

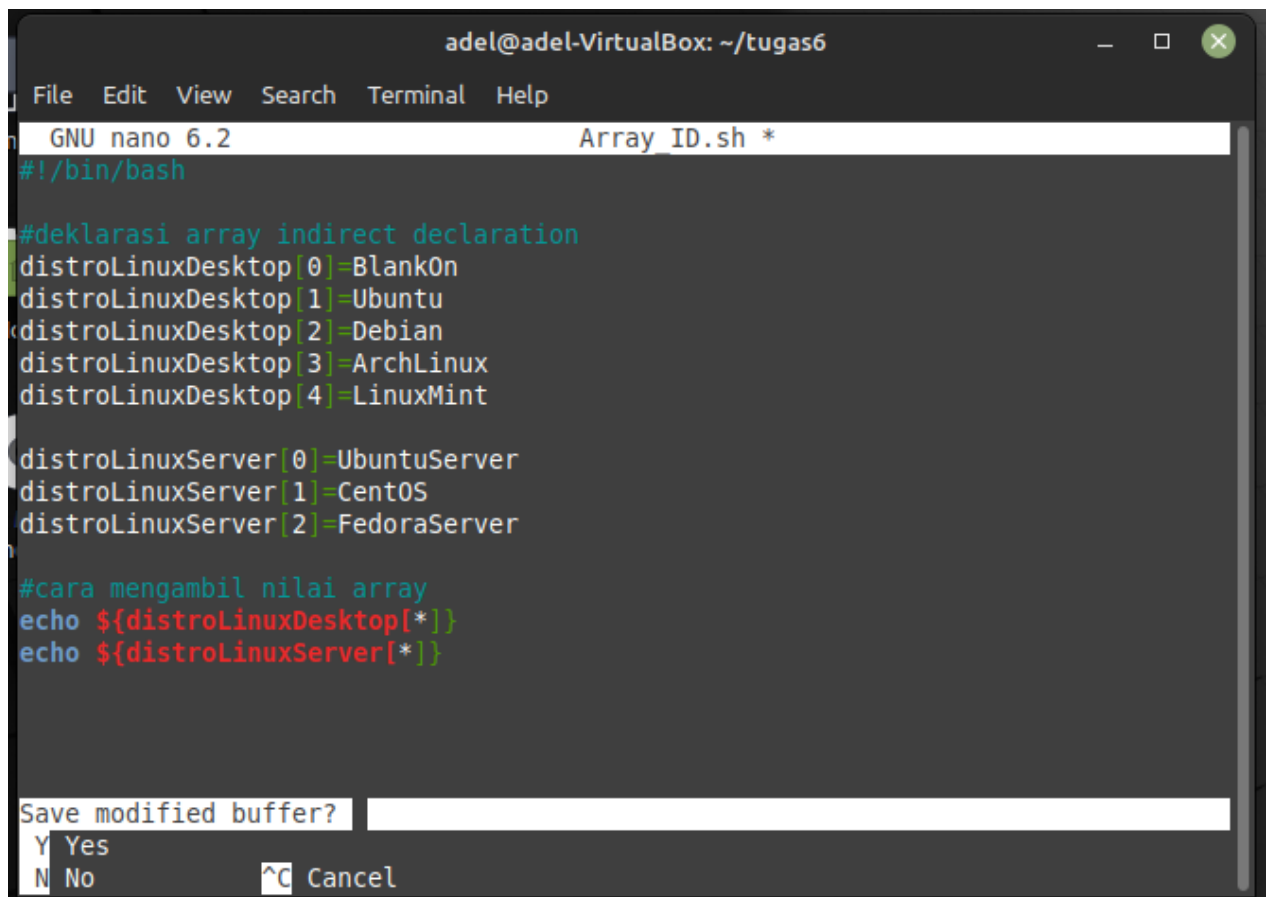
Array in Linux Shell Scripting

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. Variabel yang disimpan disebut sebagai elemen dan variabel yang menyimpan adalah Array. Lebih jelasnya, array bisa dibayangkan sebagai telur dalam karton, telur sebagai elemen di mana kotaknya adalah array.

