LAPORAN PRAKTIKUM
PEMOGRAMAN BERORIENTASI ONJEK

JUDUL : PRAKTIKUM 2
MEMBUAT FUNGSI CRUD (Create, Read, Update, Delete) USER DENGAN DATABASE MYSQL

Disusun oleh:

Putri Balqis Afradinata

2411531009

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS

SEPTEMBER

2025

i. TUJUAN

Tujuan praktikum ini yaitu mahasiswa mampu membuat fungsi CRUD data user menggunakan database MySQL, Adapun poin-poin praktikum yaitu :

- Mahasiswa mampu membuat table user pada database MySQL
- Mahasiswa mampu membuat koneksi Java dengan database MySQL
- Mahasiswa mampu membuat tampilan GUI CRUD user
- Mahasiswa mampu membuat dan mengimplementasikan interface
- Mahasiswa mampu membuat fungsi DAO (Data Access Object) dan mengimplementasikannya.
- Mahasiswa mampu membuat fungsi CRUD dengan menggunakan konsep Pemrograman Berorientasi Objek

ii. ALAT

- Computer / laptop yang telah terinstall JDK dan Eclipse
- MySQL / XAMPP
- MySQL connector atau Connector/J

iii. TEORI

XAMPP yaitu paket software yang terdiri dari Apache HTTP Server, MySQL,PHP dan Perl yang bersifat open source, xampp biasanya digunakan sebagai deleveopment environment dalam pengembangan aplikasi berbasis web secara localhost.

Apache berfungsi sebagai web server yang digunakan untuk menjalankan halaman web, MySQL digunakan untuk manajemen basis data dalam melakukan manipulasi data, PHP digunakan sebagai Bahasa pemrograman untuk membuat aplikasi berbasis web.

MySQL adalah sebuah relational database management system (RDBMS) open-source yang digunakan dalam pengelolaan database suatu aplikasi, MySQL ini dapat digunakan untuk menyimpan, mengelola dan mengambil data dalam format table.

MySQL Connection/j adalah driver yang digunakan untuk menghubungkan aplikasi

berbasis java dengan database MySQL sehingga dapat berinteraksi seperti menyimpan, mengubah, mengambil dan menghapus data. Beberapa fungsi MySQL connector yaitu :

- Membuka koneksi ke database MySQL
- Mengirimkan permintaan SQL ke server MySQL
- Menerima hasil dari permintaan SQL
- Menutup koneksi ke database

MySQL DAO (Data Access Object) merupakan object yang menyediakan abstract interface terhadap beberapa method yang berhubungan dengan database seperti mengambil data (read), menyimpan data(create), menghapus data (delete), mengubah data(update). Tujuan penggunaan DAO yaitu :

- memudahkan untuk dikelola
- Meningkatkan modularitas yaitu memisahkan logika akses data dengan logika bisnis sehingga • Meningkatkan reusabilitas yaitu DAO dapat digunakan Kembali
- Perubahan pada logika akses data dapat dilakukan tanpa mempengaruhi logika bisnis.

Interface dalam Bahasa java yaitu mendefinisikan beberapa method abstrak yang harus diimplementasikan oleh class yang akan menggunakannya.

CRUD (Create, Read, Update, Delete) merupakan fungsi dasar atau umum yang ada pada sebuah aplikasi yang mana fungsi ini dapat membuat, membaca, mengubah dan menghapus suatu data pada database aplikasi.

