

LAPORAN PRAKTIKUM PEKAN 2
PEMOGRAMAN BERORIENTASI OBJEK



Oleh :

KEVIN ANDHIKA

NIM 2311532005

MATA KULIAH PEMOGRAMAN BERORIENTASI OBJEK

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

FAKULTAS TEKNOLOGI INFORMASI

DEPARTEMEN INFORMATIKA

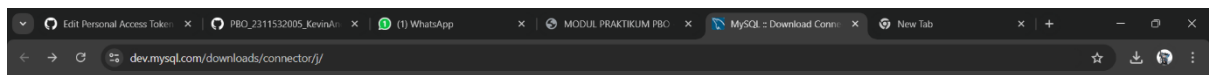
UNIVERSITAS ANDALAS

PADANG, SEPTEMBER 2024

PEMBAHASAN

1. Download MySQL Community pada link berikut

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



MySQL Community Downloads

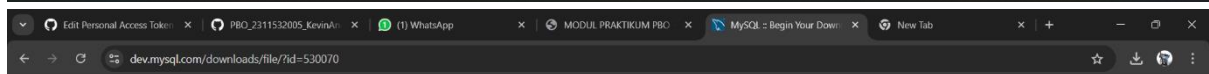
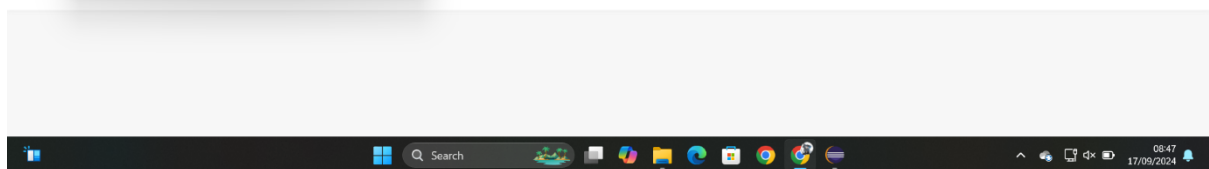
Connector/J

General Availability (GA) Releases Archives

Connector/J 9.0.0

Select Operating System:

- Platform Independent
- Select Operating System...
- Ubuntu Linux
- Debian Linux
- SUSE Linux Enterprise Server
- Red Hat Enterprise Linux / Oracle Linux
- Fedora
- Platform Independent
- Source Code



MySQL Community Downloads

Login Now or Sign Up for a free account.

An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system

Login »
using my Oracle Web account

Sign Up »
for an Oracle Web account

MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can signup for a free account by clicking the Sign Up link and following the instructions.

