Nama: Ajeng Mayang Arum

Nim: 210511130

Kelas: TI21D

**PRAKTIKUM INHERITANCE**

**Hierarchical 1**

class tepung:

    def \_\_init\_\_(self, merek, terigu):

        self.merek = merek

        self.terigu = terigu

    def get\_merek(self):

        return self.merek

    def get\_terigu(self):

        return self.terigu

class adonan(tepung):

    def \_\_init\_\_(self, merek, terigu, manis):

        super().\_\_init\_\_(merek, terigu)

        self.manis = manis

    def get\_manis(self):

        return self.manis

class cake(tepung):

    def \_\_init\_\_(self, merek, terigu, ketebalancake):

        super().\_\_init\_\_(merek, terigu)

        self.ketebalancake = ketebalancake

    def get\_ketebalancake(self):

        return self.ketebalancake

# Hierarchical Inheritance

class birthdaycake(adonan):

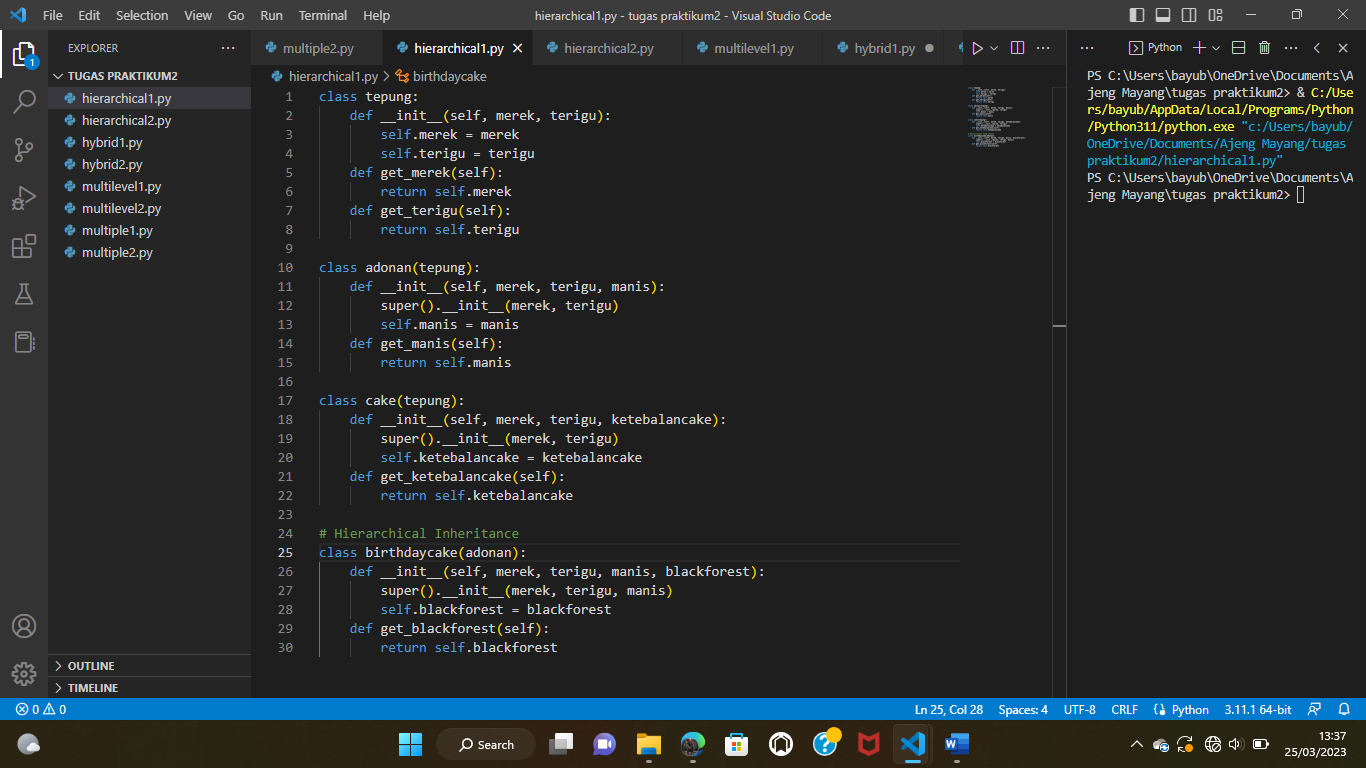
    def \_\_init\_\_(self, merek, terigu, manis, blackforest):

        super().\_\_init\_\_(merek, terigu, manis)

        self.blackforest = blackforest

    def get\_blackforest(self):

