

LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK LANJUT

2023



Prepared By:

Rian Saputra

R2(B)

210511056

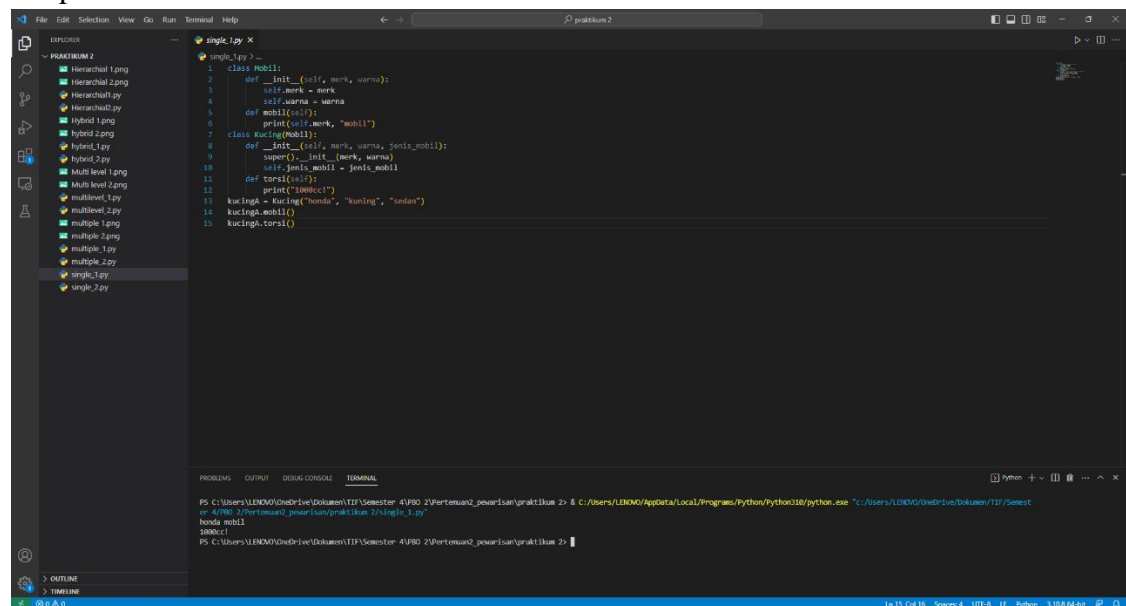
Soal Praktikum 2 Buatlah masing-masing dua jenis pewarisan di luar dari contoh yang di berikan:

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:

The screenshot shows a Python IDE with a file explorer on the left containing various files like 'Hierarchal 1.png', 'Hierarchal 2.png', etc. The main editor window shows the same Python code as in the previous block. The terminal at the bottom displays the output of the code execution: 'honda mobil' and '1000cc!'.

b. Contoh 2:

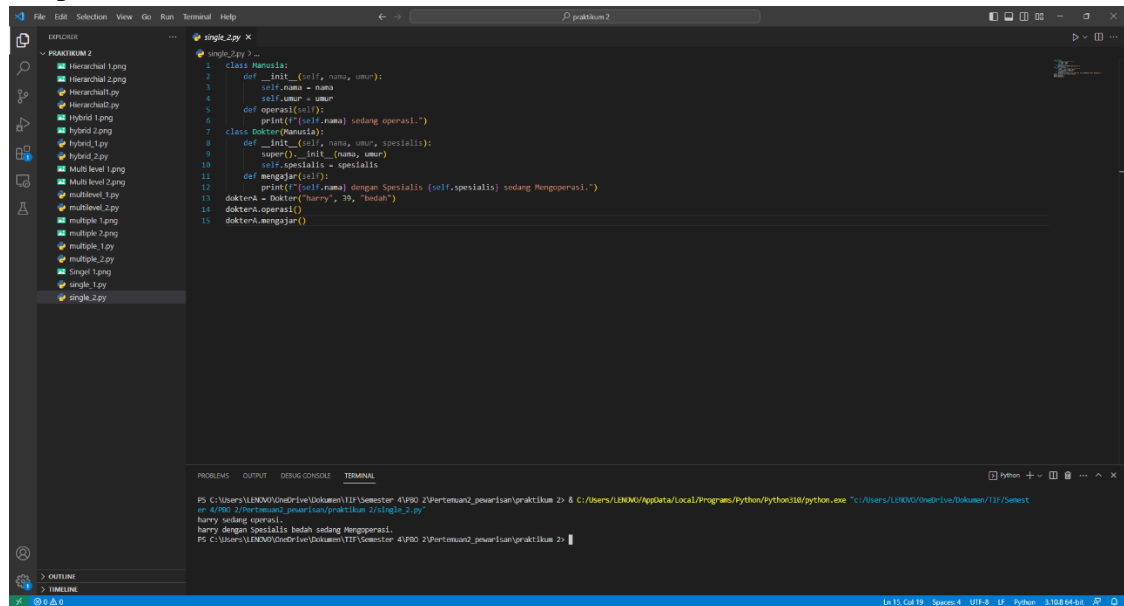
```
class Manusia:
    def __init__(self, nama, umur):
        self.nama = nama
        self.umur = umur
    def operasi(self):
        print(f"{self.nama} sedang operasi.")
```

```

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:



```

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:

```

harry sedang operasi.
harry dengan Spesialis bedah sedang Mengoperasi.

```

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:

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:

```
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)
```

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!!")

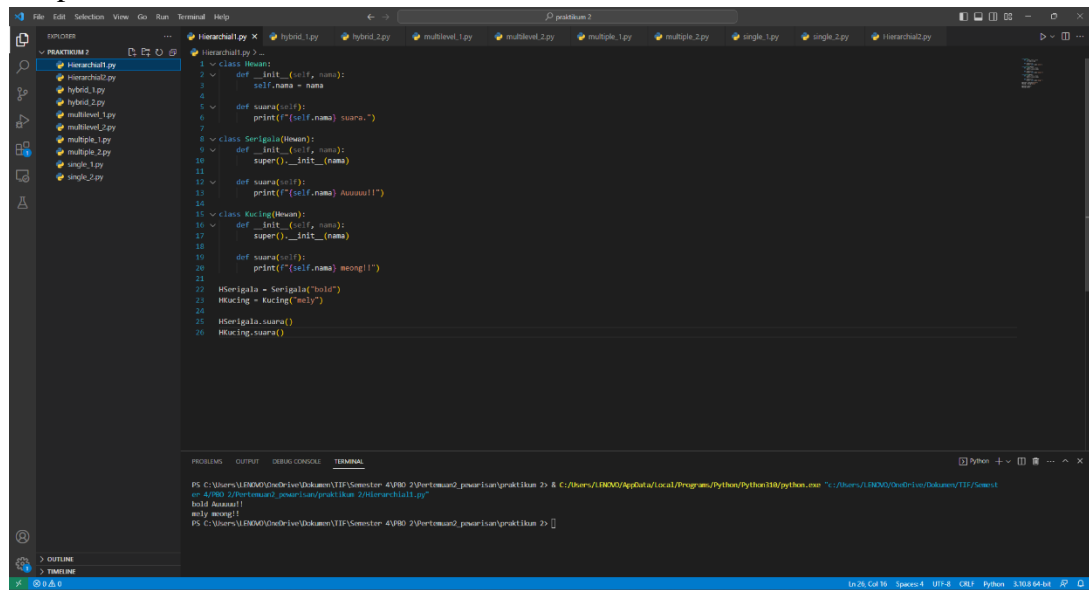
class Kucing(Hewan):
    def __init__(self, nama):
        super().__init__(nama)

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

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

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

Output:

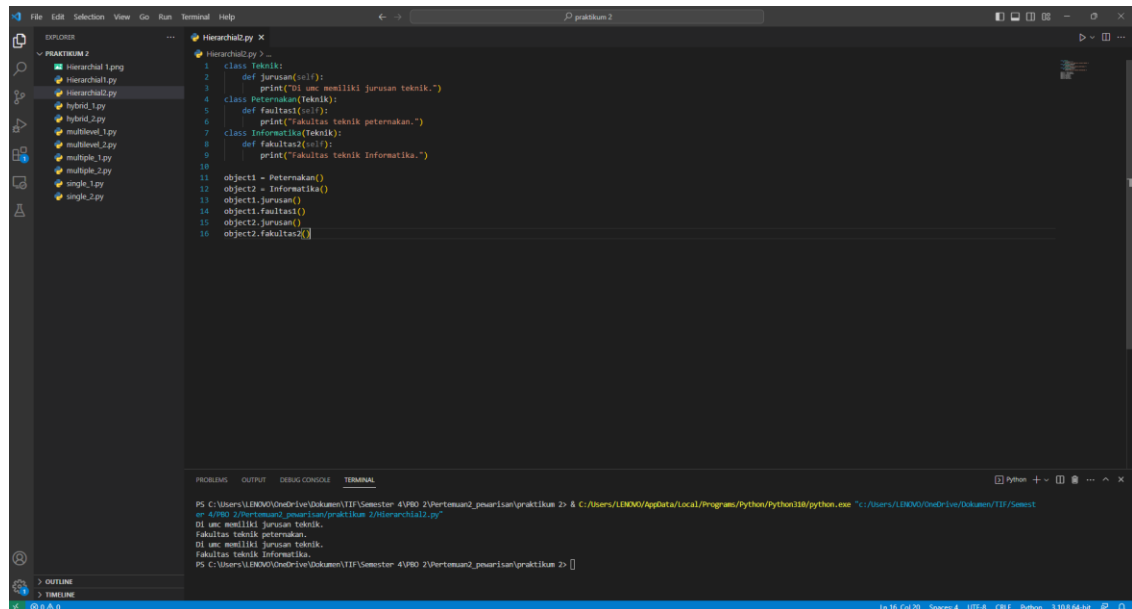


b. Contoh 2:

```
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:



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")

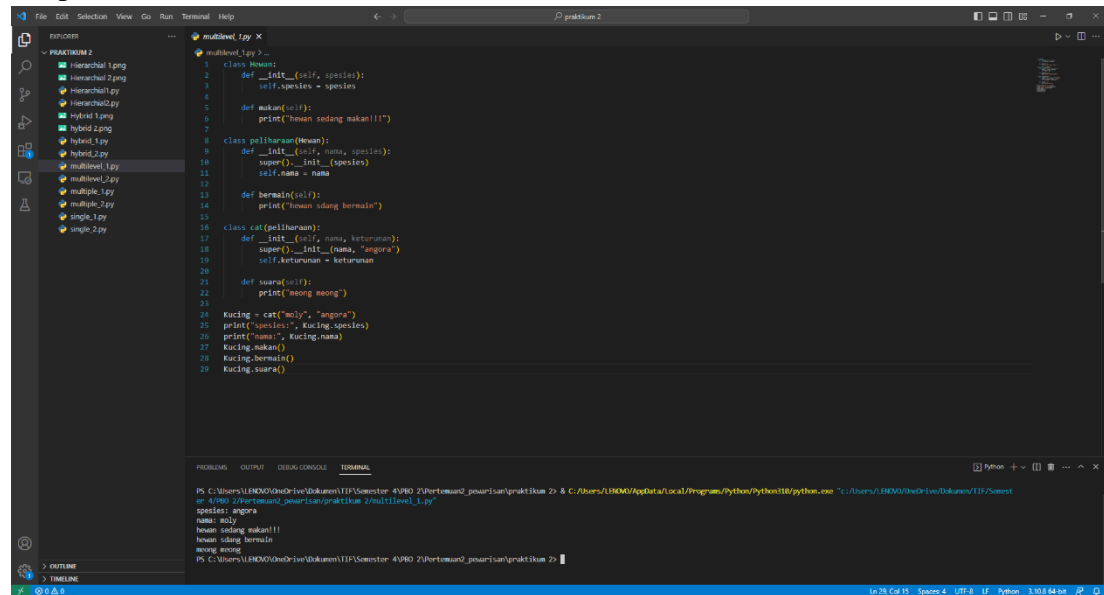
class cat(peliharaan):
    def __init__(self, nama, keturunan):
        super().__init__(nama, "angora")
        self.keturunan = keturunan

    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 Python IDE with a file explorer on the left containing files like 'Hierarchal 1.png', 'Hierarchal 2.png', 'Hybrid 1.png', 'Hybrid 2.png', 'multiple 1.py', 'multiple 2.py', 'single 1.py', and 'single 2.py'. The main editor displays a Python script named 'multilevel.py' with the following 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 poliharaam(Hewan):
9     def __init__(self, nama, spesies):
10        super().__init__(spesies)
11        self.nama = nama
12
13    def bermain(self):
14        print("hewan sedang bermain")
15
16 class cat(poliharaam):
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
24 Kucing = cat("moly", "angora")
25 print("spesies:", Kucing.spesies)
26 print("nama:", Kucing.nama)
27 Kucing.makan()
28 Kucing.bermain()
29 Kucing.suara()
```

