

LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK LANJUT

2023



Prepared By:

NAMA	: TITO FEBRI ANGGARA
NIM	: 210511126
KELAS	: R3(C)

PRAKTIKUM 7

Ketupat.py

```
class KetupatMeta(type):

    def __init__(cls, name, bases, attrs):

        super().__init__(name, bases, attrs)

        # Tambahkan method untuk menghitung luas dan volume ketupat

        def luas(cls, D1, D2):

            return 1/2 * D1 * D2

        cls.luas = classmethod(luas)

        def volume(cls, D1, D2, tinggi):

            return 1/2 * D1 * D2 * tinggi

        cls.volume = classmethod(volume)

class Ketupat(metaclass=KetupatMeta):

    pass

t = Ketupat()

# Menghitung luas permukaan ketupat dengan D1=4 dan D2=5

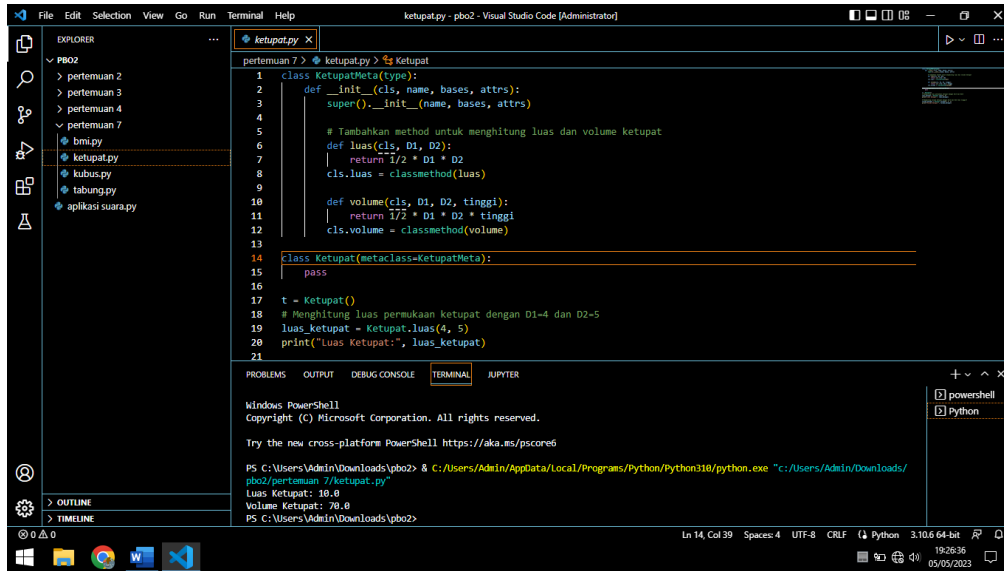
luas_ketupat = Ketupat.luas(4, 5)

print("Luas Ketupat:", luas_ketupat)

# Menghitung volume ketupat dengan D1=4 dan D2=5 dan tinggi=7

volume_ketupat = Ketupat.volume(4, 5, 7)
```

```
print("Volume Ketupat:", volume_ketupat)
```



Kubus.py

```
class KubusMeta(type):
```

```
def __init__(cls, name, bases, attrs):
    super().__init__(name, bases, attrs)
```

```
# Tambahkan method untuk menghitung luas dan volume kubus
```

```
def luas(cls, sisi):
```

```
    return sisi * sisi
```

```
cls.luas = classmethod(luas)
```

```
def volume(cls, sisi):
```

```
    return sisi * sisi * sisi
```

```
cls.volume = classmethod(volume)
```

```
class Kubus(metaclass=KubusMeta):
```

```
    pass
```

```
k = Kubus()
```

```
# Menghitung luas segitiga dengan sisi=10
```

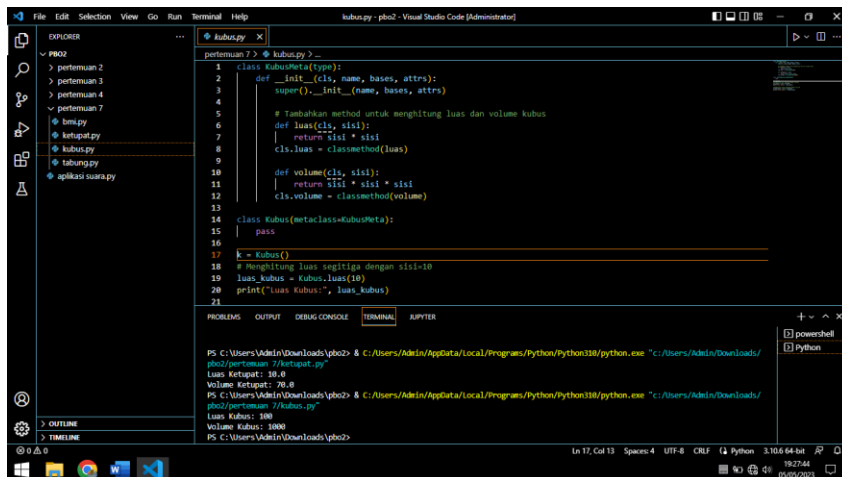
```
luas_kubus = Kubus.luas(10)
```

```
print("Luas Kubus:", luas_kubus)
```

```
# Menghitung volume segitiga dengan sisi=10
```

```
volume_kubus = Kubus.volume(10)
```

```
print("Volume Kubus:", volume_kubus)
```



