

LAPORAN PRAKTIKUM PEMROGRAMAN BERBASIS OBJEK
PEKAN 1: CLASS, OBJECT, CONSTRUCTOR DAN METHOD



Oleh :

MUHAMMAD FAZIL AFIF

NIM 2411531008

MATA KULIAH PEMOGRAMAN BERBASIS OBJEK

DOSEN PENGAMPU : NURFIAH, S.ST, M.KOM

FAKULTAS TEKNOLOGI INFORMASI

DEPARTEMEN INFORMATIKA

UNIVERSITAS ANDALAS

PADANG, 2025

A. Pendahuluan

Pemrograman berorientasi merupakan paradigma pemrograman berdasarkan konsep "objek", yang dapat berisi data, dalam bentuk field atau dikenal juga sebagai atribut; serta kode, dalam bentuk fungsi/prosedur atau dikenal juga sebagai method. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek. Bandingkan dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya.

Class adalah konsep dasar dalam pemrograman berbasis objek (OOP) yang berfungsi sebagai cetak biru atau blueprint untuk menciptakan objek, membuat pemrograman menjadi lebih terstruktur dan efisien. Objek adalah sebuah unit kode yang menggabungkan data (disebut atribut atau field) dan perilaku (disebut metode atau fungsi) untuk merepresentasikan entitas dunia nyata dalam sebuah program. Konstruktor (constructor) adalah metode khusus dalam sebuah kelas yang secara otomatis dipanggil saat sebuah objek baru dibuat untuk menginisialisasi objek tersebut. Method adalah blok kode terstruktur yang melakukan tugas tertentu dan terkait erat dengan objek atau kelas.

