LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTAS1 0 B 3EK LANJUT

2 0 2 3

Pre pa red By:

Nama : Muhammad Gifani

NIM : 210511051

Kelas : TIF21B (R2)

Praktikum 2

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

1. single1.py

```python
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("Ahmad", 16, "12345")
pelajar1.sapa()
pelajar1.belajar()
```
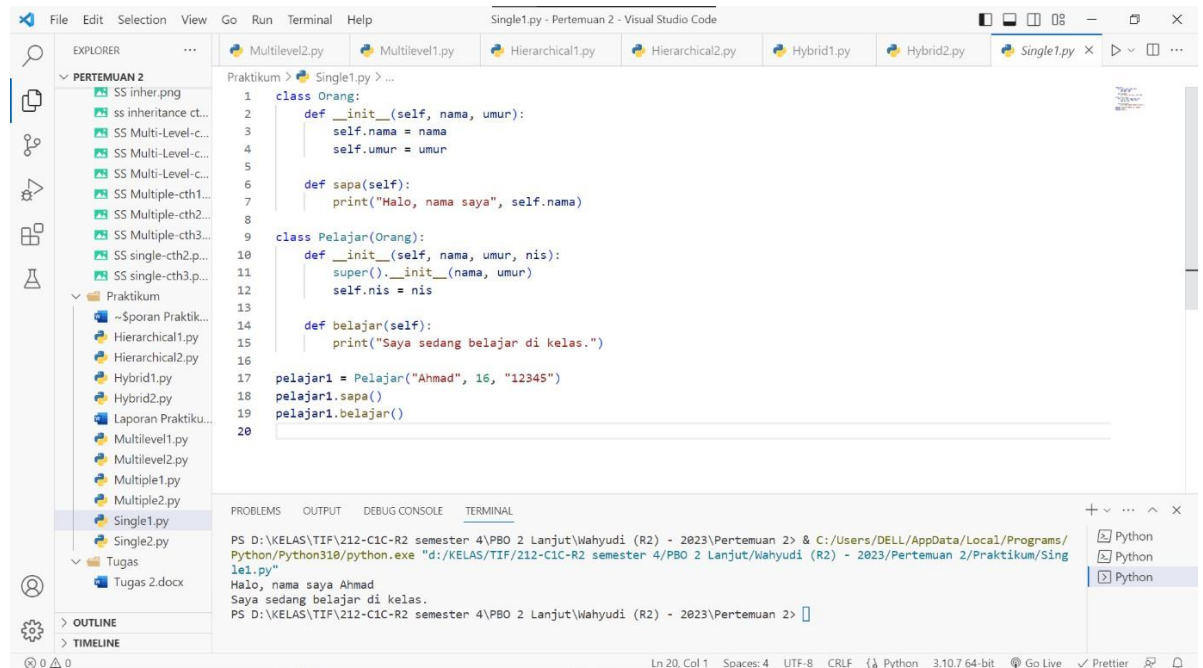
OUTPUT SS

2. single2.py

```python
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("Aldi", 12, "biru")
kucingA.jenisRambut()
kucingA.JenisKelamin()
```
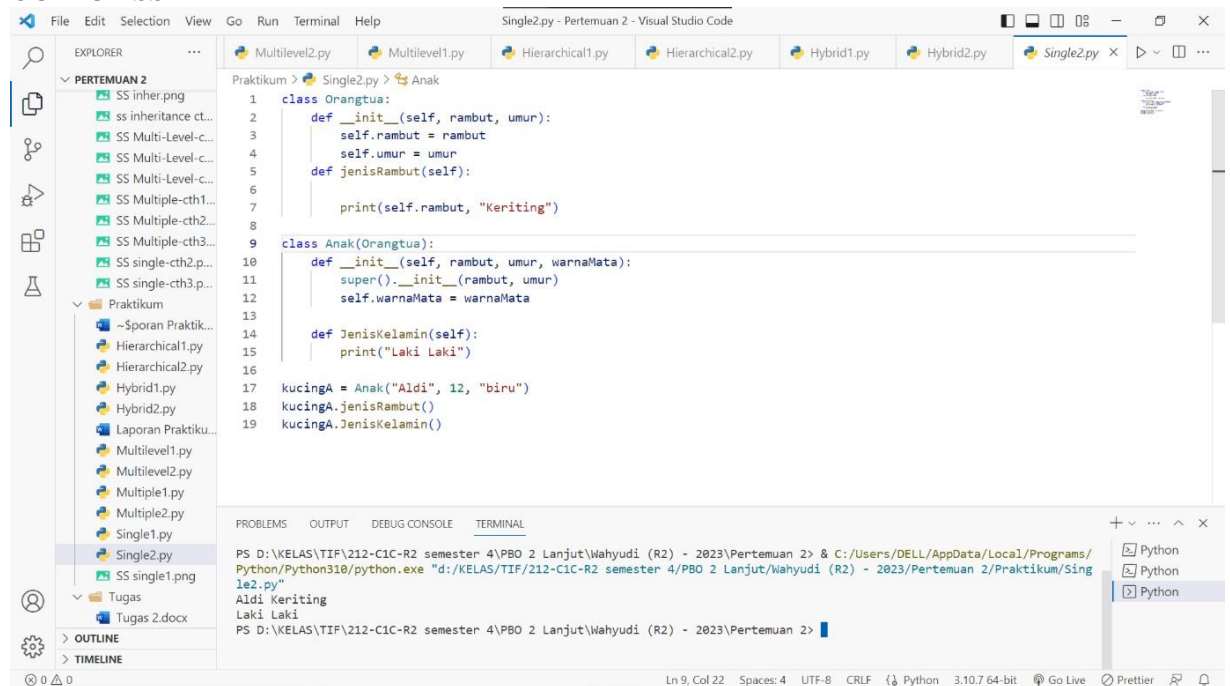
OUTPUT SS

