

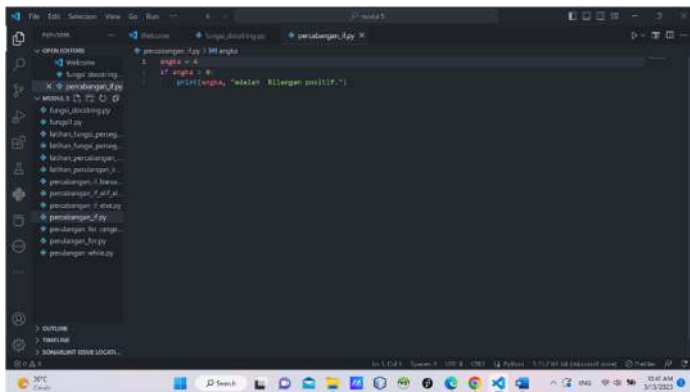
Nama : Nurul Mufliha Puasa

Nim : 20.01.013.014

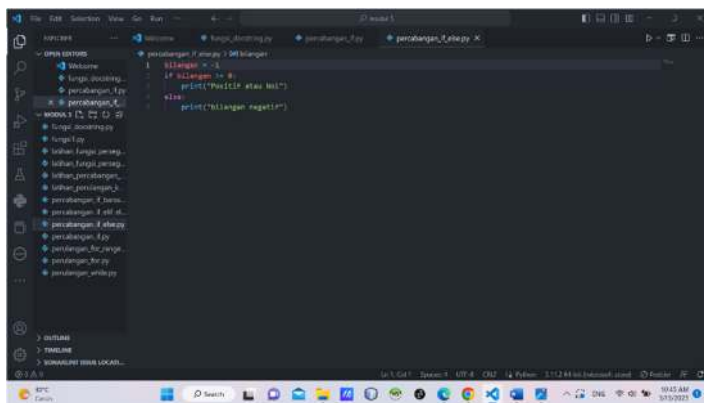
Kelas : C

Modul 5

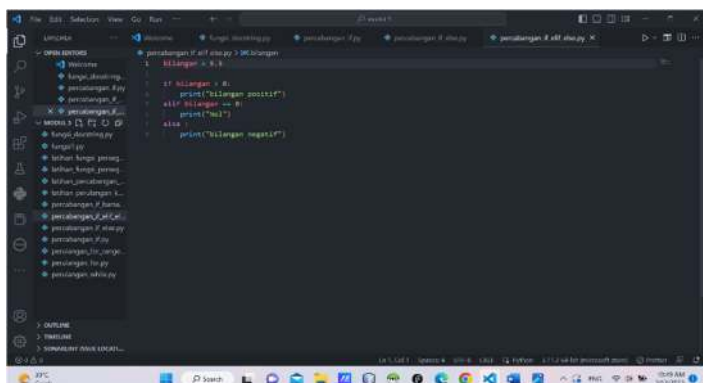
Percabangan **if** di gunakan untuk menguji satu buah kondisi contohnya seperti dibawah ini



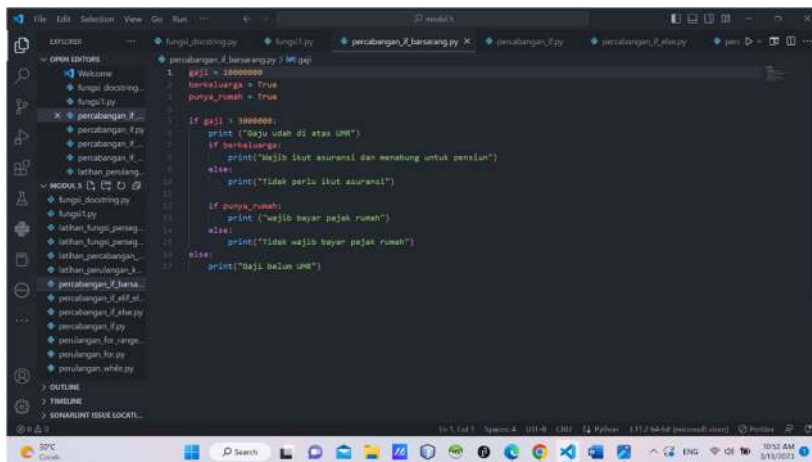
Pernyataan **If else** digunakan untuk menguji dua kondisi contohnya seperti di bawah ini