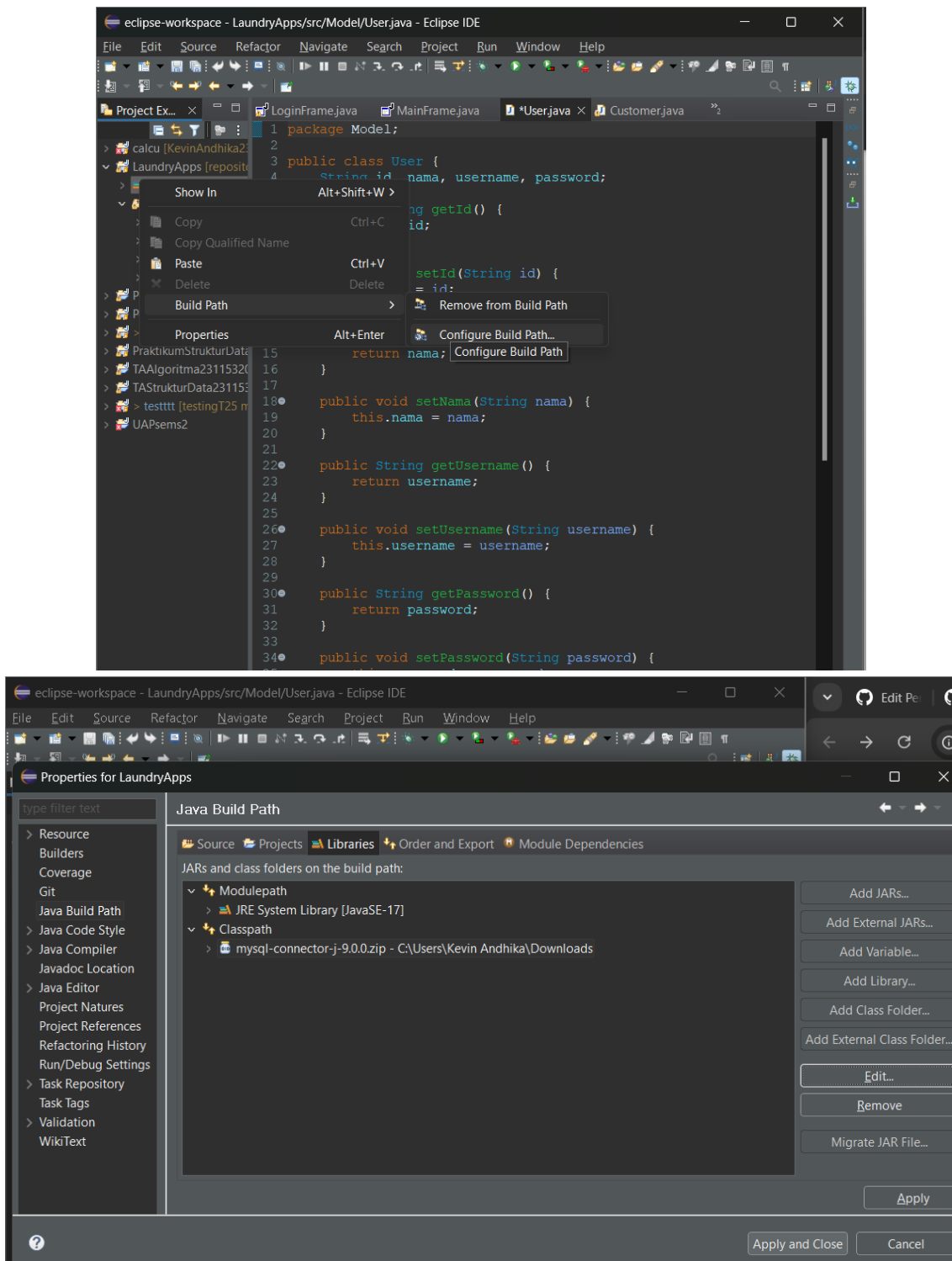
No thanks, just start my download.

ORACLE © 2024 Oracle

Privacy / Do Not Sell My Info / Terms of Use / Trademark Policy / Cookie Preferences

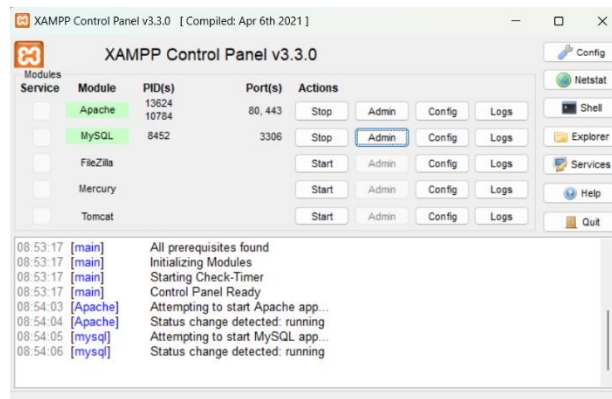


2. Tambahkan MySQL Connector ke dalam java project dengan mengklik **JRE System Library → Built Path → Configure Build Path**

