

# Pertemuan-7

## Metaprogramming

### A. Metaprogramming

Metaprogramming di Python adalah kemampuan untuk menulis kode Python yang dapat memanipulasi kode Python itu sendiri pada waktu eksekusi. Dalam hal ini, program dapat membuat, mengubah, atau memodifikasi kode Python yang sedang berjalan.

Beberapa topik yang terkait dengan Metaprogramming adalah sebagai berikut:

1. **Dynamic Attributes:** kemampuan untuk menambahkan atribut pada sebuah objek secara dinamis pada waktu eksekusi, baik pada instance maupun kelas.

**Contoh:**

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

person = Person("John", 30)

# Menambah atribut address secara dinamis
person.address = "123 Main St."

# Mengubah nilai atribut age secara dinamis
person.age = 35

print(person.name)
print(person.age)
print(person.address)
```

2. **Dynamic Classes:** kemampuan untuk membuat kelas secara dinamis pada waktu eksekusi, misalnya dengan menggunakan fungsi `type()` atau `class statement`.

Contoh 1:

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

        def luas(cls, panjang, lebar):
            return panjang * lebar
        cls.luas = classmethod(luas)

        def keliling(cls, panjang, lebar):
            return 2 * (panjang + lebar)
        cls.keliling = classmethod(keliling)
```

```
class PersegiPanjang(metaclass=PersegiPanjangMeta):  
    pass
```

```
A = PersegiPanjang()  
B = A.luas(4,5)  
C = A.keliling(4,5)  
print('Luas Persegi Panjang:',B)  
print('Keliling Persegi panjang:',C)
```

Contoh 2:

```
class SegitigaMeta(type):  
    def __init__(cls, name, bases, attrs):  
        super().__init__(name, bases, attrs)  
  
        # Tambahkan method untuk menghitung luas dan keliling segitiga  
        def luas(cls, alas, tinggi):  
            return (alas * tinggi) / 2  
        cls.luas = classmethod(luas)  
  
        def keliling(cls, sisi1, sisi2, sisi3):  
            return sisi1 + sisi2 + sisi3  
        cls.keliling = classmethod(keliling)  
  
class Segitiga(metaclass=SegitigaMeta):  
    pass
```

```
s = Segitiga()
```

```
# Menghitung luas segitiga dengan alas=4 dan tinggi=5  
luas_segitiga = Segitiga.luas(4, 5)  
print("Luas segitiga:", luas_segitiga)
```

```
# Menghitung keliling segitiga dengan sisi1=3, sisi2=4, dan sisi3=5  
keliling_segitiga = Segitiga.keliling(3, 4, 5)  
print("Keliling segitiga:", keliling_segitiga)
```

Contoh 3:

```
class CelciusMeta(type):  
    def __init__(cls, name, bases, attrs):  
        super().__init__(name, bases, attrs)  
        cls.suhu_standar = ""  
  
    def to_fahrenheit(cls, suhu):  
        return (suhu * 9/5) + 32
```

```

def to_reamur(cls, suhu):
    return suhu * 4/5

def to_kelvin(cls, suhu):
    return suhu + 273.15

class Celcius(metaclass=CelciusMeta):
    def __init__(self, suhu):
        self.suhu = suhu

    def ke_unit(self, unit):
        if unit == "Fahrenheit":
            self.suhu = self.__class__.to_fahrenheit(self.suhu)
            self.__class__.suhu_standar = "Fahrenheit"
        elif unit == "Reamur":
            self.suhu = self.__class__.to_reamur(self.suhu)
            self.__class__.suhu_standar = "Reamur"
        elif unit == "Kelvin":
            self.suhu = self.__class__.to_kelvin(self.suhu)
            self.__class__.suhu_standar = "Kelvin"
        elif unit == "Celcius":
            pass # do nothing
        else:
            raise ValueError(f"Unit {unit} tidak dikenal.")

    def __repr__(self):
        return f"{self.suhu:.2f} {self.__class__.suhu_standar}"

# Membuat objek suhu dengan nilai 100 Celcius
c = Celcius(100)

# Mengubah objek suhu menjadi Fahrenheit
c.ke_unit("Fahrenheit")

print(c)

```

Praktikum:

Buatlah 3 aplikasi untuk menghitung volume dan luas permukaan selain dari contoh diatas menggunakan teknik Metaprogramming. Hasilnya diupload ke github masing-masing di folder : **praktikum7**

Tugas:

Buatlah aplikasi untuk menentukan Berat Badan Ideal (BMI) menggunakan teknik Metaprogramming. Hasilnya diupload ke github masing-masing di folder : **tugas7**