**LAPORAN PRAKTIKUM PEMOGRAMAN BERORIENTASI OBJEK (PBO)**

**MEMBUAT FUNGSI CRUD (Create, Read, Update, Delete)**

**USER DENGAN DATABASE MYSQL**



OLEH :

NIA RAMADHANI

(2311531006)

DOSEN PENGUMPU : Nurfiah, S.ST, M.Kom.

**FAKULTAS TEKNOLOGI INFORMASI**

**DEPARTEMEN INFORMATIKA**

**UNIVERSITAS ANDALAS**

# Pendahuluan

# CRUD adalah empat operasi dasar yang digunakan untuk mengelola data dalam aplikasi berbasis basis data. CRUD terdiri dari:

# - Create (Membuat): Operasi ini digunakan untuk menambahkan data baru ke dalam tabel di database. Dalam konteks user, ini berarti menambahkan pengguna baru.

# - Read (Membaca): Operasi ini digunakan untuk membaca atau menampilkan data yang ada di dalam tabel. Dalam hal ini, data pengguna ditampilkan dari database.

# - Update (Memperbarui): Operasi ini memungkinkan kita untuk mengubah atau memperbarui data yang sudah ada dalam tabel. Ini berguna untuk memperbarui informasi pengguna, seperti nama, alamat, atau kata sandi.

# - Delete (Menghapus): Operasi ini digunakan untuk menghapus data dari tabel, dalam hal ini menghapus pengguna dari sistem.

# MySQL adalah salah satu sistem manajemen basis data relasional (RDBMS) yang paling populer dan sering digunakan dalam aplikasi berbasis web. Dengan MySQL, kita dapat menyimpan data dalam tabel yang saling berelasi, memungkinkan kita untuk dengan mudah mengelola dan memanipulasi data pengguna.

# Dalam konteks praktik pengembangan sistem manajemen user berbasis database, implementasi CRUD pada MySQL biasanya melibatkan langkah-langkah berikut:

# 1. Menghubungkan ke database MySQL: Biasanya dilakukan menggunakan koneksi berbasis JDBC (untuk Java) atau library lain tergantung bahasa pemrogramannya.

# 2. Membuat tabel user: Tabel ini akan menyimpan informasi pengguna, seperti `id`, `username`, `password`, `email`, dan atribut lain yang diperlukan.

# 3. Melakukan operasi CRUD: Dengan memanfaatkan query SQL seperti `INSERT` untuk Create, `SELECT` untuk Read, `UPDATE` untuk Update, dan `DELETE` untuk Delete.

# Penggunaan operasi CRUD pada database MySQL sangat penting untuk pengembangan aplikasi yang mengelola data dinamis, seperti sistem login, pendaftaran pengguna, atau sistem manajemen data lainnya.

# Pendahuluan ini memberi gambaran dasar tentang bagaimana CRUD digunakan dalam manajemen user pada database MySQL dan pentingnya operasi ini dalam pengelolaan data pada aplikasi berbasis basis data.

# 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

# Alat

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

# 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.



Logo MySQL

**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.

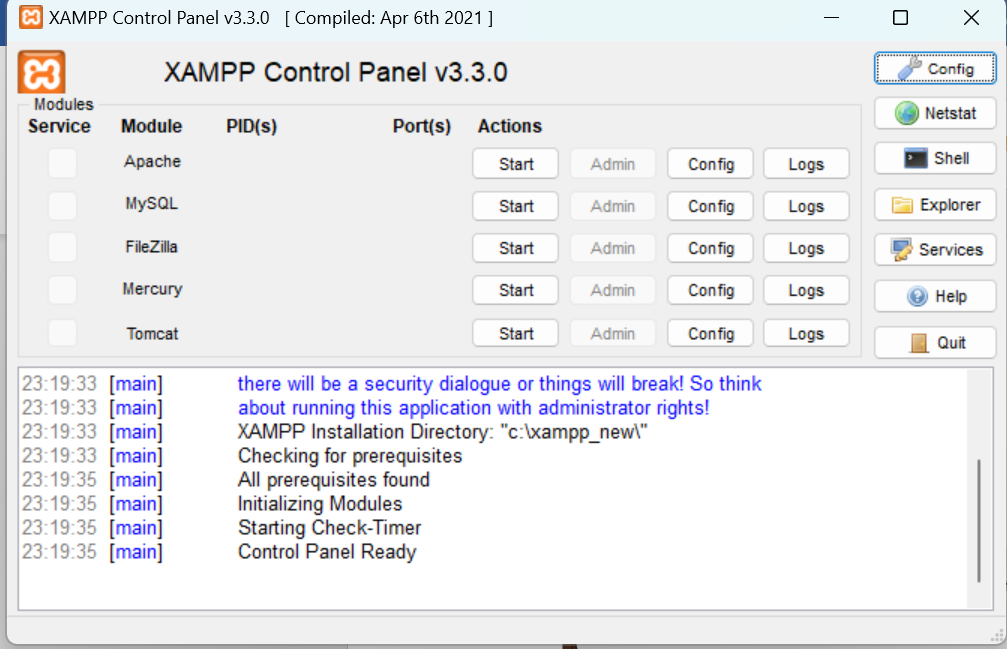
**Interface** dalam Bahasa java yaitu mendefinisikan beberapa method abstrak yang harus diimplementasikan oleh class yang akan menggunakannya.

**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.

# Langkah-langkah

## Install XAMPP

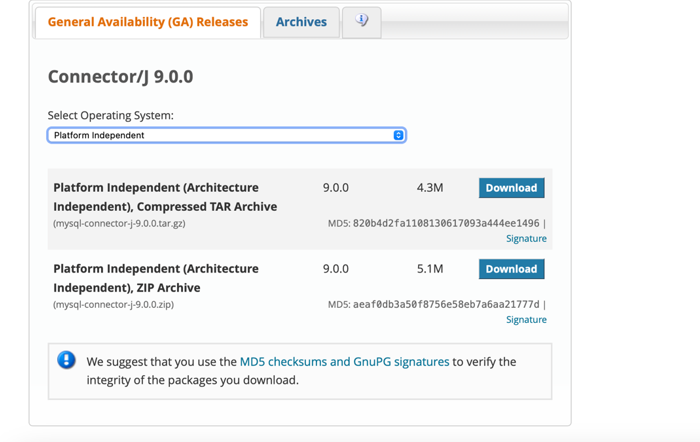
* + - Download XAMPP dari pada link berikut : <https://www.apachefriends.org/>
    - Setelah didownload install xampp pada computer/laptop masing-masing
    - Jalankan xampp dan aktifkan apache dan mysql