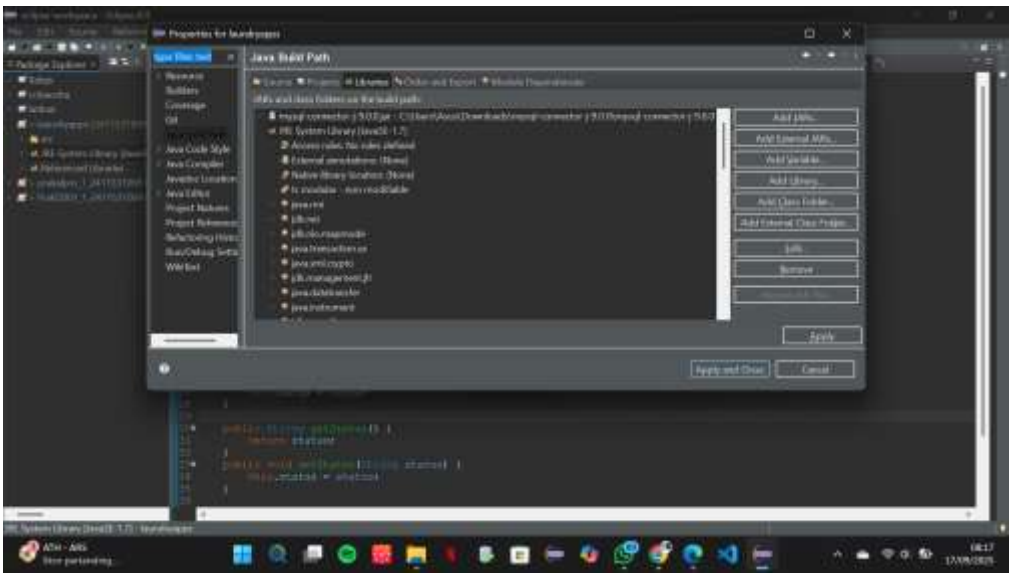
iv. LANGKAH-LANGKAH

- Install XAMPP
  - Download XAMPP dari pada link berikut : https://www.apachefriends.org/
  - Setelah didownload install xampp pada computer/laptop masing-masing
  - Jalankan xampp dan aktifkan apache dan mysql

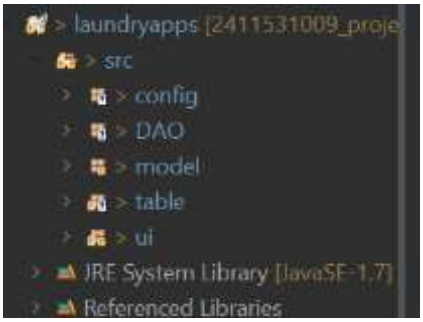
➢ Menambahkan MySQL Connector

Aplikasi java agar dapat terhubung dengan database MySQL membutuhkan sebuah driver yaitu MySQL Connection, berikut ini Langkah-langkah membuat koneksi Database MySQL.

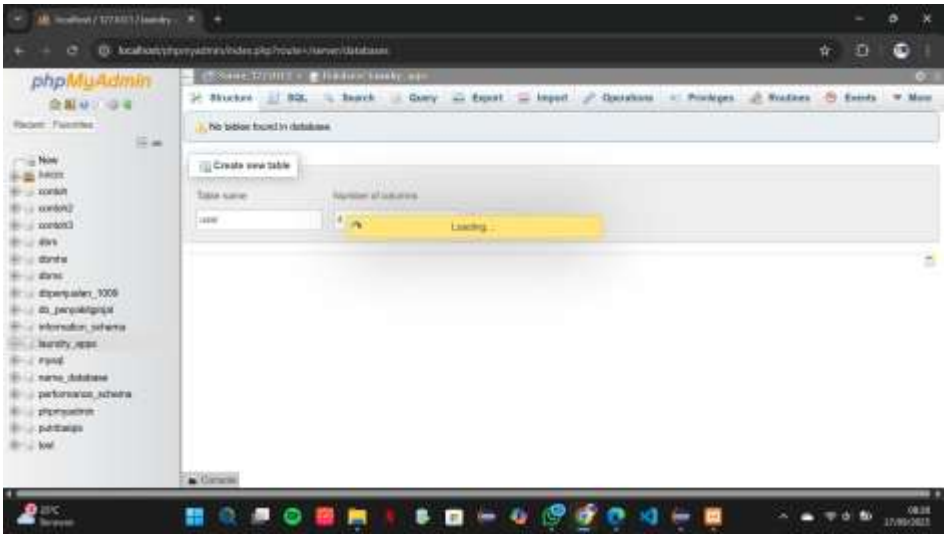
- Download MySQL connection pada link berikut https://dev.mysql.com/downloads/connector/j/
- Menambahkan MySQL Connector kedalam project dengan cara klik kanan directory JRE System Library → Built Path → Configure Build Path
- Selanjutnya pilih Libraries → Classpath
- Tambahkan file MySQL Connector dengan cara klik Add External JARs dan pilih file yang telah didownlod dan pilih Apply and Close.



