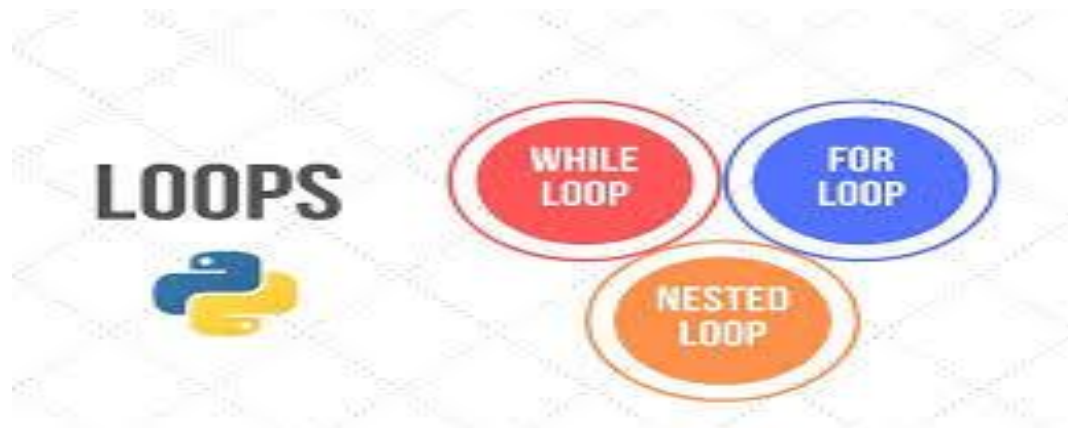


MINGGU KE-5

LOOPING

(PERULANGAN)



LOOPING

Instruksi pengulangan (*repetition*) adalah instruksi yang dapat mengulangi pelaksanaan sederetan instruksi lain berulang kali sesuai dengan persyaratan yang ditentukan.

Struktur instruksi perulangan pada dasarnya terdiri atas:

1. Kondisi perulangan. Suatu kondisi yang harus dipenuhi agar perulangan dapat terjadi
2. Badan (*body*) perulangan. Deretan instruksi yang akan diulang-ulang pelaksanaannya
3. Pencacah (*counter*) perulangan. Suatu variabel yang nilainya harus berubah agar perulangan dapat terjadi dan pada akhirnya membatasi jumlah perulangan yang dapat dilaksanakan.

Bentuk Perulangan pada Python

1. Perulangan For

Perulangan yang mengerjakan “bagian pernyataan yang sama” secara berulang-ulang berdasarkan syarat atau kondisi yang ditentukan.

2. Perulangan While

Perulangan yang mengerjakan perintah selama kondisinya bernilai benar.

3. Loop bersarang (Nested Loop)

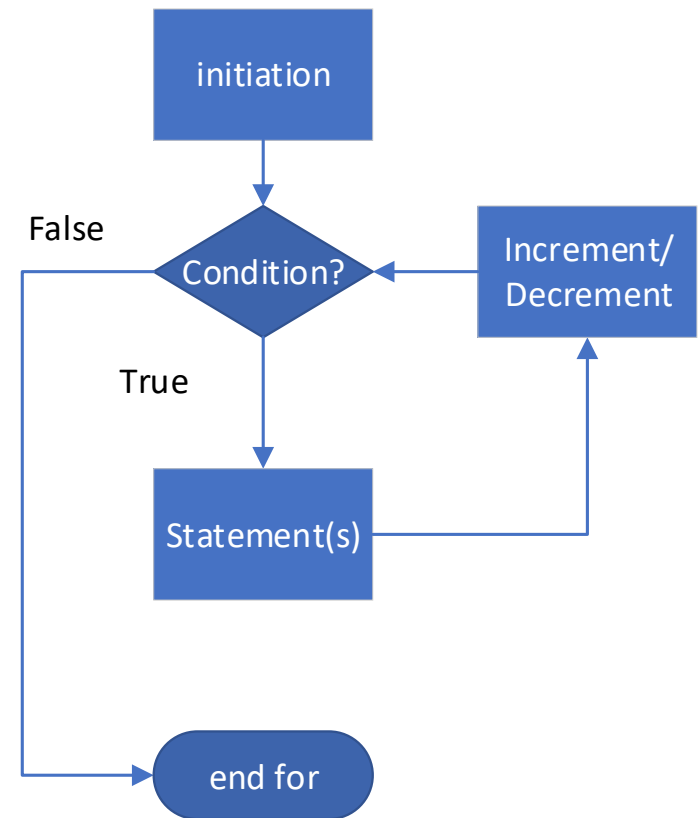
Perulangan di dalam perulangan.

