

OBJECT ORIENTED PROGRAMMING

QUIS 1



INSTIKI
INSTITUT BISNIS DAN
TEKNOLOGI INDONESIA

Nama : I Gusti Agung Raditiya Gotama

NIM : 2201010243

Kelas : L

Usulan Ide Sebuah Aplikasi “Rental Motor”

Dalam proyek ini saya membuat aplikasi “Rental Motor”. Alasan utama saya memilih untuk membuat aplikasi “Rental Motor” yaitu karena ingin mengaplikasikannya pada bisnis keluarga. Disamping itu, saya juga ingin mengembangkan proyek UTS pada beberapa waktu lalu. Dimana nantinya aplikasi tersebut harus bisa menambah data, mengubah data, menghapus data, dan terdapat fitur untung menghitung dari kalkulasi harga yang harus dibayar. Tentunya aplikasi “Rental Motor” ini harus menggunakan database. Dalam pembuatan aplikasi “Rental Motor” ini, saya mengaplikasikan teori OOP berupa akses modifier, inheritance, polymorphism dan encapsulation. Berikut penjelasan tentang bagaimana teori-teori tersebut diterapkan dalam kode ini:

1. Access Modifier

Access modifier digunakan dalam mendeklarasikan kelas, variabel, dan metode. Misalnya:

- a) Public: Kelas Rental dan beberapa metode seperti list_all, StoreData, UpdateData, dan DestroyData dideklarasikan sebagai public, yang berarti mereka dapat diakses dari mana saja.
- b) Private: Tidak ada variabel instance dalam kode ini yang dideklarasikan sebagai private, namun ini akan menjadi praktik yang baik untuk membatasi akses langsung dan menggunakan metode getter dan setter sebagai gantinya.

2. Encapsulation

Encapsulation adalah konsep menyembunyikan detail implementasi dan hanya mengekspos antarmuka yang diperlukan. Dalam kode ini:

- a) Variabel dan Metode Private: Misalnya, variabel tm dideklarasikan sebagai private, yang berarti hanya dapat diakses dari dalam kelas Rental.
- b) Metode Getter dan Setter: Kode ini tidak secara eksplisit menggunakan getter dan setter, tetapi penggunaan metode seperti list_all(), StoreData(), UpdateData(), dan DestroyData() menunjukkan cara yang baik untuk mengatur akses ke data dan logika program.

3. Inheritance (Pewarisan)

Inheritance tidak terlihat secara eksplisit dalam kode yang diberikan, tetapi dapat disimpulkan dari struktur kelas. Misalnya:

- a) Kelas Rental: Kelas Rental mewarisi dari javax.swing.JFrame, yang merupakan bagian dari hierarki pewarisan di Java. Ini memungkinkan Rental untuk memiliki properti dan metode dari JFrame.

4. Polymorphism

Polymorphism tidak digunakan secara eksplisit dalam kode ini, tetapi konsepnya dapat diterapkan. Misalnya:

- a) Method Overriding: Jika ada kelas lain yang mewarisi dari Rental dan mengoverride metode seperti StoreData, itu contoh dari polymorphism.