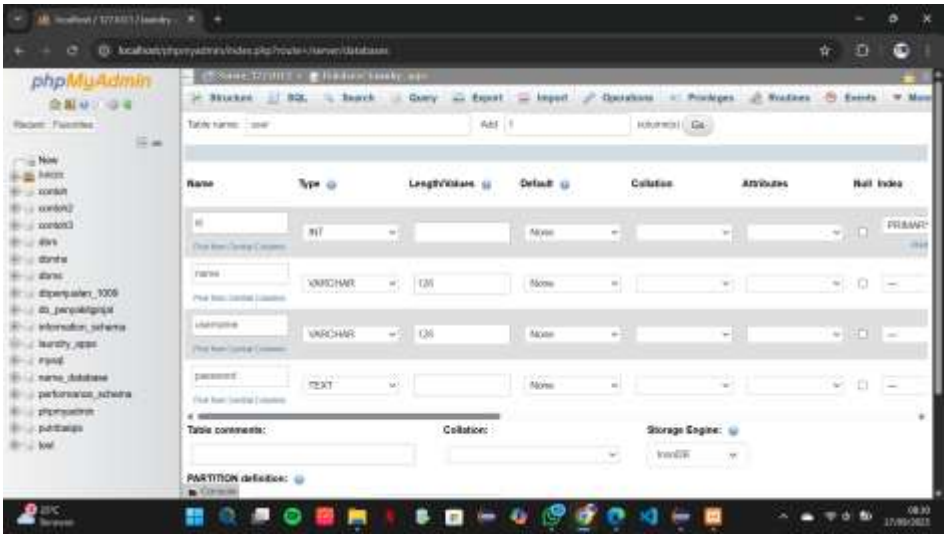
- Jika berhasil menambahkan MySQL Connector maka akan generate folder Referenced Libraries pada project yang berisi MySQL Connector.



- Membuat Database dan Table User
  - Buka http://localhost/phpmyadmin
  - Klik new dan buat database dengan nama laundry_apps
  - Buat table user dengan cara klik database laundry_apps dan buat table dengan nama user



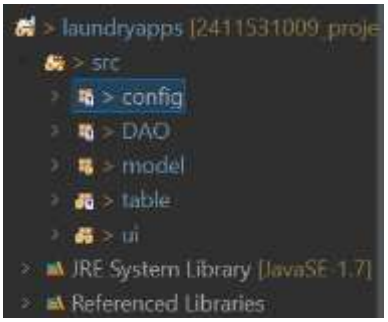
  - Klik create maka akan muncul seperti gambar dibawah ini, lalu isi seperti ketentuan di bawah dan klik save setelahnya.



- Membuat Koneksi ke Database MySQL

Setelah berhasil menambahkan MySQL Connector maka dapat membuat koneksi ke database MySQL, berikut Langkah-langkahnya.

- Buat package baru dengan nama config, package ini yang akan digunakan untuk membuat konfigurasi aplikasi yang akan dibuat termasuk dengan konfigurasi database.



