

<p>Nama: Athar Dhiya'an K</p> <p>NIM: 065002200012</p>	 <p>Praktikum Algoritma & Pemrograman</p>	<p>MODUL 3</p> <p>Nama Dosen: Anung B. Ariwibowo, M. Kom</p>
<p>Hari/Tanggal: Hari, Tanggal Bulan 2022</p>		<p>Nama Asisten Labratorium: 1. Azhar Rizki Zulma 065001900001</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*.

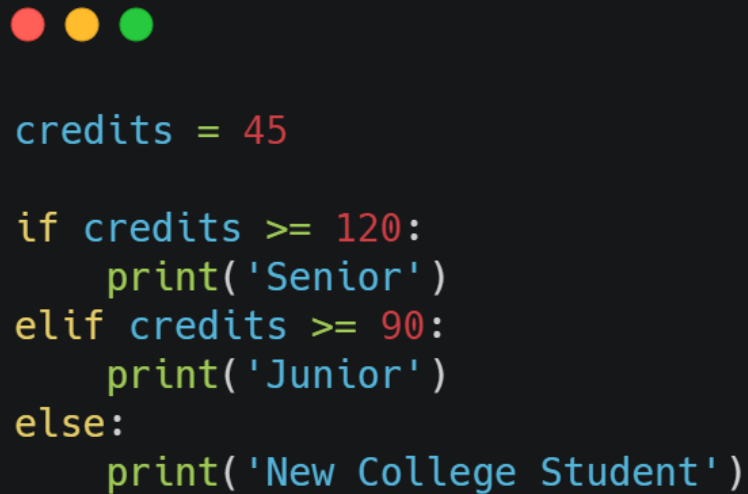
Blok program yang terdapat pada kondisi kondisi *elif* sendiri yang merupakan kepanjangan dari *else if* yang berarti jika tidak sesuai dengan kondisi sebelumnya maka akan disesuaikan dengan kondisi lainnya yang dapat bernilai *true*.

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

A screenshot of a terminal window with a dark background. At the top left, there are three colored circles: red, yellow, and green. The code is written in a light blue font. It starts with 'credits = 45', followed by an if-elif-else block. The if condition is 'credits >= 120:', the elif condition is 'credits >= 90:', and the else block is empty. The print statements are 'print('Senior')', 'print('Junior')', and 'print('New College Student')' respectively.

```
credits = 45

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

Output

A screenshot of a terminal window with a dark background. At the top left, there are three colored circles: red, yellow, and green. Below them, the text 'New College Student' is displayed in a light blue font.

```
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.

Source Code

```
sisi1 = int(input('Masukan sisi 1\n'))
sisi2 = int(input('Masukan sisi 2\n'))
alas = int(input('Masukan alas\n'))

if(sisi1 == sisi2)and(sisi2 == alas):
    print('Ini adalah jenis segitiga sama sisi')
elif(sisi1 == sisi2)and(sisi2 != alas):
    print('Ini adalah jenis segitiga sama kaki')
elif(sisi1 >= sisi2 + alas)or(sisi2 >=sisi1 + alas)or(alas >= sisi1 + sisi2):
    print('Ini bukan Segitiga!')
elif(sisi1 != sisi2)and(sisi1 != alas)and(sisi2 != alas):
    print('Ini adalah segitiga sembarang')
```



Output

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Fri Sep 30 10:59:14 2022
4
5 @author: Athardk
6 """
7
8 sisi1 = int(input("Masukan sisi 1\n"))
9 sisi2 = int(input("Masukan sisi 2\n"))
10 alas = int(input("Masukan alas\n"))
11
12 if (sisi1 == sisi2) and (sisi2 == alas):
13     print("Ini adalah jenis segitiga sama sisi")
14 elif (sisi1 == sisi2) and (sisi2 != alas):
15     print("Ini adalah jenis segitiga sama kaki")
16 elif (sisi1 >= sisi2 + alas) or (sisi2 >= sisi1 + alas) or (alas >= sisi1 + sisi2):
17     print("Ini bukan Segitiga!")
18 elif (sisi1 != sisi2) and (sisi1 != alas) and (sisi2 != alas):
19     print("Ini adalah segitiga sembarang")

```

Console Output:

```

Masukan sisi 1
10
Masukan sisi 2
2
Masukan alas
3
Ini adalah segitiga sembarang

In [6]: runcell(0, 'C:/Users/Athardk/Documents/TRISAKTI/Algoritma/untitled112.py')
Masukan sisi 1
10
Masukan sisi 2
2
Masukan alas
3
Ini bukan Segitiga!

