

# LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK LANJUT

2023



Prepared By:

Umar Faqih

R2(B)

210511066

## Tugas Praktikum 7 :

Buatlah 3 aplikasi untuk menghitung volume dan luas permukaan selain dari contoh diatas menggunakan teknik Metaprogramming. Hasilnya diupload ke github masing-masing di

### 1. Bola

```
import math

class Bola:
    def __init__(self, r):
        self.r = r

    def volume(self):
        return 4/3 * math.pi * self.r ** 3

    def luas(self):
        return 4 * math.pi * self.r ** 2

def create_bola_class(name, radius_attr):
    def volume(self):
        return 4/3 * math.pi * getattr(self,
radius_attr) ** 3

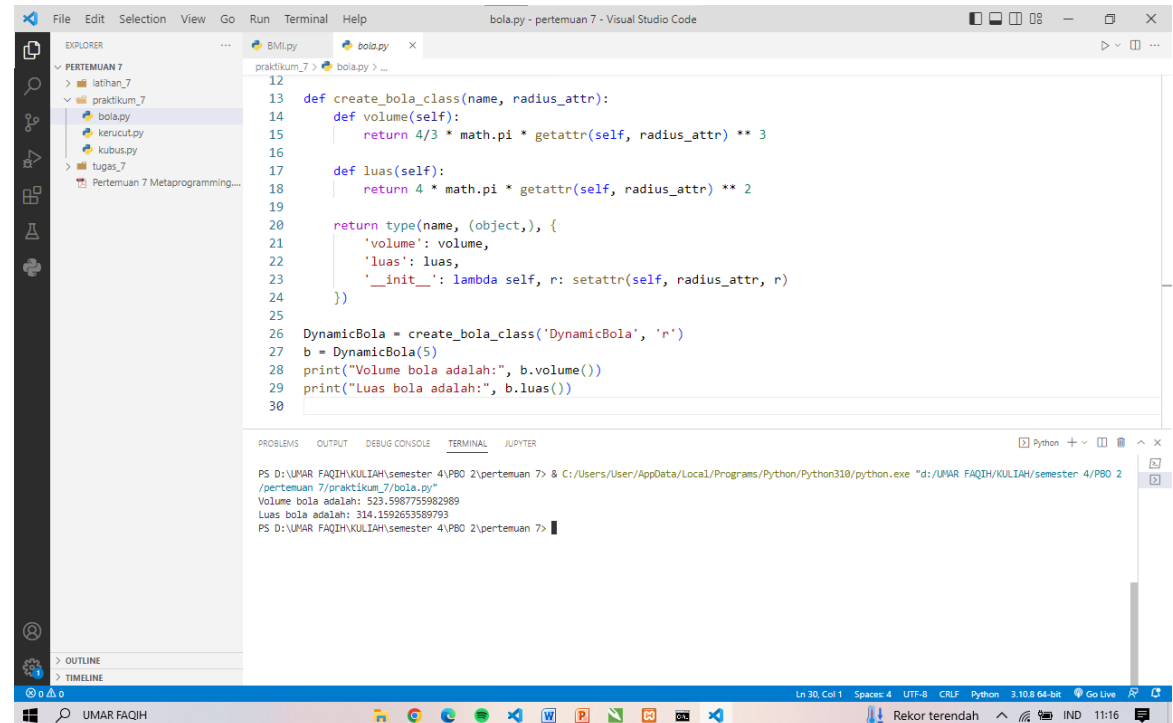
    def luas(self):
        return 4 * math.pi * getattr(self, radius_attr)
** 2

    return type(name, (object,), {
        'volume': volume,
        'luas': luas,
        '__init__': lambda self, r: setattr(self,
radius_attr, r)
    })

DynamicBola = create_bola_class('DynamicBola', 'r')
b = DynamicBola(5)
```

```
print("Volume bola adalah:", b.volume())
print("Luas bola adalah:", b.luas())
```

Hasil :