        return self.blackforest



**Hierarchical 2**

class kedelai:

    def \_\_init\_\_(self, gizi, vitamin):

        self.gizi = gizi

        self.vitamin = vitamin

    def get\_gizi(self):

        return self.gizi

    def get\_vitamin(self):

        return self.vitamin

class fermentasi(kedelai):

    def \_\_init\_\_(self, gizi,vitamin, ragi):

        super().\_\_init\_\_(gizi,vitamin)

        self.ragi = ragi

    def get\_ragi(self):

        return self.ragi

class tempe(kedelai):

    def \_\_init\_\_(self, gizi,vitamin, teksturtempe):

        super().\_\_init\_\_(gizi,vitamin)

        self.teksturtempe = teksturtempe

    def get\_teksturtempe(self):

        return self.teksturtempe

# Hierarchical Inheritance

class tempegoreng(fermentasi):

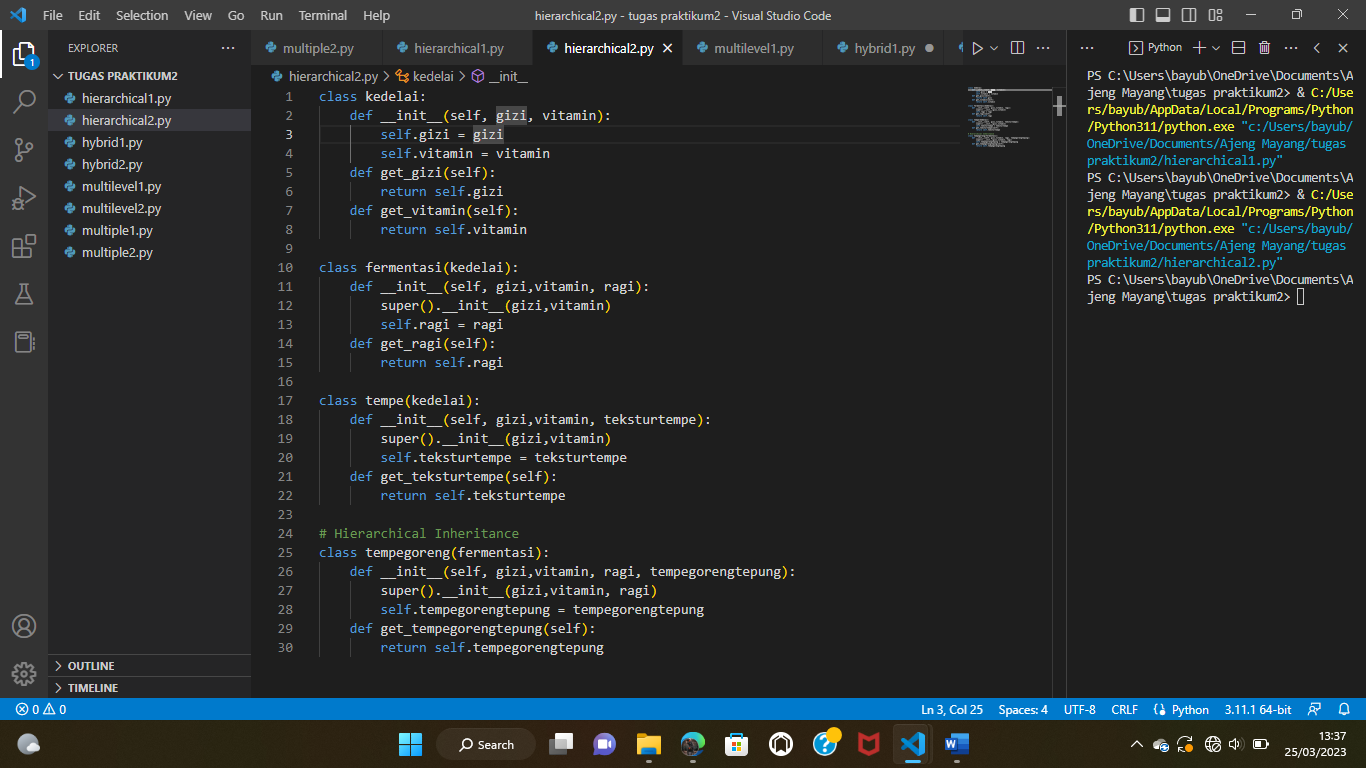
    def \_\_init\_\_(self, gizi,vitamin, ragi, tempegorengtepung):

        super().\_\_init\_\_(gizi,vitamin, ragi)

        self.tempegorengtepung = tempegorengtepung

    def get\_tempegorengtepung(self):

        return self.tempegorengtepung

****

**Hybrid 1**

# Single Inheritance

class GameObject:

    def \_\_init\_\_(self, x, y):

        self.x = x

        self.y = y

# Single Inheritance

class Drawable:

    def draw(self):

        print("Drawing object at: ", self.x, self.y)

