviendanny@calviendanny-VirtualBox:~$
```



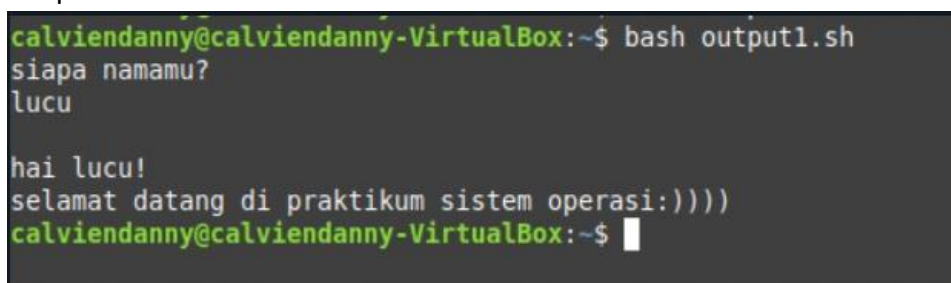
```
GNU nano 6.2 input.sh  
#!/bin/bash  
clear  
  
echo -n "halo, masukkan namamu : ";  
read nama;  
echo "selamat datang $nama";  
  
[ Read 7 lines ]  
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location  
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

Input pada shell

- Untuk membaca input dari user gunakanlah "read" dengan forma penulis ".read nama\_var"

## F. Penggunaan output

Output1



```
calviendanny@calviendanny-VirtualBox:~$ bash output1.sh  
siapa namamu?  
lucu  
  
hai lucu!  
selamat datang di praktikum sistem operasi:))))  
calviendanny@calviendanny-VirtualBox:~$
```

## Output2

```
calviendanny@calviendanny-VirtualBox:~$ bash output2.sh
OS : Ubuntu 19.04 LTS
3
0.00 float
0x0p+0.1f float
calviendanny@calviendanny-VirtualBox:~$
```

Output pada shell

- echo digunakan untuk menampilkan output dengan syntax sebagai berikut:
  - o 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"

## G. Penggunaan Percabangan

### Percabangan1

```
calviendanny@calviendanny-VirtualBox:~$ bash percabangan1.sh
a lebih besar dari b
calviendanny@calviendanny-VirtualBox:~$
```

### Percabangan2

```
calviendanny@calviendanny-VirtualBox: ~
File Edit View Search Terminal Help
calviendanny@calviendanny-VirtualBox:~$ nano percabangan2.sh
calviendanny@calviendanny-VirtualBox:~$ bash percabangan2.sh
jajan apa yang kamu suka ?
pentol ?
batagor ?
cireng ?
babi
babi rasane sampah
calviendanny@calviendanny-VirtualBox:~$
```

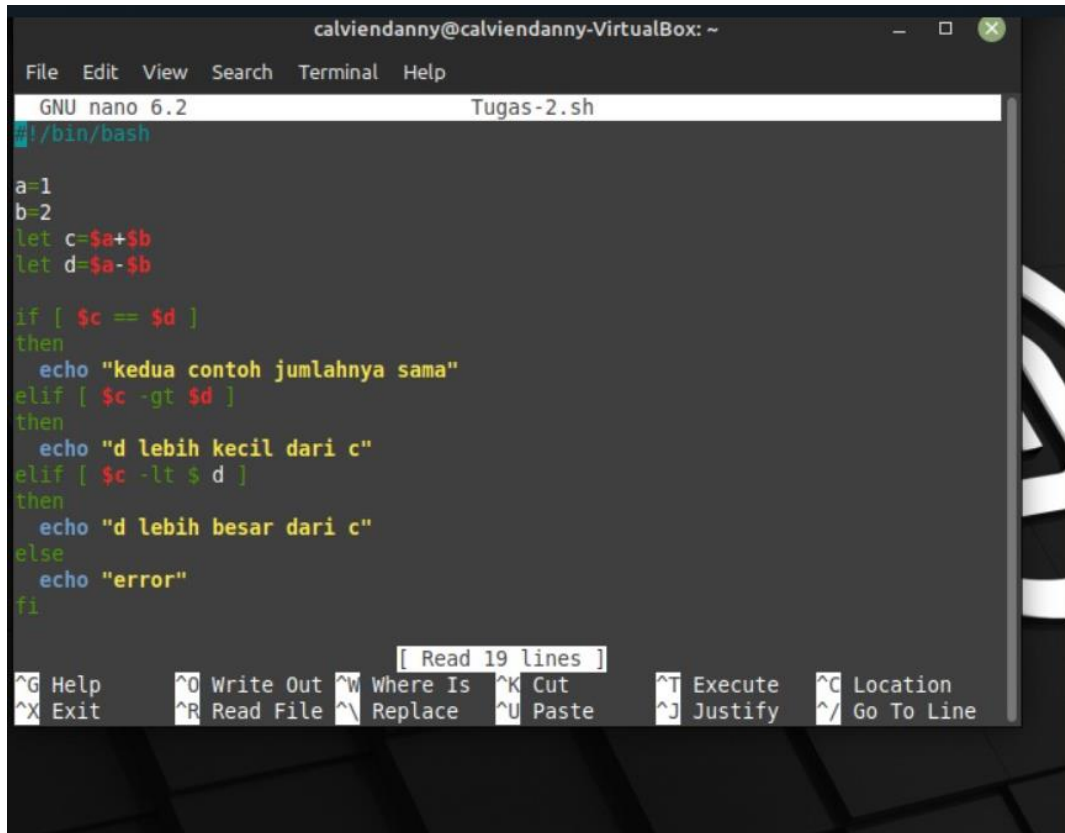
Sebelum melangkah ke percabangan alangkah baiknya terlebih dahulu mengetahui dasar operasi relasional yang mana biasanya digunakan bersama dengan conditional statements.

