

LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK LANJUT

2023



Prepared By:

Nama : Andrian Shevchenko

Kelas : TIF21B R2

Nim : 210511071

Praktikum 2

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

1. Single1

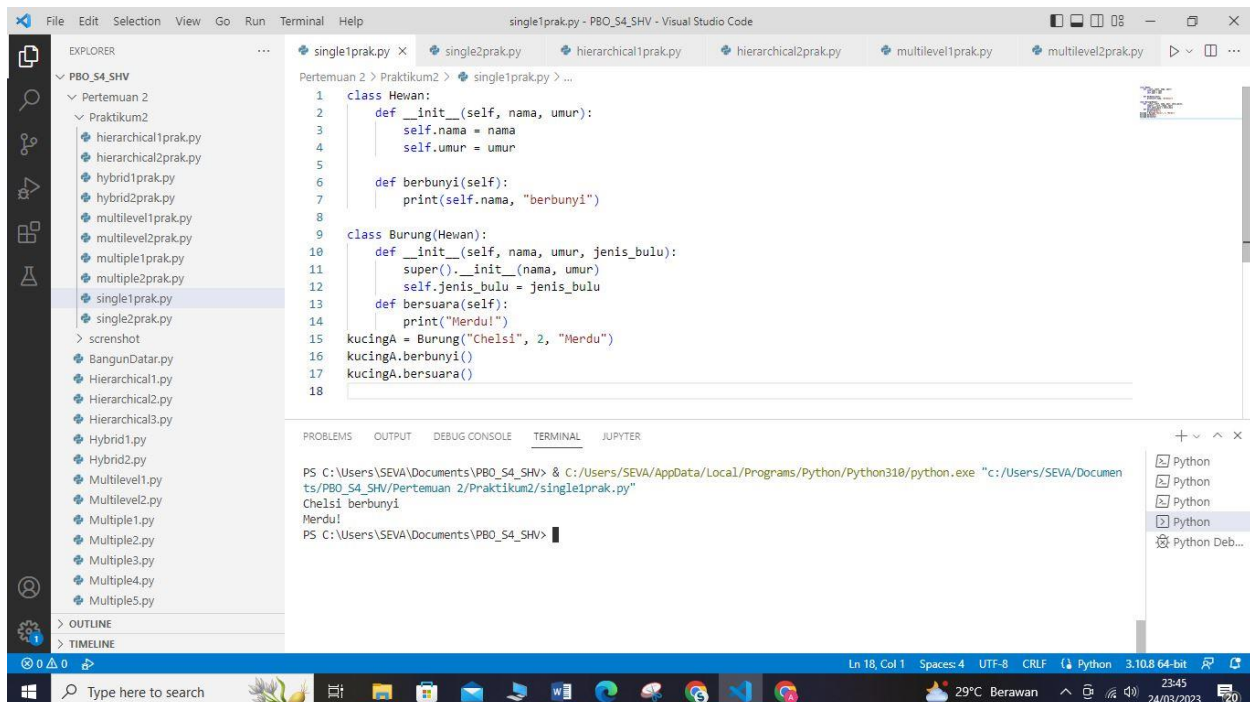
```
class Hewan:
    def __init__(self, nama, umur):
        self.nama = nama
        self.umur = umur

    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("Merdu!")

kucingA = Burung("Chelsi", 2, "Merdu")
kucingA.berbunyi()
kucingA.bersuara()
```

Output SS



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left displays a file tree for a project named 'PBO_S4_SHV'. The main editor window shows the Python code for 'single1prak.py'. The code defines a base class 'Hewan' with an '__init__' method and a 'berbunyi' method. A subclass 'Burung' inherits from 'Hewan' and adds a 'bersuara' method. The code then creates an instance 'kucingA' of the 'Burung' class and calls its methods. The Output pane at the bottom shows the execution results: 'Chelsi berbunyi' and 'Merdu!'. The status bar at the bottom indicates the file is at line 18, column 1, with 4 spaces, using UTF-8 encoding and CRLF line endings. The system tray shows the date and time as 24/03/2023, 23:45.

```
File Edit Selection View Go Run Terminal Help
single1prak.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

> OUTLINE
> TIMELINE

single1prak.py
1 class Hewan:
2     def __init__(self, nama, umur):
3         self.nama = nama
4         self.umur = umur
5
6     def berbunyi(self):
7         print(self.nama, "berbunyi")
8
9 class Burung(Hewan):
10    def __init__(self, nama, umur, jenis_bulu):
11        super().__init__(nama, umur)
12        self.jenis_bulu = jenis_bulu
13    def bersuara(self):
14        print("Merdu!")
15 kucingA = Burung("Chelsi", 2, "Merdu")
16 kucingA.berbunyi()
17 kucingA.bersuara()
18

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/Pertemuan 2/Praktikum2/single1prak.py"
Chelsi berbunyi
Merdu!
PS C:\Users\SEVA\Documents\PBO_S4_SHV>

Ln 18, Col 1 Spaces: 4 UTF-8 CRLF Python 3.10.8 64-bit 23:45
24/03/2023
```

