

LAPORAN PRAKTIKUM

PEMROGRAMAN BERORIENTASI OBJEK LANJUT

2023



Prepared By:
Rizal Trilaksana
210511054
TI21K

1. Buatlah 3 contoh dynamic attributes dan 1 dynamic classes

praktikum7_dynamic_atribute1.py

```
#NAMA : RIZALTRILAKSANA
#NIM : 210511054
#KELAS : K1

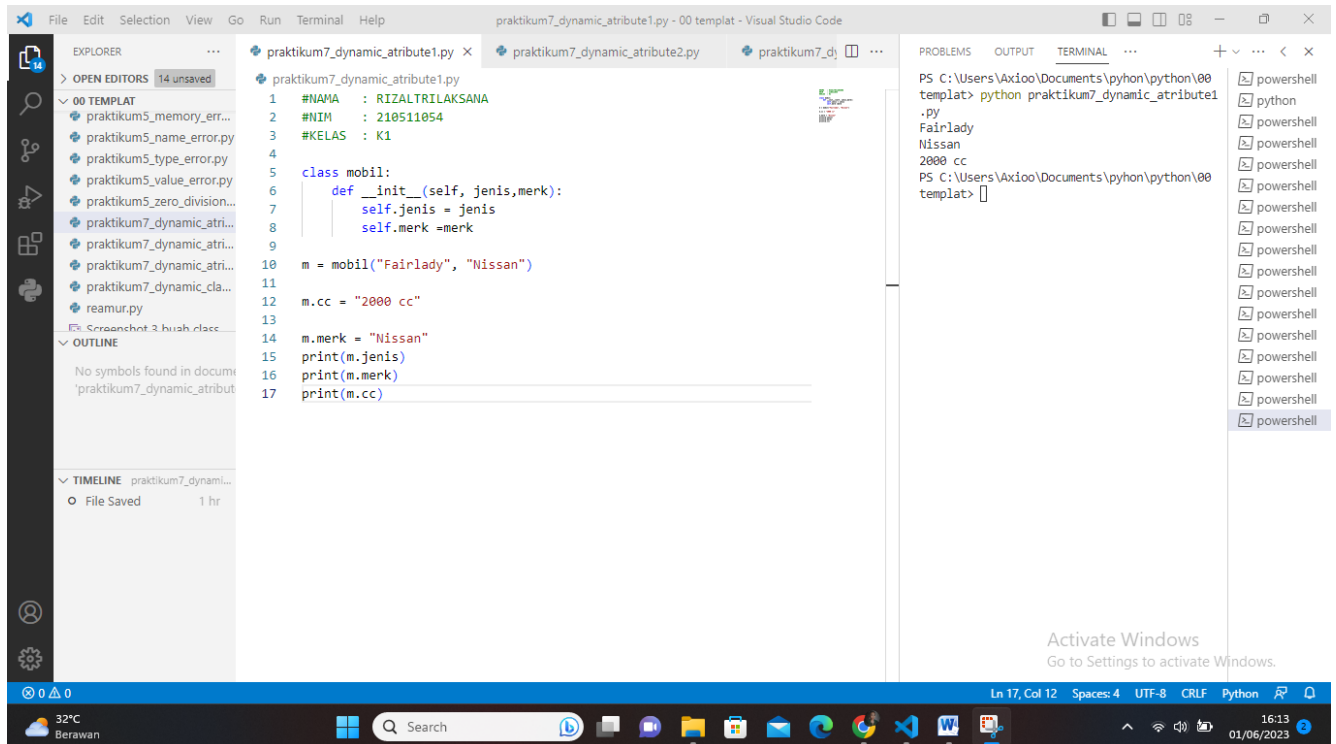
class mobil:
    def __init__(self, jenis,merk):
        self.jenis = jenis
        self.merk =merk

m = mobil("Fairlady", "Nissan")

m.cc = "2000 cc"

m.merk = "Nissan"
print(m.jenis)
print(m.merk)
print(m.cc)
```

Hasil output



Praktikum7_dynamic_atribute2.py

```
#NAMA : RIZALTRILAKSANA
```

```
#NIM : 210511054
```

```
#KELAS : K1
```

```
class menu:
```

```
    def __init__(self, nama,harga):
```

```
        self.nama = nama
```

```
        self.harga =harga
```

```
m = menu("gudeg", "10000")
```

```
m.khas = "Jogja"
```

