

## MODUL 4

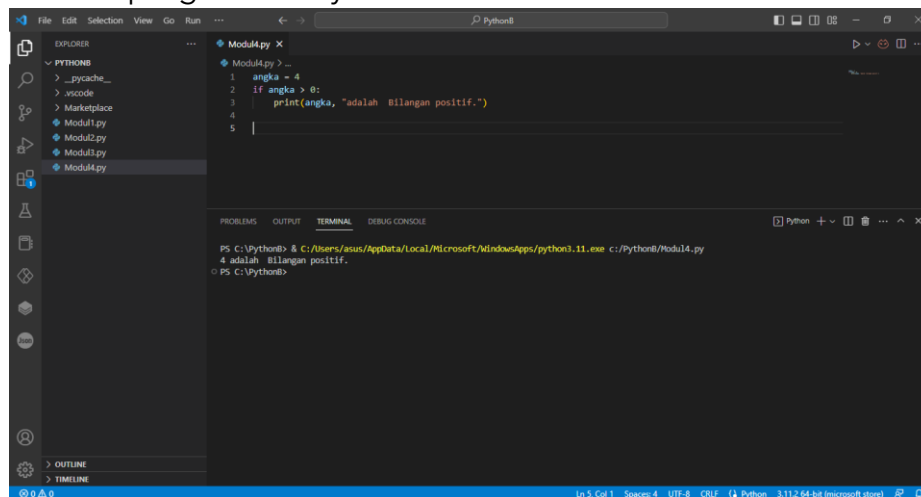
### A. Percabangan

Percabangan yaitu cara yang digunakan untuk mengambil keputusan apabila di dalam program terdapat kondisi tertentu. Jumlah kondisinya bisa satu, dua atau lebih terdapat 3 jenis pernyataan yang digunakan untuk percabangan, yaitu if, if else dan if elif else

#### 1. Pernyataan if

Pernyataan if menjalankan satu buah kondisi bila hasilnya benar maka pernyataan di dalam blok if tersebut dieksekusi jika salah, maka pernyataan tidak dieksekusi.

Contoh program Pernyataan if :

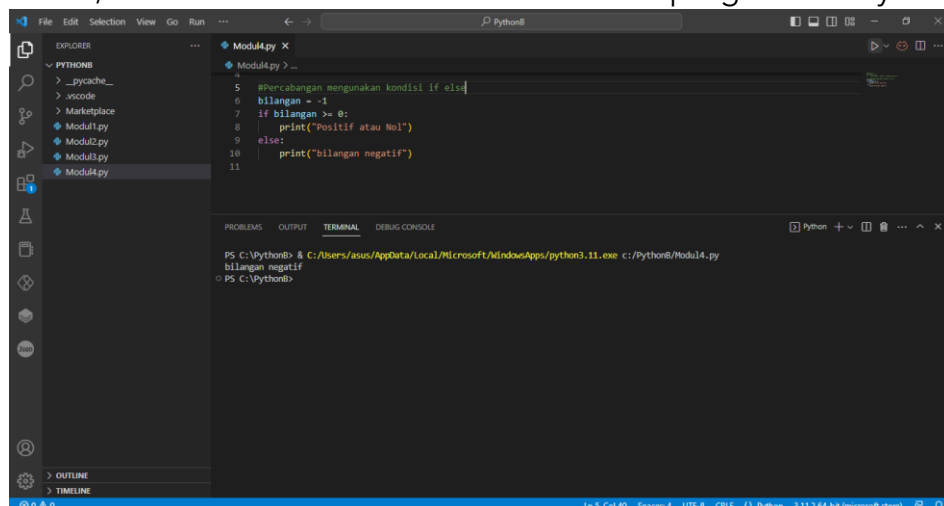


```
Modul4.py X
1  angka = 4
2  if angka > 0:
3      print(angka, "adalah Bilangan positif.")
4
5

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Python
PS C:\Python8> & C:\Users\asus\AppData\Local\Microsoft\WindowsApps/python3.11.exe c:/Python8/Modul4.py
4 adalah Bilangan positif.
PS C:\Python8>
```

#### 2. Pernyataan if else

Pernyataan if else menjalankan dua kondisi. Kondisi pertama kalau benar, dan kondisi kedua kalau salah. Contoh program Pernyataan if else:

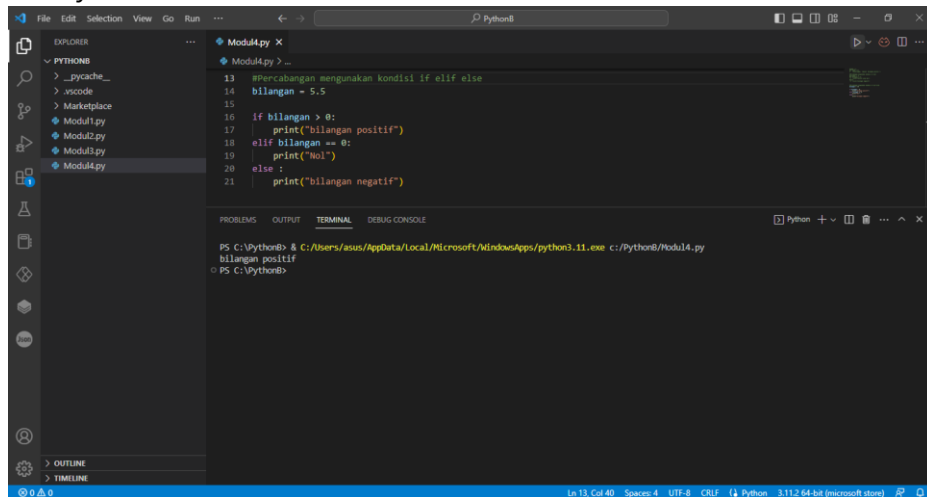


```
Modul4.py X
5  #Percabangan menggunakan kondisi if else
6  bilangan = -1
7  if bilangan >= 0:
8      print("Positif atau Nol")
9  else:
10     print("bilangan negatif")
11

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Python
PS C:\Python8> & C:\Users\asus\AppData\Local\Microsoft\WindowsApps/python3.11.exe c:/Python8/Modul4.py
bilangan negatif
PS C:\Python8>
```

### 3. Pernyataan if elif else

Pernyataan if elif else digunakan untuk menjalankan lebih dari dua kondisi. Bila kondisi pada if benar, maka pernyataan di dalamnya yang dieksekusi. Bila salah, maka masuk ke kondisi elif. Dan bila tidak ada if atau elif yang benar, maka yang dijalankan adalah yang else. Contoh Pernyataan if elif else :



