Laporan Praktikum

Pemrograman Berorientasi Objek



Disusun Oleh :

**Ahda Rindang Al-Amin (2311531003)**

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

Departemen Informatika

Fakultas Teknologi Informasi

Universitas Andalas

Tahun 2024

**Membuat Fungsi CRUD (Create, Read, Update, Delete) User Dengan Database MySQL**

1. **Tujuan Praktikum**
2. Mampu membuat table user pada database MySQL
3. Mampu membuat koneksi Java dengan database MySQL
4. Mampu membuat tampilan GUI CRUD user
5. Mampu membuat dan mengimplementasikan interface
6. Mampu membuat fungsi DAO (Data Access Object)dan mengimplementasikannya.
7. Mampu membuat fungsi CRUD dengan menggunakan konsep Pemrograman Berorientasi Objek
8. **Pendahuluan**

**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 :

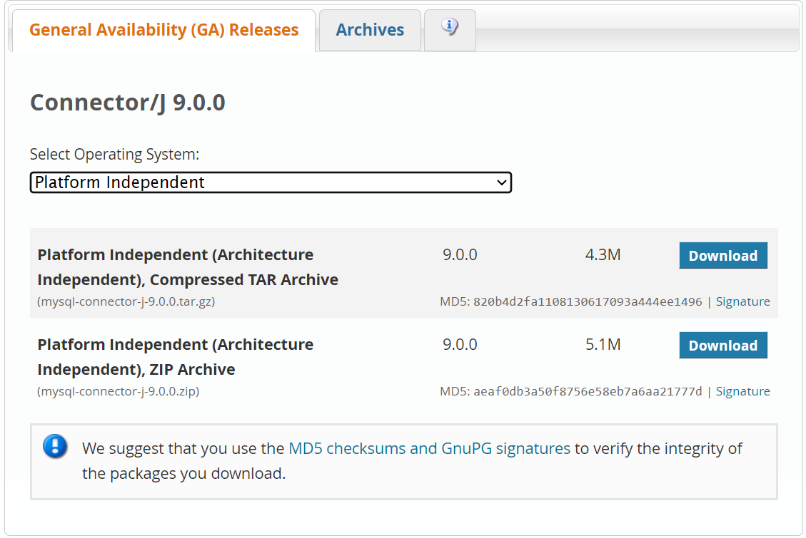
* Meningkatkan modularitas yaitu memisahkan logika akses data dengan logika bisnis sehingga memudahkan untuk dikelola
* Meningkatkan reusabilitas yaitu DAO dapat digunakan Kembali
* Perubahan pada logika akses data dapat dilakukan tanpa mempengaruhi logika bisnis.

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

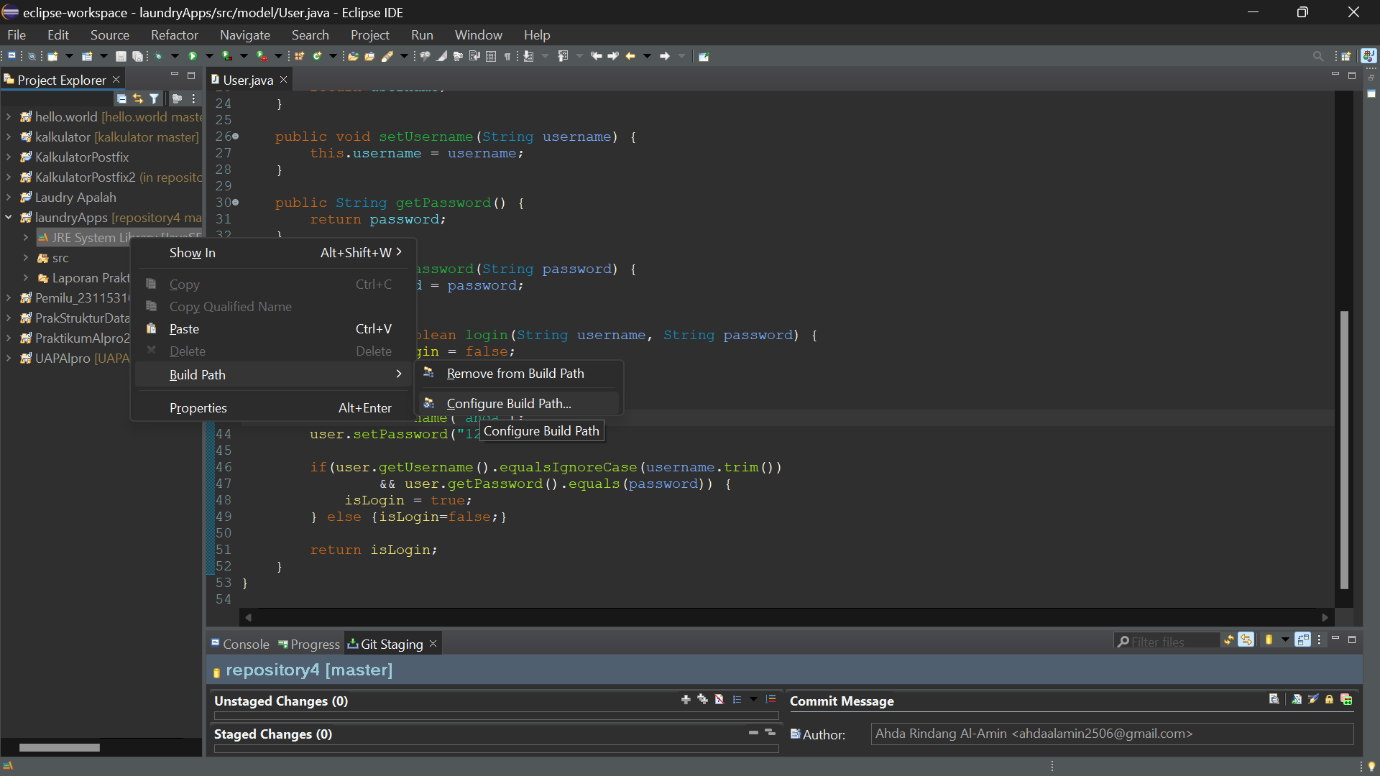
**CRUD** (Create, Read, Update, Delete) merupakan fungsi dasar atau umum yang ada pada sebuah aplikasi yang mana fungsi ini dapat membuat, membaca, mengubah dan menghapus suatu data pada database aplikasi.