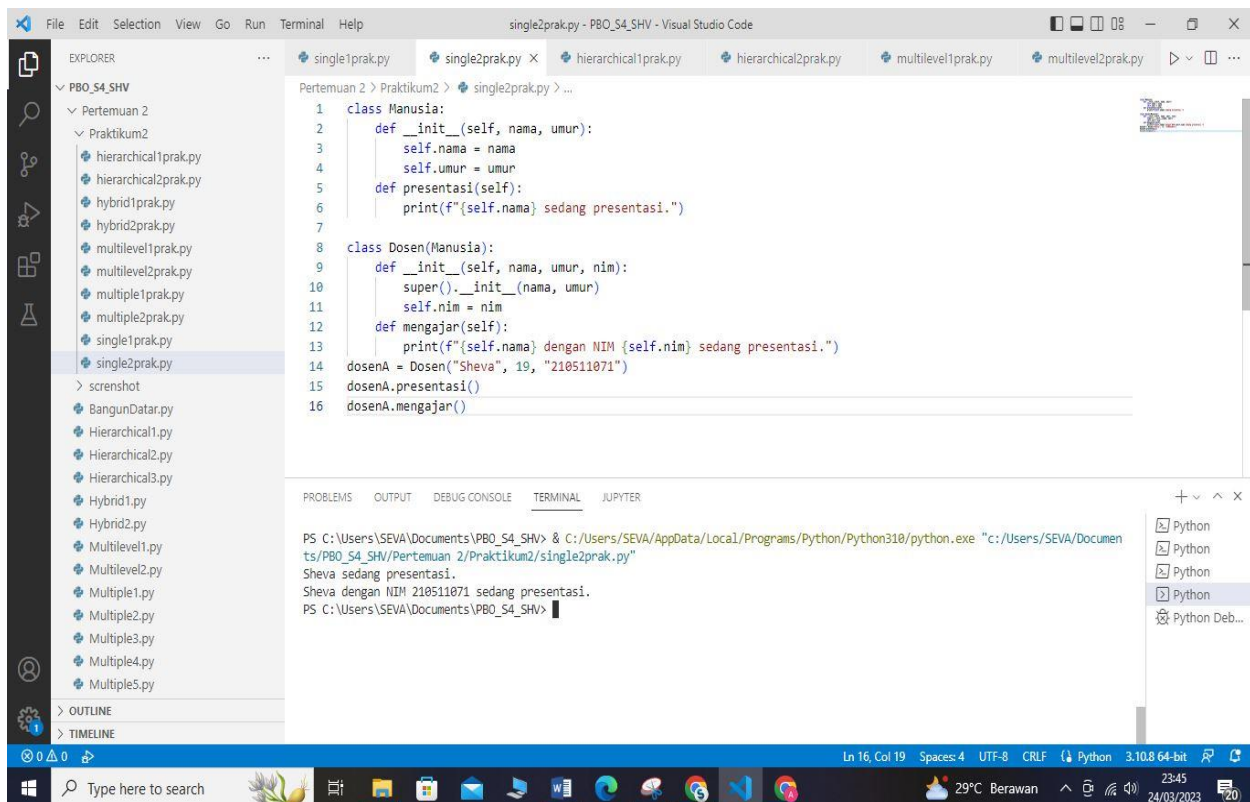
2. Single2

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

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 presentasi.")

dosenA = Dosen("Sheva", 19, "210511071")
dosenA.presentasi()
dosenA.mengajar()
```

Output SS



The screenshot shows the Visual Studio Code interface with the file explorer on the left displaying a project structure under 'PBO_S4_SHV'. The main editor window shows the Python code for 'single2prak.py'. The terminal at the bottom displays the execution output, showing the class definitions and the execution of the 'dosenA' object's methods.

