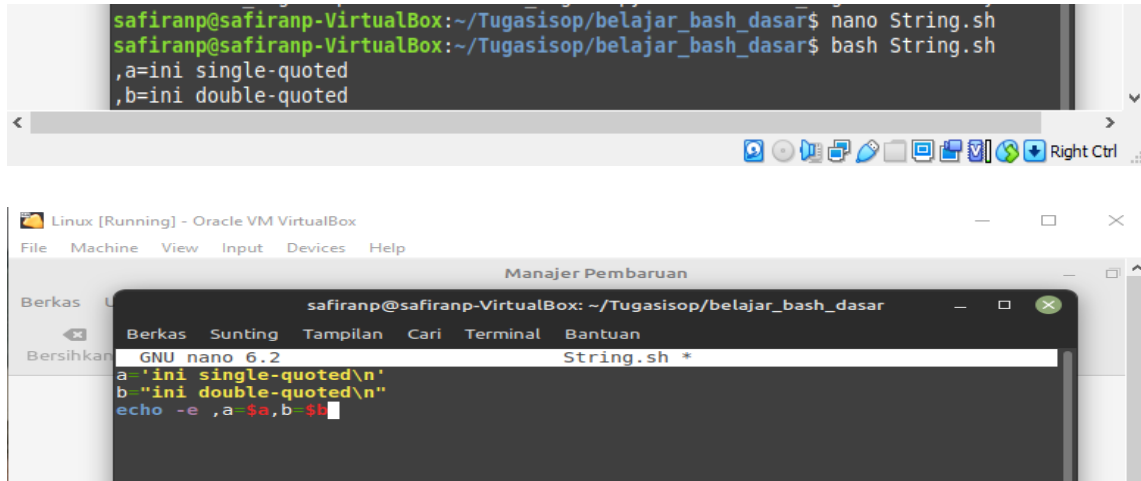


## TUGAS II SHELL SCRIPTING

### A. STRING



```
safirnp@safirnp-VirtualBox:~/Tugasisop/belajar_bash_dasar$ nano String.sh
safirnp@safirnp-VirtualBox:~/Tugasisop/belajar_bash_dasar$ bash String.sh
,a=ini single-quoted
,b=ini double-quoted
```

```
GNU nano 6.2 String.sh *
a='ini single-quoted\\n'
b="ini double-quoted\\n"
echo -e ,a=$a,b=$b
```

String adalah tipe data yang tidak dapat diubah (**hanya-baca**). Ini dapat dideklarasikan dalam tanda kutip tunggal (s=' ') atau tanda kutip ganda (s="").

Contohnya seperti gambar diatas:

- Dalam tanda kutip tunggal (single-quoted) (' ')
- Dalam tanda kutip ganda (double-quoted) (" ")

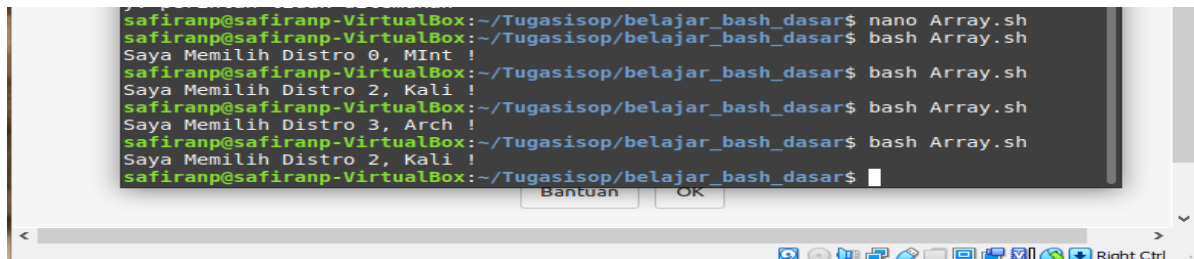
Fungsi dari :

(\n ) Menampilkan baris baru.

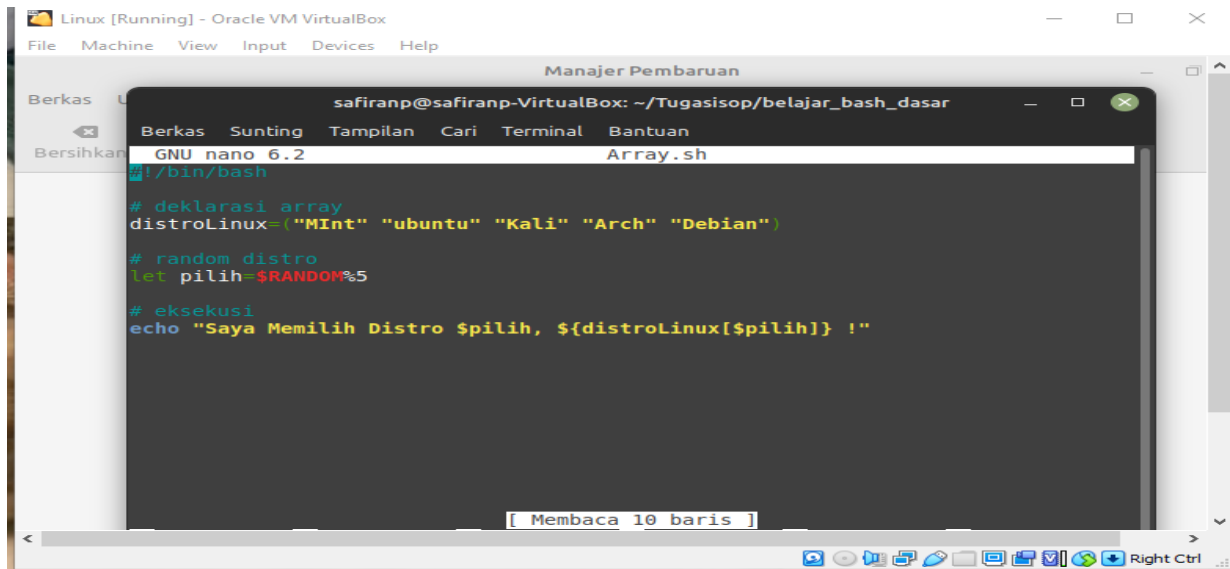
Dengan begitu terdapat opsi (-e) digunakan untuk menerjemahkan karakter (\n) yang termasuk karakter backslash-escape.

