



## MODUL VI

### Struktur Pengulangan

#### A. Tujuan

Berikut tujuan dari pelaksanaan praktikum struktur pengulangan:

1. Memahami struktur pengulangan menggunakan “while”
2. Memahami struktur pengulangan menggunakan “For”
3. Memahami struktur pengulangan bersarang
4. Memahami pernyataan loncat, break, continue, dan return
5. Memahami penggunaan fungsi sys.exit()

#### B. Struktur Pengulangan

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

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

##### 1. While loop

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 perulangan dapat berhenti.



### Contoh 6.1

```
#CONTOH 6.1
#menentukan banyak pengulangan
n = 9

#melakukan pengulangan
i = 6

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

### Contoh 6.2

```
#CONTOH 6.2
#menentukan banyak pengulangan
n = int(input("input nilai n: "))

#melakukan pengulangan
i = 6

while i <= n :
    print(i)
    i = i + 1
```

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

### Contoh 6.3

```
#CONTOH 6.3
#melakukan pengulangan
i = 6

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.



#### Contoh 6.4

```
#CONTOH 6.4
#menentukan banyak pengulangan
n = 6

#melakukan pengulangan
i = 11

while i >= n :
    print (i)
    i = i - 1 # sesuaikan dengan atas agar fungsi pengulangan dapat berhenti
```

Program diatas merupakan perintah **pengulangan** secara **menurun(*descending*)**

## 2. For Loop

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 di dalam tipe koleksi
for elemen in tipe_koleksi:
    Perintah1
    Perintah2
    ...
```

```
#Perintah "for" dengan indeks yang dihasilkan oleh fungsi range
for index in range(nilai_awal, nilai_akhir):
    Perintah1
    Perintah2
    ...
```

#### Contoh 6.5

```
#CONTOH 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 terambil.**

#### Contoh 6.6

```
#CONTOH 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.

#### Contoh 6.7

```
#CONTOH 6.7
#menggunakan "for" untuk rentang nilai tertentu
for i in range(2,9):
    print("kuadrat dari",i, "adalah", i**2)
```

Program diatas merupakan pengulangan menggunakan fungsi range(). Proses pengulangan dilakukan sebanyak 8 kali (berasal dari 2-8).

#### Contoh 6.8

```
#CONTOH 6.8
#mulai dari angka 2 sampai kurang dari 17, step inkrement 3:
for i in range(2,17,3):
    print("kuadrat dari", 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.

#### Contoh 6.9

```
#CONTOH 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).



### Contoh 6.10

```
#CONTOH 6.10
i = 1
while i <= 10:
    j = 1
    while j<=i:
        print("%d " % (i*j), end = '')
        j = j + 1
    print()
    i+=1
```

Program diatas merupakan contoh pengulangan bersarang menggunakan perintah while.

## 4. Pengulangan 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

### Contoh 6.11

```
#CONTOH 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.



### Contoh 6.12

```
#CONTOH 6.12
for i in range(1,11):
    if i == 5:
        continue
    print(1, ' x ', i, '= ', i*i)
```

### 3. Perintah `return`

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

### Contoh 6.13

```
#CONTOH 6.13
def tambah(a,b):
    c = a+ b
    return c
x = int(input("masukkan bilangan ke-1 : "))
y = int(input("masukkan bilangan ke-2 : "))

hasil = tambah(x,y)
print("%d + %d = %d"% (x,y,hasil))
```