

ACTIVITY PERTEMUAN 6

NAMA : Muhammmad Reza Rahman
NPM : 51421051
Kelas : 4IA14
Materi : Implementasi AOP dan DI pada Project Spring dan Hibernate

Pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
    http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>me.reza</groupId>
  <artifactId>Pert6_51421051</artifactId>
  <version>1.0</version>
  <packaging>jar</packaging>
  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <maven.compiler.source>18</maven.compiler.source>
    <maven.compiler.target>18</maven.compiler.target>
    <exec.mainClass>me.reza.Pert6_51421051.Pert6_51421051</exec.mainClass>
  </properties>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.3.3</version>
    <relativePath/>
  </parent>

  <dependencies>
    <!-- Hibernate + Spring Data JPA -->
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>

    <!-- MySQL Connector -->
    <dependency>
      <groupId>mysql</groupId>
```

```

        <artifactId>mysql-connector-java</artifactId>
        <version>8.0.33</version>
    </dependency>

    <!-- Spring Boot Web dependency (for MVC if needed) -->
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>

    <!-- Testing dependencies -->
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>
</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>
</project>

```

Pert6_51421051

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt
to change this license
 */

package me.reza;

import me.reza.controller.MahasiswaController;
import me.reza.service.MahasiswaService;
import me.reza.view.MahasiswaView2;
import org.springframework.boot.ApplicationArguments;
import org.springframework.boot.ApplicationRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.ApplicationContext;

/**

```

```

/*
 * @author WINDOWS 10
 */

@SpringBootApplication
public class Pert6_51421051 implements ApplicationRunner {

    @Autowired
    private MahasiswaService mahasiswaService;

    public static void main(String[] args) {
        System.setProperty("java.awt.headless", "false"); // Disable headless
mode

        // Start the Spring application and get the application context
        ApplicationContext context =
SpringApplication.run(Pert6_51421051.class, args);

        // Instantiate the view and inject the controller manually
        MahasiswaController controller =
context.getBean(MahasiswaController.class);
        MahasiswaView2 mahasiswaView = new MahasiswaView2(controller);
        mahasiswaView.setVisible(true);
    }

    @Override
    public void run(ApplicationArguments args) throws Exception {
    }
}

```

MahasiswaController.java

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt
to change this license
 */

package me.reza;

import me.reza.controller.MahasiswaController;
import me.reza.service.MahasiswaService;
import me.reza.view.MahasiswaView2;
import org.springframework.boot.ApplicationArguments;
import org.springframework.boot.ApplicationRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.beans.factory.annotation.Autowired;

```

```

import org.springframework.context.ApplicationContext;

/**
 *
 * @author WINDOWS 10
 */

@SpringBootApplication
public class Pert6_51421051 implements ApplicationRunner {

    @Autowired
    private MahasiswaService mahasiswaService;

    public static void main(String[] args) {
        System.setProperty("java.awt.headless", "false"); // Disable headless
mode

        // Start the Spring application and get the application context
        ApplicationContext context =
SpringApplication.run(Pert6_51421051.class, args);

        // Instantiate the view and inject the controller manually
        MahasiswaController controller =
context.getBean(MahasiswaController.class);
        MahasiswaView2 mahasiswaView = new MahasiswaView2(controller);
        mahasiswaView.setVisible(true);
    }

    @Override
    public void run(ApplicationArguments args) throws Exception {
    }
}

```

ModelMahasiswa

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt
to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit
this template
 */
package me.reza.model;

import jakarta.persistence.*;

@Entity
@Table(name = "mahasiswa")

```

```

public class ModelMahasiswa {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id")
    private int id;

    @Column(name = "npm", nullable = false, length = 8)
    private String npm;

    @Column(name = "nama", nullable = false, length = 50)
    private String nama;

    @Column(name = "semester")
    private int semester;

    @Column(name = "ipk")
    private float ipk;

    //Buka code -> generate -> constructor -> all
    //code -> generate -> setter and getter -> All
    //Baru buat anotasi @ nya

    // constructor kosong untuk ORM
    public ModelMahasiswa() {

    }

    //constructor ada isinya untuk dipanggil di aplikasi
    public ModelMahasiswa(int id, String npm, String nama, int semester, float
ipk) {
        this.id = id;
        this.npm = npm;
        this.nama = nama;
        this.semester = semester;
        this.ipk = ipk;
    }

    public int getId() {
        return id;
    }

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

    public String getNpm() {

