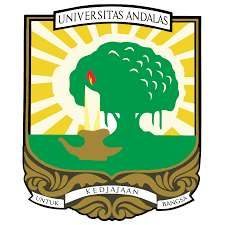
**LAPORAN PRAKTIKUM 1 PBO**

**CLASS, OBJECT, CONSTRUCTOR, METHOD**

****

**Disusun Oleh:**

**INDAH SYAHFITRI**

**2311532016**

**Dosen Pembimbing: Nurfiah, S. ST., M. Kom.**

**DEPARTEMEN INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS ANDALAS**

1. **PENDAHULUAN**

Class merupakan sekumpulan objek yang memiliki **karakteristik/sifat/behavior** dan **properti/atribut** yang sama, class juga bisa disebut sebagai template/blueprint dari objek yang akan dibuat. Misalkan ada class mahasiswa, yaitu nama-nama mahasiswa yang merupakan sebuah object. Sebuah class bukan sebuah entitas di dalam dunia nyata, melainkan sebagai blueprint untuk membuat suatu objek, class dalam java dapat berisi Data Member, Method, Constructor, Nested Class dan Interface.

Object merupakan representasi entitas yang ada pada dunia nyata, sebuah object terdiri dari:

• **State** merepresentasikan attribute dari sebuah object.

• **Behavior** merepresentasikan method dari sebuah object.

• **Identity** nama yang unik sebuah objek dan digunakan untuk interaksi dengan objek lain.

Method merupakan sebuah blok kode yang dapat dijalankan berulang kali, sehingga lebih terorganisir dan dapat digunakan kembali, contoh method yang disediakan oleh java adalah method main(), equals(), toString() dll. Aturan pembuatan method pada java yakni:

• Penamaan method **harus** menggunakan **verb** dan **dimulai dengan** **lowercase letter**.

• Jika nama method **lebih dari satu** kata maka **kata pertama harus verb** dan selanjutnya

adjective atau noun seperti sum(), area().

• jika nama method terdiri **lebih dari satu** kata, maka kata pertama adalah lowercase dan kata selanjutnya uppercase seperti areaOfCircle().

Method pada java terdiri dari beberapa jenis, berikut jenis-jenis method yang ada pada Java.

• **Predefined Method** (standard library method or built-in method) method yang sudah

disediakan oleh java seperti length(), equals(), compareTo(), print().

• **User-defined Method** yaitu method yang dibuat oleh user atau Programmer, method

dibuat sesuai dengan kebutuhan.

• **Static Method** yaitu method yang menggunakan static sebelum nama method, keutamaan method static yaitu dapat memanggil/menggunakan method tanpa harus membuat object terlebih dahulu.

• **Instance Method** merupakan method *non static* yaitu harus membuat object terlebih dahulu ketika akan menggunakannya. Instance method terdiri dari 2 jenis yaitu accessor method dan mutator method.

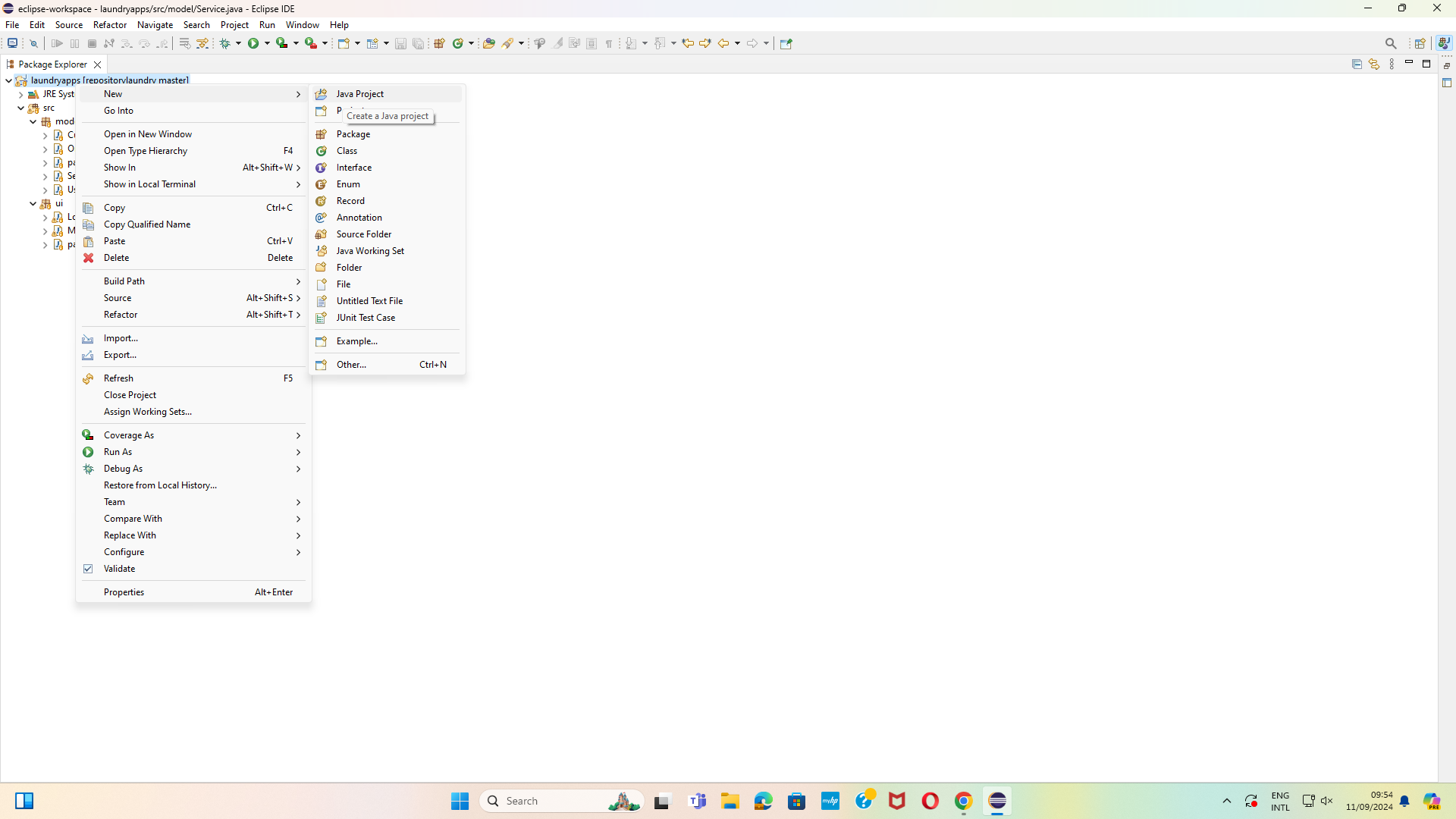
1. **Accessor Method** digunakan untuk membaca instance, menggunakan kata kunci get atau disebut getter.
2. Mutator Method digunakan untuk membaca dan mengubah nilai, menggunakan kata kunci set atau disebut juga dengan setter dan akan mengembalikan value yang bersifat private.