# Single Inheritance

class Moveable:

    def move(self, dx, dy):

        self.x += dx

        self.y += dy

# Multiple Inheritance

class Player(GameObject, Drawable, Moveable):

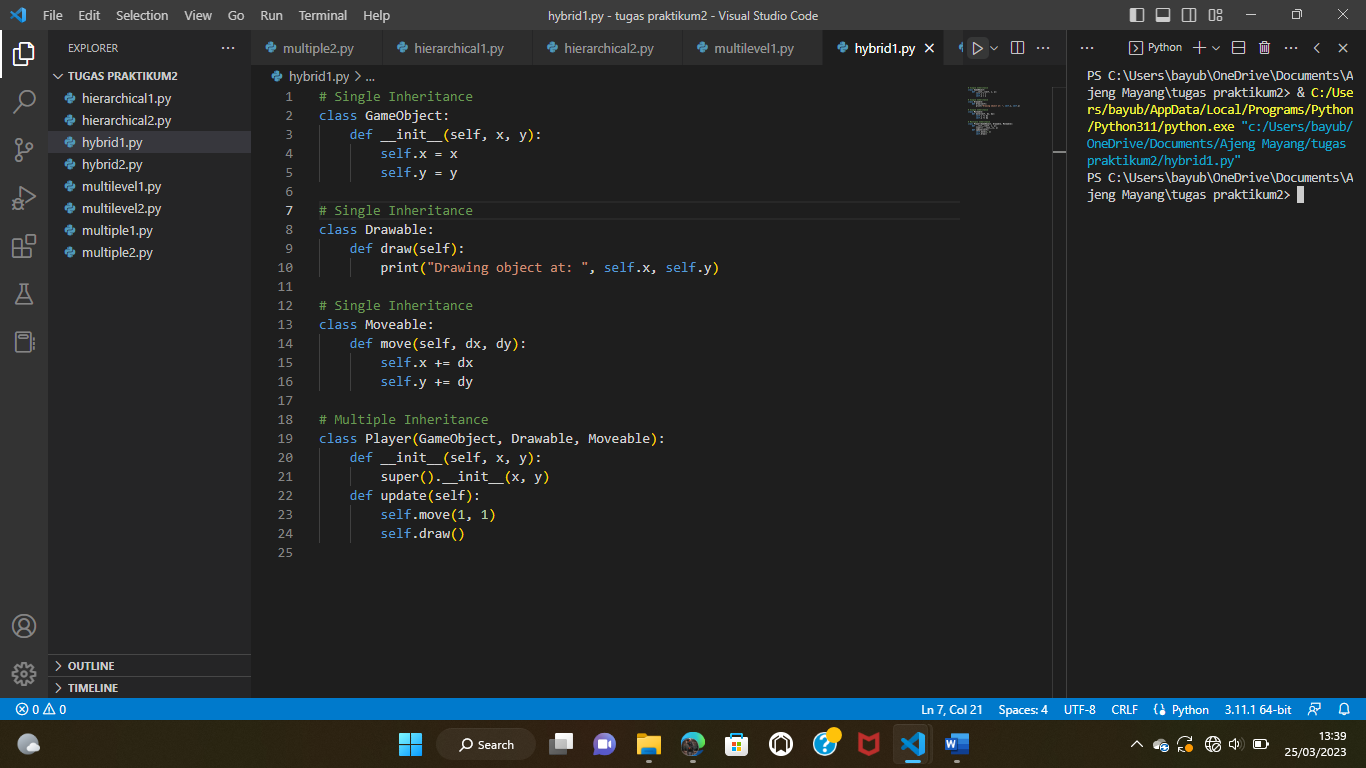
    def \_\_init\_\_(self, x, y):

        super().\_\_init\_\_(x, y)

    def update(self):

        self.move(1, 1)

        self.draw()



**Hybrid 2**

class Seseorang:

    def \_\_init\_\_(self, name, age, address):

        self.name = name

        self.age = age

        self.address = address

    def get\_info(self):

        print("Name:", self.name)

        print("Age:", self.age)

        print("Address:", self.address)

# Single Inheritance

class Mahasiswa(Seseorang):

    def \_\_init\_\_(self, name, age, address, student\_id):

        super().\_\_init\_\_(name, age, address)

        self.student\_id = student\_id

    def get\_info(self):

        super().get\_info()

        print("Student ID:", self.student\_id)

