



KONSEP PEMROGRAMAN

[Python 3.6.2]

Abstract

range, for, while, break, pass, continue

Puji Winar Cahyo
github.com/pwcahyo

A. Tujuan Pembelajaran.

Mahasiswa diharapkan dapat :

- 1) Memahami konsep perulangan
- 2) Memahami implementasi range
- 3) Memahami implementasi for, for bersarang
- 4) Memahami implementasi while, while bersarang
- 5) Memahami implementasi break, continue, pass

B. Pengantar

1) range

range merupakan suatu fungsi yang ada didalam bahasa pemrograman python yang sering digunakan untuk menyediakan nilai jarak antara start number sampai stop number dan dipengaruhi dengan step number, kemudian akan menyimpan data hasil range kedalam list.

Format penulisan range:

range(start, stop, step) # garis bawah adalah optional

Contoh 1: (secara standard, hanya menggunakan nilai stop)

```
>>> print(range(3))  
[0,1,2]
```

Contoh 2: (menggunakan inisialisasi nilai start dan stop)

```
>>> print(range(1,3))  
[1,2]
```

Contoh 3: (menggunakan inisialisasi nilai start, stop dan step)

```
>>> print(range(1,3,+2))  
[1]
```

```
>>> print(range(5,1,-2))  
[5,3]
```

2) for

a. konsep sederhana (for)

Didalam pemrograman, perulangan seringkali digunakan untuk menyelesaikan kondisi yang berulang dan kondisi tersebut bisa memiliki pola yang sama ataupun berbeda. Perulangan didalam bahasa pemrograman python secara sederhana biasanya mengakomodasi range dapat dilihat sebagai berikut:

```
>>> for i in range(3):
>>>     print(i)
0
1
2
```

BERBEDA dengan

```
>>> for i in range(1,3):
>>>     print(i)
1
2
```

BERBEDA dengan

```
>>> for i in range(1,3,+2):
>>>     print(i)
1
```

b. perulangan (for) secara bersarang

Didalam pemrograman, perulangan secara bersarang sering dijumpai pada pembentukan matrix (array multidimensi) atau pembentukan pola tertentu. Perulangan secara bertingkat didefinisikan dengan didalam for akan terdapat for atau perulangan lain, seperti contoh berikut ini :

```
>>> for i in range(0,5):
>>>     for j in range(0, i+1):
>>>         print("*", end=" ")
>>>     print()
*
* *
* * *
* * * *
* * * * *
```

c. perulangan (while)

Perulangan while hampir sama dengan for, yaitu untuk mengatasi permasalahan berulang, akan tetapi dengan memanfaatkan suatu kondisi, sampai kondisi yang dimaksud terpenuhi. Didalam bahasa pemrograman, perulangan while dapat dituliskan sebagai berikut.

```
while kondisi:  
    blok_eksekusi
```

contoh pada bahasa pemrograman python:

```
>>> suhu = 0  
>>> while suhu < 100:  
>>>     print(suhu)  
>>>     suhu+=40  
0  
40  
80
```

didalam while biasanya ada counter, counter tersebut digunakan untuk menjalankan blok_eksekusi hingga mencapai kondisi yang dimaksud, seperti contoh perulangan diatas menggunakan counter suhu+=40 untuk melakukan increment agar kondisi terpenuhi.

d. perulangan (while) secara bersarang

konsep dari perulangan while secara bersarang hampir sama dengan perulangan for secara bersarang, keduanya mengakomodasi perulangan didalam while kemungkinan ada perulangan while lebih dari sama dengan satu atau ada perulangan lain. Seperti contoh berikut, didalam while ada perulangan while.

```
>>> i = 0  
>>> while i < 5:  
>>>     j = 0  
>>>     while j <= i:  
>>>         print("*", end=" ")  
>>>         j+=1  
>>>     print()  
>>>     i+=1
```

output program while secara bersarang :

```
*  
* *  
* * *  
* * * *  
* * * * *
```

e. break

Fungsi `break` digunakan untuk menghentikan perulangan dengan mengakomodasi kondisional statement. Contoh penggunaan `break` pada bahasa pemrograman python dapat dilihat sebagai berikut. (menghentikan perulangan apabila angka 3 didapat)

`break` pada `for` :

```
>>> for i in range(0,5):  
>>>     if i == 3:  
>>>         print("angka 3 didapat")  
>>>         break  
>>>     print(i)  
0  
1  
2  
angka 3 didapat
```

`break` pada `while` :

```
>>> i = 0  
>>> while i < 5:  
>>>     if i == 3:  
>>>         print("angka 3 terambil")  
>>>         break  
>>>     print(i)  
>>>     i+=10  
1  
2  
angka 3 didapat
```

f. pass

fungsi pass digunakan untuk membiarkan, atau tidak ada sesuatu yang akan dijalankan apabila kondisi tersebut terpenuhi akan tetapi perintah setelah pass akan tetap dijalankan. pass seringkali digunakan bersamaan dengan kondisional statement. Pada bahasa pemrograman python pass dapat dilihat sebagai berikut.

```
>>> for i in range(5):
>>>     if i == 3:
>>>         pass
>>>         print("ini pass")
>>>     print("angka sekarang: ",i)
angka sekarang: 0
angka sekarang: 1
angka sekarang: 2
ini pass
angka sekarang: 3
angka sekarang: 4
```

g. continue

fungsi continue berbeda dengan pass, continue digunakan untuk melompati proses yang berjalan saat kondisi yang didefinisikan terpenuhi. proses saat itu akan dilompati dan langsung kembali ke perulangan setelah proses tersebut tanpa menghiraukan operasi yang ada. Dalam bahasa pemrograman python continue dapat dilihat sebagai berikut.

```
>>> for i in range(5):
>>>     if i == 3:
>>>         continue
>>>         print("ini continue")
>>>     print("angka sekarang: ",i)
angka sekarang: 0
angka sekarang: 1
angka sekarang: 2
angka sekarang: 4
```

Latihan 3

1. Buatlah program sederhana untuk melakukan cetak bintang sebagai berikut

```
* * * * *  
* * * *  
* * *  
* *  
*
```

2. Buatlah program sederhana untuk memasukan daftar berbagai nama, apabila saya memasukan nama python maka program akan berhenti berjalan dan menampilkan salam perpisahan goodbye python.

Ilustrasi 1 : apabila program dieksekusi akan muncul inputan sebagai berikut

Masukan nama :

Ilustrasi 2 : apabila dimasukan nama lukman maka akan tampil **Nama saya lukman**, kemudian tampil inputan baru lagi (begitu seterusnya apabila saya tidak memasukan nama python)

Masukan nama : lukman
Nama saya lukman
Masukan nama :

Ilustrasi 3 : apabila dimasukan nama python maka akan tampil **Nama saya python**, kemudian tampil **goodbye python** dan program akan berhenti.

Masukan nama : lukman
Nama saya lukman
Masukan nama : cahyo
Nama saya cahyo
Masukan nama : anggi
Nama saya anggi
Masukan nama : python
Nama saya python
goodbye python