

# LAPORAN PRAKTIKUM PBO

## PEKAN 1



Oleh :

ARYA PRATAMA HENDRI

NIM 2411533008

MATA KULIAH PBO

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

FAKULTAS TEKNOLOGI INFORMASI

DEPARTEMEN INFORMATIKA

UNIVERSITAS ANDALAS

PADANG, SEPTEMBER 2025

## 1. Tujuan

Tujuan praktikum ini adalah agar mahasiswa mampu:

1. Membuat tabel user pada database MySQL.
  2. Membuat koneksi Java dengan database MySQL.
  3. Membuat tampilan GUI CRUD user menggunakan JFrame.
  4. Membuat dan mengimplementasikan **interface** dalam Java.
  5. Membuat fungsi **DAO (Data Access Object)** dan menggunakannya pada aplikasi.
  6. Mengimplementasikan fungsi **CRUD** (Create, Read, Update, Delete) dengan konsep Pemrograman Berorientasi Objek (OOP).
- 

## 2. Alat dan Bahan

- Laptop/komputer dengan **JDK** dan **Eclipse IDE**.
  - **XAMPP** (Apache + MySQL).
  - **MySQL Connector/J** (driver untuk koneksi Java ke MySQL).
  - Database **phpMyAdmin**.
- 

## 3. Dasar Teori

1. **XAMPP**  
Merupakan paket software open source yang terdiri dari Apache, MySQL, PHP, dan Perl. Digunakan untuk development berbasis localhost.
2. **MySQL**  
Sistem manajemen database relasional (RDBMS) open source yang digunakan untuk menyimpan, mengelola, dan mengambil data dalam bentuk tabel.
3. **MySQL Connector/J**  
Driver yang menghubungkan aplikasi Java dengan MySQL. Berfungsi membuka koneksi, mengirimkan query SQL, menerima hasil, dan menutup koneksi.
4. **DAO (Data Access Object)**  
Objek yang menyediakan antarmuka abstrak terhadap method database.  
Tujuannya:
  - Memisahkan logika akses data dengan logika bisnis.
  - Memudahkan pemeliharaan dan meningkatkan reusabilitas.
5. **Interface Java**  
Kumpulan method abstrak yang wajib diimplementasikan oleh kelas yang menggunakannya.

## 6. CRUD

Operasi dasar dalam aplikasi database:

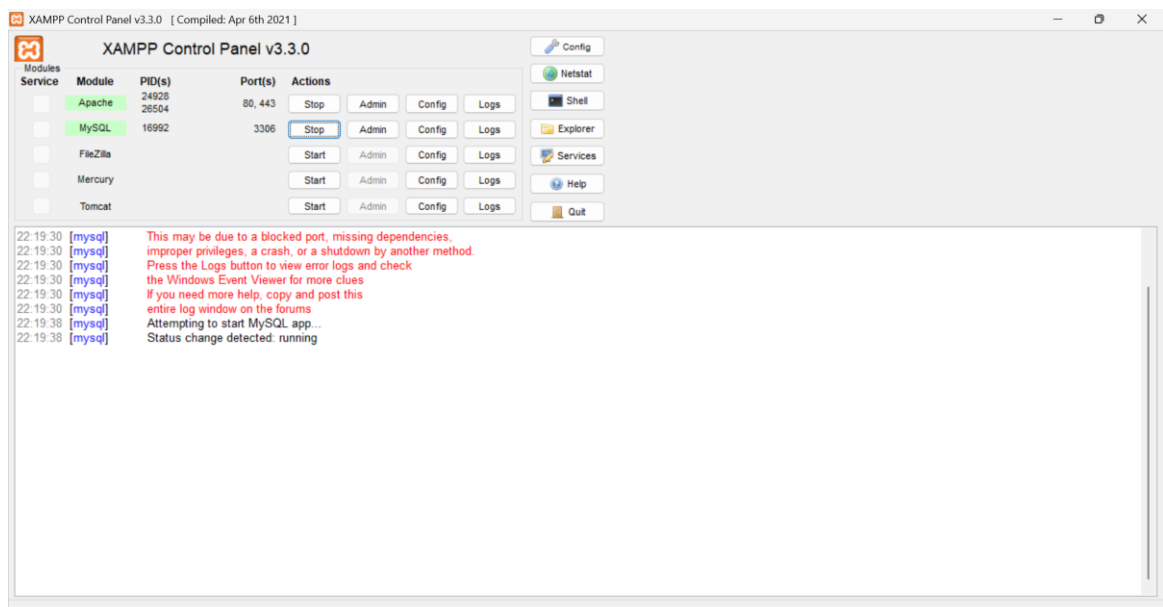
- Create → menambahkan data baru.
- Read → menampilkan data.
- Update → mengubah data.
- Delete → menghapus data.

