

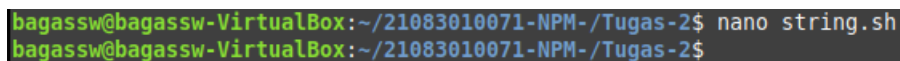
TUGAS 2

LAPORAN TUGAS

A. Tutorial

1. Penggunaan String

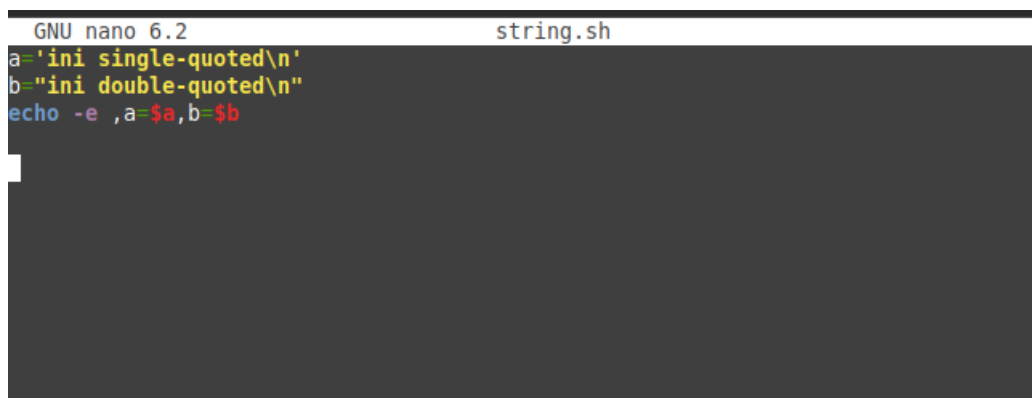
Pertama kita menggunakan perintah nano untuk membuat file string.sh seperti Gambar 1.1



```
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$ nano string.sh
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$
```

Gambar 1.1

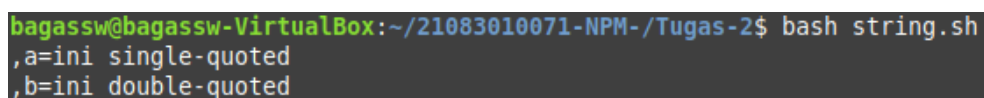
Mengisi file dengan script string yang diapit tanda kutip ganda (single quoted dan double quoted) akan ditampilkan pada layar anda, echo adalah statement (perintah) built-in bash yang berfungsi menampilkan informasi ke standard output yang defaultnya adalah layer. Percobaan ini seperti script pada Gambar 1.2



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

Gambar 1.2

Lalu kita menjalankan file string.sh dengan cara bash 'nama_file' seperti yang ada pada Gambar 1.3



```
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$ bash string.sh
,a=ini single-quoted
,b=ini double-quoted
```

Gambar 1.3

2. Penggunaan Array

Pertama kita menggunakan perintah nano untuk membuat file array.sh seperti Gambar 2.1

```
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$ nano array.sh
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$
```

Gambar 2.1

Percobaan saya mengisi file menggunakan script deklarasi array dengan array tipe string dan memberikan random distro lalu tinggal mencetak menggunakan echo. Percobaan ini seperti script pada Gambar 2.2

```
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]} !"
```

Gambar 2.2

Lalu kita menjalankan file array.sh dengan cara bash 'nama_file' seperti yang ada pada Gambar 2.3

```
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$ nano array.sh
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$ bash array.sh
Saya Memilih Distro 3, Arch !
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$
```

Gambar 2.3

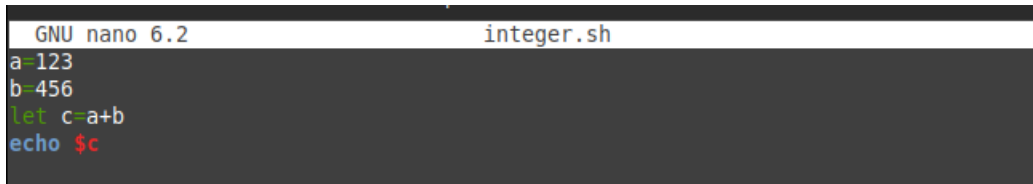
3. Penggunaan Integer

Pertama kita menggunakan perintah nano untuk membuat file integer.sh seperti yang ada pada Gambar 3.1

```
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$ nano integer.sh
```

