 Assalamu'alaikum wr.wb Nama saya IKANG FAUZI, kali ini saya akan memberikan tutorial tentang program looping pada bahasa pemrograman python. fungsi dari looping adalah memberikan perintah kepada komputer menggunakan sintax tertentu untuk menjalankan program secara berulang-ulang selama kondisi tertentu. Looping akan terus terjadi selama kondisi terpenuhi. Dalam bahasa pemrograman python,  looping dilakukan dengan perintah *for* dan *while,*dimana keduanya memberikan looping untuk melakukan pengulangan program dengan cara yang berbeda.  
  
Sedikit informasi bahwa looping adalah salah satu dari struktur kendali seperti [operator](http://tutorallprogramming.blogspot.com/2017/11/operator-pada-bahasa-pemrograman-python.html) dan [percabangan](https://tutorallprogramming.blogspot.co.id/2017/12/contoh-program-percabangan-pada-program.html). Artinya looping tak kalah pentingnya dengan kedua struktur kendali tersebut, bahkan bisa dibilang looping sangat penting untuk dipahami. Maka dari itu, disini saya akan menjelaskan tentang looping.

Seperti yang saya jelaskan sebelumnya, looping pada python memiliki 3 Macam, Dan ketiganya memiliki cara yang berbeda-beda. Berikut adalah macam-macam looping pada bahasa pemrograman python.

|  |  |  |
| --- | --- | --- |
| **No** | **Tipe Loop** | **Penjelasan** |
| 1. | **While loop** | Perulangan dilakukan selama keadaan masih TRUE, akan dilakukan pengecekan kondisi terlebih dahulu sebelum blok kode dieksekusi. |
| 2. | **For loop** | Eksekusi terhadap blok kode dilakukan berulang kali sesuai dengan variabel yang mengatur perulangan. |
| 3. | **Nested loop** | Kita bisa mengkombinasikan perulangan, dimana ada perulangan di dalam perulangan. |

Mungkin masih banyak perulangan yang ada, Namun pada kesematan ini, saya hanya akan memberikan tutorial dengan ketiga perulangan tersebut. Karena ketiga looping yang saya sebutkan diatas adalah looping yang sering digunakan oleh kebanyakan programmer. Berikut ini merupakan penjelasan dari ketiga macam perulangan.

**1. FOR LOOP**

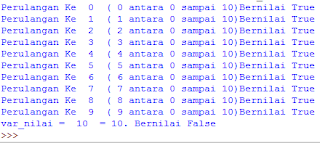
FOR LOOP adalah sintax untuk memberikan perintah pada komputer untuk melakukan program berulang-ulang selama kondisi bernilai True. Misal perulangan dilakukan selama nilai belum mencapai 5. Artinya jika nilai masih bernilai 1, kondisi tersebut bernilai True,  
  
Agar lebih jelas, lihatlah contoh program perulangan menggunakan FOR dibawah ini

#Deklarasi variabel  
var\_nilai = 0  
  
#Perulangan FOR  
**for** var\_nilai in range(0,10) :  
    print("Perulangan Ke ",var\_nilai," (",var\_nilai,"antara 0 sampai 10)Bernilai True")  
  
#diluar perulangan  
print("var\_nilai = ",**int**(var\_nilai)+1," = 10. Bernilai False")

Penjelasan Program

* Seperti yang kita ketahui, deklarasi variabel *var\_nilai* adalah 0, artinya *var\_nilai* = 0
* *range(0,10)* artinya melakukan perulangan selama *var\_nilai* bernilai antara 0 sampai 10(0 dan 10 tidak termasuk)
* ketika var\_nilai bernilai 1-9, maka program akan memunculkan output "Perulangan ke (*var\_nilai*)", Hal ini menunjukkan bahwa perulangan masih bernilai **True** (*var\_nilai* antara 0 sampai 10)
* Ketika var\_nilai bernilai 10, artinya perulangan sudah bernilai False, jadi perulangan secara otomatis dihentikan oleh program, sehingga program menampilkan output **#diluar perulangan**.

