

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERBASIS OBJEK RB
MODUL 1**

Oleh :

Indah Mutiara

(121140158)



Program Studi Teknik Informatika

Institut Teknologi Sumatera

2023

Daftar Isi

Daftar Isi	2
Bab 1 (Pembahasan)	3
1. Kelas Abstrak	3
2. Interface	4
3. Metaclass	5
Kesimpulan	6
Daftar Pustaka	6

Pembahasan

1. Kelas Abstrak

Pada Python, terdapat kelas abstrak atau *abstract class* yang merupakan kelas tanpa instansiasi secara langsung dan biasanya digunakan sebagai kerangka untuk membuat kerangka kelas abstrak lainnya yang disebut dengan kelas turunan atau *child class*. Dalam kelas abstrak sendiri, kita dapat menemukan metode abstrak yang nanti akan diimplementasikan oleh kelas turunannya. Melalui metode-metode abstrak ini, kelas turunan tersebut dapat melakukan fungsi dan tugasnya masing-masing.

Sebelum membuat kelas abstrak dalam Python, kita harus mengimport modul `abc` (*abstract base class*) dan menggunakan fungsi `ABC` sebagai decorator guna untuk mendefinisikan kelas yang akan dibuat menjadi kelas abstrak pada suatu program. Setelah itu, kita dapat mendefinisikan metode abstrak yang terdapat dalam kelas abstrak dengan menggunakan dekorator `@abstractmethod`. Penting untuk dicatat bahwa setiap kelas turunan / *child class* akan menggunakan metode yang terdapat pada kelas abstrak.

- Contoh Implementasi kode kelas abstrak

Berikut ini merupakan contoh program python yang menggunakan kelas abstrak :

```
from abc import ABC, abstractmethod
```

```
class Hewan(ABC):
```

```
    @abstractmethod
```

```
    def move(self):
```

```
        pass
```

```
class Burung(Hewan):
```

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

```
        self.nama = "burung"
```

```
    def move(self):
```

```
        print("Terbang")
```

```
class Reptil(Hewan):
```

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

```
        self.nama = "reptil"
```

```
    def move(self):
```

```
        print("Merangkak")
```

```
class Ikan(Hewan):
```

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

```
        self.nama = "Ikan"
```

```
    def move(self):
```

```
print("Berenang")
```

Jika mengeksekusi program diatas, maka kita akan menerima keluaran sebagai berikut ini.

```
Berenang
None

...Program finished with exit code 0
Press ENTER to exit console. □
```

2. Interface

Konsep interface pada bahasa pemrograman Python tidak dinyatakan secara eksplisit seperti pada bahasa pemrograman lain seperti C#, PHP, Go, atau Java. Untuk mengatasi masalah ini, interface dapat diimplementasikan dalam program dengan menggunakan ABC (*Abstract Base Class*) melalui modul abc. Untuk konsep interface sendiri, konsep ini membahas tentang bagaimana aturan atau fungsi dari bagian-bagian program dapat berinteraksi dengan satu sama lain. Namun, interface dapat dibagi menjadi dua bagian, yang pertama informal interface dan kedua adalah formal interface.

Informal interface pada python merupakan interface yang berupa kelas-kelas. Dengan kata lain, informal interface mendefinisikan metode yang dapat diganti tanpa penegakan secara paksa. Selain itu, informal interface juga sering disebut sebagai protokol dimana sifatnya informal. Untuk contohnya sendiri dapat dilihat dengan kode program yang sudah dicontohkan pada pembahasan bagian kelas abstrak dimana kelas-kelas turunan menggunakan metode (berupa abstract method dari kelas orangtua).

Di sisi lain terdapat formal interface pada python yang berbicara tentang interface yang dapat diberlakukan sesuai dengan namanya, formal. Untuk membuat interface formal diperlukan penggunaan ABC (*Abstract Base Class*) dimana library ini akan membantu kita dalam mendefinisikan kelas-kelas yang bersifat abstrak bersama dengan metode-metodenya.

