

Nama : Dina Magdalena Manurung

NPM : 21083010117

Mata Kuliah : Sistem Operasi B

Praktek script bash di Linux (Pertemuan 3)

1. Pertama, kita membuat directori terlebih dahulu agar lebih mempermudah kita menyatukan semua file nanti. Saya membuat directori dengan nama "Pertemuan3"
Pada percobaan pertama, saya membuat file bash **.sh** dengan menggunakan nano dan nama file "Penggunaan_string.sh"

```
dindin@dindin-VirtualBox:~$ mkdir Pertemuan3
dindin@dindin-VirtualBox:~$ cd Pertemuan3
dindin@dindin-VirtualBox:~/Pertemuan3$ nano Penggunaan_array.sh
dindin@dindin-VirtualBox:~/Pertemuan3$
```

- Pada script bash penggunaan_string, a & b dibuat sebagai variabel dengan perbedaan di tanda kutip (yang artinya tanda petik satu dan tanda petik dua bisa dipakai).

Untuk print bisa menggunakan **echo** dan cara nampilin isi string bisa menggunakan **\$variabel_name**

\n berfungsi agar output setelahnya berada pada paragraf berikutnya.

```
dindin@dindin-VirtualBox: ~/Pertemuan3
GNU nano 6.2      Penggunaan_string.sh
a "ini single-quoted\n"
b "ini double-quoted\n"
echo -e ,a $a,b $b

[ Read 3 lines ]
Help Write Out Where Is Cut Execute Location
Exit Read File Replace Paste Justify Go To Li
```

- Ini untuk output dari script diatas, cara nampilin output bisa menggunakan **bash variable_name**

```
dindin@dindin-VirtualBox:~/Pertemuan3$ nano Penggunaan_string.sh
dindin@dindin-VirtualBox:~/Pertemuan3$ bash Penggunaan_string.sh
,a=ini single-quoted
,b=ini double-quoted
```

- Ini contoh ketika script bash dijalankan di terminal dan tidak dijalankan dengan nano dan bash. Tapi kekurangannya adalah script di dalam terminal tidak bisa diedit maupun di save

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

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

2. Pada percobaan kedua, saya membuat file bash **.sh** dengan menggunakan nano dan nama file "Penggunaan_array.sh"

```
Penggunaan_array.sh
dinding@dinding-VirtualBox:~/Pertemuan3$ nano Penggunaan_array.sh
dinding@dinding-VirtualBox:~/Pertemuan3$
```

- **distroLinux=(array)** kita mendeklarasikan array
- **let pilih=\$RANDOM%5** disini kita memakai syntax random agar output yang dihasilkan dapat mengacak dari array yang telah ada
- **#!/bin/bash** ketika digunakan dalam skrip digunakan untuk menginstruksikan sistem operasi untuk menggunakan bash sebagai penerjemah perintah.

```
dinding@dinding-VirtualBox: ~/Pertemuan3
File Edit View Search Terminal Help
GNU nano 6.2 Penggunaan_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]} !"

Help Write Out Where Is Cut Execute Location
Exit Read File Replace Paste Justify Go To L
```

