# Laporan Praktikum PBO



**Disusun oleh:**

**Ahmad Zain Mahmud | 121140232**

**Program Studi Teknik Informatika**

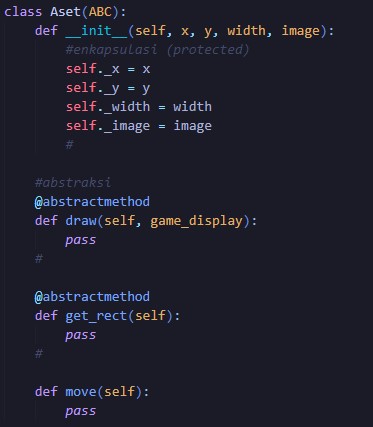
**Institut Teknologi Sumatera**

**Lampung Selatan 2023**

# Ringkasan

## 1. Kelas abstrak

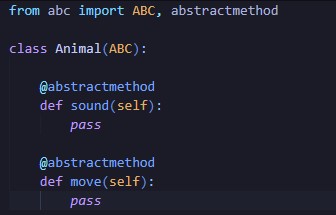
Kelas abstrak dapat dianggap sebagai kumpulan metode dari kelas lain. Kelas mendefinisikan sekumpulan metode yang harus diimplementasikan di setiap subkelas yang berasal dari kelas abstrak. Kelas abstrak dapat berisi satu atau lebih metode abstrak. Metode yang memiliki deklarasi tetapi tidak ada implementasinya dikatakan abstrak. Seperti pada gambar berikut ini.

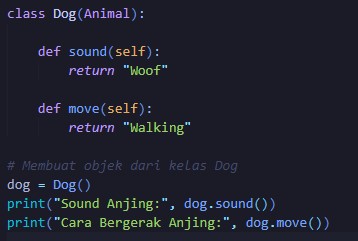


Berdasarkan kode ini, kelas turunan dari kelas abstrak harus mengimplementasikan semua metode abstrak yang dideklarasikan di kelas abstrak. Tujuan dari kelas abstrak adalah untuk menyediakan konsep dasar yang harus diikuti oleh kelas turunan.

## 2. Interface

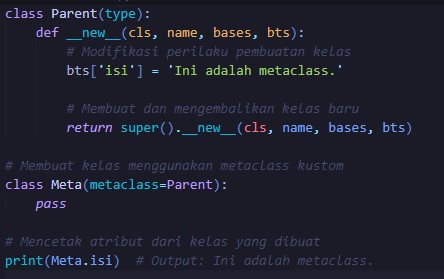
Interface adalah metode dan properti yang ada di kelas. Interface pengguna dapat dikatakan sebagai perilaku yang mengimplementasikan metode dalam objek.





## 3. Metaclass

Metaclass dari suatu kelas mendefinisikan bagaimana kelas berperilaku. Kelas itu sendiri adalah turunan dari metaclass. Kelas dalam Python menentukan bagaimana instance kelas berperilaku.



# Kesimpulan

Interface adalah konvensi yang mendefinisikan metode yang harus dimiliki kelas atau objek. Interface pengguna tidak menyediakan implementasi metode yang konkret, tetapi hanya mendefinisikan metode berdasarkan nama, parameter, dan tipe pengembaliannya. UI digunakan ketika Anda ingin menggunakan konsep polimorfisme. Ini memungkinkan kita untuk secara fleksibel menulis kode yang dapat digunakan kembali.

Kelas abstrak adalah kelas yang tidak dapat diinstansiasi secara langsung, tetapi berfungsi sebagai kerangka kerja untuk kelas turunan. Kelas abstrak mendefinisikan metode yang harus diterapkan oleh kelas turunannya. Jika kita ingin mendefinisikan kerangka kerja umum yang harus diikuti oleh kelas turunan, kita harus menggunakan kelas abstrak. Perbedaan utama antara kelas abstrak dan antarmuka adalah kelas abstrak dapat memiliki implementasi konkret untuk banyak metode, sedangkan antarmuka hanya mendefinisikan tanda tangan metode tanpa implementasi konkret.

Kelas beton adalah kelas yang dapat dipakai secara langsung dan memiliki implementasi konkret dari metode yang didefinisikan di dalamnya. Jika kita ingin membuat objek yang dapat digunakan secara langsung dan menerapkan perilaku tertentu, kita harus menggunakan kelas konkret. Kelas konkret menyediakan implementasi konkret untuk metode yang didefinisikan dalam kelas itu dan memungkinkan objek yang dibuat dari kelas itu untuk melakukan tugas tertentu secara langsung.

Metaclass adalah kelas yang digunakan untuk membuat dan mengontrol perilaku kelas lain (kelas). Metaclass memungkinkan kita untuk mengubah dan mengontrol cara kelas dibuat dan dioperasikan. Jika kita ingin mengubah perilaku sebuah instance, menambah atau mengubah metode dalam sebuah kelas, atau mengimplementasikan fungsionalitas tambahan, kita harus menggunakan metaclass. Perbedaan utama antara metaclass dan pewarisan biasa adalah bahwa metaclass bekerja pada tingkat pembuatan kelas sementara pewarisan bekerja pada tingkat objek.

# Daftar Pustaka

bestharadhakrishna. (2021, May 19). *Abstract Classes in Python*. Retrieved from https://www.geeksforgeeks.org/: https://www.geeksforgeeks.org/abstract-classes-in-python/

Choudhari, P. (2022, Februari 7). *What are metaclasses in Python*. Retrieved from https://www.python-engineer.com/: https://www.python-engineer.com/posts/metaclassespython/

Metaclasses, P. M. (2018). *Pengantar Metaclass PythonIntroduction to Python Metaclasses*. Retrieved from https://www.datacamp.com/: https://www.datacamp.com/tutorial/pythonmetaclasses#rdl

N, A. (2021, Oktober 17). *How to Use Abstract Classes in Python*. Retrieved from https://towardsdatascience.com/: https://towardsdatascience.com/how-to-use-abstractclasses-in-python-d4d2ddc02e90

naina024. (2020, Maret 26). *Python-interface module*. Retrieved from https://www.geeksforgeeks.org/: https://www.geeksforgeeks.org/python-interface-module/