

LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK LANJUT

2023



Prepared By:

Umar Faqih

R2(B)

210511066

Soal Praktikum 2 Buatlah masing-masing dua jenis pewarisan di luar dari contoh yang di berikan:

1. Single Inheritance:

a. contoh 1:

```
class Mobil:
    def __init__(self, merk, warna):
        self.merk = merk
        self.warna = warna
    def mobil(self):
        print(self.merk, "mobil")

class Kucing(Mobil):
    def __init__(self, merk, warna, jenis_mobil):
        super().__init__(merk, warna)
        self.jenis_mobil = jenis_mobil
    def torsi(self):
        print("1000cc!")

kucingA = Kucing("honda", "kuning", "sedan")
kucingA.mobil()
kucingA.torsi()
```

Output

```

1 class Mobil:
2     def __init__(self, merk, warna):
3         self.merk = merk
4         self.warna = warna
5     def mobil(self):
6         print(self.merk, "mobil")
7 class Kucing(Mobil):
8     def __init__(self, merk, warna, jenis_mobil):
9         super().__init__(merk, warna)
10        self.jenis_mobil = jenis_mobil
11    def torsi(self):
12        print("1000cc!")
13 kucingA = Kucing("honda", "kuning", "sedan")
14 kucingA.mobil()
15 kucingA.torsi()

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```

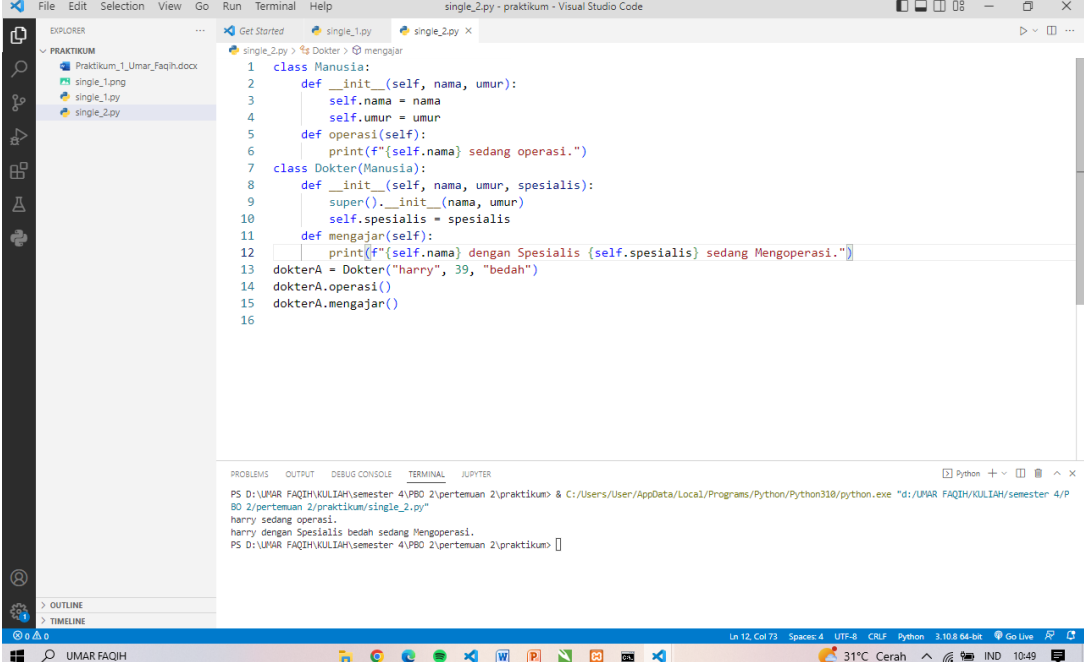
PS D:\UMAR FAQIH\KULIAH\semester 4\PBO 2\pertemuan 2\praktikum> C:\Users\User\AppData\Local\Programs\Python\Python310\python.exe "d:\UMAR FAQIH\KULIAH\semester 4\
PBO 2\pertemuan 2\praktikum\single_1.py"
honda mobil
1000cc!
PS D:\UMAR FAQIH\KULIAH\semester 4\PBO 2\pertemuan 2\praktikum>

```

b. contoh 2:

```
class Manusia:
    def __init__(self, nama, umur):
        self.nama = nama
        self.umur = umur
    def operasi(self):
        print(f"{self.nama} sedang operasi.")
class Dokter(Manusia):
    def __init__(self, nama, umur, spesialis):
        super().__init__(nama, umur)
        self.spesialis = spesialis
    def mengajar(self):
        print(f"{self.nama} dengan Spesialis {self.spesialis} sedang Mengoperasi.")
dokterA = Dokter("harry", 39, "bedah")
dokterA.operasi()
dokterA.mengajar()
```

