

LAPORAN PRAKTIKUM 1

PEMROGRAMAN BERBASIS OBJEK



Disusun Oleh :

SASKIA ALIFAH (2411531002)

Dosen Pengampu : Nurfiah, S.ST, M.KOM

Departemen Informatika
Fakultas Teknologi Informasi
Universitas Andalas
Tahun 2025

LaundryApps

A. TUJUAN PRAKTIKUM

1. Mahasiswa mampu memahami konsep dasar pemrograman berorientasi objek melalui implementasi class, object, atribut, dan method dalam project Java.
2. Mahasiswa dapat mengimplementasikan antarmuka grafis (GUI) menggunakan Java Swing untuk membangun aplikasi sederhana yang interaktif.
3. Mahasiswa dapat membangun struktur aplikasi dengan pemisahan package model dan ui sehingga kode lebih terorganisir, mudah dibaca, serta siap dikembangkan lebih lanjut.

B. PENDAHULUAN

Pemrograman berorientasi objek PBO merupakan paradigma pemrograman yang berfokus pada penggunaan object untuk memodelkan entitas dalam dunia nyata. Dalam PBO, setiap object memiliki atribut (data) dan method (perilaku) yang saling terkait. Konsep ini memudahkan pengembang untuk membuat aplikasi yang modular, terstruktur, dan dapat digunakan kembali.

Dalam praktikum ini, mahasiswa diarahkan untuk membuat sebuah aplikasi sederhana berbasis Java dengan memanfaatkan konsep PBO. Project ini diberi nama **laundryapps** yang bertujuan untuk mensimulasikan proses login pengguna dan navigasi ke menu utama menggunakan antarmuka grafis berbasis Swing. Struktur project dibagi menjadi dua package, yaitu model untuk menyimpan class-class inti, serta ui untuk menyimpan komponen tampilan.

Pembuatan aplikasi Laundry ini juga melatih mahasiswa agar terbiasa menggunakan GitHub sebagai platform version control. Dengan adanya repository, kolaborasi antara mahasiswa dan dosen menjadi lebih mudah serta memudahkan pemantauan perkembangan project. Selain itu, penggunaan GUI pada aplikasi ini dapat memberikan gambaran nyata bagaimana sebuah aplikasi desktop sederhana dikembangkan dari awal.

C. KAJIAN TEORI

1. Class

Class merupakan sekumpulan objek yang memiliki karakteristik (sifat/behavior) dan properti (atribut) yang sama. Class dapat disebut sebagai *template* atau *blueprint* dari objek yang akan dibuat. Dengan adanya class, programmer dapat membentuk objek-objek baru yang memiliki kesamaan perilaku maupun atribut. Sebagai contoh, jika terdapat class Mahasiswa, maka objek Fulan merupakan salah satu instansiasi dari class tersebut. Class dalam Java bukanlah entitas nyata, melainkan

rancangan untuk membentuk objek nyata. Di dalam Java, sebuah class dapat berisi **Data Member, Method, Constructor, Nested Class**, maupun **Interface**.

2. Object

Object adalah representasi nyata dari entitas yang ada di dunia nyata berdasarkan class yang telah dibuat. Objek memiliki tiga komponen penting, yaitu:

- a) **State**, yang merepresentasikan atribut dari sebuah objek.
- b) **Behavior**, yang merepresentasikan fungsi atau method yang dapat dilakukan oleh objek.
- c) **Identity**, yaitu nama unik suatu objek yang memungkinkan objek tersebut berinteraksi dengan objek lain. Dengan adanya objek, konsep *Object Oriented Programming* dapat berjalan, karena objek-objek dapat saling berhubungan dan berkomunikasi sesuai dengan kebutuhan sistem.

3. Method

Method adalah blok kode yang dapat dijalankan berulang kali untuk tujuan tertentu, sehingga program menjadi lebih terorganisir, efisien, dan mudah digunakan kembali. Java menyediakan beberapa method bawaan, misalnya `main()`, `equals()`, `toString()`, dan sebagainya. Aturan umum dalam pembuatan method di Java adalah sebagai berikut:

- a) Nama method harus berupa kata kerja (*verb*) dan dimulai dengan huruf kecil.
- b) Jika nama method terdiri dari lebih dari satu kata, maka menggunakan *camelCase* (contoh: `sum()`, `areaOfCircle()`).

Method dalam Java dibagi menjadi beberapa jenis, diantaranya yaitu:

- a) Predefined Method, yaitu method bawaan Java seperti `length()`, `equals()`, dan `compareTo()`.
- b) User-defined Method, yaitu method yang dibuat programmer sesuai kebutuhan.
- c) Static Method, yaitu method dengan kata kunci `static` sehingga dapat dipanggil tanpa membuat objek.
- d) Instance Method, yaitu method non-static yang memerlukan objek untuk dipanggil. Instance method terbagi menjadi:
 - a. Accessor Method (Getter), digunakan untuk membaca nilai atribut.
 - b. Mutator Method (Setter), digunakan untuk mengubah nilai atribut.
- e) Abstract Method, yaitu method tanpa isi (body) yang hanya dideklarasikan pada class abstract.

