

Nama : Radya Ardi Ninang Pudyasuti

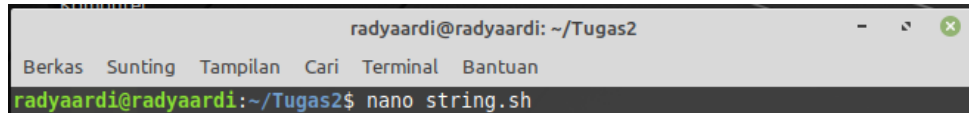
NPM : 21083010097

## Tugas Sistem Operasi 2

### 1. String

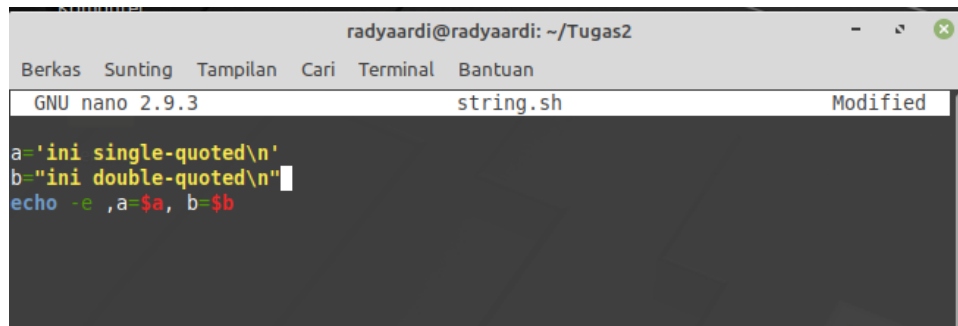
langkah dalam membuat string adalah

- Membuat nano nama file.sh (nano string.sh)



```
radyaardi@radyaardi: ~/Tugas2
Berkas  Sunting  Tampilan  Cari  Terminal  Bantuan
radyaardi@radyaardi:~/Tugas2$ nano string.sh
```

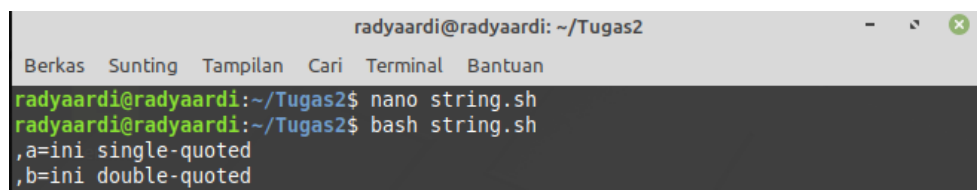
- Pada single quoted hanya digunakan oleh kata-kata atau kalimat yang sederhana. Sedangkan double quoted digunakan untuk memuat variabel. Contohnya lambang dollar (\$), backtick (`), backslash (\), dan lain-lain.
- Di sini fungsi echo sendiri adalah untuk menampilkan output yang diinginkan. Sedangkan lambang dollar atau dollar sign berfungsi untuk menampilkan value.



```
radyaardi@radyaardi: ~/Tugas2
Berkas  Sunting  Tampilan  Cari  Terminal  Bantuan
GNU nano 2.9.3      string.sh      Modified

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

- Setelah kita Menyusun script pada nano, kemudian kita dapat memunculkan output dari script yang telah kita buat tadi melalui “bash string.sh”

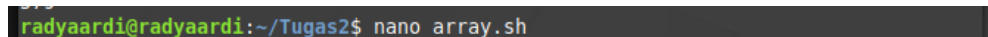