---

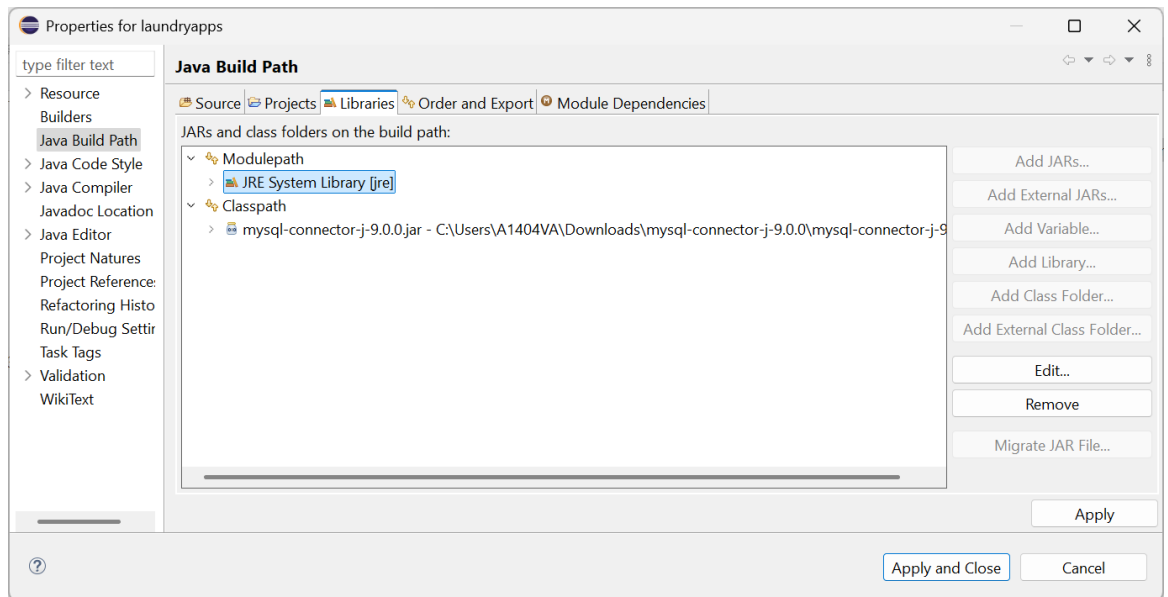
## 4. Langkah Kerja

### 4.1 Instalasi dan Persiapan

1. Install **XAMPP** dan jalankan Apache serta MySQL.

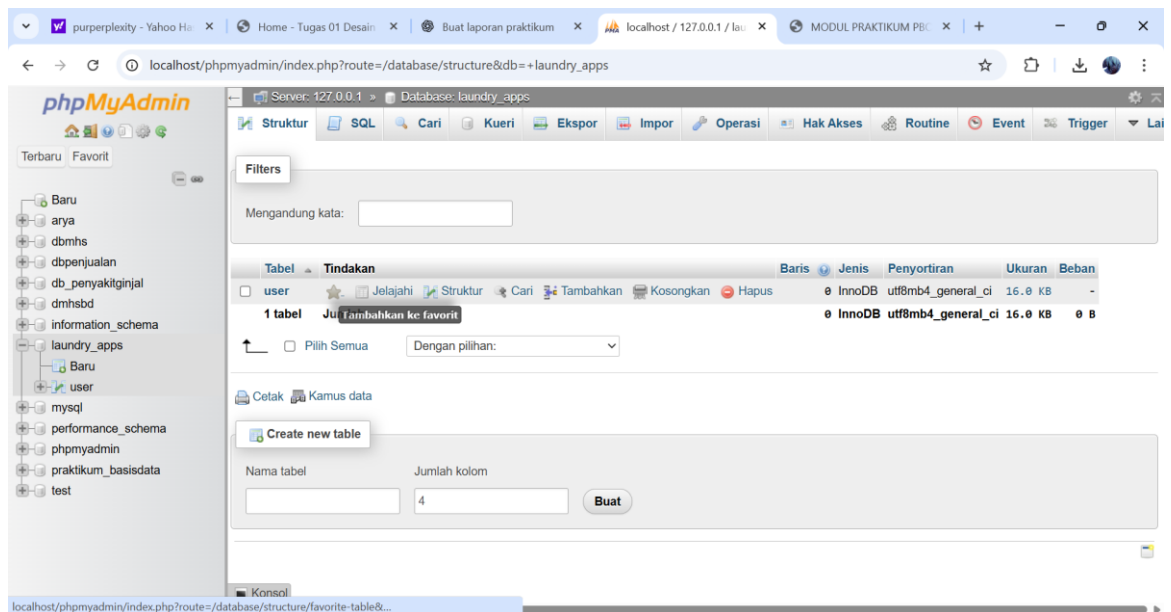


