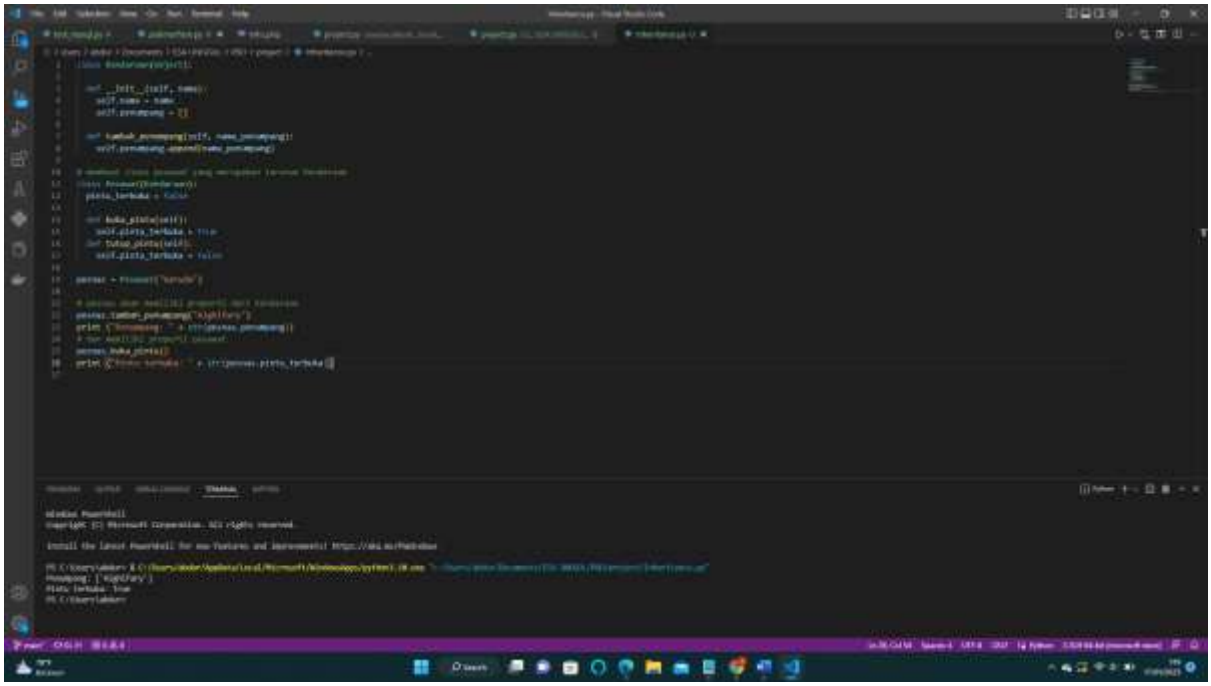
UAS PBO

Arrum Ligar
20220801389

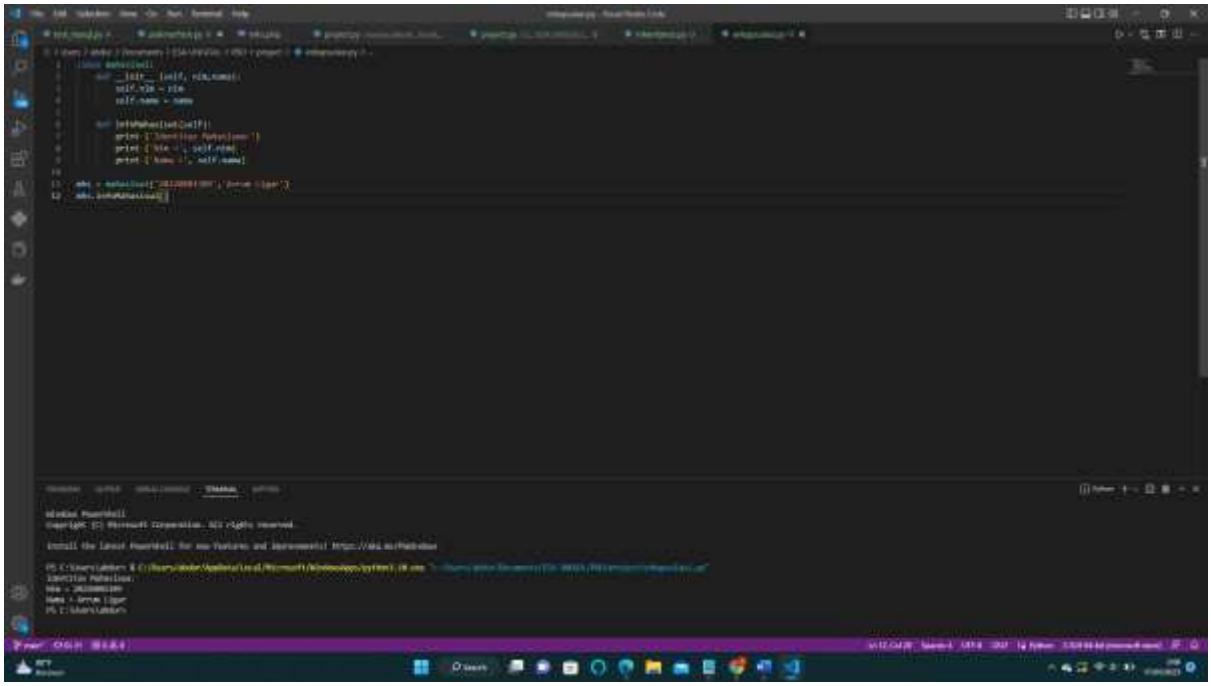
1. Inheritance (warisan atau turunan) adalah sebuah konsep yang penting di dalam Python. Inheritance adalah sebuah proses dimana sebuah class mengambil semua properti dan semua metode dari kelas lain.

```
class Kendaraan(object):

  def __init__(self, nama):
    self.nama = nama
    self.penumpang = []

  def tambah_penumpang(self, nama_penumpang):
    self.penumpang.append(nama_penumpang)

# membuat class pesawat yang merupakan turunan Kendaraan
class Pesawat(Kendaraan):
  pintu_terbuka = False

  def buka_pintu(self):
    self.pintu_terbuka = True
  def tutup_pintu(self):
    self.pintu_terbuka = False

pesnas = Pesawat("Garuda")

# pesnas akan memiliki properti dari Kendaraan
pesnas.tambah_penumpang("Alghifary")
