

LAPORAN PRATIKUM

“PEKAN 1”

Disusun Untuk Memenuhi Tugas Mata Kuliah PBO

DOSEN PENGAMPU:

Nurfiah, S.ST. M.Kom.



DISUSUN OLEH:

Karimah Irsyadiyah (2411533018)

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
T.A 2024/2025**

Daftar Pustaka

BAB I PENDAHULUAN.....	3
1.1 Latar Belakang.....	3
1.2 Tujuan	3
1.3 Alat dan Bahan.....	4
BAB II PEMBAHASAN	5
2.1 Langkah-langkah Praktikum dan Pembahasan Program	5
BAB III PENUTUP	19
3.1 Kesimpulan	19

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pemrograman Berorientasi Objek (PBO) merupakan paradigma pemrograman yang memodelkan program berdasarkan objek dari dunia nyata. Setiap objek memiliki atribut (data) dan method (fungsi) yang memungkinkan interaksi antar objek. Paradigma ini sangat penting dipahami karena banyak bahasa pemrograman modern, termasuk Java, menerapkannya sebagai dasar.

Dalam praktikum ini dibuat sebuah aplikasi sederhana bertema **Laundry Apps** menggunakan bahasa pemrograman Java. Aplikasi ini mengimplementasikan konsep class dan object untuk memodelkan tiga entitas utama: Costumer, Service, dan Order. Masing-masing entitas dibuat sebagai class dengan atribut serta setter dan getter. Untuk memudahkan interaksi dengan pengguna, program dilengkapi dengan antarmuka grafis (GUI) berbasis Swing menggunakan JTabbedPane, sehingga setiap tab mewakili satu entitas.

Pembuatan aplikasi Laundry Apps ini bertujuan untuk melatih pemahaman konsep dasar OOP, meliputi pembuatan class, object, penggunaan setter-getter, serta penerapan pada sebuah program nyata dengan tampilan GUI.

1.2 Tujuan

- Memahami konsep dasar class, atribut, method, dan object pada Java.
- Mengimplementasikan setter dan getter untuk mengelola data objek.
- Mendesain aplikasi sederhana Laundry Apps berbasis GUI dengan Swing.
- Melatih keterampilan pemodelan entitas nyata (customer, service, order) ke dalam bentuk class dalam Java.
- Memberikan pengalaman dalam menghubungkan logika program (OOP) dengan antarmuka pengguna (GUI).

1.3 Alat dan Bahan

Alat dan bahan yang digunakan dalam praktikum ini adalah:

- **Perangkat Keras (Hardware):**
 - Laptop/PC dengan sistem operasi Windows/Linux/macOS.
 - RAM minimal 4 GB.
- **Perangkat Lunak (Software):**
 - Java Development Kit (JDK) versi 8 atau lebih baru.
 - IDE Eclipse untuk coding dan desain GUI.
 - Library bawaan Java Swing dan AWT untuk membangun GUI.

BAB II

PEMBAHASAN

2.1 Langkah-langkah Praktikum dan Pembahasan Program

Langkah-langkah Praktikum (ringkasan yang diikuti saat membuat Laundry Apps):

1. Buat project baru di Eclipse (nama bebas, mis. LaundryApps), lalu buat satu package bernama PEKAN1.
2. Di package PEKAN1 buat 4 file/class: Costumer.java, Service.java, Order.java, dan MainApp.java.
3. Pada masing-masing class model (Costumer, Service, Order) deklarasikan atribut sesuai soal lalu buat getter dan setter menggunakan shortcut Eclipse (Source → Generate Getters and Setters).
4. Implementasikan GUI pada MainApp.java menggunakan Swing (extend JFrame) dengan layout berbasis JTabbedPane — satu tab untuk Customer, satu tab untuk Service, satu tab untuk Order.
5. Tambahkan komponen input (JTextField, JLabel) dan kontrol (JButton) pada setiap tab.
6. Tambahkan event listeners (ActionListener) untuk tombol Simpan dan Reset pada tiap tab:
 - **Simpan:** baca nilai dari JTextField, buat instance objek model (mis. Costumer) lalu tampilkan/cek di console dan tampilkan JOptionPane konfirmasi.
 - **Reset:** kosongkan semua JTextField pada tab tersebut.

