

BAB 3

Condition (Conditional Statement)

Condition atau percabangan merupakan sebuah fitur dari bahasa pemrograman yang berfungsi untuk melakukan tindakan berbeda sesuai dengan kondisi yang ditetapkan oleh seorang programmer. Dengan percabangan, seorang programmer dapat menciptakan percabangan pada kode pemrograman mereka dan akan tereksekusi jika kondisinya terpenuhi.

Dalam pemrograman, sering kali kita harus membuat keputusan berdasarkan kondisi tertentu. Misalnya:

- Jika nilai ujian $\geq 70 \rightarrow$ siswa dinyatakan lulus.
- Jika umur $\geq 17 \rightarrow$ boleh membuat KTP.
- Jika username dan password benar \rightarrow pengguna bisa login.

Semua keputusan ini dalam Python ditangani menggunakan percabangan (condition). Percabangan memungkinkan program untuk menjalankan bagian kode tertentu jika syarat tertentu terpenuhi, dan mengabaikannya jika tidak.

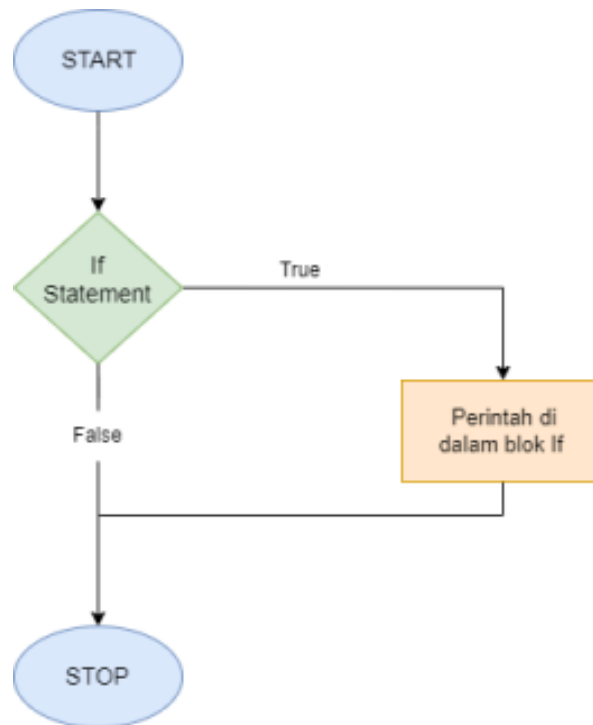
Layaknya bahasa pemrograman lain, bahasa pemrograman python menggunakan keyword if, elif, dan else statement dalam membuat sebuah alur percabangan. Hampir tidak ada bahasa pemrograman yang tidak memiliki if statement dan hampir tidak ada cara untuk memprogram tanpa cabang dalam aliran kode (setidaknya jika kode tersebut perlu memecahkan masalah yang kompleks).

A. If Statement

If statement merupakan jenis pengkondisian yang paling mendasar. Kode akan dieksekusi apabila ekspresi terpenuhi atau bernilai True (benar). Pernyataan dari if statement harus memiliki indent minimal sepanjang satu spasi di awal tiap baris kode.

```
1 if kondisi:
2     # baris kode yang dieksekusi
```

Alur Program :



Contoh Program :

```
1 angkatan = 2025
2
3 if angkatan == 2025:
4     print("Selamat Datang dan Selamat Belajar Algoritma dan Pemrograman!")
5
6
7 print("Akhir dari Program")
```

Output : Selamat Datang dan Selamat Belajar Algoritma dan Pemrograman!

- **Shorthand If Statement**

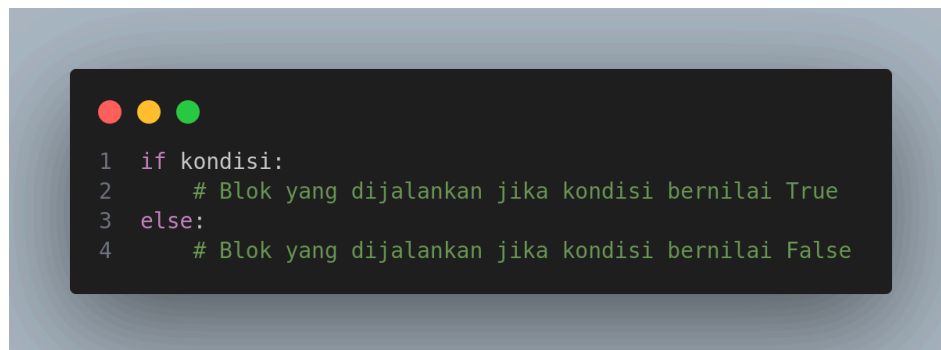
Shorthand if statement digunakan ketika hanya satu statement yang perlu dieksekusi di dalam blok if. Pernyataan ini dapat disertakan di baris yang sama dengan pernyataan If.