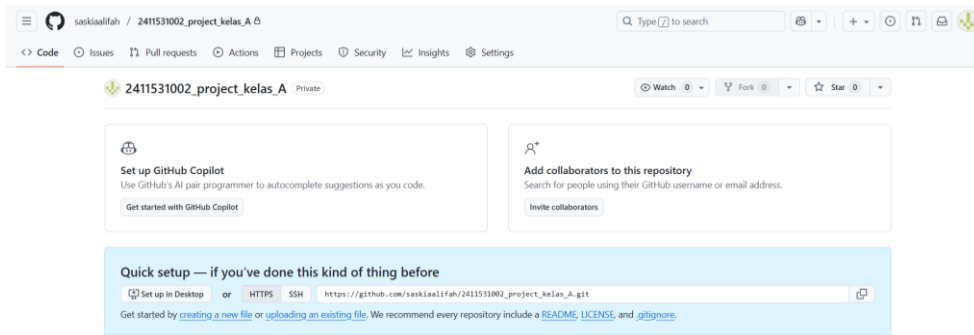
4. Constructor

Constructor merupakan blok kode khusus yang digunakan untuk menginisialisasi objek. Constructor memiliki nama yang sama dengan nama class dan dipanggil menggunakan kata kunci `new`. Tidak seperti method, constructor tidak memiliki *return type*. Dengan adanya constructor, objek dapat dibuat secara otomatis dengan nilai awal tertentu sesuai kebutuhan.

D. METODE PRAKTIKUM

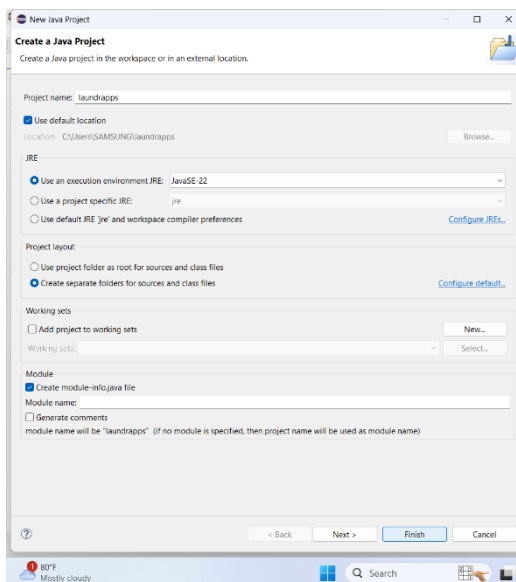
1. Membuat Repo di Github

Langkah pertama adalah membuat repository di GitHub dengan nama **2411531002_project_kelas_A**. Repository ini dibuat dalam mode **private** agar hanya bisa diakses oleh mahasiswa dan dosen pengampu. Dosen pengampu ditambahkan sebagai **collaborator** sehingga dapat memantau dan menilai progress project yang telah dibuat mahasiswa.



2. Membuat Java project dengan nama laundryapps

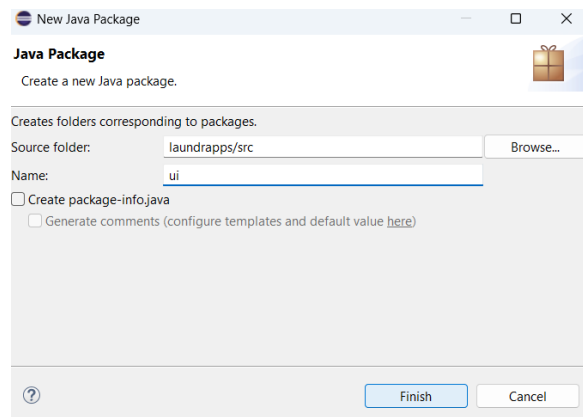
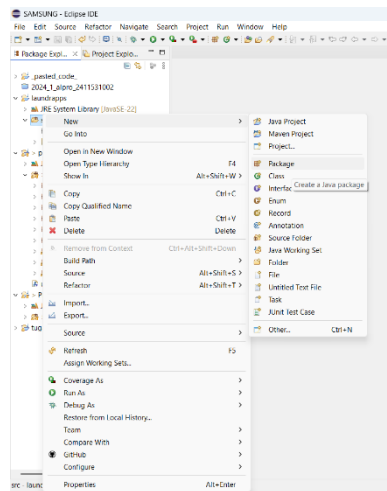
Setelah repository berhasil dibuat, langkah selanjutnya adalah membuat **Java Project** baru pada IDE Eclipse dengan nama laundryapps. Project ini menjadi dasar dari aplikasi yang akan dibangun. Struktur project nantinya akan dibagi menjadi beberapa package agar kode lebih terorganisir