```
radyaardi@radyaardi: ~/Tugas2
Berkas  Sunting  Tampilan  Cari  Terminal  Bantuan
radyaardi@radyaardi:~/Tugas2$ nano string.sh
radyaardi@radyaardi:~/Tugas2$ bash string.sh
,a=ini single-quoted
,b=ini double-quoted
```

### 2. Array

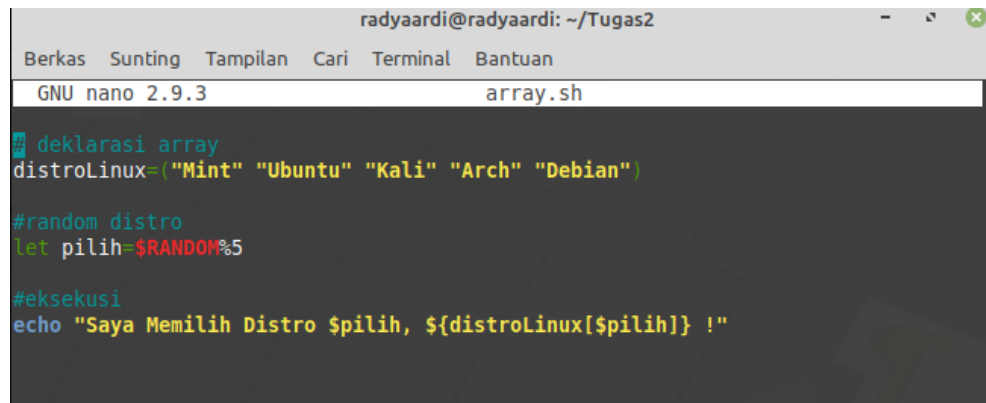
Langkah dalam membuat Array adalah

- Membuat nano nama file.sh (nano array.sh)



```
radyaardi@radyaardi:~/Tugas2$ nano array.sh
```

- Array berfungsi sebagai menampilkan daftar list secara acak dengan menggunakan Random.



```

radyaardi@radyaardi: ~/Tugas2
Berkas  Sunting  Tampilan  Cari  Terminal  Bantuan
GNU nano 2.9.3 array.sh

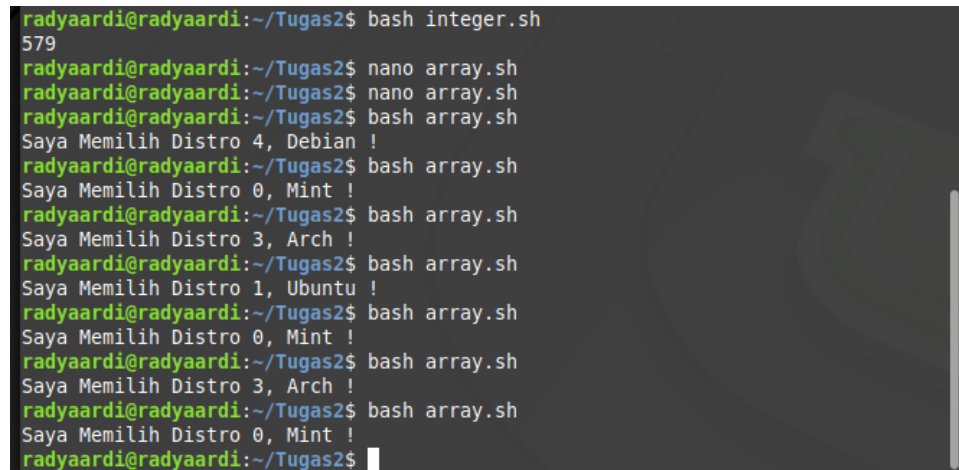
# deklarasi array
distroLinux=("Mint" "Ubuntu" "Kali" "Arch" "Debian")

#random distro
let pilih=$RANDOM%5

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

```

- Setelah kita Menyusun script pada nano, kemudian kita dapat memunculkan output dari script yang telah kita buat tadi melalui “bash array.sh”
- Di sini dapat terlihat bahwa ketika memanggil “bash array.sh” secara berulang-ulang, output yang dimunculkan pun juga berbeda.