A screenshot of a code editor with a dark background and light blue borders. At the top left, there are three colored circles: red, yellow, and green. Below them, a single line of Python code is displayed:

```
1 if kondisi: # Blok kode yang di eksekusi
```

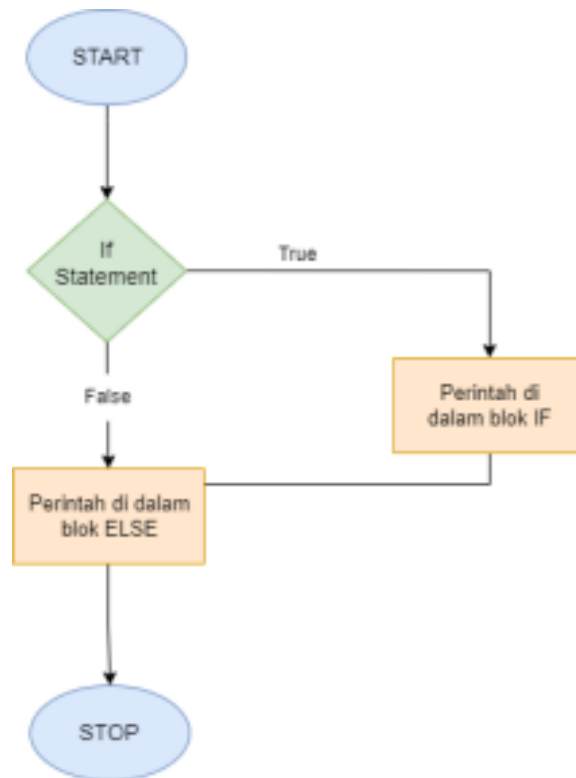
B. If-Else Statement

Pernyataan “else” digunakan ketika bagian benar dan salah dari kondisi tertentu ditentukan untuk dieksekusi. Ketika kondisinya True, maka pernyataan yang ada pada blok If akan dieksekusi. Sebaliknya, jika kondisinya bernilai False, maka pernyataan dalam blok Else yang akan dieksekusi.

A screenshot of a code editor with a dark background and light blue borders. At the top left, there are three colored circles: red, yellow, and green. Below them, a block of Python code is displayed:

```
1 if kondisi:
2     # Blok yang dijalankan jika kondisi bernilai True
3 else:
4     # Blok yang dijalankan jika kondisi bernilai False
```

Alur Program :



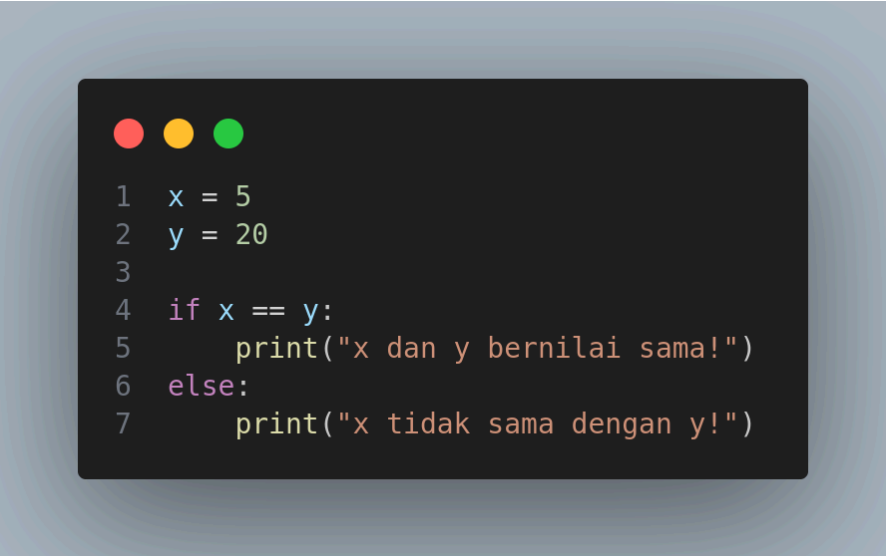
Contoh Program :

