

LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK LANJUT

2023



Prepared By:

Nama : Ryan taufiq nurdiansyah fauji

Kelas : TIF21B R2

Nim : 210511048

Praktikum 2

Buatlah masing-masing 2 contoh jenis pewarisan di luar dari contoh yang telah diberikan, beri nama:

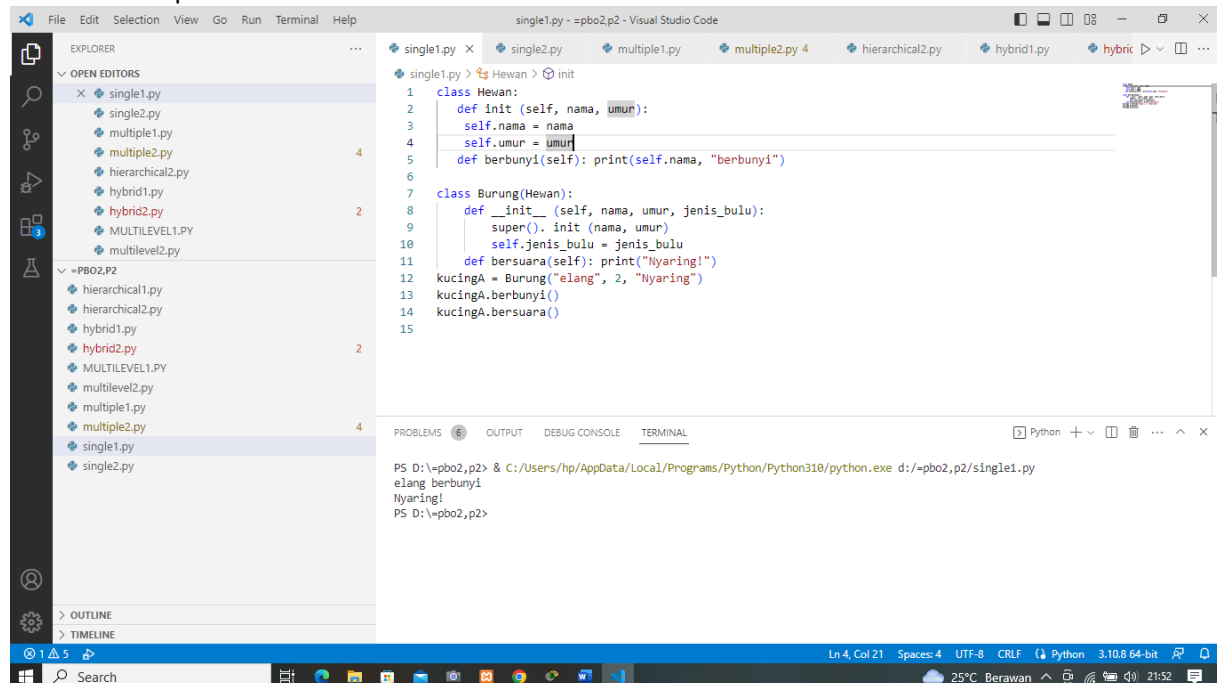
1. Single1

```
class Hewan:
    def init (self, nama, umur):
        self.nama = nama
        self.umur = umur
    def berbunyi(self): print(self.nama, "berbunyi")

class Burung(Hewan):
    def __init__ (self, nama, umur, jenis_bulu):
        super(). init (nama, umur)
        self.jenis_bulu = jenis_bulu
    def bersuara(self): print("Nyaring!")

kucingA = Burung("elang", 2, "Nyaring")
kucingA.berbunyi()
kucingA.bersuara()
```

Output SS



The screenshot shows the Visual Studio Code interface with a Python file named `single1.py` open. The code defines a base class `Hewan` and a derived class `Burung`. The `Burung` class inherits from `Hewan` and overrides the `__init__` and `bersuara` methods. The `__init__` method of `Burung` calls `super().init(nama, umur)` to initialize the `nama` and `umur` attributes from the `Hewan` class. The `bersuara` method of `Burung` prints "Nyaring!". The `init` method of `Hewan` sets `self.nama` and `self.umur`. The `berbunyi` method of `Hewan` prints the `nama` attribute followed by "berbunyi".

```
1 class Hewan:
2     def init (self, nama, umur):
3         self.nama = nama
4         self.umur = umur
5     def berbunyi(self): print(self.nama, "berbunyi")
6
7 class Burung(Hewan):
8     def __init__ (self, nama, umur, jenis_bulu):
9         super(). init (nama, umur)
10        self.jenis_bulu = jenis_bulu
11    def bersuara(self): print("Nyaring!")
12
13 kucingA = Burung("elang", 2, "Nyaring")
14 kucingA.berbunyi()
15 kucingA.bersuara()
```