• **Abstract Method** yaitu method yang tidak memiliki body, dideklarasikan di dalam class abstract dengan menggunakan kata kunci abstract.

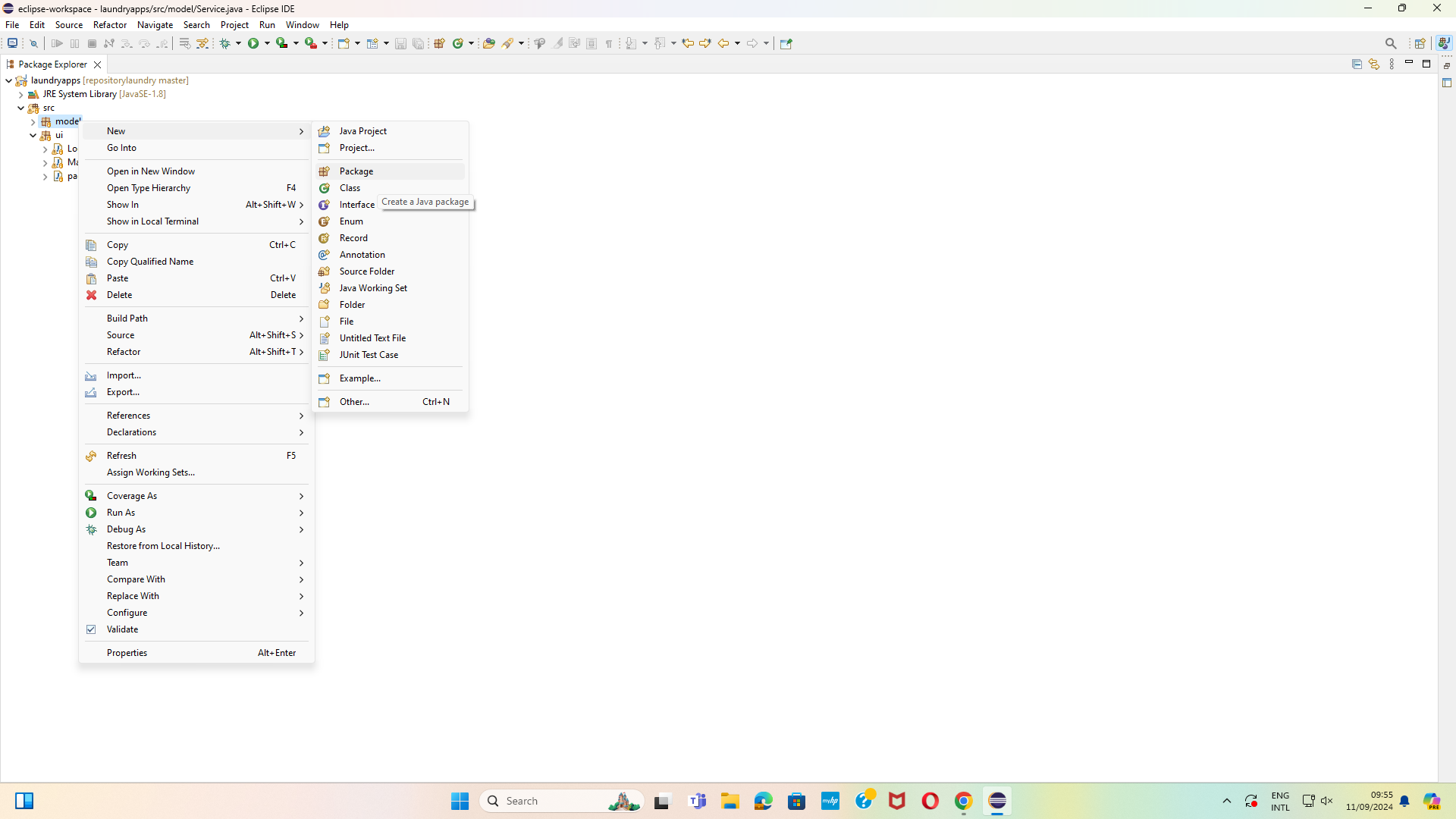
• **Factory Method** yaitu method yang digunakan untuk mengembalikan object ke class yang bersangkutan, semua method static disebut factory method contohnya NumberFormat obj = NumberFormat.getNumberInstance();.