1. **Metode Praktikum**
2. **Menambahkan MySQL Connector**
3. Download MySQL Connection pada link berikut:

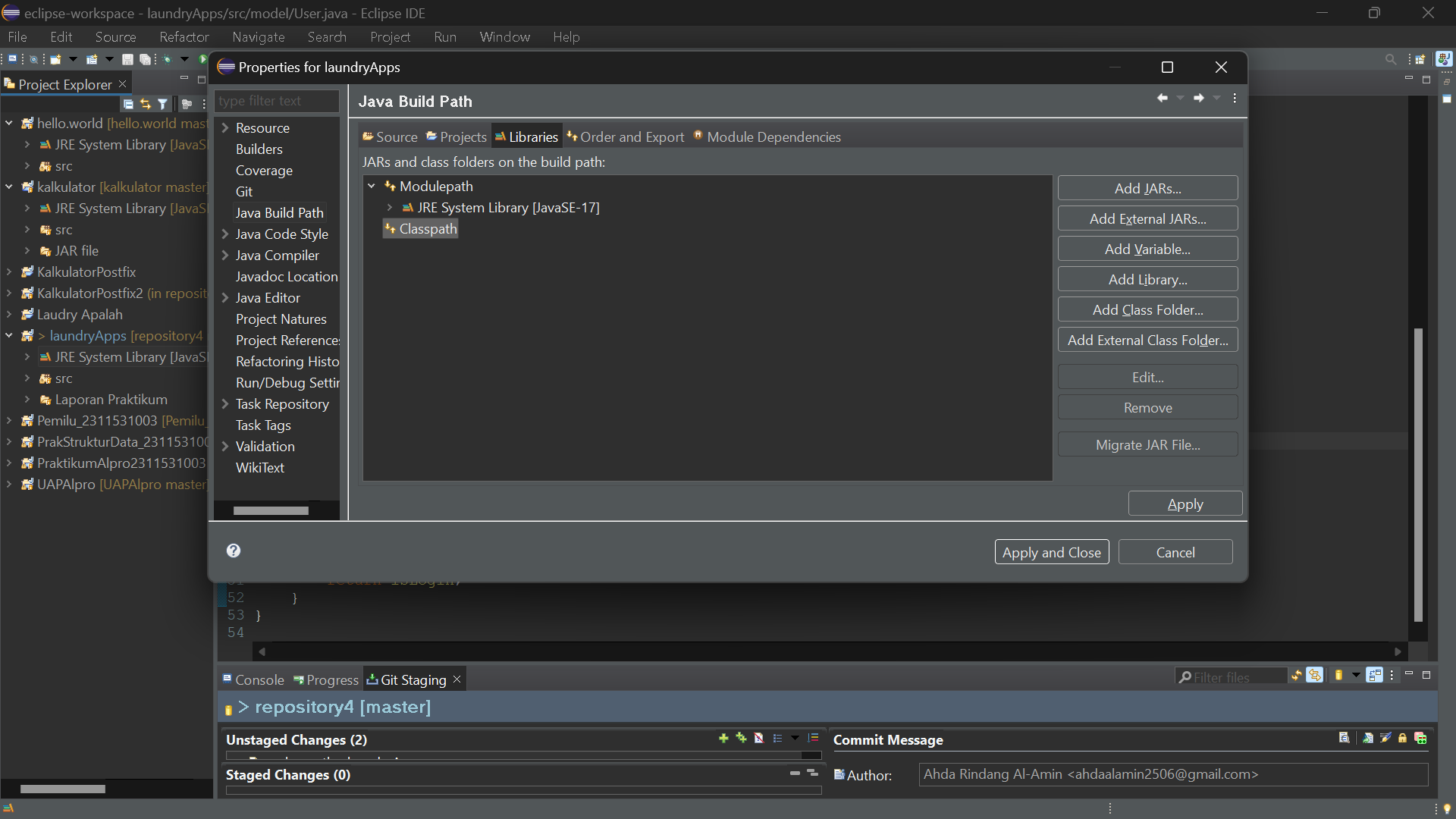
<https://dev.mysql.com/downloads/connector/j/>



