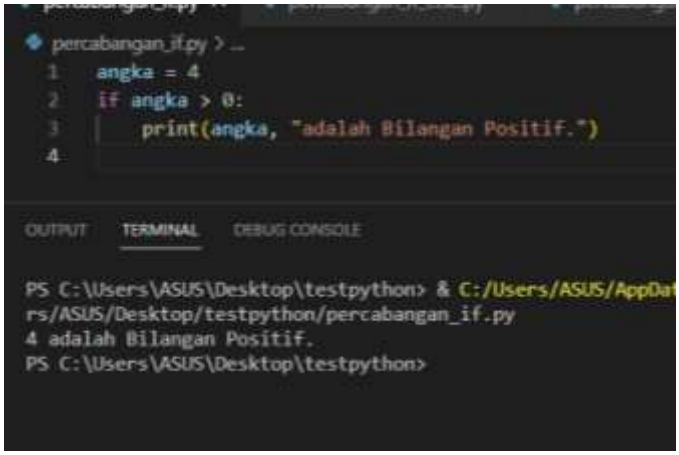
Nama : Novitasari

Nim : 20.01.013.012

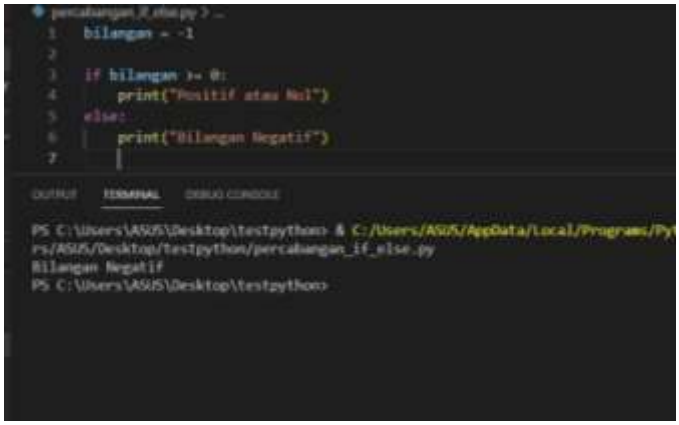
Kelas : Python_D

Modul 5

- Contoh pernyataan if:



- Contoh pernyataan if else:



- Contoh pernyataan if elif else:

```
percabangan_if.py    percabangan_if_else.py    percaba
percabangan_if_elif_else.py > ...
1   bilangan = 5.5
2
3   if bilangan > 0:
4       print("Bilangan Positif")
5   elif bilangan == 0:
6       print("Nol")
7   else:
8       print("Bilangan Negatif")
9

OUTPUT   TERMINAL   DEBUG CONSOLE
PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/Ap
rs/ASUS/Desktop/testpython/percabangan_if_elif_else.py
Bilangan Positif
PS C:\Users\ASUS\Desktop\testpython>
```

- Contoh pernyataan if bersarang:

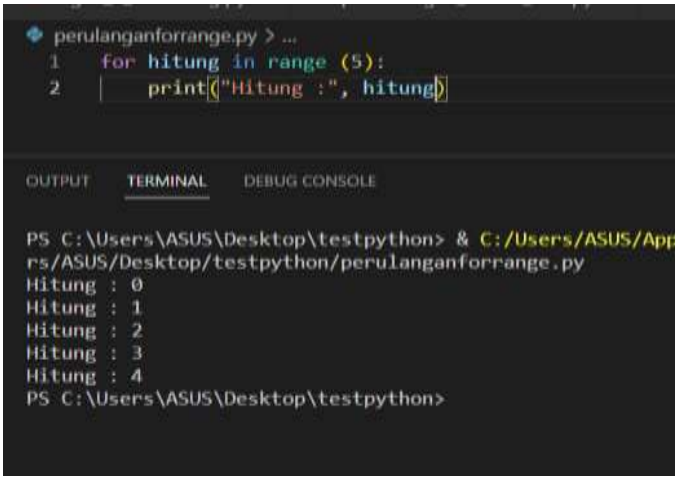
```
percabangan_if_bersarang.py > ...
1   gaji = 10000000
2   berkeluarga = True
3   punya_rumah = True
4
5   if gaji > 5000000:
6       print ("Gaji sudah diatas UMR")
7       if berkeluarga:
8           print ("Wajib ikut asuransi dan menabung untuk pensiun")
9       else:
10          print ("Tidak perlu ikut asuransi")
11      if punya_rumah:
12          print ("Wajib bayar pajak rumah")
13      else:
14          print ("Tidak wajib bayar pajak rumah")
15  else:
16      print ("Gaji belum UMR")

PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/AppData/Local/Programs/Py
rs/ASUS/Desktop/testpython/percabangan_if_bersarang.py
Gaji sudah diatas UMR
Wajib ikut asuransi dan menabung untuk pensiun
Wajib bayar pajak rumah
PS C:\Users\ASUS\Desktop\testpython>
```

