**LAPORAN PRAKTIKUM PEMOGRAMAN BERORIENTASI OBJEK (PBO)**

***CLASS, OBJECT, CONSTRUCTOR dan METHOD***



OLEH :

FAYI AMATULLAH AZHARA

(2311537001)

DOSEN PENGUMPU : Nurfiah, S.ST, M.Kom.

**FAKULTAS TEKNOLOGI INFORMASI**

**DEPARTEMEN INFORMATIKA**

**UNIVERSITAS ANDALAS**

1. **PENDAHULUAN**

Praktikum ini merupakan bagian dari pembelajaran mata kuliah Pemrograman Berorientasi Objek (PBO) menggunakan bahasa pemrograman Java. Sebagai mahasiswa Informatika, pemahaman tentang konsep dasar PBO seperti **class**, **object**, **constructor**, dan **method** sangatlah penting, terutama dalam penerapannya pada kasus nyata. Melalui praktikum ini, saya berkesempatan untuk mempelajari dan mengimplementasikan konsep tersebut dalam pembuatan aplikasi sederhana berbasis Java, yaitu aplikasi manajemen laundry.

Pada praktikum ini, saya diminta untuk membuat beberapa **class** seperti `User`, `Costumer`, `Service`, dan `Order` yang masing-masing memiliki atribut dan method tertentu untuk mendukung fungsionalitas aplikasi. Selain itu, praktikum ini juga melibatkan pembuatan antarmuka pengguna menggunakan \***JFrame**, di mana aplikasi memiliki tampilan login dan halaman utama. Proses pembuatan aplikasi ini mengajarkan saya tentang bagaimana menghubungkan antar tampilan dan memanfaatkan method untuk berpindah dari satu jendela ke jendela lainnya.

Secara keseluruhan, praktikum ini bertujuan agar mahasiswa tidak hanya memahami teori, tetapi juga mampu menerapkan konsep PBO dalam pembuatan aplikasi yang bermanfaat dan relevan dengan kebutuhan industri saat ini.

1. **TUJUAN**

Tujuan praktikum ini yaitu mahasiswa mampu membuat class, object, encapsulation, constructor

Mampu membuat fungsi berpindah dari tampilankannya pada aplikasi manajemen laundry,

Adapun pada praktikum ini mahasiswa akan mempelajari beberapa poin

yaitu :

• Membuat class, object, encapsulation, contstruktor dan method untuk aplikasi laundry

seperti class User, Costumer, Service dan Order

• Membuat desain antarmuka aplikasi Laundry Login dan Halaman Utama

• Mampu menggunakan method pada Jframe/Tampilan aplikasi

• Mampu membuat fungsi berpindah dari tampilan login ke halaman utama

1. **Alat**

Computer / laptop yang telah terinstall JDK dan Eclipse

1. **LANGKAH – LANGKAH**
2. Membuka Eclipse IDE dan membuat proyek dengan nama “laundryapps”
3. Membuat 2 Package baru directory src dengan nama “model” dan “ui”

Package **model** berfungsi untuk menyimpan class-class yang diperlukan untuk pembuatan

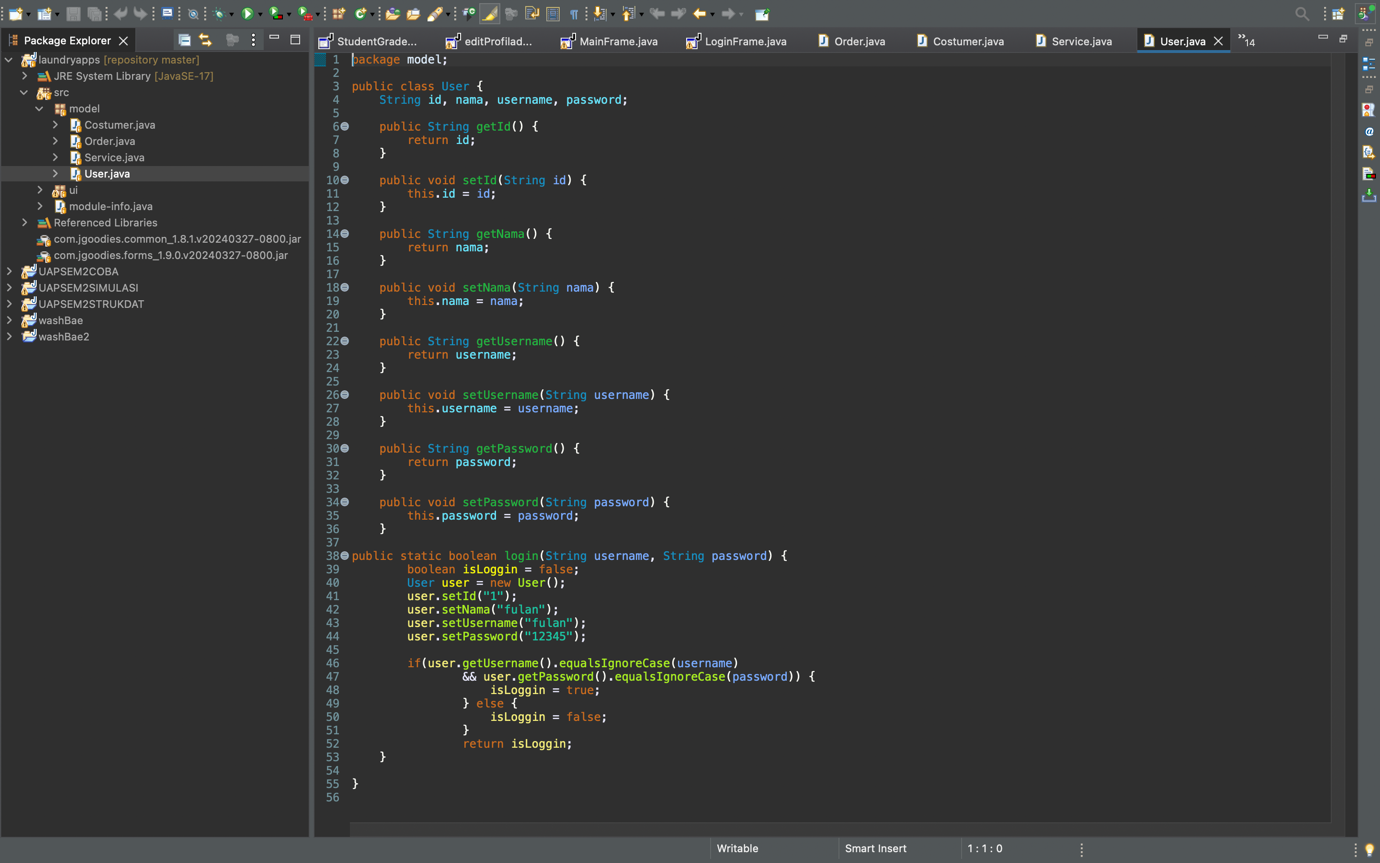
aplikasi laundry, sedangkan package **ui** digunakan untuk menyimpan tampilan atau antar muka aplikasi laundry.



