LAPORAN PRATIKUM
IMPLEMENTASI APLIKASI DESKTOP LAUNDRY SEDERHANA MENGGUNAKAN JAVA SWING DENGAN KONSEP OOP

MATA KULIAH PEMROGRAMAN BERORIENTASI OBJEK (PBO)

DOSEN PENGAMPU :

NURFIAH, S.ST, M.KOM

OLEH:

AUFAN TAUFIQURRAHMAN

NIM 2411532011

FAKULTAS TEKNOLOGI INFORMASI

DEPARTEMEN INFORMATIKA

UNIVERSITAS ANDALAS

2025

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Dalam pengembangan perangkat lunak modern, pemisahan antara logika data (model) dan tampilan antarmuka (UI/view) adalah prinsip desain yang fundamental. Java, sebagai bahasa pemrograman berorientasi objek (OOP), menyediakan berbagai *tools* untuk membangun aplikasi, salah satunya adalah *library* Swing untuk pengembangan aplikasi desktop (GUI).

Praktikum ini adalah implementasi aplikasi "Laundry Apps" sederhana yang dibangun menggunakan Java Swing. Aplikasi ini dirancang untuk mengelola data laundry, namun pada tahap ini berfokus pada implementasi struktur model data dasar dan alur autentikasi pengguna (login) serta navigasi antar jendela (frame).

1.2 Tujuan Praktikum

Tujuan dari praktikum ini adalah:

1. Memahami dan menerapkan konsep dasar Pemrograman Berorientasi Objek (OOP) seperti *Class*, *Object*, dan *Encapsulation* (menggunakan *getters* dan *setters*) dalam mendefinisikan struktur data.
2. Mengimplementasikan Graphical User Interface (GUI) dasar menggunakan Java Swing, termasuk komponen seperti JFrame, JPanel, JButton, JTextField, dan JPasswordField.
3. Memahami dan menerapkan mekanisme *event handling* (khususnya ActionListener) untuk merespons interaksi pengguna (klik tombol).
4. Membangun alur navigasi aplikasi multi-jendela, khususnya perpindahan dari frame login ke frame menu utama setelah autentikasi berhasil.

BAB 2

DASAR TEORI

2.1 Pemrograman Berorientasi Objek (OOP)

OOP adalah paradigma pemrograman yang didasarkan pada konsep "objek", yang dapat berisi data dalam bentuk *fields* (atribut) dan kode dalam bentuk *procedures* (metode). Dalam Java, cetak biru (blueprint) untuk objek disebut *Class*. Proyek ini membagi *class* menjadi dua paket utama: model dan ui.

2.2 Java Swing

Java Swing adalah *widget toolkit* GUI (Graphical User Interface) untuk Java. Ini adalah bagian dari Java Foundation Classes (JFC). Swing menyediakan komponen yang kaya untuk membangun antarmuka desktop. Komponen kunci yang digunakan dalam proyek ini meliputi:

- **JFrame:** Jendela utama (kontainer level atas) aplikasi.
- **JPanel:** Kontainer umum yang digunakan untuk mengelompokkan komponen.
- **JLabel:** Komponen untuk menampilkan teks atau ikon.
- **JTextField/JPasswordField:** Bidang input teks (satu baris), di mana JPasswordField menyembunyikan karakter input (untuk kata sandi).
- **JButton:** Tombol standar yang dapat diklik pengguna.
- **JOptionPane:** Kelas utilitas untuk memunculkan dialog standar (seperti pesan error atau dialog konfirmasi).

2.3 Event Handling (ActionListener)

Agar komponen Swing dapat merespons interaksi pengguna (seperti klik tombol), Java menggunakan model delegasi acara (event delegation model). Sebuah *Listener* (dalam hal ini `ActionListener`) didaftarkan ke sebuah komponen (seperti `JButton`). Ketika acara (event) terjadi, metode yang relevan dalam *listener* (yaitu `actionPerformed`) akan dieksekusi.

2.4 Enkapsulasi: Accessor (Getter) dan Mutator (Setter)

Dalam Pemrograman Berorientasi Objek, metode *Accessor* dan *Mutator* adalah implementasi inti dari prinsip **Enkapsulasi**. Keduanya merupakan bagian dari *Instance Method* (method non-statis).