B. Tujuan

Tujuan dari dilakukannya praktikum ini adalah

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

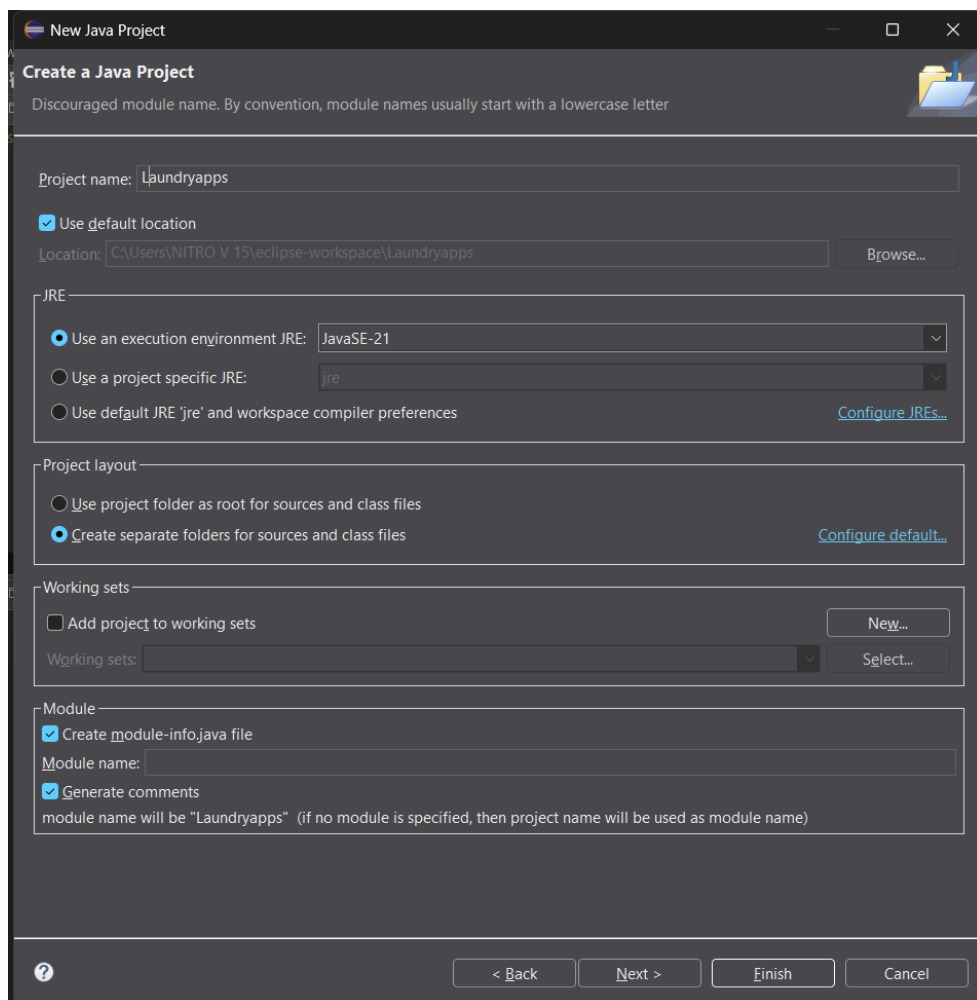
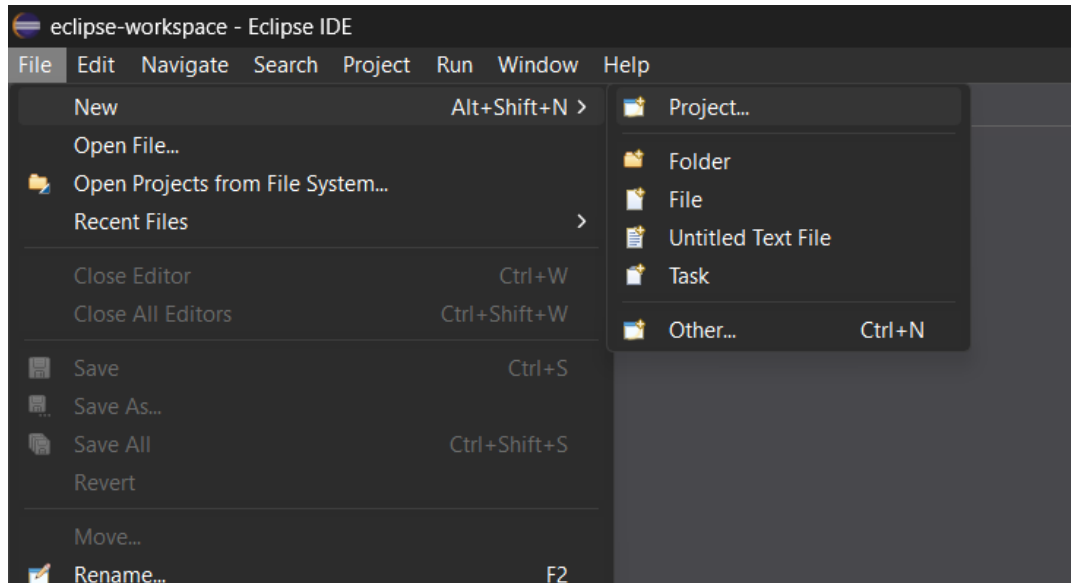
C. Alat dan Bahan

1. IDE (Integrated Development Environment) seperti Eclipse atau IntelliJIDEA
2. Java Development Kit (JDK)
3. Komputer atau Laptop

