

LAPORAN PRAKTIKUM PEMROGRAMAN BERBASIS OBJEK

PEKAN 2: MySQL dan FUNGSI CRUD



Oleh :

MUHAMMAD FAZIL AFIF

NIM 2411531008

MATA KULIAH PEMOGRAMAN BERBASIS OBJEK

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

FAKULTAS TEKNOLOGI INFORMASI

DEPARTEMEN INFORMATIKA

UNIVERSITAS ANDALAS

PADANG, 2025

A. Pendahuluan

MySQL Connector/Java, atau sekarang sering disebut MySQL Connector/J, adalah driver JDBC resmi yang memungkinkan aplikasi Java untuk terhubung dan berinteraksi dengan basis data MySQL. Ini adalah pustaka Java murni yang mengimplementasikan API Java Database Connectivity (JDBC) standar, memungkinkan Anda menjalankan kueri SQL dan mengakses data di dalam aplikasi Anda

B. Tujuan

Tujuan dari dilakukannya praktikum ini adalah

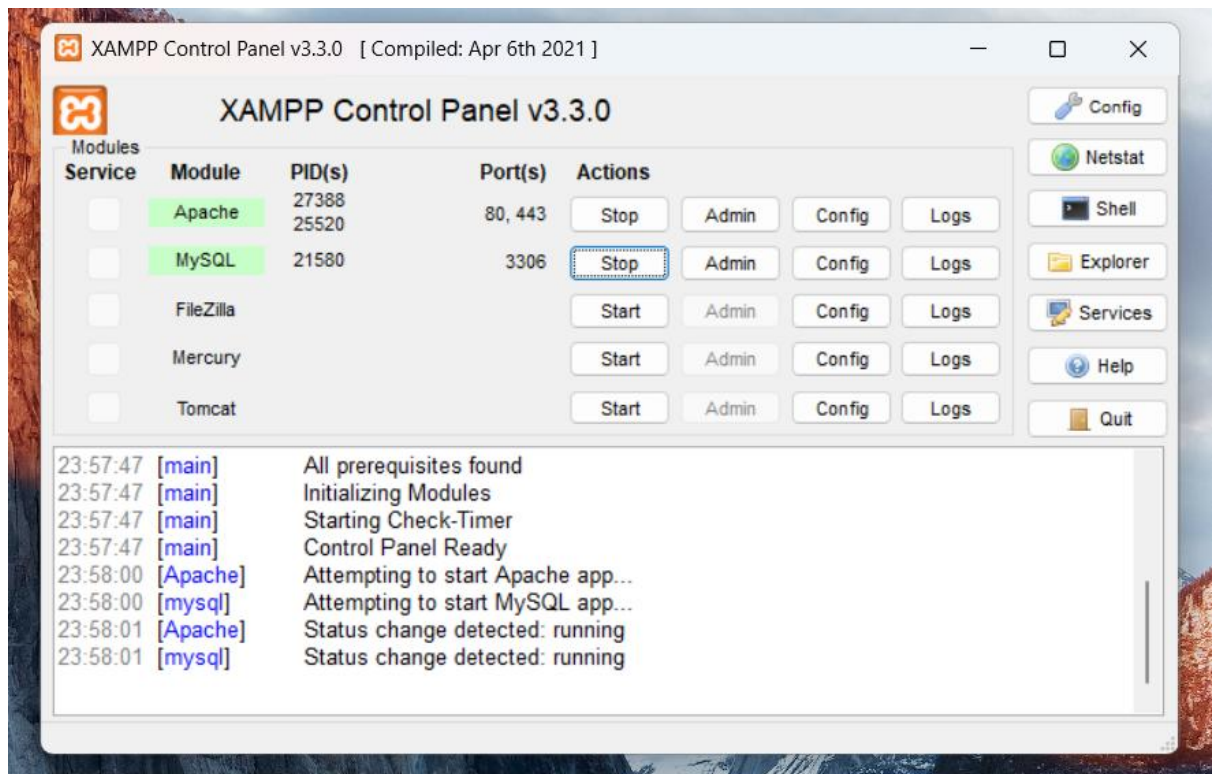
1. Mahasiswa mampu membuat table user pada database MySQL
2. Mahasiswa mampu membuat koneksi Java dengan database MySQL
3. Mahasiswa mampu membuat tampilan GUI CRUD user
4. Mahasiswa mampu membuat dan mengimplementasikan interface
5. Mahasiswa mampu membuat fungsi DAO (Data Access Object) dan mengimplementasikannya.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

