



Perulangan

For

Seperti di bahasa pemrograman lainnya, Python juga memiliki fungsi for. Bedanya di Python, For tidak hanya untuk perulangan dengan jumlah finite (terbatas), melainkan lebih ke fungsi yang dapat melakukan perulangan pada setiap jenis variabel berupa kumpulan atau urutan. Variabel yang dimaksud bisa berupa list, string, ataupun range. Jika sebuah list atau urutan berisi expression, maka ia akan dievaluasi terlebih dahulu. Kemudian item pertama pada urutan/list akan diassign sebagai variabel iterating_var. Setelahnya, blok statement akan dieksekusi, berlanjut ke item berikutnya, berulang, hingga seluruh urutan habis.

```
1. for huruf in 'Dicoding': # Contoh pertama
2.     print('Huruf: {}'.format(huruf))
3.
4. flowers = ['mawar', 'melati', 'anggrek']
5. for flower in flowers: # Contoh kedua
6.     print('Flower: {}'.format(flower))
```

Output:

```
Huruf : D
Huruf : i
Huruf : c
Huruf : o
Huruf : d
Huruf : i
Huruf : n
Huruf : g
Flower: mawar
Flower: melati
Flower: anggrek
```

Anda juga dapat melakukan perulangan berdasarkan indeks atau range dengan memanfaatkan fungsi len():



DIBANTU



```
2. for index in range(len(flowers)):
3.     print('Flowers: {}'.format(flowers[index]))
```

Output:

```
Flowers : mawar
Flowers : melati
Flowers : anggrek
```

While

While pada bahasa Python digunakan untuk mengeksekusi statement selama kondisi yang diberikan terpenuhi (True). Kondisi dapat berupa expression apapun, dan harap diingat bahwa **True di Python termasuk semua nilai non-zero**. Saat kondisi menjadi False, program akan melanjutkan ke baris setelah blok statement.

Tip: Python tidak memiliki do.. while statement

Seperti for dan semua statement percabangan, blok statement yang mengikuti kondisi while dan memiliki posisi indentasi yang sama, dianggap blok statement yang akan dieksekusi.

Contoh:

```
1. count = 0
2. while (count < 7):
3.     print('Hitungannya adalah: {}'.format(count))
4.     count = count + 1
```

Output:

```
Hitungannya adalah: 0
Hitungannya adalah: 1
Hitungannya adalah: 2
Hitungannya adalah: 3
Hitungannya adalah: 4
Hitungannya adalah: 5
Hitungannya adalah: 6
```



DIBANTU



```
1. var = 1
2. while var == 1: # This constructs an infinite loop
3.     num = input('Masukkan angka: ')
4.     print('Anda memasukkan angka: {}'.format(num))
5.
6.
7. while True: # This constructs an infinite loop
8.     num = input('Masukkan angka: ')
9.     print('Anda memasukkan angka: {}'.format(num))
```

Potongan kode di atas tidak akan pernah bernilai False karena nilai var tidak pernah berubah. Untuk menghentikan infinite loop, gunakan CTRL (atau CMD) - C untuk menghentikannya dan keluar dari program.

Anda juga dapat menyingkat penulisan blok statement While jika statement Anda cukup terwakili oleh satu baris.



```
1. while (var1): do_something()
```

Perulangan Bertingkat

Ada kalanya Anda perlu untuk melakukan perulangan bertingkat, misalnya untuk menghasilkan contoh print-out berikut:

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

Anda dapat melakukannya dengan kode berikut:



```
1. for i in range(0, 6):
2.     for j in range(0, 6 - i):
3.         print('*', end='')
4.     print()
```



DIBANTU



[← Sebelumnya](#)

[Selanjutnya →](#)



Dicoding Space
Jl. Batik Kumeli No.50, Sukaluyu,
Kec. Cibeunying Kaler, Kota Bandung
Jawa Barat 40123



Decode Ideas
Discover Potential

[Tentang Kami](#)

- [Blog](#)
- [Reward](#)
- [Showcase](#)
- [Hubungi Kami](#)
- [FAQ](#)

Penghargaan



DIBANTU