Pernyataan **if _elif_ else** digunakan untuk menguji lebih dari 2 kondisi contohnya seperti dibawah ini

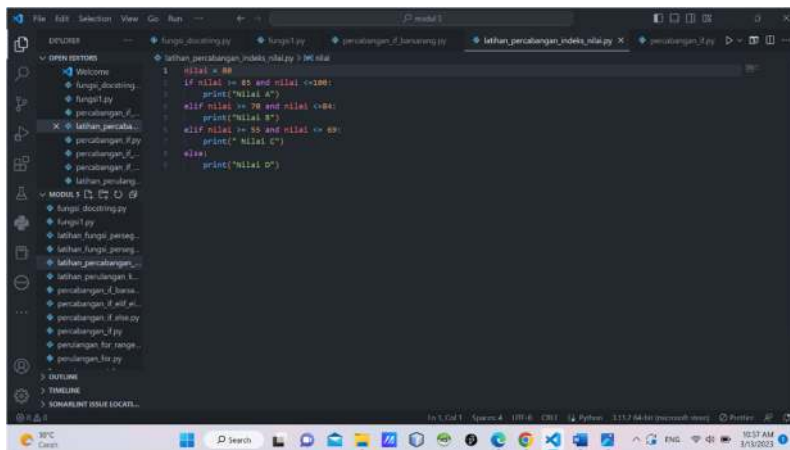


If bersarang digunakan apabila sebuah kondisi kondisional dapat disimpan di dalam **if** lain contohnya seperti dibawah ini



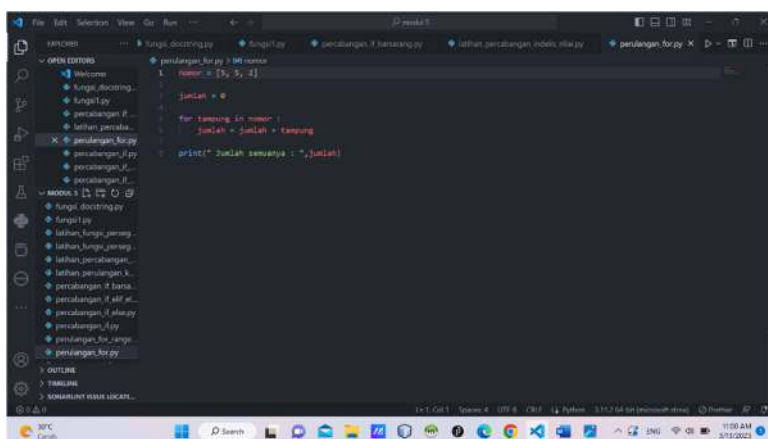
```
1 gaji = 10000000
2 berkesiua = True
3 punya_rumah = True
4
5 if gaji > 10000000:
6     print("Gaji udah di atas UMR")
7     if berkesiua:
8         print("Wajib ikut asuransi dan menabung untuk pensiun")
9     else:
10        print("Tidak perlu ikut asuransi")
11
12 if punya_rumah:
13     print("Wajib bayar pajak rumah")
14 else:
15     print("Tidak wajib bayar pajak rumah")
16
17 else:
18     print("Gaji belum UMR")
```

Percabangan indeks nilai statis contohnya seperti dibawah ini



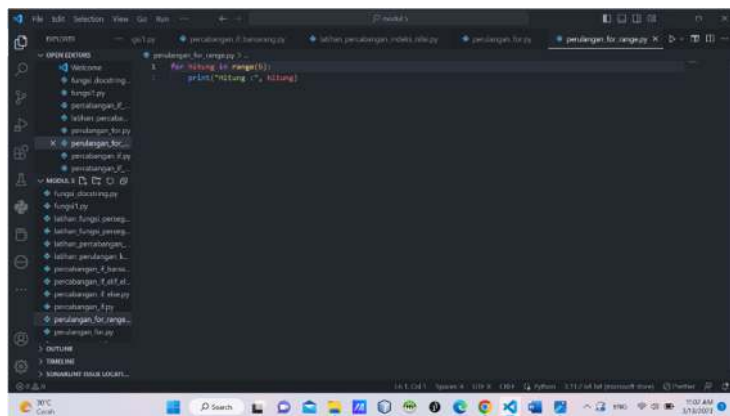
```
1 nilai = 80
2
3 if nilai >= 85 and nilai <= 100:
4     print("Nilai A")
5 elif nilai >= 70 and nilai <= 84:
6     print("Nilai B")
7 elif nilai >= 50 and nilai <= 69:
8     print("Nilai C")
9 else:
10    print("Nilai D")
```