```

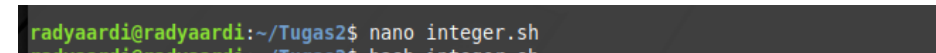
radyaardi@radyaardi:~/Tugas2$ bash integer.sh
579
radyaardi@radyaardi:~/Tugas2$ nano array.sh
radyaardi@radyaardi:~/Tugas2$ nano array.sh
radyaardi@radyaardi:~/Tugas2$ bash array.sh
Saya Memilih Distro 4, Debian !
radyaardi@radyaardi:~/Tugas2$ bash array.sh
Saya Memilih Distro 0, Mint !
radyaardi@radyaardi:~/Tugas2$ bash array.sh
Saya Memilih Distro 3, Arch !
radyaardi@radyaardi:~/Tugas2$ bash array.sh
Saya Memilih Distro 1, Ubuntu !
radyaardi@radyaardi:~/Tugas2$ bash array.sh
Saya Memilih Distro 0, Mint !
radyaardi@radyaardi:~/Tugas2$ bash array.sh
Saya Memilih Distro 3, Arch !
radyaardi@radyaardi:~/Tugas2$ bash array.sh
Saya Memilih Distro 0, Mint !
radyaardi@radyaardi:~/Tugas2$

```

### 3. Integer

Langkah dalam membuat Integer adalah

- Membuat nano nama file.sh (nano integer.sh)

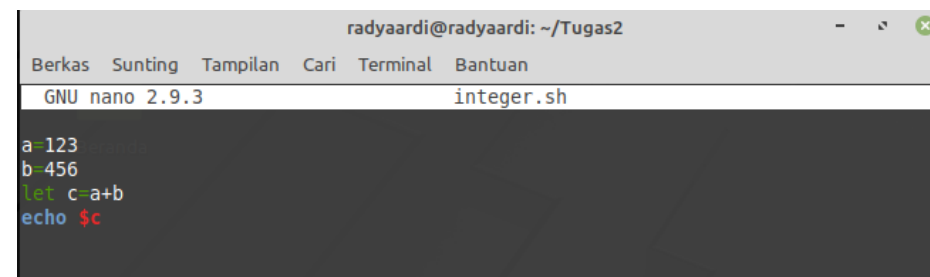


```

radyaardi@radyaardi:~/Tugas2$ nano integer.sh

```

- Integer berfungsi untuk menampilkan atau mempresentasikan hasil operasi bilangan bulat pada operasi matematika.



```

radyaardi@radyaardi: ~/Tugas2
Berkas  Sunting  Tampilan  Cari  Terminal  Bantuan
GNU nano 2.9.3 integer.sh

a=123
b=456
let c=a+b
echo $c

```

- Setelah kita Menyusun script pada nano, kemudian kita dapat memunculkan output dari script yang telah kita buat tadi melalui “bash array.sh”

- Di sini dapat terlihat jika hasil dari  $a+b=c \Rightarrow 123+456=579$

```
radyaardi@radyaardi:~/Tugas2$ nano integer.sh
radyaardi@radyaardi:~/Tugas2$ bash integer.sh
579
```

## 4. Operasi Matematika

Langkah dalam membuat Operasi Matematika adalah

- Membuat nano nama file.sh (nano op\_mtk.sh)
- Dalam melakukan operasi perhitungan matematika, kita menggunakan `let`. selain `let`, kita juga dapat menggunakan `expr` atau `awk` untuk menggunakan perintah eksternal, dan `$((ekspresi))` untuk menggunakan perintah substitusi.



```
radyaardi@radyaardi: ~/Tugas2
Berkas  Sunting  Tampilan  Cari  Terminal  Bantuan
GNU nano 2.9.3      op_mtk.sh      Modified

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"
```

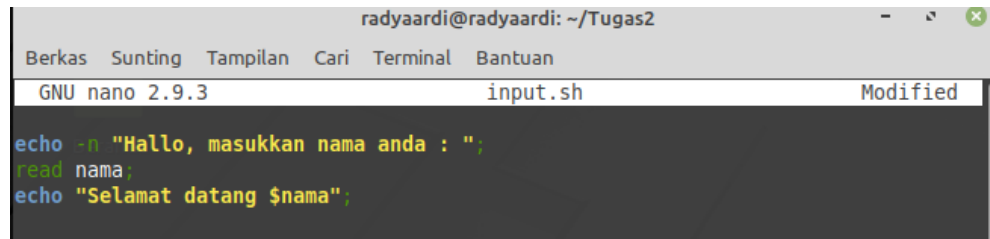
- Setelah kita Menyusun script pada nano, kemudian kita dapat memunculkan output dari script yang telah kita buat tadi melalui “`bash op_mtk.sh`”. dan dapat dilihat operasi perhitungan matematika yang telah diselesaikan.