1. Selanjutnya buat class baru pada package model dengan nama User.

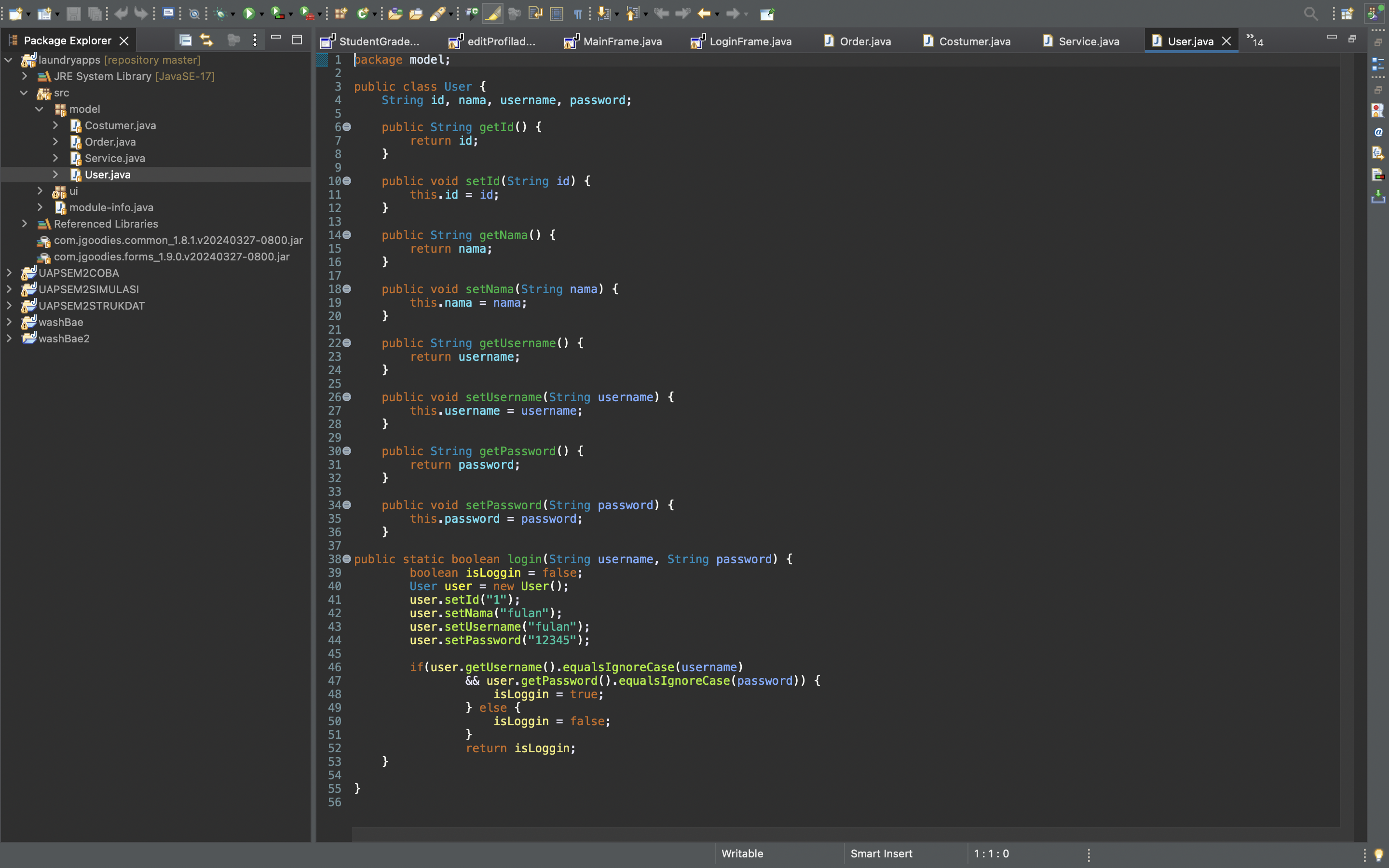
Membuat setter dan getter attribute user, method ini berfungsi untuk

memasukkan/mengubah dan menampilkan value dari suatu object.



