



Laporan Tugas Pertemuan ke-11

Identitas

- NPM : 223040024
- Nama : Diaz Alfari Rachmad
- Kelas : A
- URL Repository Github :
[https://github.com/Diazalfari/pp2_223040024_A/tree/main/Latihan/Sesi11]

Penjelasan Kode

Jika berkas **tidak memiliki output**, gunakan tabel ini.

mybatis-config.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE configuration
  PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
  "http://mybatis.org/dtd/mybatis-3-config.dtd">
<configuration>
  <environments default="development">
    <environment id="development">
      <transactionManager type="JDBC" />
      <dataSource type="POOLED">
        <property name="driver"
value="com.mysql.cj.jdbc.Driver" />
        <property name="url"
value="jdbc:mysql://localhost:3306/pp2_sesi11" />
        <property name="username" value="root" />
        <property name="password" value="" />
      </dataSource>
    </environment>
  </environments>
  <mappers>
    <mapper class="model.ProductMapper"/>
    <package name="mapper" />
  </mappers>
</configuration>
```

Penjelasan

code diatas merupakan konfigurasi untuk MyBatis, sebuah framework java untuk ORM atau pengelolaan database



User.java

```
package model;

/**
 *
 * @author Diaza
 */
public class User {
    private int id;
    private String name;
    private String email;

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }
}
```

Penjelasan



User.java

code diatas merupakan definisi sebuah kelas Java bernama User, yang berada didalam paket model. Kelas ini digunakan untuk merepresentasikan sebuah entitas atau objek User dalam aplikasi, dengan atribut dan metode terkait.

UserMapper.java

```
package model;

import java.util.List;
import org.apache.ibatis.annotations.*;

/**
 *
 * @author Diaza
 */
public interface UserMapper {
    @Select("SELECT * FROM users")
    List<User> getAllUsers();

    @Insert("INSERT INTO users (name, email) VALUES (#{name},  
#{email})")
    void insertUser(User user);
}
```

Penjelasan

code ini merupakan definisi interface java bernama userMapper, yang berada dalam paket model. Interface ini digunakan dalam kerangka MyBatis untuk memetakan operasi database (SQL) ke metode Java. Tujuannya adalah untuk mempermudah akses manipulasi data di tabel users melalui kode Java.

MyBatisUtil.java

```
package model;

import org.apache.ibatis.io.Resources;
import org.apache.ibatis.session.*;

import java.io.IOException;
```



MyBatisUtil.java

```
/**
 *
 * @author Diaza
 */
public class MyBatisUtil {
    private static SqlSessionFactory sqlSessionFactory;

    static {
        try {
            sqlSessionFactory = new SqlSessionFactoryBuilder()
                .build(Resources.getResourceAsStream("mybatis-config.xml"));
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    public static SqlSession getSqlSession() {
        return sqlSessionFactory.openSession(true);
    }
}
```

Penjelasan

code ini mendefinisikan sebuah kelas Java bernama MyBatisUtil, yang berada didalam paket model. Kelas ini berfungsi sebagai utilitas untuk mengatur dan menyediakan koneksi ke database menggunakan MyBatis. Bertujuan untuk menyederhanakan pembuatan SqlSession yang di gunakan untuk berinteraksi dengan database

UserView.java

```
package view;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionListener;
/**
 *
```



UserView.java

```
* @author Diaza
*/
public class UserView extends JFrame {
    private JTextField txtName = new JTextField(20);
    private JTextField txtEmail = new JTextField(20);
    private JButton btnAdd = new JButton("Add User");
    private JButton btnRefresh = new JButton("Refresh");
    private JList<String> userList = new JList<>();
    private DefaultListModel<String> listModel = new
DefaultListModel<>();

    public UserView() {
        setTitle("User Management");
        setSize (400, 300);
        setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);

        JPanel panel = new JPanel (new GridLayout (5, 1));
        panel.add(new JLabel ("Name:"));
        panel.add(txtName);
        panel.add(new JLabel ("Email:"));
        panel.add(txtEmail);