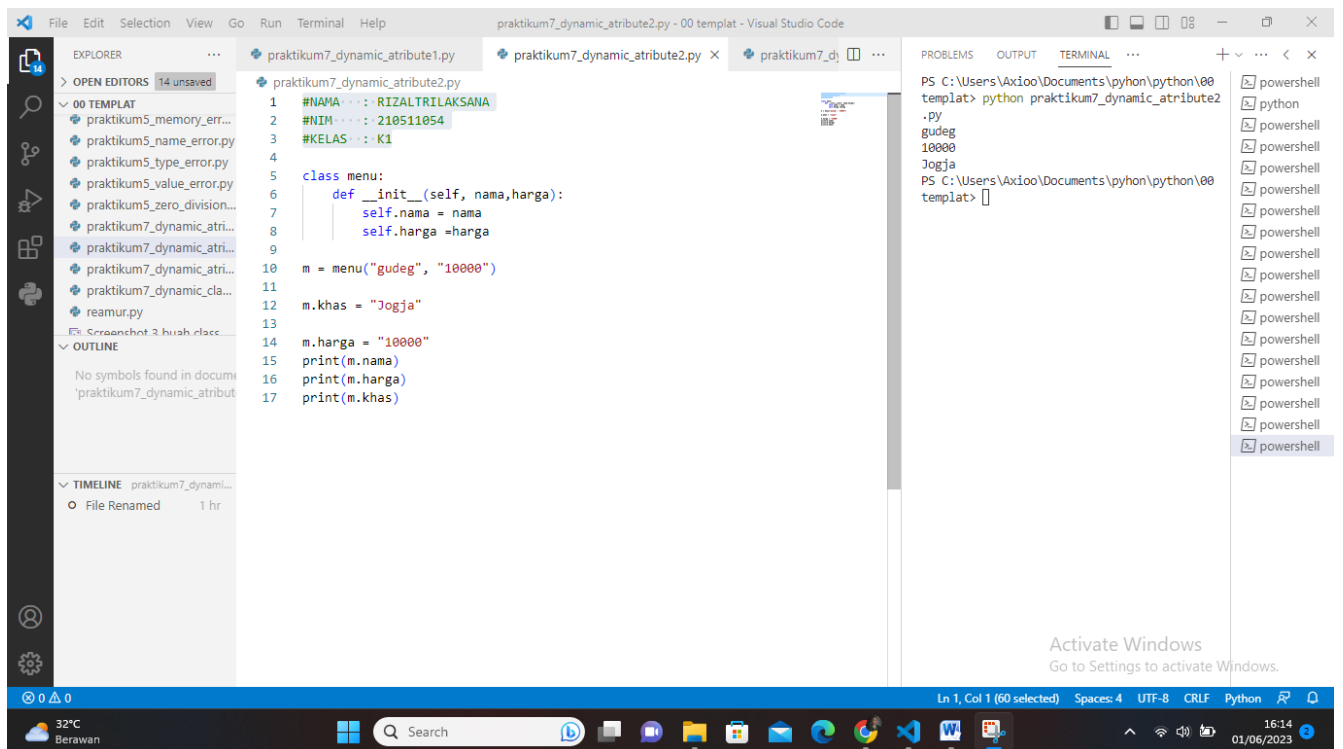
```
m.harga = "10000"
```

```
print(m.nama)
```

```
print(m.harga)
```

```
print(m.khas)
```

Hasil output



Praktikum7_dynamic_atribute3.py

```
#NAMA : RIZALTRILAKSANA
```

```
#NIM : 210511054
```

```
#KELAS : K1
```

```
class hewan:
```

```
    def __init__(self, nama,jenis):
```

```
        self.nama = nama
```

```
        self.jenis =jenis
```

```
h = hewan("kucing", "mamalia")
```

```
h.ras = "angora"
```