Perulangan **for** digunakan untuk mengeksekusi baris perbaris



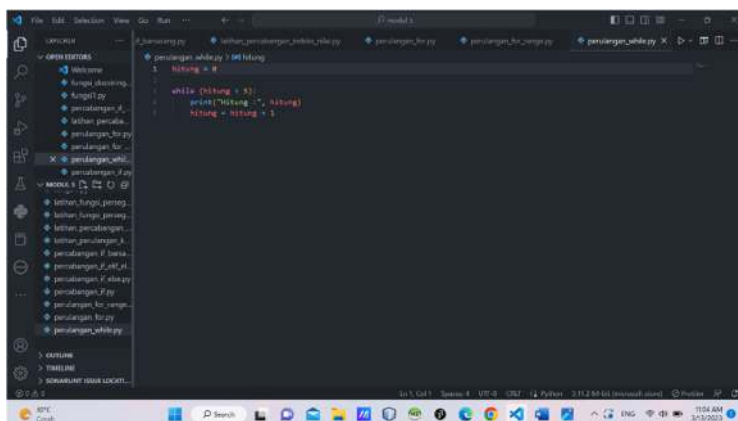
```
1 nomor = [5, 5, 2]
2 jumlah = 0
3
4 for tampung in nomor:
5     jumlah = jumlah + tampung
6
7 print("Jumlah semuanya : ", jumlah)
```

Perulangan **for range** digunakan untuk mendapat deret bilangan



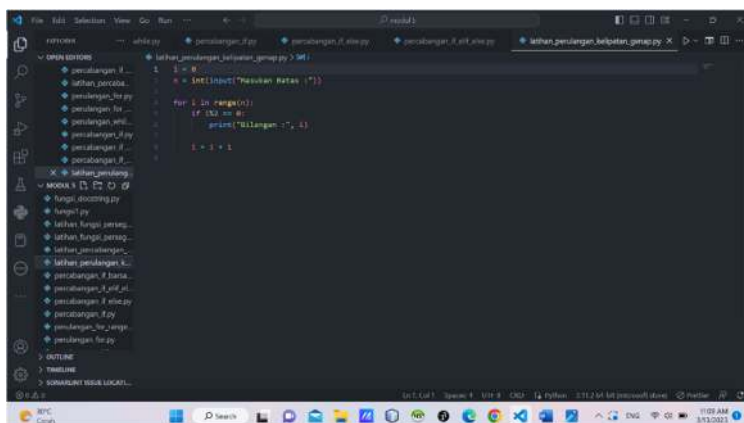
```
1 for i in range(5):
2     print("hitung :", i)
```

Perulangan **while** digunakan untuk menjalankan blok pernyataan terus menerus sampai menghasilkan nilai yang benar



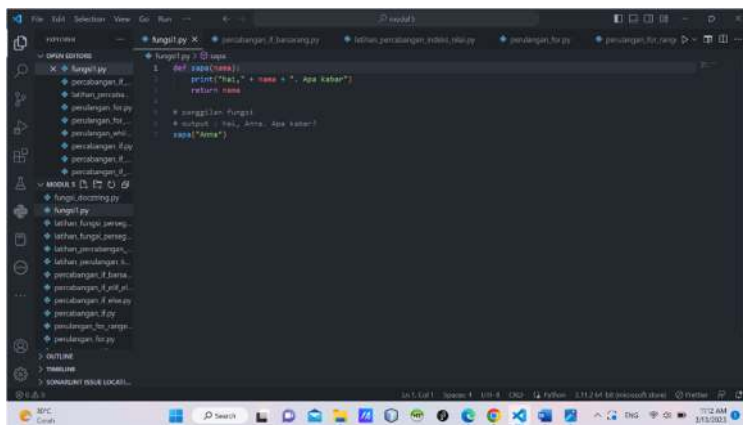
```
1 hitung = 1
2 while (hitung < 5):
3     print("hitung :", hitung)
4     hitung = hitung + 1
```

