

REPORT TUGAS PERTEMUAN 2 SISTEM OPERASI

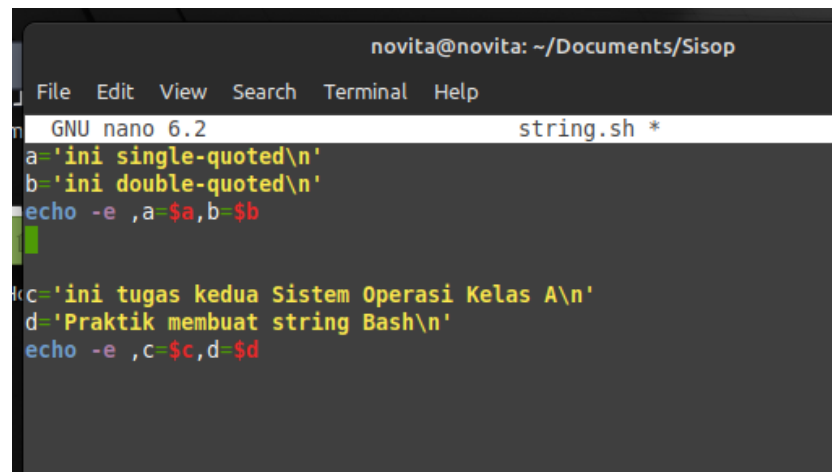
Nama : Novita Anggraini

NPM : 21083010104

Kelas : Sistem Operasi A

1. Pembuatan string.sh

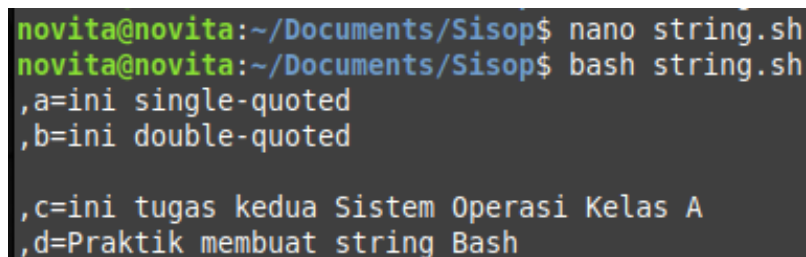
- nano string.sh



```
novita@novita: ~/Documents/Sisop
File Edit View Search Terminal Help
GNU nano 6.2 string.sh *
a='ini single-quoted\n'
b='ini double-quoted\n'
echo -e ,a=$a,b=$b

c='ini tugas kedua Sistem Operasi Kelas A\n'
d='Praktik membuat string Bash\n'
echo -e ,c=$c,d=$d
```

- contoh output



```
novita@novita:~/Documents/Sisop$ nano string.sh
novita@novita:~/Documents/Sisop$ bash string.sh
,a=ini single-quoted
,b=ini double-quoted

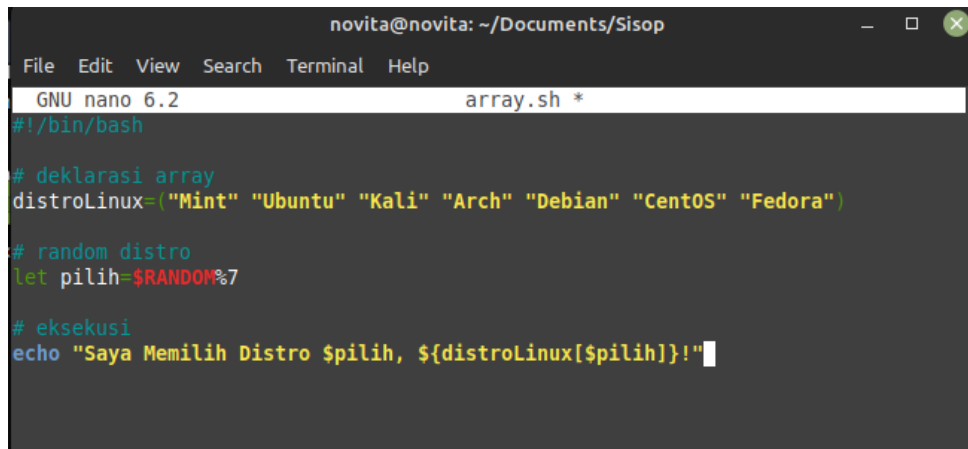
,c=ini tugas kedua Sistem Operasi Kelas A
,d=Praktik membuat string Bash
```

