Laporan Praktikum

Pemrograman Berorientasi Objek



Disusun Oleh :

**Irgi Fatihul Ihsan (2311533010)**

Dosen Pengampu : Afdhal Dinilhak, M.Kom

Departemen Informatika

Fakultas Teknologi Informasi

Universitas Andalas

Tahun 2024

**Design Pattern**

1. **Tujuan Praktikum**

Mampu menggunakan design pattern jenis builder pattern pada kasus CRUD aplikasi.

1. **Pendahuluan**

Design Patterns adalah solusi standar yang telah terbukti (teruji) untuk menyelesaikan masalah umum yang sering terjadi dalam pengembangan perangkat lunak. Design Patterns bukanlah hal yang bisa ditranslate ke dalam kode program secara langsung, melainkan hanya template atau panduan cara menyelesaikan masalah. Terdapat banyak sekali desain pattern, dan untuk implementasi ke dalam kode programnya mungkin akan sedikit berbeda tergantung teknologi dan bahasa pemrograman yang digunakan. Dengan memahami design pattern kita akan lebih mudah dalam mengatasi masalah yang terjadi dalam pembuatan aplikasi/software. Karena kita sudah tahu solusi-solusi umum yang harus dilakukan, berbeda jika belum mengerti design pattern maka kita mencari solusi dari permasalahan yang dihadapi dengan mencoba- coba solusi versi kita sendiri padahal sebenarnya sudah ada solusi yang sudah terbukti untuk masalah tersebut.

**Builder Pattern**

Builder pattern merupakan salah satu design pattern paling banyak digunakan dalam software development. Builder Pattern digunakan untuk memisahkan logika konstruksi sebuah objek dari representasi final objek tersebut. Builder Pattern dirancang untuk membantu dalam pembuatan objek kompleks dengan memberikan kontrol yang lebih besar atas proses konstruksinya dan memastikan bahwa objek tersebut dibuat secara konsisten. Pattern ini sangat berguna saat sebuah objek yang memiliki banyak atribut, beberapa di antaranya opsional, atau saat proses pembuatannya membutuhkan beberapa langkah.

1. **Metode Praktikum**
2. Modifikasi class Customer pada package model dengan menambahkan constructor dan menghapus Setter, karena Setter akan diletakkan di kelas lain.

public class Customer {

private String id,nama,email, alamat, hp;

public Customer(String id, String nama, String email, String alamat, String hp) {

this.id = id;

this.nama = nama;

this.email = email;

this.alamat = alamat;

this.hp = hp;

}

public String getId() {

return id;

}

public String getNama() {

return nama;

}

public String getEmail() {

return email;

}

public String getAlamat() {

return alamat;

}

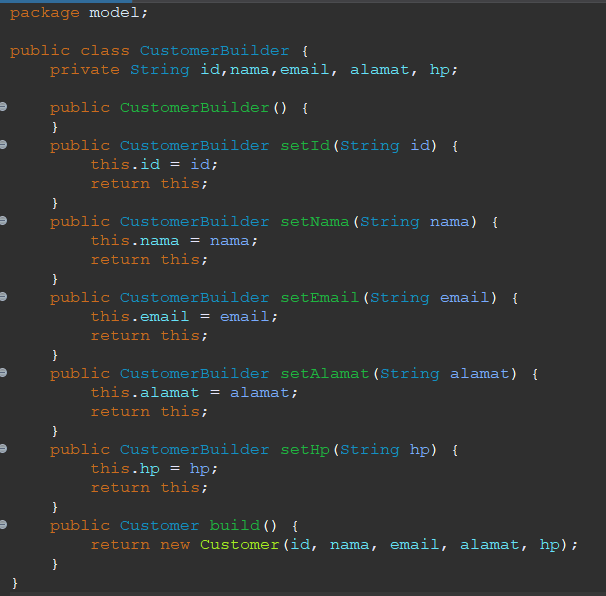
public String getHp() {

return hp;

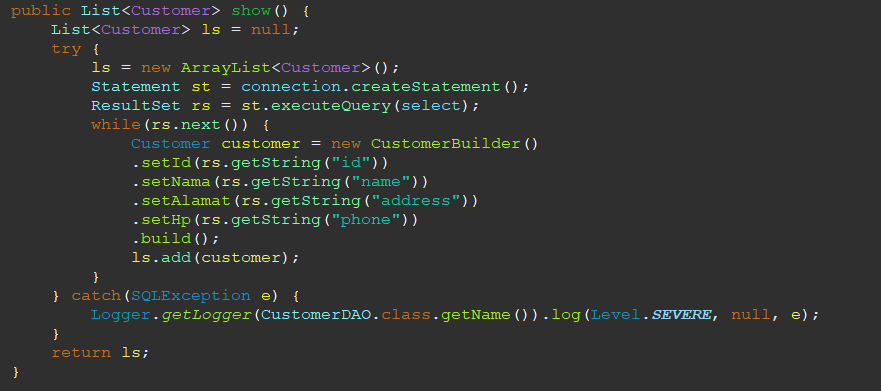
}

}

1. Buat class CustomerBuilder dalam package model
2. Copy semua atribut dari class customer ke CustomerBuilder, dan tambahkan Setter. Kemudian buat method build untuk return value Customer



1. Modifikasi method show yang ada pada class CustomerRepo yang sebelumnya mengimplementasikan dari model Customer, menjadi menggunakan tambahakn CustomerBuilder yang menerapkan Builder Pattern



1. Implementasi juga method-method yang lain



1. Selannjutnya, modifikasi kode pada tombol save dan update di CustomerFrame untuk menerapkan Builder Pattern

****