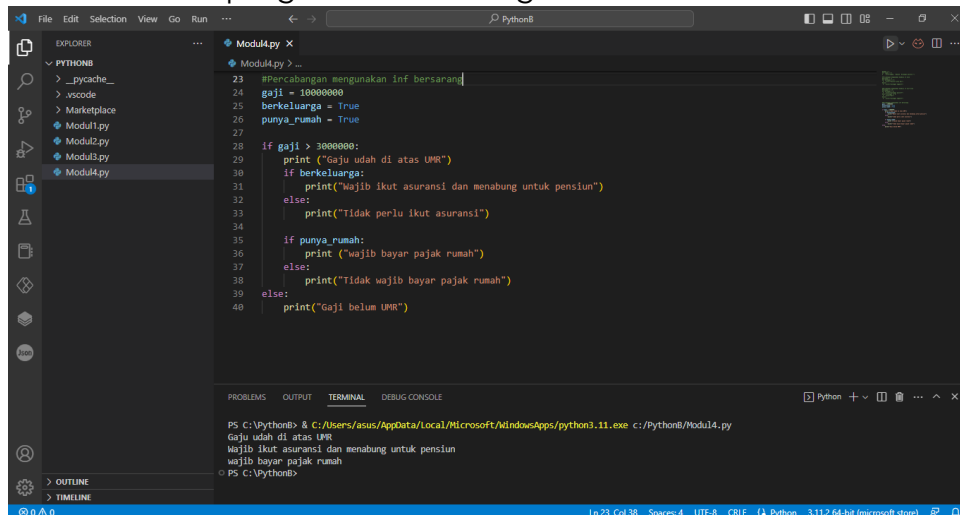
```
13 #percabangan menggunakan kondisi if elif else
14 bilangan = 5.5
15
16 if bilangan > 0:
17     print("bilangan positif")
18 elif bilangan == 0:
19     print("Nol")
20 else:
21     print("bilangan negatif")
```

Terminal Output:

```
PS C:\Python8> & C:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Python8/Modul4.py
bilangan positif
PS C:\Python8>
```

### 4. Tambahan : if Bersarang

Sebuah kondisional dapat disimpan di dalam if lain. Berikut ini adalah contoh program if bersarang :



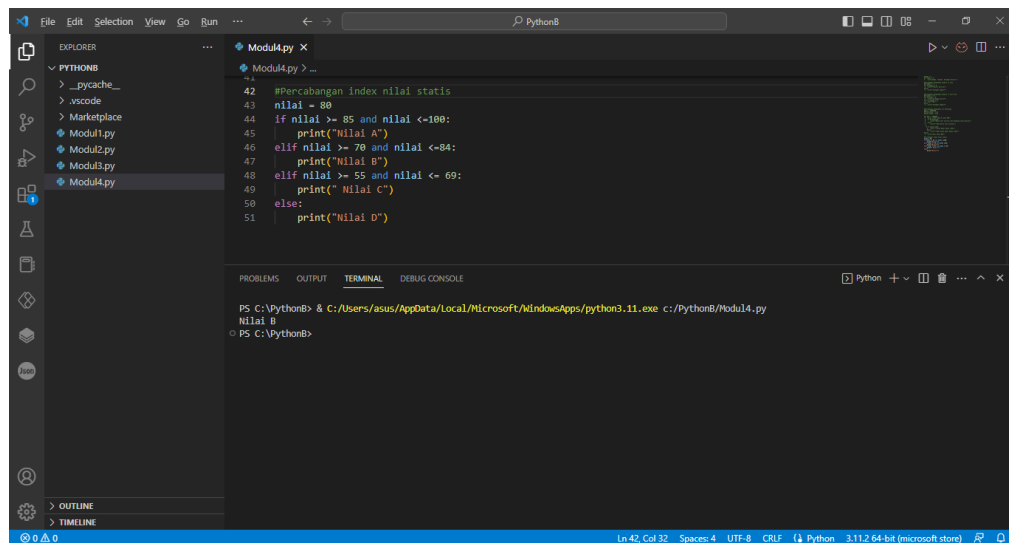
```
23 #Percabangan menggunakan inf bersarang
24 gaji = 10000000
25 berkeluarga = True
26 punya_rumah = True
27
28 if gaji > 2000000:
29     print("Gaji udah di atas UMR")
30     if berkeluarga:
31         print("wajib ikut asuransi dan menabung untuk pensiun")
32     else:
33         print("Tidak perlu ikut asuransi")
34
35     if punya_rumah:
36         print("wajib bayar pajak rumah")
37     else:
38         print("Tidak wajib bayar pajak rumah")
39 else:
40     print("Gaji belum UMR")
```

Terminal Output:

```
PS C:\Python8> & C:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Python8/Modul4.py
Gaji udah di atas UMR
wajib ikut asuransi dan menabung untuk pensiun
wajib bayar pajak rumah
PS C:\Python8>
```

## 5. Program Percabangan Indeks Nilai Statis

Ketentuan : Nilai 85 s/d 100 indeks A, nilai 70 s/d 84 indeks B, nilai 55 s/d 69 indeks C, nilai dibawah 55 indeks D. Contoh Program Percabangan Indeks Nilai Statis :



```
42 #Percabangan index nilai statis
43 nilai = 80
44 if nilai >= 85 and nilai <= 100:
45     print("Nilai A")
46 elif nilai >= 70 and nilai <= 84:
47     print("Nilai B")
48 elif nilai >= 55 and nilai <= 69:
49     print("Nilai C")
50 else:
51     print("Nilai D")
```

Terminal output:

```
PS C:\PythonB> & C:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/PythonB/Modul4.py
Nilai B
PS C:\PythonB>
```

## B. Perulangan

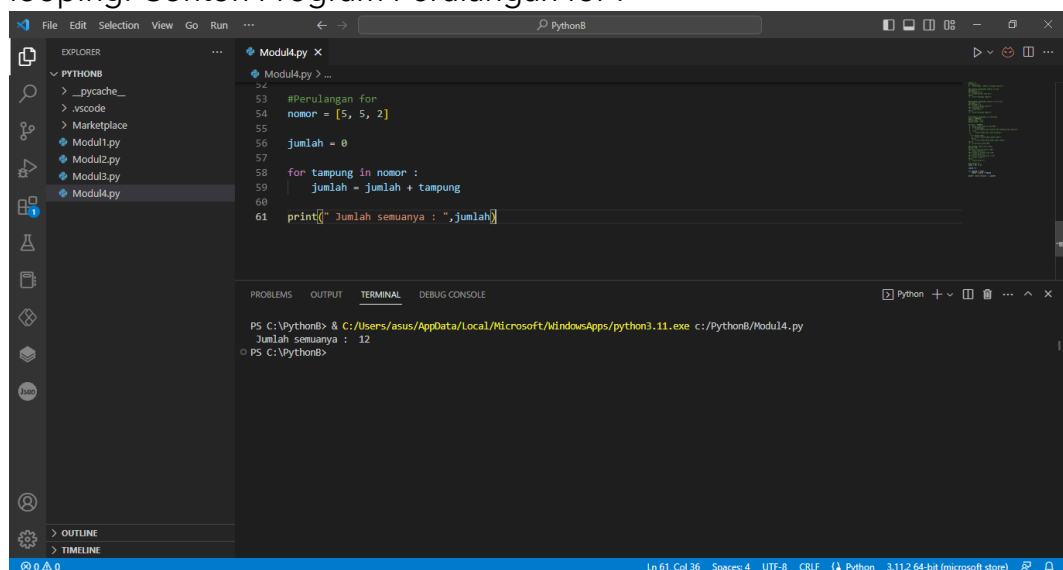
Secara umum, python mengeksekusi program baris perbaris. Mulai dari baris satu, dua, dan seterusnya. perulangan bisa dilakukan dengan dua cara atau metode, yaitu menggunakan for dan menggunakan while.