The terminal output shows the execution of the code:

```
PS D:\pbo2,p2> & C:/Users/hp/AppData/Local/Programs/Python/Python310/python.exe d:/pbo2,p2/single1.py
elang berbunyi
Nyaring!
```

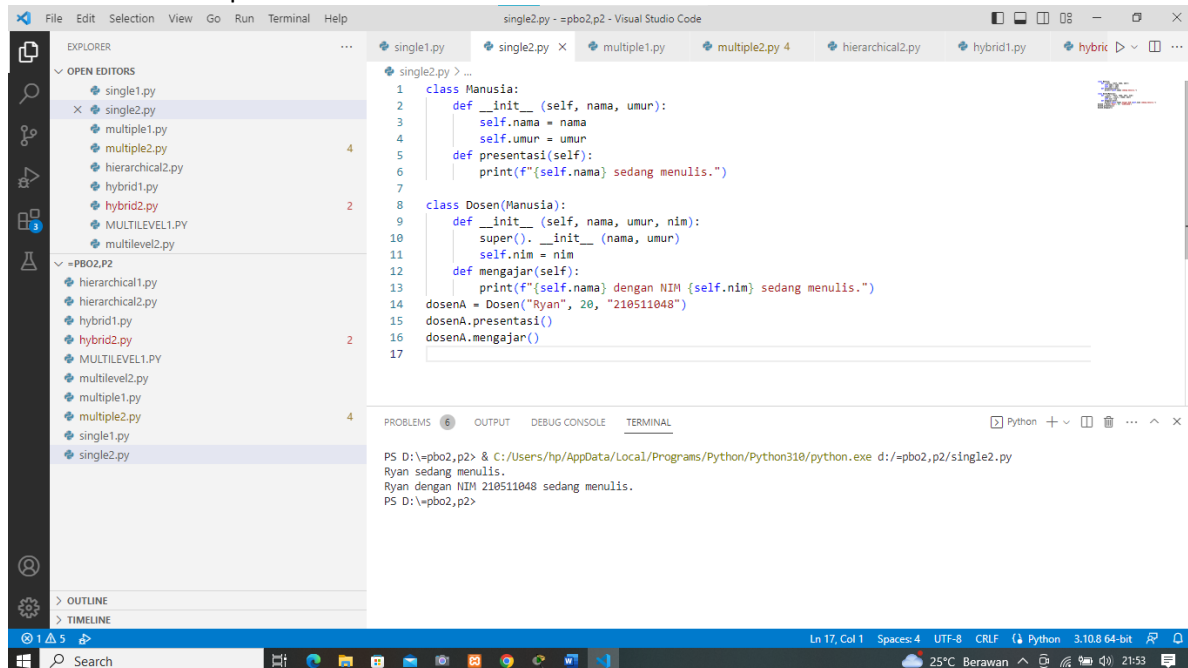
2. Single2

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

class Dosen(Manusia):
    def __init__(self, nama, umur, nim):
        super().__init__(nama, umur)
        self.nim = nim
    def mengajar(self):
        print(f"{self.nama} dengan NIM {self.nim} sedang menulis.")

dosenA = Dosen("Ryan", 20, "210511048")
dosenA.presentasi()
dosenA.mengajar()
```

Output SS



3. Multiple1

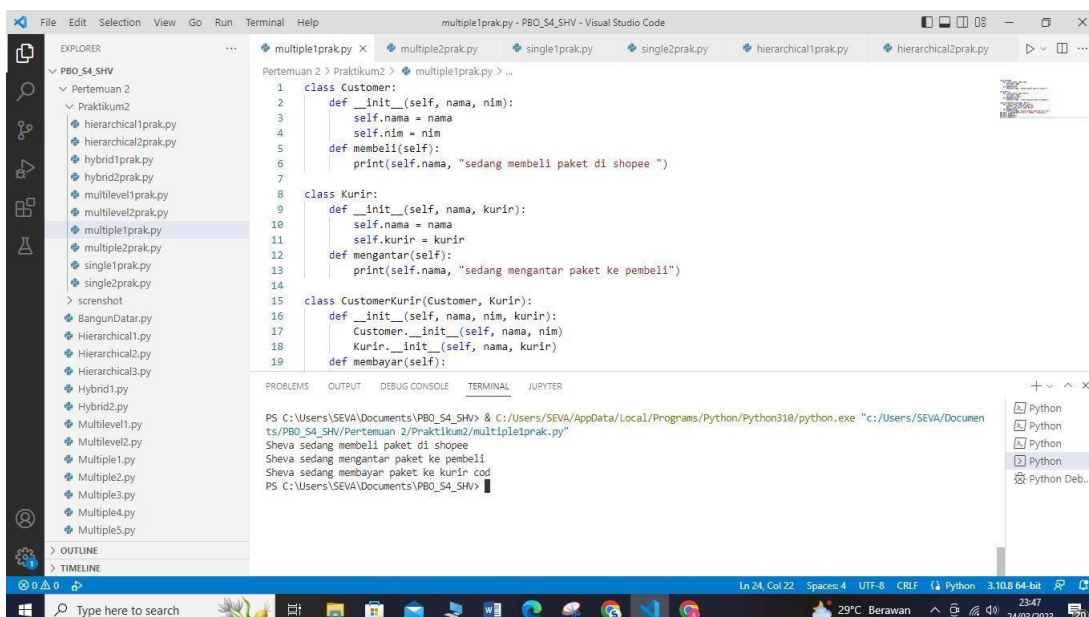
```
class Customer:
    def __init__(self, nama, nim):
        self.nama = nama
        self.nim = nim
    def membeli(self):
        print(self.nama, "sedang membeli produk di online shop ")

