

Laporan praktikum Pekan 1
Pemrograman Berorientasi Objek
Class, Object, Constructor dan Method



Oleh:

Nama: Fadil Insanus Siddik

NIM: 2411532013

Praktikum PBO A

Dosen Pengampu:

Nurfiah S.ST.,M.Kom.

Program Studi Informatika

Fakultas Teknologi Informasi

Universitas Andalas

Padang, 2025

I. PENDAHULUAN

1.1 Tujuan Praktikum

Tujuan praktikum ini yaitu kita mampu membuat class, object, encapsulation, constructor dan method menggunakan Bahasa Java dan menerapkannya pada aplikasi manajemen laundry.

Adapun pada praktikum ini, kita akan mempelajari beberapa point yaitu:

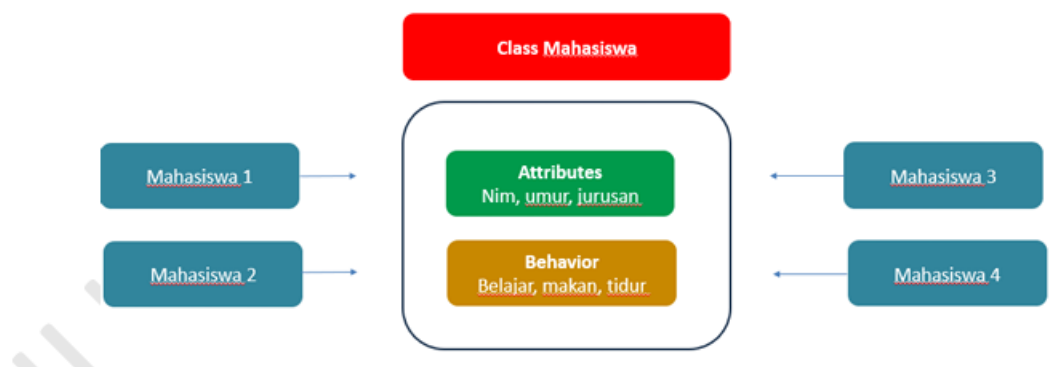
- Membuat class, object, encapsulation, constructor dan method untuk aplikasi laundry seperti class User, Customer, Service dan Order
- Membuat desain antarmuka aplikasi Laundry Login dan Halaman Utama
- Mampu menggunakan method pada JFrame/Tampilan aplikasi
- Mampu membuat fungsi berpindah dari tampilan login ke halaman utama

1.2 Alat yang diperlukan

- Computer / laptop yang telah terinstall JDK dan Eclipse

1.3 Teori

Class merupakan sekumpulan objek yang memiliki karakteristik/sifat/behavior dan properties/atribut yang sama, class juga bisa disebut sebagai template/blueprint dari objek yang akan dibuat. Misalkan misalkan ada class mahasiswa, yaitu nama-nama mahasiswa misalkan fulan merupakan sebuah object.



Sebuah class bukan sebuah entitas didalam dunia nyata melainkan sebagai blueprint untuk membuat suatu objek, class didalam java dapat berisi Data Member, Method, Constructor, Nested Class dan Interface.

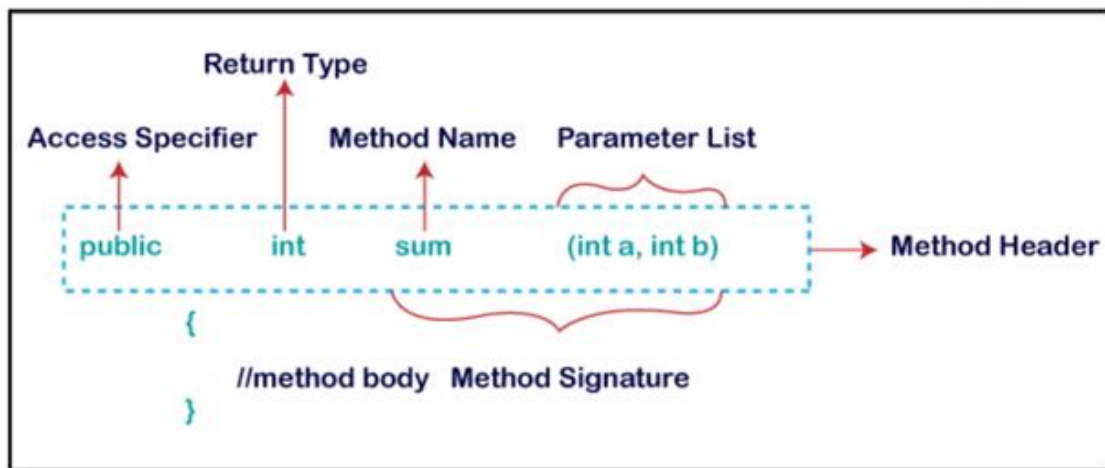
Object merupakan representasi entitis yang ada pada dunia nyata, sebuah object terdiri dari:

- 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. Pola sebuah method dapat dilihat seperti gambar berikut:



Aturan pembuatan method pada java:

- Penamaan method harus menggunakan **verb** dan dimulai dengan **lowercase** letter
- Jika nama method lebih dari satu kata maka kata pertama harus **verb** dan selanjutnya **adjectiva** atau **noun** seperti sum(), area()
- Jika nama method terdiri lebih dari 1 kata, maka kata pertama **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.
 - Accessor Method digunakan untuk membaca instance, menggunakan kata kunci **get** atau disebut **getter**.

```
public String getNim() {
    return nim;
}
```

- Mutator method digunakan untuk membaca dan mengubah nilai, menggunakan kata kunci **set** atau disebut juga dengan **setter** dan akan mengembalikan value yang bersifat private.

```
public void setNim(String nim) {
    this.nim = nim;
}
```

- Abstract Method yaitu method yang tidak memiliki body, dideklarasikan di dalam class abstract dengan menggunakan kata kunci abstract.
- Factory Method yaitu metod yang digunakan untuk mengembalikan object ke class yang bersangkutan, semua method static disebut factory method contohnya **NumberFormat obj = NumberFormat.getNumberInstance();**

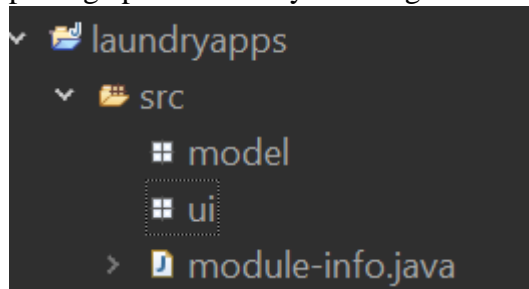
Constructor yaitu blok kode yang mirip dengan method, menggunakan kata kunci new ketika akan membuat sebuah object, nama constructor harus sama dengan class.

```
public Costumer(String id, String nama, String alamat, String hp) {
    super();
    this.id = id;
    this.nama = nama;
    this.alamat = alamat;
    this.hp = hp;
}
```