The screenshot shows the Visual Studio Code editor with a file named 'kubus.py' open. The code defines a class 'KubusMeta' with methods 'luas' and 'volume', and a class 'Kubus' that inherits from 'KubusMeta'. The 'Kubus' class has an instance 'k' and calls the 'luas' and 'volume' methods with the argument 10. The terminal output shows the execution of the script, displaying the calculated values for 'luas_kubus' (100.0) and 'volume_kubus' (1000.0).

```
1 class KubusMeta(type):
2     def __init__(cls, name, bases, attrs):
3         super().__init__(name, bases, attrs)
4
5         # Tambahkan method untuk menghitung luas dan volume kubus
6     def luas(cls, sisi):
7         return sisi * sisi
8     cls.luas = classmethod(luas)
9
10    def volume(cls, sisi):
11        return sisi * sisi * sisi
12    cls.volume = classmethod(volume)
13
14 class Kubus(metaclass=KubusMeta):
15     pass
16
17 k = Kubus()
18 # Menghitung luas segitiga dengan sisi=10
19 luas_kubus = Kubus.luas(10)
20 print("Luas Kubus:", luas_kubus)
21
```

Terminal Output:

```
PS C:\Users\Adin\Downloads\gho2> & C:\Users\Adin\AppData\Local\Programs\Python\Python38\python.exe "c:/Users/Adin/Downloads/gho2/permainan 7/ketapat.py"
Luas Ketapat: 100.0
Volume Ketapat: 1000.0
PS C:\Users\Adin\Downloads\gho2> & C:\Users\Adin\AppData\Local\Programs\Python\Python38\python.exe "c:/Users/Adin/Downloads/gho2/permainan 7/kubus.py"
Luas Kubus: 1000
Volume Kubus: 10000
PS C:\Users\Adin\Downloads\gho2>
```

Tabung.py

```
class TabungMeta(type):
```

```
def __init__(cls, name, bases, attrs):
```

```
    super().__init__(name, bases, attrs)
```

```
# Tambahkan method untuk menghitung luas dan volume tabung
```

```
def luas(cls, jari):
```

```
    return 3.14 * jari * jari
```

```
cls.luas = classmethod(luas)
```

```
def volume(cls, jari, tinggi):
```

```
    return 3.14 *jari * jari * tinggi
```

```
cls.volume = classmethod(volume)
```

```
class Tabung(metaclass=TabungMeta):
```

```
    pass
```

```
s = Tabung()
```

```
# Menghitung luas permukaan tabung dengan jari=5
```

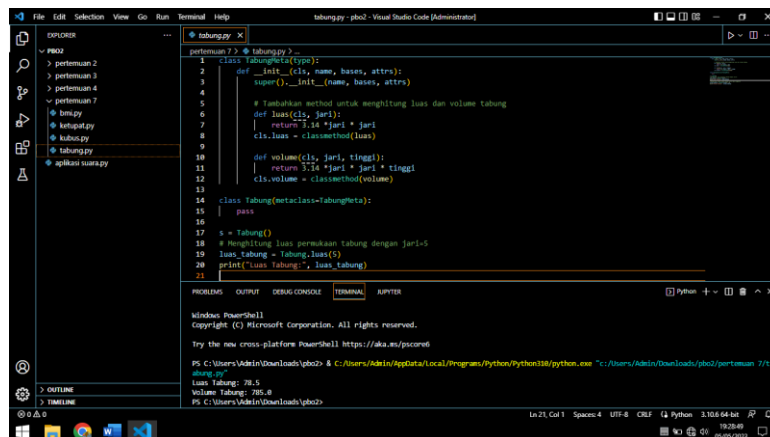
```
luas_tabung = Tabung.luas(5)
```

```
print("Luas Tabung:", luas_tabung)
```

```
# Menghitung volume tabung dengan jari=5 dan tinggi=10
```

```
volume_tabung = Tabung.volume(5, 10)
```

```
print("Volume Tabung:", volume_tabung)
```



The screenshot shows the Visual Studio Code interface with a file named 'tabung.py' open. The code defines a class 'Tabung' using the 'metaclass=TabungMeta' pattern. It includes methods for calculating the surface area ('luas') and volume ('volume') of a cylinder. The code is executed, and the output is shown in the terminal window at the bottom. The output displays the calculated surface area and volume for a cylinder with a radius of 5 and a height of 10.

```
1 class TabungMeta(type):
2     def __init__(cls, name, bases, attrs):
3         super().__init__(name, bases, attrs)
4
5         # Tambahkan method untuk menghitung luas dan volume tabung
6     def luas(cls, jari):
7         return 3.14 *jari * jari
8     cls.luas = classmethod(luas)
9
10    def volume(cls, jari, tinggi):
11        return 3.14 *jari * jari * tinggi
12    cls.volume = classmethod(volume)
13
14    class Tabung(metaclass=TabungMeta):
15        pass
16
17    s = Tabung()
18    # Menghitung luas permukaan tabung dengan jari=5
19    luas_tabung = Tabung.luas(5)
20    print("Luas Tabung:", luas_tabung)
21
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/powershell

PS C:\Users\Adelin\Downloads\gho2> & C:\Users\Adelin\AppData\Local\Programs\Python\Python38\python.exe "C:/Users/Adelin/Downloads/gho2/peremuan 7/pt
abung.py"
Luas Tabung: 78.5
Volume Tabung: 785.0
PS C:\Users\Adelin\Downloads\gho2>
```