

# MODUL ALGORITMA DAN PEMROGRAMAN

**Tahun Ajaran 2024-2025**

**D4 Manajemen Informatika**



## BAB III

### CONDITIONAL DAN LOGICAL OPERATOR

#### Tujuan Pembelajaran :

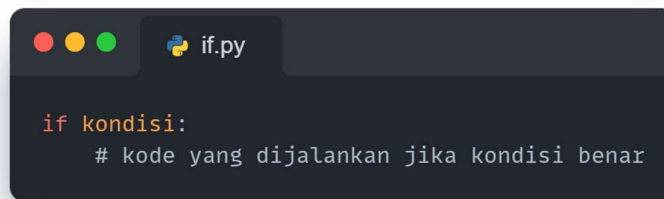
1. Memahami konsep dasar percabangan menggunakan if, elif, dan else.
2. Menggunakan operator logika seperti and, or, dan not.
3. Mampu mengimplementasikan percabangan dan operator logika dalam program sederhana.

### 3.1 Struktur Dasar Percabangan

#### 1. If Statement

Pernyataan if digunakan untuk mengevaluasi suatu kondisi. Jika kondisi tersebut bernilai True, maka blok kode di dalam pernyataan if akan dieksekusi. Jika kondisi bernilai False, program akan melanjutkan ke pernyataan berikutnya.

Contoh:

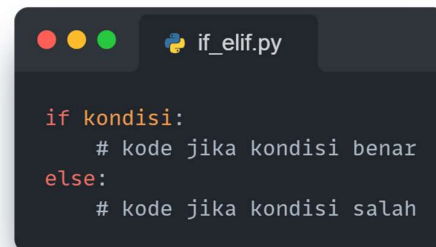


```
if kondisi:
    # kode yang dijalankan jika kondisi benar
```

#### 2. If-Else Statement

Pernyataan if-else memberikan alternatif. Jika kondisi dalam if bernilai True, maka blok kode di dalam if akan dieksekusi. Namun, jika kondisi tersebut bernilai False, maka blok kode di dalam else akan dijalankan.

Contoh:



```
if kondisi:
    # kode jika kondisi benar
else:
    # kode jika kondisi salah
```

#### 3. If-Elif-Else Statement

Pernyataan elif (singkatan dari "else if") digunakan ketika ada beberapa kondisi yang perlu dievaluasi. Program akan memeriksa setiap kondisi secara berurutan. Jika salah satu kondisi bernilai True, blok kode yang sesuai akan dieksekusi, dan program akan melewati sisa kondisi.

Contoh:

```
if_elif_else.py

if kondisi1:
    # kode jika kondisi1 benar
elif kondisi2:
    # kode jika kondisi2 benar
else:
    # kode jika semua kondisi salah
```

### 3.2 Operator Logika

Operator logika dalam pemrograman Python digunakan untuk melakukan operasi pada nilai boolean (True atau False). Operator ini sangat berguna dalam percabangan, di mana kita perlu mengevaluasi beberapa kondisi sekaligus. Ada tiga operator logika utama yang sering digunakan dalam Python: and, or, dan not. Mari kita bahas masing-masing operator ini secara mendetail.

#### 1. AND

Operator and mengembalikan True hanya jika kedua kondisi yang dievaluasi bernilai True. Jika salah satu atau kedua kondisi bernilai False, maka hasilnya adalah False.

Contoh:

```
operator_and.py

a = 10
b = 20
if a > 5 and b > 15:
    print("Kedua kondisi benar.")
```

Dalam contoh di atas, karena kedua kondisi ( $a > 5$  dan  $b > 15$ ) bernilai True, maka output yang dihasilkan adalah "Kedua kondisi benar."

#### 2. OR

Operator or mengembalikan True jika salah satu dari kondisi yang dievaluasi bernilai True. Hanya jika kedua kondisi bernilai False, maka hasilnya adalah False.

Contoh:

```
operator_or.py

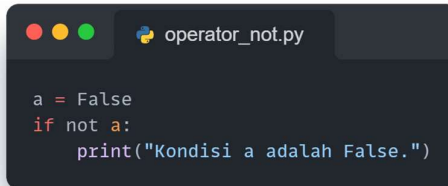
a = 10
b = 5
if a > 15 or b < 10:
    print("Salah satu kondisi benar.")
```

Dalam contoh ini, meskipun  $a > 15$  adalah False,  $b < 10$  adalah True, sehingga output yang dihasilkan adalah "Salah satu kondisi benar."

