Program Control

Perulangan (Loop)

Perulangan (Loop) Instruksi Program

- Dalam pembuatan program, kita akan sering mendapati instruksi-instruksi yang perlu dilakukan secara berulang (*Looping*).
- Kita bisa saja menuliskan instruksi tersebut secara berulang saat dibutuhkan, akan tetapi hal tersebut merupakan hal yang kurang efektif dan efisien.
- Cara yang efektif dan efisien untuk melakukan perulangan instruksi dalam pemrograman adalah menggunakan pernyataan/statemen perulangan/*loop* `while` atau `for` yang juga dikenal dalam bahasa pemrograman Python.

Jenis Perulangan (Loop)

Jenis Perulangan (Loop)

1. Perulangan terdefinisi (Definite)

- Perulangan suatu blok program yang dilakukan dengan **jumlah perulangan** yang telah **diketahui** dengan pasti oleh *programmer*.
- Jenis perulangan ini juga dikenal dengan **perulangan terbatas**.

2. Perulangan tak terdefinisi (Indefinite)

- Perulangan suatu blok program yang dilakukan dengan jumlah perulangan yang tidak diketahui (***Unknown***) oleh *programmer*.
- Jenis perulangan ini merupakan **perulangan tanpa batas** (*Unlimited*), yang menjadikan eksekusi suatu blok program akan terus diulang selama nilai kondisinya adalah **benar** atau **True**.

Perulangan (Loop)

While

1.1 Perulangan - `While`

- Prinsip umum dari ***statement*** perulangan **`while`** adalah melakukan **perulangan** terhadap satu atau beberapa baris program selama kondisi yang diharapkan bernilai **benar** (***True***). Jika kondisi tersebut bernilai **salah** (***False***), maka proses perulangan program akan dihentikan.
- Perulangan **`while`** dapat digunakan untuk melakukan perulangan jenis **terbatas** (*Definite loop*) maupun perulangan jenis **tanpa batas** (*Indefinite loop*).

1.2 Format Perulangan `While`

A. Perulangan terhadap satu baris program:

```
while kondisi:
      statement (program yang dijalankan)
```

B. Perulangan terhadap lebih dari satu baris program:

```
while kondisi:
      statement-1
      statement-2
      statement-...
      statement-n
```

Contoh 1: Perulangan While Terbatas

```
kalimat=input("Masukkan data: ")
print("\nTampilkan data 5 kali \n")
a = 1
while(a<=5):                    # Pembatas jumlah perulangan
    print(a,".",kalimat)
    a=a+1
```

```
# Output
Masukkan data: Hallo Sahabat Robonesia

Tampilkan data 5 kali

1 . Hallo Sahabat Robonesia
2 . Hallo Sahabat Robonesia
3 . Hallo Sahabat Robonesia
4 . Hallo Sahabat Robonesia
5 . Hallo Sahabat Robonesia
```

Contoh 2: Perulangan While Tanpa Batas

```
while True:                                    # Jumlah perulangan tidak ada batas
    print("Selamat Belajar Python")
    print("di Robonesia")
```

```
# Output
Selamat Belajar Python
di Robonesia
Selamat Belajar Python
di Robonesia
Selamat Belajar Python
di Robonesia
...
```

Perulangan

For

2.1 Perulangan - `For`

- Perulangan **`for`** termasuk dalam jenis **perulangan terbatas** (*Definite loop*)
- Statemen perulangan **`for`** pada pemrograman Python sangat berbeda dengan pemrograman C/C++ dan Java.
- Pada pemrograman Python, statemen **`for`** merepresentasikan teknik perulangan yang bisa diterapkan pada tipe-tipe data runtun (*Sequential data type*) seperti ***list***, ***tuple***, ***set***, dan ***dictionary***.

2.2 Format Perulangan `For`

Format umum perulangan dengan statemen **`for`**:

```
for variabel in iterable:
      statement-1                    #program yang dijalankan
      statement-2
      statement-...
      statement-n
```

Catatan:
`iterable` = bisa berupa *list*, *tuple*, *set*, atau *dictionary*

Contoh Perulangan For (1)

A. Perulangan for berdasar data suatu list

```
# Menampilkan isi suatu list dengan perulangan "for"
# Perulangan berdasar data suatu list

list_lauk = ["Tahu","Tempe","Krupuk","Perkedel","Telor ceplok","Telor dadar"]
print("Daftar lauk: ")
for t in list_lauk:
    print(t)          # Menampilkan elemen list_lauk
```

```
# Output
Daftar lauk:
Tahu
Tempe
Krupuk
Perkedel
Telor ceplok
Telor dadar
```

Contoh Perulangan For (2)

B. Perulangan for berdasar indeks suatu list

```
# Menampilkan isi suatu list dengan perulangan "for"
# Perulangan berdasarkan indeks suatu list

list_olahraga = ["Basket","Volley","Futsal","Takraw","Soccer","Base ball"]
panjangList = len(list_olahraga)
print("Daftar Olahraga dengan Bola: ")
for s in range(panjangList):                              # indeks dari list
    print("Olahraga dengan bola: ", list_olahraga[s])     # Menampilkan elemen list_olahraga
```

```
Daftar Olahraga dengan Bola:
Olahraga dengan bola:  Basket
Olahraga dengan bola:  Volley
Olahraga dengan bola:  Futsal
Olahraga dengan bola:  Takraw
Olahraga dengan bola:  Soccer
Olahraga dengan bola:  Base ball
```

Fungsi-Fungsi Terkait Perulangan `For`

2.3 Fungsi-Fungsi Terkait Perulangan `For`

- Fungsi `enumerate()`
- Fungsi `range()`
- Fungsi `sorted()`
- Fungsi `reversed()`
- Fungsi `zip()`

