

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

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

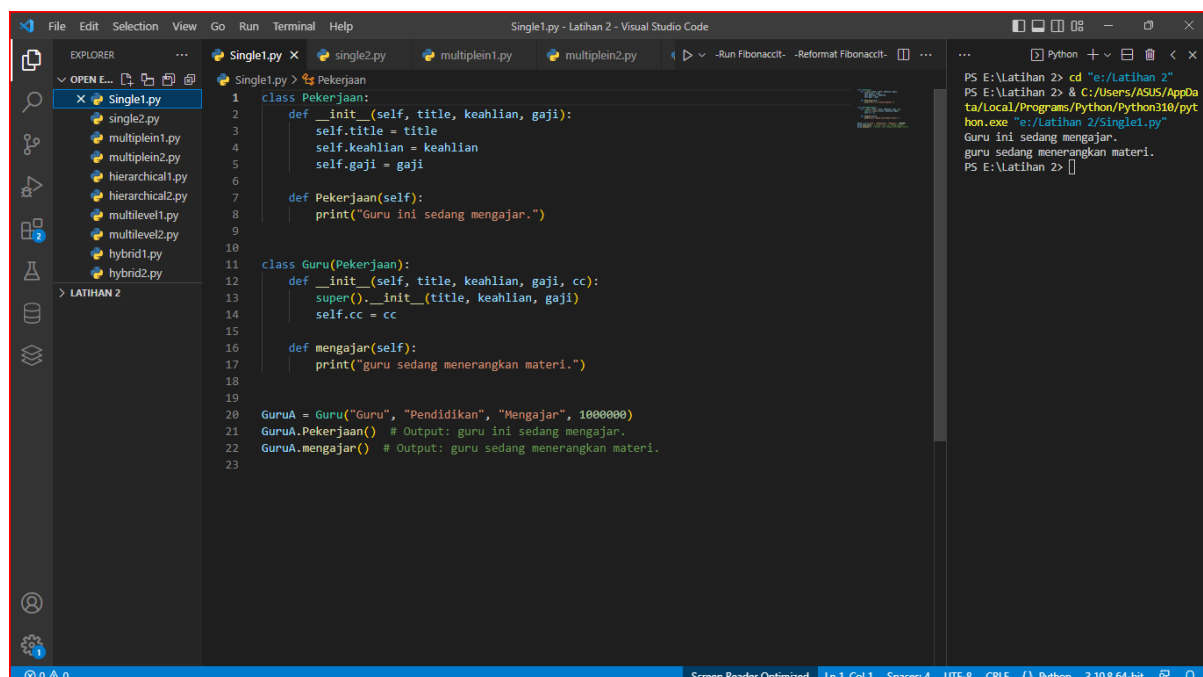
    def Pekerjajaan(self):
        print("Guru ini sedang mengajar.")

class Guru(Pekerjajaan):
    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

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

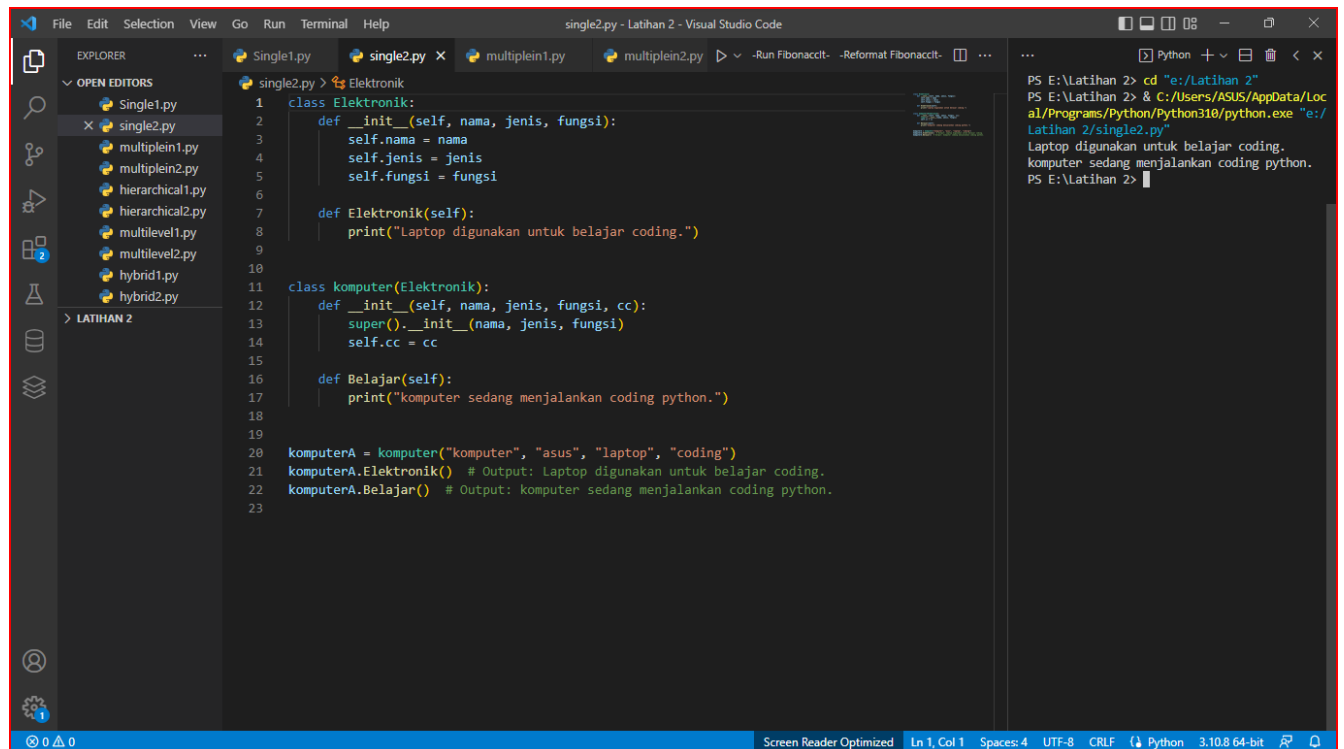
    def Elektronik(self):
        print("Laptop digunakan untuk belajar coding.")

