

LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK LANJUT

2023



Prepared By:
Farhan Saefulah
NIM. 210511059

Praktikum 2

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

Hierarchical 1 :

```
#Nama : Farhan Saefulah
#NIM : 210511059
#Kelas : R2

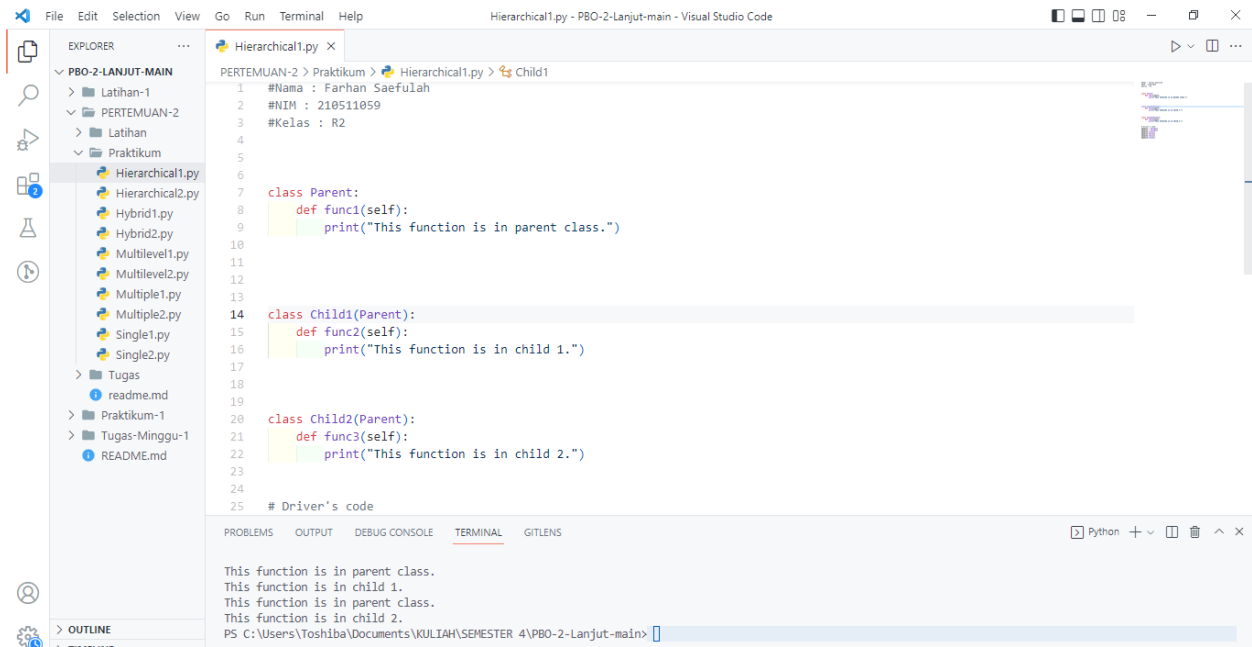
class Parent:
    def func1(self):
        print("This function is in parent class.")

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



The screenshot shows the Visual Studio Code interface with a Python file named `Hierarchical1.py` open. The file contains a class hierarchy with a `Parent` class and two child classes, `Child1` and `Child2`. The `Parent` class has a `func1` method, while the child classes have `func2` and `func3` methods respectively. The terminal output shows the execution of the code, displaying the messages from each class method.

```
1 #Nama : Farhan Saefulah
2 #NIM : 210511059
3 #Kelas : R2
4
5
6
7 class Parent:
8     def func1(self):
9         print("This function is in parent class.")
10
11
12
13
14 class Child1(Parent):
15     def func2(self):
16         print("This function is in child 1.")
17
18
19
20 class Child2(Parent):
21     def func3(self):
22         print("This function is in child 2.")
23
24
25 # Driver's code
```

The terminal output shows the following messages:

```
This function is in parent class.
This function is in child 1.
This function is in parent class.
This function is in child 2.
```

The terminal prompt is `PS C:\Users\Toshiba\Documents\KULIAH\SEMESTER 4\PBO-2-Lanjut-main>`.

Hierarchical 2 :