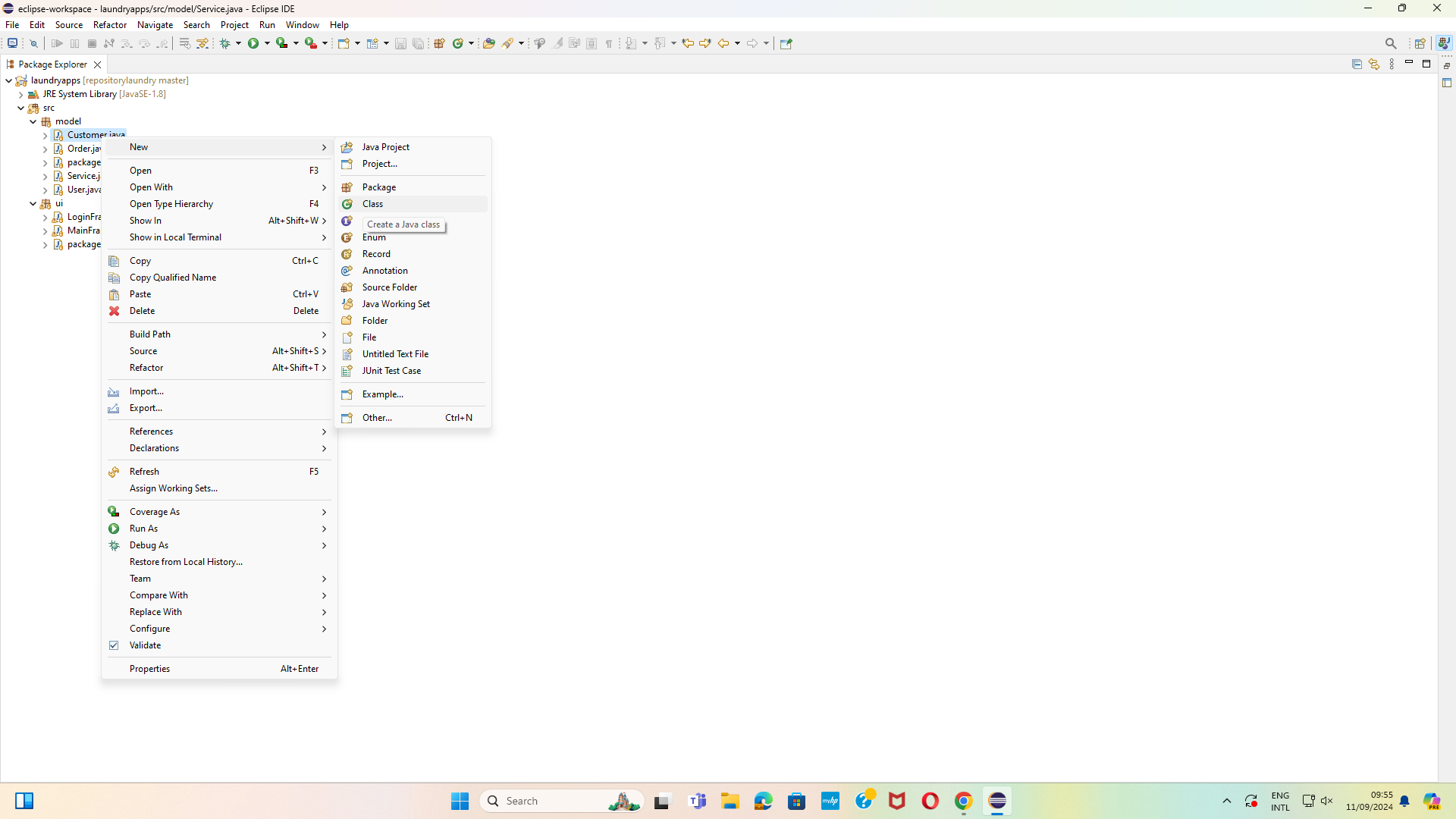
Constructor yaitu blok kode yang mirip dengan method, menggunakan kata kunci new ketika akan membuat sebuah object, nama constructor harus sama dengan nama class.

1. **TUJUAN PRAKTIKUM**
2. Tujuan praktikum ini yaitu mahasiswa mampu membuat class, object, encapsulation, constructor dan method menggunakan Bahasa java dan menerapkannya pada aplikasi manajemen laundry.
3. **LANGKAH-LANGKAH**
4. Buka aplikasi eclipse, lalu buat java project baru dengan nama laundryapps.