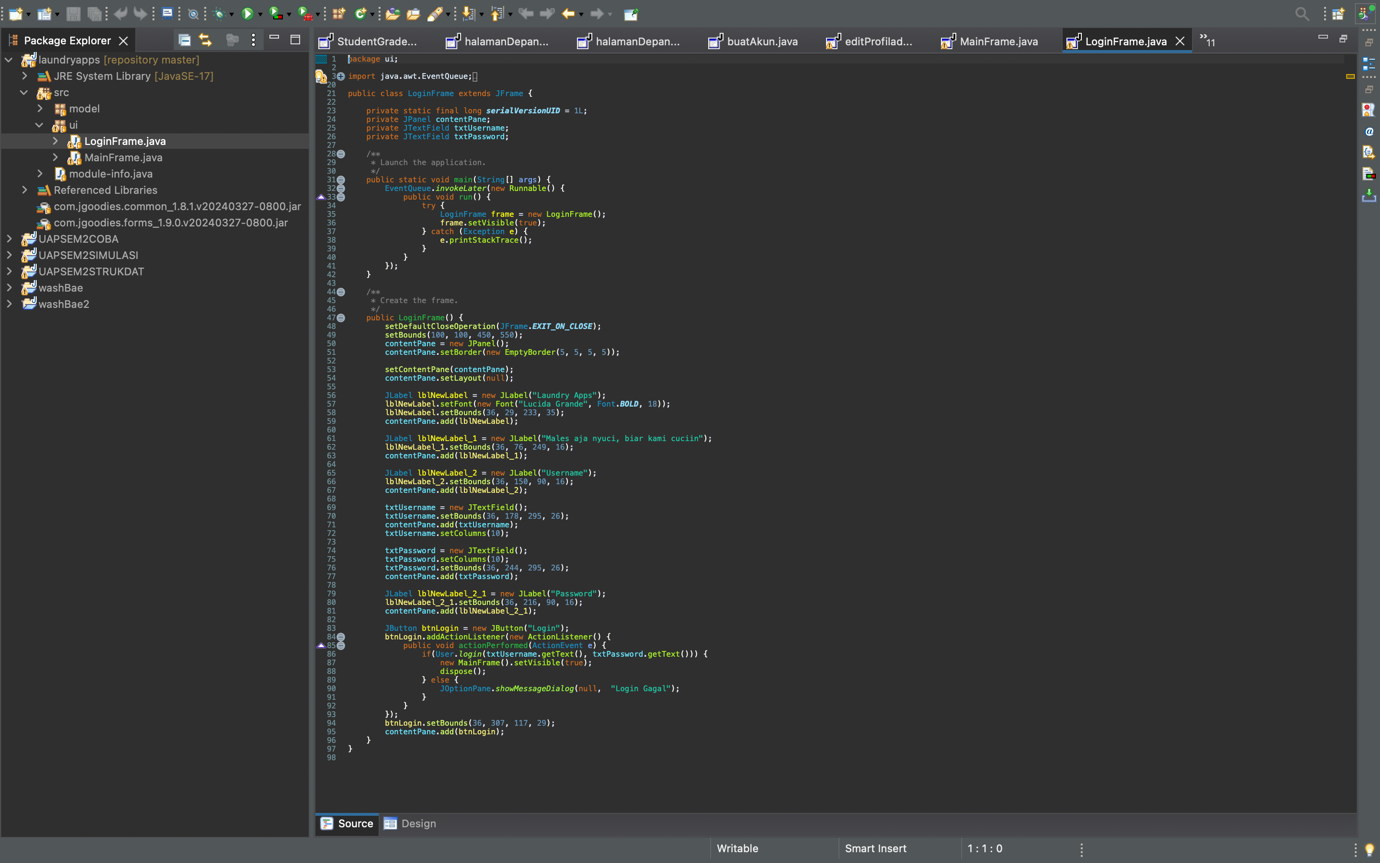
1. Membuat Method login yang nantinya akan digunakan Ketika pengguna akan login ke

aplikasi.