```
#Nama : Farhan Saefulah
#NIM : 210511059
#Kelas : R2

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

Hybrid1 :

```
#Nama : Farhan Saefulah
#NIM : 210511059
#Kelas : R2

class A:
    def method_a(self):
        print("Method A")

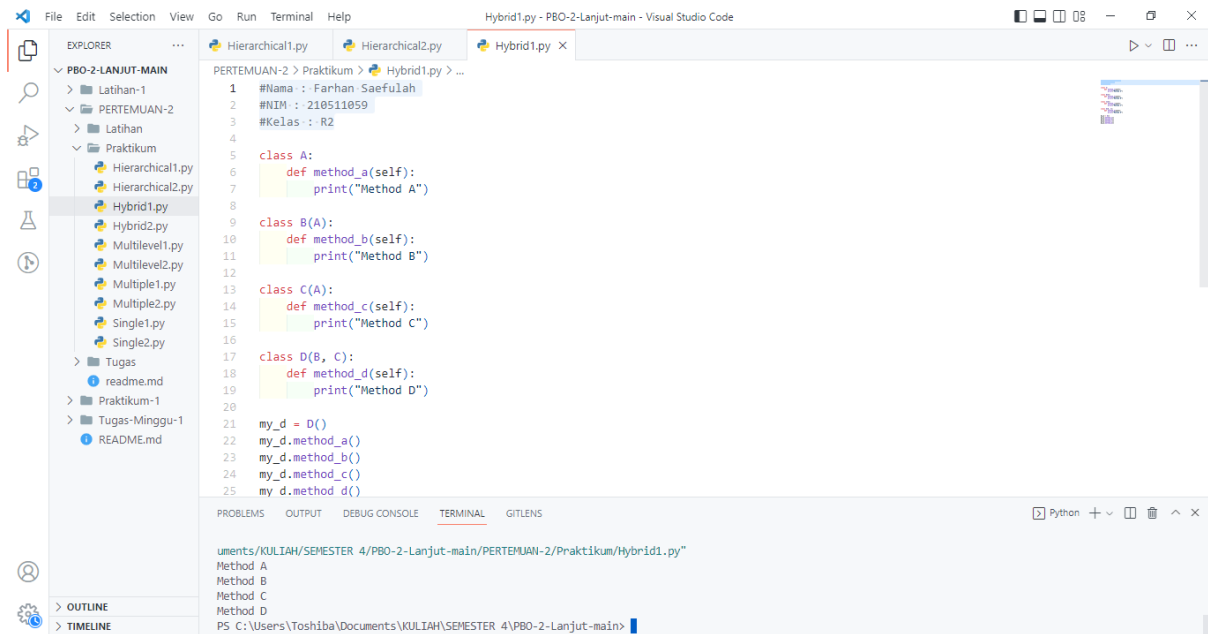
class B(A):
    def method_b(self):
        print("Method B")

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



The screenshot displays the Visual Studio Code interface. The Explorer panel on the left shows a project structure for 'PBO-2-LANJUT-MAIN' with folders 'Latihan-1', 'PERTEMUAN-2', 'Latihan', and 'Praktikum'. The 'Praktikum' folder contains several Python files, including 'Hybrid1.py' which is currently selected. The main editor window shows the code in 'Hybrid1.py'.

```
1 #Nama : Farhan Saefulah
2 #NIM : 210511059
3 #Kelas : R2
4
5 class A:
6     def method_a(self):
7         print("Method A")
8
9 class B(A):
10     def method_b(self):
11         print("Method B")
12
13 class C(A):
14     def method_c(self):
15         print("Method C")
16
17 class D(B, C):
18     def method_d(self):
19         print("Method D")
20
21 my_d = D()
22 my_d.method_a()
23 my_d.method_b()
24 my_d.method_c()
25 my_d.method_d()
```

The TERMINAL panel at the bottom shows the output of the program, which is the result of running the code in 'Hybrid1.py'.

```
ments/KULIAH/SEMESTER 4/PBO-2-Lanjut-main/PERTEMUAN-2/Praktikum/Hybrid1.py"
Method A
Method B
Method C
Method D
PS C:\Users\Toshiba\Documents\KULIAH\SEMESTER 4\PBO-2-Lanjut-main>
```

Hybrid 2 :

```
#Nama : Farhan Saefulah
#NIM : 210511059
#Kelas : R2

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"

