LAPORAN PRAKTIKUM PEKAN 1 PEMOGRAMAN BERORIENTASI OBJEK



NAMA : FIKHRI HANIF

NIM : 2311533007

DOSEN PENGAMPU :

Nurfiah, S.ST, M.Kom.

DEPARTEMEN INFORMATIKA FAKULTAS TEKNOLOGI INFORMASI UNIVERSITAS ANDALAS

PADANG

1. Tujuan
2. Membuat class, object, encapsulation, contstruktor dan method untuk aplikasi laundry
3. seperti class User, Costumer, Service dan Order
4. Membuat desain antarmuka aplikasi Laundry Login dan Halaman Utama
5. Mampu menggunakan method pada Jframe/Tampilan aplikasi
6. Mampu membuat fungsi berpindah dari tampilan login ke halaman utama
7. Kajian teori
8. Class dalam Pemrograman Java

Class adalah sebuah kerangka atau blueprint yang digunakan untuk membuat objek dalam pemrograman berorientasi objek (OOP). Class mendefinisikan karakteristik, sifat (behavior), dan atribut (properties) dari objek yang akan dibuat. Sebagai contoh, class "Mahasiswa" dapat digunakan sebagai template untuk menciptakan objek seperti mahasiswa bernama Fulan.

Class pada dasarnya tidak mewakili entitas yang nyata di dunia nyata, melainkan sebagai cetak biru yang memungkinkan programmer membuat objek berdasarkan definisi yang ada dalam class tersebut. Dalam bahasa Java, class dapat berisi beberapa elemen seperti:

* **Data Member**: Variabel yang menyimpan data atau atribut dari class.
* **Method**: Fungsi atau prosedur yang dimiliki oleh class.
* **Constructor**: Blok kode yang digunakan untuk menginisialisasi objek baru.
* **Nested Class**: Class yang dideklarasikan di dalam class lain.
* **Interface**: Definisi kontrak yang menentukan metode yang harus diimplementasikan oleh class lain.

1. Object dalam Pemrograman Java

Object merupakan representasi nyata dari entitas yang didefinisikan oleh sebuah class. Jika class adalah blueprint, maka object adalah hasil aktual yang diciptakan dari blueprint tersebut. Object memiliki tiga karakteristik utama:

* **State**: Merepresentasikan atribut dari object. Misalnya, nama dan umur dari seorang mahasiswa.
* **Behavior**: Merepresentasikan metode yang dapat dilakukan oleh object. Misalnya, method untuk menampilkan data mahasiswa.
* **Identity**: Setiap object memiliki identitas unik yang membedakannya dari object lain. Identitas ini juga digunakan untuk berinteraksi antar objek.

1. Method dalam Pemrograman Java

Method adalah blok kode yang didefinisikan dalam class yang dapat dieksekusi berulang kali. Fungsinya adalah untuk mengatur logika atau tindakan yang dapat dilakukan oleh objek. Dalam Java, ada beberapa aturan dalam penamaan dan pembuatan method:

* Nama method harus menggunakan kata kerja (verb) dan dimulai dengan huruf kecil.
* Jika terdiri dari lebih dari satu kata, kata pertama harus kata kerja dan selanjutnya menggunakan kata benda atau sifat.
* Jika nama method terdiri dari dua kata atau lebih, huruf pertama kata kedua ditulis dengan huruf kapital (camelCase), seperti areaOfCircle().

Jenis-jenis method yang ada dalam Java meliputi:

* **Predefined Method**: Method yang sudah disediakan oleh Java dan dapat langsung digunakan, seperti length(), equals(), print(), dll.
* **User-defined Method**: Method yang didefinisikan oleh programmer sesuai kebutuhan.
* **Static Method**: Method yang menggunakan keyword static, sehingga dapat dipanggil tanpa harus membuat objek dari class.
* **Instance Method**: Method yang memerlukan pembuatan objek sebelum digunakan. Terdiri dari:
  + **Accessor Method (getter)**: Digunakan untuk mendapatkan nilai dari atribut object.
  + **Mutator Method (setter)**: Digunakan untuk mengubah nilai atribut object.
* **Abstract Method**: Method yang tidak memiliki implementasi atau body, dan hanya dideklarasikan dalam class abstract.
* **Factory Method**: Method yang digunakan untuk mengembalikan object ke class yang bersangkutan, biasanya menggunakan keyword static.