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>Keahlian: {self.keahlian}")
        print(f>Rutinitas: {self.rutinitas}")

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

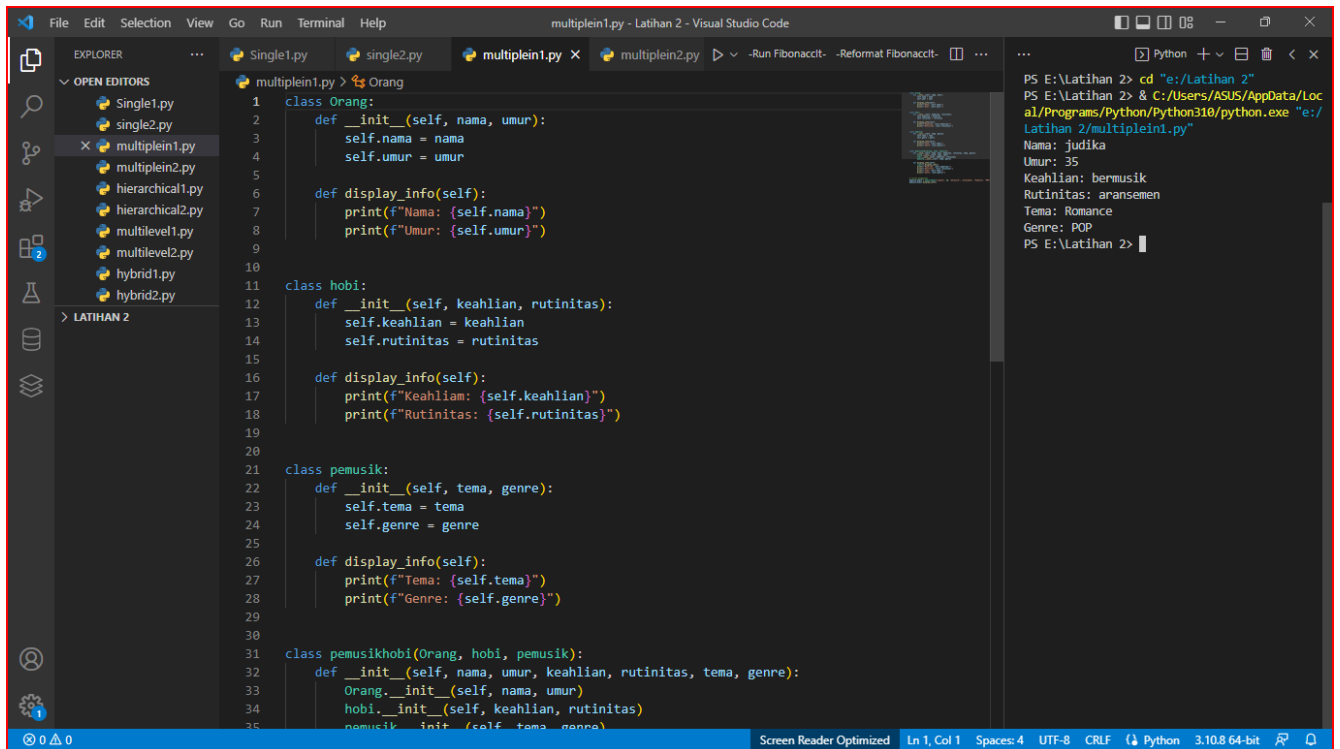
    def display_info(self):
        print(f>Tema: {self.tema}")
        print(f>Genre: {self.genre}")

