LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK LANJUT

2023

Prepared By:

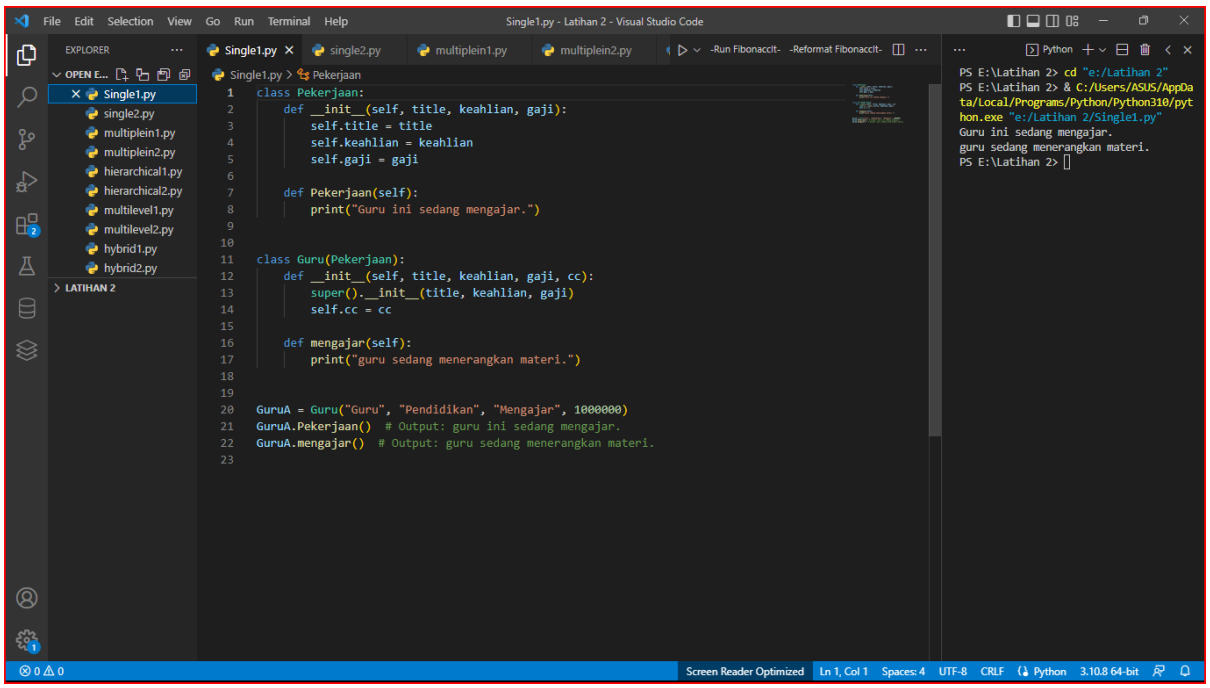
1. Single Inheritance (Pewarisan Tunggal)

```
print ("Nama  : Wira Cantika")
print ("NIM   : 210511171")
print ("Kelas : K1 Teknik Informatika")
print ("--------------------------------")
```

Single Inheritance Input 1

```python
class Pekerjaan:
    def __init__(self, title, keahlian, gaji):
        self.title = title
        self.keahlian = keahlian
        self.gaji = gaji

    def Pekerjaan(self):
        print("Guru ini sedang mengajar.")


class Guru(Pekerjaan):
    def __init__(self, title, keahlian, gaji, cc):
        super().__init__(title, keahlian, gaji)
        self.cc = cc

    def mengajar(self):
        print("guru sedang menerangkan materi.")


GuruA = Guru("Guru", "Pendidikan", "Mengajar", 1000000)
GuruA.Pekerjaan()  # Output: guru ini sedang mengajar.
GuruA.mengajar()  # Output: guru sedang menerangkan materi.
```