```
radyaardi@radyaardi:~/Tugas2$ nano op_mtk.sh
radyaardi@radyaardi:~/Tugas2$ bash op_mtk.sh
a + b = 22
a - b = 8
a * b = 105
a / b = 2
a % b = 1
a = 15
b = 15
```

## 5. Input

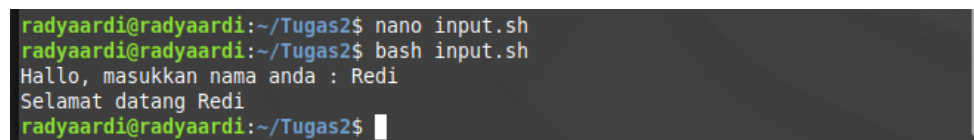
Langkah dalam membuat Input pada shell adalah

- Membuat nano nama file (`nano input.sh`)
- Dalam menggunakan fungsi input pada shell, untuk membaca inputan word dari user kita dapat menggunakan “read” dengan format penulisan “`read nama_var`”
- Untuk menampilkan output yang telah kita inputkan, maka kita menggunakan `echo`.

A screenshot of a terminal window with the nano text editor open. The title bar shows 'radyaardi@radyaardi: ~/Tugas2'. The menu bar includes 'Berkas', 'Sunting', 'Tampilan', 'Cari', 'Terminal', and 'Bantuan'. The status bar shows 'GNU nano 2.9.3' and 'input.sh Modified'. The editor content contains the following code:

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

- Setelah kita Menyusun script pada nano, kemudian kita dapat memunculkan output dari script yang telah kita buat tadi melalui “`bash input.sh`”. dan dapat dilihat bahwa kita dapat menginputkan nama sesuai yang kita inginkan pada script kita.

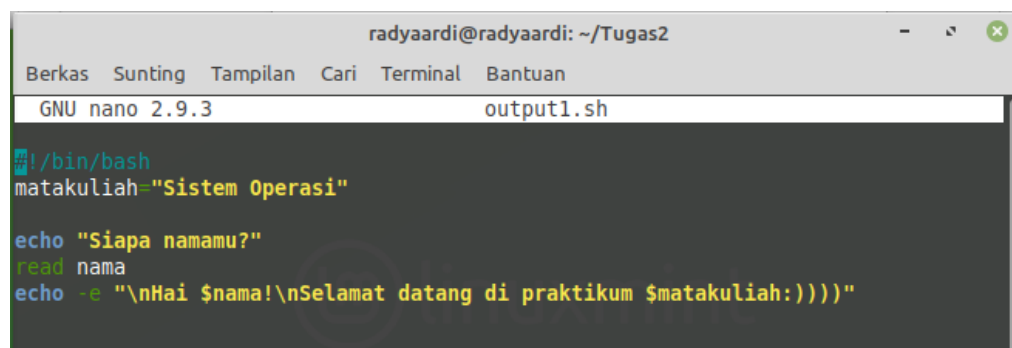
A screenshot of a terminal window showing the execution of the script. The prompt is 'radyaardi@radyaardi:~/Tugas2\$'. The user enters 'nano input.sh' and then 'bash input.sh'. The script outputs 'Hallo, masukkan nama anda : Redi' and 'Selamat datang Redi'. The prompt returns to 'radyaardi@radyaardi:~/Tugas2\$' with a cursor.

```
radyaardi@radyaardi:~/Tugas2$ nano input.sh
radyaardi@radyaardi:~/Tugas2$ bash input.sh
Hallo, masukkan nama anda : Redi
Selamat datang Redi
radyaardi@radyaardi:~/Tugas2$
```

## 6. Output 1

Langkah dalam memuat Output pada shell adalah

- Membuat nano nama file.sh (`nano Output1.sh`)
- Pada pengoperasian output kali ini menggunakan `echo`. Dimana `echo` dapat menampilkan output dari syntax yang kita masukkan. Macam output `echo` ada 2. Yaitu, menampilkan teks biasa menggunakan `echo "teks"` dan menampilkan isi dari sebuah variabel menggunakan `echo $nama_var`

A screenshot of a terminal window with the nano text editor open. The title bar shows 'radyaardi@radyaardi: ~/Tugas2'. The menu bar includes 'Berkas', 'Sunting', 'Tampilan', 'Cari', 'Terminal', and 'Bantuan'. The status bar shows 'GNU nano 2.9.3' and 'output1.sh'. The editor content contains the following code:

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

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

- Setelah kita Menyusun script pada nano, kemudian kita dapat memunculkan output dari script yang telah kita buat tadi melalui “`bash output1.sh`”. di sini kita menggunakan `echo` yang menampilkan teks biasa.