3. Membagi package menjadi 2 yaitu, package model dan package ui

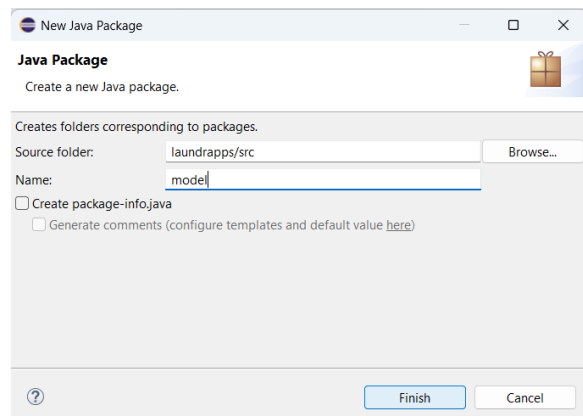
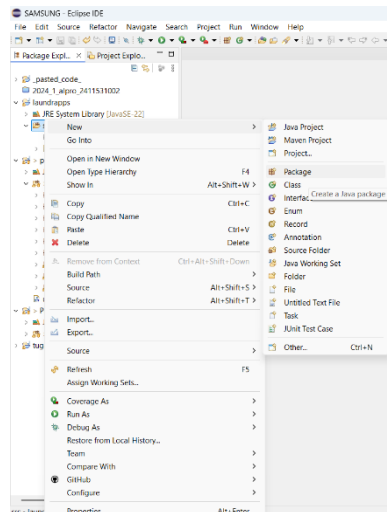
a) Package ui

digunakan untuk menyimpan tampilan antarmuka pengguna berbasis Java Swing, misalnya LoginFrame dan MainFrame. Dengan pemisahan ini, kode menjadi lebih rapi, mudah dipahami, dan mendukung konsep **MVC (Model-View-Controller)**.



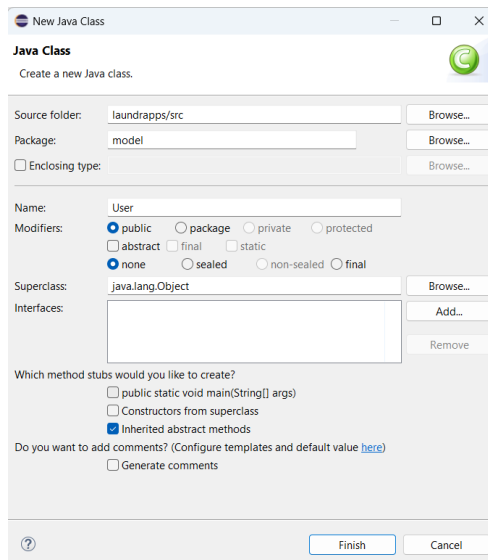
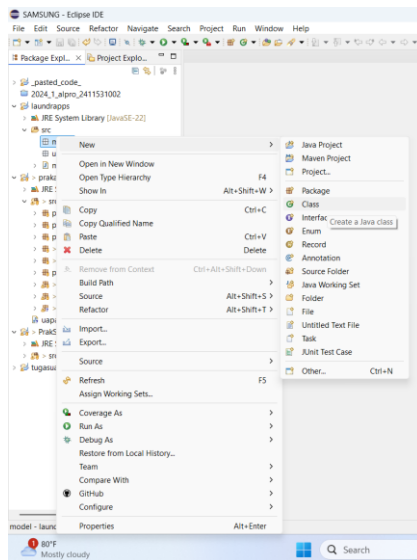
b) Package model

digunakan untuk menyimpan class-class inti seperti User, Costumer, Service, dan Order. Package ini berperan sebagai lapisan logika data (data model).



4. Membuat kelas User pada Package Model.

Pada package model, dibuat sebuah class User yang berfungsi untuk menyimpan data pengguna aplikasi seperti id, nama, username, dan password. Selain itu, pada class ini juga dibuat method login() untuk melakukan validasi username dan password. Jika data login sesuai dengan data yang telah diset pada object, maka proses login dianggap berhasil. Class ini menjadi dasar autentikasi aplikasi.



