

## MODUL V

### ARRAY

#### 5.1 TUJUAN

1. Memahami struktur Array (Larik) dalam program,
2. Mampu menerapkan Array untuk menyelesaikan berbagai kasus.

#### 5.2 DASAR TEORI

Array adalah kumpulan data untuk menyimpan item bertipe data sama. Array digunakan untuk menampung dan memproses data nilai lebih dari satu. Jika kita membutuhkan 5 data nilai atau lebih untuk diproses dan kita menggunakan variabel biasa, maka memerlukan 5 variabel. Penggunaan 5 variabel tersebut, akan memboroskan memori dan penulisan *code*. Untuk mengatasinya kita dapat menggunakan Array, untuk menyimpan 5 data nilai tersebut. Nilai variabel array bisa bertipe data primitif seperti int, string, boolean, float, double, ataupun bentuk lainnya. Hal yang membedakan antara nilai yang satu dengan nilai yang lain adalah index dari pada array sendiri. Kata index bisa dianalogikan sebagai nomor rumah dari sebuah perumahan. Tentunya setiap rumah ditempati oleh orang yang berbeda. Jadi, index merupakan lokasi data yang akan disimpan.

##### A. Definisi Array Pada Python

Array merupakan sekumpulan variabel yang memiliki tipe data yang sama dan mampu menyimpan beberapa nilai dalam satu variabel. Array merupakan tipe data terstruktur dalam pemrograman, array memungkinkan untuk menyimpan data maupun referensi objek dalam jumlah banyak dan terindeks. Sehingga sejumlah variabel dapat menggunakan nama yang sama.

##### B. Cara Mendefinisikan Sebuah Array

Untuk mendefinisikan sebuah array di python bisa menggunakan keyword ***nama\_array = [jumlah\_elemen]***. Contoh mendefinisikan suatu array:

```
#inisialisasi variabel berformat array
```

```
Angka = [ 1, 2, 3 ,4 ]
```

```
Text = ["Satu", "Dua", "Tiga","Empat"]
```

```
print(Angka) print(Text)
```

Outputnya adalah:

```
[1, 2, 3, 4]  
['Satu', 'Dua', 'Tiga', 'Empat']
```

### C. Metode Array

Python memiliki seperangkat metode bawaan yang dapat Anda gunakan pada List/ array.

Method	Deskripsi
<code>append()</code>	Menambahkan elemen di akhir list
<code>clear()</code>	Menghapus semua elemen dari list
<code>copy()</code>	Mengembalikan salinan list
<code>count()</code>	Mengembalikan jumlah elemen dengan nilai yang ditentukan
<code>extend()</code>	Tambahkan elemen list (atau iterable), ke akhir list saat ini
<code>index()</code>	Mengembalikan indeks elemen pertama dengan nilai yang ditentukan
<code>insert()</code>	Menambahkan elemen pada posisi yang ditentukan
<code>pop()</code>	Menghapus elemen pada posisi yang ditentukan
<code>remove()</code>	Menghapus item dengan nilai yang ditentukan
<code>reverse()</code>	Membalik urutan list
<code>sort()</code>	Mengurutkan list

### D. Cara menggunakan Array

#### 1. Menampilkan nilai array

Cara merujuk ke elemen array adalah dengan mengacu pada nomor indeks.

Contoh: Tampilkan nilai item array kedua

```
buah = ["Apel", "Jeruk", "Anggur", "Pisang"] nilai  
= buah[1] print(nilai)
```

Outputnya adalah: Jeruk

## 2. Merubah nilai item array

Contoh: Merubah item kedua yaitu Jeruk

```
# urutan buah asal buah = ["Apel", "Jeruk",  
"Anggur", "Pisang"] print(buah)  
buah[1] = "Melon"  
  
# urutan buah yang sudah dirubah nilainya print(buah)
```

Outputnya adalah:

```
['Apel', 'Jeruk', 'Anggur', 'Pisang']  
['Apel', 'Melon', 'Anggur', 'Pisang']
```

## 3. Mengetahui Panjang array

Gunakan metode len() untuk mengembalikan panjang array (jumlah elemen dalam array). Contoh:

```
buah = ["Apel", "Jeruk", "Anggur", "Pisang"]  
panjang = len(buah) print(panjang)
```

Outputnya adalah: 4

## 4. Looping elemen array

Kita dapat menggunakan for in loop untuk melakukan loop melalui semua elemen array.