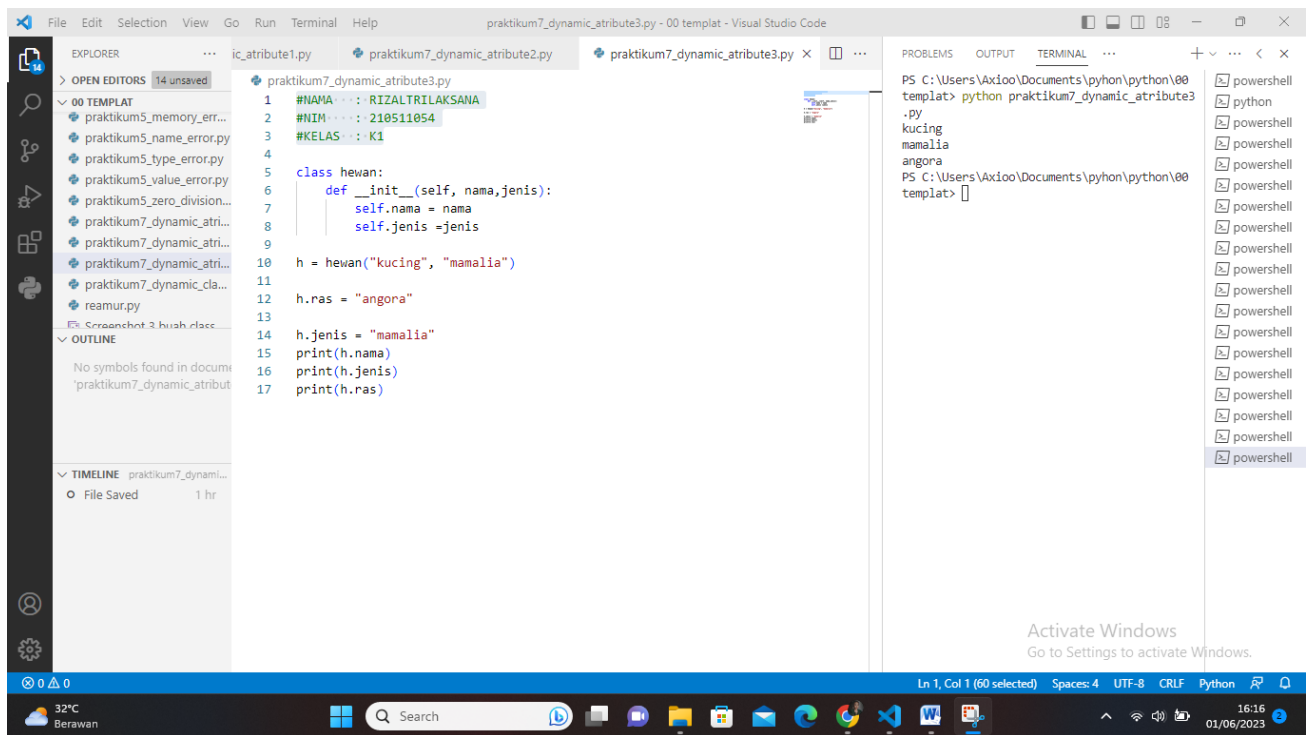
```
h.jenis = "mamalia"
```

```
print(h.nama)
```

```
print(h.jenis)
```

```
print(h.ras)
```