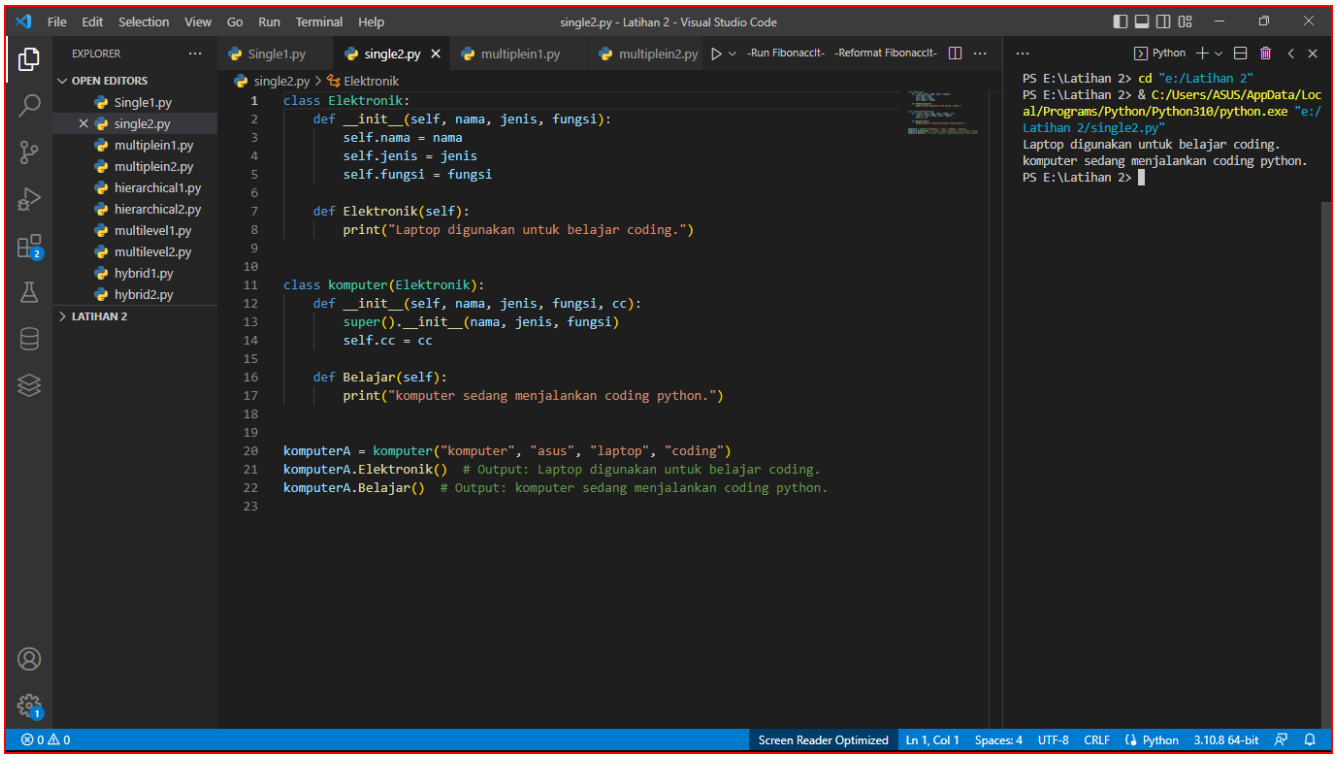
Single Inheritance Output 1



Single Inheritance Input 2

```python
class Elektronik:
    def __init__(self, nama, jenis, fungsi):
        self.nama = nama
        self.jenis = jenis
        self.fungsi = fungsi

    def Elektronik(self):
        print("Laptop digunakan untuk belajar coding.")


class komputer(Elektronik):
    def __init__(self, nama, jenis, fungsi, cc):
        super().__init__(nama, jenis, fungsi)
        self.cc = cc

    def Belajar(self):
        print("komputer sedang menjalankan coding python.")


komputerA = komputer("komputer", "asus", "laptop", "coding")
komputerA.Elektronik()  # Output: Laptop digunakan untuk belajar coding.
komputerA.Belajar()  # Output: komputer sedang menjalankan coding python.
```