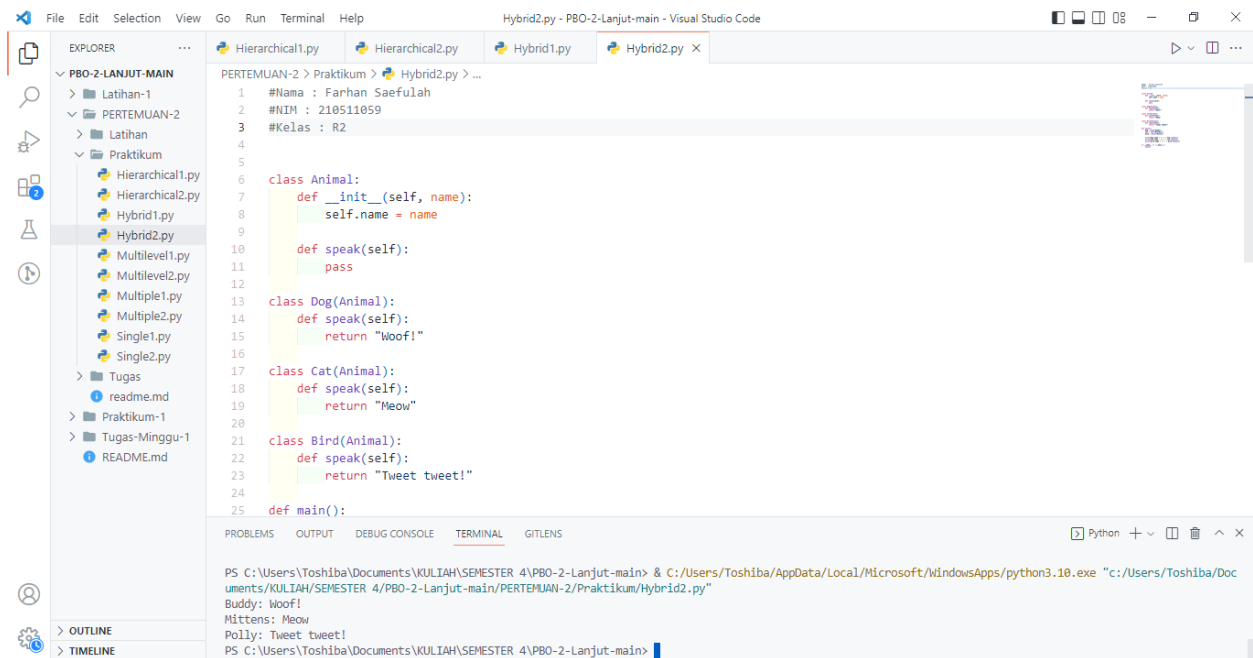
class Bird(Animal):
    def speak(self):
        return "Tweet tweet!"

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



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

- Explorer Panel:** Displays a file tree for a project named "PBO-2-LANJUT-MAIN". The tree includes folders for "Latihan-1", "PERTEMUAN-2", "Latihan", and "Praktikum". The "Praktikum" folder is expanded, showing files like "Hierarchical1.py", "Hierarchical2.py", "Hybrid1.py", "Hybrid2.py", "Multilevel1.py", "Multilevel2.py", "Multiple1.py", "Multiple2.py", "Single1.py", and "Single2.py".
- Editor Panel:** Shows the code for "Hybrid2.py". The code defines a base class "Animal" with an ".__init__" method and a "speak" method. It then defines three subclasses: "Dog", "Cat", and "Bird", each inheriting from "Animal" and implementing the "speak" method with specific sounds ("Woof!", "Meow", and "Tweet tweet!" respectively). A "main" function is also present.
- Terminal Panel:** Shows the output of running the script. The command executed is "PS C:\Users\Toshiba\Documents\KULIAH\SEMESTER 4\PBO-2-Lanjut-main> & C:/Users/Toshiba/AppData/Local/Microsoft/WindowsApps/python3.10.exe "c:/Users/Toshiba/Documents/KULIAH/SEMESTER 4/PBO-2-Lanjut-main/PERTEMUAN-2/Praktikum/Hybrid2.py"". The output shows the execution of the "main" function, which creates instances of "Dog", "Cat", and "Bird" and calls their "speak" methods, resulting in the output: "Buddy: Woof!", "Mittens: Meow", and "Polly: Tweet tweet!".

```
1 #Nama : Farhan Saefulah
2 #NIM : 210511059
3 #Kelas : R2
4
5
6 class Animal:
7     def __init__(self, name):
8         self.name = name
9
10    def speak(self):
11        pass
12
13    class Dog(Animal):
14        def speak(self):
15            return "Woof!"
16
17    class Cat(Animal):
18        def speak(self):
19            return "Meow"
20
21    class Bird(Animal):
22        def speak(self):
23            return "Tweet tweet!"
24
25    def main():
26
27        # Create instances
28        dog = Dog("Buddy")
29        cat = Cat("Mittens")
30        bird = Bird("Polly")
31
32        # Call speak method
33        dog.speak()
34        cat.speak()
35        bird.speak()
36
37        # Print output
38        print("Buddy: Woof!")
39        print("Mittens: Meow")
40        print("Polly: Tweet tweet!")
41
42    main()
```

```
PS C:\Users\Toshiba\Documents\KULIAH\SEMESTER 4\PBO-2-Lanjut-main> & C:/Users/Toshiba/AppData/Local/Microsoft/WindowsApps/python3.10.exe "c:/Users/Toshiba/Documents/KULIAH/SEMESTER 4/PBO-2-Lanjut-main/PERTEMUAN-2/Praktikum/Hybrid2.py"
Buddy: Woof!
Mittens: Meow
Polly: Tweet tweet!
PS C:\Users\Toshiba\Documents\KULIAH\SEMESTER 4\PBO-2-Lanjut-main>
```