C. Alat dan Bahan

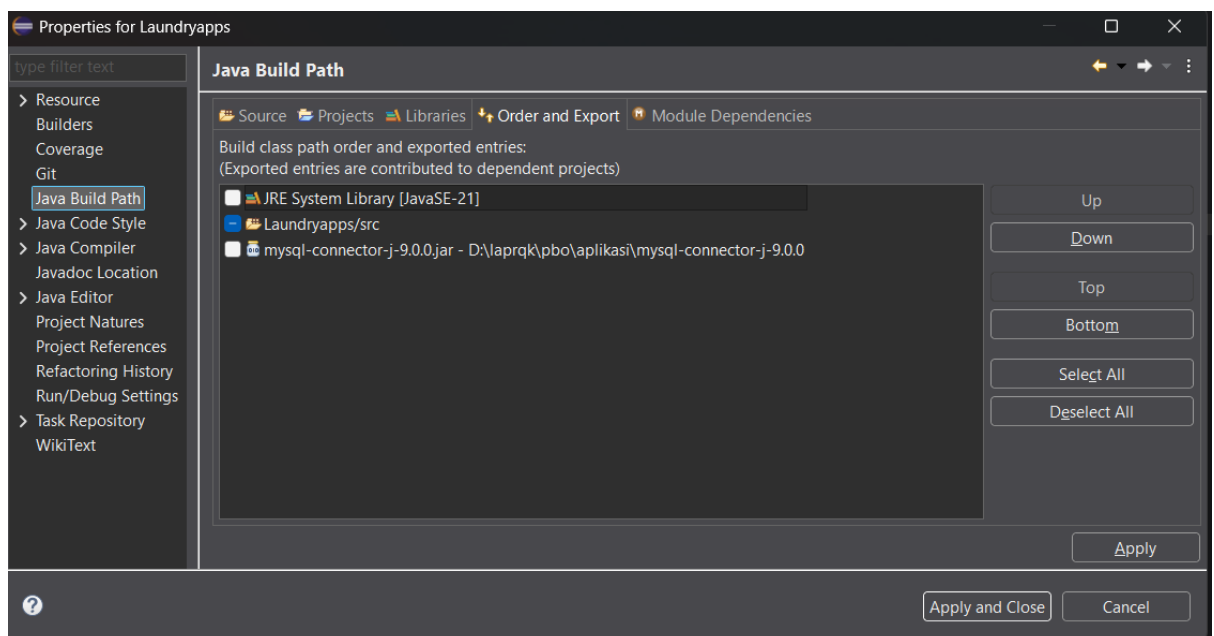
1. IDE (Integrated Development Environment) seperti Eclipse atau IntelliJIDEA
2. Java Development Kit (JDK)
3. Komputer atau Laptop
4. MySQL / XAMPP
5. MySQL connector atau Connector/J

D. Langkah Kerja Praktikum

- a. Menjalankan XAMPP dan menghubungkan MySQL
 1. Buka XAMPP dan aktifkan apache dan mysql