print ("Penumpang: " + str(pesnas.penumpang))
# dan memiliki properti pesawat
pesnas.buka_pintu()
print ("Pintu terbuka: " + str(pesnas.pintu_terbuka))
```



2. Enkapsulasi (Encapsulation) adalah sebuah teknik dalam OOP yang mengizinkan kita untuk menyembunyikan detail dari sebuah atribut dalam sebuah class. Enkapsulasi juga bisa disebut sebagai teknik membatasi akses pada method yang dimiliki sebuah object dari luar object tersebut.

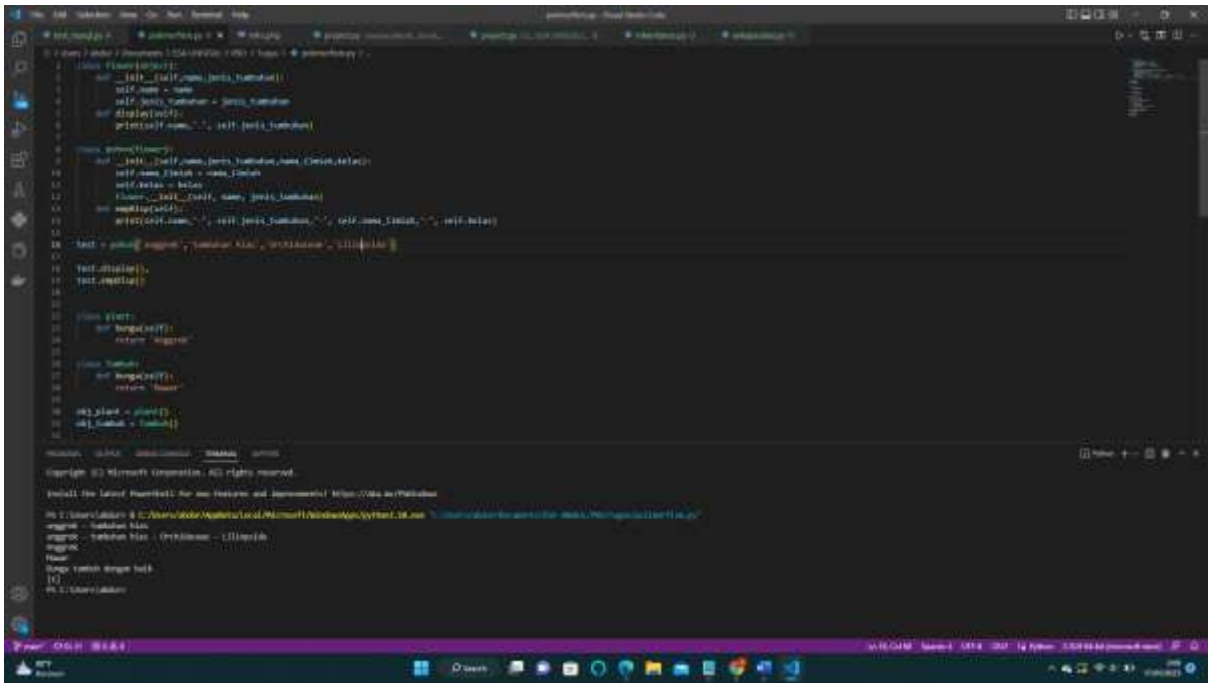
```
class mahasiswi:
    def __init__ (self, nim,nama):
        self.nim = nim
        self.nama = nama

    def infoMahasiswi(self):
        print ('Identitas Mahasiswa:')
        print ('Nim =', self.nim)
        print ('Nama =', self.nama)

mhs = mahasiswi('20220801389','Arrum Ligar')
mhs.infoMahasiswi()
```



