

A. TUJUAN

1. Menggunakan Variabel untuk mengenali (meng-identifier) data yang nilainya sewaktu-waktu dapat berubah.
2. Menggunakan Environment Variable yang digunakan khusus oleh shell atau system linux untuk proses kerja system.
3. Menggunakan Positional Parameter (parameter posisi) yang digunakan shell untuk menampung argumen yang diberikan terhadap shell.
4. Menggunakan User Define Variable yaitu variabel yang didefinisikan sendiri oleh pembuat script sesuai dengan kebutuhannya.

B. DASAR TEORI

Variabel

Secara sederhana variabel adalah pengenal (identifier) berupa satuan dasar penyimpanan yang isi atau nilainya sewaktu-waktu dapat berubah baik oleh eksekusi program (runtime program) ataupun proses lain yang dilakukan sistem operasi. Variabel dibagi menjadi 3 kategori:

1. Environment Variable
2. Positional Parameter
3. User Define Variable

❖ Environment Variabel

Environment Variabel atau variabel lingkungan adalah variabel yang digunakan khusus oleh shell atau system linux untuk proses kerja system seperti variabel

PS1, PS2, HOME, PATH, USER, SHELL, dan sebagainya. Jika variabel ini digunakan akan berdampak pada system, misalkan variabel PS1 yang digunakan untuk mengeset prompt shell pertama yaitu prompt tempat anda mengetikkan perintah - perintah shell (defaultnya "`¥s-¥v¥$`"), PS2 untuk prompt pelengkap perintah, prompt ini akan ditampilkan jika perintah yang dimasukkan dianggap belum lengkap oleh shell (defaultnya ">"). Variabel PS1 dan PS2 dapat disetting seperti berikut.

```
$PS1LAMA=$PS1
```

data string yang diinginkan dimasukkan pada variabel PS1

```
$PS1="Ini Promptku!"
Ini Promptku!PS2="Cobalah ? "
```

maka prompt pertama dan kedua akan berubah, untuk mengembalikan PS1 ke prompt semula perintahnya adalah :

```
$PS1=$PS1LAMA
```

Jika ingin mengkonfigurasi prompt shell, bash telah menyediakan beberapa backslash karakter diantaranya adalah:

<code>¥a</code>	ASCII bell character (07)
<code>¥d</code>	date dengan format "Weekday Month Date" (misalnya "Tue May 26")
<code>¥e</code>	ASCII escape character (033)
<code>¥H</code>	hostname (namahost)
<code>¥n</code>	newline (karakter baru)
<code>¥w</code>	Direktory aktif
<code>¥t</code>	time dalam 24 jam dengan format HH:MM:SS
<code>dll</code>	man bash:-)

contoh pemakaiannya:

```
$PS1="[ \t][ \u@\h:\w]\$"
```

agar prompt shell hasil konfigurasi anda dapat tetap berlaku (permanen) sisipkan pada file `.bashrc` atau `.profile`

❖ Positional Parameter

Positional Parameter / parameter posisi yaitu variabel yang digunakan shell untuk menampung argumen yang diberikan terhadap shell baik berupa argumen waktu sebuah file dijalankan atau argumen yang dikirim ke subrutin. variabel yang dimaksud adalah 1,2,3,dan seterusnya, lebih jelasnya lihat contoh script berikut :

```
echo $1 adalah salah satu $2 populer di $3
```

masing-masing argumen pada contoh diatas akan disimpan pada variabel 1,2,3 sesuai posisinya. variabel spesial lain yang dapat digunakan yaitu 0, #, dan * seperti diperlihatkan pada script berikut:

```
echo "Nama script anda : $0";  
echo "Banyak argumen : $#";  
echo "Argumennya adalah: $*";
```

❖ User Defined Variable

User Defined Variable atau variabel yang didefinisikan sendiri oleh pembuat script sesuai dengan kebutuhannya, beberapa hal yang perlu diperhatikan dalam mendefenisikan variabel adalah:

- dimulai dengan huruf atau underscore

```
VARIABLE_NAME=VALUE
```

Contoh :

```
VARIABLE1=Value  
echo "My variable holds: $VARIABLE1"
```

- hindari pemakaian spesial karakter seperti *,\$,#,dll...
- bash bersifat case sensitive, maksudnya membedakan huruf besar dan kecil, a berbeda dengan A, nama berbeda dengan Nama, NaMa, dan sebagainya.

untuk mengeset nilai variabel dengan menggunakan operator assignment (pemberi nilai)"=", contohnya :

```

myos="linux"           #double-quoted
nama='penguin'        #single-quoted
hasil=`ls -l`;        #back-quoted
angka=12

```

Jika diperhatikan ada 3 tanda kutip yang digunakan untuk memberikan nilai string ke suatu variabel, adapun perbedaannya adalah:

- dengan kutip ganda (double-quoted), bash mengizinkan untuk menyisipkan variabel di dalamnya.
- dengan kutip tunggal (single-quoted), akan ditampilkan apa adanya.
- dengan kutip terbalik (double-quoted), bash menerjemahkan sebagai perintah yang akan dieksekusi.

C. PROSEDUR

1. Masuk ke sistem operasi Linux.
2. Login sebagai stD3XXYYY.
3. Gunakan variable dalam pembuatan program : Environment Variable, Positional Parameter, dan User Define Variable.
4. Catatlah hasil dari setiap program bash script pada prosedur no.3 sebagai laporan sementara.
5. Kerjakan soal latihan yang telah disediakan pada modul praktikum.

D. PERCOBAAN

Percobaan 1 : Menggunakan Environment Variable

1. Mengeset prompt shell

```

$PS1LAMA=$PS1
$PS1="Ini Promptku!"
Ini Promptku!PS2="Cobalah ? "
$PS1=$PS1LAMA
^C

```

2. Melihat environment pada system

```

$set
$printenv

```

3. Script untuk pengecekan environment variable

```

$vi check_env.sh

#!/bin/bash
#unset DISPLAY
if [ "$DISPLAY" == "" ]
then
    echo "DISPLAY not set, using :0.0 as default."
    DISPLAY=":0.0"
fi

#unset SHELL
if [ "$SHELL" == "" ]
then
    echo "Using /bin/bash, which is the shell you should use."
    SHELL=/bin/bash
fi

#unset USER
if [ "$USER" == "" ]
then
    echo -n "Please enter your username: "
    read USER
fi

#unset HOME
if [ "$HOME" == "" ]
then
    # Check for Mac OS X home.
    if [ -d "/Users/$USER" ]
    then
        HOME="/Users/$USER"
    # Check for Linux home.
    elif [ -d "/home/$USER" ]
    then
        HOME="/home/$USER"
    else
        echo -n "Please enter your home directory: "
        read HOME
        echo
    fi
fi

# Display all the values.
echo "DISPLAY=$DISPLAY"
echo "SHELL=$SHELL"
echo "USER=$USER"
echo "HOME=$HOME"

```

4. Men-setting environment variable dengan perintah *export*

```

$vi set_env.sh

#!/bin/bash

```

```
var1="value1"
export var1
var2="value2"; export var2
export var3="value3"

echo "$var1, $var2, $var3"
```

5. Men-setting environment variable dengan perintah *export*

```
$ vi set_env1.sh

var1="value1"
var2="value2"
var3="value3"
export var1 var2 var3

echo "$var1, $var2, $var3"
```

Percobaan 2 : Menggunakan Positional Variable

1. Penggunaan beberapa variable (1,2,3)

```
$vi pos1.sh

#!/bin/bash
#argumen1
echo $1 adalah salah satu $2 populer di $3

$ ./pos1.sh bash shell linux
```

2. Penggunaan beberapa variable *spesial*

```
$ vi pos2.sh

#!/bin/bash
#argumen2
clear
echo "Nama script anda : $0";
echo "Banyak argumen : $#";
echo "Argumennya adalah: $*";

$ ./pos2.sh 1 2 3 empat
```