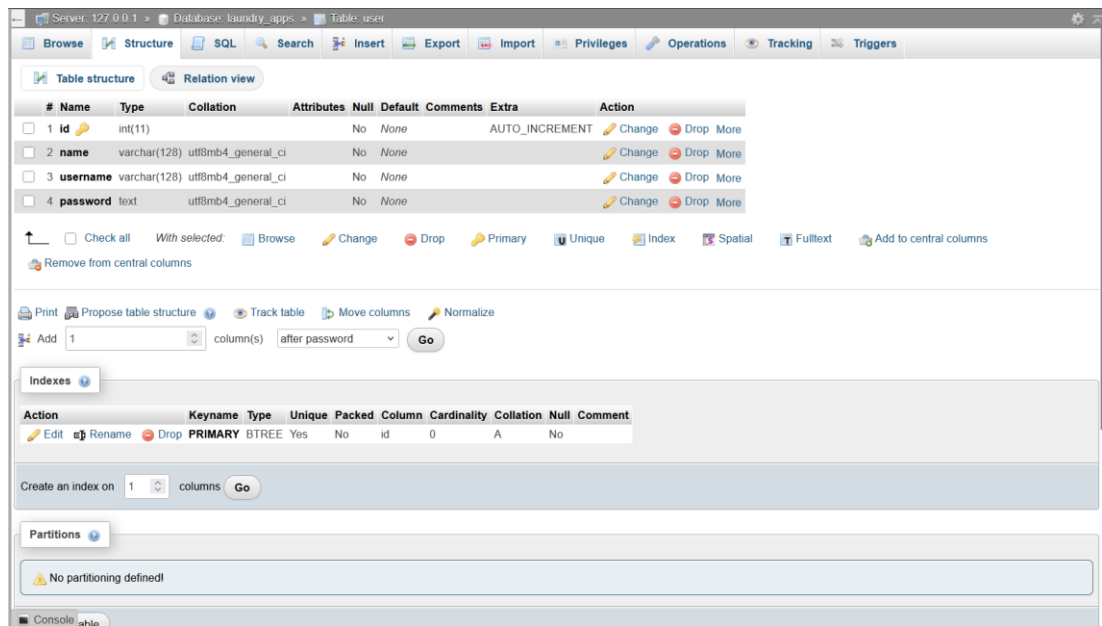


2. Untuk menghubungkan Mysql ke project dengan klik kanan directory JRE System Library lalu Built Path dan Configure Build Path, selanjutnya pilih Libraries lalu Classpath kemudian tambahkan file MySQL Connector dengan cara klik Add External JARs dan pilih file yang telah didownload dan pilih Apply and Close.



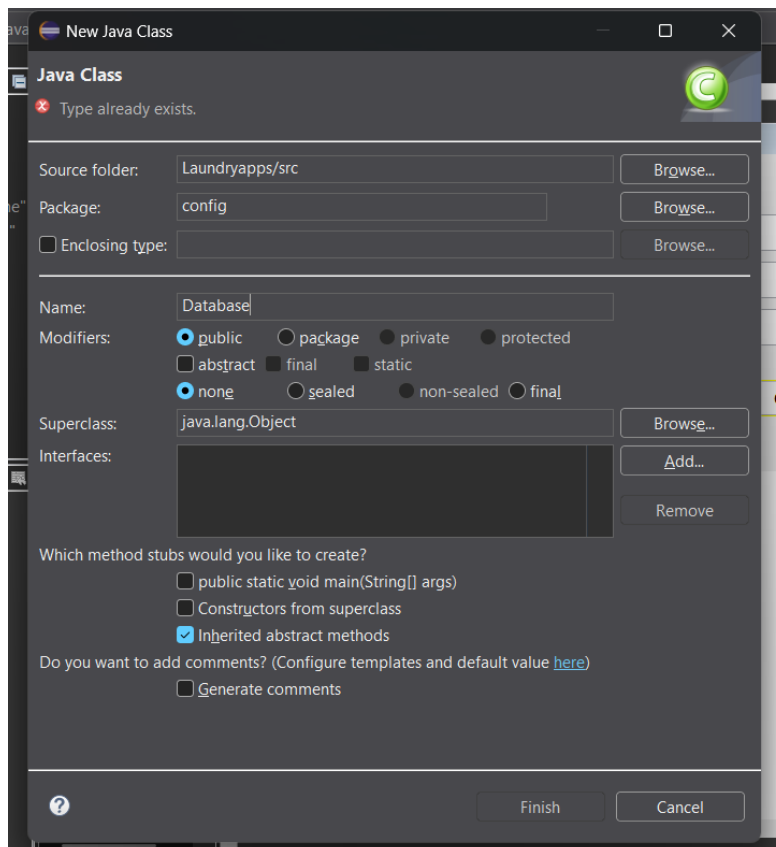
- b. Membuat database dan table user

1. 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 lalu buat attribute id, name, username, dan password