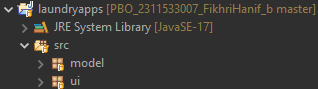
1. Constructor dalam Pemrograman Java

Constructor adalah blok kode khusus dalam class yang digunakan untuk menginisialisasi objek baru. Nama constructor harus sama dengan nama class, dan penggunaannya dilakukan dengan keyword new. Constructor berperan penting dalam memastikan bahwa setiap objek baru yang dibuat memiliki keadaan awal yang sesuai dengan definisi di dalam class.

1. Langkah Kerja

**Membuat Class User**

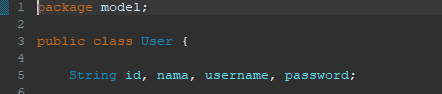
* 1. Buat project baru pada eclipse dengna nama **laundryapps**
  2. Tambahkan 2 buah package pada directory src dengan nama **model** dan **ui**
     + **model** sebagai tempat class-class
     + **ui** sebagai desainnya



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



* 1. Buat attribute class user yaitu id, nama, username dan password

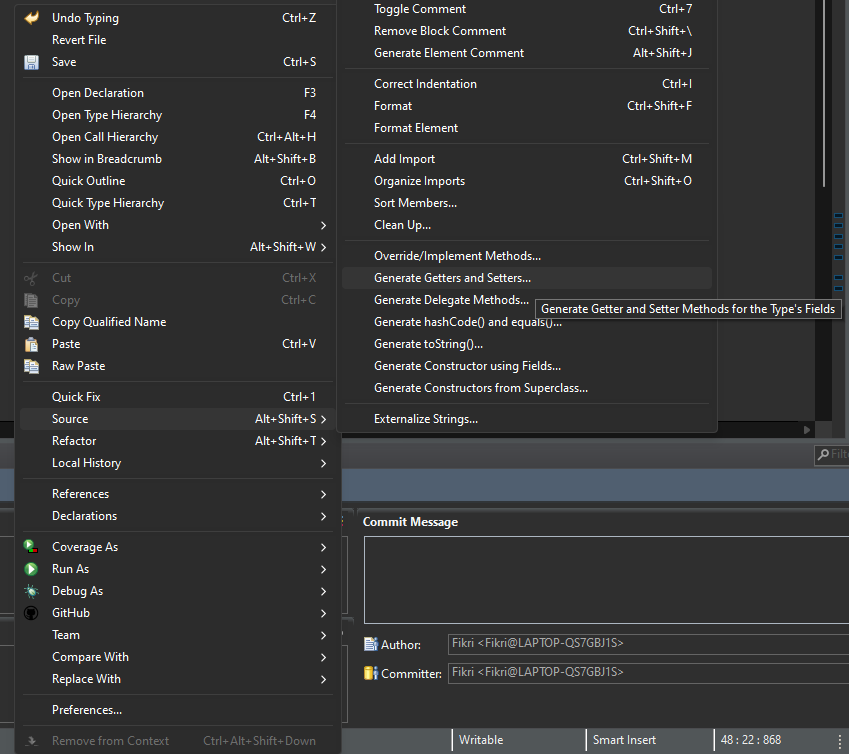


* 1. Buat setter dan getter attribute user, method ini berfungsi untuk memasukkan/mengubah dan menampilkan value dari suatu object.

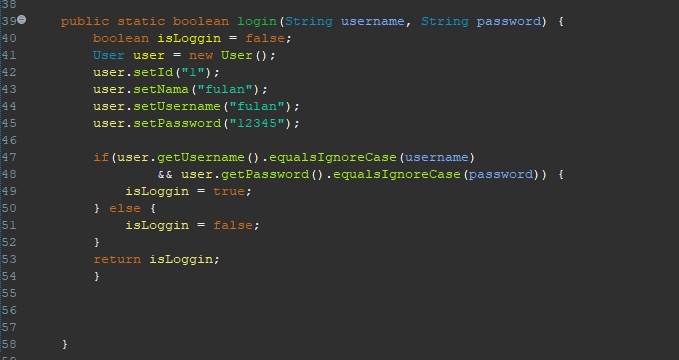


Cara cepat membuat setter dan getter yaitu :

* + - 1. Klik kanan pada bagian panel yang sudah kita deklarasikan attribute nya
      2. Kemudian pilih “Source”
      3. Pilih “Generate Getters and Setters….”