- **Accessor Method (Getter)** Modul mendefinisikan Accessor Method sebagai method yang "digunakan untuk membaca instance, menggunakan kata kunci get atau disebut getter". Fungsi utamanya adalah untuk mengambil atau membaca nilai dari sebuah atribut yang bersifat privat (private) di dalam sebuah kelas.
- **Mutator Method (Setter)** Modul mendefinisikan Mutator Method sebagai method yang "digunakan untuk membaca dan mengubah nilai, menggunakan kata kunci set atau disebut juga dengan setter". Fungsi utamanya adalah untuk mengatur, mengubah, atau menetapkan nilai baru ke atribut yang bersifat privat.

Dalam proyek ini, semua kelas di paket model (seperti User.java, Customer.java, Service.java, dan Order.java) menerapkan prinsip ini. Semua atribut (variabel) di dalamnya tidak dapat diakses langsung, namun disediakan method
getX() (getter) untuk membacanya dan setX() (setter) untuk mengubah nilainya.

BAB 3

IMPLEMENTASI DAN PEMBAHASAN

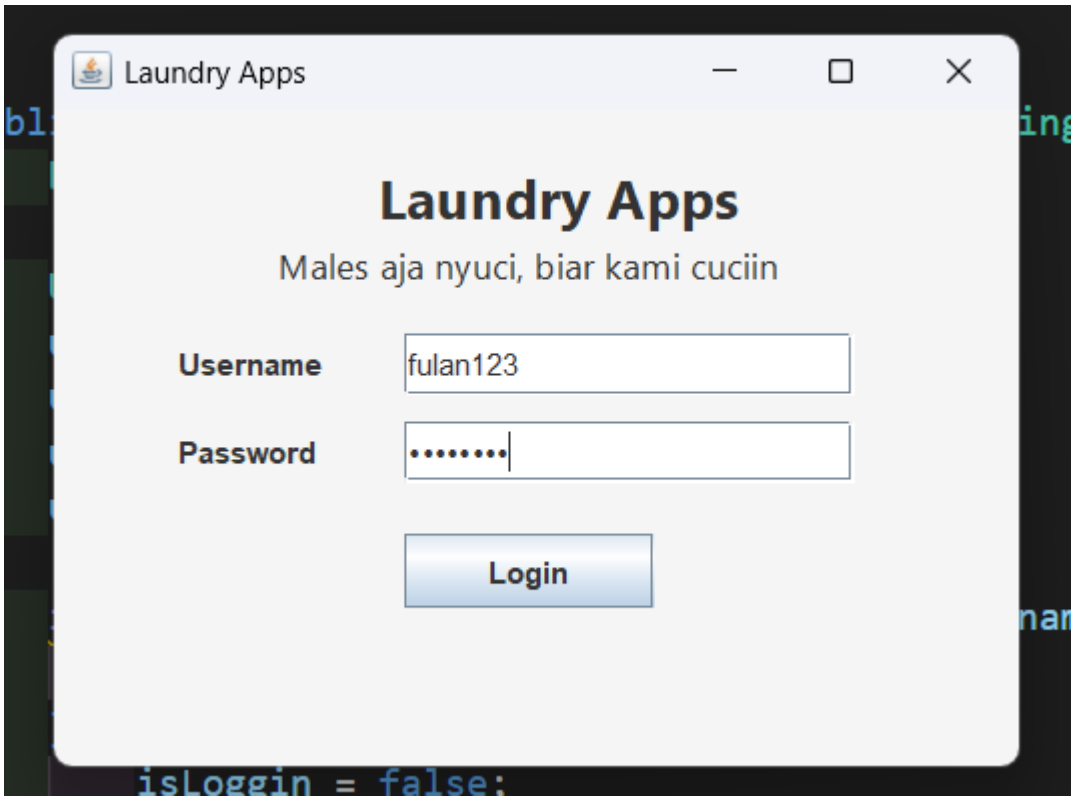
3.1. Pembuatan Struktur Proyek dan Class User

- Atribut: Class ini memiliki atribut yang diminta: id, nama, username, dan password.
- Encapsulation: Metode *Accessor* (getter) dan *Mutator* (setter) telah diimplementasikan untuk semua atribut, sesuai instruksi.
- Method Static Login: Sebuah method public static boolean login(String username, String password) dibuat seperti yang diinstruksikan.

- data login: username: "fulan" dan password: "12345".
- Implementasi kode menggunakan: username: "fulan123" dan password: "password". Logika validasi perbandingan (if-else) tetap sama dan berfungsi sesuai tujuan.

```
public static boolean login(String username, String password) {
    boolean isLoggin = false;

    User user = new User();
    user.setId(id:"1");
    user.setNama(nama:"fulan");
    user.setUsername(username:"fulan123");
    user.setPassword(password:"password");

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

}
```

3.2. Implementasi Tampilan Login (LoginFrame)



- Desain GUI: Kode LoginFrame.java berhasil mengimplementasikan desain tata letak (layout) seperti yang diinstruksikan dalam modul . Kode ini menggunakan panel.setLayout(null) (Absolute Layout) dan menempatkan komponen (JLabel, JTextField, JPasswordField, JButton) secara manual menggunakan setBounds().
- Event Handling: Logika perpindahan halaman telah diimplementasikan dengan sukses. Sebuah ActionListener ditambahkan ke btnLogin.

