

LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK LANJUT

2023



Prepared By:

Linda Novita Juliyanti

210510003

D3

Praktikum :

Buatlah masing-masing 2 contoh polymorphism statis (overload) dan polymorphism dinamis (overriding). Beri nama overload1.py, overload2, overriding1.py, overriding2.py

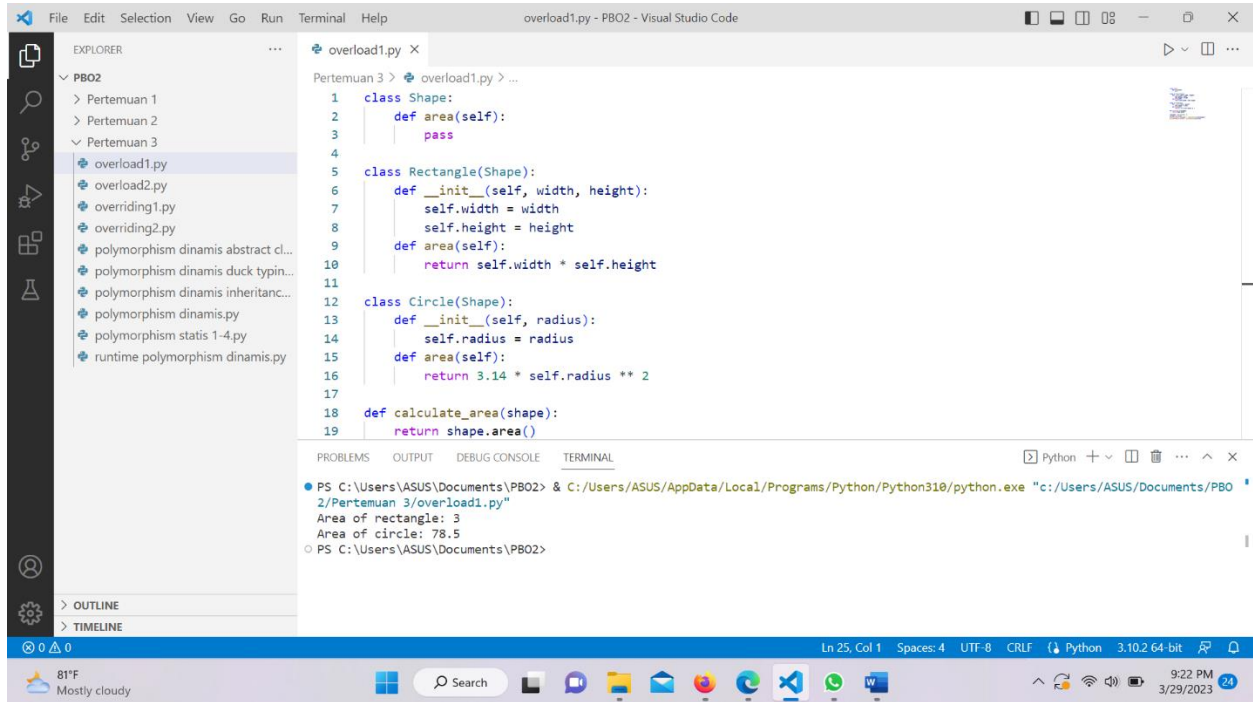
Jawaban Praktikum :

1. Script overload1.py dan overload2.py

Overload1.py

```
# LINDA NOVITA JULIYANTI  
# 210510003  
# D3
```

```
class Shape:  
    def area(self):  
        pass  
  
class Rectangle(Shape):  
    def __init__(self, width, height):  
        self.width = width  
        self.height = height  
    def area(self):  
        return self.width * self.height  
  
class Circle(Shape):  
    def __init__(self, radius):  
        self.radius = radius  
    def area(self):  
        return 3.14 * self.radius ** 2  
  
def calculate_area(shape):  
    return shape.area()  
  
rectangle = Rectangle(1, 3)  
circle = Circle(5)  
print("Area of rectangle:", calculate_area(rectangle))  
print("Area of circle:", calculate_area(circle))  
  
# output  
Area of rectangle: 3  
Area of circle: 78.5
```



Overload2.py

```
# LINDA NOVITA JULIYANTI
# 210510003
# D3
```

```
class Shape:
    def area(self):
        pass

class Rectangle(Shape):
    def __init__(self, width, height):
        self.width = width
        self.height = height
    def area(self):
        return self.width * self.height

class Circle(Shape):
    def __init__(self, radius):
        self.radius = radius
    def area(self):
        return 3.14 * self.radius * self.radius
```

```
class Triangle(Shape):
    def __init__(self, base, height):
        self.base = base
        self.height = height
    def area(self):
        return 0.5 * self.base * self.height
```

```
r = Rectangle(10, 20)
c = Circle(5)
t = Triangle(8, 12)
```

```
print("Area of rectangle:", r.area())
print("Area of circle:", c.area())
print("Area of triangle:", t.area())
```

output

Area of rectangle: 200

Area of circle: 78.5

Area of triangle: 48.0

The screenshot shows a Visual Studio Code window with a file explorer on the left and a code editor in the center. The file explorer shows a project named 'PBO2' with several files, including 'overload2.py'. The code editor displays the following Python code:

```
5 class Shape:
6     def area(self):
7         pass
8
9 class Rectangle(Shape):
10    def __init__(self, width, height):
11        self.width = width
12        self.height = height
13    def area(self):
14        return self.width * self.height
15
16 class Circle(Shape):
17    def __init__(self, radius):
18        self.radius = radius
19    def area(self):
20        return 3.14 * self.radius * self.radius
21
22 class Triangle(Shape):
23    def __init__(self, base, height):
```