```
1 class Manusia:
2     def __init__(self, nama, umur):
3         self.nama = nama
4         self.umur = umur
5     def presentasi(self):
6         print(f"{self.nama} sedang presentasi.")
7
8 class Dosen(Manusia):
9     def __init__(self, nama, umur, nim):
10        super().__init__(nama, umur)
11        self.nim = nim
12    def mengajar(self):
13        print(f"{self.nama} dengan NIM {self.nim} sedang presentasi.")
14
15 dosenA = Dosen("Sheva", 19, "210511071")
16 dosenA.presentasi()
17 dosenA.mengajar()
```

Terminal Output:

```
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/Pertemuan 2/Praktikum2/single2prak.py"
Sheva sedang presentasi.
Sheva dengan NIM 210511071 sedang presentasi.
PS C:\Users\SEVA\Documents\PBO_S4_SHV>
```

3. Multiple1

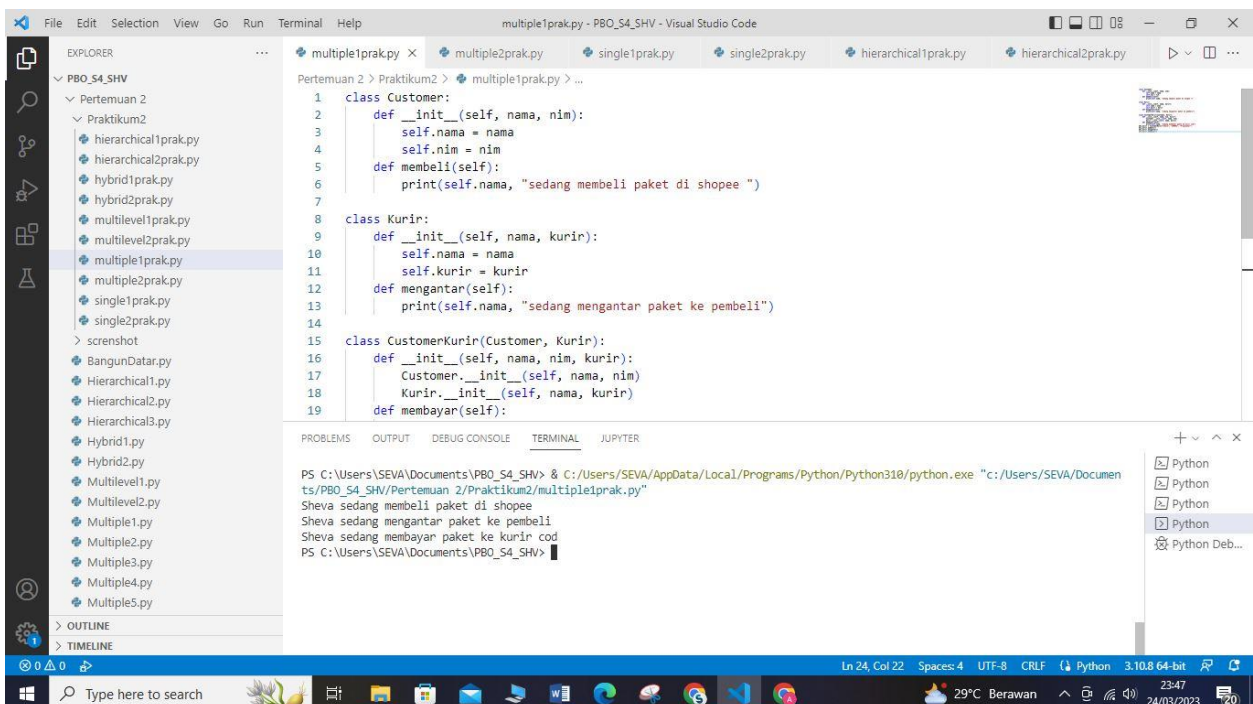
```
class Customer:
    def __init__(self, nama, nim):
        self.nama = nama
        self.nim = nim
    def membeli(self):
        print(self.nama, "sedang membeli paket di shopee ")

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 cod")

mhs_kurir = CustomerKurir("Sheva", "190001", "Programmer")
mhs_kurir.membeli()
mhs_kurir.mengantar()
mhs_kurir.membayar()
```

Output SS