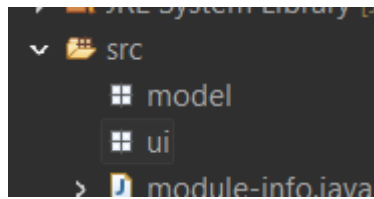
D. Langkah Kerja Praktikum

a. Membuat Class User

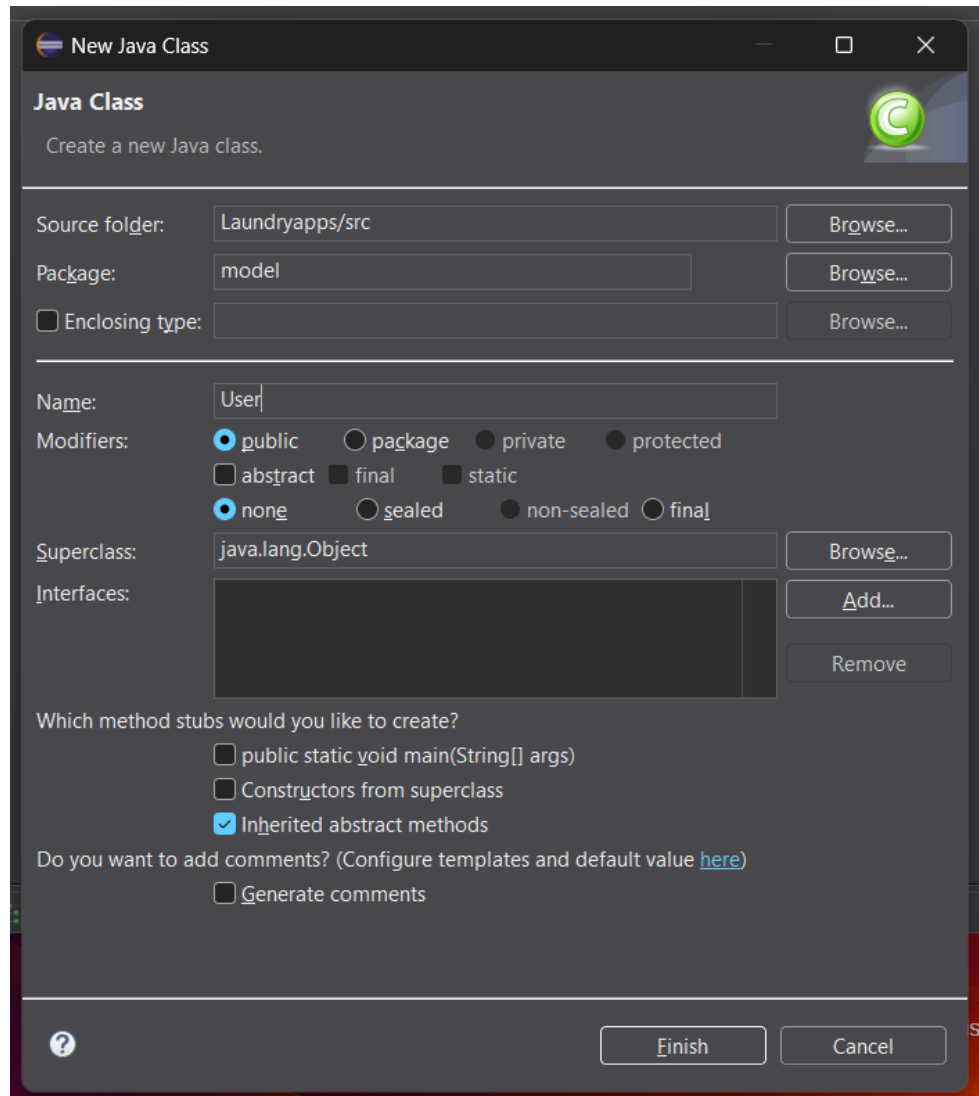
1. Buat projek baru dengan klik toolbar file di kiri atas, klik new lalu project dan project java buat nama lalu finish



2. Buatlah package baru di dalam projek dengan cara klik kanan new lalu package buat dua dengan nama model dan ui



3. Pada package model buat class baru bernama user



4. Pada user buat attribute id, nama, username, dan password

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

5. Lalu generate setter dan getter untuk attribute tadi. Method ini berfungsi untuk memasukkan/mengubah dan menampilkan value dari suatu object.

```
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;  
}
```