Multilevel 1 :

```
#Nama : Farhan Saefulah
#NIM : 210511059
#Kelas : R2

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

Output 5

The screenshot displays the Visual Studio Code interface for a Python project named "PBO-2-LANJUT-MAIN". The Explorer sidebar on the left shows a directory structure with folders "Latihan-1", "PERTEMUAN-2", and "Praktikum". The "Praktikum" folder contains several Python files, including "Multilevel1.py", which is currently open in the editor.

The code in "Multilevel1.py" defines a class hierarchy:

```
1 #Nama : Farhan Saefulah
2 #NIM : 210511059
3 #Kelas : R2
4
5
6 class Animal:
7     def __init__(self, species):
8         self.species = species
9
10    def eat(self):
11        print("The animal is eating.")
12
13    class Pet(Animal):
14        def __init__(self, name, species):
15            super().__init__(species)
16            self.name = name
17
18        def play(self):
19            print("The pet is playing.")
20
21    class cat(Pet):
22        def __init__(self, name, breed):
23            super().__init__(name, "Canine")
24            self.breed = breed
25
```

Below the code editor, the TERMINAL panel shows the output of running the script:

```
Species: Canine
Name: Ane
The animal is eating.
The pet is playing.
Meow! Meow!
PS C:\Users\Toshiba\Documents\KULIAH\SEMESTER 4\PBO-2-Lanjut-main>
```

Multilevel 2 :

```
#Nama : Farhan Saefulah
#NIM : 210511059
#Kelas : R2

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

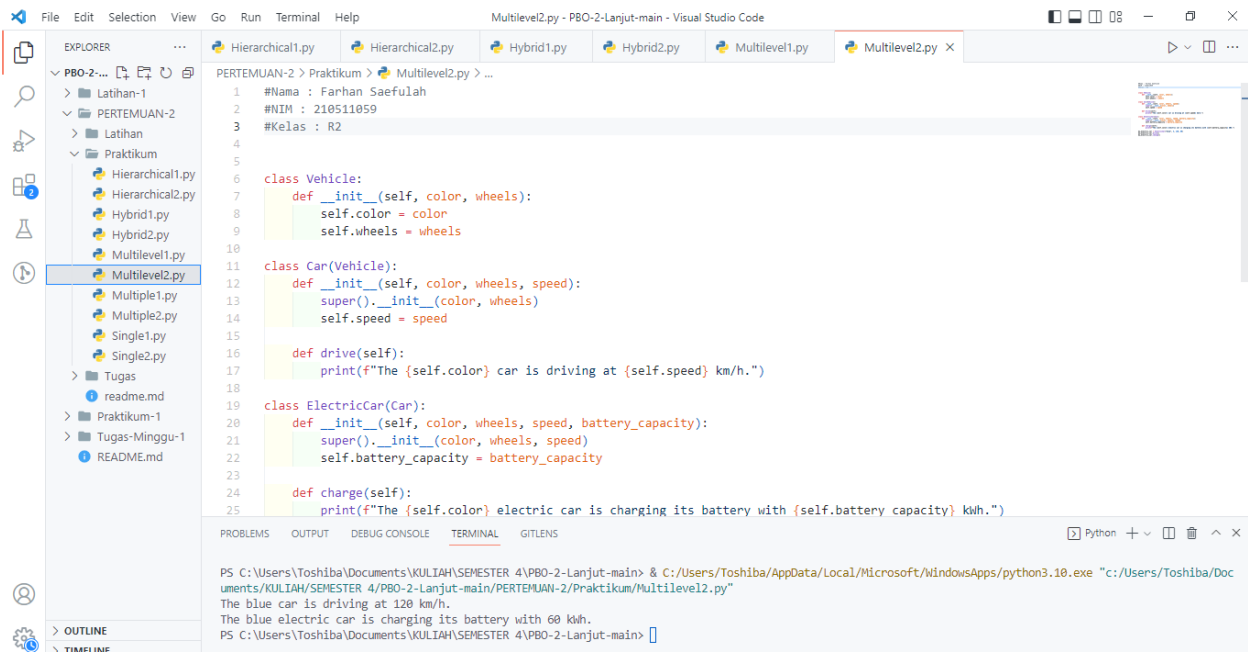
    def drive(self):
        print(f"The {self.color} car is driving at {self.speed} km/h.")

