

## 1. Array [Indirect Declaration]

Tipe deklarasi array ini menetapkan tiap nilai arraynya secara implisit dengan nama\_array[index] = value. Penerapannya seperti berikut.

```
GNU nano 6.2 Array_ID.sh *
#!/bin/bash

#deklarasi array inderect declaration
distroLinuxDesktop[0]=BlankOn
distroLinuxDesktop[1]=Ubuntu
distroLinuxDesktop[2]=Debian
distroLinuxDesktop[3]=ArchLinux
distroLinuxDesktop[4]=LinuxMint

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

Untuk print out semua elemen dari array tersebut dapat menggunakan indeks [\*] atau [@].

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

Maka nilai akan keluar

```
chrysilla@chrysilla-VirtualBox:~$ bash Array_ID.sh
BlankOn Ubuntu Debian ArchLinux LinuxMint
UbuntuServer CentOS FedoraServer
chrysilla@chrysilla-VirtualBox:~$
```

## 2. Array [Explicit Declaration]

Tipe array ini berkebalikan dengan tipe array indirect declaration dimana kita mendeklarasikan array dengan declare -a nama\_array kemudian menetapkan valuenya. Contohnya sebagai berikut.

```
GNU nano 6.2 Array_ED.sh *
#!/bin/bash

#deklarasi 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
#dengan indexnya berisi "*" atau "@"
echo ${angka[@]}
```

Pertama kita mendeklarasikan array yang diberi nama `angka` lalu pada perulangan `while` dilakukan generate elemen array dengan kondisi  $i \leq 4$  dimana  $i$  akan dikali 2 dan hasil kalinya merupakan elemen dari array `angka` tersebut.

```
chrysilla@chrysilla-VirtualBox:~$ bash Array_ED.sh
0 2 4 6 8
chrysilla@chrysilla-VirtualBox:~$
```

### 3. Array [Compound Assignment]

Tipe array ini mendeklarasikan array dengan sekelompok nilai `nama_array=(value1, value2, value3, ...)` sehingga akan lebih efisien dibandingkan dengan tipe array indirect declaration. Penerapannya sebagai berikut.

```
GNU nano 6.2 Array_CA.sh *
#!/bin/bash

#deklarasi array compound assignment
distroLinuxDesktop=('BlankOn' 'Ubuntu' 'Debian' 'ArchLinux' 'LinuxMint')
distroLinuxServer=('UbuntuServer' 'CentOS' 'FedoraServer')

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

Hasilnya

```
chrysilla@chrysilla-VirtualBox:~$ bash Array_CA.sh
BlankOn Ubuntu Debian ArchLinux LinuxMint
UbuntuServer CentOS FedoraServer
chrysilla@chrysilla-VirtualBox:~$
```

### 4. Array Multi Dimensi

Bash tidak memiliki array multi dimensi sehingga kita dapat mensimulasikan script di bawah ini yang cukup mirip dengan multi array asosiatif dimensi.

```
GNU nano 6.2 Array_MD.sh *
#!/bin/bash

#deklarasi array2dimensi " : " 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"

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

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

#test
dimensiBaris
```

Maka outputnya seperti berikut ini.

```
chrysilla@chrysilla-VirtualBox:~$ 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
chrysilla@chrysilla-VirtualBox:~$
```

## 5. Latihan Soal

Buat program array yang dapat menghitung nilai IPK mahasiswa yang menerapkan beberapa konsep pemrograman bash seperti diatas dengan ketentuan

- o user input data arrayIPSMahasiswa[index]
- o IPK = (jumlah nilai IPS) / (jumlah data IPS)

### Script Bash

```
GNU nano 6.2                                     Tugas6.sh
#!/bin/bash

#deklarasi array
declare -a arrIPS

#user input jumlah element dalam array
echo -n "Input : "
read n

#user input nilai-nilai di dalam array
for ((i=0; i<n; i=i+1))
do
    read arrIPS[$i]
done

#perhitungan nilai IPK
sumIPS=0;
for i in ${arrIPS[@]};
do
    let sumIPS+= $i;
    let hasil=$sumIPS/$n;
done

#print out
echo ""
echo -e "IPS mhs = $sumIPS / $n"
echo -e "IPK mhs = $hasil"
```

### Penjelasan Script :

- o Pertama kita mendeklarasikan array (arrIPS)
- o Kedua user menginputkan jumlah elemen array (n)
- o Ketiga user dapat menginputkan nilai-nilai dalam array selama memenuhi  $i < n$

- Keempat dilakukan perhitungan nilai IPK dengan menjumlahkan nilai-nilai dalam array (`sumIPS+= $i$` ) kemudian pada variable `hasil` (perhitungan akhir nilai IPK mahasiswa) dilakukan pembagian dari `sumIPS / n`
- Terakhir memunculkan hasil perhitungan nilai IPK

### Output

```
chrysilla@chrysilla-VirtualBox:~/Tugas6_SISOP$ bash Tugas6.sh
Input : 3
4
2
3

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