**echo** sebenarnya dapat berfungsi dalam berbagai hal, namun untuk menjelaskan gambar diatas **echo** digunakan untuk menampilkan baris pada output.

### B. ARRAY



```
safirnp@safirnp-VirtualBox:~/Tugasisop/belajar_bash_dasar$ nano Array.sh
safirnp@safirnp-VirtualBox:~/Tugasisop/belajar_bash_dasar$ bash Array.sh
Saya Memilih Distro 0, MInt !
safirnp@safirnp-VirtualBox:~/Tugasisop/belajar_bash_dasar$ bash Array.sh
Saya Memilih Distro 2, Kali !
safirnp@safirnp-VirtualBox:~/Tugasisop/belajar_bash_dasar$ bash Array.sh
Saya Memilih Distro 3, Arch !
safirnp@safirnp-VirtualBox:~/Tugasisop/belajar_bash_dasar$ bash Array.sh
Saya Memilih Distro 2, Kali !
safirnp@safirnp-VirtualBox:~/Tugasisop/belajar_bash_dasar$
```



```
safirarp@safirarp-VirtualBox: ~/Tugasisop/belajar_bash_dasar
GNU nano 6.2 Array.sh
#!/bin/bash

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

# random distro
let pilih=$RANDOM%5

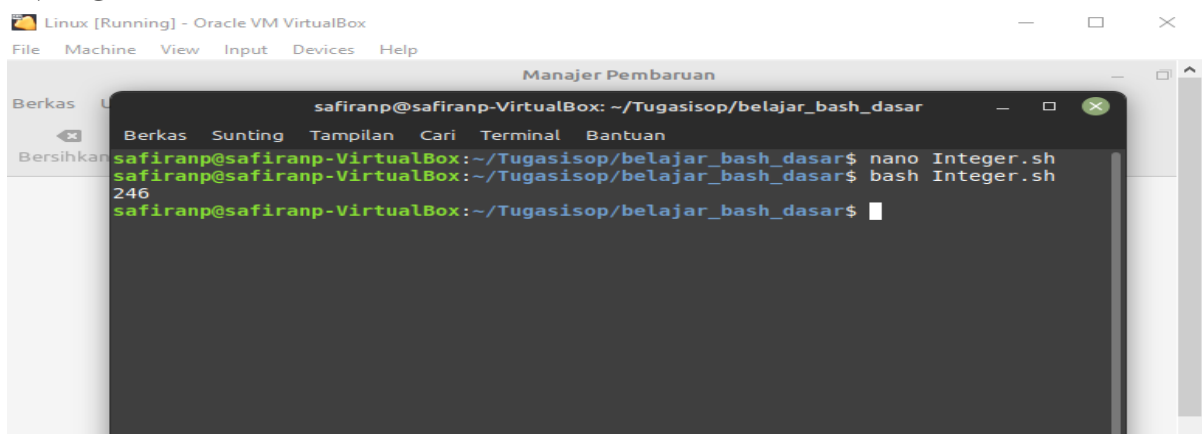
# eksekusi
echo "Saya Memilih Distro $pilih, ${distroLinux[$pilih]} !"

[ Membaca 10 baris ]
```

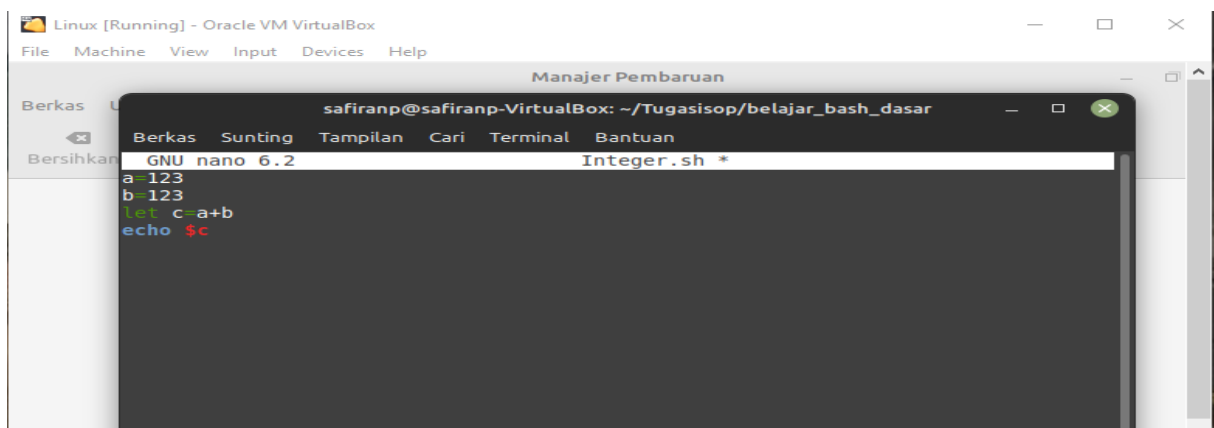
**Array** adalah kumpulan variabel dengan tipe sejenis, dimana array ini merupakan feature Bash yang cukup menarik. Array juga merupakan salah satu hal yang cukup penting dalam bahasa pemrograman, bahwa bisa dibayangkan array ini sebagai tumpukan buku - buku di meja belajar. Inisialisasi array sebagai berikut :

Array=(element1 elemen t2 ... elementN)

### C. INTEGER



```
safirarp@safirarp-VirtualBox: ~/Tugasisop/belajar_bash_dasar
safirarp@safirarp-VirtualBox: ~/Tugasisop/belajar_bash_dasar$ nano Integer.sh
safirarp@safirarp-VirtualBox: ~/Tugasisop/belajar_bash_dasar$ bash Integer.sh
246
safirarp@safirarp-VirtualBox: ~/Tugasisop/belajar_bash_dasar$
```



```
GNU nano 6.2 Integer.sh *
a=123
b=123
let c=a+b
echo $c
```

**Integer** merupakan Mencoba mencari cara mengonversi argumen ke integer untuk menjalankan aritmatika, dan kemudian mencetaknya. Contohnya seperti gambar diatas merupakan contoh penggunaan integer tipe data numerik yang biasa digunakan apabila bertemu dengan bilangan bulat, seperti 1, 27, 100, dll.