2. Tambahkan **MySQL Connector/J** ke dalam proyek Eclipse melalui *Build Path*.

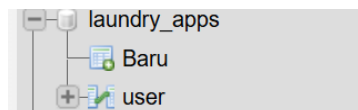


## 4.2 Pembuatan Database dan Tabel

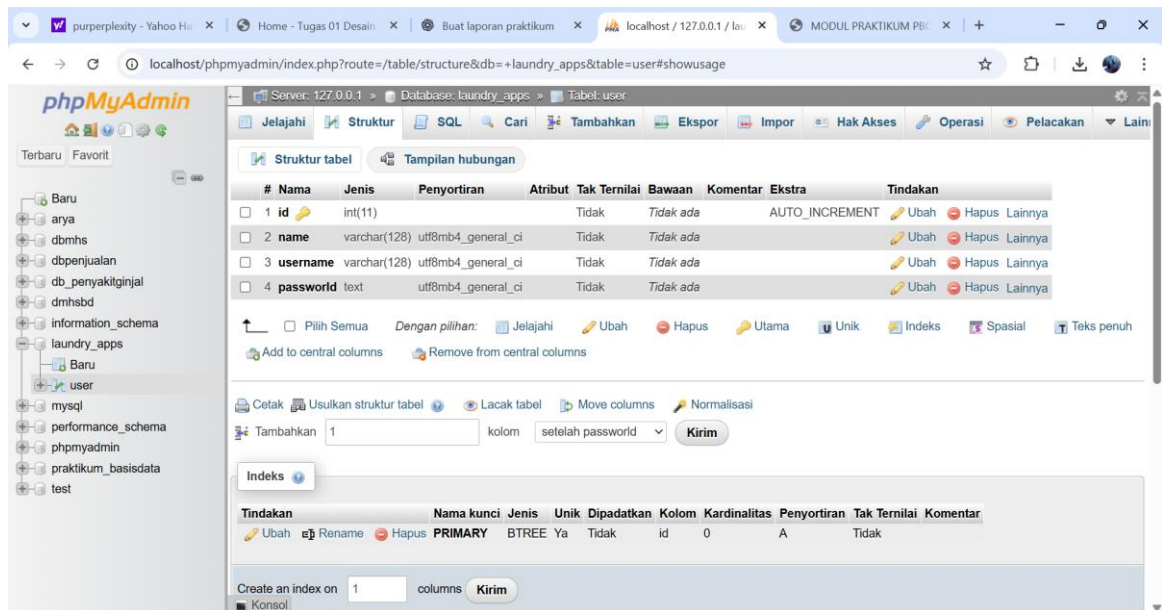
1. Buka **phpMyAdmin** pada <http://localhost/phpmyadmin>.



