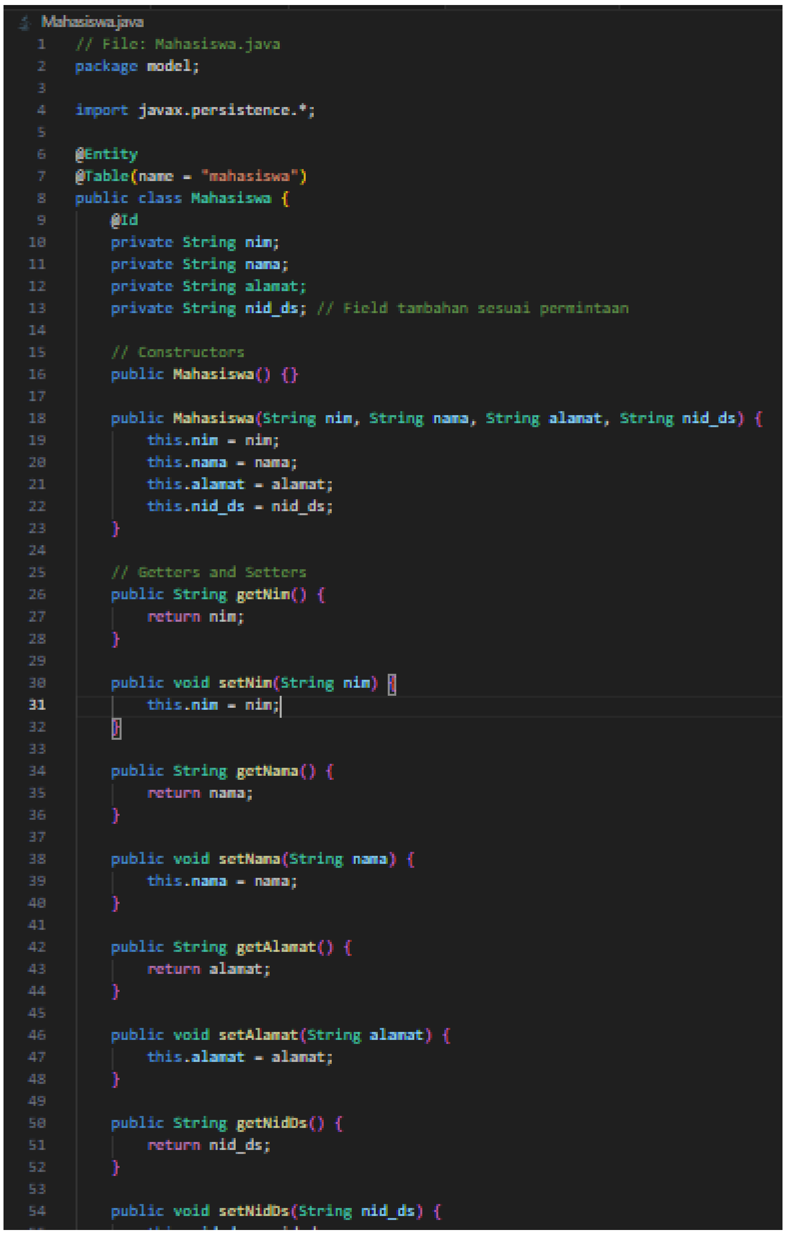
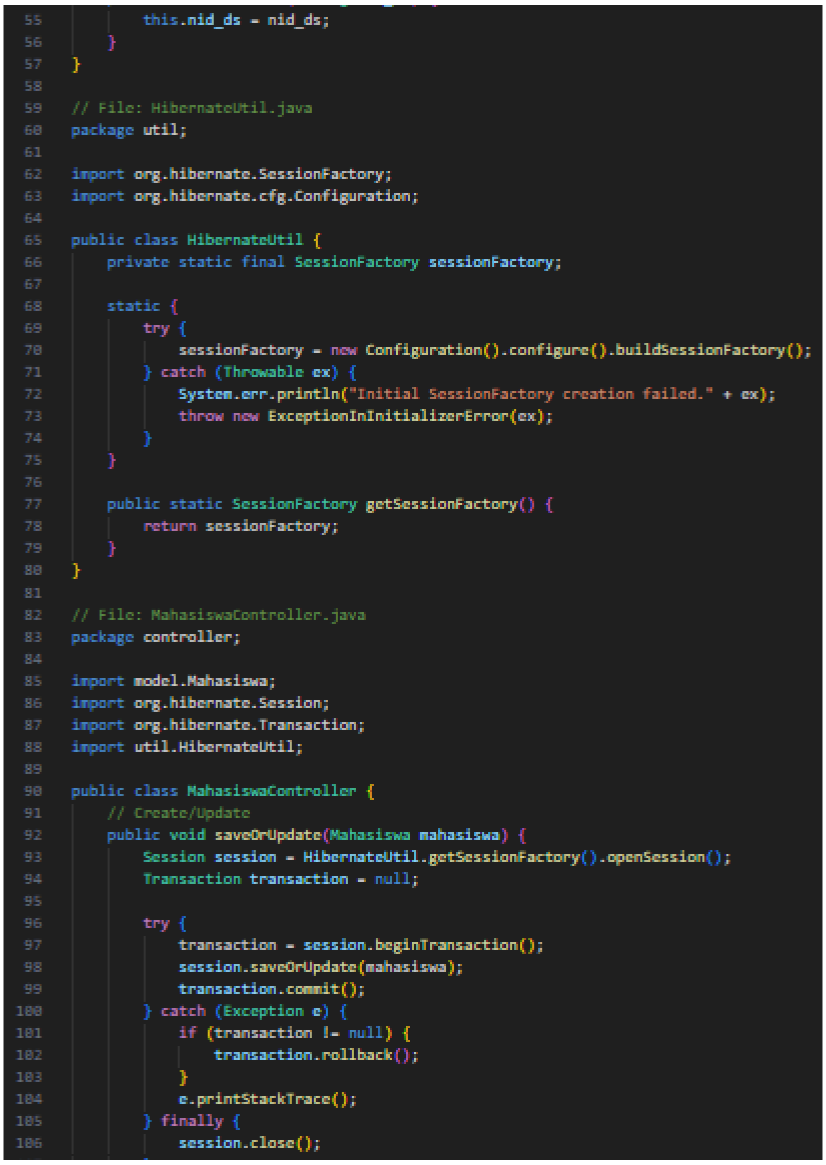
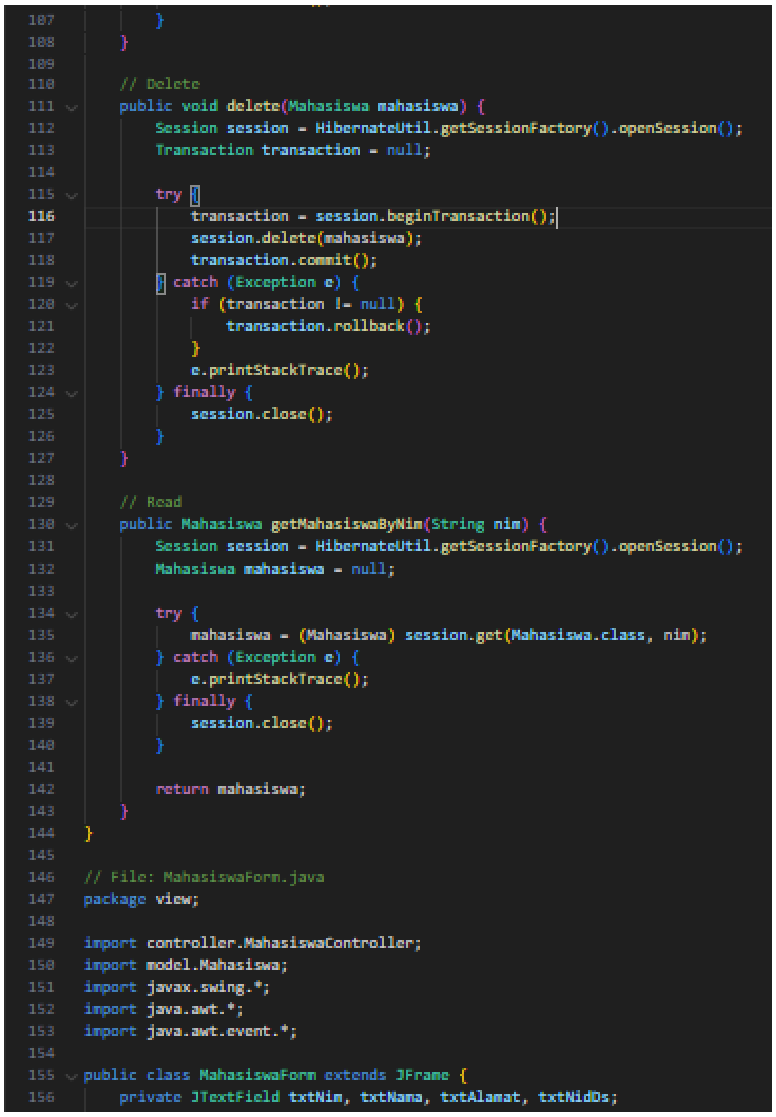
Listing



