

**LAPORAN PRAKTIKUM
PEMOGRAMAN BERORIENTASI OBJEK**

PRAKTIKUM 1

Class, Object, Constructor dan Method



Disusun oleh:

Renaldi Alwean Saputra

2411532003

Dosen Pengampu :

Nurfiah, S.ST, M.Kom

**PROGRAM STUDI S1 INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI**

UNIVERSITAS ANDALAS

SEPTEMBER

2025

A. Pendahuluan

Pemrograman Berorientasi Objek (PBO) merupakan salah satu paradigma pemrograman yang banyak digunakan dalam pengembangan perangkat lunak modern. Konsep utama dalam PBO adalah penggunaan **class** dan **object** untuk merepresentasikan permasalahan dunia nyata ke dalam bentuk program. Class berfungsi sebagai blueprint atau cetak biru, sementara object merupakan realisasi nyata dari class tersebut.

Melalui praktikum ini, mahasiswa mempelajari bagaimana membangun aplikasi sederhana dengan menerapkan konsep **class, object, constructor, method, dan encapsulation**. Studi kasus yang digunakan adalah **Aplikasi Manajemen Laundry**, di mana mahasiswa membuat beberapa class seperti **USER, COSTUMER, SERVICE dan ORDER** untuk merepresentasikan entitas dalam sistem laundry.

Selain itu, mahasiswa juga diajarkan bagaimana menghubungkan class-class tersebut dengan antarmuka pengguna berbasis **GUI (Graphical User Interface)** menggunakan **Java Swing**. Dengan demikian, aplikasi tidak hanya berjalan secara logika, tetapi juga dapat berinteraksi dengan pengguna melalui tampilan login dan halaman utama.

Praktikum ini menjadi penting karena selain memahami konsep dasar OOP, mahasiswa juga mulai membiasakan diri dalam mengintegrasikan logika pemrograman dengan tampilan antarmuka. Hasil akhirnya adalah sebuah aplikasi Laundry sederhana yang dapat menampilkan data pelanggan, layanan, pesanan, serta simulasi login pengguna.

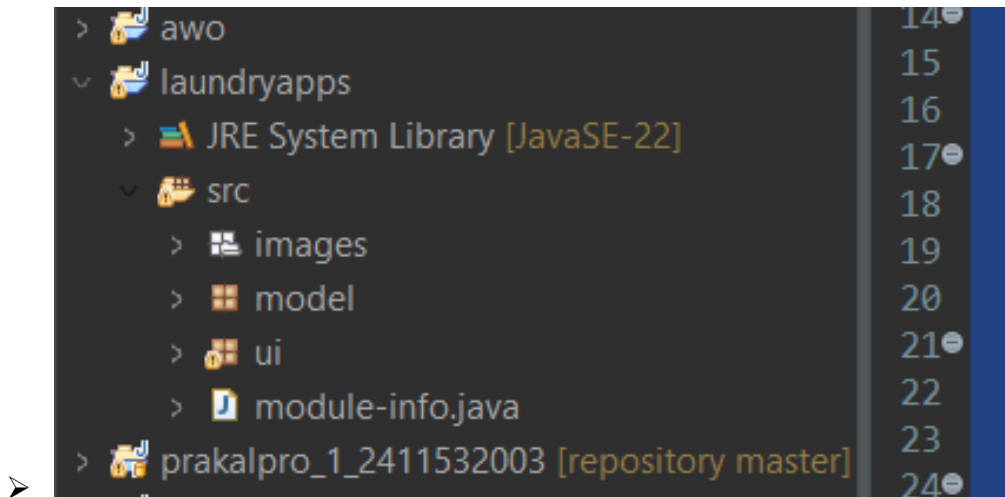
B. Tujuan

Tujuan praktikum ini yaitu mahasiswa mampu membuat class, object, encapsulation, constructor dan method menggunakan Bahasa Java serta menerapkannya pada aplikasi manajemen laundry. Selain itu, mahasiswa juga diharapkan mampu membuat antarmuka aplikasi sederhana menggunakan JFrame serta memahami proses perpindahan tampilan dari login ke halaman utama