The output console at the bottom shows the execution results:

```
PS C:\Users\LENOVO\OneDrive\Documents\TIP\Semester 4\INFO 2\Portoman2_pwarisan\praktikum 2> python multilevel_1.py
spesies: angora
nama: moly
hewan sedang makan!!!
hewan sedang bermain
meong meong
PS C:\Users\LENOVO\OneDrive\Documents\TIP\Semester 4\INFO 2\Portoman2_pwarisan\praktikum 2>
```

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:

```

1 class kendaraan:
2     def __init__(self, warna, jenis):
3         self.warna = warna
4         self.jenis = jenis
5
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 kendal {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} kWh.")
21
22 my_electric_Mobil = MobilListrik("merah", 5, 120, 60)
23 my_electric_Mobil.berjalan()
24 my_electric_Mobil.charge()

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```

PS C:\Users\LENOVO\OneDrive\Documents\TIF\Semester 4\PRO 2\Pertemuan2_pemrisan\praktikum 2> python 2.py
Mobil berwarna merah di kendal 120 km/h.
Mobil listrik berwarna merah sedang di charge dengan kapasitas batre 60 kWh.
PS C:\Users\LENOVO\OneDrive\Documents\TIF\Semester 4\PRO 2\Pertemuan2_pemrisan\praktikum 2>

```

5. Hybrid inheritance

a. Contoh 1:

```

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

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

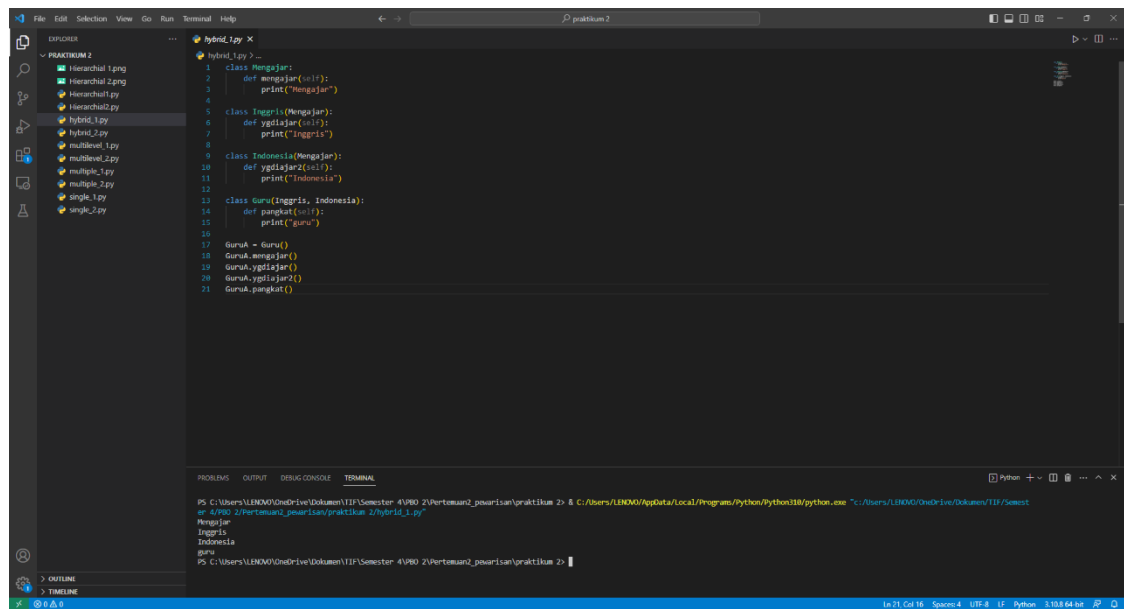
class Indonesia(Mengajar):
    def ygdiajar2(self):
        print("Indonesia")

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

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

```

Output:



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 "ciitt ciit!!"