- Alur Logika: Ketika tombol ditekan, kode memanggil User.login(). Jika true, ia membuka MainFrame dan menutup frame login (frame.dispose()). Jika false, ia menampilkan JOptionPane.showMessageDialog dengan pesan "Login Gagal

```
    btnLogin.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            if (User.login(txtUsername.getText(), new String(txtPassword.getPassword()))) {
                new MainFrame().setVisible(b:true);
                frame.dispose();
            } else {
                JOptionPane.showMessageDialog(parentComponent:null, message:"Login Gagal");
            }
        }
    });

    frame.setVisible(b:true);
}
```

3.3. Implementasi Tampilan Halaman Utama (MainFrame)

- Desain GUI: Implementasi kode di MainFrame.java mencerminkan desain yang diinstruksikan dalam modul, menampilkan judul "Laundry Apps" (dengan warna Magenta) dan serangkaian tombol menu (PESANAN, LAYANAN, PELANGGAN, PENGGUNA, LAPORAN, PROFILE, dan KELUAR).
- Fungsionalitas Keluar: Fungsionalitas tombol "KELUAR" juga telah diimplementasikan. Kode ini menggunakan JOptionPane.showConfirmDialog untuk konfirmasi sebelum menutup aplikasi via System.exit(0).


Laundry Apps - Main Menu
Laundry Apps
PESANAN
LAYANAN
PELANGGAN
PENGGUNA
PROFILE
KELUAR
Konfirmasi
Apakah yakin ingin keluar?
Yes
No

```java
public class MainFrame extends JFrame {

    public MainFrame() {
        setTitle(title:"Laundry Apps - Main Menu");
        setSize(width:600, height:400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(c:null);

        // Panel utama
        JPanel panel = new JPanel();
        panel.setBackground(new Color(r:245, g:245, b:245));
        panel.setLayout(mgr:null); // Menggunakan Absolute Layout
        add(panel);

        // Judul
        JLabel titleLabel = new JLabel(text:"Laundry Apps");
        titleLabel.setFont(new Font(name:"Segoe UI", Font.BOLD, size:26));
        titleLabel.setForeground(Color.MAGENTA);
        titleLabel.setBounds(x:200, y:20, width:200, height:35); // x, y, width, height
        panel.add(titleLabel);


        // Baris pertama: Pesanan, Layanan, Pelanggan
        JButton btnPesanan = new JButton(text:"PESANAN");
        btnPesanan.setBounds(x:50, y:80, width:150, height:40);
        panel.add(btnPesanan);

        JButton btnLayanan = new JButton(text:"LAYANAN");
```

BAB 4

PENYELESAIAN LATIHAN / TUGAS

4.1. Tugas 1: Pembuatan Class Customer

- **Permintaan Modul:** "Buatlah class dengan nama Costumer dengan attribute id, nama, Alamat dan nomor hp, buatkan setter dan getter...".
- **Hasil Implementasi:** File model/Customer.java telah dibuat.
- **Analisis:** Class ini berisi atribut yang diminta (id, nama, alamat, telepon) dan menyertakan satu set lengkap metode getter (accessor) dan setter (mutator) untuk setiap atribut, sehingga memenuhi tugas pertama.

```java
package model;

public class Customer {
    String id, nama, alamat, telepon;

    public String getId() {···

    public void setId(String id) {···

    public String getNama() {···

    public void setNama(String nama) {···

    public String getAlamat() {···

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

    public String getTelepon() {
        return telepon;
    }

    public void setTelepon(String telepon) {
        this.telepon = telepon;
    }
}
```

