TUGAS MODUL 6

LAPORAN PRATIKUM PEMROGRAMAN BERORIENTASI OBJEK

KELAS ABSTRAK , INTERFACE

ITERA

Disusun oleh :

Nama : Winnerson Laia

NIM : 121140121

PROGRAM STUDI TEKNIK INFORMATIKA

INSTITUT TEKNOLOGI SUMATERA

LAMPUNG SELATAN

2023

A. KELAS ABSTRAK

Pada suatu waktu, kita ingin membuat beberapa class yang hampir mirip dan memiliki fungsi/method yang sama kegunaanya namun memiliki definisi yang berbeda disetiap kelasnya. Jika membuat class satu persatu dan memdefinsikan fungsi/methodnya satu persatu maka pada saat membuat kode program maka bisa saja kita lupa membuat definisi fungsi/method tadi. Solusinya kita dapat menggunakan kelas yang bersifat abstrak (kelas abstrak).

Kelas abstrak bersifat seperti suatu kontrak artinya semua kelas turunan dari kelas abstrak harus mendefinisikan fungsi/method abstrak di kelas turunan. Kelas abstrak sendiri adalah kelas yang fungsi/methodnya belum memiliki implementasi, namun implemtasinya baru dilakukan pada kelas turunan dari kelas abstrak. Kelas abstrak juga dapat memiliki konstruktor namun tidak dapat dibuat objeknya.

Dalam bahasa pemrograman python, kita dapat membuat kelas dengan menggunakan modul abc (abstract base class). Kelas abstrak dibuat dengan menjadikan kelas abstrack tersebut sebagai kelas turunan ABC. Dalam kelas abstrak terdapat yang disebut abstrak method. Abstrack method adalah fungsi/method yang dimana kita ingin agar fungsi/mehtod tersebut harus didefinisikan di kelas turunan. Kita dapat menambahkan decorator @abstrackmethod diatas fungsi/method yang kita maksud. Kelas yang merupakan kelas turunan dari kelas abstrack disebut sebagai kelas konkret. Kelas konkret digunakan sebagai kelas dimana abstract method diimplementasikan.

Berikut contoh kodenya :

```python
from abc import ABC
from abc import abstractmethod

class BangunDatar:
    def __init__(self, nama_bangun) -> None:
        self.nama_bangun = nama_bangun

    @abstractmethod
    def luas(self):
        pass

    @abstractmethod
    def keliling(self):
        pass

class Persegi(BangunDatar) :
    def __init__(self, panjang, lebar) -> None:
        super().__init__("Persegi")
        self.panjang = panjang
        self.lebar = lebar

    def luas(self):
        return self.panjang * self.lebar

    def keliling(self):
        return 2 * (self.panjang + self.lebar)

    def __str__(self) -> str:
        return f"\nPersegi \nLuas = {self.luas()} \nKeliling = {self.keliling()} \n"

class Lingkaran(BangunDatar) :
    def __init__(self, jari_jari) -> None:
        super().__init__("Lingkara")
        self.jari_jari = jari_jari

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

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

    def __str__(self) -> str:
        return f"\nLingkaran \nLuas = {self.luas()} \nKeliling = {self.keliling()} \n"


# Main Program

persegi = Persegi(4, 3)
lingkaran  = Lingkaran(7)

print(persegi)
print(lingkaran)
```

```
thewinner@thewinner-Aspi
1-01-Prak PBO/Pertemuan

Persegi
Luas = 12
Keliling = 14


Lingkaran
Luas = 153.86
Keliling = 43.96
```

B. INTERFACE

Interface sendiri berarti antar muka. Interface merupakan kelas abstrak yang berisi kumpulan fungsi/method yang nantinya diimplentasikan di kelas turunan. Interface bersifat bersifat abstrak yang berati semua fungsi yang terdapat di interface harus diimplentasikan di kelas turunan. Kita dapat menggunakan interface ketika ingin agar implemtasi dari fungsi/method dari kelas abstrack berbeda-beda sehingga menjadi fleksibel. Contohnya seperti pembuatan suatu fitur, kita ingin agar fitur tersebut dapat digunakan diberbagai perangkat, maka kita tinggal menjadi fitur sebagai interface.

Interface dan kelas abstrak memang memiliki kemiripan, namun perbedaannya terdapat konstruktor dan abstrack methodnya. Pada kelas abstrak, kita dapat memberikan konstuktor dan tidak semua fungsi/methodnya dijadikan abstrack method. Sedangkan pada interface tidak memiliki konstruktor dan semua fungsi/methodnya dijadikan abstrack method.

Berikut contoh kodenya :

```python
from abc import ABC
from abc import abstractclassmethod

class VoiceAssitance(ABC) :
    @abstractclassmethod
    def aktifkan_asisten(self):
        pass

    @abstractclassmethod
    def matikan_asisten(self):
        pass

    @abstractclassmethod
    def greetings(self):
        pass


class Smartphone(VoiceAssitance) :
    def __init__(self, nama_perangkat:str) -> None:
        self.nama_perangkat = nama_perangkat
        self.__perangkat_hidup = False
        self.__voice_asisten_hidup = False
        self.__nama_user = "user"
        self.__nama_asisten_virtual = "Jeni"

    def hidupkan_smartphone(self):
        self.__perangkat_hidup  = True

    def matikan_smarthphone(self) :
        self.__perangkat_hidup  = False
        self.__voice_asisten_hidup = False

    def ubah_nama_user(self, nama_user_baru:str) :
        self.__nama_user = nama_user_baru

    def greetings(self):
        print(f"\n{self.__nama_asisten_virtual} : ")
        print(f"Halo {self.__nama_user}")
        print(f"Perkenalkan, nama saya {self.__nama_asisten_virtual}")
        print(f"Saya adalah asisten virtual pribadi anda")
        print(f"Senang bertemu dengan anda..... :) \n")

    def __saying_good_by(self):
        print(f"\n{self.__nama_asisten_virtual} : ")
        print(f"Selamat tinggal {self.__nama_user}")
        print(f"Panggil saya kembali kalau butuh sesuatu")
        print(f"Bye..... :)\n")

    def aktifkan_asisten(self):
        if self.__perangkat_hidup == True :
            self.__voice_asisten_hidup = True
            self.greetings()

    def matikan_asisten(self):
        if self.__voice_asisten_hidup == True :
            self.__saying_good_by()

        self.__voice_asisten_hidup = False


realme9pro = Smartphone("realme9pro")

realme9pro.hidupkan_smartphone()

realme9pro.ubah_nama_user("Winner")

realme9pro.aktifkan_asisten()

realme9pro.matikan_asisten()

realme9pro.matikan_smarthphone()
```

```
thewinner@thewinner-Aspire-A514-54:~/00-Ku
temuan 6/Tugas/01-Interface.py"

Jeni :
Halo Winner
Perkenalkan, nama saya Jeni
Saya adalah asisten virtual pribadi anda
Senang bertemu dengan anda..... :)


Jeni :
Selamat tinggal Winner
Panggil saya kembali kalau butuh sesuatu
Bye..... :)
```

Daftar Pustaka

https://www.geeksforgeeks.org/abstract-classes-in-python/

https://realpython.com/python-interface/