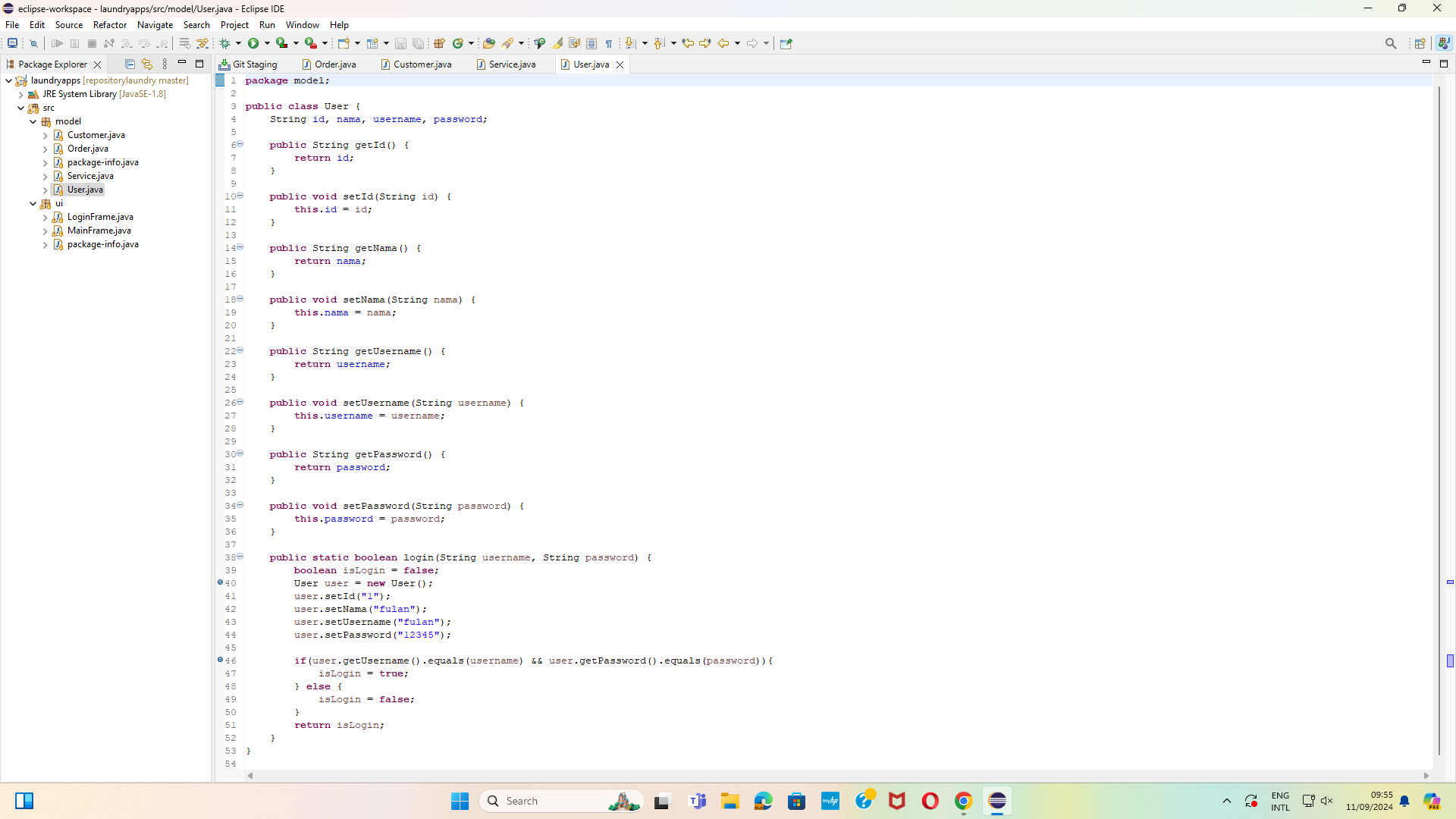


1. Setelahnya, buat dua packages yakni model dan ui. Package model berfungsi untuk menyimpan class-class yang diperlukan untuk pembuatan aplikasi laundry, sedangkan package ui digunakan untuk menyimpan tampilan atau antarmuka aplikasi laundry. Akan dibuat beberapa kelas di tiap package.