## Menambahkan MySQL Connector

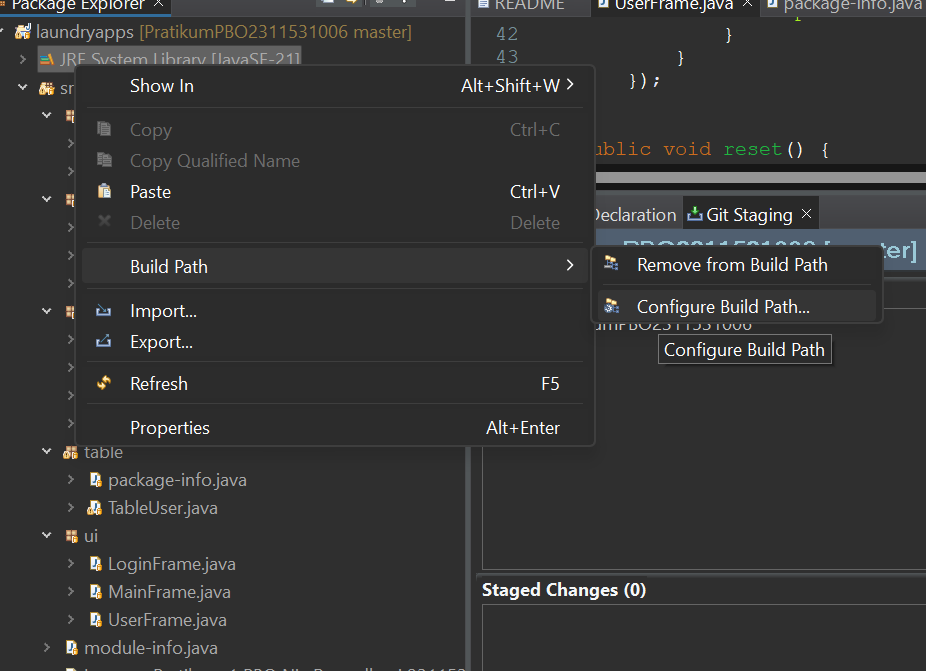
Aplikasi java agar dapat terhubung dengan database MySQL membutuhkan sebuah driver yaitu MySQL Connection, berikut ini Langkah-langkah membuat koneksi Database MySQL.

* + - Download MySQL connection pada link berikut <https://dev.mysql.com/downloads/connector/j/>

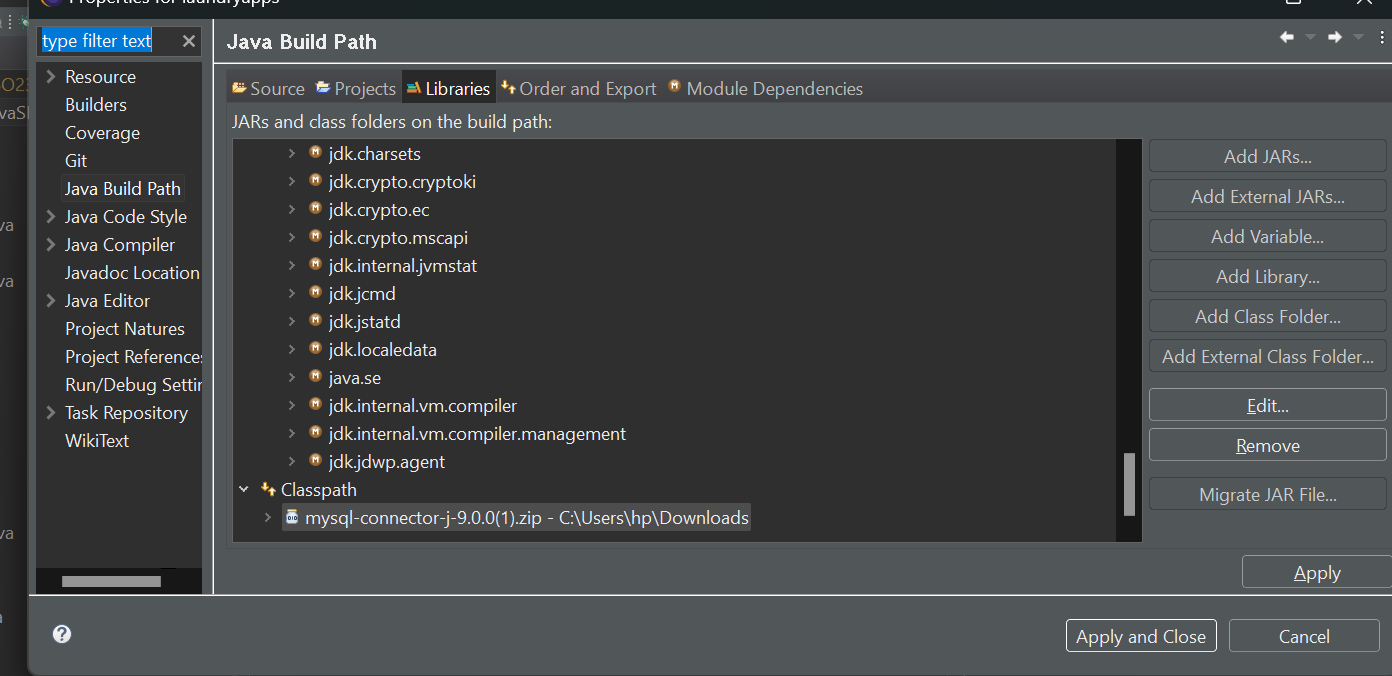


Pilih file yang berekstensi .zip

* + - Menambahkan MySQL Connector kedalam project dengan cara klik kanan directory **JRE System Library**  **Built Path**  **Configure Build Path**



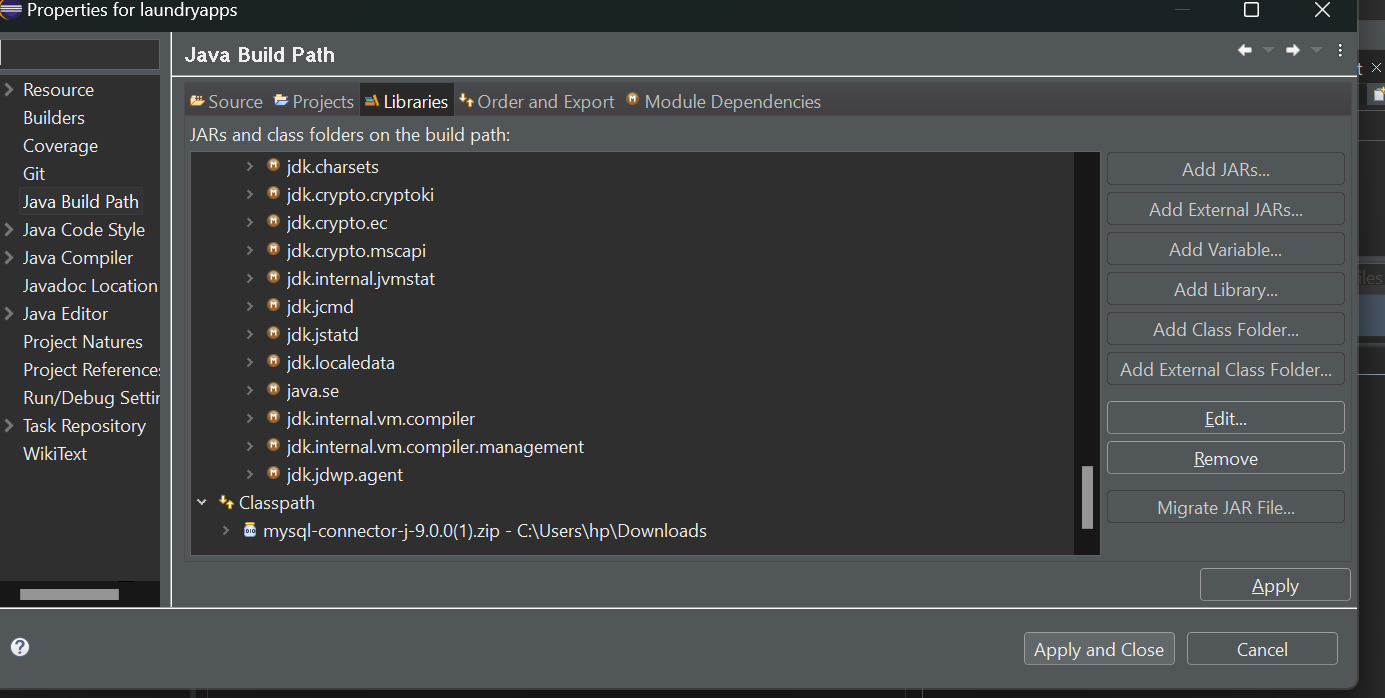
Selanjutnya pilih **Libraries**  **Classpath**