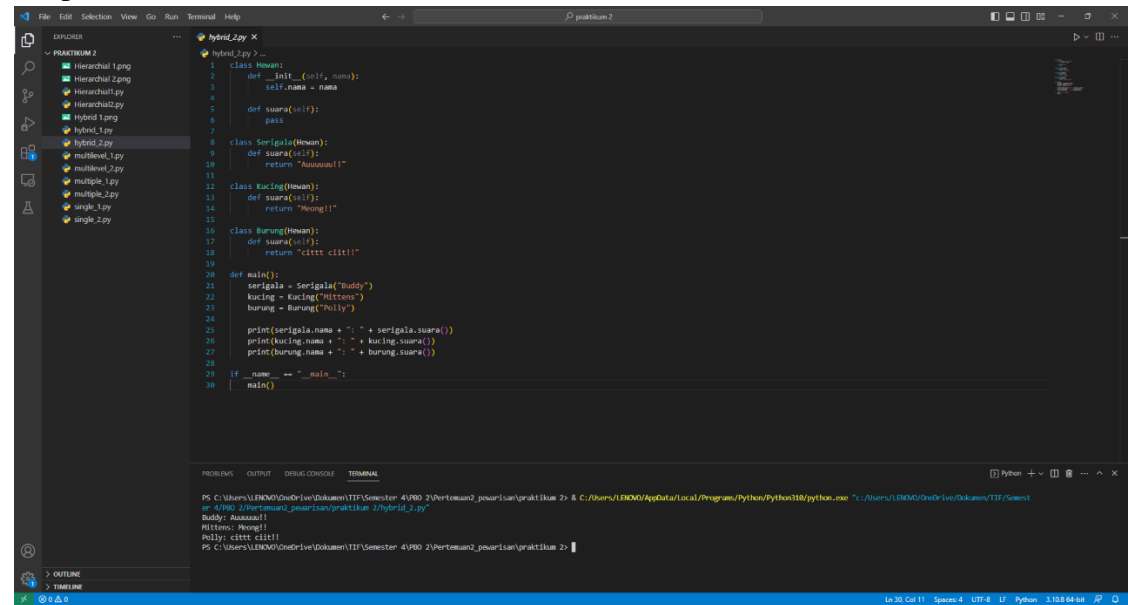
def main():
    serigala = Serigala("Buddy")
    kucing = Kucing("Mittens")
    burung = Burung("Polly")

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

```

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

Output:



The screenshot shows a Python IDE with a file explorer on the left, a code editor in the center, and a terminal at the bottom. The file explorer shows a project named 'PRAKTIKUM 2' with several files, including 'hybrid_2.py'. The code editor displays the following Python code:

```
1 class Hewan:  
2     def __init__(self, nama):  
3         self.nama = nama  
4  
5     def suara(self):  
6         pass  
7  
8 class Serigala(Hewan):  
9     def suara(self):  
10        return "Aaaaaa!!!"  
11  
12 class Kucing(Hewan):  
13     def suara(self):  
14        return "Meong!!!"  
15  
16 class Burung(Hewan):  
17     def suara(self):  
18        return "cittt cttt!!!"  
19  
20 def main():  
21     serigala = Serigala("Ruddy")  
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 at the bottom shows the output of the program:

```
PS C:\Users\LENOVO\OneDrive\Documents\Semester 4\PRO 2\pertemuan2_pemrisan\praktikum 2> & C:\Users\LENOVO\AppData\Local\Programs\Python\Python38\python.exe "C:\Users\LENOVO\OneDrive\Documents\Semester 4\PRO 2\pertemuan2_pemrisan\praktikum 2\hybrid_2.py"  
Ruddy: Aaaaaa!!!  
Mittens: Meong!!!  
Polly: cttt cttt!!!  
PS C:\Users\LENOVO\OneDrive\Documents\Semester 4\PRO 2\pertemuan2_pemrisan\praktikum 2>
```