Output



```
1 class Manusia:
2     def __init__(self, nama, umur):
3         self.nama = nama
4         self.umur = umur
5     def operasi(self):
6         print(f"{self.nama} sedang operasi.")
7 class Dokter(Manusia):
8     def __init__(self, nama, umur, spesialis):
9         super().__init__(nama, umur)
10        self.spesialis = spesialis
11    def mengajar(self):
12        print(f"{self.nama} dengan Spesialis {self.spesialis} sedang Mengoperasi.")
13 dokterA = Dokter("harry", 39, "bedah")
14 dokterA.operasi()
15 dokterA.mengajar()
16
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

Python 3.10.8 64-bit

PS D:\UMAR FAQIH\KULIAH\semester 4\PRO 2\pertemuan 2\praktikum> & C:\Users\User\AppData\Local\Programs\Python\Python310\python.exe "d:\UMAR FAQIH\KULIAH\semester 4\PRO 2\pertemuan 2\praktikum\single_2.py"

harry sedang operasi.

harry dengan Spesialis bedah sedang Mengoperasi.

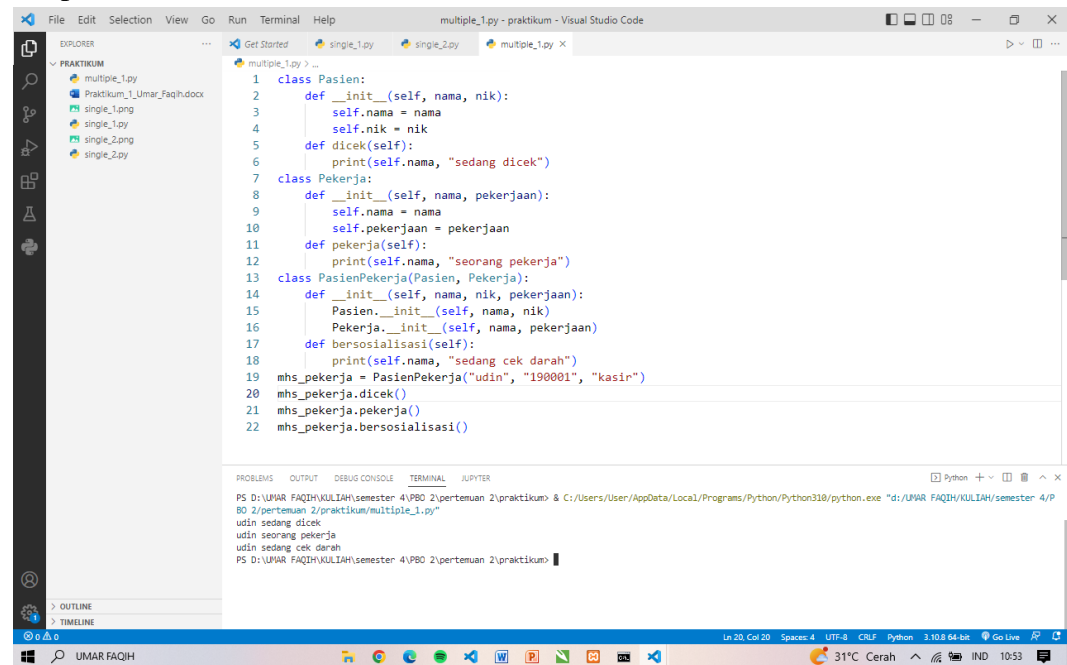
PS D:\UMAR FAQIH\KULIAH\semester 4\PRO 2\pertemuan 2\praktikum> |

2. multiple inheritance :

a. contoh 1:

```
class Pasien:
    def __init__(self, nama, nik):
        self.nama = nama
        self.nik = nik
    def dicek(self):
        print(self.nama, "sedang dicek")
class Pekerja:
    def __init__(self, nama, pekerjaan):
        self.nama = nama
        self.pekerjaan = pekerjaan
    def pekerja(self):
        print(self.nama, "seorang pekerja")
class PasienPekerja(Pasien, Pekerja):
    def __init__(self, nama, nik, pekerjaan):
        Pasien.__init__(self, nama, nik)
        Pekerja.__init__(self, nama, pekerjaan)
    def bersosialisasi(self):
        print(self.nama, "sedang cek darah")
mhs_pekerja = PasienPekerja("udin", "190001",
"kasir")
mhs_pekerja.dicek()
mhs_pekerja.pekerja()
mhs_pekerja.bersosialisasi()
```

Output



The screenshot shows a Visual Studio Code window with a file explorer on the left containing a folder named 'PRAKTIKUM' with files 'multiple_1.py', 'praktikum_1_Umar_Faqih.docx', 'single_1.py', 'single_2.py', 'single_1.png', 'single_2.png', and 'single_2.py'. The main editor displays the code for 'multiple_1.py'.