Single Inheritance Output 2

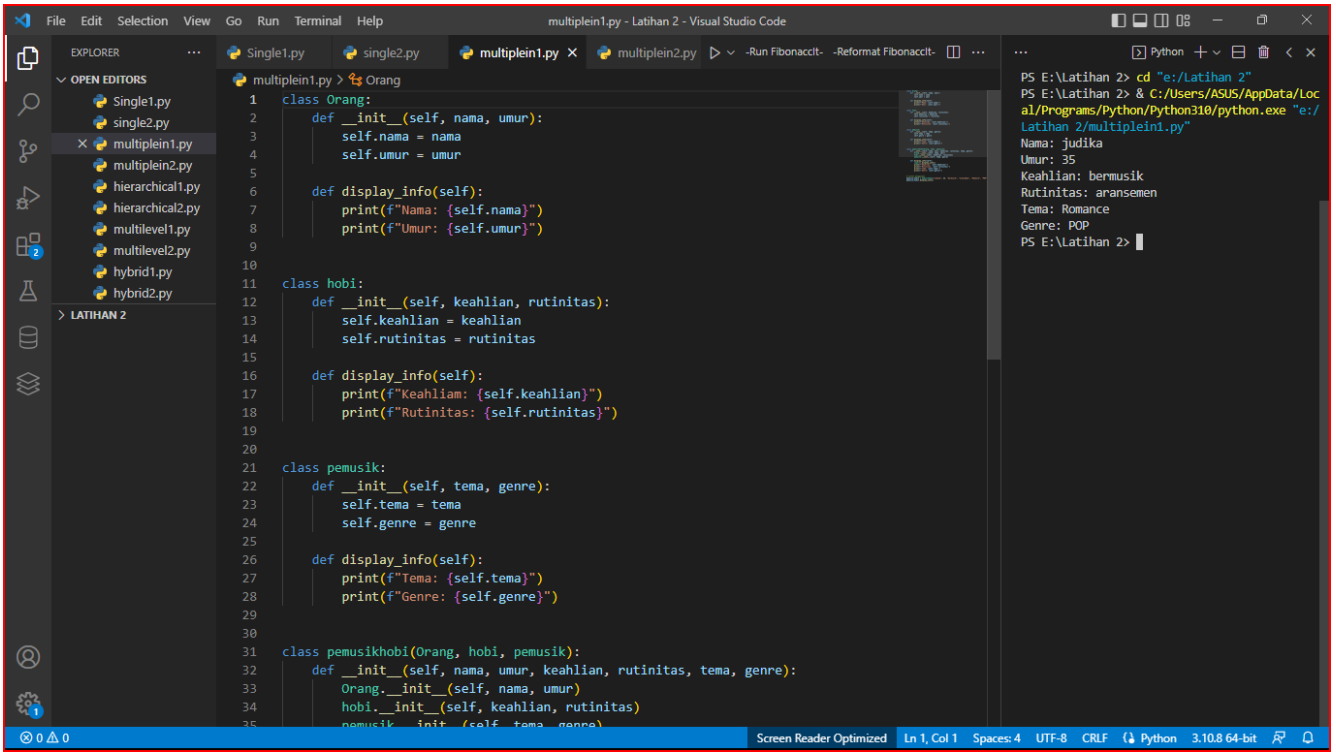


2. Multiple inheritance (Pewarisan Ganda)

Multiple inheritance Input 1

```
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 hobi:
    def __init__(self, keahlian, rutinitas):
        self.keahlian = keahlian
        self.rutinitas = rutinitas

    def display_info(self):
        print(f"Keahliam: {self.keahlian}")
        print(f"Rutinitas: {self.rutinitas}")


class pemusik:
    def __init__(self, tema, genre):
        self.tema = tema
        self.genre = genre

    def display_info(self):
        print(f"Tema: {self.tema}")
        print(f"Genre: {self.genre}")


class pemusikhobi(Orang, hobi, pemusik):
    def __init__(self, nama, umur, keahlian, rutinitas, tema, genre):
        Orang.__init__(self, nama, umur)
        hobi.__init__(self, keahlian, rutinitas)
        pemusik.__init__(self, tema, genre)

    def display_info(self):
        super().display_info()
        print(f"Keahlian: {self.keahlian}")
        print(f"Rutinitas: {self.rutinitas}")
        print(f"Tema: {self.tema}")
        print(f"Genre: {self.genre}")


# contoh penggunaan
pemusik_hobiC = pemusikhobi("judika", 35, "bermusik", "aransemen", "Romance", "POP")
pemusik_hobiC.display_info()
```