Fungsi dari **let** sendiri yaitu perintah untuk menghitung khususnya untuk perhitungan yang sederhana atau dasar.

Fungsi dari \$ yaitu untuk memanggil serta membaca jawaban dari variabel yang diituju, bukan untuk membaca keseluruhan melainkan jawaban yang dituju.

#### D. OPERASI MATEMATIKA

```
safirnp@safirnp-VirtualBox:~/Tugasisop/belajar_bash_dasar$ nano Op-matematika.sh
sh
safirnp@safirnp-VirtualBox:~/Tugasisop/belajar_bash_dasar$ bash Op-matematika.sh
sh
a + b = 26
a - b = -6
a * b = 160
a / b = 0
a % b = 10
a = 10
```

```
Linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

Manajer Pembaruan

Berkas U
Bersihkan GNU nano 6.2 Op-matematika.sh *
#!/bin/bash

a=10
b=16

#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 ))

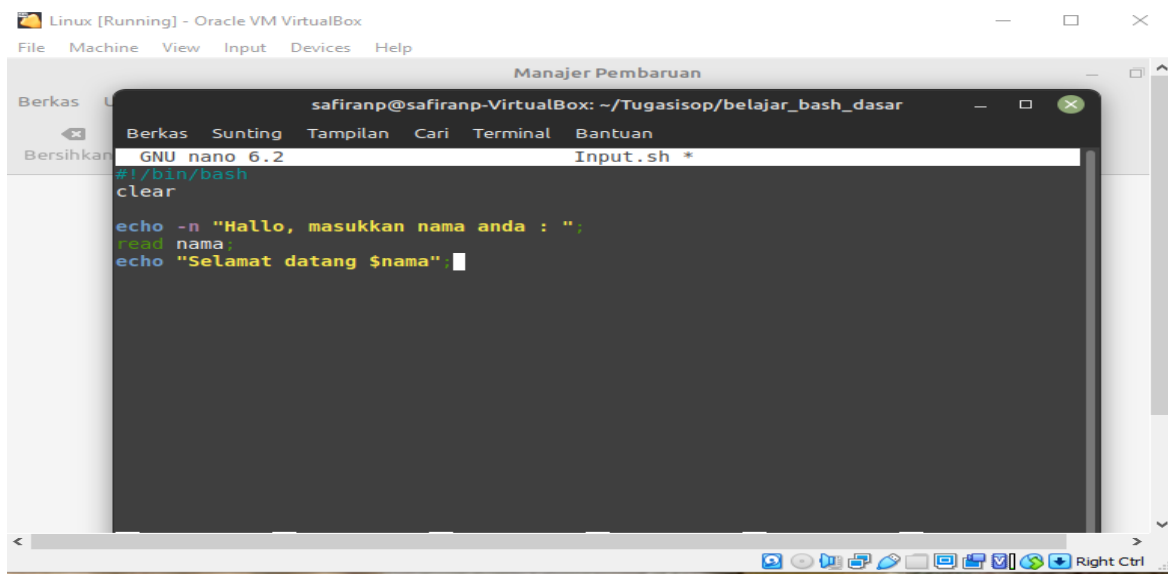
echo "a + b = $jumlah"
echo "a - b = $kurang"
echo "a * b = $kali"
echo "a / b = $bagi"
```

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

**Operasi Matematika** diatas ada yang menggunakan let dan ada yang menggunakan expr. Untuk let sendiri digunakan sebagai operasi matematika dasar atau sederhana sedangkan expr digunakan sebagai Perintah mengevaluasi dan menampilkan output standar dari ekspresi yang diberikan. Setiap ekspresi yang berbeda dianggap sebagai pernyataan. Ekspresi integer dan string dan ekspresi reguler digunakan dengan perintah "expr". Fungsi expr begitu berdaya guna baik untuk operasi matematika.

## E. INPUT

Untuk membaca inputan dari user digunakan `read`, dengan format penulisan “`read namavariabel`”.