6. Lalu buat method login yang akan digunakan nanti. Method login bersifat *static* karena menggunakan kata kunci *static*, sehingga dapat dipanggil dari class lain tanpa perlu membuat objek terlebih dahulu. Method ini bertipe data Boolean, artinya saat dijalankan akan menghasilkan nilai berupa true atau false. Method login memiliki dua parameter, yaitu username dan password. Pada awalnya, variabel *isLogin* diatur bernilai false, lalu dibuat sebuah objek user baru. Setelah itu, sistem membandingkan data username dan password dari objek tersebut dengan data yang dikirimkan dari class lain. Jika keduanya sesuai, maka nilai *isLogin* akan berubah menjadi true. Pada akhirnya, method akan mengembalikan nilai dari *isLogin*.

```

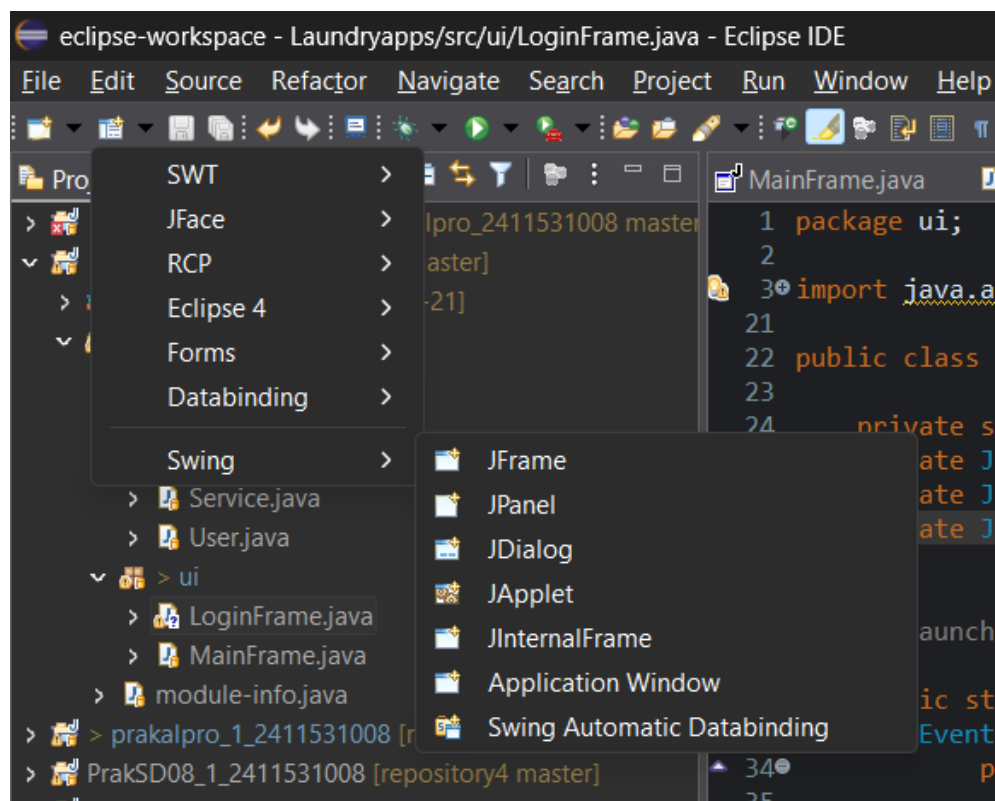
public static boolean login(String username, String password) {
    boolean isLogin = false;
    User user = new User();
    user.setId("1");
    user.setNama("fazil");
    user.setUsername("fazil");
    user.setPassword("12345");

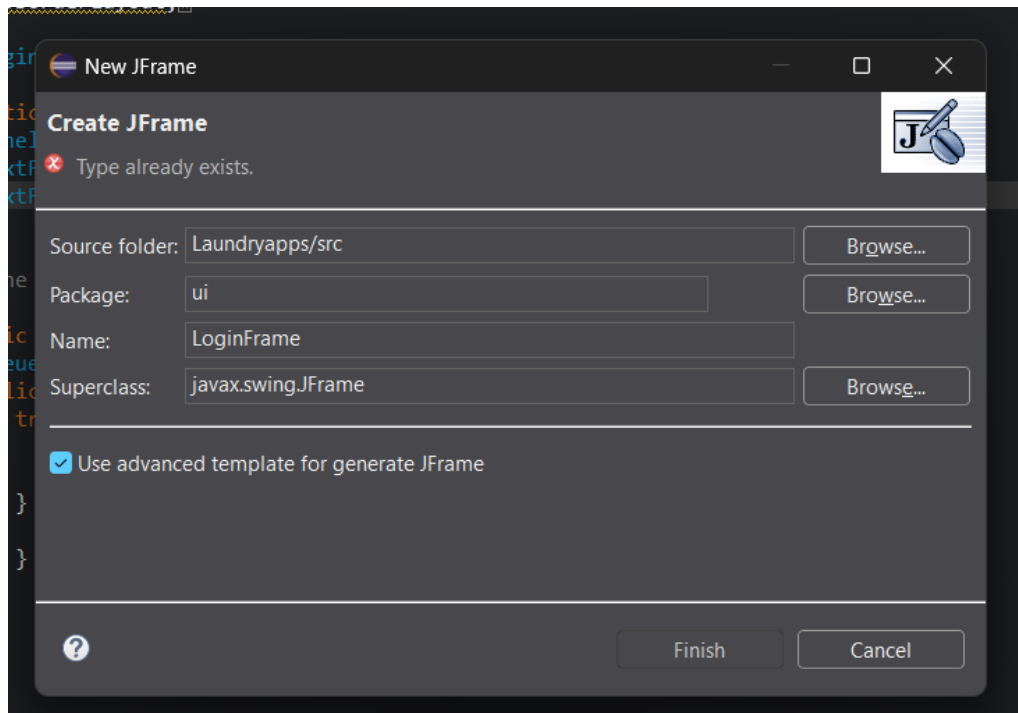
    if(user.getUsername().equalsIgnoreCase(username)
        && user.getPassword().equalsIgnoreCase(password)) {
        isLogin = true;
    } else {
        isLogin = false;
    }
    return isLogin;
}