```
1 class Pasien:
2     def __init__(self, nama, nik):
3         self.nama = nama
4         self.nik = nik
5     def dicek(self):
6         print(self.nama, "sedang dicek")
7 class Pekerja:
8     def __init__(self, nama, pekerjaan):
9         self.nama = nama
10        self.pekerjaan = pekerjaan
11    def pekerja(self):
12        print(self.nama, "seorang pekerja")
13 class PasienPekerja(Pasien, Pekerja):
14     def __init__(self, nama, nik, pekerjaan):
15         Pasien.__init__(self, nama, nik)
16         Pekerja.__init__(self, nama, pekerjaan)
17     def bersosialisasi(self):
18         print(self.nama, "sedang cek darah")
19 mhs_pekerja = PasienPekerja("udin", "190001", "kasir")
20 mhs_pekerja.dicek()
21 mhs_pekerja.pekerja()
22 mhs_pekerja.bersosialisasi()
```

The terminal at the bottom shows the execution output:

```
PS D:\UMAR FAQIH\KULIAH\semester 4\pbo 2\pertemuan 2\praktikum> python multiple_1.py
udin sedang dicek
udin seorang pekerja
udin sedang cek darah
PS D:\UMAR FAQIH\KULIAH\semester 4\pbo 2\pertemuan 2\praktikum>
```

b. contoh 2:

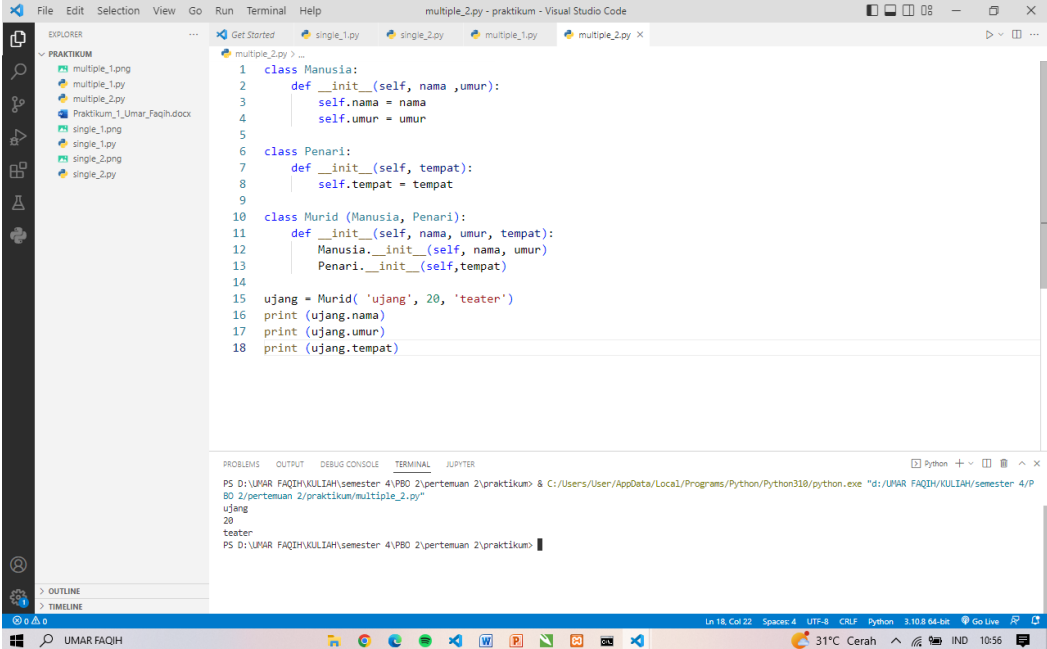
```
class Manusia:
    def __init__(self, nama ,umur):
        self.nama = nama
        self.umur = umur

class Penari:
    def __init__(self, tempat):
        self.tempat = tempat

class Murid (Manusia, Penari):
    def __init__(self, nama, umur, tempat):
        Manusia.__init__(self, nama, umur)
        Penari.__init__(self, tempat)

ujang = Murid( 'ujang', 20, 'teater')
print (ujang.nama)
print (ujang.umur)
print (ujang.tempat)
```

Output



The screenshot shows the Visual Studio Code interface with a Python file named `multiple_2.py` open. The code defines three classes: `Manusia`, `Penari`, and `Murid`. `Manusia` has attributes `nama` and `umur`. `Penari` has an attribute `tempat`. `Murid` inherits from both `Manusia` and `Penari`. An instance `ujang` of the `Murid` class is created with the values 'ujang', 20, and 'teater'. The code prints the attributes of `ujang`.

