

Pertemuan-1

Class dan Objek

A. Class

Class di dalam OOP (Object-Oriented Programming) Python adalah sebuah blueprint atau cetak biru yang mendefinisikan atribut dan metode yang dimiliki oleh sebuah objek. Class digunakan untuk membuat objek, yang merupakan instance dari class itu sendiri.

Dalam class, atribut adalah data atau informasi yang terkait dengan objek yang dibuat dari class tersebut, sedangkan metode adalah fungsi yang berhubungan dengan objek tersebut. Class digunakan untuk mengorganisasi kode program dalam bentuk yang lebih terstruktur dan mudah dipahami.

Jenis Class:

1. Class yang mengimplementasi pendekatan Objek Oriented Programming
2. Class yang mengimplementasi pendekatan prosedural

B. Objek

Objek di dalam OOP Python adalah sebuah instance atau representasi dari sebuah class. Secara sederhana, class dapat dianggap sebagai blueprint atau cetak biru dari suatu objek, sedangkan objek adalah hasil implementasi dari blueprint tersebut. Objek memiliki atribut dan metode yang ditentukan oleh class yang menjadi dasar pembuatannya. Atribut adalah variabel yang menyimpan nilai yang terkait dengan objek tersebut, sedangkan metode adalah fungsi atau perilaku yang dapat dilakukan oleh objek.

Dalam OOP Python, setiap objek memiliki state (keadaan) dan behavior (perilaku) yang terdefinisi dalam class. State objek didefinisikan oleh atribut-atribut yang dimilikinya, sementara behavior objek didefinisikan oleh metode-metode yang dapat dilakukan oleh objek tersebut.

Sebagai contoh, jika kita memiliki class "Mobil" maka objek dari class tersebut bisa berupa mobil-mobil yang berbeda dengan atribut-atribut seperti merek, tipe, warna, dan nomor polisi, serta metode-metode seperti mengendarai mobil, mengisi bahan bakar, dan membunyikan klakson.

Contoh 1:

```
class Mobil:
```

```
    def __init__(self, merk, warna):  
        self.merk = merk  
        self.warna = warna
```

```
    def info(self):  
        print(f"Mobil {self.merk} berwarna {self.warna}")
```

Constructor

Method

```
mobila = Mobil("Toyota", "Hitam")
mobila.info() # Output: Mobil Toyota berwarna Hitam
```

Contoh 2:

```
class Mahasiswa:
    def __init__(self, nama, npm):
        self.nama = nama
        self.npm = npm

    def info(self):
        print(f>Nama: {self.nama}\nNPM: {self.npm}")
```

```
mahasiswaB = Mahasiswa("Ahmad", "123456789")
mahasiswaB.info()
```

Contoh 3:

```
class Lingkaran:
    def __init__(self, jari_jari):
        self.jari_jari = jari_jari

    def luas(self):
        return 3.14 * (self.jari_jari ** 2)

lingkaranA = Lingkaran(7)
print(f>Luas lingkaran: {lingkaranA.luas()}")
```

Contoh 4:

```
class Buku:
    def __init__(self, judul, penulis):
        self.judul = judul
        self.penulis = penulis

    def info(self):
        print(f>Judul: {self.judul}\nPenulis: {self.penulis}")
```

```
bukuA = Buku("Harry Potter and the Philosopher's Stone", "J.K. Rowling")
bukuA.info()
```

Contoh 5:

```
class PesawatTerbang:
    def __init__(self, maskapai, tujuan):
        self.maskapai = maskapai
        self.tujuan = tujuan

    def info(self):
        print(f>Maskapai: {self.maskapai}\nTujuan: {self.tujuan}")
```

```
pesawataA = PesawatTerbang("Garuda Indonesia", "Jakarta - Bali")
pesawataA.info()
```

Dalam hal ini mobilA, mahasiswaB, lingkaranA, bukuA, dan pesawatA adalah **objek**

Class bertipe Prosedural dan bukan OOP:

Contoh 6:

```
class Kalkulator:
    @staticmethod
    def add(x, y):
        return x + y

    @staticmethod
    def subtract(x, y):
        return x - y

    @staticmethod
    def multiply(x, y):
        return x * y

    @staticmethod
    def divide(x, y):
        if y == 0:
            raise ValueError('Tidak dapat membagi dengan nol.')
        return x / y

# Memanggil metode statis add() dan subtract() di dalam class Math
print(Kalkulator.add(3, 5))          # Output: 8
print(Kalkulator.subtract(10, 7))    # Output: 3

# Memanggil metode statis multiply() dan divide() di dalam class Math
print(Kalkulator.multiply(4, 6))     # Output: 24
print(Kalkulator.divide(12, 4))      # Output: 3.0
```

Contoh 7:

```
class Celcius:
    @staticmethod
    def to_fahrenheit(celsius):
        return (celsius * 9/5) + 32

    @staticmethod
    def to_kelvin(celsius):
        return celsius + 273.15
```

```

    @staticmethod
    def to_reamur(celsius):
        return celsius * 4/5

mycelcius = 80
myfahrenheit = Celcius.to_fahrenheit(mycelcius)
print(myfahrenheit)

```

Dengan menambahkan decorator **@staticmethod**, metode-metode di dalam class tersebut akan menjadi metode statis yang dapat dipanggil tanpa membuat instance dari class tersebut.