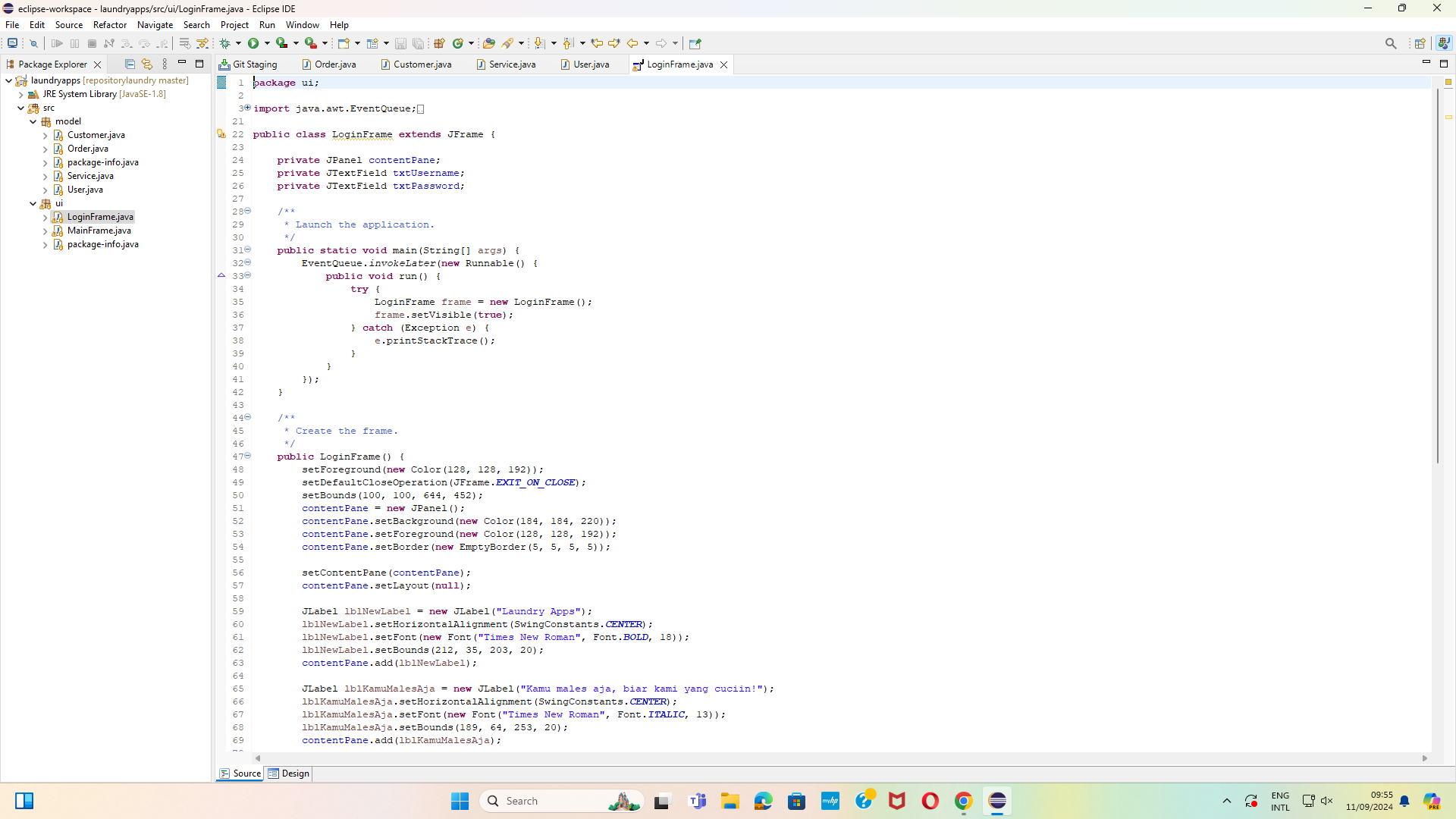




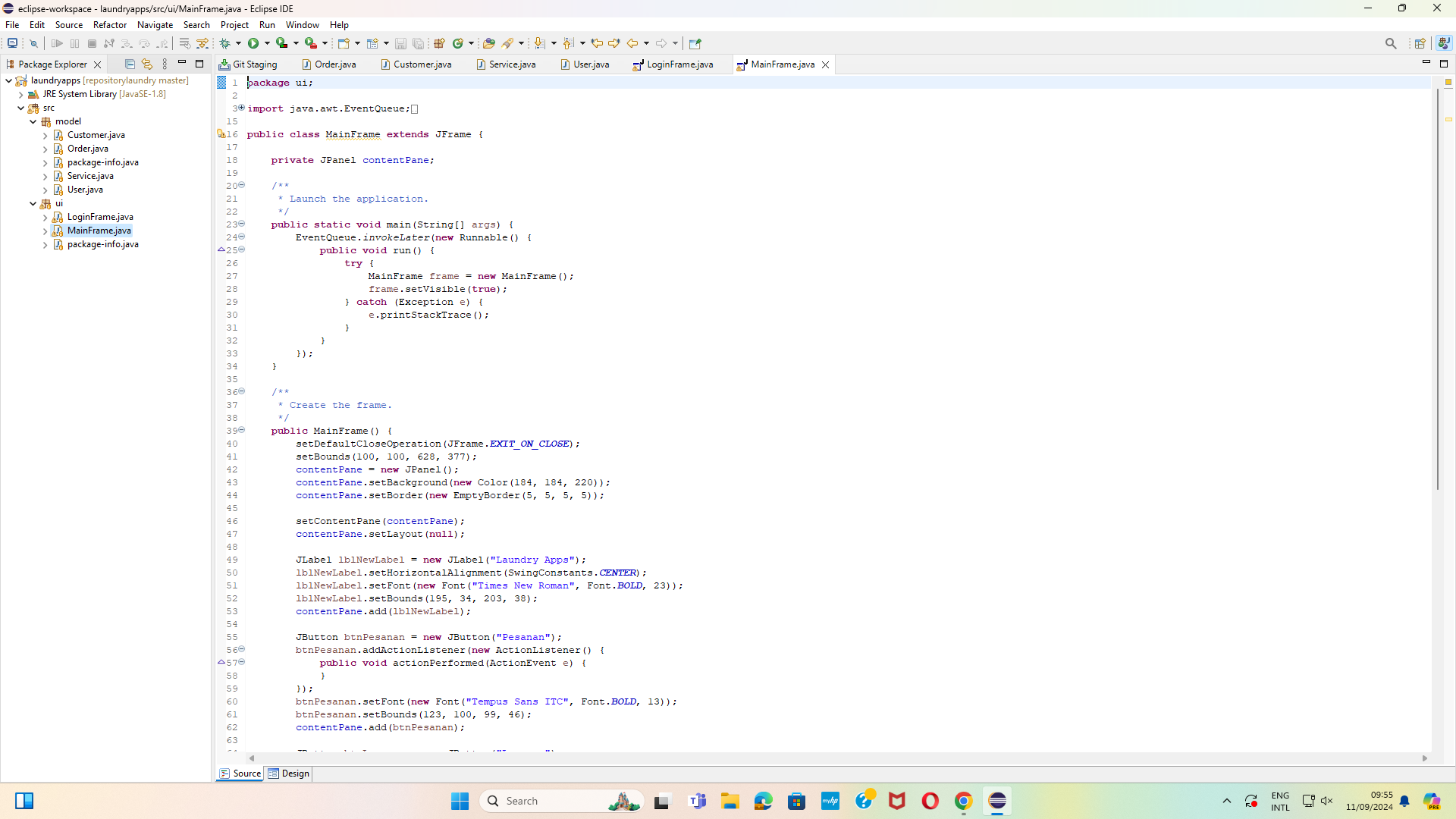
1. Pada package model, buat class baru dengan nama User. Buatlah attribute class user yaitu id, nama, username dan password. Setelah itu, buat setter dan getter dari attribute user, method ini berfungsi untuk memasukkan/mengubah dan menampilkan value dari suatu object. Lalu, buat Method login yang nantinya akan digunakan ketika pengguna akan login ke aplikasi, yakni menggunakan public static boolean. Method login ini merupakan method static dikarenakan menggunakan kata kunci static yang mana dapat diakses dari class lain tanpa harus membuat sebuah object terlebih dulu. Method login ini menggunakan tipe data Boolean yang mana akan mengembalikkan nilai Boolean. Ketika digunakan, kemudian memiliki 2 buah parameter, yaitu username dan password. Pertama, variable isLoggin diberikan nilai false, kemudian dibuatkan sebuah object user baru, selanjutnya akan dilakukan pencocokan data username dan password pada object dengan data username dan password yang dikirimkan dari class lain, jika cocok maka isLoggin akan diubah menjadi true dan terakhir method akan mengembalikkan nilai isLoggin.