```

radyaardi@radyaardi:~/Tugas2$ nano output1.sh
radyaardi@radyaardi:~/Tugas2$ bash output1.sh
Siapa namamu?
Redi

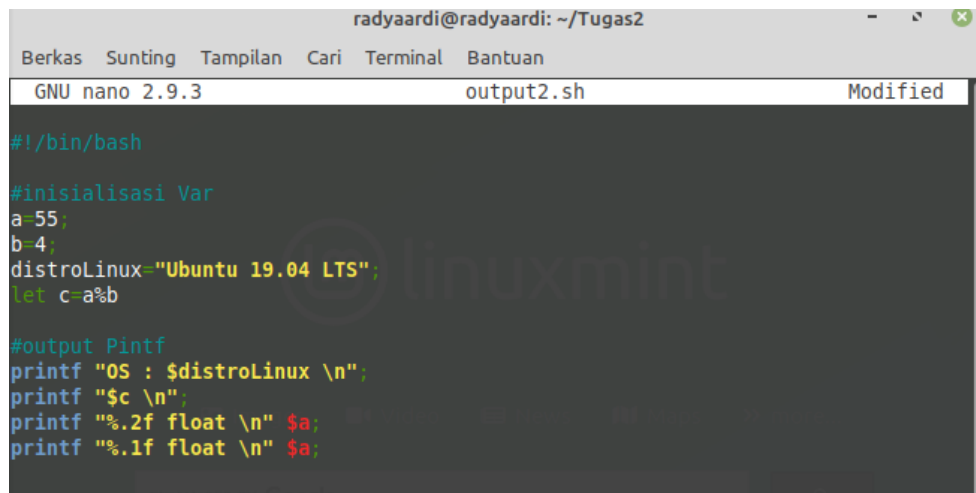
Hai Redi!
Selamat datang di praktikum Sistem Operasi:)))
radyaardi@radyaardi:~/Tugas2$

```

## 7. Output 2

Langkah dalam membuat Output pada shell adalah

- Membuat nano nama file.sh (nano Output2.sh)
- Pada Output yang kedua ini kita bisa juga menggunakan printf seperti pada bahasa pemrograman C. fungsi dari Printf sendiri adalah sama seperti echo, yaitu menampilkan output ke layar komputer.



```

radyaardi@radyaardi: ~/Tugas2
Berkas  Sunting  Tampilan  Cari  Terminal  Bantuan
GNU nano 2.9.3      output2.sh      Modified

#!/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;

```

- Setelah kita Menyusun script pada nano, kemudian kita dapat memunculkan output dari script yang telah kita buat tadi melalui “bash output2.sh”

```

radyaardi@radyaardi:~/Tugas2$ nano output2.sh
radyaardi@radyaardi:~/Tugas2$ bash output2.sh
OS : Ubuntu 19.04 LTS
3
55,00 float
55,0 float
radyaardi@radyaardi:~/Tugas2$