The screenshot shows a Visual Studio Code window with a file named 'bola.py' open. The code defines a function 'create\_bola\_class' that takes a name and a radius attribute, and returns a class with 'volume' and 'luas' methods. It then creates an instance 'DynamicBola' and prints its volume and area. The terminal output shows the execution results: 'Volume bola adalah: 523.5987755982989' and 'Luas bola adalah: 314.1592653589793'.

```
12
13 def create_bola_class(name, radius_attr):
14     def volume(self):
15         return 4/3 * math.pi * getattr(self, radius_attr) ** 3
16
17     def luas(self):
18         return 4 * math.pi * getattr(self, radius_attr) ** 2
19
20     return type(name, (object,), {
21         'volume': volume,
22         'luas': luas,
23         '__init__': lambda self, r: setattr(self, radius_attr, r)
24     })
25
26 DynamicBola = create_bola_class('DynamicBola', 'r')
27 b = DynamicBola(5)
28 print("Volume bola adalah:", b.volume())
29 print("Luas bola adalah:", b.luas())
30
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

PS D:\UMAR FAQIH\KULIAH\semester 4\PBO 2\pertemuan 7> & C:/Users/User/AppData/Local/Programs/Python/Python310/python.exe "d:/UMAR FAQIH/KULIAH/semester 4/PBO 2 /pertemuan 7/praktikum\_7/bola.py"

Volume bola adalah: 523.5987755982989  
Luas bola adalah: 314.1592653589793

PS D:\UMAR FAQIH\KULIAH\semester 4\PBO 2\pertemuan 7>

## 2. Kerucut

```
import math

class Kerucut:
    def __init__(self, r, t):
        self.r = r
        self.t = t

    def volume(self):
        return math.pi * self.r ** 2 * self.t / 3
```

```

    def luas(self):
        s = math.sqrt(self.r ** 2 + self.t ** 2)
        return math.pi * self.r * s + math.pi * self.r
** 2

def create_kerucut_class(name, radius_attr,
height_attr):
    def volume(self):
        return math.pi * getattr(self, radius_attr) **
2 * getattr(self, height_attr) / 3

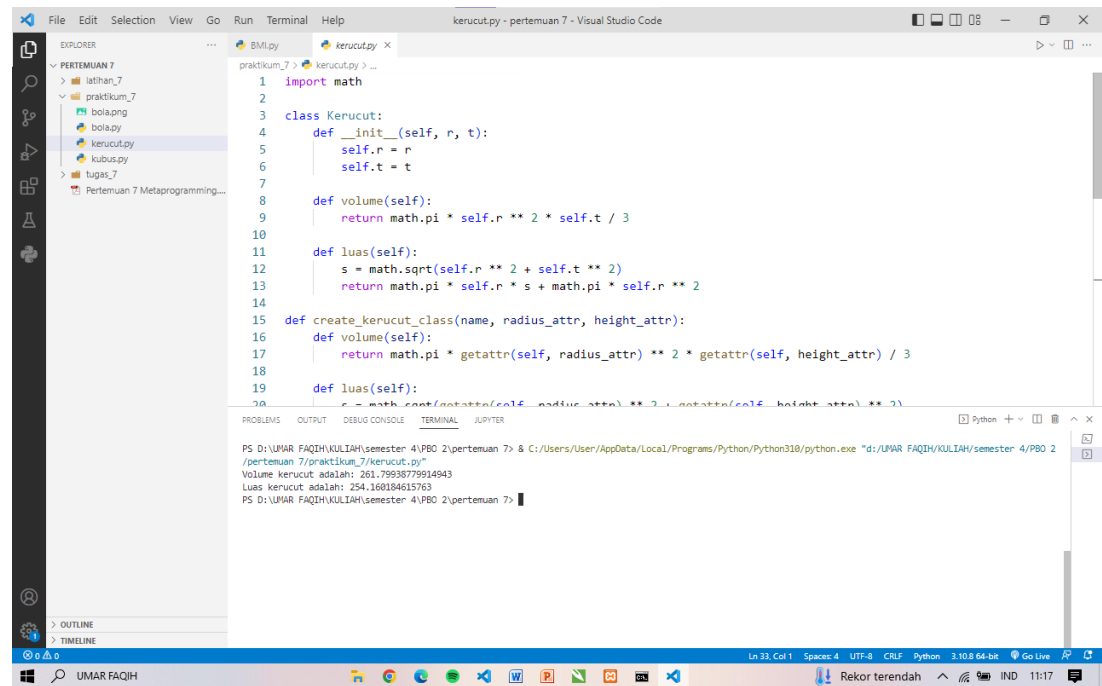
    def luas(self):
        s = math.sqrt(getattr(self, radius_attr) ** 2 +
getattr(self, height_attr) ** 2)
        return math.pi * getattr(self, radius_attr) * s
+ math.pi * getattr(self, radius_attr) ** 2