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



The screenshot shows the Visual Studio Code interface with a file explorer on the left, a code editor in the center, and a terminal on the right. The code editor displays a Python script named `multiplein1.py` with the following content:

```
1 class Orang:
2     def __init__(self, nama, umur):
3         self.nama = nama
4         self.umur = umur
5
6     def display_info(self):
7         print(f>Nama: {self.nama}")
8         print(f"> Umur: {self.umur}")
9
10
11 class hobi:
12     def __init__(self, keahlian, rutinitas):
13         self.keahlian = keahlian
14         self.rutinitas = rutinitas
15
16     def display_info(self):
17         print(f"> Keahlian: {self.keahlian}")
18         print(f"> Rutinitas: {self.rutinitas}")
19
20
21 class pemusik:
22     def __init__(self, tema, genre):
23         self.tema = tema
24         self.genre = genre
25
26     def display_info(self):
27         print(f"> Tema: {self.tema}")
28         print(f"> Genre: {self.genre}")
29
30
31 class pemusikhobi(Orang, hobi, pemusik):
32     def __init__(self, nama, umur, keahlian, rutinitas, tema, genre):
33         Orang.__init__(self, nama, umur)
34         hobi.__init__(self, keahlian, rutinitas)
35         pemusik.__init__(self, tema, genre)
```

The terminal on the right shows the execution of the script with the following output:

```
PS E:\Latihan 2> cd "E:\Latihan 2"
PS E:\Latihan 2> & C:\Users\ASUS\AppData\Local\Programs\Python\Python310\python.exe "E:\Latihan 2\multiplein1.py"
Nama: judika
Umur: 35
Keahlian: bermusik
Rutinitas: aransemen
Tema: Romance
Genre: POP
PS E:\Latihan 2>
```

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

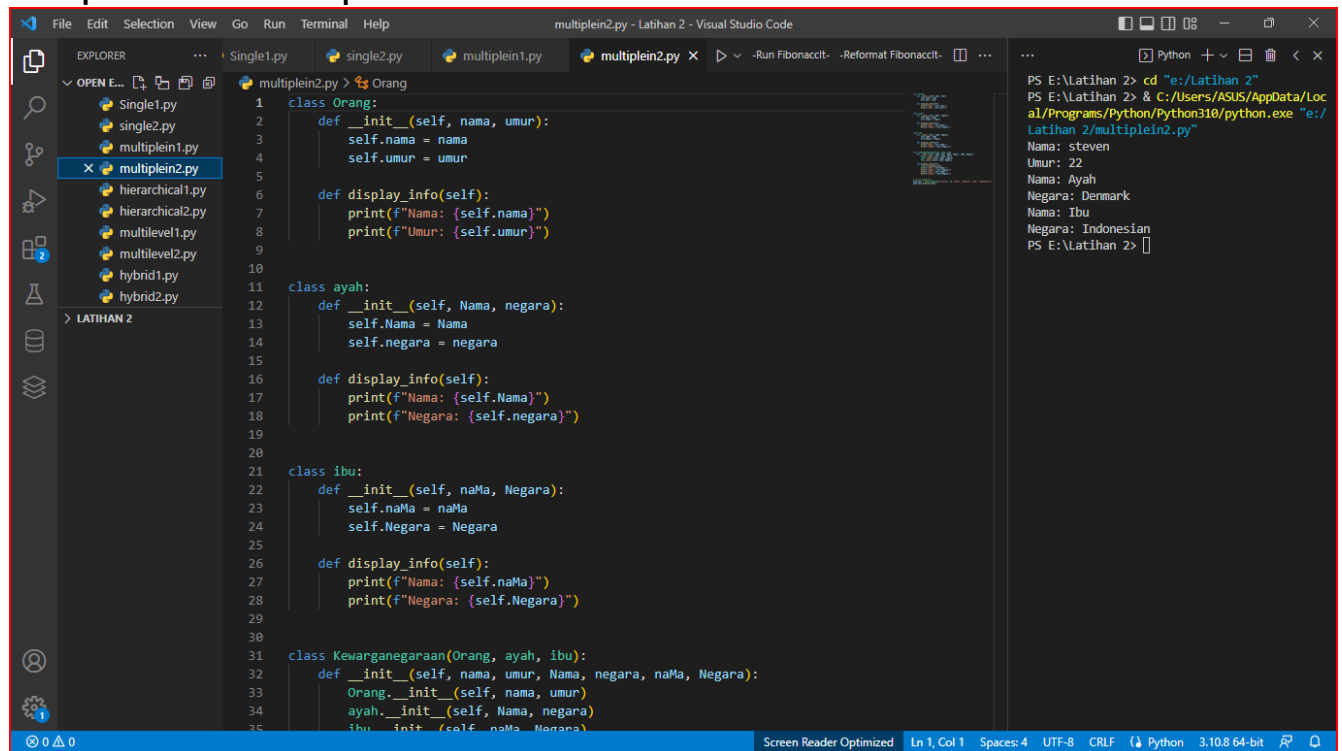
    def display_info(self):
        print(f"> Nama: {self.naMa}")
        print(f"> Negara: {self.Negara}")
```

