

## Pertemuan 5

### Proses Perulangan/Loop for, while



#### CAPAIAN PEMBELAJARAN

---

1. Memahami Proses perulangan, for dan while



#### KEBUTUHAN ALAT/BAHAN/SOFTWARE

---

1. Sistem Operasi Linux/Window 10
2. Bahasa pemrograman Python
3. Teks editor



#### DASAR TEORI

---

##### Loop For

Sebuah loop for digunakan untuk iterasi berurutan (baik list, tuple, dict, set, atau string).

Loop for kurang lebih seperti pada bahasa pemrograman lain, dan bekerja menggunakan fungsi/method iterator seperti yang ada di dalam pemrograman berorientasi objek.

Dengan perulangan for bisa mengeksekusi satu pernyataan set, sekali untuk setiap item di dalam list, tuple, set dll

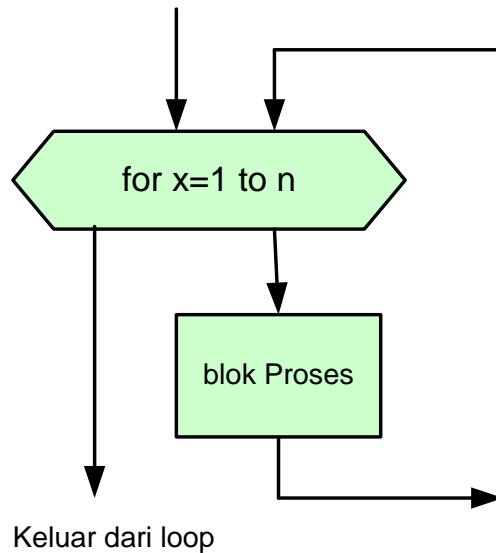
Loop for umumnya digunakan untuk melakukan pengulangan yang banyaknya sudah pasti  
Menentukan nilai awal pengulangan dan nilai akhir pengulangan

Sintaks:

```
for x in n:
```

```
    blok proses
```

list  
tuple  
dict  
set  
string  
range()



contoh:

```

1  # menggunakan pernyataan break
2  # membaca data dari list
3  # contoh51.py
4  #-----
5  buah = ["semangka", "mangga", "pisang","jambu"]
6  print(buah,"\n")
7  print("tampil menggunakan loop :")
8  for x in buah:
9      print(x)
10 print(">>>>> Selesai <<<<<<<")

```

hasil keluaran seperti gambar berikut:

```

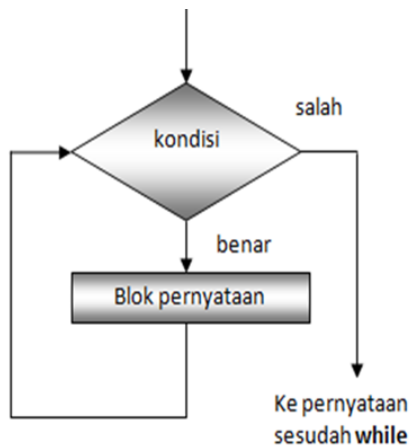
['semangka', 'mangga', 'pisang', 'jambu']

tampil menggunakan loop :
semangka
mangga
pisang
jambu
>>>>> Selesai <<<<<<<

```

## Loop while

Dengan loop while bisa mengeksekusi sekumpulan pernyataan selama kondisinya benar.



```
while kondisi-loop:
    //blok loop
    //statement-statement;
```

```
# loop while
n=10
x=1
while x <= 10:
    print(x,end=" ")
    x+=1
```

```
1 2 3 4 5 6 7 8 9 10
>>> |
```



## PRAKTIK

1. Buatlah program untuk menampilkan **list** hari, seperti pada kode program berikut berikut

```
1  #*****
2  # menggunakan pernyataan loop for
3  # membaca data dari list
4  # contoh51.py
5  #*****
6  hari = ["Minggu", "Senin", "Selasa", "Rabu", "Kamis", "Jumat", "Sabtu"]
7  print(hari, "\n")
8  print("tampil menggunakan loop :")
9  for x in hari:
10 |   print(x)
11 print(">>>>> Selesai <<<<<<<")
```

hasil keluarannya:

```
['Minggu', 'Senin', 'Selasa', 'Rabu', 'Kamis', 'Jumat', 'Sabtu']

tampil menggunakan loop :
Minggu
Senin
Selasa
Rabu
Kamis
Jumat
Sabtu
>>>>> Selesai <<<<<<<
```

2. Loop untuk membaca deretan karakter tipe data string, buatlah program seperti pada kode berikut:

```
1  #####
2  # menggunakan pernyataan for
3  # membaca data dari string
4  # praktik52.py
5  #####
6  print("tampil menggunakan loop :")
7  for x in "YOGYAKARTA":
8      | print(x)
9  print(">>>>> Selesai <<<<<<<")
```

