


<p>Nama: Dave Ryano F.M</p> <p>NIM: 064002300039</p>	 <p>Praktikum Algoritma & Pemrograman</p>	<p>MODUL 3</p> <p>Nama Dosen: Ratna Shofiati, S.Kom, M.Kom</p>
<p>Hari/Tanggal: Senin 5 Oktober 2023</p>		<p>Nama Asisten Labratorium:</p> <ol style="list-style-type: none"> Yuda Hadi Prasetyo – 065002100004 Muhammad Hasan Husein – 065002100009

Struktur Kendali (Control Structure)

1. Teori Singkat

Ekspresi Boolean

Ekspresi 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*.

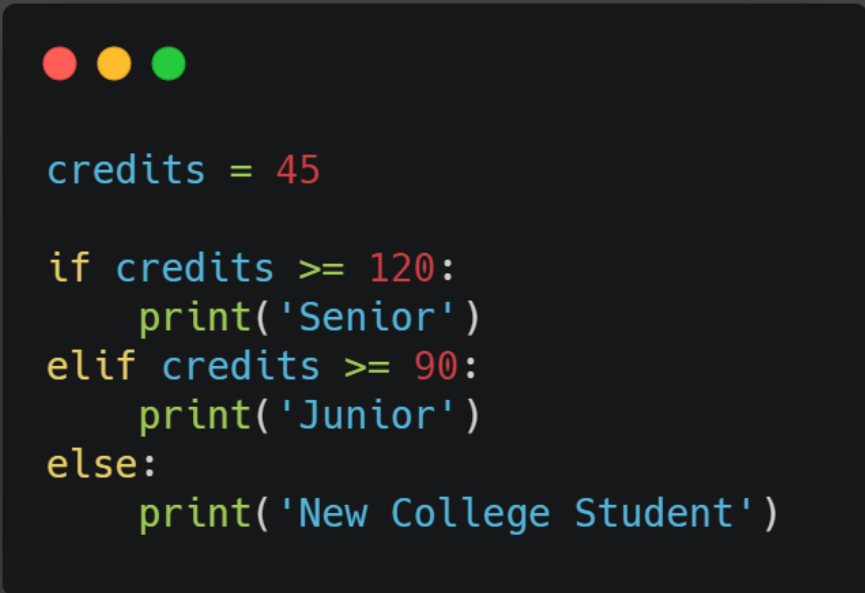


Blok program yang terdapat pada kondisi kondisi *elif* sendiri yang merupakan kepanjangan dari *else if* yang berarti jika tidak sesuai dengan kondisi sebelumnya maka akan disesuaikan dengan kondisi lainnya yang dapat bernilai *true*.

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

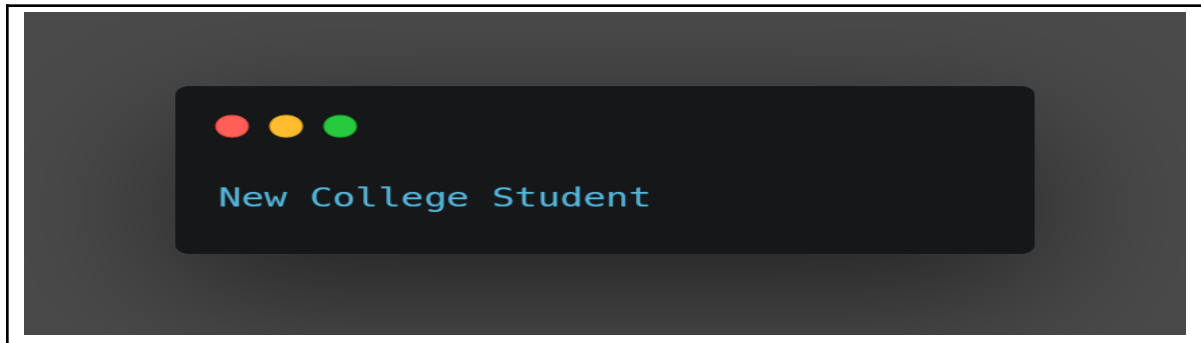


```
credits = 45

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



Output



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.

Source Code

```

1  # -*- coding: utf-8 -*-
2  """
3  Created on Thu Oct 5 19:46:24 2023
4
5  @author: Dude Herlino
6  """
7
8  sisi1 = float(input("Masukkan panjang sisi pertama: "))
9  sisi2 = float(input("Masukkan panjang sisi kedua: "))
10 sisi3 = float(input("Masukkan panjang sisi ketiga: "))
11
12 if sisi1 == sisi2 == sisi3:
13     jenis_segitiga = "segitiga sama sisi"
14 elif (sisi1 == sisi2 and sisi1 != sisi3) or (sisi1 == sisi3 and sisi1 != sisi2) or (sisi2 == sisi3 and sisi2 != sisi1):
15     jenis_segitiga = "segitiga sama kaki"
16 elif sisi1**2 == sisi2**2 + sisi3**2 or sisi2**2 == sisi1**2 + sisi3**2 or sisi3**2 == sisi1**2 + sisi2**2:
17     jenis_segitiga = "segitiga siku-siku"
18 else:
19     jenis_segitiga = "segitiga sembarang"
20
21 # Mencetak jenis segitiga
22 print("Segitiga ini adalah", jenis_segitiga)
23

```



Output