Contoh:

```
buah = ["Apel", "Jeruk", "Anggur", "Pisang"]  
for x in buah: print(x)
```

Outputnya adalah:

```
PROBLEMS OUTPUT
PS C:\Users\SISFO :
Apel
Jeruk
Anggur
Pisang
```

#### 5. Menambah elemen array

Kita bisa menggunakan metode `append()` untuk menambahkan elemen ke array.

Contoh:

```
buah = ["Apel", "Jeruk", "Anggur", "Pisang"]
buah.append("Semangka") print(buah)
```

Outputnya adalah:

```
['Apel', 'Jeruk', 'Anggur', 'Pisang', 'Semangka']
```

#### 6. Menghapus elemen array

Kita bisa menggunakan metode `pop()` atau `remove()` untuk menghapus elemen dari array. Contoh:

```
buah = ["Apel", "Jeruk", "Anggur", "Pisang"]
buah.pop(0) buah.remove("Anggur") print(buah)
```

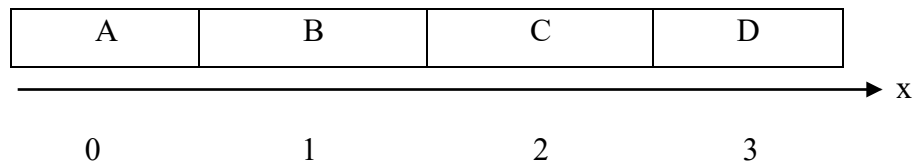
Outputnya adalah:

```
['Jeruk', 'Pisang']
```

## E. Array 1 Dimensi

Array 1 dimensi merupakan array yang hanya memiliki 1 baris saja dengan banyak kolom. Secara default, array akan dimulai dari indeks ke-0. Sehingga ketika kita mendeklarasikan array dengan ukuran  $n$ , maka indeksnya adalah dari 0 sampai  $n-1$ . Berikut ilustrasi yang dapat dilihat.

Ukuran array = 4



Contoh array 1 dimensi :

```
buah = ["Apel", "Jeruk", "Anggur", "Pisang"]  
for x in buah:  
    print(x)
```

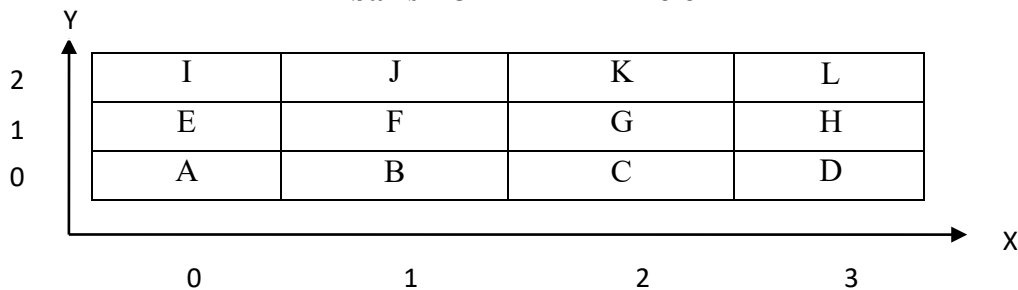
Outputnya adalah:

```
PROBLEMS  OUTPUT  
PS C:\Users\SISFO :  
Apel  
Jeruk  
Anggur  
Pisang
```

## F. ARRAY 2 DIMENSI

Array 2 dimensi merupakan array yang terdiri dari baris dan kolom. Array 2 dimensi dapat dianalogikan sebagai matriks terdiri lebih dari 1 baris dan lebih dari 1 kolom. Sama halnya dengan array 1 dimensi, array 2 dimensi hanya dapat menampung data dengan tipe data yang sama. Pada array ini, ketika kita akan memproses data di dalam array 2 dimensi, maka perlu mendeklarasikan baris dan kolom yang akan diproses. Berikut ilustrasi dari array 2 dimensi.

