LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK LANJUT

2023

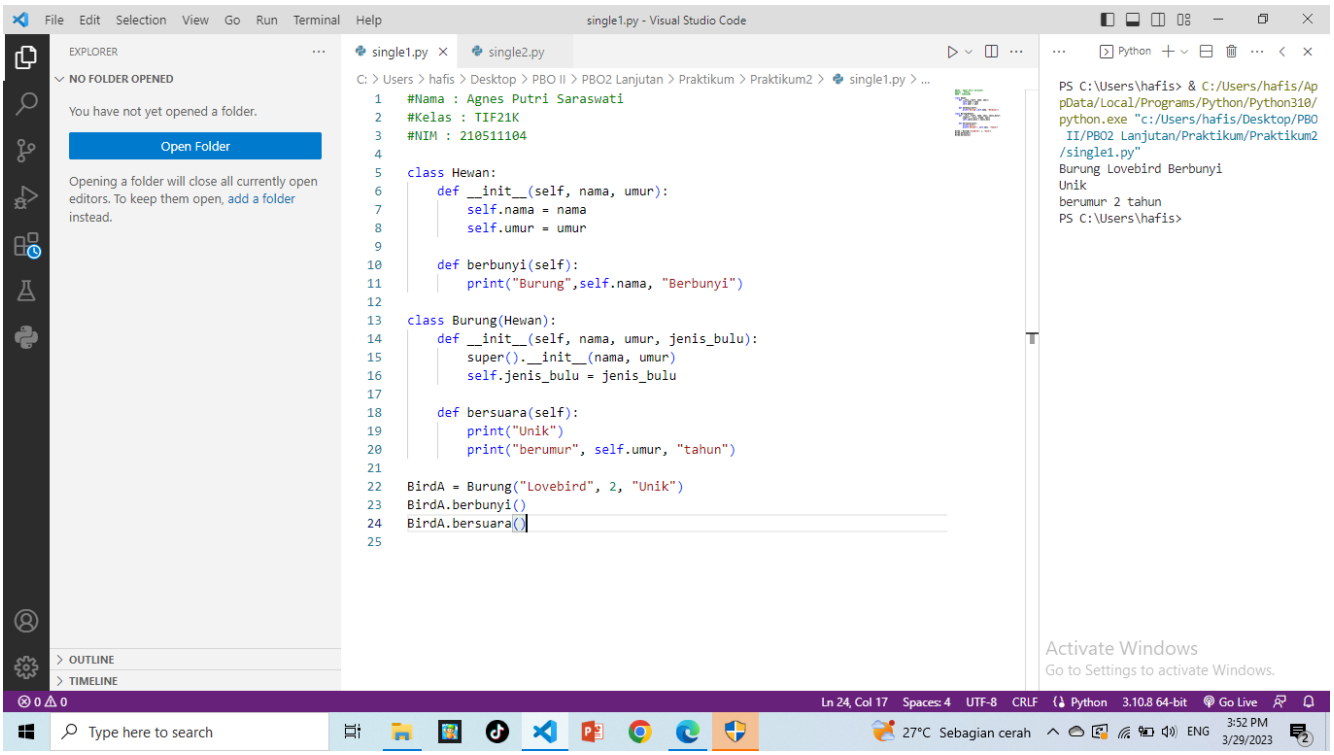
Prepared By:

Buatlah masing-masing 2 contoh jenis pewarisan di luar contoh yang telah diberikan, beri nama : single1.py, single2.py, multiple1.py, multiple2.py, hierarchical1.py, hierarchical2.py, multilevel1.py, multilevel2.py, hybrid1.py, hybrid2.py

Jawaban:

- SCRIPT SINGLE1:

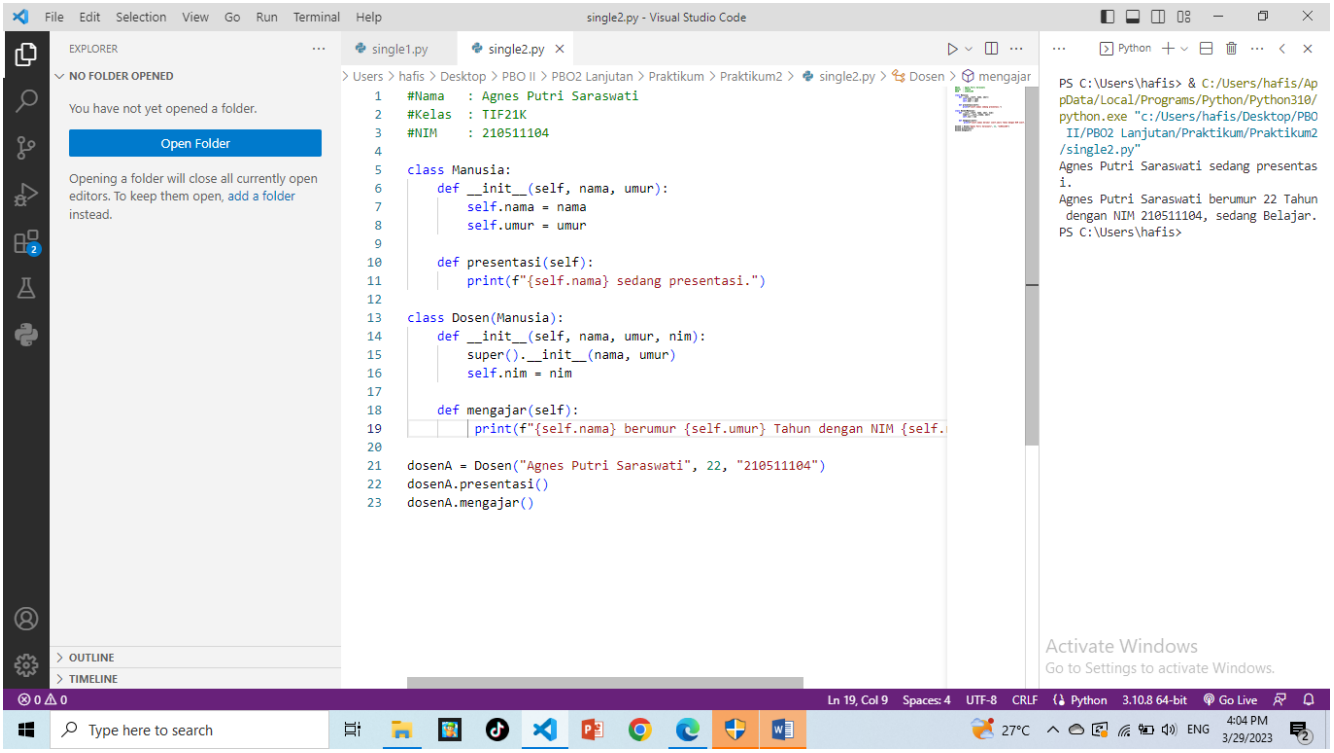
```
#Nama : Agnes Putri Saraswati
#Kelas : TIF21K
#NIM : 210511104

class Hewan:
    def __init__(self, nama, umur):
        self.nama = nama
        self.umur = umur

    def berbunyi(self):
        print("Burung",self.nama, "Berbunyi")

class Burung(Hewan):
    def __init__(self, nama, umur, jenis_bulu):
        super().__init__(nama, umur)
        self.jenis_bulu = jenis_bulu

    def bersuara(self):
        print("Unik")
        print("berumur", self.umur, "tahun")

BirdA = Burung("Lovebird", 2, "Unik")
BirdA.berbunyi()
BirdA.bersuara()
```



- SCRIPT SINGLE2:

```
#Nama   : Agnes Putri Saraswati
#Kelas  : TIF21K
#NIM    : 210511104

class Manusia:
    def __init__(self, nama, umur):
        self.nama = nama
        self.umur = umur

    def presentasi(self):
        print(f"{self.nama} sedang presentasi.")

class Dosen(Manusia):
    def __init__(self, nama, umur, nim):
        super().__init__(nama, umur)
        self.nim = nim
```

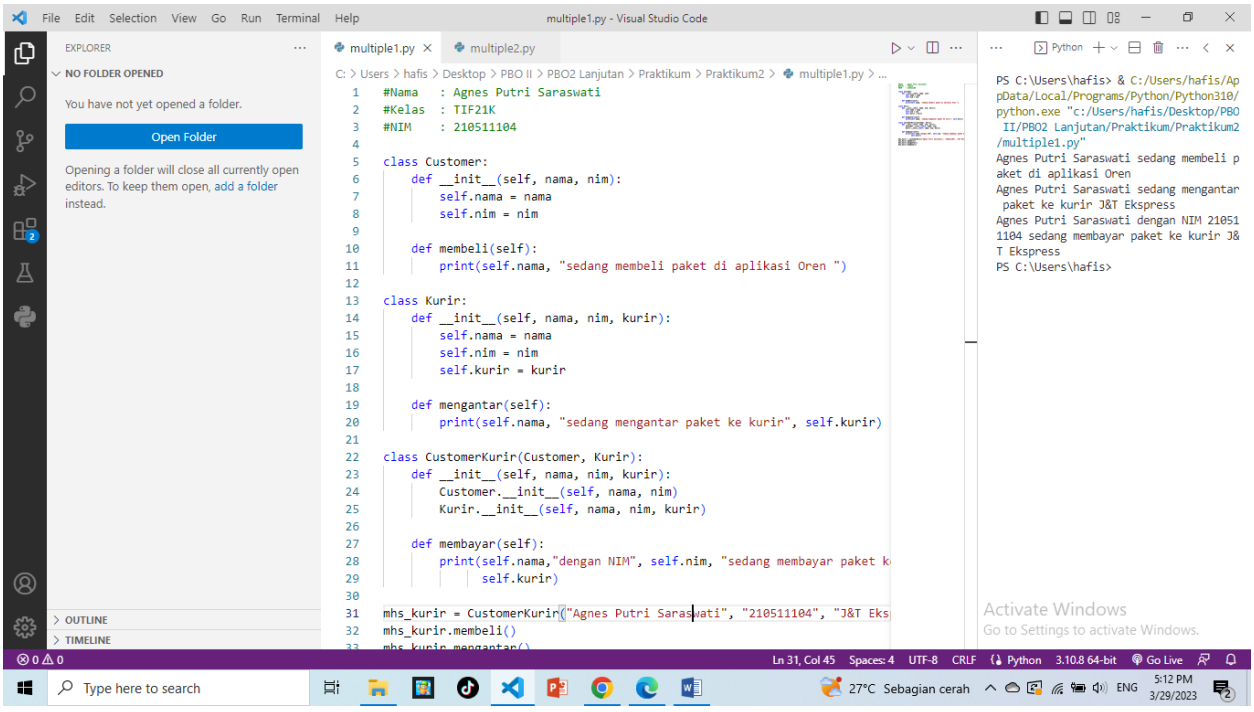
```
    def mengajar(self):
        print(f"{self.nama} berumur {self.umur} Tahun dengan NIM {self.nim},
sedang Belajar.")

dosenA = Dosen("Agnes Putri Saraswati", 22, "210511104")
dosenA.presentasi()
dosenA.mengajar()
```