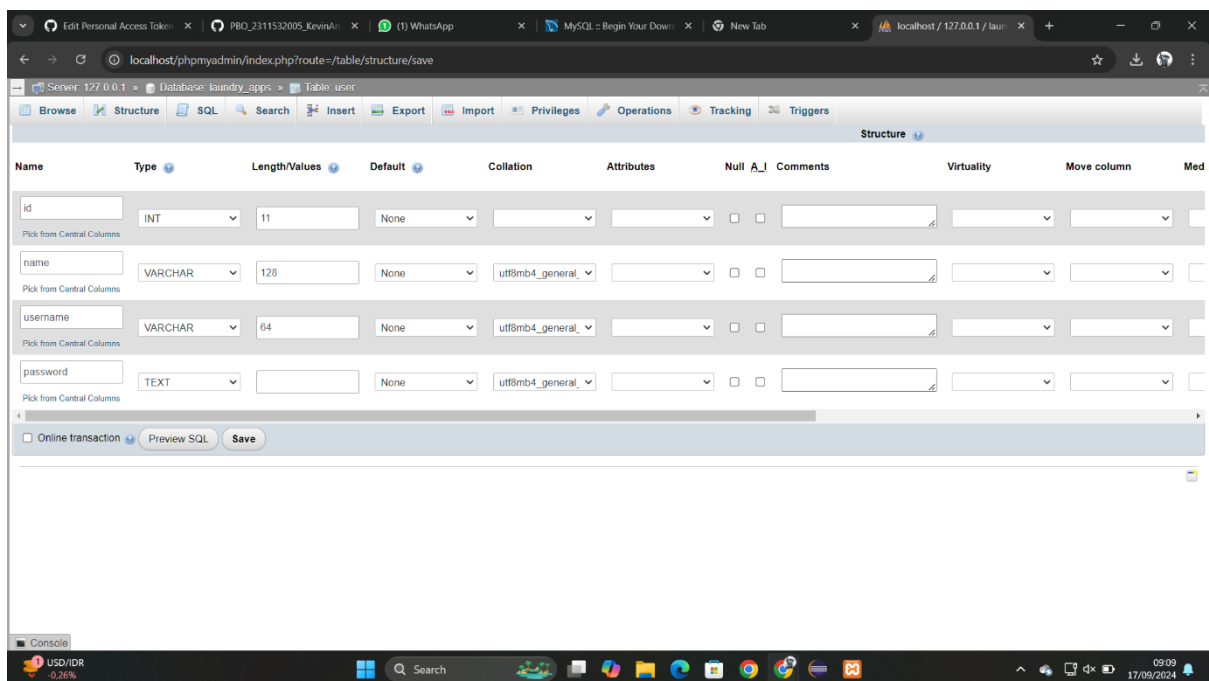
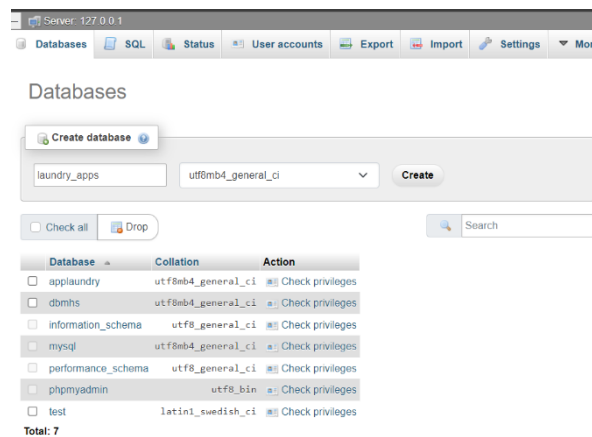


Lalu tambahkan file yang didownload sebelumnya dengan cara **Add External JARs → Apply and Close** pada libraries classpath

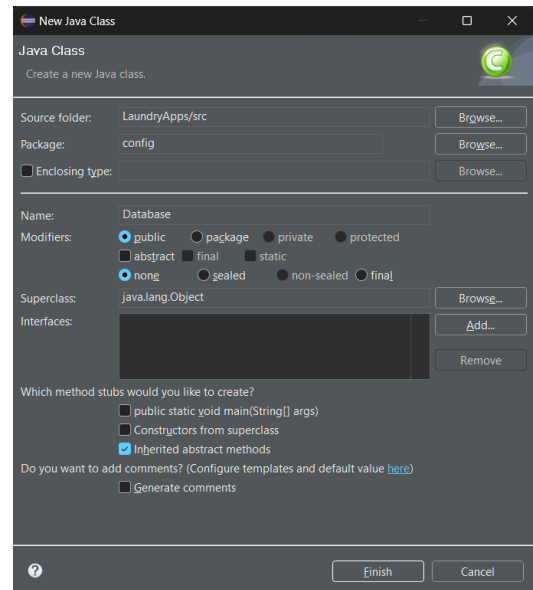
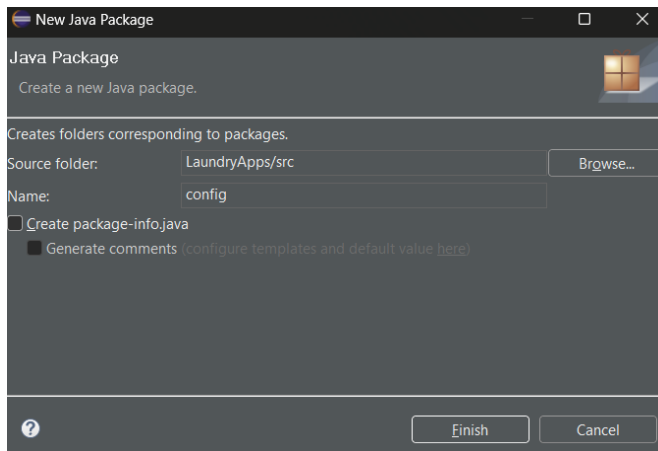
4. Buka phpMyAdmin melalui aplikasi XAMPP dengan mencentang **Apache** dan **MySQL** lalu menekan tompol admin pada **MySQL**



