

# Bab II

## Conditional Statement

*Conditional statement* (yang dalam bahasa Indonesia disebut pernyataan bersyarat atau pengkondisian) adalah fitur dari bahasa pemrograman yang melakukan perhitungan atau tindakan yang berbeda tergantung pada kondisi yang ditentukan programmer. Pengkondisian membuat program memiliki cabang yang masing-masing akan tereksekusi jika persyaratannya terpenuhi.

Pengkondisian biasanya dalam bentuk *if statement*, yaitu salah satu fitur utama dari bahasa pemrograman, tidak terkecuali Python. Hampir tidak ada bahasa pemrograman yang tidak memiliki *if statement* dan hampir tidak ada cara untuk memprogram tanpa cabang dalam aliran kode (setidaknya jika kode tersebut perlu memecahkan masalah yang kompleks).

Bahasa pemrograman seperti C, C++, dan Java setidaknya memiliki fitur conditional statement seperti *if statement* dan *switch case*, namun dalam bahasa Python *switch case* baru diadaptasi pada versi 3.10 keatas yang dikenal sebagai “*structural pattern matching*” atau “*match case*”. Artinya, pada versi sebelumnya pengkondisian hanya dapat dilakukan menggunakan *if statement*. Ekspresi terdiri dari satu atau beberapa operator perbandingan dan operator logika yang akan menghasilkan nilai True (benar) atau False (salah).

**Tabel Operator Perbandingan**

| Operator | Arti   | Contoh | Hasil |
|----------|--|--------|-------|
| >        | Lebih besar dari – True jika operand kiri lebih besar dari kanan                         | 5 > 6  | False |
| <        | Kurang dari – True jika operand kiri lebih kecil dari kanan                              | 5 < 6  | True  |
| ==       | Sama dengan – True jika kedua operand bernilai sama                                      | 5 == 5 | True  |
| !=       | Tidak sama dengan – True jika kedua operand tidak bernilai sama                          | 5 != 5 | False |
| >=       | Lebih besar atau sama dengan – True jika operand kiri lebih besar atau sama dengan kanan | 5 >= 3 | True  |

|    |  |        |      |
|----|--|--------|------|
| <= | Lebih kecil atau sama dengan – True jika operand kiri lebih kecil atau sama dengan kanan | 5 <= 5 | True |
|----|--|--------|------|

**Tabel Operator Logika**

| Operator | Arti  | Contoh        | Hasil |
|----------|---|---------------|-------|
| and      | True jika kedua operand bernilai True                 | True and True | True  |
| or       | True jika kedua atau salah satu operand bernilai True | True or False | True  |
| not      | True jika operand bernilai False                      | not False     | True  |

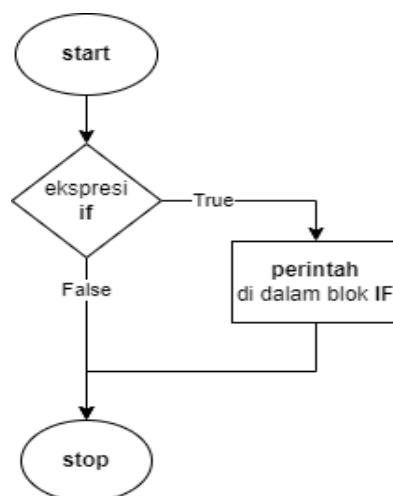
## A. If-else Statement

### 1. If statement

*If statement* adalah jenis pengkondisian yang paling mendasar, di mana kode dieksekusi apabila ekspresi terpenuhi atau bernilai *True* (benar). Pernyataan dari *if statement* harus memiliki *indent* minimal sepanjang satu spasi di awal tiap baris kode. Suatu pernyataan dapat berupa satu baris atau satu blok kode.

```
if ekspresi:
    # Pernyataan/statement(dieksekusi jika ekspresi #
    bernilai True)
```

**Alur Program :**



**Contoh 1**

```
angka = -5  
  
if angka > 0:  
    print(angka, "adalah bilangan positif")  
  
print("pernyataan ini bernilai benar!")
```

**Output:**

```
> 5 adalah bilangan positif.  
> pernyataan ini bernilai True!.
```

**Contoh 2**

```
angka = 5  
  
if angka < 0:  
    print(angka, "adalah bilangan negatif") print("pernyataan  
ini bernilai False!")
```

**Output:**

```
> pernyataan ini bernilai False!.
```

**Contoh 1.3**

```
angka = -2  
  
if angka > 0:  
    print(angka, "adalah bilangan positif") if  
angka < 0:  
    print(angka, "adalah bilangan negatif")  
  
print("pernyataan ini bernilai benar!")
```

### Output:

> -2 adalah bilangan negatif.