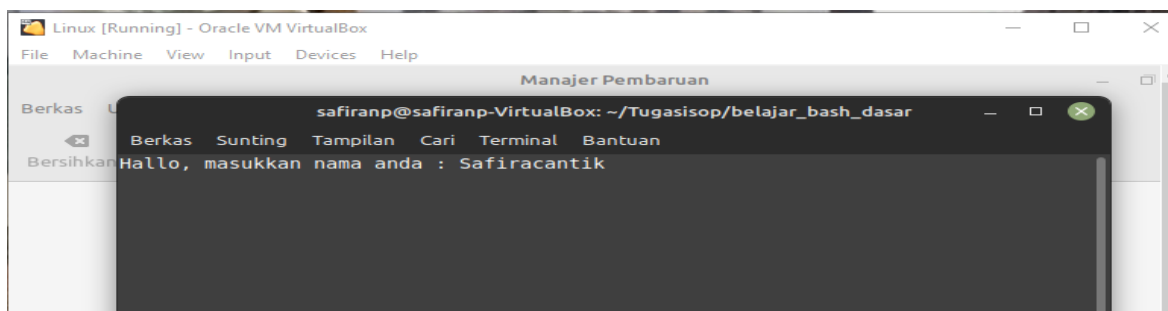
```
safirnp@safirnp-VirtualBox: ~/Tugasisop/belajar_bash_dasar
GNU nano 6.2 Input.sh *
#!/bin/bash
clear

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

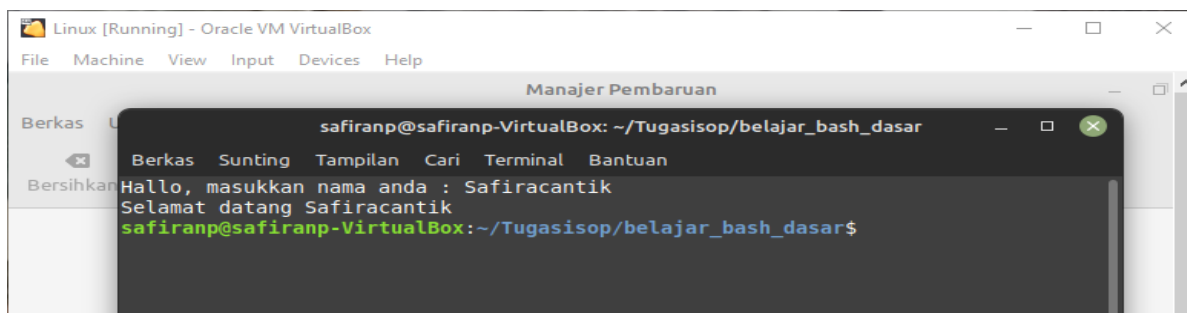
Perintah **clear** di Linux fungsinya untuk membersihkan layar terminal. Misalnya kita mengetik banyak perintah hingga memenuhi layar pada terminal. Nah untuk membersihkan ketik perintah `clear`.

Inputan dari user tersimpan ke dalam variabel `nama`.

Hasilnya



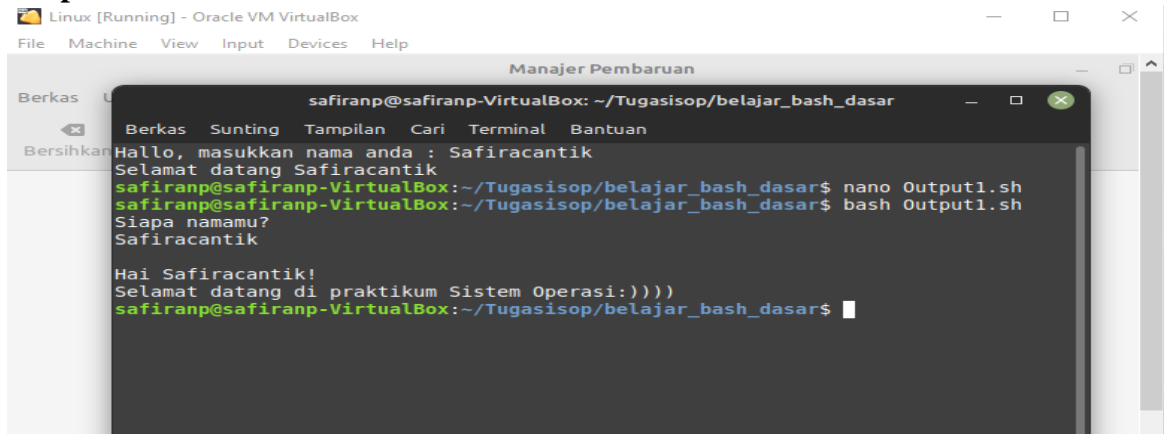
```
safirnp@safirnp-VirtualBox: ~/Tugasisop/belajar_bash_dasar
Hallo, masukkan nama anda : Safiracantik
```



```
safirnp@safirnp-VirtualBox: ~/Tugasisop/belajar_bash_dasar$
Selamat datang Safiracantik
safirnp@safirnp-VirtualBox:~/Tugasisop/belajar_bash_dasar$
```

## F. OUTPUT

### Output1.sh



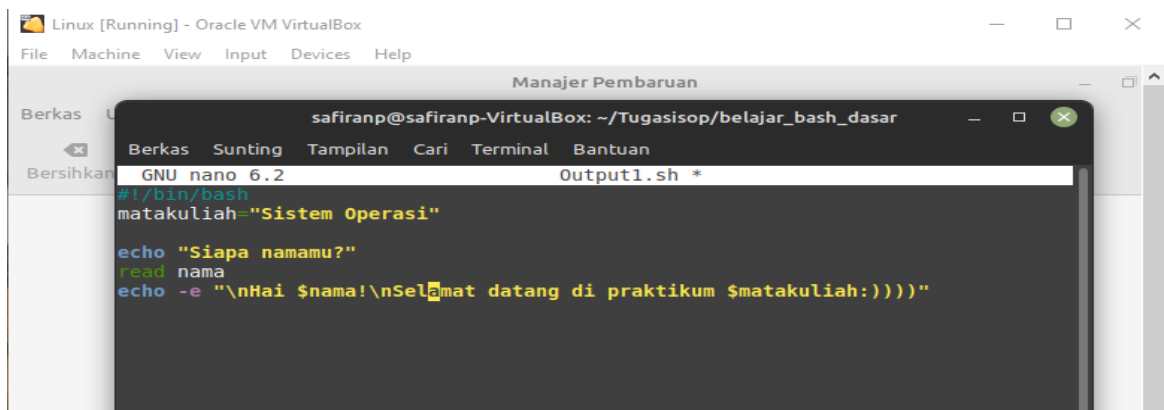
```
Linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

Manajer Pembaruan

Berkas U
Bersihkan

safirarp@safirarp-VirtualBox: ~/Tugasisop/belajar_bash_dasar
Halo, masukkan nama anda : Safiracantik
Selamat datang Safiracantik
safirarp@safirarp-VirtualBox:~/Tugasisop/belajar_bash_dasar$ nano Output1.sh
safirarp@safirarp-VirtualBox:~/Tugasisop/belajar_bash_dasar$ bash Output1.sh
Siapa namamu?
Safiracantik

Hai Safiracantik!
Selamat datang di praktikum Sistem Operasi:))))
safirarp@safirarp-VirtualBox:~/Tugasisop/belajar_bash_dasar$
```



```
Linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

Manajer Pembaruan

Berkas U
Bersihkan

GNU nano 6.2 Output1.sh *
#!/bin/bash
matakuliah="Sistem Operasi"