Gambar 3.1

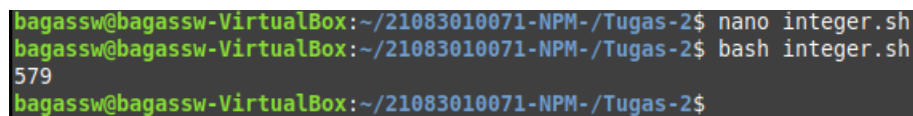
Percobaan saya mengisi file dengan menggunakan deklarasi variable $a = \text{integer } 123$, $b = \text{integer } 456$, dan $c = a + b$ lalu tinggal mencetak menggunakan echo. Percobaan ini seperti script pada Gambar 3.2



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

Gambar 3.2

Lalu kita menjalankan file integer.sh dengan cara bash 'nama_file' seperti yang ada pada Gambar 3.3

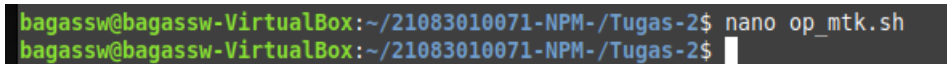


```
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$ nano integer.sh
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$ bash integer.sh
579
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$
```

Gambar 3.3

4. Penggunaan Operasi matematika

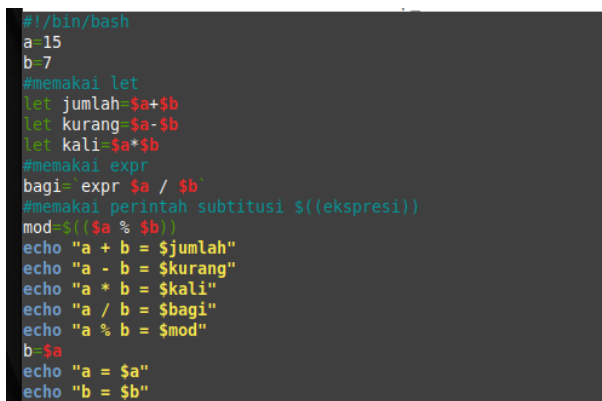
Pertama kita menggunakan perintah nano untuk membuat file op_mtk.sh seperti yang ada pada Gambar 4.1



```
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$ nano op_mtk.sh
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$
```

Gambar 4.1

Percobaan saya mengisi file dengan menggunakan deklarasi variable $a = 15$, $b = 7$, dan memndekarasikan op matematika dengan let seperti jumlah, kurang, kali lalu tinggal mencetak menggunakan echo. Percobaan ini seperti script pada Gambar 4.2



```
#!/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"
```

Gambar 4.2

Lalu kita menjalankan file op_mtk.sh dengan cara bash 'nama_file' seperti yang ada pada Gambar 4.3

```
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$ bash op_mtk.sh
a + b = 22
a - b = 8
a * b = 105
a / b = 2
a % b = 1
a = 15
b = 15
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$
```

Gambar 4.3

5. Penggunaan Input

Pertama kita menggunakan perintah nano untuk membuat file input.sh seperti yang ada pada Gambar 5.1

```
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$ nano input.sh
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$
```

Gambar 5.1

Percobaan saya mengisi file menggunakan echo -n masukkan nama anda sebagai input dan read nama untuk menyimpan lalu tinggal mencetak menggunakan echo. Percobaan ini seperti script pada Gambar 5.2

```
GNU nano 6.2      input.sh
#!/bin/bash
clear
echo -n "Hallo, masukkan nama anda : ";
read nama;
echo "Selamat datang $nama";
```

Gambar 5.2

Lalu kita menjalankan file input.sh dengan cara bash 'nama_file' seperti yang ada pada Gambar 5.3

```
Hallo, masukkan nama anda : bagas
Selamat datang bagas
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$ bash input.sh
```