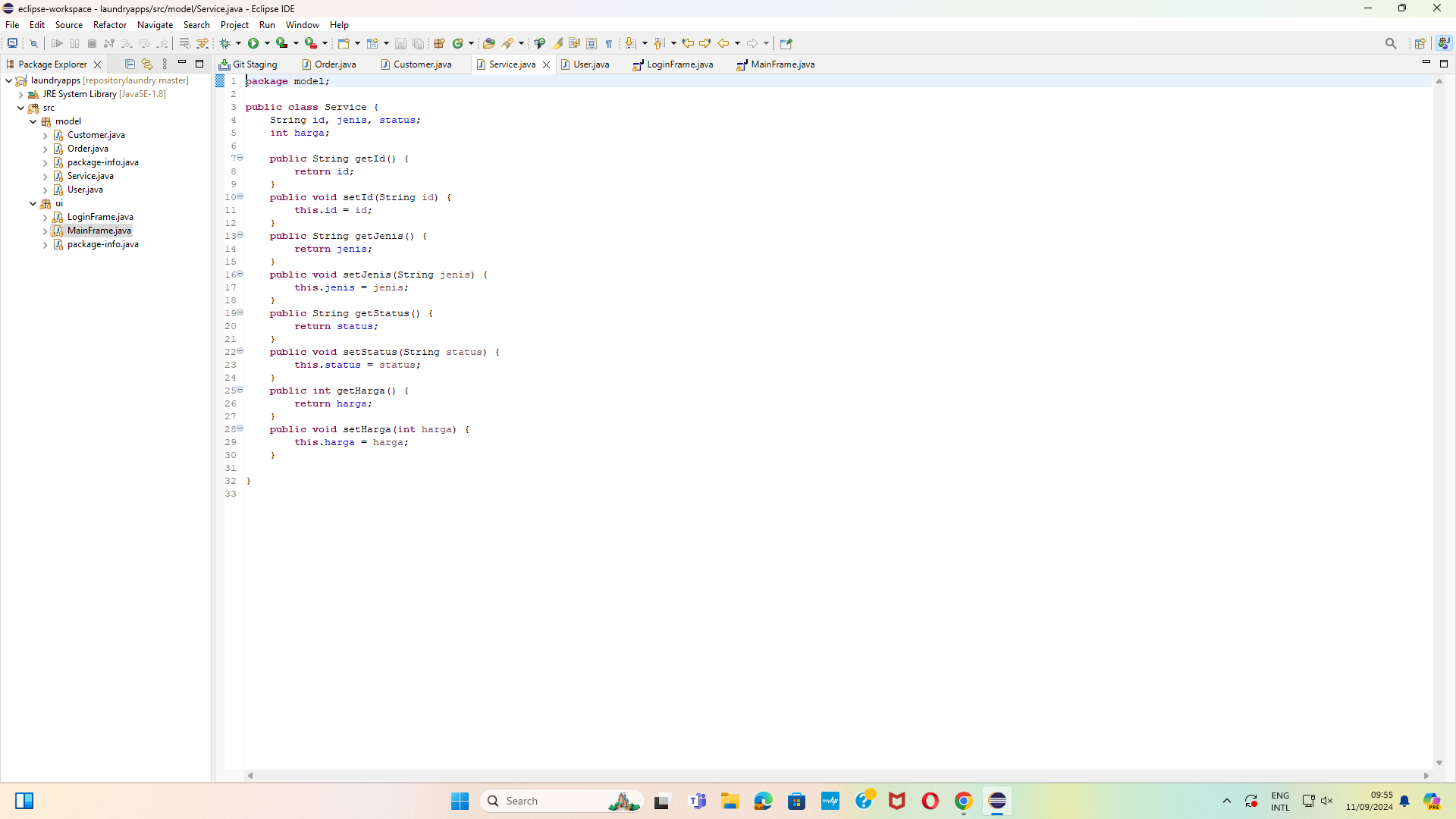
1. Lanjut, buat jFrame baru pada package ui dengan nama LoginFrame. Klik menu design, maka akan muncul tampilan UI jframe yang bisa didesain menurut preferensi masing-masing. Setelahnya, klik kanan pada button login, pilih add event handler → action → actionPerformed. Kemudian panggil method login pada class User dengan mengirimkan parameter yang diambil dari txtUsername dan txtPassword, jika cocok maka akan tampil halaman utama.