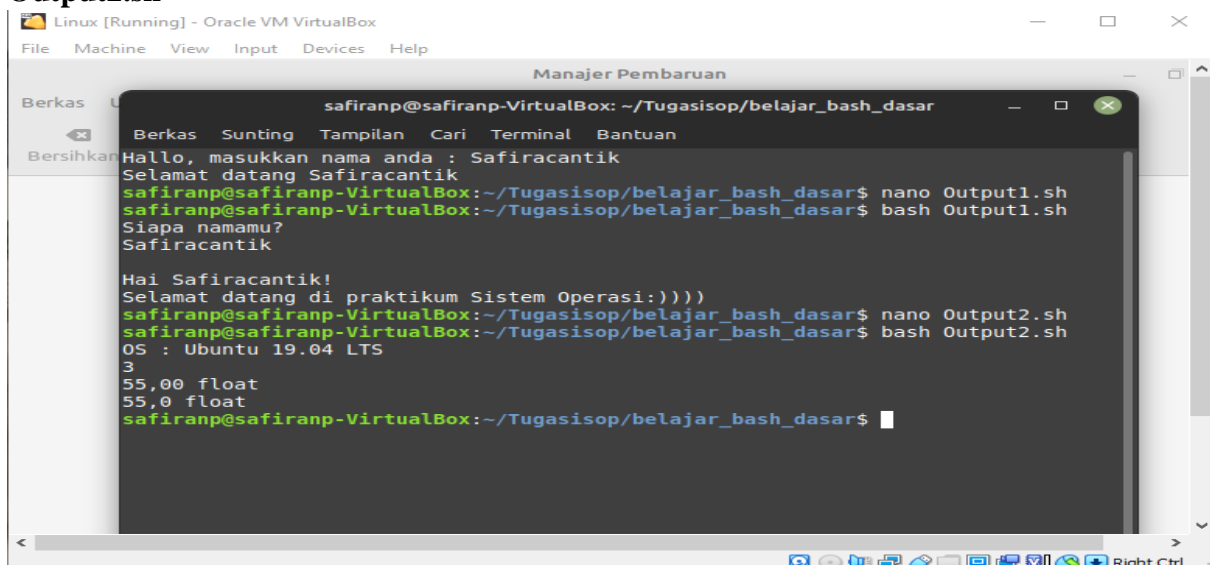
echo "Siapa namamu?"
read nama
echo -e "\nHai $nama!\nSelamat datang di praktikum $matakuliah:)))))"
```

Pertama kita membuat berkas Output1.sh dengan nano. Setelah itu kita ketik echo yang berfungsi untuk menampilkan . Lalu ketik :

- o **read nama** yang berfungsi untuk membaca nama yang sudah terinput.
- o **opsi (-e)** digunakan untuk menerjemahkan karakter (**n**) yang termasuk karakter backslash-escape.

Memanggil isi file menggunakan perintah bash output1.sh dan akan menampilkan output nama yang wajib kita isi lalu kita Enter, maka akan muncul kalimat seperti gambar diatas.

### Output2.sh



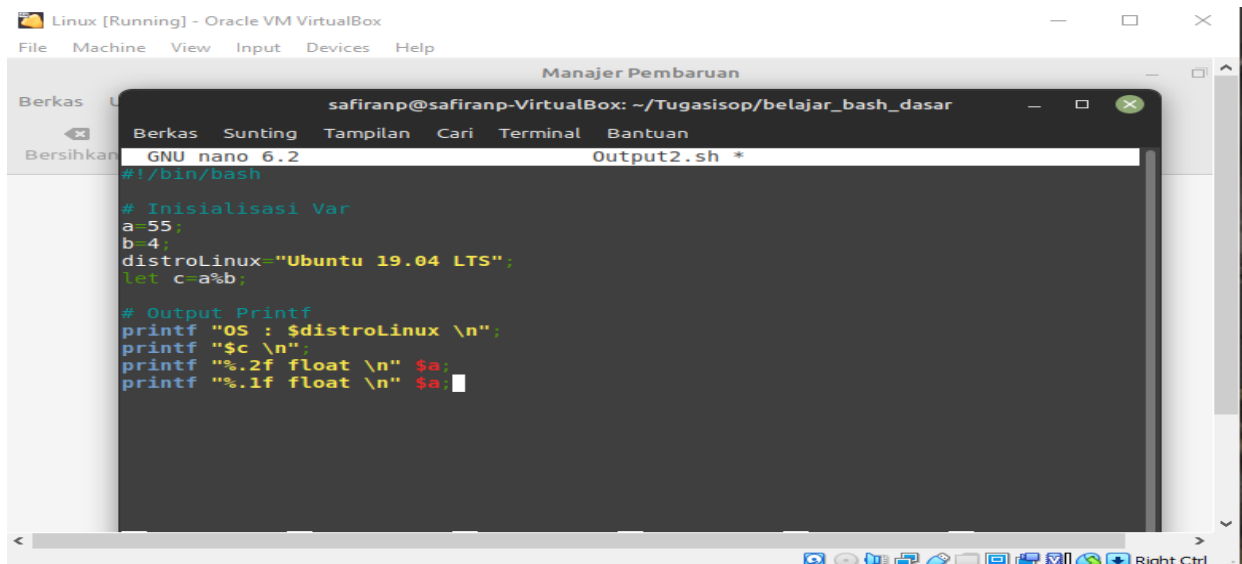
```
Linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

Manajer Pembaruan

Berkas U
Bersihkan

safirarp@safirarp-VirtualBox: ~/Tugasisop/belajar_bash_dasar
Halo, masukkan nama anda : Safiracantik
Selamat datang Safiracantik
safirarp@safirarp-VirtualBox:~/Tugasisop/belajar_bash_dasar$ nano Output1.sh
safirarp@safirarp-VirtualBox:~/Tugasisop/belajar_bash_dasar$ bash Output1.sh
Siapa namamu?
Safiracantik

Hai Safiracantik!
Selamat datang di praktikum Sistem Operasi:))))
safirarp@safirarp-VirtualBox:~/Tugasisop/belajar_bash_dasar$ nano Output2.sh
safirarp@safirarp-VirtualBox:~/Tugasisop/belajar_bash_dasar$ bash Output2.sh
OS : Ubuntu 19.04 LTS
3
55.00 float
55.0 float
safirarp@safirarp-VirtualBox:~/Tugasisop/belajar_bash_dasar$
```



```
safirnp@safirnp-VirtualBox: ~/Tugasisop/belajar_bash_dasar
GNU nano 6.2 Output2.sh *
#!/bin/bash