1. PERULANGAN FOR

Bentuk Umum :

For variabel in range :
statements

Flowchart - For



Menampilkan Deret Bilangan

Input: -

Output: 0 1 2 3 ...

Algoritma deret

Deklarasi

n, i : integer

Begin

read(n)

for i in range n

write(i)

End

Program Python Untuk

Deret Bilangan:

```
#Menampilkan Deret Bilangan
```

```
n=int(input('Banyak Data : '))
```

```
for i in range (n):
```

```
    print(i)
```

Output Program:

Banyak Data : 5

0 1 2 3 4

Note: pada Python perulangan dengan fungsi Range dimulai dari 0

Menampilkan Deret Bilangan

Penjelasan algoritma:

Mula-mula masukkan nilai n misal: 5, kemudian terjadi perulangan dengan $i=0$ sampai n . Pengulangan dikerjakan selama kondisi bernilai true. Kemudian write (i) menghasilkan keluaran 0 dst, proses **diulang lagi sampai dengan i kurang dari nilai n .**

Tabel Penyimpanan deret

Perintah	n	$i=0..n$	i	Output
Read(n)	5			
$i=0$: write(i)		true	0	0
$i=1$: write(i)		true	1	1
$i=2$: write(i)		true	2	2
$i=3$: write(i)		true	3	3
$i=4$: write(i)		true	4	4
$i=5$: write(i)		false		

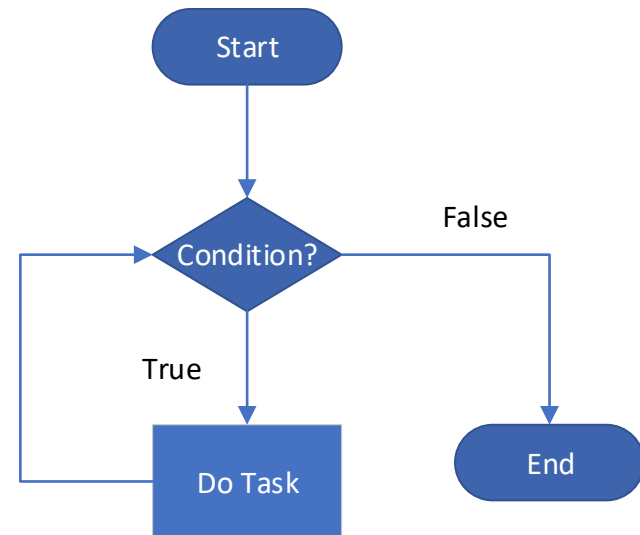
2. Perulangan While

Perulangan akan terus dilaksanakan selama kondisi bernilai True / benar.

Flowchart Perulangan while

Bentuk Umum :

**while kondisi:
statement(s)**



Perulangan While (Lanjutan)

1. Ada instruksi yang berkaitan dengan kondisi sebelum masuk ke while sehingga kondisi ini benar (terpenuhi) dan pengulangan bisa dilaksanakan.
2. Ada suatu instruksi di antara instruksi-instruksi yang diulang yang mengubah nilai variabel perulangan agar pada saat kondisi perulangan tidak terpenuhi sehingga perulangan berhenti.

Menampilkan Deret Bilangan Genap

Input : -

Output : 2, 4, 6, ...