- Ini untuk output dari code diatas, cara nampilin output bisa menggunakan **bash variable_name**.
Output yang keluar akan random sesuai code yang telah dibuat.

```
dinding@dinding-VirtualBox: ~/Pertemuan3
File Edit View Search Terminal Help
dinding@dinding-VirtualBox:~/Pertemuan3$ nano Penggunaan_array.sh
dinding@dinding-VirtualBox:~/Pertemuan3$ bash Penggunaan_array.sh
Saya Memilih Distro 2, Kali !
dinding@dinding-VirtualBox:~/Pertemuan3$ bash Penggunaan_array.sh
Saya Memilih Distro 1, Ubuntu !
dinding@dinding-VirtualBox:~/Pertemuan3$ bash Penggunaan_array.sh
Saya Memilih Distro 4, Debian !
dinding@dinding-VirtualBox:~/Pertemuan3$ bash Penggunaan_array.sh
Saya Memilih Distro 1, Ubuntu !
dinding@dinding-VirtualBox:~/Pertemuan3$ bash Penggunaan_array.sh
Saya Memilih Distro 2, Kali !
dinding@dinding-VirtualBox:~/Pertemuan3$ bash Penggunaan_array.sh
Saya Memilih Distro 0, Mint !
dinding@dinding-VirtualBox:~/Pertemuan3$ bash Penggunaan_array.sh
Saya Memilih Distro 4, Debian !
dinding@dinding-VirtualBox:~/Pertemuan3$ bash Penggunaan_array.sh
Saya Memilih Distro 4, Debian !
dinding@dinding-VirtualBox:~/Pertemuan3$ bash Penggunaan_array.sh
Saya Memilih Distro 0, Mint !
dinding@dinding-VirtualBox:~/Pertemuan3$ bash Penggunaan_array.sh
Saya Memilih Distro 0, Mint !
dinding@dinding-VirtualBox:~/Pertemuan3$
```

3. Pada percobaan ketiga, saya membuat file bash **.sh** dengan menggunakan nano dan nama file "Penggunaan_integer.sh"

```
dindin@dindin-VirtualBox: ~/Pertemuan3$ nano Penggunaan_integer.sh
dindin@dindin-VirtualBox:~/Pertemuan3$
```

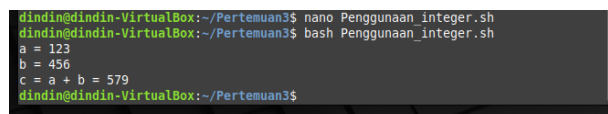
- Pada script bash penggunaan_integer.sh, a dan b sebagai variabel suatu integer (kita bisa mengubah variabel sesuka kita dan bisa memasukkan angkanya juga). Selanjutnya, kita mendeklarasikan variabel c itu penjumlahan dari variabel a dan b dengan perintah **let**.

Lalu, kita membuat **echo** agar menghasilkan output yang sesuai perintah. Pada script dibawah ini kita ingin menghasilkan output perintah variabel c dengan mengetik **\$c**.



```
GNU nano 6.2      Penggunaan_integer.sh *
a=123
b=456
let c=a+b
echo "a = $a"
echo "b = $b"
echo "c = a + b = $c"
```

- Dibawah ini adalah output dari script bash diatas, outputnya adalah 579= c yang merupakan penjumlahan dari $c = a + b \Rightarrow c = 123 + 456 \Rightarrow c = 579$

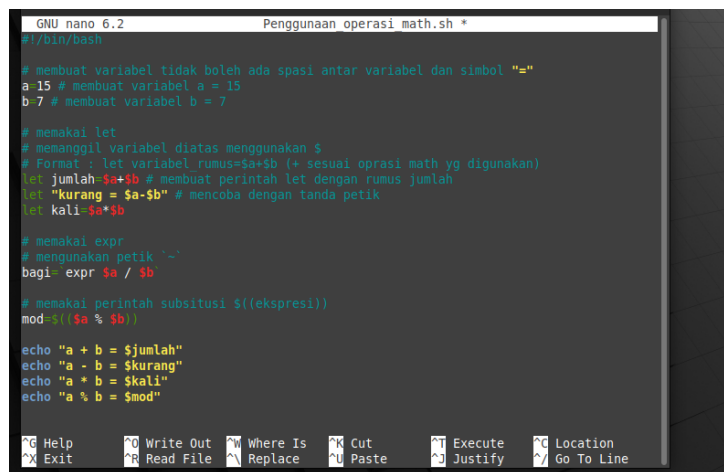


```
dindin@dindin-VirtualBox:~/Pertemuan3$ nano Penggunaan_integer.sh
dindin@dindin-VirtualBox:~/Pertemuan3$ bash Penggunaan_integer.sh
a = 123
b = 456
c = a + b = 579
dindin@dindin-VirtualBox:~/Pertemuan3$
```

