



KONSEP PEMROGRAMAN

[Python 3.6.2]

Abstract

IF, IF ELSE, IF AND/OR, Nested IF

Puji Winar Cahyo

Universitas Jenderal Achmad Yani [Yogyakarta]

3. If dengan menggunakan operator AND (DAN), OR (ATAU)

Operator AND, OR sering digunakan untuk menghubungkan dua buah kondisi atau lebih masih dalam satu premis yang menghasilkan satu buah konklusi. Seperti contoh:

<u>kondisi 1</u>	<u>kondisi 2</u>
1. JIKA <u>hari ini hujan</u> DAN <u>hari ini terang</u>	
premis	
MAKA <u>saya akan berteduh kemudian jalan jalan</u>	
konklusi	

<u>kondisi 1</u>	<u>kondisi 2</u>
2. JIKA <u>hari ini hujan</u> ATAU <u>hari ini terang</u>	
premis	
MAKA <u>saya akan dirumah</u>	
konklusi	

Keterangan :

- ✓ Kondisi AND (DAN) akan menghasilkan konklusi apabila semua kondisi terpenuhi.
- ✓ Kondisi OR (ATAU) akan menghasilkan konklusi apabila salah satu kondisi terpenuhi.

4. If bersarang (Nested IF)

Nested IF digunakan apabila hasil konklusi memiliki logika/rule yang berkelanjutan yang dipengaruhi dari premis sebelumnya. Seperti contoh:

JIKA <u>hari ini hujan</u> DAN <u>saya dirumah</u> MAKA	
JIKA <u>dirumah ada playstation</u> MAKA <u>main game</u>	} konklusi statement 1
JIKA <u>dirumah ada tv</u> MAKA <u>nonton tv</u>	
JIKA <u>hari ini tidak hujan</u> DAN <u>saya dirumah</u> MAKA	
<u>pergi ke taman</u>	} konklusi statement 2

C. Implementasi

Didalam bahasa pemrograman python definisi conditional statement selalu diawali dengan IF kemudian diikuti dengan premis yang berisikan kondisi, dan hasil kondisi tersebut akan menentukan konklusi. Contoh dari penggunaan conditional statement dapat dilihat pada implementasi berikut.

1. IF Tunggal

IF tunggal digunakan pada pendefinisian if secara tunggal dan hanya dalam satu konteks, apabila ada IF tunggal yang lain dan berdiri sendiri, maka dapat disebut dengan IF tunggal yang lain dalam konteks yang berbeda.

a) Contoh pencarian bilangan ganjil

```
>>> X = 3
>>> if (X%2)==1:
>>>     print("Bilangan Ganjil")
```

b) Contoh IF yang memiliki konteks yang berbeda untuk mencari bilangan ganjil dan pencocokan angka 3

```
>>> X = 3
>>> if (X%2)==1:
>>>     print("Bilangan Ganjil")
>>> if X==3:
>>>     print("Angka tiga")
```

2. IF ELIF ELSE

IF ELIF ELSE digunakan pada logika percabangan yang terdiri lebih dari satu kondisi dan semua statement yang didefinisikan masih dalam satu konteks yang sama. Contoh:

```
>>> X = 3
>>> if (float(X).is_integer())!=True:
>>>     print("Bukan bilangan bulat")
>>> elif (X%2)==1:
>>>     print("Bilangan Ganjil")
>>> else:
>>>     print("Bilangan Genap")
```

3. Ternary IF

Ternary IF biasanya dipakai untuk menyingkat penulisan logika percabangan.

Contoh ternary IF ELSE dari:

```
>>> X = 3
>>> if (X%2)==1:
>>>     print("Bilangan Ganjil")
>>> else:
>>>     print("Bilangan Genap")
```

adalah sebagai berikut:

```
>>> X = 3
>>> print("Bilangan ganjil" if ((X%2)==1) else "Bilangan genap")
```

atau menggunakan tuple:

```
>>> X = 3
>>> print(("Bilangan genap", "Bilangan ganjil")[((X%2)==1)])
```

atau menggunakan dictionary:

```
>>> X = 3
>>> print({True:"Bilangan ganjil", False:"Bilangan genap"}[((X%2)==1)])
```

atau menggunakan lambda function:

```
>>> X = 3
>>> print((lambda:"Bilangan genap", lambda:"Bilangan ganjil")[(X%2)==1]() )
```

4. IF AND, OR

a) IF AND

Logika IF AND seringkali dipakai untuk menghubungkan dua buah kondisi atau lebih, dengan syarat dua buah kondisi atau lebih tersebut harus bernilai benar untuk menghasilkan konklusi yang dimaksud. Seperti contoh berikut:

```
>>> X = 3
>>> if ((X%2)==1) AND (X==3):
>>>     print("X adalah bilangan ganjil")
```

b) IF OR

Logika IF OR seringkali dipakai untuk menghubungkan dua buah kondisi atau lebih, dengan syarat salah satu kondisi yang terjadi harus bernilai benar untuk menghasilkan konklusi yang dimaksud. Seperti contoh berikut:

```
>>> X = 3
>>> if ((X%2)==1) OR (X==3):
>>>     print("X adalah bilangan ganjil")
```

5. NESTED IF

Logika Nested If seringkali dipakai pada logika bersyarat, logika yang menyaratkan ketergantungan pada logika sebelumnya. dapat dikatakan, untuk mencapai persyaratan selanjutnya maka dibutuhkan persyaratan sebelumnya. Seperti contoh berikut ini logika dengan color warna **ungu** memerlukan hasil kondisi logika warna **merah** :

```
>>> X = 3
>>> if ((X%2)==1):
>>>     if (X>10):
>>>         print("X diatas 10")
>>>     elif (X==3):
>>>         print("X adalah 3")
>>>     else:
>>>         print("X dibawah 10")
```

*) Contoh Program untuk mengetahui angka ganjil genap (dengan inputan dari keyboard):

```
>>> X = int(input("Masukan X : "))
>>> print(("Bilangan genap", "Bilangan ganjil")[((X%2)==1)])
```

Latihan 2

1. Buatlah program sederhana untuk mencari angka maksimal dari 2 buah angka masukan.
2. Buatlah program sederhana untuk menghitung keliling lingkaran dan luas lingkaran.

Dengan pilihan:

- ✓ Jika memasukan angka 1 maka menghitung keliling lingkaran,
- ✓ Jika memasukan angka 2 maka menghitung luas lingkaran.

Selain itu juga ada masukan nilai untuk jari – jari.