

<p><b>Nama:</b> Zahwa Nur Azkia Putri</p> <p><b>NIM:</b> 064002300038</p>	 <p><b>Praktikum Algoritma &amp; Pemrograman</b></p>	<p><b>MODUL 3</b></p> <p><b>Nama Dosen:</b> Ratna Shofiati, S.Kom, M.Kom</p>
<p><b>Hari/Tanggal:</b> Kamis, 05 Oktober 2023</p>		<p><b>Nama Asisten Labratorium:</b>            1. Yuda Hadi Prasetyo – 064002000004            2. Muhammad Hasan Husein – 065002100009</p>

## Struktur Kendali (Control Structure)

### 1. Teori Singkat

#### Ekspresi Boolean

Eskpresi Boolean merupakan ekspresi yang mengembalikan nilai True atau False, menggunakan operator relasional/operator perbandingan, dan juga operator logika. Selain itu Ekspresi Boolean juga dapat menggunakan operator keanggotaan (*membership operator*) dan juga operator identitas dalam beberapa kasus.

#### Operator Perbandingan

Operator Perbandingan adalah operator yang melakukan perbandingan antara dua buah nilai. Operator ini juga dikenal dengan operator relasional dan sering digunakan untuk membuat sebuah logika atau kondisi. Berikut ini adalah daftar Operator Aritmatika dalam Python:

Operator	Simbol
Lebih Besar	>
Lebih Kecil	<
Sama Dengan	==
Tidak Sama Dengan	!=
Lebih Besar Sama Dengan	>=
Lebih Kecil Sama Dengan	<=



## Operator Logika

Operator Logika merupakan sebuah operator yang digunakan untuk membuat logika dalam program yang kita buat. Operator logika juga sering disebut juga sebagai Operator Aljabar Boolean, biasanya operator logika ini digunakan untuk membuat operasi percabangan pada program. Operator Logika diantaranya seperti logika AND, OR, dan NOT.

Operator logika terdiri dari:

Operator	Simbol
Logika AND	and
Logika OR	or
Logika Negasi/Kebalikan	not

## Konstruksi Percabangan & Blok Program

Konstruksi Percabangan adalah sebuah program yang ketika dijalankan akan menimbulkan percabangan kedalam sub cabangnya yang berisi sebuah blok program sesuai dengan kondisi dan logika yang diminta. Umumnya konstruksi percabangan dalam Bahasa pemrograman Python sendiri dapat dibuat dengan memanggil keyword *if/elif/else*. Berikut tabelnya

Keterangan	Keyword
Terdapat 1 pilihan keputusan	if
Terdapat 2 pilihan keputusan	if/else
Terdapat lebih dari 2 pilihan keputusan	if/elif/else

Blok program berisi sekumpulan ekspresi dan statement untuk dikerjakan oleh komputer. Dalam Bahasa pemrograman Python blok program sendiri dapat diidentifikasi dengan tanda *colon* (":") setelah pendeklarasian konstruksi *if/elif/else*, *for*, *while* ataupun ketika melakukan definisi fungsi.

Blok program yang terdapat pada kondisi *if* sendiri akan dijalankan jika kondisi yang diminta bernilai *true*.

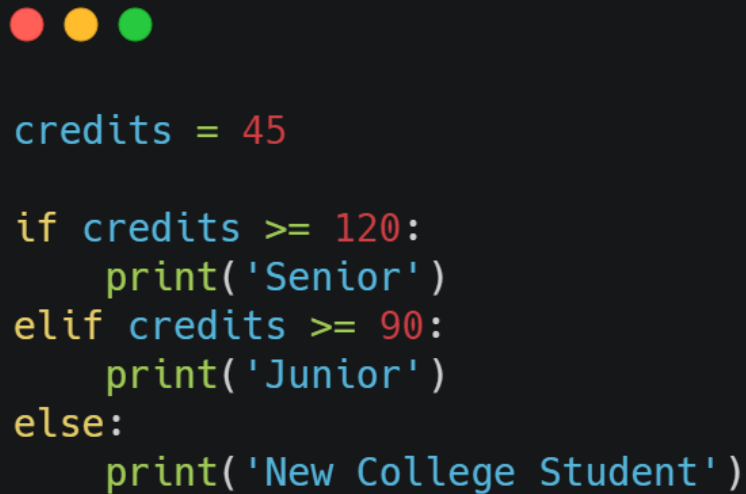
Blok program yang terdapat pada kondisi kondisi *elif* sendiri yang merupakan kepanjangan dari *else if* yang berarti jika tidak sesuai dengan kondisi sebelumnya maka akan disesuaikan dengan kondisi lainnya yang dapat bernilai *true*.