Contoh Latihan perulangan kelipatan dibawah ini



```
1 i = 5
2 n = int(input("Masukan Nanti :"))
3 for i in range(5):
4     if (i%5 == 0):
5         print("bilangan :", i)
6     i = i + 1
```

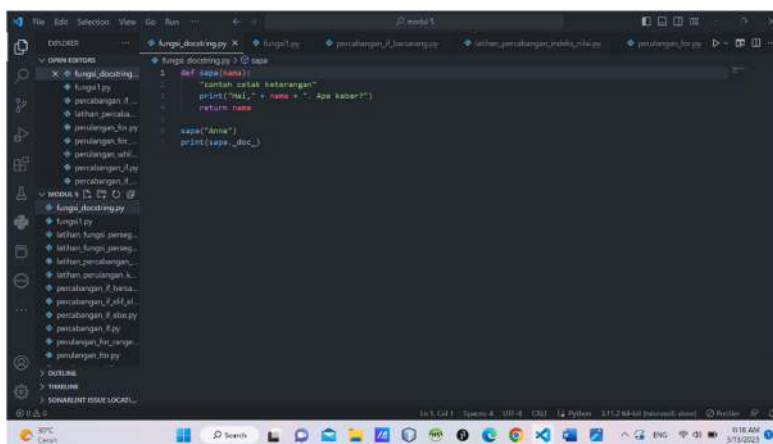
Fungsi digunakan untuk membuat kode program



```
def sapa(name):  
    print("Hai, " + name + ", Apa kabar?")  
    return name  
  
sapa("Anna")
```

The screenshot shows a Python IDE with a file explorer on the left containing various files like 'fungsi.py', 'fungsi_docstring.py', and 'fungsi_return.py'. The main editor window displays the code for a function named 'sapa' which takes a 'name' parameter, prints a greeting, and returns the name. The function is then called with the argument 'Anna'.

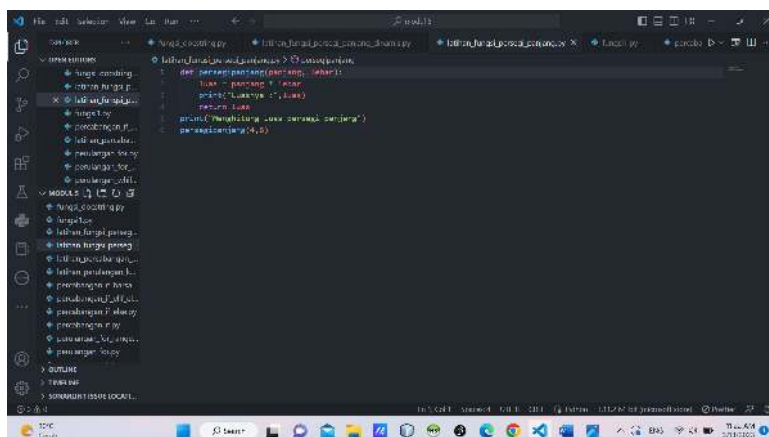
Docstring digunakan dokumentasi atau kerangka singkat fungsi yang di buat



```
def sapa(name):  
    """contoh cetak keterangan"""  
    print("Hai, " + name + ", Apa kabar?")  
    return name  
  
sapa("Anna")  
print(sapa.__doc__)
```

The screenshot shows a Python IDE with a file explorer on the left. The main editor window displays the code for a function named 'sapa' which takes a 'name' parameter, prints a greeting, and returns the name. A docstring is added to the function: """contoh cetak keterangan""". The function is then called with the argument 'Anna', and the docstring is printed using 'print(sapa.__doc__)'.