```
In [31]: runfile('C:/Users/Dude Herlino/Desktop/phyton.py', wdir='C:/Users/Dude Herlino/Desktop')
Masukkan panjang sisi pertama: 6
Masukkan panjang sisi kedua: 6
Masukkan panjang sisi ketiga: 6
Segitiga ini adalah segitiga sama sisi

In [32]: runfile('C:/Users/Dude Herlino/Desktop/phyton.py', wdir='C:/Users/Dude Herlino/Desktop')
Masukkan panjang sisi pertama: 3
Masukkan panjang sisi kedua: 6
Masukkan panjang sisi ketiga: 7
Segitiga ini adalah segitiga sembarang
```

Kuadrat dan Determinan

Source Code

```
import math

a = float(input("Masukkan koefisien a: "))
b = float(input("Masukkan koefisien b: "))
c = float(input("Masukkan koefisien c: "))

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

if determinan > 0:
    akar1 = (-b + math.sqrt(determinan)) / (2*a)
    akar2 = (-b - math.sqrt(determinan)) / (2*a)
    print(f"Akar persamaan: x1 = {akar1} dan x2 = {akar2}")
elif determinan == 0:
    akar1 = -b / (2*a)
    print(f"Akar persamaan ganda: x1 = x2 = {akar1}")
else:
    realPart = -b / (2*a)
    imaginaryPart = math.sqrt(-determinan) / (2*a)
    print(f"Akar kompleks: x1 = {realPart} + {imaginaryPart}i dan x2 = {realPart} - {imaginaryPart}i")
print(f"Determinan: {determinan}")
```



Output

```
In [4]: runfile('C:/Users/Oaan115/untitled0.py', wdir='C:/Users/Oaan115')
Masukkan koefisien a: 2
Masukkan koefisien b: 3
Masukkan koefisien c: 5
Akar kompleks: x1 = -0.75 + 1.3919410907075054i dan x2 = -0.75 - 1.3919410907075054i

In [5]: runfile('C:/Users/Oaan115/untitled0.py', wdir='C:/Users/Oaan115')
Masukkan koefisien a: 3
Masukkan koefisien b: 5
Masukkan koefisien c: 6
Akar kompleks: x1 = -0.8333333333333334 + 1.1426091000668406i dan x2 = -0.8333333333333334 - 1.1426091000668406i
Determinan: -47.0

In [6]: runfile('C:/Users/Oaan115/untitled0.py', wdir='C:/Users/Oaan115')
Masukkan koefisien a: 10
Masukkan koefisien b: 12
Masukkan koefisien c: 15
Akar kompleks: x1 = -0.6 + 1.0677078252031311i dan x2 = -0.6 - 1.0677078252031311i
Determinan: -456.0

In [7]:
```

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. Status yang diberikan kondisinya adalah nilai *true*. Alasannya karna jika kondisi *if* tidak terpenuhi atau menghasilkan *false*, maka program akan langsung melanjutkan ke blok *else*. Hal ini berarti kondisi *else* akan dieksekusi hanya jika kondisi *if* tidak terpenuhi. Kelanjutan dari program akan tergantung pada blok kode yang ada di dalam blok *else*. Biasanya, blok *else* digunakan untuk menangani kondisi alternatif atau tindakan yang



diambil jika kondisi if tidak terpenuhi. Program akan mengeksekusi perintah-perintah dalam blok else dan melanjutkan eksekusi kode setelahnya.

2. Program dimulai dengan mengimpor modul `math` untuk menggunakan fungsi `sqrt` (akar kuadrat). Lalu kita memasukkan nilai koefisien `a`, `b`, dan `c` dari persamaan kuadrat melalui fungsi `input`. Nilai akan disimpan dalam variabel `a`, `b`, dan `c`. Selanjutnya, program menghitung determinan persamaan kuadrat dengan rumus $\text{determinan} = b^2 - 4ac$. Program akan memeriksa nilai determinan: kuadrat dan mencetak hasilnya. kuadrat (akar ganda) dan mencetaknya. Terakhir, program mencetak nilai determinan yang telah dihitung.

6. Kesimpulan

- a. Dalam pengerjaan program dengan bahasa pemrograman Python, kita harus benar-benar teliti dalam menginputkan suatu fungsi untuk menampilkan suatu keluaran pada layar dengan sesuai.
- b. Kita dapat mengetahui bahwa kita dapat mengerjakan program untuk mendefinisikan macam-macam segitiga dengan cara membuat programnya lalu di run setelah itu kita tentukan angka untuk sisinya dan kita juga dapat membuat program untuk membuat Kuadrat dan Determinan.



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	Menarik
2.	Latihan Kedua	30 Menit	Menarik

Keterangan:

1. Menarik
2. Baik
3. Cukup
4. Kurang