```
File Edit Selection View Go Run Terminal Help
multiple1prak.py x multiple2prak.py single1prak.py single2prak.py hierarchical1prak.py hierarchical2prak.py
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
  OUTLINE
  TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
multiple1prak.py
1 class Customer:
2     def __init__(self, nama, nim):
3         self.nama = nama
4         self.nim = nim
5     def membeli(self):
6         print(self.nama, "sedang membeli paket di shopee ")
7
8 class Kurir:
9     def __init__(self, nama, kurir):
10        self.nama = nama
11        self.kurir = kurir
12    def mengantar(self):
13        print(self.nama, "sedang mengantar paket ke pembeli")
14
15 class CustomerKurir(Customer, Kurir):
16     def __init__(self, nama, nim, kurir):
17         Customer.__init__(self, nama, nim)
18         Kurir.__init__(self, nama, kurir)
19     def membayar(self):
20         print(self.nama, "sedang membayar paket ke kurir cod")
21
22 mhs_kurir = CustomerKurir("Sheva", "190001", "Programmer")
23 mhs_kurir.membeli()
24 mhs_kurir.mengantar()
25 mhs_kurir.membayar()
PS C:\Users\SEVA\Documents\PBO_S4_SHV> & C:/Users/SEVA/AppData/Local/Programs/Python/Python310/python.exe "C:/Users/SEVA/Document
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>