Ukuran **baris** = 3 && Ukuran **kolom** = 4



2	I	J	K	L
1	E	F	G	H
0	A	B	C	D

0                      1                      2                      3

Contoh array 2 dimensi:

```
buah = [ ["Apel", "Jeruk", "Anggur", "Pisang"], ["Nanas",  
"Melon", "Manggis", "Sawo"] ]  
for i in range(len(buah)):    for j in  
range(len(buah[i])):  
    print(buah[i][j])
```

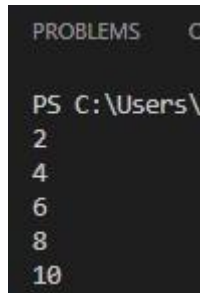
Outputnya adalah:

```
PROBLEMS  OUTPUT  
  
PS C:\Users\SISFO  
Apel  
Jeruk  
Anggur  
Pisang  
Nanas  
Melon  
Manggis  
Sawo
```

### 5.3 LATIHAN

- Buatlah sebuah program untuk menampilkan bilangan genap dari 1-10 dari data array.

Contoh Output :



```
PROBLEMS C
PS C:\Users\
2
4
6
8
10
```

Code program

```
bilangan = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
for x in bilangan:
    if x % 2 == 0:
        genap = x
        print(genap)
```

## 5.4 TUGAS

- A. Buatlah program untuk mencari data yang ada pada array. Jika ditemukan, maka akan menampilkan pesan ditemukan. Jika tidak ditemukan, maka akan menampilkan pesan tidak ditemukan. Berikut contoh outputnya.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\SISFO ST3T amr> & "C:/Users/S
Masukan jumlah kata: 3
masukan kata: aku
masukan kata: bukan
masukan kata: superman

Masukan kata yang ingin dicari: superman
superman ditemukan pada indeks ke- 2
```

- B. Buatlah sebuah program, untuk menghitung nilai rerata beserta predikatnya dengan persyaratan.

- $100 > \text{Nilai} \geq 90$  Predikat = A
- $90 > \text{Nilai} \geq 70$  Predikat = B
- $70 > \text{Nilai} \geq 50$  Predikat = C
- $50 > \text{Nilai} \geq 30$  Predikat = D
- $30 > \text{Nilai} \geq 0$  Predikat = E
- Selain itu, nilai akan dinyatakan tidak valid Contoh Output :
- Jika nilai Valid

```
Masukkan jumlah Mata Kuliah : 5

Masukkan nilai mata kuliah ke-1 : 100
Masukkan nilai mata kuliah ke-2 : 90
Masukkan nilai mata kuliah ke-3 : 95
Masukkan nilai mata kuliah ke-4 : 80
Masukkan nilai mata kuliah ke-5 : 10

Hasil Predikat B dengan nilai :
Mata kuliah ke-0 : 100.0
Mata kuliah ke-1 : 90.0
Mata kuliah ke-2 : 95.0
Mata kuliah ke-3 : 80.0
Mata kuliah ke-4 : 10.0
```



- Jika nilai tidak Valid

```
Masukkan jumlah Mata Kuliah : 5
Masukkan nilai mata kuliah ke-1 : 80
Masukkan nilai mata kuliah ke-2 : 900
Masukkan nilai mata kuliah ke-3 : 10
Masukkan nilai mata kuliah ke-4 : 100
Masukkan nilai mata kuliah ke-5 : 800
Nilai tidak valid!
```