- Contoh program percabangan indeks nilai statis:

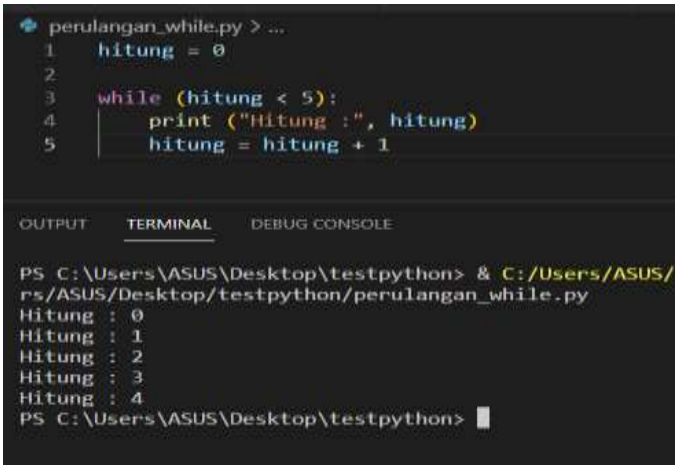
```
percabangan_indeks_nilai.py > ...
1   nilai = 80
2
3   if nilai >= 85 and nilai <= 100:
4       print ("Nilai A")
5   elif nilai >= 70 and nilai <= 84:
6       print ("Nilai B")
7   elif nilai >= 55 and nilai <= 69:
8       print ("Nilai C")
9   else:
10      print ("Nilai D")

OUTPUT   TERMINAL   DEBUG CONSOLE

PS C:\Users\ASUS\Desktop\testpython> & C:/Users/A
rs/ASUS/Desktop/testpython/percabangan_indeks_nil
Nilai B
PS C:\Users\ASUS\Desktop\testpython>
```

- Contoh perulangan for:

- Contoh perulangan for dengan range:



- Contoh perulangan while:



- Contoh program kelipatan bilangan genap:

```
perulanganfor.py    perulanganforrange.py    perulangan_v
perulangan_kelipatan_genap.py > ...
1  i = 0
2  n = int(input ("Masukkan Batas : "))
3
4  for i in range(n):
5      if i%2 == 0:
6          print("Bilangan :", i)
7      i = i + 1

OUTPUT   TERMINAL   DEBUG CONSOLE

PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/AppDa
rs/ASUS/Desktop/testpython/perulangan_kelipatan_genap.py
Masukkan Batas : 8
Bilangan : 0
Bilangan : 2
Bilangan : 4
Bilangan : 6
PS C:\Users\ASUS\Desktop\testpython>
```

- Contoh fungsi:

```
fungsi.py > ...
1  def sapa(nama):
2      print("Hai, " + nama + ". Apa kabar?")
3      return nama
4
5  # pemanggilan fungsi
6  # output: Hai, Novitasari. Apa kabar?
7  sapa("Novitasari")

OUTPUT   TERMINAL   DEBUG CONSOLE

PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/AppDat
rs/ASUS/Desktop/testpython/fungsi.py
Hai, Novitasari. Apa kabar?
PS C:\Users\ASUS\Desktop\testpython>
```

- Contoh docstring:

```
fungsi_docstring.py > ...
1  def sapa(nama):
2      "Contoh Cetak Keterangan"
3      print("Hai, "+ nama + ". Apa kabar?")
4      return nama
5
6  sapa("Novitasari")
7  print(sapa.__doc__)

OUTPUT   TERMINAL   DEBUG CONSOLE

PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/
rs/ASUS/Desktop/testpython/fungsi_docstring.py
Hai, Novitasari. Apa kabar?
Contoh Cetak Keterangan
PS C:\Users\ASUS\Desktop\testpython>
```