****

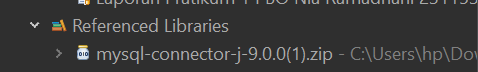
Tambahkan file MySQL Connector dengan cara klik **Add External JARs**

****

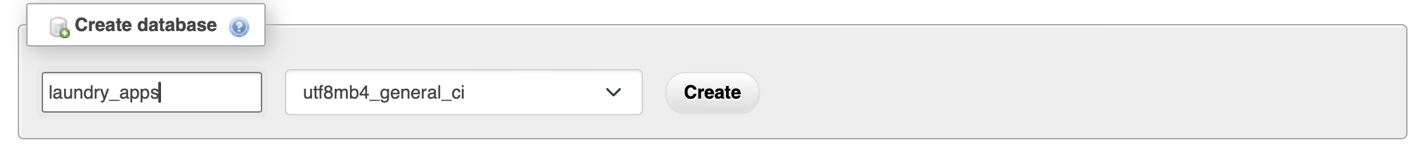
dan pilih file yang telah didownlod dan pilih **Apply and Close**.

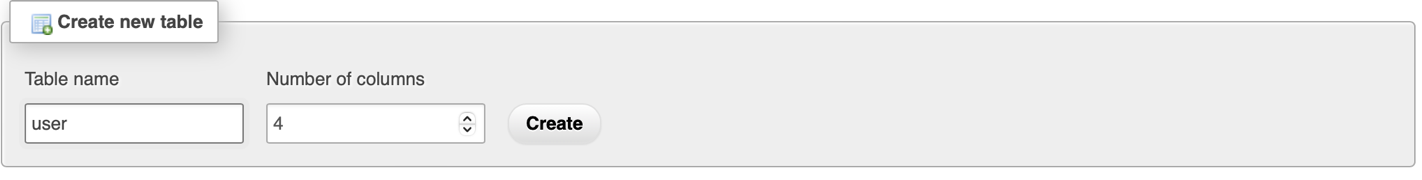


Jika berhasil menambahkan MySQL Connector maka akan generate folder Referenced Libraries pada project yang berisi MySQL Connector.

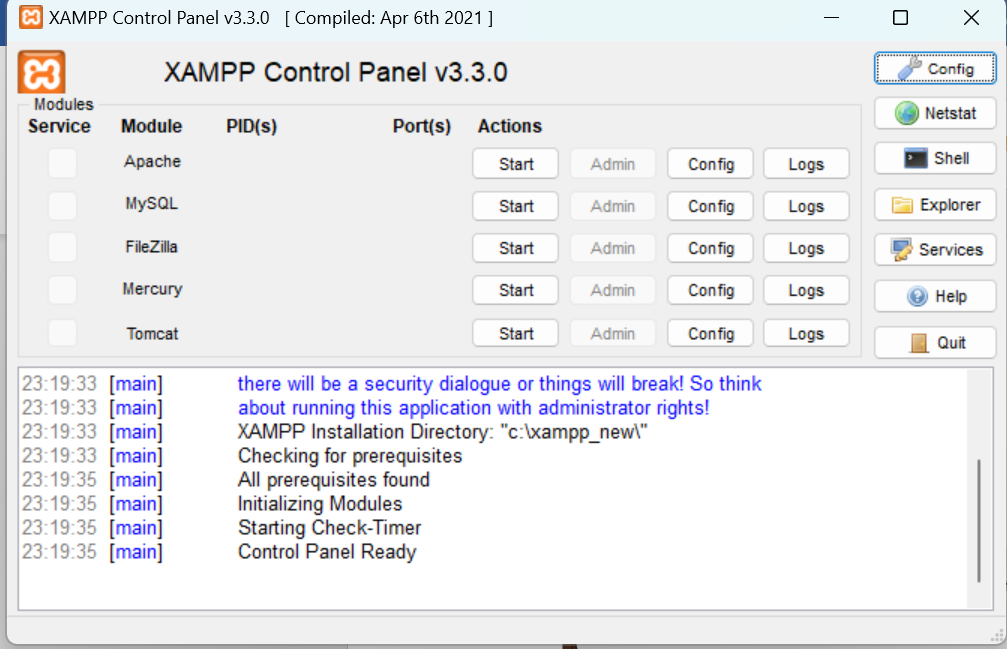


## Membuat Database dan Table User

* + - 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.

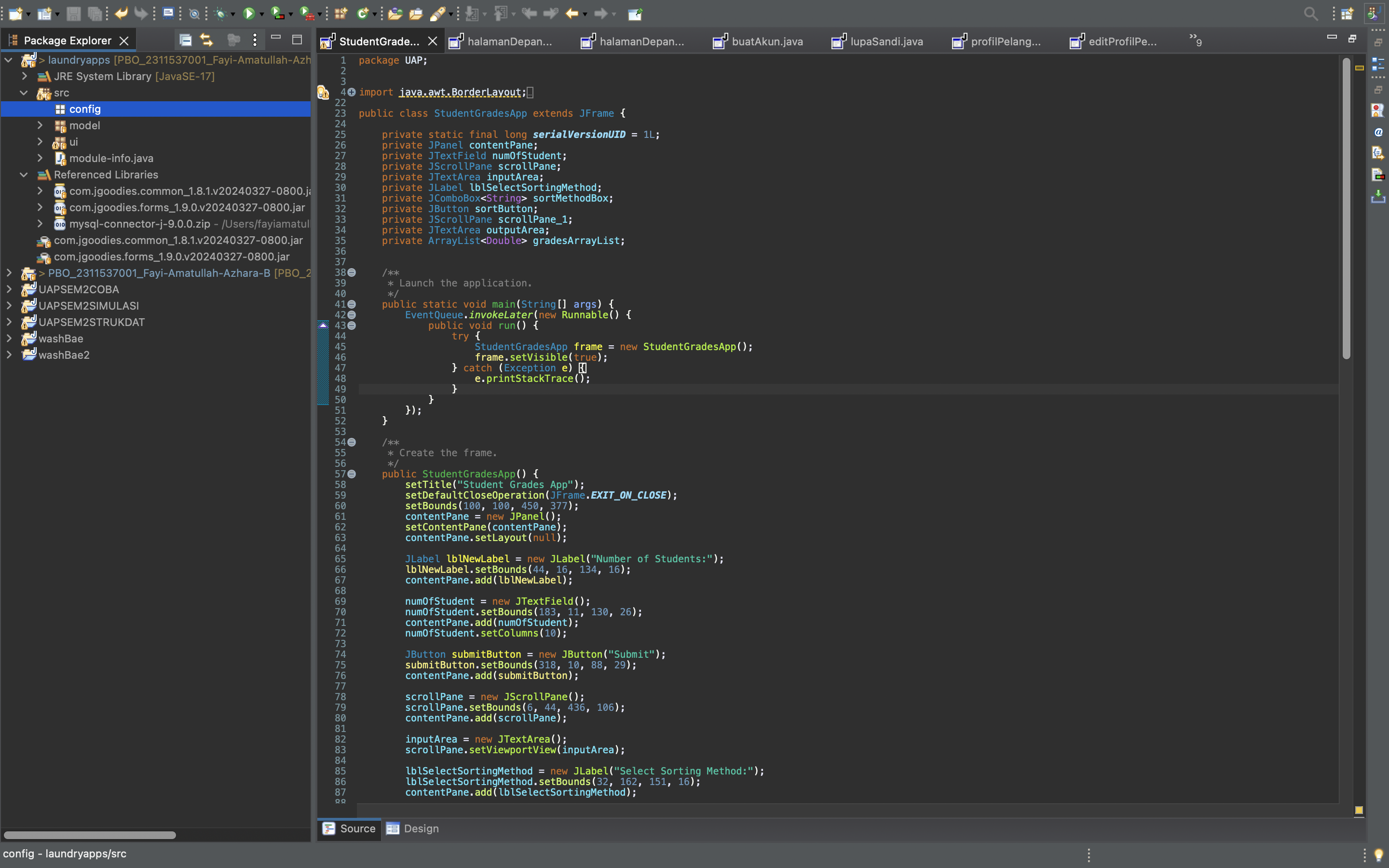


Isi seperti gambar diatas dan klik save.