```
1 class Manusia:
2     def __init__(self, nama ,umur):
3         self.nama = nama
4         self.umur = umur
5
6 class Penari:
7     def __init__(self, tempat):
8         self.tempat = tempat
9
10 class Murid (Manusia, Penari):
11     def __init__(self, nama, umur, tempat):
12         Manusia.__init__(self, nama, umur)
13         Penari.__init__(self,tempat)
14
15 ujang = Murid( 'ujang', 20, 'teater')
16 print (ujang.nama)
17 print (ujang.umur)
18 print (ujang.tempat)
```

The output window shows the execution of the code, displaying the attributes of the `ujang` object:

```
PS D:\UMAR FAQIH\KULIAH\semester 4\PRO 2\pertemuan 2\praktikum> & C:/Users/User/AppData/Local/Programs/Python/Python318/python.exe "d:/UMAR FAQIH/KULIAH/semester 4/P
RO 2/pertemuan 2/praktikum/multiple_2.py"
ujang
20
teater
PS D:\UMAR FAQIH\KULIAH\semester 4\PRO 2\pertemuan 2\praktikum>
```

3. hierarchial Inheritance :

a. contoh 1:

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

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

class Serigala(Hewan):
    def __init__(self, nama):
        super().__init__(nama)

    def suara(self):
        print(f"{self.nama} Auuuuu!!")

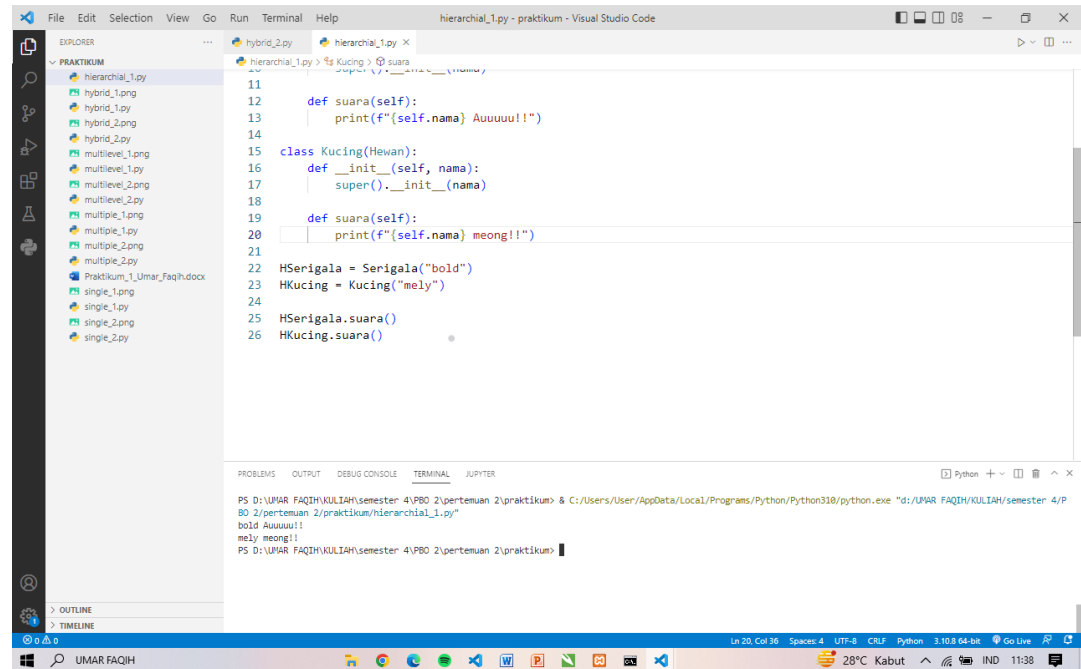
class Kucing(Hewan):
    def __init__(self, nama):
        super().__init__(nama)

    def suara(self):
        print(f"{self.nama} meong!!")

HSerigala = Serigala("bold")
HKucing = Kucing("mely")

HSerigala.suara()
HKucing.suara()
```

Output



```
11 class Kucing:
12     def suara(self):
13         print(f"{self.nama} Auuuuu!!")
14
15 class Kucing(Hewan):
16     def __init__(self, nama):
17         super().__init__(nama)
18
19     def suara(self):
20         print(f"{self.nama} meong!!")
21
22 HSerigala = Serigala("bold")
23 HKucing = Kucing("mely")
24
25 HSerigala.suara()
26 HKucing.suara()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

PS D:\UPAR FAQIH\KULIAH\semester 4\pbo 2\pertemuan 2\praktikum> C:\Users\User\AppData\Local\Programs\Python\Python310\python.exe "d:\UPAR FAQIH\KULIAH\semester 4\pbo 2\pertemuan 2\praktikum\hierarchia_1.py"

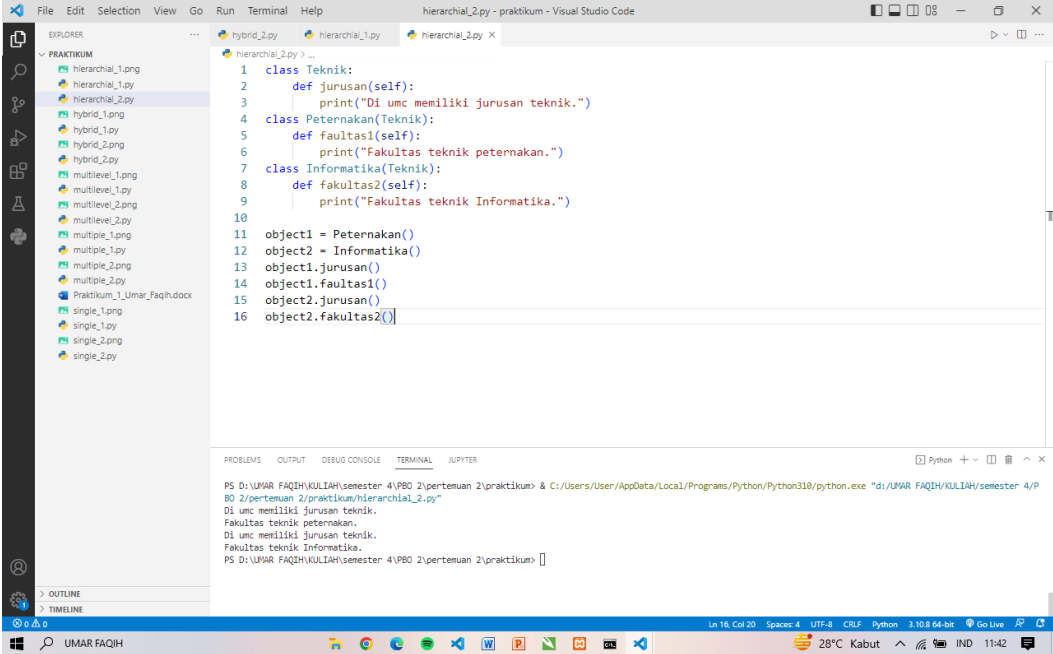
bold Auuuuu!!
mely meong!!
mely meong!!

PS D:\UPAR FAQIH\KULIAH\semester 4\pbo 2\pertemuan 2\praktikum>

b. contoh 2:

```
class Teknik:
    def jurusan(self):
        print("Di umc memiliki jurusan teknik.")
class Peternakan(Teknik):
    def faultas1(self):
        print("Fakultas teknik peternakan.")
class Informatika(Teknik):
    def fakultas2(self):
        print("Fakultas teknik Informatika.")
object1 = Peternakan()
object2 = Informatika()
object1.jurusan()
object1.faultas1()
object2.jurusan()
object2.fakultas2()
```


Output



The screenshot shows the Visual Studio Code interface with a Python file named `hierarchial_2.py` open. The file contains a class hierarchy and object creation. The Explorer sidebar on the left shows a project named `praktikum` with various image and Python files. The Output window at the bottom displays the execution results of the script.

```
1 class Teknik:
2     def jurusan(self):
3         print("Di umc memiliki jurusan teknik.")
4 class Peternakan(Teknik):
5     def faultas1(self):
6         print("Fakultas teknik peternakan.")
7 class Informatika(Teknik):
8     def fakultas2(self):
9         print("Fakultas teknik Informatika.")
10
11 object1 = Peternakan()
12 object2 = Informatika()
13 object1.jurusan()
14 object1.faultas1()
15 object2.jurusan()
16 object2.fakultas2()
```

Output:

```
PS D:\UMAR FAQIH\KULIAH\semester 4\PBO 2\pertemuan 2\praktikum> & C:/Users/User/AppData/Local/Programs/Python/Python310/python.exe "d:/UMAR FAQIH/KULIAH/semester 4/PBO 2/pertemuan 2/praktikum/hierarchial_2.py"
Di umc memiliki jurusan teknik.
Fakultas teknik peternakan.
Di umc memiliki jurusan teknik.
Fakultas teknik Informatika.
PS D:\UMAR FAQIH\KULIAH\semester 4\PBO 2\pertemuan 2\praktikum>
```

4. Multi_level inheritance :

a. Contoh 1:

```
class Hewan:
    def __init__(self, spesies):
        self.spesies = spesies

    def makan(self):
        print("hewan sedang makan!!!")

class peliharaan(Hewan):
    def __init__(self, nama, spesies):
        super().__init__(spesies)
        self.nama = nama