II. LANGKAH Pengerjaan

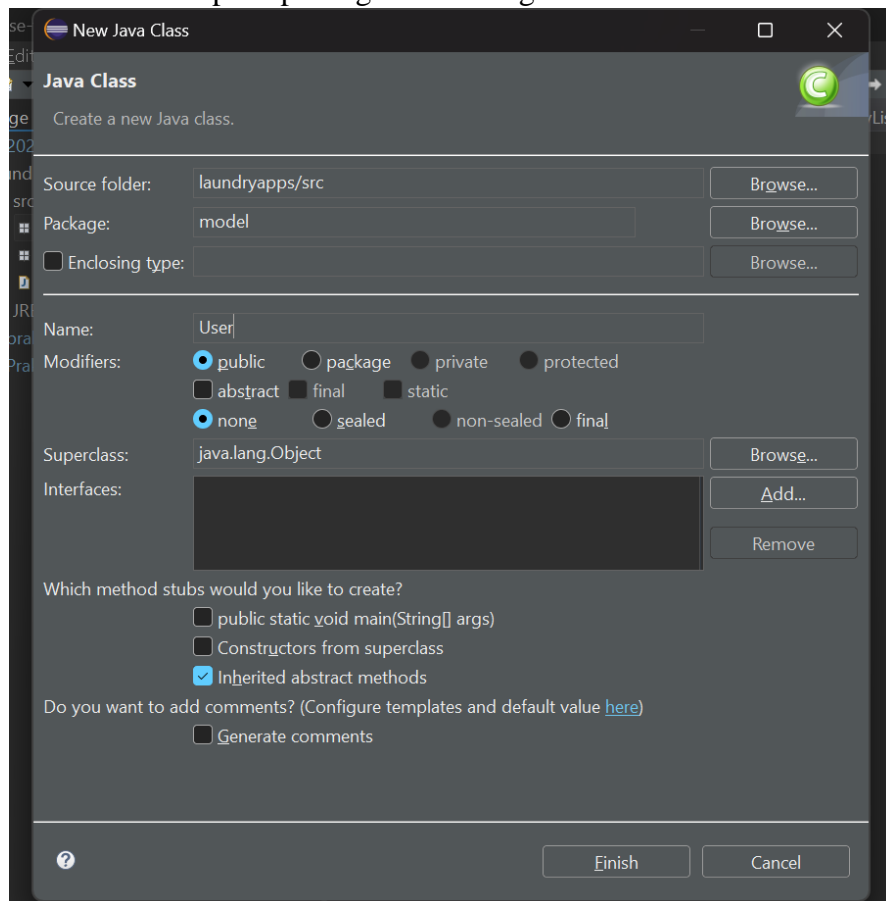
1. Membuat Class user

- a. Buat project java baru dengan nama **laundryapps**, setelah itu buat 2 buah package pada 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.

- b. Buat class baru pada package mode dengan nama **User**.



- c. Membuat attribute class user yaitu id, nama, username dan password

```

1 package model;
2
3 public class User {
4     String id, nama, username, password;
5
6 }
7

```

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

```

4     String id, nama, username, password;
5
6     public String getId() {
7         return id;
8     }
9
10    public void setId(String id) {
11        this.id = id;
12    }
13
14    public String getNama() {
15        return nama;
16    }
17
18    public void setNama(String nama) {
19        this.nama = nama;
20    }
21
22    public String getUsername() {
23        return username;
24    }
25
26    public void setUsername(String username) {
27        this.username = username;
28    }
29
30    public String getPassword() {
31        return password;
32    }
33
34    public void setPassword(String password) {
35        this.password = password;
36    }
37
38

```

Problems Javadoc Declaration Console × Coverage Git Staging
 terminated> LoginFrame [Java Application] C:\Users\LENOVO\p2\pool\plugins\org.e

- e. Membuat Method login yang nantinya akan digunakan ketika pengguna akan login ke aplikasi.

```
29
30 public String getPassword() {
31     return password;
32 }
33
34 public void setPassword(String password) {
35     this.password = password;
36 }
37
38 public static boolean login(String username, String password) {
39     boolean isLoggin = false;
40     User user = new User();
41     user.setId("1");
42     user.setNama("fulan");
43     user.setUsername("fulan");
44     user.setPassword("12345");
45
46     if(user.getUsername().equalsIgnoreCase(username)
47         && user.getPassword().equalsIgnoreCase(password)) {
48         isLoggin = true;
49     }else {
50         isLoggin = false;
51     }
52     return isLoggin = false;
53 }
54
55
56 }
57
```

Method login ini merupakan method static dikarenakan menggunakan kata kunci static yang mana dapat diakses dari class lain tanpa harus membuat sebuah object terlebih dahulu. Method login ini menggunakan tipe data Boolean yang mana akan mengembalikan nilai Boolean ketika digunakan, kemudian memiliki 2 buah parameter yaitu username dan password. Pertama akan variable isLoggin iberikan nilai false, kemudian dibuatkan sebuah object user baru, selanjutnya akan dilakukan pencocokan yang dikirimkan dari class lain, jika cocok maka isLoggin akan diubah menjadi true dan terakhir method akan mengembalikan nilai isLoggin.