A. Class : Costumer.java

a) Tujuan Program:

Merepresentasikan entitas pelanggan (customer) pada sistem laundry. Class ini hanya berfungsi sebagai *model* data dengan atribut dan method accessor/mutator untuk akses data.

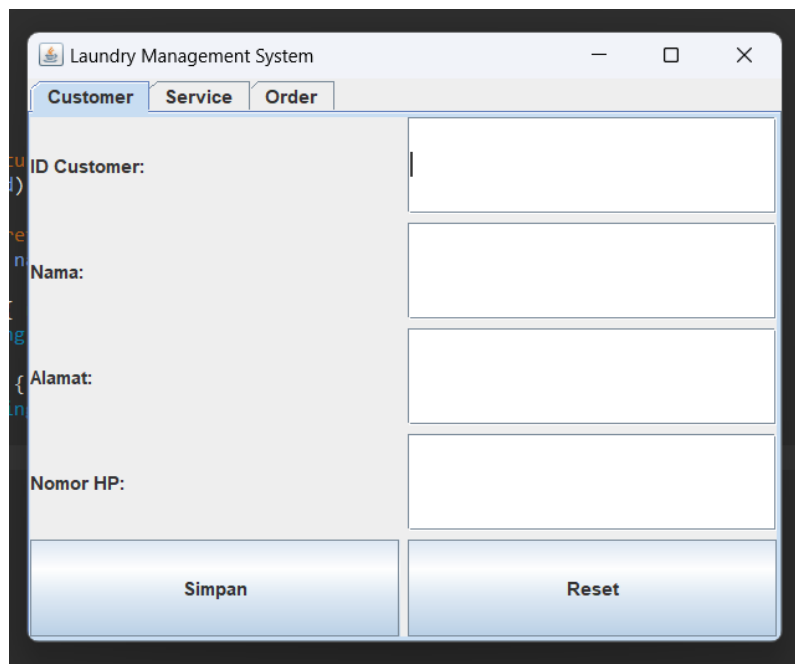
b) Struktur dan Penjelasan Komponen Program:

1. package PEKAN1;
 - file berada dalam package PEKAN1.

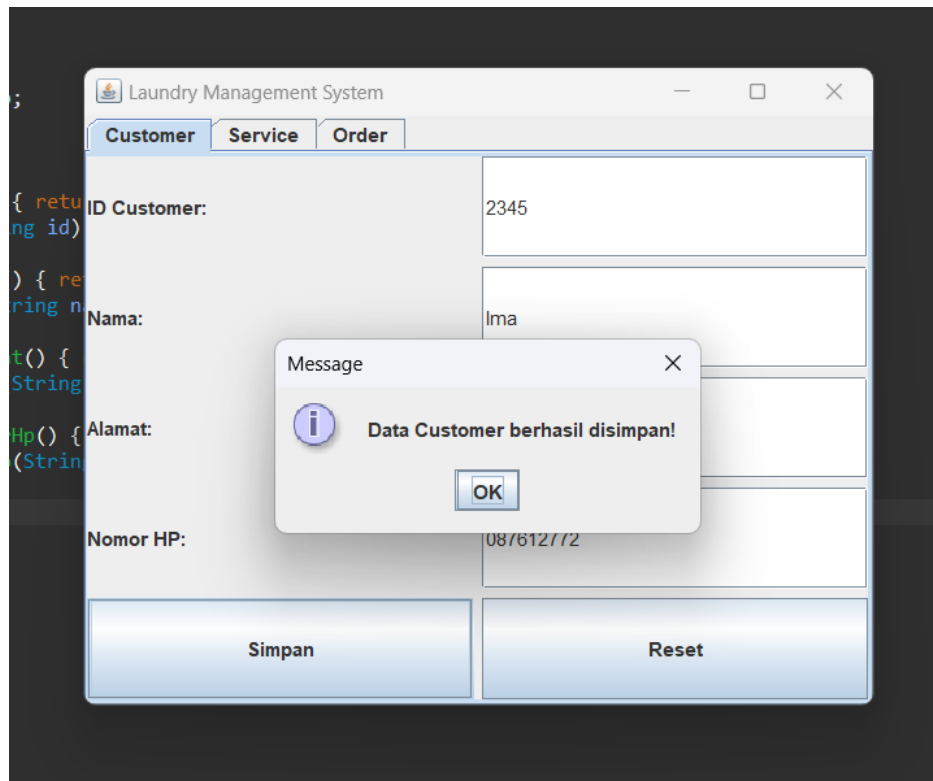
2. `public class Costumer`
 - Deklarasi class model Costumer.
3. JFrame Atribut (private):
 - `private String id` menyimpan ID customer.
 - `private String nama` menyimpan nama customer.
 - `private String Alamat` menyimpan alamat customer.
 - `private String nomorHp` menyimpan nomor HP customer.
4. Constructor:
 - `getId(), setId(String id)`
 - `getNama(), setNama(String nama)`
 - `getAlamat(), setAlamat(String alamat)`
 - `getNomorHp(), setNomorHp(String nomorHp)`
5. Getter & Setter (accessor & mutator):
 - Objek Costumer dibuat ketika pengguna menekan tombol **Simpan** pada tab Customer; data diisi dari JTextField ke objek melalui setter.

