

- ✓ Merupakan kumpulan variabel dengan tipe sejenis yang mampu menampung banyak nilai dan didefinisikan sebagai sebuah nama variabel
- √ Sintaks

```
Array=(elemen1 elemen2 ... elemenN)
```

Contoh

hari=(Senin,Selasa,Rabu,Kamis,Jumat);
echo \${hari[*]};
artinya:

hari[0] berisi Senin

hari[1] berisi Selasa

hari[2] berisi Rabu

hari[3] berisi Kamis

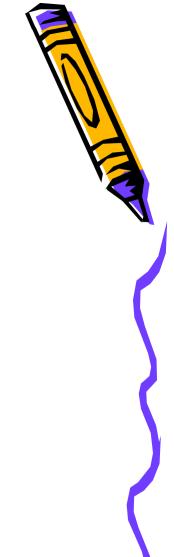
hari[4] berisi Jumat

Deklarasi array

- Menggunakan statement declare
- Sintaks:

declare -a nama variabel_array

```
Contoh 1:
     declare -a angka
     i=0;
     while [$i -le 4];
     do
     let isi=$i*2;
     angka[$i]=$isi;
     let i=$i+1;
     done
     echo ${angka[*]};
```

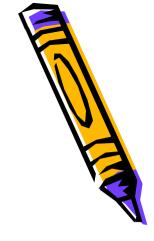


Contoh 2: Pemakaian Sederhana Array

```
#!/bin/bash
a[11]=23
a[13]=37
a[51]=UFOs
echo -n "a[11] = "
echo ${a[11]}
                              # perlu {kurung}.
echo -n "a[13] = "
echo ${a[13]}
echo "isi a[51] adalah ${a[51]}."
# Isi variabel array yang tidak diinisialisasi adalah kosong (null variable).
echo -n "a[43] = "
echo ${a[43]}
echo "(a[43] tidak dialokasikan)"
```



Jumlah dua variabel array dialokasikan ke variabel ketiga a[5]=`expr \${a[11]} + \${a[13]}` echo "a[5] = a[11] + a[13]" echo -n "a[5] = " echo \${a[5]}

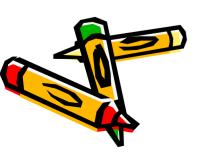


```
a[6]= expr \{a[11]\} + \{a[51]\}
echo "a[6] = a[11] + a[51]"
echo -n "a[6] = " echo ${a[6]}
# Gagal karena tidak boleh menambahkan suatu integer ke string.
# Array lain, "a2"
# Cara lain mengalokasikan variabel array ...
# nama_array=( XXX YYY ZZZ ... )
a2=( nol satu dua tiga empat )
echo -n "a2[0] = "
echo ${a2[0]}
# Pengindeksan mulai nol (elemen pertama array adalah [0], bukan [1]).
echo -n "a2[1] = "
echo ${a2[1]}
                          #[1] adalah elemen array kedua
```

```
# Array lain, "a3".
# Cara lain pendefinisian variabel array ...
# nama_array=([xx]=XXX [yy]=YYY ...)
a3=([17]=tujuh-belas [24]=dua-puluh-empat)
echo -n "a3[17] = "
echo ${a3[17]}
echo -n "a3[24] = "
echo ${a3[24]}
exit 0
```



Contoh 3:



Contoh 4: Bermacam Operasi Array

```
#1/bin/bash
# array-ops.sh: More fun with arrays.
array=(nol satu dua tiga empat lima) # Elemen 0 1 2 3 4 5
echo ${array[0]}
                           # nol
echo ${array:0}
                           # nol
                           # ekspansi parameter elemen pertama yang
                           #+ dimulai pada posisi # 0 (karakter pertama).
                           # ol
echo ${array:1}
                           # ekspansi parameter elemen pertama yang
                           #+ dimulai pada posisi # 1 (karakter kedua).
echo ${#array[0]}
                           #3
                           # Panjang elemen pertama array.
echo ${#array}
                           #3
                           # Panjang elemen pertama array.
```

(alternatif notasi)

```
echo ${#array[1]}
                           #4
                           # Panjang elemen kedua array.
echo ${#array[*]}
                           #6
                           # Jumlah elemen di array.
                           #6
echo ${#array[@]}
                           # Jumlah elemen di array.
array2=([0]="elemen pertama" [1]="elemen kedua" [3]="element ke-empat")
# Tanda petik membolehkan penambahan spasi antara elemen array.
echo ${array2[0]}
                           # elemen pertama
echo ${array2[1]}
                           # elemen kedua
echo ${array2[2]}
                           # Tidak diinisialisasi, karena itu null.
echo ${array2[3]}
                           # elemen ke-empat
          echo ${#array2[0]}
                                    # 14 (panjang elemen pertama)
          echo ${#array2[*]}
                                    #3 (jumlah elemen di array)
```

Contoh 5: Operasi String pada Array

echo \${arrayZ[@]:1:2} # two three # hanya dua elemen setelah elemen[0].

Penghapusan Substring
menghapus elemen terpendek yang cocok dari awal string.

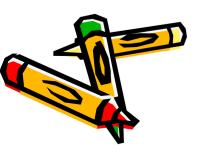
echo \${arrayZ[@]#f*r} # one two three five five # diterapkan untuk semua elemen array. # cocok dengan "four" dan menghapusnya.

```
# cocok terpanjang dari awal string
echo ${arrayZ[@]##t*e} # one two four five five
                          # diterapkan ke semua elemen array.
                          # cocok dengan "three" dan menghapusnya.
# cocok terpendek dari akhir string
echo ${arrayZ[@]%h*e} # one two t four five five
                          # diaplikasikan ke semua elemen array.
                          # cocok "hree" dan menghapusnya.
# cocok terpanjang dari akhir string
echo ${arrayZ[@]%%t*e} # one two four five five
                          # diaplikasikan ke semua elemen array.
                          # cocok "three" dan menghapusnya.
# Penggantian Substring
# Mengganti substring.
echo ${arrayZ[@]/fiv/XYZ}
                                # one two three four XYZe XYZe
                                # diterapkan ke semua elemen array.
                                # mengganti semua kejadian substring.
```

```
# one two three four fyye fyye
echo ${arrayZ[@]//iv/YY}
                                # diaplikasikan ke semua elemen array
# menghapus semua substring yang sesuai.
                                # one two three four ve ve
echo ${arrayZ[@]//fi/}
                                # diaplikasikan ke semua elemen array.
# mengganti awal-akhir substring yang sesuai
echo ${arrayZ[@]/#fi/XY}
                                # one two three four XYve XYve
                                # diaplikasikan ke semua elemen array.
# mengganti akhir substring yang sesuai.
                                # one two three four fiZZ fiZZ
echo ${arrayZ[@]/%ve/ZZ}
                                # diterapkan ke semua elemen array.
```

mengapa?

one twXX three four five five



echo \${arrayZ[@]/%o/XX}

Soal Latihan

- 1. Buatlah program array untuk mencetak dan menampilkan bilangan cacah kelipatan 5 yang kurang dari 100.
- 2. Buatlah program dengan menggunakan array yang mengurutkan 5 angka yang dimasukkan secara acak.