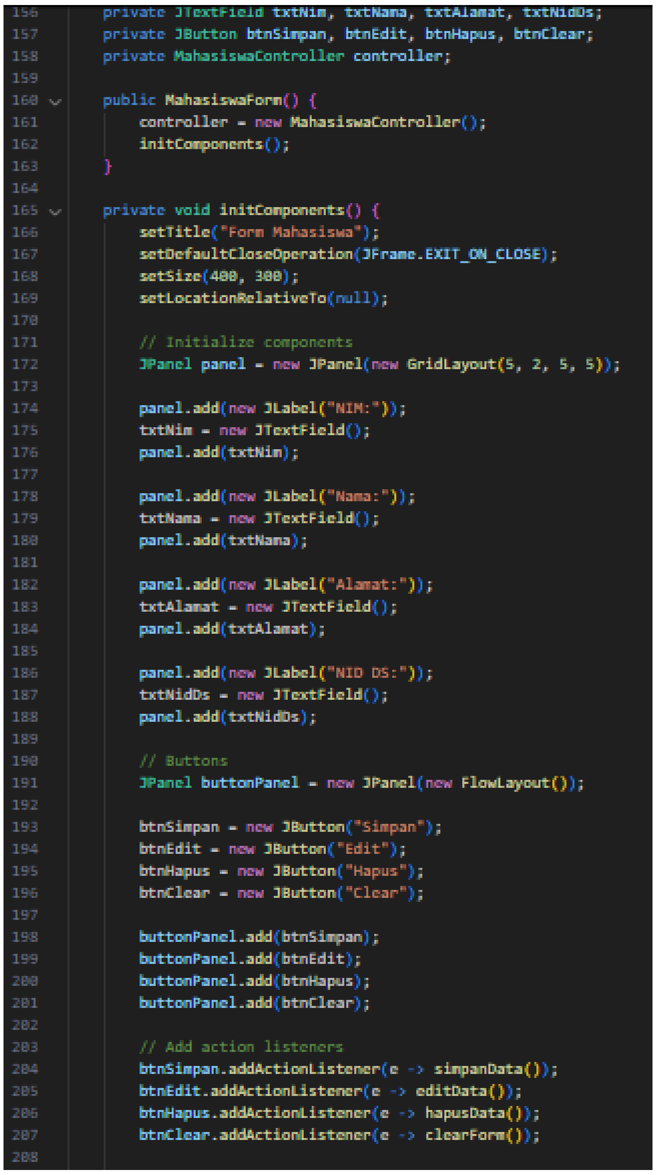
```java
// File: Mahasiswa.java
package model;

import javax.persistence.*;

@Entity
@Table(name = "mahasiswa")
public class Mahasiswa {
    @Id
    private String nim;
    private String nama;
    private String alamat;
    private String nid_ds; // Field tambahan sesuai permintaan

    // Constructors
    public Mahasiswa() {}

    public Mahasiswa(String nim, String nama, String alamat, String nid_ds) {
        this.nim = nim;
        this.nama = nama;
        this.alamat = alamat;
        this.nid_ds = nid_ds;
    }

    // Getters and Setters
    public String getNim() {
        return nim;
    }

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

    public String getNama() {
        return nama;
    }

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

    public String getAlamat() {
        return alamat;
    }

    public void setAlamat(String alamat) {
        this.alamat = alamat;
    }

    public String getNidDs() {
        return nid_ds;
    }

    public void setNidDs(String nid_ds) {
```