### No Operator Deskripsi 1

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 ( $\geq$ )
6. -le Memeriksa apakah nilai operan kiri lebih kecil atau sama dengan operan kanan ( $\leq$ )

H. Soal Latihan

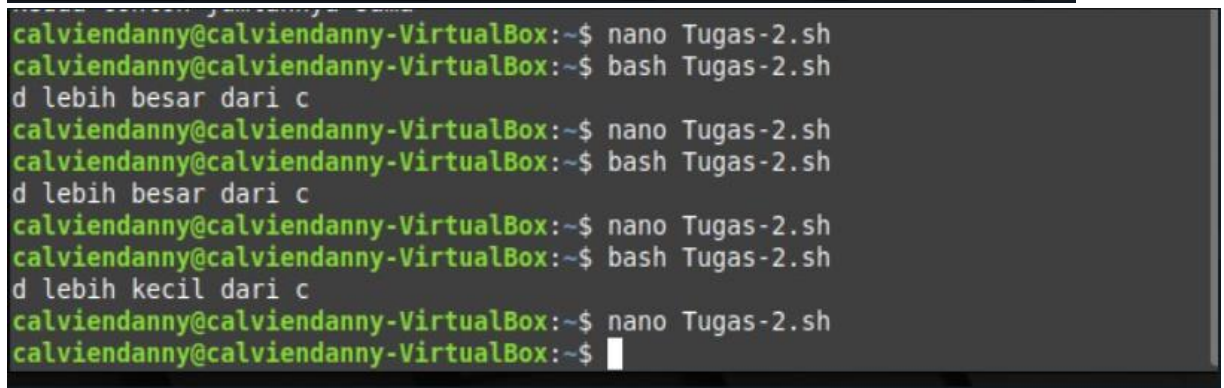


```
calviendanny@calviendanny-VirtualBox: ~
File Edit View Search Terminal Help
GNU nano 6.2 Tugas-2.sh
#!/bin/bash

a=1
b=2
let c=$a+$b
let d=$a-$b

if [ $c == $d ]
then
    echo "kedua contoh jumlahnya sama"
elif [ $c -gt $d ]
then
    echo "d lebih kecil dari c"
elif [ $c -lt $d ]
then
    echo "d lebih besar dari c"
else
    echo "error"
fi

[ Read 19 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```



```
calviendanny@calviendanny-VirtualBox:~$ nano Tugas-2.sh
calviendanny@calviendanny-VirtualBox:~$ bash Tugas-2.sh
d lebih besar dari c
calviendanny@calviendanny-VirtualBox:~$ nano Tugas-2.sh
calviendanny@calviendanny-VirtualBox:~$ bash Tugas-2.sh
d lebih besar dari c
calviendanny@calviendanny-VirtualBox:~$ nano Tugas-2.sh
calviendanny@calviendanny-VirtualBox:~$ bash Tugas-2.sh
d lebih kecil dari c
calviendanny@calviendanny-VirtualBox:~$ nano Tugas-2.sh
calviendanny@calviendanny-VirtualBox:~$
```