Latihan :

Perhatikan 3 rumus berikut:

Konversi Suhu Celcius Ke Fahrenheit, Reamur, dan Kelvin:

```

C = 75
F = (9/5) * C + 32
print("konversi ",C, "derajat celcius adalah ",F, "derajat fahrenheit")

```

Konversi Suhu Celcius Ke Reamur:

```

C = 60
R = 4/5 * C
print("konversi ",C, "derajat celcius adalah ",R, "derajat Reamur")

```

Konversi Suhu Celcius Ke Kelvin:

```

C = 90
K = C + 273
print("konversi ",C, "derajat celcius adalah ",K, "derajat Kelvin")

```

Soal Praktikum:

1. Buatlah Class yang mengimplementasikan Prosedural, beri nama: celcius_pro.py
2. Buatlah Class yang mengimplementasikan Object Oriented Programming, beri nama: celcius_oop.py

3. Tulis nama, nim, dan kelas Anda di dalam Script

Jawaban berupa 2 buah screenshot script dan output tampilan hasilnya dikirim ke email :

(freddy.wicaksono@umc.ac.id) dengan subject: **Praktikum-1 PBO2 2023** pada body email tuliskan NIM, Nama Lengkap, dan Kelas

(Kesalahan judul Subject bisa berakibat tidak dinilai)

Evaluasi:

1. Apa yang dimaksud dengan class dalam Python?
 - a. Sebuah fungsi yang mengembalikan nilai tertentu
 - b. Sebuah variabel yang dapat digunakan oleh seluruh program
 - c. Sebuah blueprint untuk membuat objek
 - d. Sebuah fungsi yang digunakan untuk mencari nilai tertentu

2. Apa yang dimaksud dengan objek dalam Python?
 - a. Sebuah fungsi yang mengembalikan nilai tertentu
 - b. Sebuah variabel yang dapat digunakan oleh seluruh program
 - c. Sebuah blueprint untuk membuat class
 - d. Sebuah instance dari sebuah class

3. Apa yang dimaksud dengan constructor dalam Python?
 - a. Sebuah method yang digunakan untuk mengubah nilai atribut objek
 - b. Sebuah method yang digunakan untuk menghapus objek
 - c. Sebuah method yang digunakan untuk membuat objek
 - d. Sebuah method yang digunakan untuk mengambil nilai atribut objek

4. Apa yang dimaksud dengan self dalam Python?
 - a. Sebuah keyword yang digunakan untuk merujuk pada objek saat ini
 - b. Sebuah keyword yang digunakan untuk merujuk pada class saat ini
 - c. Sebuah keyword yang digunakan untuk merujuk pada konstruktor saat ini
 - d. Sebuah keyword yang digunakan untuk merujuk pada method saat ini

5. Bagaimana cara membuat objek dari sebuah class?
 - A. Dengan menggunakan fungsi init
 - B. Dengan menggunakan keyword "new"
 - C. Dengan menggunakan fungsi "create"
 - D. Dengan menggunakan keyword "instance"

6. Apa yang dimaksud dengan atribut dalam Python?

- A. Variabel yang terkait dengan sebuah objek
- B. Metode yang terkait dengan sebuah objek
- C. Fungsi yang digunakan untuk mengembalikan nilai
- D. Sebuah tipe data yang menyimpan nilai

7. Apa yang dimaksud dengan metode dalam Python?

- A. Fungsi yang terkait dengan sebuah objek
- B. Variabel yang terkait dengan sebuah objek
- C. Fungsi yang digunakan untuk mengembalikan nilai
- D. Sebuah tipe data yang menyimpan nilai

8. Bagaimana cara mengakses atribut dari sebuah objek?

- A. Dengan menggunakan tanda titik (.)
- B. Dengan menggunakan tanda kurung []
- C. Dengan menggunakan tanda koma ,
- D. Dengan menggunakan tanda panah ->

9. Bagaimana cara memanggil metode dari sebuah objek?

- A. Dengan menggunakan tanda titik (.)
- B. Dengan menggunakan tanda kurung ()
- C. Dengan menggunakan tanda koma ,
- D. Dengan menggunakan tanda panah ->

Tugas Minggu 1:

Buatlah 3 buah class (Fahrenheit, Reamur, dan Kelvin) yang mengimplementasikan OOP dimana setiap class memiliki kemampuan untuk melakukan konversi ke Temperatur yang lain.

Jawaban berupa 3 buah screenshot script beserta hasilnya dikirim ke email

(freddy.wicaksono@umc.ac.id) dengan subject: **Tugas-1 PBO2 2023**