

Nama : Denisa Septalian Alhamda

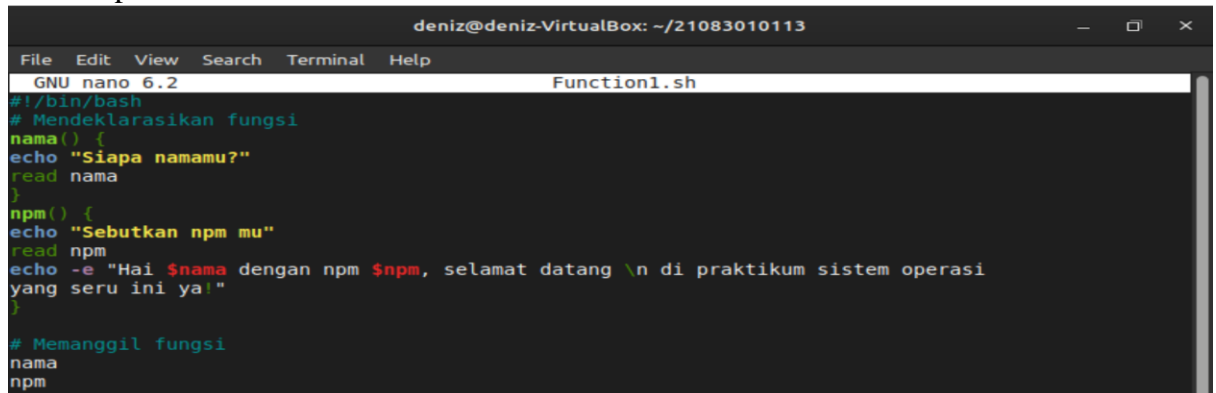
Npm : 21083010113

Kelas : sistem oprasi B

Dokumentasi Bash Function

1. Function1.sh

- Script

A screenshot of a terminal window titled "deniz@deniz-VirtualBox: ~/21083010113". The window shows the GNU nano 6.2 editor editing a file named "Function1.sh". The script content is as follows:

```
#!/bin/bash
# Mendeklarasikan fungsi
nama() {
  echo "Siapa namamu?"
  read nama
}
npm() {
  echo "Sebutkan npm mu"
  read npm
  echo -e "Hai $nama dengan npm $npm, selamat datang \n di praktikum sistem operasi yang seru ini ya!"
}

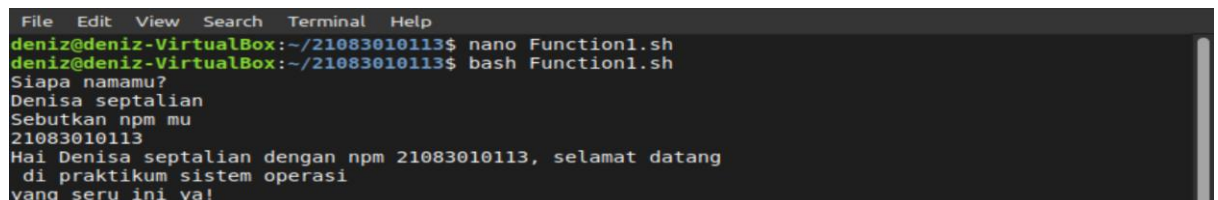
# Memanggil fungsi
nama
npm
```

Mendeklarasikan setiap fungsi, yaitu fungsi nama dengan menyimpan nama user pada variable nama dan fungsi npm dengan menyimpan npm pada variable npm, selanjutnya memanggil fungsi diluar deklarasi fungsi dengan menuliskan nama fungsi.

- Function 1 Syntax

```
nama_fungsi () {
  perintah1
  perintah2
  ...
  perintah_lain
}
```

- Output

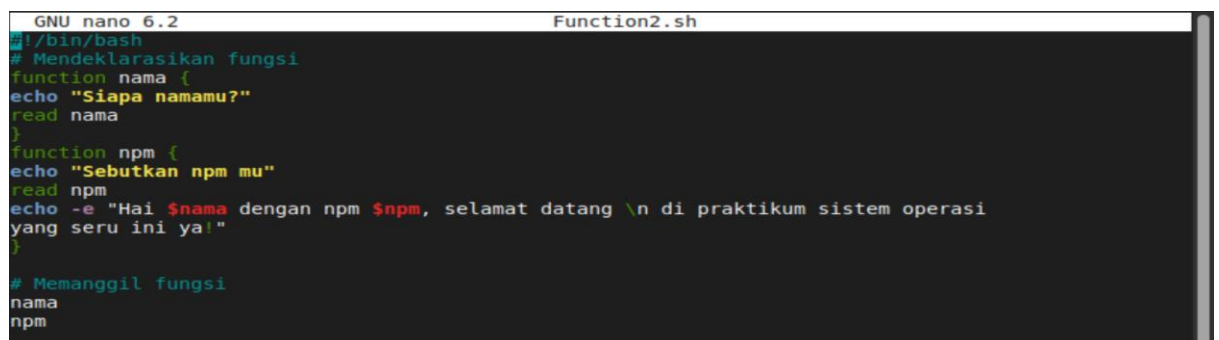
A screenshot of a terminal window showing the execution of the "Function1.sh" script. The prompt is "deniz@deniz-VirtualBox:~/21083010113\$". The user enters "nano Function1.sh" and then "bash Function1.sh". The output of the script is:

```
Siapa namamu?
Denisa septalian
Sebutkan npm mu
21083010113
Hai Denisa septalian dengan npm 21083010113, selamat datang
di praktikum sistem operasi
yang seru ini ya!
```