c) Fitur Utama Program:

- Menyimpan atribut pelanggan dalam bentuk objek.
- Memudahkan pertukaran data antara GUI dan logika (mis. disimpan ke DB di masa depan).



The screenshot displays a Java Swing window titled "Laundry Management System". It features three tabs: "Customer", "Service", and "Order", with "Customer" currently selected. The form contains four labeled text input fields: "ID Customer:", "Nama:", "Alamat:", and "Nomor HP:". Below these fields are two buttons: "Simpan" (Save) and "Reset". The window has standard OS window controls (minimize, maximize, close) in the top right corner.



```
package PEKAN1;

//Karimah Irsyadiyah
//2411533018

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

    public Costumer() {}

    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; }
}
```

B. Class : Service.java

a) Tujuan Program:

Model untuk menyimpan data layanan laundry (jenis layanan, harga, status).

b) Struktur dan Penjelasan Komponen Program:

1. package PEKAN1;
2. public class Service
3. Atribut (private):
 - private String id; — ID service.
 - private String jenis; — jenis layanan (mis. Cuci Kiloan, Ekspres).
 - private double harga; — harga layanan (tipe double).
 - private String status; — status layanan (aktif/tidak aktif).
4. Constructor: default.
5. Getter & Setter:
 - getId()/setId(...), getJenis()/setJenis(...), getHarga()/setHarga(...), getStatus()/setStatus(...)
6. Kegunaan:
 - Dibuat waktu tekan **Simpan** pada tab Service; data dipakai saat membuat Order (mengacu ke id_service).

c) Fitur Utama Program:

- Menyimpan informasi service dan harga.
- Memudahkan kalkulasi total order jika dihubungkan ke Order.

Laundry Management System

Customer Service Order

ID Service: 12345

Jenis: kadajajid

Harga: 121212

Status: adadqd

Simpan Reset

Laundry Management System

Customer Service Order

ID Service: 12345

Jenis:

Harga:

Status: adadqd

Simpan Reset

Message

Data Service berhasil disimpan!

OK

```
package PEKAN1;

//Karimah Irsyadiyah
//2411533018

public class Service {
    private String id;
    private String jenis;
    private double harga;
    private String status;

    public Service() {}

    public String getId() { return id; }
    public void setId(String id) { this.id = id; }

    public String getJenis() { return jenis; }
    public void setJenis(String jenis) { this.jenis = jenis; }
```

```

public double getHarga() { return harga; }
public void setHarga(double harga) { this.harga = harga; }

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

```

C. Class : Order.java

a) Tujuan Program:

Model untuk menyimpan data transaksi/pesanan laundry — mengikat Costumer, Service, user pembuat, status, tanggal, dan total biaya.

b) Struktur dan Penjelasan Komponen Program:

1. package PEKAN1;
2. public class Order
3. Atribut (private):
 - private String id; — ID order.
 - private String id_costumer; — referensi ID customer.
 - private String id_service; — referensi ID service.
 - private String id_user; — ID user/operator (jika diperlukan).
 - private double total; — total biaya order.
 - private String tanggal; — tanggal order (format sederhana String, mis. "2025-09-07").
 - private String tanggal_selesai; — tanggal selesai.
 - private String status; — status order (mis. "proses", "selesai").
 - private String status_pembayaran; — status pembayaran (mis. "lunas", "belum lunas").
4. Constructor: default.
5. Getter & Setter: masing-masing atribut memiliki getter & setter.
6. Kegunaan: Saat membuat Order lewat GUI Order tab, baca field dan set pada objek Order. Objek ini bisa disimpan atau ditampilkan di console.

c) Fitur Utama Program:

- Menyimpan informasi lengkap pesanan.
- Menjadi wadah data untuk proses lebih lanjut (laporan, penyimpanan DB, dsb).

