ACTIVITY PERTEMUAN 3 & 4

Nama : Maulana Hidayatulloh Mujanah

NPM : 50421797

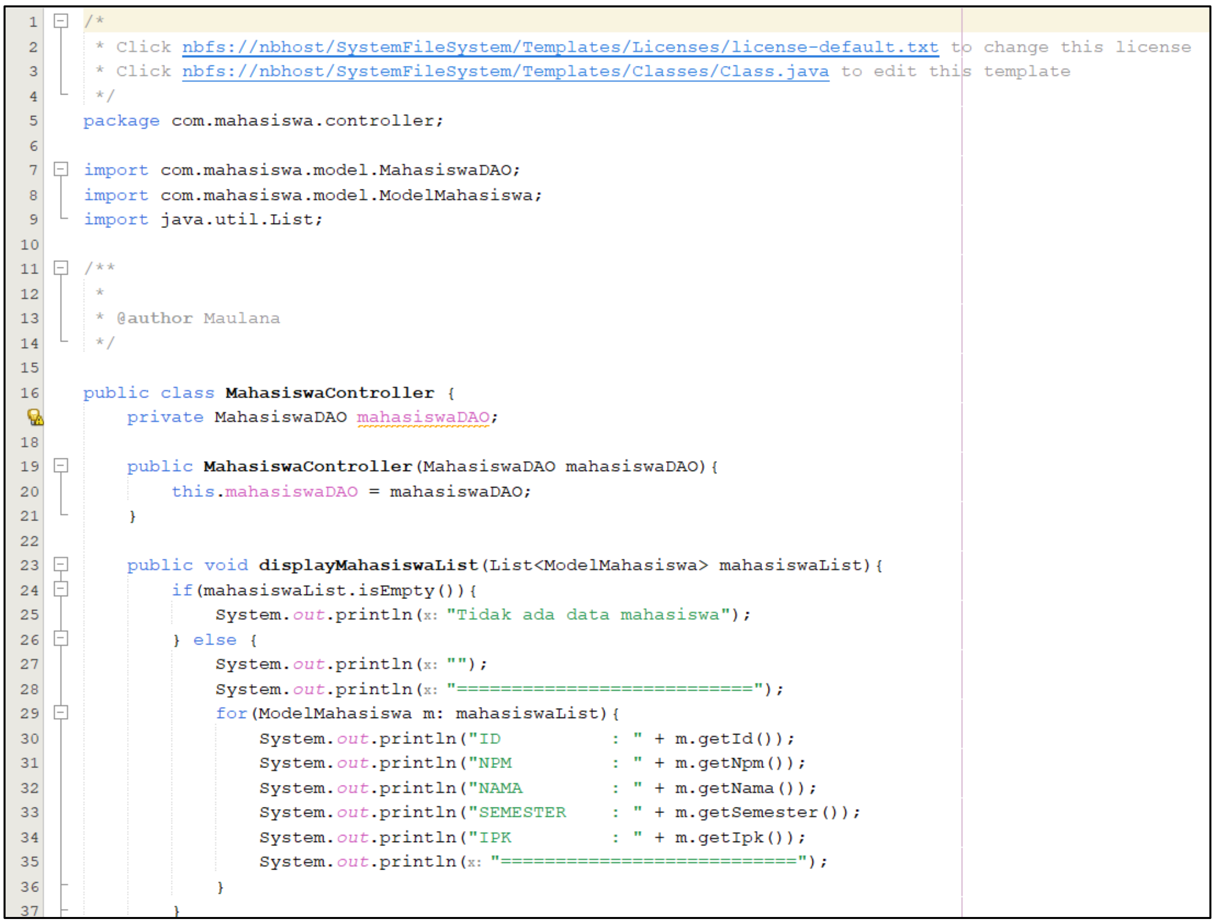
Kelas : 4IA28

Praktikum RPL 2

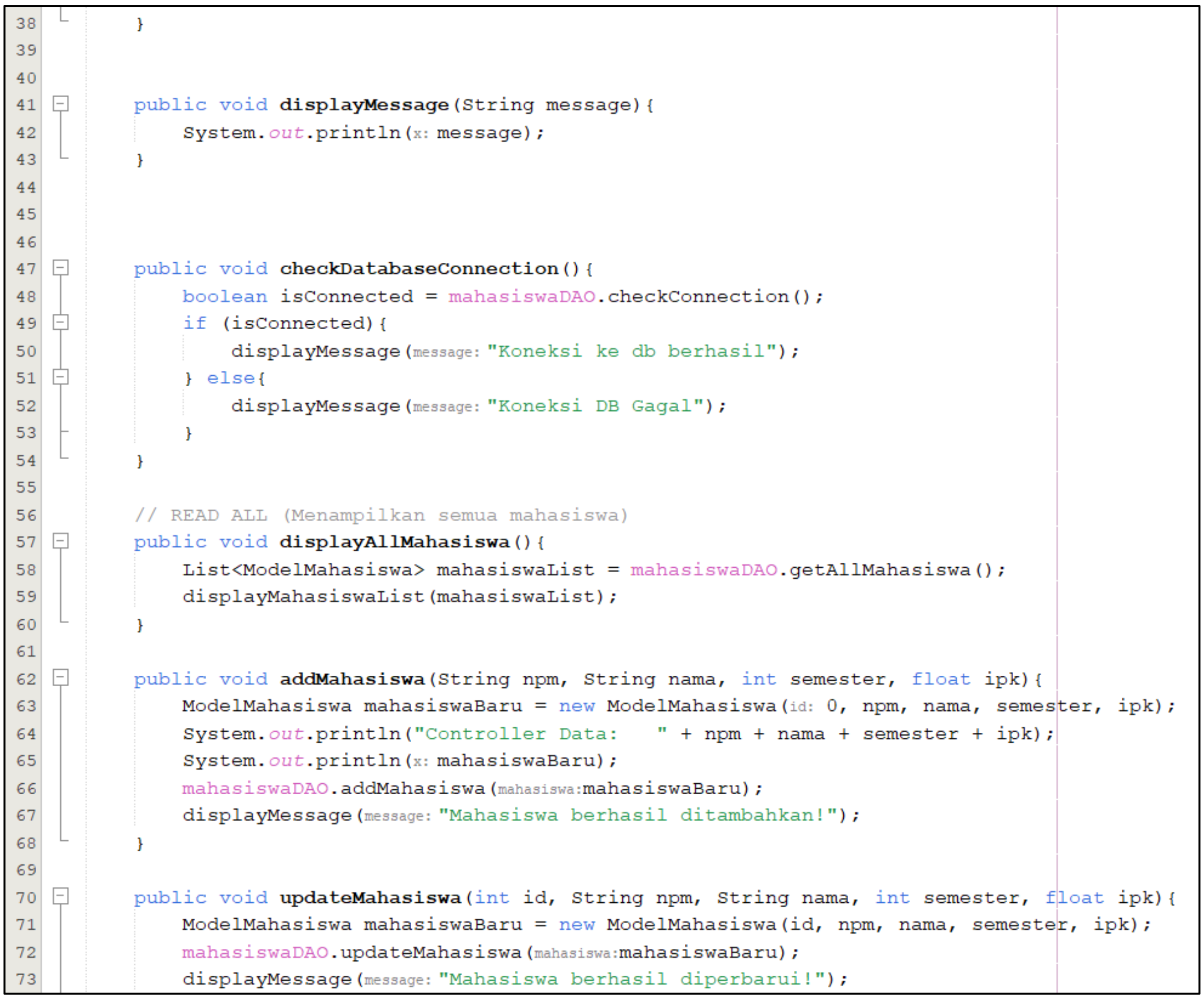
Pertemuan 3

Code :