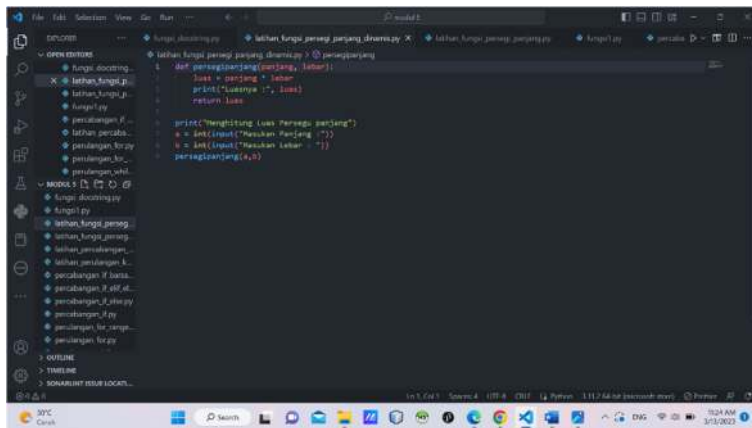
Contoh program luas persegi panjang dengan fungsi



```
def luas_persegi_panjang(p, l):  
    """luas persegi panjang"""  
    return p * l  
  
luas_persegi_panjang(4,5)
```

The screenshot shows a Python IDE with a file explorer on the left. The main editor window displays the code for a function named 'luas_persegi_panjang' which takes two parameters 'p' and 'l', calculates the area (p * l), and returns the result. A docstring is added to the function: """luas persegi panjang""". The function is then called with the arguments 4 and 5.

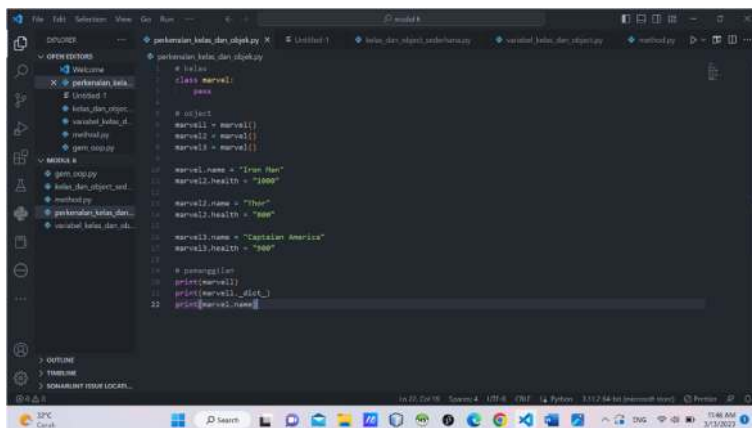
Contoh program luas persegi Panjang dengan menggunakan fungsi dinamis



```
def persegipanjang(panjang, lebar):  
    luas = panjang * lebar  
    print("Luas persegipanjang: ", luas)  
    return luas  
  
print("Menghitung Luas Persegi panjang")  
p = int(input("Masukan Panjang : "))  
l = int(input("Masukan Lebar : "))  
persegipanjang(p, l)
```