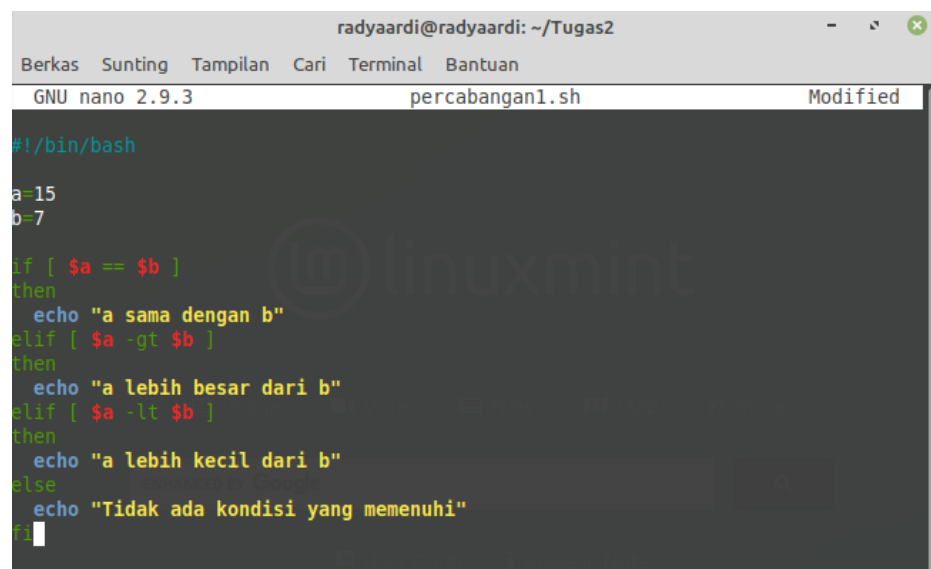
```

## 8. Percabangan1

Langkah dalam membuat Percabangan pada shell adalah

- Membuat nano nama file.sh (`nano percabangan1.sh`)
- Pada percabangan If... Else ini kita dapat menggunakan beberapa fungsi yang digunakan. Yaitu
  - `-eq (=)` untuk mengetahui apakah nilai kedua operasi itu sama.
  - `-ne (!=)` untuk memeriksa apakah nilai kedua operasi itu tidak sama
  - `-gt (>)` untuk memeriksa apakah nilai yang sebelah kiri lebih besar daripada nilai yang sebelah kanan.
  - `-lt (<)` untuk memeriksa nilai sebelah kiri lebih kecil daripada nilai sebelah kanan.
  - `-ge (>=)` untuk memeriksa apakah nilai sebelah kiri lebih besar atau sama dengan nilai sebelah kanan
  - `-le (<=)` untuk memeriksa apakah nilai sebelah kiri lebih kecil atau sama dengan nilai sebelah kanan.
  - `(=)` untuk mengetahui apakah kedua nilai inputan sama atau tidak.

Fungsi fi adalah untuk mengakhiri atau menutup suatu operasi percabangan.



```
radyaardi@radyaardi: ~/Tugas2
Berkas  Sunting  Tampilan  Cari  Terminal  Bantuan
GNU nano 2.9.3      percabangan1.sh      Modified

#!/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
```

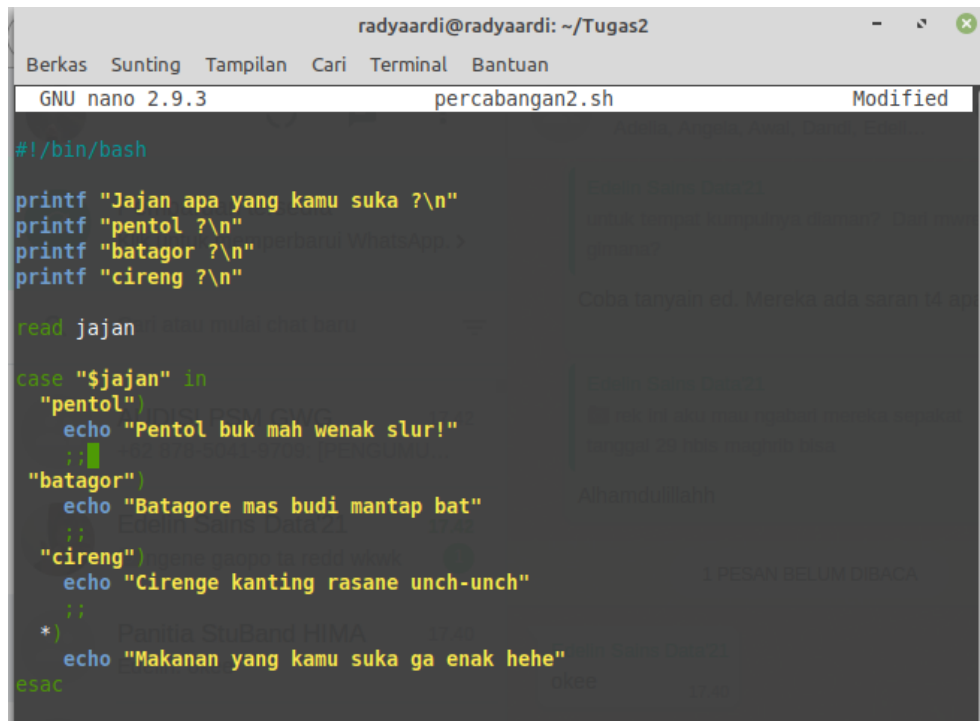
- Setelah kita Menyusun script pada nano, kemudian kita dapat memunculkan output dari script yang telah kita buat tadi melalui “`bash percabangan1.sh`”