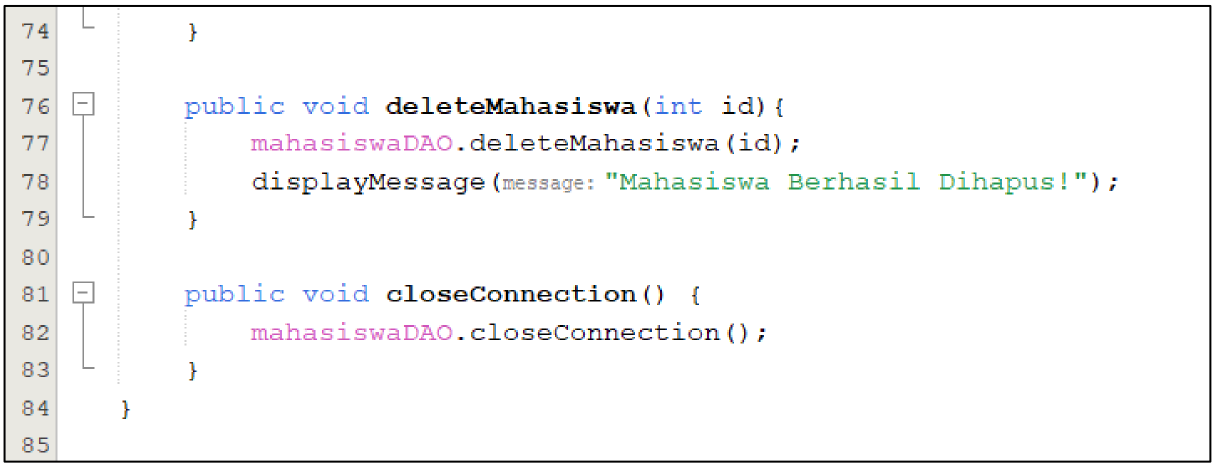
MahasiswaController.java



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

import com.mahasiswa.model.MahasiswaDAO;
import com.mahasiswa.model.ModelMahasiswa;
import java.util.List;

/**
 *
 * @author Maulana
 */

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

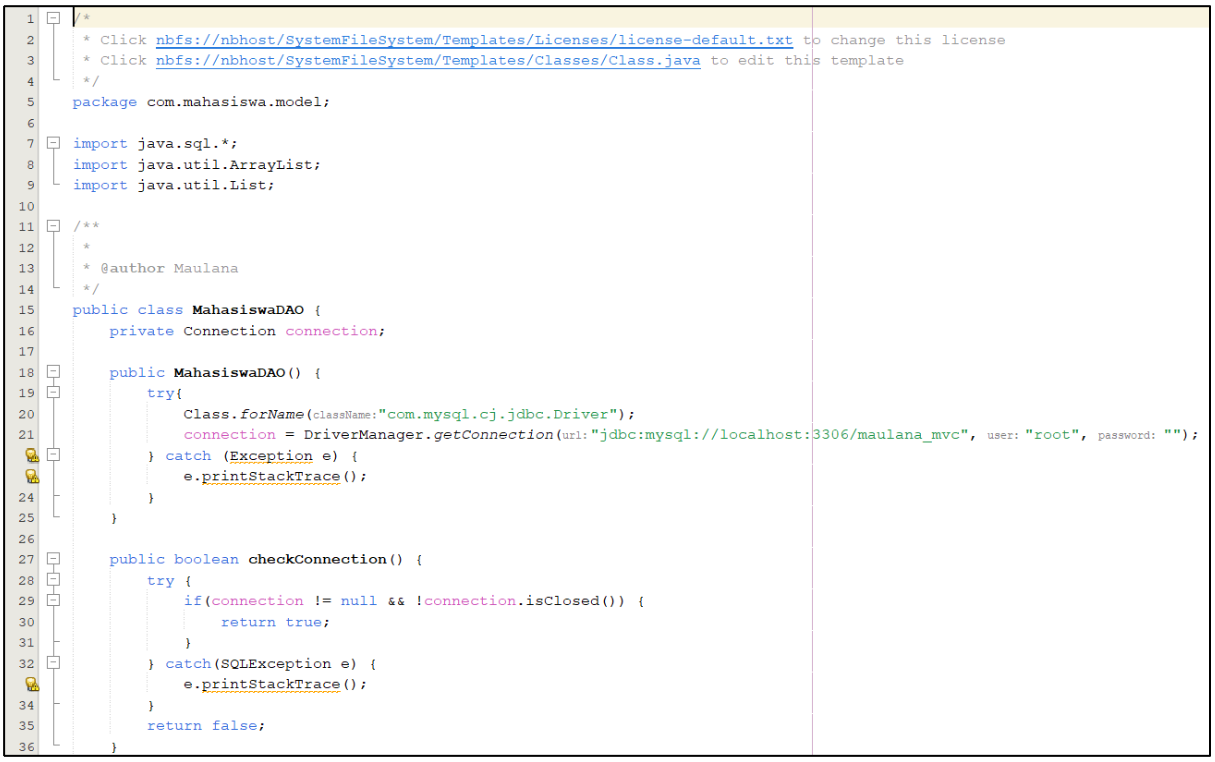


```java
    }


    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);
        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!");
```

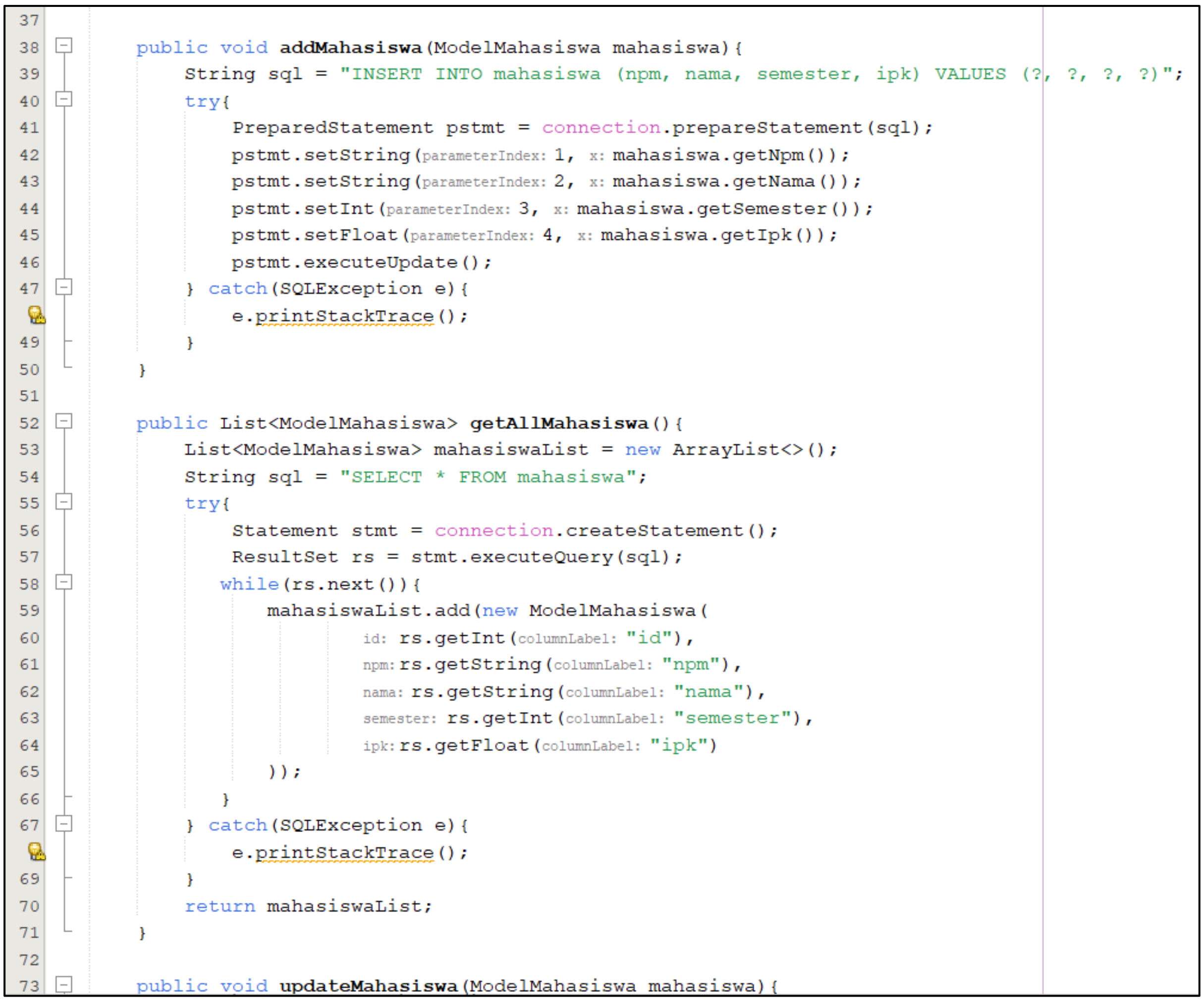


```java
    }

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

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