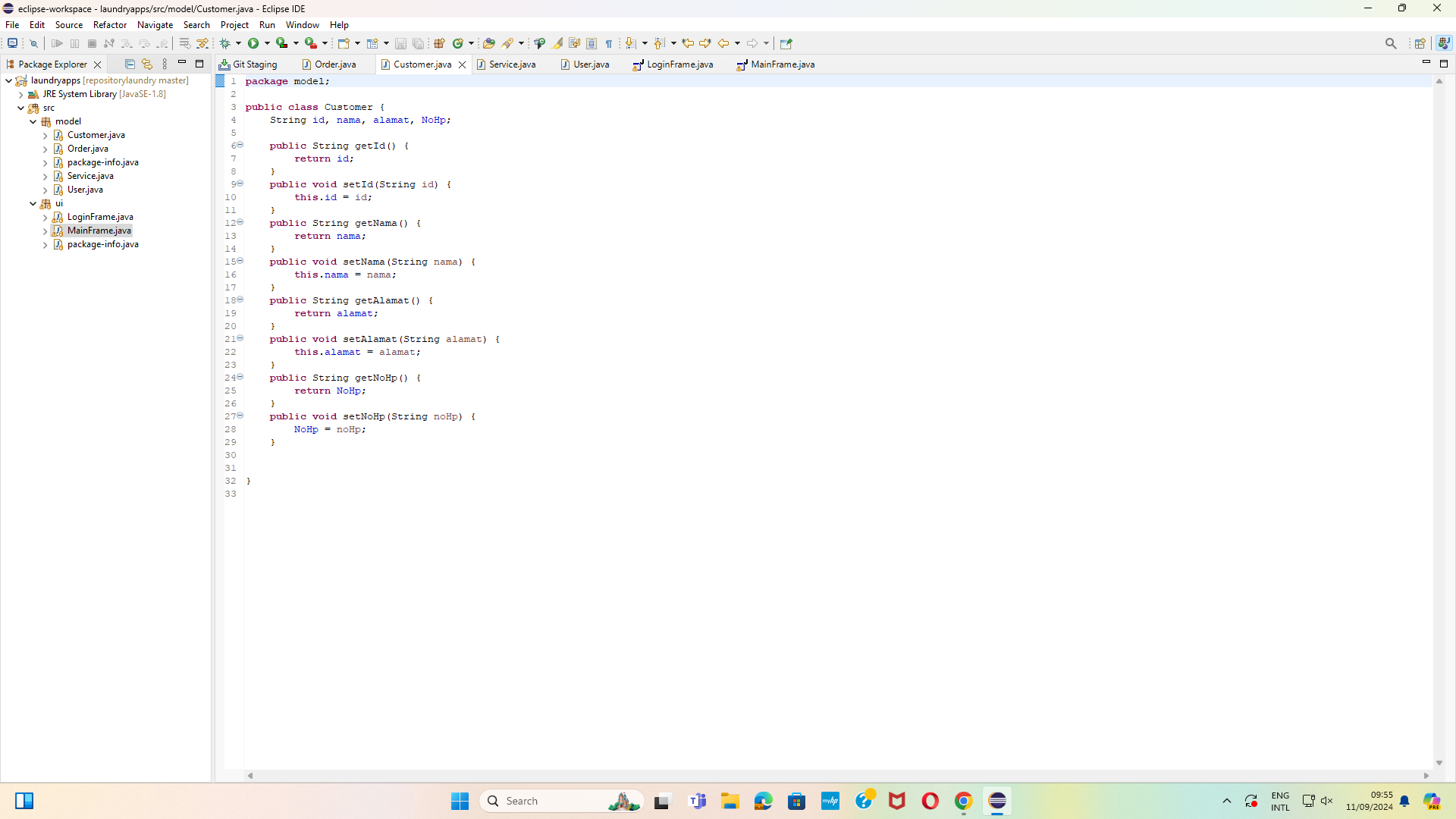
1. Kemudian, buat kelas jFrame baru dengan namanya MainFrame. Setelah user login, maka akan diarahkan langsung menuju menu utama dan menggunakan fitur-fitur yang tersedia. Fitur-fitur yang tersedia yakni Pesanan, Layanan, Pelanggan, Pengguna, Laporan, Profil, dan menu untuk log out. Desain jFramenya terlebih dahulu. Untuk penerapan fitur-fitur, kita bisa membuat kelas baru untuk setiap fitur di package model.