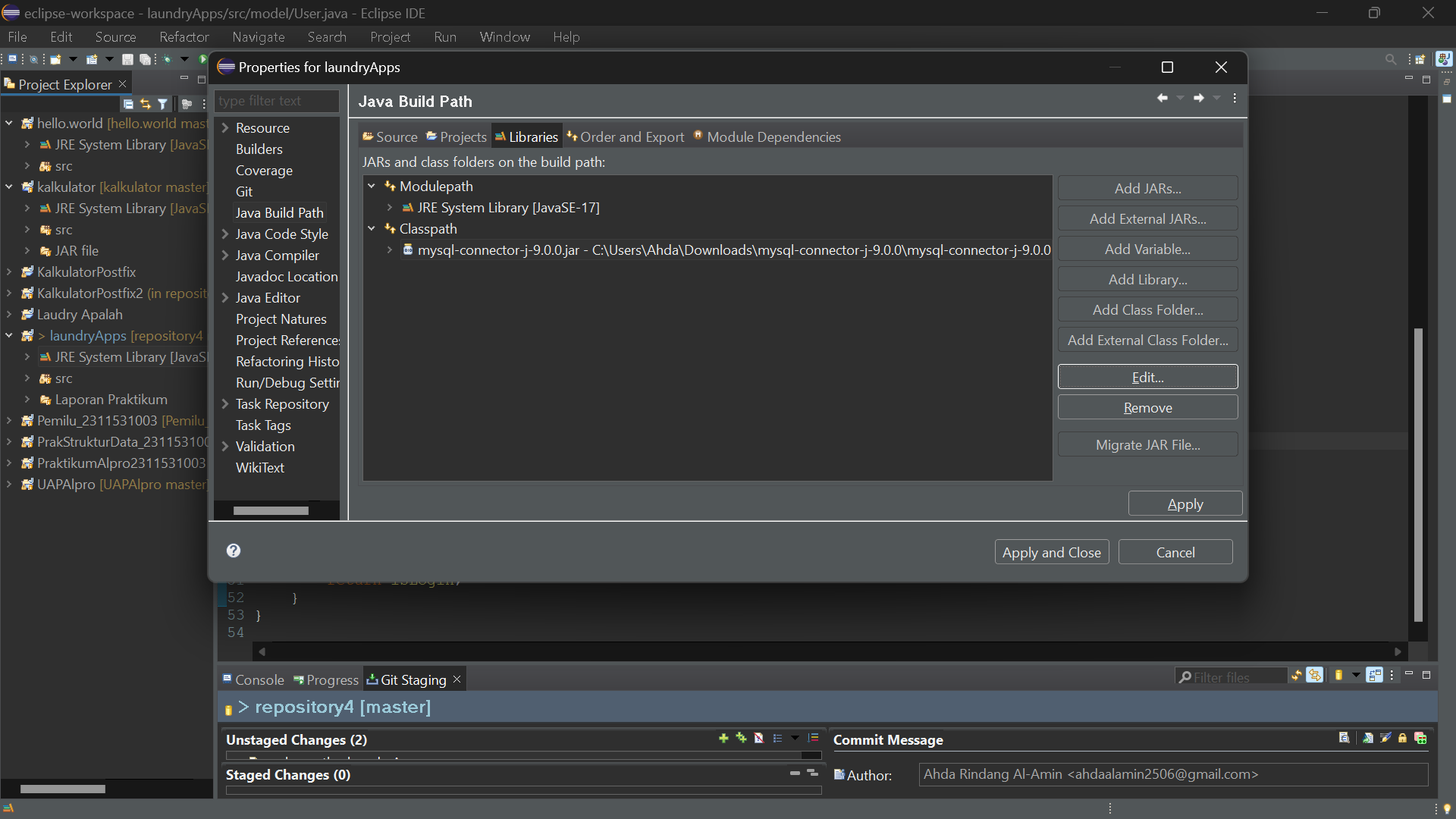
1. Pilih file berekstensi .zip lalu Download
2. Setelah selesai di-download, ekstrak file zip
3. Tambahkan MySQL Connector kedalam project dengan cara klik kanan directory **JRE System Library** > **Built Path** > **Configure Build Path**