3. Polymorphism adalah kemampuan untuk mengambil bentuk yang berbeda. Polymorphism dalam Python memungkinkan untuk mendefinisikan metode pada child class dengan menggunakan nama yang sama seperti pada parent class.

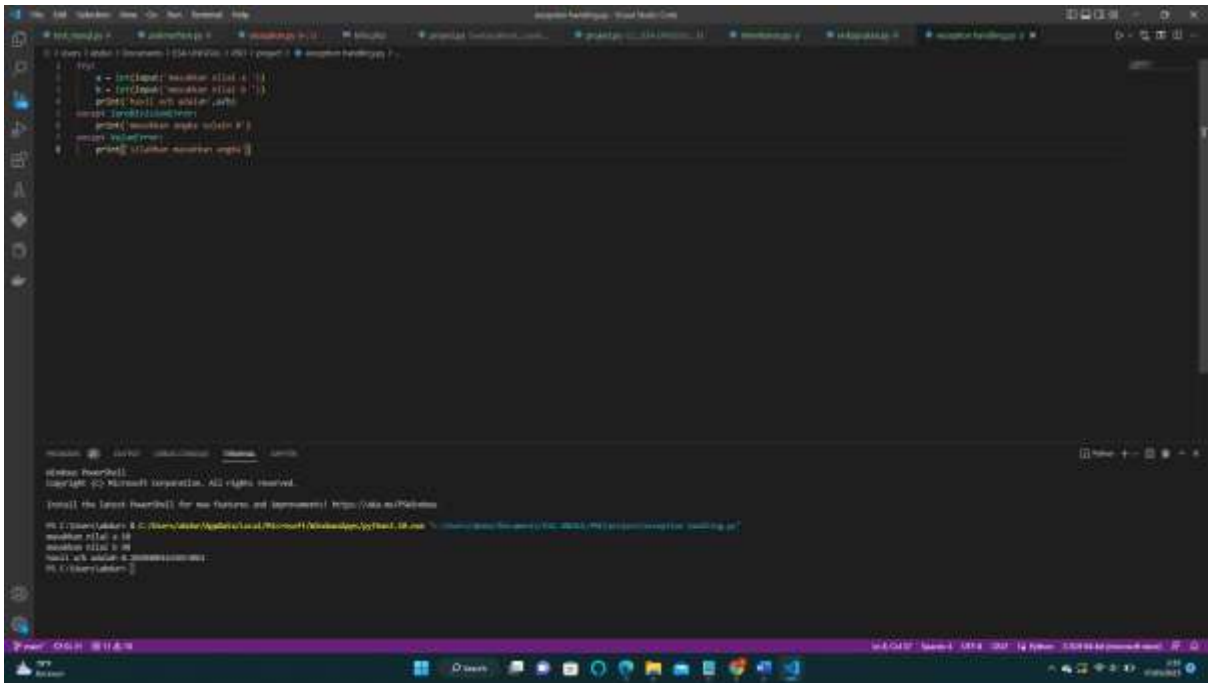
```
class flower(object):
    def __init__(self,name,jenis_tumbuhan):
        self.name = name
        self.jenis_tumbuhan = jenis_tumbuhan
    def display(self):
        print(self.name,"-", self.jenis_tumbuhan)

class pohon(flower):
    def __init__(self,name,jenis_tumbuhan,nama_ilmiah,kelas):
        self.nama_ilmiah = nama_ilmiah
        self.kelas = kelas
        flower.__init__(self, name, jenis_tumbuhan)
    def empDisp(self):
        print(self.name,"-", self.jenis_tumbuhan,"-", self.nama_ilmiah,"-",
self.kelas)

test = pohon('anggrek','tumbuhan hias','Orchidaceae','Liliopsida')

test.display(),
test.empDisp()


class plant:
    def bunga(self):
        return 'Anggrek'
```

```python
class Tumbuh:
    def bunga(self):
        return 'Mawar'

obj_plant = plant()
obj_tumbuh = Tumbuh()

for obj in [obj_plant,obj_tumbuh]:
    print(obj.bunga())

class tumbuhan(list):
    def append(self):
        return 'Bunga tumbuh dengan baik'

tumbuh_bunga = tumbuhan()
print(tumbuh_bunga.append())

list1 = list()
list1.append(1)
print(list1)
```