echo dengan sendirinya menampilkan sebaris teks, yang mana akan mengambil apa pun dalam "..." dua tanda kutip secara harfiah, dan hanya mencetak apa adanya. Namun dengan memakai echo -e, maka kita bisa memakai echo untuk mengaktifkan interpretasi *backslash escapes* (\n, \t).

String sendiri merupakan salah satu tipe data yang berisi sequence atau untaian dari beberapa karakter. Tipe data string adalah tipe data yang digunakan untuk menyimpan barisan karakter.

2. Pembuatan Array

- nano array.sh



```
novita@novita: ~/Documents/Sisop
File Edit View Search Terminal Help
GNU nano 6.2 array.sh *
#!/bin/bash

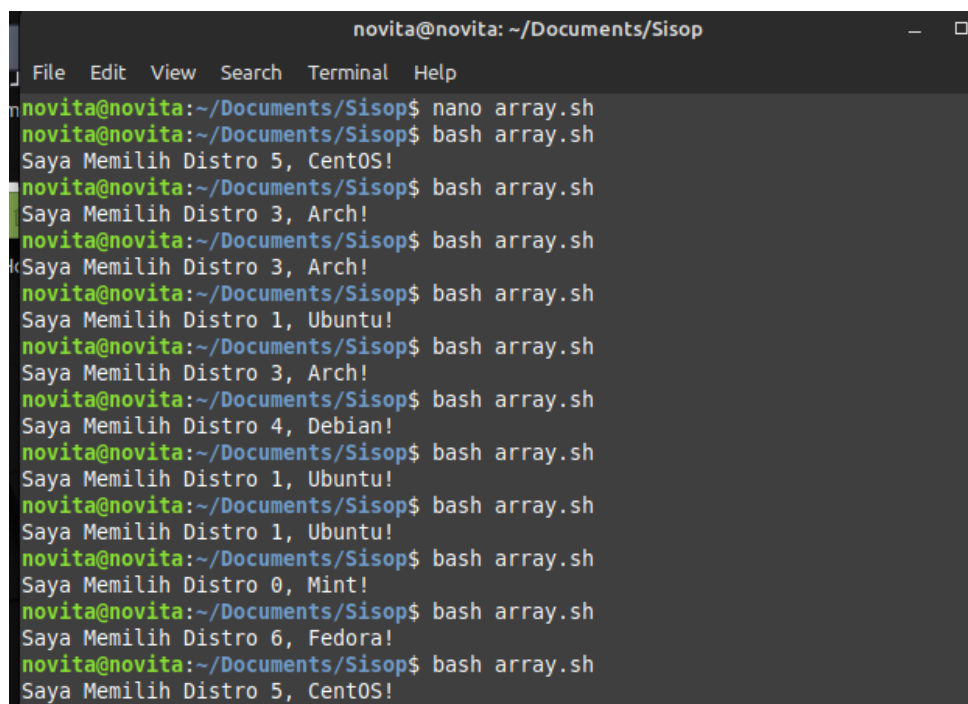
# deklarasi array
distroLinux=("Mint" "Ubuntu" "Kali" "Arch" "Debian" "CentOS" "Fedora")

# random distro
let pilih=$((RANDOM%7))

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

distroLinux adalah variable yang digunakan untuk menyimpan informasi berupa array atau kumpulan data. Lalu random digunakan untuk memilih acak sesuai range angka yang dipilih. Output dari code di atas adalah sebagai berikut:

- Output array.sh

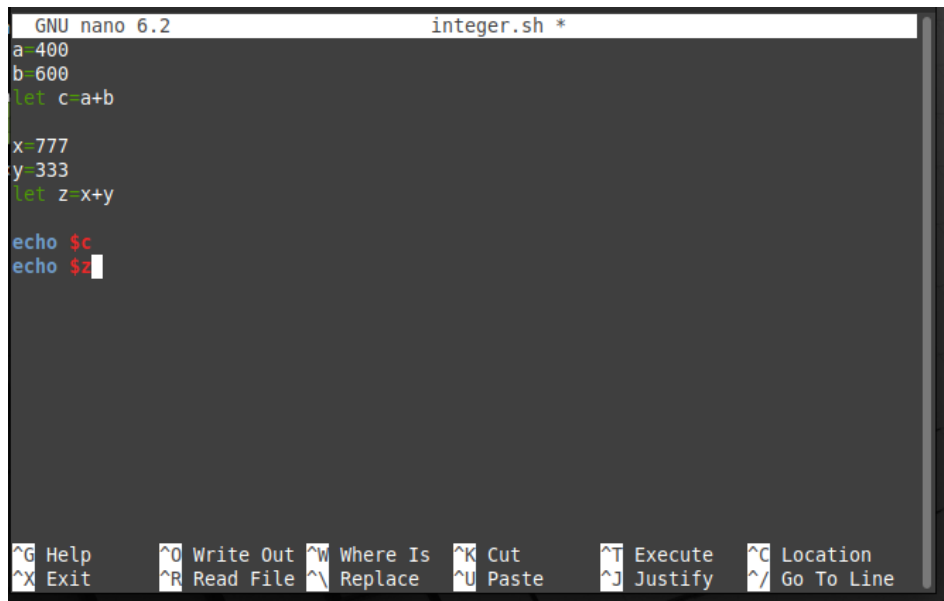