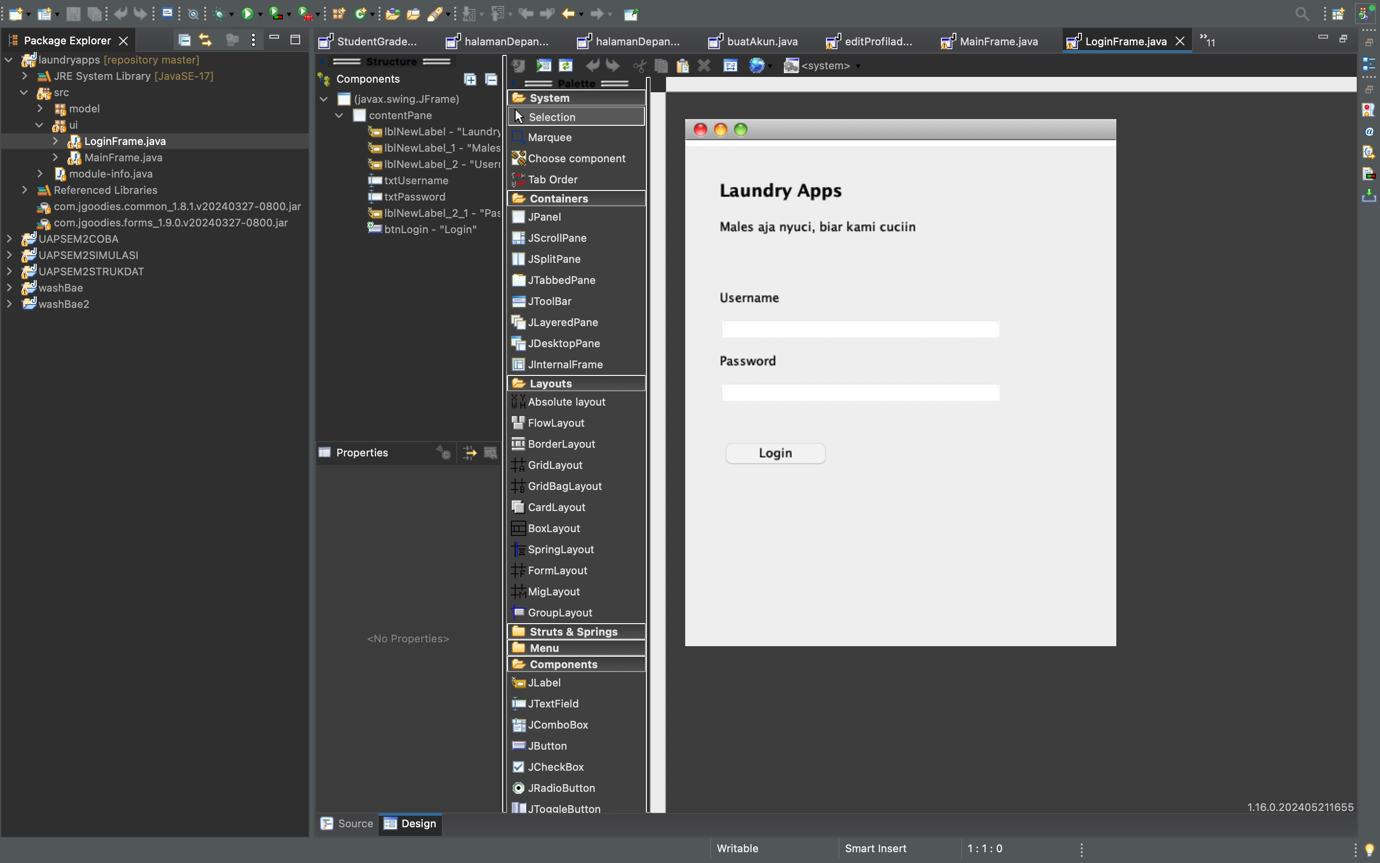


**MEMBUAT TAMPILAN JFRAME**

1. buat jFrame baru pada package ui dengan nama LoginFrame

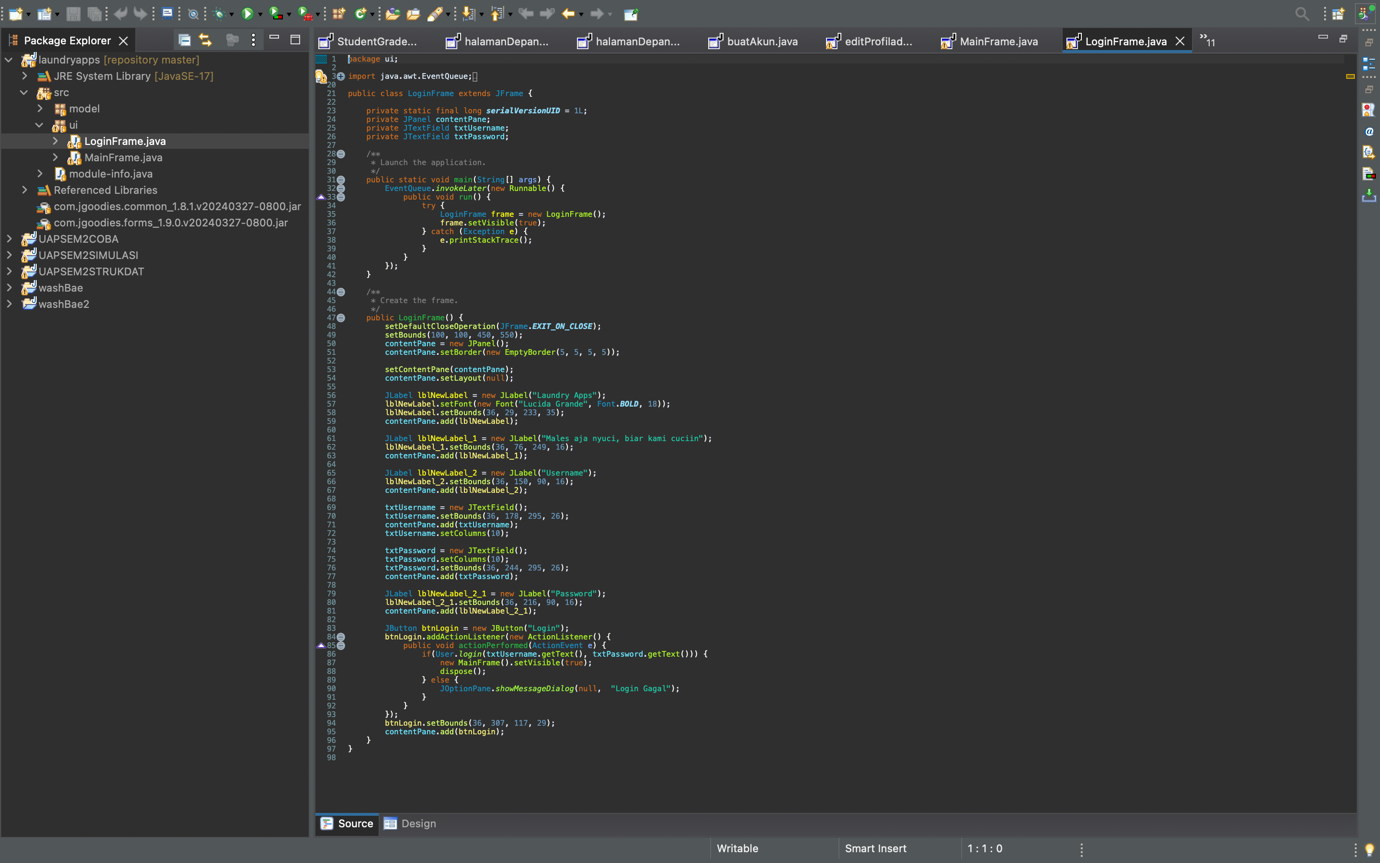


1. kemudian desain tampilan login seperti gambar dibawah ini.