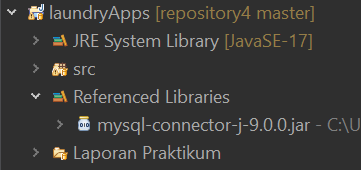
1. Pilih **Libraries** > **Classpath**



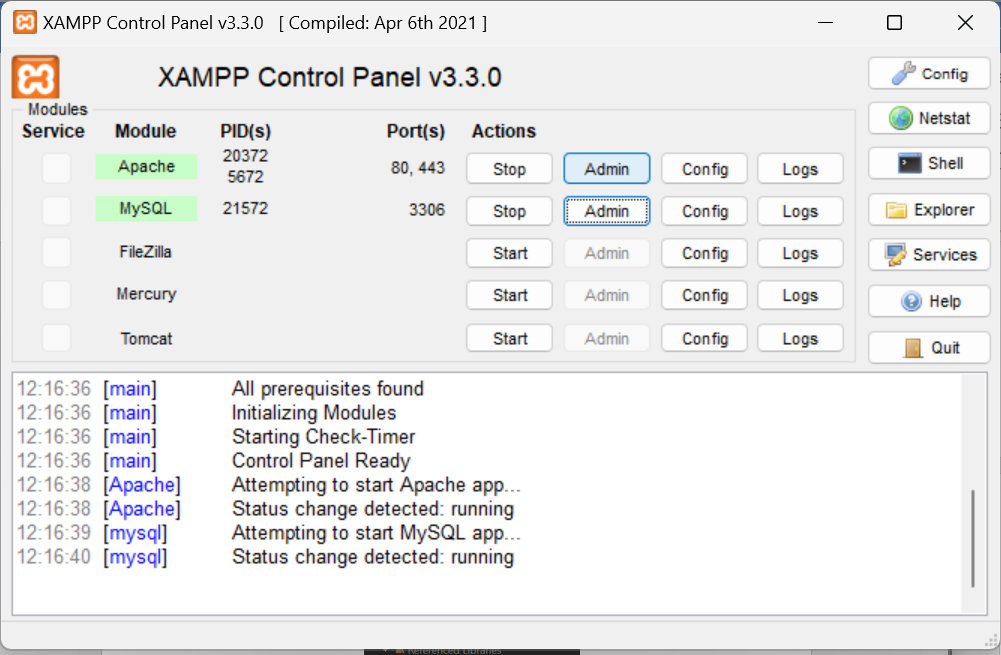
1. Tambahkan file MySQL Connector dengan cara klik **Add External JARs** dan pilih file yang telah didownlod dan pilih **Apply and Close**.



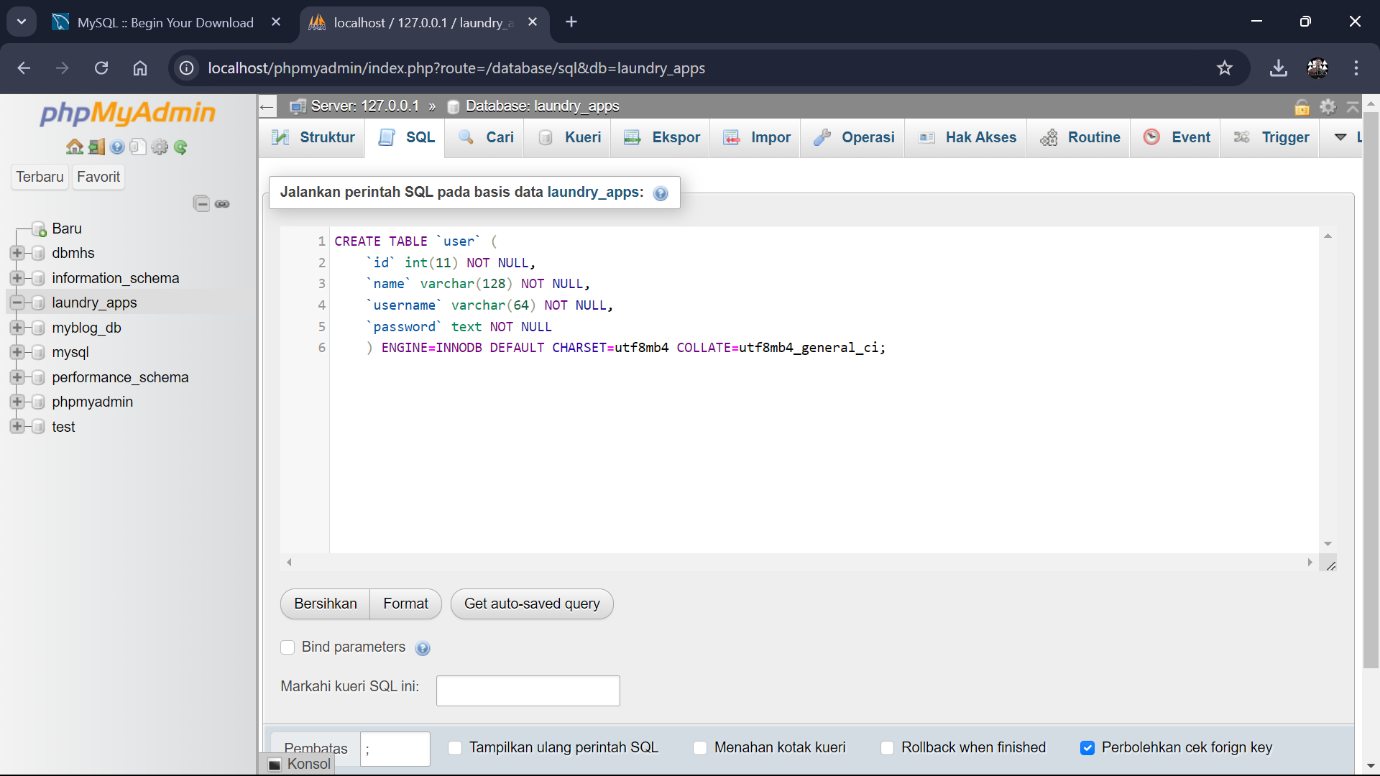
1. Jika berhasil menambahkan MySQL Connector maka akan generate folder Referenced Libraries pada project yang berisi MySQL Connector.