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

Output



The screenshot displays the Visual Studio Code interface with the file `Multilevel2.py` open. The Explorer sidebar on the left shows the project structure, including folders like `PERTEMUAN-2` and `Praktikum`, and files like `Hierarchical1.py`, `Hierarchical2.py`, `Hybrid1.py`, `Hybrid2.py`, `Multilevel1.py`, and `Multilevel2.py`.

The main editor shows the following Python code:

```
1 #Nama : Farhan Saefulah
2 #NIM : 210511059
3 #Kelas : R2
4
5
6 class Vehicle:
7     def __init__(self, color, wheels):
8         self.color = color
9         self.wheels = wheels
10
11 class Car(Vehicle):
12     def __init__(self, color, wheels, speed):
13         super().__init__(color, wheels)
14         self.speed = speed
15
16     def drive(self):
17         print(f"The {self.color} car is driving at {self.speed} km/h.")
18
19 class ElectricCar(Car):
20     def __init__(self, color, wheels, speed, battery_capacity):
21         super().__init__(color, wheels, speed)
22         self.battery_capacity = battery_capacity
23
24     def charge(self):
25         print(f"The {self.color} electric car is charging its battery with {self.battery_capacity} kWh.")
```

The TERMINAL panel at the bottom shows the execution output:

```
PS C:\Users\Toshiba\Documents\KULIAH\SEMESTER 4\PBO-2-Lanjut-main> & C:/Users/Toshiba/AppData/Local/Microsoft/WindowsApps/python3.10.exe "c:/Users/Toshiba/Doc
uments/KULIAH/SEMESTER 4/PBO-2-Lanjut-main/PERTEMUAN-2/Praktikum/Multilevel2.py"
The blue car is driving at 120 km/h.
The blue electric car is charging its battery with 60 kWh.
PS C:\Users\Toshiba\Documents\KULIAH\SEMESTER 4\PBO-2-Lanjut-main>
```

Multiple 1 :

```
#Nama : Farhan Saefulah
#NIM : 210511059
#Kelas : R2

class Tenaga :
    def setTenaga (self,tenaga):
        self.tenaga = tenaga

    def showTenaga(self):
        print(self.tenaga)

class Kelompok :
    def setKelompok(self,kelompok):
        self.kelompok = kelompok

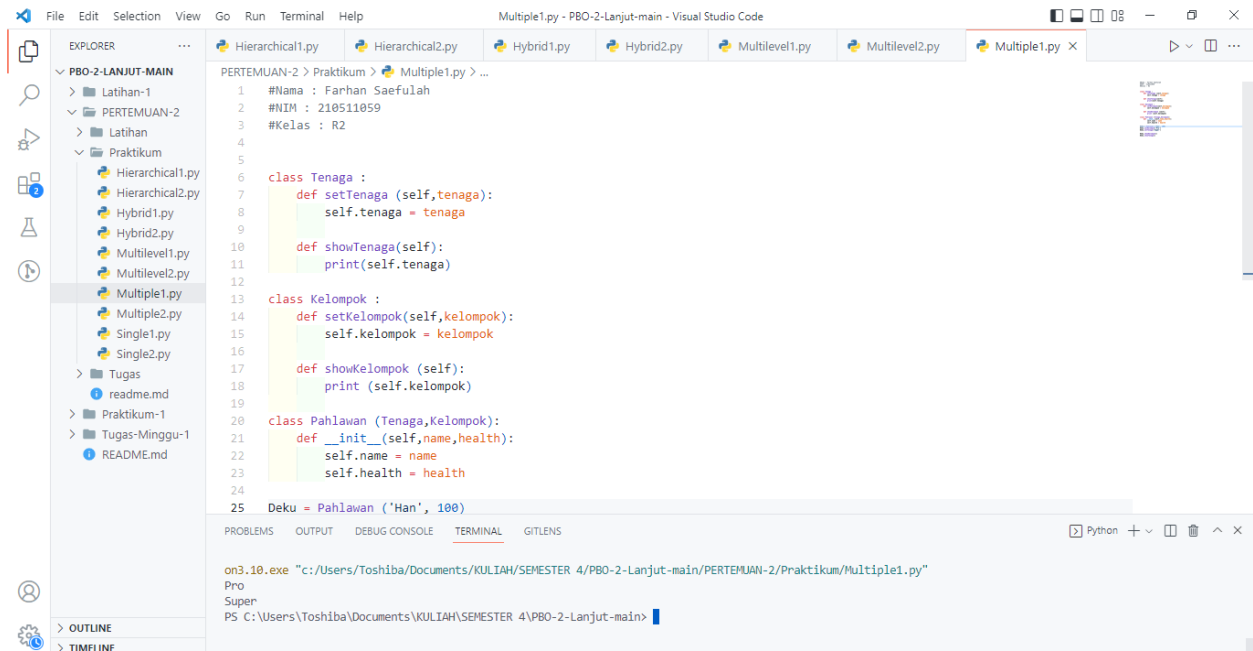
    def showKelompok (self):
        print (self.kelompok)

class Pahlawan (Tenaga,Kelompok):
    def __init__(self,name,health):
        self.name = name
        self.health = health

Deku = Pahlawan ('Han', 100)
Deku.setKelompok('Pro')
Deku.setTenaga('Super')

Deku.showKelompok()
Deku.showTenaga()
```