-SCRIPT MULTIPLE1

```
#Nama    : Agnes Putri Saraswati
#Kelas   : TIF21K
#NIM     : 210511104

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

    def membeli(self):
```

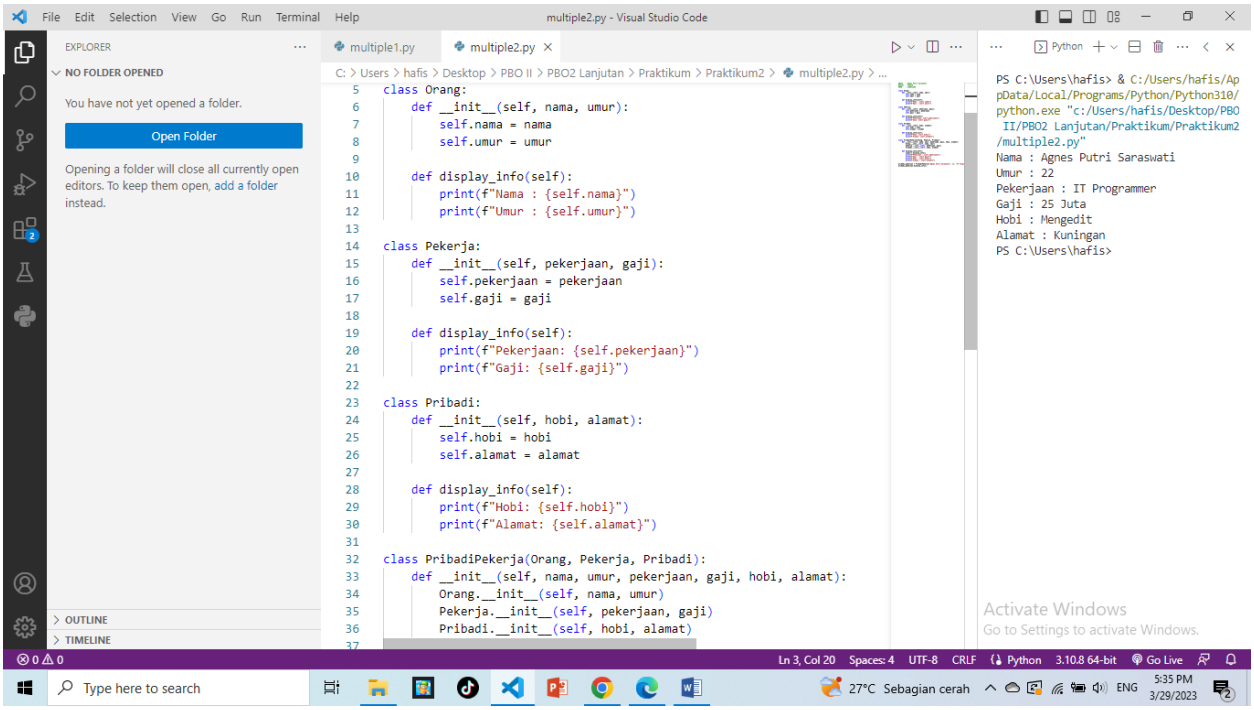
```
        print(self.nama, "sedang membeli paket di aplikasi Oren ")

class Kurir:
    def __init__(self, nama, nim, kurir):
        self.nama = nama
        self.nim = nim
        self.kurir = kurir

    def mengantar(self):
        print(self.nama, "sedang mengantar paket ke kurir", self.kurir)

class CustomerKurir(Customer, Kurir):
    def __init__(self, nama, nim, kurir):
        Customer.__init__(self, nama, nim)
        Kurir.__init__(self, nama, nim, kurir)

    def membayar(self):
        print(self.nama,"dengan NIM", self.nim, "sedang membayar paket ke kurir",
              self.kurir)

mhs_kurir = CustomerKurir("Agnes Putri Saraswati", "210511104", "J&T Ekspress")
mhs_kurir.membeli()
mhs_kurir.mengantar()
mhs_kurir.membayar()
```



- SCRIPT MULTIPLE2

```
#Nama   : Agnes Putri Saraswati
#Kelas  : TIF21K
#NIM    : 210511104

class Orang:
    def __init__(self, nama, umur):
        self.nama = nama
        self.umur = umur

    def display_info(self):
        print(f"Nama : {self.nama}")
        print(f"Umur : {self.umur}")

class Pekerja:
    def __init__(self, pekerjaan, gaji):
        self.pekerjaan = pekerjaan
        self.gaji = gaji

    def display_info(self):
        print(f"Pekerjaan: {self.pekerjaan}")
        print(f"Gaji: {self.gaji}")

class Pribadi:
    def __init__(self, hobi, alamat):
        self.hobi = hobi
        self.alamat = alamat

    def display_info(self):
        print(f"Hobi: {self.hobi}")
        print(f"Alamat: {self.alamat}")

class PribadiPekerja(Orang, Pekerja, Pribadi):
    def __init__(self, nama, umur, pekerjaan, gaji, hobi, alamat):
        Orang.__init__(self, nama, umur)
        Pekerja.__init__(self, pekerjaan, gaji)
        Pribadi.__init__(self, hobi, alamat)

    def display_info(self):
        super().display_info()
        print(f"Pekerjaan : {self.pekerjaan}")
        print(f"Gaji : {self.gaji}")
        print(f"Hobi : {self.hobi}")
```