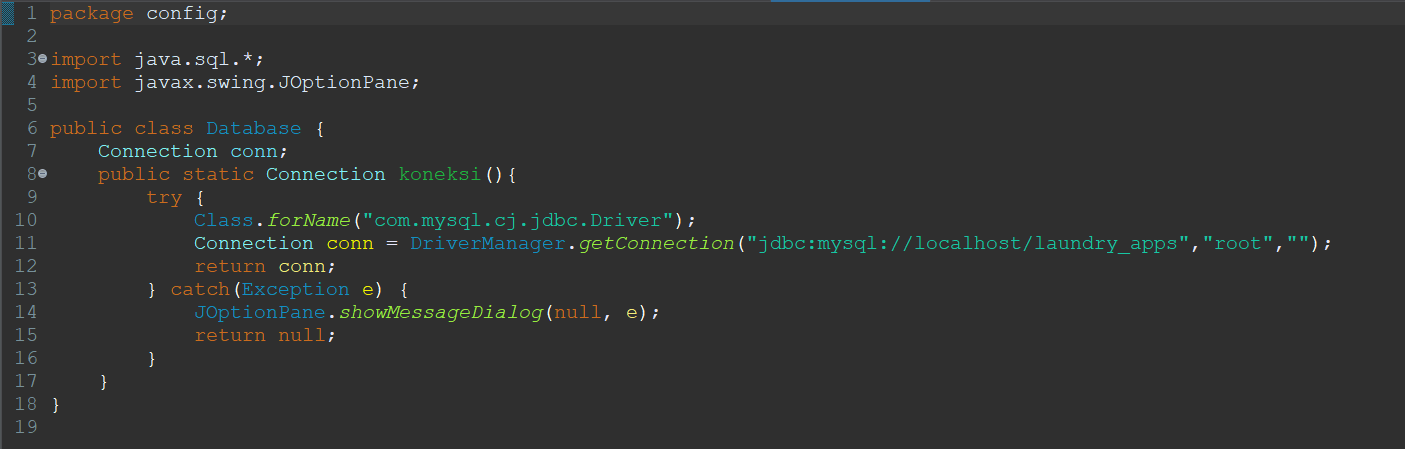
1. **Membuat Database dan Table User**
2. Buka XAMPP dan jalankan Apache dan MySQL



1. Buka phpMyAdmin, buat database dengan nama laundry\_apps
2. Buat tabel user dengan SQL sebagai berikut



1. **Membuat Koneksi ke database MySQL**
2. Buat package baru dengan nama config, package ini yang akan digunakan untuk membuat konfigurasi aplikasi yang akan dibuat termasuk dengan konfigurasi database.
3. Buat class baru dengan nama Database, kemudian konfigurasi sesuai dengan kode program berikut.