Hasil keluarannya

```
tampil menggunakan loop :
Y
O
G
Y
A
K
A
R
T
A
>>>>> Selesai <<<<<<<
```

3. Pernyataan for dan **break** bisa digunakan untuk menghentikan pengulangan sebelum perulangan melalui semua item:

```
1  #####
2  # menggunakan pernyataan loop for dan break
3  # membaca data dari list
4  # praktik53.py
5  #####
6  hari = ["Minggu", "Senin", "Selasa", "Rabu", "Kamis", "Jumat", "Sabtu"]
7  print(hari, "\n")
8  print("tampil menggunakan loop :")
9  for x in hari:
10     | print(x)
11     | if x=="Kamis":
12     |     | break
13  print(">>>>> Selesai <<<<<<<")
```

Hasilnya :

```
['Minggu', 'Senin', 'Selasa', 'Rabu', 'Kamis', 'Jumat', 'Sabtu']

tampil menggunakan loop :
Minggu
Senin
Selasa
Rabu
Kamis
>>>>> Selesai <<<<<<<
```

4. Pernyataan for dan continue, ketika menjalankan continue loop akan melewati proses untuk melanjutkan ke loop berikutnya. Buatlah seperti pada kode program berikut:

```
1  #*****
2  # menggunakan pernyataan loop for dan continue
3  # membaca data dari list
4  # praktik53.py
5  #*****
6  hari = ["Minggu", "Senin", "Selasa", "Rabu", "Kamis", "Jumat", "Sabtu"]
7  print(hari, "\n")
8  print("tampil menggunakan loop :")
9  for x in hari:
10     if x=="Kamis" or x=="Jumat":
11         continue
12     print(x)
13
14  print(">>>>> Selesai <<<<<<<")
```

Amati hasilnya

5. Loop dan fungsi `range()`, fungsi `range()` mengembalikan urutan angka, secara default dimulai dari 0, dan bertambah 1 (secara default), dan berakhir sesuai angka yang ditentukan, buatlah seperti pada kode program berikut:

```

1  # *****
2  # loop dengan fungsi range
3  # 0...5
4  # praktik55.py
5  #*****
6  print("Manggunakan range()")
7  print(range(6),"\n")
8
9  for x in range(6):
10 |   print(x,end = " ")
11
12 print("\n>>>>> Selesai <<<<<<<")

```

buatlah analisis dan amati hasilnya

6. Fungsi range () default ke 0 sebagai nilai awal, namun dimungkinkan untuk menentukan nilai awal dengan menambahkan parameter: range (2, 6) , yang berarti nilai dari 2 hingga 6 (tetapi tidak termasuk 6):

```

1  # *****
2  # loop dengan fungsi range
3  # nilai awal 2 dan akhir 6
4  # praktik56.py
5  #*****
6  print("Manggunakan range()")
7  print(range(2,6),"\n")
8
9  for x in range(2,6):
10 |   print(x,end = " ")
11
12 print("\n>>>>> Selesai <<<<<<<")

```

buatlah analisis dan amati hasilnya

7. Fungsi range() default menaikkan urutan sebesar 1, namun dimungkinkan untuk menentukan nilai kenaikan dengan menambahkan parameter ketiga: range(10, 30, 5) :

```

1  # *****
2  # loop dengan fungsi range
3  # nilai awal 10 dan akhir 30, step 5
4  # praktik57.py
5  #*****
6  print("Manggunakan range()")
7  print(range(10,30,5),"\n")
8
9  for x in range(10,30,5):
10 |   print(x)
11
12 print("\n>>>>> Selesai <<<<<<<")

```

amati hasilnya dan buatlah analisis

8. **for** dan **else** dalam **for** perulangan menentukan blok kode yang akan dieksekusi ketika perulangan selesai

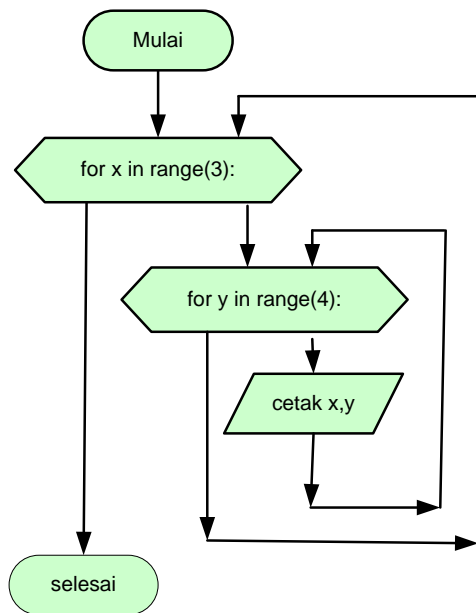
```

1  # *****
2  # loop dengan fungsi range
3  # pernyataan else
4  # praktik58.py
5  #*****
6  for x in range(5):
7  |   print ("Langkah :",x+1)
8  else:
9  |   print("bagian blok else")
10
11 print(">>> selesai <<<")

