

Nama : Fajri Uswatul Hanifah

NPM : 21083010058

Kelas : Sistem Operasi A

## Array

### A. Macam Macam Deklarasi Array

Array merupakan kumpulan variabel dengan tipe sejenis, yang disimpan ke dalam variabel dengan nama yang sama, dengan memberi indeks pada variabel untuk membedakan antara yang satu dengan yang lain.

#### 1. Array Indirect Declaration

Deklarasi dilakukan dengan menetapkan nilai dalam indeks tertentu dari variabel array. Command dari deklarasinya sebagai berikut :

“Array\_name[index]=value”

- Membuat file

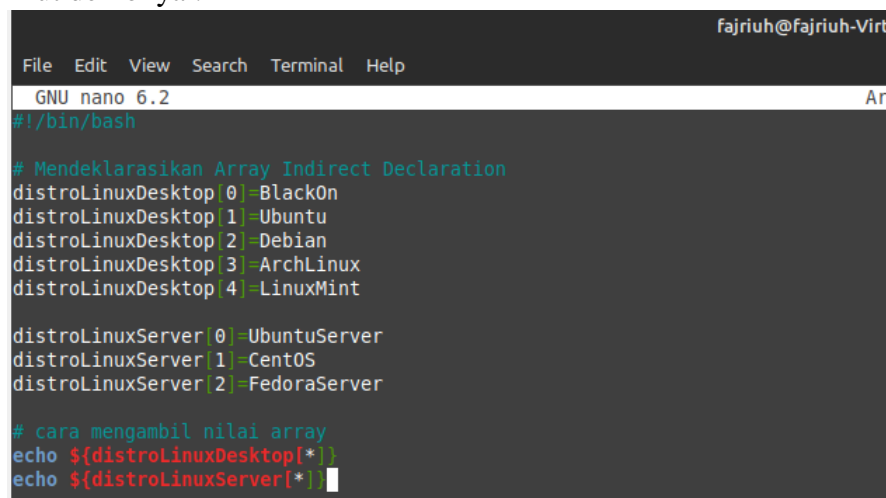
Berikut demonya :

```
fajriuh@fajriuh-VirtualBox:~/Sisop/Tugas-6$ nano Array ID.sh
```

Untuk membuat contoh dari *array indirect declaration* kita memerlukan suatu file, maka dari itu kita harus membuat file terlebih dahulu hal ini dilakukan dengan cara menjalankan command “nano Array\_ID.sh” seperti gambar scrip diatas.

- Menuliskan isi file .sh

Berikut demonya :



```
fajriuh@fajriuh-Virt
File Edit View Search Terminal Help
GNU nano 6.2
#!/bin/bash

# Mendeklarasikan Array Indirect Declaration
distroLinuxDesktop[0]=BlackOn
distroLinuxDesktop[1]=Ubuntu
distroLinuxDesktop[2]=Debian
distroLinuxDesktop[3]=ArchLinux
distroLinuxDesktop[4]=LinuxMint

distroLinuxServer[0]=UbuntuServer
distroLinuxServer[1]=CentOS
distroLinuxServer[2]=FedoraServer

# cara mengambil nilai array
echo ${distroLinuxDesktop[*]}
echo ${distroLinuxServer[*]}
```

Dari scrip diatas dapat dijelaskan kita akan mendeklarasikan dua array, yaitu distroLinuxDekstop dan distroLinuxServer sekaligus dengan elemen ke-n dari masing-masing array tersebut. Dengan menggunakan command yang telah dijelaskan diatas yaitu “Array\_name[index]=value”

Dengan menggunakan cara deklarasi ini, dapat ditetapkan urutan atau posisi indeks suatu elemen dalam array sesuai dengan yang diinginkan. Apabila ingin menginisialisasikan elemen pada indeks 1, maka tidak diperlukan inisialisasi elemen pada indeks 0. Dalam script dideklarasikan 2

array yaitu `distroLinuxDekstop` dan `distroLinuxServer` secara tidak langsung satu persatu sesuai indeks yang ingin dimasukan juga.