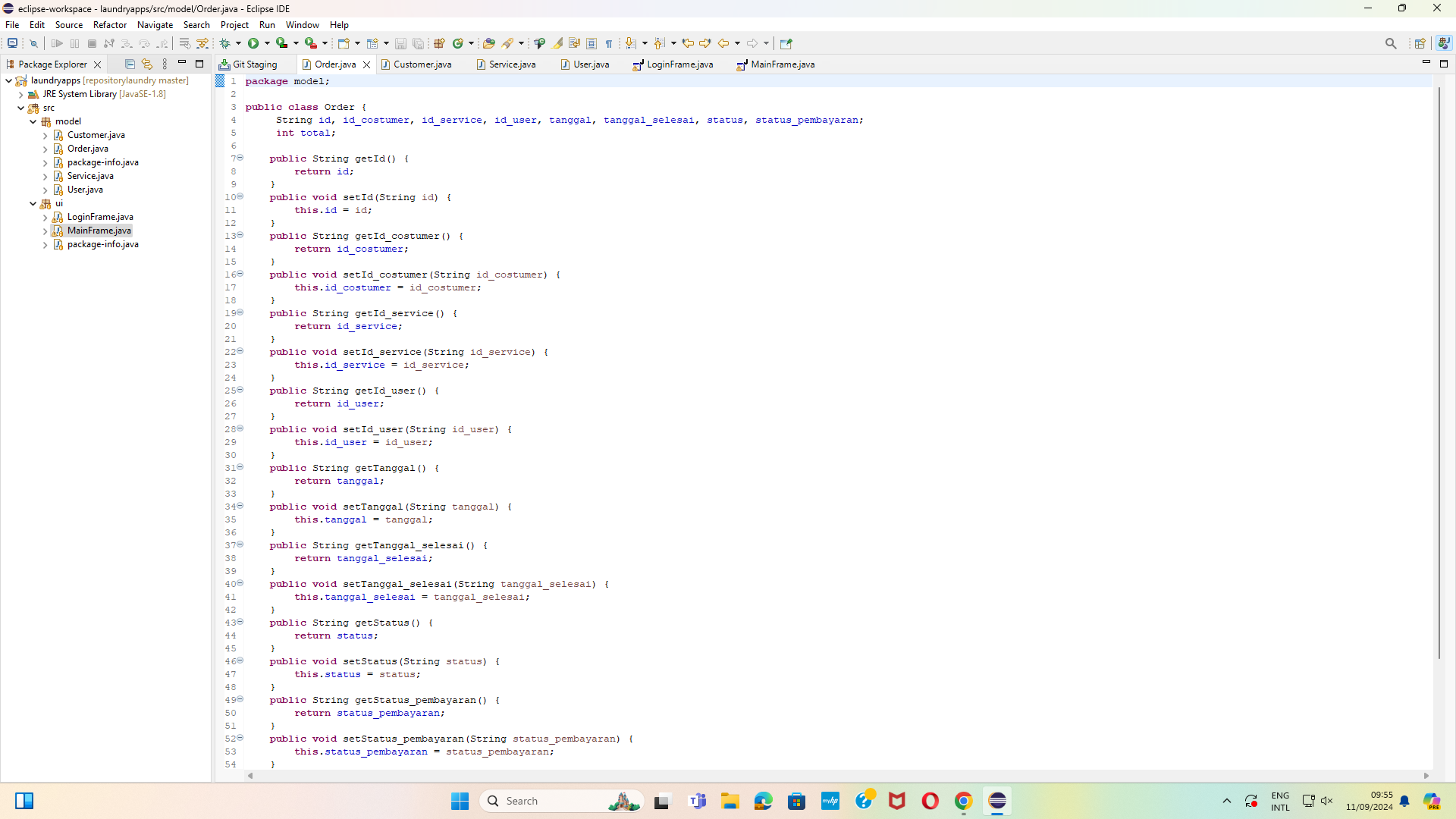
1. Untuk menjalankan fitur Layanan, kita bisa buat kelas baru di package model dengan nama Service. Kemudian, buat attribute class service yaitu id, jenis, dan status menggunakan tipe data string dan harga dengan tipe data integer. Setelah itu, buat setter dan getter dari attribute service untuk memasukkan/mengubah dan menampilkan value dari suatu object.



1. Untuk menjalankan fitur Pelanggan, kita bisa buat kelas baru di package model dengan nama Customer. Kemudian, buat attribute class customer yaitu id, nama, alamat, dan nomor telepon menggunakan tipe data string. Setelah itu, buat setter dan getter dari attribute service untuk memasukkan/mengubah dan menampilkan value dari suatu object.



1. Untuk menjalankan fitur Pesanan, kita bisa buat kelas baru di package model dengan nama Order. Kemudian, buat attribute class order yaitu id, id customer, id service, id user, tanggal, tanggal selesai, status, dan status pembayaran menggunakan tipe data string dan total dengan tipe data integer. Setelah itu, buat setter dan getter dari attribute order untuk memasukkan/mengubah dan menampilkan value dari suatu object.



1. **SIMPULAN**

Praktikum ini telah mengimplementasikan konsep dasar dari pemrograman berorientasi objek menggunakan bahasa pemrograman Java. Kelas pertama adalah kelas User sebagai pengguna dalam aplikasi manajemen laundry, dengan atribut seperti id, nama, username, dan password. Program ini juga memiliki metode getter dan setter untuk mengakses dan mengubah nilai atribut, serta metode login yang menggunakan parameter username dan password untuk memverifikasi identitas pengguna.

Untuk antarmuka pengguna (UI) telah diimplementasikan pada kelas LoginFrame dan MainFrame. Program ini menghubungkan tampilan visual dengan ptogram yang telah ada pada kelas User. Pengguna dapat login dengan memasukkan username dan password, yang akan dicek melalui metode login pada kelas User. Setelah login berhasil, pengguna akan diarahkan ke halaman utama yang berisi fitur-fitur aplikasi seperti layanan, pelanggan, dan pesanan. Setiap fitur diwakili oleh kelas-kelas terpisah seperti Service, Customer, dan Order. Kelas-kelas ini memanfaatkan konsep getter dan setter.