# Single Inheritance

class Employee(Seseorang):

    def \_\_init\_\_(self, name, age, address, employee\_id, salary):

        super().\_\_init\_\_(name, age, address)

        self.employee\_id = employee\_id

        self.salary = salary

    def get\_info(self):

        super().get\_info()

        print("Employee ID:", self.employee\_id)

        print("Salary:", self.salary)

# Multiple Inheritance

class Penulis(Employee, Mahasiswa):

    def \_\_init\_\_(self, name, age, address, employee\_id, salary, student\_id, published\_books):

        Employee.\_\_init\_\_(self, name, age, address, employee\_id, salary)

        Mahasiswa.\_\_init\_\_(self, name, age, address, student\_id)

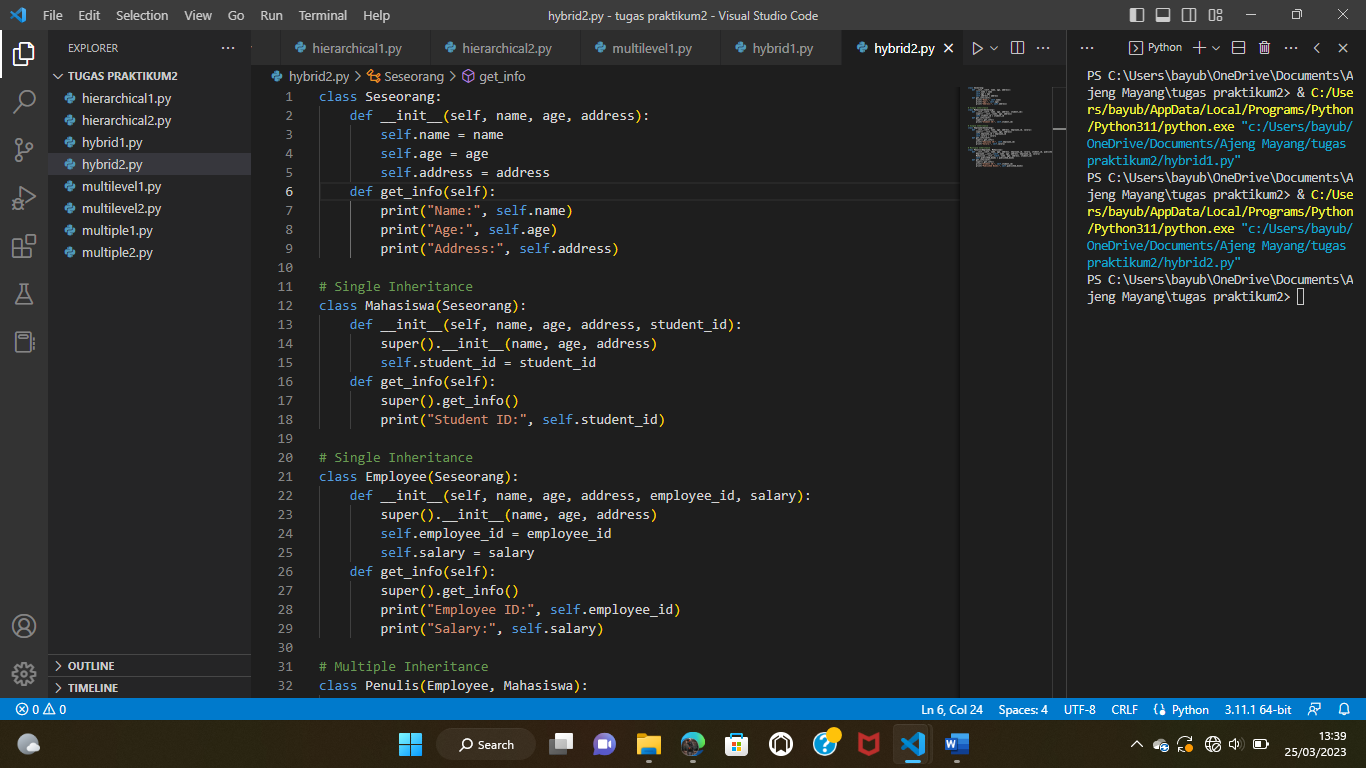
        self.published\_books = published\_books

    def get\_info(self):

        super().get\_info()

        print("Student ID:", self.student\_id)

        print("Published Books:", self.published\_books)



