Laporan Praktikum

Pemrograman Berorientasi Objek



Disusun Oleh :

**NIA RAMADHANI**

**2311531006**

Dosen Pengampu : Nurfiah, S.ST, M.Kom.

Departemen Informatika

Fakultas Teknologi Informasi

Universitas Andalas

Tahun 2024

**Desain Aplikasi Laundry**

1. **Tujuan Praktikum**
2. Memahami kembali GUI menggunakan JFrame
3. Memahami penggunaan class, object, encapsulation, constructor, dan method pada Pemograman Berorientasi Objek
4. **Pendahuluan**

Pemrograman berorientasi objek (OOP) adalah paradigma pemrograman yang berfokus pada penggunaan **class** dan **object** sebagai dasar pengembangan perangkat lunak. Konsep-konsep utama dalam OOP, seperti class, object, method, dan constructor, sangat penting untuk membangun aplikasi yang terstruktur dengan baik dan mudah dikelola.

**Class** adalah konsep fundamental dalam OOP yang berfungsi sebagai blueprint atau template untuk membuat objek. Sebuah class mendefinisikan sekumpulan karakteristik dan perilaku yang dimiliki oleh objek-objek yang dibuat berdasarkan class tersebut. Di dalam bahasa pemrograman Java, class dapat berisi data member, method, constructor, nested class, dan interface. Data member adalah atribut yang menyimpan informasi tentang objek, sedangkan method adalah fungsi yang menggambarkan perilaku objek. Constructor adalah metode khusus yang digunakan untuk menginisialisasi objek pada saat pembuatan, sementara nested class dan interface digunakan untuk mendefinisikan struktur tambahan yang mendukung fungsionalitas class.

**Object** merupakan instansi nyata dari sebuah class dan mencerminkan entitas yang ada di dunia nyata. Setiap objek memiliki **state** (atribut), **behavior** (method), dan **identity** yang membedakannya dari objek lainnya. State mencakup data yang disimpan dalam atribut objek, behavior mencakup operasi yang dapat dilakukan pada objek melalui method, dan identity memastikan bahwa setiap objek dapat diidentifikasi secara unik.

**Method** adalah blok kode yang dirancang untuk melakukan tindakan tertentu dan dapat dipanggil berulang kali sepanjang program. Di Java, terdapat berbagai method bawaan seperti `main()`, `equals()`, dan `toString()` yang menyediakan fungsionalitas dasar untuk manipulasi objek dan interaksi antar objek. Method mempermudah pengorganisasian kode dan meningkatkan keterbacaan serta pemeliharaan program.

**Constructor**, di sisi lain, adalah metode khusus yang digunakan untuk menginisialisasi objek dari sebuah class. Constructor secara otomatis dipanggil saat objek dibuat, dan bertugas untuk mengatur nilai awal dari atribut-atribut objek. Ini memastikan bahwa objek dimulai dalam kondisi yang valid dan siap digunakan.

Praktikum ini bertujuan untuk menerapkan konsep-konsep OOP dalam konteks pengembangan aplikasi berbasis Java, khususnya aplikasi laundry. Dalam proses ini, dua package utama dibuat: **model** dan **ui**. Package model berisi berbagai class seperti **User**, Customer, **Order**, dan **Service** yang mendefinisikan struktur data dan logika bisnis aplikasi. Sementara itu, package ui berisi JFrame untuk antarmuka pengguna, termasuk **LoginFrame** dan **MainFrame**, yang menyediakan tampilan grafis dan interaksi pengguna dengan aplikasi.

Dengan memisahkan logika bisnis dari antarmuka pengguna dan mengimplementasikan fitur login, praktikum ini memberikan pemahaman mendalam tentang penerapan OOP dalam pembangunan aplikasi nyata. Penekanan pada penggunaan class, object, method, dan constructor membantu menciptakan aplikasi yang terorganisir dengan baik, mudah dikelola, dan memenuhi kebutuhan fungsional pengguna.

1. **Metode Praktikum**
2. Buat Java Project baru dengan nama laundryApps, dan buat 2 package untuk model dan ui.
3. Buat class baru pada model dengan nama User dan tambahkan atribut id, nama, username, dan password.