Laundry Management System

Customer Service **Order**

ID Order: 121313

ID Customer: 13131

ID Service: 1313

Total: 1313

Tanggal: 12

Simpan Reset

Laundry Management System

Customer Service **Order**

ID Order: 121313

ID Customer: 13131

ID Service: 1313

Total: 1313

Tanggal: 12

Simpan Reset

Message

Data Order berhasil disimpan!

OK

```
package PEKAN1;
//Karimah Irsyadiyah
//2411533018

public class Order {
    private String id;
```

```

private String id_costumer;
private String id_service;
private String id_user;
private double total;
private String tanggal;
private String tanggal_selesai;
private String status;
private String status_pembayaran;

public Order() {}

public String getId() { return id; }
public void setId(String id) { this.id = id; }

public String getId_costumer() { return id_costumer; }
public void setId_costumer(String id_costumer) { this.id_costumer =
id_costumer; }

public String getId_service() { return id_service; }
public void setId_service(String id_service) { this.id_service =
id_service; }

public String getId_user() { return id_user; }
public void setId_user(String id_user) { this.id_user = id_user; }

public double getTotal() { return total; }
public void setTotal(double total) { this.total = total; }

public String getTanggal() { return tanggal; }
public void setTanggal(String tanggal) { this.tanggal = tanggal; }

public String getTanggal_selesai() { return tanggal_selesai; }
public void setTanggal_selesai(String tanggal_selesai) {
this.tanggal_selesai = tanggal_selesai; }

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

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

```

D. Class : MainApp.java (GUI)

a) Tujuan Program:

Membuat antarmuka pengguna (GUI) untuk input dan manajemen data Customer, Service, dan Order menggunakan Swing. GUI memfasilitasi operasi *Simpan* dan *Reset* pada tiap entitas.

b) Struktur dan Penjelasan Komponen Program:

1. package PEKAN1;
 - File berada di package PEKAN1.
2. import javax.swing.*; import java.awt.*; import java.awt.event.*;
 - Mengimpor pustaka Swing dan AWT untuk pembuatan GUI dan event handling.
3. public class MainApp extends JFrame
 - Class utama GUI, turunan dari JFrame.
4. Deklarasi variabel / komponen GUI (contoh nama variabel sesuai kode):
 - JTabbedPane tabbedPane;
 - **Customer tab:** JTextField txtCustId, txtCustNama, txtCustAlamat, txtCustHp;
JButton btnCustSave, btnCustReset;
 - **Service tab:** JTextField txtServId, txtServJenis, txtServHarga, txtServStatus;
JButton btnServSave, btnServReset;
 - **Order tab:** JTextField txtOrderId, txtOrderCust, txtOrderServ, txtOrderTotal, txtOrderTanggal;
JButton btnOrderSave, btnOrderReset;
5. Constructor MainApp()
 - Mengatur title, ukuran frame, default close operation.

- Membuat JTabbedPane lalu tiap tab dibuat sebagai JPanel dengan GridLayout (mis. new GridLayout(5,2,5,5) untuk customer/service; 6,2 untuk order).
- Menambahkan JLabel dan JTextField sesuai atribut tiap entitas.
- Menambahkan tombol **Simpan** dan **Reset** di setiap tab.
- Menambahkan *ActionListener* pada tombol:
 - **btn...Save:**
 - Membaca `getText()` dari `JTextField`.
 - Membuat instance model (`new Costumer()`, `new Service()`, `new Order()`), memanggil setter untuk mengisi data.
 - Menampilkan data di **console** (`System.out.println`) untuk verifikasi.
 - Menampilkan `JOptionPane.showMessageDialog(this, "Data ... berhasil disimpan!")`.
 - **btn...Reset:**
 - Memanggil `setText("")` pada semua `JTextField` tab tersebut untuk mengosongkan form.
- Menambahkan semua tab ke `tabbedPane` dan menambahkan `tabbedPane` ke frame.
- Memanggil `setLocationRelativeTo(null)` untuk memposisikan window di tengah layar.