Dan dibawah ini adalah hasil outputnya



FOR Loop memang perulangan yang paling mudah, Bukan karena Syntax yang pendek, sebenarnya untuk looping, FOR sendiri memiliki syntax yang panjang. Akan tetapi, FOR memberikan perintah dengan cara yang mudah dipahami.

**2. While LOOP**

Sama seperti FOR LOOP, WHILE LOOP juga memberikan perintah kepada program untuk menjalankan program secara berulang - ulang. Hanya saja WHILE LOOP lebih ringkas karena bisa disisipkan dengan [operator pada python](http://tutorallprogramming.blogspot.com/2017/11/operator-pada-bahasa-pemrograman-python.html).

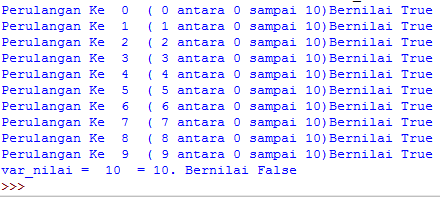
Untuk lebih jelasnya, silahkan lihat contoh perulangan menggunakan WHILE dibawah ini :

#Deklarasi variabel  
var\_nilai = 0  
  
#Perulangan WHILE  
**while** (var\_nilai < 10) :  
     print("Perulangan Ke ",var\_nilai," (",var\_nilai,"antara 0 sampai 10)Bernilai True")  
     var\_nilai +=1  
  
#Diluar Perulangan  
print("var\_nilai = ",**int**(var\_nilai)," = 10. Bernilai False")

Penjelasan Program

* Seperti yang kita ketahui, deklarasi variabel *var\_nilai* adalah 0, artinya *var\_nilai* = 0
* *(var\_nilai < 10)*adalah kondisi dimana var\_nilai bernilai kurang dari 10, dan selama kondisi itu True, maka program terus berjalan.
* *var\_nilai += 1*, artinya selama perulangan bernilai True, var\_nilai akan terus bertambah 1 hingga kondisi bernilai False.
* Ketika kondisi bernilai False(var\_nilai = 10) yang mana tidak sesuai dengan kondisi yang sudah ditentukan, maka akan muncul **#Diluar perulangan**

Dan inilah hasil dari program diatas



Itulah penjelasan mengenai WHILE LOOP dengan Looping ini, kita dapat mempersingkat program dengan seksama sehingga program bisa menjadi lebih rapi dan lebih indah.

**3. Nested LOOP**

Nested LOOP adalah lopping dalam looping. Artinya ketika kita membuat looping dengan kondisi tertentu, maka program tersebut akan melakukan looping lagi didalamnya dengan kondisi tertentu juga.  
  
Memang agak membingungkan, namun jika anda sudah melihat programnya, mungkin bisa lebih memahaminya. dan berikut ini merupakan contoh program Nested LOOP

**Nested LOOP Bersarang Menggunakan FOR**

#Deklarasi variabel  
var\_nilai = 0  
var\_i = 1  
  
#Perulangan FOR  
**for** var\_nilai in range (0,10) :  
    print("Perulangan pertama Ke ",var\_nilai)  
**for** var\_i in range (1,3) :  
        print(" Perulangan ke ", var\_nilai,", ",var\_i)  
    #diluar perulangan var\_i  
    var\_i = 1  
#diluar\_perulangan var\_nilai  
print("var\_nilai = ",**int**(var\_nilai)+1," = 10. Bernilai False")

Penjelasan Program

* *for var\_nilai in range (0,10)*. artinya selama var\_nilai nilainya berada diantara 0 sampai 10, maka program akan terus berjalan.
* *for var\_i in range(1,3)*. ini adalah perulangan kedua. Dimana jika var\_i nilainya antara 1 sampai 3, artinya kondisi terpenuhi, jika tidak, kembali ke perulangan pertama.
* *var\_i = 1*. Pada bagian ini, ketika kembali ke perulangan pertama, nilai var\_i dikembalikan menjadi 1.