Multiple inheritance Output 1

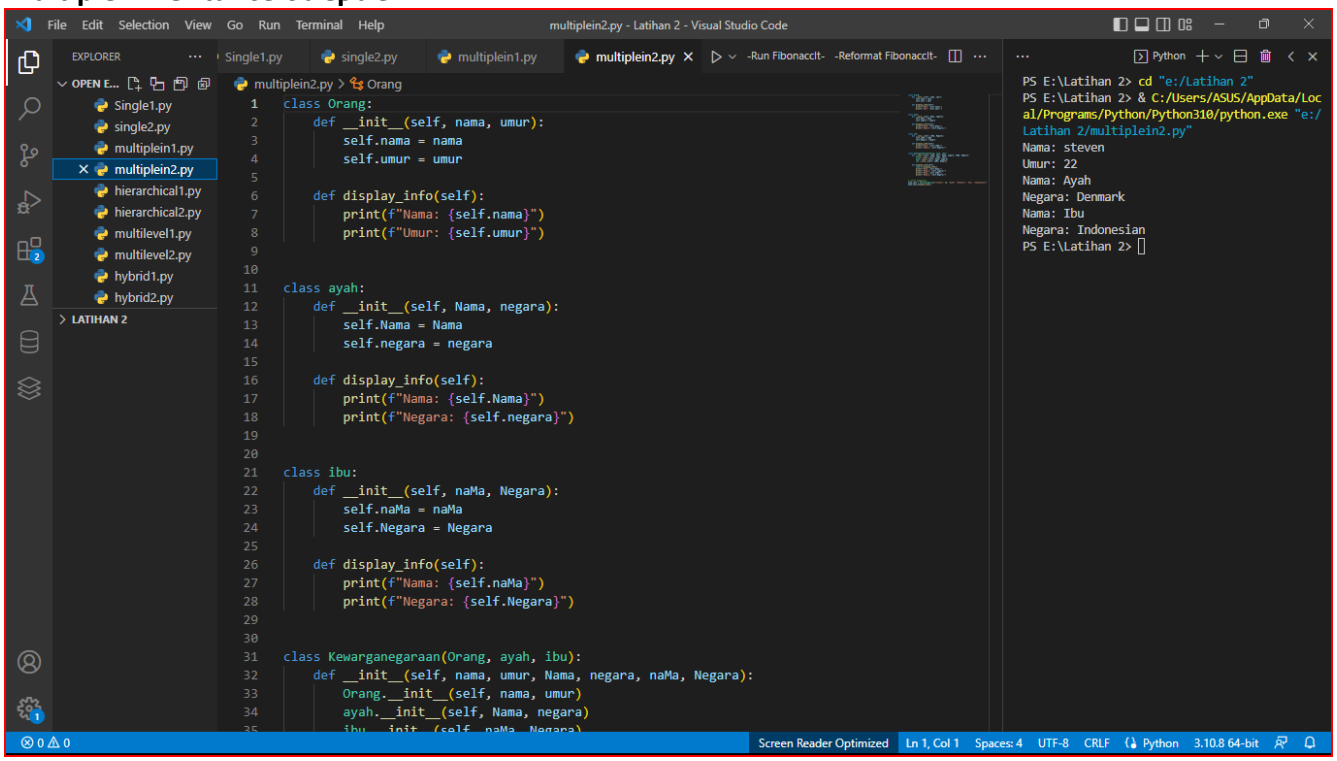


Multiple inheritance Input 2

```
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 ayah:
    def __init__(self, Nama, negara):
        self.Nama = Nama
        self.negara = negara

    def display_info(self):
        print(f"Nama: {self.Nama}")
        print(f"Negara: {self.negara}")


class ibu:
    def __init__(self, naMa, Negara):
        self.naMa = naMa
        self.Negara = Negara

    def display_info(self):
        print(f"Nama: {self.naMa}")
        print(f"Negara: {self.Negara}")
```

```python
class Kewarganegaraan(Orang, ayah, ibu):
    def __init__(self, nama, umur, Nama, negara, naMa, Negara):
        Orang.__init__(self, nama, umur)
        ayah.__init__(self, Nama, negara)
        ibu.__init__(self, naMa, Negara)

    def display_info(self):
        super().display_info()
        print(f"Nama: {self.Nama}")
        print(f"Negara: {self.negara}")
        print(f"Nama: {self.naMa}")
        print(f"Negara: {self.Negara}")


# contoh penggunaan
ayah_ibuC = Kewarganegaraan("steven", 22, "Ayah", "Denmark", "Ibu", "Indonesian")
ayah_ibuC.display_info()
```