### 1. Perulangan dengan Menggunakan for

Perulangan dengan menggunakan for memiliki sintaks seperti berikut :

for var in sequence: body of

for var adalah variabel digunakan untuk menampung sementara nilai dari sequence pada saat terjadi perulangan. Sequence adalah tipe data berurut seperti string, list, dan tuple. Perulangan terjadi sampai looping mencapai elemen atau anggota terakhir dari sequence. Bila loop sudah sampai ke elemen terakhir dari sequence, maka program akan keluar dari looping. Contoh Program Perulangan for :



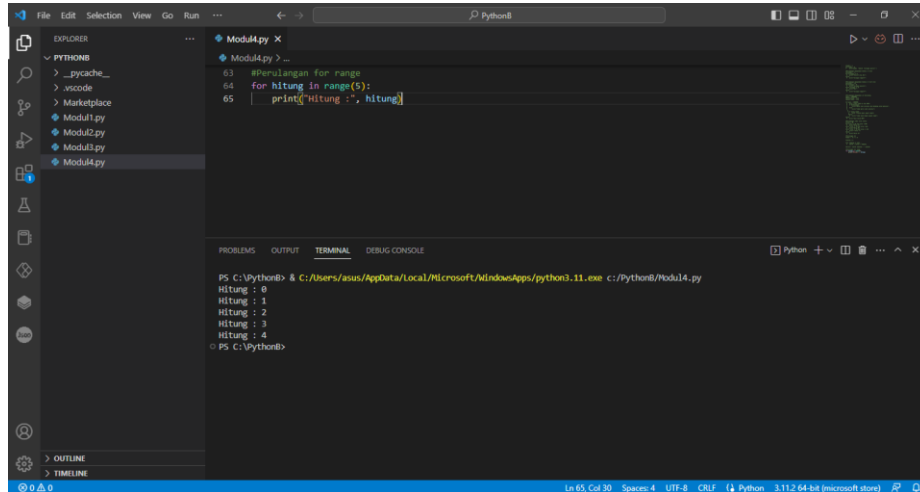
```
53 #Perulangan for
54 nomor = [5, 5, 2]
55
56 jumlah = 0
57
58 for tampung in nomor :
59     jumlah = jumlah + tampung
60
61 print(f"Jumlah semuanya : ", jumlah)
```

Terminal output:

```
PS C:\PythonB> & C:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/PythonB/Modul4.py
Jumlah semuanya : 12
PS C:\PythonB>
```

## 2. Perulangan for dengan range

Fungsi `range()` digunakan untuk menghasilkan deret bilangan. `range(10)` akan menghasilkan bilangan dari 0 sampai dengan 9 (10 bilangan). Contoh Program Perulangan for dengan Range :



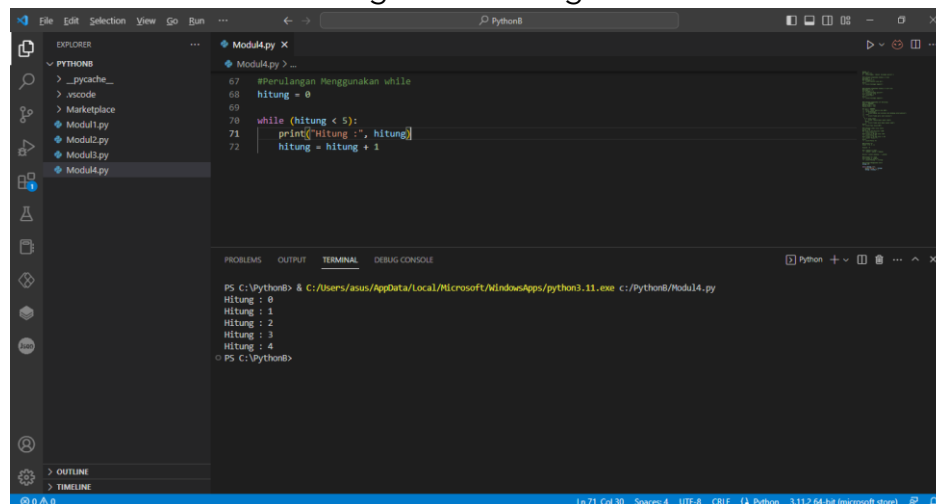
```
63 #Perulangan for range
64 for hitung in range(5):
65     print("Hitung :", hitung)
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
PS C:\Python8> & C:\Users\asus\AppData\Local\Microsoft\WindowsApps/python3.11.exe c:/Python8/Modul4.py
Hitung : 0
Hitung : 1
Hitung : 2
Hitung : 3
Hitung : 4
PS C:\Python8>
```

## 3. Perulangan Menggunakan while

Perulangan menggunakan `while` akan menjalankan blok pernyataan terus menerus selama kondisi bernilai benar. Bila kondisi salah, maka program akan keluar dari `while` dan lanjut ke baris pernyataan di luar `while`. Contoh Program Perulangan `while` :



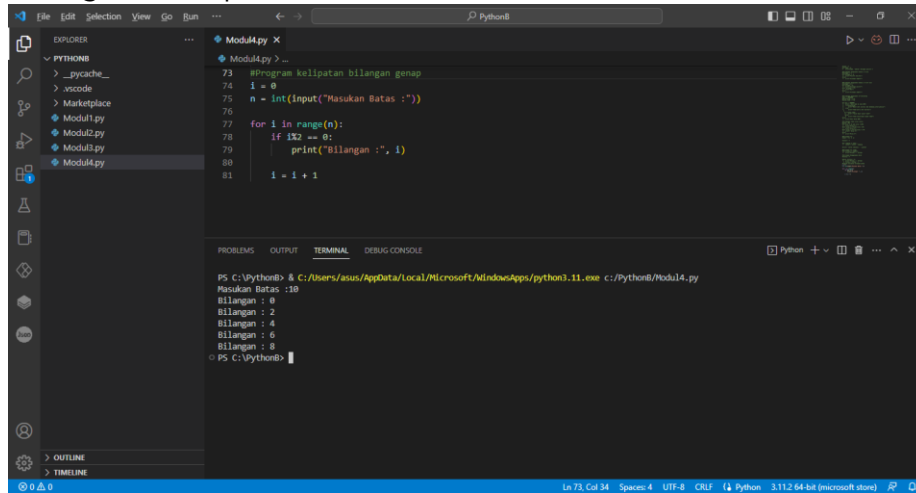
```
67 #Perulangan Menggunakan while
68 hitung = 0
69
70 while (hitung < 5):
71     print("Hitung :", hitung)
72     hitung = hitung + 1
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
PS C:\Python8> & C:\Users\asus\AppData\Local\Microsoft\WindowsApps/python3.11.exe c:/Python8/Modul4.py
Hitung : 0
Hitung : 1
Hitung : 2
Hitung : 3
Hitung : 4
PS C:\Python8>
```

#### 4. Program Kelipatan Bilangan Genap

Program pengulangan dengan for. Contoh Program Kelipatan Bilangan Genap :



The screenshot shows a VS Code editor with a Python file named 'Modul4.py'. The code is as follows:

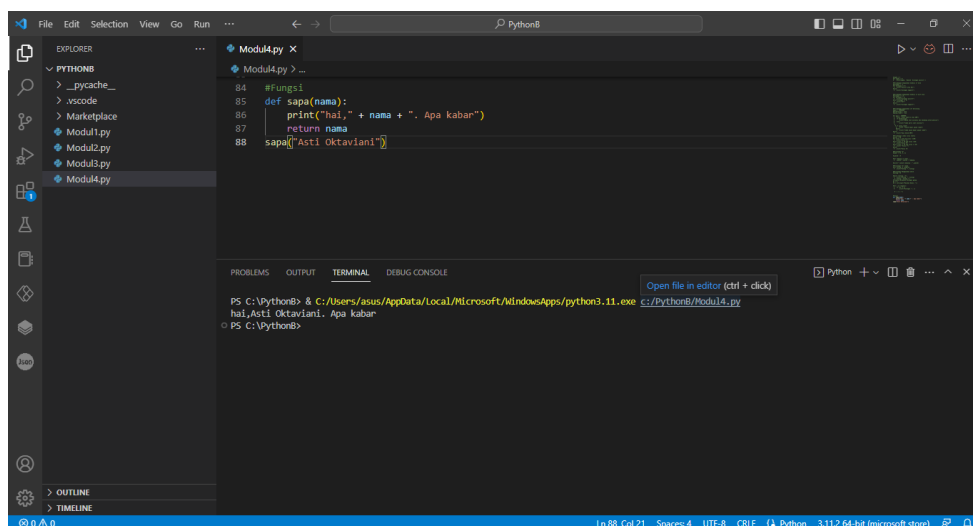
```
73 #Program kelipatan bilangan genap
74 i = 0
75 n = int(input("Masukan Batas :"))
76
77 for i in range(n):
78     if i%2 == 0:
79         print("Bilangan :", i)
80     i = i + 1
```

The terminal output shows the program running with an input of 10, printing even numbers from 0 to 8.

```
PS C:\Python8> & C:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Python8/Modul4.py
Masukan Batas :10
Bilangan : 0
Bilangan : 2
Bilangan : 4
Bilangan : 6
Bilangan : 8
PS C:\Python8>
```

#### C. Fungsi

Fungsi adalah grup/blok program untuk melakukan tugas tertentu yang berulang. Fungsi membuat kode program menjadi reusable, artinya hanya di definisikan sekali saja, dan kemudian bisa digunakan berulang kali dalam program. Contoh Program Fungsi :



The screenshot shows a VS Code editor with a Python file named 'Modul4.py'. The code is as follows:

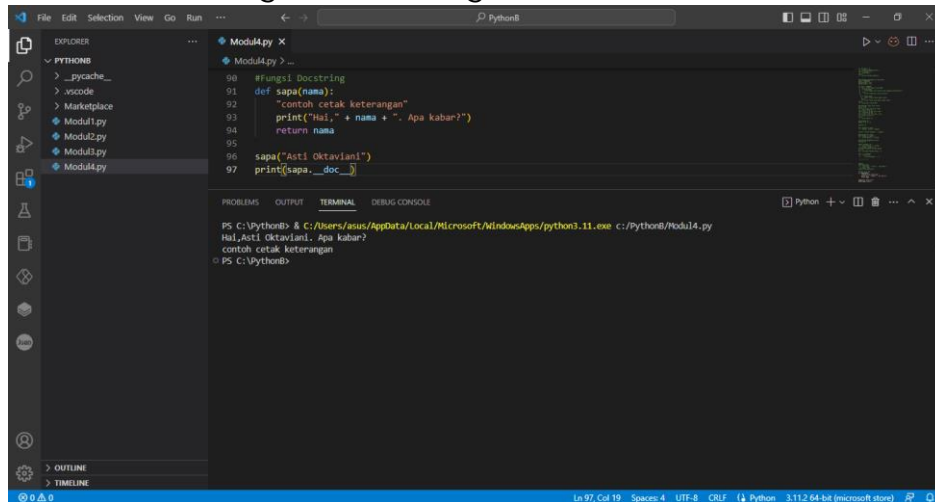
```
84 #fungsi
85 def sapa(nama):
86     print("hai," + nama + ", Apa kabar")
87     return nama
88 sapa("Asti Oktaviani")
```

The terminal output shows the function being called with the name 'Asti Oktaviani', resulting in the output 'hai,Asti Oktaviani. Apa kabar'.

```
PS C:\Python8> & C:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Python8/Modul4.py
hai,Asti Oktaviani. Apa kabar
PS C:\Python8>
```

## 1. Docstring

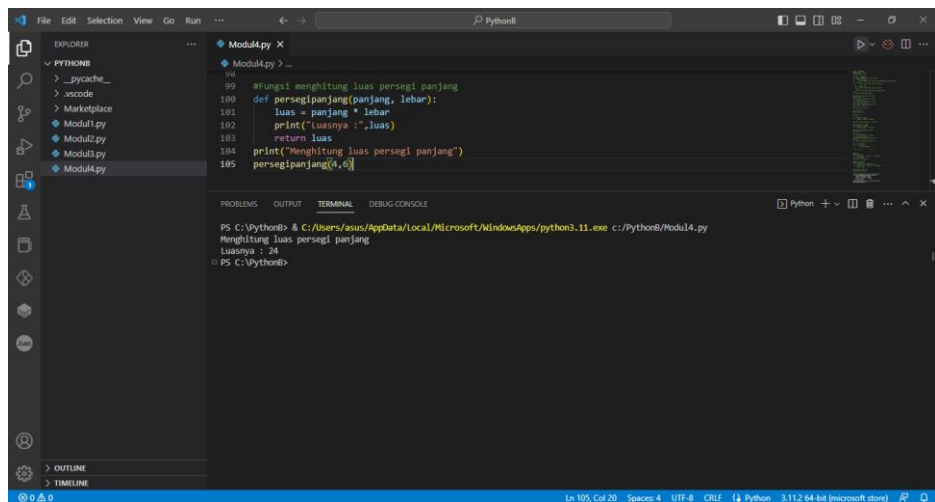
Docstring adalah singkatan dari documentation string berfungsi sebagai dokumentasi atau keterangan singkat tentang fungsi yang kita buat. Contoh Program Docstring



```
File Edit Selection View Go Run ... Python8
EXPLORER
PYTHONB
  _pycache_
  .vscode
  Marketplace
  Modul1.py
  Modul2.py
  Modul3.py
  Modul4.py
Modul4.py X
90 #Fungsi Docstring
91 def sapa(nama):
92     """contoh cetak keterangan"""
93     print("Hai," + nama + ". Apa kabar?")
94     return nama
95
96 sapa("Asti Oktaviani")
97 print(sapa.__doc__)

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Python
PS C:\PythonB> & C:\Users\asus\AppData\Local\Microsoft\WindowsApps\python3.11.exe c:/PythonB/Modul4.py
Hai,Asti Oktaviani. Apa kabar?
contoh cetak keterangan
PS C:\PythonB>
```