1. Kemudian panggil method login pada class User dengan mengirimkan parameter yang

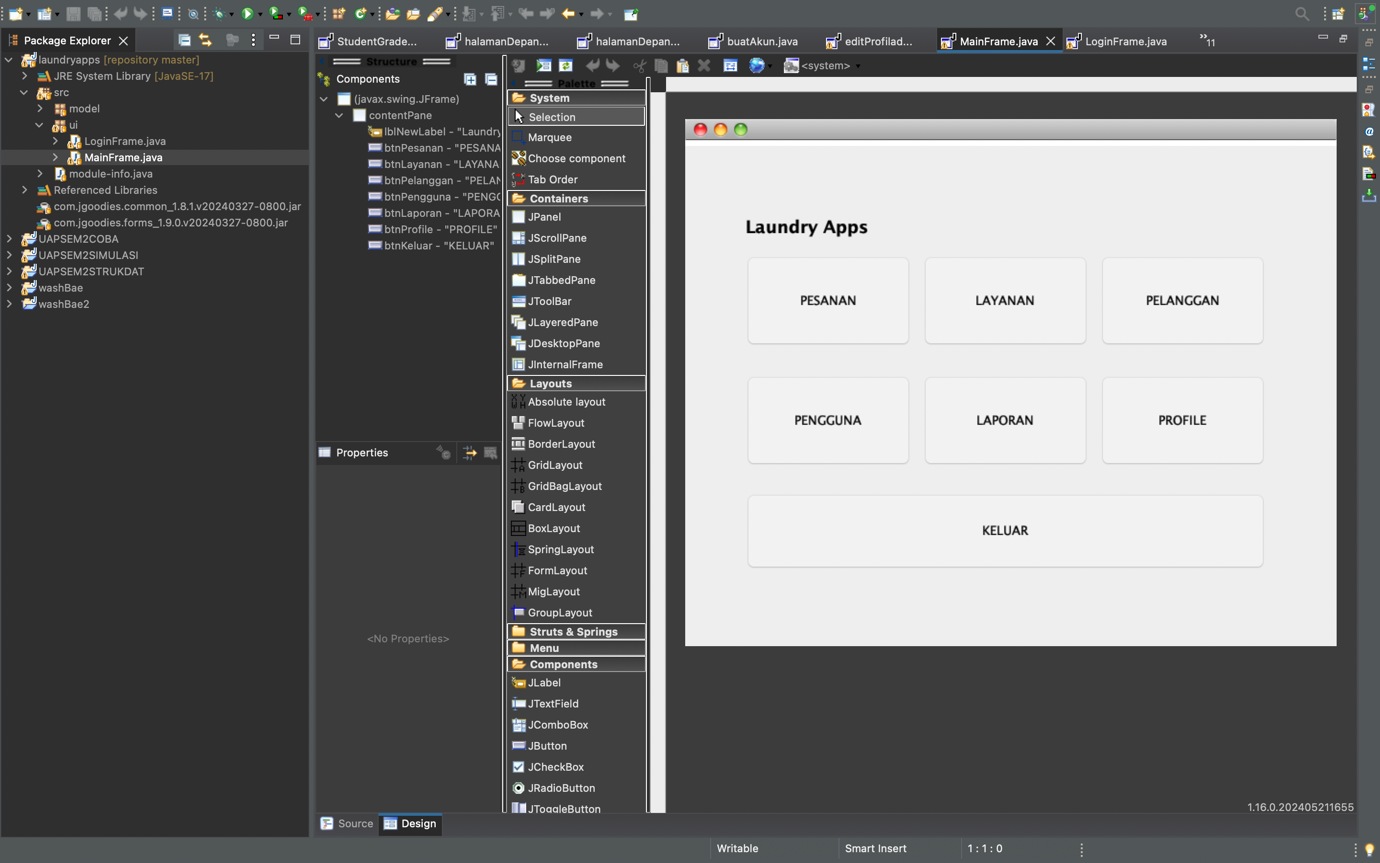
diambil dari txtUsername dan txtPassword, jika cocok maka akan tampil halaman utama.