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

```
PS C:\Users\ASUS\Documents\PBO2> & C:/Users/ASUS/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/ASUS/Documents/PBO2/Pertemuan 3/overload2.py"
Area of rectangle: 200
Area of circle: 78.5
Area of triangle: 48.0
PS C:\Users\ASUS\Documents\PBO2>
```

2. Script overriding1.py dan overriding2.py

Overriding1.py

```
# LINDA NOVITA JULIYANTI  
# 210510003  
# D3
```

```
from abc import ABC, abstractmethod
```

```
class Shape(ABC):  
    @abstractmethod  
    def draw(self):  
        pass
```

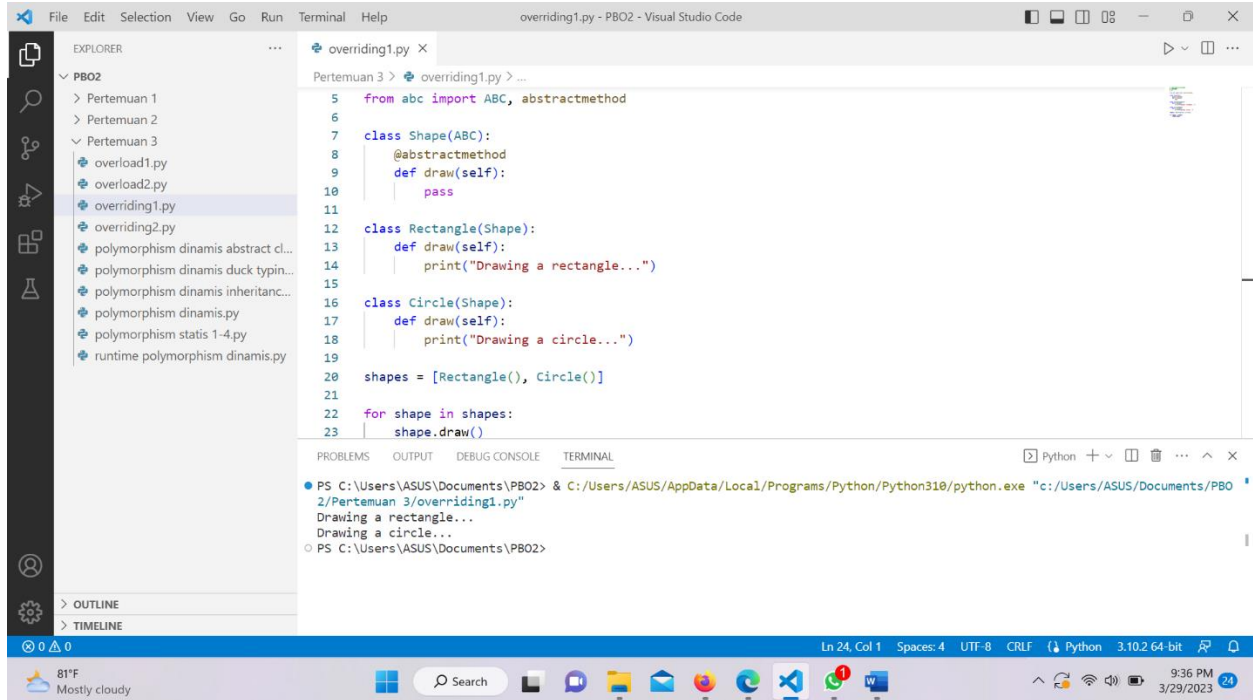
```
class Rectangle(Shape):  
    def draw(self):  
        print("Drawing a rectangle...")
```

```
class Circle(Shape):  
    def draw(self):  
        print("Drawing a circle...")
```

```
shapes = [Rectangle(), Circle()]
```

```
for shape in shapes:  
    shape.draw()
```

```
# output  
Drawing a rectangle...  
Drawing a circle...
```



Overriding2.py

LINDA NOVITA JULIYANTI

210510003

D3

```
from abc import ABC, abstractmethod
```

```
class Kendaraan(ABC):
```

```
    @abstractmethod
```

```
    def start(self):
```

```
        pass
```

```
class Mobil(Kendaraan):
```

```
    def start(self):
```

```
        print("Menjalankan mesin mobil...")
```

```
class Motor(Kendaraan):
```

```
    def start(self):
```

```
        print("Menjalankan mesin motor...")
```

```
class Sportcar(Kendaraan):
```

```
    def start(self):
```

```
        print("Menjalankan mesin sportcar...")
```