Source code lengkap class User seperti di bawah ini:

```
1 package model;
2
3 public class User {
4     String id, nama, username, password;
5
6     public String getId() {
7         return id;
8     }
9
10    public void setId(String id) {
11        this.id = id;
12    }
13
14    public String getNama() {
15        return nama;
16    }
17
18    public void setNama(String nama) {
19        this.nama = nama;
20    }
21
22    public String getUsername() {
23        return username;
24    }
25
26    public void setUsername(String username) {
27        this.username = username;
28    }
29 }
```

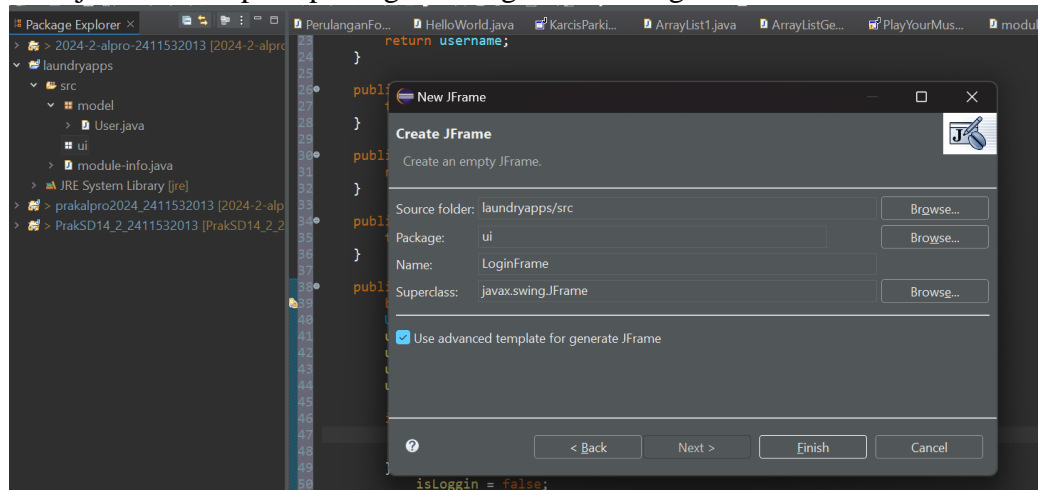
```

29
30 • public String getPassword() {
31     return password;
32 }
33
34 • public void setPassword(String password) {
35     this.password = password;
36 }
37
38 • public static boolean login(String username, String password) {
39     boolean isLoggedIn = false;
40     User user = new User();
41     user.setId("1");
42     user.setNama("fulan");
43     user.setUsername("fulan");
44     user.setPassword("12345");
45
46     if(user.getUsername().equalsIgnoreCase(username)
47         && user.getPassword().equalsIgnoreCase(password)) {
48         isLoggedIn = true;
49     }else {
50         isLoggedIn = false;
51     }
52     return isLoggedIn = false;
53 }
54
55
56 }

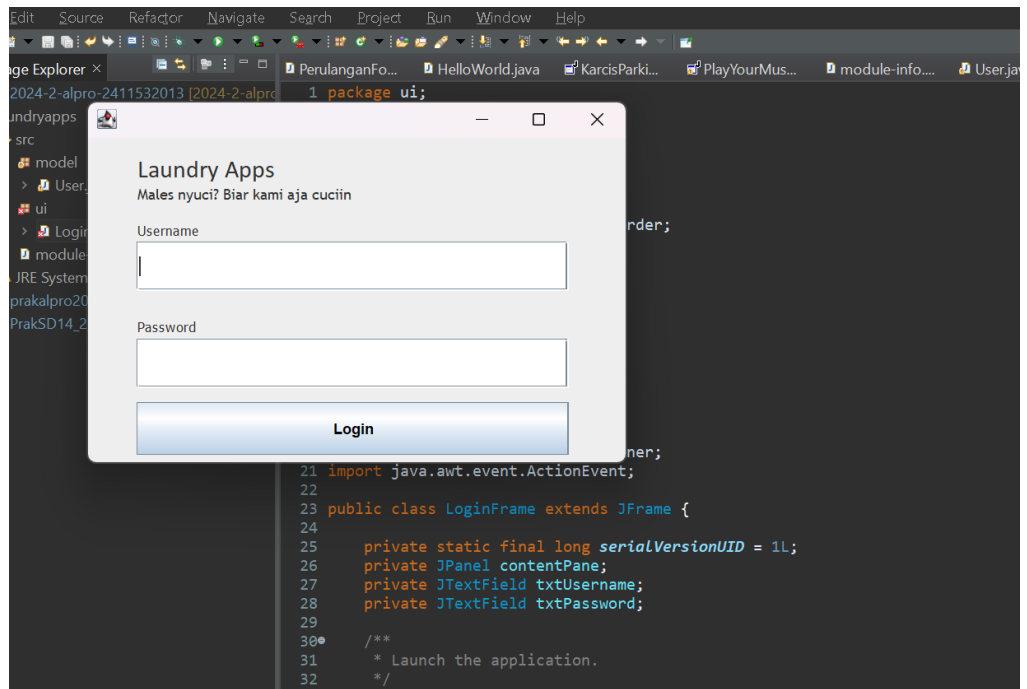
```