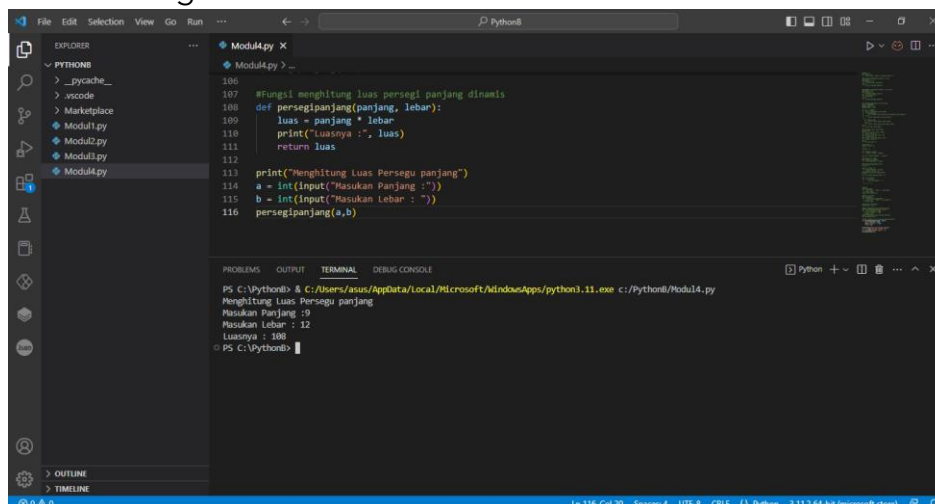
## 2. Contoh Program Luas Persegi Panjang dengan Fungsi. Contoh Program Statis



```
File Edit Selection View Go Run ... Python8
EXPLORER
PYTHONB
  _pycache_
  .vscode
  Marketplace
  Modul1.py
  Modul2.py
  Modul3.py
  Modul4.py
Modul4.py X
99 #Fungsi menghitung luas persegi panjang
100 def persegi panjang(panjang, lebar):
101     luas = panjang * lebar
102     print("Luasnya :",luas)
103     return luas
104 print("Menghitung luas persegi panjang")
105 persegi panjang(4,6)

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Python
PS C:\PythonB> & C:\Users\asus\AppData\Local\Microsoft\WindowsApps\python3.11.exe c:/PythonB/Modul4.py
Menghitung luas persegi panjang
Luasnya : 24
PS C:\PythonB>
```

Contoh Program Dinamis :



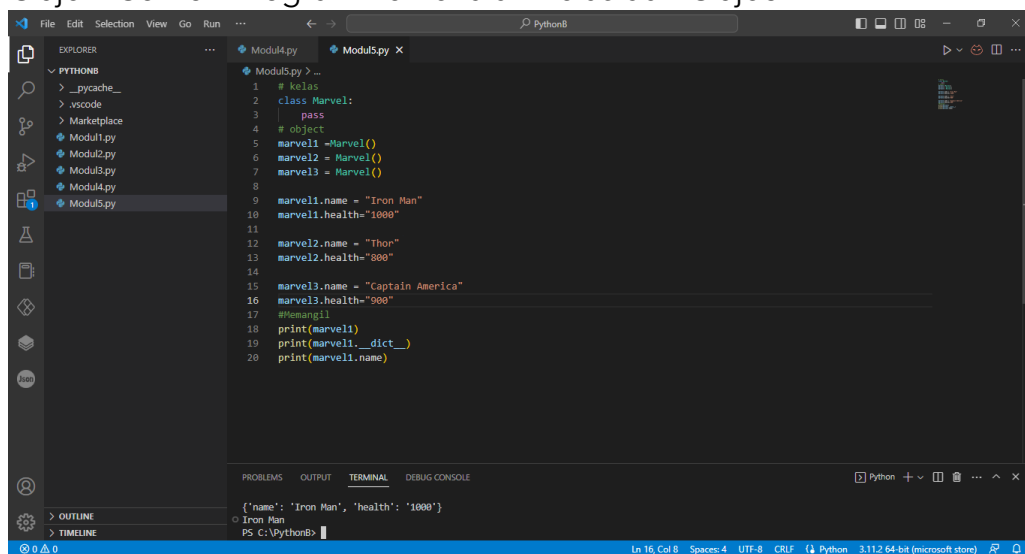
```
File Edit Selection View Go Run ... Python8
EXPLORER
PYTHONB
  _pycache_
  .vscode
  Marketplace
  Modul1.py
  Modul2.py
  Modul3.py
  Modul4.py
Modul4.py X
106
107 #Fungsi menghitung luas persegi panjang dinamis
108 def persegi panjang(panjang, lebar):
109     luas = panjang * lebar
110     print("Luasnya :", luas)
111     return luas
112
113 print("Menghitung luas Persegi panjang")
114 a = int(input("Masukan Panjang :"))
115 b = int(input("Masukan Lebar : "))
116 persegi panjang(a,b)

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Python
PS C:\PythonB> & C:\Users\asus\AppData\Local\Microsoft\WindowsApps\python3.11.exe c:/PythonB/Modul4.py
Menghitung luas Persegi panjang
Masukan Panjang : 9
Masukan Lebar : 12
Luasnya : 108
PS C:\PythonB>
```

## MODUL 5

### A. Object Oriented Programming

Object Oriented Programming (OOP) merupakan suatu konsep pemrograman yang menekankan pada paradigma atau cara pandang terhadap suatu masalah berdasarkan object. Istilah dalam OOP yaitu sebagai berikut : Kelas, Variabel, Data member. Overloading, Overloading Operator, Variabel, Pewarisan/Inheritance, Instance, Instansiasi , Metode, Objek. Contoh Program Perkenalan Kelas dan Object

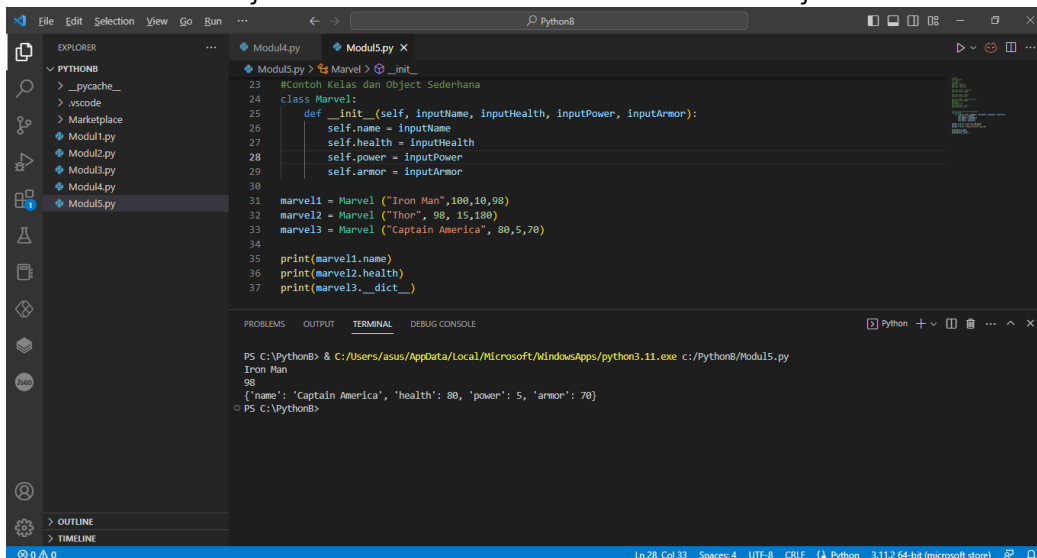


```
1 # kelas
2 class Marvel:
3     pass
4 # object
5 marvel1 = Marvel()
6 marvel2 = Marvel()
7 marvel3 = Marvel()
8
9 marvel1.name = "Iron Man"
10 marvel1.health = "1000"
11
12 marvel2.name = "Thor"
13 marvel2.health = "800"
14
15 marvel3.name = "Captain America"
16 marvel3.health = "900"
17 #Mencari
18 print(marvel1)
19 print(marvel1.__dict__)
20 print(marvel1.name)
```

Terminal output:

```
{'name': 'Iron Man', 'health': '1000'}
Iron Man
PS C:\Python8>
```