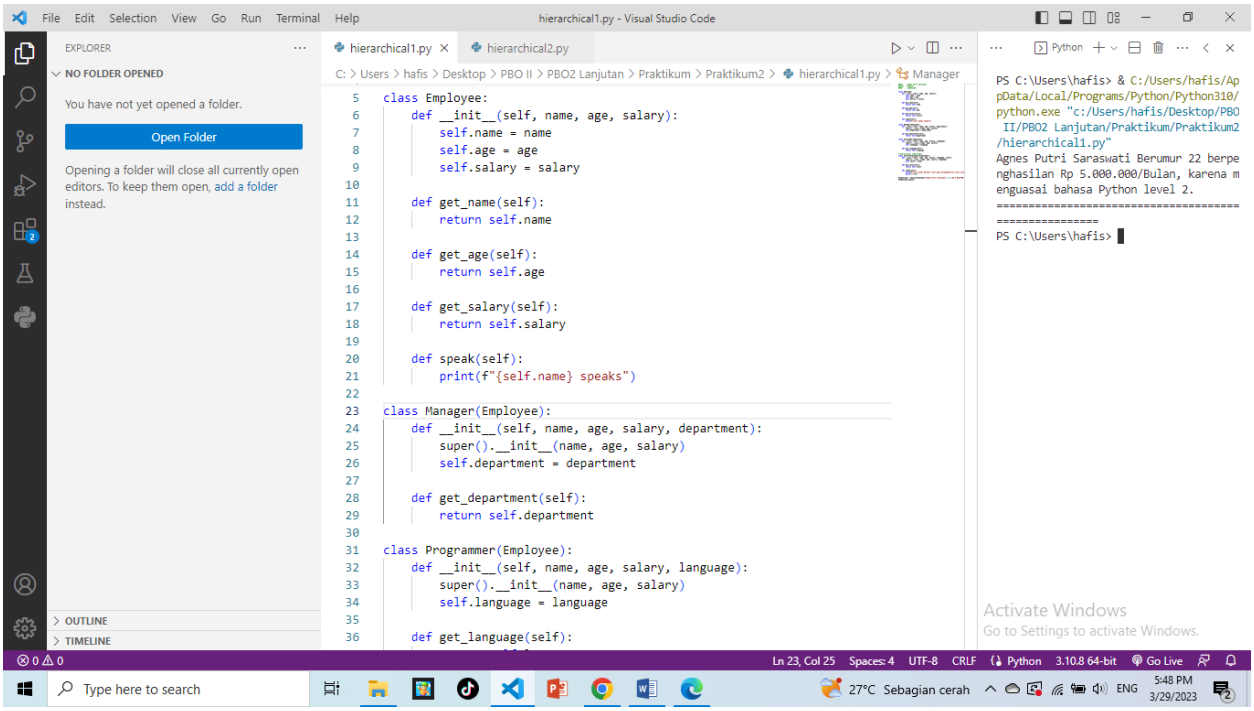
```
        print(f"Alamat : {self.alamat}")

pribadi_pekerjaC = PribadiPekerja("Agnes Putri Saraswati", 22, "IT Programmer",
"25 Juta", "Mengedit", "Kuningan")
pribadi_pekerjaC.display_info()
```



- SCRIPT HIERARCHICAL1

```
#Nama   : Agnes Putri Saraswati
#Kelas  : TIF21K
#NIM    : 210511104

class Employee:
    def __init__(self, name, age, salary):
        self.name = name
        self.age = age
        self.salary = salary

    def get_name(self):
        return self.name

    def get_age(self):
        return self.age
```

