

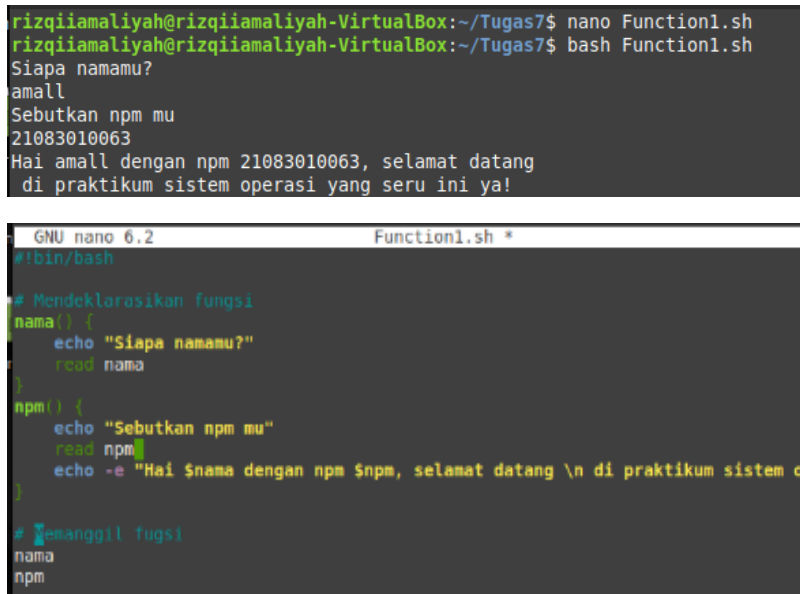
Nama: Rizqii Amaliyah M.

NPM : 21083010063

SHELL PROGRAMMING 4

Fungsi bash pada dasarnya adalah seperangkat perintah yang dapat dipanggil berkali-kali. Tujuan dari sebuah fungsi adalah untuk membantu membuat script bash lebih mudah dibaca dan untuk menghindari penulisan kode yang sama berulang kali. Dibandingkan dengan kebanyakan bahasa pemrograman, fungsi bash agak terbatas. Macam-macam fungsi bash:

1. Fungsi dapat dideklarasikan dalam dua syntax yang berbeda. Untuk pertama dimulai dengan nama fungsi, diikuti dengan tanda kurung. Biasanya syntax ini yang disukai dan lebih banyak digunakan. Perintah di antara kurung kurawal { } disebut sebagai badan fungsi. Kurung kurawal harus dipisahkan dari badan dengan spasi atau baris baru. Definisi fungsi harus ditempatkan sebelum panggilan ke fungsi.



The first screenshot shows a terminal session where a script named 'Function1.sh' is executed. The script prompts for a name and an npm username, then prints a welcome message. The second screenshot shows the source code of 'Function1.sh' in the nano editor. It defines two functions, 'nama()' and 'npm()', and then calls them at the bottom.

```
rizqiiamaliyah@rizqiiamaliyah-VirtualBox:~/Tugas7$ nano Function1.sh
rizqiiamaliyah@rizqiiamaliyah-VirtualBox:~/Tugas7$ bash Function1.sh
Siapa namamu?
amall
Sebutkan npm mu
21083010063
Hai amall dengan npm 21083010063, selamat datang
di praktikum sistem operasi yang seru ini ya!

GNU nano 6.2      Function1.sh *
#!/bin/bash

# Mendefinisikan 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 o

# Panggil fungsi
nama
npm
```

- Pada baris 4 dan 8, didefinisikan fungsi dengan memberi nama kurung kurawal { menandai awal dari badan fungsi.
- Baris 5, 6, 9, 10, dan 11 adalah fungsi badan tubuh. Badan fungsi dapat berisi beberapa perintah, pernyataan, dan deklarasi variabel.
- Baris 7 dan 12, kurung kurawal penutup } , mendefinisikan akhir fungsi.
- Baris 14 dan 15 menjalankan fungsi.