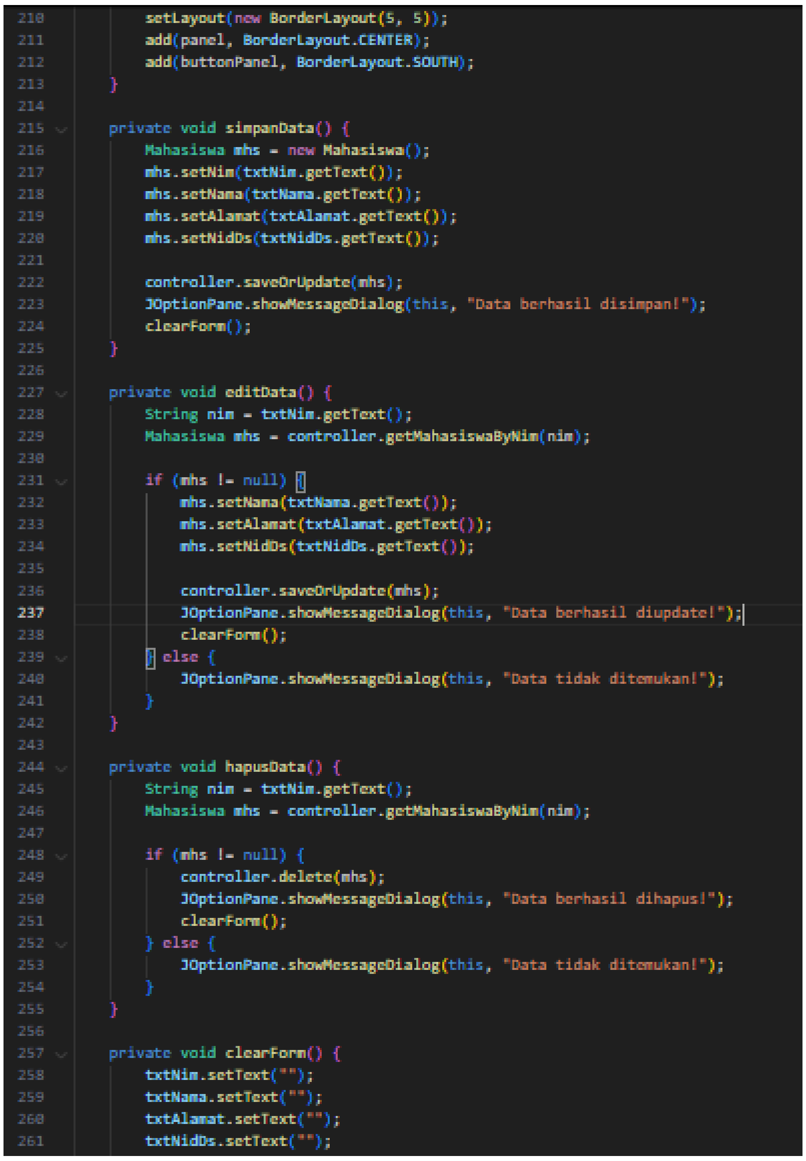
```java
        this.nid_ds = nid_ds;
    }
}

// File: HibernateUtil.java
package util;

import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

public class HibernateUtil {
    private static final SessionFactory sessionFactory;

    static {
        try {
            sessionFactory = new Configuration().configure().buildSessionFactory();
        } catch (Throwable ex) {
            System.err.println("Initial SessionFactory creation failed." + ex);
            throw new ExceptionInInitializerError(ex);
        }
    }

    public static SessionFactory getSessionFactory() {
        return sessionFactory;
    }
}

// File: MahasiswaController.java
package controller;

import model.Mahasiswa;
import org.hibernate.Session;
import org.hibernate.Transaction;
import util.HibernateUtil;

public class MahasiswaController {
    // Create/Update
    public void saveOrUpdate(Mahasiswa mahasiswa) {
        Session session = HibernateUtil.getSessionFactory().openSession();
        Transaction transaction = null;

        try {
            transaction = session.beginTransaction();
            session.saveOrUpdate(mahasiswa);
            transaction.commit();
        } catch (Exception e) {
            if (transaction != null) {
                transaction.rollback();
            }
            e.printStackTrace();
        } finally {
            session.close();
```