- Contoh program statis:

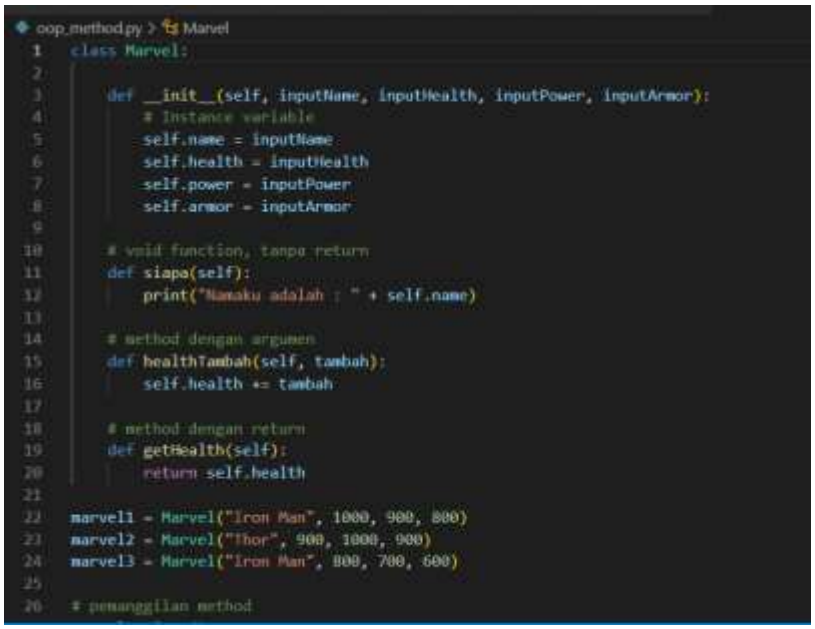
```
Fungsi_persegi_panjang.py > ...
1   def persegipanjang(panjang, lebar):
2       luas = panjang * lebar
3       print ("Luasnya :", luas)
4       return luas
5
6   print("Menghitung Luas Persegi Panjang")
7   persegipanjang(4,6)

OUTPUT  TERMINAL  DEBUG CONSOLE
PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/AppData/Loca
rs/ASUS/Desktop/testpython/Fungsi_persegi_panjang.py
Menghitung Luas Persegi Panjang
Luasnya : 24
PS C:\Users\ASUS\Desktop\testpython>
```

- Contoh program dinamis:

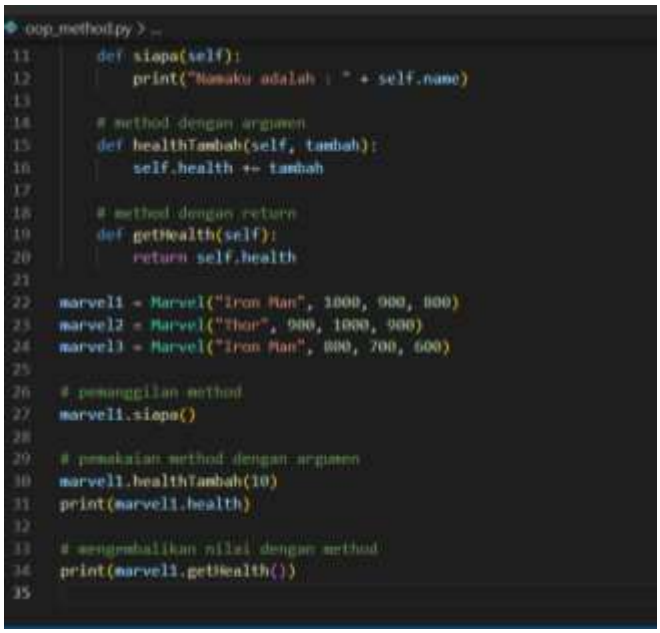
```
persegi_panjang_dinamis.py > ...
1   def persegipanjang(panjang, lebar):
2       luas =panjang * lebar
3       print ("Luasnya :", luas)
4       return luas
5   print("Menghitung Luas Persegi Panjang")
6   a = int(input("Masukkan Panjang : "))
7   b = int(input("Masukkan Lebar : "))
8   persegipanjang(a,b)

OUTPUT  TERMINAL  DEBUG CONSOLE
PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/AppData
rs/ASUS/Desktop/testpython/persegi_panjang_dinamis.py
Menghitung Luas Persegi Panjang
Masukkan Panjang : 8
Masukkan Lebar : 4
Luasnya : 32
PS C:\Users\ASUS\Desktop\testpython>
```