Penjelasan Syntax

```
package model;

public class User {
    String id, nama, username, password;

    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;
    }

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

        User user = new User();
        user.setId("1");
        user.setNama("Fulan");
        user.setUsername("fulan");
        user.setPassword("12345");

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

Package

- package model;
Mendeklarasikan bahwa kelas *User* berada pada package model. Hal ini memudahkan pengelompokan kode sesuai fungsinya.

Deklarasi Kelas

- public class User {
Mendefinisikan kelas *User* sebagai kelas publik yang dapat diakses oleh kelas lain.

Atribut

- String id, nama, username, password;
Mendeklarasikan empat atribut utama untuk menyimpan data user, yaitu:
 - **id** → identitas unik user
 - **nama** → nama lengkap user
 - **username** → nama pengguna untuk login
 - **password** → kata sandi untuk login

Getter dan Setter

- getId() → mengambil nilai atribut **id**.
- setId(String id) → mengubah nilai atribut **id** sesuai parameter.
- getNama() → mengambil nilai atribut **nama**.
- setNama(String nama) → mengubah nilai atribut **nama**.
- getUsername() → mengambil nilai atribut **username**.
- setUsername(String username) → mengubah nilai atribut **username**.
- getPassword() → mengambil nilai atribut **password**.
- setPassword(String password) → mengubah nilai atribut **password**.

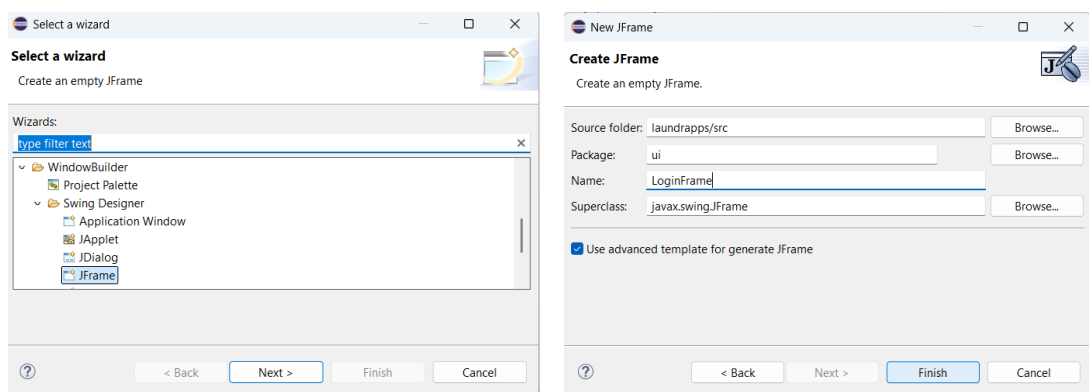
Method Login

- public static boolean login(String username, String password)
 - Method dibuat **static**, sehingga bisa dipanggil tanpa membuat objek baru.
 - Variabel isLogin digunakan sebagai penanda berhasil atau tidaknya proses login.
- User user = new User();
Membuat objek baru bernama **user**.
- user.setId("1");
user.setNama("Fulan");
user.setUsername("fulan");
user.setPassword("12345");
Mengisi data user secara hardcode dengan nilai:
 - id = "1"
 - nama = "Fulan"
 - username = "fulan"
 - password = "12345"
- if (user.getUsername().equalsIgnoreCase(username) && user.getPassword().equalsIgnoreCase(password))
Mengecek apakah username dan password input sama dengan data user.
 - Menggunakan **equalsIgnoreCase()** agar pengecekan tidak sensitif huruf besar/kecil.
 - Jika cocok → isLogin = true.

- Jika tidak cocok → isLogin = false.
- return isLogin;
Mengembalikan nilai **true** atau **false** sebagai hasil login.

5. Membuatkan Kelas LoginFrame pada Package UI

Pada package ui, dibuat sebuah class LoginFrame yang merupakan antarmuka login aplikasi. Tampilan login ini terdiri dari label, text field untuk username, password field, serta tombol login. Warna dan font disesuaikan agar tampilan lebih menarik (soft pink, biru muda, abu-abu). Tombol login memiliki event handler yang memanggil method User.login(). Jika login berhasil, maka aplikasi akan menampilkan MainFrame, sedangkan jika gagal akan muncul pesan peringatan.



Penjelasan Syntax

```
package ui;
import javax.swing.*;

public class LoginFrame extends JFrame {
    private JTextField txtUsername;
    private JPasswordField txtPassword;
    private JButton btnLogin;

    public LoginFrame() {
        setTitle("Laundry Apps - Login");
        setSize(400, 300);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setLayout(null);
        setLocationRelativeTo(null);

        // Warna background
        getContentPane().setBackground(new Color(255, 182, 193)); // soft pink

        // Judul aplikasi
        JLabel lblTitle = new JLabel("Laundry Apps", SwingConstants.CENTER);
        lblTitle.setFont(new Font("Segoe UI", Font.ITALIC, 12));
        lblTitle.setForeground(new Color(219, 112, 147)); // pink elegan
        lblTitle.setBounds(100, 20, 200, 30);
        add(lblTitle);

        JLabel lblSubtitle = new JLabel("Males aja nyuci, biar kami cuciin", SwingConstants.CENTER);
        lblSubtitle.setFont(new Font("Segoe UI", Font.ITALIC, 12));
        lblSubtitle.setForeground(new Color(185, 185, 185)); // abu-abu
        lblSubtitle.setBounds(70, 50, 260, 20);
        add(lblSubtitle);

        // Username
        JLabel lblUser = new JLabel("Username:");
        lblUser.setBounds(50, 100, 100, 25);
        add(lblUser);

        txtUsername = new JTextField();
        txtUsername.setName("txtUsername");
        txtUsername.setBounds(150, 100, 180, 25);
        add(txtUsername);
    }
}
```



```

// Password
JLabel lblPass = new JLabel("Password:");
lblPass.setBounds(50, 140, 100, 25);
add(lblPass);

txtPassword = new JPasswordField();
txtPassword.setName("txtPassword");
txtPassword.setBounds(150, 140, 180, 25);
add(txtPassword);