4. Pada percobaan keempat, saya membuat file bash **.sh** dengan menggunakan nano dan nama file “Penggunaan_operasi_math.sh”

```
dindin@dindin-VirtualBox:~/Pertemuan3$ nano Penggunaan_operasi_math.sh
```

- **Penjelasan ada dalam script.** Pada script dibawah ini, menggunakan 3 cara pengoperasian aritmatika yaitu **let**, **expr**, **substitusi \$(x)**. Urutan penempatan perintah ini berbeda-beda sesuai dengan script dibawah. Tetapi untuk mengeluarkannya dalam output tetap sama yaitu menggunakan **echo**.
 - **let a=var1+var2**, tidak menggunakan kutip jadi tidak boleh ada spasi
let “b = var1 – var2”, menggunakan kutip boleh ada spasi
 - **expr** boleh memakai spasi walaupun tanpa tanda kutip
expr langsung menampilkan hasil operasi tanpa menyimpan terlebih dahulu ke dalam variabel
a=\$(expr 12 – 1), menyimpan hasil operasi ke dalam variabel \$a
 - **a=\$((2 + 3))**, format penulisan dasar tanpa menggunakan kutip.
b=\$((a + 5)), dapat memasukkan variabel tanpa tanda \$.



```
GNU nano 6.2      Penggunaan_operasi_math.sh *
#!/bin/bash

# membuat variabel tidak boleh ada spasi antar variabel dan simbol "="
a 15 # membuat variabel a = 15
b 7  # membuat variabel b = 7

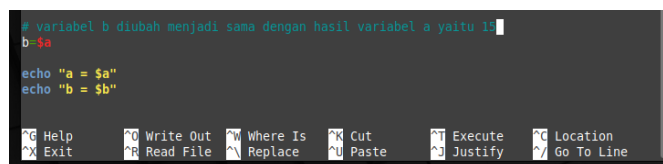
# memakai let
# memanggil variabel diatas menggunakan $
# Format : let variabel_rumus=$a+$b (+ sesuai oprasi math yg digunakan)
let jumlah=$a+$b # membuat perintah let dengan rumus jumlah
let "kurang = $a-$b" # mencoba dengan tanda petik
let kali=$a*$b

# memakai expr
# menggunakan petik '-'
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 = $mod"
```

- Pada script dibawah ini, nilai variabel b diubah menjadi sama dengan nilai variabel a. Dan menghasilkan output nilai a=b dengan menggunakan perintah **echo**.



```
# variabel b diubah menjadi sama dengan hasil variabel a yaitu 15
b $a

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

- Dibawah ini adalah output dari script diatas. Output dibawah ini berasal dari perintah **echo**.

Contohnya, pada `let kali=$a*$b` artinya kita membuat variabel kali sebagai rumus dengan a kali b. Untuk mengeluarkan outputnya kita bisa menggunakan echo, `echo "a * b = $kali"` a*b dalam echo artinya string dan untuk memanggil perintah rumus yang telah kita buat bisa menggunakan ***\$variabel_rumus***

```
dinding@dindin-VirtualBox:~/Pertemuan3$ nano Penggunaan_operasi_math.sh
dinding@dindin-VirtualBox:~/Pertemuan3$ bash Penggunaan_operasi_math.sh
a + b = 22
a - b = 8
a * b = 105
a % b = 1
a = 15
b = 15
dinding@dindin-VirtualBox:~/Pertemuan3$
```

- Dibawah ini terdapat contoh penggunaan perintah operasi aritmatika dengan langsung memasukkan angka bukan variabel yang dideklarasikan.