Multiple inheritance Output 2

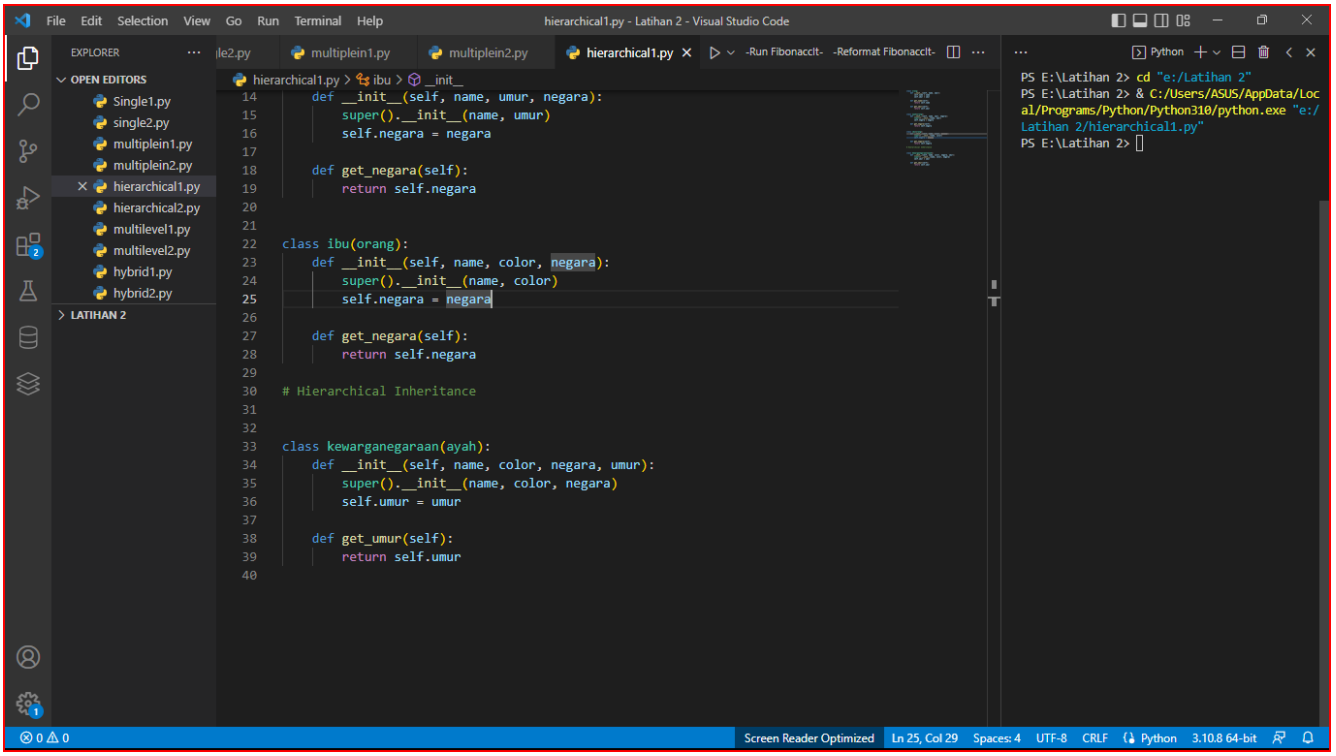


3. Hierarchical inheritance (Pewarisan Hirarki)

Hierarchical inheritance Input 1

```
class orang:
    def __init__(self, name, umur):
        self.name = name
        self.umur = umur

    def get_name(self):
        return self.name

    def get_umur(self):
        return self.umur


class ayah(orang):
    def __init__(self, name, umur, negara):
        super().__init__(name, umur)
        self.negara = negara

    def get_negara(self):
        return self.negara


class ibu(orang):
    def __init__(self, name, color, negara):
        super().__init__(name, color)
        self.negara = negara

    def get_negara(self):
        return self.negara

# Hierarchical Inheritance


class kewarganegaraan(ayah):
    def __init__(self, name, color, negara, umur):
        super().__init__(name, color, negara)
        self.umur = umur

    def get_umur(self):
        return self.umur
```

Hierarchical inheritance Output 1

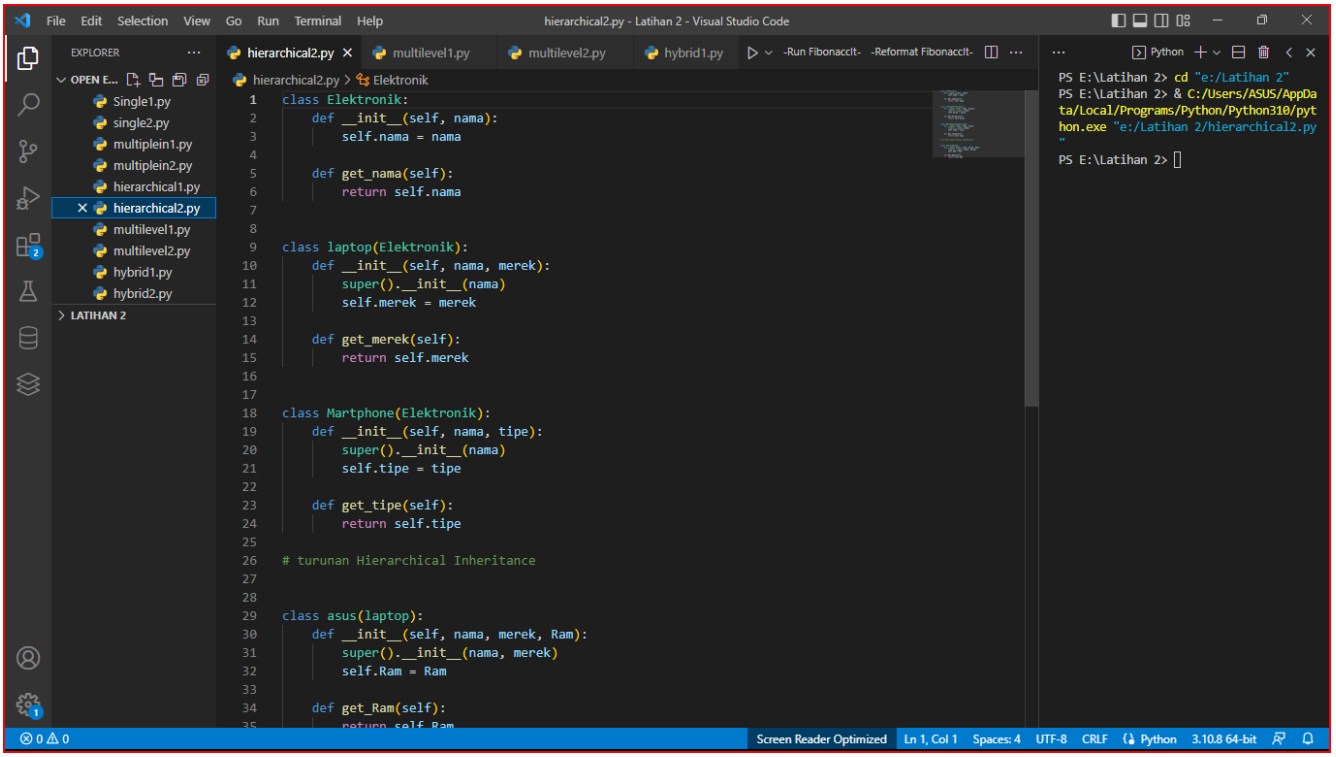


Hierarchical inheritance Input 2

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

    def get_nama(self):
        return self.nama


class laptop(Elektronik):
    def __init__(self, nama, merek):
        super().__init__(nama)
        self.merek = merek