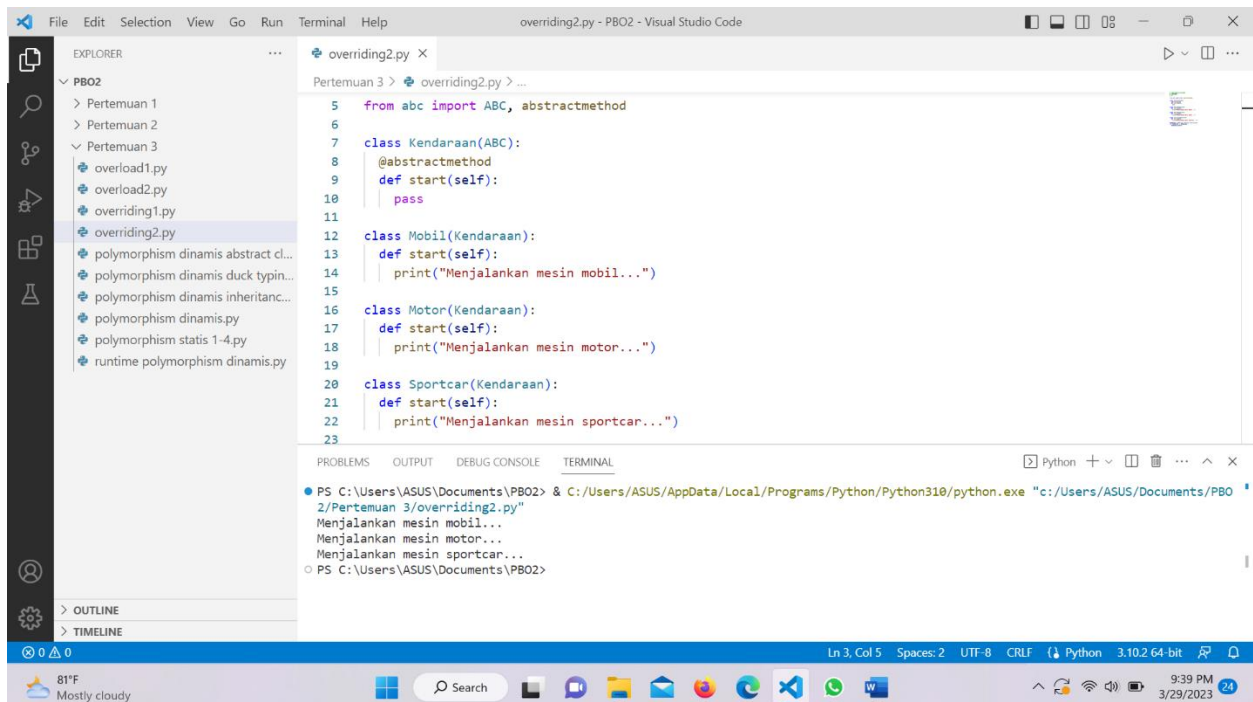
```
kendaraan = [Mobil(), Motor(), Sportcar()]
for kendaraan in kendaraan:
    kendaraan.start()
```

output

Menjalankan mesin mobil...

Menjalankan mesin motor...

Menjalankan mesin sportcar...



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left shows a project named 'PBO2' with several files, including 'overriding2.py'. The main editor displays the code for 'overriding2.py', which defines an abstract class 'Kendaraan' and three subclasses: 'Mobil', 'Motor', and 'Sportcar'. Each subclass has a 'start' method that prints a specific message. The Terminal pane at the bottom shows the output of running the script, which matches the expected output shown above the screenshot.

```
5 from abc import ABC, abstractmethod
6
7 class Kendaraan(ABC):
8     @abstractmethod
9     def start(self):
10         pass
11
12 class Mobil(Kendaraan):
13     def start(self):
14         print("Menjalankan mesin mobil...")
15
16 class Motor(Kendaraan):
17     def start(self):
18         print("Menjalankan mesin motor...")
19
20 class Sportcar(Kendaraan):
21     def start(self):
22         print("Menjalankan mesin sportcar...")
23
```

```
PS C:\Users\ASUS\Documents\PBO2> & C:\Users\ASUS\AppData\Local\Programs\Python\Python310\python.exe "c:/Users/ASUS/Documents/PBO2/2/Pertemuan 3/overriding2.py"
Menjalankan mesin mobil...
Menjalankan mesin motor...
Menjalankan mesin sportcar...
PS C:\Users\ASUS\Documents\PBO2>
```

Soal Tugas :

Buatlah sebuah aplikasi sederhana untuk memperdengarkan suara 10 hewan yang berbeda.

Format suara bisa .wav atau mp3

Jawaban Tugas :

https://github.com/lindanovitaj/pemrograman_berorientasi_objek2/tree/main/tugas3