### 3. NOT

Operator not digunakan untuk membalikkan nilai boolean. Jika kondisi bernilai True, maka not akan mengembalikan False, dan sebaliknya.

Contoh:



```
operator_not.py

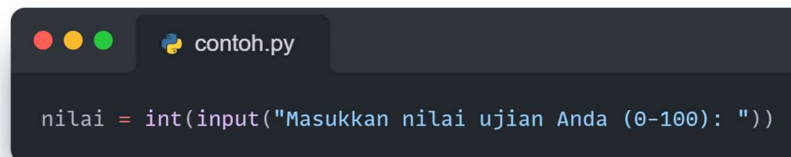
a = False
if not a:
    print("Kondisi a adalah False.")
```

Dalam contoh ini, karena a bernilai False, maka not a akan bernilai True, sehingga output yang dihasilkan adalah "Kondisi a adalah False."

#### 3.1 Latihan

Berikut adalah contoh program dalam Python yang meminta pengguna untuk memasukkan nilai ujian dan status kehadiran, kemudian memberikan kategori berdasarkan kriteria yang telah ditentukan.

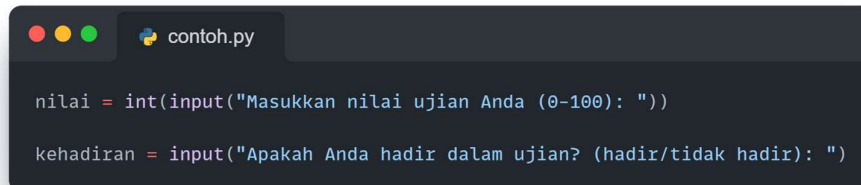
1. Program meminta pengguna untuk memasukkan nilai ujian dalam rentang 0 hingga 100.



```
contoh.py

nilai = int(input("Masukkan nilai ujian Anda (0-100): "))
```

2. Program meminta pengguna untuk memasukkan status kehadiran, apakah "hadir" atau "tidak hadir".



```
contoh.py

nilai = int(input("Masukkan nilai ujian Anda (0-100): "))

kehadiran = input("Apakah Anda hadir dalam ujian? (hadir/tidak hadir): ")
```

3. Percabangan:

- Jika nilai  $\geq 75$  dan status kehadiran adalah "hadir", maka program mencetak "Kategori: Lulus".
- Jika nilai  $< 75$  dan status kehadiran adalah "hadir", maka program mencetak "Kategori: Remedial".
- Jika status kehadiran adalah "tidak hadir", maka program mencetak "Kategori: Tidak Lulus".

```
contoh.py

nilai = int(input("Masukkan nilai ujian Anda (0-100): "))

kehadiran = input("Apakah Anda hadir dalam ujian? (hadir/tidak hadir): ")

if nilai >= 75 and kehadiran.lower() == "hadir":
    print("Kategori: Lulus")
elif nilai < 75 and kehadiran.lower() == "hadir":
    print("Kategori: Remedial")
else:
    print("Kategori: Tidak Lulus")
```

### 3.2 Tugas Praktikum

1. Buatlah program yang meminta pengguna memasukkan angka dan menentukan apakah angka tersebut genap atau ganjil.

Contoh output :

```
Masukkan sebuah angka: 13
Angka 13 adalah Ganjil.
```

2. Kembangkan program yang meminta pengguna memasukkan nilai ujian (0-100) dan memberikan feedback sebagai berikut:  
Nilai 90-100: "Sangat Baik"  
Nilai 80-89: "Baik"  
Nilai 70-79: "Cukup"  
Nilai 60-69: "Kurang"  
Nilai di bawah 60: "Sangat Kurang"

Contoh output :

```
Masukkan nilai ujian Anda (0-100): 80
Feedback: Baik
```

3. Buatlah program yang meminta pengguna memasukkan usia dan tekanan darah. Tentukan status kesehatan sebagai berikut:  
Jika usia  $\geq 60$  dan tekanan darah  $> 140$ , maka "Tinggi".  
Jika usia  $\geq 60$  dan tekanan darah  $\leq 140$ , maka "Normal".  
Jika usia  $< 60$  dan tekanan darah  $> 130$ , maka "Tinggi".  
Jika usia  $< 60$  dan tekanan darah  $\leq 130$ , maka "Normal".

Contoh output

```
Masukkan usia Anda: 40
Masukkan tekanan darah Anda: 120
Status Kesehatan: Normal
```