    def get_merek(self):
        return self.merek


class Martphone(Elektronik):
    def __init__(self, nama, tipe):
        super().__init__(nama)
        self.tipe = tipe

    def get_tipe(self):
        return self.tipe

# turunan Hierarchical Inheritance


class asus(laptop):
    def __init__(self, nama, merek, Ram):
```

```
        super().__init__(nama, merek)
        self.Ram = Ram

    def get_Ram(self):
        return self.Ram
```

Hierarchical inheritance Output 2

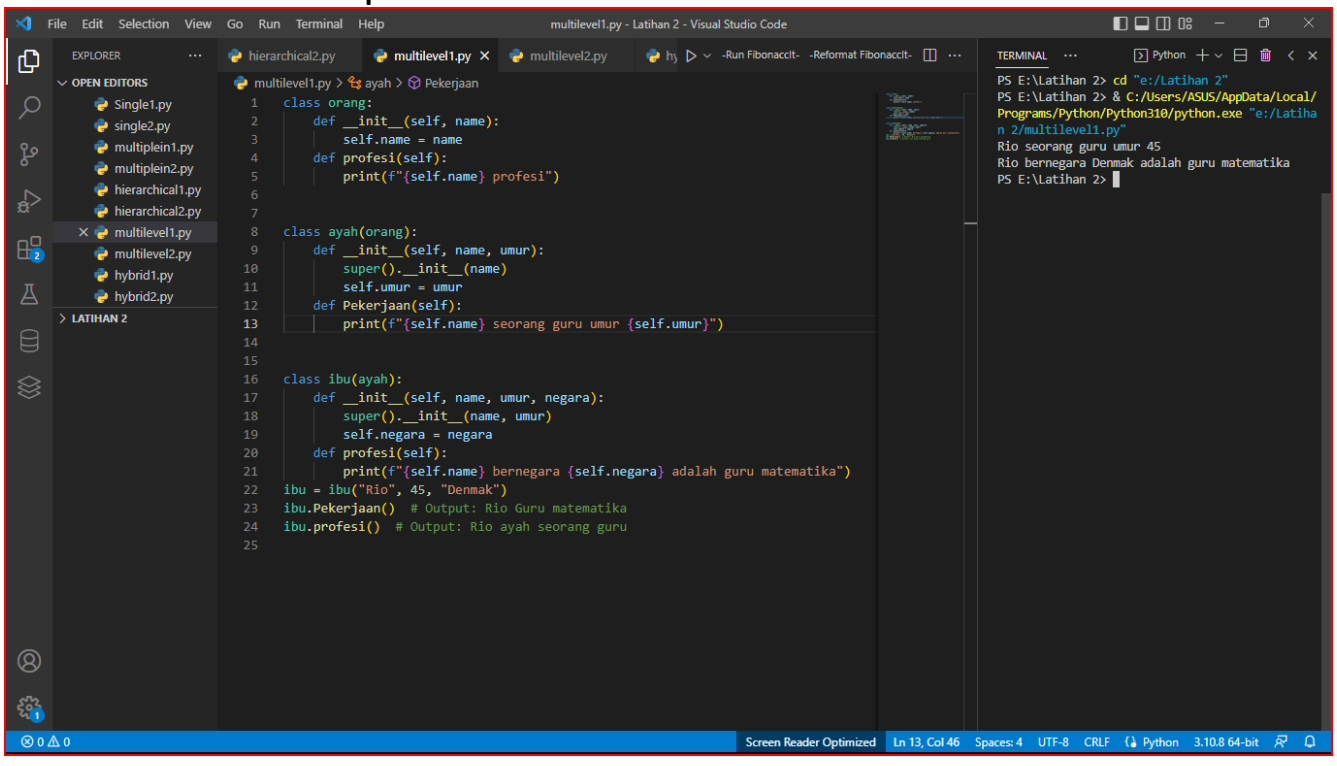


4. Multi-level Inheritance (Pewarisan Bertingkat)

Multi-level Inheritance Input 1

```python
class orang:
    def __init__(self, name):
        self.name = name
    def profesi(self):
        print(f"{self.name} profesi")


class ayah(orang):
    def __init__(self, name, umur):
        super().__init__(name)
        self.umur = umur
    def Pekerjaan(self):
        print(f"{self.name} seorang guru umur {self.umur}")