```
novita@novita: ~/Documents/Sisop
File Edit View Search Terminal Help
novita@novita:~/Documents/Sisop$ nano array.sh
novita@novita:~/Documents/Sisop$ bash array.sh
Saya Memilih Distro 5, CentOS!
novita@novita:~/Documents/Sisop$ bash array.sh
Saya Memilih Distro 3, Arch!
novita@novita:~/Documents/Sisop$ bash array.sh
Saya Memilih Distro 3, Arch!
novita@novita:~/Documents/Sisop$ bash array.sh
Saya Memilih Distro 1, Ubuntu!
novita@novita:~/Documents/Sisop$ bash array.sh
Saya Memilih Distro 3, Arch!
novita@novita:~/Documents/Sisop$ bash array.sh
Saya Memilih Distro 4, Debian!
novita@novita:~/Documents/Sisop$ bash array.sh
Saya Memilih Distro 1, Ubuntu!
novita@novita:~/Documents/Sisop$ bash array.sh
Saya Memilih Distro 1, Ubuntu!
novita@novita:~/Documents/Sisop$ bash array.sh
Saya Memilih Distro 0, Mint!
novita@novita:~/Documents/Sisop$ bash array.sh
Saya Memilih Distro 6, Fedora!
novita@novita:~/Documents/Sisop$ bash array.sh
Saya Memilih Distro 5, CentOS!
```

Dapat dilihat bahwa setiap run, system dapat menampilkan output yang berbeda-beda.

3. Pembuatan Integer

- nano integer.sh



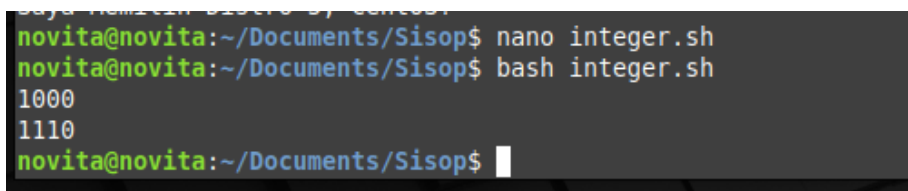
```
GNU nano 6.2 integer.sh *
a=400
b=600
let c=a+b