Hasil output



praktikum7_dynamic_classes1.py

#NAMA : RIZALTRILAKSANA

#NIM : 210511054

#KELAS : K1

```
def custom_kendaraan(tipe_kendaraan):
    class kendaraan:
        def __init__(self, brand, warna):
            self.tipe_kendaraan = tipe_kendaraan
            self.brand = brand
            self.warna = warna

        def __repr__(self):
            return f"{self.brand} {self.tipe_kendaraan} ({self.warna})"

    return kendaraan

Car = custom_kendaraan("Mobil")
Motorcycle = custom_kendaraan("Motor")

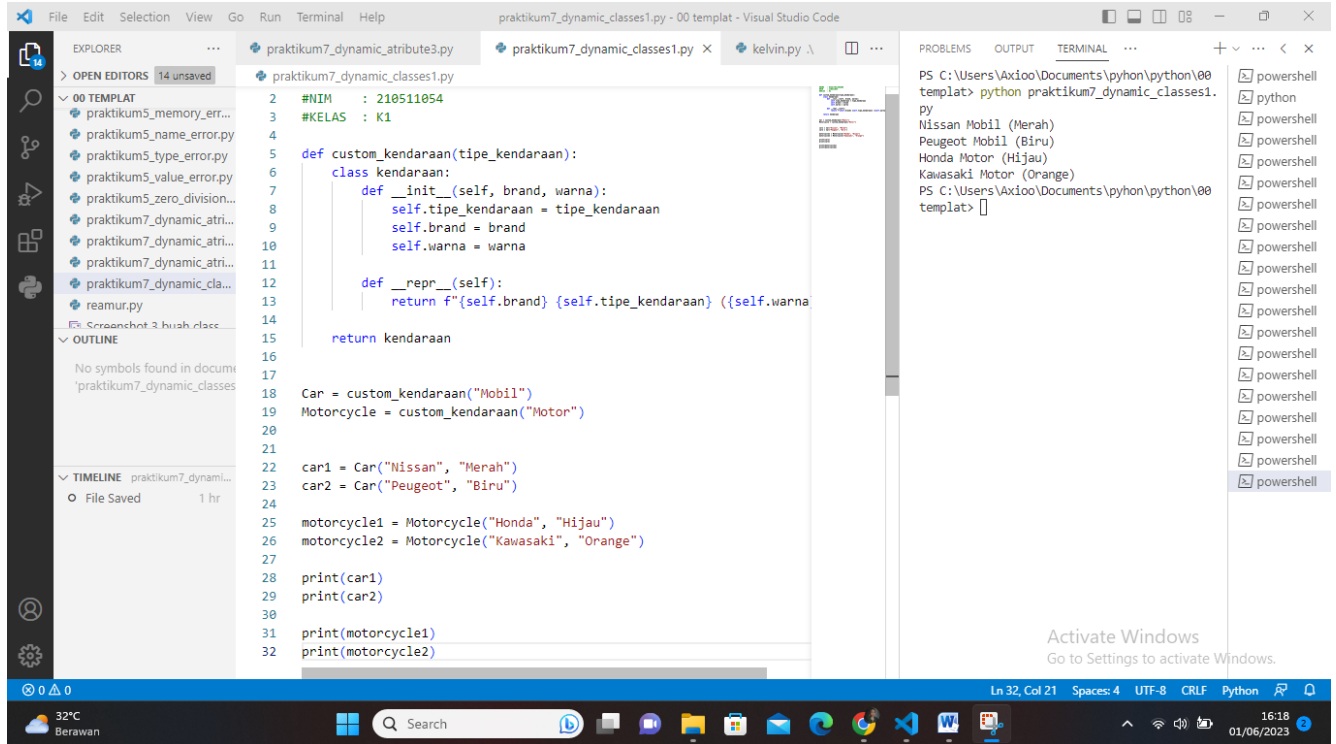
car1 = Car("Nissan", "Merah")
car2 = Car("Peugeot", "Biru")

motorcycle1 = Motorcycle("Honda", "Hijau")
motorcycle2 = Motorcycle("Kawasaki", "Orange")

print(car1)
print(car2)

print(motorcycle1)
print(motorcycle2)
```

Hasil output



The screenshot displays the Visual Studio Code interface with a Python file named `praktikum7_dynamic_classes1.py` open. The code defines a `custom_kendaraan` function that takes a `type_kendaraan` argument and returns a `kendaraan` class. The `kendaraan` class has an `__init__` method that sets `self.tipe_kendaraan`, `self.brand`, and `self.warna` based on the input. It also has a `__repr__` method that returns a string representation of the object. The code then creates two car objects (`car1` and `car2`) and two motorcycle objects (`motorcycle1` and `motorcycle2`), and prints them.

```
2 #NIM : 210511054
3 #KELAS : K1
4
5 def custom_kendaraan(tipe_kendaraan):
6     class kendaraan:
7         def __init__(self, brand, warna):
8             self.tipe_kendaraan = tipe_kendaraan
9             self.brand = brand
10            self.warna = warna
11
12            def __repr__(self):
13                return f'{self.brand} {self.tipe_kendaraan} ({self.warna})'
14
15            return kendaraan
16
17
18 Car = custom_kendaraan("Mobil")
19 Motorcycle = custom_kendaraan("Motor")
20
21
22 car1 = Car("Nissan", "Merah")
23 car2 = Car("Peugeot", "Biru")
24
25 motorcycle1 = Motorcycle("Honda", "Hijau")
26 motorcycle2 = Motorcycle("Kawasaki", "Orange")
27
28 print(car1)
29 print(car2)
30
31 print(motorcycle1)
32 print(motorcycle2)
```

The terminal output shows the execution of the script, displaying the string representations of the created objects:

```
PS C:\Users\Axioo\Documents\pyhon\python\00
templat> python praktikum7_dynamic_classes1.
py
Nissan Mobil (Merah)
Peugeot Mobil (Biru)
Honda Motor (Hijau)
Kawasaki Motor (Orange)
PS C:\Users\Axioo\Documents\pyhon\python\00
templat>
```

The status bar at the bottom indicates the file is at line 32, column 21, with 4 spaces, UTF-8 encoding, CRLF line endings, and the Python interpreter is active.