class ibu(ayah):
    def __init__(self, name, umur, negara):
        super().__init__(name, umur)
        self.negara = negara
    def profesi(self):
        print(f"{self.name} bernegara {self.negara} adalah guru matematika")
ibu = ibu("Rio", 45, "Denmak")
ibu.Pekerjaan()  # Output: Rio Guru matematika
ibu.profesi()  # Output: Rio ayah seorang guru
```

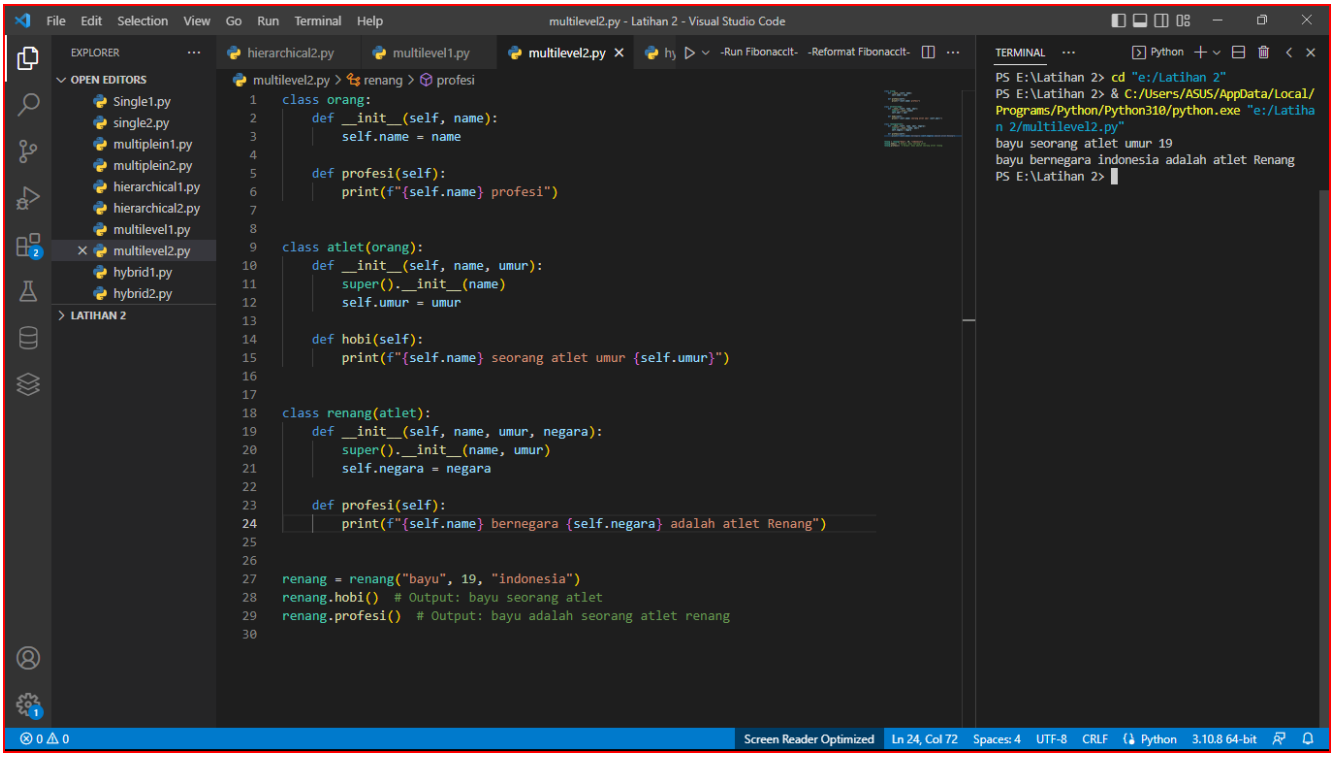
Multi-level Inheritance Output 1



Multi-level Inheritance Input 2

```python
class orang:
    def __init__(self, name):
        self.name = name

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


class atlet(orang):
    def __init__(self, name, umur):
        super().__init__(name)
        self.umur = umur

    def hobi(self):
        print(f"{self.name} seorang atlet umur {self.umur}")


class renang(atlet):
    def __init__(self, name, umur, negara):
        super().__init__(name, umur)
        self.negara = negara

    def profesi(self):
        print(f"{self.name} bernegara {self.negara} adalah atlet Renang")