Kemudian setelah kedua array dideklarasikan, kita dapat menampilkan dan memanggil elem dalam kedua array tersebut dengan command `echo` terhadap variabel array dengan indeks yang telah ditentukan. Untuk memanggil elemen dalam array dilakukan dengan menuliskan syntax “`${nama_array[indeks]}`”. Apabila ingin memanggil semua elemen dalam array dilakukan dengan mengisi opsi indeks dengan `*` atau `@` seperti yang ada pada gambar di atas.

- Menjalankan skrip

Setelah file yang kita buat disimpan, maka kita dapat menguji skrip kita dengan menjalankan nya menggunakan command “`bash Array_ID.sh`”

Berikut demonya :

```
fajriuh@fajriuh-VirtualBox:~/Sisop/Tugas-6$ bash Array_ID.sh
BlackOn Ubuntu Debian ArchLinux LinuxMint
UbuntuServer CentOS FedoraServer
```

Dari scrip diatas dapat dijelaskan bahwa output yang keluar adalah seluruh elemen dari array yang kita deklarasikan tadi, yaitu dari array `distroLinuxDekstop` dan `distroLinuxServer`.

## 2. *Array Explicit Declaration*

Deklarasi dilakukan dengan mendeklarasikan array kemudian menetapkan nilainya. Command dari deklarasinya sebagai berikut :

“`declare -a Array_name Membuat file`”

- Membuat file

Berikut demonya :

```
fajriuh@fajriuh-VirtualBox:~/Sisop/Tugas-6$ nano Array ED.sh
```

Untuk membuat contoh dari *array explicit declaration* kita memerlukan suatu file, maka dari itu kita harus membuat file terlebih dahulu hal ini dilakukan dengan cara menjalankan command “`nano Array_ED.sh`” seperti gambar scrip diatas.

- Menuliskan isi file .sh

Berikut demonya :

```

fajriuh
File Edit View Search Terminal Help
GNU nano 6.2
#!/bin/bash

# Mendeklarasikan Array Explicit Declaration :
declare -a angka

# clear
i=0;
while [ $i -le 4 ];
do
    let isi=$i*2;
    angka[$i]=$isi;
    let i=$((i+1));
done

# tampilkan semua elemen array indexnya berisi "+" atau "@"
echo ${angka[@]}

```

Dari scrip diatas dapat dijelaskan sebagai berikut :

Pertama kita deklarasikan array dengan nama angka menggunakan command “declare -a angka”. Selanjutnya kita membuat while loop untuk mengisi array “angka”. Hal ini dilakukan dengan cara menginisialisasi variable I dengan nilai angka 0 dengan kondisi nilai variabel i kurang dari atau sama dengan 4 ([ \$a -lt 10 ]) pada dalam dengan command (let isi=\$i\*2) Variabel dengan nama isi yang memiliki nilai dari perkalian nilai variabel i dengan bilangan 2, (angka[\$i]=\$isi) Diinisialisasikan elemen array angka dengan indeks kei adalah nilai variabel isi, (let i=\$((i+1))) Nilai variabel i ditambah dengan bilangan 1.

Kemudian digunakan command echo terhadap variabel array dengan indeks yang telah ditentukan. Untuk memanggil elemen dalam array dilakukan dengan syntax berikut: \${nama\_array[indeks]} Apabila ingin memanggil semua elemen dalam array dilakukan dengan mengisi opsi indeks dengan \* atau @ seperti yang ada pada gambar di atas.

- Menjalankan skrip  
Setelah file yang kita buat disimpan, maka kita dapat menguji skrip kita dengan menjalankan nya menggunakan command “bash Array\_ED.sh”  
Berikut demonya :

```

fajriuh@fajriuh-VirtualBox:~/Sisop/Tugas-6$ bash Array_ED.sh
0 2 4 6 8

```

### 3. Array Compound Assignment

Deklarasi dilakukan dengan Mendeklarasikan array dengan sekelompok nilai.

