

PEMBELAJARAN DASAR PYTHON

BAB 6 : PERULANGAN

1.1 Tujuan & Manfaat

1. Mahasiswa mengerti konsep perulangan pada bahasa pemrograman python

1.2 Ulasan Materi

PERULANGAN

1. Perulangan Python

Secara umum, pernyataan pada bahasa pemrograman akan dieksekusi secara berurutan. Pernyataan pertama dalam sebuah fungsi dijalankan pertama, diikuti oleh yang kedua, dan seterusnya. Tetapi akan ada situasi dimana Anda harus menulis banyak kode, dimana kode tersebut sangat banyak. Jika dilakukan secara manual maka Anda hanya akan membuang-buang tenaga dengan menulis beratus-ratus bahkan beribu-ribu kode. Untuk itu Anda perlu menggunakan pengulangan di dalam bahasa pemrograman Python.

Di dalam bahasa pemrograman Python pengulangan dibagi menjadi 2 bagian, yaitu: *While Loop* dan *For Loop*

2. While Loop

Pengulangan While Loop di dalam bahasa pemrograman Python dieksekusi statement berkali-kali selama kondisi bernilai benar atau True.

Contoh:

```
i = 1
while i < 6:
    print(i)
    i += 1
```

Output :

```
1
2
3
4
5
```

Catatan: Ingat untuk menambah i, atau loop akan berjalan selamanya

While Loop membutuhkan variabel yang relevan untuk ready, dalam contoh ini kita perlu mendefinisikan variabel pengindeksan, i, yang kita set ke 1.

1. Pernyataan Break

Dengan pernyataan break kita dapat menghentikan perulangan meskipun kondisi while true.

Contoh:

```
i = 1
while i < 6:
    print(i)
    if (i == 3):
        break
    i += 1
```

Output :

```
1
2
3
```

2. Pernyataan Continue

Dengan pernyataan continue kita dapat menghentikan iterasi saat ini, dan melanjutkan dengan yang berikutnya..

Contoh:

```
i = 0
while i < 6:
    i += 1
    if i == 3:
        continue
    print(i)
```

Output :

```
1
2
3
4
5
6
```

3. Pernyataan Else

Dengan pernyataan else kita dapat menjalankan blok kode satu kali ketika kondisinya tidak lagi true

Contoh:

```
i = 1
while i < 6:
    print(i)
    i += 1
else:
    print("i is no longer less than 6")
```

Output :

```
1
2
3
4
5
i is no longer less than 6
```

3. For Loop

Perulangan for digunakan untuk mengulangi urutan (yaitu list, tuple, dictionary, set, atau string). Ini kurang seperti kata kunci for dalam bahasa pemrograman lain, dan bekerja lebih seperti method iterator seperti yang ditemukan dalam bahasa pemrograman berorientasi objek lainnya.

Dengan for loop kita dapat mengeksekusi satu set pernyataan, sekali untuk setiap item dalam list, tuple, set dll.

Contoh:

```
fruits = ["apel", "jeruk", "pisang"]
for x in fruits:
    print(x)
```

Output :

```
apel
jeruk
pisang
```

Perulangan for tidak memerlukan variabel pengindeksan untuk disetel sebelumnya.

1. Perulangan Melalui String

Bahkan string adalah objek yang dapat diubah, mereka berisi urutan karakter:

Contoh:

```
for x in "pisang":
    print(x)
```

Output :

```
p
i
s
a
n
g
```

2. Pernyataan Break

Dengan pernyataan break kita dapat menghentikan loop sebelum loop melewati semua item.

Contoh:

```
fruits = ["apel", "jeruk", "pisang"]
for x in fruits:
    print(x)
    if x == "pisang":
        break
```

Output :

```
apel  
jeruk
```

3. Pernyataan Continue

Dengan pernyataan continue kita dapat menghentikan iterasi loop saat ini, dan melanjutkan dengan yang berikutnya.

Contoh:

```
fruits = ["apel", "jeruk", "pisang"]  
for x in fruits:  
    if x == "jeruk":  
        continue  
    print(x)
```

Output :

```
apel  
pisang
```

4. Fungsi range()

Untuk mengulang satu set kode beberapa kali, kita dapat menggunakan fungsi range(), Fungsi range() mengembalikan urutan angka, mulai dari 0 secara default, dan bertambah 1 (secara default), dan berakhir pada angka yang ditentukan.

Contoh:

```
for x in range(6):  
    print(x)
```

Output :

```
0
1
2
3
4
5
```

Perhatikan bahwa range(6) bukanlah nilai 0 hingga 6, tetapi nilai 0 hingga 5

5. Else di Perulangan For

Kata kunci else dalam perulangan for menentukan blok kode yang akan dieksekusi ketika perulangan selesai.

Contoh:

```
for x in range(6):
    print(x)
else:
    print("Selesai!")
```

Output :

```
0
1
2
3
4
5
Selesai!
```

6. Pernyataan Pass

Perulangan loop tidak boleh kosong, tetapi jika programmer karena alasan tertentu memiliki perulangan for tanpa konten, masukkan pernyataan pass untuk menghindari kesalahan.

Contoh:

```
for x in [0, 1, 2]:  
    pass
```

Output :

Memiliki for loop kosong seperti ini, akan menimbulkan kesalahan tanpa pernyataan pass

1.3 Praktikum

Contoh : While Loop

While merupakan perulangan dimana mengecek kondisi terlebih dahulu, apabila terpenuhi maka perulangan akan berjalan, sampai selesai. Untuk implementasinya akan diuraikan pada poin dibawah ini.

1. Inisialisasikan variabel i dengan bertipe integer dan memiliki nilai 1

```
i = 1
```

2. Tambahkan perulangan while dengan menambahkan operator perbandingan kurang dari 6. Penjelasan dari sintaks ini mendeskripsikan tetap dijalankan apabila variabel i kurang dari 6.

```
While i < 6:
```

3. Didalam while tambahkan output menggunakan perintah print dengan memiliki nilai i

```
print(i)
```

4. Tambahkan perintah increment untuk variabel i. Perintah increment digunakan untuk menambahkan nilai secara kontinyu selama perulangan berjalan. Penambahan nilai memiliki pola +1

```
i += 1
```

1.4 Percobaan 1

Sekarang kerjakan percobaan 1 dengan mengikuti langkah-langkah dibawah ini !

1. Buatlah variabel dengan nama i yang memiliki nilai integer 6
2. Buatlah perulangan while untuk menampilkan angka 6 sampai 1 secara terbalik, dengan menambahkan perulangan while dan menambahkan operator perbandingan lebih dari sama dengan 1
3. Didalam while tampilkan output variabel i
4. Tambahkan perintah increment untuk variabel i. Perintah increment digunakan untuk menambahkan nilai secara kontinyu selama perulangan berjalan. Penambahan nilai memiliki pola -=
5. Jalankan kode program percobaan 1 diatas dengan menekan tombol Check Code Validity