**Contoh Nested LOOP Menggunakan WHILE**

#Deklarasi variabel  
var\_nilai = 0  
var\_i = 1  
  
#Perulangan WHILE  
**while** (var\_nilai < 10) :  
     print("Perulangan pertama Ke ",var\_nilai)  
**while**(var\_i < 3) :  
         print(" Perulangan ke ", var\_nilai,", ",var\_i)  
         var\_i +=1  
     #diluar perulangan var\_i  
     var\_i = 1  
     var\_nilai +=1  
#diluar\_perulangan var\_nilai  
print("var\_nilai = ",**int**(var\_nilai)," = 10. Bernilai False")

Penjelasan Program

* *while (var\_nilai < 10)*. Artinya adalah ketika var\_nilai bernilai kurang dari 10, maka perulangan tetap berjalan
* *while (var\_i < 3)*. Artinya ketika var\_i bernilai kurang dari 3, maka perulangan terus berjalan.
* *var\_i += 1*. Artinya menambahkan nilai var\_i. Hal ini diperlukan karena while tidak menyediakan penambahan var\_i secara otomatis. Jika tidak ditambahkan, maka nilai dari var\_i tetap 1 dan perulangan tidak akan berhenti
* *var\_i = 1*. Ketika perulangan kedua berjalan, nilai dari var\_i terus berjalan. Disini kita mengembalikan nilai var\_i menjadi 1.
* *var\_nilai += 1*. artinya menambahkan nilai var\_nilai agar bertamba 1.

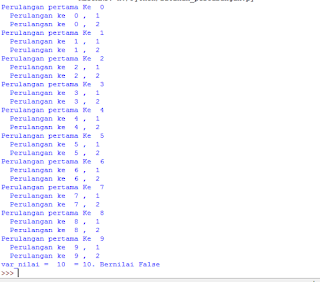
**Contoh Nested LOOP Menggunakan FOR dan WHILE**

#Deklarasi variabel  
var\_nilai = 0  
var\_i = 1  
  
#Perulangan FOR  
**for** var\_nilai in range (0,10) :  
    print("Perulangan pertama Ke ",var\_nilai)  
**while**(var\_i < 3) :  
        print(" Perulangan ke ", var\_nilai,", ",var\_i)  
        var\_i +=1  
    #diluar perulangan var\_i  
    var\_i = 1  
#diluar\_perulangan var\_nilai  
print("var\_nilai = ",**int**(var\_nilai)+1," = 10. Bernilai False")

Penjelasan Program

* *for var\_nilai in range (0,10)*. artinya selama var\_nilai nilainya berada diantara 0 sampai 10, maka program akan terus berjalan.
* *while (var\_i < 3)*. Artinya ketika var\_i bernilai kurang dari 3, maka perulangan terus berjalan.
* *var\_i += 1*. Artinya menambahkan nilai var\_i. Hal ini diperlukan karena while tidak menyediakan penambahan var\_i secara otomatis. Jika tidak ditambahkan, maka nilai dari var\_i tetap 1 dan perulangan tidak akan berhenti
* *var\_i = 1*. Ketika perulangan kedua berjalan, nilai dari var\_i terus berjalan. Disini kita mengembalikan nilai var\_i menjadi 1.

Dari ketiga Contoh Nested LOOP ditas, menghasilkan output yang sama, yaitu seperti dibawah ini :



**Penutup**

Perulangan adalah salah satu dari struktur kendali. terdapat 3 macam looping yang dibahas di postingan ini yaitu FOR, WHILE, dan Nested . Masing-masing looping memiliki cara dan tingkat kesulitan yang berbeda-beda. Demikian tutorial yang dapat saya sampaikan saat ini. Assalamu'alaikum wr.wb