#### 1. Kelas dan Object Sederhana. Contoh Kelas dan Object Sederhana

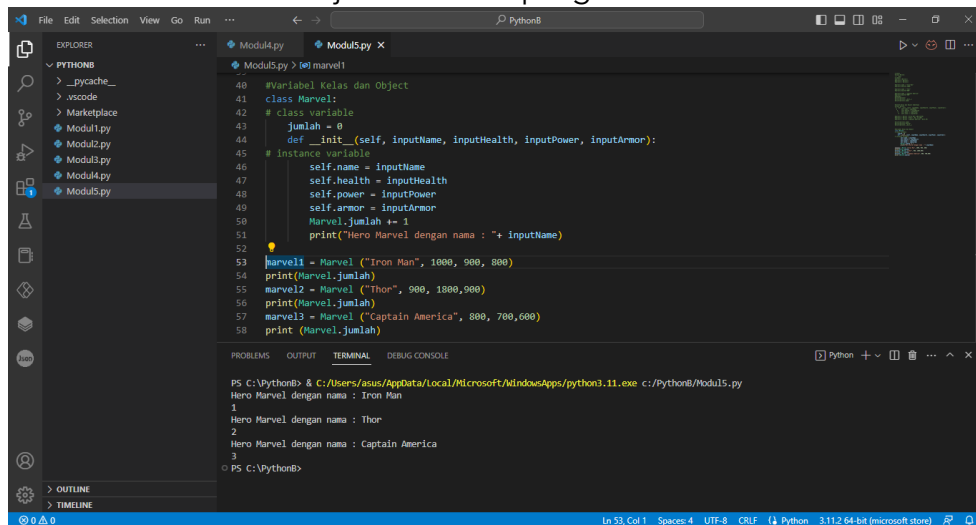


```
23 #Contoh Kelas dan Object Sederhana
24 class Marvel:
25     def __init__(self, inputName, inputHealth, inputPower, inputArmor):
26         self.name = inputName
27         self.health = inputHealth
28         self.power = inputPower
29         self.armor = inputArmor
30
31 marvel1 = Marvel ("Iron Man",100,10,98)
32 marvel2 = Marvel ("Thor", 98, 15,100)
33 marvel3 = Marvel ("Captain America", 80,5,70)
34
35 print(marvel1.name)
36 print(marvel2.health)
37 print(marvel3.__dict__)
```

Terminal output:

```
PS C:\Python8> & C:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Python8/Modul5.py
Iron Man
98
{'name': 'Captain America', 'health': 80, 'power': 5, 'armor': 70}
PS C:\Python8>
```

## 2. Variabel Kelas dan Object. Contoh program Variabel Kelas dan Object

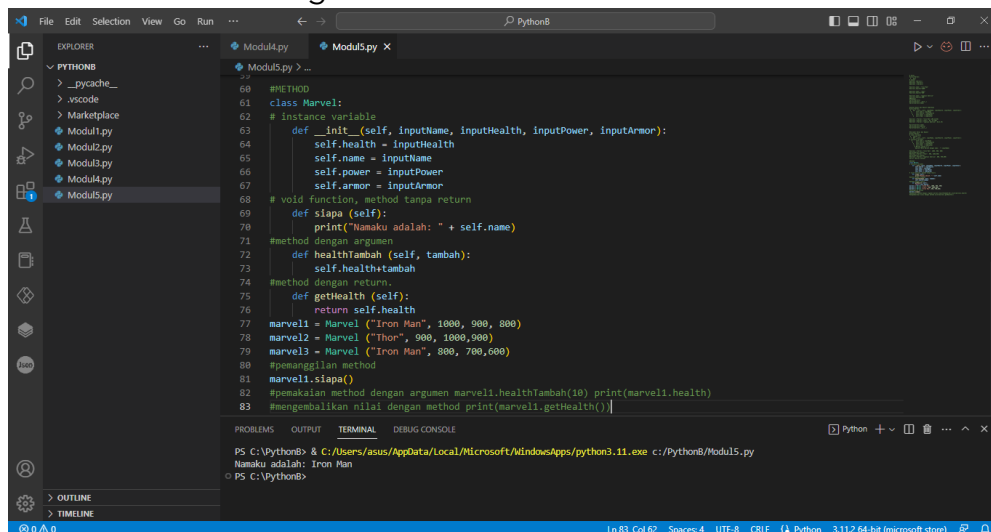


```
40 #Variabel Kelas dan Object
41 class Marvel:
42     # class variable
43     jumlah = 0
44     def __init__(self, inputName, inputHealth, inputPower, inputArmor):
45         # instance variable
46         self.name = inputName
47         self.health = inputHealth
48         self.power = inputPower
49         self.armor = inputArmor
50         Marvel.jumlah += 1
51         print("Hero Marvel dengan nama : "+ inputName)
52
53 marvel1 = Marvel ("Iron Man", 1000, 900, 800)
54 print(Marvel.jumlah)
55 marvel2 = Marvel ("Thor", 900, 1800, 900)
56 print(Marvel.jumlah)
57 marvel3 = Marvel ("Captain America", 800, 700, 600)
58 print(Marvel.jumlah)
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
PS C:\Python8> & C:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Python8/Modul5.py
1 Hero Marvel dengan nama : Iron Man
2 Hero Marvel dengan nama : Thor
3 Hero Marvel dengan nama : Captain America
4 PS C:\Python8>
```

## 3. Method. Contoh Program Method



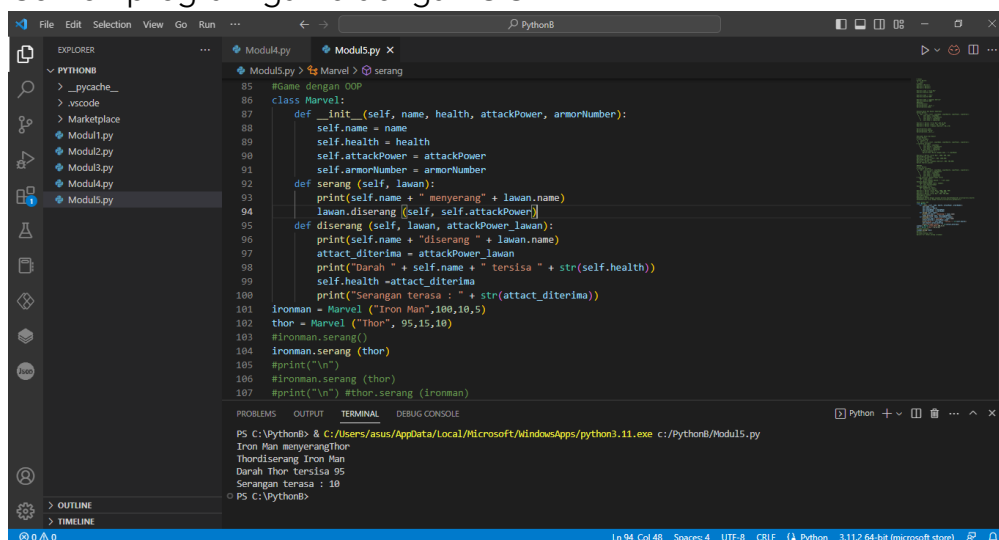
```
60 #METHOD
61 class Marvel:
62     # instance variable
63     def __init__(self, inputName, inputHealth, inputPower, inputArmor):
64         self.health = inputHealth
65         self.name = inputName
66         self.power = inputPower
67         self.armor = inputArmor
68     # void function, method tanpa return
69     def siapa(self):
70         print("Namaku adalah: " + self.name)
71     #method dengan argumen
72     def healthTambah(self, tambah):
73         self.health+=tambah
74     #method dengan return.
75     def getHealth(self):
76         return self.health
77     marvel1 = Marvel ("Iron Man", 1000, 900, 800)
78     marvel2 = Marvel ("Thor", 900, 1800, 900)
79     marvel3 = Marvel ("Iron Man", 800, 700, 600)
80     #pemanggilan method
81     marvel1.siapa()
82     #pemakaian method dengan argumen marvel1.healthTambah(10) print(marvel1.health)
83     #mengembalikan nilai dengan method print(marvel1.getHealth())
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
PS C:\Python8> & C:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Python8/Modul5.py
1 Namaku adalah: Iron Man
2 PS C:\Python8>
```