// Tombol Login
btnLogin = new JButton("Login");
btnLogin.setName("btnLogin");
btnLogin.setBounds(150, 190, 100, 30);
btnLogin.setBackground(new Color(135, 206, 250)); // biru muda
btnLogin.setForeground(Color.WHITE);
btnLogin.setFocusPainted(false);
btnLogin.setFont(new Font("Segoe UI", Font.BOLD, 14));
add(btnLogin);

// Event sesuai modul
btnLogin.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String username = txtUsername.getText();
        String password = new String(txtPassword.getPassword());

        if (User.Login(username, password)) {
            new MainFrame().setVisible(true);
            dispose();
        } else {
            JOptionPane.showMessageDialog(null, "Login Gagal!");
        }
    }
});
}

public static void main(String[] args) {
    new LoginFrame().setVisible(true);
}
}

```

Package dan Import

- package ui;
Menyatakan bahwa kelas LoginFrame ada pada package ui, khusus untuk tampilan antarmuka aplikasi.
- import javax.swing.*;
Mengimpor library Swing untuk membuat GUI.
- import java.awt.*; import java.awt.event.*;
Mengimpor library AWT untuk pengaturan warna, font, tata letak, serta event (aksi tombol).
- import model.User;
Mengimpor kelas User dari package model, agar fungsi login bisa digunakan di sini.

Deklarasi Kelas dan Komponen

```

public class LoginFrame extends JFrame {
    private JTextField txtUsername;
    private JPasswordField txtPassword;
    private JButton btnLogin;

```

- LoginFrame adalah kelas turunan (extends) dari JFrame → berfungsi sebagai jendela utama login.
- txtUsername → field input untuk username.
- txtPassword → field input password dengan karakter tersembunyi.
- btnLogin → tombol untuk login.

Konstruktur LoginFrame()

- Pengaturan jendela
 - setTitle("Laundry Apps - Login"); → memberi judul aplikasi.
 - setSize(400, 300); → ukuran jendela 400x300 piksel.

- `setDefaultCloseOperation(EXIT_ON_CLOSE);` → menutup aplikasi jika jendela ditutup.
 - `setLayout(null);` → layout manual (posisi diatur dengan `setBounds`).
 - `setLocationRelativeTo(null);` → jendela muncul di tengah layar.
- Desain GUI
 - `getContentPane().setBackground(new Color(255,182,193));` → warna latar belakang soft pink.
 - Judul "Laundry Apps" ditampilkan besar, tebal, warna pink elegan.
 - Subjudul "Males aja nyuci, biar kami cuciin" ditampilkan kecil, miring, abu-abu.

Input Username dan Password

- Username
Label "Username:" dan `txtUsername` (input teks).
- Password
Label "Password:" dan `txtPassword` (input password dengan teks tersembunyi).

Tombol Login

```
btnLogin = new JButton("Login");
btnLogin.setBounds(150, 190, 100, 30);
btnLogin.setBackground(new Color(135, 206, 250));
btnLogin.setForeground(Color.WHITE);
btnLogin.setFont(new Font("Segoe UI", Font.BOLD, 14));
```

- Membuat tombol Login ukuran 100x30.
- Warna biru muda, teks putih, dan font tebal.

Event Handling Tombol Login

```
btnLogin.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String username = txtUsername.getText();
        String password = new String(txtPassword.getPassword());

        if (User.login(username, password)) {
            new MainFrame().setVisible(true);
            dispose();
        } else {
            JOptionPane.showMessageDialog(null, "Login Gagal!");
        }
    }
});
```

- Saat tombol ditekan:
 - Ambil input username dan password.

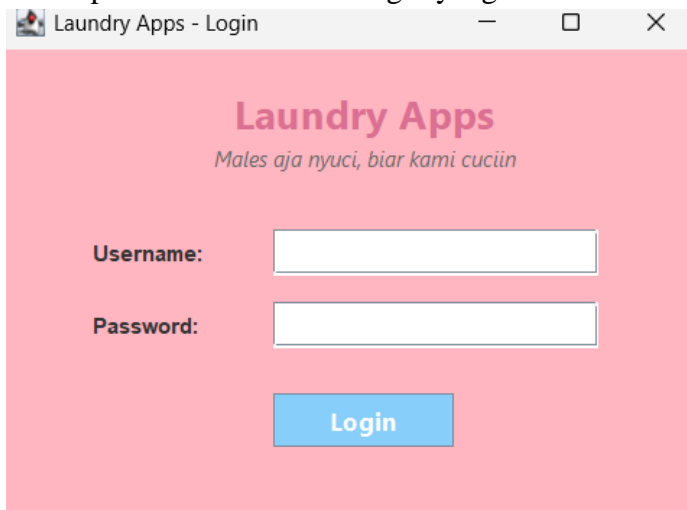
- Cek dengan `User.login()`.
- Jika berhasil → buka `MainFrame`, tutup `LoginFrame`.
- Jika gagal → muncul pesan "**Login Gagal!**".

Method Main