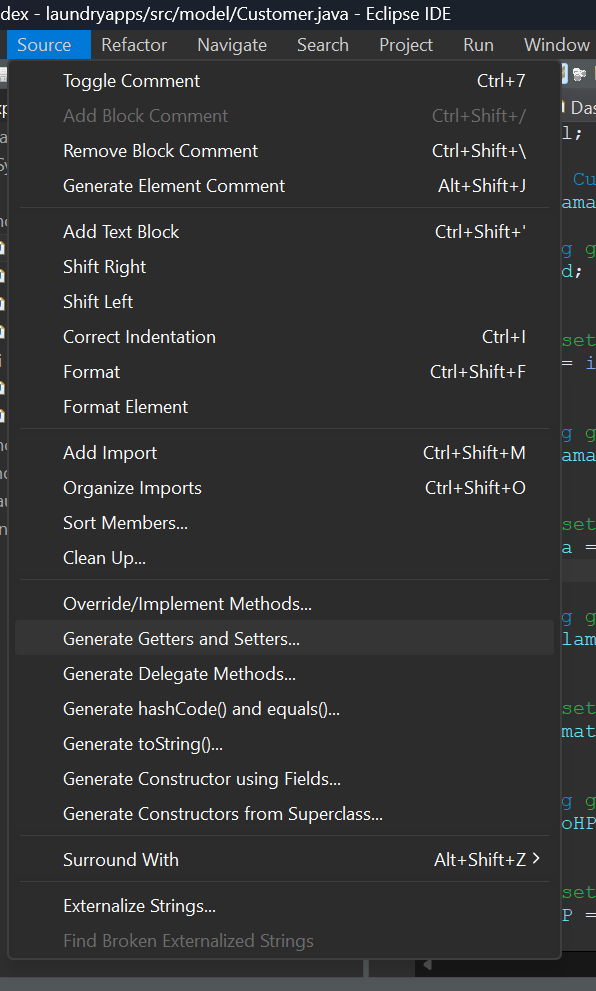
package model;

public class User {

String id, nama, username, password;

}

1. Tambahkan getter setter untuk masing-masing atribut. Untuk ini, bisa menggunakan pintasan dengan klik kanan pada laman source > Source > Generate Getters and Setters.



1. Setelah itu, getter dan setter akan ditambahkan secara otomatis.

package model;

public class User {

String id, nama, username, password;

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

public String getNama() {

return nama;

