

Nama : Agustinus Pardamean Lumban Tobing  
Kelas : PPTI 15  
NIM : 2602191251

### Self Learning 5 & Quiz OOP

#### 1. Jelaskan Encapsulation!

Encapsulation adalah prinsip dasar dalam object-oriented programming yang berfokus pada penyembunyian implementasi dari informasi yang dibutuhkan oleh objek, dan memberikan akses melalui method atau properti yang sudah ditentukan. Ini memungkinkan pengembang untuk membuat bagian dari kode mereka yang terisolasi dan tidak dapat dipengaruhi oleh bagian lain dari kode, sehingga membuat program lebih stabil dan mudah dalam memahami dan menjaga.

Dalam Java, encapsulation diterapkan dengan memberikan hak akses private pada data member (properti/field) dan memberikan metode publik yang memungkinkan akses dan modifikasi terhadap properti tersebut. Ini dikenal sebagai getter dan setter. Dengan menggunakan encapsulation, pengembang dapat membangun program yang lebih stabil, mudah dalam memahami dan memelihara, dan mempermudah penggunaan ulang kode.

Beberapa kegunaan dari encapsulation adalah sebagai berikut:

- Meningkatkan keamanan data: dengan menyembunyikan implementasi dan membatasi akses ke properti, kita dapat memastikan bahwa properti hanya dapat diakses dan dimodifikasi melalui metode yang sudah ditentukan.
- Mempermudah maintenance: dengan memisolasi implementasi dalam objek, pengembang dapat dengan mudah memodifikasi objek tanpa mempengaruhi bagian lain dari kode.
- Meningkatkan modularitas: encapsulation memungkinkan pengembang untuk membuat objek yang terisolasi dan terintegritas sehingga mempermudah dalam memahami dan menjaga kode.
- Mempermudah penggunaan ulang kode: encapsulation membuat objek dapat digunakan ulang tanpa perlu memahami implementasi detail.

#### 2. Jelaskan Inheritance!

Inheritance adalah prinsip dasar dalam object-oriented programming, yang memungkinkan pengembang untuk membuat kelas baru dengan menggunakan properti dan metode dari kelas yang sudah ada. Ini membantu pengembang dalam mengurangi duplikasi kode dan membuat struktur program lebih mudah dipahami dan dikelola.

Dalam Java, inheritance diterapkan dengan menggunakan kata kunci "extends" pada deklarasi kelas baru. Kelas yang diwarisi disebut sebagai kelas induk (superclass) dan kelas yang mewarisi disebut sebagai kelas turunan (subclass). Kelas turunan memiliki akses ke semua properti dan metode kelas induk, dan juga dapat memiliki properti dan metode tambahan. Inheritance membantu pengembang dalam membangun program yang lebih terstruktur, mudah dipahami dan dikelola, dan mempermudah reusability kode.

Beberapa kegunaan dari inheritance adalah sebagai berikut:

- Reusability: dengan mewarisi properti dan metode dari kelas lain, pengembang dapat mengurangi duplikasi kode dan mempermudah pemeliharaan.
- Abstraction: inheritance memungkinkan pengembang untuk membuat kelas abstrak yang tidak dapat diinstantiasi, tetapi dapat diwarisi oleh kelas lain.
- Polymorphism: inheritance memungkinkan pengembang untuk membuat kelas turunan yang memiliki metode yang sama namun memiliki implementasi yang berbeda.

3. Jelaskan perbedaan abstract class and interface!

Abstract class dan interface adalah dua konsep yang berbeda dalam pemrograman Java yang digunakan untuk mengimplementasikan inheritance dan polymorphism. Berikut adalah perbedaan antara keduanya:

- Abstract Class: Sebuah kelas abstrak adalah kelas yang tidak dapat diinstantiasi dan harus diteruskan oleh kelas turunannya. Sebuah kelas abstrak dapat mengandung metode abstrak (tanpa implementasi) dan metode yang memiliki implementasi.
- Interface: Sebuah interface adalah kumpulan deklarasi method tanpa implementasi. Implementasi method harus diberikan oleh kelas yang mengimplementasikan interface tersebut. Sebuah kelas dapat mengimplementasikan beberapa interface sekaligus.
- Method Implementation: Dalam sebuah kelas abstrak, Anda dapat memiliki metode abstrak dan metode dengan implementasi. Dalam sebuah interface, hanya ada deklarasi method tanpa implementasi, dan implementasi harus diberikan oleh kelas yang mengimplementasikan interface tersebut.
- Multiple Inheritance: Dalam Java, sebuah kelas hanya dapat mewarisi dari satu kelas lain. Namun, sebuah kelas dapat mengimplementasikan beberapa interface sekaligus, sehingga memberikan efek multiple inheritance.
- Keyword: Kata kunci yang digunakan untuk mendefinisikan kelas abstrak adalah **abstract**, sedangkan untuk interface adalah **interface**.

Dengan demikian, pemilihan antara kelas abstrak dan interface akan tergantung pada kebutuhan dan tujuan dari aplikasi Anda. Jika Anda ingin membuat sebuah kelas yang mengandung beberapa metode abstrak dan memiliki implementasi, Anda dapat menggunakan kelas abstrak. Namun, jika Anda ingin membuat sebuah kontrak yang harus diterapkan oleh kelas lain, Anda dapat menggunakan interface.

4. Jelaskan Polymorphism!

Polymorphism adalah yang memungkinkan suatu objek untuk memiliki beberapa bentuk atau tipe, dan menentukan tindakan yang diambil oleh objek tersebut berdasarkan tipe yang sesuai. Dalam Java, polymorphism dapat diterapkan melalui method overriding dan method overloading.

Kegunaan dari polymorphism adalah membuat kode lebih flexible dan reusable (dapat digunakan kembali). Ini juga mempermudah proses debugging dan testing karena memungkinkan kita untuk membuat abstraksi dari suatu tipe objek. Dalam beberapa kasus, polymorphism memungkinkan kita untuk menggantikan implementasi suatu objek dengan implementasi lain tanpa mengubah bagian kode yang menggunakan objek tersebut, yang membuat proses perbaikan bug atau perubahan fitur lebih mudah dilakukan.

Maka, dapat disimpulkan bahwa polymorphism merupakan teknik penting dalam pemrograman OOP yang membantu membuat kode yang lebih baik dalam hal modularitas, fleksibilitas, dan maintainability.