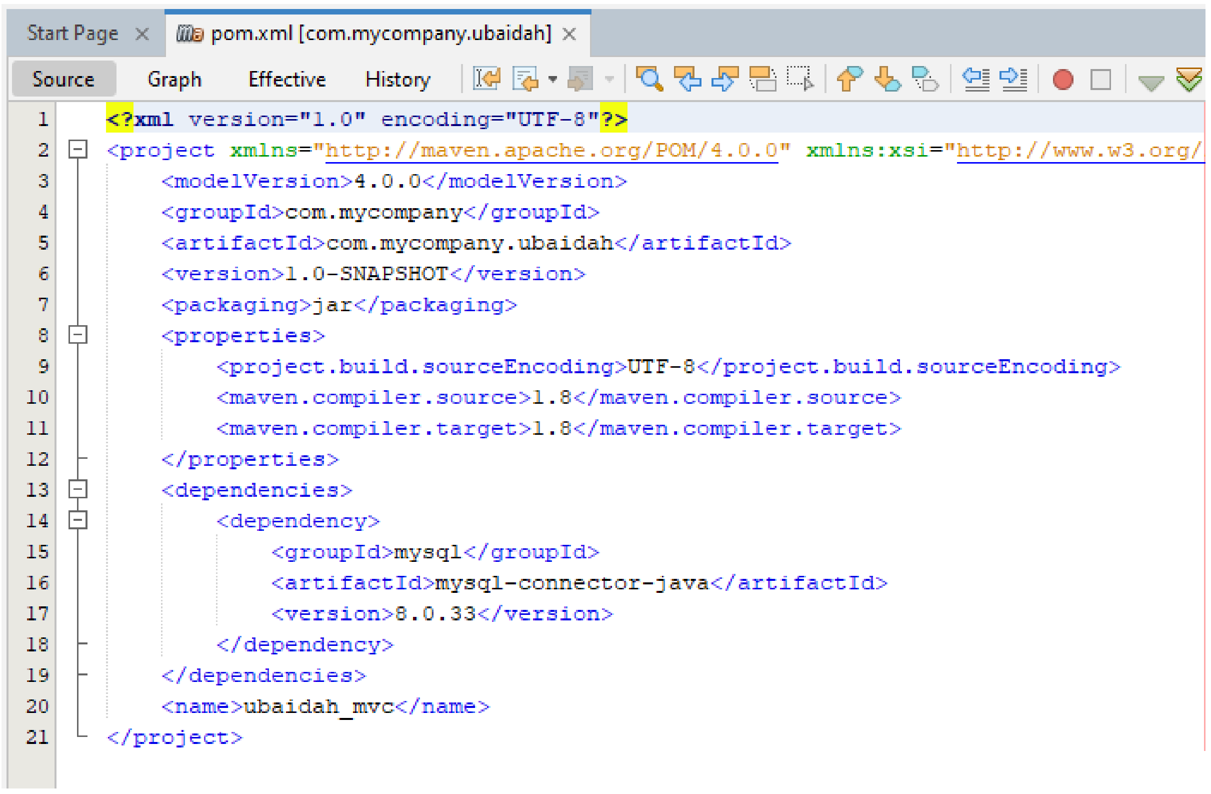
ACTIVITY PERTEMUAN 3

Nama : Ubaidah Luthfiyah Zain

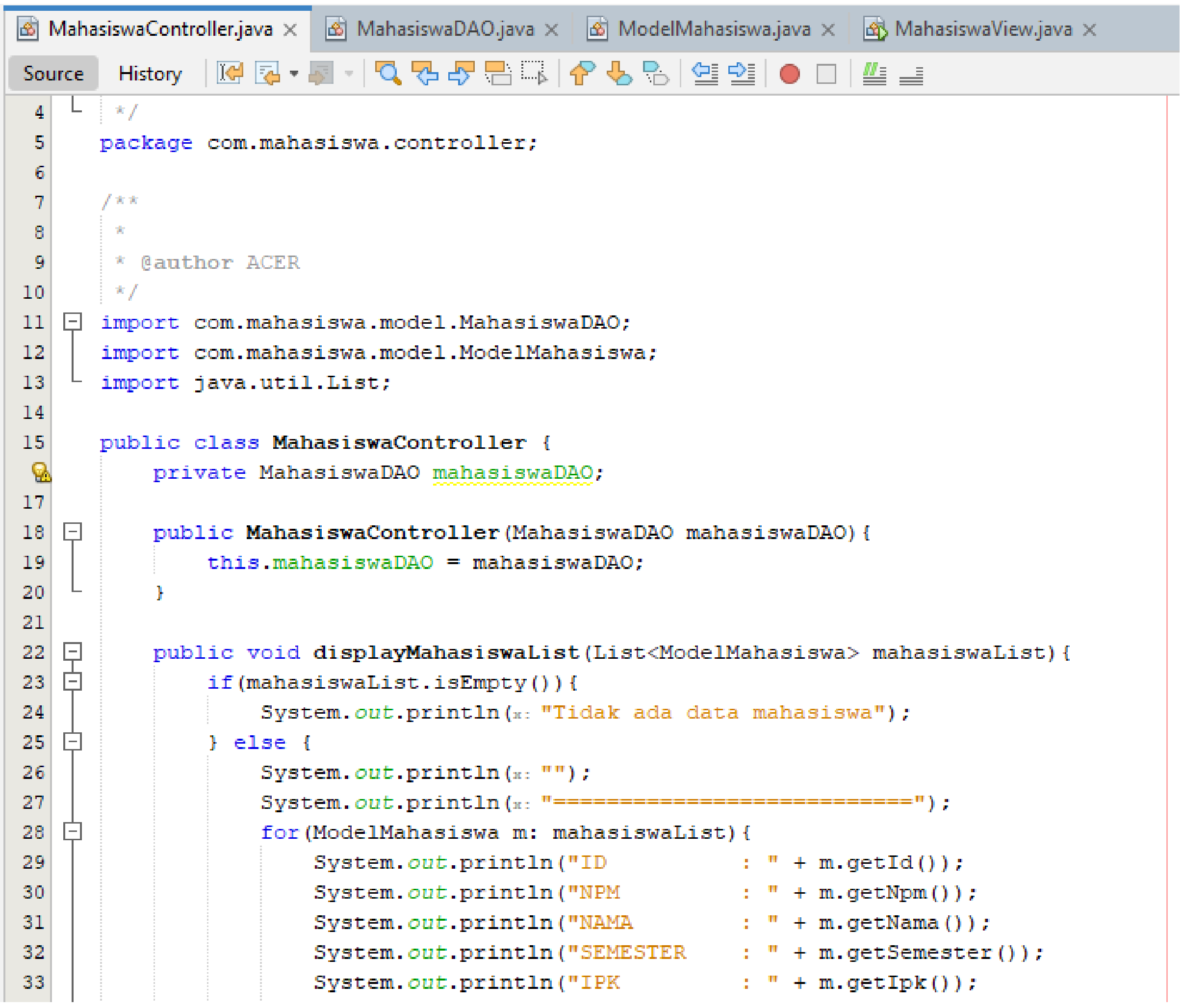
NPM : 51421491

Kelas : 4IA28

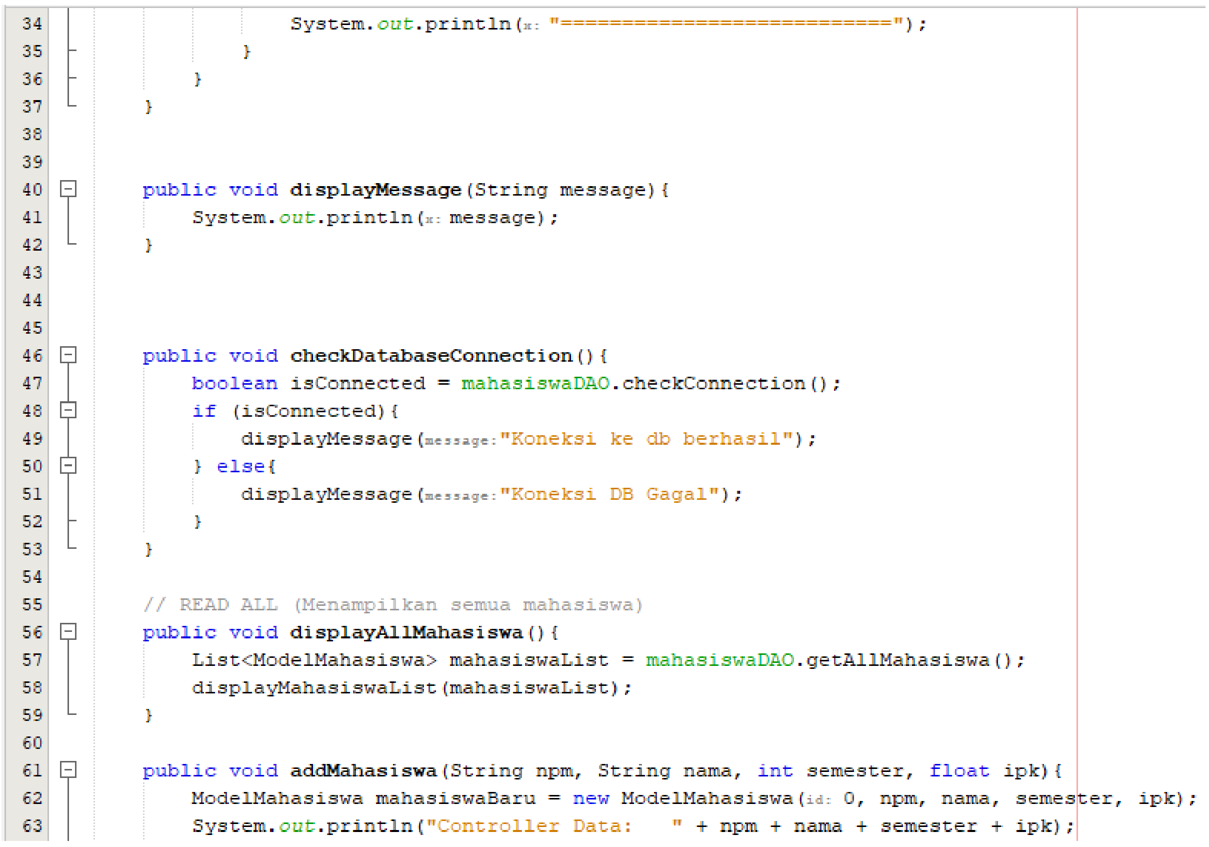
Source Code



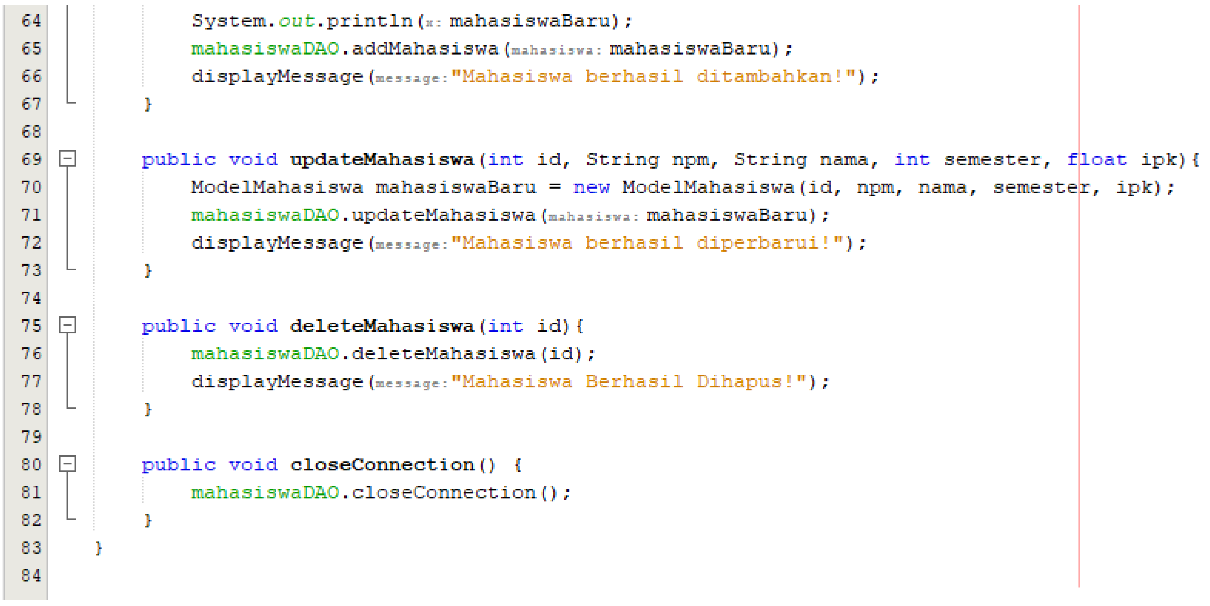
```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.mycompany</groupId>
    <artifactId>com.mycompany.ubaidah</artifactId>
    <version>1.0-SNAPSHOT</version>
    <packaging>jar</packaging>
    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
    </properties>
    <dependencies>
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>8.0.33</version>
        </dependency>
    </dependencies>
    <name>ubaidah_mvc</name>
</project>
```