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

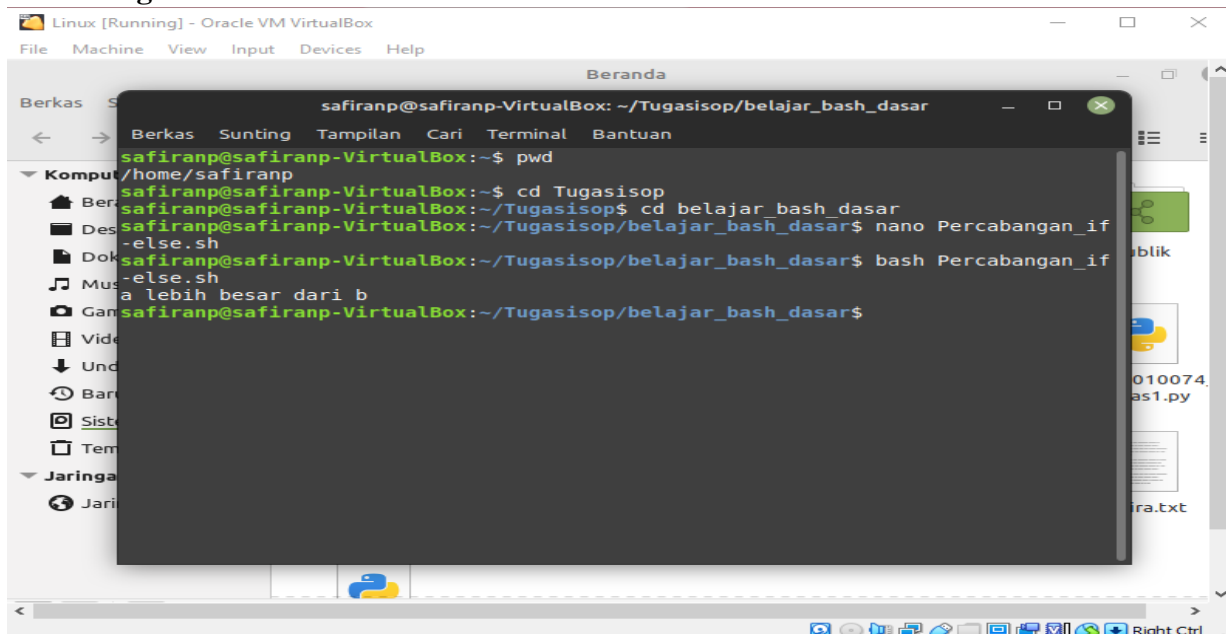
# Output Printf
printf "OS : $distroLinux \n"
printf "$c \n"
printf "%.2f float \n" $a
printf "%.1f float \n" $a
```

Menampilkan output di layar merupakan komponen penting dari setiap bahasa pemrograman. Fungsi puts di C digunakan untuk melakukan ini. Panggilan puts () di C adalah metode penanganan file yang digunakan untuk mencetak baris ke layar output. Fungsi puts () sangat mirip dengan fungsi printf(). Fungsi puts () digunakan untuk menampilkan string yang telah dibaca sebelumnya menggunakan fungsi get() atau scanf() pada terminal. Nilai integer yang dikembalikan oleh fungsi puts () mewakili jumlah karakter yang ditulis pada terminal.

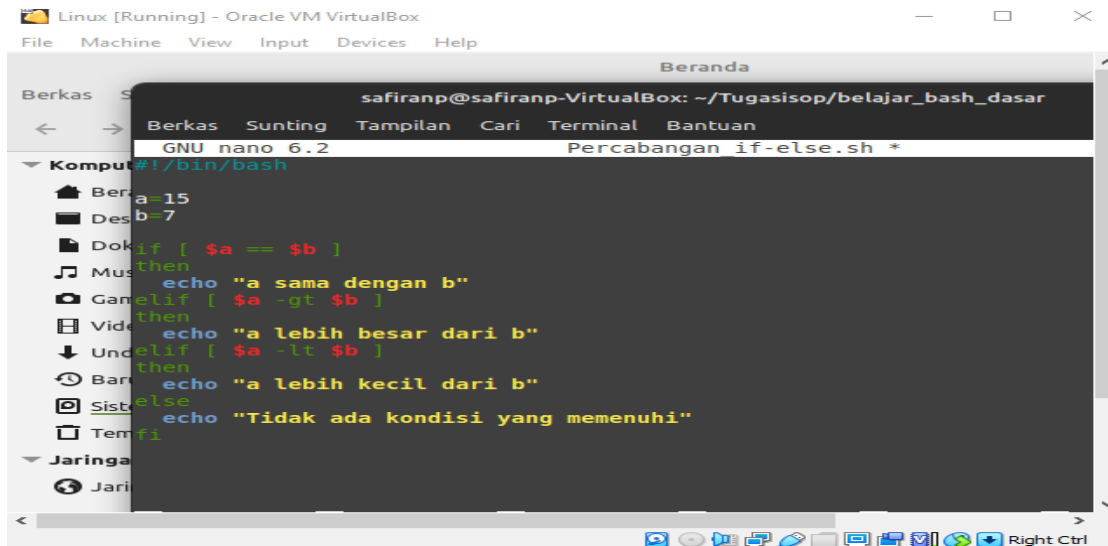
Selain menggunakan output "echo", output bisa juga dilakukan dengan menggunakan printf seperti pada bahasa pemrograman C. Jika string berisi karakter pemformatan seperti '% s,' printf() akan memberikan hasil yang tidak terduga dalam fungsi puts(). Juga, jika "str" adalah string yang disediakan user.

## G. PERCABANGAN

### Percabangan IF-ELSE



```
safirnp@safirnp-VirtualBox: ~$ pwd
/home/safirnp
safirnp@safirnp-VirtualBox:~$ cd Tugasisop
safirnp@safirnp-VirtualBox:~/Tugasisop$ cd belajar_bash_dasar
safirnp@safirnp-VirtualBox:~/Tugasisop/belajar_bash_dasar$ nano Percabangan_if-else.sh
safirnp@safirnp-VirtualBox:~/Tugasisop/belajar_bash_dasar$ bash Percabangan_if-else.sh
a lebih besar dari b
safirnp@safirnp-VirtualBox:~/Tugasisop/belajar_bash_dasar$
```