Output



The screenshot displays the Visual Studio Code interface. The Explorer sidebar on the left shows a project named 'PBO-2-LANJUT-MAIN' with a folder structure including 'Latihan-1', 'PERTEMUAN-2', 'Latihan', and 'Praktikum'. The 'Praktikum' folder contains several Python files, with 'Multiple1.py' selected. The main editor area shows the code for 'Multiple1.py', which defines three classes: 'Tenaga', 'Kelompok', and 'Pahlawan'. The 'Tenaga' class has methods 'setTenaga' and 'showTenaga'. The 'Kelompok' class has methods 'setKelompok' and 'showKelompok'. The 'Pahlawan' class inherits from 'Tenaga' and 'Kelompok', with an '__init__' method that sets 'name' and 'health'. An instance 'Deku' of the 'Pahlawan' class is created with 'Han' as the name and 100 as health. The bottom panel shows the 'TERMINAL' tab with the command 'on3.10.exe "c:/Users/Toshiba/Documents/KULIAH/SEMESTER 4/PBO-2-Lanjut-main/PERTEMUAN-2/Praktikum/Multiple1.py"' and the output 'Pro Super'.

```
1 #Nama : Farhan Saefu1ah
2 #NIM : 210511059
3 #Kelas : R2
4
5
6 class Tenaga :
7     def setTenaga (self,tenaga):
8         self.tenaga = tenaga
9
10    def showTenaga(self):
11        print(self.tenaga)
12
13 class Kelompok :
14     def setKelompok(self,kelompok):
15         self.kelompok = kelompok
16
17     def showKelompok (self):
18         print (self.kelompok)
19
20 class Pahlawan (Tenaga,Kelompok):
21     def __init__(self,name,health):
22         self.name = name
23         self.health = health
24
25 Deku = Pahlawan ('Han', 100)
```

on3.10.exe "c:/Users/Toshiba/Documents/KULIAH/SEMESTER 4/PBO-2-Lanjut-main/PERTEMUAN-2/Praktikum/Multiple1.py"
Pro
Super
PS C:\Users\Toshiba\Documents\KULIAH\SEMESTER 4\PBO-2-Lanjut-main>