Fungsi merupakan bagian script atau program yang berisi kumpulan beberapa statement yang melaksanakan tugas tertentu. dengan subrutin kode script kita tentunya lebih sederhana dan terstruktur, karena sekali fungsi telah dibuat dan berhasil maka akan dapat digunakan kapan saja kita inginkan

2. Function2.sh

- Script

A screenshot of a terminal window titled "deniz@deniz-VirtualBox: ~/21083010113". The window shows the GNU nano 6.2 editor editing a file named "Function2.sh". The script content is as follows:

```
#!/bin/bash
# Mendeklarasikan fungsi
function nama {
  echo "Siapa namamu?"
  read nama
}
function npm {
  echo "Sebutkan npm mu"
  read npm
  echo -e "Hai $nama dengan npm $npm, selamat datang \n di praktikum sistem operasi yang seru ini ya!"
}

# Memanggil fungsi
nama
npm
```

Hampir sama dengan function 1, pada function 2 pada nama fungsi diberikan function namafungsi tanpa menggunakan tanda kurung.

Function 2 Syntax

```
function nama_fungsi {  
  
    Statement-1 command...  
  
    Statement-2 command...  
  
    Statement-3 command...  
  
    Etc  
  
}
```

- Output

```
deniz@deniz-VirtualBox:~/21083010113$ nano Function2.sh  
deniz@deniz-VirtualBox:~/21083010113$ bash Function2.sh  
Siapa namamu?  
denisa septalian  
Sebutkan npm mu  
21083010113  
Hai denisa septalian dengan npm 21083010113, selamat datang  
di praktikum sistem operasi  
yang seru ini ya!
```

3. Nested_function.sh

- Script

```
GNU nano 6.2 Nested Function.sh  
#!/bin/bash  
  
# Mendeklarasikan fungsi  
nama() {  
    echo "Siapa namamu?"  
    read nama  
    npm # Memanggil fungsi di dalam fungsi(fungsi bersarang)  
}  
npm() {  
    echo "Sebutkan npm mu"  
    read npm  
    echo -e "Hai $nama dengan npm $npm, selamat datang \n di praktikum sistem operasi  
yang seru ini ya!"  
}  
  
# Memanggil fungsi  
nama
```

Pendeklarasian Nested_function.sh ini hampir sama dengan Function1.sh, namun bedanya ada pada function nama(), yaitu menyelipkan function npm setelah read nama untuk memanggil function npm() diluar deklarasi fungsi.

- Output

```
deniz@deniz-VirtualBox:~/21083010113$ nano Nested_Function.sh  
deniz@deniz-VirtualBox:~/21083010113$ bash Nested_Function.sh  
Siapa namamu?  
Denisa septalian  
Sebutkan npm mu  
21083010113  
Hai Denisa septalian dengan npm 21083010113, selamat datang  
di praktikum sistem operasi  
yang seru ini ya!
```

4. Function_parameter.bash

- Script

```
GNU nano 6.2 Function Parameter.sh
#!/bin/bash

# Mendeklarasikan fungsi
identitas() {
parameter1=$1
parameter2=$2
parameter3=$3
echo "$parameter1"
echo "$parameter2"
echo "$parameter3"
}
echo "Masukkan Nama : "
read a
echo "Masukkan Npm : "
read b
echo "Hobimu Apa : "
read c
printf "\n"
identitas $a $b $c
```

Function Parameter atau juga disebut sebagai Parameter Formal, adalah variabel lokal yang didirikan di dalam deklarasi function (bukan definisi), yang merupakan tempat penyimpanan nilai dari argument yang diberikan saat pemanggilan function.

- Function Parameter Syntax

```
function nama_fungsi atau nama_fungsi() {

par1=$1

par2=$2

command on $par1

}
```

- Output

```
deniz@deniz-VirtualBox:~/21083010113$ nano Function_Parameter.sh
deniz@deniz-VirtualBox:~/21083010113$ bash Function_Parameter.sh
Masukkan Nama :
Denisa septalian
Masukkan Npm :
21083010113
Hobimu Apa :
turu

Denisa
septalian
21083010113
```

Pada function parameter, variable yang dipanggil di deklarasikan didalam function terlebih dahulu dengan variable parameter1=\$1, parameter2=\$2, parameter3=\$3 yang selanjutnya di panggil diluar function pada variable a, b, c. Selanjutnya memanggil function identitas \$a \$b \$c diluar function identitas, sehingga variable a menggantikan posisi \$1 yang mana telah di deklarasikan sebagai parameter1, variable b menggantikan posisi \$2 yang mana telah di deklarasikan sebagai parameter2 dan variable c menggantikan posisi \$3 yang mana telah dideklarasikan sebagai parameter3.

5. Tugas_7

- Script

```
GNU nano 6.2 Tugas 7.sh
#!/bin/bash

{
echo "==luas bidang persegi=="
echo "Masukan Panjang"
read panjang
echo "Masukan Lebar"
read lebar
let luasbidangpersegi=$panjang*$lebar
echo "Luas persegi :
$luasbidangpersegi"
}
```

Melakukan deklarasi 3 function, yaitu ada function panjang() untuk menyimpan variable panjang, function lebar() untuk menyimpan variable lebar, dan function luas() untuk memanggil function panjang dan luas, lalu dilakukan perhitungan variable luas dengan rumus panjang * lebar yang disimpan pada variable luas. Lalu memanggil function luas diluar deklarasi function.

- Output

```
deniz@deniz-VirtualBox:~/21083010113$ bash Tugas_7.sh
==luas bidang persegi==
Masukan Panjang
10
Masukan Lebar
5
Luas persegi :
50
```