## 4. Game dengan OOP

Contoh program game dengan OOP :



```
85 #Game dengan OOP
86 class Marvel:
87     def __init__(self, name, health, attackPower, armorNumber):
88         self.name = name
89         self.health = health
90         self.attackPower = attackPower
91         self.armorNumber = armorNumber
92     def serang(self, lawan):
93         print(self.name + " menyerang" + lawan.name)
94         lawan.diserang(self, self.attackPower)
95     def diserang(self, lawan, attackPower_lawan):
96         print(self.name + " diserang " + lawan.name)
97         attack_diterima = attackPower_lawan
98         print("Darah " + self.name + " tersisa " + str(self.health))
99         self.health -=attack_diterima
100         print("Serangan terasa : " + str(attack_diterima))
101     ironman = Marvel ("Iron Man",100,10,5)
102     thor = Marvel ("Thor", 95,15,10)
103     #ironman.serang()
104     ironman.serang(thor)
105     #print("\n")
106     #ironman.serang(thor)
107     #print("\n") #thor.serang(ironman)
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
PS C:\Python8> & C:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Python8/Modul5.py
1 Iron Man menyerangThor
2 Thordiserang Iron Man
3 Darah Thor tersisa 95
4 Serangan terasa : 10
5 PS C:\Python8>
```



## MODUL 6

### A. Tkinter

Tkinter adalah graphic user interface (GUI) standar python digunakan untuk membuat tampilan aplikasi dengan komponen-komponen yang ada di modul tkinter seperti Button, Textbox, Label, Frame, Window yang mana sangat mendukung dalam penciptaan aplikasi GUI.

#### 1. Memanggil Modul Tkinter

Modul Tkinter berisi berbagai kelas dan komponen yang diperlukan untuk membangun sebuah aplikasi GUI. Untuk menggunakan komponen Tkinter, kita harus memanggilya terlebih dahulu. `import Tkinter` Kita juga bisa memanggil semua metode, kelas, dan atribut dari modul Tkinter dengan `from Tkinter import *`

#### 2. Membuat Jendela Aplikasi GUI

Untuk membuat jendela utama dari suatu aplikasi, kita menggunakan pernyataan seperti berikut: `root = Tkinter.Tk()` Jika kita menggunakan pernyataan `from-import` untuk memanggil modul Tkinter, maka kita cukup menuliskan : `root = Tk()`

#### 3. Menambahkan Komponen

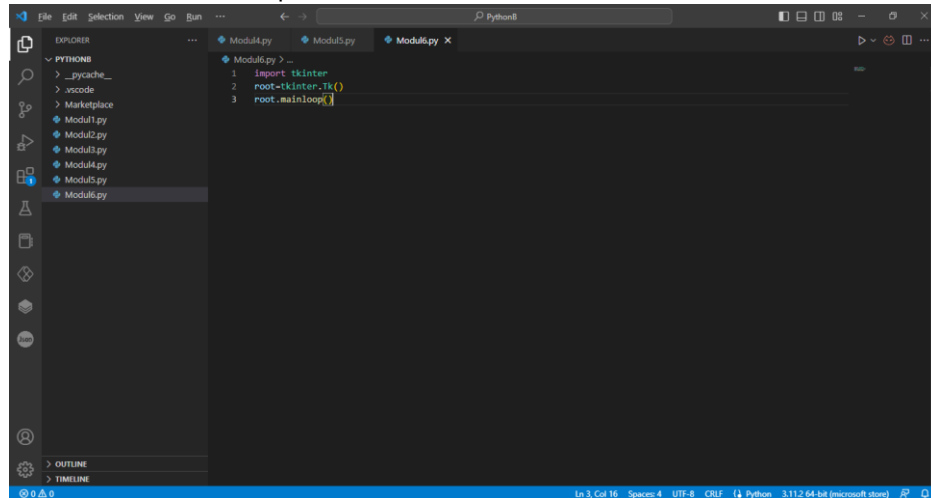
Pada Aplikasi Dengan menggunakan Tkinter, kita dapat menambahkan beberapa komponen pada aplikasi yang kita buat. Komponen ini bisa berdiri sendiri atau dapat diisi komponen lain (disebut Container). Komponen yang berdiri sendiri merupakan komponen yang tidak dapat diisi oleh komponen lain, misalkan tombol, checkbox, dan label. Komponen container merupakan komponen yang bisa berisi/ditempeli oleh komponen lain seperti frame dan window.

#### 4. Menambahkan Kode mainLoop

Setelah kita mendesain sebuah aplikasi, Kode untuk memasukkan program kita dalam kondisi perputaran ini yaitu : `Tkinter.mainloop()` Jika kita menggunakan `from-import` untuk mengambil semua fungsi pada Tkinter, kita cukup menuliskan `root.mainloop()` Dengan root pada kode di atas mengacu pada window utama (toplevel ).

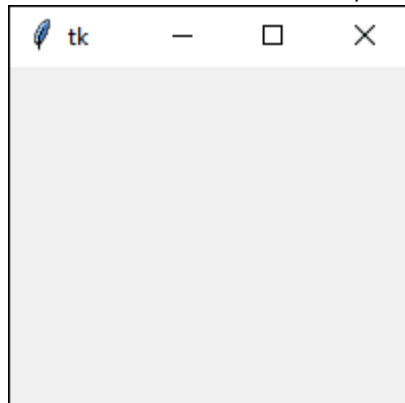
## 5. Tampilan GUI Sederhana

### Contoh Kode Tampilan GUI Sederhana.



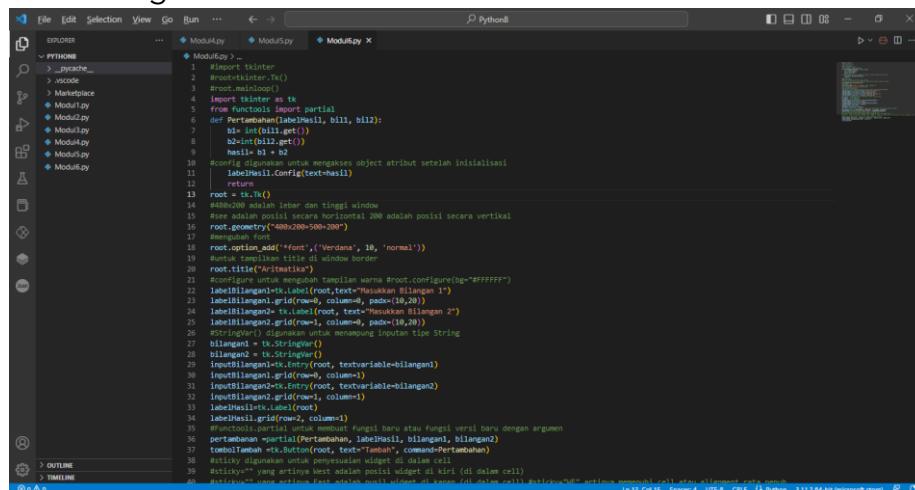
```
1 import tkinter
2 root=tkinter.Tk()
3 root.mainloop()
```

Setelah di-run akan tampil seperti gambar dibawah.