```
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

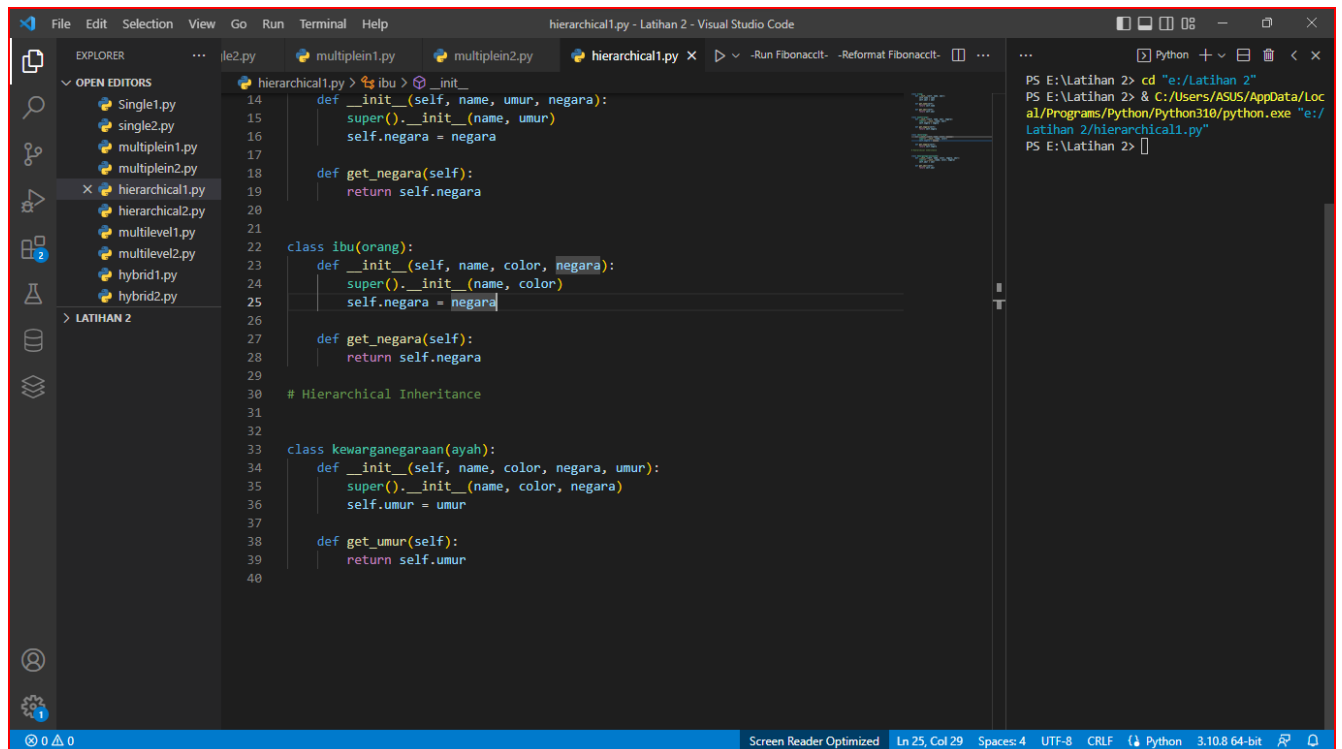
    def get_negara(self):
        return self.negara