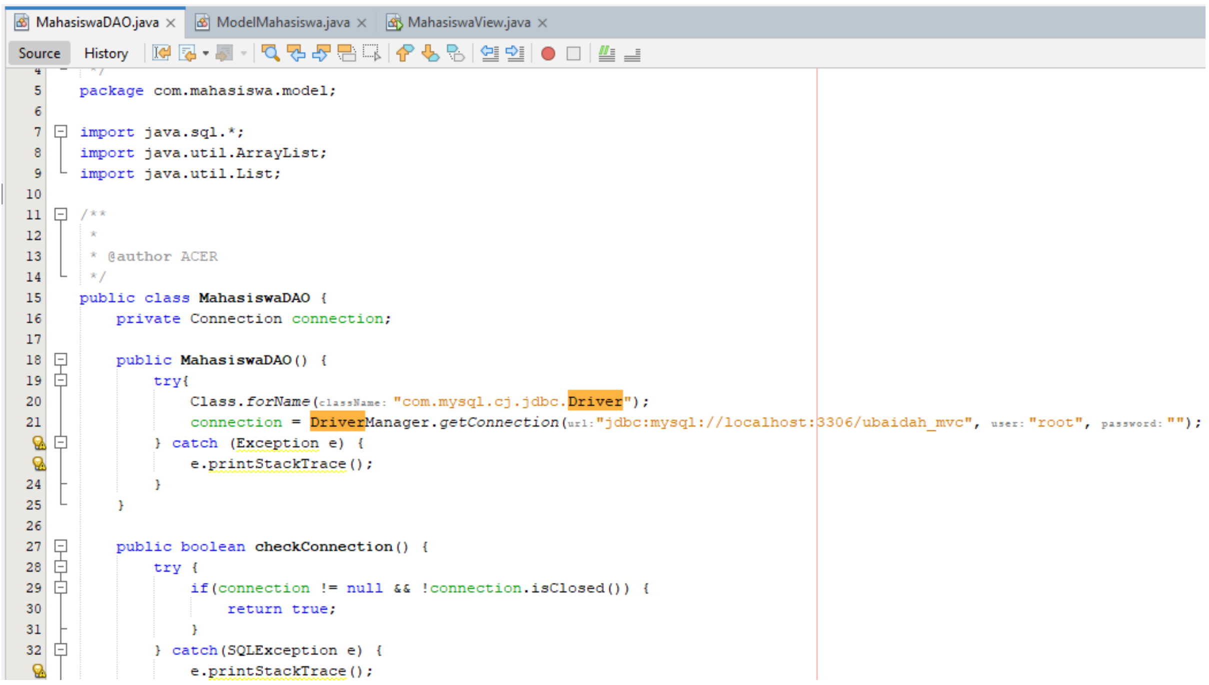
```java
 */
package com.mahasiswa.controller;

/**
 *
 * @author ACER
 */
import com.mahasiswa.model.MahasiswaDAO;
import com.mahasiswa.model.ModelMahasiswa;
import java.util.List;

public class MahasiswaController {
    private MahasiswaDAO mahasiswaDAO;

    public MahasiswaController(MahasiswaDAO mahasiswaDAO){
        this.mahasiswaDAO = mahasiswaDAO;
    }

    public void displayMahasiswaList(List<ModelMahasiswa> mahasiswaList){
        if(mahasiswaList.isEmpty()){
            System.out.println(x: "Tidak ada data mahasiswa");
        } else {
            System.out.println(x: "");
            System.out.println(x: "===========================");
            for(ModelMahasiswa m: mahasiswaList){
                System.out.println("ID          : " + m.getId());
                System.out.println("NPM         : " + m.getNpm());
                System.out.println("NAMA        : " + m.getNama());
                System.out.println("SEMESTER    : " + m.getSemester());
                System.out.println("IPK         : " + m.getIpk());
```



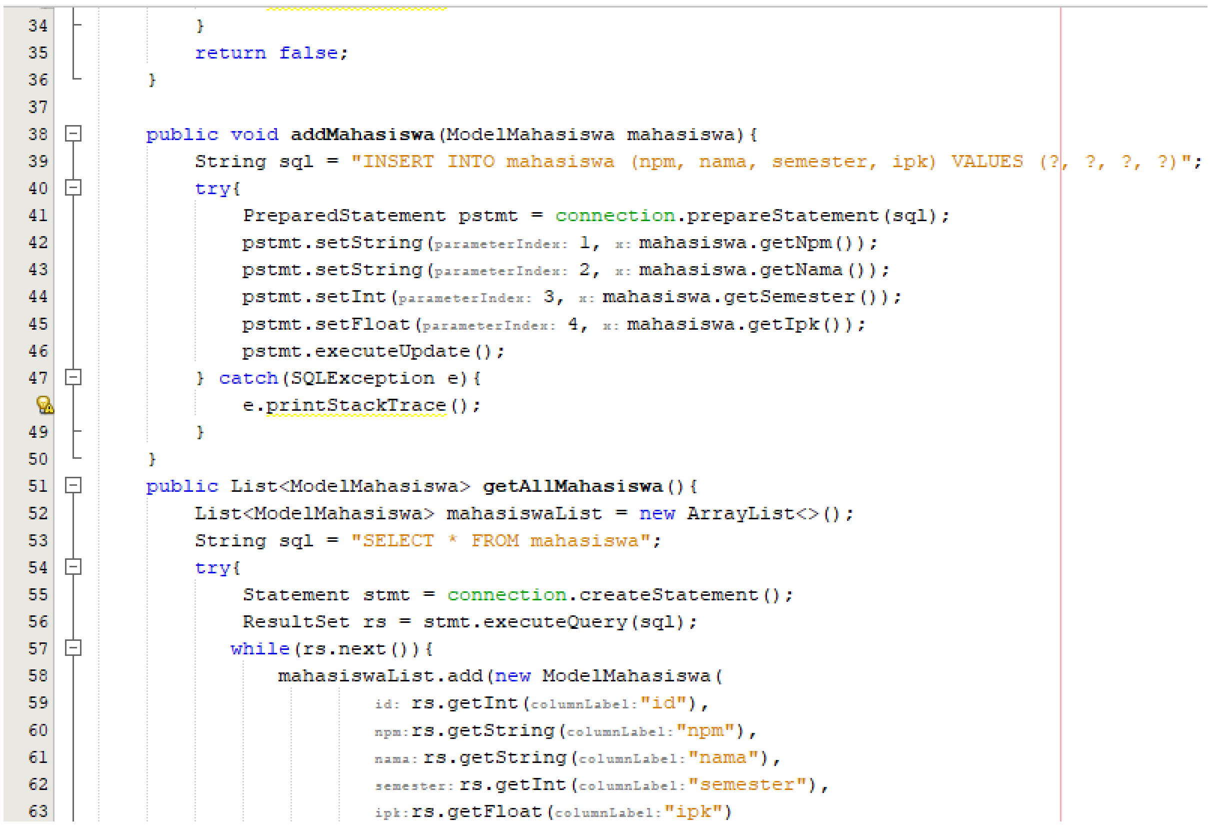
```java
                    System.out.println(x: "===========================");
                }
            }
        }


        public void displayMessage(String message){
            System.out.println(x: message);
        }



        public void checkDatabaseConnection(){
            boolean isConnected = mahasiswaDAO.checkConnection();
            if (isConnected){
                displayMessage(message:"Koneksi ke db berhasil");
            } else{
                displayMessage(message:"Koneksi DB Gagal");
            }
        }

        // READ ALL (Menampilkan semua mahasiswa)
        public void displayAllMahasiswa(){
            List<ModelMahasiswa> mahasiswaList = mahasiswaDAO.getAllMahasiswa();
            displayMahasiswaList(mahasiswaList);
        }

        public void addMahasiswa(String npm, String nama, int semester, float ipk){
            ModelMahasiswa mahasiswaBaru = new ModelMahasiswa(id: 0, npm, nama, semester, ipk);
            System.out.println("Controller Data:    " + npm + nama + semester + ipk);
```



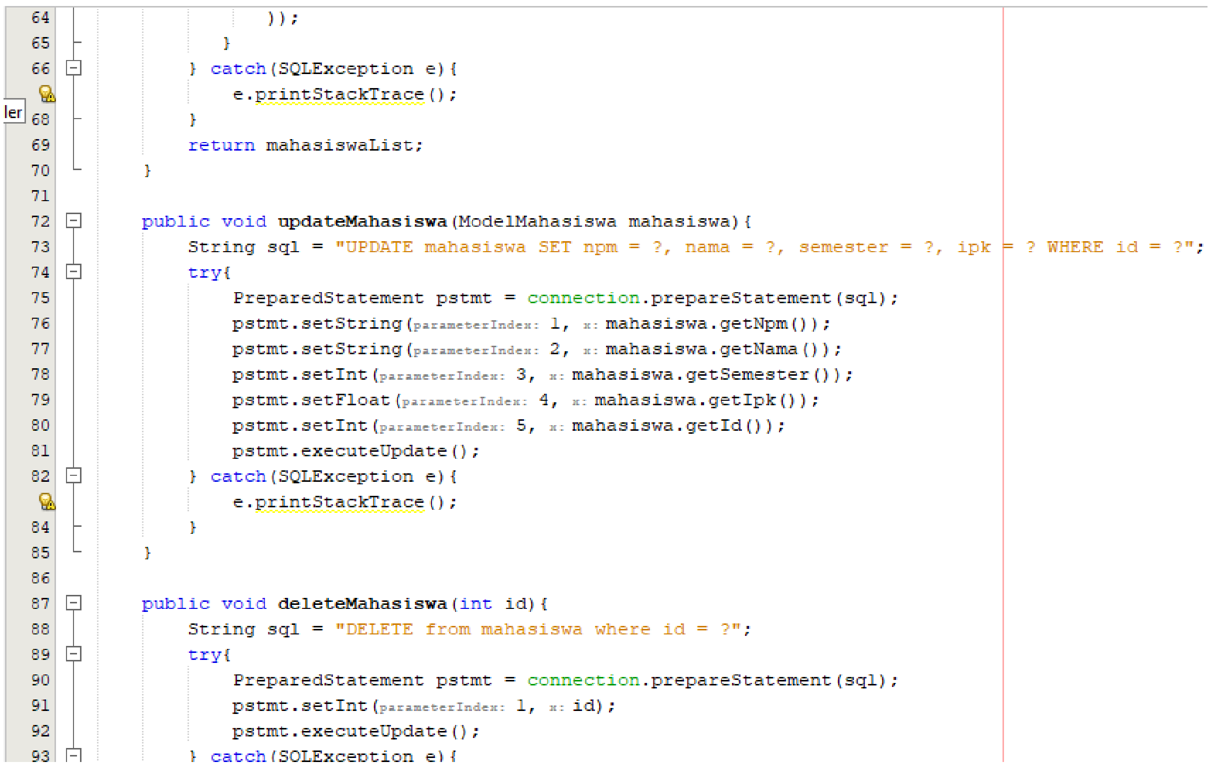
```java
            System.out.println(x: mahasiswaBaru);
            mahasiswaDAO.addMahasiswa(mahasiswa: mahasiswaBaru);
            displayMessage(message:"Mahasiswa berhasil ditambahkan!");
        }

        public void updateMahasiswa(int id, String npm, String nama, int semester, float ipk){
            ModelMahasiswa mahasiswaBaru = new ModelMahasiswa(id, npm, nama, semester, ipk);
            mahasiswaDAO.updateMahasiswa(mahasiswa: mahasiswaBaru);
            displayMessage(message:"Mahasiswa berhasil diperbarui!");
        }

        public void deleteMahasiswa(int id){
            mahasiswaDAO.deleteMahasiswa(id);
            displayMessage(message:"Mahasiswa Berhasil Dihapus!");
        }

        public void closeConnection() {
            mahasiswaDAO.closeConnection();
        }
    }
```



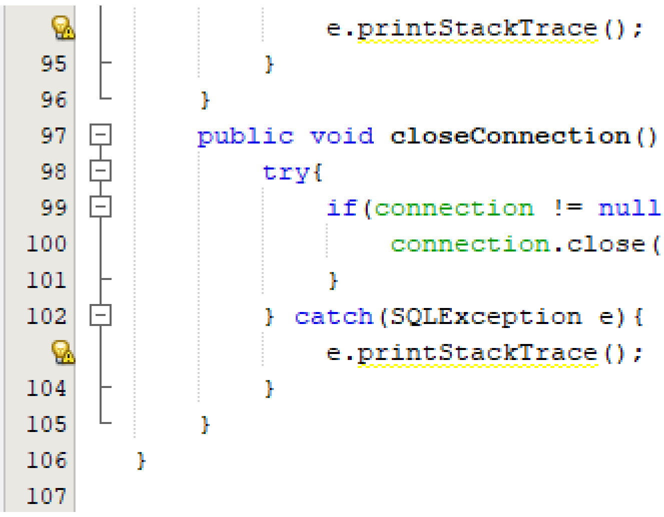
```java
package com.mahasiswa.model;

import java.sql.*;
import java.util.ArrayList;
import java.util.List;

/**
 *
 * @author ACER
 */
public class MahasiswaDAO {
    private Connection connection;

    public MahasiswaDAO() {
        try{
            Class.forName(className: "com.mysql.cj.jdbc.Driver");
            connection = DriverManager.getConnection(url:"jdbc:mysql://localhost:3306/ubaidah_mvc", user: "root", password: "");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public boolean checkConnection() {
        try {
            if(connection != null && !connection.isClosed()) {
                return true;
            }
        } catch(SQLException e) {
            e.printStackTrace();
```



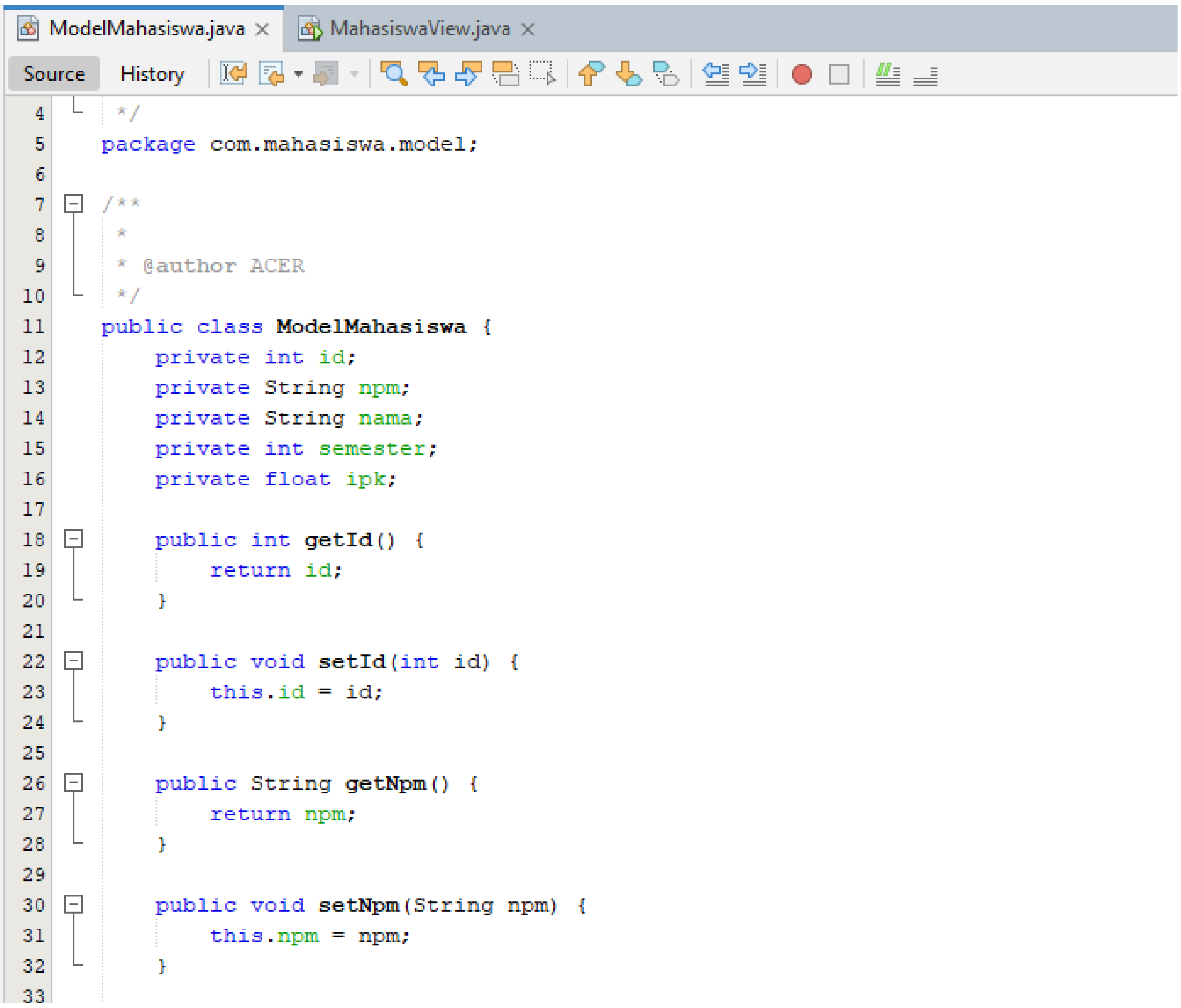
```java
            }
            return false;
        }

    public void addMahasiswa(ModelMahasiswa mahasiswa){
        String sql = "INSERT INTO mahasiswa (npm, nama, semester, ipk) VALUES (?, ?, ?, ?)";
        try{
            PreparedStatement pstmt = connection.prepareStatement(sql);
            pstmt.setString(parameterIndex: 1, x: mahasiswa.getNpm());
            pstmt.setString(parameterIndex: 2, x: mahasiswa.getNama());
            pstmt.setInt(parameterIndex: 3, x: mahasiswa.getSemester());
            pstmt.setFloat(parameterIndex: 4, x: mahasiswa.getIpk());
            pstmt.executeUpdate();
        } catch(SQLException e){
            e.printStackTrace();
        }
    }
    public List<ModelMahasiswa> getAllMahasiswa(){
        List<ModelMahasiswa> mahasiswaList = new ArrayList<>();
        String sql = "SELECT * FROM mahasiswa";
        try{
            Statement stmt = connection.createStatement();
            ResultSet rs = stmt.executeQuery(sql);
           while(rs.next()){
               mahasiswaList.add(new ModelMahasiswa(
                       id: rs.getInt(columnLabel:"id"),
                       npm:rs.getString(columnLabel:"npm"),
                       nama: rs.getString(columnLabel:"nama"),
                       semester: rs.getInt(columnLabel:"semester"),
                       ipk:rs.getFloat(columnLabel:"ipk")
```