**Multilevel 1**

class binatang:

    def \_\_init\_\_(self, nama):

        self.nama = nama

    def bicara(self):

        print(f"{self.nama} bicara")

class kucing(binatang):

    def \_\_init\_\_(self, nama, keturunan):

        super().\_\_init\_\_(nama)

        self.keturunan = keturunan

    def turunan(self):

        print(f"{self.nama} sejenis kucing anggora {self.keturunan}")

class kucinganggora(kucing):

    def \_\_init\_\_(self, nama, keturunan, asal):

        super().\_\_init\_\_(nama, keturunan)

        self.asal = asal

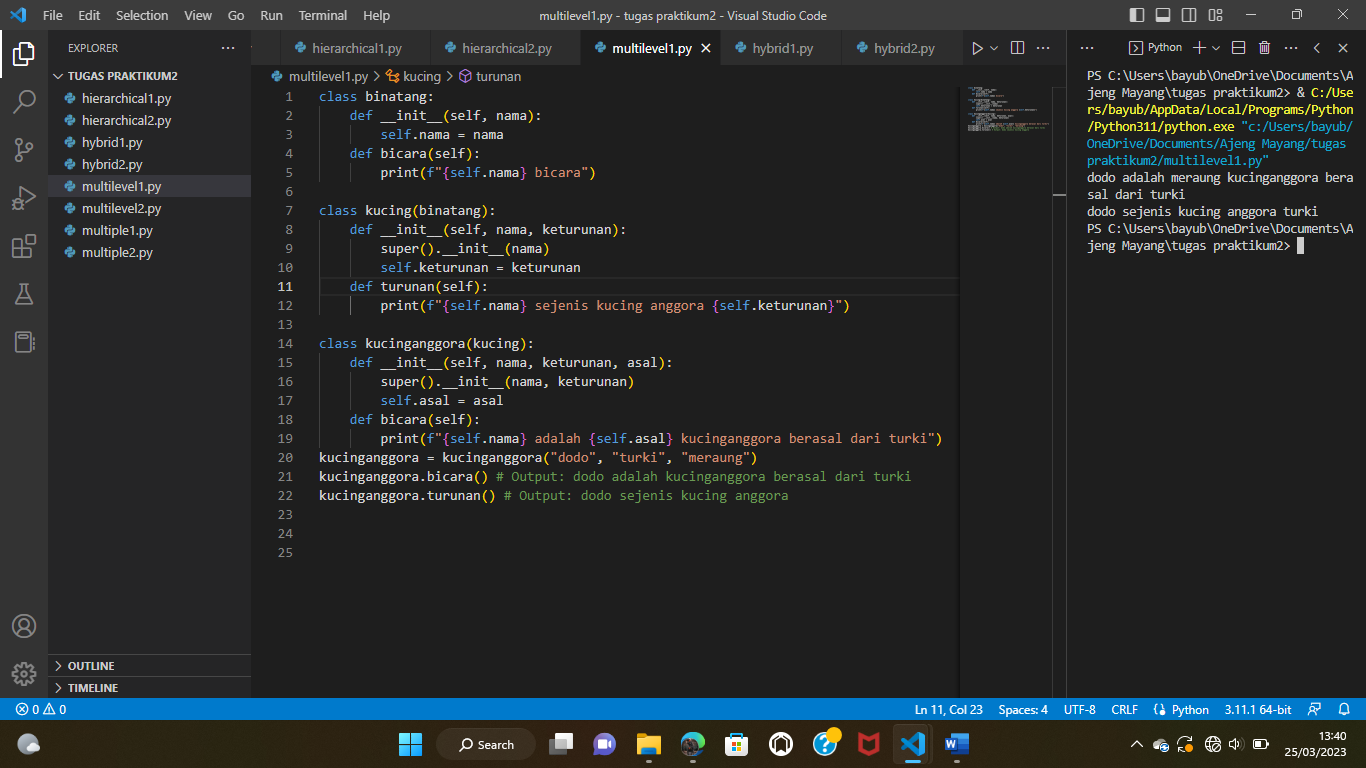
    def bicara(self):

        print(f"{self.nama} adalah {self.asal} kucinganggora berasal dari turki")

kucinganggora = kucinganggora("dodo", "turki", "meraung")

kucinganggora.bicara() # Output: dodo adalah kucinganggora berasal dari turki

kucinganggora.turunan() # Output: dodo sejenis kucing angora



**Multilevel 2**

class binatang:

    def \_\_init\_\_(self, nama):

        self.nama = nama

    def bicara(self):

        print(f"{self.nama} bicara")

class mamalia(binatang):

    def \_\_init\_\_(self, nama, habitat):