Gambar 5.3

6. Penggunaan Penampilan ouput 1

Pertama kita menggunakan perintah nano untuk membuat file output1.sh seperti yang ada pada Gambar 6.1

```
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$ nano output1.sh
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$
```

Gambar 6.1

Percobaan saya mengisi file menggunakan script echo siapa namamu? Dan read nama untuk menyimpan lalu tinggal mencetak menggunakan echo -e untuk mencetak variable nama dan matakuliah. Percobaan ini seperti script pada Gambar 6.2

```
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:)))))"
```

Gambar 6.2

Lalu kita menjalankan file output1.sh dengan cara bash 'nama_file' seperti yang ada pada Gambar 6.3

```
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$ bash output1.sh
Siapa namamu?
bagas

Hai bagas!
Selamat datang di praktikum Sistem Operasi:)))))
```

Gambar 6.3


7. Penggunaan Penampilan output 2

Pertama kita menggunakan perintah nano untuk membuat file output2.sh seperti yang ada pada Gambar 7.1

```
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$ nano output2.sh
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$
```

Gambar 7.1

Percobaan saya mengisi file menggunakan script deklarasi variable a,b dan distrolinux let c = a%b lalu print atau cetak output. Percobaan ini seperti script pada Gambar 7.2



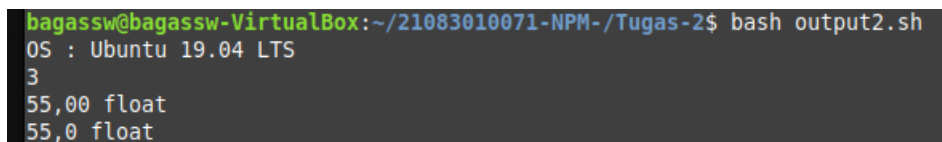
```

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;

```

Gambar 7.2

Lalu kita menjalankan file output2.sh dengan cara bash 'nama_file' seperti yang ada pada Gambar 7.3



```

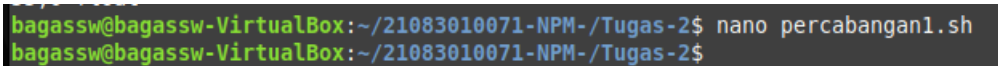
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$ bash output2.sh
OS : Ubuntu 19.04 LTS
3
55,00 float
55,0 float

```

Gambar 7.3

8. Penggunaan Percabangan 1

Pertama kita menggunakan perintah nano untuk membuat file percabangan1.sh seperti yang ada pada Gambar 8.1



```

bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$ nano percabangan1.sh
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$

```

Gambar 8.1

Percobaan saya mengisi file menggunakan script deklarasi variable a,b dan menyesuaikan percabangan dan aturan menggunakan if dan elif lalu echo untuk cetak output.Percobaan ini seperti script pada Gambar 8.2



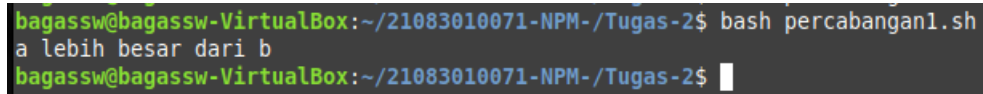
```

GNU nano 6.2                                percabangan1.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

```

Gambar 8.2

Lalu kita menjalankan file percabangan1.sh dengan cara bash 'nama_file' seperti yang ada pada Gambar 8.3

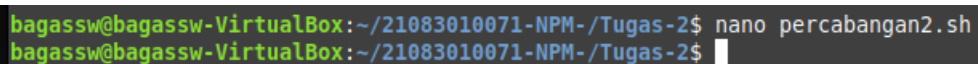


```
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$ bash percabangan1.sh
a lebih besar dari b
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$
```