```
public static void main(String[] args) {
    new LoginFrame().setVisible(true);
}
```

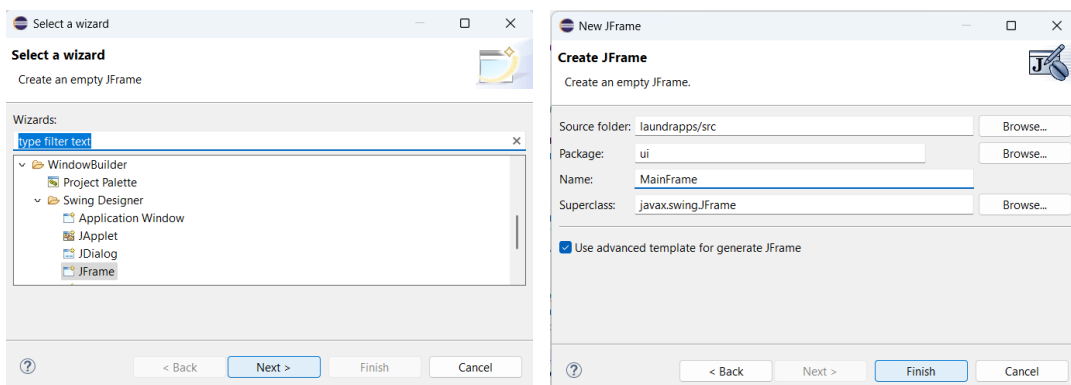
Menjalankan aplikasi dengan menampilkan jendela login.

Tampilan GUI Halaman Login yang telah di edit.



6. Membuat Kelas `MainFrame` pada Package `UI`

`MainFrame` merupakan halaman utama aplikasi Laundry setelah login berhasil. Pada halaman ini terdapat beberapa tombol menu seperti **Pesanan, Layanan, Pelanggan, Pengguna, Laporan, Profile, Dan Keluar**. Tampilan ini menggunakan `GridLayout` agar tombol tersusun rapi. Tombol diberi warna (soft blue dan hot pink) agar lebih menarik. Tombol keluar difungsikan untuk menutup aplikasi. Halaman utama ini menjadi dasar navigasi ke menu-menu lain yang dapat dikembangkan lebih lanjut.



Penjelasan Syntax:

```

package ui;

import javax.swing.*;

public class MainFrame extends javax.swing.JFrame {

    public MainFrame() {
        initComponents();
        styleButtons(); // styling tombol
        setLocationRelativeTo(null); // center
    }

    @SuppressWarnings("unchecked")
    private void initComponents() {

        mainPanel = new javax.swing.JPanel();
        titleLabel = new javax.swing.JLabel();
        buttonPanel = new javax.swing.JPanel();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("Laundry Apps - Main");

        mainPanel.setBackground(new java.awt.Color(255, 182, 193)); // pink
        mainPanel.setLayout(new java.awt.BorderLayout());

        titleLabel.setFont(new java.awt.Font("Segoe UI", 1, 30));
        titleLabel.setForeground(new java.awt.Color(219, 112, 147));
        titleLabel.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
        titleLabel.setText("Laundry Apps");
        titleLabel.setBorder(javax.swing.BorderFactory.createEmptyBorder(20, 0, 20, 0));
        mainPanel.add(titleLabel, java.awt.BorderLayout.NORTH);

        buttonPanel.setBackground(new java.awt.Color(255, 182, 193));
        buttonPanel.setLayout(new java.awt.GridLayout(3, 3, 15, 15));
        buttonPanel.setBorder(javax.swing.BorderFactory.createEmptyBorder(20, 50, 20, 50));

        btnPesanan = new JButton("PESANAN");
        btnLayanan = new JButton("LAYANAN");
        btnPelanggan = new JButton("PELANGGAN");
        btnPengguna = new JButton("PENGGUNA");
        btnLaporan = new JButton("LAPORAN");

        btnLaporan = new JButton("LAPORAN");
        btnProfile = new JButton("PROFILE");
        btnKeluar = new JButton("KELUAR");

        buttonPanel.add(btnPesanan);
        buttonPanel.add(btnLayanan);
        buttonPanel.add(btnPelanggan);
        buttonPanel.add(btnPengguna);
        buttonPanel.add(btnLaporan);
        buttonPanel.add(btnProfile);
        buttonPanel.add(new JLabel());
        buttonPanel.add(btnKeluar);

        mainPanel.add(buttonPanel, java.awt.BorderLayout.CENTER);

        getContentPane().add(mainPanel);
        pack();
    }

    private void styleButtons() {
        Color softBlue = new Color(135, 206, 250);
        Color hotPink = new Color(255, 105, 180);