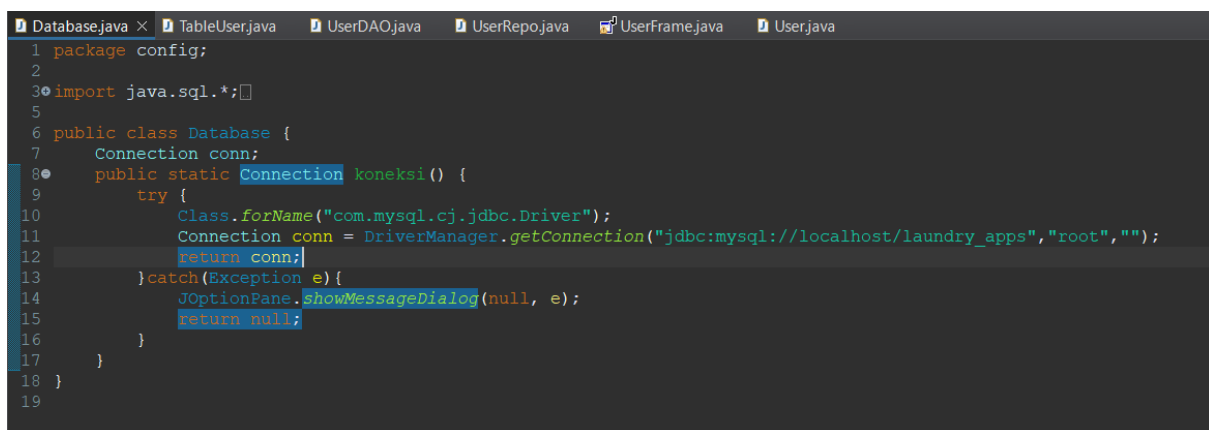
5. Setelah masuk ke dalam phpMyAdmin buat Database bernama **laundry_apps** dengan tipe data sebagai berikut



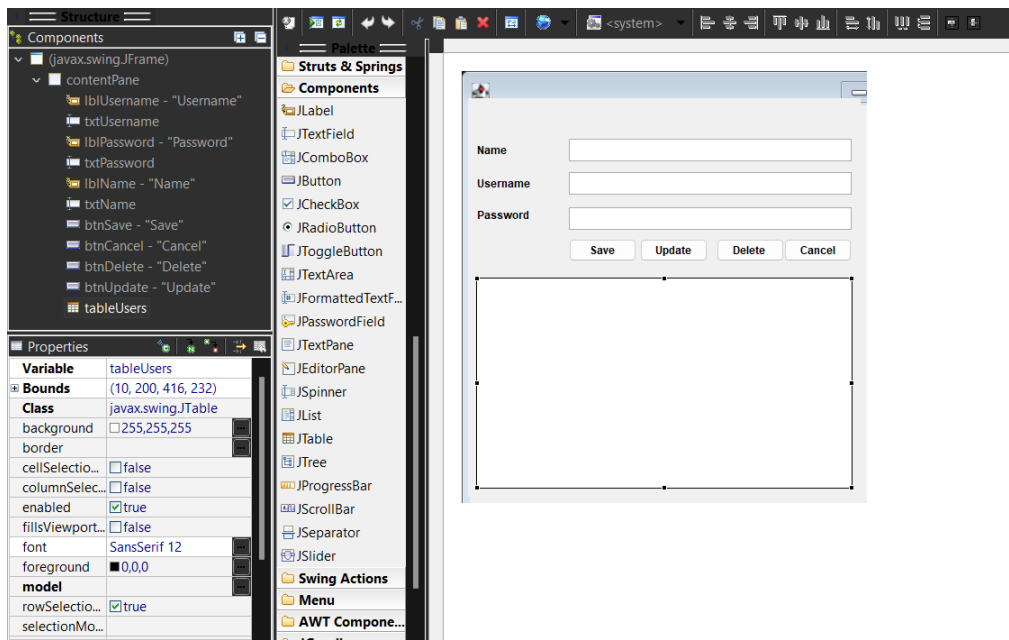
6. Buat package config untuk membuat konfigurasi aplikasi dengan database, di dalamnyaaa buat class **Database** untuk isi konfigurasinya