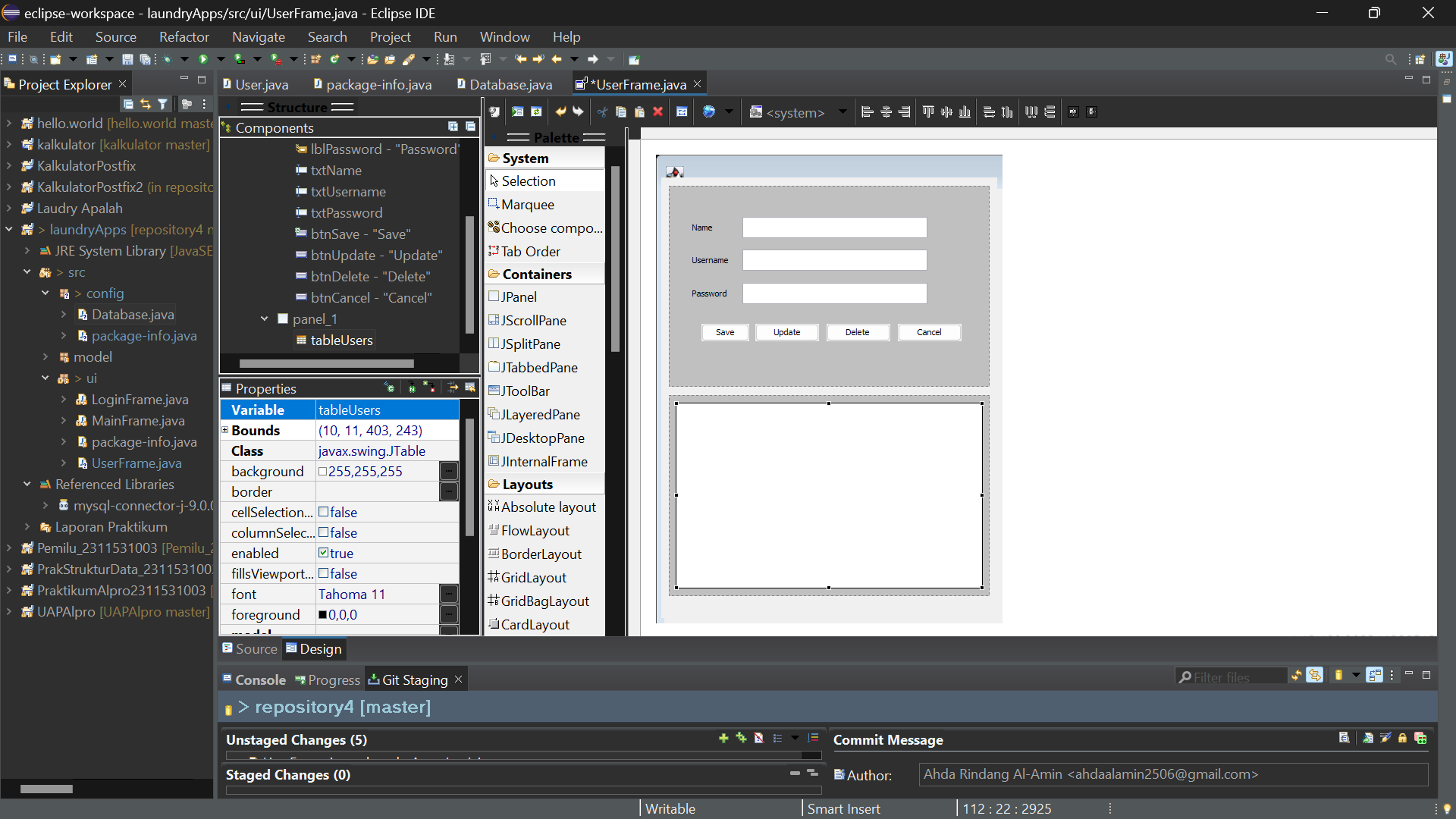


Penjelasan :

* 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 10-12 membuat koneksi database, jika koneksi berhasil maka akan mengembalikkan nilai Connection
* Line 14-15 jika koneksi gagal maka akan ditampilkan pesan error menggunakan JOptionPane.

1. **Membuat tampilan CRUD User**

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



1. **Membuat Table Model**
2. Buat package baru dengan nama table
3. Buat class baru didalam package table dengan nama TableUser

package table;

import java.util.List;

import javax.swing.table.AbstractTableModel;

import model.User;

public class TableUser extends AbstractTableModel {

List<User> ls;

private String[] columnNames = {"ID", "Name", "Username","Password"};

public TableUser(List<User> ls) {

this.ls = ls;

}

*@Override*

public int getRowCount() {

return ls.size();

}

*@Override*

public int getColumnCount() {

return 4;

}

public String getColumnName(int column) {

return columnNames[column];

}

*@Override*

public Object getValueAt(int rowIndex, int columnIndex) {

switch (columnIndex) {

case 0:

return ls.get(rowIndex).getId();

case 1:

return ls.get(rowIndex).getNama();

case 2:

return ls.get(rowIndex).getUsername();

case 3:

return ls.get(rowIndex).getPassword();

default:

return null;

}

}

}

1. **Membuat Fungsi DAO**
2. Buat package baru dengan nama DAO
3. Buat interface baru dengan nama UserDAO

public interface UserDAO {

void save(User user);

public List<User> show();

public void delete(String id);

public void update(User user);

}

1. **Menggunakan Fungsi DAO**

Buat class baru pada package DAO dengan nama UserRepo yang digunakan untuk mengimplementasikan DAO yang telah dibuat pada interface UserDAO

//Impelementasi UserDAO

public class UserRepo implements UserDAO{

//Instanisasi Connection

private Connection connection;

//String untuk manipulasi database

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=?;";

//Constructor

public UserRepo() {

connection = Database.*koneksi*();

}

//Method save untuk menambahkan data ke database

*@Override*

public void save(User user) {

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();

}

}

}

//Method show untuk mengambil data dari database

*@Override*

public List<User> show() {

List<User> ls = null;

try {

ls = new ArrayList<User>();

Statement st = connection.createStatement();

ResultSet rs = st.executeQuery(select);

while(rs.next()) {

User user = new User();

user.setId(rs.getString("id"));

user.setNama(rs.getString("name"));

user.setUsername(rs.getString("username"));

user.setPassword(rs.getString("password"));

ls.add(user);

}

} catch(SQLException e) {

Logger.*getLogger*(UserDAO.class.getName()).log(Level.***SEVERE***, null, e);

}

return ls;

}

//Method delete untuk menghapus data

*@Override*

public void delete(String id) {

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();

}

}

}

//method update untuk mengubah data

*@Override*

public void update(User user) {

PreparedStatement st = null;

try {

st= connection.prepareStatement(update);

st.setString(1, user.getNama());

st.setString(2, user.getUsername());

st.setString(3, user.getPassword());

st.setString(4, user.getId());

st.executeUpdate();

} catch (SQLException e) {

e.printStackTrace();

} finally {

try {

st.close();

} catch (SQLException e) {

e.printStackTrace();

}

}}

}

1. **Menggunakan Fungsi CRUD pada GUI**
2. Buat method reset pada UserFrame untuk menghapus value inputan ketika suatu proses berhasil dilaksanakan

public void reset() {

txtName.setText("");

txtUsername.setText("");

txtPassword.setText("");

}

1. Buat Instance pada UserFrame

UserRepo usr = new UserRepo();

List<User> ls;

public String id;

1. Buat method loadTable , lalu panggil pada method main

public void loadTable() {

ls = usr.show();

TableUser tu = new TableUser(ls);

tableUsers.setModel(tu);

tableUsers.getTableHeader().setVisible(true);

}

/\*\*

\* Launch the application.