class Kurir:
    def __init__(self, nama, kurir):
        self.nama = nama
        self.kurir = kurir
    def mengantar(self):
        print(self.nama, "sedang mengantar paket ke pembeli")

class CustomerKurir(Customer, Kurir):
    def __init__(self, nama, nim, kurir):
        Customer.__init__(self, nama, nim)
        Kurir.__init__(self, nama, kurir)
    def membayar(self):
        print(self.nama, "sedang membayar paket ke kurir ")

mhs_kurir = CustomerKurir("Ryan", "1922", "Programmer")
mhs_kurir.membeli()
mhs_kurir.mengantar()
mhs_kurir.membayar()
```

Output SS



```
PS C:\Users\SEVA\Documents\PBO_S4_SHV> & C:/Users/SEVA/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/SEVA/Documen
ts/PBO_S4_SHV/Pertemuan 2/Praktikum2/multiple1prak.py"
Sheva sedang membeli paket di shopee
Sheva sedang mengantar paket ke pembeli
Sheva sedang membayar paket ke kurir cod
PS C:\Users\SEVA\Documents\PBO_S4_SHV>
```

4. Multiple2

```
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 Pekerja:
    def __init__(self, pekerjaan, gaji):
        self.pekerjaan = pekerjaan
        self.gaji = gaji
    def display_info(self):
        print(f>Pekerjaan: {self.pekerjaan}")
        print(f>Gaji: {self.gaji}")

class Pribadi:
    def __init__(self, hobi, alamat):
        self.hobi = hobi
        self.alamat = alamat
    def display_info(self):
        print(f>Hobi: {self.hobi}")
        print(f>Alamat: {self.alamat}")

class PribadiPekerja(Orang, Pekerja, Pribadi):
    def __init__(self, nama, umur, pekerjaan, gaji, hobi, alamat):
        Orang.__init__(self, nama, umur)
        Pekerja.__init__(self, pekerjaan, gaji)
        Pribadi.__init__(self, hobi, alamat)
    def display_info(self):
        super().display_info()
        print(f>Pekerjaan: {self.pekerjaan}")
        print(f>Gaji: {self.gaji}")
        print(f>Hobi: {self.hobi}")
        print(f>Alamat: {self.alamat}")

# contoh penggunaan
pribadi_pekerjaC = PribadiPekerja("Sheva", 19, "Digital Marketing", "20 Juta",
    "Bermain Bola", "Arjawinangun")
pribadi_pekerjaC.display_info()
```

Output SS Multiple2

```
File Edit Selection View Go Run Terminal Help
multiple2prak.py - PBO_S4_SHV - Visual Studio Code

EXPLORER
PBO_S4_SHV
  Pertemuan 2
    Praktikum2
      hierarchical1prak.py
      hierarchical2prak.py
      hybrid1prak.py
      hybrid2prak.py
      multilevel1prak.py
      multilevel2prak.py
      multiple1prak.py
      multiple2prak.py
      single1prak.py
      single2prak.py
      screenshot
      BangunDatar.py
      Hierarchical1.py
      Hierarchical2.py
      Hierarchical3.py
      Hybrid1.py
      Hybrid2.py
      Multilevel1.py
      Multilevel2.py
      Multiple1.py
      Multiple2.py
      Multiple3.py
      Multiple4.py
      Multiple5.py

multiple2prak.py
Pertemuan 2 > Praktikum2 > multiple2prak.py > ...
0 print(r"nama: {self.nama}")
7 print(f"Umur: {self.umur}")
8
9 class Pekerja:
10     def __init__(self, pekerjaan, gaji):
11         self.pekerjaan = pekerjaan
12         self.gaji = gaji
13     def display_info(self):
14         print(f"Pekerjaan: {self.pekerjaan}")
15         print(f"Gaji: {self.gaji}")
16
17 class Pribadi:
18     def __init__(self, hobi, alamat):
19         self.hobi = hobi
20         self.alamat = alamat
21     def display_info(self):
22         print(f"Hobi: {self.hobi}")
23         print(f"Alamat: {self.alamat}")
24
25 class PribadiPekerja(Orang, Pekerja, Pribadi):
26     def __init__(self, nama, umur, pekerjaan, gaji, hobi, alamat):
27         super().__init__(nama, umur)
28         super().__init__(pekerjaan, gaji)
29         super().__init__(hobi, alamat)