```

```

        return npm;
    }

    public void setNpm(String npm) {
        this.npm = npm;
    }

    public String getNama() {
        return nama;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }

    public int getSemester() {
        return semester;
    }

    public void setSemester(int semester) {
        this.semester = semester;
    }

    public float getIpk() {
        return ipk;
    }

    public void setIpk(float ipk) {
        this.ipk = ipk;
    }
}

```

ModelTabelMahasiswa

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt
to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit
this template
 */
package me.reza.model;
import javax.swing.table.AbstractTableModel;
import java.util.List;

public class ModelTabelMahasiswa extends AbstractTableModel{
    private List<ModelMahasiswa> mahasiswaList;

```

```

private String[] columnNames = {"ID", "NPM", "Nama", "Semester", "IPK"};

public ModelTabelMahasiswa(List<ModelMahasiswa> mahasiswaList) {
    this.mahasiswaList = mahasiswaList;
}

@Override
public int getRowCount() {
    return mahasiswaList.size(); // Jumlah baris sesuai dengan jumlah data
mahasiswa
}

@Override
public int getColumnCount() {
    return columnNames.length; // Jumlah kolom sesuai dengan jumlah elemen
dalam columnNames
}

@Override
public Object getValueAt(int rowIndex, int columnIndex) {
    ModelMahasiswa mahasiswa = mahasiswaList.get(rowIndex);
    switch (columnIndex) {
        case 0:
            return mahasiswa.getId();
        case 1:
            return mahasiswa.getNpm();
        case 2:
            return mahasiswa.getNama();
        case 3:
            return mahasiswa.getSemester();
        case 4:
            return mahasiswa.getIpk();
        default:
            return null;
    }
}

@Override
public String getColumnName(int column) {
    return columnNames[column]; // Mengatur nama kolom
}

@Override
public boolean isCellEditable(int rowIndex, int columnIndex) {
    return false; // Semua sel tidak dapat diedit
}

// Method untuk menambahkan atau memodifikasi data, jika dibutuhkan

```

```

    public void setMahasiswaList(List<ModelMahasiswa> mahasiswaList) {
        this.mahasiswaList = mahasiswaList;
        fireTableDataChanged(); // Memberitahu JTable bahwa data telah berubah
    }
}

```

MahasiswaRepository

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt
to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit
this template
 */
package me.reza.repository;

import me.reza.model.ModelMahasiswa;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;
/**
 *
 * @author WINDOWS 10
 */

@Repository
public interface MahasiswaRepository extends JpaRepository<ModelMahasiswa,
Integer> {

}

```

MahasiswaRepository

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt
to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit
this template
 */
package me.reza.service;

import me.reza.model.ModelMahasiswa;
import me.reza.repository.MahasiswaRepository;
import java.util.List;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

```



```
/**
 *
 * @author WINDOWS 10
 */
@Service
public class MahasiswaService {

    @Autowired
    private MahasiswaRepository repository;

    public void addMhs(ModelMahasiswa mhs) {
        repository.save(mhs);
    }

    public ModelMahasiswa getMhs(int id) {
        return repository.findById(id).orElse(null);
    }

    public void updateMhs(ModelMahasiswa mhs) {
        repository.save(mhs);
    }

    public void deleteMhs(int id) {
        repository.deleteById(id);
    }

    public List<ModelMahasiswa> getAllMahasiswa() {
        return repository.findAll();
    }
}
```

MahasiswaView

The image shows a Java Swing window titled "MahasiswaView". It features a form with five input fields labeled "Nama", "NPM", "Semester", and "IPK". Below the fields are three buttons: "Simpan", "Refresh", and "Buang". At the bottom of the window is a table with four columns labeled "Title 1", "Title 2", "Title 3", and "Title 4". The table is currently empty.

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt
to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit
this template
 */
package me.reza.view;

import me.reza.controller.MahasiswaController;
import me.reza.model.ModelMahasiswa;
import java.util.List;
import javax.swing.*;
import me.reza.model.ModelTabelMahasiswa;
/**
 *
 * @author WINDOWS 10
 */
public class MahasiswaView extends javax.swing.JFrame {

    private MahasiswaController controller;

