Tugas
Praktikum Pemrograman II

UNPAS
Teknik Informatika
Universitas Pasundan

[PERTEMUAN KE 13]

Identitas

- NPM : 223040033
- Nama : Muhammad Fauzan Dwi Putera
- Kelas : A
- URL Repository Github :

[https://github.com/Fauzan2617/pp2-2024_223040033_A/tree/main/Semua%20Latihan/latihan%20Pertemuan%20-%2013/konkurensi]

Penjelasan Kode

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

[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;

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.");
            }
        }
    }

    public void generateDummyData() {
        for (int i = 1; i <= 100; i++) {
            User user = new User();
            user.setName("User " + i);
            user.setEmail("user" + i + "@example.com");
            mapper.insertUser(user);
        }
    }




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

```
package controller;

import model.*;
import view.UserView;

import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.List;

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.");
            }
        }
    }

    public void generateDummyData() {
        for (int i = 1; i <= 100; i++) {
            User user = new User();
            user.setName("User " + i);
            user.setEmail("user" + i + "@example.com");
            mapper.insertUser(user);
        }
    }




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

Penjelasan

[Code bagian Controller]

[MyBatisUtil.java]

```
[package model;

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

public class MyBatisUtil {
    private static SqlSessionFactory sqlSessionFactory;

    static {
try {
            sqlSessionFactory = new SqlSessionFactoryBuilder()
.build(Resources.getResourceAsStream("mybatisconfig.xml"));
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    public static SqlSession getSqlSession() {
return sqlSessionFactory.openSession();     }
```

```
}
]
```

Penjelasan

[Code bagian model]

[User.java]

```
[package model;
 public class User {
private int id;
private String name;
private String email;

    // Getters dan Setters
    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

[Code bagian model]

[UserMapper.java]

```
[package model;

import java.util.List;
import org.apache.ibatis.annotations.*;
 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 bagian model]

[UserPdf.java]

```
[package view;

import model.User;

import com.lowagie.text.Document; import
com.lowagie.text.DocumentException; import
com.lowagie.text.Font; import
com.lowagie.text.FontFactory; import
com.lowagie.text.PageSize; import
com.lowagie.text.Phrase; import
com.lowagie.text.pdf.PdfPCell; import
com.lowagie.text.pdf.PdfPTable; import
com.lowagie.text.pdf.PdfWriter;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.List;

public class UserPdf {
    public void exportPdf(List<User> users) {
        System.out.println(users.size());
        Document document = new Document(PageSize.A4);

        try {
            PdfWriter writer = PdfWriter.getInstance(document,
new FileOutputStream(System.getProperty("user.dir") +
File.separator + "users.pdf")
            );
             float width = document.getPageSize().getWidth();
float height = document.getPageSize().getHeight();

            document.open();

            float[] columnDefinitionSize = {33.33F, 33.33F, 33.33F};

            float pos = height / 2;
PdfPTable table = null;
            PdfPCell cell = null;

            table = new PdfPTable(columnDefinitionSize);
table.getDefaultCell().setBorder(1);
table.setHorizontalAlignment(0);
table.setTotalWidth(width - 72);
table.setLockedWidth(true);

            table.addCell(new Phrase("No"));
table.addCell(new Phrase("Name"));
table.addCell(new Phrase("Email"));

            int no = 1;
            for (User user : users) {
                table.addCell(new
Phrase(String.valueOf(no++)));
table.addCell(new Phrase(user.getName()));
table.addCell(new Phrase(user.getEmail()));             }
```

```
            document.add(table);
            } catch (DocumentException | IOException ex) {
                System.err.println(ex.getMessage());
            }

            // step 5
            document.close();

        }
    }


]
```

Penjelasan

[Code bagian view]

[UserView.java]

```
[package view;
 import
javax.swing.*;
import java.awt.*;
import java.awt.event.ActionListener;

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 JButton btnExport = new JButton("Export");
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);
buttonPanel.add(btnExport);
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();
    }
    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);
    }
    public void addExportListener(ActionListener listener) {
btnExport.addActionListener(listener);
    }
}