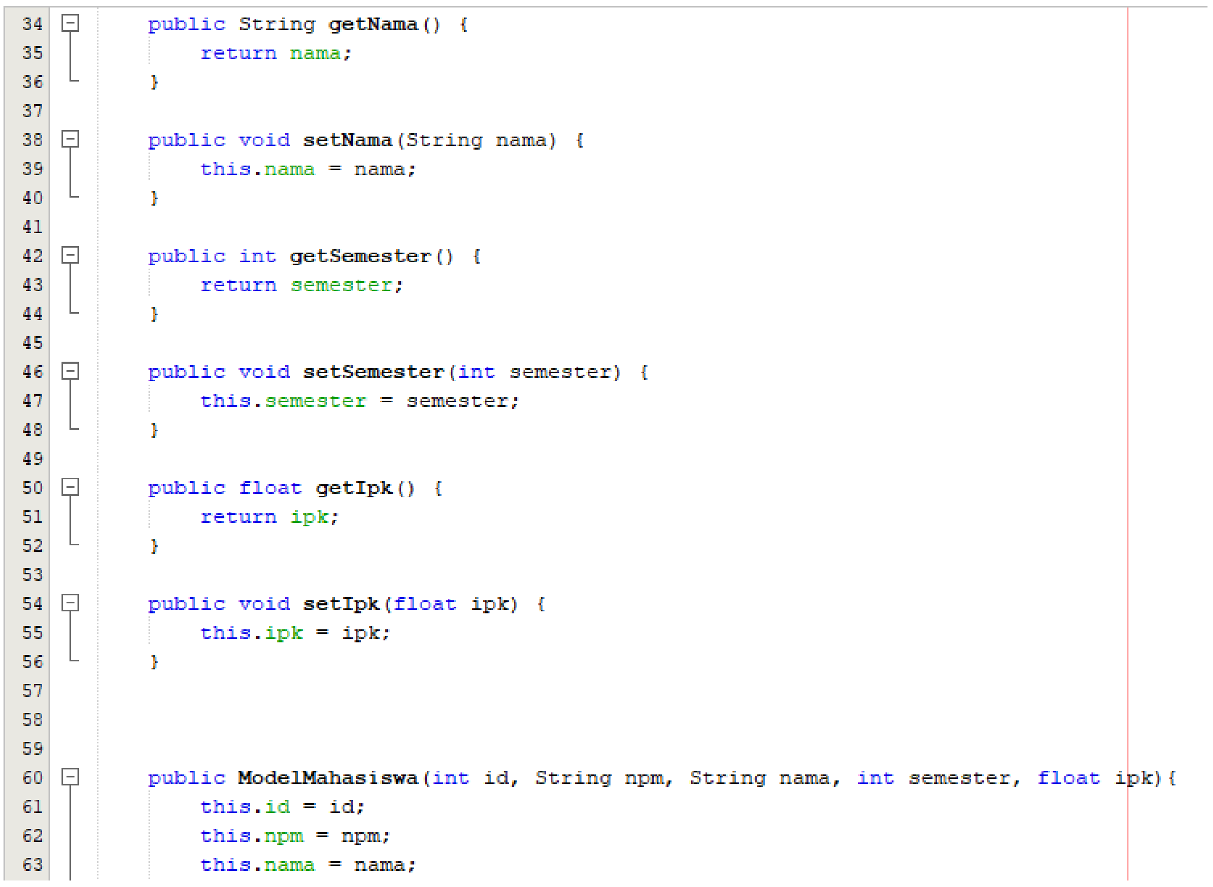
```
                ));
            }
        } catch(SQLException e){
            e.printStackTrace();
        }
        return mahasiswaList;
    }

    public void updateMahasiswa(ModelMahasiswa mahasiswa){
        String sql = "UPDATE mahasiswa SET npm = ?, nama = ?, semester = ?, ipk = ? WHERE id = ?";
        try{
            PreparedStatement pstmt = connection.prepareStatement(sql);
            pstmt.setString(parameterIndex: 1, x: mahasiswa.getNpm());
            pstmt.setString(parameterIndex: 2, x: mahasiswa.getNama());
            pstmt.setInt(parameterIndex: 3, x: mahasiswa.getSemester());
            pstmt.setFloat(parameterIndex: 4, x: mahasiswa.getIpk());
            pstmt.setInt(parameterIndex: 5, x: mahasiswa.getId());
            pstmt.executeUpdate();
        } catch(SQLException e){
            e.printStackTrace();
        }
    }

    public void deleteMahasiswa(int id){
        String sql = "DELETE from mahasiswa where id = ?";
        try{
            PreparedStatement pstmt = connection.prepareStatement(sql);
            pstmt.setInt(parameterIndex: 1, x: id);
            pstmt.executeUpdate();
        } catch(SQLException e){
```



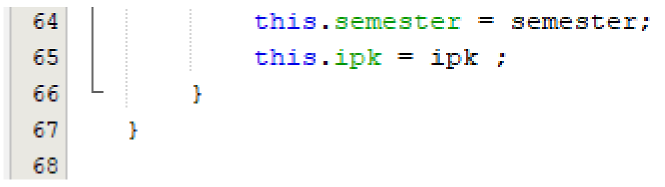
```
            e.printStackTrace();
        }
    }
    public void closeConnection(){
        try{
            if(connection != null){
                connection.close();
            }
        } catch(SQLException e){
            e.printStackTrace();
        }
    }
}
```



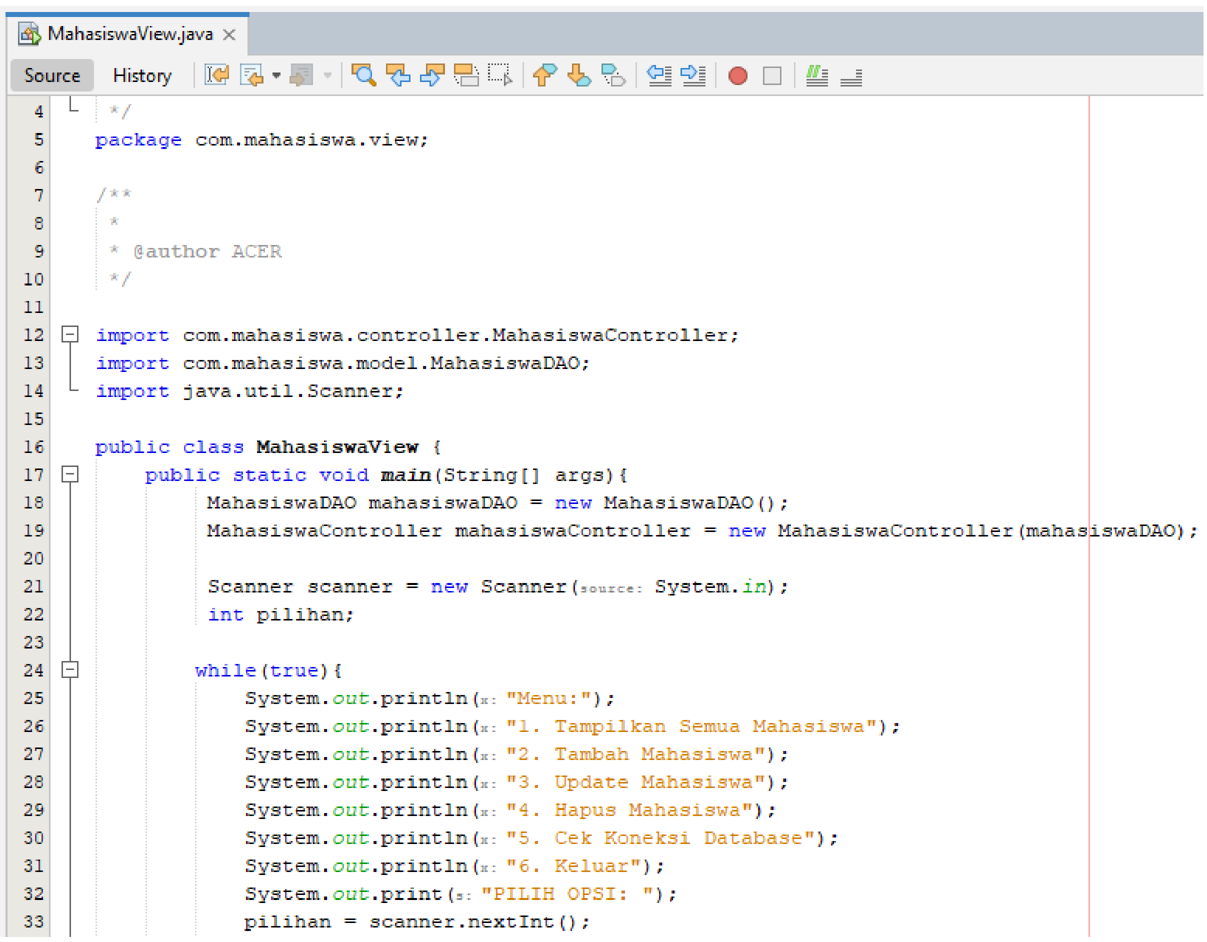
```
 */
package com.mahasiswa.model;

/**
 *
 * @author ACER
 */
public class ModelMahasiswa {
    private int id;
    private String npm;
    private String nama;
    private int semester;
    private float 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;
    }

```



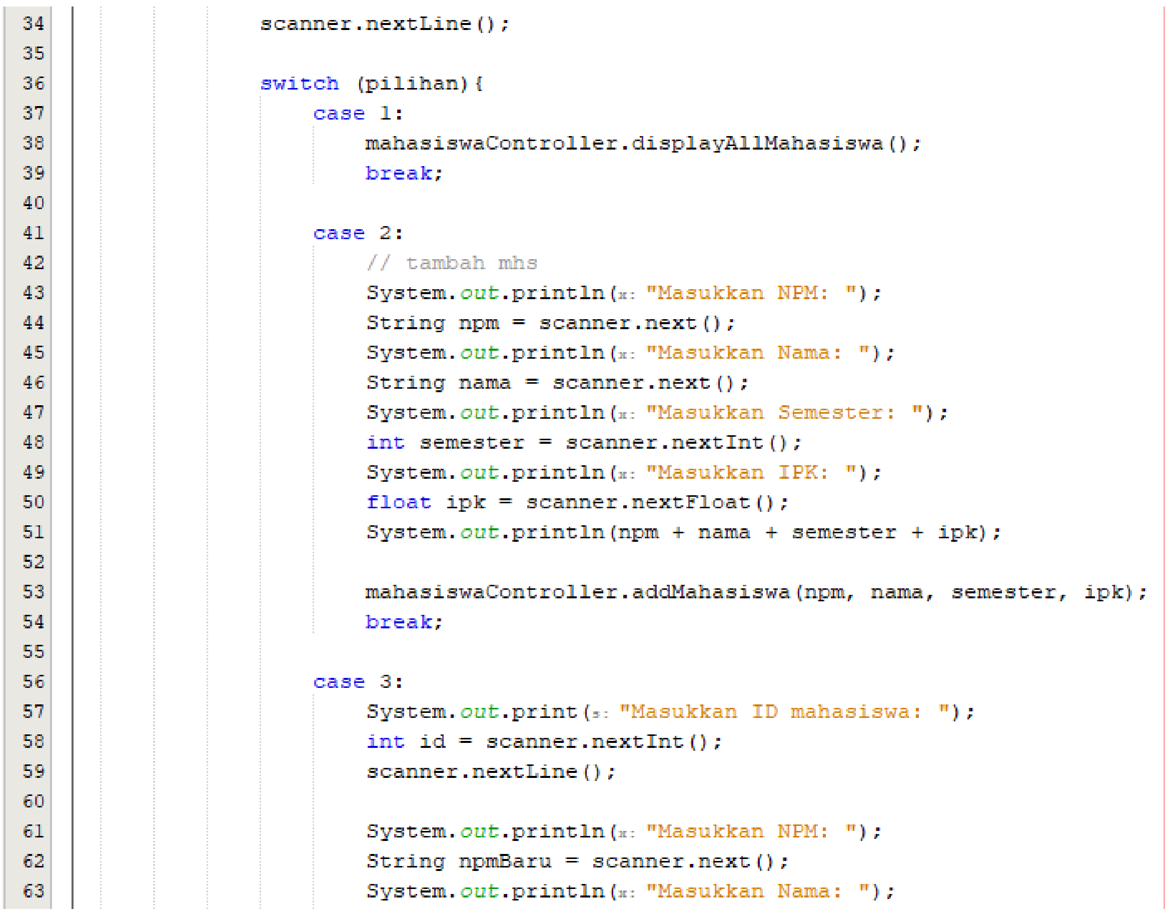
```java
    public String getNama() {
        return nama;
    }

    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;
    }



    public ModelMahasiswa(int id, String npm, String nama, int semester, float ipk){
        this.id = id;
        this.npm = npm;
        this.nama = nama;
```