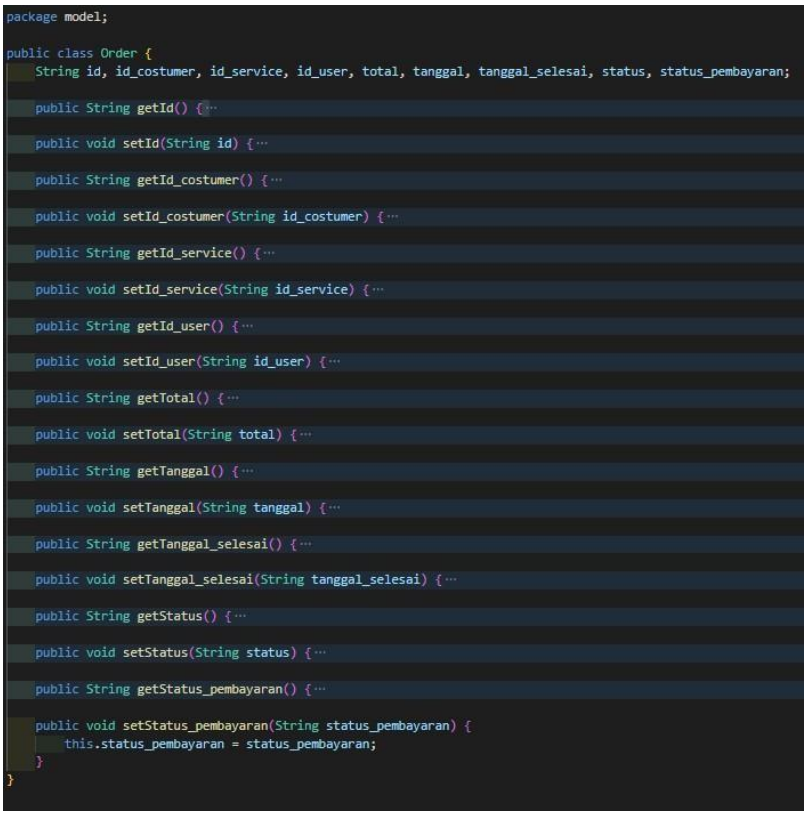
4.2. Tugas 2: Pembuatan Class Service

- **Permintaan Modul:** "Buatlah class dengan nama Service dengan attribute id, jenis, harga dan status, buatkan setter dan getter...".
- **Hasil Implementasi:** File model/Service.java telah dibuat.
- **Analisis:** Class ini berisi atribut yang diminta (id, jenis, harga, status) dan menyertakan metode getter dan setter yang sesuai untuk semua atribut, sehingga memenuhi tugas kedua.

```java
package model;

public class Service {
    String id, jenis, harga, status;

    public String getId() {
        return id;
    }

    public void setId(String id) {...

    public String getJenis() {
        return jenis;
    }

    public void setJenis(String jenis) {...

    public String getHarga() {
        return harga;
    }

    public void setHarga(String harga) {...

    public String getStatus() {
        return status;
    }

    public void setStatus(String status) {
        this.status = status;
    }

}
```

4.3. Tugas 3: Pembuatan Class Order

- **Permintaan Modul:** "Buatlah class dengan nama Order dengan attribute id, id_costumer, id_service, id_user, total, tanggal, tanggal_selesai, status, status_pembayaran, buatkan setter dan getter...".
- **Hasil Implementasi:** File model/Order.java telah dibuat.
- **Analisis:** Class ini secara komprehensif mendefinisikan semua atribut yang diminta oleh modul (termasuk id_costumer, id_service, status_pembayaran, dll.) dan menyediakan metode getter dan setter lengkap untuk setiap atribut tersebut, sehingga memenuhi tugas ketiga.



```
package model;

public class Order {
    String id, id_costumer, id_service, id_user, total, tanggal, tanggal_selesai, status, status_pembayaran;

    public String getId() {...

    public void setId(String id) {...

    public String getId_costumer() {...

    public void setId_costumer(String id_costumer) {...

    public String getId_service() {...

    public void setId_service(String id_service) {...

    public String getId_user() {...

    public void setId_user(String id_user) {...

    public String getTotal() {...

    public void setTotal(String total) {...

    public String getTanggal() {...

    public void setTanggal(String tanggal) {...

    public String getTanggal_selesai() {...

    public void setTanggal_selesai(String tanggal_selesai) {...

    public String getStatus() {...

    public void setStatus(String status) {...

    public String getStatus_pembayaran() {...

    public void setStatus_pembayaran(String status_pembayaran) {
        this.status_pembayaran = status_pembayaran;
    }
}
```

BAB 5

KESIMPULAN

Praktikum ini berhasil mendemonstrasikan implementasi aplikasi Java Swing sederhana dengan prinsip OOP. Pemisahan struktur data dalam paket model dan antarmuka dalam paket ui telah diterapkan dengan baik. Alur autentikasi pengguna dan navigasi antar-frame (Login ke MainMenu) telah berhasil diimplementasikan menggunakan ActionListener dan validasi logika statis.