c. Membuat Koneksi ke Database MySQL

1. Buatlah package baru dengan nama config dan buat class baru dengan nama Database.

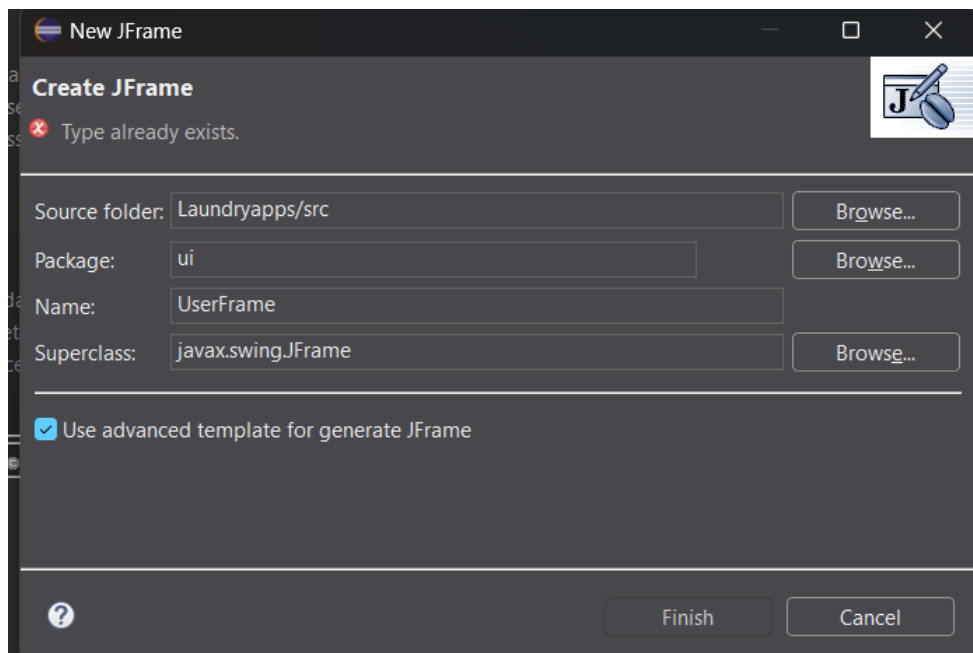


2. Kelas ini digunakan untuk membuat koneksi dengan database MySQL. Pertama, kode melakukan import terhadap library `java.sql.*`. Di dalam kelas Database, terdapat atribut `Connection conn` yang berfungsi sebagai objek koneksi. Selanjutnya terdapat sebuah method static bernama `koneksi()` yang mengembalikan objek bertipe `Connection`. Di dalam blok `try`, kode memanggil `Class.forName("com.mysql.cj.jdbc.Driver")` untuk memuat driver MySQL. Kemudian dibuat koneksi menggunakan `DriverManager.getConnection("jdbc:mysql://localhost/laundry_apps", "root", "")`, di mana alamat database adalah localhost, nama databasenya laundry_apps, username root, dan password kosong.