```
radyaardi@radyaardi:~/Tugas2$ nano percabangan1.sh
radyaardi@radyaardi:~/Tugas2$ bash percabangan1.sh
a lebih besar dari b
radyaardi@radyaardi:~/Tugas2$
```

## 9. Percabangan2

Langkah dalam membuat Percabangan pada shell adalah

- Membuat nano nama file.sh (nano percabangan2.sh)
- Pada percabangan kedua ini kita menggunakan statement Case...Esac. Perbedaan dengan statement If... else adalah pada statement case... esac dinilai lebih efisien dalam mengeluarkan hasil daripada menggunakan if...else.



```
radyaardi@radyaardi: ~/Tugas2
Berkas  Sunting  Tampilan  Cari  Terminal  Bantuan
GNU nano 2.9.3      percabangan2.sh      Modified

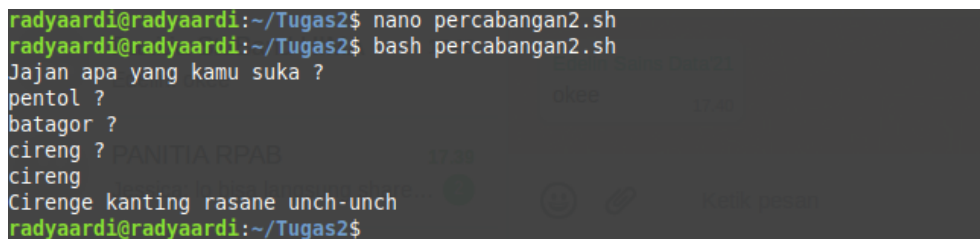
#!/bin/bash

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 kancing rasane unch-unch"
        ;;
    *)
        echo "Makanan yang kamu suka ga enak hehe"
esac
```

- Setelah kita Menyusun script pada nano, kemudian kita dapat memunculkan output dari script yang telah kita buat tadi melalui “bash percabangan2.sh”



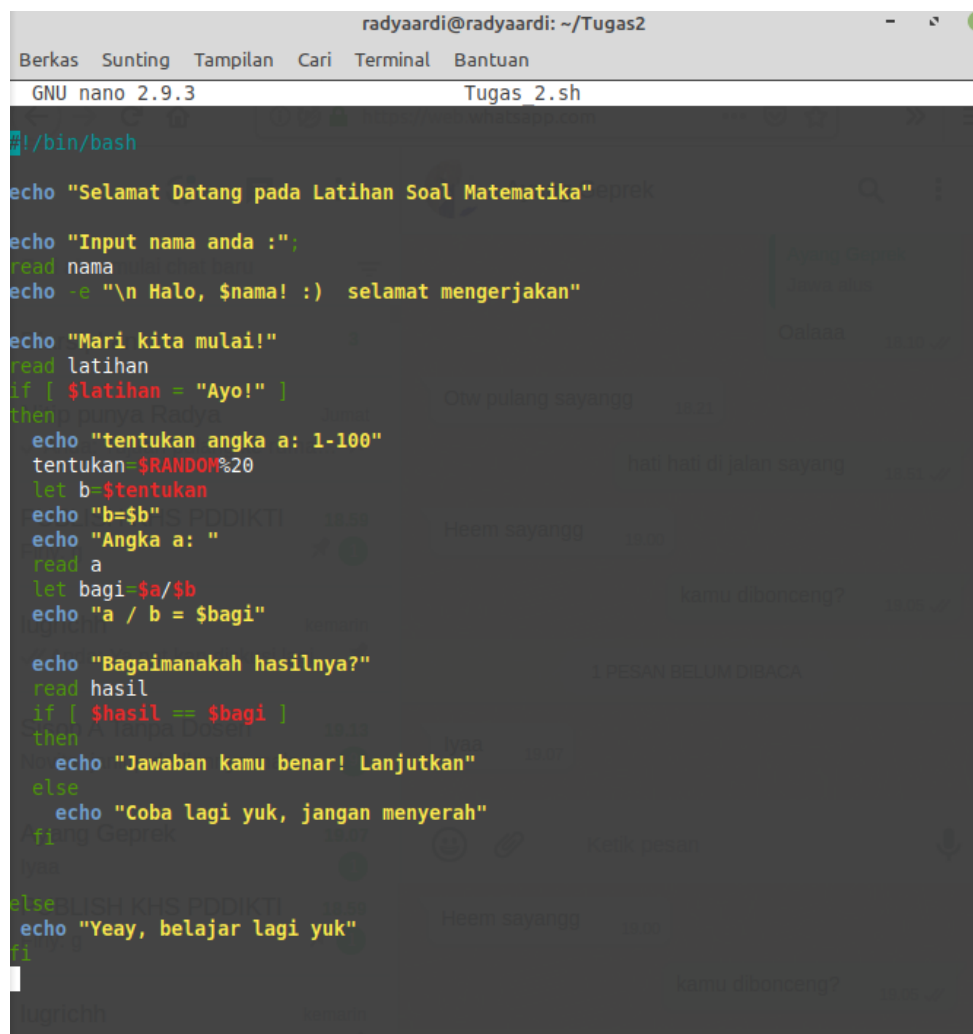
```
radyaardi@radyaardi:~/Tugas2$ nano percabangan2.sh
radyaardi@radyaardi:~/Tugas2$ bash percabangan2.sh
Jajan apa yang kamu suka ?
pentol ?
batagor ?
cireng ?
Cireng
Cireng kancing rasane unch-unch
radyaardi@radyaardi:~/Tugas2$
```

# SOAL LATIHAN

## 10. Operasi Percabangan Aritmatika

Langkah dalam membuat operasi percabangan aritmetika adalah

- Membuat nano nama file (nano Tugas\_2.sh)
- Pada tugas membuat program percabangan sederhana ini saya menggunakan konsep pemrograman input, output, operasi aritmetika, dan percabangan if...else.
- Pertama menggunakan program input untuk memasukkan nama orang yang akan mengerjakan soal Latihan matematika tersebut. Kemudian pada pembuatan angka nya saya menggunakan random untuk menentukan variabel b pada operasi pembagian yang saya terapkan. Kemudian, jika dia menjawab benar akan muncul kalimat "Jawaban kamu benar! Lanjutkan". Dan jika salah, maka akan muncul kalimat "Coba lagi yuk, jangan menyerah."