Blok program yang terdapat pada kondisi *else* akan dijalankan ketika nilai dari kondisi sebelumnya yaitu *if/elif* bernilai *false*.



Berikut ini adalah contoh sederhana program konstruksi percabangan yang menggunakan operator perbandingan:

#### Source Code



```
credits = 45

if credits >= 120:
    print('Senior')
elif credits >= 90:
    print('Junior')
else:
    print('New College Student')
```

#### Output



```
New College Student
```



## 2. Alat dan Bahan

Hardware : Laptop/PC

Software : Spyder (Anaconda Python)

## 3. Elemen Kompetensi

### a. Latihan pertama

Sebuah segitiga dibangun dari tiga garis lurus. Berdasarkan panjang dari sisi-sisinya, segitiga dapat dibedakan menjadi tiga jenis. Ada segitiga sama sisi, segitiga sama kaki, segitiga siku-siku, atau segitiga sembarang. Buatlah sebuah program yang menerima tiga bilangan yang merupakan panjang dari sisi-sisi sebuah segitiga. Berdasarkan panjang yang diberikan, program anda akan mencetak jenis segitiganya (sama sisi, sama kaki, atau sembarang). Hati-hati: Tidak semua kombinasi tiga bilangan dapat membentuk segitiga. Contoh: 1, 2, 3 tidak mungkin membentuk segitiga.

### Contoh Output

```
C:\Users\Muhammad Hasan Husei\OneDrive\Documents\Algo\Source Code\Praktikum 3>python modul3lat1.py
program ini untuk mengetahui jenis dari suatu segitiga dengan berdasarkan panjang sisi yang diberikan
Panjang sisi A = 6
Panjang sisi B = 4
Panjang sisi C = 9
ini merupakan segitiga sembarang

C:\Users\Muhammad Hasan Husei\OneDrive\Documents\Algo\Source Code\Praktikum 3>python modul3lat1.py
program ini untuk mengetahui jenis dari suatu segitiga dengan berdasarkan panjang sisi yang diberikan
Panjang sisi A = 9
Panjang sisi B = 40
Panjang sisi C = 41
ini merupakan segitiga siku siku

C:\Users\Muhammad Hasan Husei\OneDrive\Documents\Algo\Source Code\Praktikum 3>python modul3lat1.py
program ini untuk mengetahui jenis dari suatu segitiga dengan berdasarkan panjang sisi yang diberikan
Panjang sisi A = 10
Panjang sisi B = 10
Panjang sisi C = 10
ini merupakan segitiga sama sisi

C:\Users\Muhammad Hasan Husei\OneDrive\Documents\Algo\Source Code\Praktikum 3>python modul3lat1.py
program ini untuk mengetahui jenis dari suatu segitiga dengan berdasarkan panjang sisi yang diberikan
Panjang sisi A = 9
Panjang sisi B = 3
Panjang sisi C = 14
ini bukan merupakan segitiga
```



## Source Code

```
print("program ini untuk mengetahui jenis dari salah satu segitiga dengan
berdasarkan panjang sisi yang di berikan")

# Menerima input panjang sisi-sisi segitiga dari pengguna
sisi1 = float(input("Masukkan panjang sisi 1: "))
sisi2 = float(input("Masukkan panjang sisi 2: "))
sisi3 = float(input("Masukkan panjang sisi 3: "))

# Memeriksa jenis segitiga
if sisi1 == sisi2 == sisi3:
    jenis_segitiga = "segitiga sama sisi"
elif sisi1 == sisi2 or sisi2 == sisi3 or sisi1 == sisi3:
    jenis_segitiga = "segitiga sama kaki"
elif (sisi1**2 == sisi2**2 + sisi3**2) or (sisi2**2 == sisi1**2 + sisi3**2) or
(sisi3**2 == sisi1**2 + sisi2**2):
    jenis_segitiga = "segitiga siku-siku"
else:
    jenis_segitiga = "segitiga sembarang"

# Menampilkan jenis segitiga
print(f"Segitiga ini adalah {jenis_segitiga}.")
```