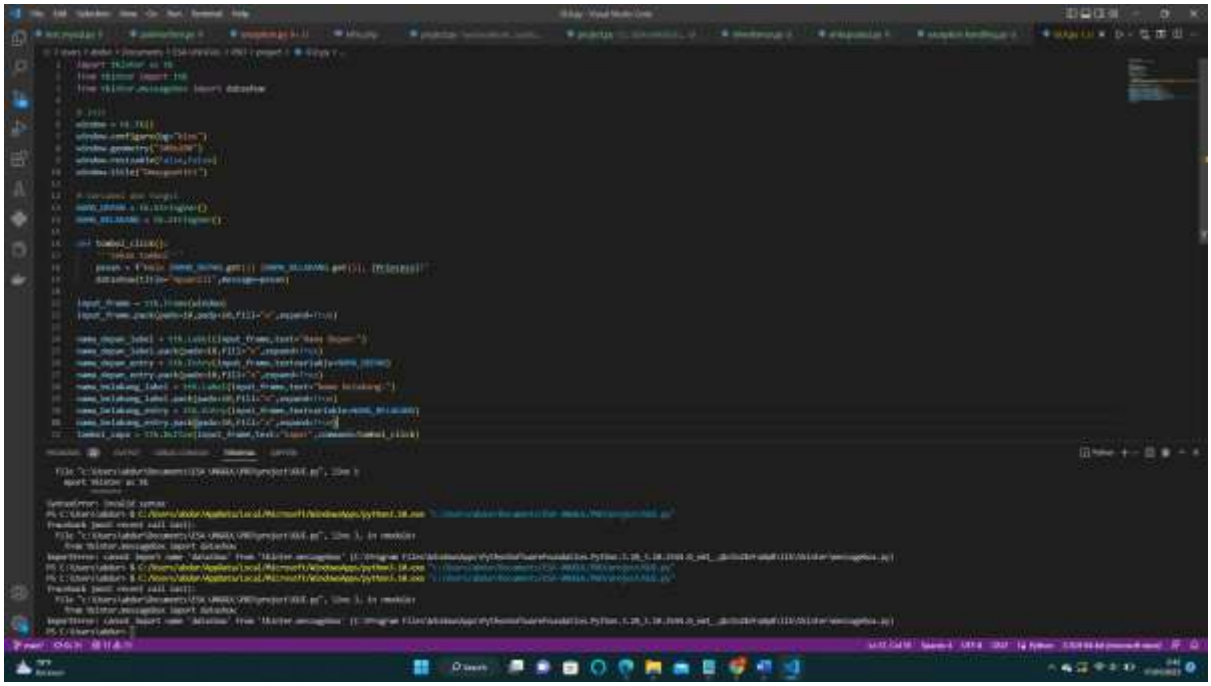
4. Python Exception Handling adalah salah satu Python Exception yang banyak digunakan. Exception handling merupakan Teknik penanganan kasus khusus. *Exception handling* pada kode program diperlukan saat kita perlu menyiasati adanya perintah yang gagal dieksekusi karena adanya kondisi yang tidak sesuai dengan kondisi yang diharapkan pada kasus normal.