3. Metaclass

Metaclass dalam Python adalah sebuah kelas yang digunakan untuk membuat kelas baru. Dalam Python, kelas sendiri adalah objek dan metaclass adalah kelas yang digunakan untuk membuat objek kelas tersebut.

Ketika kita mendefinisikan sebuah kelas, Python akan menggunakan metaclass bawaan yang disebut type untuk membuat objek kelas tersebut. Namun, kita dapat menentukan metaclass yang berbeda untuk membuat kelas dengan perilaku yang berbeda pula.

Untuk membuat sebuah metaclass, kita perlu mendefinisikan sebuah kelas dengan mengimplementasikan metode khusus `__new__` atau `__init_subclass__`. Metode `__new__`

digunakan untuk membuat objek kelas baru, sedangkan metode `__init_subclass__` digunakan untuk menginisialisasi kelas turunan baru.

- Contoh Implementasi kode kelas abstrak
Berikut ini merupakan contoh program python yang menggunakan konsep interface dalam pemrograman :

```
from abc import ABC, abstractmethod

class Mobil(ABC):
    @abstractmethod
    def harga_disewakan(self):
        pass

class Toyota(Mobil):
    def harga_disewakan(self):
        print("Mobil Toyota disewakan dengan rate 5jt per bulan")

class Honda(Mobil):
    def harga_disewakan(self):
        print("Mobil Honda disewakan dengan rate 8jt per bulan")

class Hyundai(Mobil):
    def harga_disewakan(self):
        print("Mobil Hyundai disewakan dengan rate 5.5jt per bulan")
```

Pada contoh program di atas ini, interface yang dapat dilihat pada program ini berupa metode-metode yang diturunkan dari metode abstract kelas parent (Mobil) kepada kelas anak (Toyota, Honda, dan Hyundai).

Kesimpulan

1. Pada praktikum ini dapat disimpulkan bahasa interface merupakan konsep yang dapat dijadikan sebagai spesifikasi/dasar dari setiap fungsi yang akan dibuat dan digunakan pada suatu kelas dalam program. Hal ini tentu dapat dilakukan tanpa memberikan implementasi pada fungsi-fungsi tersebut. Selain itu, interface juga diperlukan saat kita perlu untuk memastikan kelas-kelas yang memiliki fungsi tertentu.
2. Untuk kelas abstrak, kita dapat menggunakan jenis kelas ini untuk membuat suatu basis atau kerangka dari kelas-kelas turunannya yang akan sering digunakan pada suatu program. Hal ini tentu akan meningkatkan efisiensi dan optimasi kode program.
3. Pada metaclass, kita belajar bahwa konsep ini akan berguna sesaat kita membutuhkan untuk memodifikasi perilaku antar kelas dalam suatu program. Dengan metaclass, kita dapat menentukan bagaimana kelas baru dapat berinteraksi dengan kelas lain dan juga perilakunya terhadap dirinya dan kelas-kelas lain.

Daftar Pustaka

- Korpa, J. (2021, October 16). *How to Use Abstract Classes in Python*. Towards Data Science. Retrieved April 12, 2023, from <https://towardsdatascience.com/how-to-use-abstract-classes-in-python-d4d2ddc02e90?gi=d11945472cdb>
- Murphy, W. (n.d.). *Implementing an Interface in Python – Real Python*. Real Python. Retrieved April 12, 2023, from <https://realpython.com/python-interface/>
- Rafael. (n.d.). *What is an Abstract Class in Object Oriented Programming*. Computing Learner. Retrieved April 12, 2023, from <https://computinglearner.com/abstract-class-in-object-oriented-programming/>
- W, J. (2022, July 12). *What are Python Metaclasses?* Sentry. Retrieved April 12, 2023, from <https://sentry.io/answers/what-are-Python-metaclasses/>