```
GNU nano 6.2          coba.sh *
# Percobaan tanpa variabel yang artinya langsung memasukkan angka

# memakai let
let jumlah=4+5

# memakai expr
kurang=$(expr 6 - 5)
bagi=$(expr 6 / 3)

# memakai perintah substitusi ${(ekspresi)}
kali=$((4 * 5))

echo "4 + 5 = $jumlah"
echo "6 - 5 = $kurang"
echo "6 / 3 = $bagi"
echo "4 * 5 = $kali"

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

```
dinding@dindin-VirtualBox:~$ nano coba.sh
dinding@dindin-VirtualBox:~$ bash coba.sh
4 + 5 = 9
6 - 5 = 1
6 / 3 = 2
4 * 5 = 20
dinding@dindin-VirtualBox:~$
```

5. Pada percobaan kelima, saya membuat file bash **.sh** dengan menggunakan nano dan nama file "Penggunaan_input.sh"

```
dindin@dindin-VirtualBox:~/Pertemuan3$ nano Penggunaan_input.sh
dindin@dindin-VirtualBox:~/Pertemuan3$
```

- Perintah clear digunakan untuk menghapus perintah/script sebelumnya yang ada pada didalam terminal.

Perintah read digunakan ketika kita ingin memasukkan (input) value dalam suatu variabel secara sendiri [format **read variabel_name**].

Setelah itu kita bisa memanggil variabel dengan **\$** dan mengeluarkan output dengan **echo**.

```
GNU nano 6.2      Penggunaan_input.sh *
#!/bin/bash
clear

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

- Setelah itu, kita bisa menjalankan script diatas dengan bash pada terminal.

```
dindin@dindin-VirtualBox:~/Pertemuan3$ nano Penggunaan_input.sh
dindin@dindin-VirtualBox:~/Pertemuan3$ bash Penggunaan_input.sh
```

- Setelah di enter, maka seperti perintah clear. Output akan ada pada terminal baru. Dibawah ini adalah outputnya, sesuai fungsi read kita bisa memasukkan nama secara sendiri.

```
dindin@dindin-VirtualBox: ~/Pertemuan3
File Edit View Search Terminal Help
Hallo, masukkan nama anda : Dina Magdalena Manurung
```

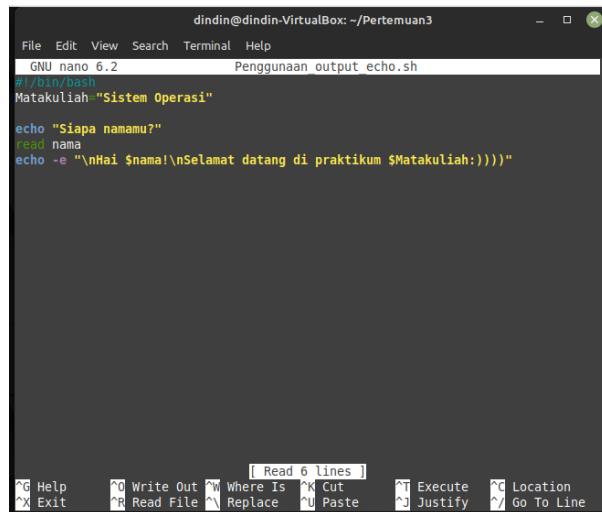
- Setelah input nama dan klik enter, maka output lainnya dalam script akan keluar.

```
dindin@dindin-VirtualBox: ~/Pertemuan3
File Edit View Search Terminal Help
Hallo, masukkan nama anda : Dina Magdalena Manurung
SElamat datang Dina Magdalena Manurung
dindin@dindin-VirtualBox:~/Pertemuan3$
```

6. Pada percobaan keenam, saya membuat file bash **.sh** dengan menggunakan nano dan nama file “Penggunaan_output_echo.sh”