```

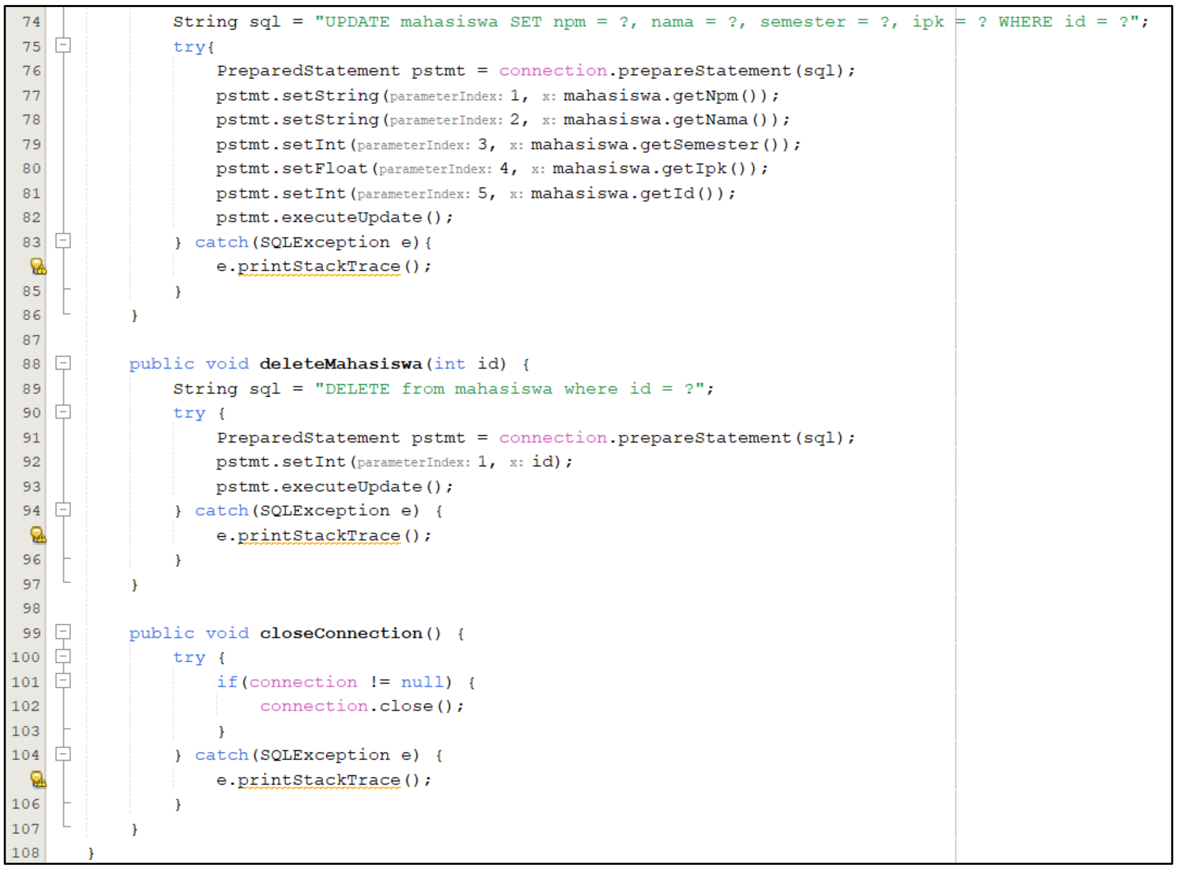
MahasiswaDAO.java



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

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

/**
 *
 * @author Maulana
 */
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/maulana_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();
        }
        return false;
    }
```

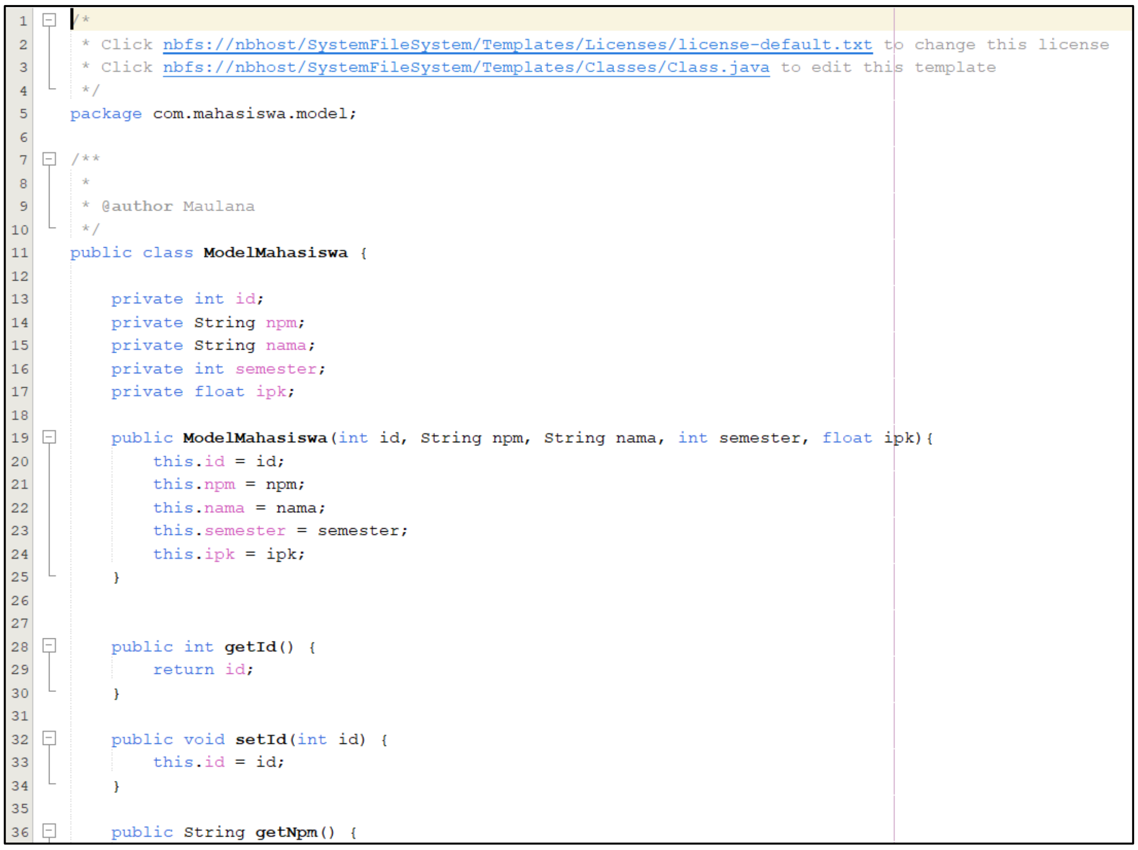