3. Penggunaan beberapa variable *spesial*

```
$ vi pos3.sh

#!/bin/bash
echo "alice: $@"
echo "$0: $1 $2 $3 $4"
echo "$# arguments"

$ ./pos3.sh alice in wonderland
```

4. Penggunaan parameter dan argumen

```
$ vi pos4.sh

#!/bin/bash
#menggunakan parameter dan argumen
```

```
echo "Argumen pertama adalah : $1"
echo "Argumen kedua adalah : $2"
echo "Argumen ketiga adalah : $3"

$ ./pos4.sh 3 4 5
```

5. Penggunaan positional variable

```
$ vi pos5.sh

#!/bin/bash
var1="outside function"
echo var1: $var1
echo $0: $1 $2

$ ./pos5.sh 3 4 5
```

6. Penggunaan positional variable dan fungsi

```
$ vi pos6.sh

#!/bin/bash
function afunc
{
    echo in function:$0 $1 $2
    var1= "in function"
    echo var1:$var1
}
var1="outside function"
echo var1: $var1
echo $0: $1 $2
afunc funcarg1 funcarg2
echo var1: $var1
echo $0 $1 $2

$ ./pos6.sh hai selamat datang
```

Percobaan 3 : Penggunaan User Define Variable

1. Penggunaan tanda kutip ganda

```
$ vi define.sh

variable1="A value"
echo "My variable holds: $variable1"

echo "MYVAR is: $MYVAR"
MYVAR="hi there"
echo "MYVAR is: $MYVAR"

$ ./define.sh
```

2. Menyisipkan variable nama menggunakan tanda kutip ganda (double-quoted)

```
$ vi defi.sh

#!/bin/bash
```

```
nama="penguin"
kata="Hi $nama, apa khabarmu"
echo $kata;

$ ./defi.sh
```

3. Menyisipkan variable nama menggunakan tanda kutip tunggal (single-quoted)

```
$ vi defi1.sh

#!/bin/bash

nama="penguin"
kata='Hi $nama, apa khabarmu'
echo $kata;

$ ./defi1.sh
```

4. Menyisipkan variable nama menggunakan tanda kutip terbalik

```
$ vi defi2.sh

#!/bin/bash
hapus=`clear`;
isi=`ls -l`;          #simpan hasil perintah ls -l
#hapus layar
echo $hapus

#ls -l
echo $isi;

$ ./defi2.sh
```

5. Penggunaan variable dan perintah dasar shell dalam program

```
$ vi defi3.sh

VARIABLE1=Cool
VARIABLE2="this system runs"
echo -n "$VARIABLE1 $VARIABLE2 "
uname

$ ./defi3.sh
```

E. Latihan

1. Hasil eksekusi suatu program bash script adalah sebagai berikut :

Masukkan nama depan anda : **Joni**
 Masukkan nama belakang anda : **Ar**
 Masukkan nama kota tempat tinggal anda : **Jakarta**
 Terima kasih, **Joni Ar** dari **Jakarta**, selamat datang di perusahaan.
 Anda telah terdaftar sebagai pekerja pada Unit 10236.

- a. Buatlah program bash scriptnya
 - b. gunakan variabel FULLNAME untuk menyimpan nama lengkap pekerja, dan variable STATE untuk menyimpan kota tempat tinggal pekerja.
 - c. Gunakan variabel MESSAGE untuk menyimpan pesan (Terima kasih, Joni Ar dari jakarta di perusahaan)
2. Buatlah program bash script untuk menampilkan tulisan dibawah ini.

```

Test 1 =====
abc
abc
{abc}
Test 2 =====
myvar
myvar
abc
$myvar
Test 3 =====
abcdef
Test 4 =====
abcabc
abcabc
Test 5 =====
a b c
abc

```

3. Buatlah program bash script untuk menjalankan perintah shell dari sebuah variable, yaitu perintah yang dapat menampilkan isi dari direktori /usr/local.