```
dindin@dindin-VirtualBox: ~/Pertemuan3$ nano Penggunaan_output_echo.sh
dindin@dindin-VirtualBox: ~/Pertemuan3$
```

- **echo** adalah salah satu perintah untuk output pada script bash

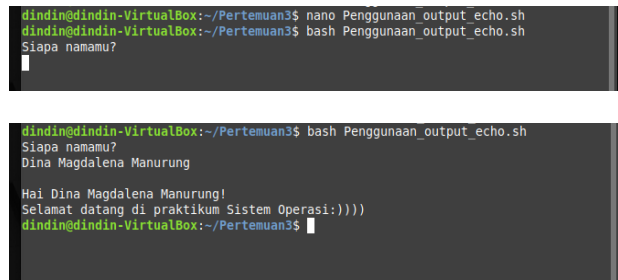


The screenshot shows a terminal window with the nano editor open. The file being edited is 'Penggunaan_output_echo.sh'. The content of the file is as follows:

```
#!/bin/bash
Matakuliah "Sistem Operasi"

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

- Seperti pada script, maka kita bisa memasukkan nama sendiri dan akan keluar kalimat lainnya.



The first screenshot shows the command to run the script:

```
dindin@dindin-VirtualBox:~/Pertemuan3$ nano Penggunaan_output_echo.sh
dindin@dindin-VirtualBox:~/Pertemuan3$ bash Penggunaan_output_echo.sh
Siapa namamu?
```

The second screenshot shows the output of the script after the user has entered their name:

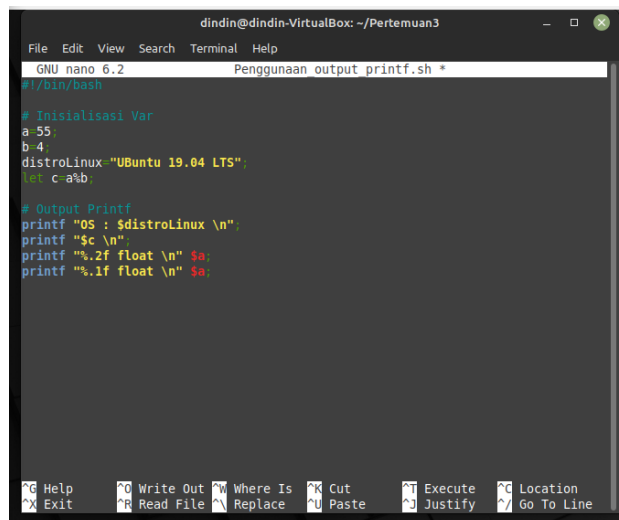
```
dindin@dindin-VirtualBox:~/Pertemuan3$ bash Penggunaan_output_echo.sh
Siapa namamu?
Dina Magdalena Manurung

Hai Dina Magdalena Manurung!
Selamat datang di praktikum Sistem Operasi:)))))
dindin@dindin-VirtualBox:~/Pertemuan3$
```


7. Pada percobaan ketujuh, saya membuat file bash **.sh** dengan menggunakan nano dan nama file “Penggunaan_output_printf.sh”

```
dindin@dindin-VirtualBox:~/Pertemuan3$ nano Penggunaan_output_printf.sh
dindin@dindin-VirtualBox:~/Pertemuan3$
```

- Pada script dibawah ini, kita membuat variabel dan valuenya a=55, b=4, dan distroLinux= UBuntu 19.04 LTS” dengan membuat fungsi c=a%4 (modulus) **printf** adalah perintah output pada script bash selain echo.

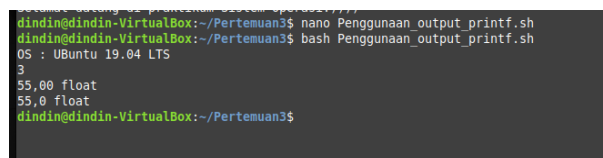


```
GNU nano 6.2  Penggunaan_output_printf.sh *
#! /bin/bash

# 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;
```

- Dibawah ini adalah output dari script diatas dengan menggunakan perintah printf