1. buat sebuah JFrame baru pada package ui dengan nama MainFrame

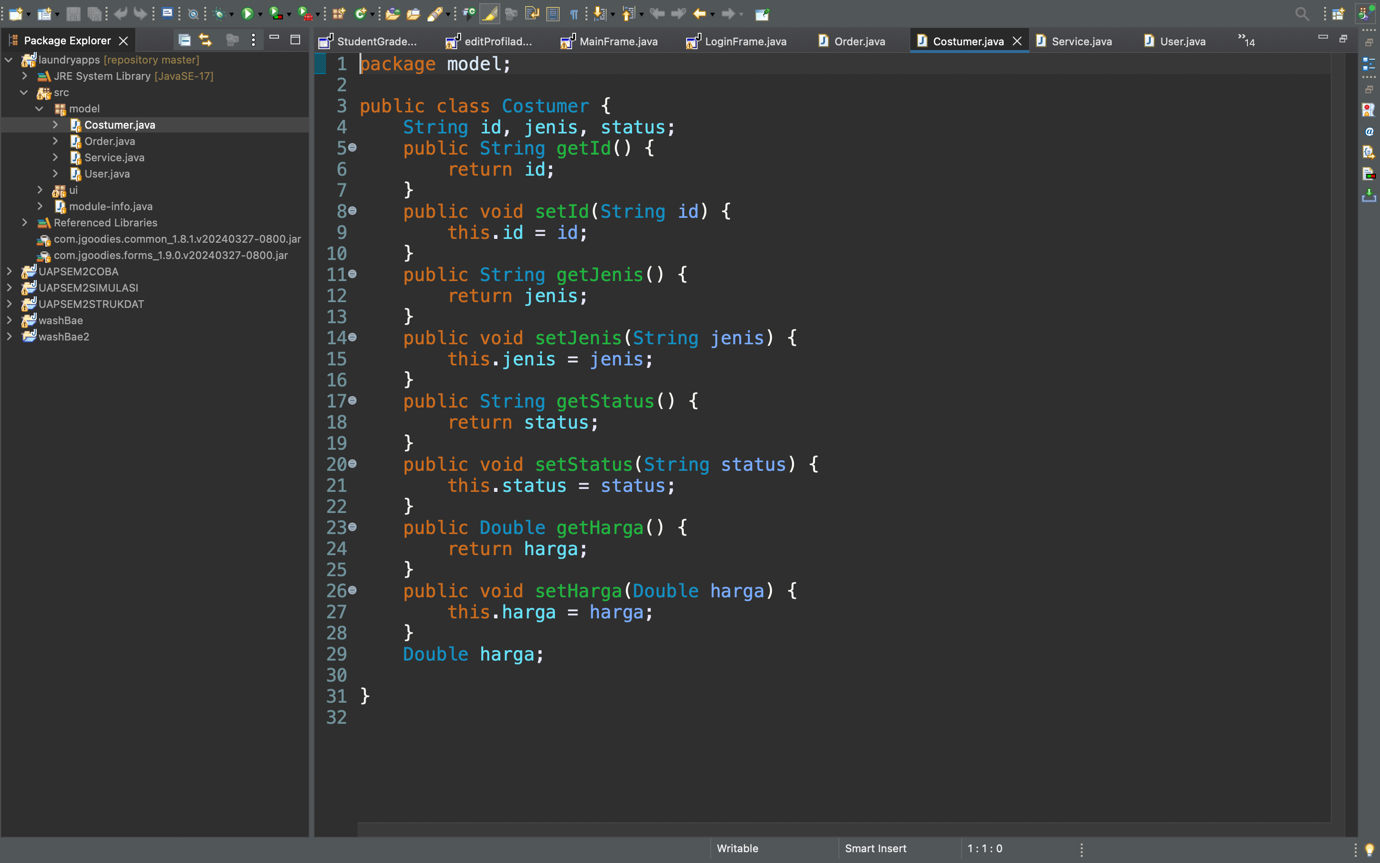


1. selanjutnya desain halaman utama seperti gambar berikut :



1. Buatlah class dengan nama Costumer dengan attribute id, nama, Alamat dan nomor hp

buatkan setter dan getter untuk menambahkan object Costumer



Kelas `Customer` pada kode di atas adalah sebuah model Java yang merepresentasikan entitas pelanggan. Kelas ini memiliki beberapa atribut atau properti dan metode untuk mengakses serta memodifikasi data pelanggan. Berikut adalah penjelasan singkat dari isi kode tersebut:

1. Atribut:

- id: Menyimpan ID pelanggan dalam bentuk String.

- jenis: Menyimpan jenis pelanggan dalam bentuk String.

- status: Menyimpan status pelanggan dalam bentuk String.

- harga: Menyimpan harga yang berkaitan dengan pelanggan dalam bentuk Double.

2. Metode Get dan Set:

- getId(): Mengembalikan nilai ID pelanggan.

- setId(String id): Mengatur nilai ID pelanggan.

- getJenis(): Mengembalikan nilai jenis pelanggan.

- setJenis(String jenis): Mengatur nilai jenis pelanggan.

- getStatus(): Mengembalikan nilai status pelanggan.