```

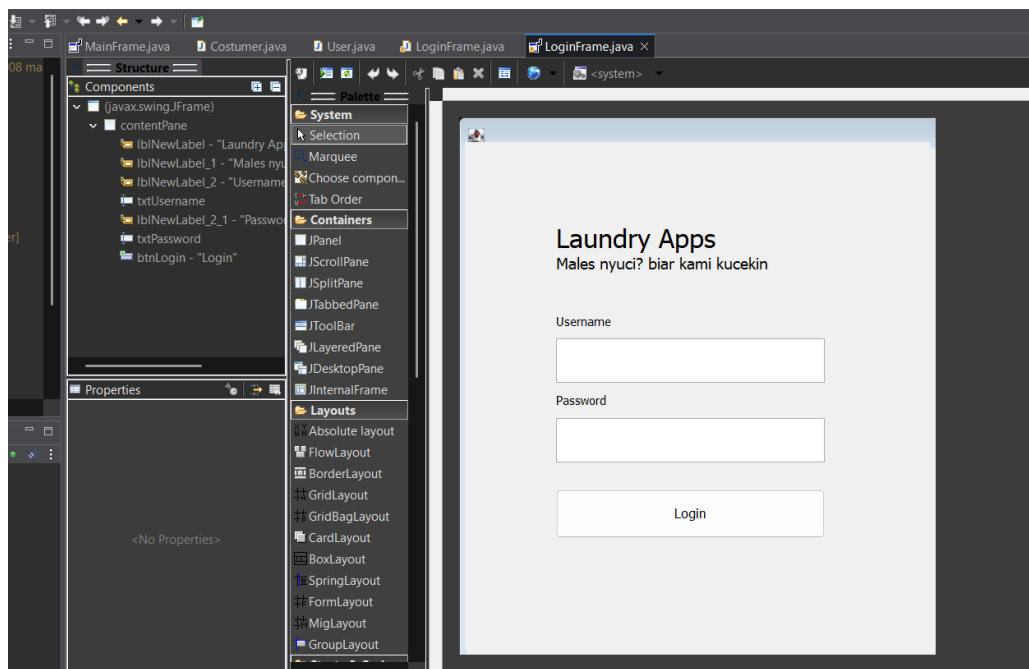
b. Membuat Tampilan Login menggunakan JFrame

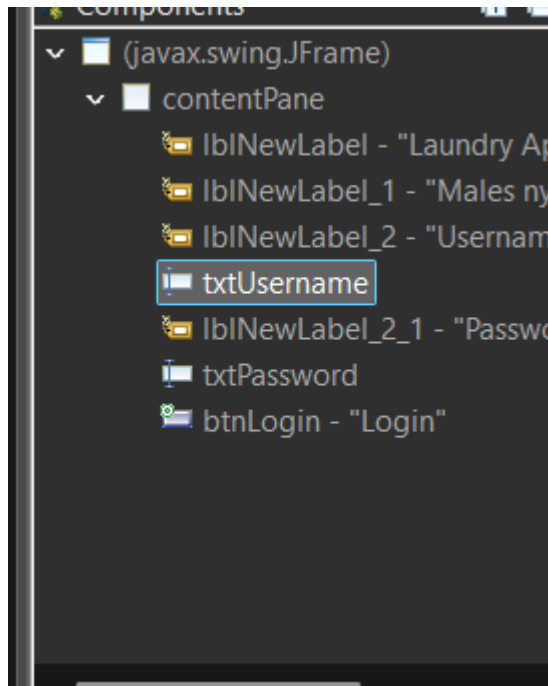
1. Buatlah JFrame baru di package ui dengan nama LoginFrame