Ln 24, Col 22 Spaces: 4 UTF-8 CRLF Python 3.10.8 64-bit
23:47
24/03/2023
```

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})

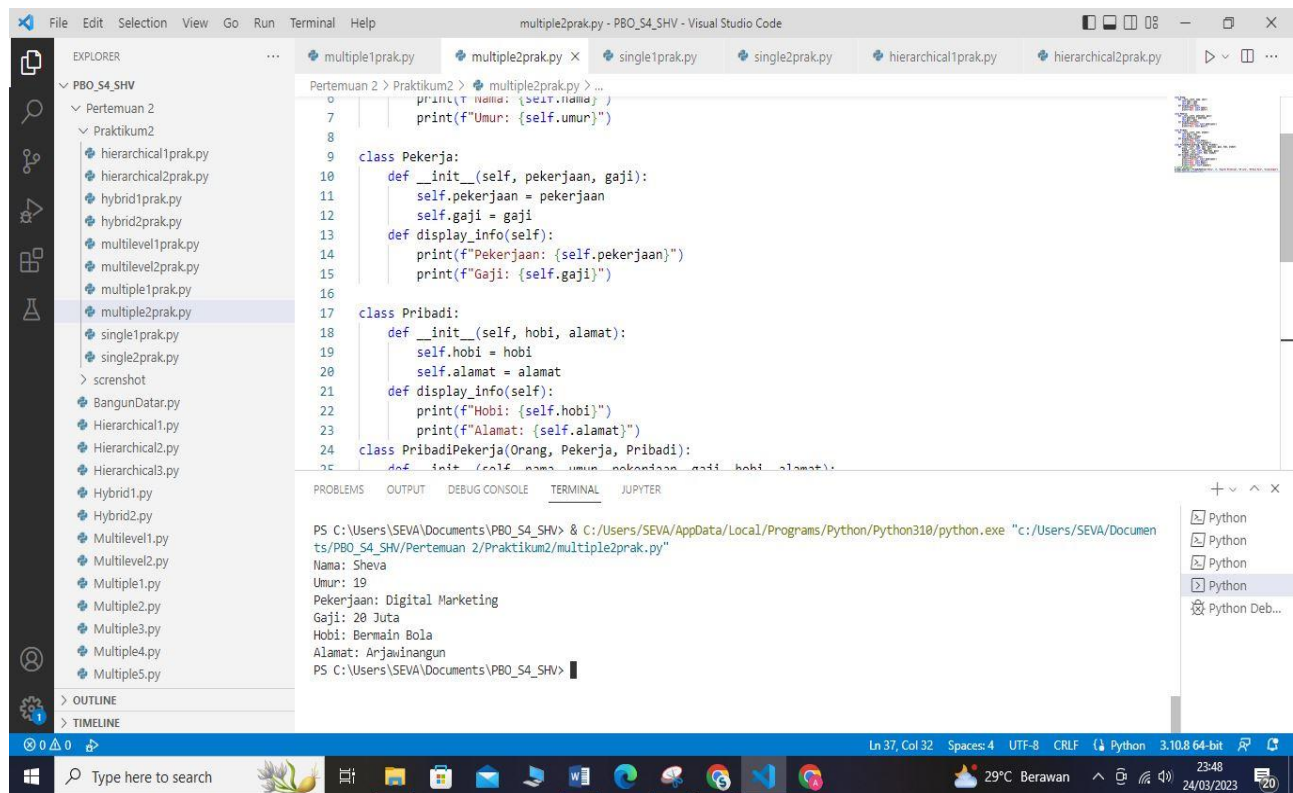
class Pekerja:
    def __init__(self, pekerjaan, gaji):
        self.pekerjaan = pekerjaan
        self.gaji = gaji
    def display_info(self):
        print(f>Pekerjaan: {self.pekerjaan})
        print(f>Gaji: {self.gaji})

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
  > OUTLINE
  > TIMELINE

multiple2prak.py
0 print(f"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 class PribadiPekerja(Orang, Pekerja, Pribadi):
25     def __init__(self, nama, umur, pekerjaan, gaji, hobi, alamat):
26         super().__init__(nama, umur, pekerjaan, gaji, hobi, alamat)

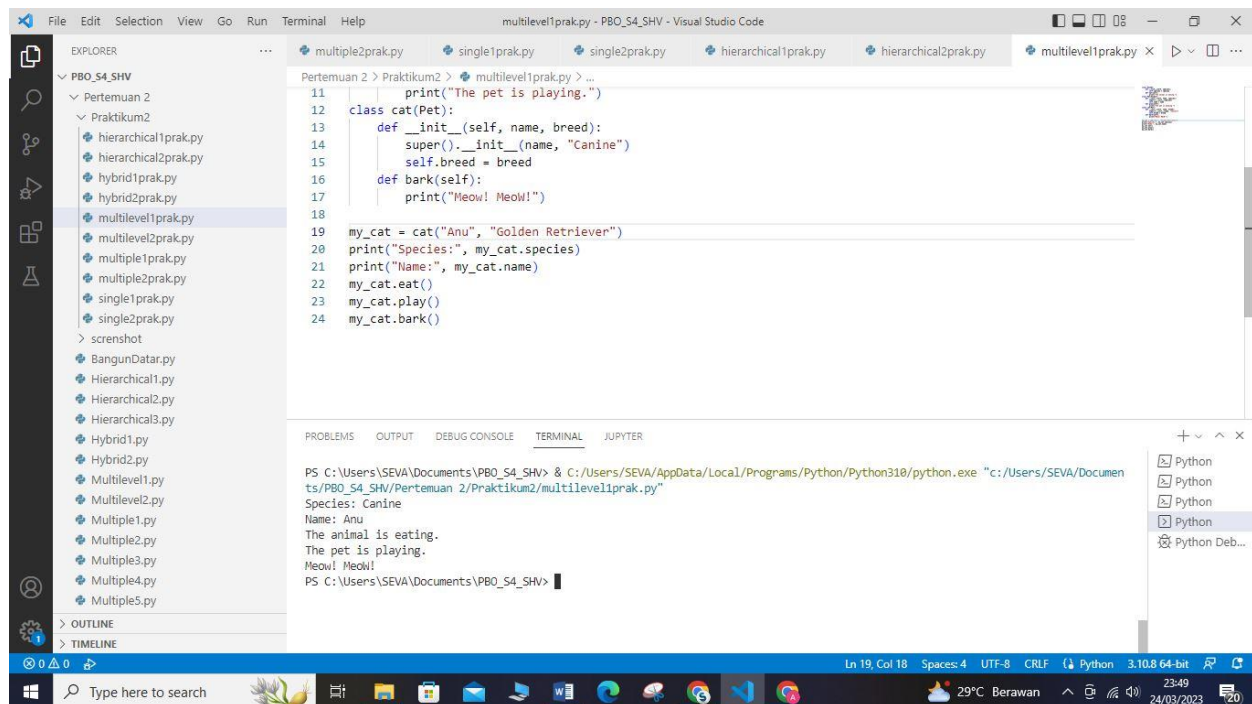
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
PS C:\Users\SEVA\Documents\PBO_S4_SHV> & C:\Users\SEVA\AppData\Local\Programs\Python\Python310\python.exe "c:/Users/SEVA/Documen
ts/PBO_S4_SHV/Pertemuan 2/Praktikum2/multiple2prak.py"
Nama: Sheva
Umur: 19
Pekerjaan: Digital Marketing
Gaji: 20 Juta
Hobi: Bermain Bola
Alamat: Anjalinangun
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("Anu", "Golden Retriever")
print("Species:", my_cat.species)
print("Name:", my_cat.name)
my_cat.eat()
my_cat.play()
my_cat.bark()
```