6. Method `main(String[] args)`

- Menjalankan GUI aman dengan `SwingUtilities.invokeLater(() -> new MainApp().setVisible(true));`

c) **Fitur Utama Program:**

- setup UI (di constructor): menyiapkan layout dan komponen setiap tab.
- event handlers pada tombol Simpan/Reset: menghubungkan UI ke model (membuat objek model dan menampilkan hasil).

- validasi sederhana (jika ditambahkan): memastikan field tidak kosong sebelum menyimpan (pada implementasi sekarang validasi minimal atau belum lengkap).
- entry point aplikasi via main().
- GUI berbasis Swing dengan Tabbed Pane: tab Customer, Service, Order.
- Input data untuk setiap entitas melalui JTextField.
- Tombol Simpan: membuat object model (Costumer/Service/Order), menampilkan konfirmasi, dan mem-print ke console.
- Tombol Reset: membersihkan semua field input di tab terkait.
- Struktur kode OOP: tiap entitas punya class sendiri (memudahkan pengembangan/penyimpanan data ke DB nanti).
- main() yang menjalankan GUI di Event Dispatch Thread (prinsip best practice Swing).

d) Alur kerja ketika user menekan tombol Simpan (contoh Customer):

- User mengisi txtCustId, txtCustNama, txtCustAlamat, txtCustHp.
- Tekan btnCustSave.
- Event listener terbuka: mengambil getText() dari tiap field.
- Membuat Costumer c = new Costumer(); c.setId(...); c.setNama(...); ...
- System.out.println untuk menampilkan data di console (untuk verifikasi praktikum).
- Menampilkan dialog JOptionPane bahwa data tersimpan. (Di masa depan, langkah 5 bisa diganti menyimpan ke database atau file.)

```
package PEKAN1;

//Karimah Irsyadiyah
//2411533018

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class MainApp extends JFrame {
    public MainApp() {
        setTitle("Laundry Management System");
        setSize(500, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JTabbedPane tabbedPane = new JTabbedPane();
```

```

// CUSTOMER
JTextField txtCustId = new JTextField();
JTextField txtCustNama = new JTextField();
JTextField txtCustAlamat = new JTextField();
JTextField txtCustHp = new JTextField();

JButton btnCustSave = new JButton("Simpan");
JButton btnCustReset = new JButton("Reset");

JPanel panelCustomer = new JPanel(new GridLayout(5, 2, 5, 5));
panelCustomer.add(new JLabel("ID Customer:"));
panelCustomer.add(txtCustId);
panelCustomer.add(new JLabel("Nama:"));
panelCustomer.add(txtCustNama);
panelCustomer.add(new JLabel("Alamat:"));
panelCustomer.add(txtCustAlamat);
panelCustomer.add(new JLabel("Nomor HP:"));
panelCustomer.add(txtCustHp);
panelCustomer.add(btnCustSave);
panelCustomer.add(btnCustReset);

// Event tombol Customer
btnCustSave.addActionListener(e -> {
    System.out.println("=== Data Customer ===");
    System.out.println("ID: " + txtCustId.getText());
    System.out.println("Nama: " + txtCustNama.getText());
    System.out.println("Alamat: " + txtCustAlamat.getText());
    System.out.println("Nomor HP: " + txtCustHp.getText());
    JOptionPane.showMessageDialog(this, "Data Customer berhasil
disimpan!");
});

btnCustReset.addActionListener(e -> {
    txtCustId.setText("");
    txtCustNama.setText("");
    txtCustAlamat.setText("");
    txtCustHp.setText("");
});

//SERVICE
JTextField txtServId = new JTextField();
JTextField txtServJenis = new JTextField();
JTextField txtServHarga = new JTextField();
JTextField txtServStatus = new JTextField();

JButton btnServSave = new JButton("Simpan");
JButton btnServReset = new JButton("Reset");

