Nama: I Kadek Aditia Pradipta

Nim : 2201010022

Ide dan Penjelasan Proyek

Proyek: Aplikasi Transaksi Penjualan Barang Elektronik

Deskripsi:

Proyek ini adalah aplikasi transaksi penjualan barang elektronik yang dirancang menggunakan Java dan konsep-konsep dasar OOP (Object-Oriented Programming). Aplikasi ini memungkinkan pengguna untuk melakukan transaksi dengan memilih barang, menghitung total harga, membayar, dan mendapatkan kembalian. Data transaksi juga dapat ditampilkan dalam sebuah tabel.

Mengapa Menggunakan Teori OOP?

1. Modularitas dan Reusability:

- Inheritance: Dengan mewarisi dari `javax.swing.JFrame`, kita bisa memanfaatkan fungsionalitas yang sudah ada di kelas `JFrame` tanpa harus menulis ulang kode tersebut. Hal ini meningkatkan reusability dan mengurangi duplikasi kode.

- Encapsulation: Dengan menggunakan akses modifier seperti `private` dan `public`, kita dapat menyembunyikan implementasi detail dari kelas dan hanya mengekspos bagian yang diperlukan. Ini membuat kode lebih modular dan mudah dimengerti.

2. Maintainability:

- Encapsulation: Dengan membungkus data dan metode dalam satu unit (kelas), kita dapat menjaga integritas data dan mencegah perubahan yang tidak diinginkan dari luar kelas. Jika ada perubahan dalam implementasi, kita hanya perlu mengubah dalam satu tempat tanpa mempengaruhi kode lainnya.

- Polymorphism: Meskipun tidak terlihat jelas dalam kode ini, dengan menggunakan polymorphism kita dapat memperlakukan objek dari berbagai kelas terkait sebagai objek dari kelas dasar yang sama. Hal ini memudahkan kita untuk menambah atau mengubah fungsionalitas di masa depan tanpa mengubah kode yang sudah ada.

3. Efisiensi Pengembangan:

- Inheritance: Dengan mewarisi fungsionalitas dari kelas lain, kita bisa mengurangi waktu pengembangan dan fokus pada fungsionalitas spesifik yang dibutuhkan oleh aplikasi kita.

- Encapsulation: Dengan membungkus data dan metode dalam satu unit (kelas), kita bisa memastikan bahwa perubahan dalam satu bagian kode tidak mempengaruhi bagian lain, yang bisa mengurangi waktu debugging dan testing.

Detail Implementasi OOP dalam Proyek

1. Access Modifiers:

- Public: Digunakan untuk metode dan variabel yang harus diakses dari luar kelas, seperti event handler dan variabel GUI.

-Private: (Idealnya) digunakan untuk variabel yang hanya digunakan di dalam kelas untuk menjaga enkapsulasi. Meskipun dalam kode ini belum diterapkan, ini adalah praktik yang baik untuk menjaga data tetap aman.

2. Inheritance:

- `public class transaksi1 extends javax.swing.JFrame`:

- Kelas `transaksi1` mewarisi semua fungsionalitas dari `JFrame`, yang memungkinkan kita untuk membuat GUI dengan komponen seperti tombol, text field, dan tabel.

3. Polymorphism:

- Polymorphism tidak digunakan secara eksplisit dalam kode ini, tetapi bisa diterapkan dalam skenario yang lebih kompleks, seperti menggunakan interface untuk berbagai jenis transaksi.

4. Encapsulation:

- Variabel-variabel instance seperti `nm_barang`, `harga_brg`, `jml_beli`, `jumlah_harga`, `kembalian`, dan `jumlahbayar` seharusnya dienkapsulasi menggunakan `private` dan diakses melalui getter dan setter untuk menjaga integritas data.

- Misalnya:

```
java
public class Transaksi {
    private String nm_barang;
    private long harga_brg;
    private int jml_beli;
    private long jumlah_harga;
    private long kembalian;
    private long jumlahbayar;
    // Getter and Setter methods
    public String getNm_barang() {
        return nm_barang;
    }
    public void setNm_barang(String nm_barang) {
        this.nm_barang = nm_barang;
    }
// Similar getters and setters for other fields
}
```

Kesimpulan

Menggunakan teori OOP dalam proyek ini memberikan beberapa keuntungan seperti modularitas, reusability, maintainability, dan efisiensi pengembangan. Dengan menerapkan inheritance, kita dapat

memanfaatkan fungsionalitas yang sudah ada. Encapsulation membantu menjaga integritas data dan mempermudah maintenance. Polymorphism, meskipun tidak digunakan secara eksplisit dalam kode ini, dapat meningkatkan fleksibilitas dan extensibility aplikasi di masa depan. Implementasi OOP dalam proyek ini menciptakan struktur kode yang lebih terorganisir dan mudah untuk dikembangkan lebih lanjut.