Output SS Multilevel1



```
11 print("The pet is playing.")
12 class cat(Pet):
13     def __init__(self, name, breed):
14         super().__init__(name, "Canine")
15         self.breed = breed
16     def bark(self):
17         print("Meow! Meow!")
18
19 my_cat = cat("Anu", "Golden Retriever")
20 print("Species:", my_cat.species)
21 print("Name:", my_cat.name)
22 my_cat.eat()
23 my_cat.play()
24 my_cat.bark()
```

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/Pertemuan 2/Praktikum2/multilevel1prak.py"
Species: Canine
Name: Anu
The animal is eating.
The pet is playing.
Meow! Meow!
```

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("red", 4, 120, 60)
my_electric_car.drive()
my_electric_car.charge()
```

Output SS Multilevel2

```
File Edit Selection View Go Run Terminal Help
multilevel2prak.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
  > OUTLINE
  > TIMELINE

multilevel2prak.py
7
8
9
10
11
12
13
14
15
16
17
18
19
20

class Car:
    def __init__(self, color, wheels, speed):
        self.color = color
        self.wheels = 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("red", 4, 120, 60)
my_electric_car.drive()
my_electric_car.charge()
```

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/multilevel2prak.py"
The red car is driving at 120km/h.
The red electric car is charging its battery with 60 kwh.
PS C:\Users\SEVA\Documents\PBO_S4_SHV>
```

Ln 18, Col 35 Spaces: 4 UTF-8 CRLF Python 3.10.8 64-bit 23:49 24/03/2023

7. Hierarchical1

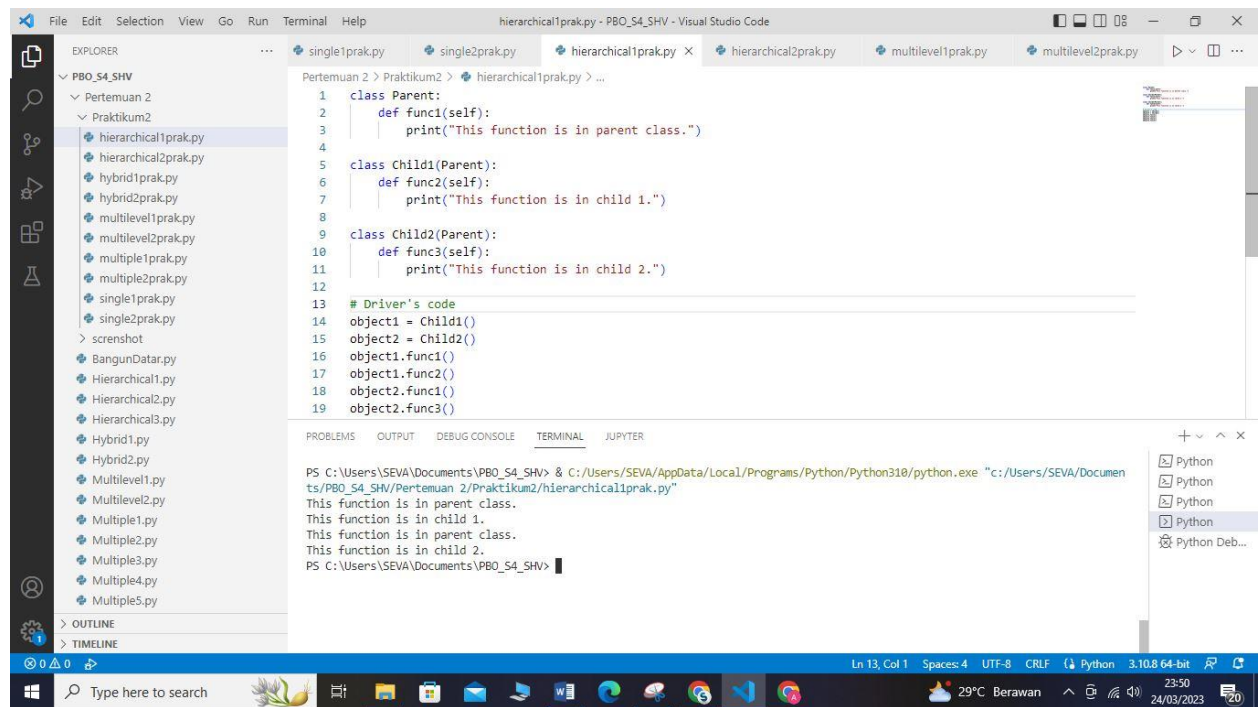
```
class Parent:
    def func1(self):
        print("This function is in parent class.")

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



The screenshot shows a Visual Studio Code window 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 structure with a folder 'PBO_S4_SHV' containing a subfolder 'Praktikum2' and several Python files. The code editor displays the content of 'hierarchichal1prak.py' (note the typo in the filename). The code defines a base class 'Parent' with a method 'func1', and two child classes 'Child1' and 'Child2' that inherit from 'Parent'. 'Child1' has a method 'func2' and 'Child2' has a method 'func3'. The 'Driver's code' section creates instances of 'Child1' and 'Child2' and calls their respective methods. The terminal shows the output of the program, which prints the function names for each instance.

```
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()
```

PS C:\Users\SEVA\Documents\PBO_S4_SHV> & C:/Users/SEVA/AppData/Local/Programs/Python/Python318/python.exe "c:/Users/SEVA/Documents/PBO_S4_SHV/Praktikum2/hierarchichal1prak.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.