3. multiple1.py

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


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

```python
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 ('Deku', 200)
Deku.setKelompok('Pro')
Deku.setTenaga('Super')

Deku.showKelompok()
Deku.showTenaga()
```
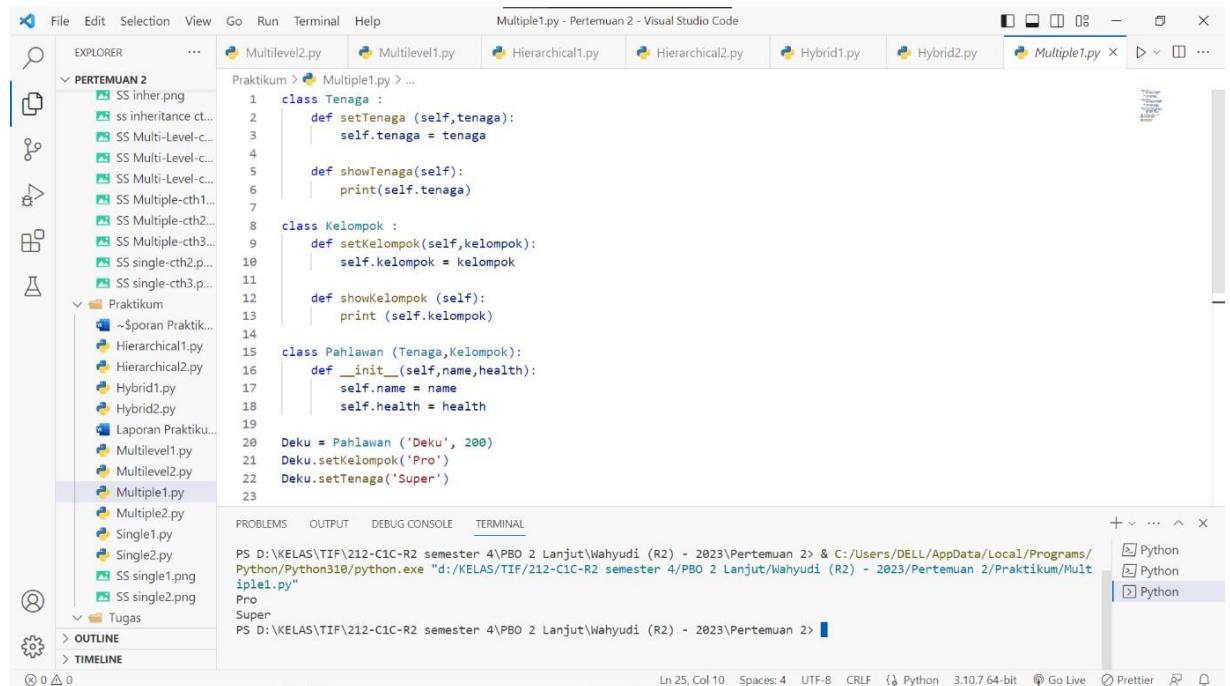
OUTPUT SS

