Nama   : Muh. Nur Afrizal

NIM   : 222212738

Kelas / No : 2KS1 / 25

TUGAS PERTEMUAN 9

PEMROGRAMAN BERORIENTASI OBJEK (PBO)

Seluruh code dapat dilihat pada: https://github.com/afrzl/Pemrograman-Berorientasi-Objek/tree/main/Week%209/Pra/Pertemuan%209

1. **Student.java (Model)**

```java
package model;

public class Student {
    private String nim;
    private String firstName;
    private String lastName;
    private int umur;
    private String asal;

    public String getNim() {
        return nim;
    }

    public String getFirstName() {
        return firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public int getUmur() {
        return umur;
    }

    public String getAsal() {
        return asal;
    }

    public void setNim(String nim) {
        this.nim = nim;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public void setUmur(int umur) {
        this.umur = umur;
    }

    public void setAsal(String asal) {
        this.asal = asal;
```

```
        }
}
```

2. **StudentController.java (Controller)**

```java
package controller;

import com.sun.source.tree.ThrowTree;
import model.Student;
import view.StudentView;
import view.StudentForm;
import java.util.*;

public class StudentController {
    private ArrayList<Student> model = new ArrayList<>();
    private StudentView view;

    public StudentController() {}

    public StudentController(ArrayList <Student> model, StudentView
view) {
        this.model = model;
        this.view = view;
    }

    public void addStudent(Student student) {
        model.add(student);
    }

    public void removeStudent(Student student) {
        model.remove(student);
    }

    public void setStudentNim(Student student, String nim) {
        for(Student mdl : this.model) {
            if(mdl == student) {
                mdl.setNim(nim);
            }
        }
    }

    public String getStudentNim(Student student) {
        for (Student std : this.model) {
            if(std == student) {
                return std.getNim();
            }
        }
        return "No data.";
    }

    public void setStudentFirstName(Student student, String firstName)
{
        for(Student mdl : this.model) {
            if(mdl == student) {
                mdl.setFirstName(firstName);
            }
        }
    }
```

```java
public String getStudentFirstName(Student student) {
    for (Student std : this.model) {
        if(std == student) {
            return std.getFirstName();
        }
    }
    return "No data.";
}

public void setStudentLastName(Student student, String lastName) {
    for(Student mdl : this.model) {
        if(mdl == student) {
            mdl.setLastName(lastName);
        }
    }
}

public String getStudentLastName(Student student) {
    for (Student std : this.model) {
        if(std == student) {
            return std.getLastName();
        }
    }
    return "No data.";
}

public void setStudentUmur(Student student, int umur) {
    for(Student mdl : this.model) {
        if(mdl == student) {
            mdl.setUmur(umur);
        }
    }
}

public int getStudentUmur(Student student) {
    for (Student std : this.model) {
        if(std == student) {
            return std.getUmur();
        }
    }
    return 0;
}

public void setStudentAsal(Student student, String asal) {
    for(Student mdl : this.model) {
        if(mdl == student) {
            mdl.setAsal(asal);
        }
    }
}

public String getStudentAsal(Student student) {
    for (Student std : this.model) {
        if(std == student) {
            return std.getAsal();
        }
    }
    return "No data.";
```

```java
    }

    public void updateView(Student student) {
        view.print(student);
    }

    public String updateAllView() {
        view = new StudentView();
        String res = "Daftar NIM dan Nama Mahasiswa yang telah
berhasil disimpan : \n";
        for(Student std : this.model) {
            res += view.print(std) + '\n';
        }

        return res;
    }

    public void validateForm(String nim, String firstName, String
lastName, String umur, String asal) {
        String err = "";
        if (nim.isBlank() || firstName.isBlank() || lastName.isBlank()
|| asal.isBlank()) {
            err += "Terjadi kesalahan input pada: form isian wajib
diisi semua.\n";
        } else {
            if (nim.length() != 6) {
                err += "Terjadi kesalahan input pada: NIM harus
memiliki panjang 6 digit.\n";
            }
            String name = firstName + lastName;
            if (name.length() > 50) {
                err += "Terjadi kesalahan input pada: Nama depan +
nama belakang tidak boleh lebih dari 50 karakter.\n";
            }
            if (!umur.matches("\\d+")) {
                err += "Terjadi kesalahan input pada: Umur hanya dapat
diisi dengan angka.\n";
            }
        }

        if (err != "") {
            throw new IllegalArgumentException(err);
        }
    }
}
```

3. StudentForm.java (View)

```java
package view;

import controller.StudentController;
import model.Student;

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

```java
public class StudentForm {
    private JPanel panelMain;
    private JTextField nim;
    private JTextField lastName;
    private JTextField firstName;
    private JSpinner umur;
    private JComboBox asal;
    private JButton reset;
    private JButton simpan;
    private static StudentController studentController;
    private StudentView studentView;

    public StudentForm() {
        simpan.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                Student student = new Student();
                String nimField = nim.getText();
                String firstNameField = firstName.getText();
                String lastNameField = lastName.getText();
                String umurField = umur.getValue().toString();
                String asalField =
String.valueOf(asal.getSelectedItem());

                try {
                    studentController.validateForm(nimField,
firstNameField, lastNameField, umurField, asalField);

                    student.setNim(nimField);
                    student.setFirstName(firstNameField);
                    student.setLastName(lastNameField);
                    student.setUmur(Integer.parseInt(umurField));
                    student.setAsal(asalField);

                    studentController.addStudent(student);
                    JOptionPane.showMessageDialog(panelMain,
studentController.updateAllView());
                    reset.doClick();

                } catch (IllegalArgumentException err) {
                    JOptionPane.showMessageDialog(panelMain,
err.getMessage());
                }
            }
        });
        reset.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                nim.setText("");
                firstName.setText("");
                lastName.setText("");
                umur.setValue(0);
                asal.setSelectedIndex(0);
            }
        });
    }

    public static void main(String[] args) {
        JFrame frame = new JFrame("StudentForm");
```

```java
        frame.setContentPane(new StudentForm().panelMain);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.pack();
        frame.setVisible(true);

        studentController = new StudentController();
    }
}
```

4. StudentView.java (View)

```java
package view;

import model.Student;

public class StudentView {
    public StudentView() {}

    public String print(Student student) {
        return student.getNim() + "-" + student.getFirstName() + " " +
student.getLastName();
    }
}
```

5. Output