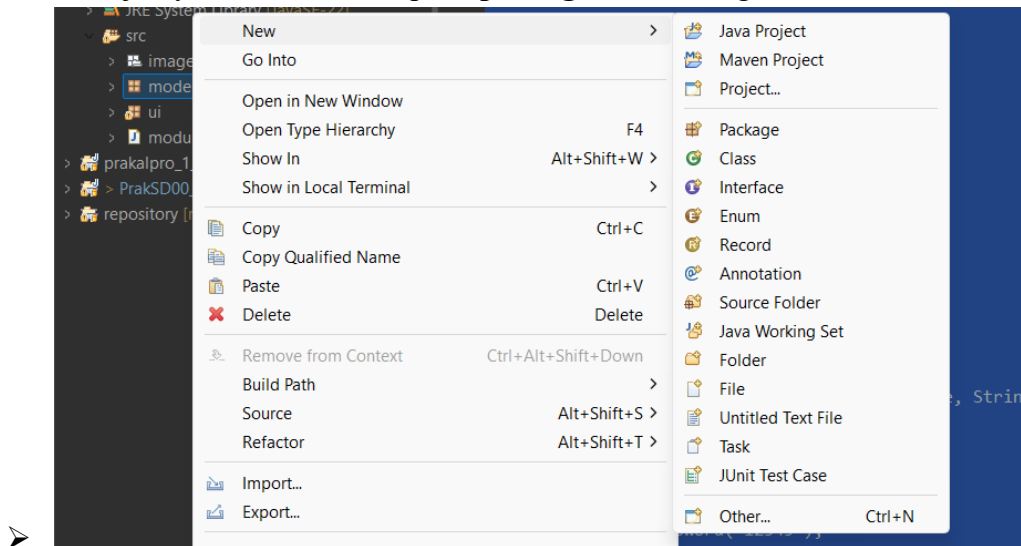
C. Langkah kerja

1. Membuat Class user

- Buat **Project** baru di Eclipse dengan nama **laundryapps**
- Selanjutnya buat 2 buah package dan 1 folder images (jika ingin menambahkan logo dan background) 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**