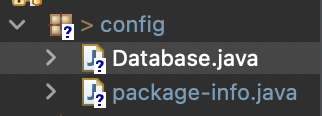
## Membuat Koneksi ke Database MySQL

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

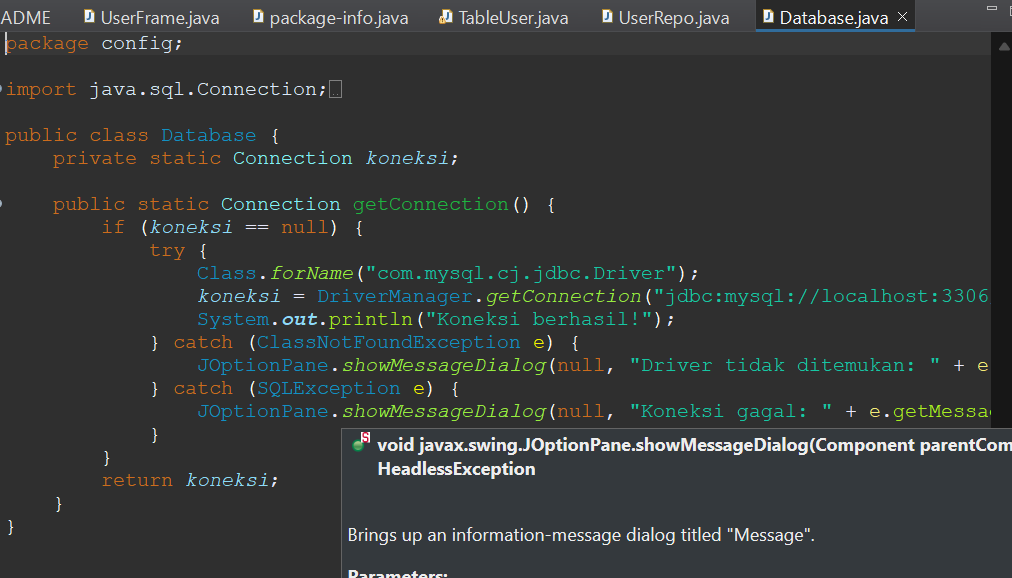
* + - Buat package baru dengan nama config, package ini yang akan digunakan untuk membuat konfigurasi aplikasi yang akan dibuat termasuk dengan konfigurasi database.



* + - Buat class baru dengan nama Database,

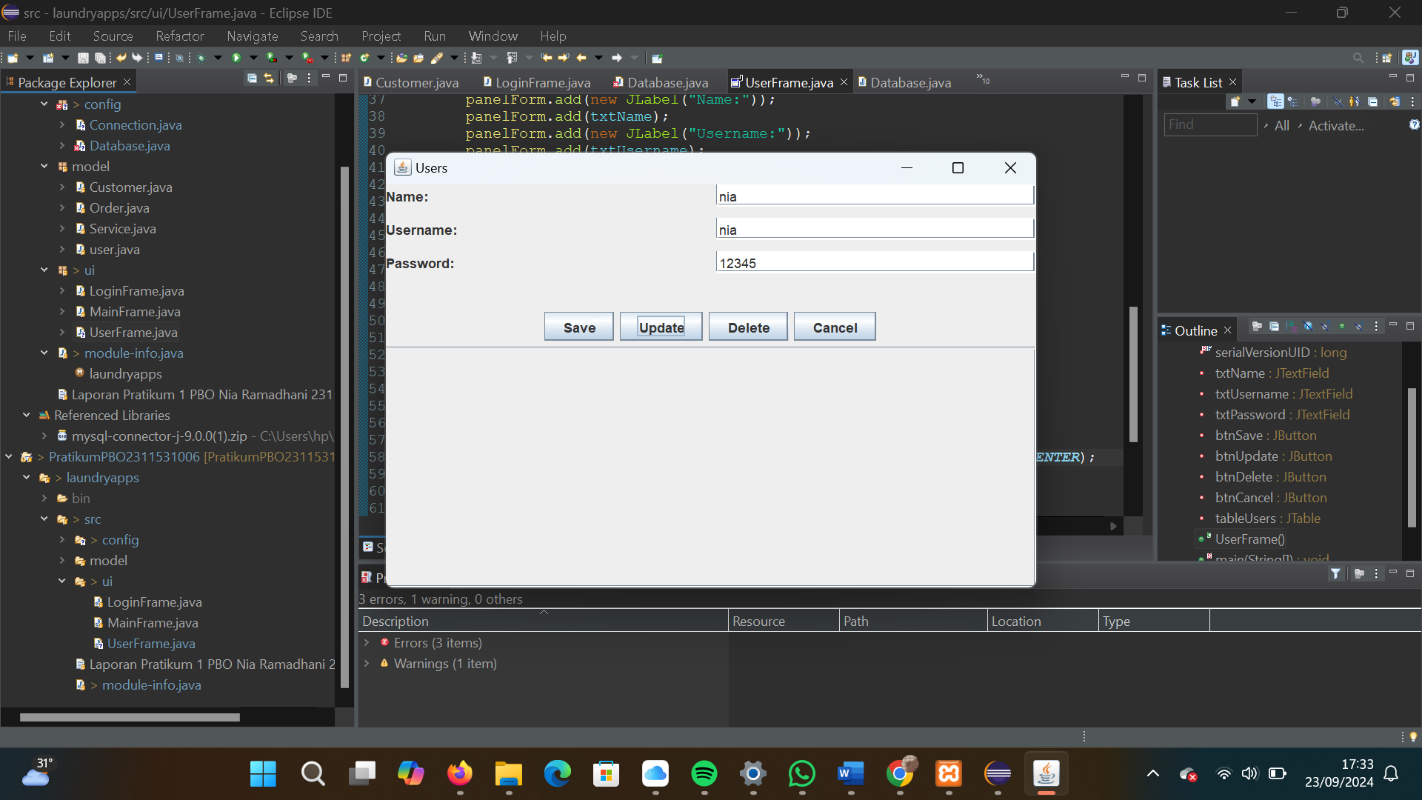


kemudian konfigurasi sesuai dengan kode program beriku



## Membuat Tampilan CRUD User

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



Keterangan :

|  |  |  |
| --- | --- | --- |
| **Component** | **Variable** | **Keterangan** |
| JTextField | txtName | Name |
| JTextField | txtUsername | Username |

|  |  |  |
| --- | --- | --- |
| JTextField | txtPassword | Password |
| JButton | btnSave | Save |
| JButton | btnUpdate | Update |
| JButton | btnDelete | Delete |
| JButton | btnCancel | Cancel |
| JTable | tableUsers | Table Users |