Contoh 1 :

```
1 umur = int(input("Masukkan umur anda : "))
2
3 if umur <= 18:
4     print("Anda masih dibawah umur!")
5 else:
6     print("Anda telah berusia legal!")
```

Program tersebut akan menerima sebuah input dengan tipe data integer. Apabila inputan umur yang diterima nilainya kurang dari atau sama dengan 18, maka program tersebut akan menampilkan output “Anda masih dibawah umur!”. Sedangkan jika inputan umur bernilai lebih dari 18 tahun maka program akan menampilkan output “Anda telah berusia legal!”

Contoh 2 :

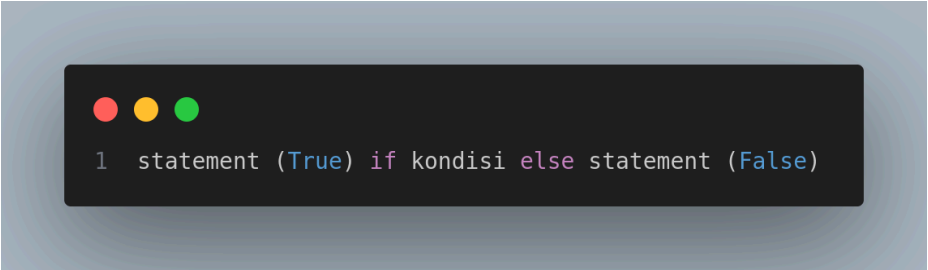
A screenshot of a terminal window with a dark background and three colored window control buttons (red, yellow, green) at the top left. The terminal displays a Python script with seven lines of code. The first two lines initialize variables x and y. The next three lines form an if-else statement that prints a message based on whether x and y are equal.

```
1 x = 5
2 y = 20
3
4 if x == y:
5     print("x dan y bernilai sama!")
6 else:
7     print("x tidak sama dengan y!")
```

Pada program tersebut, di-inisiasikan dua buah variabel x dan y yang masing-masing bernilai 5 dan 20. Pada blok if statement, apabila x dan y memiliki value yang sama, maka program tersebut akan menampilkan output yang menyatakan bahwa “x dan y bernilai sama!”. Sedangkan apabila kondisi `x == y` tidak terpenuhi (x dan y memiliki nilai yang berbeda) maka program tersebut akan menjalankan blok else yaitu “x tidak sama dengan y!”

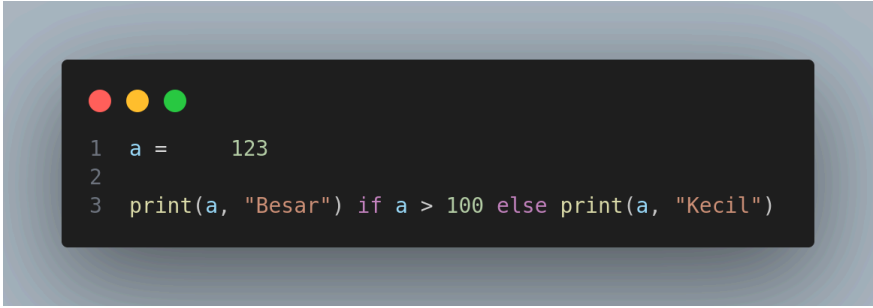
- **Shorthand If-Else Statement**

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.

A screenshot of a terminal window with a dark background and three colored window control buttons (red, yellow, green) at the top left. The terminal displays a single line of Python code using the shorthand if-else statement.