x=777
y=333
let z=x+y

echo $c
echo $z
```

Integer merupakan tipe sebuah data. Dalam hal ini adalah bilangan bulat. Dengan mengetikkan angka tanpa koma, maka angka tersebut disebut tipe data integer yang bisa dioperasikan matematika seperti tambah, kurang, kali, bagi, atau operasi matematika lainnya.

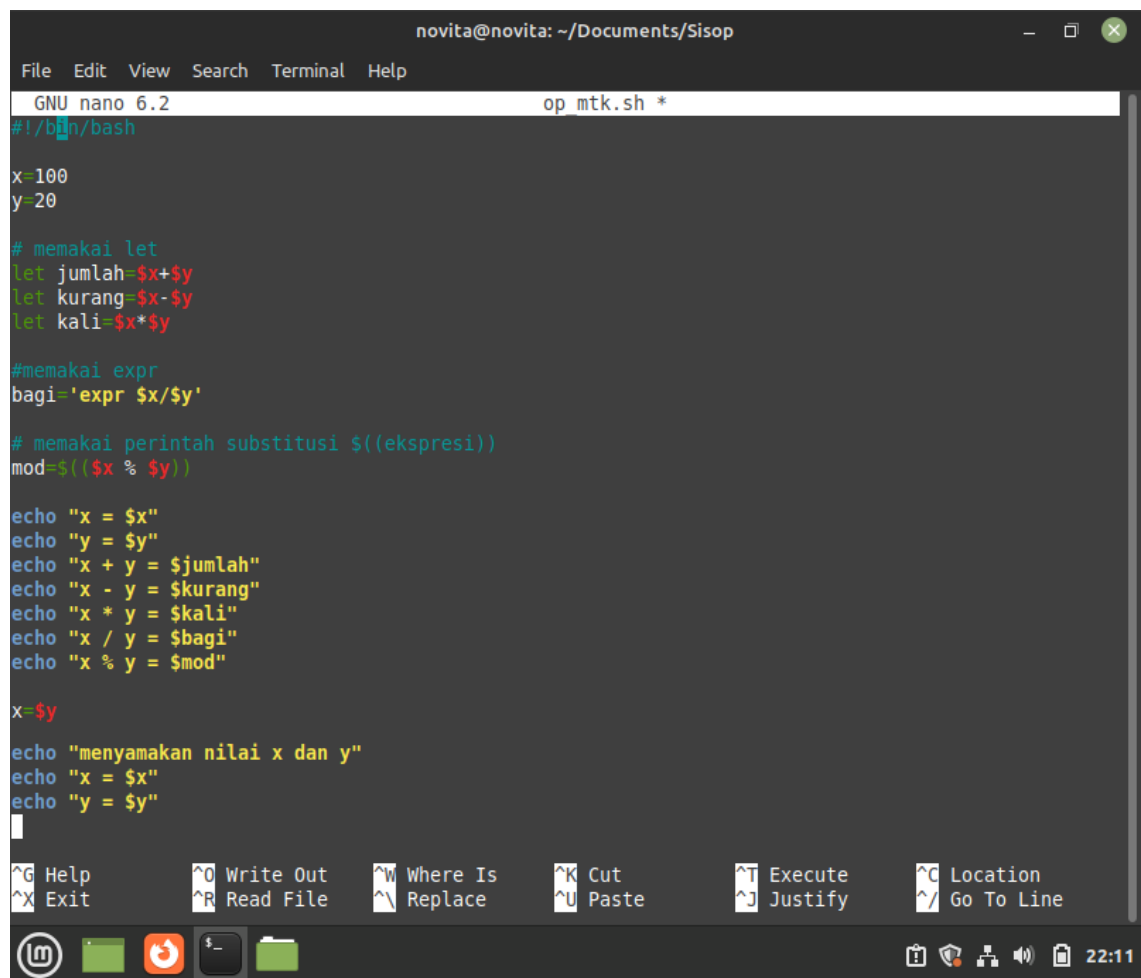
- Output integer.sh