## 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.

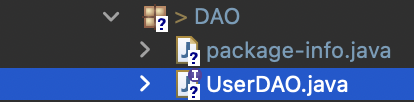


## Membuat Fungsi DAO

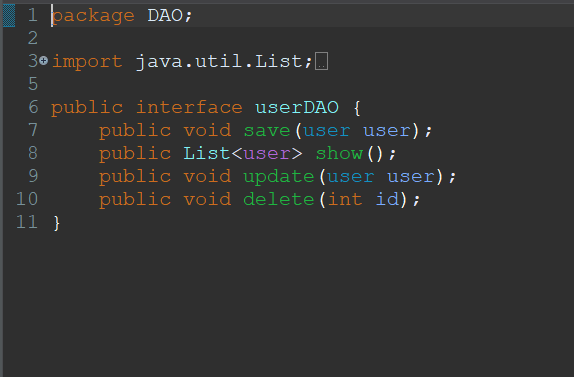
* + - Buat package baru dengan nama DAO



* + - Buat class Interface baru dengan nama **UserDAO**,

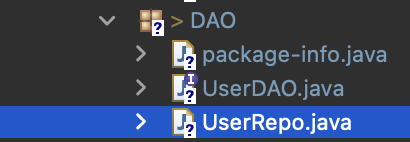


kemudian isikan dengan kode program berikut.



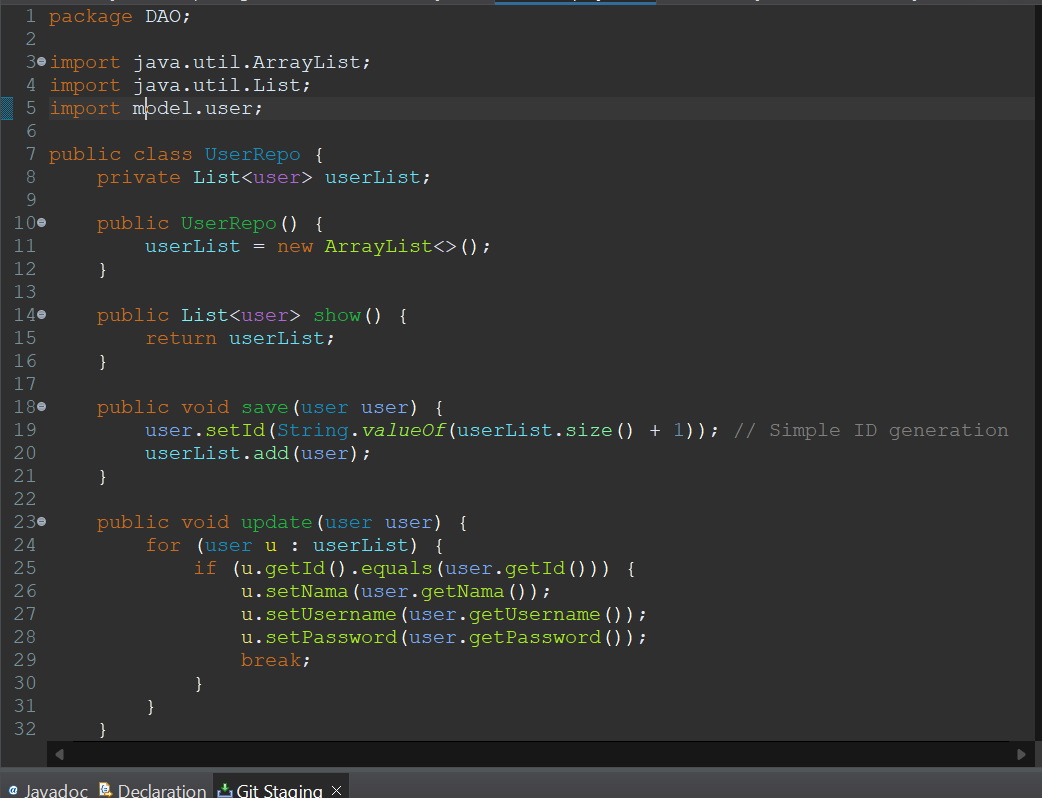
Terdapat method **save, show, delete dan update**. Method pada class interface digunakan sebagai method utama yang wajib diimplementasikan pada class yang menggunakannya.

## Menggunakan Fungsi DAO

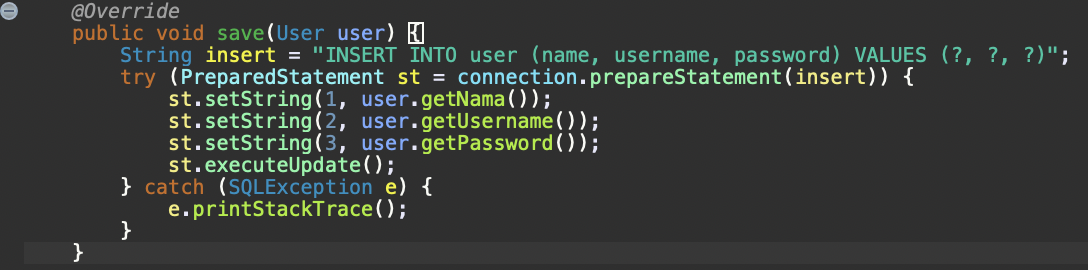
* + - Buat class baru pada package DAO dengan nama **UserRepo** yang mana akan digunakan untuk mengimplementasikan DAO yang telah dibuat.  
      
    - Implementasikan UserDao dengan kata kunci **implements**

****

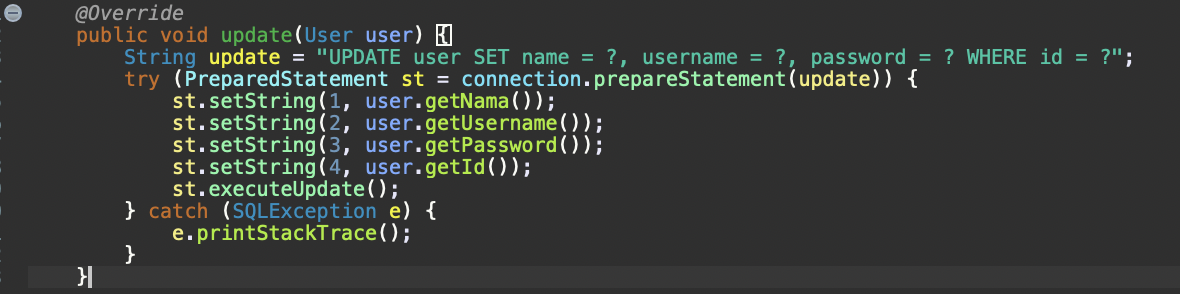
* + - Masukkan kode di bawah ini



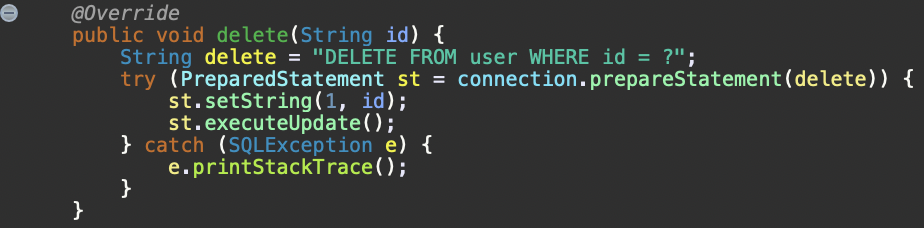
* + - untuk method **save**, isikan dengan kode program berikut.