- Selanjutnya buat class baru pada **package model** dengan nama class **User**



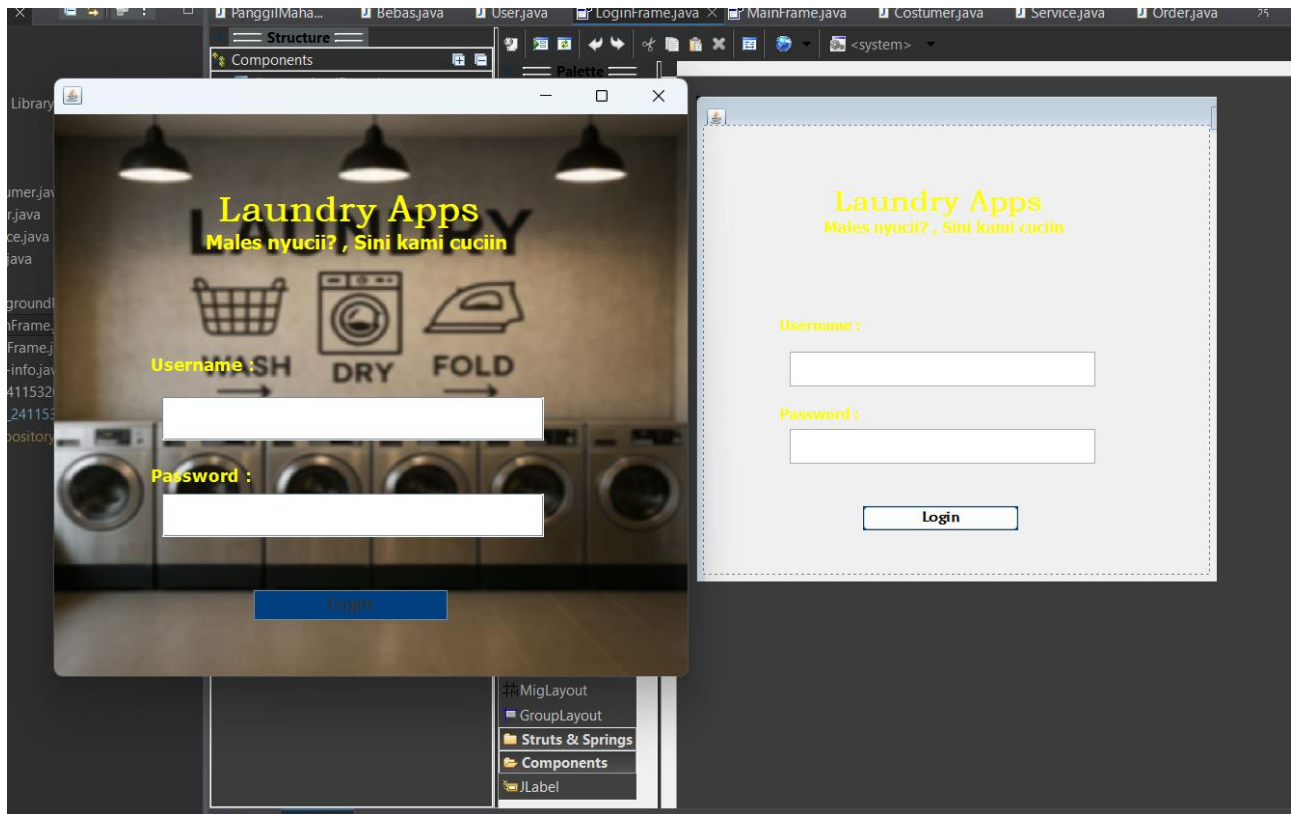
- Lalu buat attribute class **User : id,name,username,password** dengan menggunakan **String** , Selanjutnya buat method **Setter** dan **Getter** (berfungsi untuk memasukkan/mengubah dan menampilkan value dari suatu object), dan method **login** yang akan digunakan user ketika login ke aplikasi (method **Login** ini berhubungan dengan **LoginFrame** yang akan kita buat dibagian **package UI**
- Berikut contoh codingan nya :

```

2
3 public class User {
4     String nama, username, password;
5     private int id;
6
7     public int getId() {
8         return id;
9     }
10    public void setId(int id) {
11        this.id = id;
12    }
13
14    public String getName() {
15        return nama;
16    }
17    public void setName(String nama) {
18        this.nama = nama;
19    }
20    //RENALDI ALWEAN SAPUTRA
21    public String getUsername() {
22        return username;
23    }
24    public void setUsername(String username) {
25        this.username = username;
26    }
27
28    public String getPassword() {
29        return password;
30    }
31    public void setPassword(String password) {
32        this.password = password;
33    }
34    public static boolean login(String username, String password) {
35        boolean isLogin = false;
36        User user = new User();
37        user.setId(1);
38        user.setName("fulan");
39        user.setUsername("fulan");
40        user.setPassword("12345");
41
42

```

- Jika sudah membuat Class **User**, kita lanjut membuat **LoginFrame** dengan cara :
Package UI > New > Swing Designer > JFrame, jika sudah buat desain seperti dibawah



- Jika Desain sudah siap jangan lupa tambahkan codingan pada button **Login** agar pengguna bisa berinteraksi dengan aplikasi yang dibuat, berikut contoh codingan nya

```

2
3 JButton btnLogin = new JButton("Login");
4 btnLogin.setBackground(new Color(0, 64, 128));
5 btnLogin.addActionListener(new ActionListener() {
6     public void actionPerformed(ActionEvent e) {
7         if (model.User.Login(txtUsername.getText(), txtPassword.getText())) {
8             new MainFrame().setVisible(true);
9             dispose();
10        } else {
11            JOptionPane.showMessageDialog(null, "Login Gagal!");
12        }
13    }
14 }