        super().\_\_init\_\_(nama)

        self.habitat = habitat

    def hutan(self):

        print(f"{self.nama} habitat di hutan {self.habitat}")

class harimau(mamalia):

    def \_\_init\_\_(self, nama, habitat, makanan):

        super().\_\_init\_\_(nama, habitat)

        self.makanan = makanan

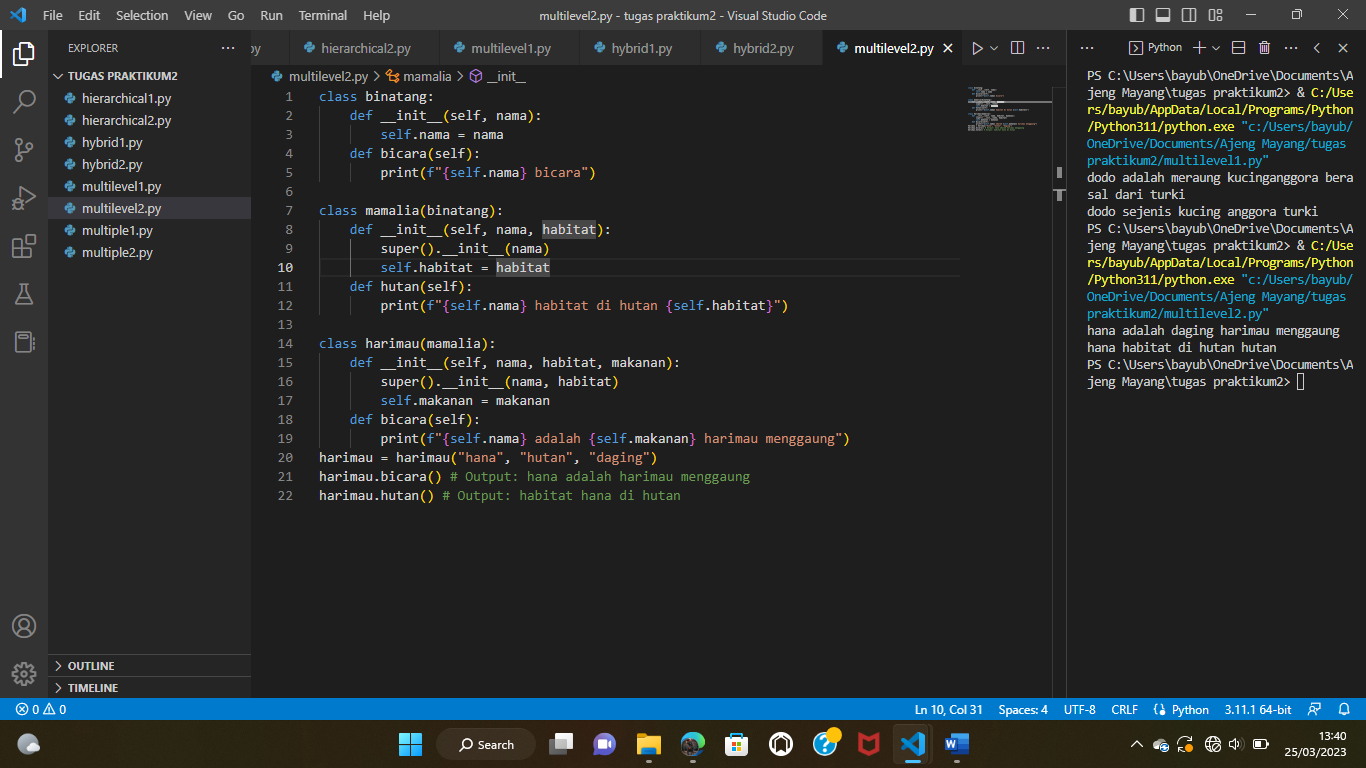
    def bicara(self):

        print(f"{self.nama} adalah {self.makanan} harimau menggaung")

harimau = harimau("hana", "hutan", "daging")

harimau.bicara() # Output: hana adalah harimau menggaung

harimau.hutan() # Output: habitat hana di hutan

****

**Multiple 1**

class karyawan:

    def \_\_init\_\_(self, nama, alamat):

        self.nama = nama

        self.alamat = alamat

    def pergikekantor(self):

        print(self.nama, "sedang pergi ke kantor")

class kantor:

    def \_\_init\_\_(self, nama, pekerjaan):

        self.nama = nama

        self.pekerjaan = pekerjaan

    def bekerja(self):

        print(self.nama, "sedang bekerja")

class karyawankantor(karyawan, kantor):

    def \_\_init\_\_(self, nama, alamat, pekerjaan):

        karyawan.\_\_init\_\_(self, nama, alamat)

        kantor.\_\_init\_\_(self, nama, pekerjaan)

    def bersosialisasi(self):