    return type(name, (object,), {
        'volume': volume,
        'luas': luas,
        '__init__': lambda self, r, t: setattr(self,
radius_attr, r) or setattr(self, height_attr, t)
    })

DynamicKerucut = create_kerucut_class('DynamicKerucut',
'r', 't')
k = DynamicKerucut(5, 10)
print("Volume kerucut adalah:", k.volume())
print("Luas kerucut adalah:", k.luas())

```

Hasil :



The screenshot shows a Visual Studio Code window with a file explorer on the left containing a folder named 'PERTEMUAN 7' with files like 'latihan\_7', 'praktikum\_7', 'bola.png', 'bola.py', 'kerucut.py', 'kubus.py', and 'lugas\_7'. The main editor displays the 'kerucut.py' file with the following Python code:

```
1 import math
2
3 class Kerucut:
4     def __init__(self, r, t):
5         self.r = r
6         self.t = t
7
8     def volume(self):
9         return math.pi * self.r ** 2 * self.t / 3
10
11     def luas(self):
12         s = math.sqrt(self.r ** 2 + self.t ** 2)
13         return math.pi * self.r * s + math.pi * self.r ** 2
14
15 def create_kerucut_class(name, radius_attr, height_attr):
16     def volume(self):
17         return math.pi * getattr(self, radius_attr) ** 2 * getattr(self, height_attr) / 3
18
19     def luas(self):
20         s = math.sqrt(getattr(self, radius_attr) ** 2 + getattr(self, height_attr) ** 2)
```

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

```
PS D:\UMAR FAQIH\KULIAH\semester 4\PBO 2\pertemuan 7> & C:/Users/User/AppData/Local/Programs/Python/Python310/python.exe "d:/UMAR FAQIH/KULIAH/semester 4/PBO 2/pertemuan 7/praktikum_7/kerucut.py"
Volume kerucut adalah: 261.79938779914943
Luas kerucut adalah: 254.168184615763
PS D:\UMAR FAQIH\KULIAH\semester 4\PBO 2\pertemuan 7>
```

### 3. Kubus

```
class Kubus:
    def __init__(self, sisi):
        self.sisi = sisi

    def volume(self):
        return self.sisi ** 3

    def luas(self):
        return 6 * self.sisi ** 2

def create_kubus_class(name, sisi_attr):
    def volume(self):
        return getattr(self, sisi_attr) ** 3

    def luas(self):
```

```

        return 6 * getattr(self, sisi_attr) ** 2

    return type(name, (object,), {
        'volume': volume,
        'luas': luas,
        '__init__': lambda self, sisi: setattr(self,
sisi_attr, sisi)
    })