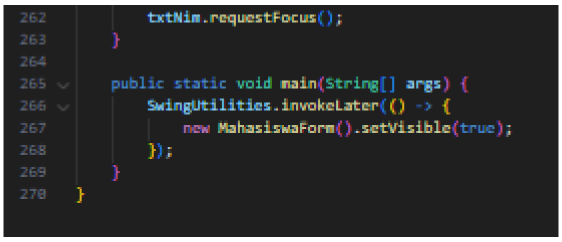
```java
        }
    }

    // Delete
    public void delete(Mahasiswa mahasiswa) {
        Session session = HibernateUtil.getSessionFactory().openSession();
        Transaction transaction = null;

        try {
            transaction = session.beginTransaction();
            session.delete(mahasiswa);
            transaction.commit();
        } catch (Exception e) {
            if (transaction != null) {
                transaction.rollback();
            }
            e.printStackTrace();
        } finally {
            session.close();
        }
    }

    // Read
    public Mahasiswa getMahasiswaByNim(String nim) {
        Session session = HibernateUtil.getSessionFactory().openSession();
        Mahasiswa mahasiswa = null;

        try {
            mahasiswa = (Mahasiswa) session.get(Mahasiswa.class, nim);
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            session.close();
        }

        return mahasiswa;
    }
}

// File: MahasiswaForm.java
package view;

import controller.MahasiswaController;
import model.Mahasiswa;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class MahasiswaForm extends JFrame {
    private JTextField txtNim, txtNama, txtAlamat, txtNidDs;
```