Multiple 2 :

```
#Nama : Farhan Saefulah
#NIM : 210511059
#Kelas : R2

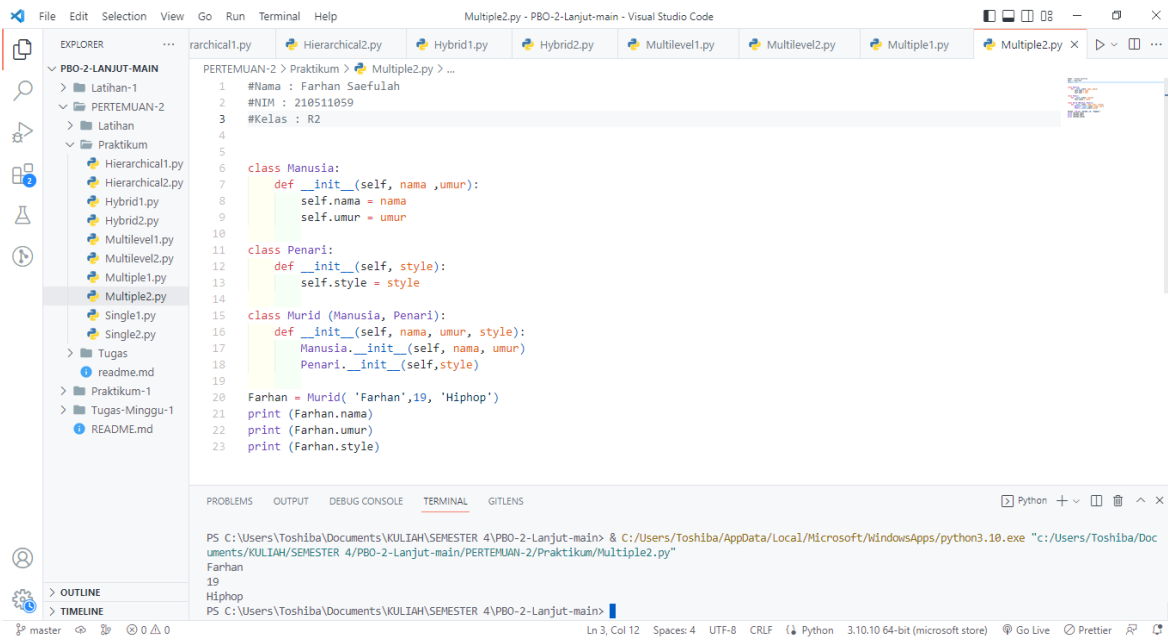
class Manusia:
    def __init__(self, nama ,umur):
        self.nama = nama
        self.umur = umur

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

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

Farhan = Murid( 'Farhan',19, 'Hiphop')
print (Farhan.nama)
print (Farhan.umur)
print (Farhan.style)
```

Output Screenshot



Single 1 :

```
#Nama : Farhan Saefulah
#NIM : 210511059
#Kelas : R2

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

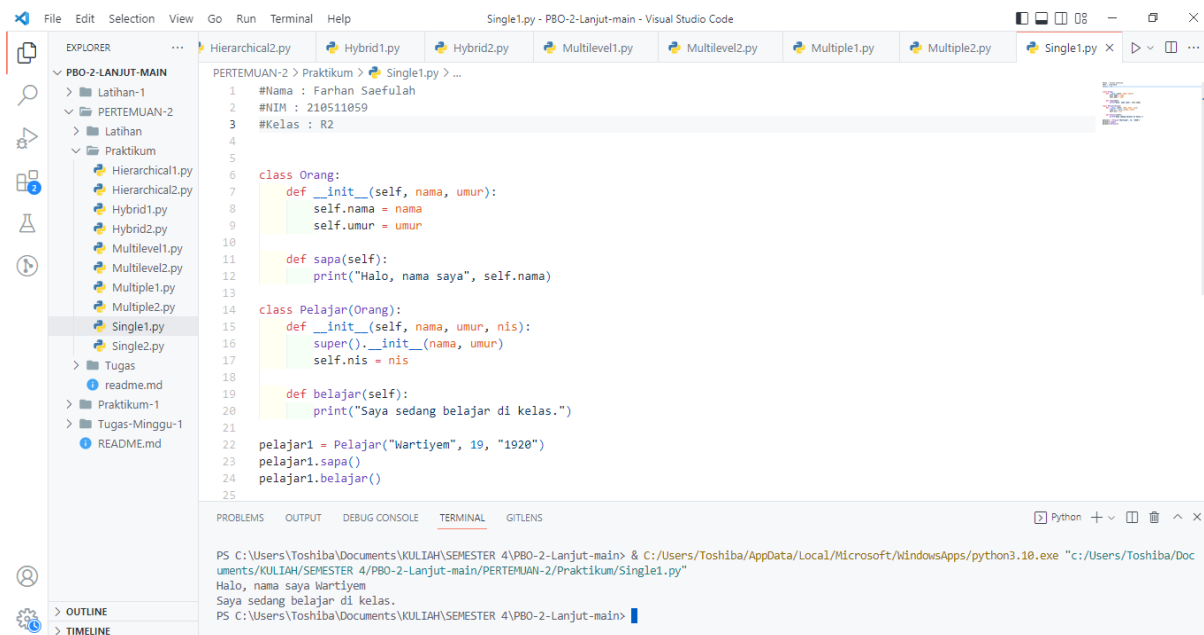
    def sapa(self):
        print("Halo, nama saya", self.nama)

class Pelajar(Orang):
    def __init__(self, nama, umur, nis):
        super().__init__(nama, umur)
        self.nis = nis

    def belajar(self):
        print("Saya sedang belajar di kelas.")

pelajar1 = Pelajar("Wartiyem", 19, "1920")
pelajar1.sapa()
pelajar1.belajar()
```