```
novita@novita:~/Documents/Sisop$ nano integer.sh
novita@novita:~/Documents/Sisop$ bash integer.sh
1000
1110
novita@novita:~/Documents/Sisop$
```

4. Pembuatan Operasi Matematika

- nano op_mtk.sh



```
novita@novita: ~/Documents/Sisop
GNU nano 6.2 op_mtk.sh *
#!/bin/bash

x=100
y=20

# memakai let
let jumlah=$((x+y))
let kurang=$((x-y))
let kali=$((x*y))

#memakai expr
bagi=$((x/y))

# memakai perintah substitusi $((ekspresi))
mod=$((x % y))

echo "x = $x"
echo "y = $y"
echo "x + y = $jumlah"
echo "x - y = $kurang"
echo "x * y = $kali"
echo "x / y = $bagi"
echo "x % y = $mod"

x=$y
y=$x

echo "menyamakan nilai x dan y"
echo "x = $x"
echo "y = $y"
```

- Output op_mtk.sh

```
novita@novita:~/Documents/Sisop$ nano op_mtk.sh
novita@novita:~/Documents/Sisop$ bash op_mtk.sh
x = 100
y = 20
x + y = 120
x - y = 80
x * y = 2000
x / y = expr $x/$y
x % y = 0
menyamakan nilai x dan y
x = 20
y = 20
```

Operasi matematika berupa tambah, kurang, kali, bagi, pangkat, dan lain sebagainya dapat digunakan untuk mengoperasikan angka. Kita tidak perlu membuat kode baru untuk bisa menggunakan operasi matematika, namun kita bisa langsung menggunakan simbol matematika.

5. Pembuatan Input

- nano input.sh

```
novita@novita: ~/Documents/Sisop
File Edit View Search Terminal Help
GNU nano 6.2 input.sh *
#!/bin/bash
clear

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

echo -n "Berapa umur Anda?"
read umur;
echo "$nama saat ini berumur $umur"
```

Fungsi input dalam bash menggunakan syntax *read*. Dengan menggunakan *read*, kita bisa memberikan wadah untuk memasukkan informasi dari *user*.

- Output dari file input.sh

```
novita@novita: ~/Documents/Sisop
File Edit View Search Terminal Help
Hallo, masukkan nama Anda: 
```

Memasukkan input nama yang disimpan dalam variable *nama*, dan memasukkan input umur dalam variable *umur*

```
novita@novita: ~/Documents/Sisop
File Edit View Search Terminal Help
Hallo, masukkan nama Anda: Novita
Selamat datang, Novita
Berapa umur Anda? 20
Novita saat ini berumur 20
novita@novita:~/Documents/Sisop$ 
```

6. Pembuatan Output1

- nano output1.sh

```
novita@novita: ~/Documents/Sisop
File Edit View Search Terminal Help
GNU nano 6.2 output1.sh *
#!/bin/bash
matakuliah="Sistem Operasi"

echo "Siapa namamu? "
read nama
echo "Kelas yang dipilih (A/B/C): "
read kelas

echo -e "\nHalo, $nama!\n Selamat datang di praktikum $matakuliah kelas $kelas! :)"
```

Dengan menggunakan echo dan read, maka kita bisa mengeluarkan output sekaligus memasukkan input. Input yang dimasukkan disimpan ke dalam sebuah variabel.

- Output dari file output1.sh

```
novita@novita:~/Documents/Sisop$ nano output1.sh
novita@novita:~/Documents/Sisop$ bash output1.sh
Siapa namamu?

```

Memasukkan input nama yang disimpan dalam variabel nama.