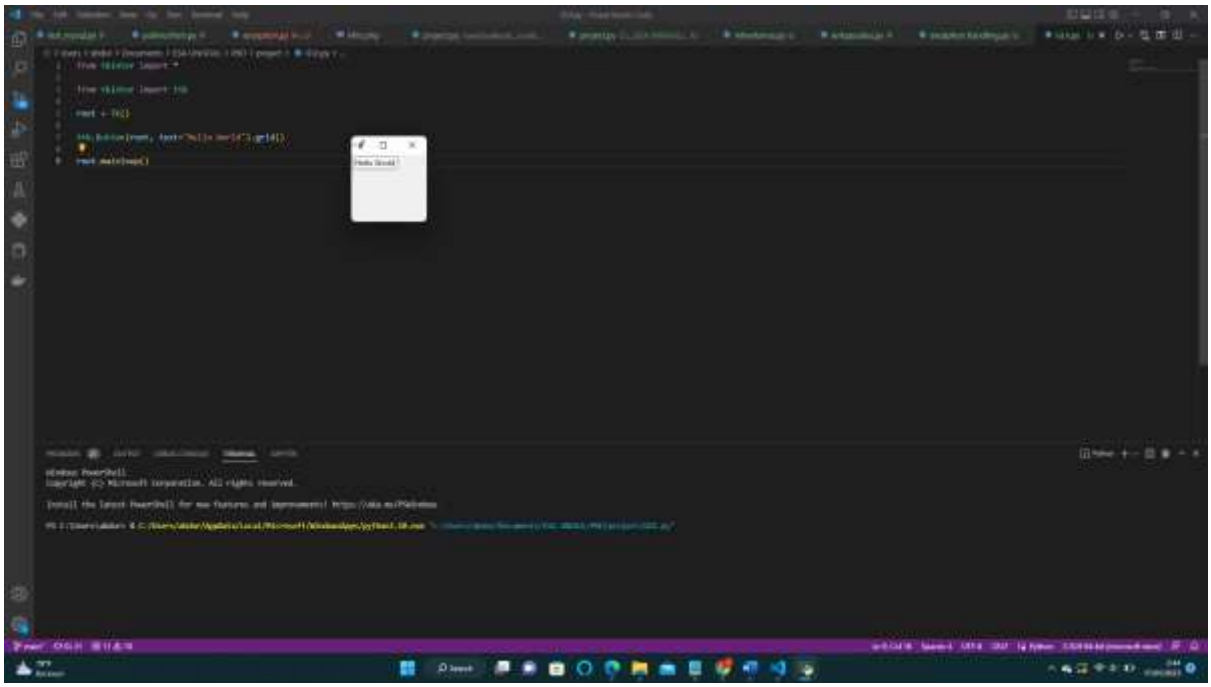
```
try:
    a = int(input('masukkan nilai a '))
    b = int(input('masukkan nilai b '))
    print('hasil a/b adalah',a/b)
except ZeroDivisionError:
    print('masukkan angka selain 0')
except ValueError:
    print('silahkan masukkan angka')
```



5. Graphical user interface (GUI) adalah sistem komponen visual interaktif untuk software komputer. GUI adalah suatu sistem yang membuat para pengguna atau user memapu berinteraksi dengan suatu perangkat komputer yang digunakan oleh si user tersebut. GUI sendiri dapat dikendalikan menggunakan beberapa macam alat input, seperti mouse, keyboard, touchscreen, dan lain sebagainya. Sistem Operasi GUI secara umum akan ada jendela, menu, tombol, ikon, dan lainnya yang didesain supaya penggunanya lebih mudah dalam berinteraksi dengan sistem operasi atau aplikasi. Untuk GUI disaya masih belum bisa keluar pak, hanya bisa menampilkan ini.

```python
import tkinter as tk
from tkinter import ttk
from tkinter.messagebox import datashow

# Init
window = tk.Tk()
window.configure(bg="blue")
window.geometry("300x200")
window.resizable(False,False)
window.title("Omaygaatttt")

# Variabel dan Fungsi
NAMA_DEPAN = tk.StringVar()
NAMA_BELAKANG = tk.StringVar()

def tombol_click():
    '''tekan tombol'''
    pesan = f"Halo {NAMA_DEPAN.get()} {NAMA_BELAKANG.get()}, {Princess}!"
    datashow(title="Apaaniii",message=pesan)

input_frame = ttk.Frame(window)
input_frame.pack(padx=10,pady=10,fill="x",expand=True)

nama_depan_label = ttk.Label(input_frame,text="Nama Depan:")
nama_depan_label.pack(padx=10,fill="x",expand=True)
nama_depan_entry = ttk.Entry(input_frame,textvariable=NAMA_DEPAN)
nama_depan_entry.pack(padx=10,fill="x",expand=True)
nama_belakang_label = ttk.Label(input_frame,text="Nama belakang:")
nama_belakang_label.pack(padx=10,fill="x",expand=True)
nama_belakang_entry = ttk.Entry(input_frame,textvariable=NAMA_BELAKANG)
nama_belakang_entry.pack(padx=10,fill="x",expand=True)
tombol_sapa = ttk.Button(input_frame,text="Sapa!",command=tombol_click)
tombol_sapa.pack(fill='x',expand=True,padx=10,pady=10)

window.mainloop()
```