\*/

public static void main(String[] args) {

EventQueue.*invokeLater*(new Runnable() {

public void run() {

try {

UserFrame frame = new UserFrame();

frame.setVisible(true);

frame.loadTable();

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

1. Klik dua kali pada tombol save untuk menambahkan event handlers actionPerformed, kemudian isi dengan program berikut

public void actionPerformed(ActionEvent e) {

User user = new User();

user.setNama(txtName.getText());

user.setUsername(txtUsername.getText());

user.setPassword(txtPassword.getText());

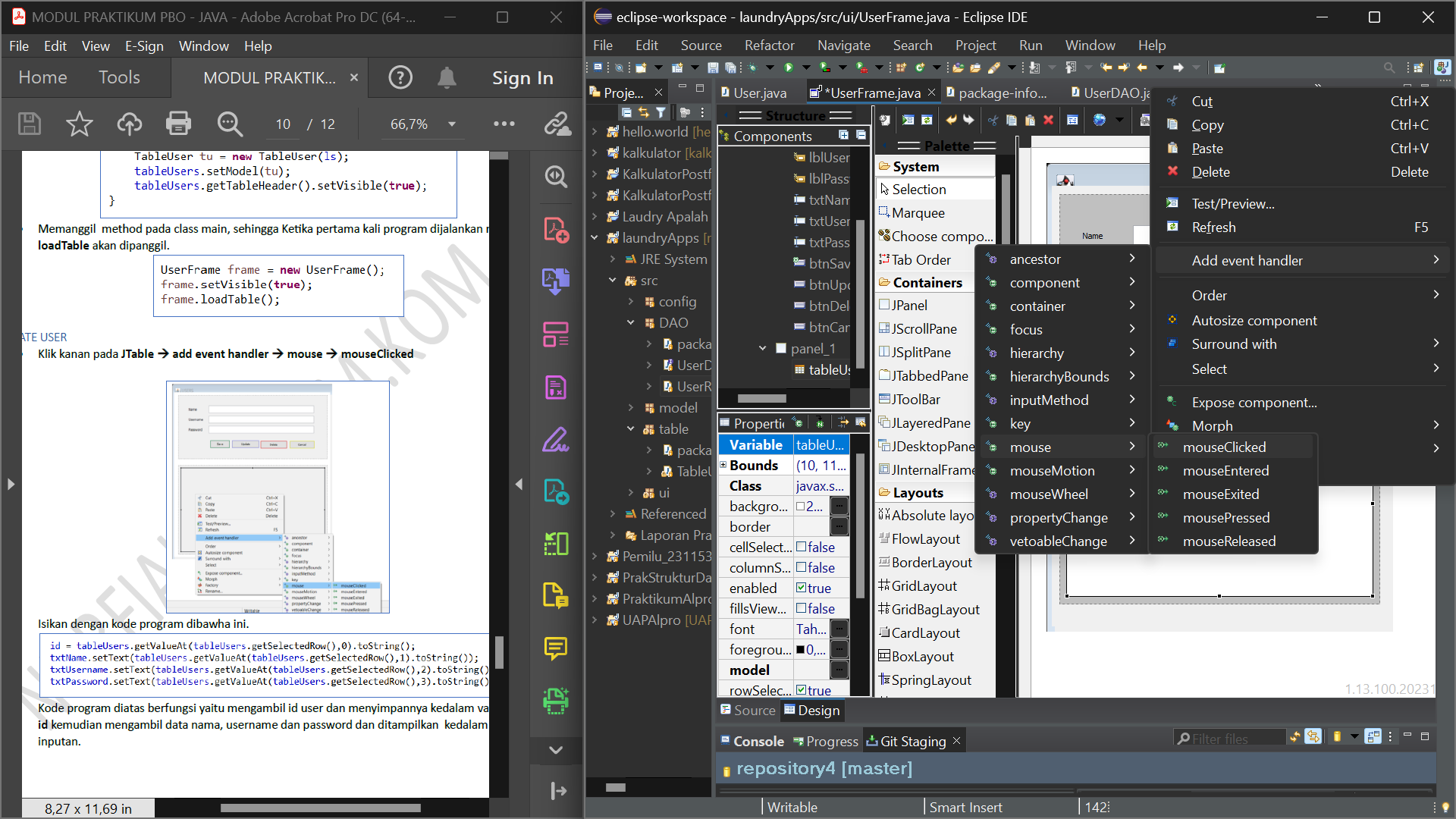
usr.save(user);

reset();

loadTable();

}

1. Klik kanan pada JTable > add event handler > mouse > mouseClicked



1. Isi mouseClicked dengan program berikut

public void mouseClicked(MouseEvent e) {

id =tableUsers.getValueAt(tableUsers.getSelectedRow(),0).toString();

txtName.setText(tableUsers.getValueAt(tableUsers.getSelectedRow(),1).toString());

txtUsername.setText(tableUsers.getValueAt(tableUsers.getSelectedRow(),2).toString());

txtPassword.setText(tableUsers.getValueAt(tableUsers.getSelectedRow(),3).toString());

}

Kode program diatas berfungsi yaitu mengambil id user dan menyimpannya kedalam variable id kemudian mengambil data nama, username dan password dan ditampilkan kedalam form inputan.