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

Contoh Program :

A screenshot of a terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The terminal displays three lines of Python code:

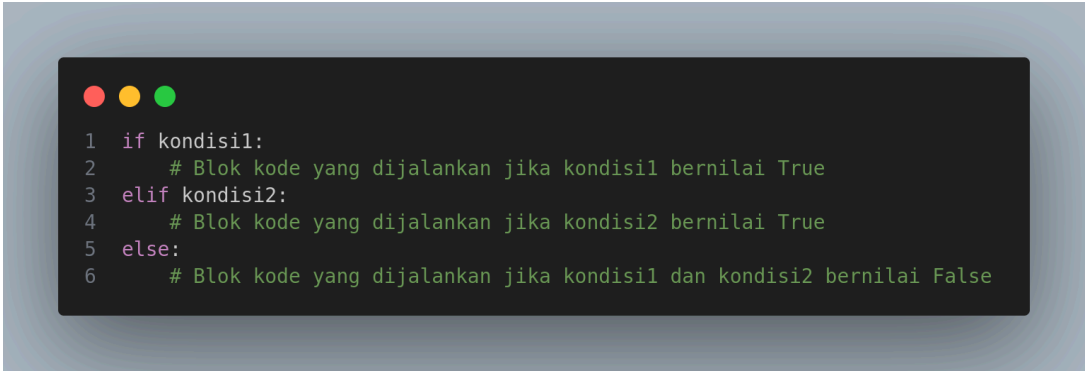
```
1 a = 123
2
3 print(a, "Besar") if a > 100 else print(a, "Kecil")
```

```
1 a = 123
2
3 print(a, "Besar") if a > 100 else print(a, "Kecil")
```

Output : 123 Besar

C. If-Elif-Else Statement

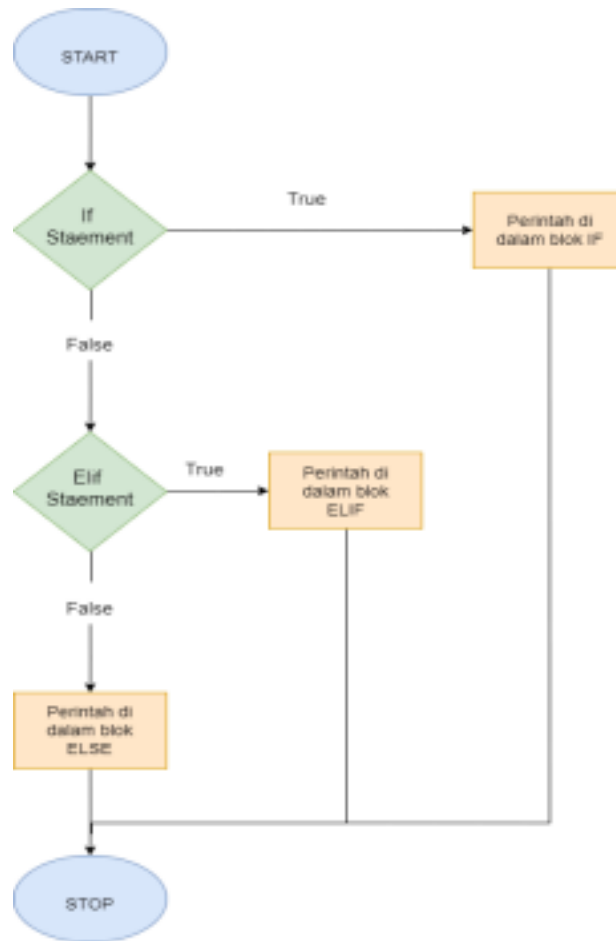
Pernyataan elif memungkinkan Anda untuk memeriksa beberapa ekspresi dan mengeksekusi blok kode segera setelah salah satu kondisi mengevaluasi ke True. Dalam hal ini, kondisi if dievaluasi terlebih dahulu. Jika salah, pernyataan elif akan dieksekusi, jika itu juga salah, pernyataan else akan dieksekusi.

A screenshot of a terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The terminal displays six lines of Python code illustrating the If-Elif-Else statement:

```
1 if kondisi1:
2     # Blok kode yang dijalankan jika kondisi1 bernilai True
3 elif kondisi2:
4     # Blok kode yang dijalankan jika kondisi2 bernilai True
5 else:
6     # Blok kode yang dijalankan jika kondisi1 dan kondisi2 bernilai False
```

```
1 if kondisi1:
2     # Blok kode yang dijalankan jika kondisi1 bernilai True
3 elif kondisi2:
4     # Blok kode yang dijalankan jika kondisi2 bernilai True
5 else:
6     # Blok kode yang dijalankan jika kondisi1 dan kondisi2 bernilai False
```

Alur Program :