* 1. Kemudian buat Method login yang nantinya akan digunakan Ketika pengguna akan login ke aplikasi.

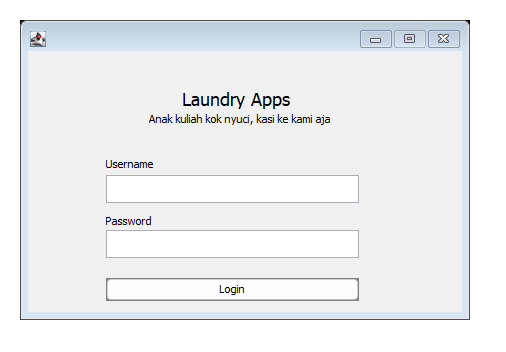


**Membuat Tampilan Login Menggunakan JFrame**

1. Buat jFrame baru pada package ui dengan nama LoginFrame



1. Kemudian desain tampilan login



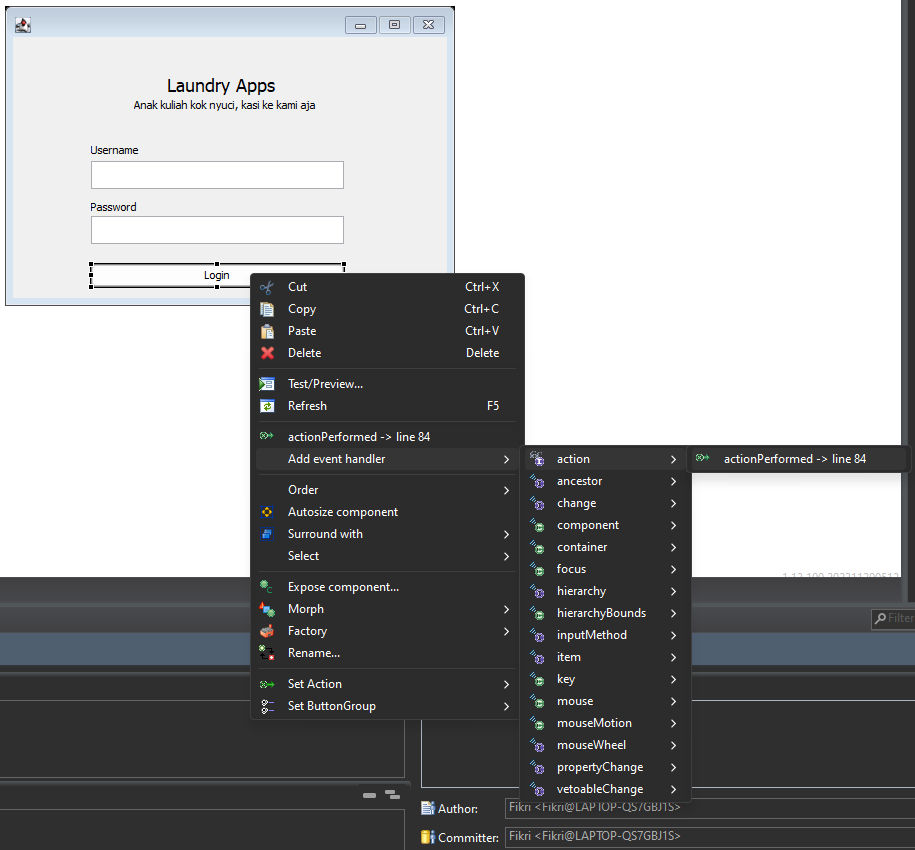
Ubah id JTextField username, password dan JButton