6. Python menggunakan apa yang disebut *Python Database API* dengan tujuan untuk menjadi antarmuka dengan database. API ini mengijinkan kita untuk memprogram database management system (DBMS) yang berbeda.

```
import mysql.connector

mydb = mysql.connector.connect(
  host = "localhost",
    user = "root",
    password = "",
    database = "esa_unggul"
```

```
)

mycursor = mydb.cursor()

mycursor.execute("select * from mahasiswa")

myresult = mycursor.fetchall()

for x in myresult:
  print(x)
```

7. Saya mengambil dari membuat project plant.

```
import tkinter as tk
from tkinter import ttk

class Plant:
    def __init__(self):
        self.statusTumbuh = 0
        self.jumlahAir = 0
        self.jumlahPupuk = 0

    def beri_air(self):
        self.jumlahAir += 1
        self.cek_kondisi_tumbuh()

    def beri_pupuk(self):
        self.jumlahPupuk += 1
        self.cek_kondisi_tumbuh()

    def cek_kondisi_tumbuh(self):
        if self.jumlahAir >= 3 and self.jumlahPupuk >= 1:
            self.tumbuh()

    def tumbuh(self):
        if self.statusTumbuh < 4:
            self.jumlahAir -= 3
            self.jumlahPupuk -= 1
            self.statusTumbuh += 1

    def display_plant(self):
        print(self.get_status_tumbuh_text())
        print("Jumlah Air:" + str(self.jumlahAir))
        print("Jumlah Pupuk:" + str(self.jumlahPupuk))

    def get_status_tumbuh_text(self):
        if self.statusTumbuh == 0:
```

```
            return "Benih"
        elif self.statusTumbuh == 1:
            return "Tunas"
        elif self.statusTumbuh == 2:
            return "Tanaman Kecil"
        elif self.statusTumbuh == 3:
            return "Tanaman Dewasa"
        else:
            return "Berbunga"

    def get_status_tumbuh(self):
        return self.statusTumbuh

    def get_image_path(self):
        tImagePath = "img/seed.png"
        if self.statusTumbuh == 0:
            tImagePath = "img/seed.png"
        elif self.statusTumbuh == 1:
            tImagePath = "img/sprout.png"
        elif self.statusTumbuh == 2:
            tImagePath = "img/small.png"
        elif self.statusTumbuh == 3:
            tImagePath = "img/big.png"
        else:
            tImagePath = "img/blossom.png"
        return tImagePath

class UgardenMain (tk.Tk):
    def __init__(self):
        super().__init__()

        self.db = mysql.connector.connect(
            host='localhost',
            user = "root",
            password = "",
            database = "esa_unggul"
        )


class PlantMainSwing(tk.Tk):
    def __init__(self):
        super().__init__()

        self.plant = Plant()

        self.title("WELCOME TO UGARDEN")
        self.geometry("500x300")
        self.resizable(False, False)
```

```
        # Add label
        self.label = tk.Label(self, text="")
        self.label.pack()

        # Add buttons
        ttk.Button(self, text="Beri Air", command=self.give_water).pack()
        ttk.Button(self, text="Beri Pupuk",
command=self.give_fertilizer).pack()

        # Add widget
        self.text_field = tk.Text(self, width=20, height=1)
        self.text_field.pack()
        self.set_plant_image()

    def give_water(self):
        self.plant.beri_air()
        self.text_field.delete("1.0", tk.END)
        self.text_field.insert("1.0", self.plant.get_status_tumbuh_text())
        self.set_plant_image()

    def give_fertilizer(self):
        self.plant.beri_pupuk()
        self.text_field.delete("1.0", tk.END)
        self.text_field.insert("1.0", self.plant.get_status_tumbuh_text())
        self.set_plant_image()

    def set_plant_image(self):
        self.photo = tk.PhotoImage(file = self.plant.get_image_path())
        self.label.config(image=self.photo)

if __name__ == "__main__":
        app = PlantMainSwing()
        app.mainloop()
```