


<p><b>Nama:</b> Nino Sebastiano Emerline Napitupulu</p> <p><b>NIM:</b> 064102400025</p>	 <p><b>Praktikum Algoritma &amp; Pemrograman</b></p>	<p><b>MODUL 3</b></p> <p><b>Nama Dosen:</b> Binti solihah, S.T, M.KOM</p>
<p><b>Hari/Tanggal:</b> Jumat, 4 Oktober 2022</p>		<p><b>Nama Asisten Labratorium:</b></p> <ol style="list-style-type: none"> <li>1. Yustianas Rombon - 064002300015</li> <li>2. Vira Aditya Kurniawan - 065002300012</li> </ol>

## 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	$\geq$
Lebih Kecil Sama Dengan	$\leq$

## 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*.

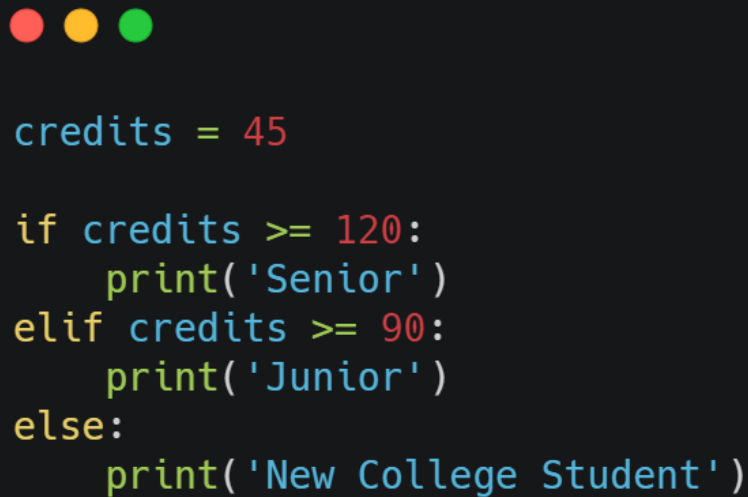


Blok program yang terdapat pada kondisi kondisi *elif* sendiri yang merupakan kepanjangan dari *else if* yang berarti jika tidak sesuai dengan kondisi sebelumnya maka akan disesuaikan dengan kondisi lainnya yang dapat bernilai *true*.