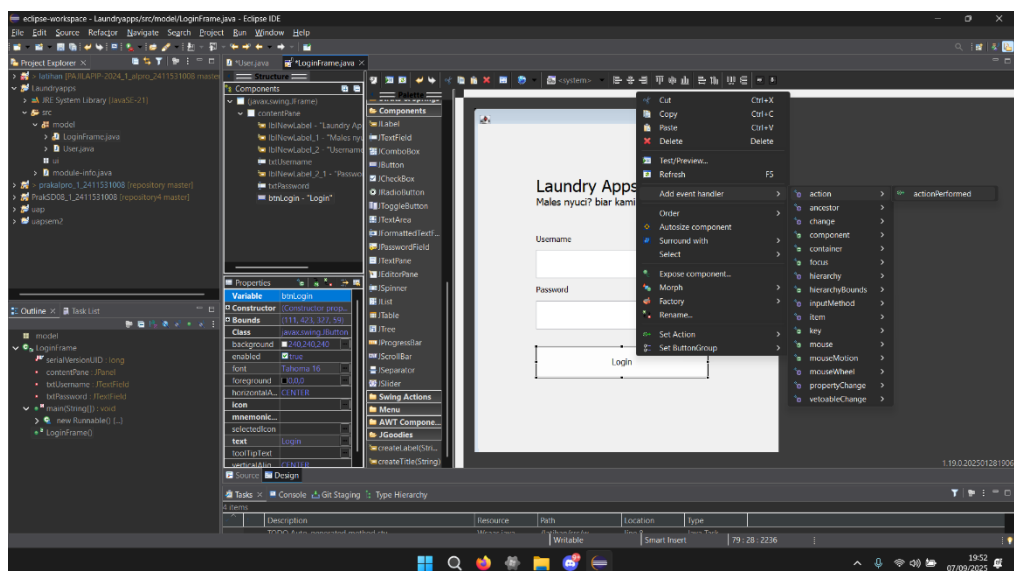


2. Buat desain seperti gambar dibawah, lalu ubah variable Jtextfield pada username menjadi txtUsername, password menjadi txtPassword, JButton menjadi btnLogin.