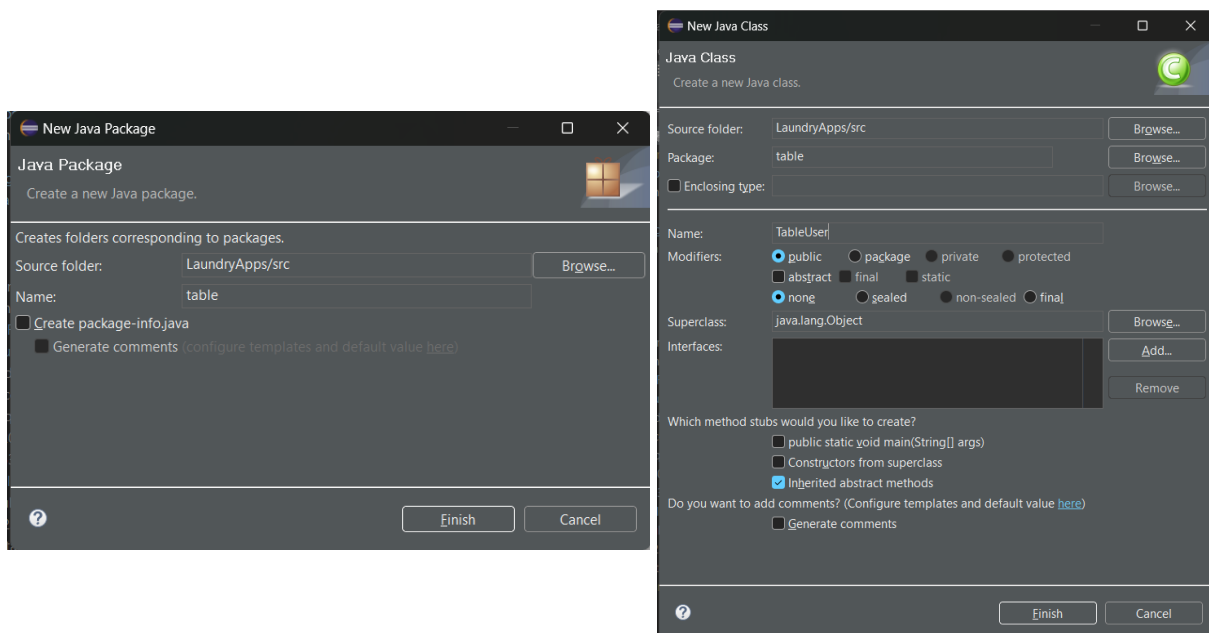
7. Berikut kode program dari class Database



8. Buat file baru pada package UI dengan nama JFrame **UserFrame**



9. Tambahkan package baru **table** lalu class **TableUser** beserta kode programnya

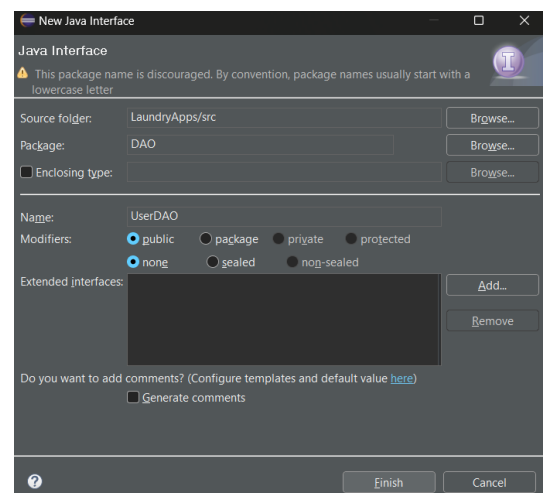
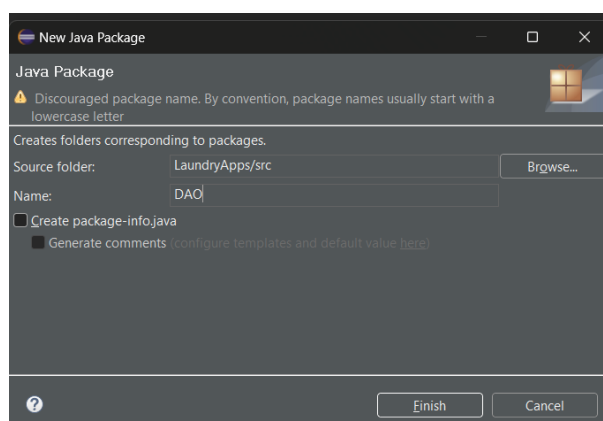


```

Database.java TableUser.java X UserDao.java UserRepo.java UserFrame.java User.java
7 public class TableUser extends AbstractTableModel {
8     List<User> ls;
9     private String[] columnNames = {"Id", "Name", "Username", "Password"};
10    public TableUser(List<User> ls) {
11        this.ls = ls;
12    }
13
14    public int getRowCount() {
15        return ls.size();
16    }
17
18    public int getColumnCount() {
19        return 4;
20    }
21
22    public String getColumnName(int column) {
23        return columnNames[column];
24    }
25
26    public Object getValueAt(int rowIndex, int columnIndex) {
27        switch(columnIndex) {
28            case 0:
29                return ls.get(rowIndex).getId();
30            case 1:
31                return ls.get(rowIndex).getNama();
32            case 2:
33                return ls.get(rowIndex).getUsername();
34            case 3:
35                return ls.get(rowIndex).getPassword();
36            default:
37                return null;
38    }