```
novita@novita:~/Documents/Sisop$ nano output1.sh
novita@novita:~/Documents/Sisop$ bash output1.sh
Siapa namamu?
Novita
Kelas yang dipilih (A/B/C):

```

Memasukkan input kelas yang disimpan dalam variabel kelas.

```
novita@novita:~/Documents/Sisop$ nano output1.sh
novita@novita:~/Documents/Sisop$ bash output1.sh
Siapa namamu?
Novita
Kelas yang dipilih (A/B/C):
A

Halo, Novita!
Selamat datang di praktikum Sistem Operasi kelas A! :)
novita@novita:~/Documents/Sisop$
```

Output keseluruhan dari file output1.sh

7. Pembuatan output2.sh

- nano output2.sh

```
novita@novita: ~/Documents/Sisop
File Edit View Search Terminal Help
GNU nano 6.2 output2.sh *
#!/bin/bash

# inisialisasi var
x=50;
y=4;
distroLinux="Ubuntu 19.04 LTS";
let z=x*y;

# output printf
printf "x = $x\n"
printf "y = $y\n"
printf "OS : $distroLinux \n";
printf "$z \n";
printf "%.2f float \n" $x;
printf "%.1f float \n" $x;
printf "%.2f float \n" $y;
printf "%.1f float \n" $y;
```

- Output dari file output2.sh

```
novita@novita:~/Documents/Sisop$ bash output2.sh
x = 50
y = 4
OS : Ubuntu 19.04 LTS
2
50,00 float
50,0 float
4,00 float
4,0 float
```

Untuk menampilkan output, kita biasa menggunakan echo. Selain echo, kita juga bisa menggunakan syntax printf. Echo hanya bisa mengeksekusi teks apa adanya. Namun dengan printf, system dapat membaca *backslash code* seperti \n atau \t.

8. Pembuatan percabangan2.sh

- nano percabangan1.sh

```
# If - Else

x=102
y=52

echo "x = $x"
echo "y = $y"

if [ $x == $y ]
then
    echo "x sama dengan y"
elif [ $x -gt $y ]
then
    echo "x lebih besar daripada y"
elif [ $x -lt $y ]
then
    echo "x lebih kecil daripada y"
else
    echo "Tidak ada kondisi yang memenuhi"
fi
```

Percabangan dalam *shell scripting* hampir sama dengan Python. *Syntax* yang digunakan adalah *if* untuk membuka percabangan, *then* untuk menampilkan output yang ditampilkan ketika kondisi pertama sudah terpenuhi. *Syntax* digunakan untuk membuat kondisi lain, dan *else* digunakan untuk membuat kondisi terakhir. Dalam kata lain, *else* digunakan untuk kondisi alternatif setelah semua kondisi di atasnya tidak terpenuhi.

- Output file percabangan1.sh

```
novita@novita:~/Documents/Sisop$ nano percabangan2.sh
novita@novita:~/Documents/Sisop$ bash percabangan2.sh
x = 102
y = 52
x lebih besar daripada y
novita@novita:~/Documents/Sisop$
```


9. Pembuatan Percabangan2

- nano percabangan2.sh

```
GNU nano 6.2 percabangan2.sh
#!/bin/bash
printf "Model looseleaf apa yang kamu suka?"
printf "Polos?\n"
printf "Line?\n"
printf "Grid?\n"
printf "Atau berwarna?\n"

read model

case $model in
    "polos")
        echo "model polos cocok untuk coret-core"
        ;;
    "line")
        echo "model line cocok untuk mencatat"
        ;;
    "grid")
        echo "model grid cocok untuk matematika"
        ;;
    "berwarna")
        echo "model berwarna cocok untuk hiasan"
        ;;
    *)
        echo "model yang kamu masukkan tidak ada."
        ;;
esac

^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
```

- Output dari file percabangan2.sh