Algoritma deret

Deklarasi

n, x : integer

Begin

read(n)

$x \leftarrow 2$

while x <= n do

write(x)

$x \leftarrow x + 2$

End

Program Python Menampilkan
Deret - Bilangan:

```
#Menampilkan Deret Bilangan  
Genap
```

```
n=int(input('Masukkan Nilai N : '))
```

```
x=2
```

```
while x<=n:
```

```
    print(x,end=" ")
```

```
    x=x+2
```

Output Program:

Masukkan Nilai N : 8

2 4 6 8

Menampilkan Deret Bilangan Genap

Mula-mula inputkan nilai $n = 8$, kemudian x diberi nilai 2, setelah itu x dibandingkan dengan n . Jika $(x \leq n)$ bernilai benar maka x ditampilkan kemudian x ditambah 2 dan menghasilkan nilai x baru. Setelah itu data kembali keatas dan menguji $(X \leq N)$ bernilai benar, jika ya proses yang sama dgn sebelumnya dilakukan kembali. Demikian seterusnya sampai $(x \leq n)$ bernilai salah

Tabel Penyimpanan Deret

Perintah	$X \leq n$	N	X	output
Read(n)		8		
$X \leftarrow 2$			2	
x=2: write(x) $x \leftarrow x+2$	true		4	2
x=4: write(x) $x \leftarrow x+2$	true		6	4
x=6: write(x) $x \leftarrow x+2$	true		8	6
x=8: write(x) $x \leftarrow x+2$	true		10	8
x=10: write(x) $x \leftarrow x+2$	False			

Algoritma while menampilkan angka 1 hingga 15

Algoritma Perulangan_while
{mencetak angka 1 hingga 15}

Deklarasi
angka = 1

Deskripsi
while angka <= 15:
 cetak angka
 angka \leftarrow angka + 1

Program Mencetak bilangan 1-15

```
# Perulangan While  
angka = 1  
  
while angka <= 15:  
    print ("Bilangan ke-: ", angka)  
    angka = angka + 1  
  
print ("Terima Kasih")
```

Hasil Running :

```
Bilangan ke-: 1  
Bilangan ke-: 2  
Bilangan ke-: 3  
Bilangan ke-: 4  
Bilangan ke-: 5  
Bilangan ke-: 6  
Bilangan ke-: 7  
Bilangan ke-: 8  
Bilangan ke-: 9  
Bilangan ke-: 10  
Bilangan ke-: 11  
Bilangan ke-: 12  
Bilangan ke-: 13  
Bilangan ke-: 14  
Bilangan ke-: 15  
Terima Kasih
```

Algoritma Cetak Bilangan menurun

Algoritma while_mencetak angka menurun
#mencetak angka 10 sampai angka 1

Deklarasi

bil = 10

Deskripsi

while bil > 0:

 cetak bil

 bil = bil - 1

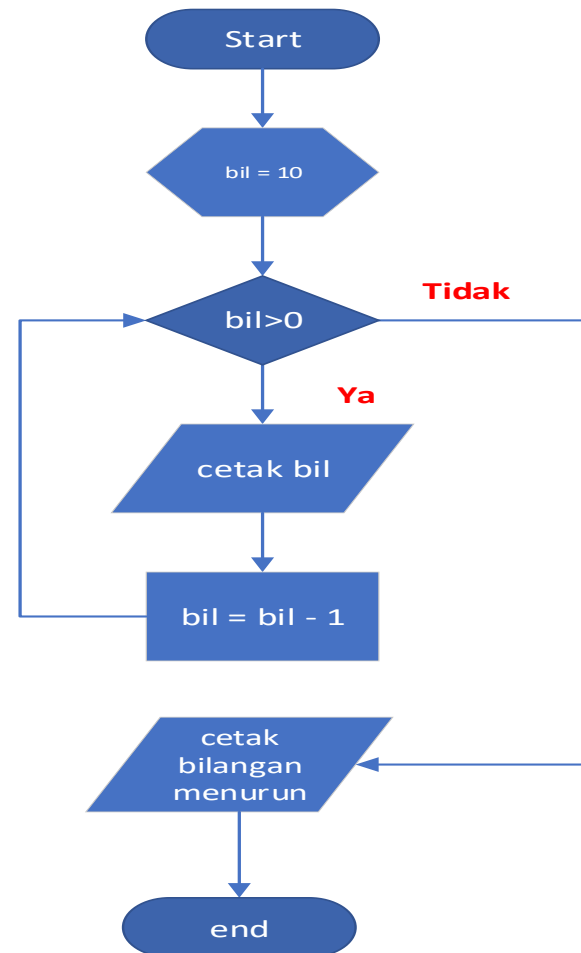
Program Cetak Bilangan Menurun

```
#Perulangan While  
#Mencetak bilangan 10 sampai 1  
bil = 10  
  
while (bil > 0):  
    print(bil, end=' ')  
    bil = bil - 1  
print ("\n Mencetak Bilangan menurun")
```