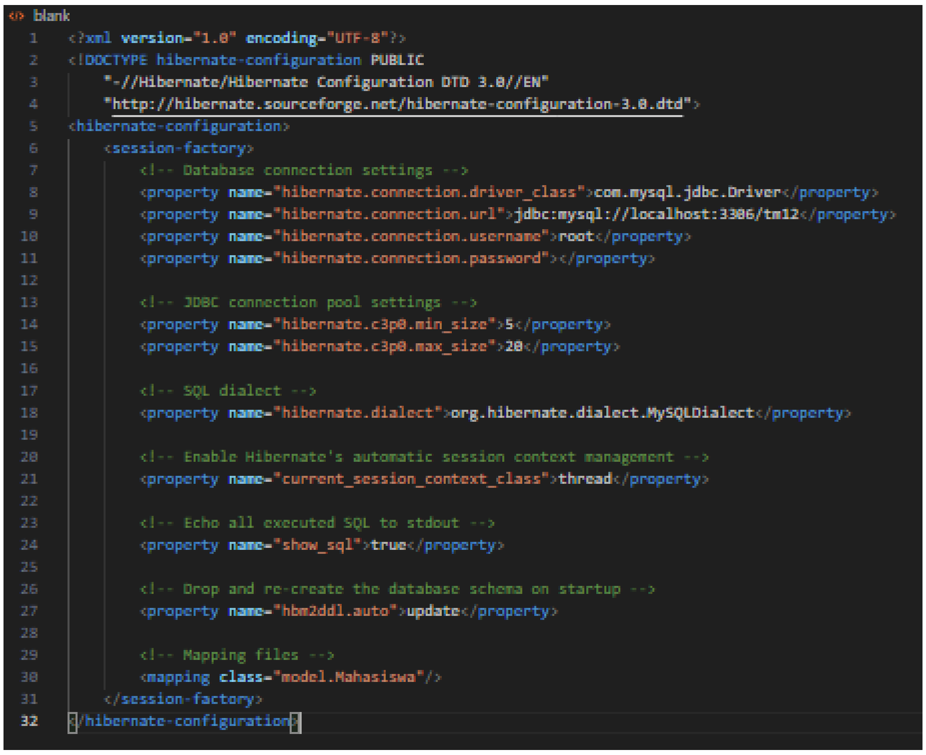
```java
    private JTextField txtNim, txtNama, txtAlamat, txtNidDs;
    private JButton btnSimpan, btnEdit, btnHapus, btnClear;
    private MahasiswaController controller;

    public MahasiswaForm() {
        controller = new MahasiswaController();
        initComponents();
    }

    private void initComponents() {
        setTitle("Form Mahasiswa");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(400, 300);
        setLocationRelativeTo(null);

        // Initialize components
        JPanel panel = new JPanel(new GridLayout(5, 2, 5, 5));

        panel.add(new JLabel("NIM:"));
        txtNim = new JTextField();
        panel.add(txtNim);

        panel.add(new JLabel("Nama:"));
        txtNama = new JTextField();
        panel.add(txtNama);

        panel.add(new JLabel("Alamat:"));
        txtAlamat = new JTextField();
        panel.add(txtAlamat);

        panel.add(new JLabel("NID DS:"));
        txtNidDs = new JTextField();
        panel.add(txtNidDs);

        // Buttons
        JPanel buttonPanel = new JPanel(new FlowLayout());

        btnSimpan = new JButton("Simpan");
        btnEdit = new JButton("Edit");
        btnHapus = new JButton("Hapus");
        btnClear = new JButton("Clear");

        buttonPanel.add(btnSimpan);
        buttonPanel.add(btnEdit);
        buttonPanel.add(btnHapus);
        buttonPanel.add(btnClear);

        // Add action listeners
        btnSimpan.addActionListener(e -> simpanData());
        btnEdit.addActionListener(e -> editData());
        btnHapus.addActionListener(e -> hapusData());
        btnClear.addActionListener(e -> clearForm());