```
radyaardi@radyaardi: ~/Tugas2
Berkas  Sunting  Tampilan  Cari  Terminal  Bantuan
GNU nano 2.9.3  Tugas_2.sh

#!/bin/bash

echo "Selamat Datang pada Latihan Soal Matematika"
echo "Input nama anda :";
read nama
echo -e "\n Halo, $nama! :) selamat mengerjakan"

echo "Mari kita mulai!"
read latihan
if [ $latihan = "Ayo!" ]
then
    echo "tentukan angka a: 1-100"
    tentukan=$((RANDOM%20))
    let b=$tentukan
    echo "b=$b"
    echo "Angka a: "
    read a
    let bagi=$((a/$b))
    echo "a / b = $bagi"

    echo "Bagaimanakah hasilnya?"
    read hasil
    if [ $hasil == $bagi ]
    then
        echo "Jawaban kamu benar! Lanjutkan"
    else
        echo "Coba lagi yuk, jangan menyerah"
    fi
else
    echo "Yeay, belajar lagi yuk"
fi
```

- Setelah kita Menyusun script pada nano, kemudian kita dapat memunculkan output dari script yang telah kita buat tadi melalui "bash Tugas\_2.sh". di sini dapat terlihat hasil dari program operasi percabangan sederhana aritmetika yang telah saya buat.



```
radyaardi@radyaardi:~/Tugas2$ nano Tugas_2.sh
radyaardi@radyaardi:~/Tugas2$ bash Tugas_2.sh
Selamat Datang pada Latihan Soal Matematika
Input nama anda :DDIKTI
Radya

    Halo, Radya! :) selamat mengerjakan
Mari kita mulai!
Ayo!
tentukan angka a: 1-100
b=6
Angka a:
48
a / b = 8
Bagaimanakah hasilnya?
8
Jawaban kamu benar! Lanjutkan
radyaardi@radyaardi:~/Tugas2$
```