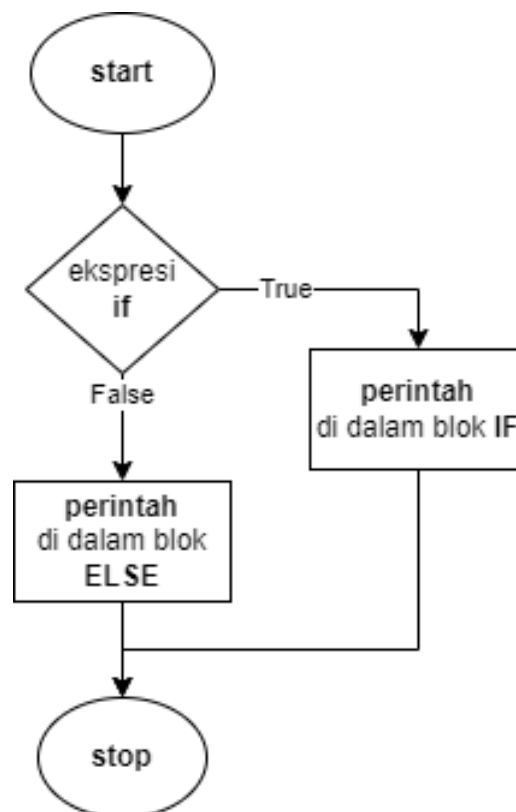
## 2. if-else statement

Pernyataan “else” digunakan ketika bagian benar dan salah dari kondisi tertentu ditentukan untuk dieksekusi. Ketika kondisinya benar, pernyataan di dalam blok if dieksekusi; jika kondisinya salah, program pada blok else akan dieksekusi.

Pernyataan if...else dalam Python memiliki sintaks berikut:

```
if ekspresi:  
  
    # pernyataan/statement(dieksekusi jika #  
    ekspresi bernilai True)  
else:  
  
    # pernyataan/statement(dieksekusi jika #  
    ekspresi bernilai False
```

### Alur program:



### Contoh 1

```
angka = 5  
if angka >= 0:  
    print("bilangan positif atau nol") else:  
    print(angka, "adalah bilangan negatif")
```

### Output:

```
> bilangan positif atau nol
```

### Contoh 2.2

```
x = 5  
y = 20  
if x == y:  
    print("x dan y bernilai sama")  
else:  
    print("x tidak sama dengan y")
```

### Output:

```
> x tidak sama dengan y
```

## 1. If-elif-else statement

Pernyataan elif memungkinkan Anda untuk memeriksa beberapa ekspresi dan mengeksekusi blok kode segera setelah salah satu kondisi mengevaluasi ke True. Dalam hal ini, kondisi if dievaluasi terlebih dahulu. Jika salah, pernyataan elif akan dieksekusi, jika itu juga salah, pernyataan else akan dieksekusi.

Pernyataan If...Elif..else dalam Python memiliki sintaks berikut:

if ekspresi:

#dieksekusi jika ekspresi bernilai True#

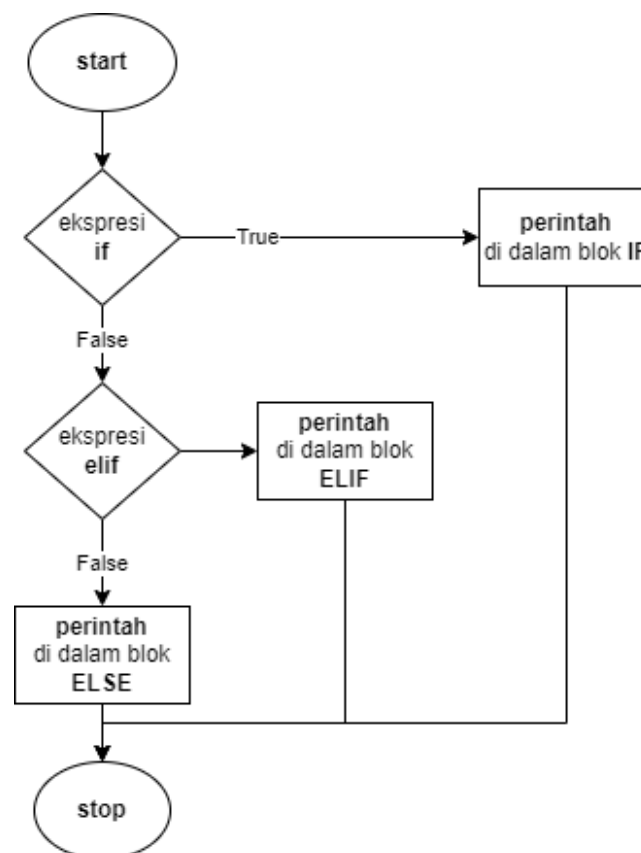
elif ekspresi:

#dieksekusi jika ekspresi (elif) bernilai True

else:

#dieksekusi jika ekspresi bernilai False

**Alur program:**



**Contoh 3**

angka = 0

if angka > 0:

```
print(angka, "adalah bilangan positif")
elif angka == 0:
    print(angka, "adalah bilangan nol")
else:
    print(angka, "adalah bilangan negatif")
```

**Output:**

```
> 0 adalah bilangan nol
```

## Contoh 2

```
angka = 0
if angka > 0:
    print(angka, "adalah bilangan positif")
elif angka == 0:
    print(angka, "adalah bilangan nol")
else:
    print(angka, "adalah bilangan negatif")
```

**Output:**

```
> 0 adalah bilangan nol
```

## 2. Nested IF Statement

Pernyataan IF bersarang adalah pernyataan di mana pernyataan If terletak di dalam pernyataan If lainnya. Ini digunakan ketika variabel harus diproses lebih dari sekali. Pernyataan if, if-else, dan if...elif...else dapat digunakan dalam program.

Pernyataan if bersarang dalam Python memiliki sintaks berikut:

if ekspresi:

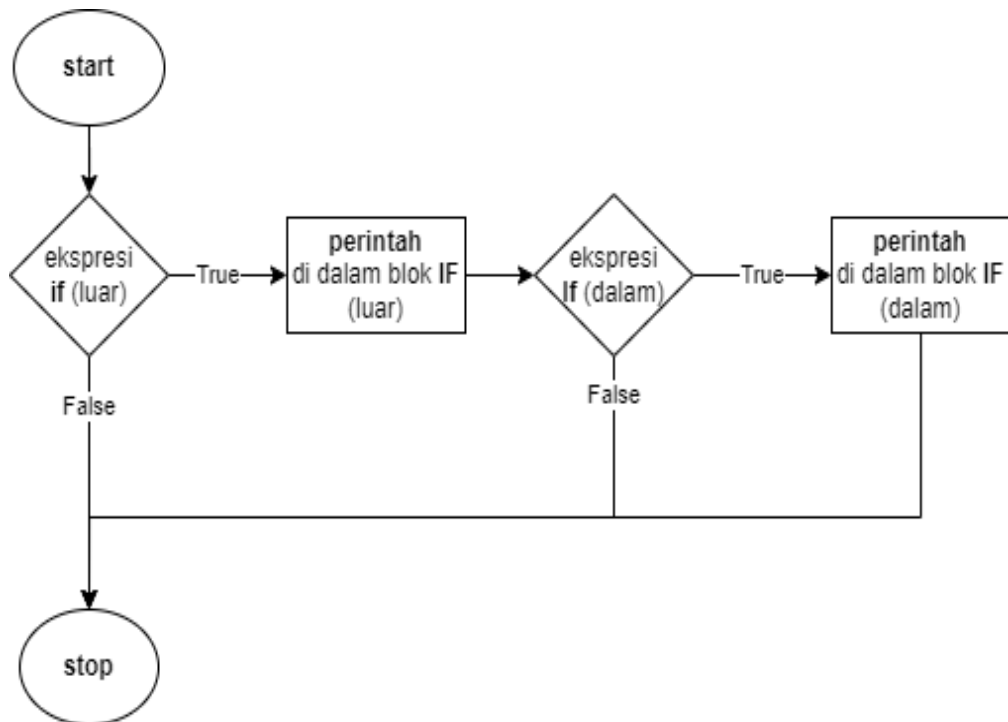
#dieksekusi jika ekspresi bernilai True# if

ekspresi:

# dieksekusi jika ekspresi if (luar) dan

# if (dalam) bernilai True

**Alur program:**



**Contoh 1**

angka = 0

if angka >= 0:

```
print(angka, "adalah bilangan positif") if
angka == 0:
    print(angka, "adalah bilangan nol")
else:
    print(angka, "adalah bilangan negatif")
```

**Output:**

```
> 0 adalah bilangan nol
```

**Contoh 2**

```
angka = 4
if angka >= 0:
    print(angka "adalah bilangan positif") if
(angka % 2) == 0:
        print(angka, "adalah bilangan genap
        positif")
else:
    print(angka, "adalah bilangan negatif")
```