```

amati hasilnya dan buatlah analisis

9. Loop bersarang adalah **loop** di dalam **loop**. Loop dalam akan dijalankan satu kali untuk setiap iterasi loop luar:



```

1  #####
2  # loop di dalam loop
3  # praktik59.py
4  #####
5  for x in range(3):
6      for y in range(4):
7          print(x,":",y)
8
9  print(">>> selasi <<<")
10

```

Amati hasilnya dan buatlah analisis

10. For loop tidak boleh kosong, tetapi jika Anda karena alasan tertentu memiliki for loop tanpa konten, masukkan pernyataan `pass` untuk menghindari kesalahan.

```

1  # #####
2  # loop dengan fungsi range
3  # pass (tidak ada proses)
4  # praktik510.py
5  #####
6  for x in range(100000):
7      pass
8
9  print("\n>>>>> Selesai <<<<<<<")
10

```

11. Loop **while**, akan melakukan perulangan jika kondisi benar. ketika kondisi salah akan keluar dari loop while. Untuk mencoba buat seperti pada kode program berikut:



```

1  #*****
2  # menggunakan pernyataan while
3  # praktik511.py
4  #*****:
5  print("Menggunakan loop while :")
6  x=0
7  while x < 10:
8      x=x+1
9      print(x)
10
11 print(">>>>> Selesai <<<<<<<")

```

12. Pernyataan **break** bisa menghentikan loop walaupun kondisi while benar:

```

1  #*****
2  # menggunakan pernyataan while dan break
3  # praktik512.py
4  #*****
5  print("Menggunakan loop while :")
6  x=0
7  while x < 10:
8      x=x+1
9      if x==4:
10         break
11         print(x)
12
13 print(">>>>> Selesai <<<<<<<")

```

amati hasilnya

13. **While** pernyataan pernyataan **continue** bisa menghentikan iterasi saat ini, dan melanjutkan dengan yang berikutnya

```

1  #####
2  # menggunakan pernyataan while dan continue
3  # praktik513.py
4  #####
5  print("Menggunakan loop while :")
6  x=0
7  while x < 10:
8      x=x+1
9      if x%2==0:
10         continue
11         print(x)
12
13  print(">>>>> Selesai <<<<<<<")
14

```

amati dan lakukan analisis, apa hasilnya?

14. **while** dengan pernyataan **else** bisa menjalankan satu blok kode satu kali ketika kondisinya tidak lagi benar.

```

1  #####
2  # menggunakan pernyataan while dan else
3  # praktik514.py
4  #####
5  print("Menggunakan loop while :")
6  x=0
7  while x < 10:
8      x=x+1
9      print(x)
10  else:
11      print("proses berakhir\n karena x lebih besar 10")

```

15. Studi kasus masukkan bilangan x, sebanyak n kali, menggunakan loop, dan jumlahkan kemudian hitung rata-ratanya, dan tampilkan hasilnya.

```

1  #####
2  # studi kasus
3  # masukan nilai n kali lop
4  # masukan nilai x sejumlah n kali
5  # jumlahkan tiap x kali loop simpan
6  # hitung rata-rata
7  #####
8  import os
9  lagi="Y"
10 while lagi=="Y" or lagi=="y":
11     os.system('cls')
12     n=int(input("MasukkanJumlah loop : "))
13     j=0
14     for i in range(n):
15         x=int(input("Masukkan nilai x ke : {} = ".format(i)))
16         j=j+x
17     print("Jumlah =",j)
18     print("Rata-rata =",j/n)
19
20     lagi=input("Mencoba lagi [Y/T] :")
..

```



## LATIHAN

---

1. Terdapat list angka=(10,20,10,100,20,20,3,0,10,50,10), tampilkan dengan menggunakan loop for
2. dari latihan nomor 1. hitung dengan ada bearpa? angka 10, dan angka 20
3. dari latihan nomor 1. cari nilai terbesar dengan membandikan lewat pernyataan if



## TUGAS

---

1. Buatlah program untuk menyandikan password dengan kunci seperti pada tabel, dengan memasukan password, untuk mengetahui sandinya ditampilkan menggunakan print(..),

A	B	C	D	E	F	G	H	lainya
x	*	#	v	a	m	q	%	?

sehingga hasilnya seperti berikut:

Masukan password : BUKANGHFD

Tersandikan : \*??x?q%mv



## REFERENSI

---