- Buat class baru dengan nama Database, kemudian konfigurasi sesuai dengan kode program berikut

```
1  package config;
2
3  import java.sql.*;
4
5
6  public class Database {
7      Connection conn;
8      public static Connection koneksi() {
9          try {
10             Class.forName("com.mysql.cj.jdbc.Driver");
11             Connection conn = DriverManager.getConnection(
12                 "jdbc:mysql://localhost/laundry_apps",
13                 "root", ""
14             );
15             return conn;
16         } catch(Exception e) {
17             JOptionPane.showMessageDialog(null, e);
18             return null;
19         }
20     }
21 }
22
23
```

a) Import java.sql.* digunakan untuk import seluruh fungsi-fungsi SQL
b) Line 8 membuka method Connection dengan nama koneksi, yang mana method ini akan digunakan untuk membuka koneksi ke database
c) Line 10-13 membuat koneksi database, jika koneksi berhasil maka akan mengembalikkan nilai Connection
d) Line 15-16 jika koneksi gagal maka akan ditampilkan pesan error menggunakan JOptionPane.

➢ Membuat Tampilan CRUD User

• Buat file baru menggunakan JFrame pada package ui dengan nama UserFrame seperti gambar berikut ini.

```java
package ui;

import java.awt.EventQueue;

public class UserFrame extends JFrame {

    private static final long serialVersionUID = 1L;
    private JPanel contentPane;
    private JTextField txtName;
    private JTextField txtUsername;
    private JTextField txtPassword;
    private JButton btnUpdate;
    private JButton btnDelete;
    private JButton btnCancel;
    private JTable tableUsers;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    UserFrame frame = new UserFrame();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    /**
     * Create the frame.
```

```java
    public UserFrame() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 583, 744);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

        setContentPane(contentPane);
        contentPane.setLayout(null);

        JLabel lblNewLabel = new JLabel("Name");
        lblNewLabel.setFont(new Font("Tahoma", Font.PLAIN, 14));
        lblNewLabel.setBounds(32, 52, 60, 29);
        contentPane.add(lblNewLabel);

        txtName = new JTextField();
        txtName.setBounds(102, 54, 396, 27);
        contentPane.add(txtName);
        txtName.setColumns(10);

        JLabel lblUsername = new JLabel("Username");
        lblUsername.setFont(new Font("Tahoma", Font.PLAIN, 14));
        lblUsername.setBounds(32, 96, 60, 29);
        contentPane.add(lblUsername);

        txtUsername = new JTextField();
        txtUsername.setColumns(10);
        txtUsername.setBounds(102, 99, 396, 27);
        contentPane.add(txtUsername);

        JLabel lblPassword = new JLabel("Password");
        lblPassword.setFont(new Font("Tahoma", Font.PLAIN, 14));
        lblPassword.setBounds(32, 140, 60, 29);
        contentPane.add(lblPassword);
```

a) Package & Import → Kode ada di package ui, menggunakan library Swing (JFrame, JPanel, JButton, JLabel, JTextField, JTable) untuk membuat GUI Java.
b) Class UserFrame → Turunan dari JFrame, artinya ini adalah window utama aplikasi.
c) Komponen GUI:
   - JLabel → label teks seperti Name, Username, Password.
   - JTextField → input field untuk nama, username, dan password.
   - JButton → tombol aksi (Save, Update, Delete, Cancel).
   - JTable → tabel untuk menampilkan daftar user.
d) Constructor UserFrame() → mengatur layout, ukuran window, posisi tiap komponen, warna tombol, dll.
e) main() method → menjalankan aplikasi dengan menampilkan frame (setVisible(true)).

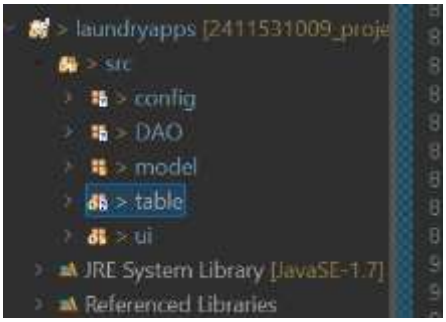
Kode ini bikin aplikasi GUI sederhana untuk manajemen user (input nama, username, password, lalu bisa *Save, Update, Delete, Cancel,* dan menampilkan data user di tabel).