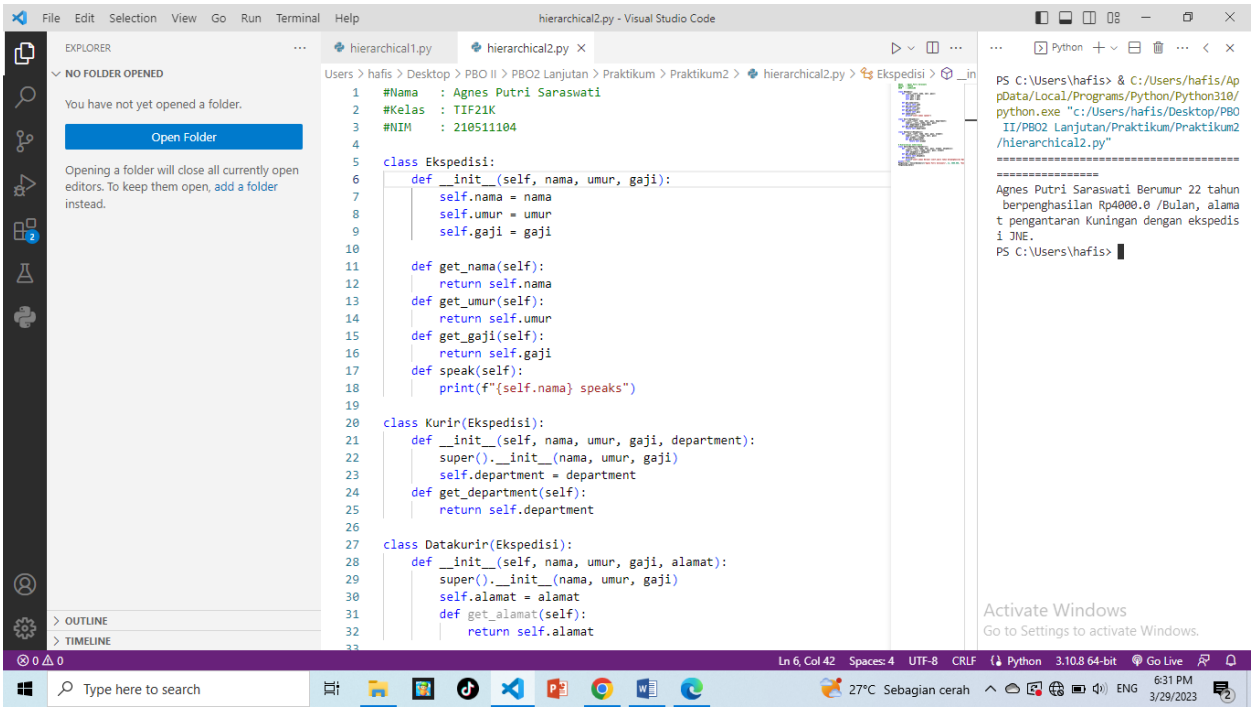
```
    def get_salary(self):
        return self.salary

    def speak(self):
        print(f"{self.name} speaks")

class Manager(Employee):
    def __init__(self, name, age, salary, department):
        super().__init__(name, age, salary)
        self.department = department

    def get_department(self):
        return self.department

class Programmer(Employee):
    def __init__(self, name, age, salary, language):
        super().__init__(name, age, salary)
        self.language = language

    def get_language(self):
        return self.language

# Hierarchical Inheritance
class SeniorProgrammer(Programmer):
    def __init__(self, name, age, salary, language, level):
        super().__init__(name, age, salary, language)
        self.level = level

    def get_level(self):
        return self.level

    def speak(self):
        print(f"{self.name} Berumur {self.age} berpenghasilan
{self.salary}/Bulan, karena menguasai bahasa {self.language} level
{self.level}.")
        print("="*54)

Pengunjung = SeniorProgrammer("Agnes Putri Saraswati", 22,"Rp 5.000.000",
"Python", 2)
Pengunjung.speak()
```



- SCRIPT HIERARCHICAL2

```
#Nama   : Agnes Putri Saraswati
#Kelas  : TIF21K
#NIM    : 210511104

class Ekspedisi:
    def __init__(self, nama, umur, gaji):
        self.nama = nama
        self.umur = umur
        self.gaji = gaji

    def get_nama(self):
        return self.nama
    def get_umur(self):
        return self.umur
    def get_gaji(self):
        return self.gaji
    def speak(self):
        print(f"{self.nama} speaks")

class Kurir(Ekspedisi):
    def __init__(self, nama, umur, gaji, department):
```

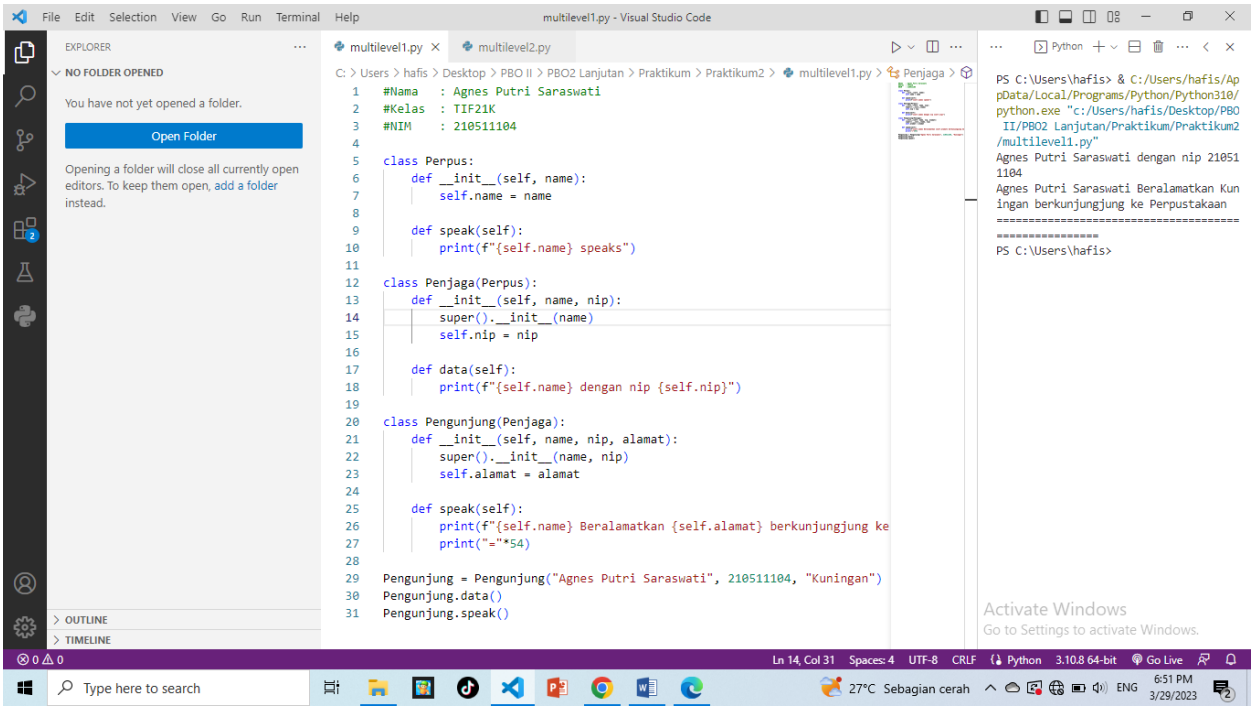
```
        super().__init__(nama, umur, gaji)
        self.department = department
    def get_department(self):
        return self.department

class Datakurir(Ekspedisi):
    def __init__(self, nama, umur, gaji, alamat):
        super().__init__(nama, umur, gaji)
        self.alamat = alamat
        def get_alamat(self):
            return self.alamat

# Hierarchical Inheritance
class SeniorDatakurir(Datakurir):
    def __init__(self, nama, umur, gaji, alamat, ekspedisi):
        super().__init__(nama, umur, gaji, alamat)
        self.ekspedisi = ekspedisi
    def get_ekspedisi(self):
        return self.ekspedisi
    def data(self):
        print(f"{self.nama} Berumur {self.umur} tahun berpenghasilan
Rp{self.gaji} /Bulan, alamat pengantaran {self.alamat} dengan ekspedisi
{self.ekspedisi}.")
print("="*54)
Pengunjung = SeniorDatakurir("Agnes Putri Saraswati", 22, 4000.000, "Kuningan",
"JNE")
Pengunjung.data()
```



