

FOR LOOP

Pertemuan 12



APA ITU PERULANGAN FOR (PADA PYTHON)?

- Perulangan for pada python adalah perintah yang digunakan untuk **melakukan iterasi dari sebuah nilai sequence** atau data koleksi pada python seperti [List](#), [Tuple](#), [String](#) dan lain-lain [\[2\]](#).
- For pada python memiliki **perilaku yang berbeda** dengan for pada kebanyakan bahasa pemrograman yang lain, karena pada python ia sangat berkaitan dengan data *sequence* atau data kolektif. Mungkin kalau dibandingkan dengan bahasa lain, for pada python lebih dikenal sebagai foreach.



SYNTAX FOR

- Berikut ini adalah struktur sintaks metode for:

for nilai **in** sequence:

blok kode for



JADI, ADA 3 BAGIAN PENTING.

- 1.sequence: adalah sebuah nilai yang bersifat *iterable* alias bisa diulang-ulang.
- 2.Di antara tipe data yang bersifat *sequence* atau *iterable* adalah:
 1. list
 2. tuple
 3. string
 4. dan lain sebagainya
- 3.nilai: adalah setiap item yang diekstrak dari sequence
- 4.**Blok kode**: yaitu statemen-statemen atau perintah-perintah tertentu yang akan dieksekusi secara berulang.



Buat sebuah variabel dengan tipe data list.

Setelah itu kita akan menampilkan tiap item dari list yang telah kita buat.

Perhatikan contoh berikut:

```
listKota = [  
    'Jakarta', 'Surabaya', 'Depok', 'Bekasi', 'Solo', 'Jogjakarta', 'Semarang', 'Makassar'  
]
```

```
for kota in listKota:
```

```
    print(kota)
```



JIKA DIEKSEKUSI, PROGRAM DI ATAS AKAN MENGHASILKAN OUTPUT:

Jakarta

Surabaya

Depok

Bekasi

Solo

Jogjakarta

Semarang

Makassar



Untuk mengetahui urutan iterasi for dengan list, kita bisa menggunakan fungsi enumerate.

Fungsi tersebut akan mengekstrak 2 buah nilai:

- 1.yang pertama adalah index: yaitu urutan iterasi yang ke berapa
- 2.dan item yang mana itu adalah nilai dari list itu sendiri.



```
listKota = [ '
    Jakarta', 'Surabaya', 'Depok', 'Bekasi', 'Solo', 'Jogjakarta', 'Semarang', 'Makassar'
]
for i, kota in enumerate(listKota):
    print(i, kota)
```



Kode program di atas sama saja seperti sebelumnya, kita hanya menambahkan fungsi `enumerate()` dan mem *passing* variabel `listKota` sebagai parameter. Kita juga mengekstrak dua buah nilai yang kita kasih nama `i` dan `kota`.





JILA DIJALANKAN, BERIKUT ADALAH OUTPUT YANG KITA DAPAT:

- 0 Jakarta
- 1 Surabaya
- 2 Depok
- 3 Bekasi
- 4 Solo
- 5 Jogjakarta
- 6 Semarang
- 7 Makassar

Perhatikan, urutannya dimulai dari **0**, bukan dari angka **1**.



FOR DENGAN FUNGSI RANGE()

Selain dengan list, kita juga bisa menggunakan for dengan fungsi range().

Perhatikan contoh berikut:

```
## 0 sampai 4  
for i in range(5):  
    print("Perulangan ke -", i)
```

```
Perulangan ke - 0  
Perulangan ke - 1  
Perulangan ke - 2  
Perulangan ke - 3  
Perulangan ke - 4
```

Dengan fungsi range, kita bisa melakukan perulangan dari 0, sampai **kurang dari** nilai range yang kita definisikan (yaitu 5 dalam contoh di atas). Sehingga hasil perulangan yang didapatkan adalah 0 sampai 4.



Perhatikan contoh berikut:

```
## 10 sampai 15
```

```
for i in range(10, 16):
```

```
    print('i =', i)
```

Perulangan di atas akan menghasilkan output:

```
i = 10
```

```
i = 11
```

```
i = 12
```

```
i = 13
```

```
i = 14
```

```
i = 15
```



Perhatikan contoh berikut:

```
## Bilangan genap kelipatan 2
```

```
for i in range(2, 12, 2):
```

```
print('i =', i)
```

Pada contoh di atas, sistem akan melakukan perulangan dimulai dari angka 2, hingga kurang dari 12 dengan interval/kelipatan sebanyak 2.