Hasil Running:

10 9 8 7 6 5 4 3 2 1

Mencetak Bilangan Menurun



Program Python Menentukan Bilangan Prima

```
#input bilangan
bilangan = int(input("Masukkan Bilangan : "))
#bilangan prima harus lebih besar dari 1
if bilangan > 1:
    for i in range(2,bilangan):
        if (bilangan % i) == 0:
            print(bilangan, "bukan bilangan prima")
            print(i, "kali", bilangan//i, "=", bilangan)
            break
    else:
        print(bilangan,"adalah bilangan prima")
else:
    print(bilangan, "bukan bilangan prima")
```


Program Python Lanjutan

Hasil Running:

Masukkan Bilangan : 137

137 adalah bilangan prima

Masukkan Bilangan : 147

147 bukan bilangan prima

3 kali 49 = 147

Perintah BREAK

Berfungsi untuk keluar dari suatu loop for atau while.
Bentuk Umumnya adalah:

```
.....
```

```
.....
```

```
break
```

```
.....
```

```
.....
```

Perintah Break Pada Python

```
#Perintah break pada perulangan for  
#Program akan keluar setelah mencetak angka sampai 6 karena  
perintah break
```

```
bil = 6
```

```
for i in range(0,10):  
    print(i)  
    if i is bil:  
        break
```

Note:

Looping akan dikerjakan terus sampai dipaksa keluar oleh instruksi **break**;

Program Python Lanjutan

Hasil Running:

0
1
2
3
4
5
6



Program keluar setelah mencetak angka 6 karena instruksi break

Perintah Continue

Fungsi Continue akan melakukan pengulangan mulai dari awal lagi.

```
#penggunaan continue pada while
bil = 0
pilihan = 'y'
while (pilihan != 'n'):
    bil = int(input("Masukkan bilangan dibawah 50: "))

    if (bil > 50):
        print("Bilangan melebihi angka 50, Silahkan diulangi.")
        continue
    print("Pangkat dua dari bilangan ini adalah: ",bil*bil)
    pilihan = input("Apakah Anda ingin mengulang kembali (y/n)? ")
```

Program Continue lanjutan

Hasil Running:

Masukkan bilangan dibawah 50: 20

Pangkat dua dari bilangan ini adalah: 400

Apakah Anda ingin mengulang kembali (y/n)? y

Masukkan bilangan dibawah 50: 36

Pangkat dua dari bilangan ini adalah: 1296

Apakah Anda ingin mengulang kembali (y/n)? y

Masukkan bilangan dibawah 50: 70

Bilangan melebihi angka 50, Silahkan diulangi.

Masukkan bilangan dibawah 50: 25

Pangkat dua dari bilangan ini adalah: 625

Apakah Anda ingin mengulang kembali (y/n)? n

3. Nested Loop (Loop Bersarang)

Bentuk Umum Nested While:

while kondisi:

 while kondisi:

 statement(s)

 statement(s)