- SCRIPT MULTILEVEL1

```
#Nama    : Agnes Putri Saraswati
#Kelas   : TIF21K
#NIM     : 210511104

class Perpus:
    def __init__(self, name):
        self.name = name

    def speak(self):
        print(f"{self.name} speaks")

class Penjaga(Perpus):
    def __init__(self, name, nip):
        super().__init__(name)
        self.nip = nip

    def data(self):
        print(f"{self.name} dengan nip {self.nip}")

class Pengunjung(Penjaga):
    def __init__(self, name, nip, alamat):
```

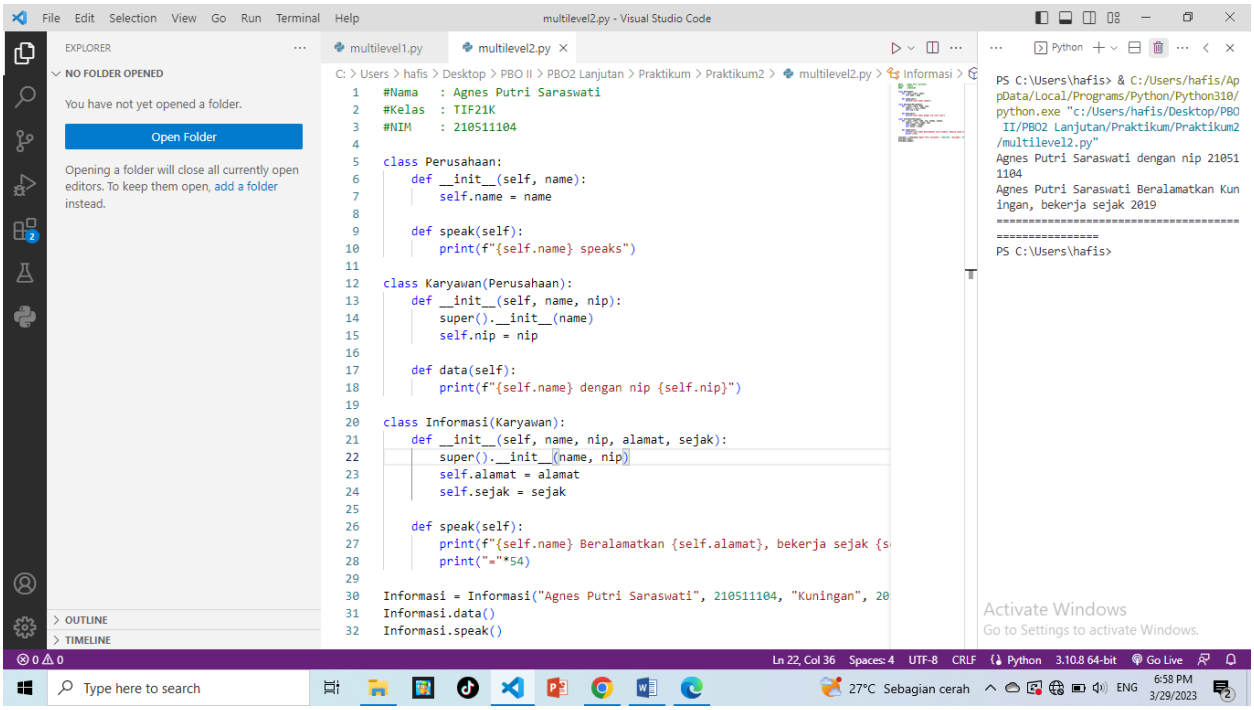
```
        super().__init__(name, nip)
        self.alamat = alamat

    def speak(self):
        print(f"{self.name} Beralamatkan {self.alamat} berkunjungjung ke
Perpustakaan")
        print("="*54)

Pengunjung = Pengunjung("Agnes Putri Saraswati", 210511104, "Kuningan")
Pengunjung.data()
Pengunjung.speak()
```



- SCRIPT MULTILEVEL2

```
#Nama    : Agnes Putri Saraswati
#Kelas   : TIF21K
#NIM     : 210511104

class Perusahaan:
    def __init__(self, name):
        self.name = name

    def speak(self):
```