4. multiple2.py,

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

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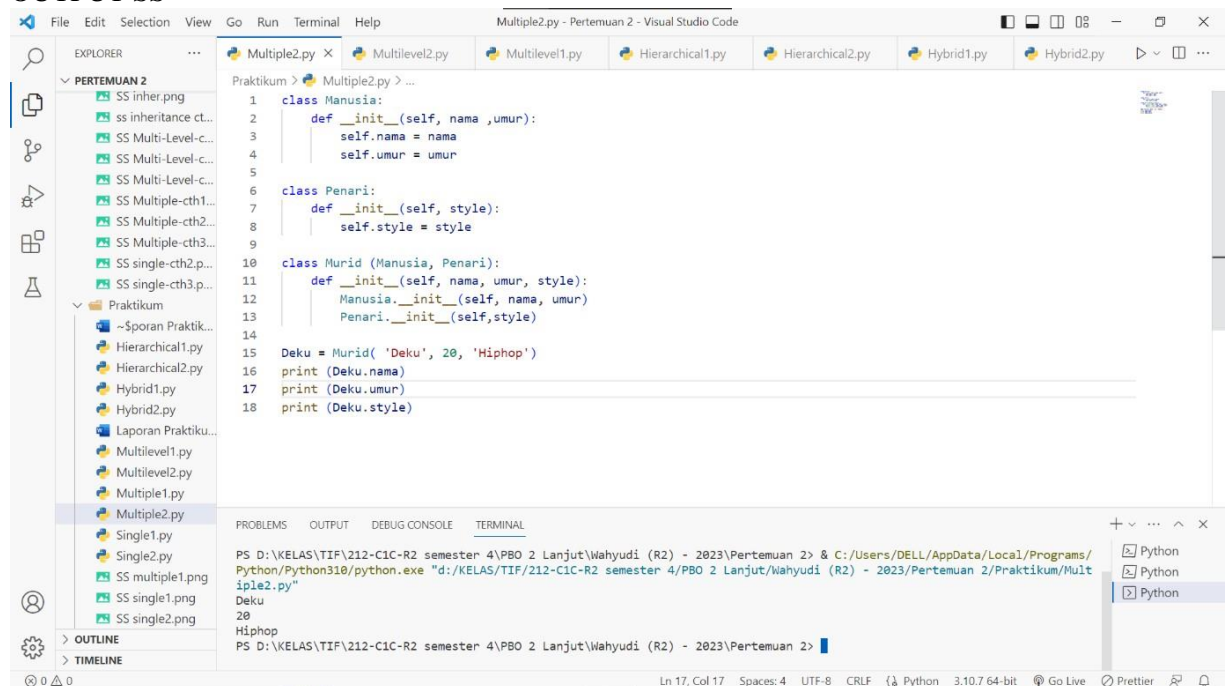
```
            self.style = style

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

Deku = Murid( 'Deku', 20, 'Hiphop')
print (Deku.nama)
print (Deku.umur)
print (Deku.style)
```

OUTPUT SS

5. hierarchical1.Py

```python
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

6. hierarchical2.py

```python
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):
```

```python
    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()
```
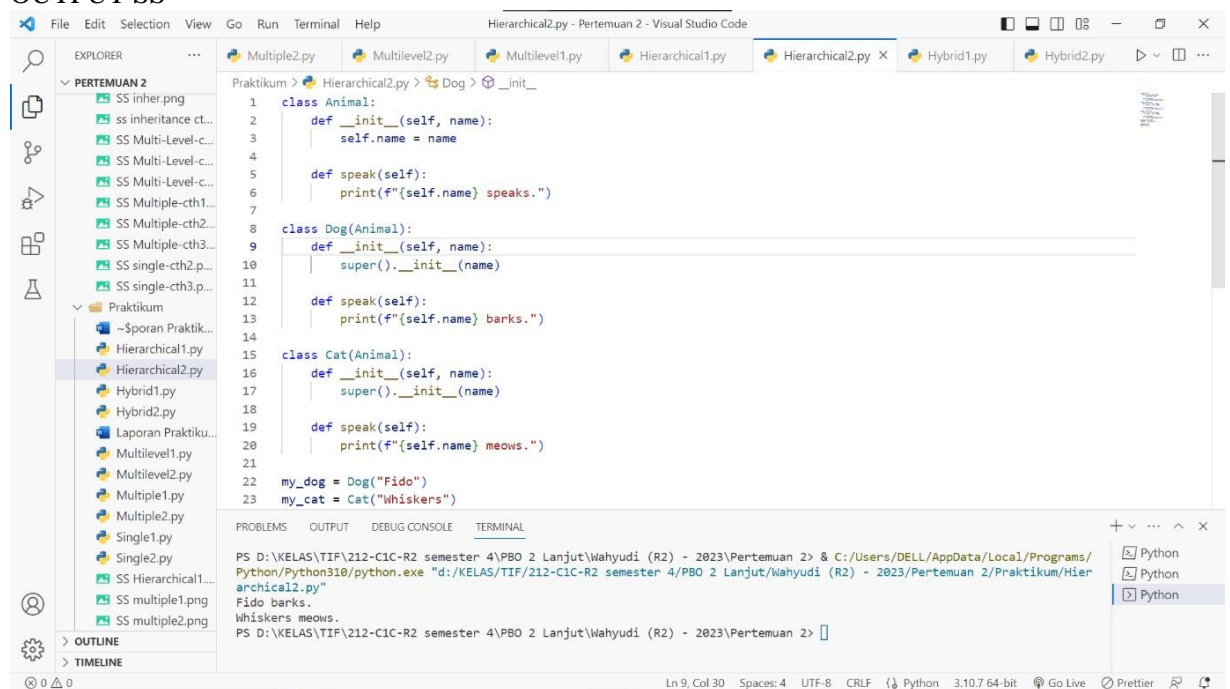
OUTPUT SS

7. multilevel1.py