JTextField 🡪 txtUsername

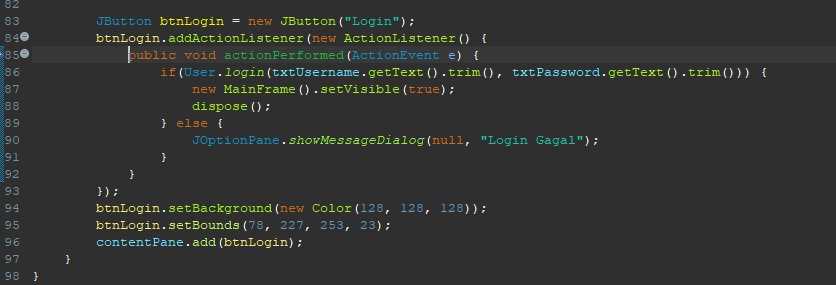
JTextField 🡪 txtPassword

JButton 🡪 btnLogin

1. selanjutnya klik kanan pada button login, pilih add event handler → action → actionPerformed

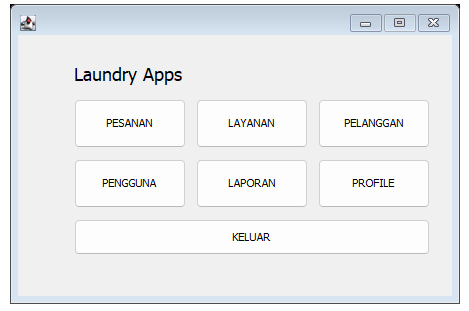


Kemudian panggil method login pada class User dengan mengirimkan parameter yang diambil dari txtUsername dan txtPassword, jika cocok maka akan tampil halaman utama.



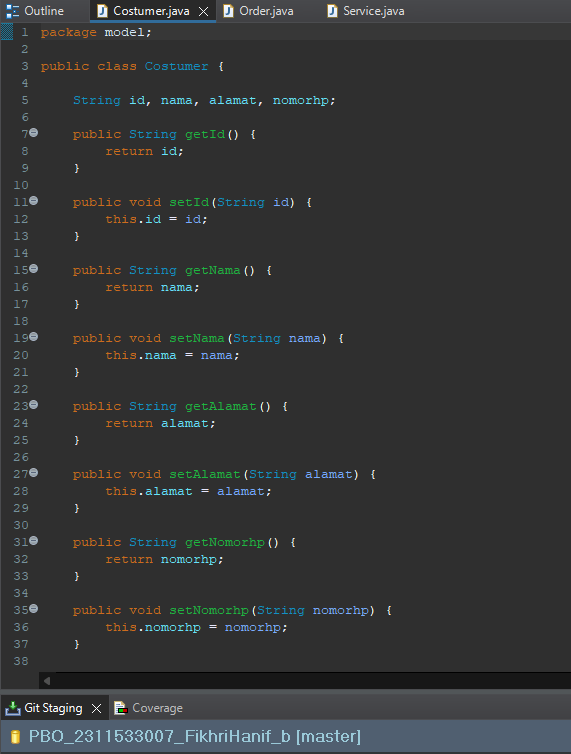
**Membuat Halaman Utama Menggunakan JFrame**

buat sebuah JFrame baru pada package ui dengan nama MainFrame, selanjutnya desain halaman utama