```java

    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){
```

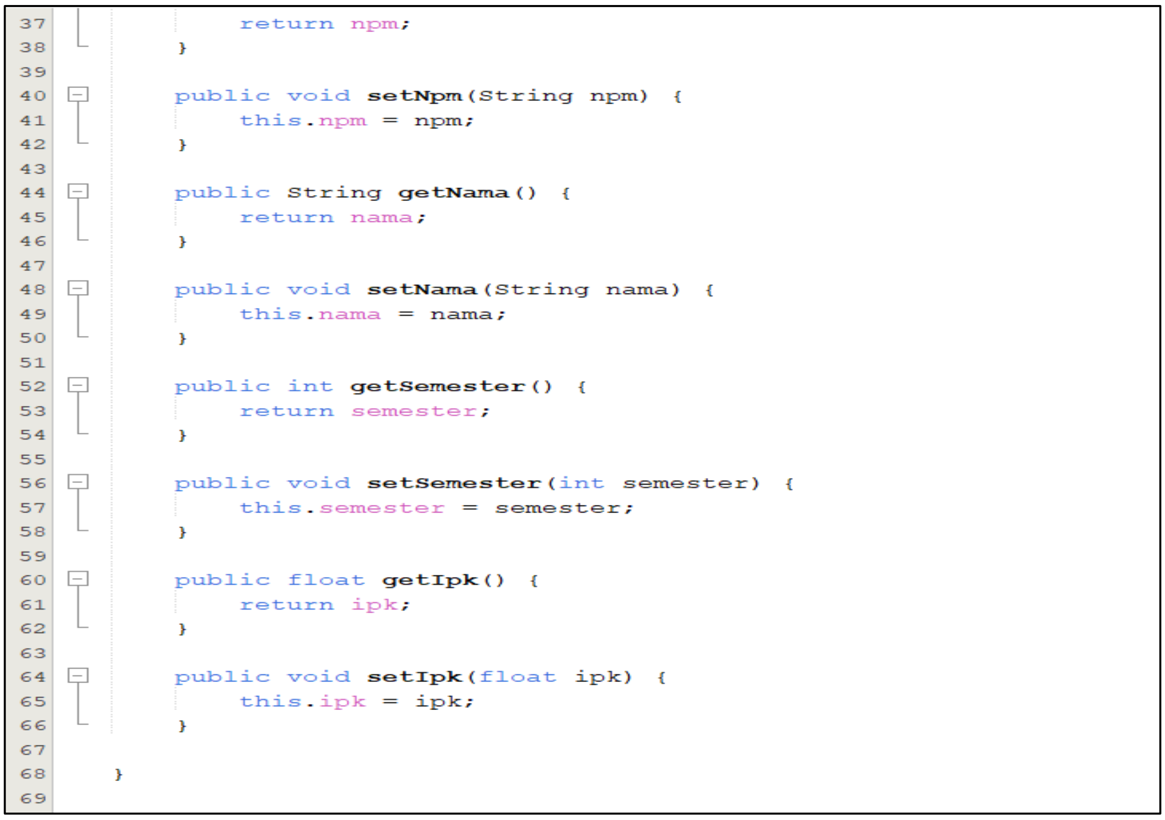


```java
        String sql = "UPDATE mahasiswa SET npm = ?, nama = ?, semester = ?, ipk = ? WHERE id = ?";
        try{
            PreparedStatement pstmt = connection.prepareStatement(sql);
            pstmt.setString(parameterIndex: 1, x: mahasiswa.getNpm());
            pstmt.setString(parameterIndex: 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();
        }
    }
}
```

ModelMahasiswa.java

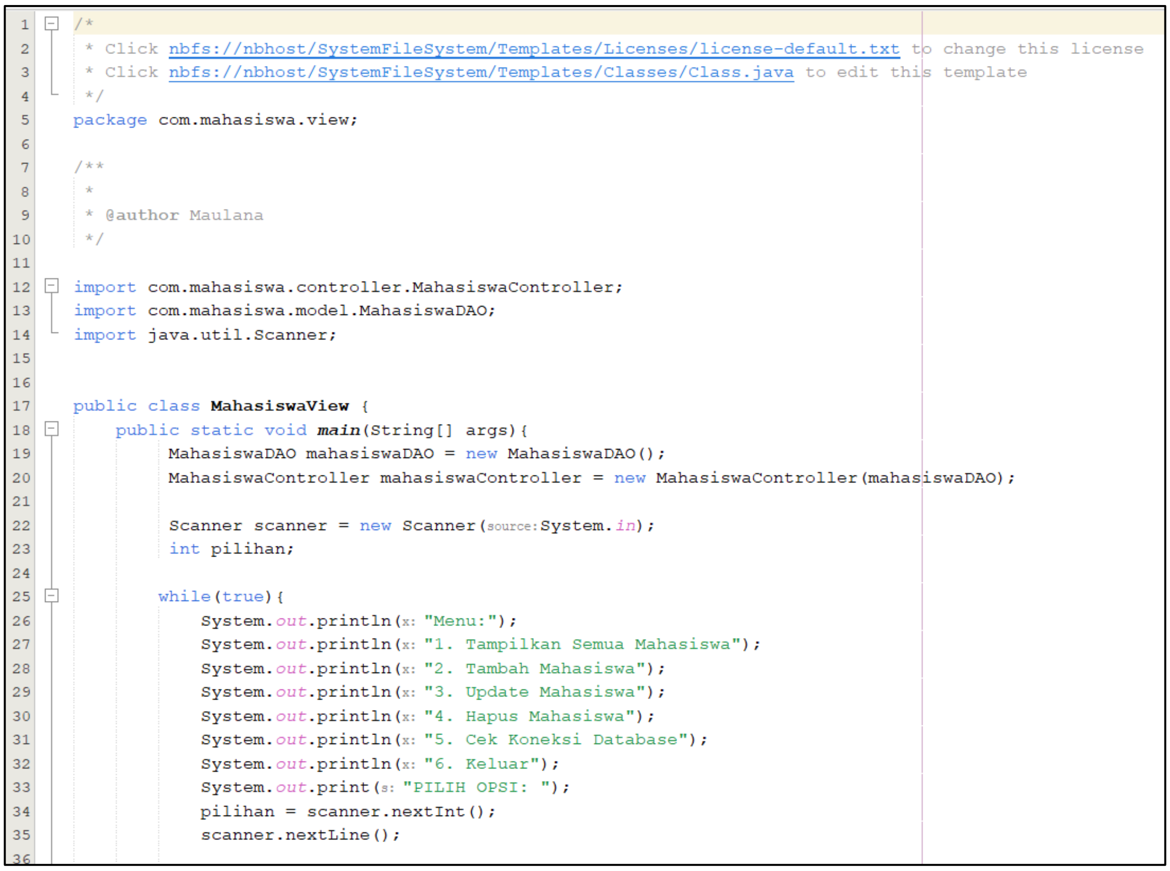


```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.mahasiswa.model;

/**
 *
 * @author Maulana
 */
public class ModelMahasiswa {

    private int id;
    private String npm;
    private String nama;
    private int semester;
    private float ipk;