Modul 6

- Contoh program OOP Method:



```python
class Marvel:

    def __init__(self, inputName, inputHealth, inputPower, inputArmor):
        # Instance variable
        self.name = inputName
        self.health = inputHealth
        self.power = inputPower
        self.armor = inputArmor

    # void function, tanpa return
    def siapa(self):
        print("Namaku adalah : " + self.name)

    # method dengan argumen
    def healthTambah(self, tambah):
        self.health += tambah

    # method dengan return
    def getHealth(self):
        return self.health

marvel1 = Marvel("Iron Man", 1000, 900, 800)
marvel2 = Marvel("Thor", 900, 1000, 900)
marvel3 = Marvel("Iron Man", 800, 700, 600)

# pemanggilan method
```



```python
    def siapa(self):
        print("Namaku adalah : " + self.name)

    # method dengan argumen
    def healthTambah(self, tambah):
        self.health += tambah

    # method dengan return
    def getHealth(self):
        return self.health

marvel1 = Marvel("Iron Man", 1000, 900, 800)
marvel2 = Marvel("Thor", 900, 1000, 900)
marvel3 = Marvel("Iron Man", 800, 700, 600)

# pemanggilan method
marvel1.siapa()

# pemakaian method dengan argumen
marvel1.healthTambah(10)
print(marvel1.health)

# mengembalikan nilai dengan method
print(marvel1.getHealth())
```

```
oop_method.py > ...
11      def siapa(self):
12          print("Namaku adalah : " + self.name)
13
14      # method dengan argumen
15      def healthTambah(self, tambah):
16          self.health += tambah
17
18      # method dengan return
19      def getHealth(self):
20          return self.health
21

OUTPUT   TERMINAL   DEBUG CONSOLE

PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/AppData/Local
rs/ASUS/Desktop/testpython/oop_method.py
Namaku adalah : Iron Man
1010
1010
PS C:\Users\ASUS\Desktop\testpython>
```

- Contoh program OOP dasar:

```
oop_dasar.py > ...
1   # kelas
2   class Marvel:
3       pass
4
5   # object
6
7   marvel1 = Marvel()
8   marvel2 = Marvel()
9   marvel3 = Marvel()
10
11  marvel1.name = "Iron Man"
12  marvel1.health = "1000"
13
14  marvel2.name = "thor"
15  marvel2.health = "800"
16
17  marvel3.name = "Captain America"
18  marvel3.health = "900"
19
20  # pemanggilan
21  print(marvel1)
22  print(marvel1.__dict__)
23  print(marvel1.name)
24
```

```
oop_dasar.py > ...
1   # kelas
2   class Marvel:
3       pass
4
5   # object
6
7   marvel1 = Marvel()
8   marvel2 = Marvel()
9   marvel3 = Marvel()
10
11  marvel1.name = "Iron Man"

OUTPUT   TERMINAL   DEBUG CONSOLE

PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/A
rs/ASUS/Desktop/testpython/oop_dasar.py
<__main__.Marvel object at 0x0000021FB9F8E550>
{'name': 'Iron Man', 'health': '1000'}
Iron Man
PS C:\Users\ASUS\Desktop\testpython>
```

- Contoh Program OOP Game

```python
class Marvel:
    def __init__(self, name, health, attackPower, armorNumber):
        self.name = name
        self.health = health
        self.attackPower = attackPower
        self.armorNumber = armorNumber

    def serang(self, lawan):
        print(self.name + " menyerang " + lawan.name)
        lawan.diserang(self, self.attackPower)

    def diserang(self, lawan, attackPower_lawan):
        print(self.name + " diserang " + lawan.name)
        attack_diterima = attackPower_lawan
        print("Serangan terasa : " + str(attack_diterima))
        self.health -= attack_diterima
        print("Darah " + self.name + " tersisa " + str(self.health))

ironman = Marvel("Iron Man", 100, 10, 5)
thor = Marvel("Thor", 95, 15, 10)

#ironman.serang()
ironman.serang(thor)
#print("\n")
#ironman.serang(thor)
#print("\n")
```

```
PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/AppData/Local/Pro
rs/ASUS/Desktop/testpython/oop_game.py
Iron Man menyerang Thor
Thor diserang Iron Man
Serangan terasa : 10
Darah Thor tersisa 85
PS C:\Users\ASUS\Desktop\testpython>
```