2. Membuat Tampilan Login menggunakan JFrame

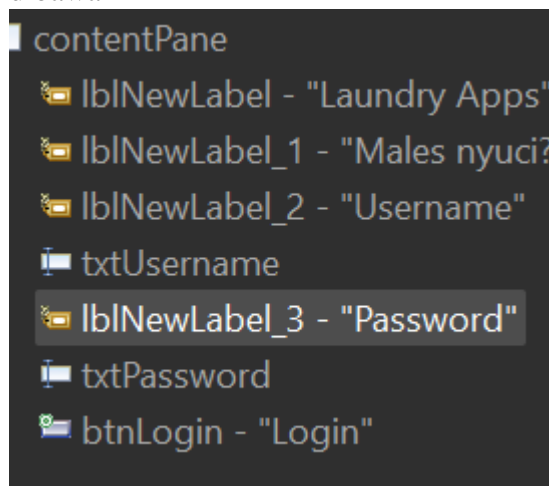
a. Buat JFrame baru pada package ui dengan nama LoginFrame



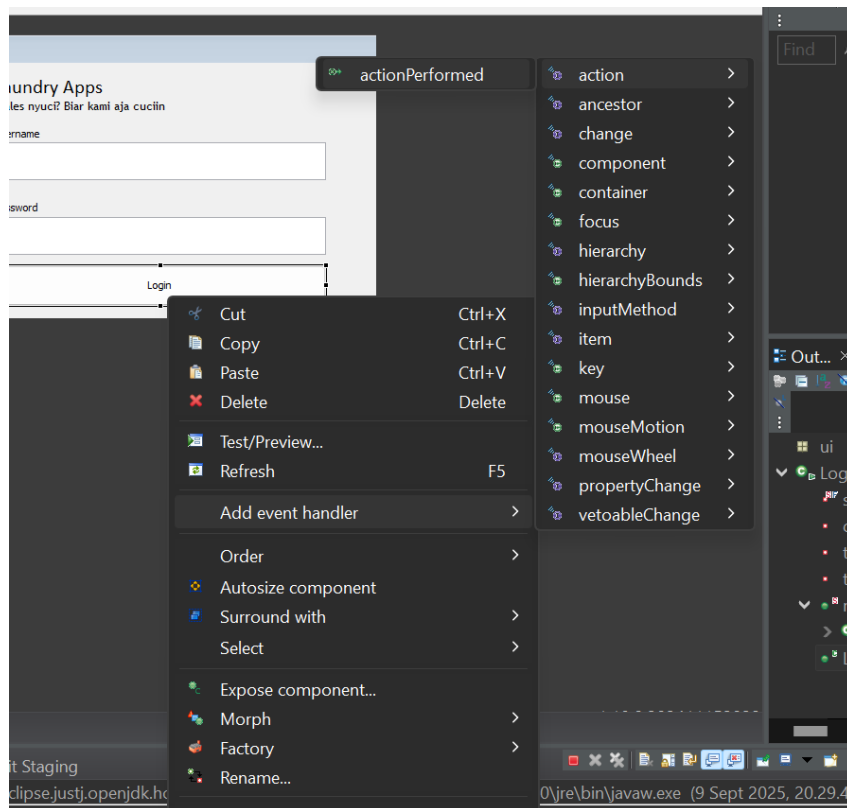
b. Kemudian buat desain tampilan login



- c. Kemudian ubah id JTextField username, password dan JButton seperti dibawah



- d. Selanjutnya klik kanan pada button login, pilih add event handler → action
→ actionPerformed