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

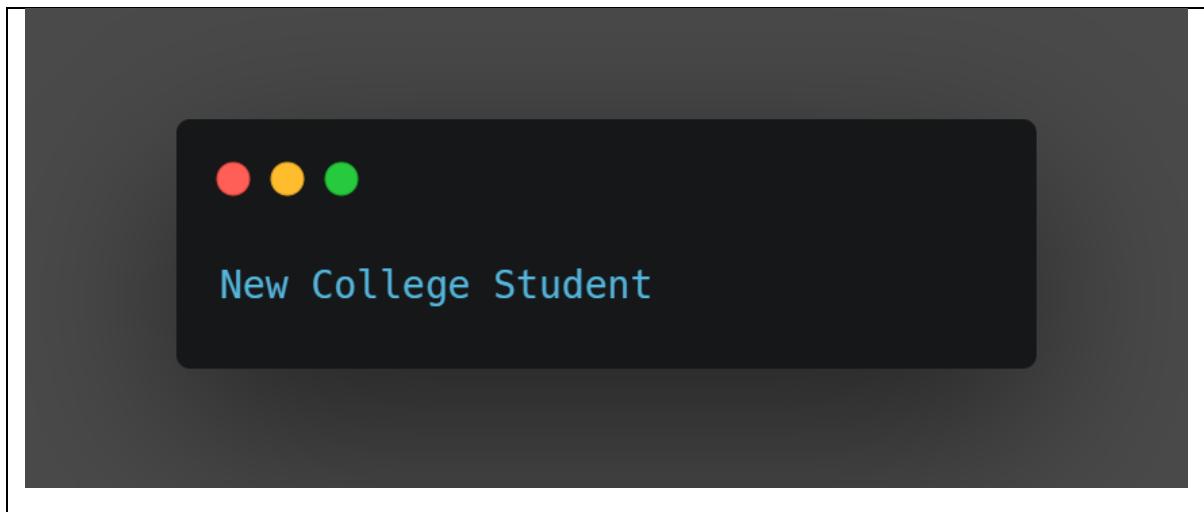


```
credits = 45

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

#### Output



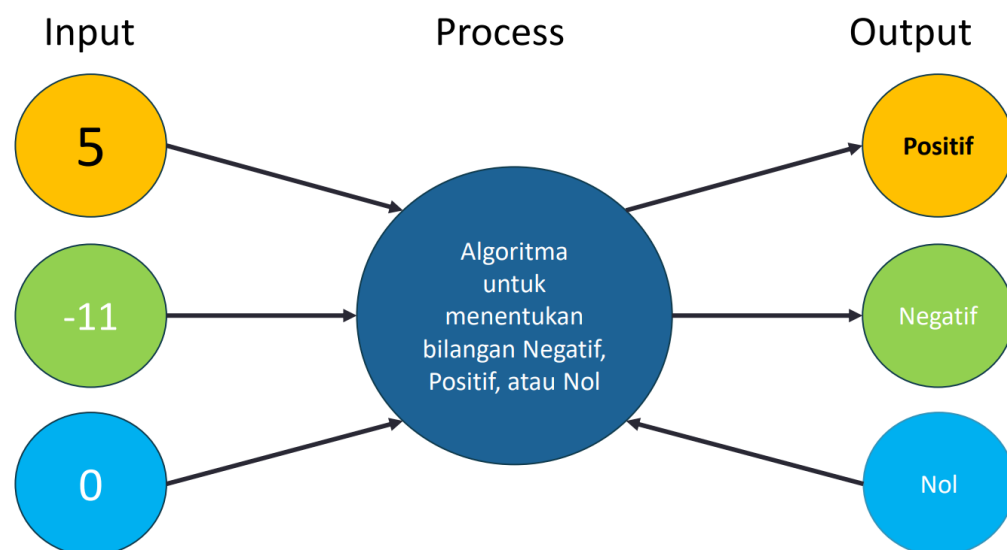


## IPO (Input Process Output)

Konsep Dasar Input, Process, dan Output (IPO)

- Konsep input, process, dan output adalah prinsip dasar dalam pemrograman dan pengembangan algoritma.
- Setiap algoritma melibatkan tiga tahap utama: mengambil data masukan (input), melakukan operasi atau pengolahan data (process), dan menghasilkan hasil akhir (output).
- Konsep ini menggambarkan bagaimana algoritma beroperasi untuk memproses informasi.

## Gambaran IPO (Menentukan Bilangan)



## Pseudocode

Pseudocode adalah suatu bentuk deskripsi informal yang mirip dengan bahasa manusia dan digunakan untuk menggambarkan algoritma atau proses secara naratif. Ini tidak terikat pada bahasa pemrograman tertentu, tetapi memberikan panduan tentang langkah-langkah yang harus diambil dalam suatu algoritma dengan bahasa yang lebih mudah dimengerti.

### Contoh PseudoCode

#### Inisiasi Variabel:

```
N      = 0
total  = 0.0
```

#### Pengulangan:

```
UNTUK i DARI 1 SAMPAI 10 LANGKAH 2
  CETAK i
END UNTUK
```

#### Pengkondisional (Conditional):

```
JIKA nilai > 10
  CETAK "Nilai lebih dari 10"
SELAINNYA JIKA nilai = 10
  CETAK "Nilai sama dengan 10"
SELAINNYA
  CETAK "Nilai kurang dari 10"
AKHIR JIKA
```

#### Fungsi atau Prosedur:

```
FUNGSI tambah(a, b)
  KEMBALIKAN a + b
AKHIR FUNGSI
```

#### Contoh Lengkap:

```
DEKLARASI variabel n, bilangan, total, rata_rata FLOAT
MINTA "Masukkan jumlah bilangan: " SIMPAN
total = 0.0

UNTUK i DARI 1 SAMPAI n
  MINTA "Masukkan bilangan ke-" + i + ": " SIMPAN bilangan
  total = total + bilangan
END UNTUK

rata_rata = total / n
CETAK "Rata-rata adalah: " + rata_rata
```

## 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.

#### Pseudocode

1. Menaruh format math
2. Buat input bilangan untuk menentukan besar bilangan a
3. Buat input bilangan untuk menentukan besar bilangan b
4. Buat input bilangan untuk menentukan besar bilangan c
5. Masukan format perhitungannya bilangan a dan b dan c
6. Cetak hasil format perhitungan dari bilangan a dan b dan c

#### Input Process Output

Input:

```
a=int(float(input("masukan bilangan:")))
```

```
b=int(float(input("masukan bilangan:")))
```

```
c=int(float(input("masukan bilangan:")))
```

proses:

```
segitiga_sama_sisi = a==b==c
```

```
segitiga_sama_kaki = b==c and a==c and a==b
```

```
segitiga_siku_siku = a**2+b**2==c**2 or a**2+c**2==b**2 or b**2+c**2==a**2
```

print:

```
#segitiga_sama_sisi
```

```
if (a==b==c):
```



```
print("Segi tiga sama sisi")
#segitiga_sama_kaki
elif (b==c or a==c or a==b):
    print("Segi tiga sama kaki")
#segitiga_siku_siku
elif (a**2+b**2==c**2 or a**2+c**2==b**2 or b**2+c**2==a**2):
    print("Segi tiga sama siku siku")
else: "Bukan Segitiga1"
```

### Source Code



```
a=int(float(input("masukan bilangan:")))
b=int(float(input("masukan bilangan:")))
c=int(float(input("masukan bilangan:")))

segitiga_sama_sisi = a==b==c
segitiga_sama_kaki = b==c and a==c and a==b
segitiga_siku_siku = a**2+b**2==c**2 or a**2+c**2==b**2 or b**2+c**2==a**2

#segitiga sama sisi
if (a==b==c):
    print("Segi tiga sama sisi")
#segitiga sama kaki
elif (b==c or a==c or a==b):
    print("Segi tiga sama kaki")
#segitiga siku siku
elif (a**2+b**2==c**2 or a**2+c**2==b**2 or b**2+c**2==a**2):
    print("Segi tiga sama siku siku")
else: "Bukan Segitiga"
```

## Output





```
masukan bilangan:12
masukan bilangan:45
masukan bilangan:45
Segi tiga sama kaki
```

b. Latihan Kedua

Buatlah program untuk mencari Akar Persamaan Kuadrat dan Determinan

Pseudocode

1. Menaruh format math
2. Buat input bilangan untuk menentukan besar bilangan a
3. Buat input bilangan untuk menentukan besar bilangan b
4. Buat input bilangan untuk menentukan besar bilangan c
5. Buat input bilangan untuk menentukan besar bilangan d
6. Masukan format perhitungannya bilangan a dan b dan c
7. Cetak hasil format perhitungan dari bilangan a dan b dan c dan d



### Input Output Process

```
Input:
def main():
    print("Masukkan nilai a:")
    a = float(input())
    print("Masukkan nilai b:")
    b = float(input())
    print("Masukkan nilai c:")
    c = float(input())

    D = hitung_determinan(a, b, c)
    print(f"Determinan D: {D}")