Tidak ada batasan dalam menetapkan sejumlah elemen di dalam array. Elemen array direferensikan oleh nomer indeks, yang biasanya dimulai dengan nol. Array digunakan dalam mengimplementasikan struktur data, yang merupakan pendekatan untuk mengatur dan mengelola data secara efektif. Array lebih fungsional daripada struktur lainnya.

A. Indirect Declaration

Indirect Declaration atau array terindeks adalah bentuk utama dari array yang menyimpan elemen yang direferensikan melalui nomor indeks mulai dari 0. Berikut ialah contoh dari *Indirect Declaration* :



```
adel@adel-VirtualBox: ~/tugas6
File Edit View Search Terminal Help
GNU nano 6.2 Array ID.sh *
#!/bin/bash

#deklarasi array indirect 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

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

Save modified buffer?
Y Yes
N No ^C Cancel
```

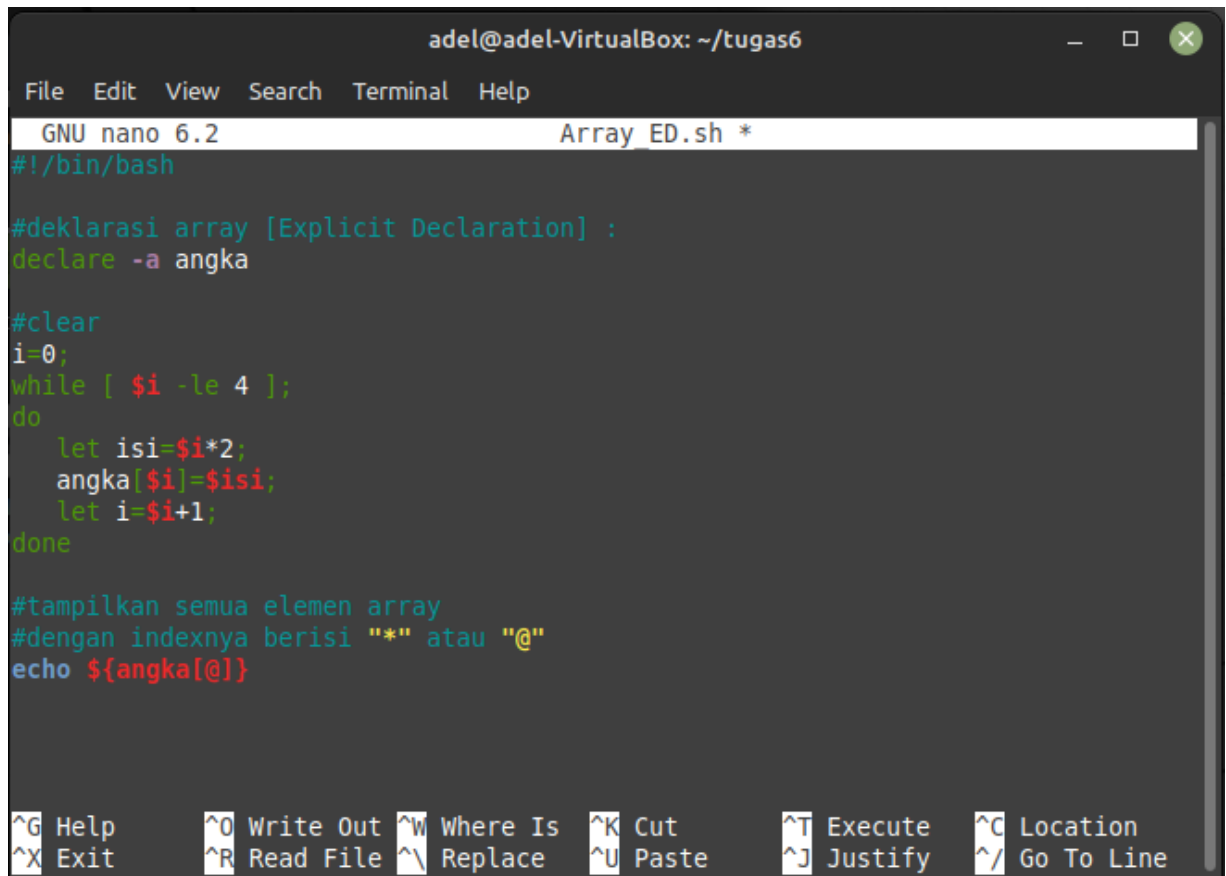
Dalam contoh di atas, `distroLinuxDesktop` dan `distroLinuxServer` adalah sebuah array. Kemudian, `BlankOn`, `Ubuntu`, `Debian`, `ArchLinux`, `LinuxMint` adalah elemen `distroLinuxDesktop` dan `UbuntuServer`, `CentOS`, `FedoraServer` adalah elemen `distroLinuxServer`. Panjang array `distroLinuxDesktop` adalah 5 dan pada array `distroLinuxServer` adalah 3. Kemudian, nomor indeks dimulai dari 0 seperti pada elemen `blankOn` pada array `distroLinuxServer`. Berikut ketika script pemrograman tersebut dieksekusi:

```
adel@adel-VirtualBox:~/tugas6$ nano Array_ID.sh
adel@adel-VirtualBox:~/tugas6$ bash Array_ID.sh
BlankOn Ubuntu Debian ArchLinux LinuxMint
UbuntuServer CentOS FedoraServer
```