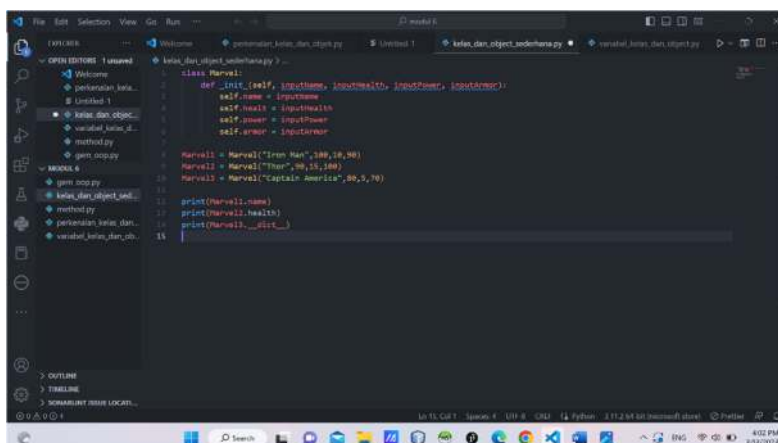
Modul 6

Perkenalan kelas dan object



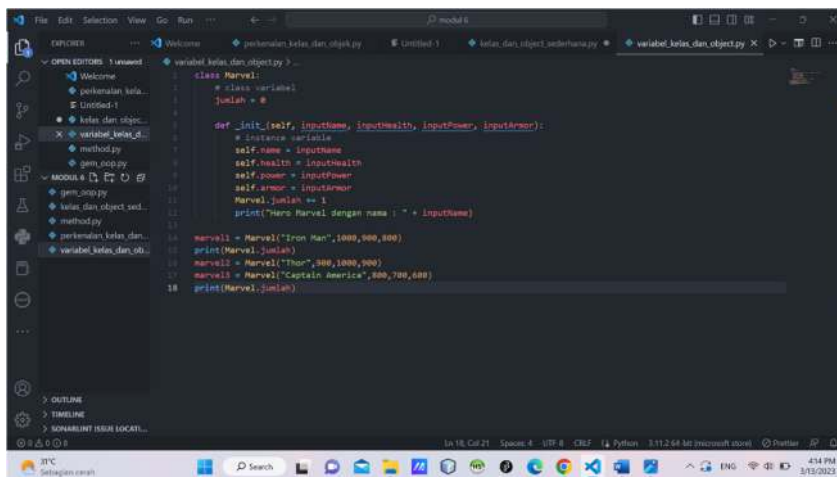
```
class Marvel:  
    name  
    health  
  
# object  
marvel1 = Marvel()  
marvel2 = Marvel()  
marvel3 = Marvel()  
  
marvel1.name = "Iron Man"  
marvel1.health = "1000"  
  
marvel2.name = "Thor"  
marvel2.health = "900"  
  
marvel3.name = "Captain America"  
marvel3.health = "800"  
  
# pengujian  
print(marvel1)  
print(marvel1.__dict__)  
print(marvel2.name)
```

Kelas dan objek sederhana



```
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, 50)  
Marvel2 = Marvel("Thor", 90, 15, 100)  
Marvel3 = Marvel("Captain America", 80, 5, 70)  
  
print(Marvel1.name)  
print(Marvel2.health)  
print(Marvel1.__dict__)  
11
```

Variabel kelas dan object

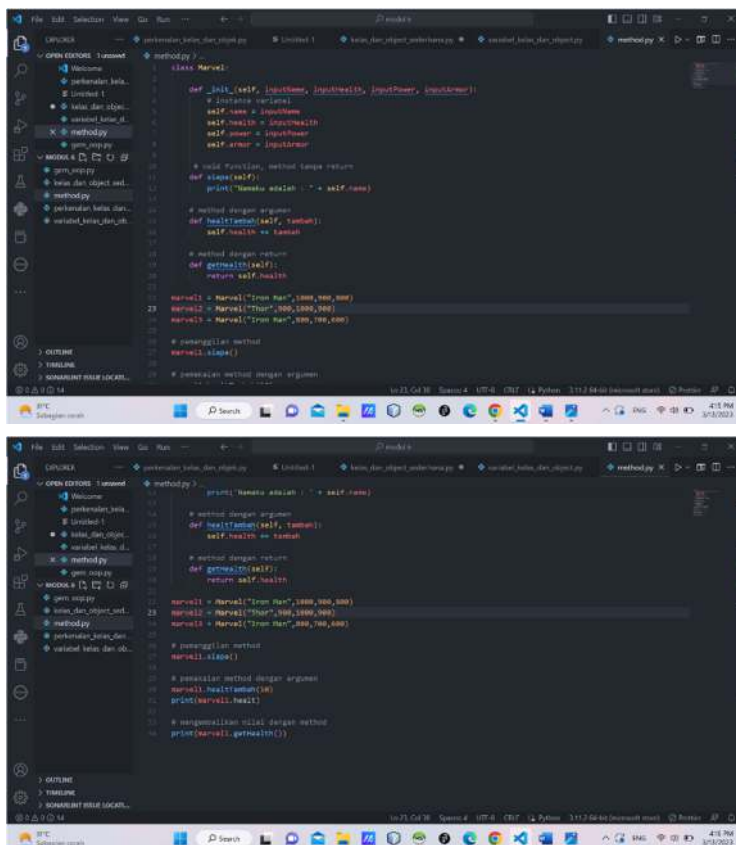


```
class Marvel:
    # class variabel
    jumlah = 0

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

marvel1 = Marvel("Iron Man", 1000, 900, 1000)
print(Marvel.jumlah)
marvel2 = Marvel("Thor", 300, 1000, 900)
marvel3 = Marvel("Captain America", 800, 700, 600)
print(Marvel.jumlah)
```