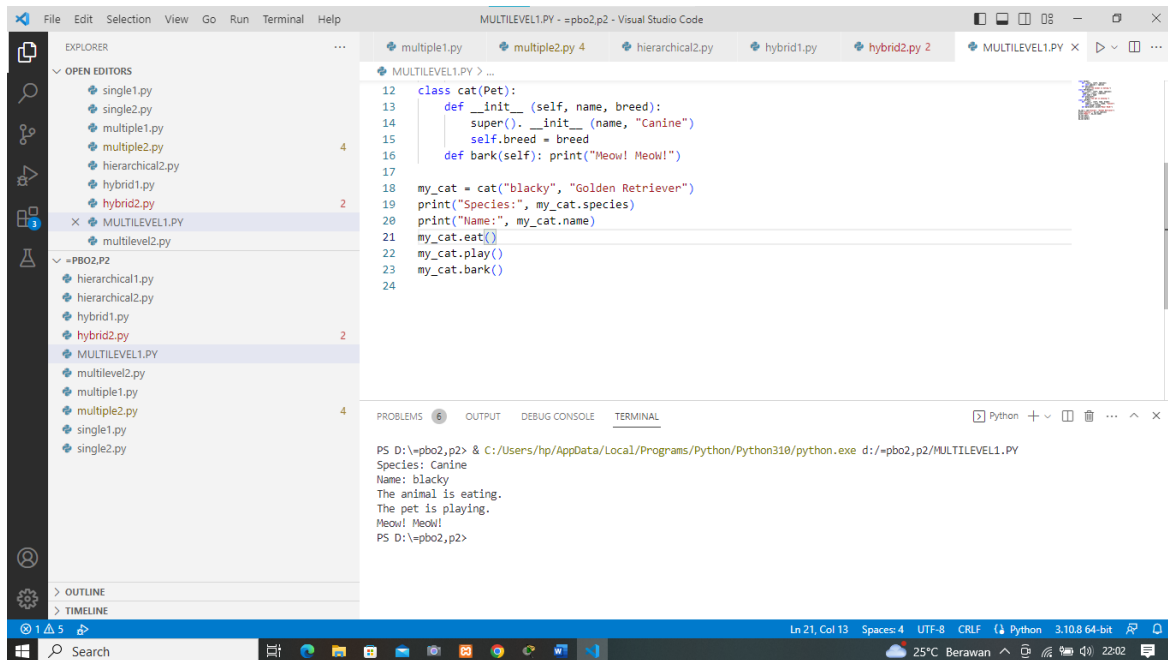
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
PS C:\Users\SEVA\Documents\PBO_S4_SHV> & C:\Users\SEVA\AppData\Local\Programs\Python\Python310\python.exe "c:/Users/SEVA/Documen
ts/PBO_S4_SHV/Pertemuan 2/Praktikum2/multiple2prak.py"
Nama: Sheva
Umur: 19
Pekerjaan: Digital Marketing
Gaji: 20 Juta
Hobi: Bermain Bola
Alamat: Arjawinangun
PS C:\Users\SEVA\Documents\PBO_S4_SHV>
```

5. Multilevel1

```
class Animal:
    def __init__(self, species):
        self.species = species
    def eat(self):
        print("The animal is eating.")
class Pet(Animal):
    def __init__(self, name, species):
        super().__init__(species)
        self.name = name
    def play(self):
        print("The pet is playing.")
class cat(Pet):
    def __init__(self, name, breed):
        super().__init__(name, "Canine")
        self.breed = breed
    def bark(self): print("Meow! Meow!")