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

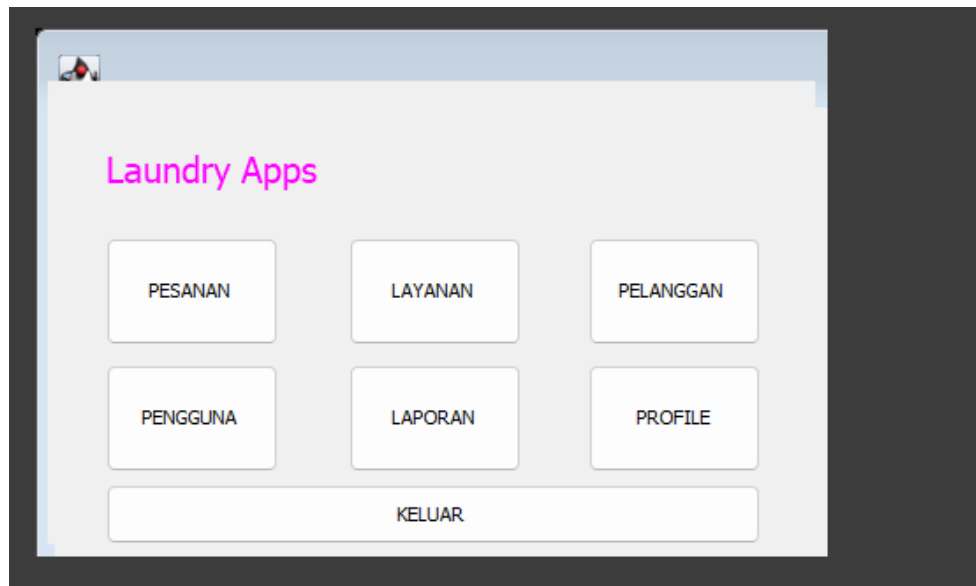
```

JButton btnLogin = new JButton("Login");
btnLogin.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if(user.login(txtUsername.getText(), txtPassword.getText())) {
            new MainFrame().setVisible(true);
            dispose();
        }else {
            JOptionPane.showMessageDialog(null, "Login Gagal");
        }
    }
});

```

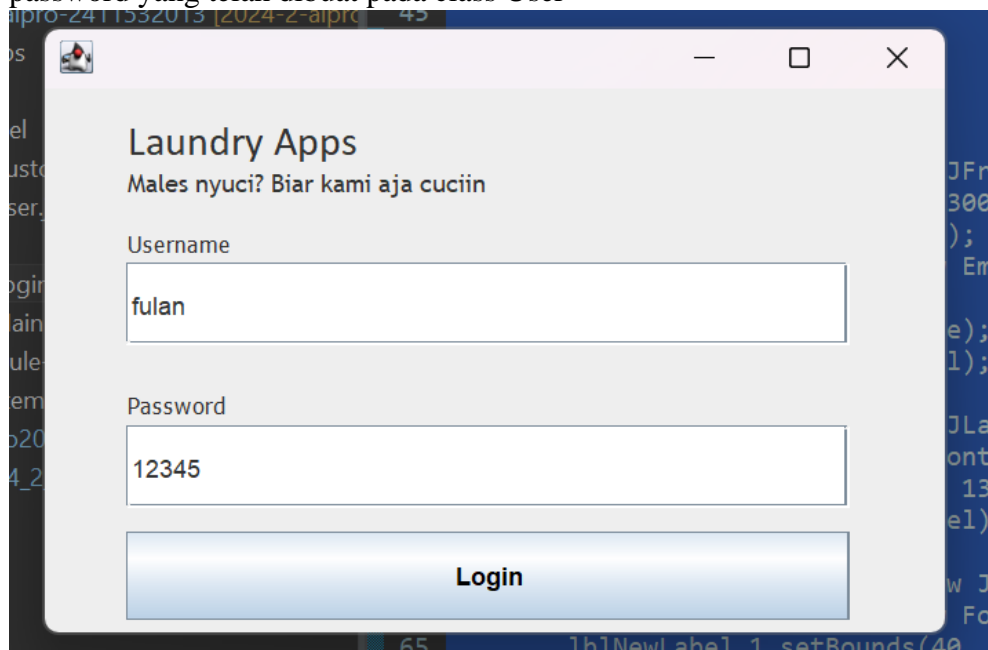
3. Membuat Tampilan Halaman Utama dengan JFrame