- Membuat Table Model

Table model user ini berguna untuk mengambil data dari database dan

ditampilkan kedalam table.

- Buat package baru dengan nama table



- Buat file baru didalam package table dengan nama TableUser, kemudian isikan dengan kode program berikut.

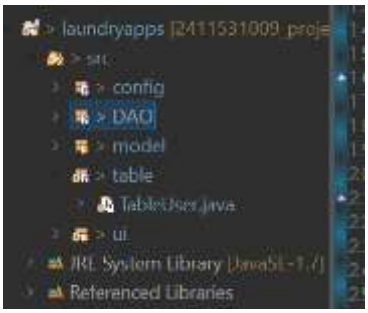
a) Kelas TableUser → turunan dari AbstractTableModel, dipakai untuk menghubungkan data user dengan JTable di GUI.

b) Atribut utama:
   - List<user> ls → menampung daftar objek user.
   - columnNames → nama kolom tabel: ID, Name, Username, Password.

c) Method override:
   - getRowCount() → jumlah baris = banyaknya data user.
   - getColumnCount() → jumlah kolom = 4.
   - getColumnName(int column) → memberi nama kolom sesuai array columnNames.
   - getValueAt(int row, int col) → ambil nilai dari objek user berdasarkan baris & kolom.

Kelas ini adalah model tabel khusus untuk data user, supaya data user bisa otomatis ditampilkan di JTable dengan format kolom *ID, Name, Username, Password*.

- Membuat Fungsi DAO
  - Buat package baru dengan nama DAO



  - Buat class Interface baru dengan nama UserDAO, kemudian isikan dengan kode program berikut.

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

Terdapat method save, show, delete dan update. Method pada class interface digunakan sebagai method utama yang wajib diimplementasikan pada class yang menggunakannya.

- Menggunakan Fungsi DAO
  - Buat class baru pada package DAO dengan nama UserRepo yang mana akan digunakan untuk mengimplementasikan DAO yang telah dibuat.
  - Implementasikan UserDao dengan kata kunci implements

```
public class UserRepo implements UserDAO {
```

- Membuat instanisasi Connection, membuat constructor dan membuat String untuk melakukan manipulas database.

```
private Connection connection;
final String insert = "INSERT INTO user (name, username, password) VALUES (?,?,?);";
final String select = "SELECT * FROM user;";
final String delete = "DELETE FROM user WHERE id=?;";
final String update = "UPDATE user SET name=?, username=?, password=? WHERE id=?;";

public UserRepo() {
    connection = Database.koneksi();
}
```

- Membuat method save, isikan dengan kode program berikut.

```
public void save(User user) {
    // TODO Auto-generated method stub
    PreparedStatement st = null;
    try {
        st = connection.prepareStatement(insert);
        st.setString(1, user.getNama());
        st.setString(2, user.getUsername());
        st.setString(3, user.getPassword());
        st.executeUpdate();
    } catch(SQLException e) {
        e.printStackTrace();
    } finally {
        try {
            st.close();
        } catch(SQLException e) {
            e.printStackTrace();
        }
    }
}
```

a) Fungsinya → menyimpan data user baru ke database.

b) Alurnya:

- Buat PreparedStatement dari query insert.
- Isi parameter query dengan data dari objek user (nama, username, password).
- Jalankan query dengan executeUpdate().
- Kalau ada error → ditangani dengan catch(SQLException e).
- Setelah selesai → PreparedStatement ditutup di blok finally.

Method ini dipakai untuk menambahkan data user ke tabel database dengan aman pakai PreparedStatement.