```python
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.")
```

```python
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 SS

8. multilevel2.py

```python
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
```

```python
    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 SS

9. hybrid1.py

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

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

class C(A):
```
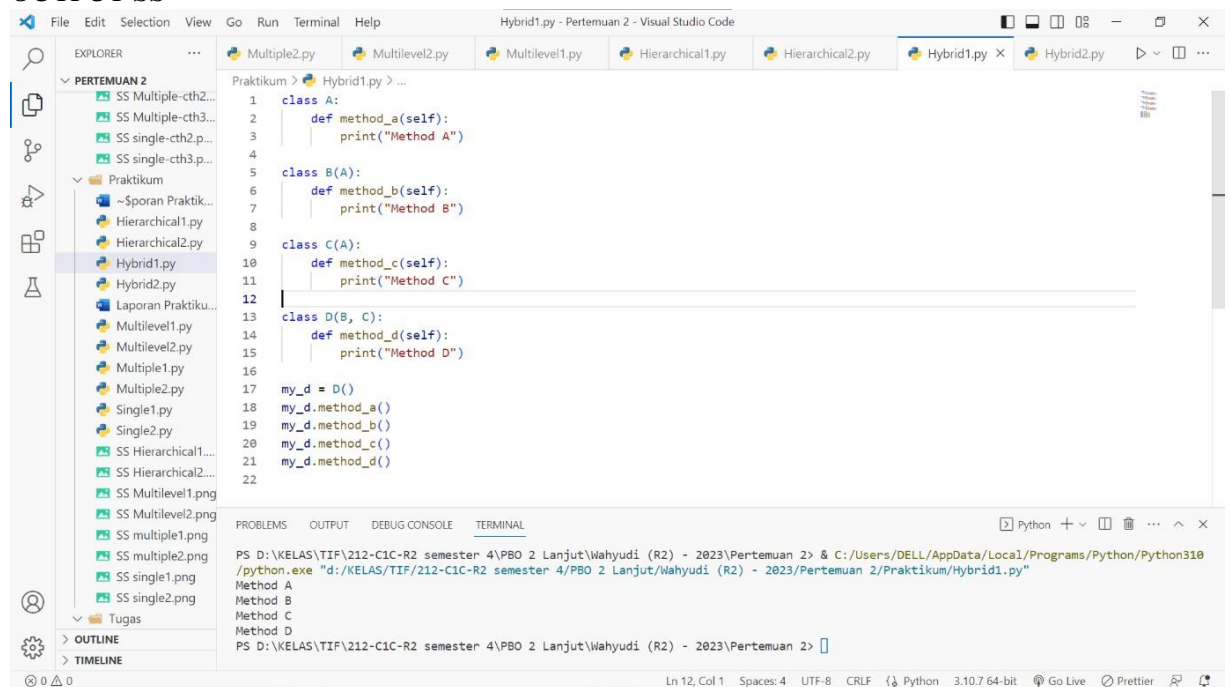
```python
    def method_c(self):
        print("Method C")

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

10. hybrid2.py

```python
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):
```

```python
        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