```
novita@novita:~/Documents/Sisop$ nano percabangan2.sh
novita@novita:~/Documents/Sisop$ bash percabangan2.sh
Model looseleaf apa yang kamu suka?
Polos?
Line?
Grid?
Atau berwarna?
line
model line cocok untuk mencatat
novita@novita:~/Documents/Sisop$ bash percabangan2.sh
Model looseleaf apa yang kamu suka?
Polos?
Line?
Grid?
Atau berwarna?
grid
model grid cocok untuk matematika
novita@novita:~/Documents/Sisop$ bash percabangan2.sh
Model looseleaf apa yang kamu suka?
Polos?
Line?
Grid?
Atau berwarna?
bigboss
model yang kamu masukkan tidak ada.
novita@novita:~/Documents/Sisop$
```

Selain menggunakan if then, untuk membuat percabangan kita bisa menggunakan case \$variabel in. Untuk strukturnya hampir samas seperti if then.

10. Pembuatan Tugas_2.sh

- *Shell scripting* Tugas_2.sh

```
#!/bin/bash

printf "Apakah anda bisa matematika? (y/n) "
read bisa

case "$bisa" in
    "y")
        echo "Oke, lanjut ke soal!"
        ;;
    "n")
        echo "Kamu harus berlatih soal!"
        ;;
    *)
        echo "Yuk, kamu harus berlatih."
        ;;
esac

x=110
y=101

let jumlah=$x+$y
kurang='expr $x - $y'
kali=$(( $x*$y ))
bagi=$(( $x/$y ))
mod=$(( $x%$y ))

printf "Berapakah $x + $y? "
read jawab1

if [ $jawab1 == $jumlah ]
Then
    echo "Nice!"
Else
    echo "Wah, salah!"
Fi

printf "Berapakah $x - $y? "
read jawab2

case "$jawab2" in
    "9")
        echo "Nice!"
        ;;
    *)
        echo "Wah, salah!"
        ;;
esac

printf "Yuk lihat operasi matematika lainnya!\n"
printf "Ketik apapun untuk melanjutkan\n"
read lanjut

echo "x * y = $kali"
echo "x / y = $bagi"
echo "x % y = $mod"

echo "x = $x"
echo "y = $y"

if [ $x == $y ]
Then
```

```

    echo "x sama dengan y"
elif [ $x -gt $y ]
Then
    echo "x lebih besar daripada y"
elif [ $x -lt $y ]
Then
    echo "x lebih kecil daripada y"
Else
    echo "Tidak ada kondisi yang memenuhi"
Fi

```

Shell scripting di atas memadukan fungsi-fungsi yang sudah dipelajari. Syntax yang ada di dalam script di atas antara lain echo, printf, operasi matematika, percabangan if-then, dan percabangan case-in.

- Output dari file Tugas_2.sh

```

novita@novita:~/Documents/Sisop$ bash Tugas_2.sh
Apakah anda bisa matematika? (y/n) n
Kamu harus berlatih soal!
Berapakah 110 + 101? 90
Wah, salah!
Berapakah 110 - 101? 0
Wah, salah!
Yuk lihat operasi matematika lainnya!
Ketik apapun untuk melanjutkan
d
x * y = 11110
x / y = 1
x % y = 9
x = 110
y = 101
x lebih besar daripada y

```

Output jika hasil salah.

```

novita@novita:~/Documents/Sisop$ nano Tugas_2.sh
novita@novita:~/Documents/Sisop$ bash Tugas_2.sh
Apakah anda bisa matematika? (y/n) y
Oke, lanjut ke soal!
Berapakah 110 + 101? 211
Nice!
Berapakah 110 - 101? 9
Nice!
Yuk lihat operasi matematika lainnya!
Ketik apapun untuk melanjutkan
OK
x * y = 11110
x / y = 1
x % y = 9
x = 110
y = 101
x lebih besar daripada y

```

Output jika hasil benar.