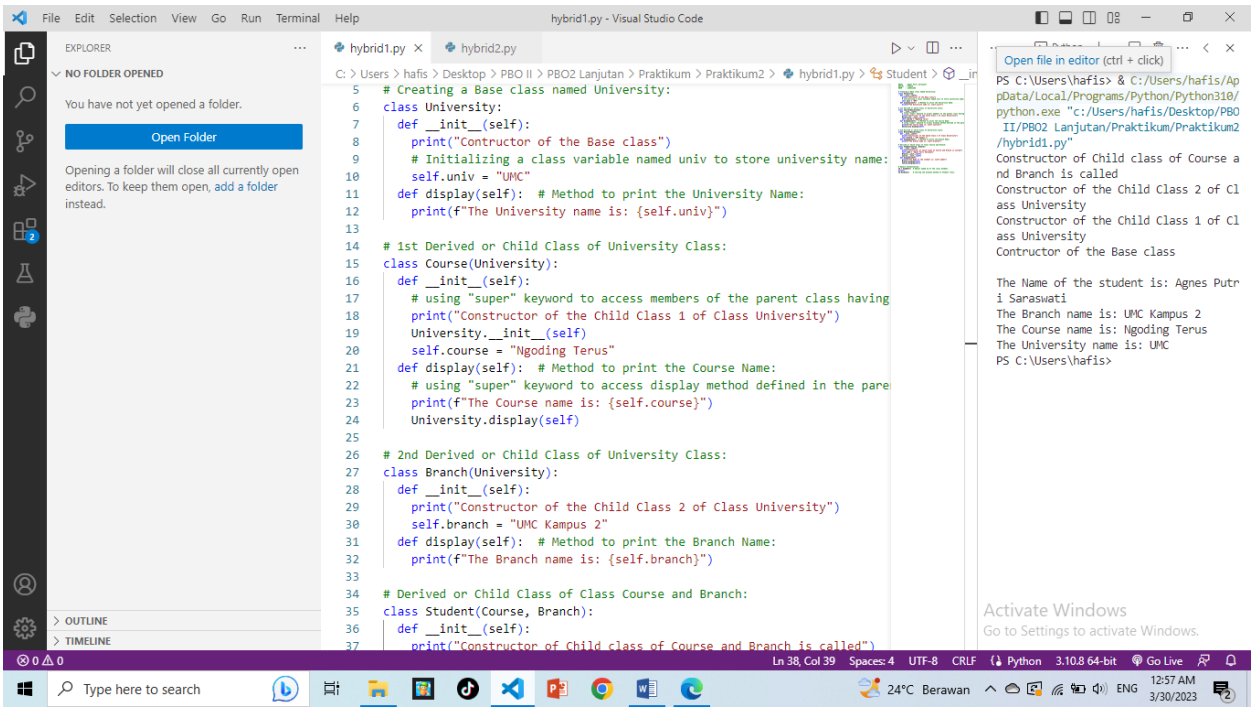
```
        print(f"{self.name} speaks")

class Karyawan(Perusahaan):
    def __init__(self, name, nip):
        super().__init__(name)
        self.nip = nip

    def data(self):
        print(f"{self.name} dengan nip {self.nip}")

class Informasi(Karyawan):
    def __init__(self, name, nip, alamat, sejak):
        super().__init__(name, nip)
        self.alamat = alamat
        self.sejak = sejak

    def speak(self):
        print(f"{self.name} Beralamatkan {self.alamat}, bekerja sejak
{self.sejak}")
        print("="*54)

Informasi = Informasi("Agnes Putri Saraswati", 210511104, "Kuningan", 2019)
Informasi.data()
Informasi.speak()
```



- SCRIPT HYBRID1

```
#Nama   : Agnes Putri Saraswati
#Kelas  : TIF21K
#NIM    : 210511104

# Creating a Base class named University:
class University:
  def __init__(self):
    print("Contructor of the Base class")
    # Initializing a class variable named univ to store university name:
    self.univ = "UMC"
  def display(self):  # Method to print the University Name:
    print(f"The University name is: {self.univ}")

# 1st Derived or Child Class of University Class:
class Course(University):
  def __init__(self):
    # using "super" keyword to access members of the parent class having same
name:
    print("Constructor of the Child Class 1 of Class University")
    University.__init__(self)
    self.course = "Ngoding Terus"
  def display(self):  # Method to print the Course Name:
    # using "super" keyword to access display method defined in the parent class:
    print(f"The Course name is: {self.course}")
    University.display(self)

# 2nd Derived or Child Class of University Class:
class Branch(University):
  def __init__(self):
    print("Constructor of the Child Class 2 of Class University")
    self.branch = "UMC Kampus 2"
  def display(self):  # Method to print the Branch Name:
    print(f"The Branch name is: {self.branch}")

# Derived or Child Class of Class Course and Branch:
class Student(Course, Branch):
  def __init__(self):
    print("Constructor of Child class of Course and Branch is called")
    self.name = "Agnes Putri Saraswati"
    Branch.__init__(self)
    Course.__init__(self)
  def display(self):
```