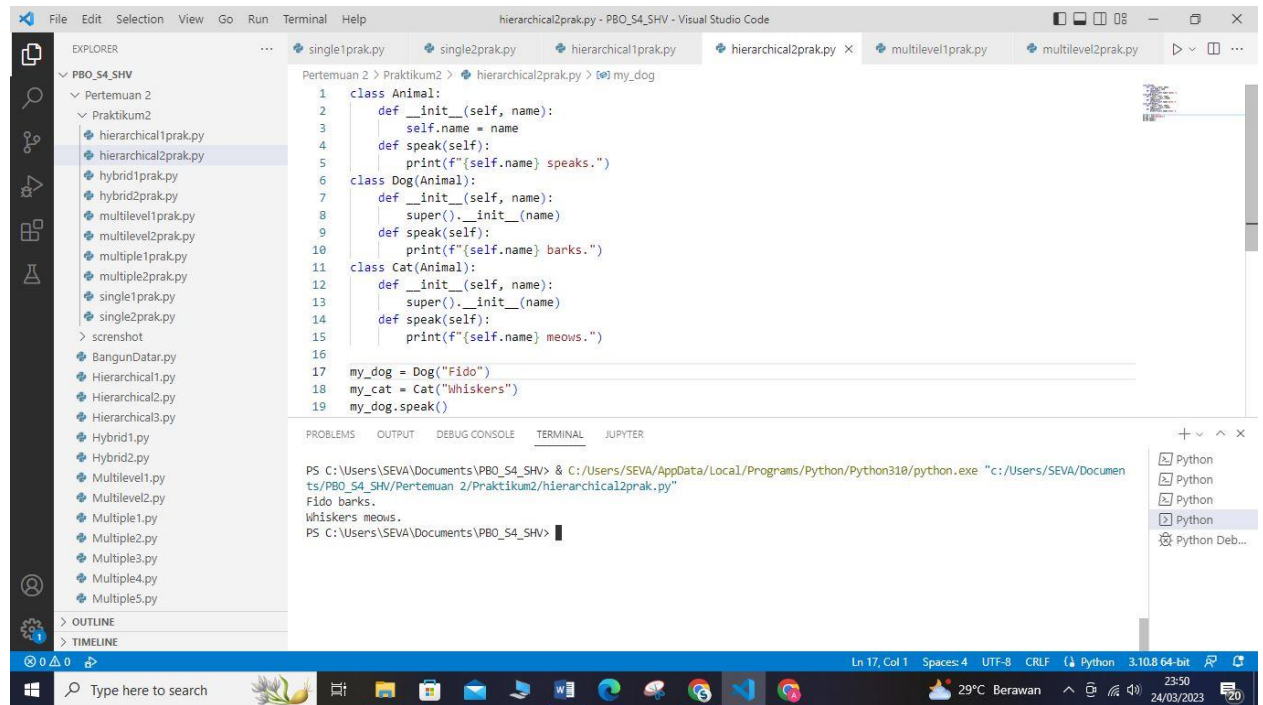
PS C:\Users\SEVA\Documents\PBO_S4_SHV>

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("Fido")
my_cat = Cat("Whiskers")
my_dog.speak()
my_cat.speak()
```

Output SS Hierarchical2



```
1 class Animal:
2     def __init__(self, name):
3         self.name = name
4     def speak(self):
5         print(f"{self.name} speaks.")
6 class Dog(Animal):
7     def __init__(self, name):
8         super().__init__(name)
9     def speak(self):
10        print(f"{self.name} barks.")
11 class Cat(Animal):
12     def __init__(self, name):
13         super().__init__(name)
14     def speak(self):
15        print(f"{self.name} meows.")
16
17 my_dog = Dog("Fido")
18 my_cat = Cat("Whiskers")
19 my_dog.speak()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
PS C:\Users\SEVA\Documents\PBO_S4_SHV> & C:\Users\SEVA\AppData\Local\Programs\Python\Python318\python.exe "c:\Users\SEVA\Documents\PBO_S4_SHV\Pertemuan 2\Praktikum2\hierarchical2prak.py"
Fido barks.
Whiskers meows.
PS C:\Users\SEVA\Documents\PBO_S4_SHV>
```

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 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 'PBO_S4_SHV' with a subfolder 'Praktikum2' containing several Python files. The code editor displays the file 'hybrid1prak.py' with the following Python 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()
```