```

```

public MahasiswaView(MahasiswaController controller) {
    this.controller = controller;
    initComponents();
    loadMahasiswaTable();
}

private MahasiswaView() {
    throw new UnsupportedOperationException("Not supported yet."); //
Generated from
nbfs://nbhost/SystemFileSystem/Templates/Classes/Code/GeneratedMethodBody
}

public void loadMahasiswaTable() {
    // Ambil data dari controller
    List<ModelMahasiswa> listMahasiswa = controller.getAllMahasiswa();

    // Buat model tabel kustom dengan data mahasiswa
    ModelTabelMahasiswa tableModel = new ModelTabelMahasiswa(listMahasiswa);

    // Set model pada JTable
    dataTable.setModel(tableModel);
}

/**
 * This method is called from within the constructor to initialize the
form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated
Code">
private void initComponents() {

    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    namaField = new javax.swing.JTextField();
    npmField = new javax.swing.JTextField();
    semesterField = new javax.swing.JTextField();
    ipkField = new javax.swing.JTextField();
    simpanButton = new javax.swing.JButton();
    jScrollPane1 = new javax.swing.JScrollPane();
    jScrollPane2 = new javax.swing.JScrollPane();
    dataTable = new javax.swing.JTable();
    buangButton = new javax.swing.JButton();
    refreshButton = new javax.swing.JButton();

```

```
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jLabel1.setText("Nama");

jLabel2.setText("NPM");

jLabel3.setText("Semester");

jLabel4.setText("IPK");

namaField.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        namaFieldActionPerformed(evt);
    }
});

ipkField.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        ipkFieldActionPerformed(evt);
    }
});

simpanButton.setText("Simpan");
simpanButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        simpanButtonActionPerformed(evt);
    }
});

dataTable.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
));
jScrollPane2.setViewportViewView(dataTable);

jScrollPane1.setViewportViewView(jScrollPane2);

uangButton.setText("Uang");
uangButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        uangButtonActionPerformed(evt);
    }
});
```

```

    }
});

refreshButton.setText("Refresh");
refreshButton.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        refreshButtonActionPerformed(evt);
    }
});

javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGroup(layout.createSequentialGroup()
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addGroup(layout.createSequentialGroup()
                                .addGap(48, 48, 48)
                                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                    .addComponent(jLabel1)
                                    .addComponent(jLabel2)
                                    .addComponent(jLabel3)
                                    .addComponent(jLabel4))
                                .addGap(31, 31, 31)
                                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                                    .addComponent(namaField)
                                    .addComponent(npmField)
                                    .addComponent(semesterField)
                                    .addComponent(ipkField,
                                        javax.swing.GroupLayout.DEFAULT_SIZE, 161, Short.MAX_VALUE)))
                                .addGroup(layout.createSequentialGroup()
                                    .addGap(36, 36, 36)
                                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                        .addGroup(layout.createSequentialGroup()
                                            .addComponent(simpanButton)
                                            .addGap(29, 29, 29)
                                            .addComponent(refreshButton)
                                            .addGap(30, 30, 30)
                                            .addComponent(buangButton))
                                        .addComponent(jScrollPane1,
                                            javax.swing.GroupLayout.PREFERRED_SIZE, 452,
                                            javax.swing.GroupLayout.PREFERRED_SIZE))))))
                            .addGroup(layout.createSequentialGroup()
                                .addComponent(namaField)
                                .addComponent(npmField)
                                .addComponent(semesterField)
                                .addComponent(ipkField,
                                    javax.swing.GroupLayout.DEFAULT_SIZE, 161, Short.MAX_VALUE)))
                                .addGroup(layout.createSequentialGroup()
                                    .addGap(36, 36, 36)
                                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                        .addGroup(layout.createSequentialGroup()
                                            .addComponent(simpanButton)
                                            .addGap(29, 29, 29)
                                            .addComponent(refreshButton)
                                            .addGap(30, 30, 30)
                                            .addComponent(buangButton))
                                        .addComponent(jScrollPane1,
                                            javax.swing.GroupLayout.PREFERRED_SIZE, 452,
                                            javax.swing.GroupLayout.PREFERRED_SIZE))))))
            .addGap(10, 10, 10)
        )
    )
);

```

```

        .addContainerGap(43, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(28, 28, 28)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabel1)
                .addComponent(namaField,
                    javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
                    javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(18, 18, 18)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel2)
                .addComponent(npmField,
                    javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
                    javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(18, 18, 18)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel3)
                .addComponent(semesterField,
                    javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
                    javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(18, 18, 18)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jLabel4)
                .addComponent(ipkField,
                    javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
                    javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGap(18, 18, 18)
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(buangButton)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(simpanButton)
                    .addComponent(refreshButton)))
            .addGap(31, 31, 31)
            .addComponent(jScrollPane1,
                javax.swing.GroupLayout.PREFERRED_SIZE, 201,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(36, Short.MAX_VALUE))
    );