2. Buat database dengan nama **laundry\_apps**.

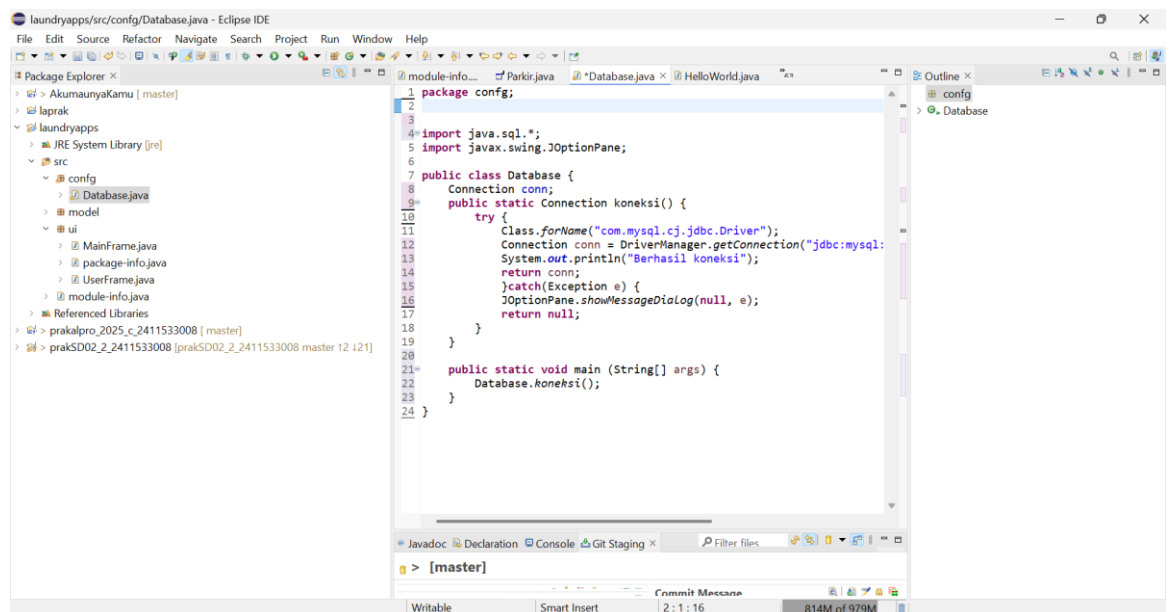


3. Buat tabel user dengan struktur:
  - o id (INT, Auto Increment, Primary Key)
  - o name (VARCHAR)
  - o username (VARCHAR)
  - o password (VARCHAR).