“Array\_name=([1]=10 [2]=20 [3]=30)”

- Membuat file  
Berikut demonya :

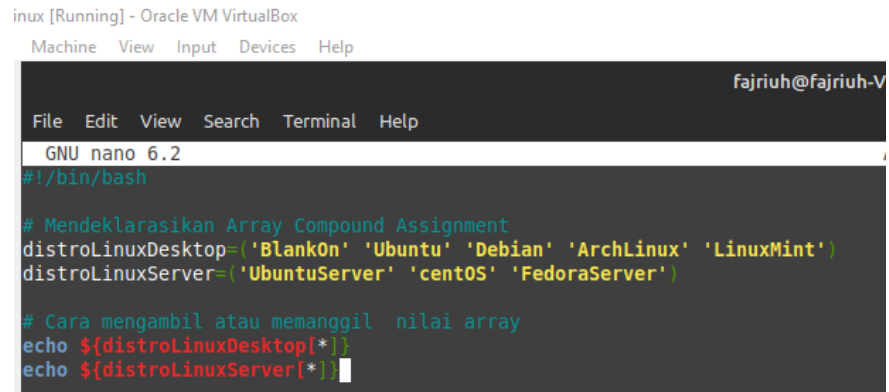
```

fajriuh@fajriuh-VirtualBox:~/Sisop/Tugas-6$ nano Array CA.sh

```

Untuk membuat contoh dari *array compound assignment* kita memerlukan suatu file, maka dari itu kita harus membuat file terlebih dahulu hal ini dilakukan dengan cara menjalankan command “nano Array\_CA.sh” seperti gambar scrip diatas.

- Menuliskan isi file .sh  
Berikut demonya :



```

inux [Running] - Oracle VM VirtualBox
Machine View Input Devices Help
fajriuh@fajriuh-V
File Edit View Search Terminal Help
GNU nano 6.2
#!/bin/bash

# Mendeklarasikan Array Compound Assignment
distroLinuxDesktop=('BlankOn' 'Ubuntu' 'Debian' 'ArchLinux' 'LinuxMint')
distroLinuxServer=('UbuntuServer' 'centOS' 'FedoraServer')

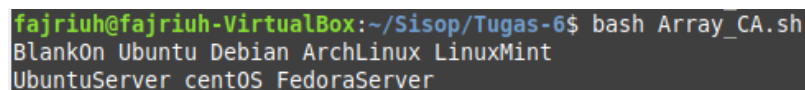
# Cara mengambil atau memanggil nilai array
echo ${distroLinuxDesktop[*]}
echo ${distroLinuxServer[*]}

```

Dari scrip diatas dapat dijelaskan kita akan mendeklarasikan dua array, yaitu distroLinuxDekstop dan distroLinuxServer sekaligus dengan elemen dari masing-masing array tersebut. Struktur syntax dari deklarasi array ini adalah sebagai berikut: nama\_array=(‘elemen-1’ ‘elemen-2’ ... ‘elemen-n’) Deklarasi array dengan cara ini akan memaksa inisialisasi setiap elemen dalam array sesuai dengan indeks yang terurut. Apabila ingin menetapkan elemen pada indeks tertentu semisal pada indeks 2, maka elemen indeks 0 dan 1 harus diinisialisasikan terlebih dahulu meskipun hanya berupa string kosong.

Digunakan command echo terhadap variabel array dengan indeks yang telah ditentukan. Untuk memanggil elemen dalam array dilakukan dengan menuliskan syntax berikut: \${nama\_array[indeks]} Apabila ingin memanggil semua elemen dalam array dilakukan dengan mengisi opsi indeks dengan \* atau @ seperti yang ada pada gambar di atas.

- Menjalankan skrip  
Setelah file yang kita buat disimpan, maka kita dapat menguji skrip kita dengan menjalankannya menggunakan command “bash Array\_CA.sh”  
Berikut demonya :



```

fajriuh@fajriuh-VirtualBox:~/Sisop/Tugas-6$ bash Array_CA.sh
BlankOn Ubuntu Debian ArchLinux LinuxMint
UbuntuServer centOS FedoraServer

```

#### 4. Array Multi Dimensi

Didalam Script Bash, Bash tidak memiliki array multi dimensi. Dikarenakan Bash menyediakan variabel array terindeks dan asosiatif satu dimensi. Variabel apa saja dapat digunakan sebagai array yang diindeks, mendeklarasikan builtin akan secara eksplisit mendeklarasikan array, dll. Tetapi kita dapat mensimulasikan efek yang agak mirip dengan multi array asosiatif dimensi, Deklarasi dilakukan dengan menetapkan nilai dalam indeks tertentu dari variabel array.