## Output

```
program ini untuk mengetahui jenis dari salah satu segitiga dengan berdasarkan panjang sisi yang di berikan
Masukkan panjang sisi 1: 24
Masukkan panjang sisi 2: 24
Masukkan panjang sisi 3: 24
Segitiga ini adalah segitiga sama sisi.

In [13]: runfile('C:/ZAHWA/Zahwa/pratikum2algo.py', wdir='C:/ZAHWA/Zahwa')
program ini untuk mengetahui jenis dari salah satu segitiga dengan berdasarkan panjang sisi yang di berikan
Masukkan panjang sisi 1: 12
Masukkan panjang sisi 2: 6
Masukkan panjang sisi 3: 12
Segitiga ini adalah segitiga sama kaki.

In [14]: runfile('C:/ZAHWA/Zahwa/pratikum2algo.py', wdir='C:/ZAHWA/Zahwa')
program ini untuk mengetahui jenis dari salah satu segitiga dengan berdasarkan panjang sisi yang di berikan
Masukkan panjang sisi 1: 9
Masukkan panjang sisi 2: 40
Masukkan panjang sisi 3: 41
Segitiga ini adalah segitiga siku-siku.

In [15]: runfile('C:/ZAHWA/Zahwa/pratikum2algo.py', wdir='C:/ZAHWA/Zahwa')
program ini untuk mengetahui jenis dari salah satu segitiga dengan berdasarkan panjang sisi yang di berikan
Masukkan panjang sisi 1: 11
Masukkan panjang sisi 2: 65
Masukkan panjang sisi 3: 3
Segitiga ini adalah segitiga sembarang.
```



b. Latihan Kedua

Buatlah program untuk mencari Akar Persamaan Kuadrat dan Determinan

Source Code

```
import math
# Menerima input koefisien a, b, dan c dari pengguna
a = float(input("Masukkan koefisien a: "))
b = float(input("Masukkan koefisien b: "))
c = float(input("Masukkan koefisien c: "))

# Menghitung determinan
determinan = b**2 - 4*a*c

# Menentukan jenis akar berdasarkan determinan
if determinan > 0:
    akar1 = (-b + math.sqrt(determinan)) / (2*a)
    akar2 = (-b - math.sqrt(determinan)) / (2*a)
    jenis_akar = "Akar-akar berbeda"
elif determinan == 0:
    akar1 = akar2 = -b / (2*a)
    jenis_akar = "Akar-akar sama"
else:
    real_part = -b / (2*a)
    imaginary_part = math.sqrt(-determinan) / (2*a)
    akar1 = complex(real_part, imaginary_part)
    akar2 = complex(real_part, -imaginary_part)
    jenis_akar = "Akar-akar kompleks"

# Menampilkan hasil
print(f"Determinan: {determinan}")
print(f"Jenis akar: {jenis_akar}")
print(f"Akar 1: {akar1}")
print(f"Akar 2: {akar2}")
```



#### Output

```
In [24]: runfile('C:/ZAHWA/Zahwa/untitled0.py', wdir='C:/ZAHWA/Zahwa')
Masukkan koefisien a: 12
Masukkan koefisien b: 6
Masukkan koefisien c: 4
Determinan: -156.0
Jenis akar: Akar-akar kompleks
Akar 1: (-0.25+0.5204164998665332j)
Akar 2: (-0.25-0.5204164998665332j)
```

#### 4. File Praktikum

Github Repository:

```
print("Taruh link github repository praktikum kalian disini yaa...")
```

#### 5. Soal Latihan

Soal:

1. Dalam sebuah kasus program, terdapat sebuah kondisi percabangan *if/else*. Jika program yang dijalankan pada kondisi *if* tidak sesuai dengan kondisinya, maka itu akan menghasilkan status nilai *false* pada percabangan *if* tersebut, dan program tersebut akan masuk ke kondisi *else*, apakah status yang diberikan kondisi *else* tersebut? Jelaskan dan berikan alasannya serta deskripsikan kelanjutan dari program tersebut!
2. Deskripsikan serta narasikan jalannya alur source code program yang sebelumnya telah kalian buat pada Elemen Kompetensi Latihan Kedua!



