

**MODUL VI**  
**STRUKTUR PENGULANGAN**  
**PRAKTIKUM PROGRAM KOMPUTER TI 20 - UNS**

---

### **Struktur Pengulangan**

Pengulangan dalam bahasa pemrograman berfungsi memerintahkan komputer melakukan sesuatu secara berulang-ulang. Terdapat dua jenis pengulangan dalam bahasa pemrograman python, yaitu pengulangan dengan for dan while.

Pengulangan for disebut *counted loop* (pengulangan yang terhitung), sementara pengulangan while disebut *uncounted loop* (pengulangan yang tak terhitung). Perbedaannya adalah pengulangan for biasanya digunakan untuk mengulangi kode yang sudah diketahui banyak pengulangannya. Sementara while untuk pengulangan yang memiliki syarat dan tidak tentu berapa banyak pengulangannya.

#### **1. Struktur pengulangan menggunakan perintah While**

Python menyediakan perintah while untuk melakukan pengulangan satu atau sekelompok perintah. While pada bahasa Python digunakan untuk mengeksekusi statement selama kondisi yang diberikan terpenuhi (True). Bentuk umum penggunaan perintah while didalam Python :

```
While kondisi:  
    Perintah1  
    Perintah2  
    ...
```

Perintah-perintah tersebut akan terus di eksekusi/ diulang selama kondisi bernilai TRUE. Maka dari itu, harus ada suatu perintah yang digunakan untuk mengubah nilai kondisi menjadi FALSE agar proses pengulangan dapat berhenti.

#### **Exercise 6.1**

```
#menentukan banyak pengulangan
n=5

#melakukan pengulangan
i = 1
while i ≤ n:
    print(i)
    i = i + 1 # bisa ditulis: i += 1
```

Pada program diatas, dua perintah yang akan diulang yaitu print(i) dan i = i + 1. Perintah tersebut akan dieksekusi/diulang sebanyak 5 kali

### Exercise 6.2

```
#input untuk nilai n
n = int(input("masukan banyak pengulangan : "))

#melakukan pengulangan
i = 1
while i ≤ n:
    print(i)
    i = i + 1
```

Program diatas menunjukan banyaknya pengulangan akan ditentukan oleh user melalui proses masukan dari keyboard.

### Exercise 6.3

```
#melakukan pengulangan
i = 1
while i ≤ 10:
    print("Baris ke-%d: Hello World" % i)
    i = i + 1
```

Program diatas berisi perintah mencetak teks “Hello world” ke layar sebanyak 10 kali, mulai dari indeks ke-1 sampai ke-10

### Exercise 6.4

```
#menentukan banyak pengulangan
n = 5
#melakukan pengulangan
i = n
while i ≥ 1:
    print(i)
    i = i - 1
```

Program diatas merupakan perintah pengulangan secara menurun(descending)

## 2. Struktur pengulangan menggunakan perintah For

Perintah for dirancang untuk mengambil atau menelusuri elemen yang terdapat pada tipe-tipe koleksi seperti string, list, tuple, dictionary, set, dan objek bertipe range. Bentuk umum perintah for dapat dituliskan seperti berikut:

Perintah for untuk penelusuran elemen didalam tipe koleksi:

```
for elemen in tipe_koleksi :
    perintah1
    perintah2
    ...
```

Perintah for dengan indeks yang dihasilkan oleh fungsi range()

```
for indeks in range(nilai_awal, nilai_akhir, step):
    perintah1
    perintah2
    ...
```

### Exercise 6.5

```
#menggunakan for pada tipe string
for ch in "Praktikum":
    print(ch)
```

Program diatas merupakan pengulangan yang mengambil karakter dari suatu string. Proses akan dihentikan ketika semua karakter dalam string “Praktikum” sudah diambil

### Exercise 6.6

```
#menggunakan for pada tipe list
for matkul in ["Kalkulus", "Fisika", "Kimia"]:
    print(matkul)
```

Program diatas merupakan pengulangan yang mengambil karakter dari suatu list. Perintah for akan mengambil setiap nilai atau elemen yang terdapat di dalam list.

#### Exercise 6.7

```
#membuat for untuk rentang nilai tertentu
for i in range(2, 9): # lakukan pengulangan mulai dari i=2 sampai i < 9
    print("kuadrat dari ", i, " adalah ", i**2)
```

Program diatas merupakan pengulangan menggunakan fungsi range(). Proses pengulangan dilakukan sebanyak 7 kali (berasal dari 9-2).

#### Exercise 6.8

```
for i in range(2, 17, 3): #mulai dari 2 sampai kurang dari 20, step inkremen 3
    print(i, " kuadrat adalah ", i**2)
```

Program diatas merupakan pengulangan menggunakan fungsi range()

### 3. Pengulangan bersarang

Pengulangan bersarang merupakan pengulangan di dalam badan pengulangan lain. Dalam proses ini kita dapat menggunakan perintah while, for maupun kombinasi dari keduanya.

#### Exercise 6.9

```
for i in range(1, 11):
    for j in range(1, i + 1):
        print('%d ' % (i*j), end='')
    print()
```

Program diatas merupakan contoh pengulangan bersarang menggunakan perintah for. Setiap langkah pada pengulangan bagian luar(pertama), program akan mengeksekusi pengulangan bagian dalam (kedua).

#### Exercise 6.10

```

i = 1
while i <= 10:
    j = 1
    while j <= i:
        print("%d " % (i*j), end= '')
        j += 1
    print()
    i += 1

```

Program diatas merupakan contoh pengulangan bersarang menggunakan perintah while.

#### 4. Pernyataan Loncat

Pernyataan loncat adalah perintah yang digunakan memindahkan eksekusi program dari satu bagian tertentu kebagian lain.

##### 1. Perintah break

Perintah break digunakan untuk menghentikan secara paksa proses pengulangan, meskipun kondisi pengulangan sebenarnya masih bernilai True

##### Exercise 6.11

```

for i in range(11):
    print(i, end='')
    if i == 7:
        break

```

Pada program diatas seharusnya pengulangan dilakukan sebanyak 10 kali, namun karena pada saat i bernilai 7 kita memanggil perintah break, maka pengulangan langsung dihentikan

##### 2. Perintah continue

Perintah continue berguna untuk memaksa pengulangan agar memproses indeks selanjutnya meskipun perintah-perintah didalam blok pengulangan sebenarnya belum selesai dieksekusi.

##### Exercise 6.12

```

for i in range(1,11):
    if i == 5:
        continue
    print(i, ' x ', i, ' = ', i*i)

```

##### 3. Perintah return

Perintah return akan menghentikan proses eksekusi kode di dalam fungsi dan eksekusi akan diarahkan ke baris pemanggil

### Exercise 6.13

```
def tambah(a, b):  
    c = a + b  
    return c  
x= int(input("masukan bilangan ke 1:"))  
y= int(input("masukan bilangan ke 2:"))  
  
hasil = tambah(x,y)  
print("%d + %d = %d" % (x, y, hasil))
```