In [7]:

```

b. Latihan Kedua

Buatlah program untuk mencari Akar Persamaan Kuadrat dan Determinan

Source Code

```

import math as m
#input
print('program pencari persamaan kuadrat dan determinant')
x1=int(input('nilai a(x2): '))
x2=int(input('nilai b(x): '))
x3=int(input('nilai c: '))
#determinan
if x1==0:
    print('bukan merupakan persamaan kuadrat')
else:
    D = pow(x2, 2)-(4*x1*x3)
    if (D>0):
        d1 = ((-x2)+m.sqrt(D))/(2*x1)
        d2 = ((-x2)-m.sqrt(D))/(2*x1)
        print('persamaan kuadrat:',x1,'x2 +',x2,'x +',x3)
        print('determinan: ',D)

```



```

print('memiliki akar x yang berbeda')
print('akar (x) pertama:',d2)
print('akar (x) kedua:',d1)
elif D==0:
    d1 = (x2)/(2*x1)
    d2 = (x2)/(2*x1)
    print('persamaan kuadrat:',x1,'x2 +',x2,'x +',x3)
    print('determinan: ',D)
    print('memiliki akar x yang sama/kembar')
    print('akar (x) pertama:',d2)
    print('akar (x) kedua:',d1)
elif D<0:
    print('persamaan kuadrat:',x1,'x2 +',x2,'x +',x3)
    print('determinan: ',D)
    print('merupakan akar imajiner')
    print('akar (x) pertama: ',x2,'+√',D,'/2',x1)
    print('akar (x) pertama: ',x2,'-√',D,'/2',x1)
else:
    print('tidak bisa di hitung')

```

Output

The screenshot shows the Spyder Python IDE interface. The left pane displays a Python script named 'untitled12.py' with the following code:

```

4
5 @author: Athardk
6 ***
7
8 import math as m
9 #input
10 print('program pencari persamaan kuadrat dan determinan')
11 x1=int(input('nilai a(x): '))
12 x2=int(input('nilai b(x): '))
13 x3=int(input('nilai c: '))
14 #determinan
15 if x1==0:
16     print('bukan merupakan persamaan kuadrat')
17 else:
18     D = pow(x2, 2)-(4*x1*x3)
19     if (D==0):
20         d1 = ((-x2)+m.sqrt(D))/(2*x1)
21         d2 = ((-x2)-m.sqrt(D))/(2*x1)
22         print('persamaan kuadrat:',x1,'x2 +',x2,'x +',x3)
23         print('determinan: ',D)
24         print('memiliki akar x yang berbeda')
25         print('akar (x) pertama:',d2)
26         print('akar (x) kedua:',d1)
27     elif D==0:
28         d1 = (x2)/(2*x1)
29         d2 = (x2)/(2*x1)
30         print('persamaan kuadrat:',x1,'x2 +',x2,'x +',x3)
31         print('determinan: ',D)
32         print('memiliki akar x yang sama/kembar')
33         print('akar (x) pertama:',d2)
34         print('akar (x) kedua:',d1)
35     elif D<0:
36         print('persamaan kuadrat:',x1,'x2 +',x2,'x +',x3)
37         print('determinan: ',D)
38         print('merupakan akar imajiner')
39         print('akar (x) pertama: ',x2,'+√',D,'/2',x1)
40         print('akar (x) pertama: ',x2,'-√',D,'/2',x1)
41     else:
42         print('tidak bisa di hitung')

```

The right pane shows the console output for the execution of the script. The output is as follows:

```

In [2]: runcell(0, 'C:/Users/Athardk/Documents/TRISAKTI/Algoritma/untitled12.py')
program pencari persamaan kuadrat dan determinan
nilai a(x): 16
nilai b(x): 10
nilai c: 8
persamaan kuadrat: 16 x2 + 10 x + 8
determinan: -432
merupakan akar imajiner
akar (x) pertama: 10 +√ -432 /2 16
akar (x) pertama: 10 -√ -432 /2 16

```

4. File Praktikum

Github Repository:

<https://github.com/Athardk/Praktikum-Algoritma>



Jurusan Teknik Informatika & Sistem Informasi
Fakultas Teknologi Industri – Universitas Trisakti

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. Jika kondisi *if* adalah *true*, sedangkan *else* bernilai *false*, maka perintah yang akan di eksekusi adalah *else*. Alasannya adalah perintah tersebut bernilai *false* sehingga yang dieksekusi adalah *else*. Indentasi digunakan untuk memisahkan blok
2. Masukkan nilai input ke dalam 3 variabel Jika variabel *x1* determinan nya adalah 0 maka akan muncul *print('bukan merupakan kuadrat')*

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 banyak operasi matematika yang bis akita selesaikan dengan menggunakan python ini diantaranya hal hal yang telah kita coba selama beberapa pertemuan praktikum dimulai dari

7. Cek List (✓)

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

8. Formulir Umpan Balik

No	Elemen Kompetensi	Waktu Pengerjaan	Kriteria
1.	Latihan Pertama	45 Menit	cukup



2.	Latihan Kedua	30 Menit	cukup
----	---------------	----------	-------

Keterangan:

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