Hasilnya:

```
i = 2
```

```
i = 4
```

```
i = 6
```

```
i = 8
```

```
i = 10
```



FOR DENGAN TUPLE

Tuple adalah di antara tipe data yang bersifat *iterable*, sehingga kita juga bisa memperlakukannya sebagai objek perulangan menggunakan for.

Perhatikan contoh di bawah:

```
tupleBuah = ('Mangga', 'Jeruk',  
'Apel', 'Pepaya')  
for buah in tupleBuah:  
    print(buah)
```

Mangga

Jeruk

Apel

Pepaya



FOR DENGAN STRING

String pun demikian, bersifat **iterable**, sehingga bisa kita jadikan objek perulangan.

Perhatikan contoh berikut:

```
for karakter in "Indonesia":  
    print(karakter)
```

```
I  
n  
d  
o  
n  
e  
s  
i  
a
```



Pada python, kita bisa **menginterupsi** dan juga **men-skip** suatu iterasi pada perulangan.

Terdapat 2 perintah yang bisa kita gunakan, yaitu:

- **break** untuk interupsi (memberhentikan paksa) sebuah perulangan
- **continue** untuk menskip ke iterasi selanjutnya



PERHATIKAN CONTOH BERIKUT:

```
for i in range(10, 20):  
    # skip jika i == 15  
    if (i == 15):  
        continue  
  
    print(i)
```

```
10  
11  
12  
13  
14  
16 <-- Habis 14 langsung 16  
17  
18  
19
```

Perhatikan output di atas, pada saat `i == 15`, perintah `print(i)` tidak dieksekusi dan justru di-skip ke iterasi berikutnya.

Atau...

Kita justru bisa memberhentikan paksa suatu perulangan sekalipun belum sampai ke iterasi yang terakhir.



PERHATIKAN CONTOH BERIKUT:

```
for i in range(10, 20):  
    # hentikan jika i == 15  
    if (i == 15):  
        break  
  
print(i)
```

```
10  
11  
12  
13  
14 <-- print terakhir sebelum terjadi break pada i == 15
```

Sistem akan memberhentikan perulangan ketika `i == 15` dan belum sempat melakukan perintah `print()`.



FOR ... ELSE?

Hampir mirip dengan if ... else.

Tapi tugasnya berbeda.

Perulangan for jika ditambahkan blok else, maka perintah yang ada pada blok else hanya akan dieksekusi ketika perulangan selesai **secara natural** –*tanpa interupsi*.



PERHATIKAN CONTOH BERIKUT:

```
listKota = [  
    'Jakarta', 'Surabaya', 'Depok', 'Bekasi', 'Solo',  
    'Jogjakarta', 'Semarang', 'Makassar'  
]
```

```
for kota in listKota:  
    print(kota)  
else:  
    print('Tidak ada lagi item yang tersisa')
```

Jakarta

Surabaya

Depok

Bekasi

Solo

Jogjakarta

Semarang

Makassar

Tidak ada lagi item yang tersisa

FOR ... ELSE + BREAK

- Jika kita gabungkan for ... else dengan break, maka blok else **hanya akan** dieksekusi jika perintah break tidak dieksekusi.
- Kita bisa memanfaatkan for ... else + break untuk pencarian sebuah item pada list.



PERHATIKAN CONTOH BERIKUT:

```
listKota = [  
    'Jakarta', 'Surabaya', 'Depok', 'Bekasi', 'Solo',  
    'Jogjakarta', 'Semarang', 'Makassar'  
]  
  
kotaYangDicari = input('Ketik nama kota yang kamu cari: ')  
  
for i, kota in enumerate(listKota):  
    # kita ubah katanya ke lowercase agar  
    # menjadi case insensitive  
    if kota.lower() == kotaYangDicari.lower():  
        print('Kota yang anda cari berada pada indeks', i)  
        break  
else:  
    print('Maaf, kota yang anda cari tidak ada')
```

Program ini akan meminta user untuk menginputkan nama kota yang ingin dicari. Jika kotanya maka akan kita kasih info indeks-nya berapa (dalam **listKota**), dan jika tidak ada maka perintah **print()** yang ada di blok **else** akan dieksekusi.



Coba jalankan. Kemudian kita input kata solo, ini hasilnya:

Ketik nama kota yang kamu cari: solo Kota yang anda cari berada pada indeks **4**

Jika kita cari pakai kota yang tidak ada di dalam list, begini hasilnya:

Ketik nama kota yang kamu cari: pontianak Maaf, kota yang anda cari tidak ada



REFERENSI

- [1] <https://techterms.com/definition/loop>
- [2] https://www.w3schools.com/python/python_for_loops.asp