Method



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

    # method function, method tanpa return
    def siapa(self):
        print(f"Nama adalah : {self.name}")

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

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

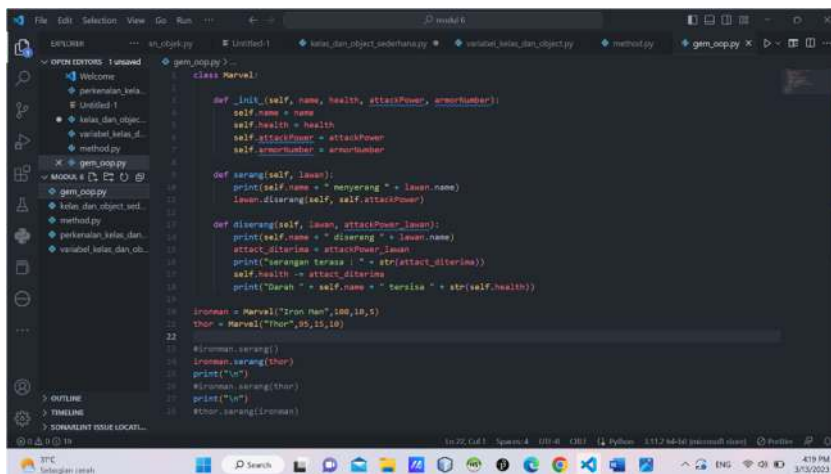
marvel1 = Marvel("Iron Man", 1000, 900, 1000)
marvel2 = Marvel("Thor", 300, 1000, 900)
marvel3 = Marvel("Iron Man", 800, 700, 600)

# memanggil method
marvel1.siapa()

# parameter method dengan argumen
marvel1.healthTambah(50)
print(marvel1.health)

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

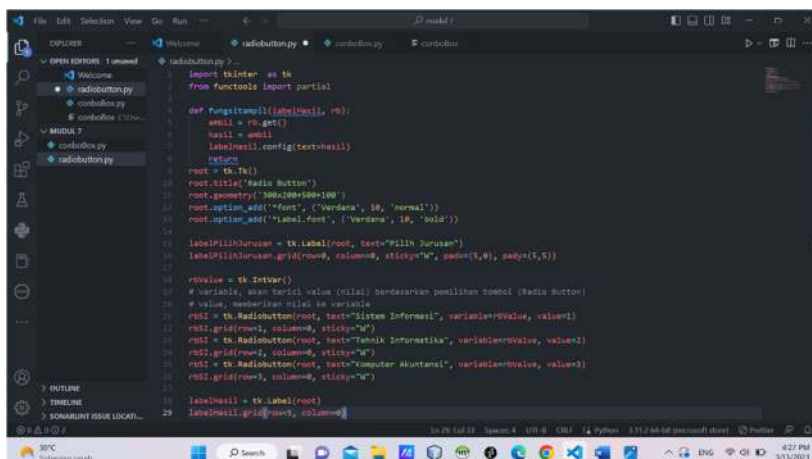

Game dan OOP



```
1 class Marvel:
2     def __init__(self, name, health, attackPower, armorNumber):
3         self.name = name
4         self.health = health
5         self.attackPower = attackPower
6         self.armorNumber = armorNumber
7
8     def serang(self, lawan):
9         print(self.name + " menyerang " + lawan.name)
10        lawan.diserang(self, self.attackPower)
11
12    def diserang(self, lawan, attackPower_lawan):
13        print(self.name + " diserang " + lawan.name)
14        attack_diterima = attackPower_lawan
15        print("serangan terasa : " + str(attack_diterima))
16        self.health -= attack_diterima
17        print("Darah " + self.name + " tersisa " + str(self.health))
18
19    Ironman = Marvel("Iron Man", 100, 10, 5)
20    Thor = Marvel("Thor", 95, 15, 10)
21
22    Ironman.serang()
23    Ironman.serang(Thor)
24    print("\n")
25    Ironman.serang(Thor)
26    print("\n")
27    Thor.serang(Ironman)
```