* + - Membuat method **show** untuk mengambil data dari database
    - Membuat method **update** yang digunakan untuk mengubah data.

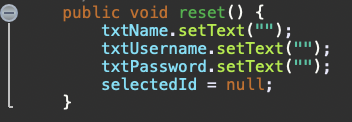


* + - Membuat method **delete** yang digunakan untuk menghapus data.



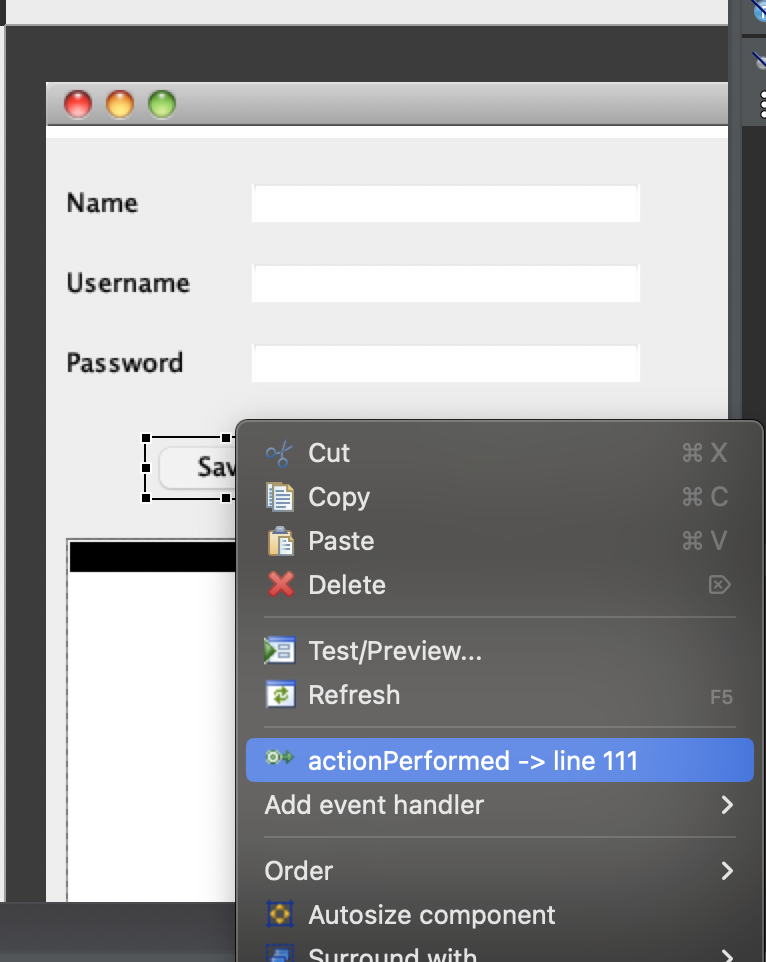
## Menggunakan Fungsi CRUD DAO pada GUI

* + - method digunakan untuk menghapus value inputan Ketika suatu proses berhasil dilakukan, buat method **reset** pada JFrame seperti kode program dibawah ini.

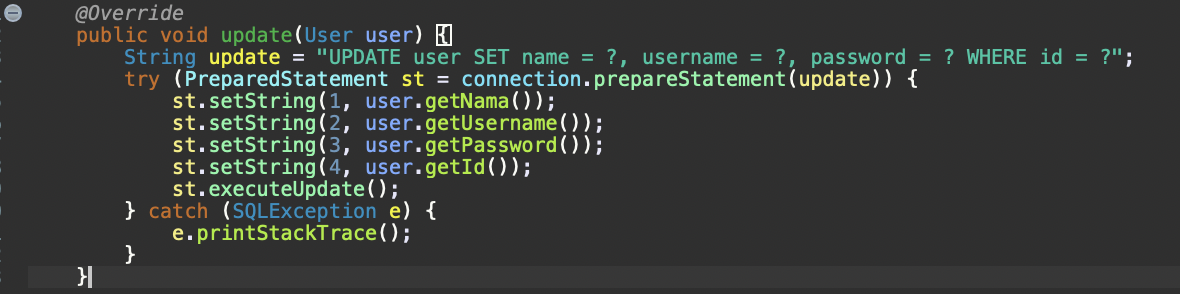


CREATE USER

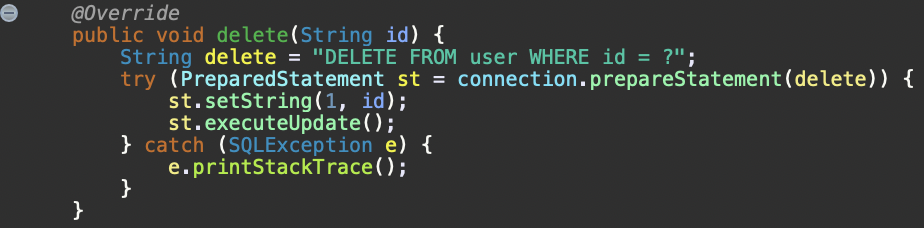
* + - Klik kanan pada tombol **save**  **add event handlers**  **actionPerformed**

****

* + - Membuat method **update** yang digunakan untuk mengubah data.

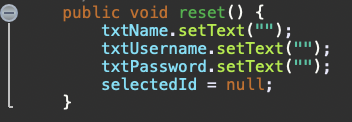


* + - Membuat method **delete** yang digunakan untuk menghapus data.



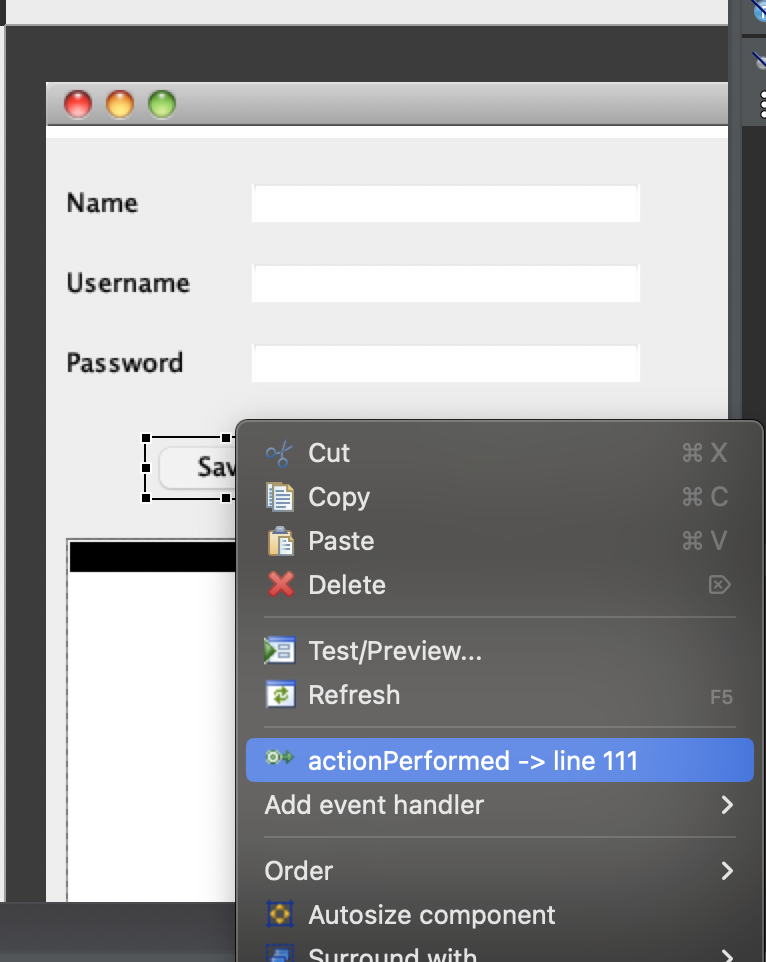
## Menggunakan Fungsi CRUD DAO pada GUI

* + - method digunakan untuk menghapus value inputan Ketika suatu proses berhasil dilakukan, buat method **reset** pada JFrame seperti kode program dibawah ini.