- a. Buat JFrame baru pada package ui dengan nama MainFrame
- b. Selanjutnya desain halaman utama seperti gambar di bawah

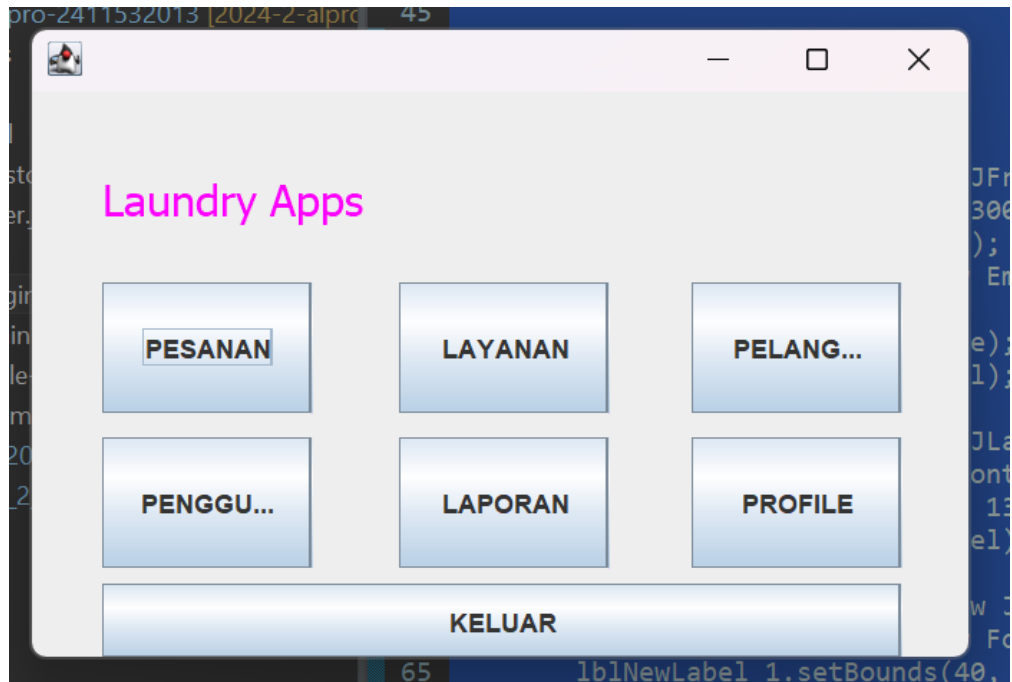


4. Hasil

- a. Login pada JFrame LoginFrame dengan menggunakan username dan password yang telah dibuat pada class User