        JButton[] blueButtons = {btnPesanan, btnLayanan, btnPelanggan,
            btnPengguna, btnLaporan, btnProfile};

        for (JButton btn : blueButtons) {
            btn.setBackground(softBlue);
            btn.setForeground(Color.WHITE);
            btn.setFont(new Font("Segoe UI", Font.BOLD, 16));
            btn.setFocusPainted(false);
            btn.setOpaque(true);
            btn.setBorderPainted(false);
        }

        btnKeluar.setBackground(hotPink);
        btnKeluar.setForeground(Color.WHITE);
        btnKeluar.setFont(new Font("Segoe UI", Font.BOLD, 16));
        btnKeluar.setFocusPainted(false);
        btnKeluar.setOpaque(true);
        btnKeluar.setBorderPainted(false);

        btnKeluar.addActionListener(e -> System.exit(0));
    }

    // ◦ Tambahkan main()
    public static void main(String[] args) {
        try {
            UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
        } catch (Exception e) {
            e.printStackTrace();
        }

        java.awt.EventQueue.invokeLater(() -> {
            new MainFrame().setVisible(true);
        });
    }

    // Variables
    private javax.swing.JPanel mainPanel;
    private javax.swing.JLabel titleLabel;
    private javax.swing.JPanel buttonPanel;

    private javax.swing.JButton btnPesanan;
    private javax.swing.JButton btnLayanan;
    private javax.swing.JButton btnPelanggan;
    private javax.swing.JButton btnPengguna;
    private javax.swing.JButton btnLaporan;
    private javax.swing.JButton btnProfile;
    private javax.swing.JButton btnKeluar;
}

```

Package

- `package ui;`
Menandakan bahwa kelas `MainFrame` berada pada package `ui`, yang berfungsi untuk menampung tampilan antarmuka (GUI) aplikasi.

Import

- `import javax.swing.*;`
Mengimpor library Swing untuk membuat GUI.
- `import java.awt.*;`
Mengimpor library AWT untuk mengatur warna, font, dan layout.

Deklarasi Kelas

- `public class MainFrame extends javax.swing.JFrame`
Kelas `MainFrame` merupakan turunan dari `JFrame`, sehingga berfungsi sebagai jendela utama aplikasi setelah login berhasil.

Konstruktor `MainFrame()`

- `initComponents();`
Memanggil method untuk menginisialisasi dan menambahkan komponen GUI.
- `styleButtons();`
Memanggil method untuk mengatur desain tombol agar lebih menarik.
- `setLocationRelativeTo(null);`
Membuat jendela aplikasi tampil di posisi tengah layar.

Method `initComponents()`

Berisi pengaturan awal semua komponen tampilan.

1. **Panel Utama (`mainPanel`)**

- Background diberi warna pink lembut (`new Color(255, 182, 193)`).
- Menggunakan `BorderLayout` untuk membagi tampilan menjadi bagian atas (judul) dan tengah (tombol).

2. **Judul Aplikasi (`titleLabel`)**

- Teks "Laundry Apps" dengan font Segoe UI Bold ukuran 30.
- Warna teks pink elegan (`new Color(219, 112, 147)`).
- Posisi teks di tengah (`CENTER`).
- Diletakkan di bagian atas jendela (`BorderLayout.NORTH`).

3. **Panel Tombol (`buttonPanel`)**

- Latar belakang pink lembut.
- Layout `GridLayout(3, 3, 15, 15)` → membagi panel menjadi grid 3 baris 3 kolom dengan jarak antar tombol 15 piksel.
- Menggunakan `EmptyBorder` agar tombol tidak menempel pada pinggir.

4. **Tombol-tombol Menu**

- Dibuat beberapa tombol utama:
 - `btnPesanan` → untuk menu pesanan.
 - `btnLayanan` → untuk menu layanan.

- btnPelanggan → untuk menu pelanggan.
- btnPengguna → untuk menu pengguna.
- btnLaporan → untuk menu laporan.
- btnProfile → untuk menu profil.
- btnKeluar → untuk keluar dari aplikasi.
- Semua tombol ditambahkan ke panel tombol sesuai urutan.

Method styleButtons()

Berfungsi untuk memberikan gaya pada tombol agar lebih menarik.

- **Warna dan Font**

- Tombol utama (Pesanan, Layanan, Pelanggan, Pengguna, Laporan, Profile) diberi warna biru muda dengan teks putih dan font tebal ukuran 16.
- Tombol Keluar diberi warna pink terang (Hot Pink) agar berbeda dan mudah dikenali.

- **Desain**

- Tombol dibuat tidak menampilkan border (setBorderPainted(false)) dan fokus tersembunyi (setFocusPainted(false)), sehingga tampil lebih modern.

- **Aksi Tombol Keluar**

- btnKeluar.addActionListener(e → System.exit(0));
Jika tombol Keluar ditekan, maka aplikasi langsung berhenti.

Method main()

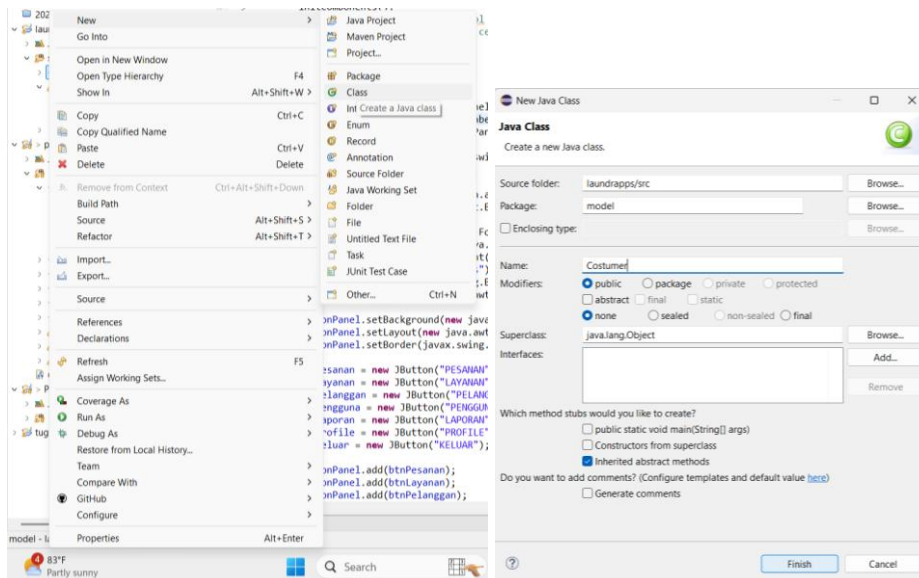
- Digunakan untuk menjalankan aplikasi secara langsung.
- **UIManager.setLookAndFeel()** → menyesuaikan tampilan GUI agar mengikuti sistem operasi (native look).
- **EventQueue.invokeLater(...)** → menjalankan MainFrame agar thread GUI lebih aman.

Tampilan Halaman Main yang telah diedit



Tugas

Membuat Kelas Costumer



- Buatlah class dengan nama Costumer dengan attribute id, nama, Alamat dan nomor hp, buatkan setter dan getter untuk menambahkan object Costumer