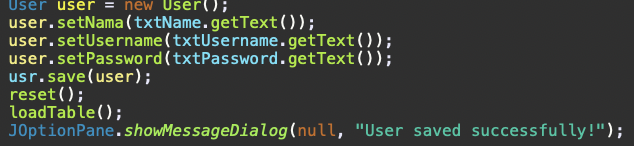


CREATE USER

* + - Klik kanan pada tombol **save**  **add event handlers**  **actionPerformed**

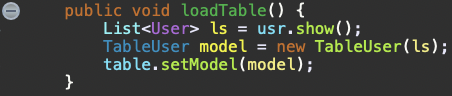
****

kemdian isi dengan kode program berikut.



READ USERS

* + - Buat method dengan nama **loadTable**() kemudian isikan dengna kode program berikut.



* + - Memanggil method pada class main, sehingga Ketika pertama kali program dijalankan maka

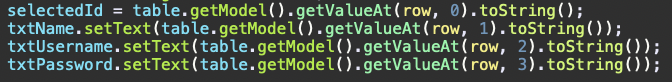
**loadTable** akan dipanggil.



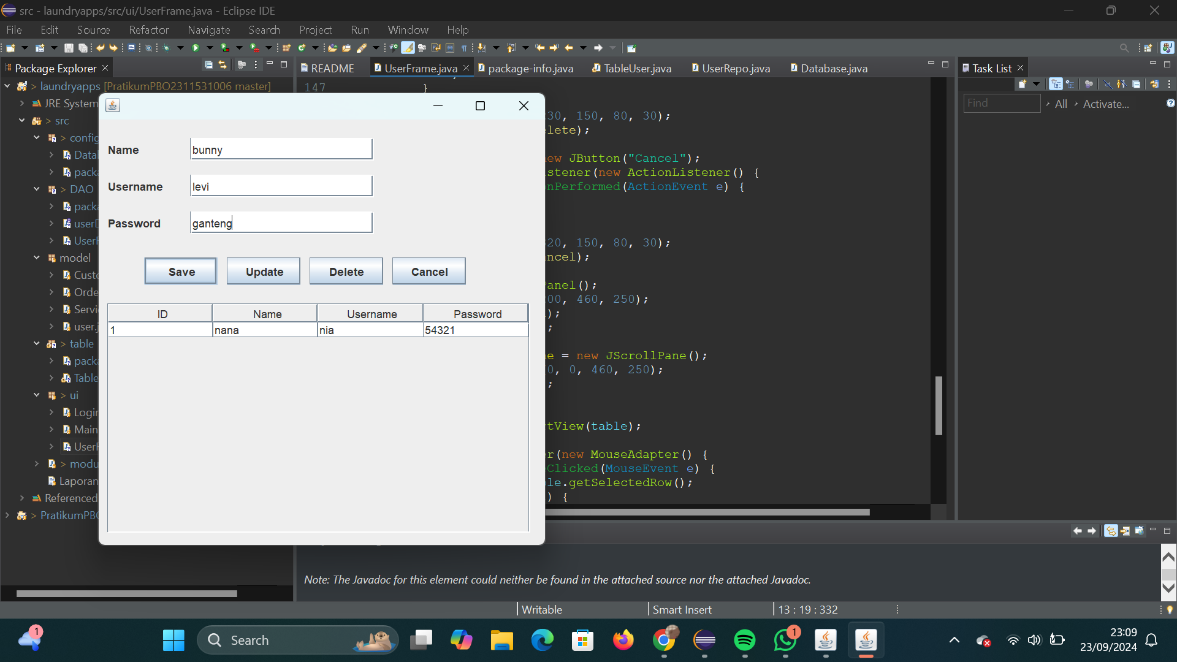
UPDATE USER

* + - Klik kanan pada **JTable**  **add event handler**  **mouse**  **mouseClicked**

Isikan dengan kode program dibawha ini.



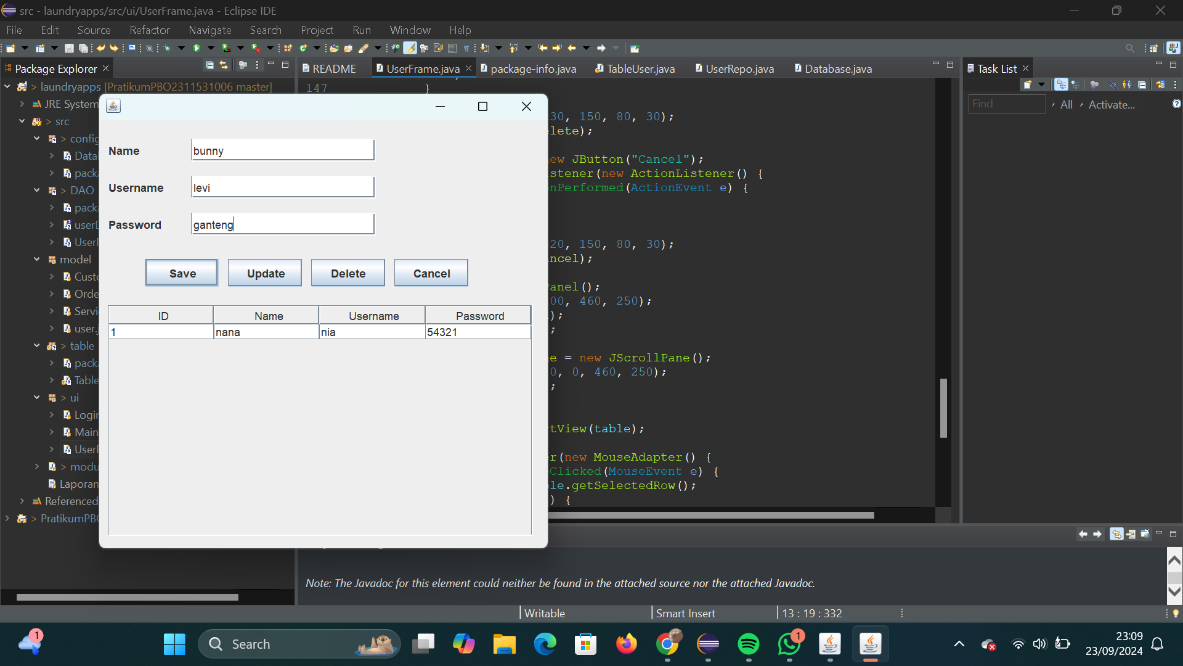
Klik salah aatu isi table maka akan secara otomatis tampil pada form inputan.



* + - Klik kanan tombol **update**  **add event handler**  **action**  **actionPeformed** dan isikan dengan kode program berikut.