    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() {
```

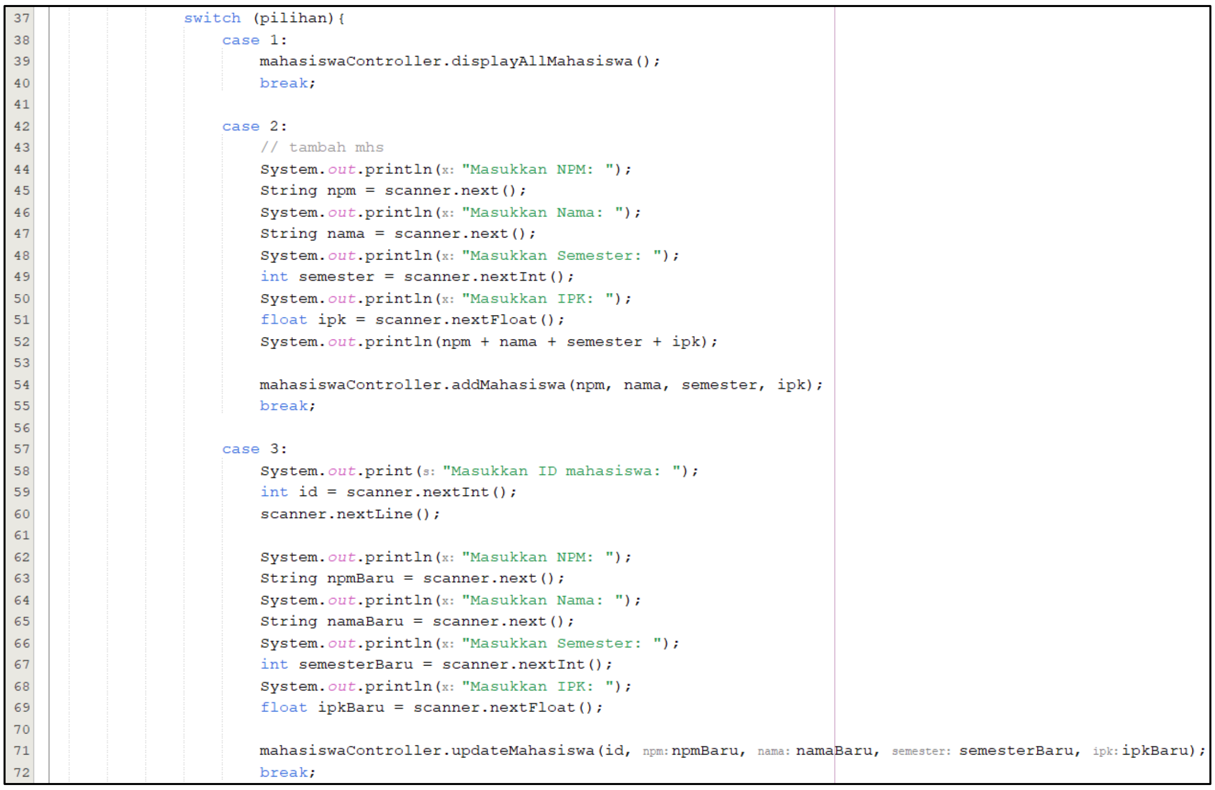


```java
        return npm;
    }

    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;
    }

}

```

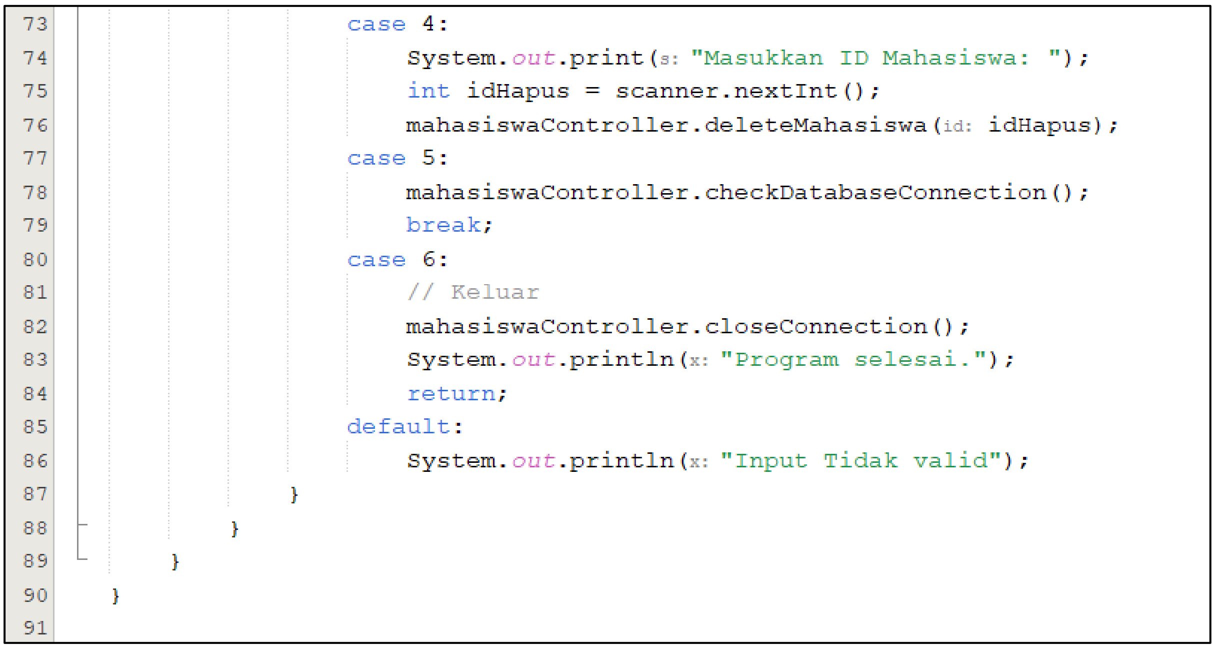
MahasiswaView.java



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

/**
 *
 * @author Maulana
 */

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();
            scanner.nextLine();
```

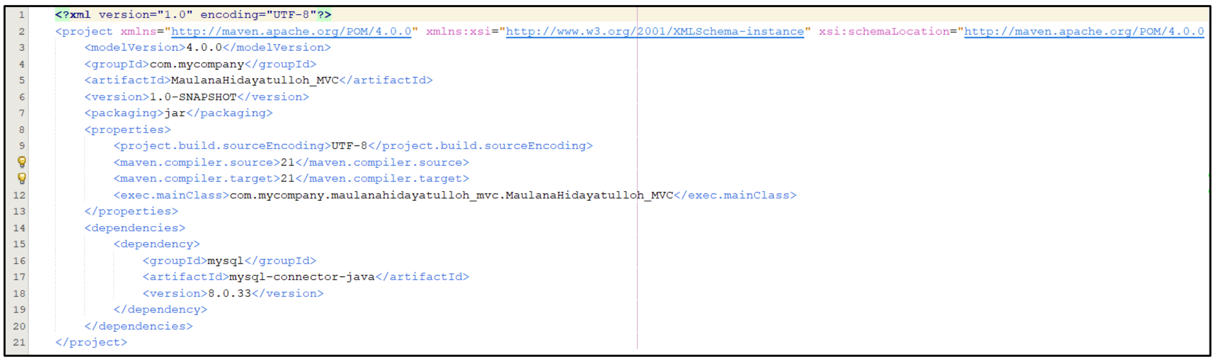


```java
        switch (pilihan){
            case 1:
                mahasiswaController.displayAllMahasiswa();
                break;

            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: ");
                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;
```



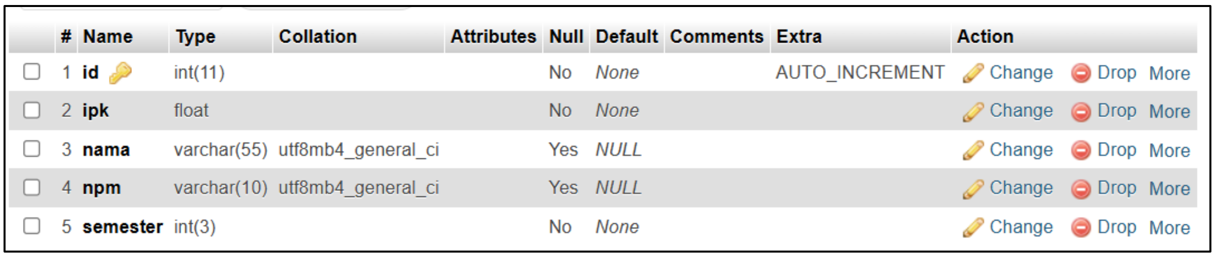
```java
            case 4:
                System.out.print(s: "Masukkan ID Mahasiswa: ");
                int idHapus = scanner.nextInt();
                mahasiswaController.deleteMahasiswa(id: idHapus);
            case 5:
                mahasiswaController.checkDatabaseConnection();
                break;
            case 6:
                // Keluar
                mahasiswaController.closeConnection();
                System.out.println(x: "Program selesai.");
                return;
            default:
                System.out.println(x: "Input Tidak valid");
        }
    }
  }
}
```

pom.xml



```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
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.mycompany</groupId>
    <artifactId>MaulanaHidayatulloh_MVC</artifactId>
    <version>1.0-SNAPSHOT</version>
    <packaging>jar</packaging>
    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <maven.compiler.source>21</maven.compiler.source>
        <maven.compiler.target>21</maven.compiler.target>
        <exec.mainClass>com.mycompany.maulanahidayatulloh_mvc.MaulanaHidayatulloh_MVC</exec.mainClass>
    </properties>
    <dependencies>
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>8.0.33</version>
        </dependency>
    </dependencies>
</project>
```