Dengan perintah script **echo \${namaArray[*]}** dapat menampilkan seluruh elemen dari array tersebut. Maka dari itu, ketika script dieksekusi, seluruh elemen dari array akan ditampilkan.

B. Explicit Declaration

Explicit Declaration adalah mendeklarasikan array dengan perintah **declare -a Array_name** kemudian menetapkan nilainya. Berikut ialah contoh dari Explicit Declaration:



```
adel@adel-VirtualBox: ~/tugas6
File Edit View Search Terminal Help
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

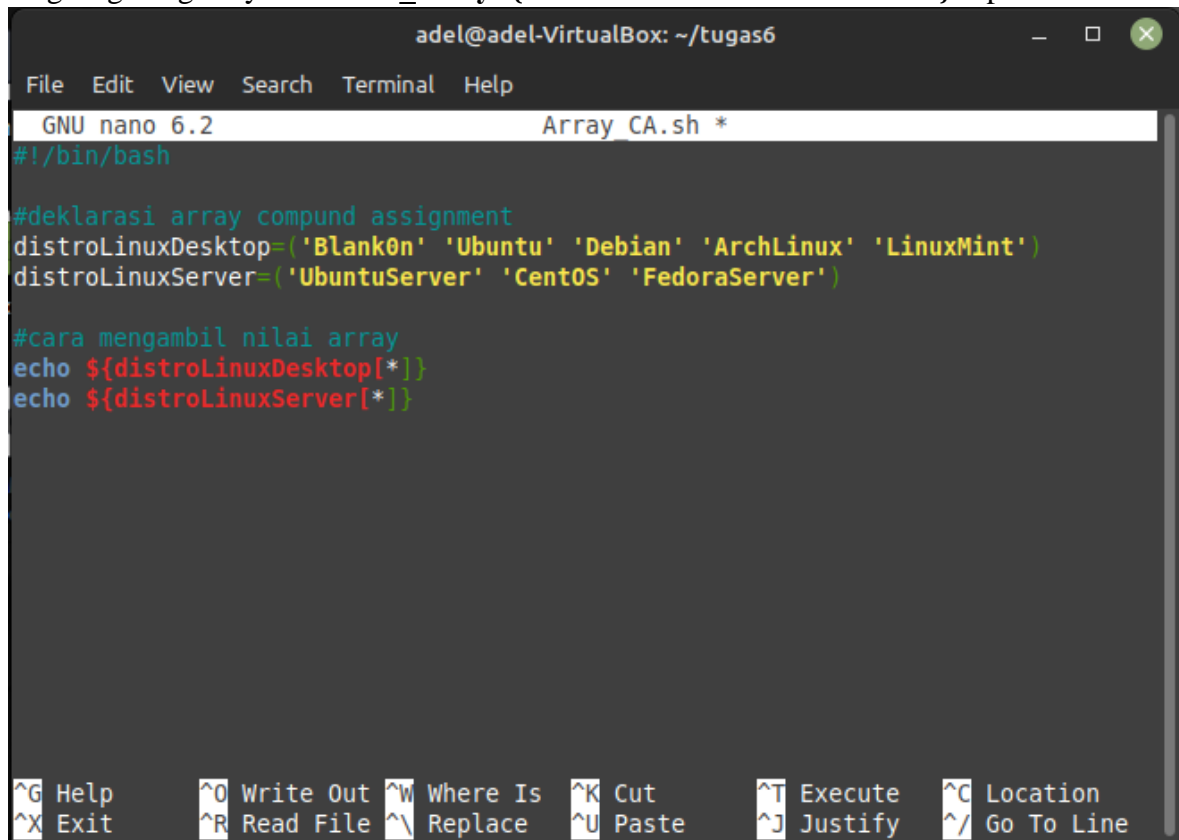
#tampilkan semua elemen array
#dengan indexnya berisi "*" atau "@"
echo ${angka[@]}
```