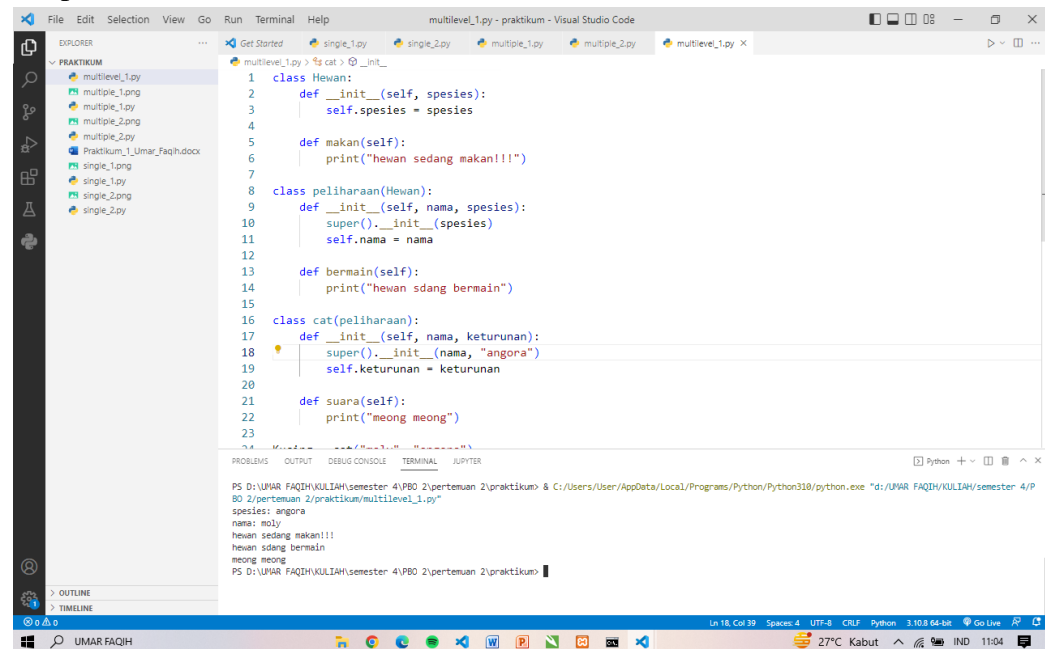
    def bermain(self):
        print("hewan sdang bermain")

class cat(peliharaan):
    def __init__(self, nama, keturunan):
        super().__init__(nama, "angora")
        self.keturunan = keturunan

    def suara(self):
        print("meong meong")

Kucing = cat("moly", "angora")
print("spesies:", Kucing.spesies)
print("nama:", Kucing.nama)
Kucing.makan()
Kucing.bermain()
Kucing.suara()
```

Output



The screenshot shows a Visual Studio Code window with a file explorer on the left and a code editor in the center. The file explorer shows a project named 'PRAKTIKUM' with several files including 'multilevel_1.py', 'multilevel_1.png', 'multilevel_2.png', 'Praktikum_1_Umar_Faqih.docx', 'single_1.png', 'single_2.png', and 'single_2.py'. The code editor displays the following Python code:

```
1 class Hewan:
2     def __init__(self, spesies):
3         self.spesies = spesies
4
5     def makan(self):
6         print("hewan sedang makan!!!")
7
8 class peliharaan(Hewan):
9     def __init__(self, nama, spesies):
10        super().__init__(spesies)
11        self.nama = nama
12
13    def bermain(self):
14        print("hewan sdang bermain")
15
16 class cat(peliharaan):
17     def __init__(self, nama, keturunan):
18         super().__init__(nama, "angora")
19         self.keturunan = keturunan
20
21     def suara(self):
22         print("meong meong")
23
```

The terminal at the bottom shows the output of the script:

```
PS D:\UMAR FAQIH\KULIAH\semester 4\PBO 2\pertemuan 2\praktikum> python.exe "d:/UMAR FAQIH/KULIAH/semester 4/PBO 2/pertemuan 2/praktikum/multilevel_1.py"
spesies: angora
nama: moly
hewan sedang makan!!!
hewan sdang bermain
meong meong
PS D:\UMAR FAQIH\KULIAH\semester 4\PBO 2\pertemuan 2\praktikum>
```

b. Contoh 2:

```
class Kendaraan:
    def __init__(self, warna, jenis):
        self.warna = warna
        self.jenis = jenis

class Mobil(Kendaraan):
    def __init__(self, warna, jenis, kecepatan):
        super().__init__(warna, jenis)
        self.kecepatan = kecepatan

    def berjalan(self):
        print(f"mobil berwarna {self.warna} di kendarai {self.kecepatan} km/h.")

class MobilListrik(Mobil):
    def __init__(self, warna, jenis, kecepatan, kapasitas_batre):
        super().__init__(warna, jenis, kecepatan)
        self.kapasitas_batre = kapasitas_batre
```

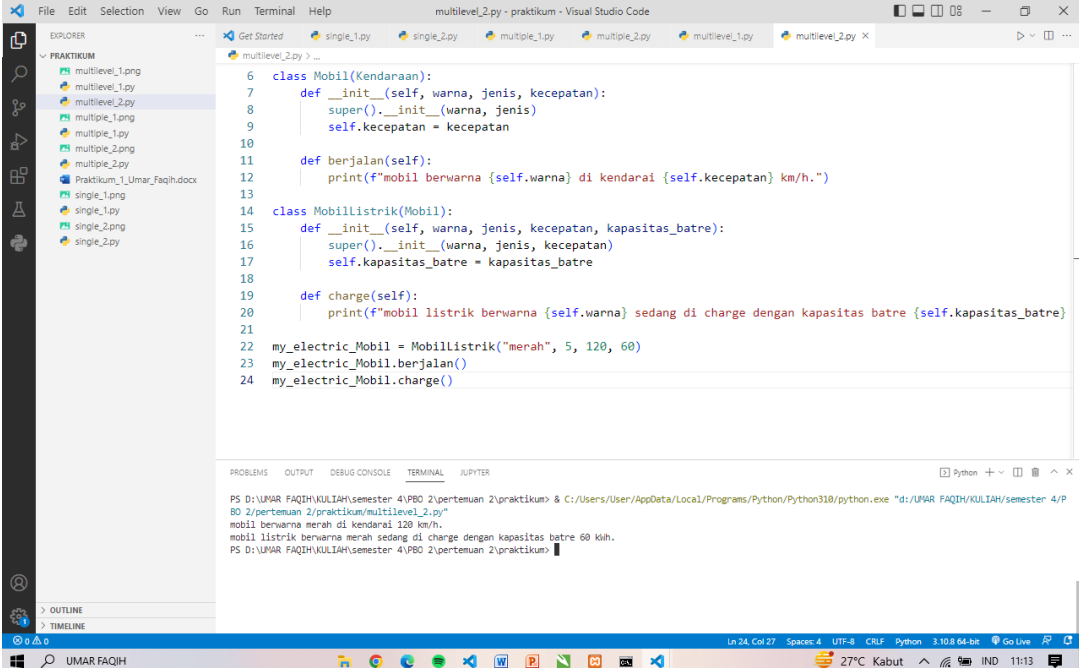
```