my_cat = cat("blacky", "Golden Retriever")
print("Species:", my_cat.species)
print("Name:", my_cat.name)
my_cat.eat()
my_cat.play()
my_cat.bark()
```

Output SS Multilevel1



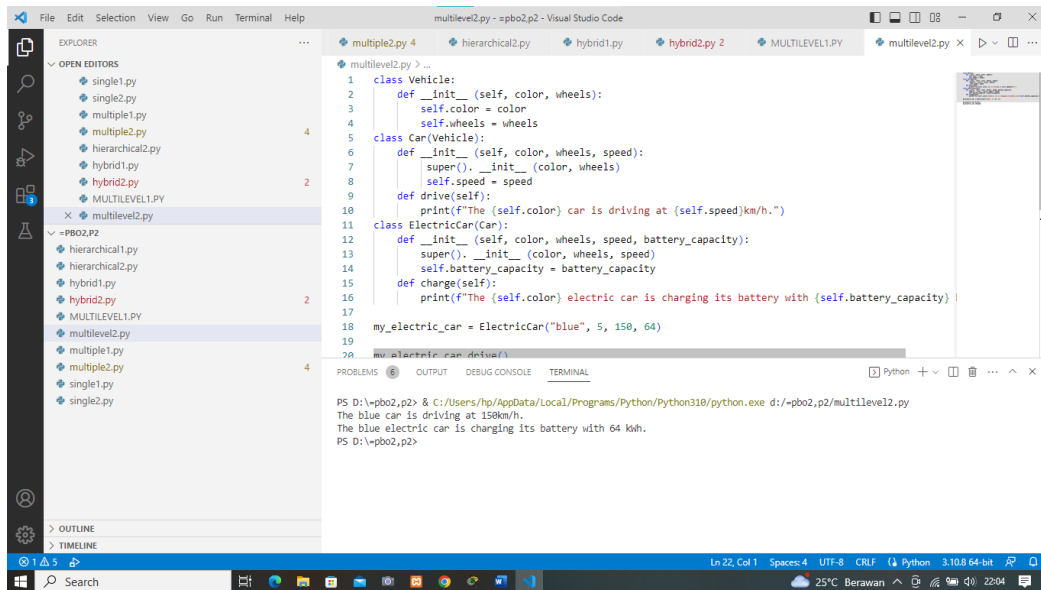
6. Multilevel2

```
class Vehicle:
    def __init__(self, color, wheels):
        self.color = color
        self.wheels = wheels
class Car(Vehicle):
    def __init__(self, color, wheels, speed):
        super().__init__(color, wheels)
        self.speed = speed
    def drive(self):
        print(f"The {self.color} car is driving at {self.speed}km/h.")
class ElectricCar(Car):
    def __init__(self, color, wheels, speed, battery_capacity):
        super().__init__(color, wheels, speed)
        self.battery_capacity = battery_capacity
    def charge(self):
        print(f"The {self.color} electric car is charging its battery with {self.battery_capacity} kWh.")

my_electric_car = ElectricCar("blue", 5, 150, 64)

my_electric_car.drive()
my_electric_car.charge()
```

Output SS Multilevel2



```
1 class Vehicle:
2     def __init__(self, color, wheels):
3         self.color = color
4         self.wheels = wheels
5     class Car(Vehicle):
6     def __init__(self, color, wheels, speed):
7         super().__init__(color, wheels)
8         self.speed = speed
9     def drive(self):
10        print(f"The {self.color} car is driving at {self.speed}km/h.")
11    class ElectricCar(Car):
12    def __init__(self, color, wheels, speed, battery_capacity):
13        super().__init__(color, wheels, speed)
14        self.battery_capacity = battery_capacity
15    def charge(self):
16        print(f"The {self.color} electric car is charging its battery with {self.battery_capacity} kWh.")
17
18 my_electric_car = ElectricCar("blue", 5, 150, 64)
19
20 my_electric_car.drive()
```

PS D:\pbo2,p2> & C:\Users\hp\AppData\Local\Programs\Python\Python310\python.exe d:/pbo2,p2/multilevel2.py
The blue car is driving at 150km/h.
The blue electric car is charging its battery with 64 kWh.
PS D:\pbo2,p2>

7. Hierarchical1

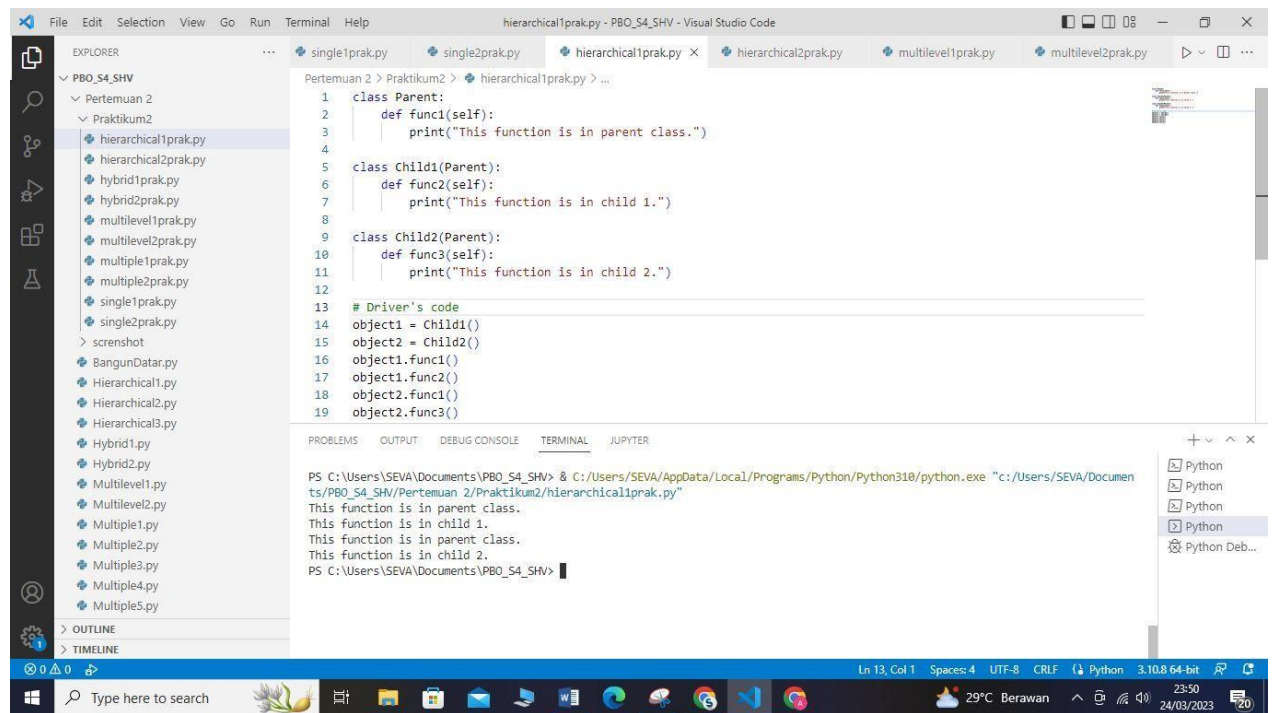
```
class Parent:
    def func1(self):
        print("This function is in parent class.")