```



```java
        setLayout(new BorderLayout(5, 5));
        add(panel, BorderLayout.CENTER);
        add(buttonPanel, BorderLayout.SOUTH);
    }

    private void simpanData() {
        Mahasiswa mhs = new Mahasiswa();
        mhs.setNim(txtNim.getText());
        mhs.setNama(txtNama.getText());
        mhs.setAlamat(txtAlamat.getText());
        mhs.setNidDs(txtNidDs.getText());

        controller.saveOrUpdate(mhs);
        JOptionPane.showMessageDialog(this, "Data berhasil disimpan!");
        clearForm();
    }

    private void editData() {
        String nim = txtNim.getText();
        Mahasiswa mhs = controller.getMahasiswaByNim(nim);

        if (mhs != null) {
            mhs.setNama(txtNama.getText());
            mhs.setAlamat(txtAlamat.getText());
            mhs.setNidDs(txtNidDs.getText());

            controller.saveOrUpdate(mhs);
            JOptionPane.showMessageDialog(this, "Data berhasil diupdate!");
            clearForm();
        } else {
            JOptionPane.showMessageDialog(this, "Data tidak ditemukan!");
        }
    }

    private void hapusData() {
        String nim = txtNim.getText();
        Mahasiswa mhs = controller.getMahasiswaByNim(nim);

        if (mhs != null) {
            controller.delete(mhs);
            JOptionPane.showMessageDialog(this, "Data berhasil dihapus!");
            clearForm();
        } else {
            JOptionPane.showMessageDialog(this, "Data tidak ditemukan!");
        }
    }

    private void clearForm() {
        txtNim.setText("");
        txtNama.setText("");
        txtAlamat.setText("");
        txtNidDs.setText("");
```