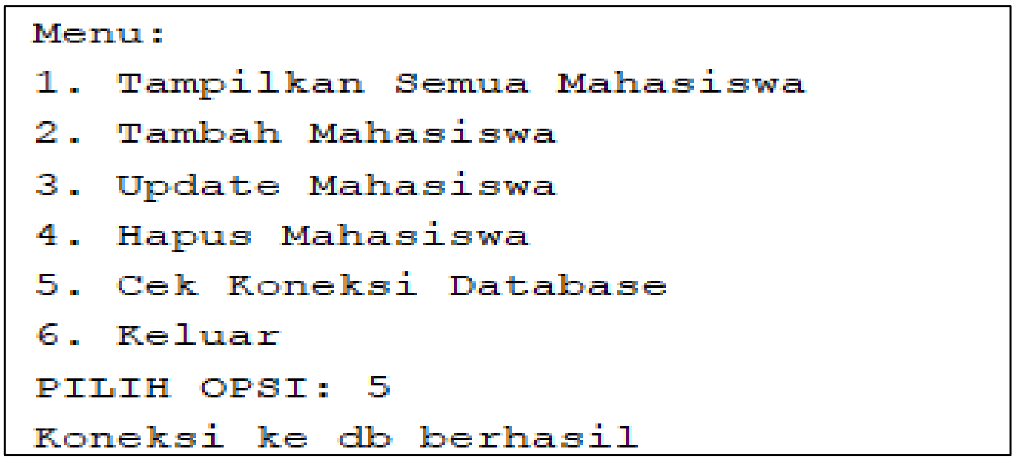
Database maulana_mvc table mahasiswa



	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
☐	1	**id**	int(11)			No	*None*		AUTO_INCREMENT	Change Drop More
☐	2	**ipk**	float			No	*None*			Change Drop More
☐	3	**nama**	varchar(55)	utf8mb4_general_ci		Yes	*NULL*			Change Drop More
☐	4	**npm**	varchar(10)	utf8mb4_general_ci		Yes	*NULL*			Change Drop More
☐	5	**semester**	int(3)			No	*None*			Change Drop More

Output :

Cek Koneksi Database



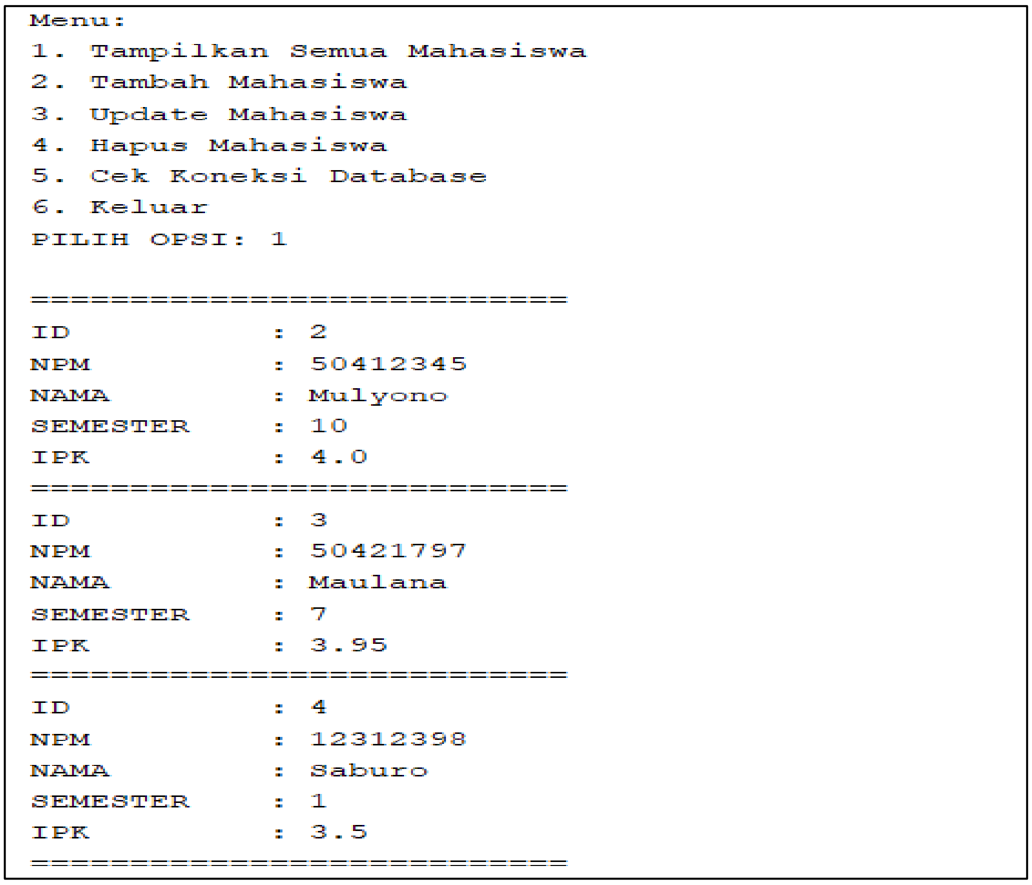
```
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI: 5
Koneksi ke db berhasil
```

Tambah Data Mahasiswa (3 Data)

```
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI: 5
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: 2
Masukkan NPM:
50412345
Masukkan Nama:
Mulyono
Masukkan Semester:
10
Masukkan IPK:
4.0
50412345Mulyono104.0
Controller Data:    50412345Mulyono104.0
com.mahasiswa.model.ModelMahasiswa@1990a65e
Mahasiswa berhasil ditambahkan!
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
```

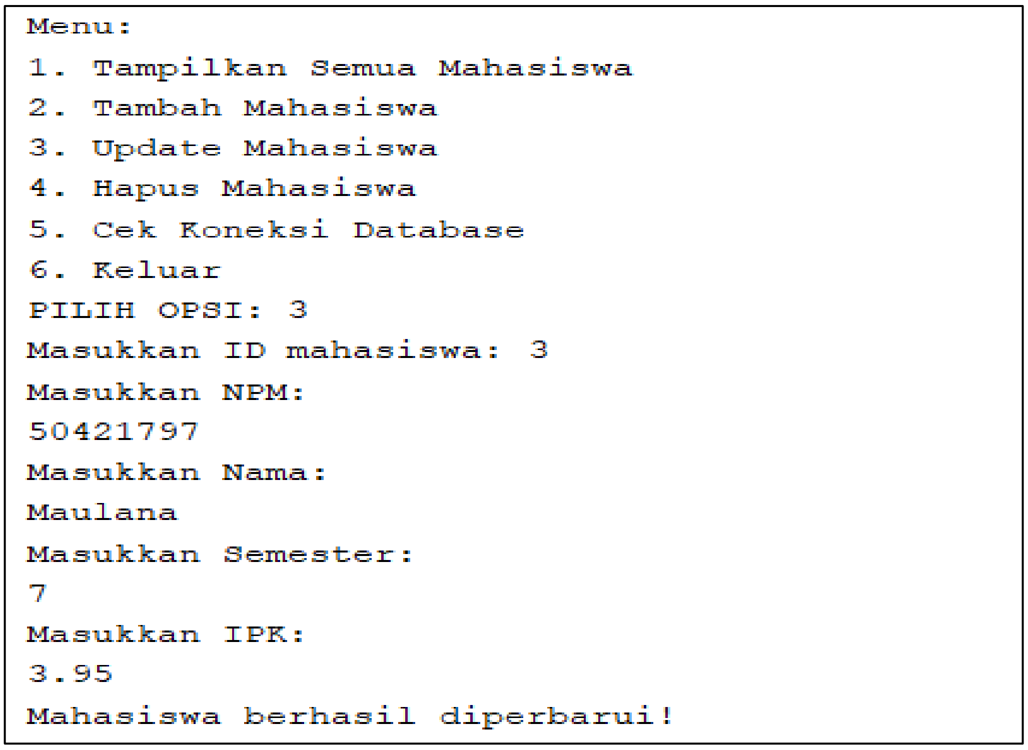
```
PILIH OPSI: 2
Masukkan NPM:
12345678
Masukkan Nama:
Ulyad
Masukkan Semester:
8
Masukkan IPK:
3.9
12345678Ulyad83.9
Controller Data:   12345678Ulyad83.9
com.mahasiswa.model.ModelMahasiswa@68ceda24
Mahasiswa berhasil ditambahkan!
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI: 2
Masukkan NPM:
12312398
Masukkan Nama:
Saburo
Masukkan Semester:
1
Masukkan IPK:
3.5
12312398Saburo13.5
Controller Data:   12312398Saburo13.5
com.mahasiswa.model.ModelMahasiswa@281e3708
Mahasiswa berhasil ditambahkan!
Menu:
1. Tampilkan Semua Mahasiswa
```