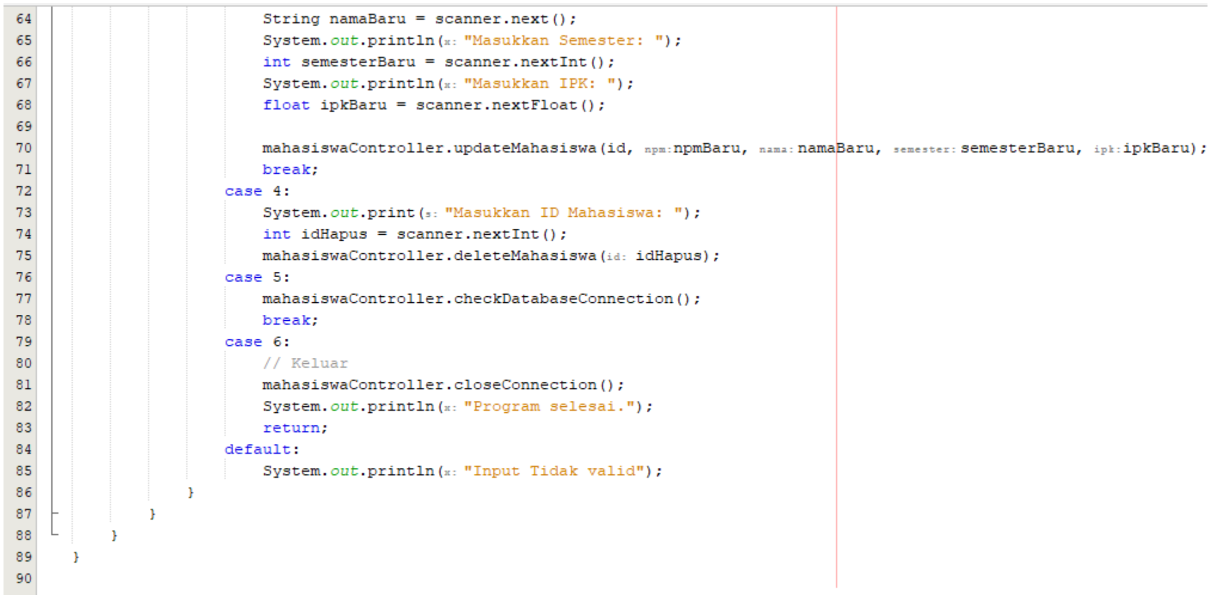
```java
        this.semester = semester;
        this.ipk = ipk ;
    }
}

```



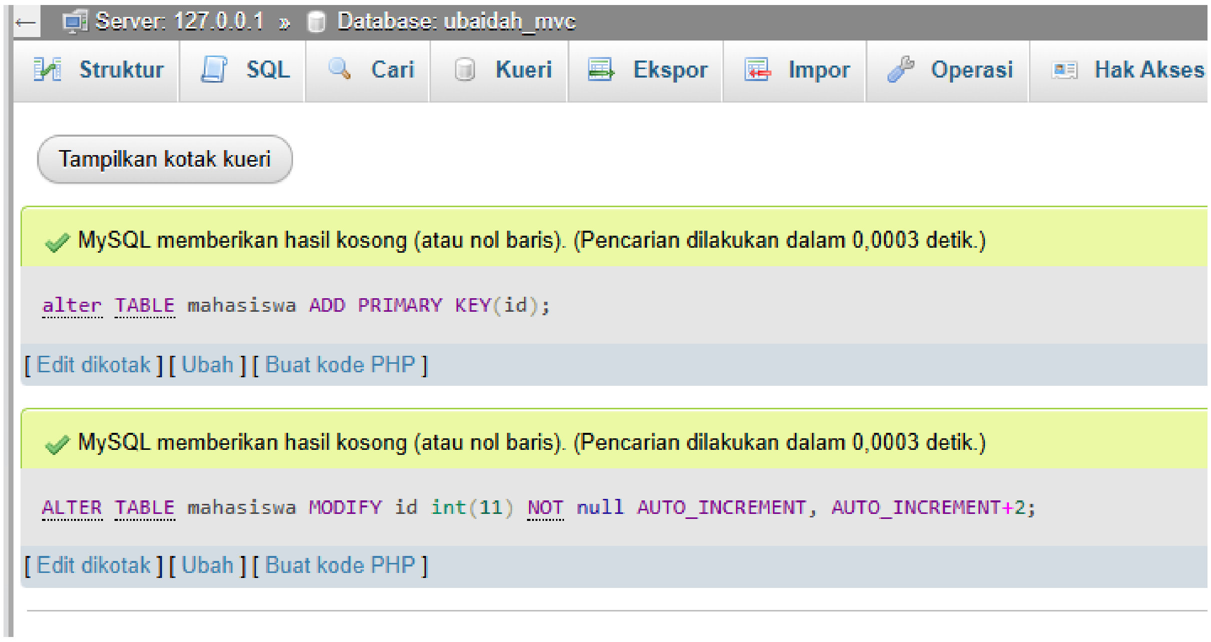
```java
 */
package com.mahasiswa.view;

/**
 *
 * @author ACER
 */

import com.mahasiswa.controller.MahasiswaController;
import com.mahasiswa.model.MahasiswaDAO;
import java.util.Scanner;

public class MahasiswaView {
    public static void main(String[] args){
         MahasiswaDAO mahasiswaDAO = new MahasiswaDAO();
         MahasiswaController mahasiswaController = new MahasiswaController(mahasiswaDAO);

         Scanner scanner = new Scanner(source: System.in);
         int pilihan;

        while(true){
            System.out.println(x: "Menu:");
            System.out.println(x: "1. Tampilkan Semua Mahasiswa");
            System.out.println(x: "2. Tambah Mahasiswa");
            System.out.println(x: "3. Update Mahasiswa");
            System.out.println(x: "4. Hapus Mahasiswa");
            System.out.println(x: "5. Cek Koneksi Database");
            System.out.println(x: "6. Keluar");
            System.out.print(s: "PILIH OPSI: ");
            pilihan = scanner.nextInt();
```



```java
            scanner.nextLine();

            switch (pilihan){
                case 1:
                    mahasiswaController.displayAllMahasiswa();
                    break;

                case 2:
                    // tambah mhs
                    System.out.println(x: "Masukkan NPM: ");
                    String npm = scanner.next();
                    System.out.println(x: "Masukkan Nama: ");
                    String nama = scanner.next();
                    System.out.println(x: "Masukkan Semester: ");
                    int semester = scanner.nextInt();
                    System.out.println(x: "Masukkan IPK: ");
                    float ipk = scanner.nextFloat();
                    System.out.println(npm + nama + semester + ipk);

                    mahasiswaController.addMahasiswa(npm, nama, semester, ipk);
                    break;

                case 3:
                    System.out.print(s: "Masukkan ID mahasiswa: ");
                    int id = scanner.nextInt();
                    scanner.nextLine();

                    System.out.println(x: "Masukkan NPM: ");
                    String npmBaru = scanner.next();
                    System.out.println(x: "Masukkan Nama: ");
```



```java
                        String namaBaru = scanner.next();
                        System.out.println(x: "Masukkan Semester: ");
                        int semesterBaru = scanner.nextInt();
                        System.out.println(x: "Masukkan IPK: ");
                        float ipkBaru = scanner.nextFloat();

                        mahasiswaController.updateMahasiswa(id, npm:npmBaru, nama: namaBaru, semester: semesterBaru, ipk:ipkBaru);
                        break;
                    case 4:
                        System.out.print(s: "Masukkan ID Mahasiswa: ");
                        int idHapus = scanner.nextInt();
                        mahasiswaController.deleteMahasiswa(id: idHapus);
                    case 5:
                        mahasiswaController.checkDatabaseConnection();
                        break;
                    case 6:
                        // Keluar
                        mahasiswaController.closeConnection();
                        System.out.println(x: "Program selesai.");
                        return;
                    default:
                        System.out.println(x: "Input Tidak valid");
                }
            }
        }
    }
```



MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,0003 detik.)

```sql
alter TABLE mahasiswa ADD PRIMARY KEY(id);
```

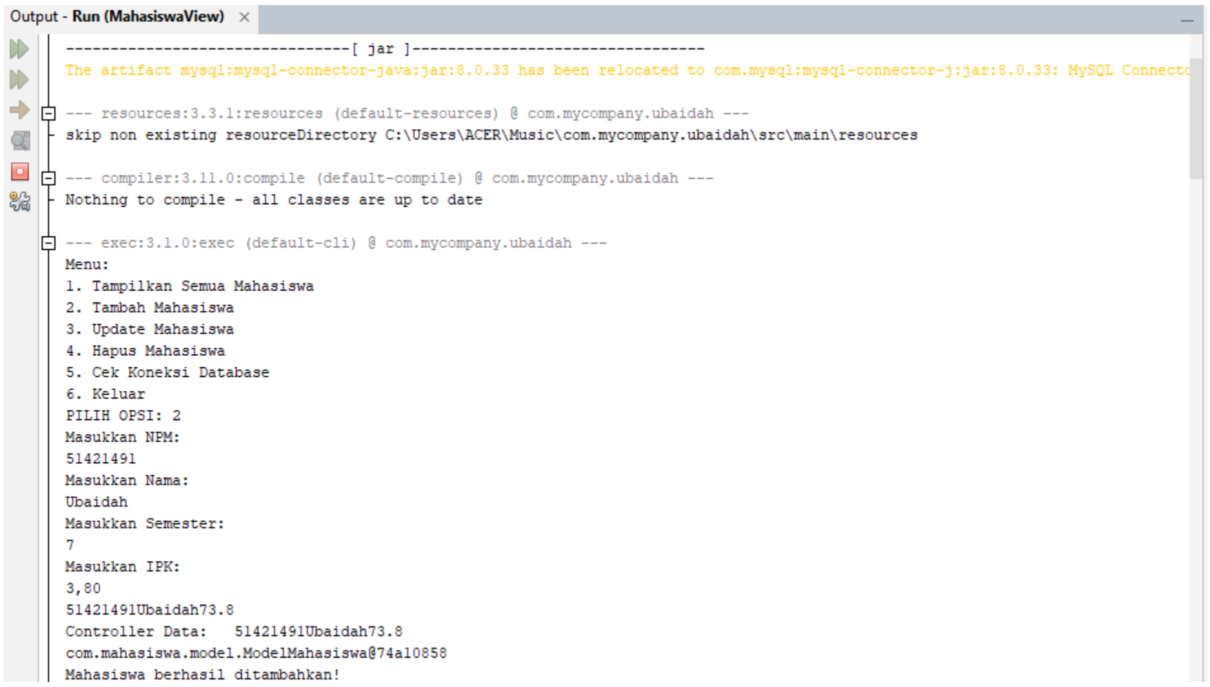
[Edit dikotak] [Ubah] [Buat kode PHP]

MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,0003 detik.)

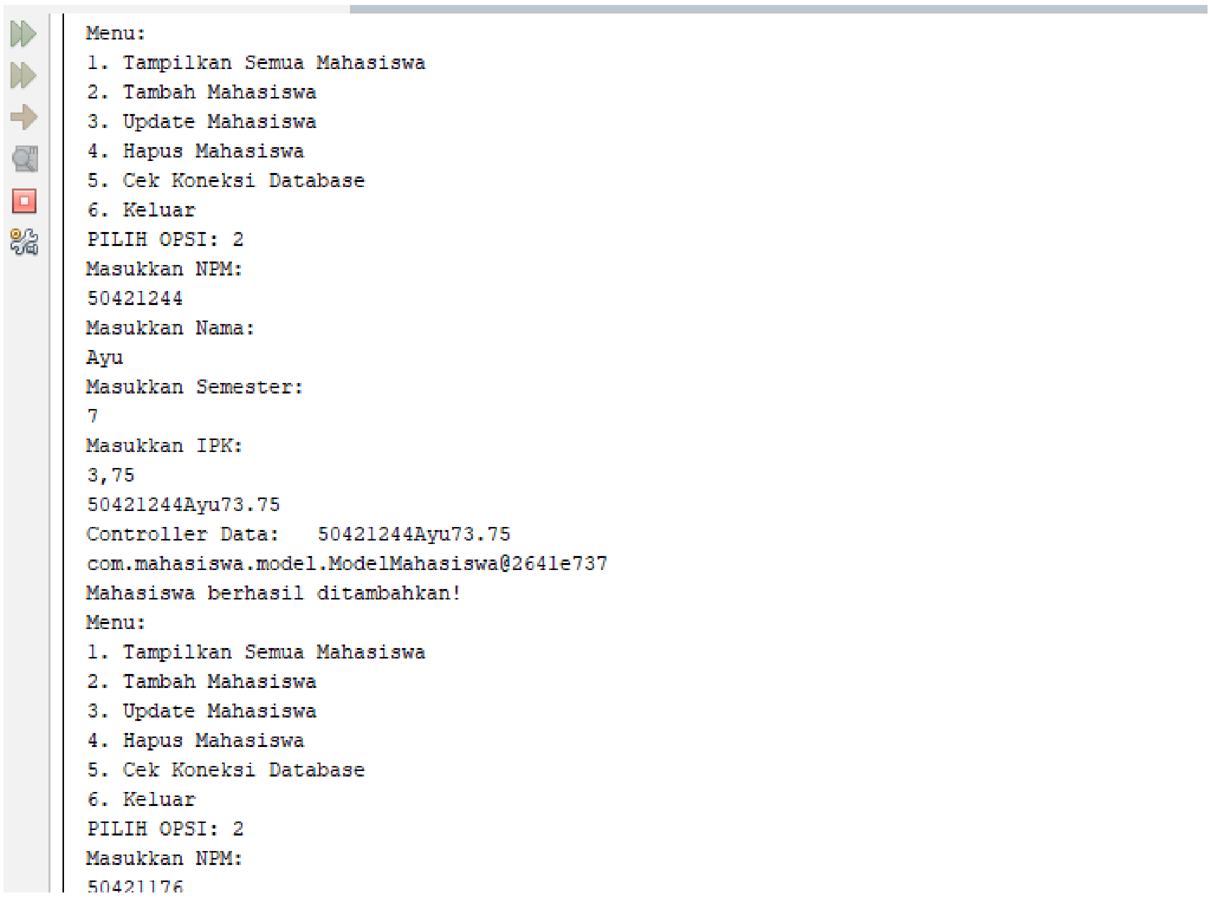
```sql
ALTER TABLE mahasiswa MODIFY id int(11) NOT null AUTO_INCREMENT, AUTO_INCREMENT+2;
```