Dalam contoh di atas, mendeklarasikan variabel angka dengan perintah **declare -a angka**. Kemudian memasukkan nilai dengan memanfaatkan perintah while-do. Pertama, mendeskripsikan variabel i bernilai 0. Kemudian menggunakan perintah while dengan kondisi i kurang dari sama dengan 4, melakukan perintah i dikalikan dengan 2 dimasukkan ke dalam variabel isi. Kemudian memasukkan nilai ke array dengan **angka[\$i]=\$isi**. Untuk menampilkan seluruh elemen dari array menggunakan perintah **echo \${angka[@]}**. Berikut ialah tampilan dari script tersebut ketika di eksekusi : menampilkan 0 2 4 6 8 sebagai perkalian dari i dengan 2 pada perintah **let isi = \$i*2**.

```
adel@adel-VirtualBox:~/tugas6$ nano Array_ED.sh
adel@adel-VirtualBox:~/tugas6$ bash Array_ED.sh
0 2 4 6 8
```

C. Compound Assignment

Compound Assignment adalah mendeklarasikan array dengan sekelompok nilai secara langsung dengan syntax **nama_array={‘elemen1’ ‘elemen2’ ‘elemen3’}** seperti berikut:



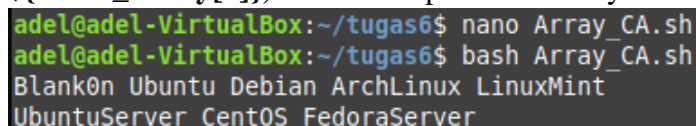
```
adel@adel-VirtualBox: ~/tugas6
File Edit View Search Terminal Help
GNU nano 6.2 Array_CA.sh *
#!/bin/bash

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

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

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location
^X Exit      ^R Read File ^_ Replace   ^U Paste     ^J Justify  ^_ Go To Line
```

Pada contoh tersebut terdapat array `distroLinuxDesktop` dan `distroLinuxServer` seperti sebelumnya. Array tersebut dideklarasikan secara langsung dengan compound assignment yang tiap elemennya diapit dengan tanda petik dan seluruh elemennya terdapat di dalam tanda kurung kurawal. Kemudian menampilkan elemen array dengan perintah **echo \${nama_Array[*]}**, berikut tampilan setelah syntax dieksekusi:

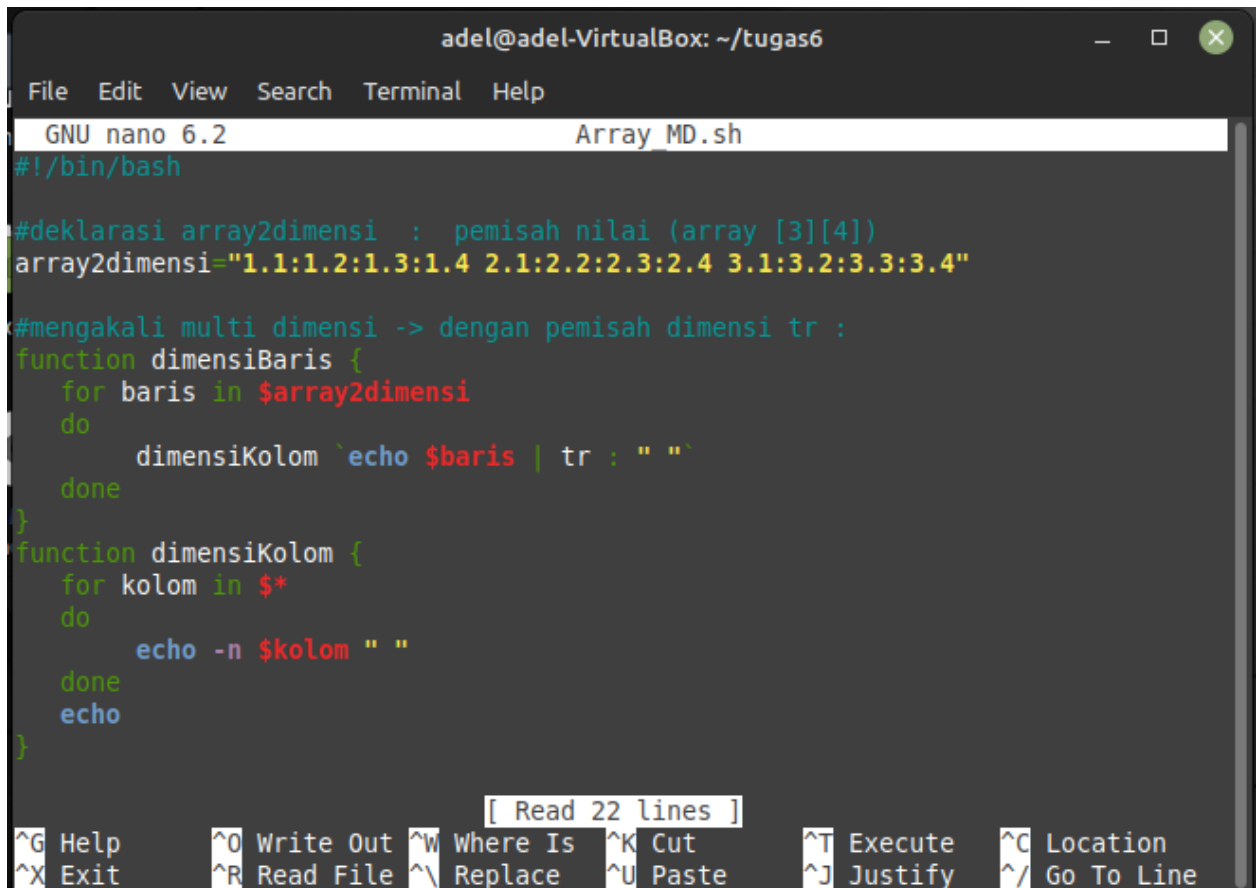


```
adel@adel-VirtualBox:~/tugas6$ nano Array_CA.sh
adel@adel-VirtualBox:~/tugas6$ bash Array_CA.sh
BlankOn Ubuntu Debian ArchLinux LinuxMint
UbuntuServer CentOS FedoraServer
```

D. 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 built-in akan secara eksplisit mendeklarasikan array, dll. Tetapi dapat mensimulasikan efek yang agak mirip dengan multi array asosiatif dimensi.

Dalam hal ini, mendeklarasikan array dengan compound assignment dan elemen dipisahkan dengan tanda “:” dan “ “. Lalu kemudian membuat dua fungsi yaitu fungsi `dimensiBaris` dan fungsi `dimensiKolom`. Pada fungsi `dimensiBaris` memanfaatkan perintah **tr** : untuk memisahkan nilai array kemudian dimasukkan ke dalam `dimensiKolom`. Kemudian pada fungsi `dimensiKolom` menambahkan spasi di antara nilai array.



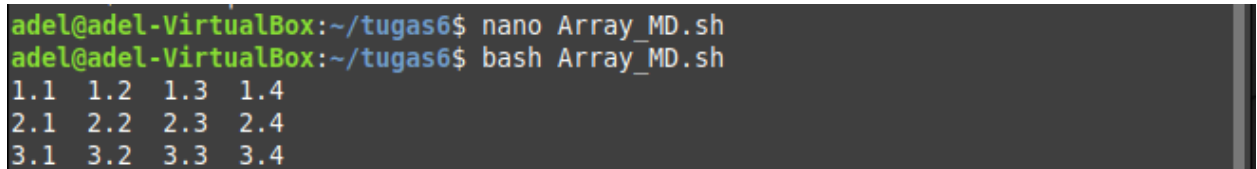
```
adel@adel-VirtualBox: ~/tugas6
File Edit View Search Terminal Help
GNU nano 6.2 Array MD.sh
#!/bin/bash

#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
}

[ Read 22 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

Berikut ialah tampilan ketika syntax di atas telah dieksekusi :