Tampilkan Semua Data Mahasiswa (Setelah Diupdate)



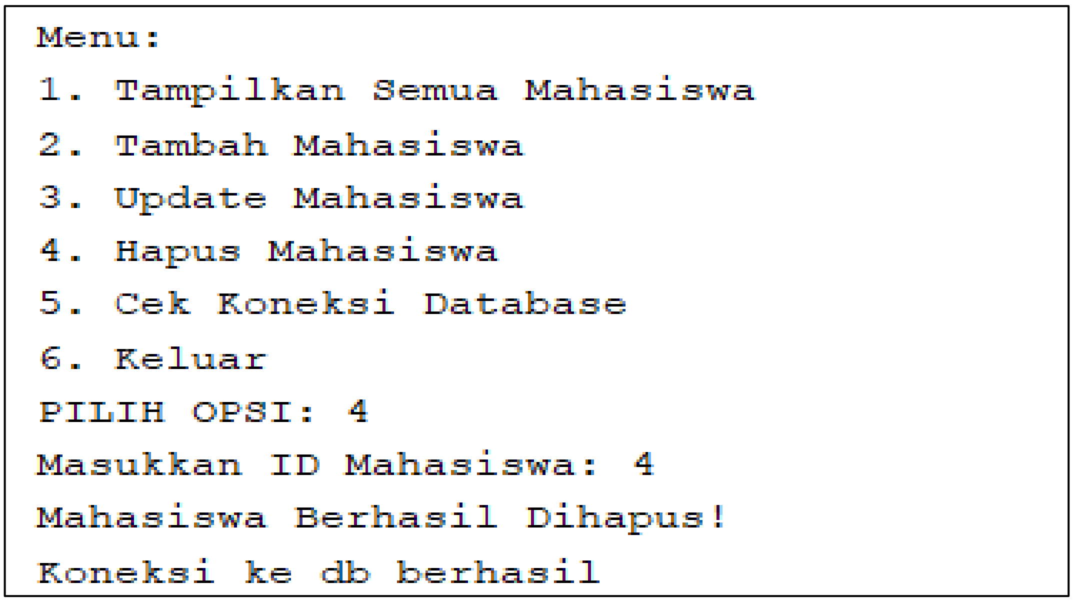
```
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI: 1

===========================
ID           : 2
NPM          : 50412345
NAMA         : Mulyono
SEMESTER     : 10
IPK          : 4.0
===========================
ID           : 3
NPM          : 50421797
NAMA         : Maulana
SEMESTER     : 7
IPK          : 3.95
===========================
ID           : 4
NPM          : 12312398
NAMA         : Saburo
SEMESTER     : 1
IPK          : 3.5
===========================
```

Update Data Mahasiswa



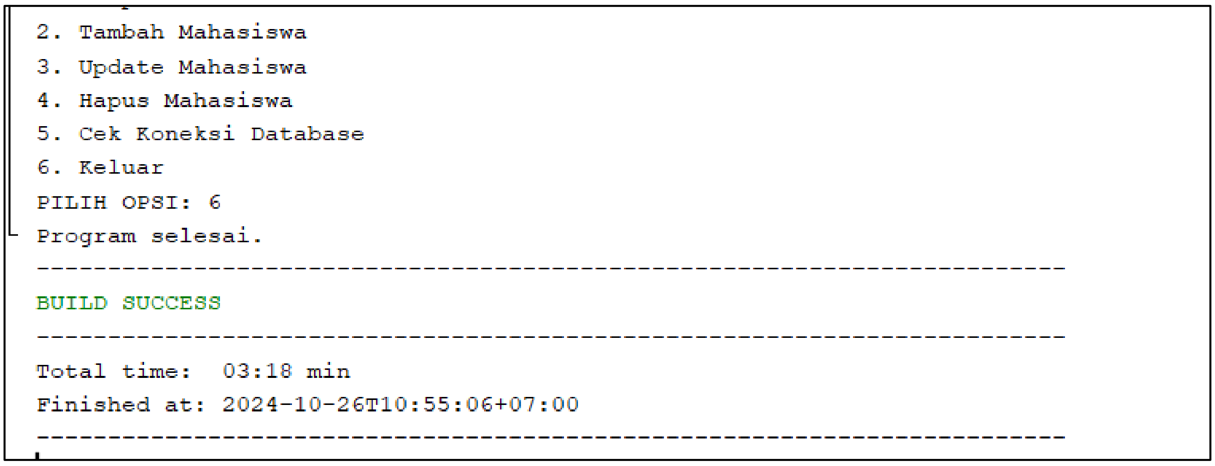
```
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI: 3
Masukkan ID mahasiswa: 3
Masukkan NPM:
50421797
Masukkan Nama:
Maulana
Masukkan Semester:
7
Masukkan IPK:
3.95
Mahasiswa berhasil diperbarui!
```

Hapus Data Mahasiswa



```
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: 4
Mahasiswa Berhasil Dihapus!
Koneksi ke db berhasil
```

Keluar

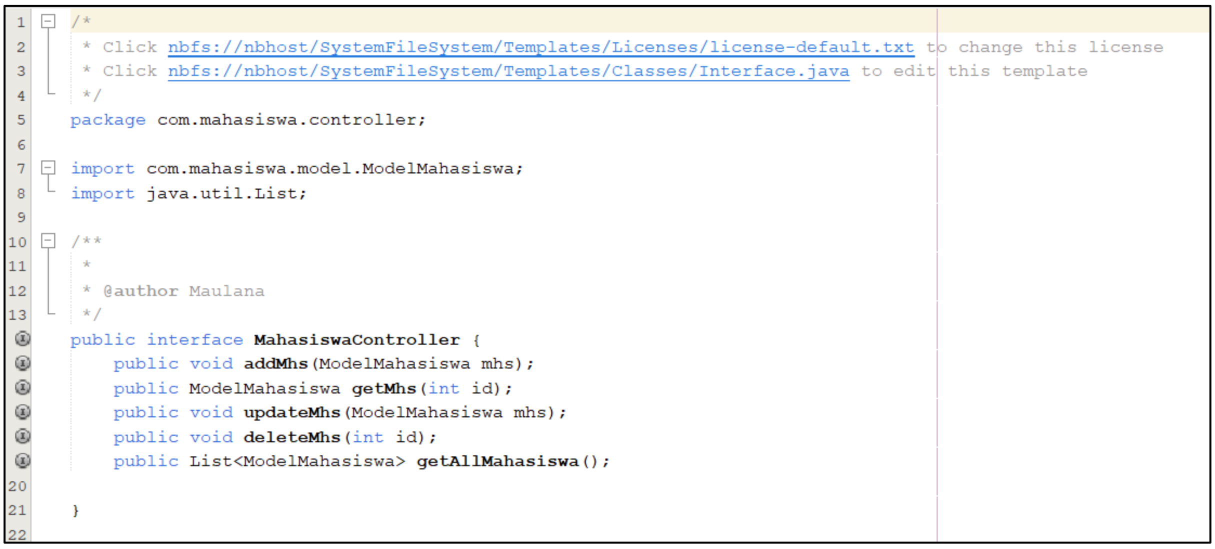


```
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI: 6
Program selesai.
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  03:18 min
Finished at: 2024-10-26T10:55:06+07:00
------------------------------------------------------------------------
```