}

public void setNama(String nama) {

this.nama = nama;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

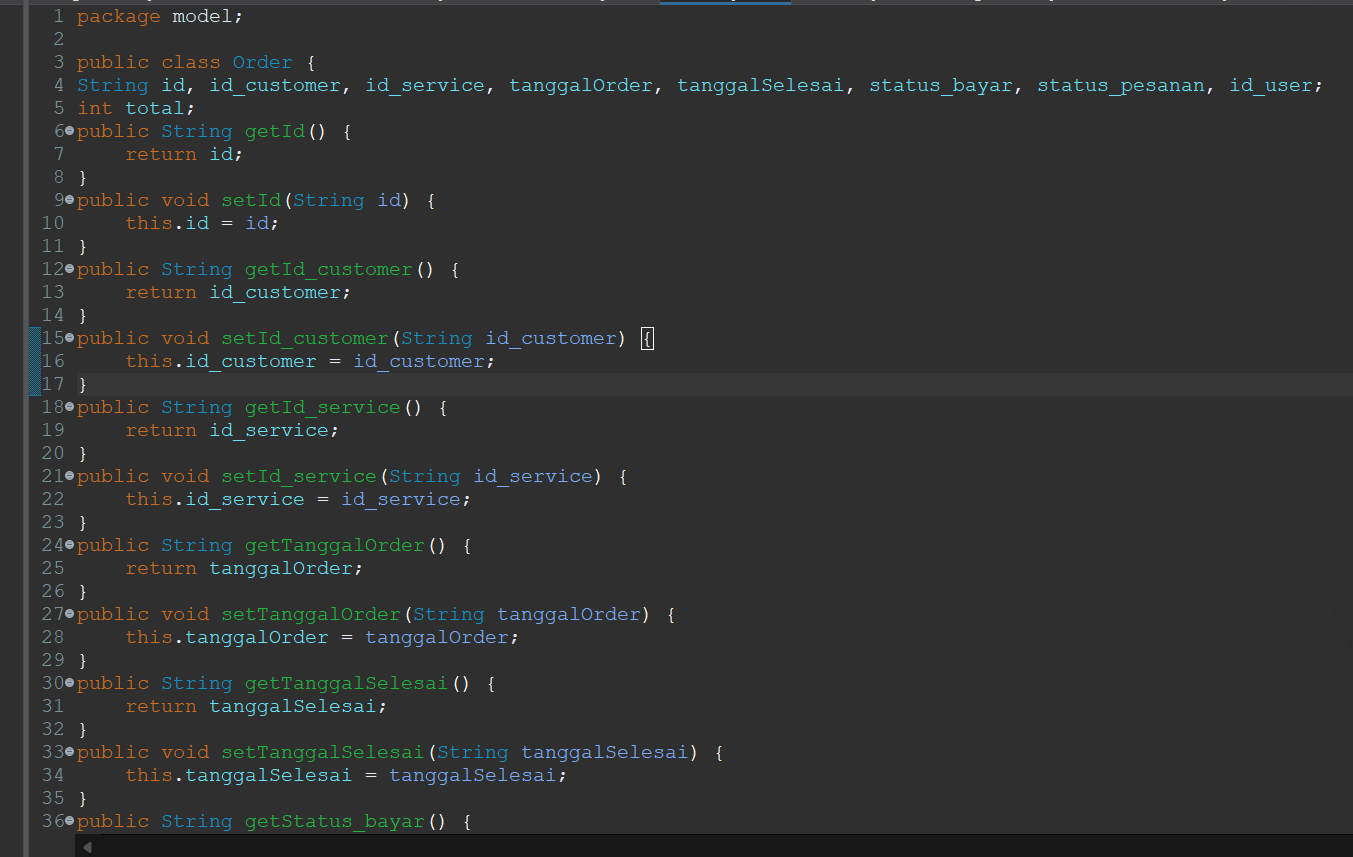
}

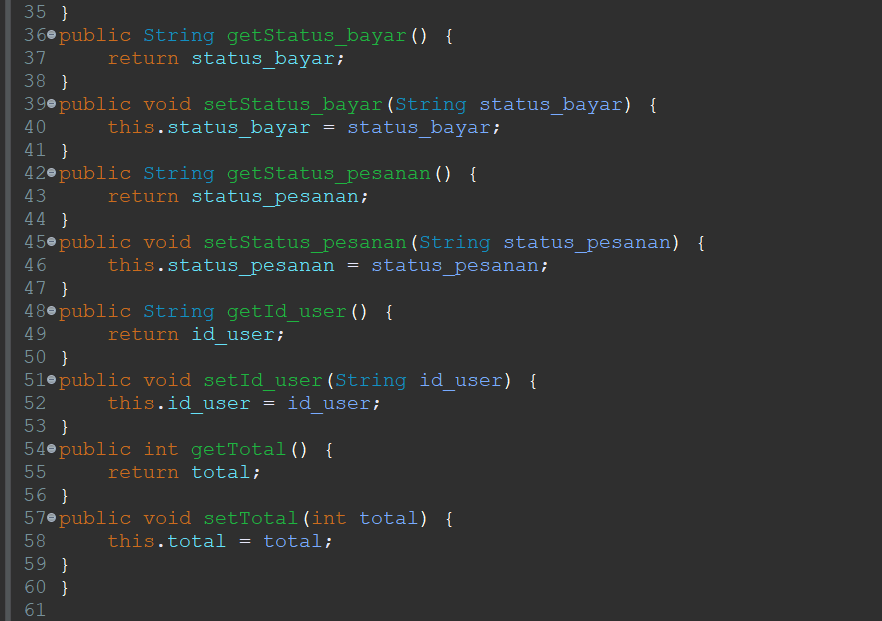
1. Tambahkan class lain seperti Customer, Order dan class Service. Masing-masing class diberi atribut dan getter setter nya.