**Output:**

```
> 0 adalah bilangan nol
```

**3. Shorthand if statement**

Shorthand if statement digunakan ketika hanya satu statement yang perlu dieksekusi di dalam blok if. Pernyataan ini dapat disertakan di baris yang sama dengan pernyataan If. Pernyataan

Short Hand if dalam Python memiliki sintaks berikut:

```
if kondisi: statement
```

**Contoh program:**

```
a = 123  
  
if a > 100: print(a, "lebih besar dari seratus")
```

**Output:**

```
> 123 lebih besar dari 100
```

#### 4. ShortHand if-else statement

Shorthand if-else digunakan untuk menyebutkan pernyataan If-else dalam satu baris di mana hanya ada satu pernyataan untuk dieksekusi di blok if dan else.

Short Hand if-else dalam Python memiliki sintaks berikut:

```
statement (True) if kondisi else statement (False)
```

**Contoh program:**

```
a = 123  
  
print(a, " besar") if a > 100 else print(a, " kecil")
```

**Output:**

```
> 123 besar
```

## B. Match Case

Pernyataan switch mengevaluasi ekspresi, mencocokkan nilai ekspresi terhadap serangkaian klausa kasus, dan mengeksekusi pernyataan setelah klausa kasus pertama dengan nilai yang cocok, hingga pernyataan break ditemukan. Klausa default dari pernyataan switch akan dilompati jika tidak ada kasus yang cocok dengan nilai ekspresi.

Short Hand if dalam Python memiliki sintaks berikut:

```
match variabel:

    case nilai-1:

        statement-1

    case nilai-2:

        statement-2

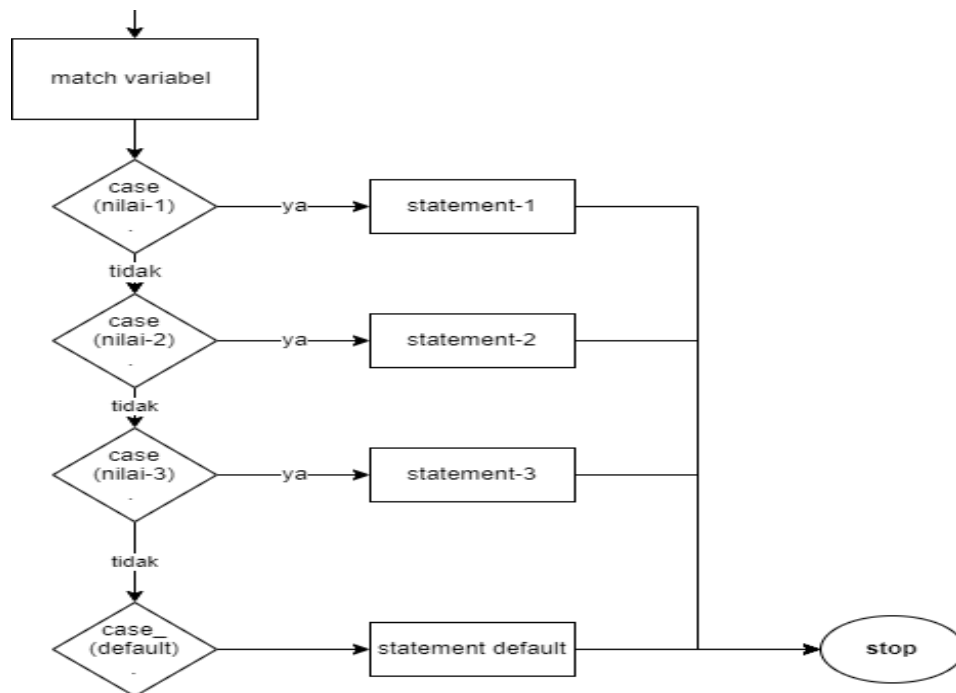
    case nilai-...:

        statement-...

    case _:

        statement default (jika tidak ada case yang sesuai)
```

**Alur program:**



**Contoh program:**

```
bahasa = "python"
match bahasa:
    case "JavaScript":
        print("kamu akan menjadi web developer.")

    case "Python":
        print("kamu akan menjadi Data Scientist")

    case "PHP":
        print("kamu akan menjadi backend developer")

    case "Solidity":
        print("kamu akan menjadi Blockchain developer")

    case "Java":
        print("kamu akan menjadi mobile app developer")

    case _:
        print("bahasa tidak penting, yang penting adalah
              mampu untuk menyelesaikan masalah.")
```

**Output:**

```
> kamu akan menjadi Data Scientist
```