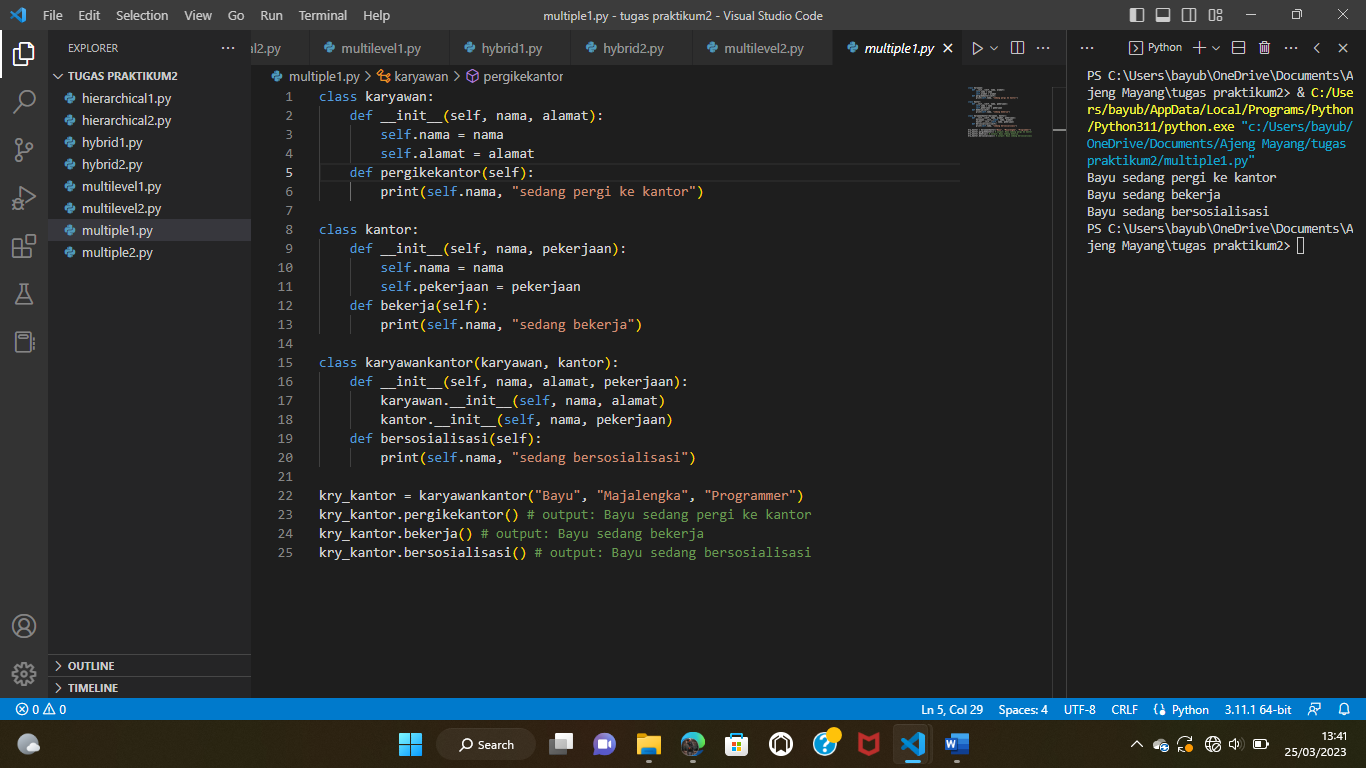
        print(self.nama, "sedang bersosialisasi")

kry\_kantor = karyawankantor("Bayu", "Majalengka", "Programmer")

kry\_kantor.pergikekantor() # output: Bayu sedang pergi ke kantor

kry\_kantor.bekerja() # output: Bayu sedang bekerja

kry\_kantor.bersosialisasi() # output: Bayu sedang bersosialisasi



**Multiple 2**

class Hewan:

    def \_\_init\_\_(self, nama, makanan):

        self.nama = nama

        self.makanan = makanan

    def display\_info(self):

        print(f"Nama: {self.nama}")

        print(f"makanan: {self.makanan}")

class mamalia:

    def \_\_init\_\_(self, jenis, habitat):

        self.jenis = jenis

        self.habitat = habitat

    def display\_info(self):

        print(f"Jenis: {self.jenis}")

        print(f"Habitat: {self.habitat}")

class nokturnal:

    def \_\_init\_\_(self, metamorfosis, habitat):

        self.metamorfosis = metamorfosis

        self.habitat = habitat

    def display\_info(self):

        print(f"Metamorfosis: {self.metamorfosis}")

        print(f"Habitat: {self.habitat}")

class kelelawar(mamalia, nokturnal):

    def \_\_init\_\_(self, nama, makanan, jenis, habitat, metamorfosis):

