

## **Modul Pemrograman Lanjut**

**Sevi Nurafni**

**Catatan**

1. Modul ini dirancang untuk dapat menjadi pegangan mata kuliah Pemrograman Lanjut.
2. Anda dapat membuka modul ini saat latihan praktikum.
3. Anda sangat disarankan untuk mencoba menjalankan semua program modul ini di komputer Anda, supaya Anda dapat mengetahui keluaran dari program yang ada.
4. Anda sangat disarankan untuk bereksperimen dari program-program yang ada di modul ini supaya Anda mendapat gambaran lebih jelas mengenai apa yang program Anda lakukan.
5. Anda sangat disarankan membaca tutorial dari tempat lain dan mengeksplor sendiri bahasa yang Anda gunakan.

## Modul 3

### 1. Array

#### 1.1. Definisi Array

Array adalah variabel dengan satu buah nama, tetapi mengandung banyak nilai. Akses nilai-nilainya dilakukan dengan indeks. Perhatikanlah contoh berikut!

Indeks	0	1	2	3	4	5	6	7	8	9
A	3	10	5	7	11	19	23	35	37	12

- Keterangan Tabel:

**A[0] = 3**

**A[1] = 10**

**A[7] = 35**

Pada contoh diatas, kita memiliki sebuah variabel yang bernama A. Variabel A tersebut memiliki 10 buah nilai, dimana nilai-nilai tersebut dapat diakses dengan indeks (mengakses dengan memerhatikan posisi urutan setiap elemen berada, *default indexing* biasanya dimulai dari urutan ke-0). Untuk mengakses indeks ke – x, gunakanlah A[x]. Dan nilai A[x] itu bisa kita anggap sebagai variabel yang berdiri sendiri. Konsep inilah yang kita sebut dengan array. Perhatikan pula bahwa indeks dimulai dari 0.

#### 1.2. Deklarasi Array

Karena array juga merupakan sebuah variabel, maka array juga memerlukan deklarasi seperti variabel lainnya. Contoh deklarasi array:

```
x = [0 for i in range(n)] membuat array berukuran n dengan isi 0.  
x = ["*" for i in range(100)] membuat array berukuran 100 dengan isi karakter "*".
```

Untuk contoh tersebut array A yang terdefinisi adalah **A[0], A[1], A[2], . . . , A[9]**. Mengakses nilai indeks di luar batasan tersebut akan menyebabkan *runtime error*. Oleh karena itu, tentukanlah rentang indeks yang akan digunakan saat deklarasi dengan tepat (sesuai dengan kebutuhan).

#### 1.3. Array dan Variabel

Suatu array dapat kita anggap sebagai variabel, sehingga segala jenis operasi pada variabel juga berlaku pada array. Sebagai contoh, kita memiliki suatu array bernama *tabel* berikut.

```
tabel = [0 for i in range(10)]
```

Maka array *tabel* tersebut akan terdefinisi untuk indeks 0 sampai dengan indeks 9. Maka kita bisa melakukan instruksi ini,

```
tabel[2] = int(input())
```

Jika diberikan 5 buah bilangan, dan kita perlu menyimpan bilangan tersebut pada *tabel*, kita bisa melakukan:

```
tabel[0] = int(input("Masukkan nilai ke -0: "))
tabel[1] = int(input("Masukkan nilai ke -1: "))
tabel[2] = int(input("Masukkan nilai ke -2: "))
tabel[3] = int(input("Masukkan nilai ke -3: "))
tabel[4] = int(input("Masukkan nilai ke -4: "))
```

Namun, cara meng-input tersebut kurang efisien (harus mengetikkan kode, yang sama, berkali-kali). Akan lebih efisien jika kita menginput menggunakan perulangan (LOOPING).

```
for i in range(5):
    tabel[i] = int(input('Masukkan nilai ke-' + str(i) + ': '))
```

#### 1.4.String

Pada modul 1 kita telah mempelajari tipe data *string*. *String* dapat dianggap sebagai array of character, namun tipe data string memiliki karakteristik yang berbeda dengan *array of character* yaitu *string* merupakan tipe data *immutable*. Artinya, *string* tidak dapat diubah nilainya setelah dideklarasikan. Sebagai contoh, jika kita memiliki *string* s1:

```
s1 = "Hello"
```

Kita tidak dapat mengupdate nilai s1, misal melakukan *assignment* s1[1] = 'a' akan menyebabkan *error*. Berbeda halnya dengan *array of character*, kita dapat mengubah nilai *array of character* setelah dideklarasikan. Sebagai contoh, jika kita memiliki *array of character* s2:

```
s2 = ['H', 'e', 'l', 'l', 'o']
```

Kita dapat mengubah nilai s2, misal melakukan *assignment* s2[1] = 'a' akan mengubah nilai s2 menjadi ['H', 'a', 'l', 'l', 'o']. Namun, kita dapat mengakses nilai *string* dengan indeks, misal s1[1] akan bernilai 'e'.

### Latihan Modul 3

#### 1. Reverse

Buatlah program yang menerima N buah bilangan dan menuliskan kembali bilangan-bilangan tersebut, namun dengan urutan terbalik.

Contoh:

Masukkan N: <u>5</u>
<u>5</u>
<u>2</u>
<u>1</u>
<u>6</u>
<u>3</u>
Hasil dibalik: 3 6 1 2 5

#### 2. Anagram

Diberikan dua array A dan B. Buatlah sebuah program yang dapat memeriksa apakah B merupakan anagram dari A. Asumsikan elemen pada array A dan B maksimal 10. Array B didefinisikan sebagai anagram dari array A bila elemen-elemen di array A dapat ditukar-tukar sehingga menghasilkan array B.

😊 Hint: gunakan tabel frekuensi, yakni sebuah array yang indeks ke-1 menunjukkan banyaknya bilangan 1, indeks ke-2 menunjukkan banyaknya bilangan 2, dan seterusnya.

##### Contoh 1

Masukkan banyaknya elemen A: 3 Masukkan elemen A ke -1: 8
Masukkan elemen A ke -2: 4
Masukkan elemen A ke -3: 3
Masukkan banyaknya elemen B: 3
Masukkan elemen B ke -1: 4
Masukkan elemen B ke -2: 8
Masukkan elemen B ke -3: 3
B adalah anagram dari A

##### Contoh 2

Masukkan banyaknya elemen A: 2
Masukkan elemen A ke -1: 1
Masukkan elemen A ke -2: 4
Masukkan banyaknya elemen B: 2
Masukkan elemen B ke -1: 1
Masukkan elemen B ke -2: 5
B bukan anagram dari A

### 3. Palindrom

Buatlah program yang menerima sebuah string dan menuliskan apakah string tersebut palindrom. Asumsikan string hanya berisi huruf kecil (a-z).

Contoh 1:

```
Masukkan panjang string: 5
Masukkan string: makan

makan bukan palindrom
```

Contoh 2:

```
Masukkan panjang string: 10
Masukkan string: kasur rusak

kasur rusak adalah palindrom
```

Contoh 3:

```
Masukkan panjang string: 3
Masukkan string: apa

apa adalah palindrom
```