```

```

        pack();
    }// </editor-fold>

    private void namaFieldActionPerformed(java.awt.event.ActionEvent evt)
    {
        // TODO add your handling code here:
    }

    private void ipkFieldActionPerformed(java.awt.event.ActionEvent evt)
    {
        // TODO add your handling code here:
    }

    private void simpanButtonActionPerformed(java.awt.event.ActionEvent evt)
    {
        // TODO add your handling code here:
        String npm = getNpmField().getText();
        String nama = getNamaField().getText();
        int semester = Integer.parseInt(getSemesterField().getText());
        float ipk = Float.parseFloat(getIpkField().getText());
        ModelMahasiswa mahasiswa = new ModelMahasiswa(0, npm, nama, semester,
ipk);
        System.out.println(mahasiswa.getIpk());
        System.out.println(mahasiswa.getNama());
        System.out.println(mahasiswa.getSemester());
        System.out.println(mahasiswa.getNpm());

        controller.addMahasiswa(mahasiswa);
        loadMahasiswaTable();
    }

    private void buangButtonActionPerformed(java.awt.event.ActionEvent evt)
    {
        // TODO add your handling code here:
        JTextField idField = new JTextField(10);

        // Membuat panel untuk menampung JTextField
        JPanel panel = new JPanel();
        panel.add(new JLabel("Masukkan ID yang ingin dihapus:"));
        panel.add(idField);

        // Menampilkan dialog box dengan JTextField, tombol OK, dan Cancel
        int result = JOptionPane.showConfirmDialog(null, panel,
            "Hapus Mahasiswa", JOptionPane.OK_CANCEL_OPTION,
JOptionPane.PLAIN_MESSAGE);

        // Jika tombol OK ditekan

```

```

        if (result == JOptionPane.OK_OPTION) {
            try {
                // Mengambil input ID dan memanggil metode deleteMhs
                int id = Integer.parseInt(idField.getText());
                controller.deleteMahasiswa(id);
                JOptionPane.showMessageDialog(null, "Data berhasil dihapus.",
                "Sukses", JOptionPane.INFORMATION_MESSAGE);
                loadMahasiswaTable();
            } catch (NumberFormatException e) {
                // Menangani error jika ID yang dimasukkan bukan angka
                JOptionPane.showMessageDialog(null, "ID harus berupa angka.",
                "Error", JOptionPane.ERROR_MESSAGE);
            }
        }
    }

    private void refreshButtonActionPerformed(java.awt.event.ActionEvent evt)
    {
        // TODO add your handling code here:
        loadMahasiswaTable();
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting
        code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay with the
        default look and feel.
         * For details see
        http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
            javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(MahasiswaView.class.getName()).
            log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(MahasiswaView.class.getName()).
            log(java.util.logging.Level.SEVERE, null, ex);
        }
    }

```



```

        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(MahasiswaView.class.getName()).
log(java.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(MahasiswaView.class.getName()).
log(java.util.logging.Level.SEVERE, null, ex);
        }
    //</editor-fold>
    //</editor-fold>

    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(MahasiswaView.class.getName()).
log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(MahasiswaView.class.getName()).
log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(MahasiswaView.class.getName()).
log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(MahasiswaView.class.getName()).
log(java.util.logging.Level.SEVERE, null, ex);
    }
}

    java.awt.EventQueue.invokeLater(new Runnable() {
        @Override
        public void run() {
            new MahasiswaView().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton buangButton;
private javax.swing.JTable dataTable;
private javax.swing.JTextField ipkField;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;

```

```

private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JTextField namaField;
private javax.swing.JTextField npmField;
private javax.swing.JButton refreshButton;
private javax.swing.JTextField semesterField;
private javax.swing.JButton simpanButton;
// End of variables declaration
public JTextField getIpkField() {
    return ipkField;
}

public JTextField getNamaField() {
    return namaField;
}

public JTextField getNpmField() {
    return npmField;
}

public JTextField getSemesterField() {
    return semesterField;
}
}

```

application.properties


```

# Konfigurasi MySQL Hibernate
spring.datasource.url=jdbc:mysql://localhost:3306/spring_51421051?useSSL=false
&serverTimezone=UTC
spring.datasource.username=root
spring.datasource.password=
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

# Hibernate settings
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true

```

Output :



—

□

×

Nama

Juju

NPM

55555555

Semester

5

IPK

5.55

Simpan

Refresh

Buang

| ID | NPM | Nama | Semester | IPK |
|----|----------|------|----------|------|
| 1 | 51421051 | reza | 7 | 3.2 |
| 4 | 55555555 | Juju | 5 | 5.55 |