```

10. Tambahkan lagi package baru **DAO** lalu class **UserDAO** beserta kode programnya

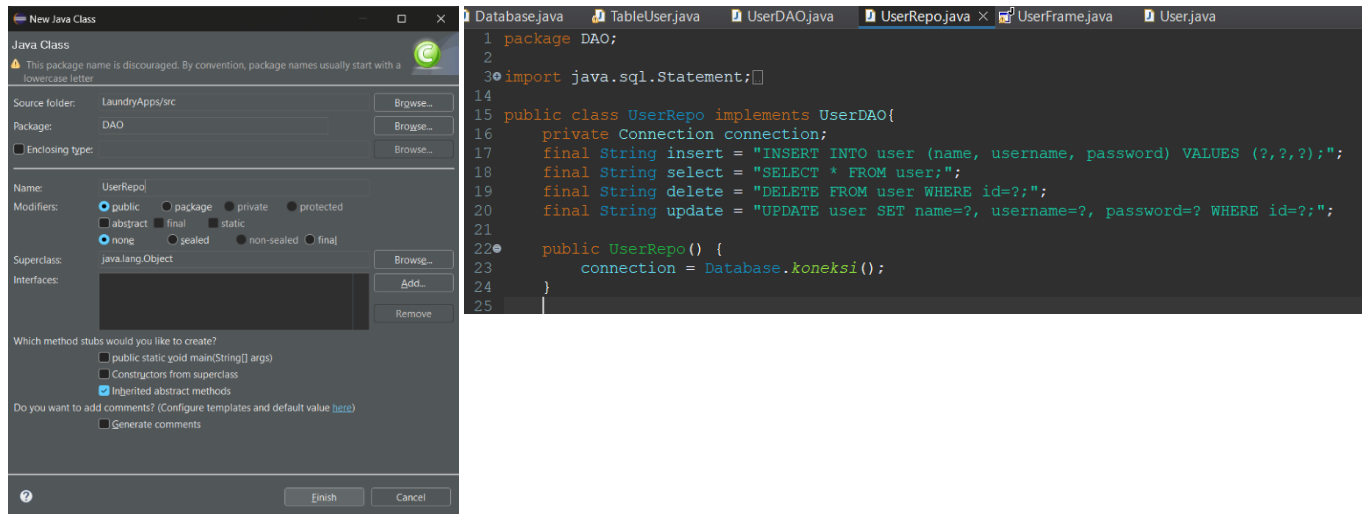


```

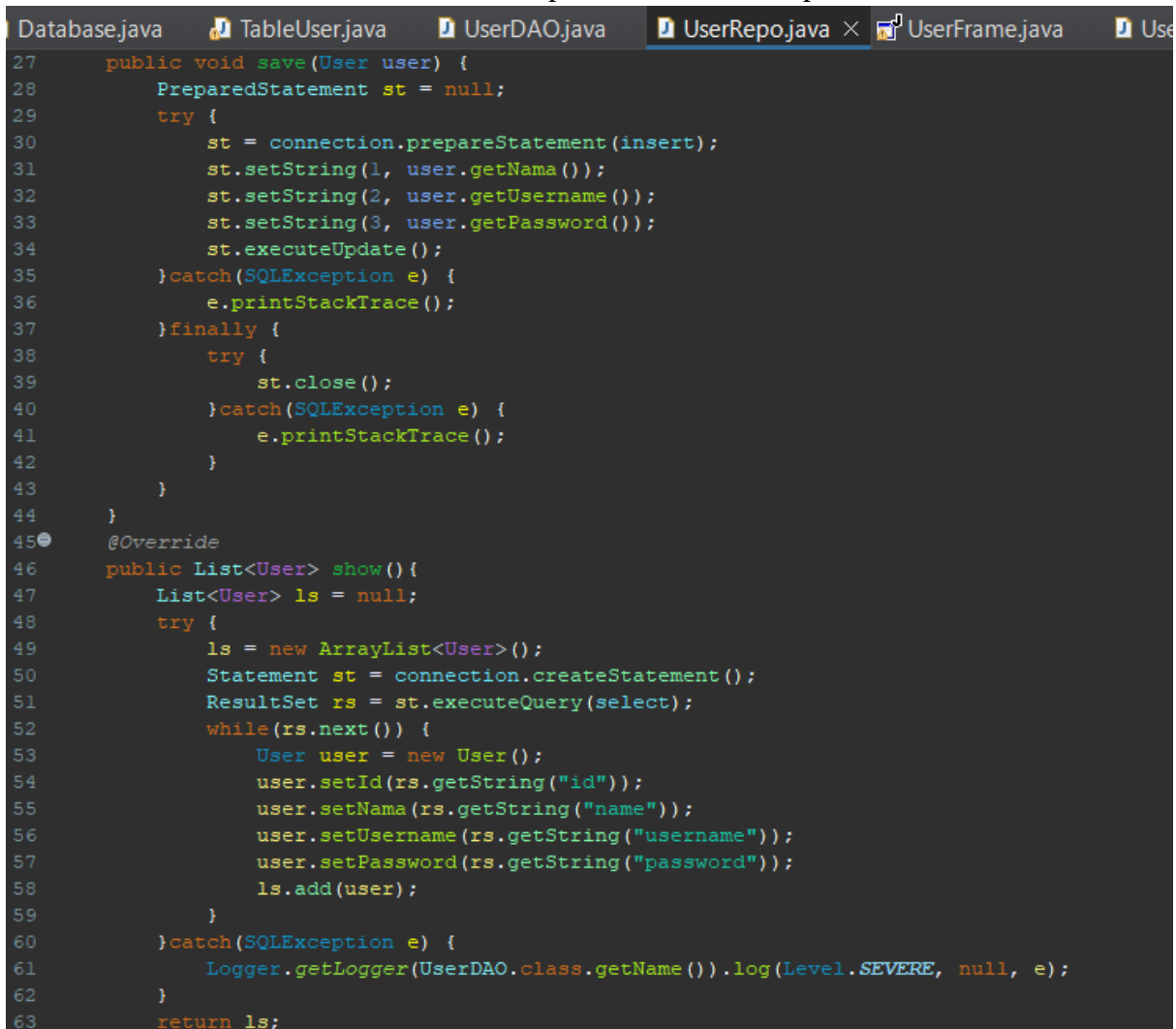
Database.java *UserFrame.java *TableUser.java *UserDAO.java X
1 package DAO;
2
3 import java.util.List;
4 import Model.User;
5
6 public interface UserDAO {
7     void save(User user);
8     List<User> show();
9     void delete(String id);
10    void update(User user);
11 }
12