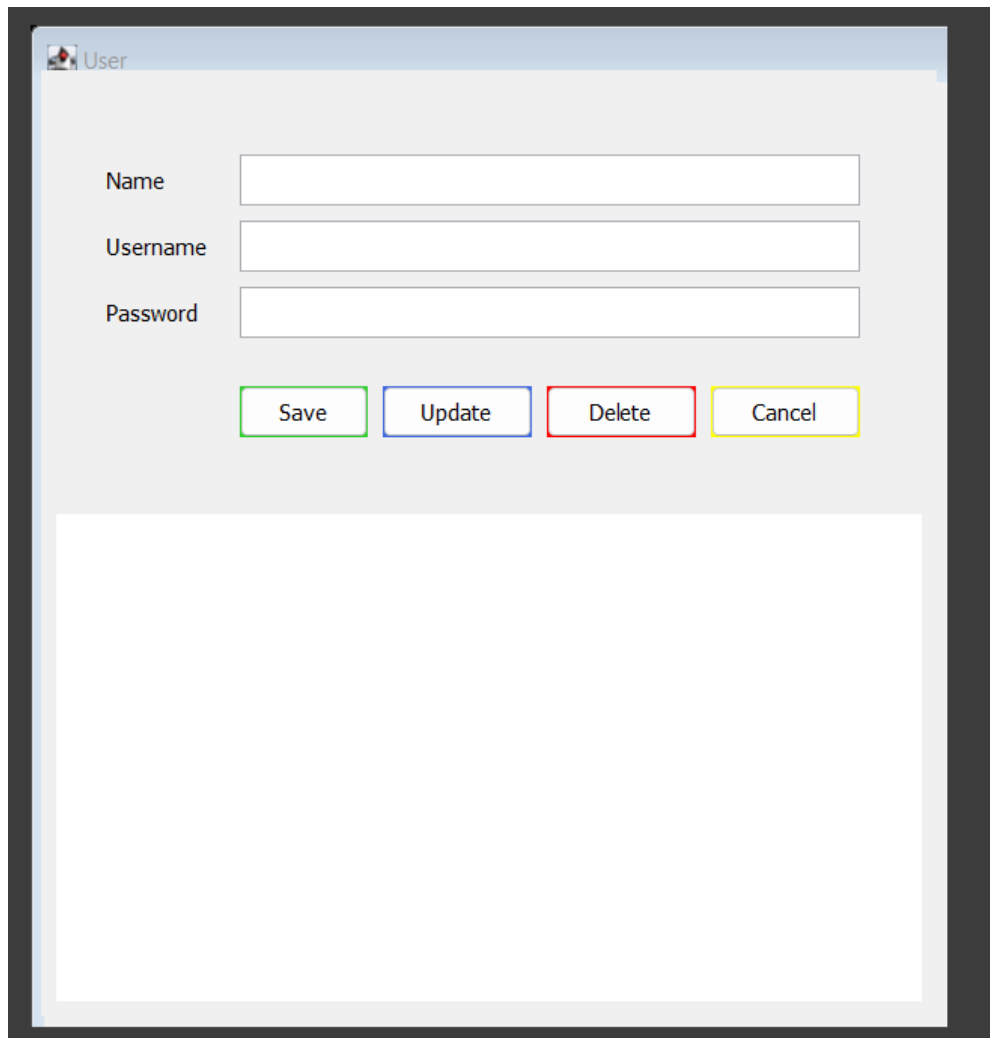
```
1 package config;
2
3 import java.sql.*;
4 import javax.swing.JOptionPane;
5
6 public class Database {
7     Connection conn;
8     public static Connection koneksi() {
9         try {
10             Class.forName("com.mysql.cj.jdbc.Driver");
11             Connection conn = DriverManager.getConnection("jdbc:mysql://localhost/laundry_apps",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

d. Membuat Tampilan CRUD User

1. Buat file baru menggunakan JFrame pada package ui dengan nama UserFrame



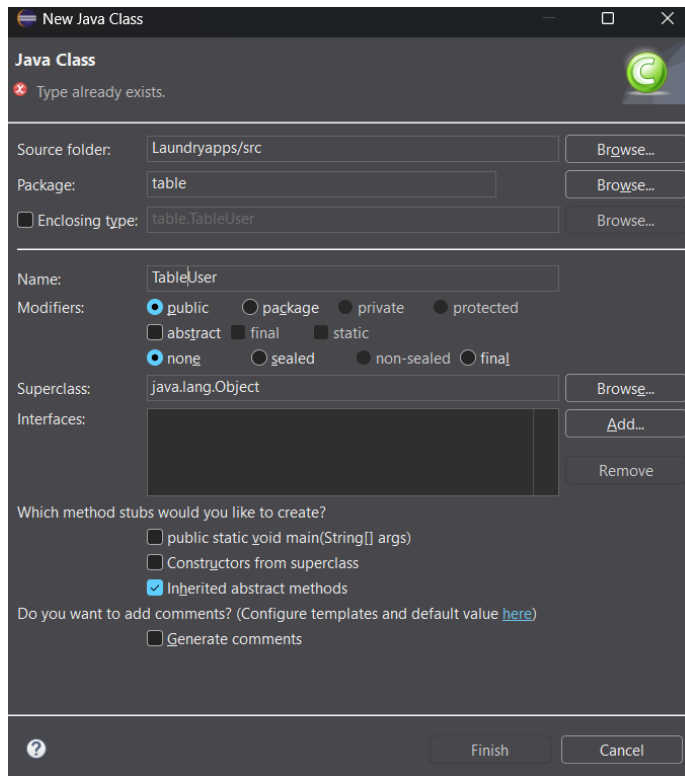
2. Lalu buatlah desain sebagai berikut



A screenshot of a software window titled "User" with a small icon on the left. The window has a light gray background and a dark gray border. It contains three text input fields stacked vertically, labeled "Name", "Username", and "Password" on