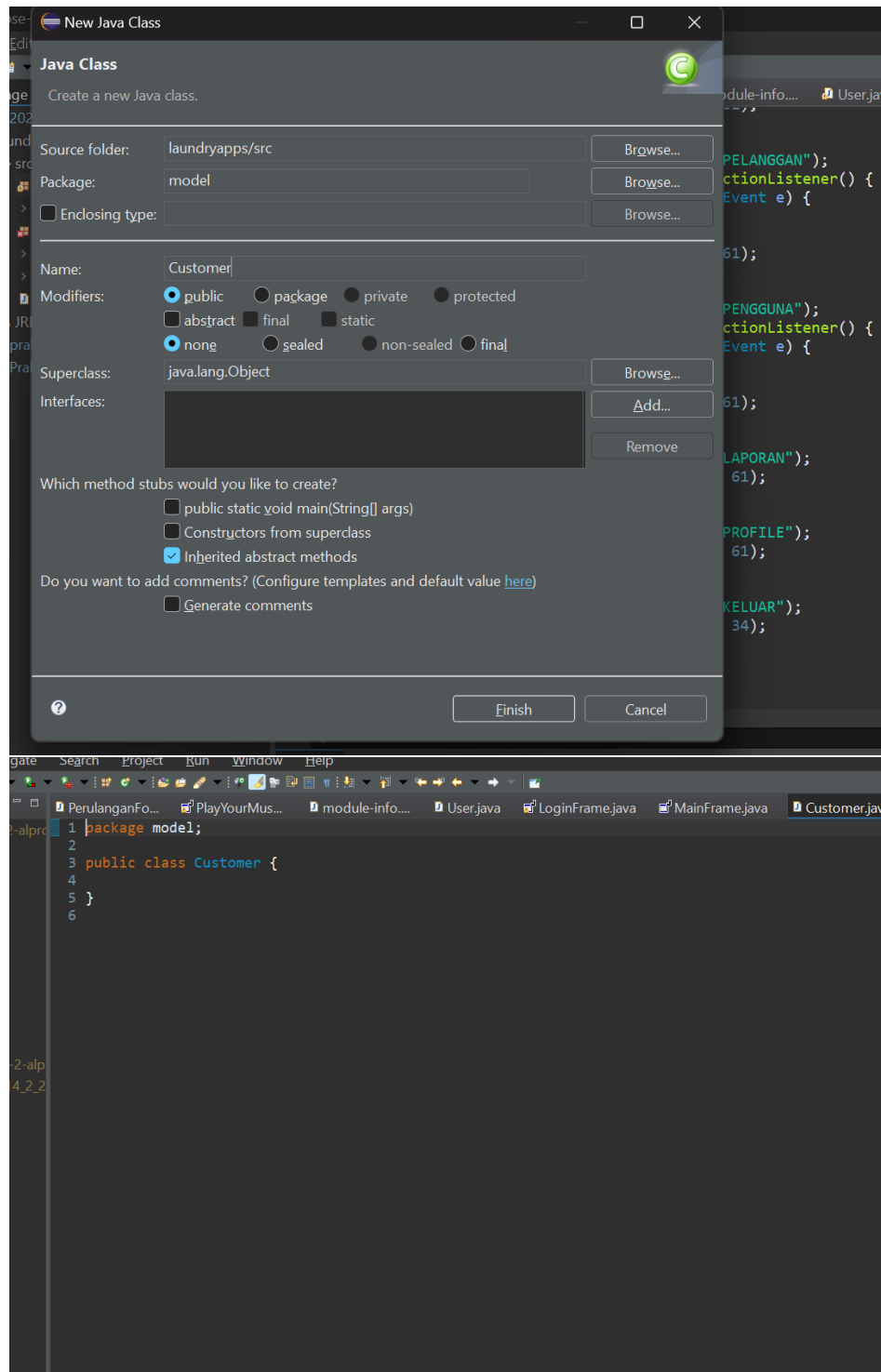
- b. Masuk ke halaman utama yang menandakan jika kode program berjalan dengan benar



III. Latihan/Tugas

Buatlah class dengan nama Customer dengan attribute id, nama, alamat dan nomor HP, buatkan setter dan getter untuk menambahkan object customer

- a. Membuat class dengan nama Customer



- b. Tambahkan attribute id, nama, alamat, dan nomor HP

```
1 package model;
2
3 public class Customer {
4     String id, nama, alamat, nomorHP;
5
6 }
7 |
```

- c. Buatkan setter dan getter untuk menambahkan object Customer

```
2
3 public class Customer {
4     String id, nama, alamat, nomorHP;
5
6     public String getId() {
7         return id;
8     }
9
10    public void setId(String id) {
11        this.id = id;
12    }
13
14    public String getNama() {
15        return nama;
16    }
17
18    public void setNama(String nama) {
19        this.nama = nama;
20    }
21
22    public String getAlamat() {
23        return alamat;
24    }
25
26    public void setAlamat(String alamat) {
27        this.alamat = alamat;
28    }
29
30    public String getNomorHP() {
31        return nomorHP;
32    }
33
34    public void setNomorHP(String nomorHP) {
35        this.nomorHP = nomorHP;
36    }
37 }
```