### 4.3 Pembuatan Koneksi Database

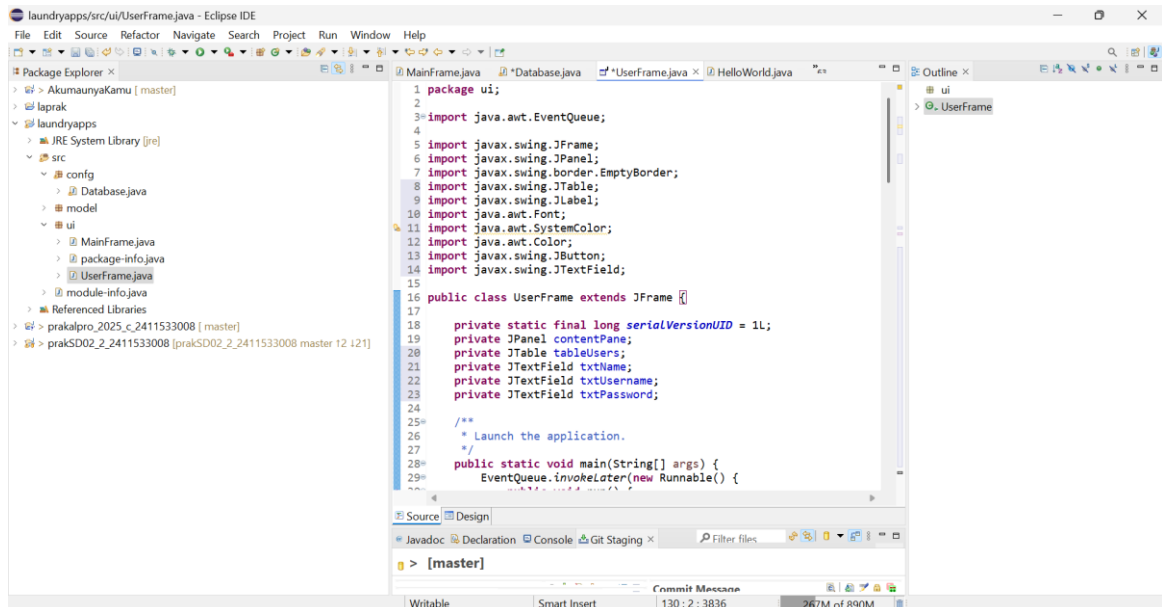
- Buat package config → class Database.
- Gunakan Connection dari java.sql.\* untuk menghubungkan aplikasi Java dengan MySQL.



**Penjelasan :** ▪ Import java.sql.\* digunakan untuk import seluruh fungsi-fungsi SQL ▪ Line 8 membuka method Connection dengan nama koneksi, yang mana method ini akan digunakan untuk membuka koneksi ke database ▪ Line 10-13 membuat koneksi database, jika koneksi berhasil maka akan mengembalikan nilai Connection ▪ Line 15-16 jika koneksi gagal maka akan ditampilkan pesan error menggunakan JOptionPane

## 4.4 Membuat GUI CRUD User

- Buat package ui → class UserFrame.
- Tambahkan komponen:
  - JTextField → txtName, txtUsername, txtPassword.
  - JButton → btnSave, btnUpdate, btnDelete, btnCancel.
  - JTable → tableUsers.



## 4.5 Membuat Table Model

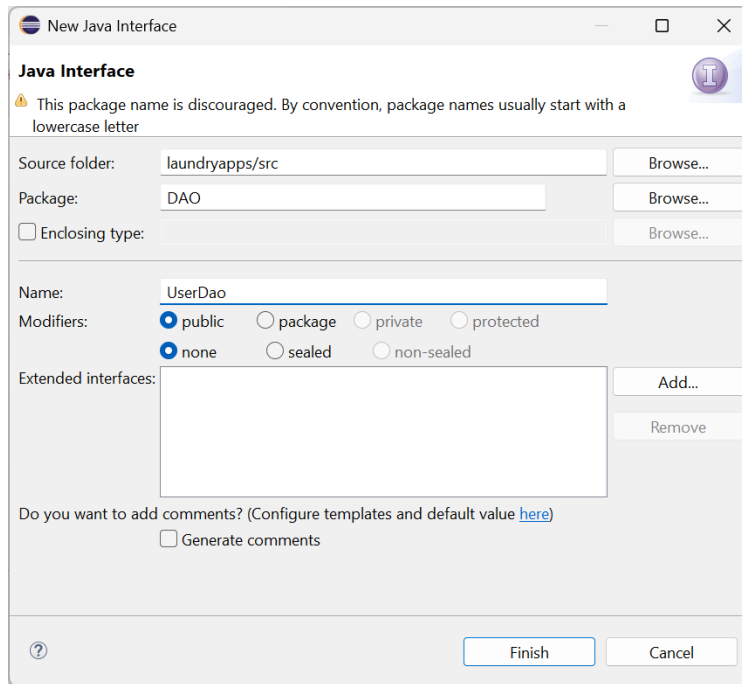
- Buat package table → class TableUser.