# 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



```
File Edit Selection View Go Run Terminal Help
hierarchical1.py - Latihan 2 - Visual Studio Code

EXPLORER
... ile2.py multiplein1.py multiplein2.py hierarchical1.py
OPEN EDITORS
Single.py
single2.py
multiplein1.py
multiplein2.py
hierarchical1.py
hierarchical2.py
multilevel1.py
multilevel2.py
hybrid1.py
hybrid2.py
LATIHAN 2

hierarchical1.py
14 def __init__(self, name, umur, negara):
15     super().__init__(name, umur)
16     self.negara = negara
17
18 def get_negara(self):
19     return self.negara
20
21
22 class ibu(orang):
23     def __init__(self, name, color, negara):
24         super().__init__(name, color)
25         self.negara = negara
26
27     def get_negara(self):
28         return self.negara
29
30 # Hierarchical Inheritance
31
32
33 class kewarganegaraan(ayah):
34     def __init__(self, name, color, negara, umur):
35         super().__init__(name, color, negara)
36         self.umur = umur
37
38     def get_umur(self):
39         return self.umur
40

Python
PS E:\Latihan 2> cd "E:/Latihan 2"
PS E:\Latihan 2> & C:/Users/ASUS/AppData/Local/Programs/Python/Python310/python.exe "e:/Latihan 2/hierarchical1.py"
PS E:\Latihan 2>
```

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

```

1 class Elektronik:
2     def __init__(self, nama):
3         self.nama = nama
4
5     def get_nama(self):
6         return self.nama
7
8
9 class laptop(Elektronik):
10    def __init__(self, nama, merek):
11        super().__init__(nama)
12        self.merek = merek
13
14    def get_merek(self):
15        return self.merek
16
17
18 class Martphone(Elektronik):
19    def __init__(self, nama, tipe):
20        super().__init__(nama)
21        self.tipe = tipe
22
23    def get_tipe(self):
24        return self.tipe
25
26 # turunan Hierarchical Inheritance
27
28
29 class asus(laptop):
30    def __init__(self, nama, merek, Ram):
31        super().__init__(nama, merek)
32        self.Ram = Ram
33
34    def get_Ram(self):
35        return self.Ram

```

PS E:\Latihan 2> cd "e:/Latihan 2"

PS E:\Latihan 2> & C:/Users/ASUS/AppData/Local/Programs/Python/Python310/python.exe "e:/Latihan 2/hierarchical2.py"

PS E:\Latihan 2>

4. Multi-level Inheritance (Pewarisan Bertingkat)

Multi-level Inheritance Input 1

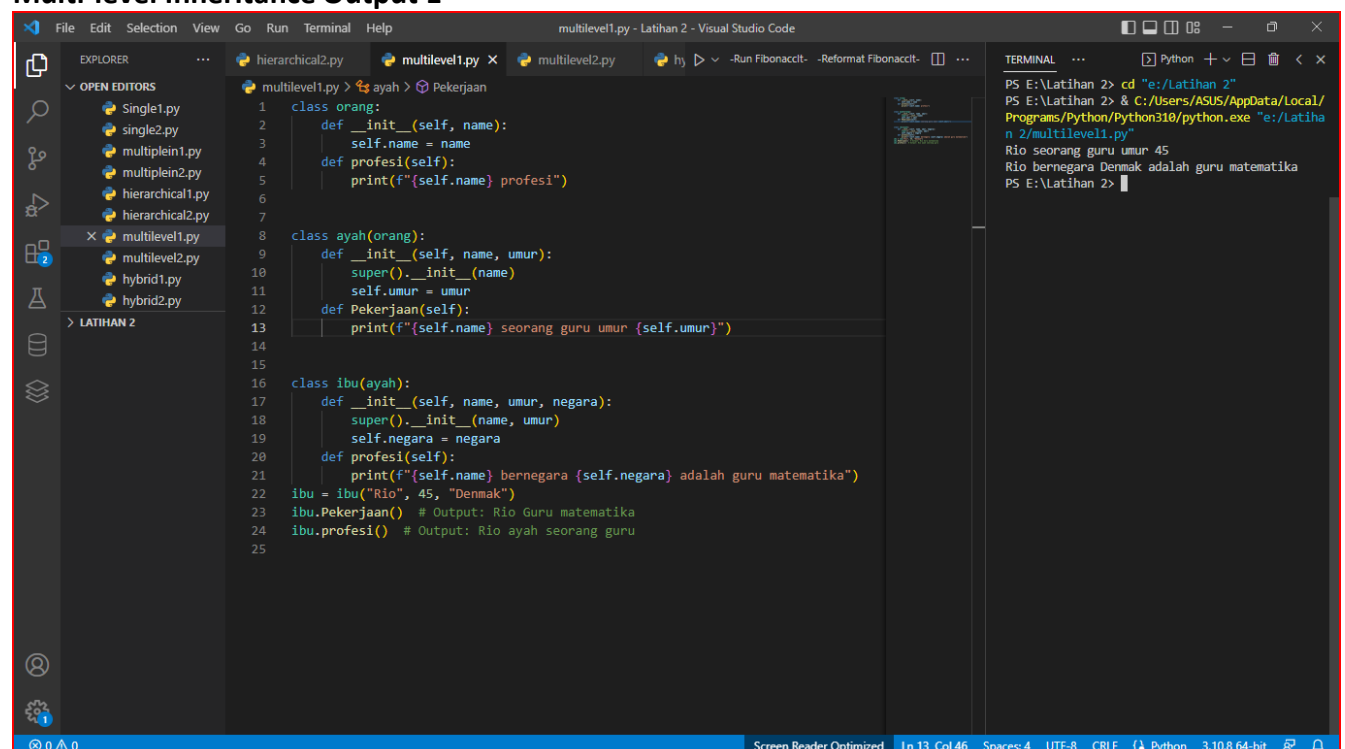
```
class orang:
    def __init__(self, name):
        self.name = name
    def profesi(self):
        print(f"{self.name} profesi")

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