[Edit dikotak] [Ubah] [Buat kode PHP]

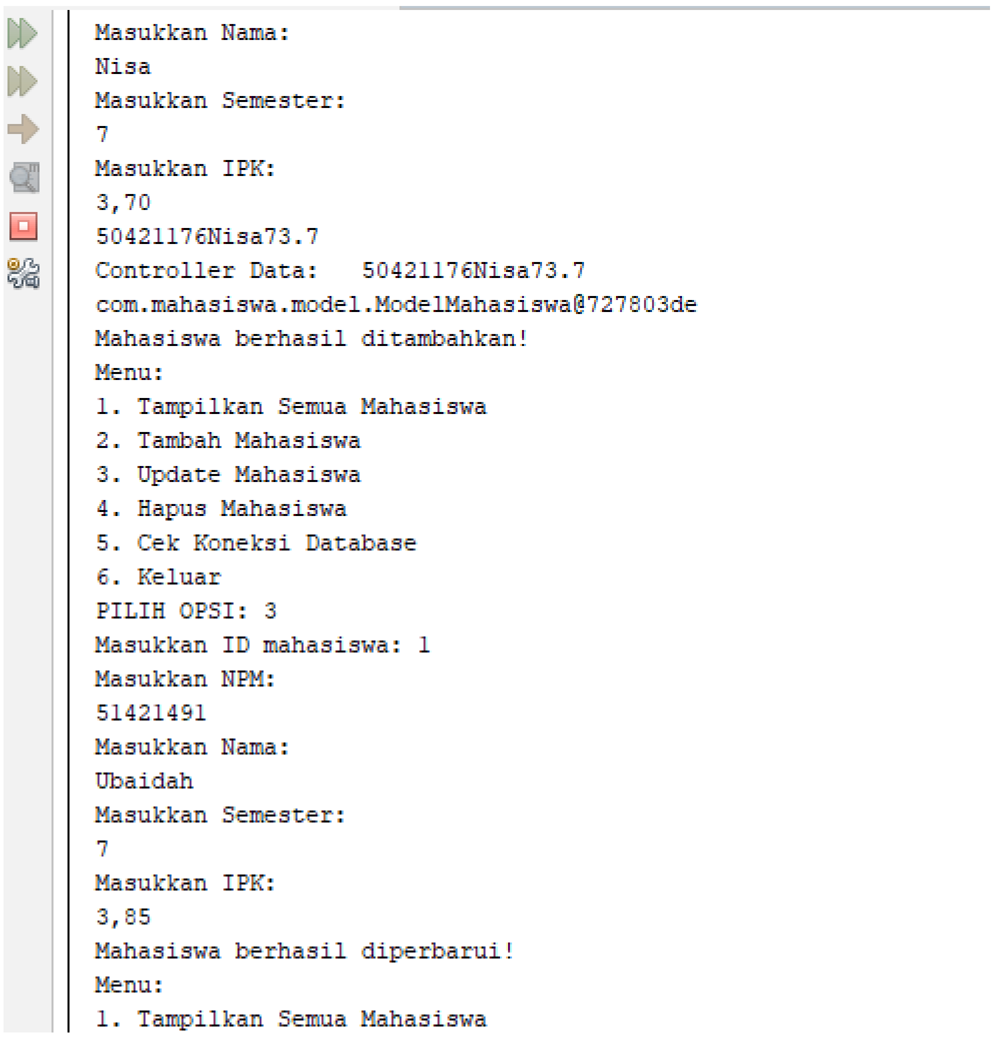
Output



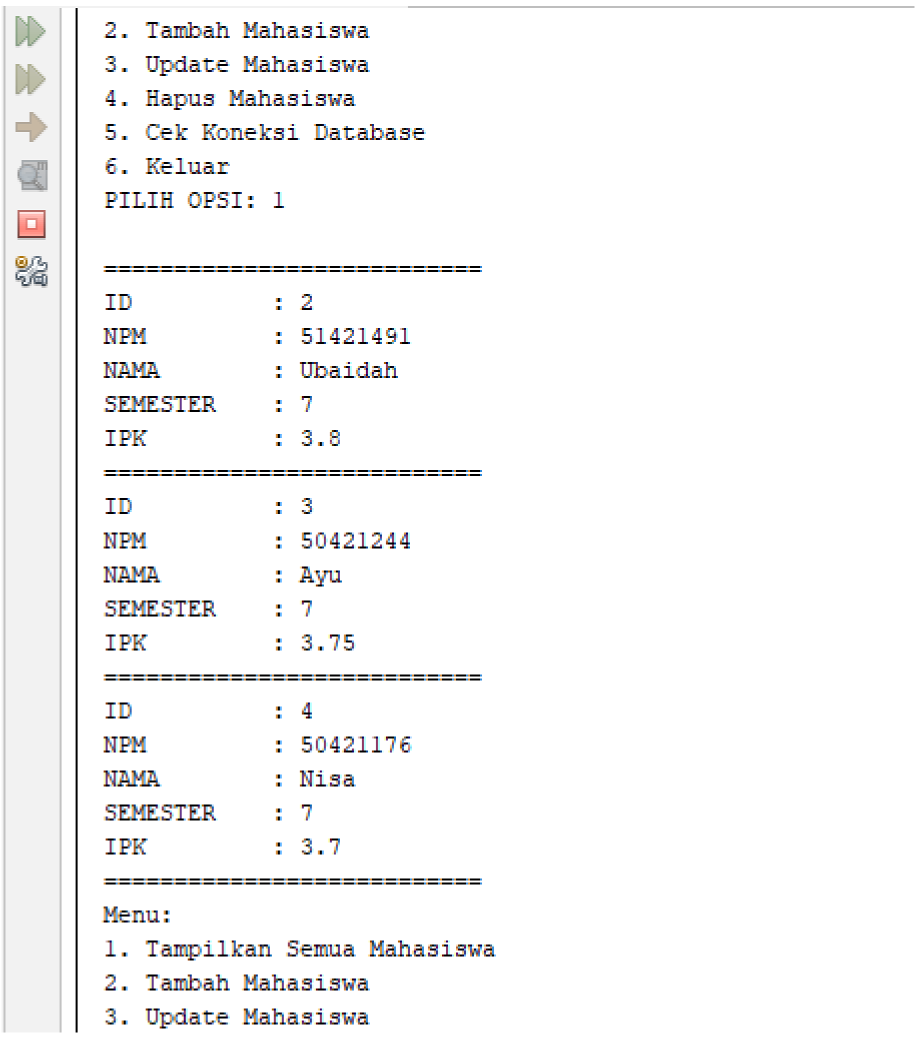
```
Output - Run (MahasiswaView)
--------------------------------[ jar ]---------------------------------
The artifact mysql:mysql-connector-java:jar:8.0.33 has been relocated to com.mysql:mysql-connector-j:jar:8.0.33: MySQL Connecto

--- resources:3.3.1:resources (default-resources) @ com.mycompany.ubaidah ---
skip non existing resourceDirectory C:\Users\ACER\Music\com.mycompany.ubaidah\src\main\resources

--- compiler:3.11.0:compile (default-compile) @ com.mycompany.ubaidah ---
Nothing to compile - all classes are up to date

--- exec:3.1.0:exec (default-cli) @ com.mycompany.ubaidah ---
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI: 2
Masukkan NPM:
51421491
Masukkan Nama:
Ubaidah
Masukkan Semester:
7
Masukkan IPK:
3,80
51421491Ubaidah73.8
Controller Data:   51421491Ubaidah73.8
com.mahasiswa.model.ModelMahasiswa@74a10858
Mahasiswa berhasil ditambahkan!
```

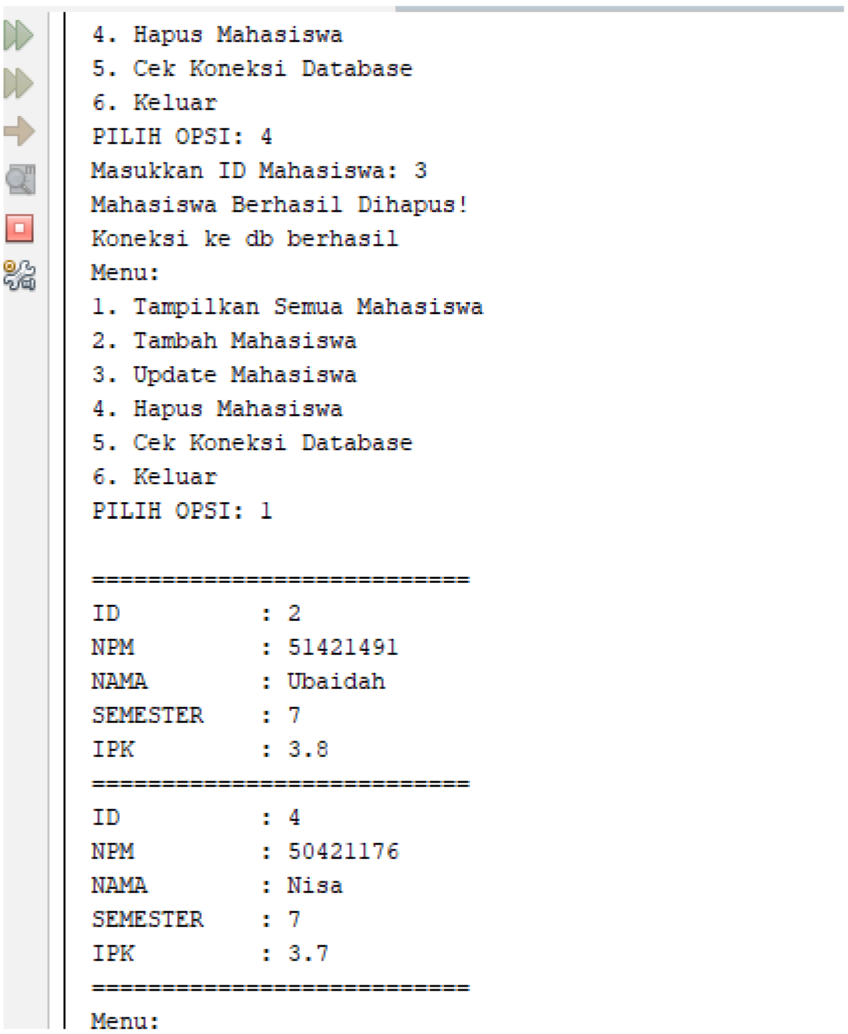


```
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI: 2
Masukkan NPM:
50421244
Masukkan Nama:
Ayu
Masukkan Semester:
7
Masukkan IPK:
3,75
50421244Ayu73.75
Controller Data:   50421244Ayu73.75
com.mahasiswa.model.ModelMahasiswa@2641e737
Mahasiswa berhasil ditambahkan!
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI: 2
Masukkan NPM:
50421176
```

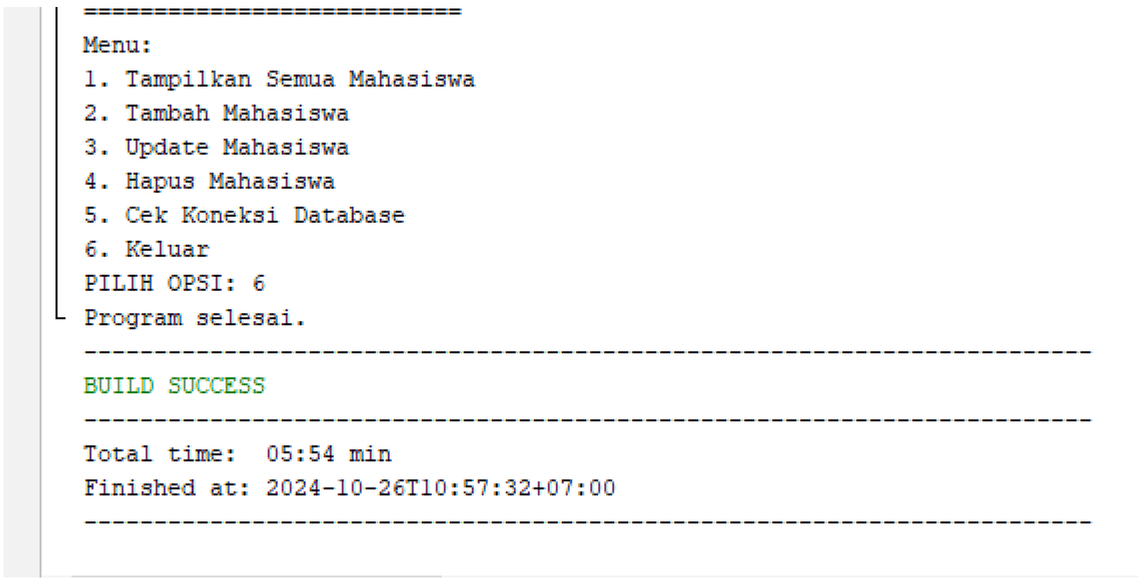


```
Masukkan Nama:
Nisa
Masukkan Semester:
7
Masukkan IPK:
3,70
50421176Nisa73.7
Controller Data:   50421176Nisa73.7
com.mahasiswa.model.ModelMahasiswa@727803de
Mahasiswa berhasil ditambahkan!
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI: 3
Masukkan ID mahasiswa: 1
Masukkan NPM:
51421491
Masukkan Nama:
Ubaidah
Masukkan Semester:
7
Masukkan IPK:
3,85
Mahasiswa berhasil diperbarui!
Menu:
1. Tampilkan Semua Mahasiswa
```


```
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI: 1

============================
ID          : 2
NPM         : 51421491
NAMA        : Ubaidah
SEMESTER    : 7
IPK         : 3.8
============================
ID          : 3
NPM         : 50421244
NAMA        : Ayu
SEMESTER    : 7
IPK         : 3.75
============================
ID          : 4
NPM         : 50421176
NAMA        : Nisa
SEMESTER    : 7
IPK         : 3.7
============================
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
```



```
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI: 4
Masukkan ID Mahasiswa: 3
Mahasiswa Berhasil Dihapus!
Koneksi ke db berhasil
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI: 1

============================
ID           : 2
NPM          : 51421491
NAMA         : Ubaidah
SEMESTER     : 7
IPK          : 3.8
============================
ID           : 4
NPM          : 50421176
NAMA         : Nisa
SEMESTER     : 7
IPK          : 3.7
============================
Menu:
```



```
============================
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI: 6
Program selesai.
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  05:54 min
Finished at: 2024-10-26T10:57:32+07:00
------------------------------------------------------------------------
```