class Child1(Parent):
    def func2(self):
        print("This function is in child 1.")

class Child2(Parent):
    def func3(self):
        print("This function is in child 2.")

# Driver's code
object1 = Child1()
object2 = Child2()
object1.func1()
object1.func2()
object2.func1()
object2.func3()
```


Output SS Hierarchical1



```
1 class Parent:
2     def func1(self):
3         print("This function is in parent class.")
4
5 class Child1(Parent):
6     def func2(self):
7         print("This function is in child 1.")
8
9 class Child2(Parent):
10    def func3(self):
11        print("This function is in child 2.")
12
13 # Driver's code
14 object1 = Child1()
15 object2 = Child2()
16 object1.func1()
17 object1.func2()
18 object2.func1()
19 object2.func3()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

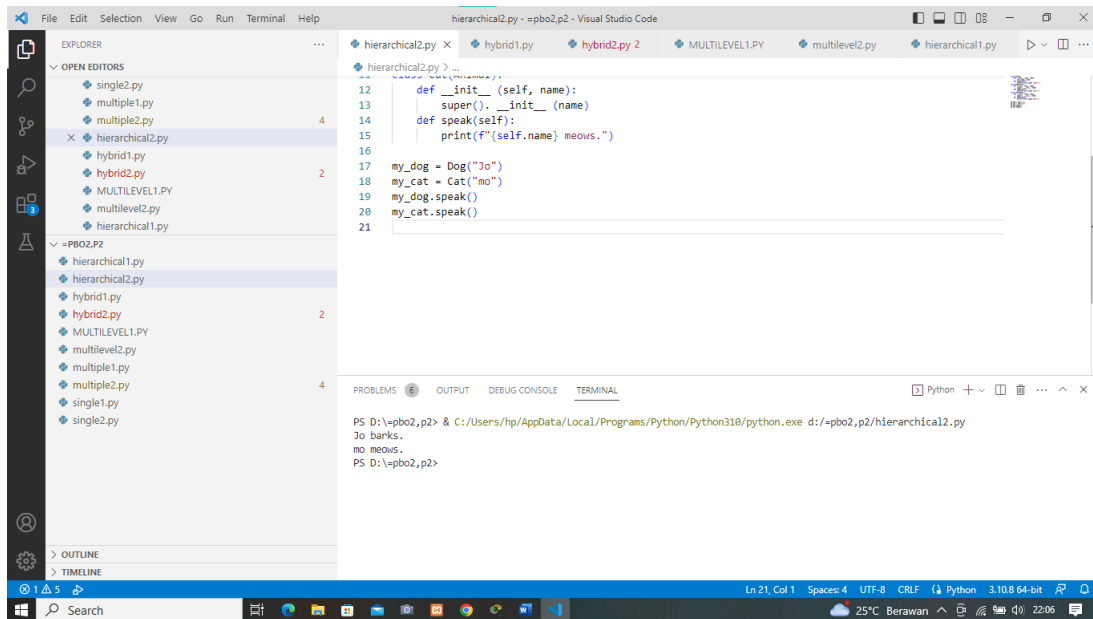
```
PS C:\Users\SEVA\Documents\PBO_S4_SHV> & C:/Users/SEVA/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/SEVA/Documen
ts/PBO_S4_SHV/Pertemuan 2/Praktikum2/hierarchical1prak.py"
This function is in parent class.
This function is in child 1.
This function is in parent class.
This function is in child 2.
PS C:\Users\SEVA\Documents\PBO_S4_SHV>
```