3. Selanjutnya klik kanan pada button login, pilih add event handler, lalu action, dan terakhir actionPerformed.



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

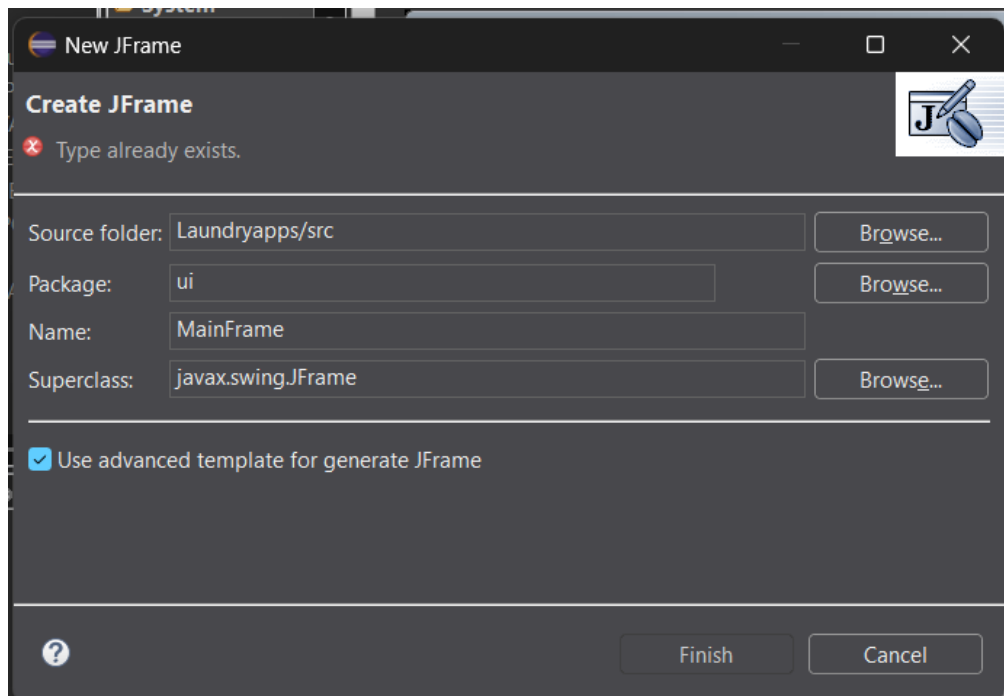

```

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");
        }
    }
});

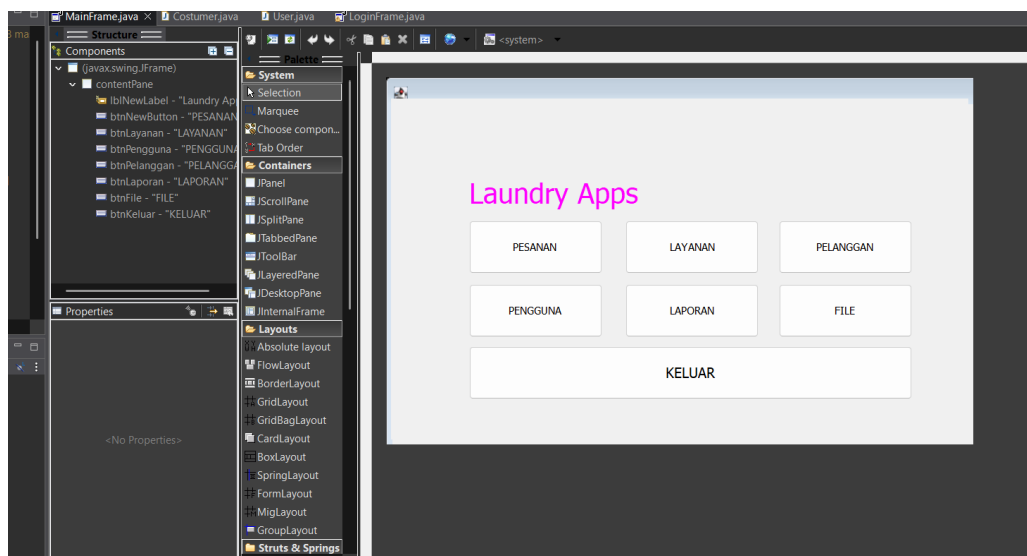
```