Gambar 8.3

9. Penggunaan Percabangan 2

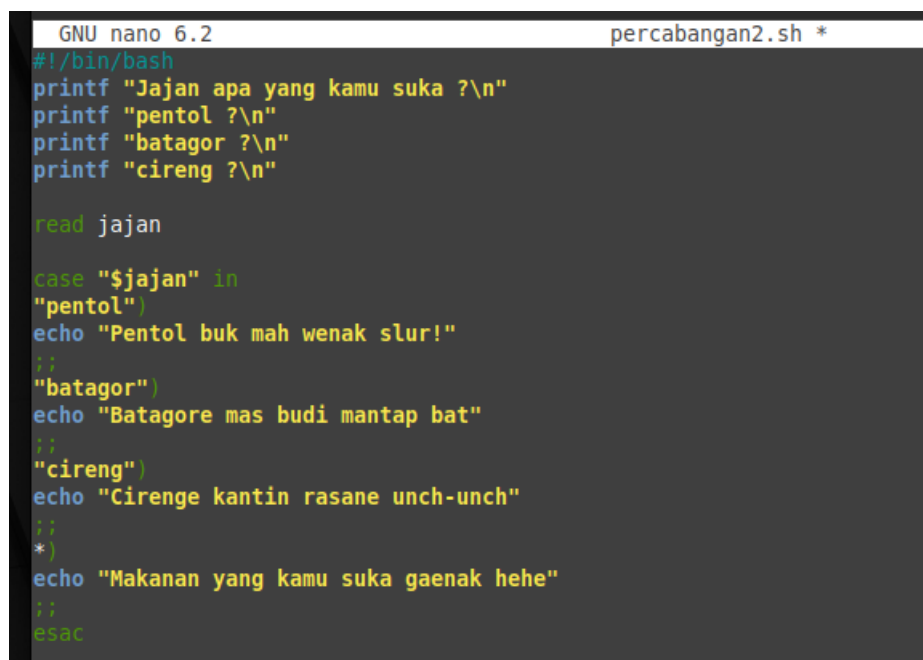
Pertama kita menggunakan perintah nano untuk membuat file percabangan2.sh seperti yang ada pada Gambar 9.1



```
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$ nano percabangan2.sh
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$
```

Gambar 9.1

Percobaan saya mengisi file menggunakan script deklarasi variable jajan dan menyesuaikan percabangan dan aturan menggunakan if dan elif lalu echo untuk cetak output. Percobaan ini seperti script pada Gambar 9.2



```
GNU nano 6.2                                percabangan2.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
```

Gambar 9.2

Lalu kita menjalankan file percabangan2.sh dengan cara bash 'nama_file' seperti yang ada pada Gambar 9.3

```

bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$ bash percabangan2.sh
Jajan apa yang kamu suka ?
pentol ?
batagor ?
cireng ?
pentol
Pentol buk mah wenak slur!
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$

```

Gambar 9.3

B. Latihan Soal

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

Pertama kita menggunakan perintah nano untuk membuat file Tugas-2.sh seperti yang ada pada Gambar 11.1

```

bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$ nano Tugas-2.sh
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$ nano Tugas-2.sh

```

Gambar 11.1

Latihan soal saya mengisi file menggunakan script deklarasi variable a,b,c,e dan menyesuaikan percabangan dan aturan menggunakan if dan elif lalu echo untuk cetak output. Percobaan ini seperti script pada Gambar 11.2

```

GNU nano 6.2          Tugas-2.sh
#!/bin/bash
a=15
b=7
c=6
e=2
let ab=$a*$b
let ce=$c*$e
if [ $ab == $ce ]
then
echo "ab sama dengan ce"
elif [ $ab -gt $ce ]
then
echo "ab lebih besar dari ce"
elif [ $ab -lt $ce ]
then
echo "ab lebih kecil dari ce"
else
echo "Tidak ada kondisi yang memenuhi"
fi

```

Gambar 11.2

Lalu kita menjalankan Tugas-2.sh dengan cara bash 'nama_file' seperti yang ada pada Gambar 11.3

```

bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$ bash Tugas-2.sh
ab lebih besar dari ce
bagassw@bagassw-VirtualBox:~/21083010071-NPM-/Tugas-2$ bash Tugas-2.sh
ab lebih besar dari ce

```

Gambar 11.3