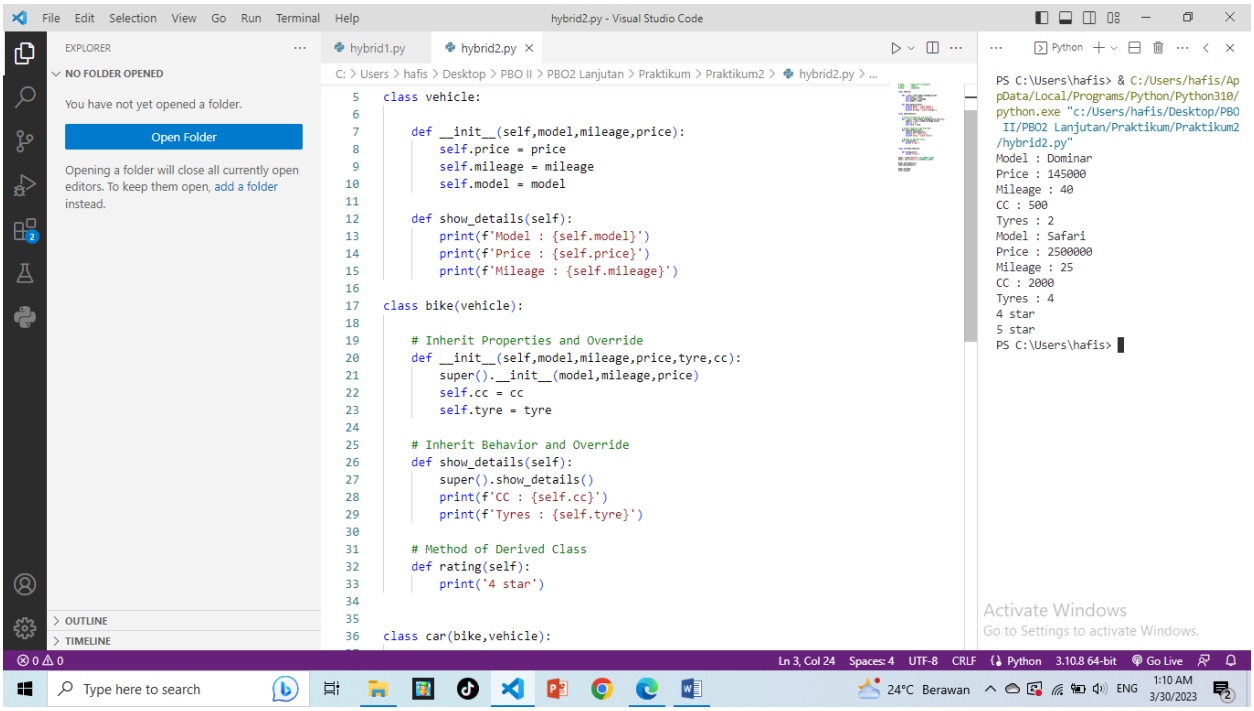
```
    print(f"The Name of the student is: {self.name}")
    Branch.display(self)
    Course.display(self)

# Object Instantiation:
ob = Student()  # Object named ob of the class Student.
print()
ob.display()    # Calling the display method of Student class.
```



- SCRIPT HYBIRD2

```
# Nama      : Agnes Putri Saraswati
# Kelas     : TIF21K
# NIim      : 210511104

class vehicle:

    def __init__(self,model,mileage,price):
        self.price = price
        self.mileage = mileage
```

```
        self.model = model

    def show_details(self):
        print(f'Model : {self.model}')
        print(f'Price : {self.price}')
        print(f'Mileage : {self.mileage}')

class bike(vehicle):

    # Inherit Properties and Override
    def __init__(self,model,mileage,price,tyre,cc):
        super().__init__(model,mileage,price)
        self.cc = cc
        self.tyre = tyre

    # Inherit Behavior and Override
    def show_details(self):
        super().show_details()
        print(f'CC : {self.cc}')
        print(f'Tyres : {self.tyre}')

    # Method of Derived Class
    def rating(self):
        print('4 star')


class car(bike,vehicle):

    def rating(self):
        print('5 star')

bajaj = bike("Dominar",40,145000,2,500)
tata = car("Safari",25,2500000,4,2000)

bajaj.show_details()
tata.show_details()

bajaj.rating()
tata.rating()
```


File Edit Selection View Go Run Terminal Help
hybrid2.py - Visual Studio Code
EXPLORER
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
Opening a folder will close all currently open editors. To keep them open, add a folder instead.
OUTLINE
TIMELINE
hybrid1.py
hybrid2.py
C: > Users > hafis > Desktop > PBO II > PBO2 Lanjutan > Praktikum > Praktikum2 > hybrid2.py > ...
class vehicle:
def __init__(self,model,mileage,price):
self.price = price
self.mileage = mileage
self.model = model
def show_details(self):
print(f'Model : {self.model}')
print(f'Price : {self.price}')
print(f'Mileage : {self.mileage}')
class bike(vehicle):
# Inherit Properties and Override
def __init__(self,model,mileage,price,tyre,cc):
super().__init__(model,mileage,price)
self.cc = cc
self.tyre = tyre
# Inherit Behavior and Override
def show_details(self):
super().show_details()
print(f'CC : {self.cc}')
print(f'Tyres : {self.tyre}')
# Method of Derived Class
def rating(self):
print('4 star')
class car(bike,vehicle):
Python
PS C:\Users\hafis> & C:/Users/hafis/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/hafis/Desktop/PBO II/PBO2 Lanjutan/Praktikum/Praktikum2/hybrid2.py"
Model : Dominar
Price : 145000
Mileage : 40
CC : 500
Tyres : 2
Model : Safari
Price : 2500000
Mileage : 25
CC : 2000
Tyres : 4
4 star
5 star
PS C:\Users\hafis>
Activate Windows
Go to Settings to activate Windows.
Ln 3, Col 24 Spaces: 4 UTF-8 CRLF Python 3.10.8 64-bit Go Live
Type here to search
24°C Berawan
ENG
1:10 AM
3/30/2023