```
package model;

public class Costumer {
    private String id;
    private String nama;
    private String alamat;
    private String nomorHp;

    // Getter & Setter
    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 getAlamat() {
        return alamat;
    }
    public void setAlamat(String alamat) {
        this.alamat = alamat;
    }

    public String getNomorHp() {
        return nomorHp;
    }
    public void setNomorHp(String nomorHp) {
        this.nomorHp = nomorHp;
    }
}
```

Kelas **Costumer** merupakan salah satu kelas pada package model yang berfungsi untuk merepresentasikan data pelanggan dalam aplikasi Laundry. Kelas ini memiliki empat atribut utama, yaitu **id**, **nama**, **alamat**, dan **nomorHp** yang semuanya bertipe data String. Atribut **id** digunakan untuk menyimpan identitas unik pelanggan, **nama** menyimpan nama pelanggan, **alamat** digunakan untuk menyimpan informasi domisili, sedangkan **nomorHp** menyimpan nomor handphone pelanggan yang dapat dihubungi.

Untuk mengakses dan memodifikasi setiap atribut, kelas ini dilengkapi dengan metode **getter** dan **setter**. Getter berfungsi mengambil nilai dari atribut, sedangkan setter berfungsi mengubah atau memberikan nilai baru pada atribut tersebut. Misalnya, `getNama()` digunakan untuk mengambil nama pelanggan, sedangkan `setNama(String nama)` digunakan untuk mengubah atau menetapkan nama pelanggan. Dengan penerapan getter dan setter ini, konsep **enkapsulasi** dalam pemrograman berorientasi objek dapat tercapai, sehingga data pelanggan tetap terjaga dan hanya bisa diakses melalui metode yang disediakan.

Secara keseluruhan, kelas **Costumer** berperan penting dalam menyimpan dan mengelola data pelanggan yang nantinya akan digunakan dalam berbagai proses transaksi maupun layanan pada sistem Laundry Apps.

E. KESIMPULAN

Berdasarkan praktikum yang telah dilakukan, dapat disimpulkan bahwa penerapan konsep **Pemrograman Berorientasi Objek (PBO)** melalui pembuatan class, object, atribut, method, serta penggunaan getter dan setter mampu membantu mahasiswa memahami dasar-dasar pengelolaan data dalam sebuah aplikasi. Dengan membagi project menjadi package model dan ui, kode menjadi lebih terstruktur, rapi, serta mudah untuk dikembangkan.

Selain itu, implementasi antarmuka grafis menggunakan **Java Swing** memberikan pengalaman langsung dalam membangun aplikasi desktop sederhana yang interaktif. Aplikasi LaundryApps berhasil dibuat dengan fitur login, navigasi menu utama, serta pengelolaan data pelanggan melalui class Costumer. Hal ini membuktikan bahwa mahasiswa tidak hanya memahami teori, tetapi juga mampu mengaplikasikannya dalam bentuk program nyata.