Untuk syntax yang kedua ialah dengan kata yang dicadangkan `function`, diikuti dengan nama fungsi.

```

rizqiiamaliyah@rizqiiamaliyah-VirtualBox:~/Tugas7$ nano Function2.sh
rizqiiamaliyah@rizqiiamaliyah-VirtualBox:~/Tugas7$ bash Function2.sh
Siapa namamu?
amall
Sebutkan npm mu
21083010063
Hai amall dengan npm 21083010063, selamat datang
di praktikum sistem operasi yang seru ini ya!

```

```

GNU nano 6.2 Function2.sh
#!/bin/bash

# Mendefinisikan 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 o
}

# Memanggil fungsi
nama
npm

```

2. Nested function ialah jika suatu fungsi memanggil fungsi lain, mengarah ke fungsi bersarang. Suatu fungsi dapat memanggil fungsi apa pun terlepas dari apakah itu didefinisikan di atasnya dalam kode sumber atau di bawahnya karena fungsi adalah kode statis.

```

rizqiiamaliyah@rizqiiamaliyah-VirtualBox:~/Tugas7$ nano Nested_function.sh
rizqiiamaliyah@rizqiiamaliyah-VirtualBox:~/Tugas7$ bash Nested_function.sh
Siapa namamu?
amall
Sebutkan npm mu
21083010063
Hai amall dengan npm 21083010063, selamat datang
di sistem operasi yang seru ini ya!

```

```

GNU nano 6.2 Nested_function.sh
#!/bin/bash

# Mendefinisikan fungsi
nama() {
    echo "Siapa namamu?"
    read nama
    npm
}

npm() {
    echo "Sebutkan npm mu"
    read npm
    echo -e "Hai $nama dengan npm $npm, selamat datang \n di sistem operasi yan
}

# Memanggil fungsi
nama

```

- Function nama dipanggil oleh lingkungan panggilan dan mencetak pesan “Siapa namamu?”
- Function nama memanggil function npm dan mencetak pesan

- Function npm mengembalikan kontrol ke function nama. Karena tidak ada pernyataan lain yang ditentukan setelah pemanggilan fungsi, function nama mengembalikan kontrol ke lingkungan pemanggilan utama.
3. Parameter dapat diteruskan ke fungsi di shell untuk operasi lebih lanjut dalam suatu fungsi. Parameter di shell dilewatkan menggunakan syntax berikut:
- a. `$0` : mendefinisikan nama file dan diteruskan secara default
 - b. `$k` : mendefinisikan parameter ke-k

```

rizqiiamaliyah@rizqiiamaliyah-VirtualBox:~/Tugas7$ nano Function_parameter.sh
rizqiiamaliyah@rizqiiamaliyah-VirtualBox:~/Tugas7$ bash Function_parameter.sh
Masukkan Nama :
amall
Masukkan Npm :
21083010063
Hobimu Apa :
membaca buku

amall
21083010063
membaca

```

```

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"

```

```

printf "\n"
identitas $a $b $c

```

Disini tiga parameter dilewatkan yaitu `$1`, `$2`, dan `$3` yang dicetak ke output standar.

LATIHAN SOAL

```
rizqiamaliyah@rizqiamaliyah-VirtualBox:~/Tugas7$ nano Tugas_7.sh
```

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

function panjang() {
    echo "Masukkan Panjang :"
    read panjang
}
function lebar() {
    echo "Masukkan Lebar :"
    read lebar
}
function luas() {
    let luas=$panjang*$lebar
    echo "Luas Persegi :"
    echo "$luas"
}

panjang
lebar
luas persegi
```

Pada script diatas ialah menggunakan pencadangan `function`. Diatas diperintahkan untuk menghitung luas persegi dengan output memasukkan panjang dan lebar. Untuk menghitung luas persegi dengan menggunakan command `let` dengan rumus panjang x lebar.

```
rizqiamaliyah@rizqiamaliyah-VirtualBox:~/Tugas7$ bash Tugas_7.sh
Masukkan Panjang :
4
Masukkan Lebar :
10
Luas Persegi :
40
rizqiamaliyah@rizqiamaliyah-VirtualBox:~/Tugas7$
```