proses:
import math

def hitung_determinan(a, b, c):
    return b**2 - 4*a*c

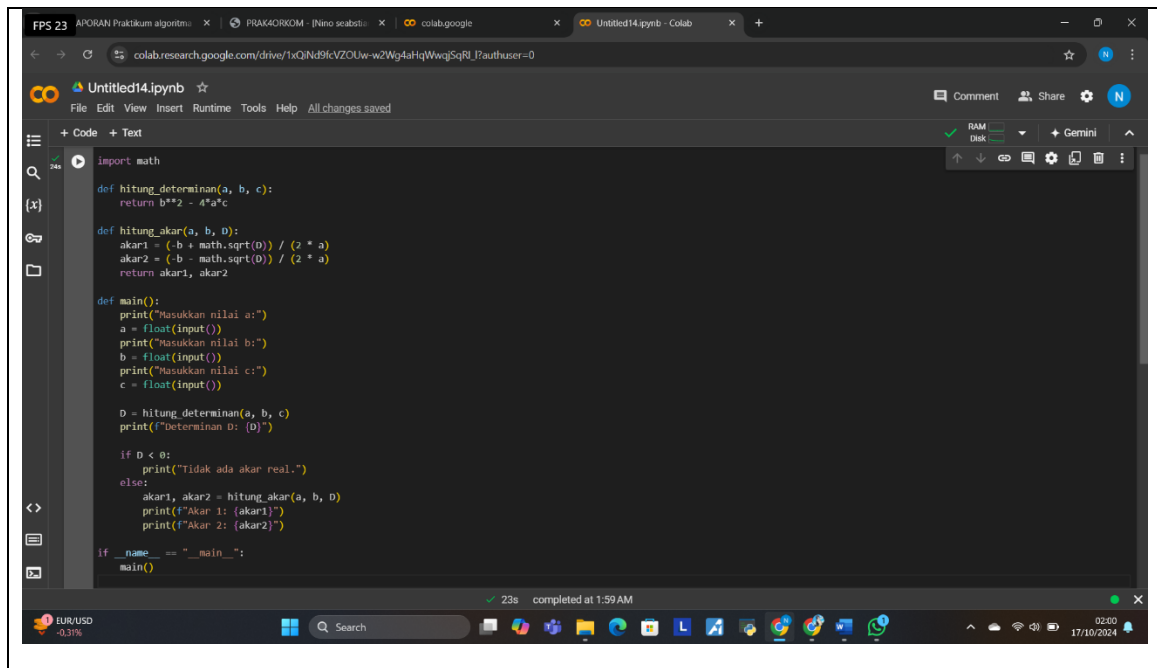
def hitung_akar(a, b, D):
    akar1 = (-b + math.sqrt(D)) / (2 * a)
    akar2 = (-b - math.sqrt(D)) / (2 * a)
    return akar1, akar2

output:
if D < 0:
    print("Tidak ada akar real.")
else:
    akar1, akar2 = hitung_akar(a, b, D)
    print(f"Akar 1: {akar1}")
    print(f"Akar 2: {akar2}")

if __name__ == "__main__":
    main()
```

### Source Code





```
import math

def hitung_determinan(a, b, c):
    return b**2 - 4*a*c

def hitung_akar(a, b, D):
    akar1 = (-b + math.sqrt(D)) / (2 * a)
    akar2 = (-b - math.sqrt(D)) / (2 * a)
    return akar1, akar2

def main():
    print("Masukkan nilai a:")
    a = float(input())
    print("Masukkan nilai b:")
    b = float(input())
    print("Masukkan nilai c:")
    c = float(input())

    D = hitung_determinan(a, b, c)
    print(f"Determinan D: {D}")

    if D < 0:
        print("Tidak ada akar real.")
    else:
        akar1, akar2 = hitung_akar(a, b, D)
        print(f"Akar 1: {akar1}")
        print(f"Akar 2: {akar2}")

if __name__ == "__main__":
    main()
```

23s completed at 1:59 AM

## Output



```
➔ Masukkan nilai a:  
45  
Masukkan nilai b:  
56  
12  
Masukkan nilai c:  
Determinan D: 976.0  
Akar 1: -0.2751000144041487  
Akar 2: -0.9693444300402957
```

#### 4. File Praktikum

Github Repository:

<https://github.com/NinoGanteng123/Ganteng>

#### 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 *if* menyatakan *false*, kode yang terdapat di dalam pernyataan *else* akan menjalankan blok, Status yang diberikan *else* adalah *true* untuk blok *else* karna merupakan bagian yang dijalankan Ketika kondisi *if* tidak terpenuhi.

Kondisi *if* akan memeriksa pernyataannya *true* atau tidak, jika *true* maka *if* dieksekusi sedangkan kondisi tidak *true*, *else* yang akan dieksekusi.

2. Pertama kita menaruh format matematikanya yaitu untuk mencari akar dan determinan, Setelah itu kita masukan kode untuk menentukan angka yang kita mau,



setelah itu program akan mengeksekusi hasil input dengan menggunakan format matematika yang kita buat tadi.

## 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 pentingnya pemahaman fungsi cara kerja yang digunakan karna itu adalah kunci untuk mendapatkan hasil yang diinginkan selain itu, kita juga harus seing melakukan pengujian pada program dengan berbagai input yang akan membantu memastikan bahwa program berfungsi dengan baik dalam berbagai kondisi.

## 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	10Menit	Menarik
2.	Latihan Kedua	10 Menit	Menarik

Keterangan:

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