the left. Below the input fields are four buttons: "Save" (green border), "Update" (blue border), "Delete" (red border), and "Cancel" (yellow border). At the bottom of the window is a large, empty white rectangular area.

e. Membuat Table Model

1. Buat package baru dengan nama table lalu Buat file baru didalam package table dengan nama TableUser,



2. Kode ini membuat sebuah model tabel kustom untuk komponen.JTable di Java Swing yang digunakan menampilkan data user. Model ini mengambil data dari sebuah List<User> lalu mengatur bagaimana data tersebut ditampilkan dalam tabel dengan 4 kolom utama, yaitu ID, Name, Username, dan Password. Dengan kata lain, kelas ini berfungsi sebagai penghubung antara data User dengan tampilan tabel, sehingga data bisa ditampilkan secara terstruktur dalam bentuk baris dan kolom.

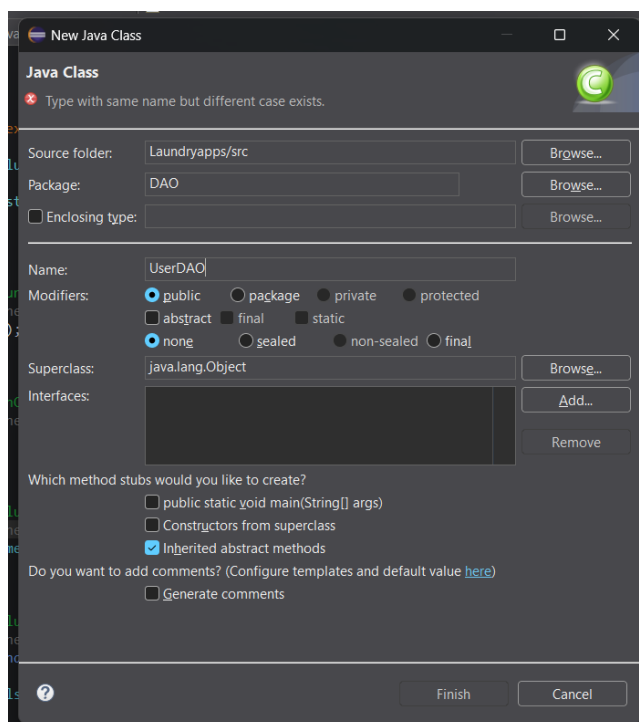
```