        Hewan.\_\_init\_\_(self, nama, makanan)

        mamalia.\_\_init\_\_(self, jenis, habitat)

        nokturnal.\_\_init\_\_(self, metamorfosis, habitat)

    def display\_info(self):

        super().display\_info()

        print(f"Nama: {self.nama}")

        print(f"makanan: {self.makanan}")

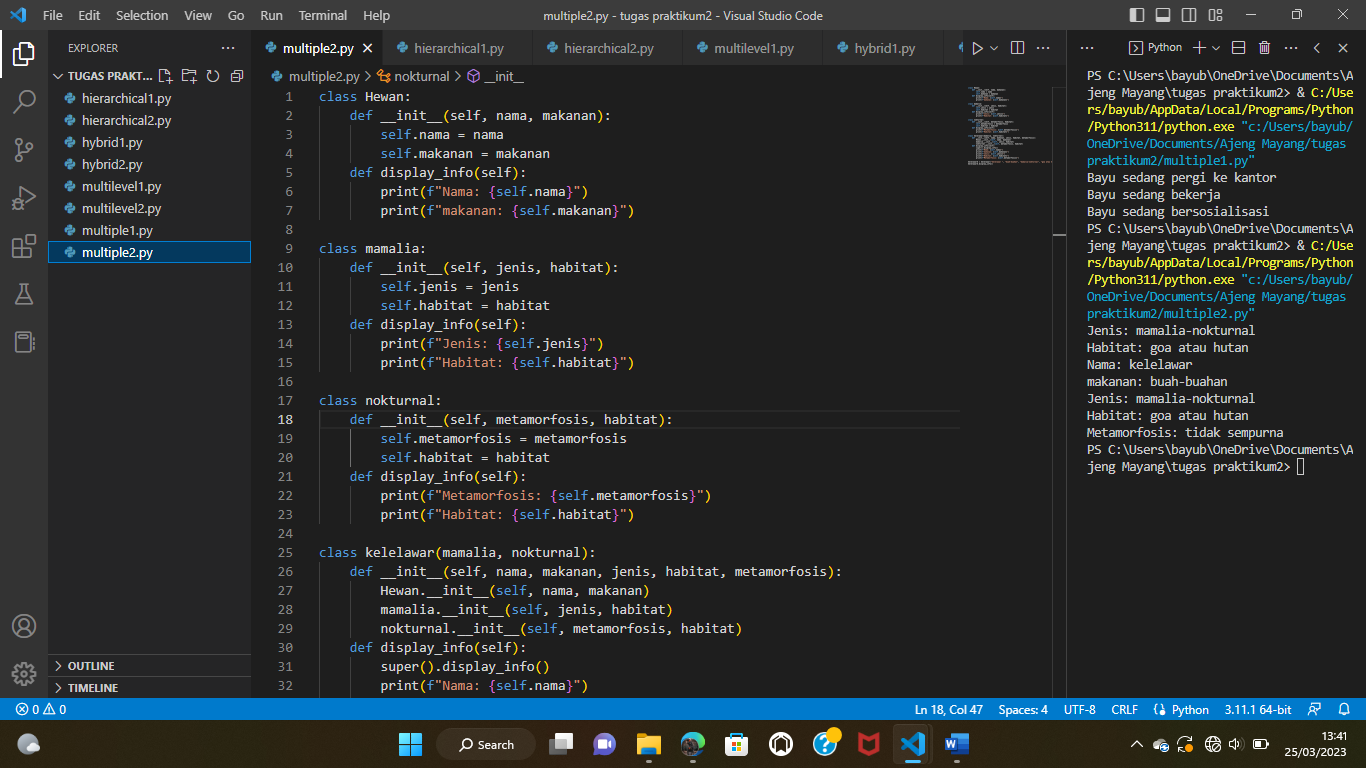
        print(f"Jenis: {self.jenis}")

        print(f"Habitat: {self.habitat}")

        print(f"Metamorfosis: {self.metamorfosis}")

kelelawarA = kelelawar("kelelawar ", "buah-buahan", "mamalia-nokturnal", "goa atau hutan", "tidak sempurna")

kelelawarA.display\_info()

****