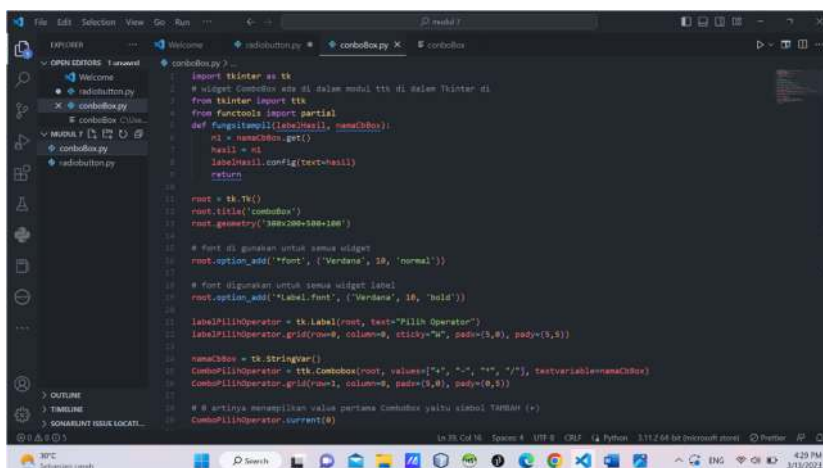
Modul 7

Radiobutton Dengan root pada kode di atas mengacu pada window utama (toplevel).



```
1 import tkinter as tk
2 from functools import partial
3
4 def fungsi_tampil(label_hasil, rb):
5     hasil = rb.get()
6     label_hasil.config(text=hasil)
7
8 root = tk.Tk()
9 root.title("Radio Button")
10 root.geometry("300x200+100+100")
11 root.option_add('*font', ('Verdana', 10, 'normal'))
12 root.option_add('*label.font', ('Verdana', 10, 'bold'))
13
14 label_pilih_jurusan = tk.Label(root, text="Pilih Jurusan")
15 label_pilih_jurusan.grid(row=0, column=0, sticky="w", padx=5, pady=5)
16
17 rb_value = tk.IntVar()
18 # variabels akan berisi value (nilai) berdasarkan pemilihan tombol (Radio Button)
19 # value, memberikan nilai ke variabel
20 rb1 = tk.Radiobutton(root, text="Sistem Informasi", variable=rb_value, value=1)
21 rb1.grid(row=1, column=0, sticky="w")
22 rb2 = tk.Radiobutton(root, text="Teknik Informatika", variable=rb_value, value=2)
23 rb2.grid(row=2, column=0, sticky="w")
24 rb3 = tk.Radiobutton(root, text="Komputer Akuntansi", variable=rb_value, value=3)
25 rb3.grid(row=3, column=0, sticky="w")
26
27 label_hasil = tk.Label(root)
28 label_hasil.grid(row=4, column=0)
```

conboBox



```
1 import tkinter as tk
2 # widget Combobox ada di dalam modul ttk di dalam Tkinter di
3 from tkinter import ttk
4 from functools import partial
5
6 def fungsi_tampil(label_hasil, namaCombobox):
7     ni = namaCombobox.get()
8     hasil = ni
9     label_hasil.config(text=hasil)
10
11 root = tk.Tk()
12 root.title("Combobox")
13 root.geometry("300x200+100+100")
14
15 # font di gunakan untuk semua widget
16 root.option_add('*font', ('Verdana', 10, 'normal'))
17
18 # font digunakan untuk semua widget label
19 root.option_add('*label.font', ('Verdana', 10, 'bold'))
20
21 label_pilih_operator = tk.Label(root, text="Pilih Operator")
22 label_pilih_operator.grid(row=0, column=0, sticky="w", padx=5, pady=5)
23
24 namaCombobox = tk.StringVar()
25 Combobox_pilih_operator = ttk.Combobox(root, values=["+", "-", "*", "/"], textvariable=namaCombobox)
26 Combobox_pilih_operator.grid(row=1, column=0, padx=5, pady=5)
27
28 # # artinya menampilkan value pertama Combobox yaitu simbol TAMBAR (+)
29 Combobox_pilih_operator.current(0)
```