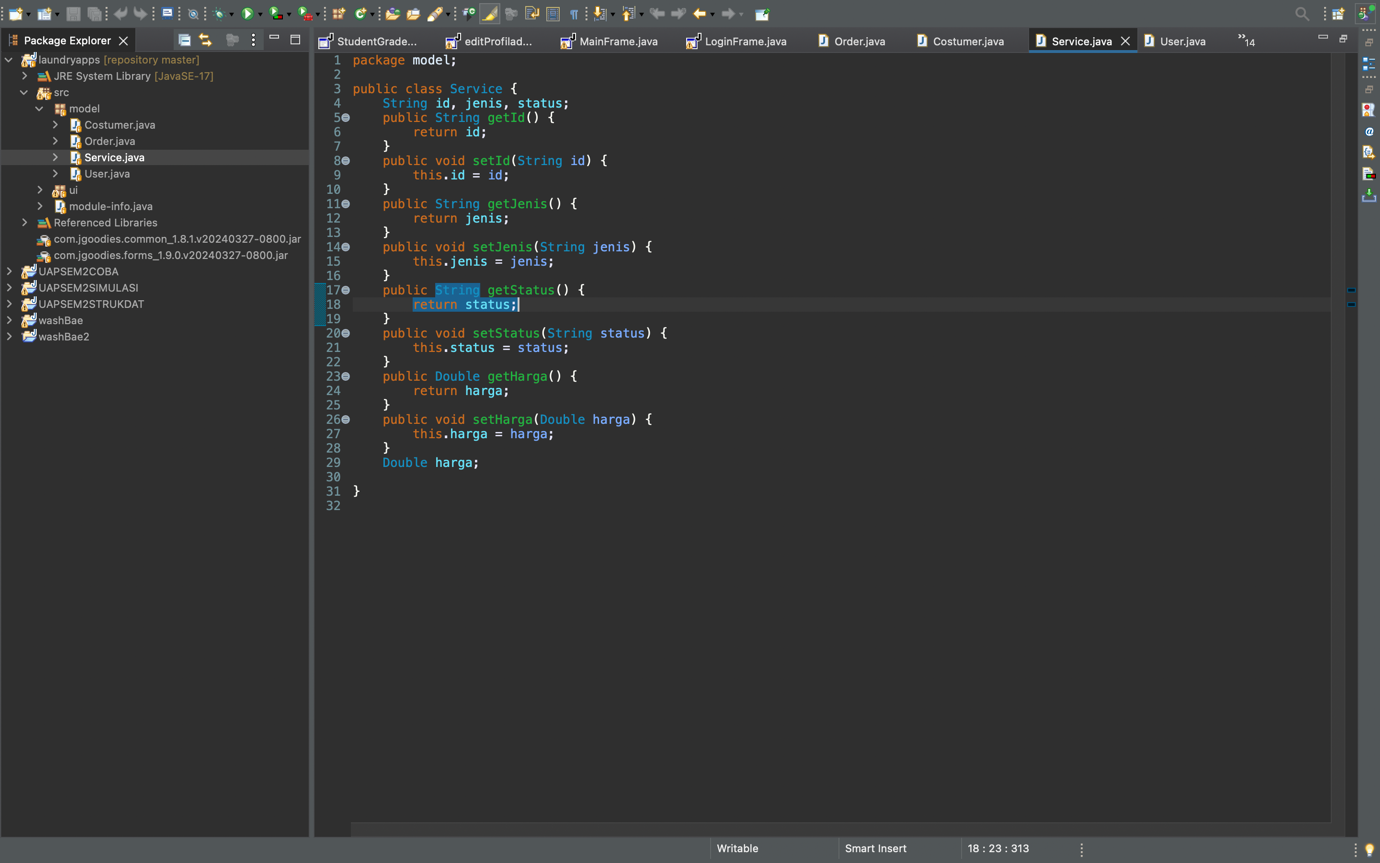
- setStatus(String status): Mengatur nilai status pelanggan.

- getHarga(): Mengembalikan nilai harga.

- setHarga(Double harga): Mengatur nilai harga.

Kelas ini menggunakan encapsulation di mana akses ke properti hanya bisa dilakukan melalui metode getter dan setter.

1. Buatlah class dengan nama Service dengan attribute id, jenis, harga dan status, buatkan setter dan getter untuk menambahkan object Service



Kelas `Service` pada kode di atas adalah sebuah model Java yang merepresentasikan entitas layanan. Kelas ini memiliki beberapa atribut atau properti dan metode untuk mengakses serta memodifikasi data layanan. Berikut adalah penjelasan singkat dari isi kode tersebut:

1. Atribut:

- id: Menyimpan ID layanan dalam bentuk String.

- jenis: Menyimpan jenis layanan dalam bentuk String.

- status: Menyimpan status layanan dalam bentuk String.

- harga: Menyimpan harga yang berkaitan dengan layanan dalam bentuk Double.

2. Metode Get dan Set:

- getId(): Mengembalikan nilai ID layanan.

- setId(String id): Mengatur nilai ID layanan.

- getJenis(): Mengembalikan nilai jenis layanan.

- setJenis(String jenis): Mengatur nilai jenis layanan.

- getStatus(): Mengembalikan nilai status layanan.

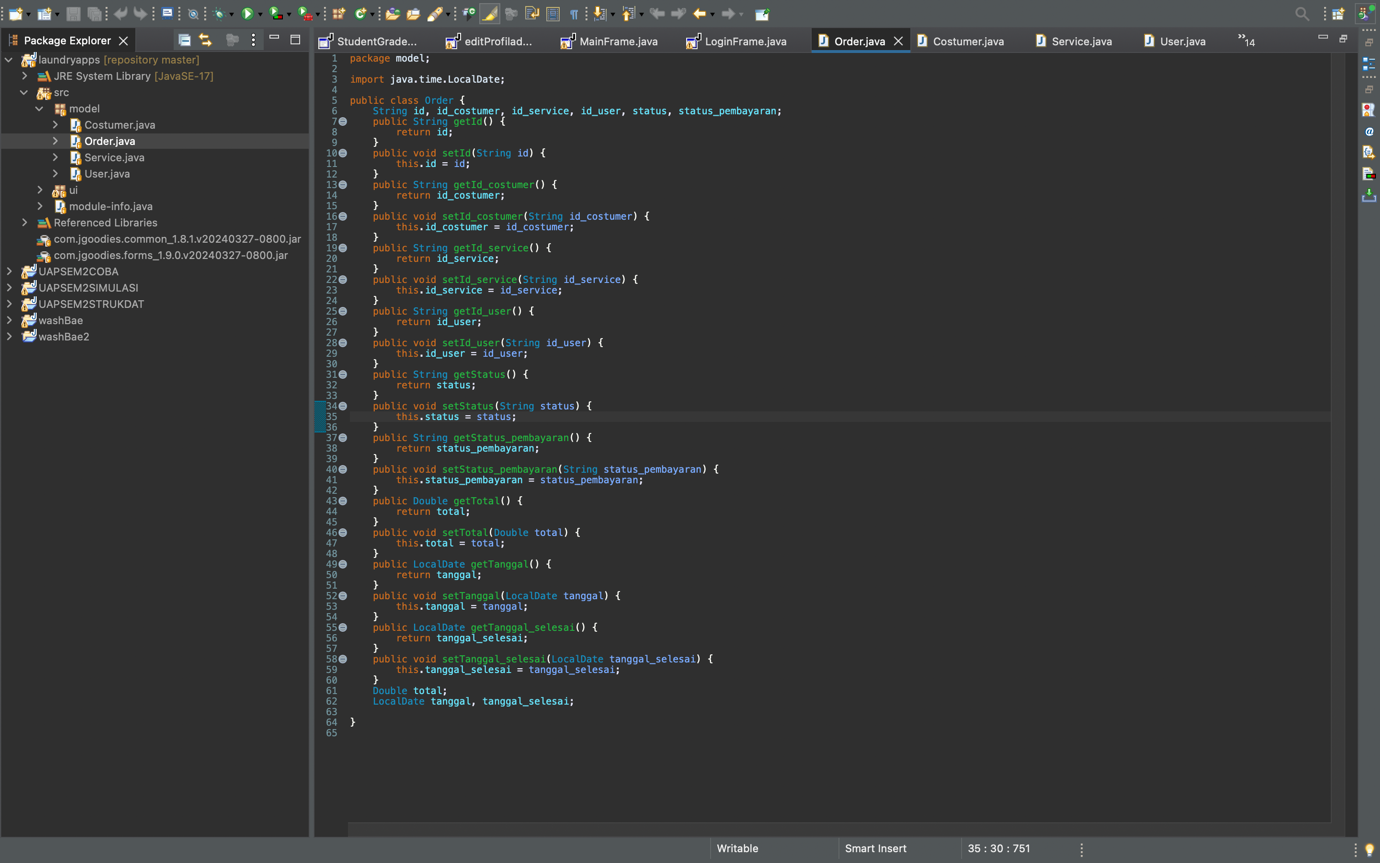
- setStatus(String status): Mengatur nilai status layanan.