```
adel@adel-VirtualBox:~/tugas6$ nano Array_MD.sh
adel@adel-VirtualBox:~/tugas6$ 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
```

LATIHAN SOAL

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

1. User input data array IPSMahasiswa[index]
2. $IPK = (\text{jumlah nilai IPS}) / (\text{jumlah data IPS})$

JAWABAN

Membuat array IPSMahasiswa dengan user input, kemudian menghitung nilai IPK dengan rumus jumlah nilai IPS dibagi jumlah data IPS. Maka tentunya yang pertama perlu dilakukan adalah membaca input sebagai semester dengan perintah **read semester**. Kemudian mendeklarasikan array atau *explicit declaration* dengan perintah **declare -a IPSmahasiswa** karena nama array-nya adalah IPSMahasiswa. Kemudian, mendeklarasikan nilai 0 pada i dan jumlah. Memanfaatkan perintah while-do dengan kondisi i kurang dari user input semester. Kemudian mendeklarasikan nilai i+1 sebagai urutan semester melalui perintah **let urutansemester=\$i+1**. Membaca user input sebagai nilai tiap semester melalui perintah **read nilaisemester**. Kemudian mendeklarasikan array IPSMahasiswa indeks ke i dengan nilai dari user input nilaisemester melalui perintah **IPSMahasiswa[\$i]=\$nilaisemester**. Kemudian menjumlahkan tiap anggota array IPSMahasiswa dengan perintah **let jumlah=\$jumlah+\${IPSMahasiswa[i]}**. Selanjutnya menghitung IPK dengan perintah **let IPK=\$jumlah/\$semester**. Berikut ialah script bahasa bash untuk menghitung nilai IPS:

```
#!/bin/bash
#Memasukkan jumlah semester
echo "Jumlah semester yang telah ditempuh : "
read semester
#Deklarasi array [Explicit Declaration]
declare -a IPSMahasiswa
#Memanfaatkan perintah while-do
i=0
jumlah=0
while [ $i -lt $semester ];
do
    let urutansemester=$i+1
    echo "Semester " $urutansemester " : ";
    read nilaisemester;
    IPSMahasiswa[$i]=$nilaisemester;
    let jumlah=$jumlah+${IPSMahasiswa[i]};
    let i=$i+1;
done
#Menghitung IPK
let IPK=$jumlah/$semester
#Menampilkan IPS dan IPK mahasiswa
echo "IPK mahasiswa = " $jumlah "/" $semester
echo "IPK mahasiswa = " $IPK
```

```
adel@adel-VirtualBox: ~/tugas6
File Edit View Search Terminal Help
GNU nano 6.2 Tugas6.sh *
#!/bin/bash
#Memasukkan jumlah semester
echo "Jumlah semester yang telah ditempuh : "
read semester

#Deklarasi array [Explicit Declaration]
declare -a IPSMahasiswa

#Memanfaatkan perintah while-do
i=0
jumlah=0
while [ $i -lt $semester ];
do
    let urutansemester=$i+1
    echo "Semester " $urutansemester " : ";
    read nilaisemester;
    IPSMahasiswa[$i]=$nilaisemester;
    let jumlah=$jumlah+${IPSMahasiswa[i]};
    let i=$i+1;
done

#Menghitung IPK
let IPK=$jumlah/$semester

#Menampilkan IPS dan IPK mahasiswa
echo "IPK mahasiswa = " $jumlah "/" $semester
echo "IPK mahasiswa = " $IPK
```

Berikut ialah tampilan hasil eksekusi syntax dengan jumlah 3 semester, nilai semester pertama adalah 4, semester kedua ialah 2, dan semester ketiga mendapatkan nilai 3. Nilai IPS yang didapatkan adalah 3.

```
adel@adel-VirtualBox:~/tugas6$ bash Tugas6.sh
Jumlah semester yang telah ditempuh :
3
Semester 1 :
4
Semester 2 :
2
Semester 3 :
3
IPK mahasiswa = 9 / 3
IPK mahasiswa = 3
```