

BAB 3

Perulangan

Perulangan dalam bahasa pemrograman memiliki fungsi untuk menyuruh 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.

3.1 Perulangan For

Bentuk umum perulangan for adalah sebagai berikut:

```
for indeks in range(banyak_perulangan):  
    # jalankan kode ini  
    # jalankan juga kode ini  
#kode ini tidak akan diulang karena berada di luar for
```

Contoh program:

```
ulang = 5  
  
for i in range(ulang):  
    print(f"Perulangan ke-{i}")
```

Variabel i berfungsi untuk menampung indeks, dan fungsi range() berfungsi untuk membuat list dengan range dari 0-10. 'f' atau 'F' di depan string memberi tahu Python untuk melihat nilai di dalam {} dan menggantinya dengan nilai variabel jika ada.

Contoh lain menggunakan senarai (list):

```
item = ['Aku', 'ganteng', 'sekali', 'nih']  
  
for isi in item:  
    print(isi)
```

3.2 Perulangan While

Bentuk umum:

```
while(True):  
    # jalankan kode ini  
# kode ini berada di luar perulangan while
```

Contoh:

```
jawab = 'Iya'  
hitung = 0  
  
while(jawab == 'Iya'):  
    hitung += 1  
    jawab = input("Ulang lagi tidak? ")  
  
print(f"Total perulangan: {hitung}")
```

Atau bisa juga dengan bentuk yang seperti ini, dengan menggunakan kata kunci 'break'.

```
jawab = 'Iya'  
hitung = 0  
  
while(jawab == 'Iya'):  
    hitung += 1  
    jawab = input("Ulang lagi tidak? ")  
    if jawab == 'tidak':  
        break  
  
print(f"Total perulangan: {hitung}")
```

Pertama menentukan variabel untuk menghitung, dan menentukan kapan perulangan berhenti. kalau pengguna menjawab tidak maka perulangan akan terhenti.

3.3 Break

Pernyataan 'break' di Python mengakhiri loop saat ini dan melanjutkan eksekusi pada pernyataan berikutnya, sama seperti 'break' tradisional yang ditemukan di C.

Contoh program:

```
for val in "string":  
    if val == "i":  
        break
```

```
print(val)

print("Selesai")
```

3.4 Continue

Pernyataan 'continue' dalam Python mengembalikan kontrol ke awal loop 'while'. Pernyataan 'continue' menolak semua pernyataan yang tersisa dalam iterasi loop saat ini dan memindahkan kontrol kembali ke atas loop. Pernyataan 'continue' dapat digunakan pada perulangan 'while' dan 'for'.

```
for val in "string":
    if val == "i":
        continue
    print(val)

print("Selesai")
```