1 package table;
2
3 import java.util.List;
4
5
6
7
8
9 public class TableUser extends AbstractTableModel {
10     List<User> ls;
11     private String[] columnNames = {"ID", "Name", "Username", "Password"};
12
13     public TableUser(List<User> ls) {
14         this.ls = ls;
15     }
16
17     @Override
18     public int getRowCount() {
19         // TODO Auto-generated method stub
20         return ls.size();
21     }
22
23     @Override
24     public int getColumnCount() {
25         // TODO Auto-generated method stub
26         return 4;
27     }
28
29     @Override
30     public String getColumnName(int column) {
31         // TODO Auto-generated method stub
32         return columnNames[column];
33     }
34
35     @Override
36     public Object getValueAt(int rowIndex, int columnIndex) {
37         // TODO Auto-generated method stub
38         switch (columnIndex) {
39             case 0:
40                 return ls.get(rowIndex).getId();
41             case 1:
42                 return ls.get(rowIndex).getNama();

```

f. Fungsi DAO

1. Buat package baru dengan nama DAO dan Buat class Interface baru dengan nama UserDao

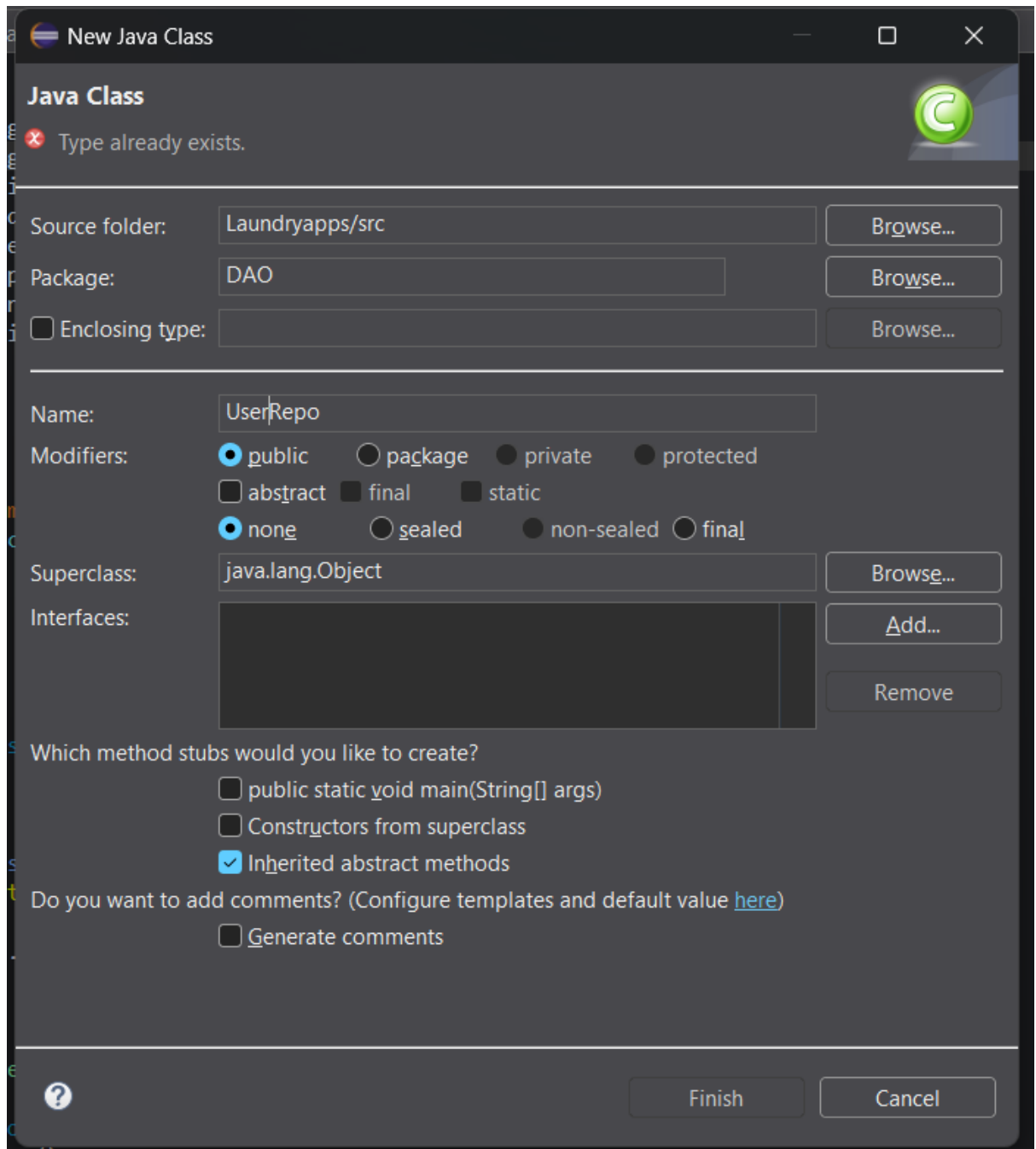


2. Kode di atas mendefinisikan interface UserDao dalam package DAO yang berisi empat method untuk operasi pada data User. Method save(User user) digunakan menambah data baru, Show() menampilkan seluruh data dalam bentuk list, delete(String id) menghapus data berdasarkan ID, dan update(User user) memperbarui data user yang ada. Interface ini hanya mendeklarasikan kontrak, sementara implementasi detailnya dibuat di kelas lain.

A screenshot of a Java IDE with a dark theme. The code defines a package DAO, imports java.util.List, and declares a public interface UserDao with four methods: save, Show, delete, and update. The code is as follows:

```
1 package DAO;
2
3 import java.util.List;
4
5
6
7 public interface UserDao {
8     void save(User user);
9     public List<User> Show();
10    public void delete(String id);
11    public void update(User user);
12 }
13
```

3. Buat class baru pada package DAO dengan nama UserRepo



4. Kode tersebut adalah implementasi dari interface UserDao melalui kelas UserRepo yang berfungsi untuk mengelola data User di database. Pada kelas ini, terdapat query SQL untuk melakukan insert, select, delete, dan update. Konstruktor UserRepo() membuat koneksi ke database menggunakan class Database. Method save(User user) menyimpan data user baru ke tabel. Method Show() menampilkan semua data user dari database dalam bentuk list. Method delete(String id) menghapus data berdasarkan ID, sedangkan update(User user) memperbarui data user sesuai dengan ID yang diberikan.

```

24 import model.User;
25
26 public class UserRepo implements UserDao {
27     private Connection connection;
28     final String insert = "INSERT INTO user (name, username, password) VALUES (?, ?, ?);";
29     final String select = "SELECT * FROM user;";
30     final String delete = "DELETE FROM user WHERE id=?;";
31     final String update = "UPDATE user SET name=?, username=?, password=? WHERE id=?;";
32
33     public UserRepo() {
34         connection = Database.koneksi();
35     }
36
37     @Override
38     public void save(User user) {
39         PreparedStatement st = null;
40         try {
41             st = connection.prepareStatement(insert);
42             st.setString(1, user.getNama());
43             st.setString(2, user.getUsername());
44             st.setString(3, user.getPassword());
45             st.executeUpdate();
46
47         } catch (SQLException e) {
48             e.printStackTrace();
49         } finally {
50             try {
51                 st.close();
52             } catch (SQLException e) {
53                 e.printStackTrace();
54             }
55         }
56     }
57
58     @Override
59     public List<User> Show() {

```

E. Kesimpulan

Setelah melakukan praktikum ini, kita memperoleh pemahaman yang lebih mendalam mengenai cara menghubungkan aplikasi Java dengan MySQL menggunakan XAMPP dan MySQL Connector. Kita juga mampu mengimplementasikan konsep tersebut dengan membuat database dan tabel user, membangun koneksi melalui class Database, serta merancang antarmuka CRUD menggunakan JFrame. Selain itu, kita dapat menampilkan data dengan TableUser dan mengelola operasi CRUD melalui DAO serta implementasinya di UserRepo, sehingga praktikum ini memberikan pengalaman langsung dalam mengembangkan aplikasi berbasis Java yang terhubung dengan database.