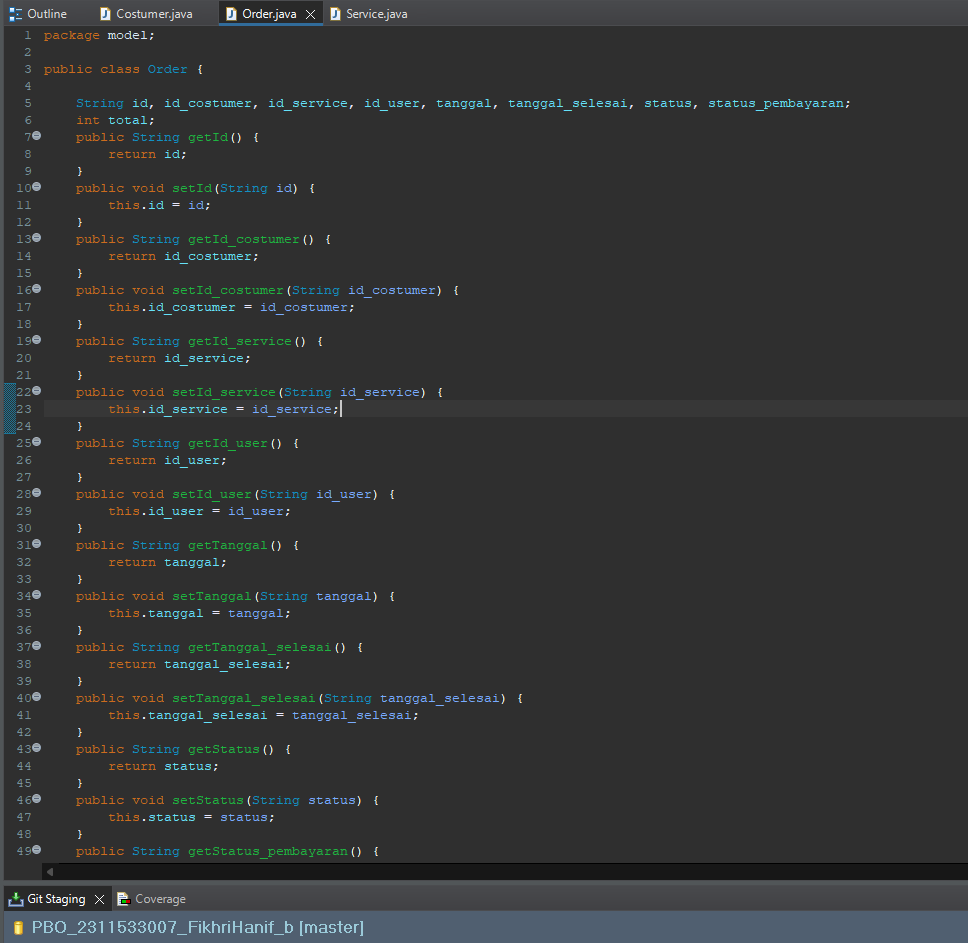


Tugas Membuat Getter Setter untuk class :

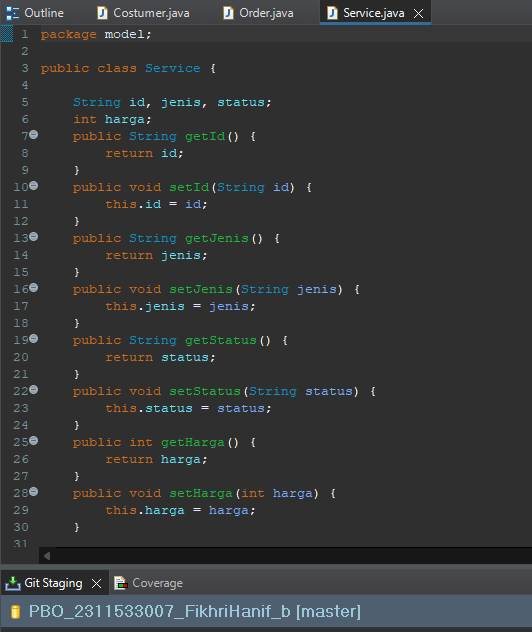
* 1. Customer.java



* 1. Order.java



* 1. Service.java



1. Kesimpulan

Praktikum ini bertujuan agar mahasiswa mampu memahami konsep dasar Pemrograman Berorientasi Objek (PBO) seperti **class**, **object**, **encapsulation**, **constructor**, dan **method**, serta mampu menerapkannya dalam pengembangan aplikasi sederhana, khususnya dalam kasus manajemen laundry.

Melalui praktikum ini, mahasiswa diajarkan cara membuat berbagai class seperti **User**, **Customer**, **Service**, dan **Order**, serta merancang antarmuka aplikasi yang meliputi login dan halaman utama menggunakan Java Swing. Salah satu poin penting yang dipelajari adalah cara berpindah antar tampilan aplikasi menggunakan metode yang diimplementasikan pada Jframe.

Selain itu, praktikum ini memperkenalkan konsep penting dalam Java seperti **getter** dan **setter**, **method login** statis, serta desain aplikasi berbasis event-driven programming. Sebagai latihan, mahasiswa diharuskan membuat class tambahan dengan atribut spesifik dan mengimplementasikan metode untuk menangani data terkait pelanggan, layanan, dan pesanan dalam aplikasi manajemen laundry.

Dengan demikian, praktikum ini memberikan pemahaman yang komprehensif mengenai bagaimana prinsip-prinsip PBO diterapkan dalam pengembangan aplikasi berbasis Java​