8. Hierarchical2

```
class Animal:
    def __init__(self, name):
        self.name = name
    def speak(self):
        print(f"{self.name} speaks.")
class Dog(Animal):
    def __init__(self, name):
        super().__init__(name)
    def speak(self):
        print(f"{self.name} barks.")
class Cat(Animal):
    def __init__(self, name):
        super().__init__(name)
    def speak(self):
        print(f"{self.name} meows.")

my_dog = Dog("Jo")
my_cat = Cat("mo")
my_dog.speak()
my_cat.speak()
```

Output SS Hierarchical2

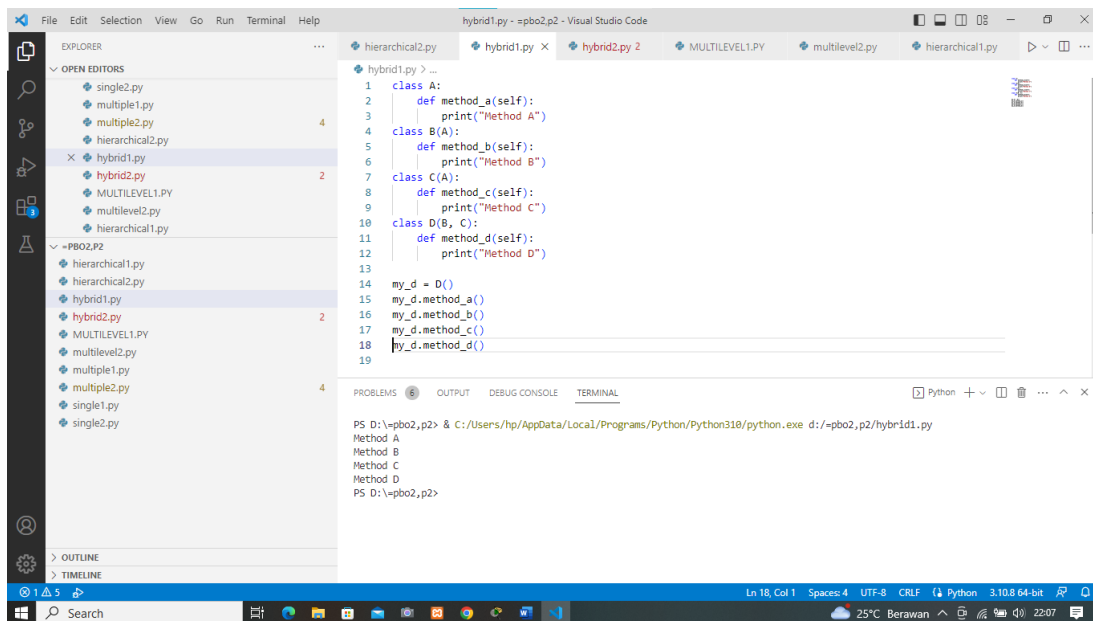


9. Hybrid1

```
class A:
    def method_a(self):
        print("Method A")
class B(A):
    def method_b(self):
        print("Method B")
class C(A):
    def method_c(self):
        print("Method C")
class D(B, C):
    def method_d(self):
        print("Method D")
```

```
my_d = D()
my_d.method_a()
my_d.method_b()
my_d.method_c()
my_d.method_d()
```

Output SS Hybrid1



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left lists several Python files, including hybrid1.py, hybrid2.py, MULTILEVEL1.PY, multilevel2.py, and hierarchical1.py. The hybrid1.py file is open in the editor, showing the following code:

```
1 class A:
2     def method_a(self):
3         print("Method A")
4 class B(A):
5     def method_b(self):
6         print("Method B")
7 class C(A):
8     def method_c(self):
9         print("Method C")
10 class D(B, C):
11     def method_d(self):
12         print("Method D")
13
14 my_d = D()
15 my_d.method_a()
16 my_d.method_b()
17 my_d.method_c()
18 my_d.method_d()
19
```

