

IF Logic, Function, dan Loop

DQLab LiveClass



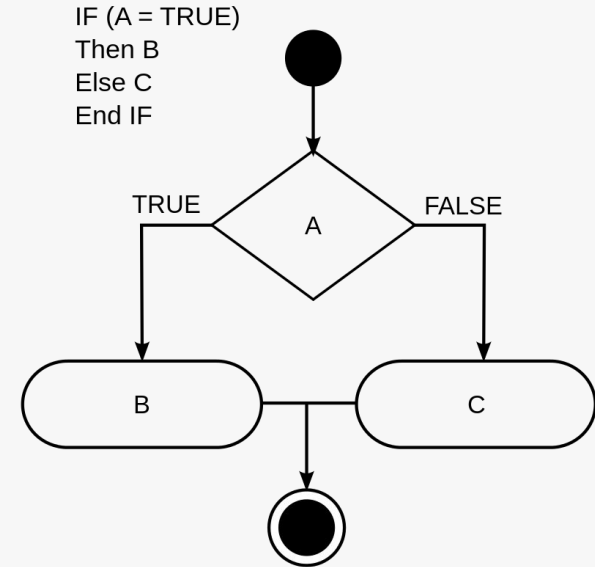
Outline

- Logika IF
- Fungsi
- Loop

IF Logic

IF Logic

- IF adalah logika yang digunakan dalam percabangan sehingga komputer dapat memilih blok kode mana yang akan dijalankan berdasarkan kondisi tertentu.



Kondisi IF

Syntax

```
if <kondisi> :  
  
    <blok kode>
```

- Komputer akan memeriksa kondisi di depan if
- Jika kondisi bernilai benar, blok kode yang ada di dalam if akan dijalankan
- Jika kondisi bernilai salah, tidak ada yang dilakukan



```
# Ganti nilai kebenaran dari variable lapar  
  
lapar = True  
  
if lapar == True:  
    print("Makan")
```

Makan



```
# Ganti nilai kebenaran dari variable lapar  
# Karena kondisi bernilai False maka "Makan" tidak ditampilkan  
  
lapar = False  
  
if lapar == True:  
    print("Makan")
```

Kondisi IF ELSE

Syntax

```
if <kondisi> :  
    <blok kode>  
else :  
    <blok kode>
```

- Komputer akan memeriksa kondisi di depan if
- Jika kondisi bernilai benar, blok kode yang ada di dalam if akan dijalankan
- Blok kode di dalam else akan dieksekusi jika sama sekali tidak ada kondisi yang benar

```
▶ lapar = False  
  
if lapar == True:  
    print("Makan")  
else:  
    print("Terus belajar")
```

```
☞ Terus belajar
```

“Terus belajar” ditampilkan karena kondisi di atasnya tidak ada yang bernilai benar

Kondisi IF ELIF

Syntax

```
if <kondisi> :  
    <blok kode>  
elif <kondisi> :  
    ...  
else :  
    <blok kode>
```

- Gunakan elif jika kita ingin mencoba beberapa kemungkinan kondisi
- Kondisi yang paling dahulu bernilai benar, blok kodenya akan dijalankan kemudian komputer akan keluar dari blok if
- Kita dapat menambahkan beberapa elif

```
▶ kondisi = 'capek'  
  
if kondisi == 'lapar' :  
    print("Makan")  
elif kondisi == 'capek' :  
    print("Istirahat")  
elif kondisi == 'ngantuk' :  
    print("Tidur")  
else :  
    print("Terus belajar")
```

```
☞ Istirahat
```

Fungsi

Fungsi

- Fungsi adalah blok kode yang memiliki tugas yang spesifik
- Fungsi cukup sekali didefinisikan
- Fungsi dapat digunakan secara berulang
- Syntax :

```
def <nama_fungsi> (parameter) :  
    <blok kode>  
  
    return <kode atau variable>
```

Contoh Fungsi

- Fungsi bisa tidak memiliki parameter dan return, contoh:

```
def tulis_hello():  
    print('Hello')
```

- Untuk menjalankan fungsi di atas cukup eksekusi tulis_hello()
- Jika fungsi tidak memiliki return maka setelah dieksekusi, fungsi tersebut tidak memiliki nilai atau None

Contoh fungsi

```
def luas_persegi(panjang, lebar):
```

```
    return panjang*lebar
```

- Fungsi di atas digunakan untuk menghitung luas persegi dengan mengembalikan nilai berupa hasil perkalian panjang dengan lebar

Loop

Loop pada Python

- Terdapat dua metode looping/perulangan di python yaitu For loop dan While loop
- For loop digunakan untuk iterasi terhadap sequence (list, tuple, dictionary, set, string)
- Sedangkan While loop adalah metode iterasi di mana blok kode akan terus dieksekusi selama kondisi bernilai benar

For

- Syntax

```
for <variable> in <sequence>:  
    <blok kode>
```

- Contoh

```
for x in ['apel','mangga','anggur']:  
    print(x)
```

Pada contoh di atas, kode akan menampilkan x sebanyak 3x di mana x adalah nama-nama buah yang ada di dalam list.

Kode akan dijalankan secara berulang sampai element terakhir pada sequence

Fungsi range()

- Misalkan kita ingin membuat iterasi sebanyak bilangan tertentu, maka sequence dapat dibangun menggunakan fungsi range
- `range(n)` akan menghasilkan bilangan bulat dari 0 sampai $n-1$
- `range(a,b)` akan menghasilkan bilangan bulat dari a sampai $b-1$
- `range(a,b,s)` akan menghasilkan bilangan bulat dari a hingga $b-1$ dengan loncat sebanyak s

While

- Jika dengan for loop kita dapat mengetahui dengan pasti berapa kali iterasi akan berlangsung (yaitu sepanjang sequence), pada while kode akan terus dijalankan selama kondisi memenuhi
- Syntax

while <kondisi>:

 <blok_kode>

Contoh while

```
x = 100
```

```
while x > 0 :
```

```
    print(x)
```

```
    x = x - 1
```

- Pada contoh di samping, nilai x akan ditampilkan dan dikurangi 1 selama $x > 0$
- Maka dari itu, pastikan nilai x selalu berubah pada setiap iterasi untuk menghindari infinite loop

break dan continue

- break digunakan untuk keluar dari loop meskipun iterasi belum selesai
- continue digunakan untuk menghentikan iterasi yang sedang berlangsung dan lanjut ke iterasi selanjutnya
- pass pada dasarnya tidak melakukan apa-apa, pass hanya sebuah placeholder saat kita akan membuat looping tapi belum memiliki kode untuk dituliskan setelahnya

Terimakasih!

Thanks!