Nested Loop Lanjutan

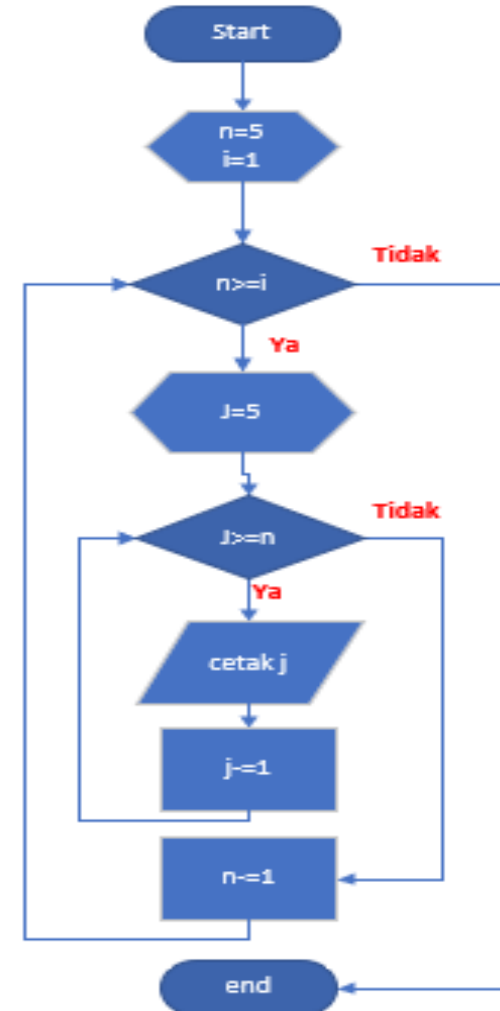
Bentuk Umum Nested For:
for variabel in range:
 for variabel in range:
 statement(s)
 statement(s)



Mencetak Bilangan Menurun

Mencetak bilangan menurun
dengan pola angka sbb:

5
5 4
5 4 3
5 4 3 2
5 4 3 2 1



Mencetak Bilangan Menurun Lanjutan

n=5 i=1 while n>=i j=5 while (j >=n) cetak j j = j-1 n = n-1
 5>=1? Ya 5>=5? Ya cetak j=5 j = 5-1=4
 4>=1? Ya j=4 4>=5? Tidak - n = 5-1=4

n=4 i=1 4>=1? Ya j=5 5>=4? Ya cetak j=5 j=5-1=4 j=4
 4>=4? Ya cetak j=5,4 j=4-1=3 j=3
 3>=4? Tidak - - n=4-1=3

n=3 i=1 3>=1? Ya j=5 5>=3? Ya cetak j=5 j=5-1=4
 4>=3? Ya cetak j=5,4 j=4-1=3
 3>=3? Ya cetak j=5,4,3 j=3-1=2
 2>=3? Tidak - - n=3-1=2

n=1 i=1 1>=1? Ya j=5 5>=1? Ya cetak j=5 j=5-1=4
 4>=1? Ya cetak j=5,4 j=4-1=3
 3>=1? Ya cetak j=5,4,3 j=3-1=2
 2>=1? Ya cetak j=5,4,3,2 j=2-1=1
 1>=1? Ya cetak j=5,4,3,2,1 j=1-1=0
 0>=1? Tidak - - n=1-1=0

n=0 i=1 0>=1? Tidak → karena kondisi sudah tidak dipenuhi maka perulangan tidak dilanjutkan.

Source Program:

```
n = 5
i = 1
while n >= i:
    j = 5
    while j >= n:
        print(j, end=" ")
    j -= 1
    print()
    n -= 1
```

Program Mencetak Bil. Prima 1 - 50 dengan Nested While

```
#Program Menggunakan Nested While
#Untuk mencetak bilangan prima antara 1 sampai 50
i = 2
while(i < 50):
    j = 2
    while(j <= (i/j)):
        if not(i%j): break
        j = j + 1
    if (j > i/j) : print (i, "adalah Bilangan Prima")
    i = i + 1
print("Terima Kasih")
```

Program Python Nested While Lanjutan

Hasil Running:

2 adalah Bilangan Prima
3 adalah Bilangan Prima
5 adalah Bilangan Prima
7 adalah Bilangan Prima
11 adalah Bilangan Prima
13 adalah Bilangan Prima
17 adalah Bilangan Prima
19 adalah Bilangan Prima
23 adalah Bilangan Prima
29 adalah Bilangan Prima

31 adalah Bilangan Prima
37 adalah Bilangan Prima
41 adalah Bilangan Prima
43 adalah Bilangan Prima
47 adalah Bilangan Prima
Terima Kasih