```
safirnp@safirnp-VirtualBox: ~/Tugasisop/belajar_bash_dasar
GNU nano 6.2 Percabangan if-else.sh *
#!/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
```

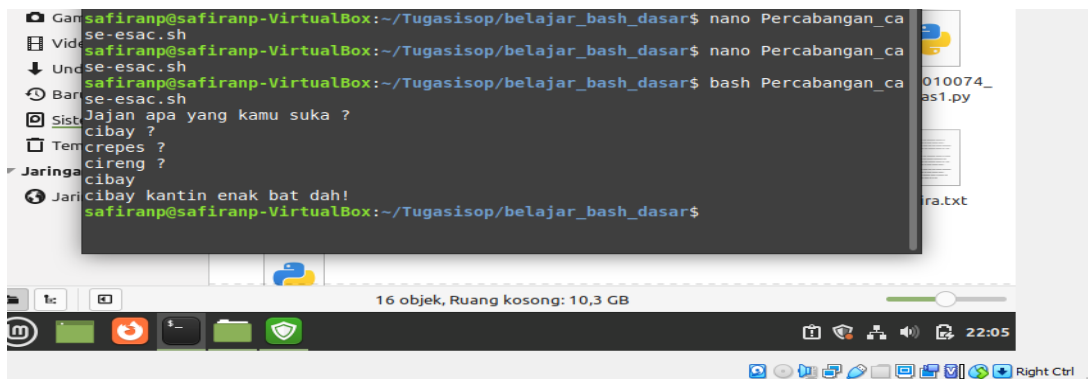
Pertama kita buka file dengan ketik nano.sh

Karena kita ingin menggunakan Perintah If-Else maka, diketahui disini terdapat integer a = 15, b = 7. Terdapat 4 kemungkinan kondisi yang paling tepat untuk mendeskripsikan a dan b. Dengan begitu kita gunakan if-else untuk memberikan kemungkinan terdekatnya. Fungsi dari beberapa Perintah diatas sebagai berikut ::

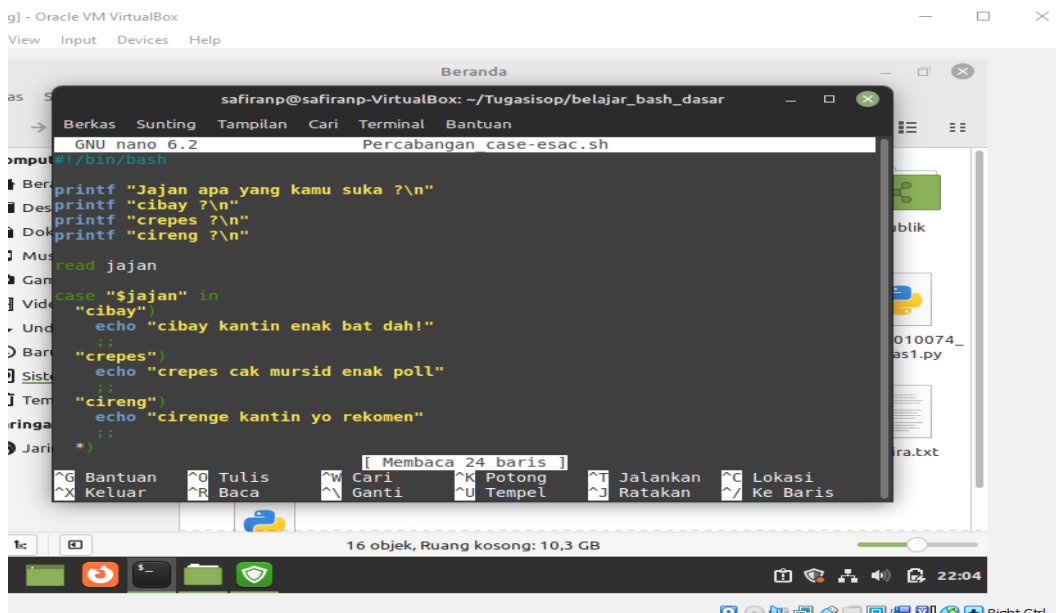
1. **(-gt)** INTEGER1 lebih besar dari INTEGER2
2. **(-lt)** INTEGER1 lebih kecil dari INTEGER2

Sehingga di dapat bahwa output (a lebih besar dari b)

## Percabangan CASE-ESAC



```
safirnp@safirnp-VirtualBox: ~/Tugasisop/belajar_bash_dasar$ nano Percabangan_case-esac.sh
safirnp@safirnp-VirtualBox: ~/Tugasisop/belajar_bash_dasar$ nano Percabangan_case-esac.sh
safirnp@safirnp-VirtualBox: ~/Tugasisop/belajar_bash_dasar$ bash Percabangan_case-esac.sh
Jajan apa yang kamu suka ?
cibay ?
crepes ?
cireng ?
cibay
cibay kantin enak bat dah!
```