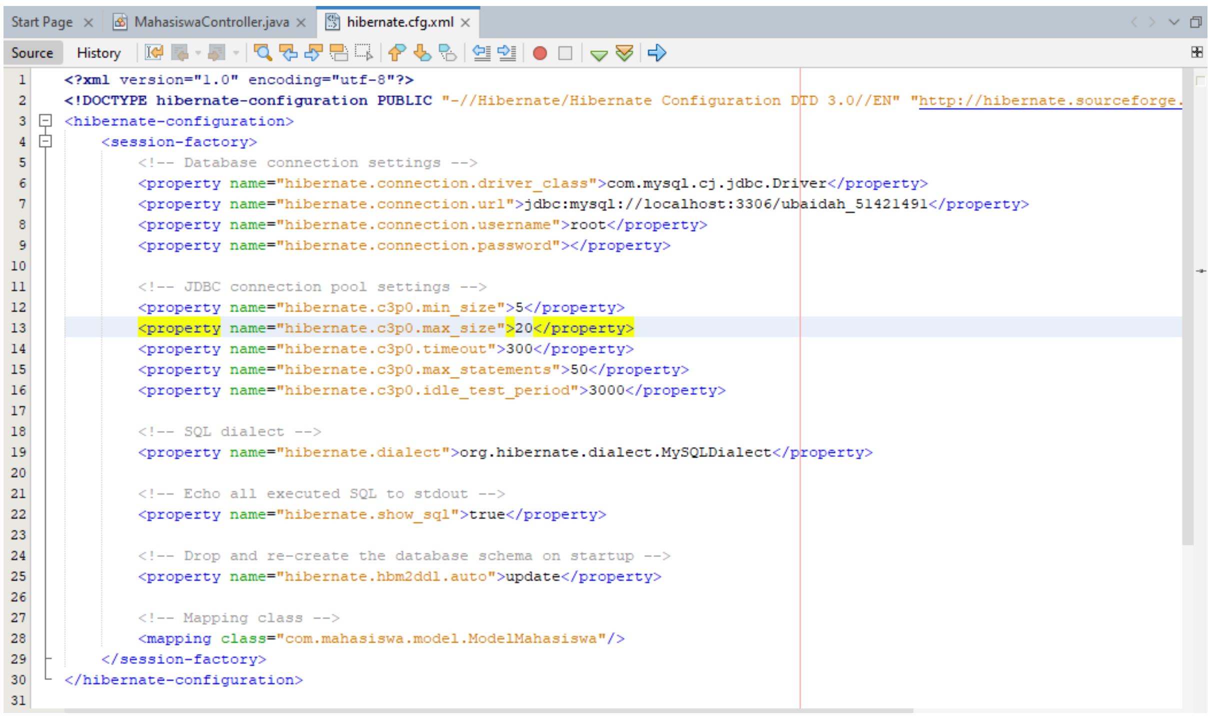
ACTIVITY PERTEMUAN 4

Nama : Ubaidah Luthfiyah Zain

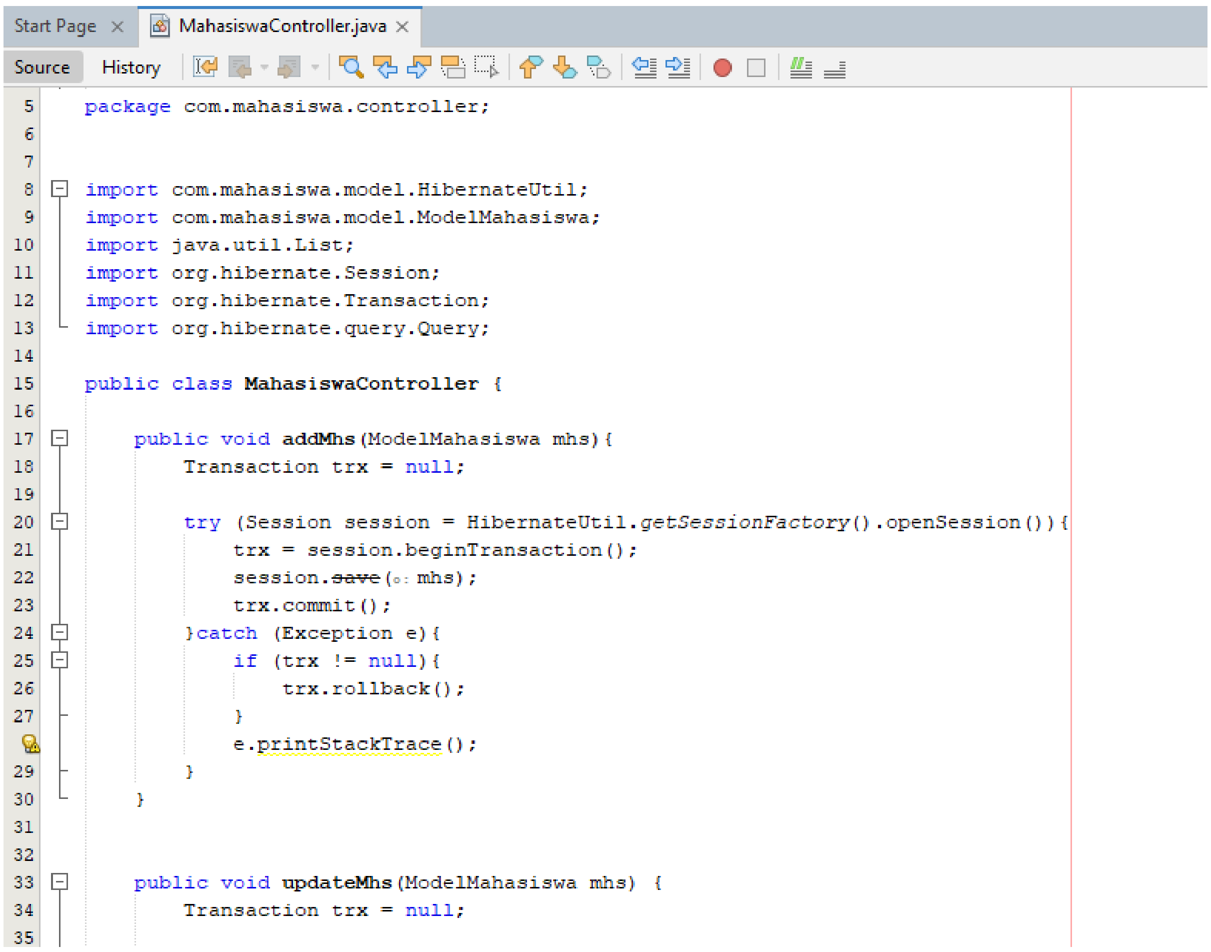
NPM : 51421491

Kelas : 4IA28

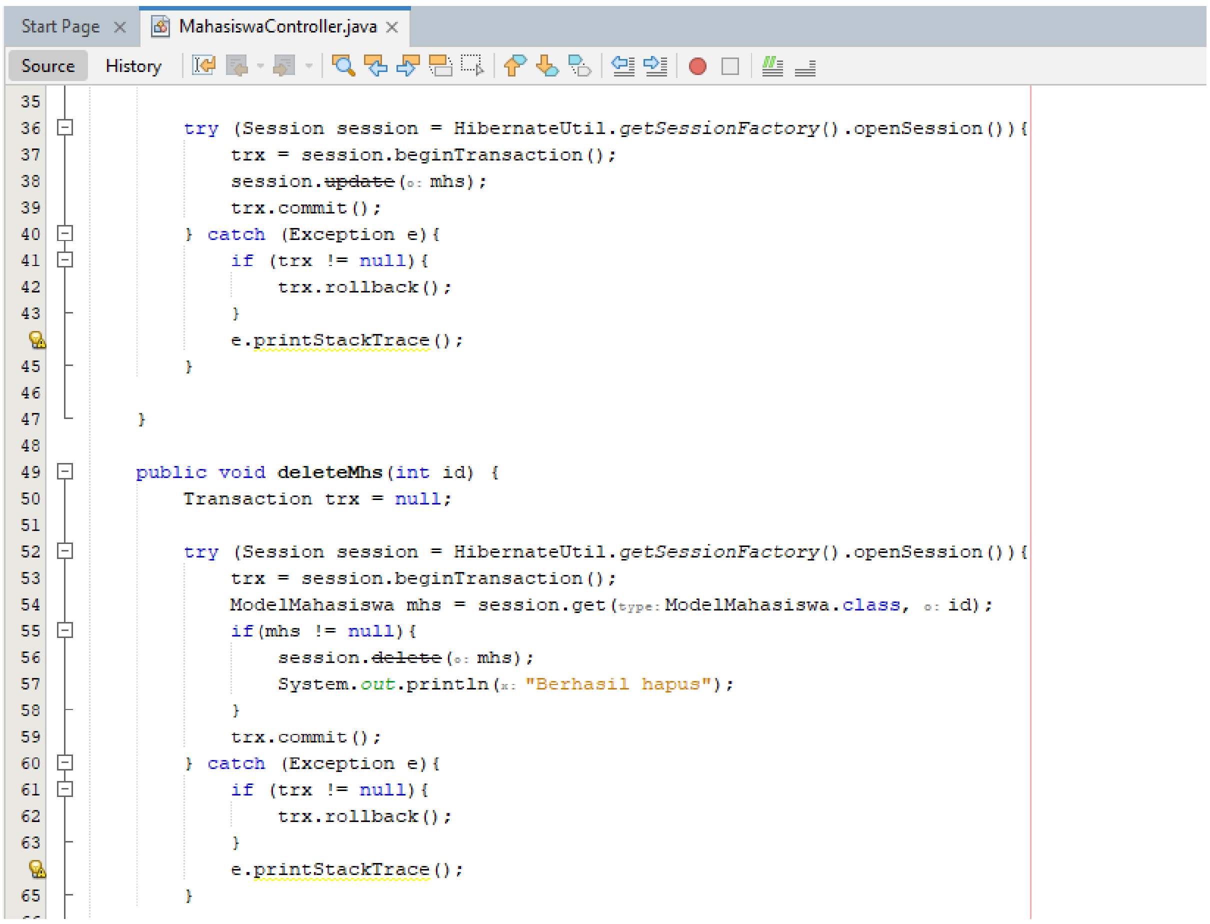
Source Code



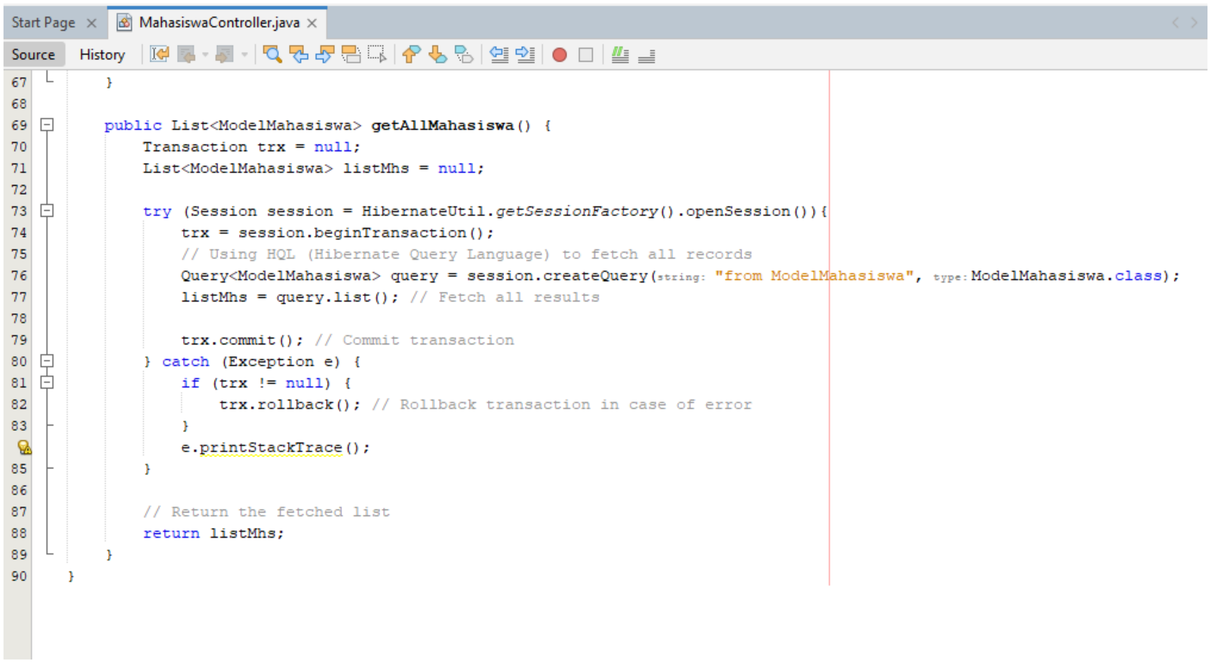
```xml
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN" "http://hibernate.sourceforge.
<hibernate-configuration>
    <session-factory>
        <!-- Database connection settings -->
        <property name="hibernate.connection.driver_class">com.mysql.cj.jdbc.Driver</property>
        <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/ubaidah_51421491</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>
        <property name="hibernate.c3p0.timeout">300</property>
        <property name="hibernate.c3p0.max_statements">50</property>
        <property name="hibernate.c3p0.idle_test_period">3000</property>

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

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

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

        <!-- Mapping class -->
        <mapping class="com.mahasiswa.model.ModelMahasiswa"/>
    </session-factory>
</hibernate-configuration>
```