Studi Kasus 1

1. Membuat Program untuk mencetak bilangan genap 1 sampai 10:

```
for i in range(2,12,2):  
    print(i)
```

Hasil Running

2
4
6
8
10

Studi Kasus 2

2. Membuat program menjumlahkan Bilangan 1 sampai 10.

```
jum = 0
```

```
for i in range(10):
```

```
    i = i + 1
```

```
    print(i)
```

```
    jum = jum + i
```

```
print()
```

```
print("Jumlah Bilangan 1 - 10 adalah: ",jum)
```

Hasil Running:

1 2 3 4 5 6 7 8 9 10

Jumlah Bilangan 1-10 adalah 55

Studi Kasus 3

3. Buatlah program untuk menggambar segitiga siku-siku dengan melakukan masukan bilangan bulat.

Format Masukan & keluaran:

Masukan adalah bilangan bulat dengan range : $1 \leq N \leq 100$.

Keluaran program adalah karakter '*' yang menggambarkan pola segitiga siku-siku.

Program Python

Masukan	Keluaran
5	<pre> * * * * * * * * * * * * * * *</pre>
10	<pre> *</pre>

Program Python

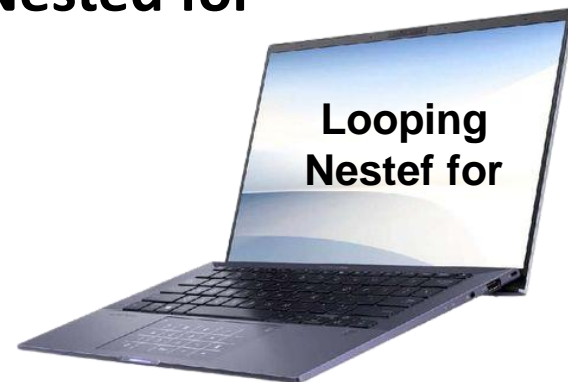
#Masukan Bilangan Bulat Integer pada variabel n

```
n = int(input("Masukan Bilangan/karakter : "))
```

#Lakukan perulangan nested for untuk menghasilkan pola siku-siku

```
for i in range(0, n):  
    for j in range(0, i + 1):  
        print('* ', end="")  
    print("")
```

Nested for



Tugas Berkelompok

1. Buatlah program untuk menggambar pola segitiga dengan sebuah karakter. Karakter dapat berupa huruf atau karakter khusus seperti: *, #, %, @, &, dan lain-lain.

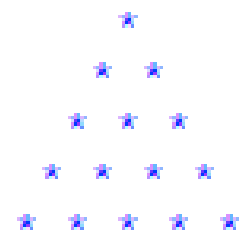

Format Masukan & keluaran:

Masukan terdiri dari dua baris, baris pertama untuk menginput bilangan bulat N dengan : $1 \leq N \leq 100$. baris kedua masukan sebuah karakter.

Keluaran program adalah huruf/karakter khusus yang menggambarkan pola segitiga sama kaki

Tugas Lanjutan

Contoh Masukkan/Keluaran

Masukan	Keluaran
5 *	
10 \$	

Tugas Lanjutan

2. Buatlah program tentang lagu anak ayam sebagai berikut:

Tek kotek kotek kotek, anak ayam turun berkotek
anak ayam turunlah 5 mati satu tinggalah 4
anak ayam turunlah 4 mati satu tinggalah 3
anak ayam turunlah 3 mati satu tinggalah 2
anak ayam turunlah 2 mati satu tinggalah 1
anak ayam turunlah 1 mati satu tinggalah induknya

Format Masukan & keluaran:

Masukan dari program adalah bilangan bulat N dengan batasan: $1 \leq N \leq 100$.

Keluaran program adalah berupa baris lirik lagu yang akan diuraikan sesuai dengan bilangan bulat yang diinput.

Tugas Lanjutan

Contoh Masukan/Keluaran

Masukan	Keluaran
5	tek kotek kotek kotek, anak ayam turun berkotek anak ayam turunlah 5 mati satu tinggalah 4 anak ayam turunlah 4 mati satu tinggalah 3 anak ayam turunlah 3 mati satu tinggalah 2 anak ayam turunlah 2 mati satu tinggalah 1 anak ayam turunlah 1 mati satu tinggalah induknya
1	tek kotek kotek kotek, anak ayam turun berkotek anak ayam turunlah 1, mati satu tinggalah induknya