* Class Customer

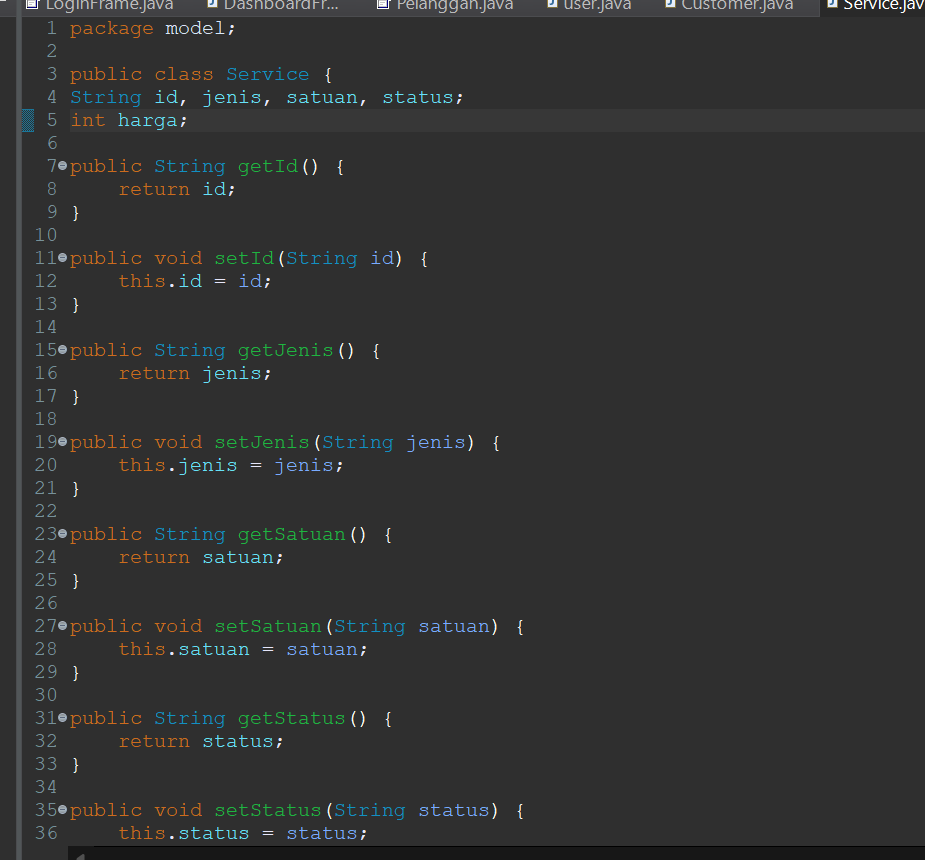


* Class Order



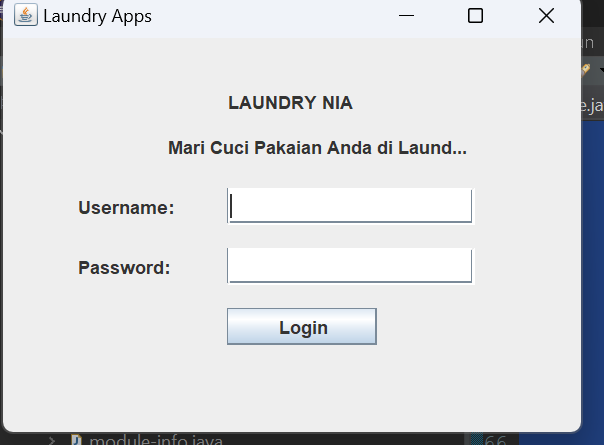


* Class Service

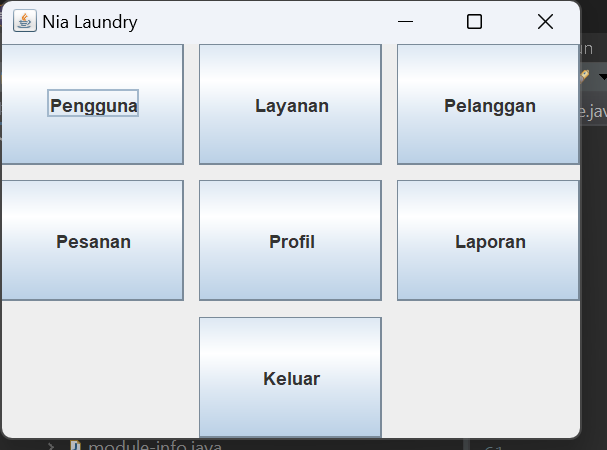


1. Selanjutnya, buat GUI menggunakan JFrame. GUI yang dibuat adalah LoginFrame dan MainFrame.

* LoginFrame



* MainFrame



1. Tambahkan method login pada class Users

public static boolean login(String username, String password) {

boolean isLogin = false;

User user = new User();

user.setId("1");

user.setNama("nia");

user.setUsername("nia");

user.setPassword("12345");

if(user.getUsername().equalsIgnoreCase(username.trim())

&& user.getPassword().equals(password)) {

isLogin = true;