```
File Edit Selection View Go Run Terminal Help
multilevel1.py - Latihan 2 - Visual Studio Code

EXPLORER
OPEN EDITORS
Single1.py
single2.py
multiplein1.py
multiplein2.py
hierarchical1.py
hierarchical2.py
X multilevel1.py
multilevel2.py
hybrid1.py
hybrid2.py
> LATIHAN 2

multilevel1.py
1 class orang:
2     def __init__(self, name):
3         self.name = name
4     def profesi(self):
5         print(f"{self.name} profesi")
6
7
8 class ayah(orang):
9     def __init__(self, name, umur):
10        super().__init__(name)
11        self.umur = umur
12    def Pekerjaan(self):
13        print(f"{self.name} seorang guru umur {self.umur}")
14
15
16 class ibu(ayah):
17     def __init__(self, name, umur, negara):
18         super().__init__(name, umur)
19         self.negara = negara
20     def profesi(self):
21         print(f"{self.name} bernegara {self.negara} adalah guru matematika")
22
23 ibu = ibu("Rio", 45, "Denmak")
24 ibu.Pekerjaan() # Output: Rio Guru matematika
25 ibu.profesi() # Output: Rio ayah seorang guru

TERMINAL
Python
PS E:\Latihan 2> cd "e:/Latihan 2"
PS E:\Latihan 2> & C:/Users/ASUS/AppData/Local/Programs/Python/Python310/python.exe "e:/Latihan 2/multilevel1.py"
Rio seorang guru umur 45
Rio bernegara Denmak adalah guru matematika
PS E:\Latihan 2>
```

Multi-level Inheritance Input 2

```
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

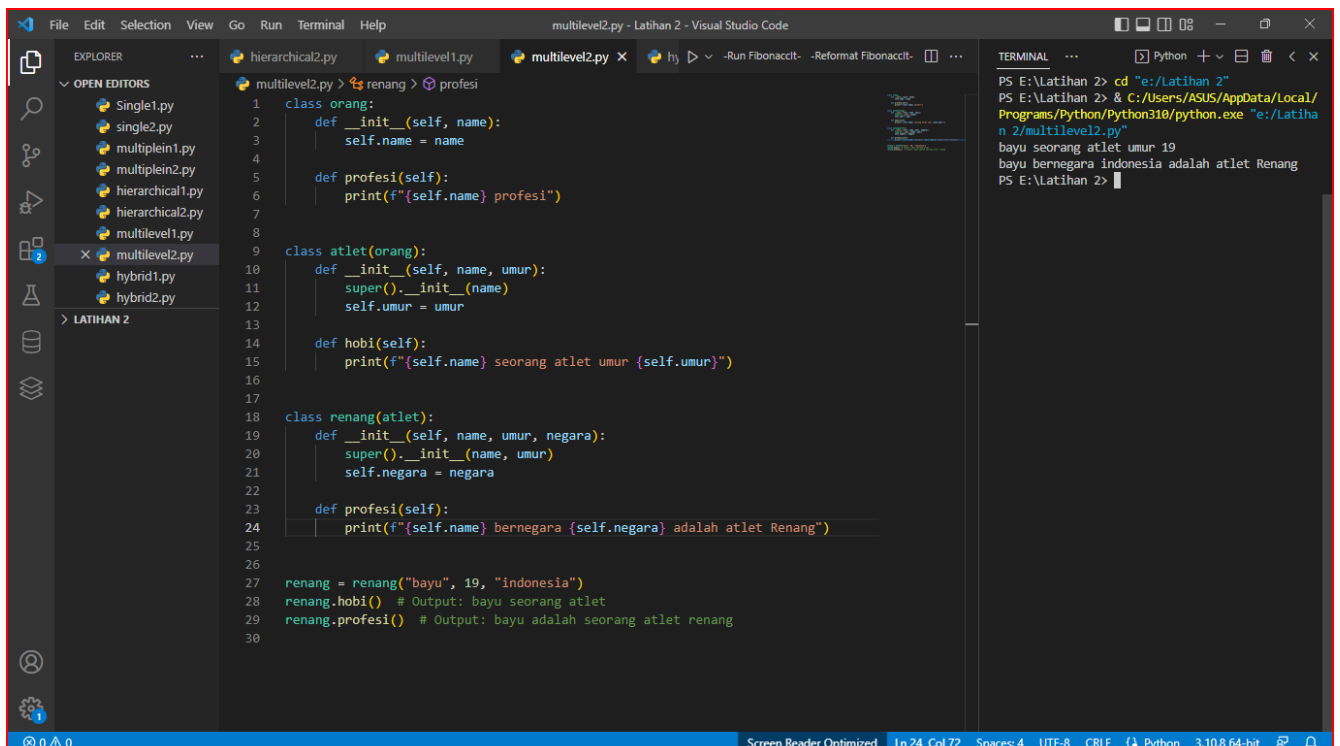
    def hobi(self):
        print(f"{self.name} seorang atlet umur {self.umur}")