        JPanel buttonPanel = new JPanel();
        buttonPanel.add(btnAdd);
        buttonPanel.add(btnRefresh);
        panel.add(buttonPanel);

        userList.setModel(listModel);
        add(panel, BorderLayout.NORTH);
        add(new JScrollPane (userList), BorderLayout.CENTER);
    }

    public String getNameInput () {
        return txtName.getText();
    }

    public String getEmailInput() {
        return txtEmail.getText();
    }
}
```



UserView.java

```
public void setUserList (String[] users) {  
    listModel.clear();  
    for (String user : users) {  
        listModel.addElement(user);  
    }  
}  
  
public void addAddUserListener (ActionListener listener) {  
    btnAdd.addActionListener(listener);  
}  
  
public void addRefreshListener (ActionListener listener) {  
    btnRefresh.addActionListener(listener);  
}  
}
```

Penjelasan

Code ini mendefinisikan sebuah kelas Java bernama UserView, yang berada dalam paket view. Kelas ini merupakan bagian dari lapisan view dalam pola arsitektur MVC . Bertujuan untuk menyediakan antarmuka pengguna berbasis Swing untuk mengelola data pengguna

UserController.java

```
package controller;  
  
import model.*;  
import view.UserView;  
  
import javax.swing.*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.util.List;  
import org.apache.ibatis.session.SqlSession;  
import org.apache.ibatis.session.SqlSessionFactory;  
  
/**  
 *  
 * @author Diaza
```



UserController.java

```
*/  
public class UserController {  
    private UserView view;  
    private UserMapper mapper;  
  
    public UserController(UserView view, UserMapper mapper) {  
        this.view = view;  
        this.mapper = mapper;  
  
        this.view.addAddUserListener(new AddUserListener());  
        this.view.addRefreshListener(new RefreshListener());  
    }  
  
    class AddUserListener implements ActionListener {  
        @Override  
        public void actionPerformed(ActionEvent e) {  
            String name = view.getNameInput();  
            String email = view.getEmailInput();  
            if (!name.isEmpty() && !email.isEmpty()) {  
                User user = new User();  
                user.setName(name);  
                user.setEmail(email);  
                mapper.insertUser(user);  
                JOptionPane.showMessageDialog(view, "User added  
successfully!");  
            } else {  
                JOptionPane.showMessageDialog(view, "please fill in  
all fields.");  
            }  
        }  
    }  
  
    class RefreshListener implements ActionListener {  
        @Override  
        public void actionPerformed(ActionEvent e) {  
            List<User> users = mapper.getAllUsers();  
            String[] userArray = users.stream()  
                .map(u -> u.getName() + " (" + u.getEmail() +  
")")  
                .toArray(String[]::new);  
        }  
    }  
}
```



UserController.java

```
        view.setUserList(userArray);  
    }  
}  
}
```

Penjelasan

Code ini mendefinisikan kelas UserController, yang berada didalam paket controller. Kelas ini adalah bagian menghubungkan lapisan View dengan lapisan Model (logika dan data) , sehingga memungkinkan aplikasi merespon tindakan pengguna dan memperbarui data serta tampilan secara sinkron.

Penjelasan Kode

Jika **berkas memiliki output**, gunakan tabel ini.

Main.java

```
package main;  
  
import model.MyBatisUtil;  
import model.UserMapper;  
import org.apache.ibatis.session.SqlSession;  
import view.UserView;  
import controller.UserController;  
  
/**  
 *  
 * @author Diaza  
 */  
public class Main {  
    public static void main(String[] args) {  
        SqlSession session = MyBatisUtil.getSqlSession();  
        UserMapper mapper = session.getMapper(UserMapper.class);  
  
        UserView view = new UserView();  
        new UserController(view, mapper);  
  
        view.setVisible(true);  
    }  
}
```




Penjelasan

Code ini mendefinisikan Kelas Main, yang merupakan entry point dari aplikasi . Fungsinya untuk menginisialisasi dan menghubungkan seluruh komponen dalam arsitektur MVC sehingga aplikasi dalam berjalan. Berkas ini memanfaatkan MyBatis sebagai ORM untuk akses database , serta Swing untuk antarmuka pengguna.

Output

User Management

Name:

diaz

Email:

diaz@gmail.com

Add User Refresh

diaz (diaz@gmail.com)