1. Klik dua kali pada tombol delete untuk menambahkan event handlers actionPerformed, kemudian isi dengan program berikut

public void actionPerformed(ActionEvent e) {

if(id != null) {

usr.delete(id);

reset();

loadTable();

} else {

JOptionPane.*showMessageDialog*(null,

"SIlahkan pilih data yang akan dihapus");

}

}

1. Klik dua kali pada tombol update untuk menambahkan event handlers actionPerformed, kemudian isi dengan program berikut

public void actionPerformed(ActionEvent e) {

User user = new User();

user.setNama(txtName.getText());

user.setUsername(txtUsername.getText());

user.setPassword(txtPassword.getText());

user.setId(id);

usr.update(user);

reset();

loadTable();

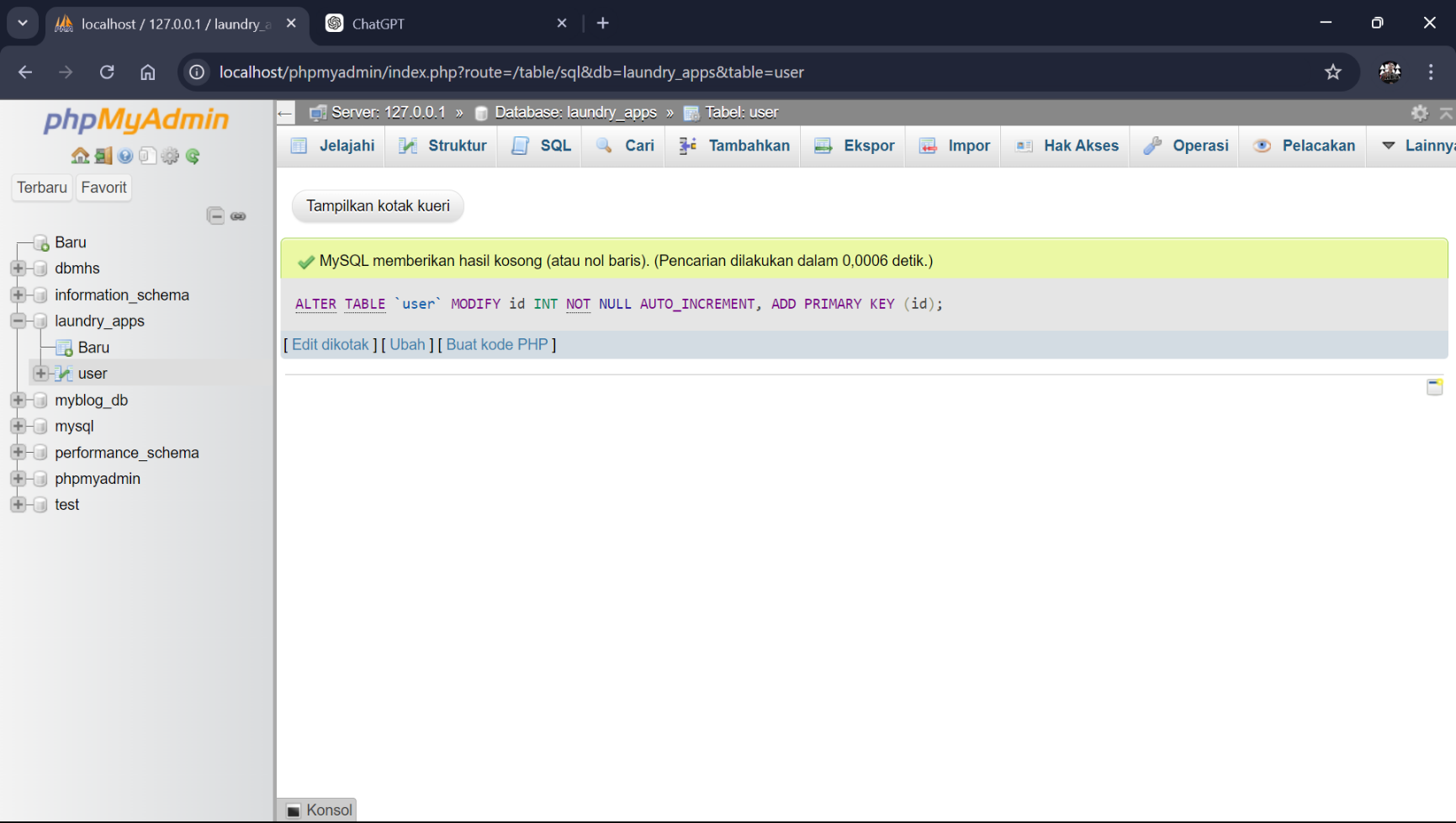
}

1. Saat dijalankan akan muncul errror yang mengatakan bahwa kolom id tidak memiliki default value, maka modifikasi pada table user pada database dengan menjalankan sintaks SQL berikut

ALTER TABLE `user`

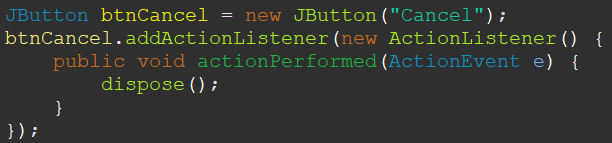
MODIFY id INT NOT NULL AUTO\_INCREMENT,

ADD PRIMARY KEY (id);



Modifikasi yang dilakukan adalah menjadikan kolom id sebagai PRIMARY KEY dan menambahkan atribut AUTO\_INCREMENT ke kolom id agar ketika menambahkan data ke kolom lain, data di kolom id secara otomatis terisi dengan tipe data int yang bertambah 1 ketika menginputkan data baru ke tabel.

1. Klik dua kali pada tombol cancel untuk menambahkan event handlers actionPerformed, kemudian isi dengan perintah dispose() untuk menutup GUI ketika tombol cancel ditekan.



1. Fungsi CRUD sudah bisa digunakan