```
safirnp@safirnp-VirtualBox: ~/Tugasisop/belajar_bash_dasar
GNU nano 6.2 Percabangan case-esac.sh
#!/bin/bash
printf "Jajan apa yang kamu suka ?\n"
printf "cibay ?\n"
printf "crepes ?\n"
printf "cireng ?\n"
read jajan
case "$jajan" in
  "cibay")
    echo "cibay kantin enak bat dah!"
  ;;
  "crepes")
    echo "crepes cak mursid enak poll"
  ;;
  "cireng")
    echo "cireng kantin yo rekomen"
  ;;
  *)
  ;;
esac
```

Konstruksi case ini digunakan untuk menyederhanakan pemakaian if yang berantai atau kompleks, sehingga dengan menggunakan case, kondisi dapat dikelompokkan secara logis dan lebih mudah dan jelas dalam penulisannya. Dengan Penjelasan

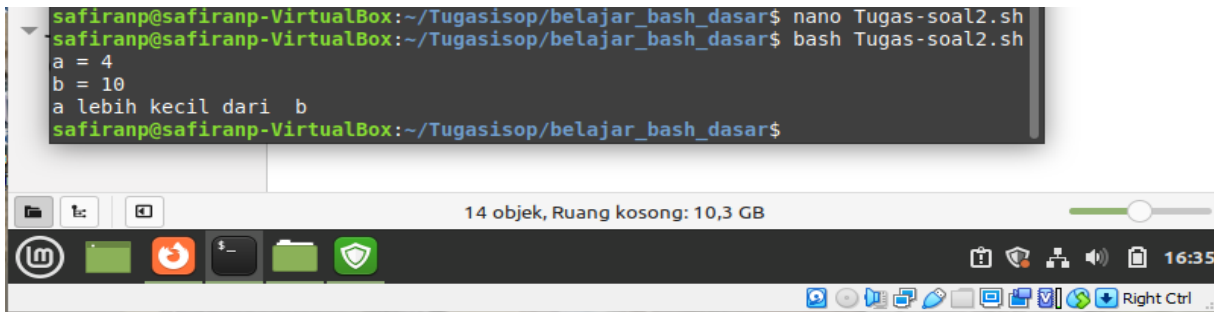
- Variabel adalah nilai yang akan dibandingkan dengan match\_1 sampai match\_n
- misalkan jika nilai dari variable sama dengan nilai match\_1 maka sistem akan mengeksekusi instruksi 1.x.
- Setiap kelompok instruksi diakhiri dengan tanda ;;
- Di dalam \*) adalah instruksi-instruksi yang dijalankan dari match\_1 sampai match\_n tidak ada yang bernilai true.
- Case diakhiri dengan esac.

Contoh seperti gambar diatas

1. Membuat Berkas dengan nano namafile.sh
2. Masukkan script
3. Panggil hasil dari Script yang sudah dibuat dengan format bash namafile.sh



## PENJELASAN SOAL TUGAS 2

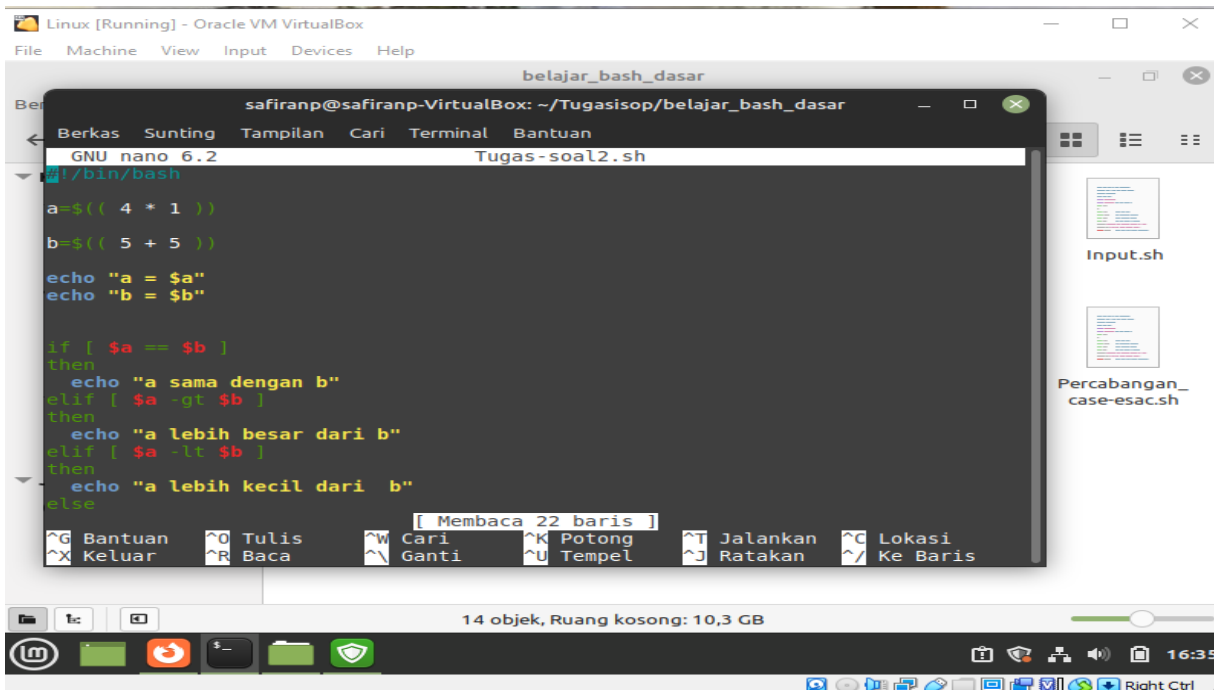


```
safirarp@safirarp-VirtualBox:~/Tugasisop/belajar_bash_dasar$ nano Tugas-soal2.sh
safirarp@safirarp-VirtualBox:~/Tugasisop/belajar_bash_dasar$ bash Tugas-soal2.sh
a = 4
b = 10
a lebih kecil dari b
safirarp@safirarp-VirtualBox:~/Tugasisop/belajar_bash_dasar$
```

Pertama, kita buat berkas nya dengan format namafile.sh.

Kedua kita ketik dalam terminal aritmatika dengan Percabangan. Disini saya menaruh 2 variabel yaitu a dan b (Perkalian dan Penjumlahan). Dapat dilihat dalam Gambar dibawah saya menggunakan Percabangan IF-ELSE dimana dapat mengeluarkan output if else sesuai dengan kondisi variabel yang berisi aritmatika.

Dapat dilihat di Gambar diatas merupakan Output dengan mengetik bash namafile.sh sudah sesuai dengan Kondisi aritmatika yang saya jalankan. Dimana Variabel a lebih kecil dari b.



```
safirarp@safirarp-VirtualBox: ~/Tugasisop/belajar_bash_dasar
GNU nano 6.2 Tugas-soal2.sh
a=$(( 4 * 1 ))
b=$(( 5 + 5 ))
echo "a = $a"
echo "b = $b"

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
[ Membaca 22 baris ]
^G Bantuan ^O Tulis ^W Cari ^K Potong ^T Jalankan ^C Lokasi
^X Keluar ^R Baca ^_ Ganti ^U Tempel ^J Ratakan ^_ Ke Baris
```



```
else
echo "Tidak ada kondisi yang memenuhi"
fi
^G Bantuan ^O Tulis ^W Cari ^K Potong ^T Jalankan ^C Lokasi
^X Keluar ^R Baca ^_ Ganti ^U Tempel ^J Ratakan ^_ Ke Baris
```