c. Membuat Tampilan Halaman Utama dengan JFrame

1. Buatlah JFrame baru di package ui dengan nama MainFrame

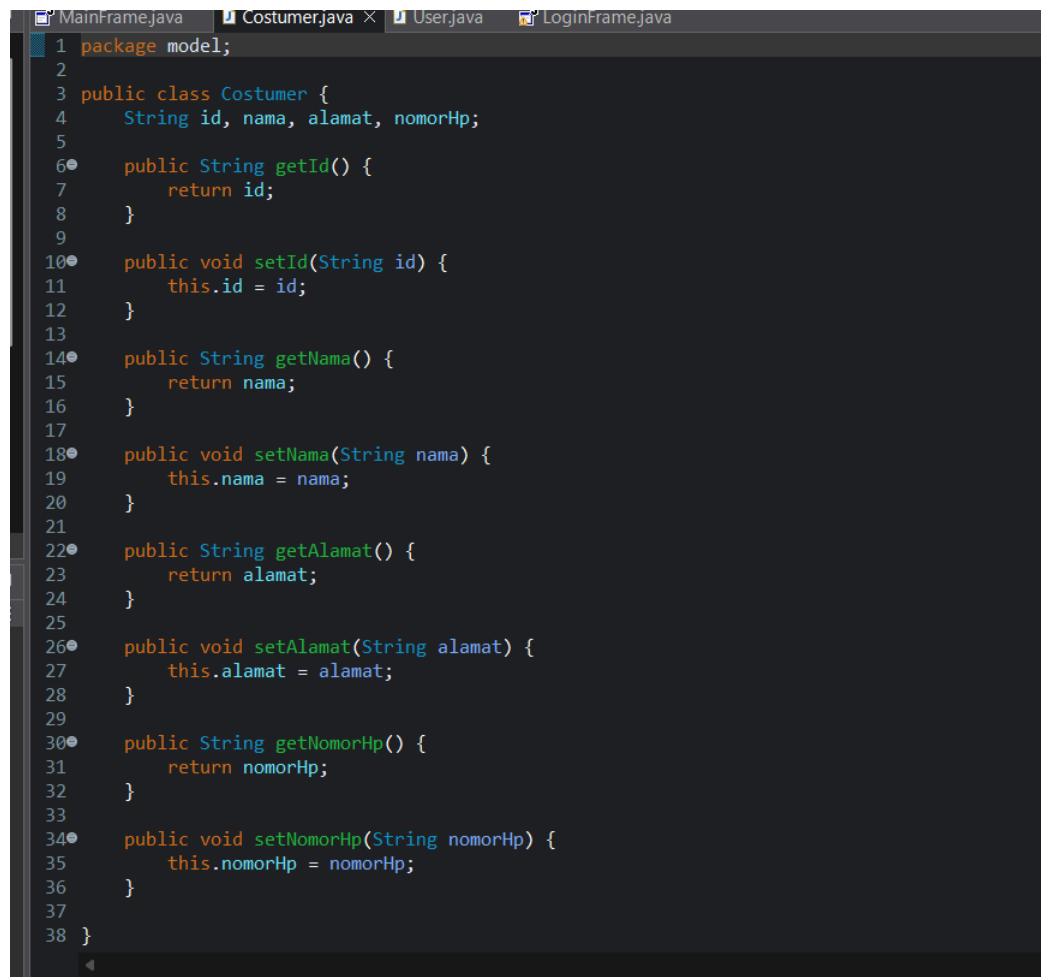


2. Lalu buat desain seperti dibawah



d. Latihan /Tugas

1. Buat class baru untuk costumer dengan attribute id, nama, Alamat dan nomor hp, Buatkan setter dan getter untuk menambahkan object Costumer



```
1 package model;
2
3 public class Costumer {
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 getName() {
15        return nama;
16    }
17
18    public void setName(String nama) {
19        this.nama = nama;
20    }
21
22    public String getAddress() {
23        return alamat;
24    }
25
26    public void setAddress(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
38 }
```

2. Buat class baru untuk Service dengan attribute id, jenis, harga dan status, buatkan setter dan getter untuk menambahkan object Service

```

1 package model;
2
3 public class Service {
4     String id, jenis, harga, status;
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 getJenis() {
15        return jenis;
16    }
17
18    public void setJenis(String jenis) {
19        this.jenis = jenis;
20    }
21
22    public String getHarga() {
23        return harga;
24    }
25
26    public void setHarga(String harga) {
27        this.harga = harga;
28    }
29
30    public String getStatus() {
31        return status;
32    }
33
34    public void setStatus(String status) {
35        this.status = status;
36    }
37 }
38

```

3. Buat class baru untuk Order dengan attribute id, id_costumer, id_service, id_user, total, tanggal, tanggal_selesai, status, status_pembayaran , buat setter dan getter untuk menambahkan object Order

```
1 package model;
2
3 public class Order {
4     String id, id_costumer, id_service, id_user, total, tanggal, tanggal_selesai, status, status_pembayaran;
5
6
7     public String getId() {
8         return id;
9     }
10
11     public void setId(String id) {
12         this.id = id;
13     }
14
15     public String getId_costumer() {
16         return id_costumer;
17     }
18
19     public void setId_costumer(String id_costumer) {
20         this.id_costumer = id_costumer;
21     }
22
23     public String getId_service() {
24         return id_service;
25     }
26
27     public void setId_service(String id_service) {
28         this.id_service = id_service;
29     }
30
31     public String getId_user() {
32         return id_user;
33     }
34
35     public void setId_user(String id_user) {
36         this.id_user = id_user;
37     }
38 }
```

E. Kesimpulan

Setelah melakukan praktikum ini, kita memperoleh pemahaman yang lebih mendalam mengenai konsep Pemrograman Berorientasi Objek (OOP), khususnya dalam penggunaan class, object, constructor, method, dan encapsulation. Kita juga mampu mengimplementasikan konsep tersebut ke dalam aplikasi Laundry sederhana dengan membuat class User, Customer, Service, dan Order, serta mendesain antarmuka menggunakan JFrame seperti LoginFrame dan MainFrame. Selain itu, kita dapat menghubungkan logika program dengan tampilan, misalnya berpindah dari halaman login ke halaman utama, sehingga praktikum ini memberikan pengalaman langsung dalam mengembangkan aplikasi berbasis Java.