

PROJECT OOP
APLIKASI PELANGGAN

NAMA:HERMALIA PUTRI HANAFI

NIM:2201010279

KELAS:TI-KAB C

Saya Akan Membuat Sebuah Project Aplikasi Pelanggan Dengan Menggunakan Sebuah Program Yang Mengaplikasikan Access Modifier, Inheritance, Polymorphism, Dan Encapsulation. Project Yang Akan Saya Buat Ini Adalah Lanjutan Project Saya Di Ujian Tengah semester.

Encapsulation

Encapsulation adalah prinsip OOP (Object-Oriented Programming) yang mengatur akses terhadap data dan fungsi dalam sebuah objek. Dalam Java, encapsulation dicapai dengan cara:

1. **Menggunakan modifier akses** seperti private, protected, dan public untuk mengendalikan akses ke variabel dan metode.
2. **Menyediakan metode getter dan setter** untuk mengakses dan memodifikasi variabel privat.

Pada contoh aplikasi ini, encapsulation diterapkan dalam kelas Customer:

- **Variabel name dan purchaseAmount dideklarasikan sebagai private.** Ini berarti variabel-variabel ini hanya dapat diakses langsung dari dalam kelas Customer itu sendiri.
- **Metode getter dan setter disediakan** untuk mengakses dan memodifikasi variabel purchaseAmount. Variabel name hanya memiliki getter, karena tidak ada kebutuhan untuk mengubah nama pelanggan setelah objek Customer dibuat.

Polymorphism

Polymorphism adalah kemampuan objek untuk mengambil banyak bentuk. Dalam Java, ini biasanya dicapai melalui **inheritance** (pewarisan) dan **interface**.

Pada contoh aplikasi ini, polymorphism diterapkan melalui kelas Customer dan subclass (RegularCustomer dan PremiumCustomer):

- **Kelas Customer adalah kelas abstrak** yang mendeklarasikan metode abstrak getDiscount(). Kelas ini tidak dapat diinstansiasi secara langsung dan harus diturunkan oleh kelas lain.
- **Kelas RegularCustomer dan PremiumCustomer meng-override metode getDiscount()** dari kelas Customer. Ini memungkinkan objek dari kedua kelas tersebut untuk memiliki implementasi spesifik dari metode getDiscount().

Pada saat runtime, metode yang dipanggil tergantung pada tipe objek yang mengimplementasikannya. Ini adalah esensi dari polymorphism, yang memungkinkan metode yang sama untuk bekerja dengan cara yang berbeda tergantung pada objek yang menggunakannya.

Penggunaan dalam PelangganApp

Dalam kelas PelangganApp, kita dapat melihat bagaimana polymorphism dan encapsulation bekerja bersama-sama:

- **Polymorphism:** Meskipun customer1 dan customer2 dideklarasikan sebagai tipe Customer, objek yang sebenarnya adalah RegularCustomer dan PremiumCustomer.

Ketika `getDiscount()` dipanggil, metode yang benar dipilih berdasarkan tipe objek yang sebenarnya.

- **Encapsulation:** Variabel `purchaseAmount` diakses dan dimodifikasi melalui metode setter dan getter. Nama pelanggan hanya dapat dibaca melalui metode getter.

Kesimpulan

Dengan menggunakan prinsip encapsulation dan polymorphism, aplikasi ini menjadi lebih modular, mudah untuk dipelihara, dan fleksibel. Encapsulation melindungi data dari akses langsung yang tidak sah, sementara polymorphism memungkinkan metode yang sama untuk berperilaku berbeda tergantung pada objek yang mengimplementasikannya.