- Contoh program OOP Init:

```python
class Marvel:
    def __init__(self, inputName, inputHealth, inputPower, inputArmor):
        self.name = inputName
        self.health = inputHealth
        self.power = inputPower
        self.armor = inputArmor

marvel1 = Marvel("Iron Man", 100, 10, 90)
marvel2 = Marvel("Thor", 90, 15, 100)
marvel3 = Marvel("Captain America", 80, 5, 70)

print(marvel1.name)
print(marvel2.health)
print(marvel3.__dict__)
```

```
PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/AppData/Local/Programs/Python/
rs/ASUS/Desktop/testpython/oop_init.py
Iron Man
90
{'name': 'Captain America', 'health': 80, 'power': 5, 'armor': 70}
PS C:\Users\ASUS\Desktop\testpython>
```

- Contoh program OOP kelas instance:

```python
class Marvel:
    # class variable
    jumlah = 0

    def __init__(self, inputName, inputHealth, inputPower, inputArmor):
        # instance variable
        self.name = inputName
        self.health = inputHealth
        self.power = inputPower
        self.armor = inputArmor
        Marvel.jumlah +=1
        print("Hero Marvel dengan nama : " + inputName)

marvel1 = Marvel("Iron Man", 1000, 900, 800)
print(Marvel.jumlah)
marvel2 = Marvel("Thor", 900, 1000, 900)
print(Marvel.jumlah)
marvel3 = Marvel("Captain America", 800, 700, 600)
print(Marvel.jumlah)
```

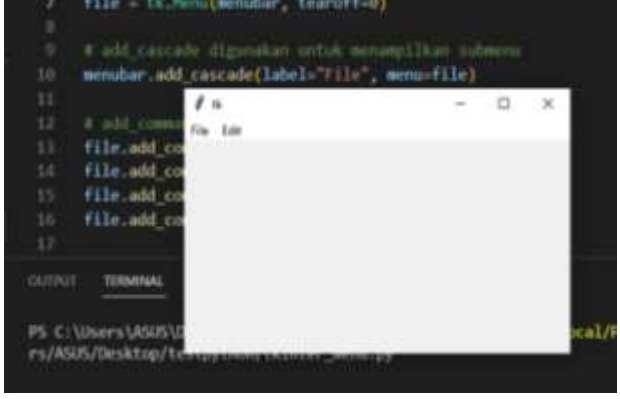
```
PS C:\Users\ASUS\Desktop\testpython> & C:/Users/ASUS/AppData/Local/Programs/Python/
rs/ASUS/Desktop/testpython/oop_kelas_instance.py
Hero Marvel dengan nama : Iron Man
1
Hero Marvel dengan nama : Thor
2
Hero Marvel dengan nama : Captain America
3
PS C:\Users\ASUS\Desktop\testpython>
```

Module 7

- Contoh program Tkinter menu

```python
import tkinter as tk

root = tk.Tk()
root.geometry('400x200+500+200')

menubar = tk.Menu(root)
file = tk.Menu(menubar, tearoff=0)

# add_cascade digunakan untuk menampilkan submenu
menubar.add_cascade(label="File", menu=file)

# add_command digunakan untuk menambah menu item
file.add_command(label="New")
file.add_command(label="Open")
file.add_command(label="Save")
file.add_command(label="Close")

# add_separator digunakan untuk menampilkan pemisah
file.add_separator()
file.add_command(label="Exit", command=root.quit)

edit = tk.Menu(menubar, tearoff=0)
menubar.add_cascade(label="Edit", menu=edit)
edit.add_command(label="Cut")
edit.add_command(label="Copy")
edit.add_command(label="Paste")
```

```python
file = tk.Menu(menubar, tearoff=0)

# add_cascade digunakan untuk menampilkan submenu
menubar.add_cascade(label="File", menu=file)

# add_command digunakan untuk menambah menu item
file.add_command(label="New")
file.add_command(label="Open")
file.add_command(label="Save")
file.add_command(label="Close")

# add_separator digunakan untuk menampilkan pemisah
file.add_separator()
file.add_command(label="Exit", command=root.quit)

edit = tk.Menu(menubar, tearoff=0)
menubar.add_cascade(label="Edit", menu=edit)
edit.add_command(label="Cut")
edit.add_command(label="Copy")
edit.add_command(label="Paste")

# menampilkan menu
root.config(menu=menubar)
root.mainloop()
```