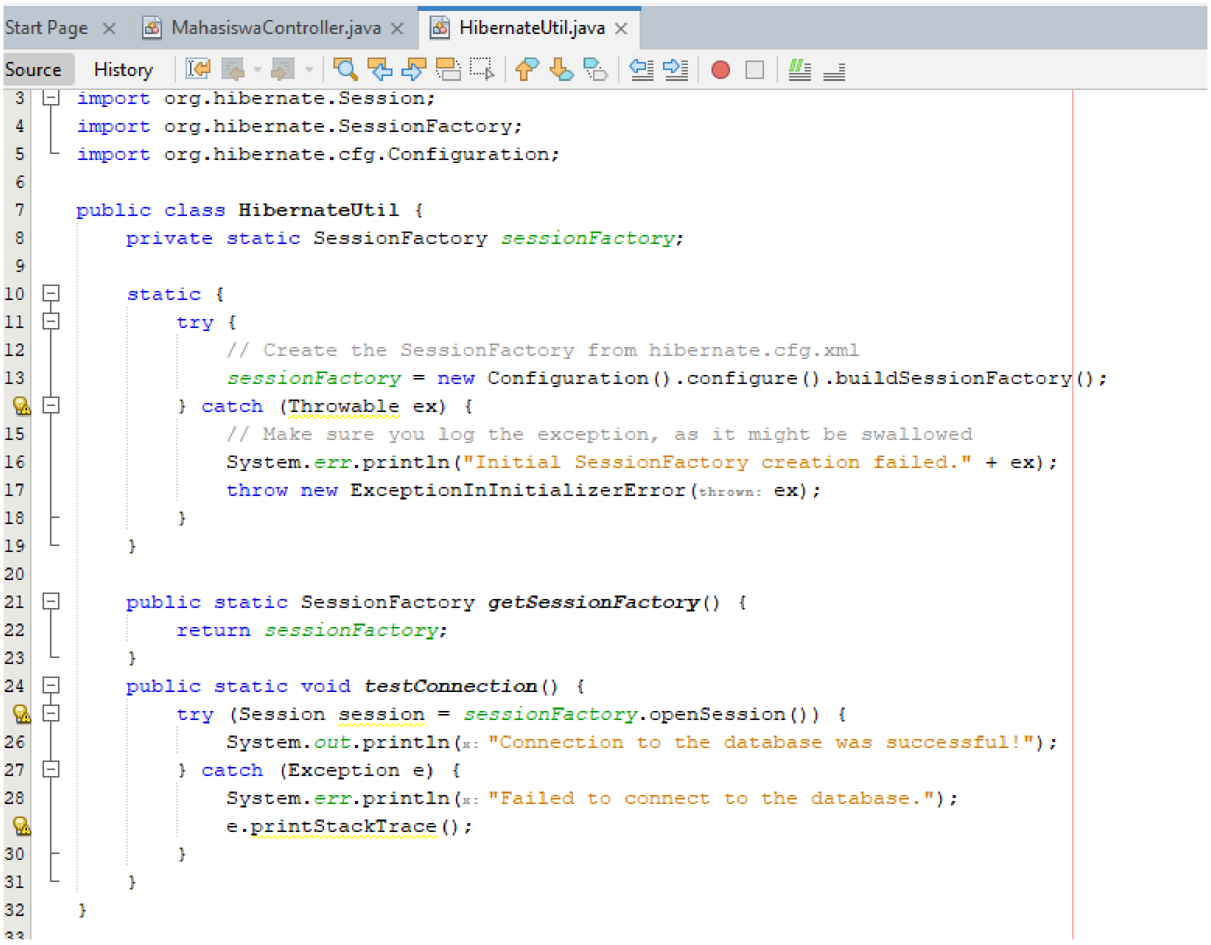
```java
package com.mahasiswa.controller;


import com.mahasiswa.model.HibernateUtil;
import com.mahasiswa.model.ModelMahasiswa;
import java.util.List;
import org.hibernate.Session;
import org.hibernate.Transaction;
import org.hibernate.query.Query;

public class MahasiswaController {

    public void addMhs(ModelMahasiswa mhs){
        Transaction trx = null;

        try (Session session = HibernateUtil.getSessionFactory().openSession()){
            trx = session.beginTransaction();
            session.~~save~~(o: mhs);
            trx.commit();
        }catch (Exception e){
            if (trx != null){
                trx.rollback();
            }
            e.printStackTrace();
        }
    }


    public void updateMhs(ModelMahasiswa mhs) {
        Transaction trx = null;

```



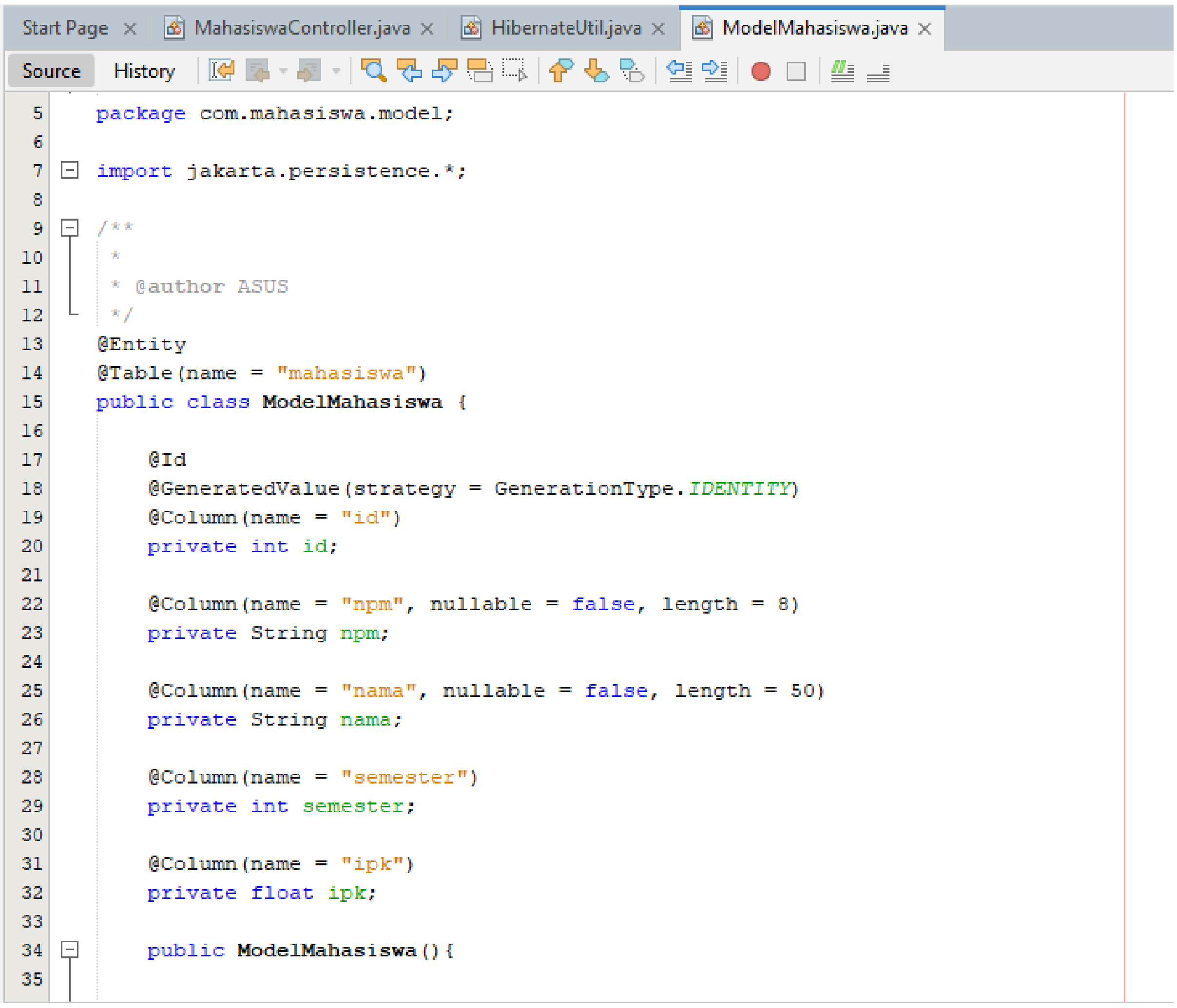
```java

        try (Session session = HibernateUtil.getSessionFactory().openSession()){
            trx = session.beginTransaction();
            session.~~update~~(o: mhs);
            trx.commit();
        } catch (Exception e){
            if (trx != null){
                trx.rollback();
            }
            e.printStackTrace();
        }

    }

    public void deleteMhs(int id) {
        Transaction trx = null;

        try (Session session = HibernateUtil.getSessionFactory().openSession()){
            trx = session.beginTransaction();
            ModelMahasiswa mhs = session.get(type: ModelMahasiswa.class, o: id);
            if(mhs != null){
                session.~~delete~~(o: mhs);
                System.out.println(x: "Berhasil hapus");
            }
            trx.commit();
        } catch (Exception e){
            if (trx != null){
                trx.rollback();
            }
            e.printStackTrace();
        }
```



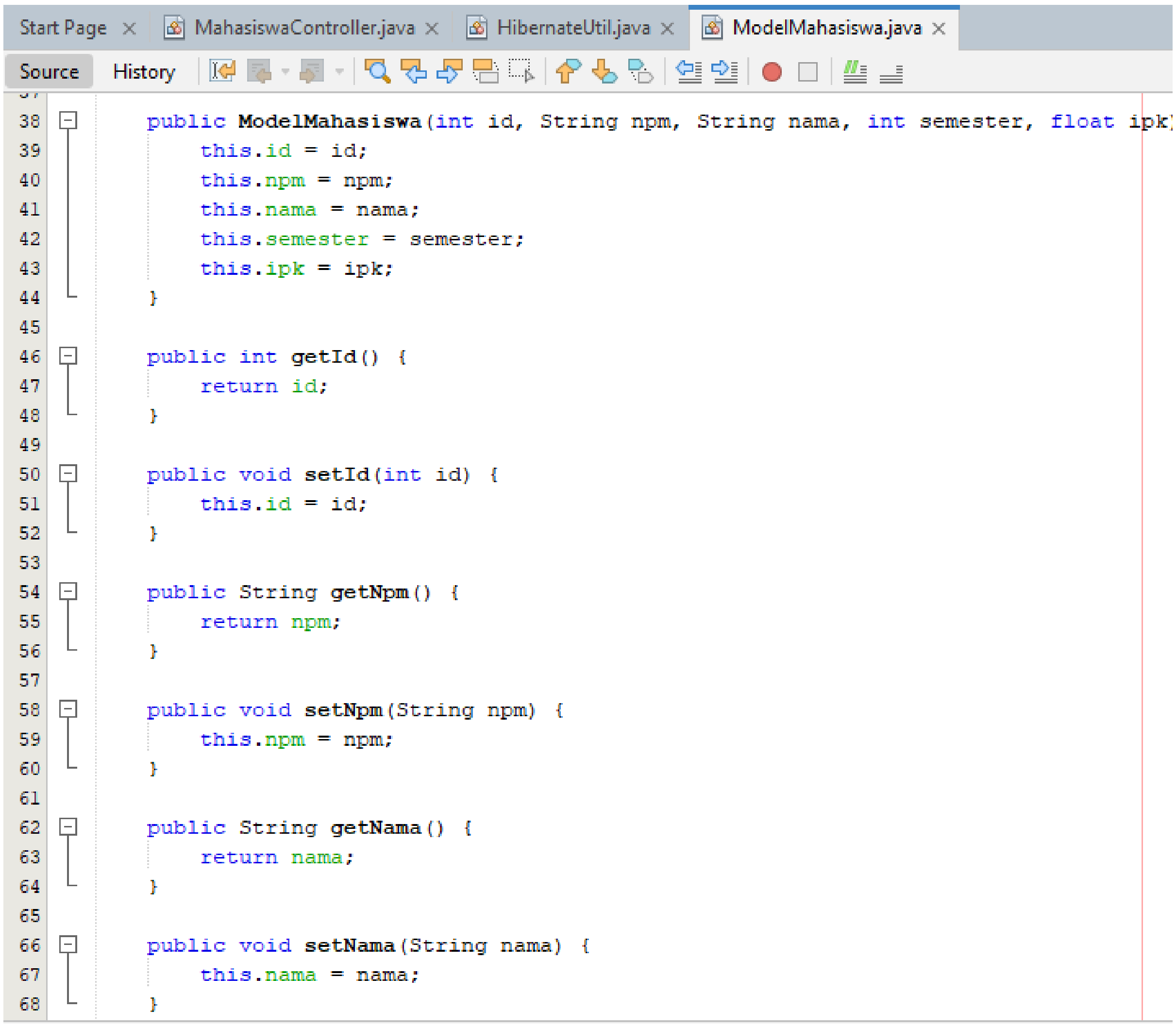
```java
    }

    public List<ModelMahasiswa> getAllMahasiswa() {
        Transaction trx = null;
        List<ModelMahasiswa> listMhs = null;

        try (Session session = HibernateUtil.getSessionFactory().openSession()){
            trx = session.beginTransaction();
            // Using HQL (Hibernate Query Language) to fetch all records
            Query<ModelMahasiswa> query = session.createQuery(string: "from ModelMahasiswa", type: ModelMahasiswa.class);
            listMhs = query.list(); // Fetch all results

            trx.commit(); // Commit transaction
        } catch (Exception e) {
            if (trx != null) {
                trx.rollback(); // Rollback transaction in case of error
            }
            e.printStackTrace();
        }

        // Return the fetched list
        return listMhs;
    }
}
```



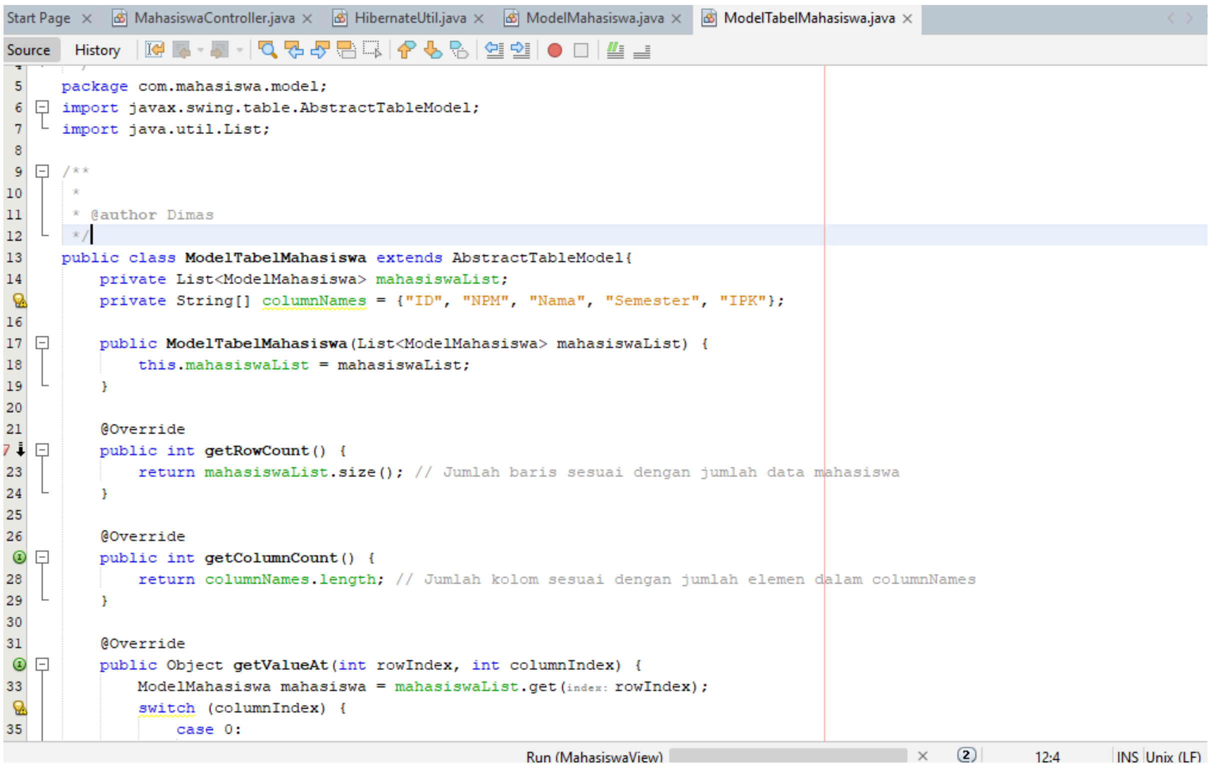
```java
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

public class HibernateUtil {
    private static SessionFactory sessionFactory;

    static {
        try {
            // Create the SessionFactory from hibernate.cfg.xml
            sessionFactory = new Configuration().configure().buildSessionFactory();
        } catch (Throwable ex) {
            // Make sure you log the exception, as it might be swallowed
            System.err.println("Initial SessionFactory creation failed." + ex);
            throw new ExceptionInInitializerError(thrown: ex);
        }
    }

    public static SessionFactory getSessionFactory() {
        return sessionFactory;
    }
    public static void testConnection() {
        try (Session session = sessionFactory.openSession()) {
            System.out.println(x: "Connection to the database was successful!");
        } catch (Exception e) {
            System.err.println(x: "Failed to connect to the database.");
            e.printStackTrace();
        }
    }
}
```