Contoh Program :

```
1 angka = int(input("Masukkan angka : "))
2
3 if angka == 0:
4     print("Angka ini bernilai 0!")
5 elif angka > 0:
6     print("Bilangan Positif!")
7 else:
8     print("Bilangan Negatif!")
```

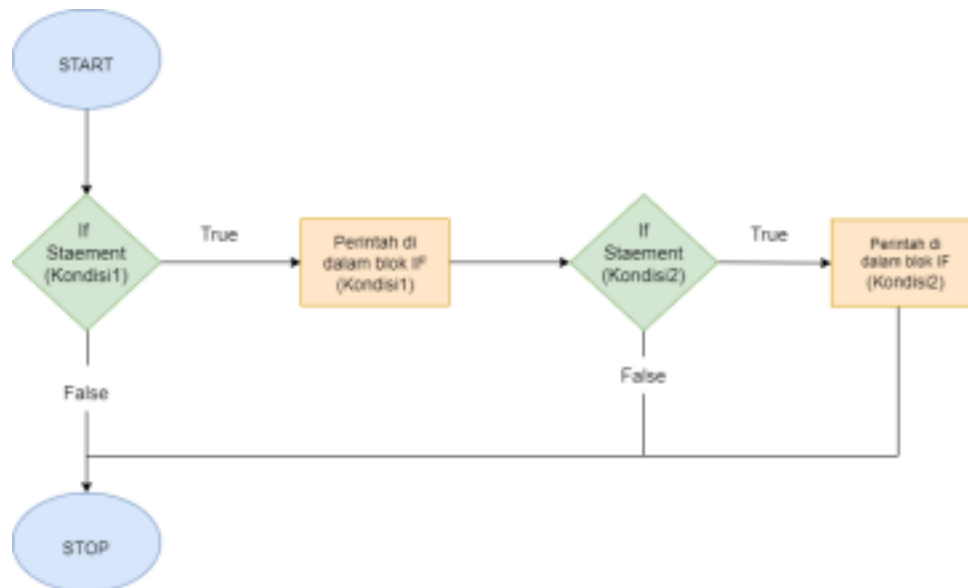
Program tersebut akan menerima inputan bertipe data integer. Apabila inputan angka yang dimasukkan bernilai sama dengan 0, maka program tersebut akan menampilkan output berupa “Angka ini bernilai 0!”. Namun, jika inputan angka lebih besar dari 0, program akan membaca angka sebagai sebuah bilangan positif. Sedangkan jika kondisi pertama dan kondisi kedua tidak terpenuhi, maka program akan langsung menampilkan output “Bilangan Negatif!”.

D. 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.

```
1  if kondisi1 :  
2      # Blok kode yang dijalankan jika kondisi1 bernilai True  
3      if kondisi2:  
4          # Blok kode yang dijalankan jika kondisi1 dan kondisi2 bernilai True
```

Alur Program :



Contoh Program :

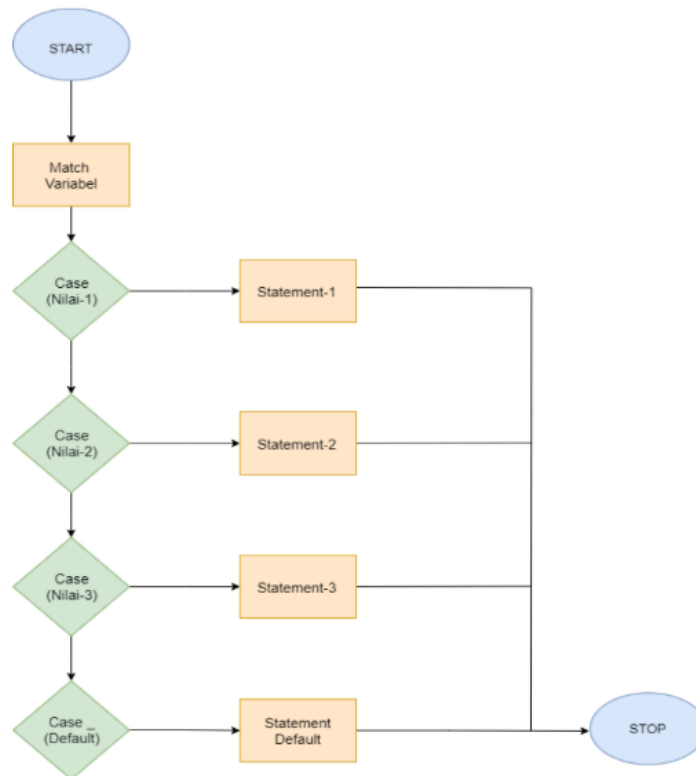
```
1 umur = int(input("Masukkan Umur Anda : "))
2
3 nilai = int(input("Masukkan Nilai Tes Anda : "))
4
5 if umur > 17:
6     if nilai < 60:
7         print("Maaf, Anda gagal mendapatkan SIM Anda!")
8     else:
9         print("Selamat, Anda berhak mendapatkan SIM Anda!")
10 else:
11     print("Maaf, Anda gagal mendapatkan SIM Anda!")
```