- Membuat file  
Berikut demonya :

```
fajriuh@fajriuh-VirtualBox:~/Sisop/Tugas-6$ nano Array MD.sh
```

Untuk membuat contoh dari *array multi dimensi* kita memerlukan suatu file, maka dari itu kita harus membuat file terlebih dahulu hal ini dilakukan dengan cara menjalankan command “nano Array\_MD.sh” seperti gambar scrip diatas.

- Menuliskan isi file .sh  
Berikut demonya :

```
fajriuh@fajriuh-VirtualBox:~/Sisop/Tugas-6$ nano Array MD.sh
File Edit View Search Terminal Help
GNU nano 6.2
#!/bin/bash

# Mendeklarasikan Array 2 Dimensi " : " pemisah nilai (array [3][4])
array2dimensi="1.1:1.2:1.3:1.4 2.1:2.2:2.3:2.4 3.1:3.2:3.3:3.4"

# Mengkali multi dimensi -> dengan pemisah dimensi "tr :"
# Membuat Function dimensi baris dan dimensi kolom
function dimensiBaris {
    for baris in $array2dimensi
    do
        dimensiKolom `echo $baris | tr : " "`
    done
}

function dimensiKolom {
    for kolom in $*
    do
        echo -n $kolom " "
    done
    echo
}

# Melakukan pemanggilan fungsi
dimensiBaris
```

- Menjalankan skrip  
Setelah file yang kita buat disimpan, maka kita dapat menguji skrip kita dengan menjalankan nya menggunakan command “bash Array\_MD.sh”  
Berikut demonya :

```
fajriuh@fajriuh-VirtualBox:~/Sisop/Tugas-6$ bash Array_MD.sh
1.1 1.2 1.3 1.4
2.1 2.2 2.3 2.4
3.1 3.2 3.3 3.4
```

## 5. Latihan Soal

## Soal Latihan

- Buatlah program array yang dapat menghitung nilai IPK mahasiswa yang menerapkan beberapa konsep pemrograman bash seperti diatas dengan ketentuan sbb!
  - user input data arrayIPSMahasiswa[index]
  - $IPK = (\text{jumlah nilai IPS}) / (\text{jumlah data IPS})$

Contoh Output :

```
Input : 3
4
2
3

IPS mhs = 9 / 3
IPK mhs = 3
```

- Membuat file  
Berikut demonya :

```
fajriuh@fajriuh-VirtualBox:~/Sisop/Tugas-6$ nano Tugas6.sh
```

Untuk membuat suatu perulangan kita memerlukan suatu file, maka dari itu kita harus membuat file terlebih dahulu hal ini dilakukan dengan cara menjalankan command “nano Tugas6.sh” seperti gambar scrip diatas.

- Menuliskan isi file .sh  
Berikut demonya :

```
GNU nano 6.2                                     Tugas6.s
echo "--Simple Program for Counting Student's IPK--"
echo "-----"

declare -a ips_num

echo -n "Enter Total IPS: "
read ips

i=0
while [ $i -lt $ips ];
do
    echo -n "Score Subject( $i ): "
    read ips_num[$i];
    let i=i+1;
done

j=0
while [ $j -le $i ];
do
    let ips_sum=ips_sum+ips_num[$j];
    let j=j+1;
done

let IPK=$ips_sum/$ips

echo
echo "--Student's Final Result--"
echo "-----"
echo "IPS Student = $ips_sum / $ips"
echo "IPK Student = $IPK"
echo
```

Menuliskan scrip sesuai dengan kriteria tugas yang telah ditentukan

- Menjalankan skrip

Setelah file yang kita buat disimpan, maka kita dapat menguji skrip kita dengan menjalankan nya menggunakan command “bash Tugas\_6.sh”

Berikut demonya :

```
fajriuh@fajriuh-VirtualBox:~/Sisop/Tugas-6$ nano Tugas6.sh
fajriuh@fajriuh-VirtualBox:~/Sisop/Tugas-6$ bash Tugas6.sh
--Simple Program for Counting Student's IPK--
-----
Enter Total IPS: 4
Score Subject( 0 ): 3
Score Subject( 1 ): 4
Score Subject( 2 ): 2
Score Subject( 3 ): 4

--Student's Final Result--
-----
IPS Student = 13 / 4
IPK Student = 3
```