Jawaban:

1. Kondisi percabangan if dan else dalam sebuah program digunakan untuk mengatur alur eksekusi berdasarkan kebenaran atau ke-falsan (true atau false) dari suatu kondisi. Jika kondisi di dalam if evaluasi menjadi False, maka program akan melanjutkan eksekusi pada blok else. Status yang diberikan oleh kondisi else adalah "jalankan kode dalam blok else jika kondisi di dalam if bernilai False." Ini berarti bahwa jika kondisi dalam if tidak terpenuhi, maka program akan menjalankan kode dalam blok else.

```
nilai = 70
if nilai >= 80:
    print("Nilai Anda tinggi.")
else:
    print("Nilai Anda rendah.")
```

Dalam contoh di atas, jika nilai kurang dari 80 (kondisi di dalam if tidak terpenuhi), maka program akan mencetak "Nilai Anda rendah." Ini adalah bagian dari aliran program yang akan dijalankan jika kondisi if adalah False.

Jadi, dalam kelanjutan program tersebut, setelah menjalankan kode dalam blok else, program akan melanjutkan eksekusi pernyataan-pernyataan di luar blok percabangan if/else, kecuali jika ada blok percabangan lain yang mungkin ada di dalamnya.

2. Program dimulai dengan mengimpor modul math untuk digunakan dalam perhitungan matematika. Selanjutnya, program meminta pengguna untuk memasukkan tiga koefisien a, b, dan c dari persamaan kuadrat. Kita diminta untuk memasukkan nilai-nilai ini satu per satu melalui input(). Setelah mendapatkan input, program menghitung determinan persamaan kuadrat menggunakan rumus  $determinan = b^2 - 4*a*c$ . Program selanjutnya menggunakan kondisi percabangan if, elif, dan else untuk menentukan jenis akar dari persamaan kuadrat berdasarkan nilai determinan.

Terdapat tiga kemungkinan jenis akar:

- Jika determinan positif ( $determinan > 0$ ), program menghitung dua akar berbeda menggunakan rumus kuadrat.
- Jika determinan nol ( $determinan == 0$ ), program menghitung satu akar tunggal yang merupakan akar ganda.
- Jika determinan negatif ( $determinan < 0$ ), program menghitung dua akar kompleks menggunakan bilangan kompleks.

Program mencetak hasilnya ke layar sesuai dengan jenis akar yang dihasilkan. Hasil yang dicetak meliputi nilai determinan dan akar-akarnya.

Program selesai, dan eksekusi program selesai.





## 6. Kesimpulan

- Dalam pengerjaan program dengan bahasa pemrograman Python, kita harus benar-benar teliti dalam menginputkan suatu fungsi untuk menampilkan suatu keluaran pada layar dengan sesuai.
- Kita dapat mengetahui pada praktikum ini belajar bagaimana menggunakan berbagai jenis struktur kendali, seperti pernyataan pengkondisian (if, else if, else), perulangan (for, while, do-while), serta pernyataan percabangan dan perulangan bersarang. Kita juga dapat belajar cara mengidentifikasi, memahami, dan memperbaiki kesalahan dalam algoritma dan struktur kendali mereka melalui proses debugging.

## 7. Cek List (✓)

No	Elemen Kompetensi	Penyelesaian	
		Selesai	Tidak Selesai
1.	Latihan Pertama	✓	
2.	Latihan Kedua	✓	

## 8. Formulir Umpan Balik

No	Elemen Kompetensi	Waktu Pengerjaan	Kriteria
1.	Latihan Pertama	30 Menit	Baik
2.	Latihan Kedua	30 Menit	Baik

Keterangan:

- Menarik
- Baik
- Cukup
- Kurang