2.3.1 Fungsi enumerate()

Perulangan for menggunakan fungsi enumerate()

```
# Program menampilkan indeks dan elemen tuple

tuple_kota = ("Lampung","Palembang","Jambi","Riau","Medan","Pekanbaru")
print("Nama kota di Sumatera:")
for index,elemen in enumerate(tuple_kota):    # Perulangan dengan fungsi enumerate()
    print(index+1,". ",elemen)                # Menampilkan index & elemen tuple
```

```
# Output
Nama kota di Sumatera:
1 .  Lampung
2 .  Palembang
3 .  Jambi
4 .  Riau
5 .  Medan
6 .  Pekanbaru
```

2.3.2 Fungsi `range()`

Format perulangan for mengunakan fungsi **`range()`**:

```
# Format-1
range(nilai_awal,nilai_akhir,selisih)
```

```
# Format-2
range(nilai_awal,nilai_akhir)
```

```
# Format-3
range(nilai_akhir)
```

A. Fungsi range(awal,akhir,selisih)

```
# Penggunaan fungsi range() format ke-1
awal=int(input("Masukkan nilai awal : "))    # Memasukkan nilai awal dari keyboard
akhir=int(input("Masukkan nilai akhir : "))   # Memasukkan nilai akhir dari keyboard
selisih=int(input("Masukkan nilai selisih : "))    # Memasukkan nilai selisih dari keyboard
print('')
print("Nilai dalam range adalah:")
print("-------------------------")
for t in range(awal,akhir,selisih):
    print(t)                                  # Menampilkan nilai dalam range
```

```
# Output
Masukkan nilai awal : 20
Masukkan nilai akhir : 40
Masukkan nilai selisih : 5

Nilai dalam range adalah:
-------------------------
20
25
30
35
```

B. Fungsi range(awal,akhir)

```
# Penggunaan fungsi range() format ke-2
awal=int(input("Masukkan nilai awal : "))    # Memasukkan nilai awal dari keyboard
akhir=int(input("Masukkan nilai akhir : "))   # Memasukkan nilai akhir dari keyboard
print('')
print("Nilai dalam range adalah:")
print("-------------------------")
for u in range(awal,akhir):
    print(u)                                  # Menampilkan nilai dalam range
```

```
# Output
Masukkan nilai awal : 10
Masukkan nilai akhir : 15

Nilai dalam range adalah:
-------------------------
10
11
12
13
14
```

C. Fungsi range(akhir)

```
# Penggunaan fungsi range() format ke-3
akhir = 3
for a in range(akhir):
    print ("Hello Sahabat Robonesia")
```

```
# Output
Hello Sahabat Robonesia
Hello Sahabat Robonesia
Hello Sahabat Robonesia
```

2.3.3 Fungsi `sorted()`

Fungsi `sorted()` pada perulangan **for** digunakan untuk **mengurutkan** suatu data.

```
# Menampilkan data list setelah diurutkan
nilai_list =[13,7,3,11]                    # List yang belum diurutkan
print("Data list sebelum diurutkan:")
for b in nilai_list:
    print(b)

print()

print("Data list setelah diurutkan:")
for c in sorted(nilai_list):
    print(c)
```

```
# Output
Data list sebelum diurutkan:
13
7
3
11

Data list setelah diurutkan:
3
7
11
13
```

2.3.4 Fungsi `reversed()`

Fungsi **`reversed()`** pada perulangan **for** digunakan untuk **membalik urutan** suatu data.

```
# Menampikan data tuple setelah dibalik urutannya
data_tuple=("Jogja","Surabaya","Bandung")                    # Tuple yang belum diurutkan
print("Data nama kota sebelum dibalik:")
for d in data_tuple:
    print(d)
print()
print("Data nama kota setelah dibalik:")
for e in reversed(data_tuple):
    print(e)
```

```
# Output
Data nama kota sebelum dibalik:
Jogja
Surabaya
Bandung

Data nama kota setelah dibalik:
Bandung
Surabaya
Jogja
```

2.3.5 Fungsi `zip()`

Fungsi **`zip()`** pada perulangan **for** digunakan untuk **mempaketkan** atau **mengompres** beberapa obyek data menjadi satu.

```
# Fungsi zip() pada struktur for
data_nama=["Faruq","Farros","Farrel"]      # List nama
data_tahun=[2010,2017,2016]                # List tahun

for f,g in zip(data_nama,data_tahun):
    print(f," lahir pada tahun ",g)
```

```
# Output
Faruq  lahir pada tahun  2010
Farros  lahir pada tahun  2017
Farrel  lahir pada tahun  2016
```

Perulangan For Bersarang (Nested)

2.4 Statemen `For` Bersarang (*Nested For*)

- Untuk kasus-kasus tertentu, seperti pada **list**, **tuple**, ataupun **dictionary** dua dimensi atau lebih, kita memerlukan statemen perulangan **for** lebih dari satu yang bersarang (Nested **for**).
- Statemen **for** pertama digunakan sebagai baris dan yang kedua sebagai kolom.
- Umumnya statemen **for** bersarang digunakan dalam kasus yang melibatkan matriks.

Contoh Perulangan For Bersarang

```
list=[[1,2,3,4],["Jogja","Solo","Klaten"],["C","C++","Python"]]

for x in range(len(list)):
    print("------------"," <--- Baris", x+1)
    for y in range(len(list[x])):
        print(list[x][y])
```

```
# Output
------------  <--- Baris 1
1
2
3
4
------------  <--- Baris 2
Jogja
Solo
Klaten
------------  <--- Baris 3
C
C++
Python
```

Terima Kasih