

Kegiatan 8

Menerapkan Konsep *Abstract Class*

1. Capaian Pembelajaran

Mahasiswa mampu menjelaskan konsep interface dan Class Abstract dalam PBO

2. Pokok Bahasan

A. Pendahuluan

Abstract Class (kelas abstrak) merupakan class yang masih bersifat abstrak. Class abstrak tidak bisa dibuat objek secara langsung. Agar disebut *class* abstrak, setidaknya sebuah *class* memiliki satu atau lebih method abstrak. Method abstrak adalah method yang tidak memiliki bentuk konkrit atau tidak memiliki implementasi hanya nama saja. Abstraction merupakan konsep PBO memiliki tujuan untuk menyembunyikan detail yang tidak terlalu penting dari user.

Lalu mengapa harus menggunakan class abstrak jika tidak bisa dibuat objeknya? Pada suatu kondisi tertentu, class induk tidak ingin kita buat objeknya karena method yang kita buat belum jelas ingin diimplementasikan seperti apa. Atau biasanya dikarenakan ada parameter tertentu yang masih belum jelas diisi apa. Maka sebaiknya class ini dijadikan abstrak.

B. Menerapkan Abstract Class dengan Python

Dikarenakan Python tidak support abstraksi *class*, untuk bisa menerapkan abstract *class* di Python kita menggunakan modul Abstract Base Classes (ABC). Import terlebih dahulu modul ABC ke dalam program python kita dengan perintah

```
from abc import *
```

kemudian untuk membuat method menjadi abstrak maka kita tambahkan dekorator `@abstractmethod` diatas definisi method pada class tersebut. Perhatikan contoh berikut ini.

```
from abc import ABC, abstractmethod
class Bentuk(ABC):
    @abstractmethod
    def luas(self):
        return self.__sisi * self.__sisi

    @abstractmethod
    def keliling(self):
        return 4 * self.__sisi

class Persegi(Bentuk):
    def __init__(self, sisi):
        self.__sisi = sisi

persegi = Persegi(6)
```

Ketika sebuah objek persegi dibuat, class persegi menggunakan method yang ada di class induk yaitu class Bentuk. Dikarenakan class Bentuk semua methodnya adalah abstrak, maka akan muncul error seperti berikut, bahwa sebuah objek tidak dapat dibuat.

```
persegi = Persegi(6)
TypeError: Can't instantiate abstract class Persegi with abstract methods keliling,
luas
```

Bagaimana cara menggunakan class abstract tersebut? Caranya adalah dengan menimplementasikan method tersebut di class Persegi. Perhatikan contoh berikut.

```
from abc import ABC, abstractmethod
class Bentuk(ABC):
    @abstractmethod
    def luas(self):
        return self.__sisi * self.__sisi

    @abstractmethod
    def keliling(self):
        return 4 * self.__sisi

class Persegi(Bentuk):
    def __init__(self, sisi):
        self.__sisi = sisi
    def luas(self):
        return self.__sisi * self.__sisi
    def keliling(self):
        return 4 * self.__sisi

persegi = Persegi(7)
print(persegi.luas())
print(persegi.keliling())
```

Pada class Persegi kita sudah tambahkan method luas() dan keliling(). Ketika dijalankan akan menghasilkan keluaran seperti berikut.

```
49
28
```

3. Tugas

1. Buatlah contoh lain penerapan Abstract Class kemudian jelaskan bagaimana program anda bekerja.