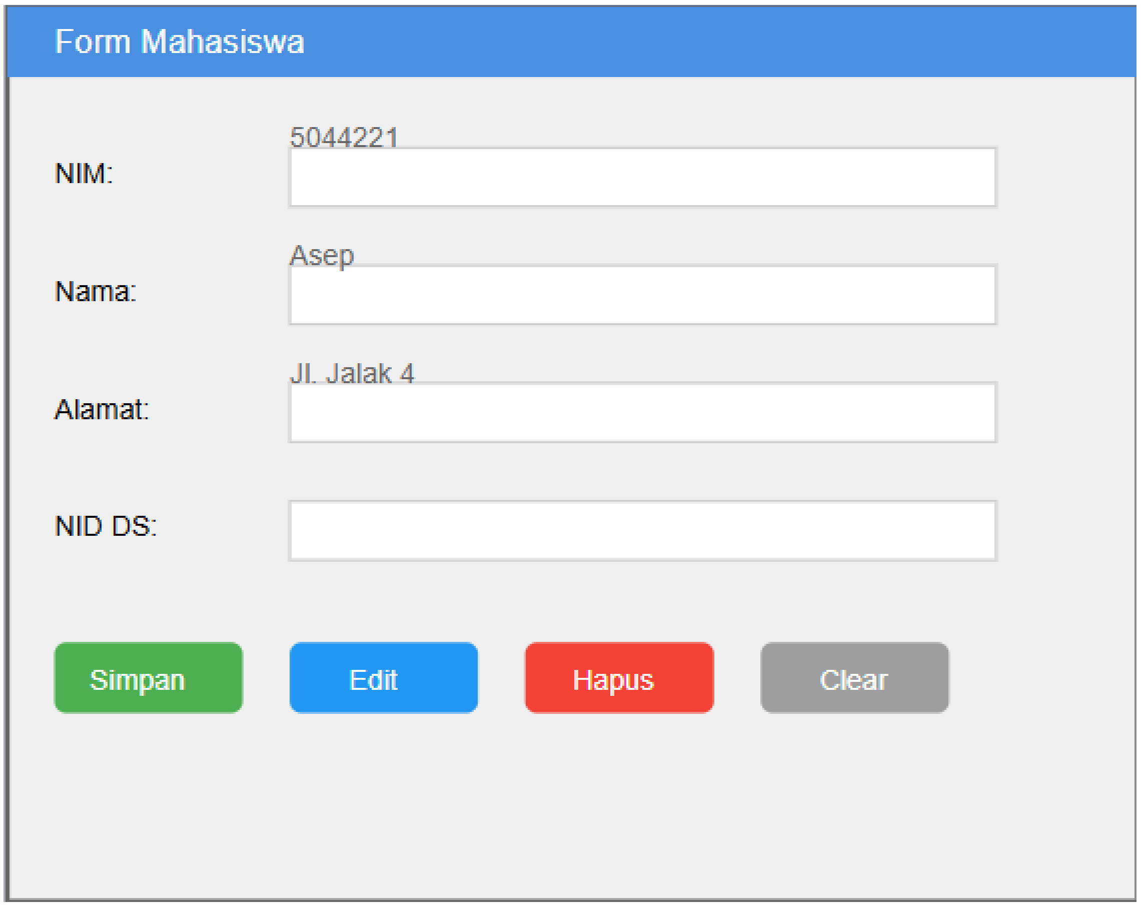
```java
        txtNim.requestFocus();
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            new MahasiswaForm().setVisible(true);
        });
    }
}
```



blank

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
    <session-factory>
        <!-- Database connection settings -->
        <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
        <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/tm12</property>
        <property name="hibernate.connection.username">root</property>
        <property name="hibernate.connection.password"></property>

        <!-- JDBC connection pool settings -->
        <property name="hibernate.c3p0.min_size">5</property>
        <property name="hibernate.c3p0.max_size">20</property>

        <!-- SQL dialect -->
        <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>

        <!-- Enable Hibernate's automatic session context management -->
        <property name="current_session_context_class">thread</property>

        <!-- Echo all executed SQL to stdout -->
        <property name="show_sql">true</property>

        <!-- Drop and re-create the database schema on startup -->
        <property name="hbm2ddl.auto">update</property>

        <!-- Mapping files -->
        <mapping class="model.Mahasiswa"/>
    </session-factory>
</hibernate-configuration>
```

Output


Form Mahasiswa

NIM:	5044221
Nama:	Asep
Alamat:	Jl. Jalak 4
NID DS:	