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



The screenshot shows the Visual Studio Code interface with the following components:

- EXPLORER:** A list of files including `Single1.py`, `single2.py`, `multiplein1.py`, `multiplein2.py`, `hierarchical1.py`, `hierarchical2.py`, `multilevel1.py`, `multilevel2.py` (selected), `hybrid1.py`, and `hybrid2.py`. Below this is a folder named `LATIHAN 2`.
- EDITOR:** The `multilevel2.py` file is open, displaying the Python code for the multi-level inheritance classes and the execution of the `renang` object.
- TERMINAL:** The output of the code execution is shown, including the command to run the script and the resulting output: `bayu seorang atlet umur 19` and `bayu bernegara indonesia adalah atlet Renang`.

```
PS E:\Latihan 2> cd "e:/Latihan 2"
PS E:\Latihan 2> & C:/Users/ASUS/AppData/Local/Programs/Python/Python310/python.exe "e:/Latihan 2/multilevel2.py"
bayu seorang atlet umur 19
bayu bernegara indonesia adalah atlet Renang
PS E:\Latihan 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

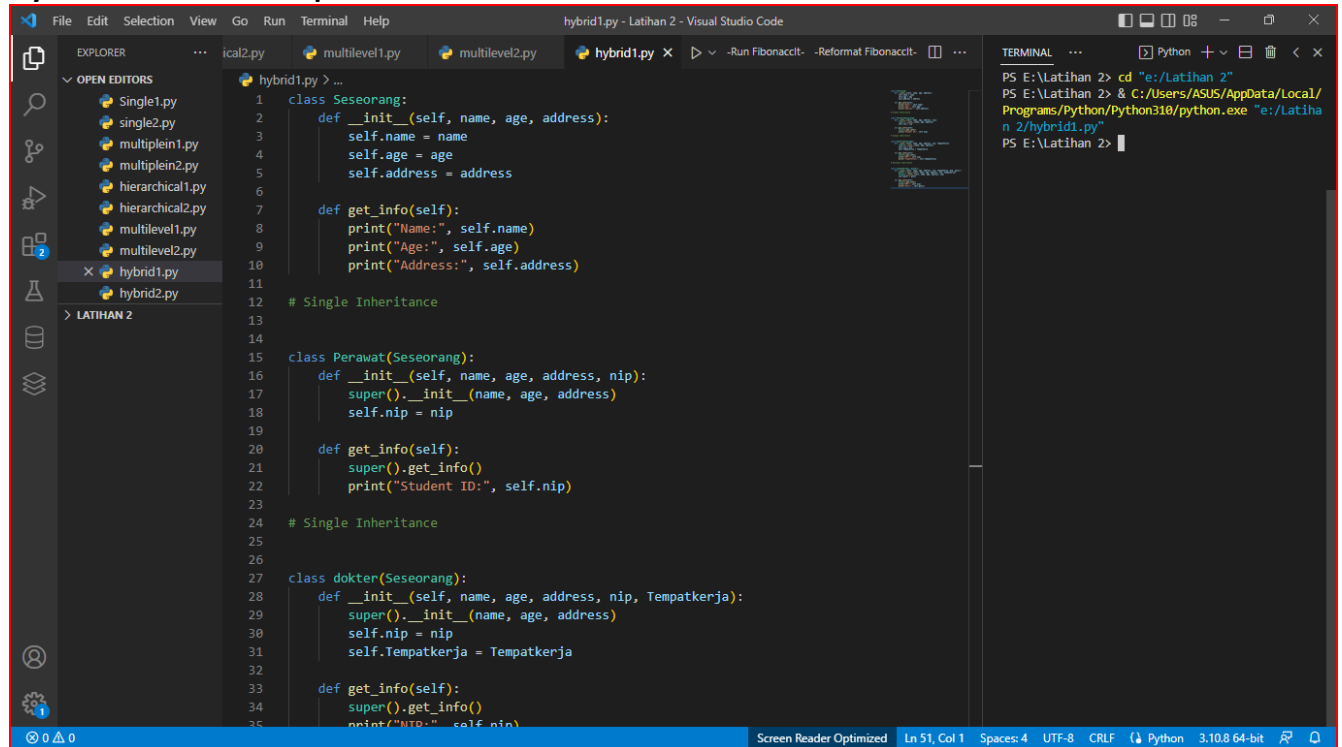
    def get_info(self):
        super().get_info()
        print("NIP:", self.nip)
        print("Tempatkerja:", self.Tempatkerja)

# 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