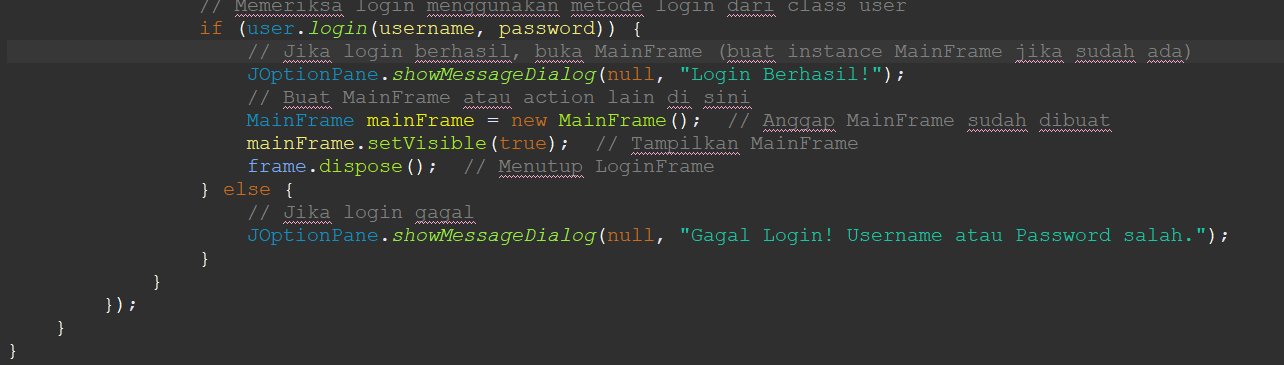
} else {isLogin=false;}

return isLogin;

}

Method tersebut menggunakan getter setter yang telah dibuat pada class User. equalsIgnoreCase merupakan fungsi untuk mengembalikan String yang menjadi instance tanpa melihat aturan case String tersebut. trim() berfungsi untuk menghapus karakter spasi pada String.

1. Hubungkan LoginFrame dengan class User. Pada source tombol Masuk pada LoginFrame, tambahkan kode berikut:



### Penjelasan Penghubungan:

* **user.login(username, password)**: Method login dari class user dipanggil untuk memverifikasi apakah kombinasi username dan password yang dimasukkan benar.
* Jika metode login mengembalikan true, maka login dianggap berhasil dan akan ditampilkan pesan "Login Berhasil!" serta ditampilkan MainFrame.
* Jika metode login mengembalikan false, akan ditampilkan pesan "Gagal Login! Username atau Password salah."

Method login ini adalah jembatan yang menghubungkan antara input user (username dan password) dari GUI (LoginFrame) dan logika bisnis yang ada di class user untuk verifikasi login.

Jadi, **kode penghubung utama antara LoginFrame dan class user adalah pemanggilan user.login(username, password)**.

Method login dari class User digunakan, dengan instance adalah isi dari text field username dan password. setVisible(true) untuk menampilkan JFrame MainFrame, dan dispose() untuk menutup halaman ini, yaitu LoginFrame.

1. **Kesimpulan Praktikum**

Praktikum ini menunjukkan penerapan konsep pemrograman berorientasi objek dalam pengembangan aplikasi laundry. Dengan membagi aplikasi menjadi dua package, yaitu **model** dan **ui**, kita memisahkan logika bisnis dari antarmuka pengguna. Pada package model, class-class seperti **User**, **Customer**, **Order**, dan **Service** diciptakan dengan atribut dan getter-setter yang sesuai untuk mendukung fungsionalitas aplikasi. Sementara itu, package ui berisi JFrame untuk tampilan login dan tampilan utama, yang menghubungkan antarmuka pengguna dengan logika verifikasi login.

Metode **login** dalam class **User** digunakan untuk memverifikasi kredensial pengguna dan menghubungkan antarmuka login dengan logika bisnis. Dengan demikian, pengguna dapat login dan mengakses tampilan utama aplikasi setelah berhasil memasukkan username dan password yang benar.

Secara keseluruhan, praktikum ini memberikan pemahaman yang jelas tentang bagaimana class, objek, method, dan constructor bekerja dalam konteks pengembangan aplikasi berbasis GUI di Java, serta bagaimana memisahkan logika aplikasi dari antarmuka pengguna untuk menghasilkan sistem yang lebih terstruktur dan mudah dikelola.