def charge(self):
    print(f"mobil listrik berwarna {self.warna}
sedang di charge dengan kapasitas batre
{self.kapasitas_batre} kWh.")

my_electric_Mobil = MobilListrik("merah", 5, 120, 60)
my_electric_Mobil.berjalan()
my_electric_Mobil.charge()

```

Output



The screenshot shows a Visual Studio Code window with a Python file named 'multilevel_2.py'. The code defines a base class 'Mobil(Kendaraan)' with an '__init__' method and a 'berjalan' method. It also defines a subclass 'MobilListrik(Mobil)' with an '__init__' method and a 'charge' method. The script creates an instance 'my_electric_Mobil' and calls 'berjalan()' and 'charge()'. The terminal output shows the execution results: 'mobil berwarna merah di kendalai 120 km/h.' and 'mobil listrik berwarna merah sedang di charge dengan kapasitas batre 60 Kwh.'

```

6 class Mobil(Kendaraan):
7     def __init__(self, warna, jenis, kecepatan):
8         super().__init__(warna, jenis)
9         self.kecepatan = kecepatan
10
11     def berjalan(self):
12         print(f"mobil berwarna {self.warna} di kendalai {self.kecepatan} km/h.")
13
14 class MobilListrik(Mobil):
15     def __init__(self, warna, jenis, kecepatan, kapasitas_batre):
16         super().__init__(warna, jenis, kecepatan)
17         self.kapasitas_batre = kapasitas_batre
18
19     def charge(self):
20         print(f"mobil listrik berwarna {self.warna} sedang di charge dengan kapasitas batre {self.kapasitas_batre}
21
22 my_electric_Mobil = MobilListrik("merah", 5, 120, 60)
23 my_electric_Mobil.berjalan()
24 my_electric_Mobil.charge()

```

```

PS D:\UMAR FAQIH\KULIAH\semester 4\PRO 2\pertemuan 2\praktikum> & C:/Users/User/AppData/Local/Programs/Python/Python310/python.exe "d:/UMAR FAQIH/KULIAH/semester 4/P
RO 2/pertemuan 2/praktikum/multilevel_2.py"
mobil berwarna merah di kendalai 120 km/h.
mobil listrik berwarna merah sedang di charge dengan kapasitas batre 60 Kwh.
PS D:\UMAR FAQIH\KULIAH\semester 4\PRO 2\pertemuan 2\praktikum>

```

5. Hybrid inheritance :

a. Contoh 1

```

class Mengajar:
    def mengajar(self):
        print("Mengajar")

class Inggris(Mengajar):

```

```

def ygdiajar(self):
    print("Inggris")

class Indonesia(Mengajar):
    def ygdiajar2(self):
        print("Indonesia")

class Guru(Inggris, Indonesia):
    def pangkat(self):
        print("guru")

GuruA = Guru()
GuruA.mengajar()
GuruA.ygdiajar()
GuruA.ygdiajar2()
GuruA.pangkat()

```

Output

```

class Mengajar:
    def mengajar(self):
        print("Mengajar")

class Inggris(Mengajar):
    def ygdiajar(self):
        print("Inggris")

class Indonesia(Mengajar):
    def ygdiajar2(self):
        print("Indonesia")

class Guru(Inggris, Indonesia):
    def pangkat(self):
        print("guru")