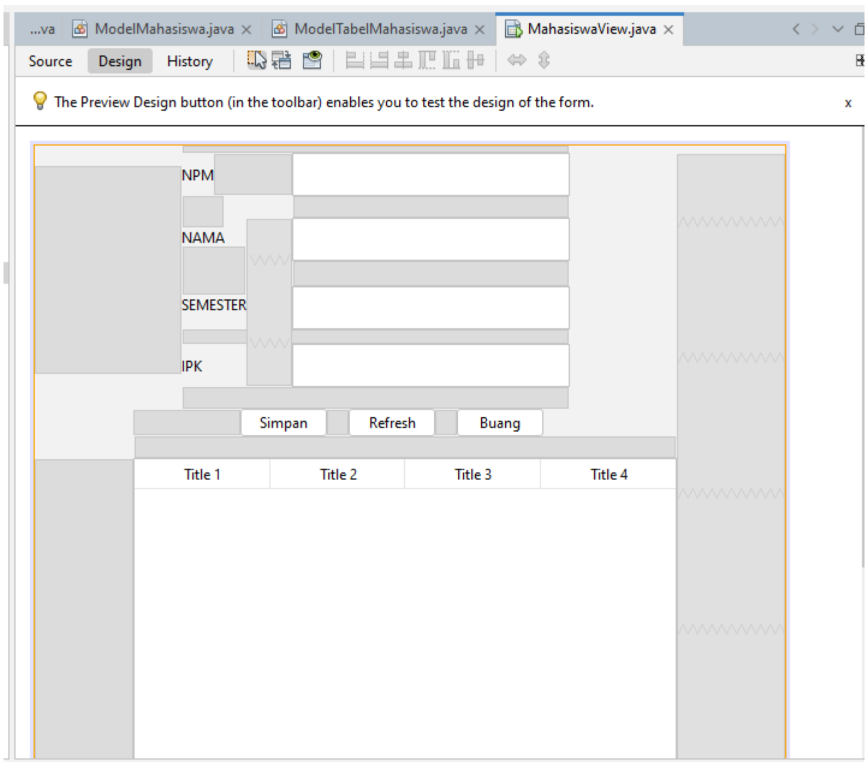
```java
package com.mahasiswa.model;

import jakarta.persistence.*;

/**
 *
 * @author ASUS
 */
@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;

    public ModelMahasiswa(){

```



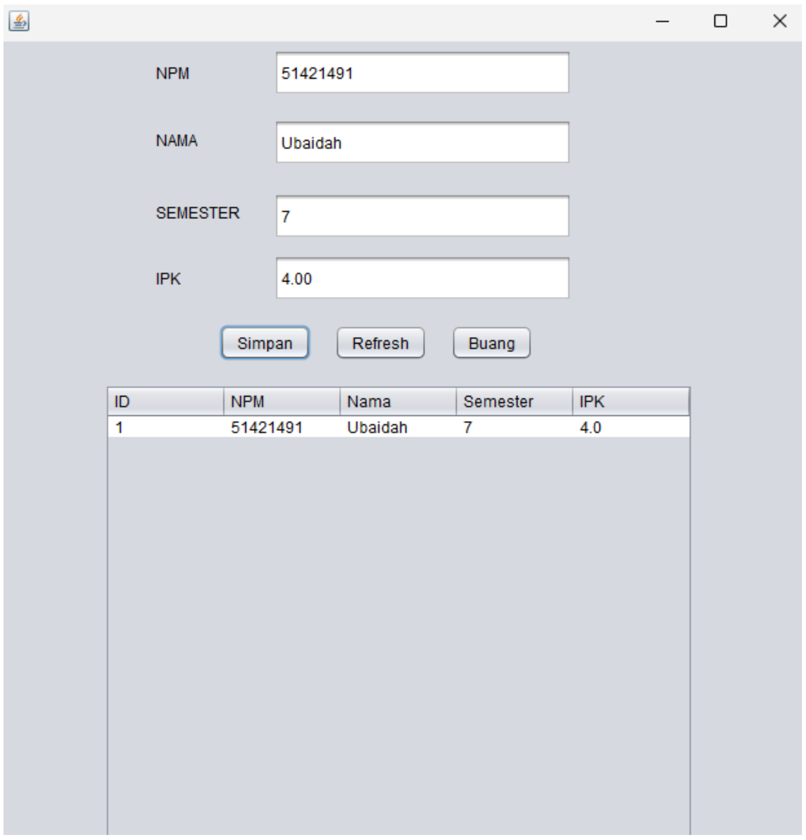
```java
    public ModelMahasiswa(int id, String npm, String nama, int semester, float ipk
        this.id = id;
        this.npm = npm;
        this.nama = nama;
        this.semester = semester;
        this.ipk = ipk;
    }

    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;
    }
```



```java
package com.mahasiswa.model;
import javax.swing.table.AbstractTableModel;
import java.util.List;

/**
 *
 * @author Dimas
 */
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(index: rowIndex);
        switch (columnIndex) {
            case 0:
```

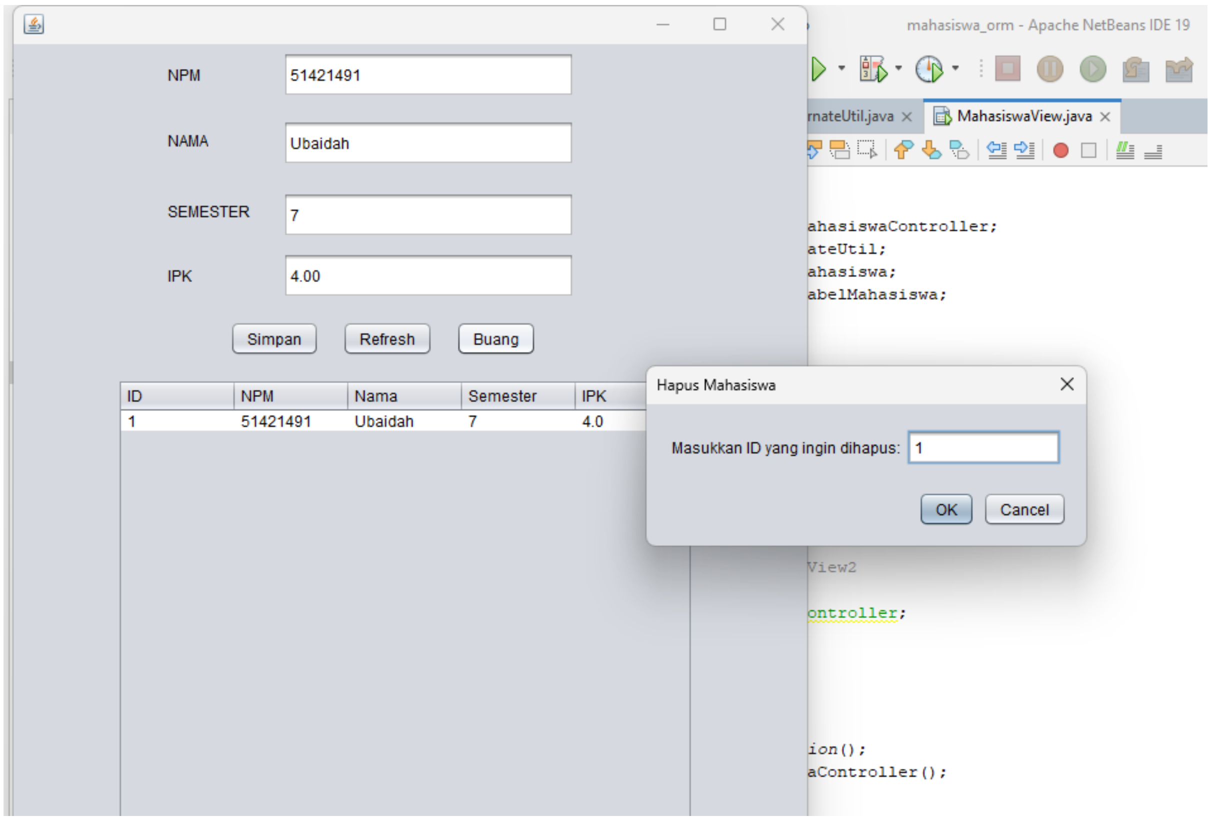


Output



NPM: 51421491
NAMA: Ubaidah
SEMESTER: 7
IPK: 4.00

ID	NPM	Nama	Semester	IPK
1	51421491	Ubaidah	7	4.0



mahasiswa_orm - Apache NetBeans IDE 19

rnateUtil.java MahasiswaView.java

NPM: 51421491

NAMA: Ubaidah

SEMESTER: 7

IPK: 4.00

Simpan Refresh Buang

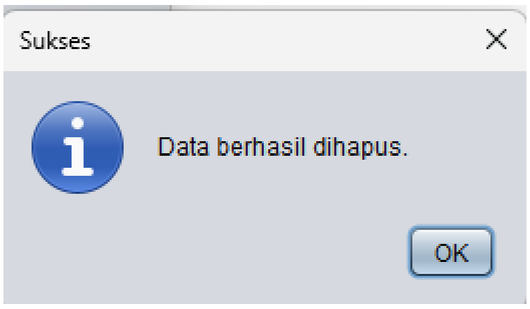
ID	NPM	Nama	Semester	IPK
1	51421491	Ubaidah	7	4.0

Hapus Mahasiswa

Masukkan ID yang ingin dihapus: 1

OK Cancel

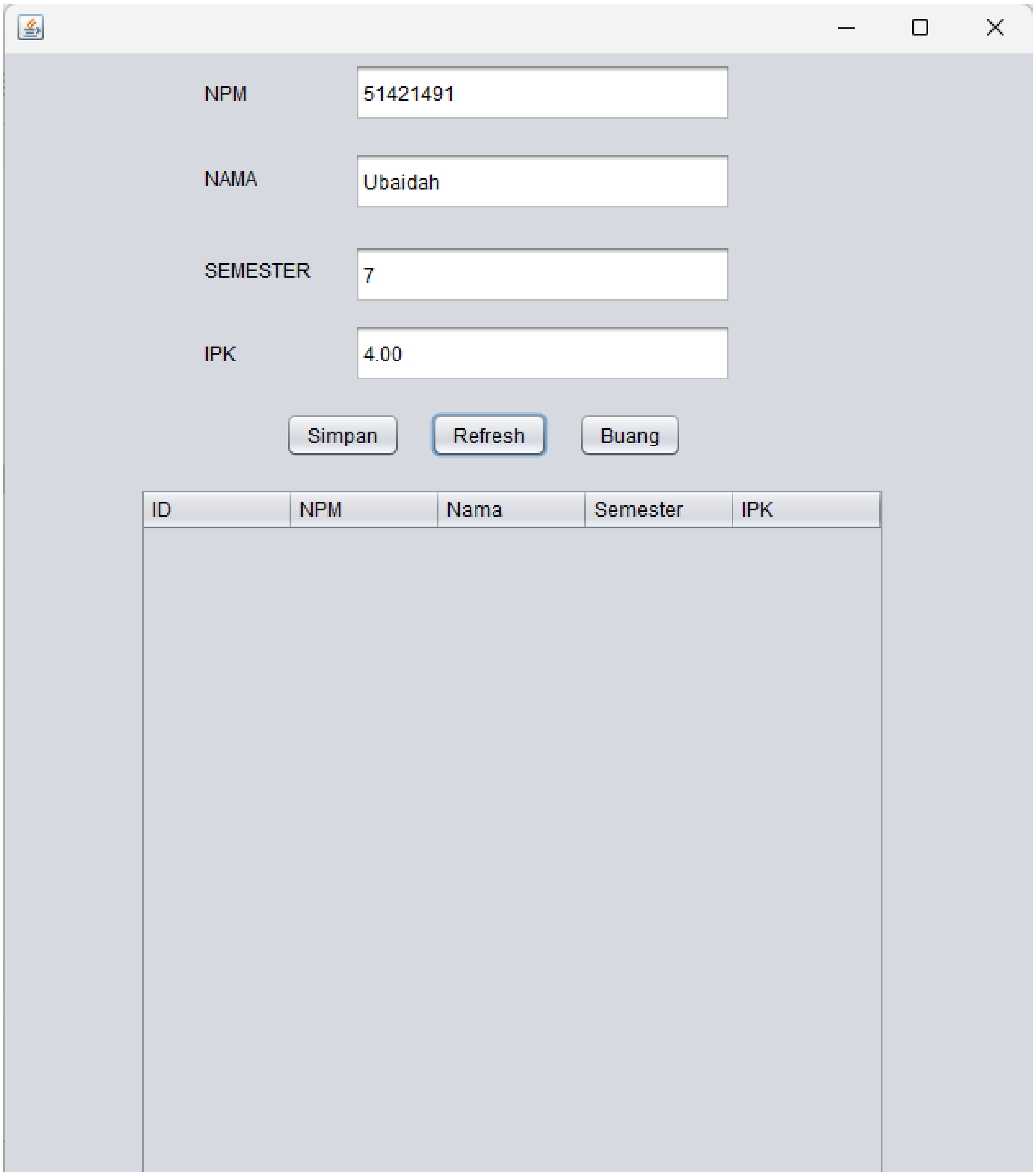
```
ahasiswaController;
ateUtil;
ahasiswa;
abelMahasiswa;
View2
ontroller;
ion();
aController();
```



Sukses

Data berhasil dihapus.

OK



NPM 51421491

NAMA Ubaidah

SEMESTER 7

IPK 4.00

ID	NPM	Nama	Semester	IPK