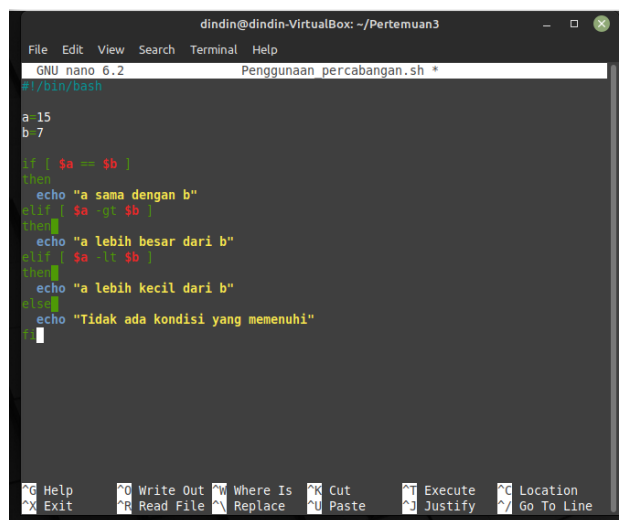


```
dindin@dindin-VirtualBox:~/Pertemuan3$ nano Penggunaan_output_printf.sh
dindin@dindin-VirtualBox:~/Pertemuan3$ bash Penggunaan_output_printf.sh
OS : UBuntu 19.04 LTS
3
55.00 float
55.0 float
dindin@dindin-VirtualBox:~/Pertemuan3$
```

8. Pada percobaan kedelapan, saya membuat file bash **.sh** dengan menggunakan nano dan nama file "Penggunaan_percabangan_IfElse.sh"

```
dindin@dindin-VirtualBox:~/Pertemuan3$ nano Penggunaan_percabangan_IfElse.sh
dindin@dindin-VirtualBox:~/Pertemuan3$
```

- Script dibawah ini kita bisa ambil contoh aritmatika dijadikan percabangan sederhana. Kita mendeklarasikan value dari variabel a dan b.
Kita langsung memasukkan perintah if yaitu **if [\$a == \$b] =>** antar kurung harus dispasi. Dan memasukkan output yang kita ingin dengan then dan echo.
Jika fungsi pada if tidak sesuai, maka kita bisa membuat jika fungsi lainnya dengan elif yaitu **elif [\$a -gt \$b]**. Fungsi -gt artinya lebih besar (ada pada page operasi aritmatika). Dan seterusnya bisa memasukkan fungsi aritmatika jika pernyataan pada if elif diatas tidak sesuai.
Jika a dan b tidak sesuai pada if maupun elif. Maka kita bisa menambahkan else (tanpa memasukkan fungsi aritmatika sebab else adalah pilihan terakhir ketika semua pernyataan tidak sesuai.



```
dindin@dindin-VirtualBox: ~/Pertemuan3
File Edit View Search Terminal Help
GNU nano 6.2 Penggunaan_percabangan.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
```

- Dibawah ini adalah output dari script diatas. 15 lebih besar dari 7, maka dari itu outputnya adalah a lebih besar dari b.

```
dindin@dindin-VirtualBox:~/Pertemuan3$ nano Penggunaan_percabangan_IfElse.sh
dindin@dindin-VirtualBox:~/Pertemuan3$ bash Penggunaan_percabangan_IfElse.sh
a lebih besar dari b
dindin@dindin-VirtualBox:~/Pertemuan3$
```

9. Pada percobaan kesembilan, saya membuat file bash **.sh** dengan menggunakan nano dan nama file "Penggunaan_percabangan_CaseEsac.sh"

```
dindingdinding-VirtualBox:~/Pertemuan3$ nano Penggunaan_percabangan_CaseEsac.sh
dindingdinding-VirtualBox:~/Pertemuan3$
```