DELETE USER

* + - Klik salah satu data pada JTable
    - Klik kanan tombol **delete**  **add event handler**  **action**  **actionPerformed**



* Untuk tombol **save**, digunakan untuk memasukkan data baru   
  
* Untuk tombol **update**, digunakan untuk memperbarui data

# Penutup

Melalui praktikum ini, saya telah mempelajari dan mempraktikkan pembuatan fungsi CRUD menggunakan Java dan MySQL dengan menerapkan konsep Pemrograman Berorientasi Objek. Saya juga telah berhasil membuat GUI untuk mengelola data pengguna serta memanfaatkan fungsi DAO untuk modularitas dan pemisahan logika. Dengan pemahaman yang didapatkan, saya diharapkan mampu mengimplementasikan konsep-konsep ini pada proyek-proyek pengembangan perangkat lunak di masa mendatang, terutama yang melibatkan pengelolaan data berbasis database.

Kesimpulan

* Penerapan **CRUD (Create, Read, Update, Delete)**, MySQL, dan pola desain **DAO (Data Access Object)** adalah elemen penting dalam pengembangan aplikasi berbasis basis data. Dalam sistem pengelolaan data, CRUD adalah fondasi utama yang memungkinkan kita untuk melakukan operasi dasar pada basis data seperti menambah, menampilkan, memperbarui, dan menghapus data. Dalam implementasinya, MySQL merupakan salah satu sistem manajemen basis data relasional (RDBMS) yang paling populer, berfungsi untuk mengelola dan menyimpan data dengan efisien dan aman.
* **XAMPP** adalah perangkat lunak berbasis open-source yang menggabungkan beberapa komponen utama seperti Apache (web server), MySQL (database server), PHP, dan Perl, yang memudahkan pengembang dalam membuat dan menjalankan server lokal untuk menguji aplikasi sebelum di-deploy secara langsung ke server produksi. XAMPP menawarkan lingkungan yang lengkap dan terintegrasi untuk pengembangan web, terutama dalam mengelola basis data dengan MySQL dan menjalankan aplikasi berbasis web.
* Pada konteks pengelolaan data pengguna (user management), operasi CRUD menjadi hal yang sangat vital. Operasi **Create** memungkinkan kita menambahkan data baru ke tabel pengguna, **Read** memungkinkan kita mengambil dan menampilkan data, **Update** memungkinkan kita memperbarui informasi pengguna yang sudah ada, dan **Delete** digunakan untuk menghapus data yang tidak diperlukan lagi. Dalam hal ini, query SQL seperti INSERT, SELECT, UPDATE, dan DELETE adalah dasar dari interaksi antara aplikasi dan MySQL.
* Namun, untuk menjaga keteraturan, struktur, dan modularitas kode dalam pengembangan aplikasi yang lebih kompleks, penerapan pola desain **DAO** menjadi sangat penting. DAO adalah pola desain yang memisahkan logika bisnis aplikasi dari interaksi langsung dengan basis data, memberikan interface yang bersih dan terisolasi untuk mengelola data. Dengan memanfaatkan DAO, developer tidak perlu memahami detail teknis tentang bagaimana query SQL dieksekusi, karena DAO bertindak sebagai penghubung yang menyederhanakan operasi data.
* Penerapan DAO dalam aplikasi berfungsi sebagai perantara yang mengabstraksi seluruh operasi database, menjadikan aplikasi lebih **terstruktur, terorganisir, dan mudah di-maintain**. DAO mengatur koneksi ke database, eksekusi query, dan manajemen transaksi dengan cara yang memungkinkan kita untuk memodifikasi logika database tanpa harus mengubah kode logika bisnis di bagian lain aplikasi. Misalnya, dalam sistem manajemen pengguna, DAO dapat berinteraksi dengan objek **User** dan menangani penyimpanan, pengambilan, dan penghapusan data pengguna dari basis data MySQL tanpa mempengaruhi bagian lain dari aplikasi.
* Dengan menggunakan **XAMPP** sebagai lingkungan server lokal, pengembang dapat dengan mudah membuat dan menguji aplikasi berbasis MySQL. Melalui **phpMyAdmin**, salah satu tool yang disediakan oleh XAMPP, pengembang dapat mengelola database secara visual. Ini mempermudah operasi seperti membuat tabel, memasukkan data, atau menjalankan query SQL secara langsung tanpa harus menggunakan baris perintah. XAMPP menyediakan platform yang fleksibel bagi pengembang untuk men-deploy aplikasi lokal secara cepat dan efisien.
* Dalam hal teknis, implementasi DAO memerlukan **interface DAO** yang mendefinisikan metode-metode CRUD yang akan digunakan dalam aplikasi. Setelah interface ini ditentukan, **kelas implementasi DAO** kemudian diisi dengan logika untuk menjalankan query-query SQL yang sesuai menggunakan JDBC (Java Database Connectivity) untuk bahasa Java, atau library lain sesuai dengan bahasa pemrograman yang digunakan. Misalnya, pada operasi "Create", kelas DAO akan menjalankan query INSERT INTO, sedangkan pada operasi "Read", DAO akan menjalankan query SELECT untuk mengambil data dari tabel.
* DAO juga berfungsi untuk mempermudah pengelolaan koneksi database. Dengan membuat kelas koneksi yang terpisah, seperti kelas MySQLConnection, pengembang hanya perlu mengelola satu koneksi ke database yang dapat digunakan secara konsisten di seluruh aplikasi. Ini tidak hanya membuat kode lebih mudah dibaca dan dipelihara, tetapi juga mengurangi risiko kesalahan teknis yang mungkin terjadi jika pengelolaan koneksi dilakukan secara manual di berbagai bagian aplikasi.
* Dari segi keamanan dan efisiensi, penerapan DAO juga memberikan beberapa keuntungan. Misalnya, penggunaan **PreparedStatement** pada implementasi DAO di Java, selain lebih aman karena mencegah serangan SQL Injection, juga lebih efisien karena query SQL yang sering digunakan dapat di-precompile. Hal ini juga memungkinkan penanganan kesalahan yang lebih baik karena DAO dapat menangani pengecualian (exception) secara terpusat, membuat aplikasi lebih robust dan mudah di-debug.
* Dengan menggunakan pendekatan DAO dan XAMPP, pengembang dapat dengan mudah membuat aplikasi yang scalable dan maintainable. DAO memungkinkan kita untuk menambahkan fungsionalitas baru ke dalam aplikasi tanpa harus mengubah seluruh kode yang sudah ada. Misalnya, jika di masa depan kita ingin mengganti MySQL dengan sistem basis data lain, kita hanya perlu mengubah kelas DAO tanpa harus merombak keseluruhan aplikasi.
* Secara keseluruhan, kombinasi antara **CRUD**, **MySQL**, **DAO**, dan **XAMPP** adalah pendekatan yang sangat baik dalam pengembangan aplikasi berbasis basis data. CRUD memberikan dasar untuk mengelola data, MySQL menyediakan sistem manajemen basis data yang efisien, DAO memberikan struktur dan modularitas kode, sementara XAMPP menyediakan platform yang cepat dan mudah digunakan untuk pengujian aplikasi. Implementasi ini menciptakan aplikasi yang lebih terstruktur, aman, dan mudah dikembangkan di masa depan.
* Melalui penggunaan DAO dan XAMPP, pengembang dapat mencapai keterpisahan antara logika bisnis dan logika database, yang pada akhirnya memudahkan pemeliharaan dan pengembangan lebih lanjut. Implementasi ini cocok untuk proyek skala kecil hingga besar, menjadikannya pilihan ideal untuk pengembangan aplikasi berbasis basis data di lingkungan lokal sebelum melakukan deploy ke server produksi.