```
File Edit Selection View Go Run Terminal Help
hybrid1.py - Latihan 2 - Visual Studio Code

EXPLORER
... ical2.py multilevel1.py multilevel2.py hybrid1.py x
OPEN EDITORS
Single1.py
single2.py
multiplein1.py
multiplein2.py
hierarchical1.py
hierarchical2.py
multilevel1.py
multilevel2.py
x hybrid1.py
hybrid2.py
LATIHAN 2

hybrid1.py > ...
1 class Seseorang:
2     def __init__(self, name, age, address):
3         self.name = name
4         self.age = age
5         self.address = address
6
7     def get_info(self):
8         print("Name:", self.name)
9         print("Age:", self.age)
10        print("Address:", self.address)
11
12    # Single Inheritance
13
14
15    class Perawat(Seseorang):
16        def __init__(self, name, age, address, nip):
17            super().__init__(name, age, address)
18            self.nip = nip
19
20        def get_info(self):
21            super().get_info()
22            print("Student ID:", self.nip)
23
24    # Single Inheritance
25
26
27    class dokter(Seseorang):
28        def __init__(self, name, age, address, nip, Tempatkerja):
29            super().__init__(name, age, address)
30            self.nip = nip
31            self.Tempatkerja = Tempatkerja
32
33        def get_info(self):
34            super().get_info()
35            print("NIP:", self.nip)

TERMINAL
Python
P5 E:\Latihan 2> cd "e:/Latihan 2"
P5 E:\Latihan 2> & C:/Users/ASUS/AppData/Local/Programs/Python/Python310/python.exe "e:/Latihan 2/hybrid1.py"
P5 E:\Latihan 2>
```

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

```
hybrid2.py > ...
1 class tumbuhan:
2     def __init__(self, name, struktur, jenis):
3         self.name = name
4         self.struktur = struktur
5         self.jenis = jenis
6
7     def get_info(self):
8         print("Name:", self.name)
9         print("Struktur:", self.struktur)
10        print("jenis:", self.jenis)
11
12 # Single Inheritance
13
14
15 class akar(tumbuhan):
16     def __init__(self, epidermis, korteks, endodermis, stele):
17         super().__init__(epidermis, korteks, endodermis)
18         self.stele = stele
19
20     def get_info(self):
21         super().get_info()
22         print("Student ID:", self.stele)
23
24 # Single Inheritance
25
26
27 class batang(tumbuhan):
28     def __init__(self, epidermis, korteks, bekaspengangkut, stele):
29         super().__init__(epidermis, korteks, bekaspengangkut)
30         self.bekaspengangkut = bekaspengangkut
31         self.stele = stele
32
33     def get_info(self):
34         super().get_info()
35         print("Bekas Pengangkut:", self.bekaspengangkut)
```

```
PS E:\Latihan 2> cd "e:/Latihan 2"
PS E:\Latihan 2> & C:/Users/ASUS/AppData/Local/Programs/Python/Python310/python.exe "e:/Latihan 2/hybrid2.py"
PS E:\Latihan 2>
```