]
```

Penjelasan
[Code bagian view]

Penjelasan Kode

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

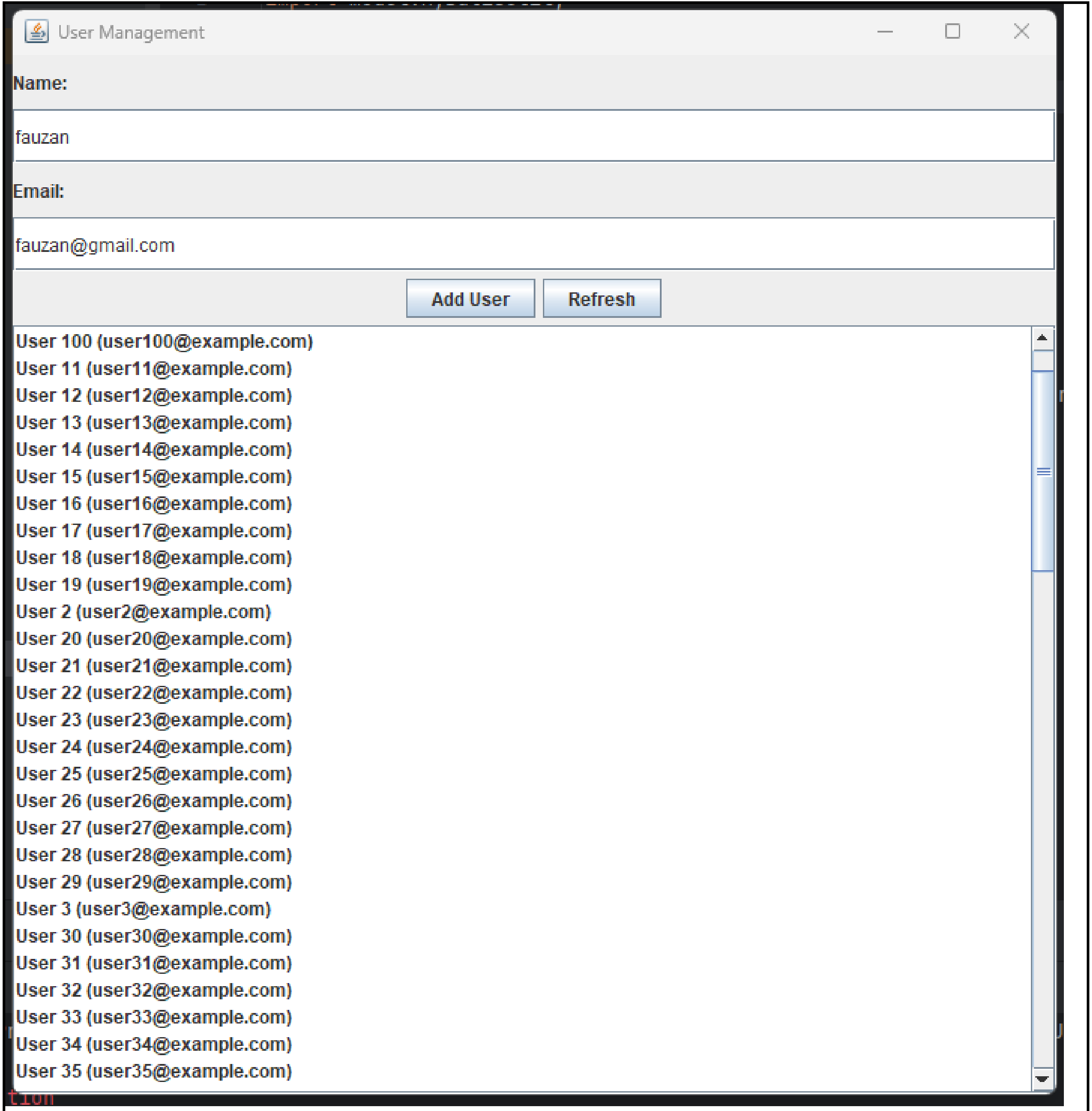
[Main.java]

```
import model.MyBatisUtil;
import model.UserMapper;
import
org.apache.ibatis.session.SqlSession;
import view.UserView;
import controller.UserController;

public class Main {
    public static void main(String[] args) {
        SqlSession session =
MyBatisUtil.getSqlSession();
        UserMapper mapper =
session.getMapper(UserMapper.class);

        UserView view = new UserView();
        UserController controller = new
UserController(view, mapper);

        // Generate dummy data
        controller.generateDummyData();

        view.setVisible(true);
    }
}
```

Penjelasan

[Code bagian main program]

Output



Name:

fauzan

Email:

fauzan@gmail.com

```
User 100 (user100@example.com)
User 11 (user11@example.com)
User 12 (user12@example.com)
User 13 (user13@example.com)
User 14 (user14@example.com)
User 15 (user15@example.com)
User 16 (user16@example.com)
User 17 (user17@example.com)
User 18 (user18@example.com)
User 19 (user19@example.com)
User 2 (user2@example.com)
User 20 (user20@example.com)
User 21 (user21@example.com)
User 22 (user22@example.com)
User 23 (user23@example.com)
User 24 (user24@example.com)
User 25 (user25@example.com)
User 26 (user26@example.com)
User 27 (user27@example.com)
User 28 (user28@example.com)
User 29 (user29@example.com)
User 3 (user3@example.com)
User 30 (user30@example.com)
User 31 (user31@example.com)
User 32 (user32@example.com)
User 33 (user33@example.com)
User 34 (user34@example.com)
User 35 (user35@example.com)
```

Catatan:

- Silakan modifikasi dan tambahkan informasi sesuai dengan proyek yang Anda kerjakan baik pada tanda [kalimat].
- Sesuaikan nama file laporan dengan format **Tugas[Pertemuan ke-X]_PP1_NPM** Contoh: Tugas[P10]_PP1_203040104
- Sesuaikan judul laporan dengan format **Laporan Tugas Pertemuan ke-X** Contoh: Laporan Tugas Pertemuan ke-10