renang = renang("bayu", 19, "indonesia")
renang.hobi()  # Output: bayu seorang atlet
renang.profesi()  # Output: bayu adalah seorang atlet renang
```

Multi-level Inheritance Output 2

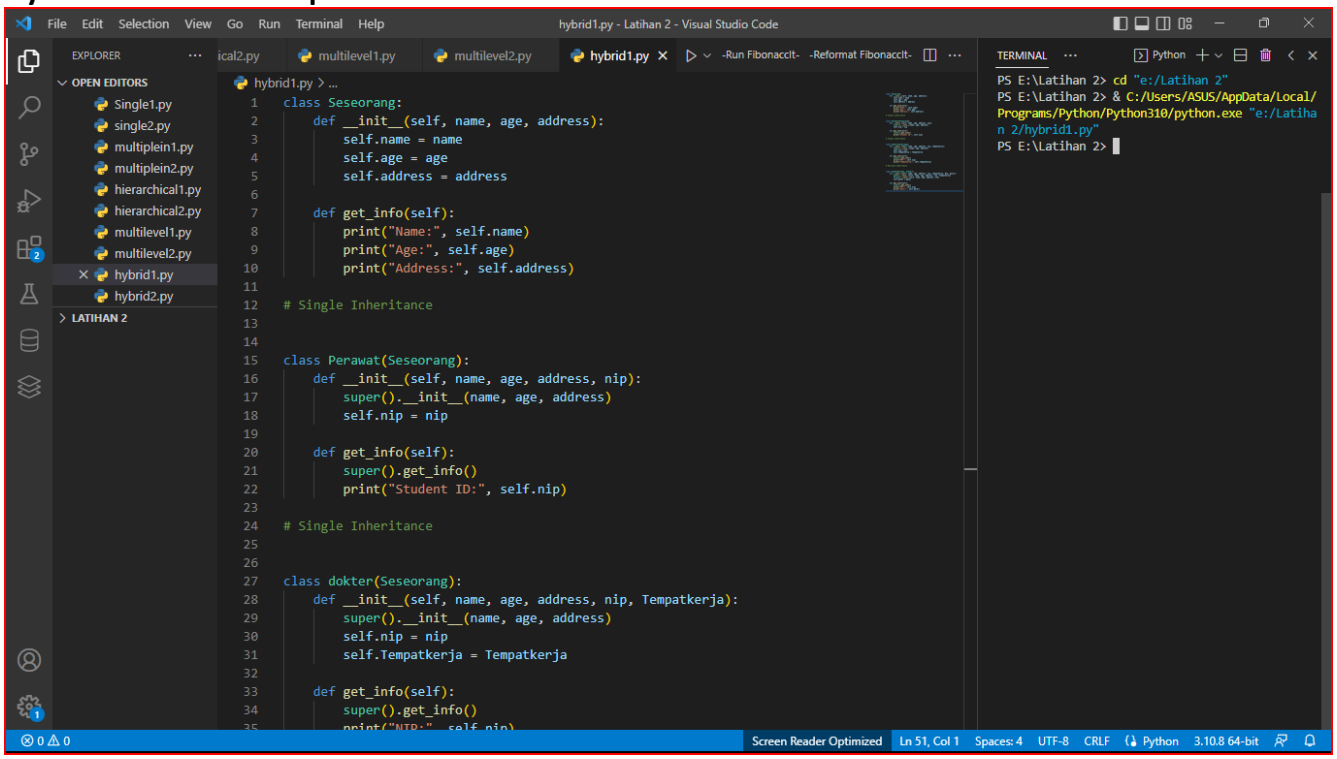


5. Hybrid Inheritance (Pewarisan Campuran)

Hybrid Inheritance Input 1

```
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 Perawat(Seseorang):
    def __init__(self, name, age, address, nip):
        super().__init__(name, age, address)
        self.nip = nip

    def get_info(self):
        super().get_info()
        print("Student ID:", self.nip)

# Single Inheritance


class dokter(Seseorang):
    def __init__(self, name, age, address, nip, Tempatkerja):
        super().__init__(name, age, address)
        self.nip = nip
        self.Tempatkerja = Tempatkerja

    def get_info(self):
        super().get_info()
        print("NIP:", self.nip)
        print("Tempatkerja:", self.Tempatkerja)

# Multiple Inheritance


class bidan(Perawat, dokter):
    def __init__(self, name, age, address, nip, Tempatkerja, Nip, gelar):
        dokter.__init__(self, name, age, address, nip, Tempatkerja)
        Perawat.__init__(self, name, age, address, nip)
        self.gelar = gelar

    def get_info(self):
        super().get_info()
        print("NIP:", self.nip)
        print("Gelar:", self.gelar)
```

Hybrid Inheritance Output 1



Hybrid Inheritance Input 2

```
class tumbuhan:
    def __init__(self, name, struktur, jenis):
        self.name = name
        self.struktur = struktur
        self.jenis = jenis

    def get_info(self):
        print("Name:", self.name)
        print("Struktur:", self.struktur)
        print("jenis:", self.jenis)

# Single Inheritance


class akar(tumbuhan):
    def __init__(self, epidermis, korteks, endodermis, stele):
        super().__init__(epidermis, korteks, endodermis)
        self.stele = stele

    def get_info(self):
        super().get_info()
        print("Student ID:", self.stele)

# Single Inheritance


class batang(tumbuhan):
```

```
    def __init__(self, epidermis, korteks, bekaspengangkut, stele):
        super().__init__(epidermis, korteks, bekaspengangkut)
        self.bekaspengangkut = bekaspengangkut
        self.stele = stele

    def get_info(self):
        super().get_info()
        print("Bekas Pengangkut:", self.bekaspengangkut)
        print("Stele:", self.stele)

# Multiple Inheritance


class daun(akar, batang):
    def __init__(self, epidermis, korteks, endodermis, bekaspengangkut, stele, jenis):
        batang.__init__(self, epidermis, korteks, bekaspengangkut, stele)
        akar.__init__(self, epidermis, korteks, endodermis, stele)
        self.jenis = jenis

    def get_info(self):
        super().get_info()
        print("NIP:", self.stele)
        print("Gelar:", self.jenis)
```

Hybrid Inheritance Output 2