- Fungsinya mengambil data dari database untuk ditampilkan ke JTable.

## 4.6 Membuat Fungsi DAO

1. Buat package DAO → interface UserDao.



**New Java Interface**

⚠ This package name is discouraged. By convention, package names usually start with a lowercase letter

Source folder: laundryapps/src Browse...

Package: DAO Browse...

☐ Enclosing type: Browse...

Name: UserDao

Modifiers: ☒ public ☐ package ☐ private ☐ protected  
☒ none ☐ sealed ☐ non-sealed

Extended interfaces: Add... Remove

Do you want to add comments? (Configure templates and default value [here](#))  
☐ Generate comments

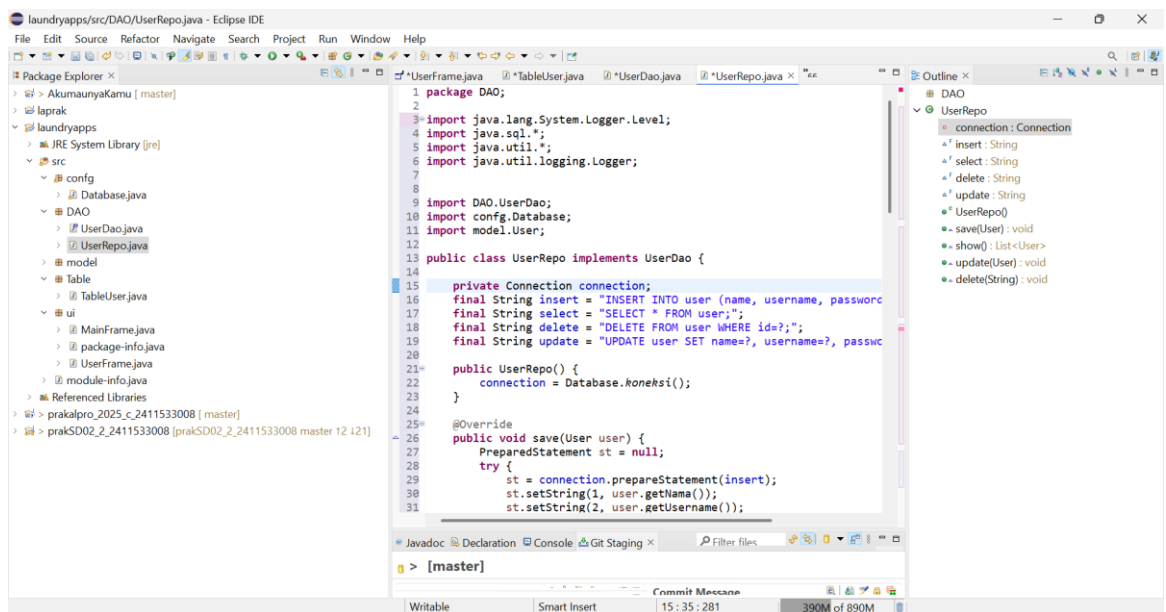
Finish Cancel

```
1 package DAO;  
2  
3 import java.util.List;  
4 import model.User;  
5  
6 public interface UserDao {  
7     void save(User user);  
8     public List<User> show();  
9     public void delete(String id);  
10    public void update(User user);  
11 }  
12
```

2. Method utama:

- save()
- show()
- update()
- delete().

### 3. Implementasikan interface dengan class UserRepo.



..... lanjutan minggu 3