Output



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left displays a project structure for 'PBO-2-LANJUT-MAIN' with folders 'Latihan-1', 'PERTEMUAN-2', and 'Praktikum'. The 'Praktikum' folder contains several Python files, including 'Single1.py'. The main editor window shows the code for 'Single1.py', which defines two classes: 'Orang' and 'Pelajar'. The 'Orang' class has an '.__init__' method and a 'sapa' method. The 'Pelajar' class inherits from 'Orang' and has an '.__init__' method and a 'belajar' method. The terminal at the bottom shows the execution of the code, displaying the output: 'Halo, nama saya Wartiyem' and 'Saya sedang belajar di kelas.'

```
1 #Nama : Farhan Saefulah
2 #NIM : 210511059
3 #Kelas : R2
4
5
6 class Orang:
7     def __init__(self, nama, umur):
8         self.nama = nama
9         self.umur = umur
10
11     def sapa(self):
12         print("Halo, nama saya", self.nama)
13
14 class Pelajar(Orang):
15     def __init__(self, nama, umur, nis):
16         super().__init__(nama, umur)
17         self.nis = nis
18
19     def belajar(self):
20         print("Saya sedang belajar di kelas.")
21
22 pelajar1 = Pelajar("Wartiyem", 19, "1920")
23 pelajar1.sapa()
24 pelajar1.belajar()
25
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS

PS C:\Users\Toshiba\Documents\KULIAH\SEMESTER 4\PBO-2-Lanjut-main> & C:/Users/Toshiba/AppData/Local/Microsoft/WindowsApps/python3.10.exe "c:/Users/Toshiba/Doc
uments/KULIAH/SEMESTER 4/PBO-2-Lanjut-main/PERTEMUAN-2/Praktikum/Single1.py"
Halo, nama saya Wartiyem
Saya sedang belajar di kelas.
PS C:\Users\Toshiba\Documents\KULIAH\SEMESTER 4\PBO-2-Lanjut-main>

Single 2 :

```
#Nama : Farhan Saefulah
#NIM : 210511059
#Kelas : R2

class Orangtua:
    def __init__(self, rambut, umur):
        self.rambut = rambut
        self.umur = umur
    def jenisRambut(self):

        print(self.rambut, "Keriting")

class Anak(Orangtua):
    def __init__(self, rambut, umur, warnaMata):
        super().__init__(rambut, umur)
        self.warnaMata = warnaMata

    def JenisKelamin(self):
        print("Laki Laki")

kucingA = Anak("Wartiyem", 15, "Ungu")
kucingA.jenisRambut()
kucingA.JenisKelamin()
```

Output

The image shows a Visual Studio Code editor window with the following details:

- File Explorer (Left):** Shows a project structure with a folder named "PBO-2-LANJUT-MAIN" containing subfolders "Latihan-1", "PERTEMUAN-2", and "Tugas". The "PERTEMUAN-2" folder contains a subfolder "Latihan" which includes a "Praktikum" folder. The "Praktikum" folder contains several Python files, including "Single2.py", which is currently selected.
- Code Editor (Center):** Displays the content of "Single2.py". The code defines a class hierarchy for a pet management system.


```

1  #Nama : Farhan Saefulah
2  #NIM : 210511059
3  #Kelas : R2
4
5  class Orangtua:
6      def __init__(self, rambut, umur):
7          self.rambut = rambut
8          self.umur = umur
9      def jenisRambut(self):
10         print(self.rambut, "Keriting")
11
12     class Anak(Orangtua):
13         def __init__(self, rambut, umur, warnaMata):
14             super().__init__(rambut, umur)
15             self.warnaMata = warnaMata
16
17         def jenisKelamin(self):
18             print("Laki Laki")
19
20         kucingA = Anak("Wartiyem", 15, "Ungu")
21         kucingA.jenisRambut()
22         kucingA.jenisKelamin()
      
```
- Terminal (Bottom):** Shows the command prompt output for the command `python3 Single2.py`. The output is:


```

PS C:\Users\Toshiba\Documents\KULIAH\SEMESTER 4\PBO-2-Lanjut-main> python3 Single2.py
Wartiyem Keriting
Laki Laki
      
```