```

- Lanjut Langkah selanjutnya karena di Codingan sebelumnya pada tombol **Login** terdapat **MainFrame** maka kita buat **MainFrame** agar tombol **Login** jika di interaksi pengguna memungkinkan pengguna masuk ke dalam halaman utama aplikasi atau yang disebut **MainFrame**



- Desain kurang lebih seperti gambar diatas lalu jangan lupa tambahkan codingan agar bisa di interaksi oleh pengguna, nah pada kasus ini kita perlu membuat class baru pada **Package Model** sama hal nya seperti **class User** kita buatkan **String, Setter dan Getter** serta Method pelanggan pada kasus ini Method Static

```

package model;

public class Costumer {
    private int id;
    private String nama;
    private String alamat;
    private String noHp;
    public int getId() { return id; }

    public void setId(int id) { this.id = id; }
    public String getNama() { return nama; }

    public void setName(String nama) { this.nama = nama; }
    public String getAlamat() { return alamat; }

    public void setAlamat(String alamat) { this.alamat = alamat; }
    public String getNoHp() { return noHp; }

    public void setNoHp(String noHp) { this.noHp = noHp; }

    public static Costumer createSampleCustomer() {
        Costumer c = new Costumer();
        c.setId(1);
        c.setName("Andi");
        c.setAlamat("Jl. Melati No. 5");
        c.setNoHp("08123456789");
        return c;
    }
}

```

- Contoh nya pada **class Costumer** sama halnya seperti **class User** ada **String, Getter dan Setter** dan method costumer nya static , pada bagian String dipisahkan dikarenakan **id** menggunakan **int** karena hanya angka namun kita tetap bisa mengubah menjadi **String** jika pada setId kita tambahkan tanda (“ “), dan sedikit Kesimpulan dari **Setter dan Getter** :

Getter

Fungsi untuk **mengambil/memanggil** nilai dari atribut private dalam class

Contoh : `public String getNama() { return nama; }`

Setter

Fungsi untuk **memberi atau mengubah nilai** atribut private dalam class

Contoh : `public void setNama(String nama) { this.nama = nama; }`

Selain itu untuk atribut **Private** tidak bisa di akses langsung di luar class, sehingga **SETTER DAN GETTER** bisa menjadi penghubung untuk membaca dan menulis

D. Kesimpulan

Praktikum Pemrograman Berorientasi Objek (PBO) ini berfokus pada penerapan konsep dasar PBO untuk membangun sebuah aplikasi manajemen laundry sederhana. Mahasiswa berhasil mengaplikasikan konsep-konsep inti PBO, termasuk **class, object, constructor, method**, dan **encapsulation**.

Secara spesifik, laporan ini menunjukkan pencapaian-pencapaian berikut:

- **Pemahaman Konsep PBO:** Mahasiswa mampu membedakan antara *class* sebagai cetak biru (*blueprint*) dan *object* sebagai realisasi nyata dari *class* tersebut.
- **Pembuatan Class dan Object:** Praktikum ini melibatkan pembuatan beberapa kelas, seperti User, Customer, Service, dan Order, untuk merepresentasikan entitas dalam sistem laundry. Setiap kelas ini dilengkapi dengan atribut, *getter* (untuk mengambil nilai), dan *setter* (untuk mengubah nilai) untuk mengimplementasikan konsep enkapsulasi.
- **Integrasi dengan GUI:** Aplikasi ini tidak hanya dibangun berdasarkan logika PBO, tetapi juga diintegrasikan dengan antarmuka pengguna grafis (GUI) menggunakan Java Swing. Mahasiswa berhasil membuat tampilan LoginFrame dan MainFrame yang memungkinkan interaksi langsung dengan pengguna.
- **Penerapan Logika Bisnis:** Dengan menggunakan method login pada class User, aplikasi ini mampu memvalidasi kredensial pengguna dan mengarahkan mereka ke halaman utama (MainFrame) jika login berhasil.

Sebagai hasilnya, praktikum ini berhasil mewujudkan sebuah aplikasi laundry sederhana yang dapat menampilkan data pelanggan, layanan, dan pesanan, serta mensimulasikan proses login pengguna.