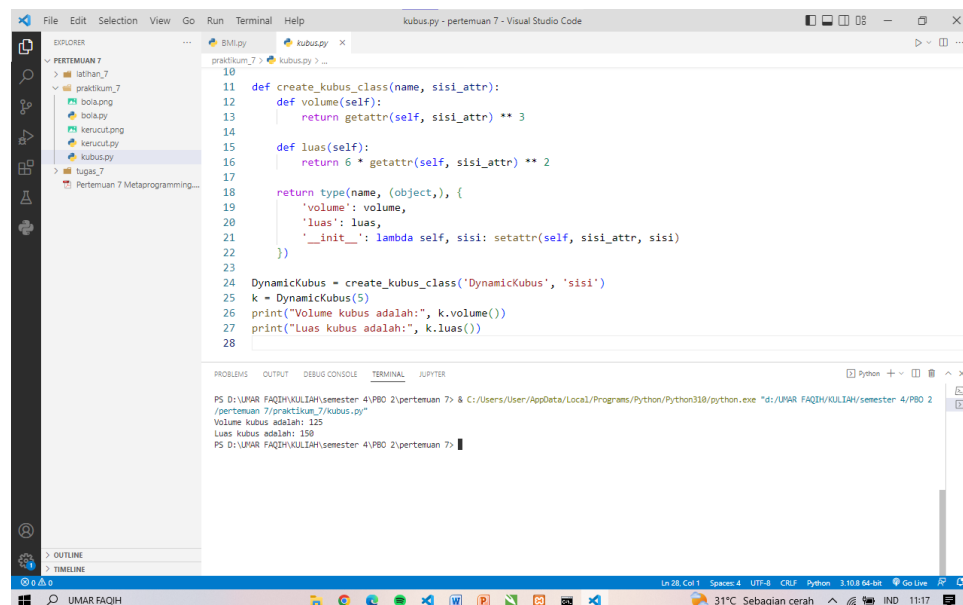
DynamicKubus = create_kubus_class('DynamicKubus',
'sisi')
k = DynamicKubus(5)
print("Volume kubus adalah:", k.volume())
print("Luas kubus adalah:", k.luas())

asyncio.run(main())

except StopAsyncIteration as e:
    print("Iterasi berhenti:", e)

```

Hasil :



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 'PERTEMUAN 7' with subfolders 'latihan\_7' and 'praktikum\_7'. The 'praktikum\_7' folder contains files 'latihan\_7', 'bobot.py', 'kerucut.py', 'kubus.py', and 'tugas\_7'. The 'kubus.py' file is selected and its content is displayed in the code editor. The code defines a function 'create\_kubus\_class' that takes a name and a side attribute, and returns a class 'DynamicKubus'. The class has methods 'volume' and 'luas' that calculate the volume and surface area of a cube. The class is instantiated as 'k' with a side length of 5. The code then prints the volume and surface area of the cube. The terminal at the bottom shows the output of the code: 'Volume kubus adalah: 125' and 'Luas kubus adalah: 150'.

```

10
11 def create_kubus_class(name, sisi_attr):
12     def volume(self):
13         return getattr(self, sisi_attr) ** 3
14
15     def luas(self):
16         return 6 * getattr(self, sisi_attr) ** 2
17
18     return type(name, (object,), {
19         'volume': volume,
20         'luas': luas,
21         '__init__': lambda self, sisi: setattr(self, sisi_attr, sisi)
22     })
23
24 DynamicKubus = create_kubus_class('DynamicKubus', 'sisi')
25 k = DynamicKubus(5)
26 print("Volume kubus adalah:", k.volume())
27 print("Luas kubus adalah:", k.luas())
28

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

Python + Python 3.10.8 64-bit

PS D:\UMAR FAQIH\KULIAH\semester 4\PRO 2\pertemuan 7> & C:\Users\User\AppData\Local\Programs\Python\Python310\python.exe "d:\UMAR FAQIH\KULIAH\semester 4\PRO 2\pertemuan 7\praktikum\_7\kubus.py"

Volume kubus adalah: 125

Luas kubus adalah: 150

PS D:\UMAR FAQIH\KULIAH\semester 4\PRO 2\pertemuan 7>

Link github < <https://github.com/Umar-Faqih/PBO-2/tree/main/Praktikum> >