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

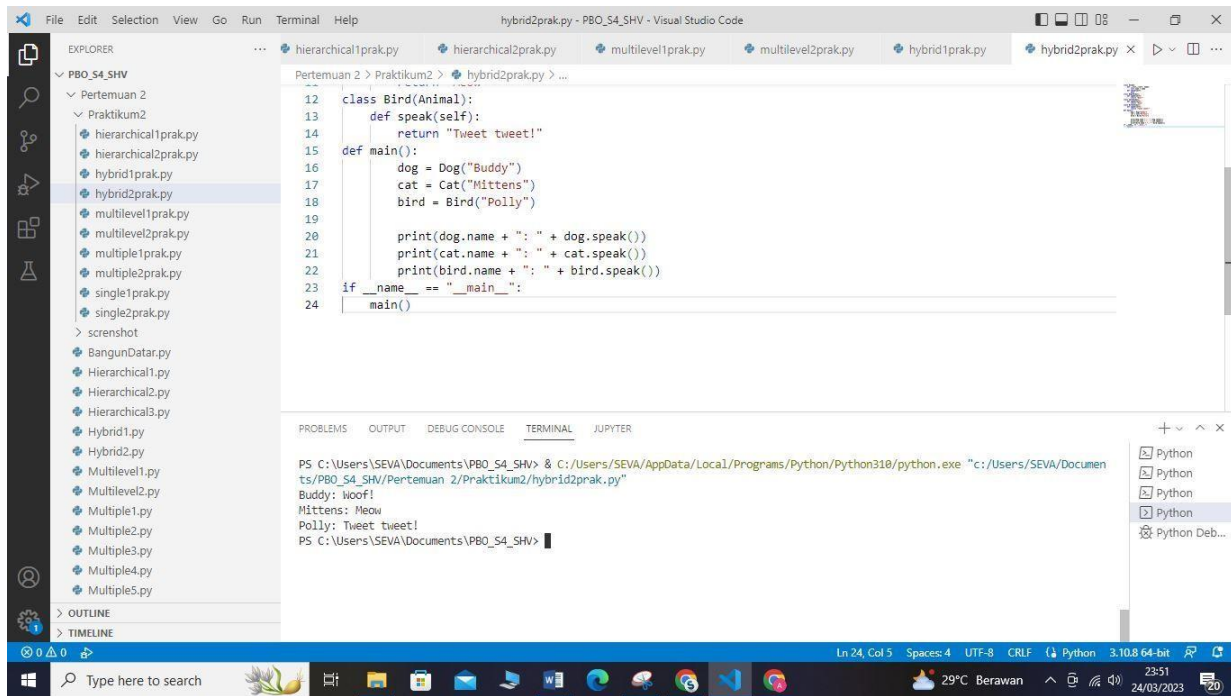
```
PS D:\pbo2,p2> & C:\Users\hp\AppData\Local\Programs\Python\Python310\python.exe d:/pbo2,p2/hybrid1.py
Method A
Method B
Method C
Method D
PS D:\pbo2,p2>
```

10. Hybrid2

```
class Animal:
    def __init__(self, name):
        self.name = name
    def speak(self):
        pass
class Dog(Animal):
    def speak(self):
        return "Woof!"
class Cat(Animal):
    def speak(self):
        return "Meow"
class Bird(Animal):
    def speak(self):
        return "Tweet tweet!"
def main():
    dog = Dog("Buddy")
    cat = Cat("Mittens")
    bird = Bird("Polly")

    print(dog.name + ": " + dog.speak())
    print(cat.name + ": " + cat.speak())
    print(bird.name + ": " + bird.speak())
if __name__ == "__main__":
    main()
```

Output SS Hybrid2



The screenshot displays the Visual Studio Code interface. The Explorer sidebar on the left shows a project structure with a folder named 'PBO_S4_SHV' containing a 'Praktikum2' subfolder. Inside 'Praktikum2', several Python files are listed, including 'hybrid2prak.py' which is currently selected and open in the editor. The editor shows a Python script with a class hierarchy and a main function. The script defines a 'Bird' class that inherits from 'Animal' (implied), with a 'speak' method that returns 'Tweet tweet!'. The 'main' function creates instances of 'Dog' (Buddy), 'Cat' (Mittens), and 'Bird' (Polly), and prints their names and speech. The output window at the bottom shows the execution of the script, displaying the names and speech of the created objects: 'Buddy: woof!', 'Mittens: Meow', and 'Polly: Tweet tweet!'. The status bar at the bottom indicates the current line and column (Ln 24, Col 5), the encoding (UTF-8), the line ending (CRLF), the Python version (3.10.8 64-bit), and the system clock (23:51 on 24/03/2023).

```
12 class Bird(Animal):
13     def speak(self):
14         return "Tweet tweet!"
15
16 def main():
17     dog = Dog("Buddy")
18     cat = Cat("Mittens")
19     bird = Bird("Polly")
20
21     print(dog.name + ": " + dog.speak())
22     print(cat.name + ": " + cat.speak())
23     print(bird.name + ": " + bird.speak())
24
25 if __name__ == "__main__":
26     main()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
PS C:\Users\SEVA\Documents\PBO_S4_SHV> & C:\Users\SEVA\AppData\Local\Programs\Python\Python310\python.exe "c:\Users\SEVA\Documents\PBO_S4_SHV\Praktikum2\hybrid2prak.py"
Buddy: woof!
Mittens: Meow
Polly: Tweet tweet!
PS C:\Users\SEVA\Documents\PBO_S4_SHV>
```