The terminal at the bottom shows the execution of the code, displaying the output of each method call:

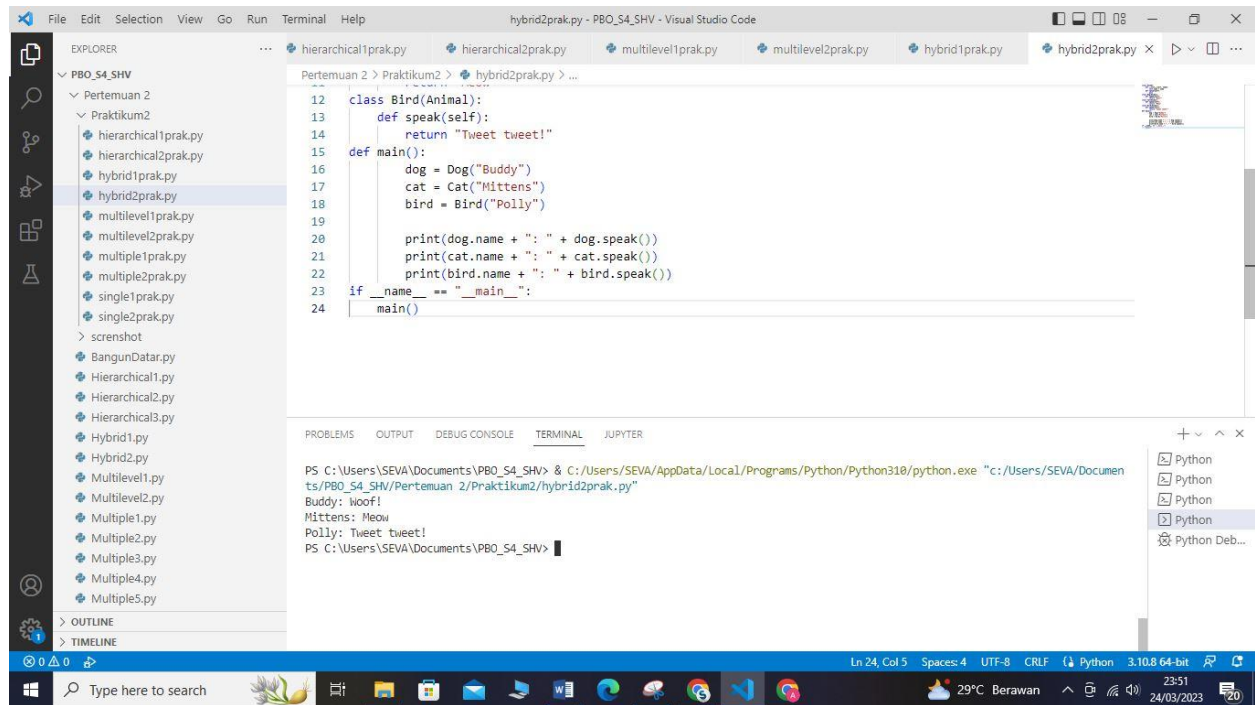
```
PS C:\Users\SEVA\Documents\PBO_S4_SHV> & C:/Users/SEVA/AppData/Local/Programs/Python/Python318/python.exe "c:/Users/SEVA/Documen
ts/PBO_S4_SHV/Praktikum2/hybrid1prak.py"
Method A
Method B
Method C
Method D
PS C:\Users\SEVA\Documents\PBO_S4_SHV>
```

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 with a Python file named `hybrid2prak.py` open. The file contains a class `Bird` that inherits from `Animal` and has a `speak` method. The `main` function creates instances of `Dog`, `Cat`, and `Bird`, and prints their names and the output of their `speak` methods. The terminal window shows the execution of the script, which runs successfully and produces the expected output.

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

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
PS C:\Users\SEVA\Documents\PBO_S4_SHV> & C:/Users/SEVA/AppData/Local/Programs/Python/Python318/python.exe "c:/Users/SEVA/Documen
ts/PBO_S4_SHV/Pertemuan 2/Praktikum2/hybrid2prak.py"
Buddy: Woof!
Mittens: Meow
Polly: Tweet tweet!
PS C:\Users\SEVA\Documents\PBO_S4_SHV>
```

Ln 24, Col 5 Spaces: 4 UTF-8 CRLF Python 3.10.8 64-bit 23:51 24/03/2023