```

11. Pada package DAO sebelumnya buat lagi class baru dengan nama **UserRepo** beserta dengan codingannya instalasi connection



Beserta membuat method untuk proses save, show, update, dan delete




```

Database.java  TableUser.java  UserDao.java  UserRepo.java ×  UserFra
65  @Override
66  public void update(User user) {
67      PreparedStatement st = null;
68      try {
69          st = connection.prepareStatement(update);
70          st.setString(1, user.getNama());
71          st.setString(2, user.getUsername());
72          st.setString(3, user.getPassword());
73          st.setString(4, user.getId());
74          st.executeUpdate();
75      } catch (SQLException e) {
76          e.printStackTrace();
77      } finally {
78          try {
79              st.close();
80          } catch (SQLException e) {
81              e.printStackTrace();
82          }
83      }
84  }
85  @Override
86  public void delete(String id) {
87      PreparedStatement st = null;
88      try {
89          st = connection.prepareStatement(delete);
90          st.setString(1, id);
91          st.executeUpdate();
92      } catch (SQLException e) {
93          e.printStackTrace();
94      } finally {
95          try {
96              st.close();
97          } catch (SQLException e) {
98              e.printStackTrace();
99          }
100     }
101 }
102

```

12. Kembali pada class UI UserFrame, buat instance serta method **reset** dan **loadTable** pada bagian luar main method

```
Database.java TableUser.java UserDao.java UserRepo.java *UserFrame.java X
1 package UI;
2
3 import java.awt.EventQueue;
28
29 public class UserFrame extends JFrame {
30
31     private static final long serialVersionUID = 1L;
32     private JPanel contentPane;
33     private JTextField txtUsername;
34     private JTextField txtPassword;
35     private JTextField txtName;
36     private JTable tableUsers;
37
38     UserRepo usr = new UserRepo();
39     List<User> ls;
40     String id;
41
42     public void reset() {
43         txtName.setText("");
44         txtUsername.setText("");
45         txtPassword.setText("");
46     }
47     public void loadTable() {
48         ls = usr.show();
49         TableUser tu = new TableUser(ls);
50         tableUsers.setModel(tu);
51         tableUsers.getTableHeader().setVisible(true);
52     }
```

13. Pada bagian button, berikan event handler → action performed. Sedangkan pada Jtable, berikan event handler → mouse clicked

Menggunakan Fi
• method d
buat metl

• Membuat

CREATE USER

ls = usr.show();
TableUser tu = new TableUser(ls);
tableUsers.setModel(tu);
tableUsers.getTableHeader().setVisible(true);

Memanggil method pada class main, sehingga loadTable akan dipanggil.

UserFrame frame = new UserFrame();
frame.setVisible(true);
frame.loadTable();

DATE USER

Klik kanan pada JTable → add event handler

program dibawah ini.

```
getValueAt(tableUsers.getRowCount(), 0);  
tableUsers.getValueAt(0, 0);
```

mouseClicked
mouseEntered
mouseExited
mousePressed
mouseReleased

14. Berikut kodingan untuk tiap tombol button

SAVE

```
JButton btnSave = new JButton("Save");
btnSave.setBackground(new Color(0, 255, 0));
btnSave.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        User user = new User();
        user.setNama(txtName.getText());
        user.setUsername(txtUsername.getText());
        user.setPassword(txtPassword.getText());
        reset();
        usr.save(user);
        loadTable();
    }
});
```

UPDATE

```
JButton btnUpdate = new JButton("Update");
btnUpdate.addActionListener(new ActionListener() {
    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();
    }
});
```

DELETE

```
JButton btnDelete = new JButton("Delete");
btnDelete.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        if(id != null) {
            usr.delete(id);
            reset();
            loadTable();
        } else {
            JOptionPane.showMessageDialog(null, "Silahkan pilih data yang akan di hapus");
        }
    }
});
```

JTABLE

```
tableUsers = new JTable();
tableUsers.setToolTipText("");
tableUsers.setFillViewportHeight(true);
tableUsers.setBackground(new Color(255, 255, 255));
tableUsers.setBounds(10, 200, 416, 232);
scrollPane.setViewportView(tableUsers);
tableUsers.addMouseListener(new MouseAdapter() {
    @Override
    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());
    }
});
tableUsers.setFont(new Font("SansSerif", Font.PLAIN, 12));
```

15. Terakhir, tambahkan juga method pada class main agar program loadTable akan dipanggil

```
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();
            }
        }
    });
}
```

HASIL PRAKTIK UM

A screenshot of a user management application window. It features three input fields for 'Name', 'Username', and 'Password'. Below these fields are four buttons: 'Save' (green), 'Update' (blue), 'Delete' (red), and 'Cancel' (grey). At the bottom, there is a table with four columns: 'ID', 'Name', 'Username', and 'Password'. The table is currently empty.

ID	Name	Username	Password
----	------	----------	----------

A screenshot of the same user management application window, but with data entered in the input fields. The 'Name' field contains 'Kevin Andhika', the 'Username' field contains 'Flan', and the 'Password' field contains '12345678'. The buttons and table remain the same as in the previous screenshot.

ID	Name	Username	Password
----	------	----------	----------

A screenshot of the user management application window showing the data saved in the table. The input fields are now empty. The table at the bottom contains one row of data: ID 1, Name Kevin Andhika, Username Flan, and Password 12345678.

ID	Name	Username	Password
1	Kevin Andhika	Flan	12345678

Name

Kevin Andhika

Username

Flan

Password

AFAAHFAHFAHFA

Save

Update

Delete

Cancel

ID	Name	Username	Password
1	Kevin Andhika	Flan	12345678
2	Testing	test01	1234*x
3	Asal Isi	asIX	jhffkfvsd

Name

Username

Password

Save

Update

Delete

Cancel

ID	Name	Username	Password
1	Kevin Andhika	Flan	AFAAHFAHFAHFA
2	Testing	test01	1234*x
3	Asal Isi	asIX	jhffkfvsd

Name

Username

Password

Save

Update

Delete

Cancel

ID	Name	Username	Password
1	Kevin Andhika	Flan	AFAAHFAHFAHFA
2	Testing	test01	1234*x