GuruA = Guru()
GuruA.mengajar()
GuruA.ygdiajar()
GuruA.ygdiajar2()
GuruA.pangkat()

```

Mengajar
 Inggris
 Indonesia
 guru

b. Contoh 2

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

    def suara(self):
        pass

class Serigala(Hewan):
    def suara(self):
        return "Auuuuuu!!"

class Kucing(Hewan):
    def suara(self):
        return "Meong!!"

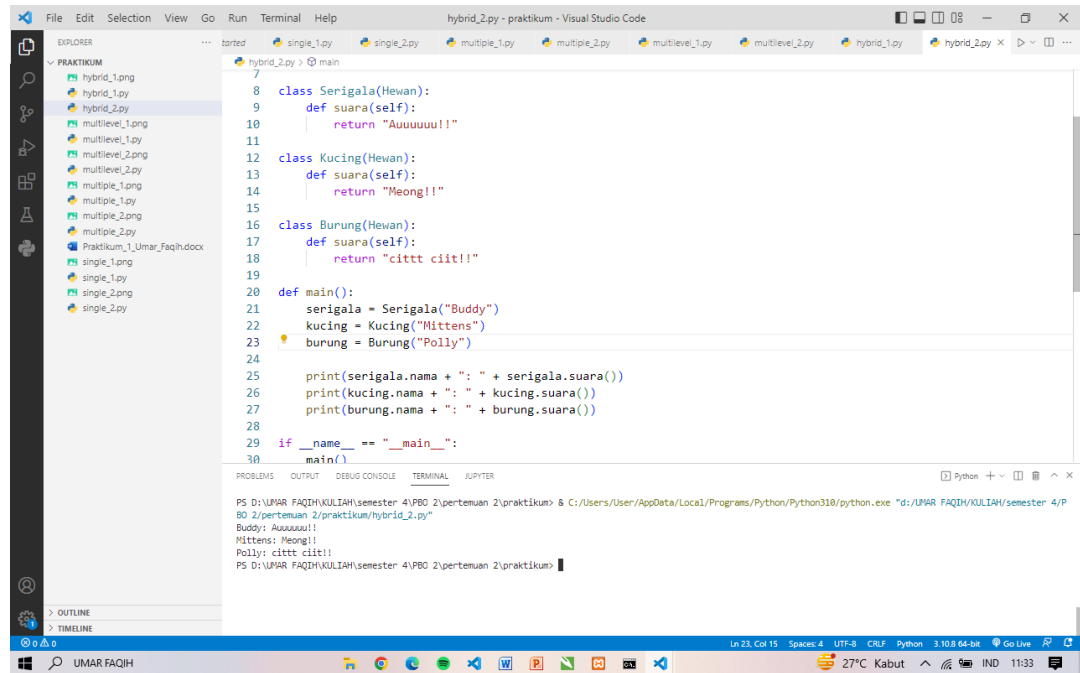
class Burung(Hewan):
    def suara(self):
        return "cittt ciit!!"

def main():
    serigala = Serigala("Buddy")
    kucing = Kucing("Mittens")
    burung = Burung("Polly")

    print(serigala.nama + ": " + serigala.suara())
    print(kucing.nama + ": " + kucing.suara())
    print(burung.nama + ": " + burung.suara())

if __name__ == "__main__":
    main()
```

Output



The screenshot displays the Visual Studio Code interface with a Python file named `hybrid_2.py` open. The Explorer sidebar on the left shows a project named "PRAKTIKUM" containing various files, including `hybrid_1.png`, `hybrid_1.py`, `hybrid_2.py`, `multilevel_1.png`, `multilevel_1.py`, `multiple_1.png`, `multiple_1.py`, `multiple_2.png`, `multiple_2.py`, `Praktikum_1_Umar_Faqih.docx`, `single_1.png`, `single_1.py`, and `single_2.py`.

The main editor window shows the following Python code:

```
7
8 class Serigala(Hewan):
9     def suara(self):
10         return "Auuuuuu!!"
11
12 class Kucing(Hewan):
13     def suara(self):
14         return "Meong!!"
15
16 class Burung(Hewan):
17     def suara(self):
18         return "cittt ciit!!"
19
20 def main():
21     serigala = Serigala("Buddy")
22     kucing = Kucing("Mittens")
23     burung = Burung("Polly")
24
25     print(serigala.nama + ": " + serigala.suara())
26     print(kucing.nama + ": " + kucing.suara())
27     print(burung.nama + ": " + burung.suara())
28
29 if __name__ == "__main__":
30     main()
```

The TERMINAL panel at the bottom shows the execution output:

```
PS D:\UMAR FAQIH\KULIAH\semester 4\PBO 2\pertemuan 2\praktikum> & C:/Users/User/AppData/Local/Programs/Python/Python310/python.exe "d:/UMAR FAQIH/KULIAH/semester 4/PBO 2/pertemuan 2/praktikum/hybrid_2.py"
Buddy: Auuuuuu!!
Mittens: Meong!!
Polly: cittt ciit!!
PS D:\UMAR FAQIH\KULIAH\semester 4\PBO 2\pertemuan 2\praktikum>
```

The status bar at the bottom indicates the file is at line 23, column 15, using UTF-8 encoding with CRLF line endings, and is running Python 3.10.8 64-bit. The system tray shows the temperature as 27°C in Kabut, with the time set to 11:33.