E. Macth 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.

```
1 match variabel:
2
3     case nilai-1:
4
5         statement-1:
6
7     case nilai-2:
8
9         statement-2:
10
11     case nilai-...:
12
13         statement-...:
14
15     case _:
16
17         statement-default (jika tidak ada case yang sesuai)
```

Alur Program :



Contoh Program :

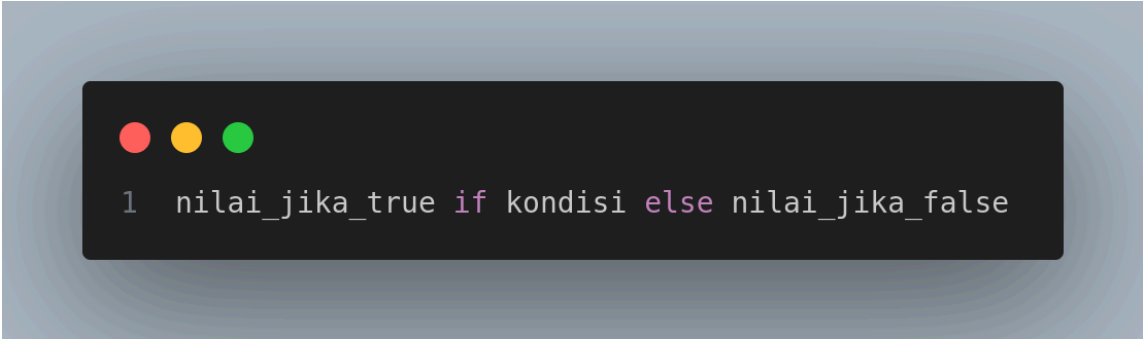
```
1  bahasa = "Python"
2
3  match bahasa:
4
5      case "Javascript":
6          print("Kamu akan menjadi Web Developer!")
7
8      case "Python":
9          print("Kamu akan menjadi Data Scientist!")
10
11     case "PHP":
12         print("Kamu akan menjadi Backend Developer!")
13
14     case "Solidity":
15         print("Kamu akan menjadi Blockchain Developer!")
16
17     case "Java":
18         print("Kamu akan menjadi Mobile App Developer!")
19
20     case _:
21         print("Bahasa tidak penting, yang penting adalah mampu untuk menyelesaikan masalah!")
```

Output : Kamu akan menjadi Data Scientist!

F. Ternary

Cara singkat untuk menulis ekspresi kondisional dalam satu baris kode. Dalam bahasa pemrograman, terutama seperti Python, ternary operator digunakan untuk melakukan operasi yang membutuhkan evaluasi kondisi dan pemilihan nilai berdasarkan hasil evaluasi tersebut.

Biasanya digunakan ketika kamu ingin menetapkan nilai pada variabel berdasarkan kondisi tertentu dalam satu baris kode. Ini memudahkan penulisan kode yang lebih singkat dan sering digunakan ketika kondisi yang dievaluasi tidak terlalu kompleks.



```
1 nilai_jika_true if kondisi else nilai_jika_false
```

Penjelasan :

- `kondisi`: Ekspresi yang akan dievaluasi (True atau False).
- `nilai_jika_true`: Nilai yang akan dikembalikan jika kondisi adalah True.
- `nilai_jika_false`: Nilai yang akan dikembalikan jika kondisi adalah False.

Contoh Program :



```
1 x = 10
2
3 hasil = "Positif" if x > 0 else "Negatif"
4
5 print(hasil)
```

Output : Positif