```



```

JPanel panelService = new JPanel(new GridLayout(5, 2, 5, 5));
panelService.add(new JLabel("ID Service:"));
panelService.add(txtServId);
panelService.add(new JLabel("Jenis:"));
panelService.add(txtServJenis);
panelService.add(new JLabel("Harga:"));
panelService.add(txtServHarga);
panelService.add(new JLabel("Status:"));
panelService.add(txtServStatus);
panelService.add(btnServSave);
panelService.add(btnServReset);

// Event tombol Service
btnServSave.addActionListener(e -> {
    System.out.println("=== Data Service ===");
    System.out.println("ID: " + txtServId.getText());
    System.out.println("Jenis: " + txtServJenis.getText());
    System.out.println("Harga: " + txtServHarga.getText());
    System.out.println("Status: " + txtServStatus.getText());
    JOptionPane.showMessageDialog(this, "Data Service berhasil
disimpan!");
});

btnServReset.addActionListener(e -> {
    txtServId.setText("");
    txtServJenis.setText("");
    txtServHarga.setText("");
    txtServStatus.setText("");
});

// ORDER
JTextField txtOrderId = new JTextField();
JTextField txtOrderCust = new JTextField();
JTextField txtOrderServ = new JTextField();
JTextField txtOrderTotal = new JTextField();
JTextField txtOrderTanggal = new JTextField();

JButton btnOrderSave = new JButton("Simpan");
JButton btnOrderReset = new JButton("Reset");

JPanel panelOrder = new JPanel(new GridLayout(6, 2, 5, 5));
panelOrder.add(new JLabel("ID Order:"));
panelOrder.add(txtOrderId);
panelOrder.add(new JLabel("ID Customer:"));
panelOrder.add(txtOrderCust);
panelOrder.add(new JLabel("ID Service:"));
panelOrder.add(txtOrderServ);

```

```

        panelOrder.add(new JLabel("Total:"));
        panelOrder.add(txtOrderTotal);
        panelOrder.add(new JLabel("Tanggal:"));
        panelOrder.add(txtOrderTanggal);
        panelOrder.add(btnOrderSave);
        panelOrder.add(btnOrderReset);

        // Event tombol Order
        btnOrderSave.addActionListener(e -> {
            System.out.println("=== Data Order ===");
            System.out.println("ID: " + txtOrderId.getText());
            System.out.println("ID Customer: " +
txtOrderCust.getText());
            System.out.println("ID Service: " +
txtOrderServ.getText());
            System.out.println("Total: " + txtOrderTotal.getText());
            System.out.println("Tanggal: " +
txtOrderTanggal.getText());
            JOptionPane.showMessageDialog(this, "Data Order berhasil
disimpan!");
        });

        btnOrderReset.addActionListener(e -> {
            txtOrderId.setText("");
            txtOrderCust.setText("");
            txtOrderServ.setText("");
            txtOrderTotal.setText("");
            txtOrderTanggal.setText("");
        });

        // TAMBAH TAB
        tabbedPane.addTab("Customer", panelCustomer);
        tabbedPane.addTab("Service", panelService);
        tabbedPane.addTab("Order", panelOrder);

        add(tabbedPane);
        setLocationRelativeTo(null);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            new MainApp().setVisible(true);
        });
    }
}

```

BAB III

PENUTUP

3.1 Kesimpulan

Berdasarkan praktikum yang telah dilakukan, dapat disimpulkan bahwa:

1. Pemrograman Berorientasi Objek (PBO) dapat diaplikasikan secara nyata dalam pembuatan aplikasi sederhana, seperti Laundry Apps.
2. Pembuatan class dengan atribut dan method setter-getter memungkinkan data disimpan dalam bentuk object yang lebih terstruktur dan mudah diakses.
3. Implementasi GUI menggunakan Java Swing dengan JTabbedPane memberikan antarmuka yang lebih terorganisir, di mana data Customer, Service, dan Order dapat dikelola pada tab masing-masing.
4. Tombol **Simpan** pada GUI berhasil menghubungkan input dari pengguna dengan pembuatan object (Customer, Service, dan Order), sementara tombol **Reset** memudahkan penghapusan data input.
5. Praktikum ini memberikan pemahaman menyeluruh tentang bagaimana konsep OOP tidak hanya dipelajari secara teori, tetapi juga dapat diintegrasikan dengan tampilan antarmuka pengguna untuk aplikasi sehari-hari.