- Membuat method show untuk mengambil data dari database

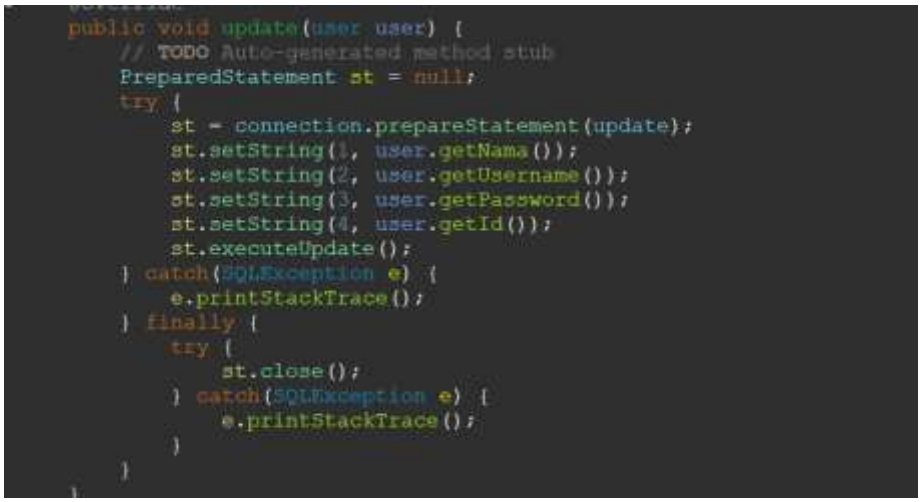
a) Fungsinya → mengambil semua data user dari database dan mengembalikannya dalam bentuk List<user>.

b) Alurnya:

- Buat ArrayList<user> kosong.
- Jalankan query select dengan Statement.
- Loop ResultSet → ambil tiap baris data dari kolom id, name, username, password.
- Simpan ke dalam objek user, lalu tambahkan ke list.
- Kalau ada error SQL → ditangani pakai Logger.
- Return list berisi semua data user.

method ini dipakai untuk **menampilkan daftar user dari database** dalam bentuk list, supaya bisa dipakai lagi di tabel atau logika program.

- Membuat method update yang digunakan untuk mengubah data.



a) Fungsinya → memperbarui data user di database berdasarkan id.

b) Alurnya:

- Siapkan PreparedStatement dari query update.
- Isi parameter query dengan data baru (nama, username, password) dan id user.
- Jalankan query dengan executeUpdate().
- Kalau ada error SQL → ditangani di catch.

method ini dipakai untuk edit/update data user yang sudah ada di tabel database sesuai id**.**

- Membuat method delete yang digunakan untuk menghapus data.

```
@Override
public void delete(String id) {
    // TODO Auto-generated method stub
    PreparedStatement st = null;
    try {
        st = connection.prepareStatement(delete);
        st.setString(1, id);
        st.executeUpdate();
    } catch(SQLException e) {
        e.printStackTrace();
    } finally {
        try {
            st.close();
        } catch(SQLException e) {
            e.printStackTrace();
        }
    }
}
```

a) Fungsinya → menghapus data user di database berdasarkan id.

b) Alurnya:

- Buat PreparedStatement dari query delete.
- Set parameter pertama query dengan nilai id.
- Jalankan perintah SQL dengan executeUpdate().
- Jika ada error SQL → ditangani di catch.
- Setelah selesai → PreparedStatement ditutup di finally.

method ini dipakai untuk menghapus 1 user dari tabel database sesuai id**.**

v. KESIMPULAN

1. DAO (UserRepo) → berisi method save, show, update, delete untuk mengelola data user di database.
2. TableUser → custom TableModel untuk menampilkan data user di JTable dengan kolom ID, Name, Username, Password.
3. UserFrame (GUI) → form utama tempat user berinteraksi: input data, klik tombol (Save, Update, Delete), dan melihat daftar user di tabel.
4. Integrasi → UserFrame memanggil DAO, lalu hasilnya ditampilkan lewat TableUser.
    - Save → tambah user baru.
    - Show → tampilkan semua user.
    - Update → ubah data user terpilih.
    - Delete → hapus user terpilih.