- getHarga(): Mengembalikan nilai harga.

- setHarga(Double harga): Mengatur nilai harga.

Kelas ini menggunakan encapsulation di mana akses ke properti hanya bisa dilakukan melalui metode getter dan setter.

1. Buatlah class dengan nama Order dengan attribute id, id\_costumer, id\_service, id\_user, untuk menambahkananggal\_selesai, status, status\_pembayaran , buatkan setter dan getter untuk menambahkan object Order



Kelas `Order` pada kode di atas adalah sebuah model Java yang merepresentasikan entitas pesanan. Kelas ini memiliki beberapa atribut atau properti, serta metode untuk mengakses dan memodifikasi data terkait pesanan. Berikut adalah penjelasan singkat dari isi kode tersebut:

1. Atribut:

- id: Menyimpan ID pesanan dalam bentuk String.

- id\_costumer: Menyimpan ID pelanggan dalam bentuk String.

- id\_service: Menyimpan ID layanan yang dipesan dalam bentuk String.

- id\_user: Menyimpan ID pengguna yang memproses pesanan dalam bentuk String.

- status: Menyimpan status pesanan dalam bentuk String.

- status\_pembayaran: Menyimpan status pembayaran pesanan dalam bentuk String.

- total: Menyimpan total harga pesanan dalam bentuk Double.

- tanggal: Menyimpan tanggal pesanan dibuat dalam bentuk LocalDate.

- tanggal\_selesai: Menyimpan tanggal pesanan selesai dalam bentuk LocalDate.

2. Metode Get dan Set:

- getId(): Mengembalikan nilai ID pesanan.

- setId(String id): Mengatur nilai ID pesanan.

- getId\_costumer(): Mengembalikan nilai ID pelanggan.

- setId\_costumer(String id\_costumer): Mengatur nilai ID pelanggan.

- getId\_service(): Mengembalikan nilai ID layanan.

- setId\_service(String id\_service): Mengatur nilai ID layanan.

- getId\_user(): Mengembalikan nilai ID pengguna.

- setId\_user(String id\_user): Mengatur nilai ID pengguna.

- getStatus(): Mengembalikan status pesanan.

- setStatus(String status): Mengatur status pesanan.

- getStatus\_pembayaran(): Mengembalikan status pembayaran.

- setStatus\_pembayaran(String status\_pembayaran): Mengatur status pembayaran.

- getTotal(): Mengembalikan total harga pesanan.

- setTotal(Double total): Mengatur total harga pesanan.

- getTanggal(): Mengembalikan tanggal pesanan dibuat.

- setTanggal(LocalDate tanggal): Mengatur tanggal pesanan dibuat.

- getTanggal\_selesai(): Mengembalikan tanggal pesanan selesai.

- setTanggal\_selesai(LocalDate tanggal\_selesai): Mengatur tanggal pesanan selesai.

Kelas ini menggunakan prinsip encapsulation, di mana akses ke properti hanya bisa dilakukan melalui metode getter dan setter, memastikan kontrol penuh terhadap perubahan nilai properti tersebut.

1. **PENUTUP**

Praktikum ini telah memberikan saya pemahaman yang lebih mendalam mengenai konsep Pemrograman Berorientasi Objek (PBO) dalam pengembangan aplikasi berbasis Java. Dalam proses pembuatan aplikasi manajemen laundry sederhana, saya belajar menerapkan konsep-konsep penting seperti pembuatan kelas, objek, penggunaan constructor, dan penerapan method. Saya juga memperoleh pengalaman dalam membuat antarmuka pengguna menggunakan JFrame, serta memahami bagaimana menghubungkan antar komponen dalam aplikasi tersebut.

Melalui implementasi berbagai class seperti `User`, `Costumer`, `Service`, dan `Order`, saya semakin memahami pentingnya prinsip encapsulation serta bagaimana method getter dan setter digunakan untuk mengelola data secara efisien. Praktikum ini tidak hanya membantu saya memahami teori, tetapi juga memberikan kemampuan praktis dalam pengembangan aplikasi yang relevan dengan kebutuhan industri saat ini.