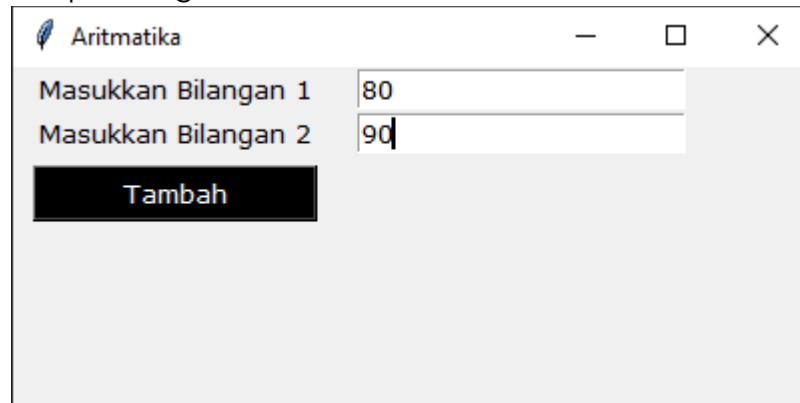
## Contoh Aplikasi Aritmatika Pertambahan

### Code Program

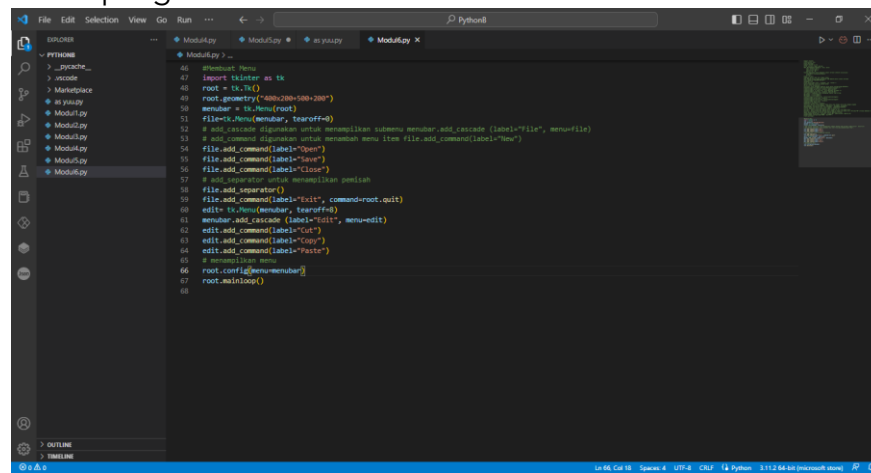


```
1 import tkinter
2 #root=tkinter.Tk()
3 from tkinter import *
4 import tkinter as tk
5 from functools import partial
6 def Pertambahan(labelMas1, bil1, bil2):
7     b1= int(bil1.get())
8     b2= int(bil2.get())
9     hasil= b1 + b2
10    #config digunakan untuk mengakses object atribut setelah inialisasi
11    labelMas1.config(text=hasil)
12    return
13    root = tk.Tk()
14    #400x200 adalah lebar dan tinggi window
15    #size adalah posisi secara horizontal 200 adalah posisi secara vertikal
16    root.geometry("400x200+500+200")
17    #mengatur font
18    root.option_add('*font','Verdana', 18, 'normal')
19    #untuk tampilan title di window border
20    root.title("Aritmatika")
21    #untuk part untuk menampilkan hasil warna #root.configure(bg="FFFFFF")
22    labelBilangan1=tk.Label(root,text="Masukkan Bilangan 1")
23    labelBilangan1.grid(row=0, column=0, padx=10, pady=10)
24    labelBilangan2=tk.Label(root, text="Masukkan Bilangan 2")
25    labelBilangan2.grid(row=1, column=0, padx=10, pady=10)
26    #StringVar() digunakan untuk menangkap inputan tipe String
27    bilangan1 = tk.StringVar()
28    bilangan2 = tk.StringVar()
29    inputBilangan1=tk.Entry(root, textvariable=bilangan1)
30    inputBilangan1.grid(row=0, column=1)
31    inputBilangan2=tk.Entry(root, textvariable=bilangan2)
32    inputBilangan2.grid(row=1, column=1)
33    labelMas1=tk.Label(root)
34    labelMas1.grid(row=2, column=1)
35    #fungsi partial akan membuat fungsi baru atau fungsi untuk baru dengan argumen
36    pertambahan=partial(Pertambahan, labelMas1, bilangan1, bilangan2)
37    tombolTambah=tk.Button(root, text="Tambah", command=pertambahan)
38    #tombol digunakan untuk pemrosesan input di dalam cell
39    #tombol yang artinya akan sebuah hasil widget di sini (di dalam cell)
40    tombolTambah.grid(row=3, column=1)
```

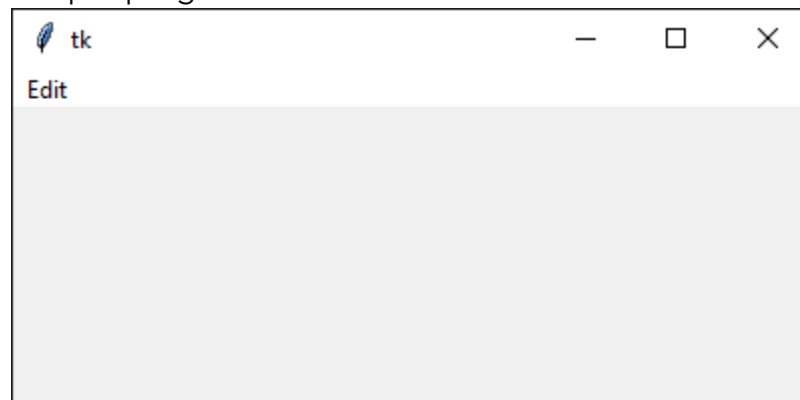
## Output Program



## Membuat Menu contoh Kode Tampilan Menu Code program :



## Output program :



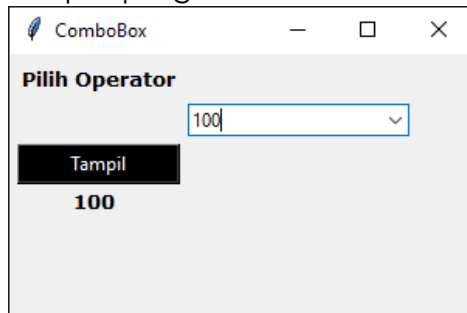
## MODUL 7

### A. ComboBox . ContohTampilan ComboBox

Code program :

[illegible]

Output program :



## B. RadioButton

## Code program radiobotton

[illegible]

## Tampilan RadioButton