- Pada script dibawah ini dibuat contoh tentang pemilihan jajanan kesukaan. Pada printf diawal menjelaskan jajanan apa saja pilihannya. Dan menambahkan perintah read agar bisa input jajanan sendiri. Pada percabangan ini kita menggunakan Case Esac. **Case "\$jajan" in** yang artinya dengan membaca jajan pada variabel read. Setelah itu, memasukkan pilihan yang mungkin terjadi dengan output masing-masing. Pada percabangan ini kita tidak mengulang kata case melainkan diganti dengan ;; Pilihan terakhir ketika pernyataan pada Case tidak ada yang sesuai dapat menggunakan *)

```
dindingdinding-VirtualBox: ~/Pertemuan3
File Edit View Search Terminal Help
GNU nano 6.2 Penggunaan_percabangan_CaseEsac.sh *
#!/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
```

- Dibawah ini adalah output dari script diatas. Kita bisa memasukkan jajanan yang disukai.

```
dindingdinding-VirtualBox:~/Pertemuan3$ nano Penggunaan_percabangan_CaseEsac.sh
dindingdinding-VirtualBox:~/Pertemuan3$ bash Penggunaan_percabangan_CaseEsac.sh
Jajan apa yang kamu suka ?
pentol ?
batagor ?
cireng ?
```

- Ketika kita coba memasukkan “pentol” maka yang keluar adalah output sesuai script diatas.

```
dinding@dindin-VirtualBox:~/Pertemuan3$ bash Penggunaan_percabangan_CaseEsac.sh
Jajan apa yang kamu suka ?
pentol ?
batagor ?
cireng ?
pentol
Pentol buk mah wenak slur!
```

- Dibawah ini adalah percobaan ketika kita input jajanan yang tidak ada dalam pilihan.

```
dinding@dindin-VirtualBox:~/Pertemuan3$ bash Penggunaan_percabangan_CaseEsac.sh
Jajan apa yang kamu suka ?
pentol ?
batagor ?
cireng ?
bakso
Makanan yang kamu suka gaenak hehe
```

Tugas 2

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

- Membuat file bash melalui nano

```
hasil penjumlahan 2 dan 9 (2 + 9) = 11
dinding@dindin-VirtualBox:~/Pertemuan3$ nano tugas_2.sh
dinding@dindin-VirtualBox:~/Pertemuan3$
```

- Script

```
dinding@dindin-VirtualBox: ~/Pertemuan3
File Edit View Search Terminal Help
GNU nano 6.2 tugas_2.sh *
echo "Masukkan angka pertama : "
read x
echo "Masukkan angka kedua : "
read y
echo "Pilihlah operasi yang anda inginkan : "
echo "1. Penjumlahan"
echo "2. Pengurangan"
echo "3. Perkalian"
echo "4. Pembagian"
echo "Masukkan nomor operasi yang anda inginkan (1/2/3/4) : "
read operasi
```

```
if [ $operasi -eq 1 ]
then
    let hasil=x+y
    echo "Hasil penjumlahan $x dan $y ($x + $y) = $hasil"
elif [ $operasi -eq 2 ]
then
    let hasil=x-y
    echo "Hasil pengurangan $x dan $y ($x - $y) = $hasil"
elif [ $operasi -eq 3 ]
then
    let hasil=x*y
    echo "Hasil perkalian $x dan $y ($x * $y) = $hasil"
elif [ $operasi -eq 4 ]
then
    let hasil=x/y
    echo "Hasil pembagian $x dan $y ($x/$y) = $hasil"
else
    echo "Maaf pilihan anda tidak bisa dideteksi"
fi
```

- Output

```
dinding@dindin-VirtualBox:~/Pertemuan3$ nano tugas_2.sh
dinding@dindin-VirtualBox:~/Pertemuan3$ bash tugas_2.sh
Masukkan angka pertama :
2
Masukkan angka kedua :
9
Pilihlah operasi yang anda inginkan :
1. Penjumlahan
2. Pengurangan
3. Perkalian
4. Pembagian
Masukkan nomor operasi yang anda inginkan (1/2/3/4) :
2
Hasil pengurangan 2 dan 9 (2 - 9) = -7
```