- Contoh Program Tkinter Aritmatika

tkinter_aritmatika.py > ...

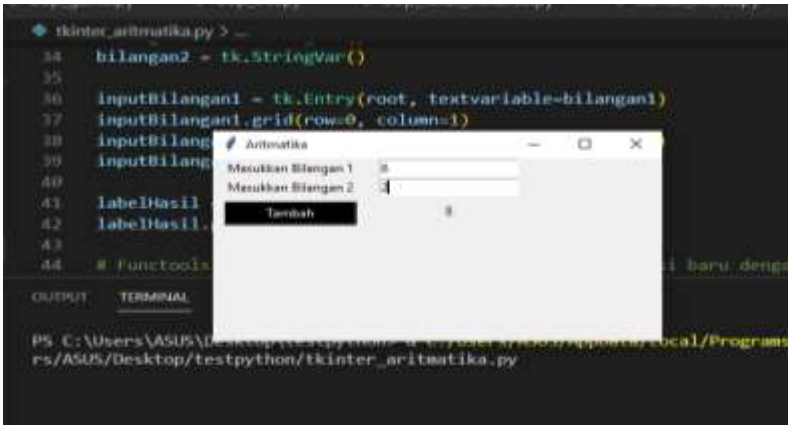
```python
import tkinter as tk
from functools import partial

def pertambahan (labelhasil, bil1, bil2):
    b1 = int(bil1.get())
    b2 = int(bil2.get())
    hasil = b1 + b2

    # config digunakan untuk mengakses object atribut setelah inisialisasi
    labelhasil.config(text=hasil)
    return

root = tk.Tk()
# 400x200 adalah lebar dan tinggi
# 500 adalah posisi secara horizontal
# 200 adalah posisi secara vertikal
root.geometry("400x200+500+200")

# mengubah font
root.option_add("font", ("Verdana", 10, "normal"))

# Menampilkan title di window border
root.title('Aritmatika')

# configure untuk mengubah warna
# root.configure(bg= "#FFFFFF")
```

oop_game.py | oop_init.py | oop_kelas_instance.py | tkinter_menu.py | tkinter_aritmatika

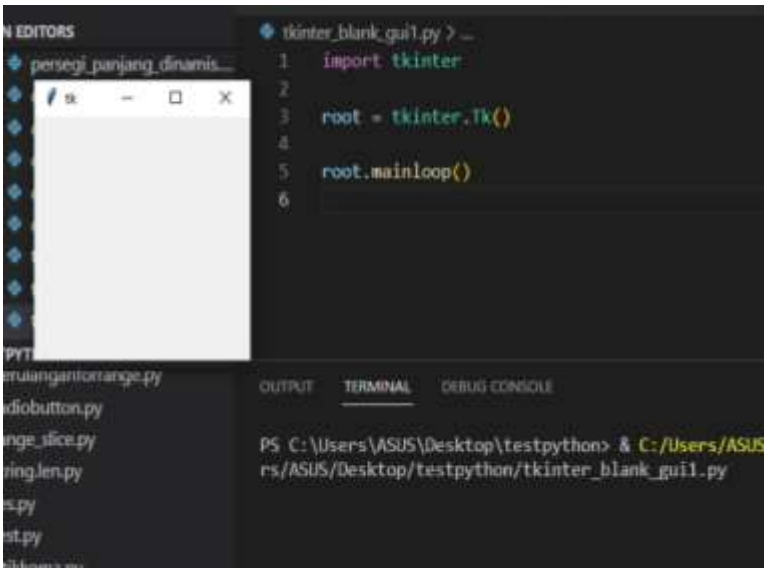
tkinter_aritmatika.py > ...

```python
# root.configure(bg= "#FFFFFF")

labelBilangan1 = tk.Label(root, text= "Masukkan Bilangan 1")
labelBilangan1.grid(row=0, column=0, padx=(10,20))
labelBilangan2 = tk.Label(root, text="Masukkan Bilangan 2")
labelBilangan2.grid(row=1, column=0, padx=(10,20))
# StringVar() digunakan untuk menampung inputan tipe String
bilangan1 = tk.StringVar()
bilangan2 = tk.StringVar()

inputBilangan1 = tk.Entry(root, textvariable=bilangan1)
inputBilangan1.grid(row=0, column=1)
inputBilangan2 = tk.Entry(root, textvariable=bilangan2)
inputBilangan2.grid(row=1, column=1)

labelHasil = tk.Label(root)
labelHasil.grid(row=2, column=1)

# functools.partial untuk membuat fungsi baru atau versi baru dengan argumen
pertambahan = partial(pertambahan, labelHasil, bilangan1, bilangan2)
tombolTambah = tk.Button(root, text = "Tambah", command=pertambahan)
# sticky digunakan untuk penyesuaian widget di dalam cell
# sticky="W" yang artinya adalah West adalah posisi widget di kiri (di dalam cell)
# sticky="E" yang artinya adalah East adalah posisi widget di kanan (di dalam cell)
# sticky = "WE" artinya memenuhi cell atau alignment rata penuh
# padx adalah padding horizontal
```

```
bilangan2 = tk.StringVar()

inputBilangan1 = tk.Entry(root, textvariable=bilangan1)
inputBilangan1.grid(row=0, column=1)
inputBilangan2 = tk.Entry(root, textvariable=bilangan2)
inputBilangan2.grid(row=1, column=1)

labelHasil = tk.Label(root)
labelHasil.grid(row=2, column=1)

# Functools.partial untuk membuat fungsi baru atau versi baru dengan argumen
pertambahan = partial(pertambahan, labelHasil, bilangan1, bilangan2)
tombolTambah = tk.Button(root, text = "Tambah", command=pertambahan)
# sticky digunakan untuk penyesuaian widget di dalam cell
# sticky="W" yang artinya adalah West adalah posisi widget di kiri (di dalam cell)
# sticky="E" yang artinya adalah East adalah posisi widget di kanan (di dalam cell)
# sticky = "WE" artinya memenuhi cell atau alignment rata penuh
# padx adalah padding horizontal
# pady adalah padding vertikal

tombolTambah.grid(row=2, column=0, sticky="WE", padx=(10,20), pady=(5,0))
tombolTambah.config(bg="#000", fg="#fff")

root.mainloop()
```



- Contoh Program Tkinter blank gui1:



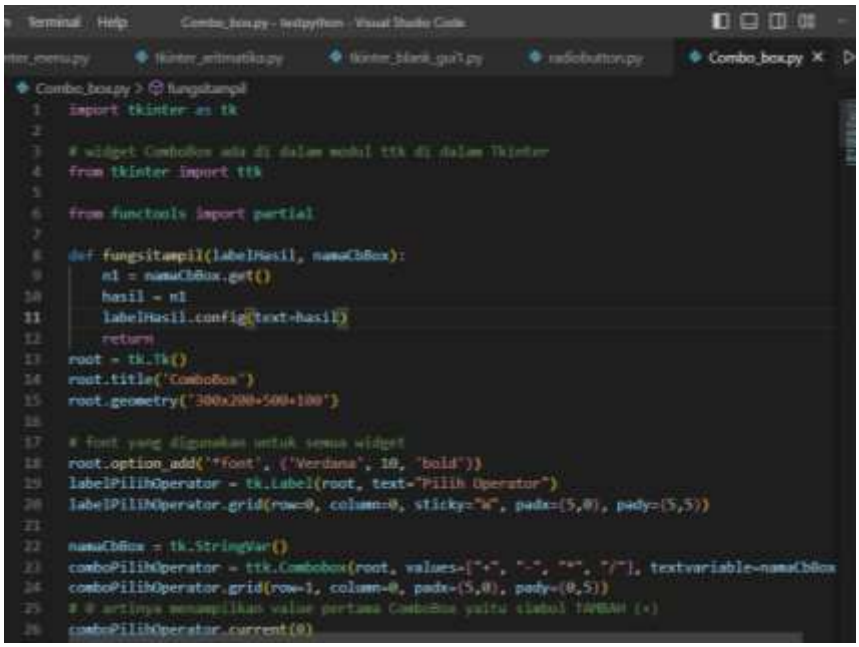
Modul 8

- Contoh Program Radiobutton

```
radiobutton.py > ...
1   import tkinter as tk
2   from functools import partial
3
4   def fungsitampil(labelhasil, rb):
5       ambil = rb.get()
6       hasil = ambil
7       labelhasil.config(text=hasil)
8       return
9
10  root = tk.Tk()
11  root.title("Radio Button")
12  root.geometry("300x200+500+100")
13  root.option_add('*Font', ('Verdana', 10, 'normal'))
14  root.option_add('*Label.font', ('Verdana', 10, 'bold'))
15
16  labelPilihJurusan = tk.Label(root, text="Pilih Jurusan")
17  labelPilihJurusan.grid(row=0, column=0, sticky="W", padx=(5,0), pady=(5,5))
18
19  rbValue = tk.IntVar()
20  # Variable akan terisi value (nilai) berdasarkan pemilihan tombol
21  # value, memberikan nilai ke variabel
22  rbSI = tk.Radiobutton(root, text="Sistem Informasi", variable=rbValue, value=1)
23  rbSI.grid(row=1, column=0, sticky="W")
24  rbIF = tk.Radiobutton(root, text="Teknik Informatika", variable=rbValue, value=2)
25  rbIF.grid(row=2, column=0, sticky="W")
26  rbKA = tk.Radiobutton(root, text="Komputer Akuntansi", variable=rbValue, value=3)
```

```
radiobutton.py > ...
14  root.option_add('*Label.font', ('Verdana', 10, 'bold'))
15
16  labelPilihJurusan = tk.Label(root, text="Pilih Jurusan")
17  labelPilihJurusan.grid(row=0, column=0, sticky="W", padx=(5,0), pady=(5,5))
18
19  rbValue = tk.IntVar()
20  # Variable akan terisi value (nilai) berdasarkan pemilihan tombol
21  # value, memberikan nilai ke variabel
22  rbSI = tk.Radiobutton(root, text="Sistem Informasi", variable=rbValue, value=1)
23  rbSI.grid(row=1, column=0, sticky="W")
24  rbIF = tk.Radiobutton(root, text="Teknik Informatika", variable=rbValue, value=2)
25  rbIF.grid(row=2, column=0, sticky="W")
26  rbKA = tk.Radiobutton(root, text="Komputer Akuntansi", variable=rbValue, value=3)
27  rbKA.grid(row=3, column=0, sticky="W")
28
29  labelHasil = tk.Label(root)
30  labelHasil.grid(row=5, column=0)
31
32  tampil = partial(fungsitampil, labelHasil, rbValue)
33  tombolTampil = tk.Button(root, text="Tampil", command=tampil)
34  tombolTampil.grid(row=4, column=0, sticky="W", padx=(5,0))
35  tombolTampil.configure(bg="#000", fg="#fff")
36
37  root.mainloop()
38
```

- Contoh program Combo Box



```python
root = tk.Tk()
root.title('ComboBox')
root.geometry('300x200+500+100')

# font yang digunakan untuk semua widget
root.option_add('*font', ('Verdana', 10, 'bold'))
labelPilihOperator = tk.Label(root, text="Pilih Operator")
labelPilihOperator.grid(row=0, column=0, sticky="w", padx=(5,0), pady=(5,5))

namaCbBox = tk.StringVar()
comboPilihOperator = ttk.Combobox(root, values=["+", "-", "*", "/"], textvariable=namaCbBox
comboPilihOperator.grid(row=1, column=0, padx=(5,0), pady=(0,5))
# 0 artinya menampilkan value pertama ComboBox yaitu simbol TAMBAH (+)
comboPilihOperator.current(0)

labelHasil = tk.Label(root)
labelHasil.grid(row=3, column=0)

tampil = partial(fungsitampil, labelHasil, namaCbBox)
tombolTampil = tk.Button(root, text="Tampil", command=tampil)
tombolTampil.grid(row=2, column=0, sticky="WE", padx=(5,0))
tombolTampil.configure(bg="#000", fg="#fff")

root.mainloop()
```