Pertemuan 4

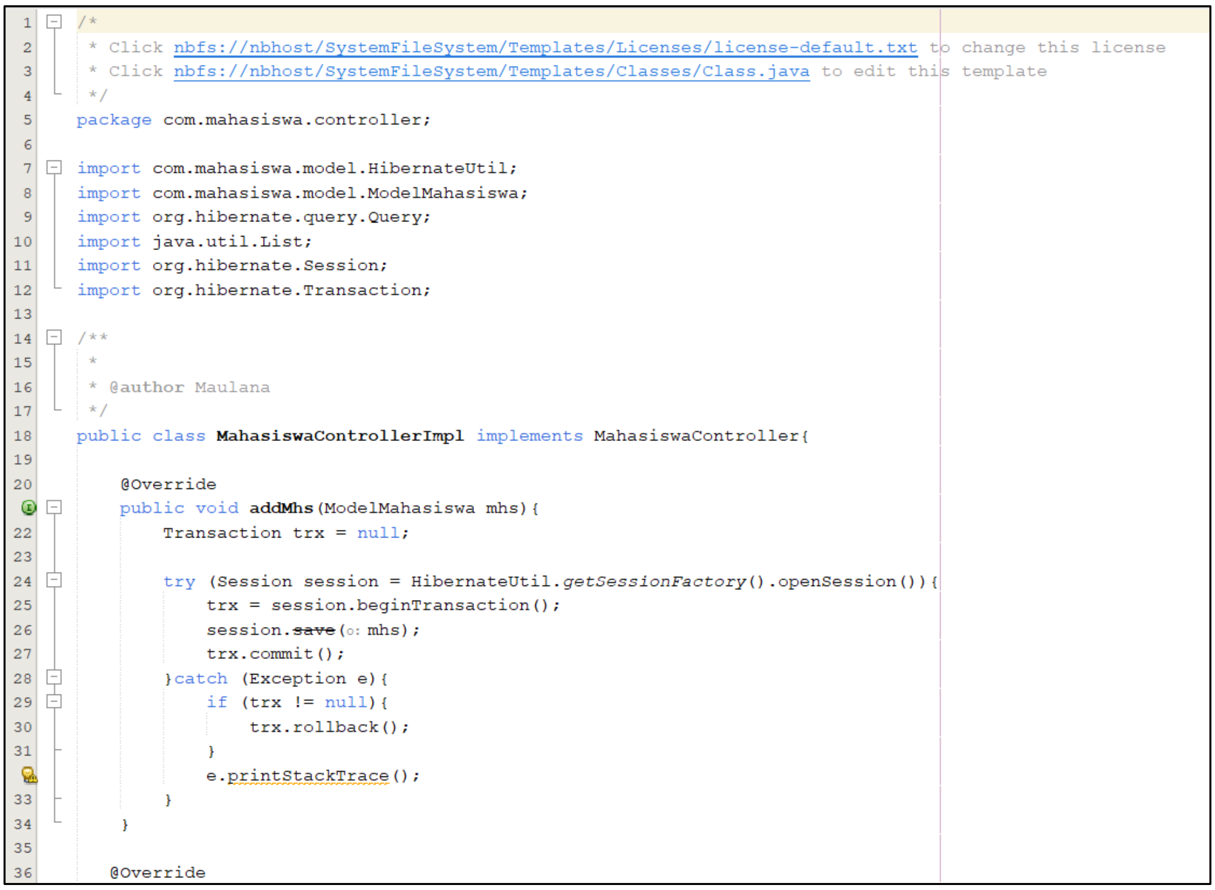
Code :

MahasiswaController.java

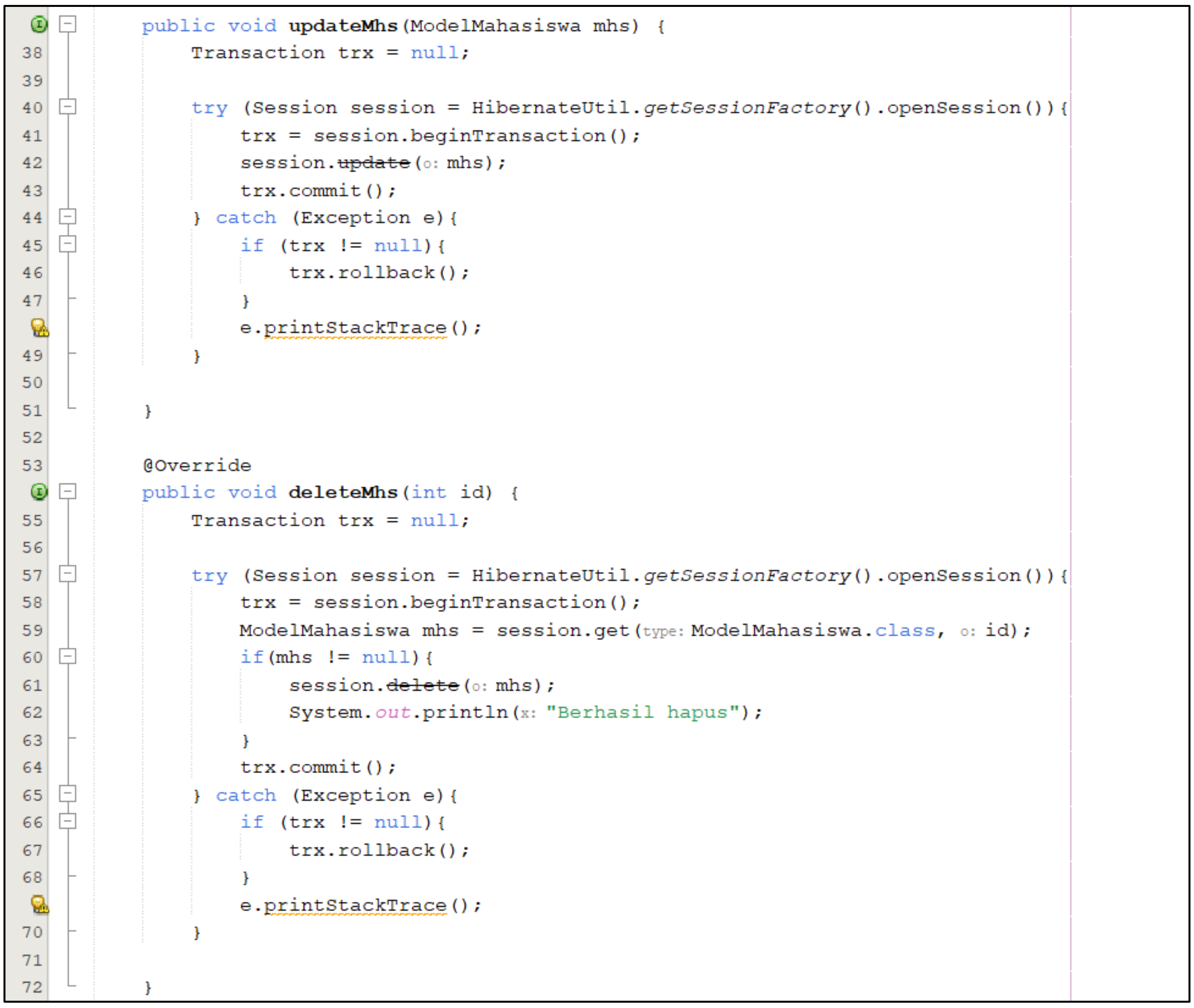


```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Interface.java to edit this template
 */
package com.mahasiswa.controller;

import com.mahasiswa.model.ModelMahasiswa;
import java.util.List;

/**
 *
 * @author Maulana
 */
public interface MahasiswaController {
    public void addMhs(ModelMahasiswa mhs);
    public ModelMahasiswa getMhs(int id);
    public void updateMhs(ModelMahasiswa mhs);
    public void deleteMhs(int id);
    public List<ModelMahasiswa> getAllMahasiswa();

}
```

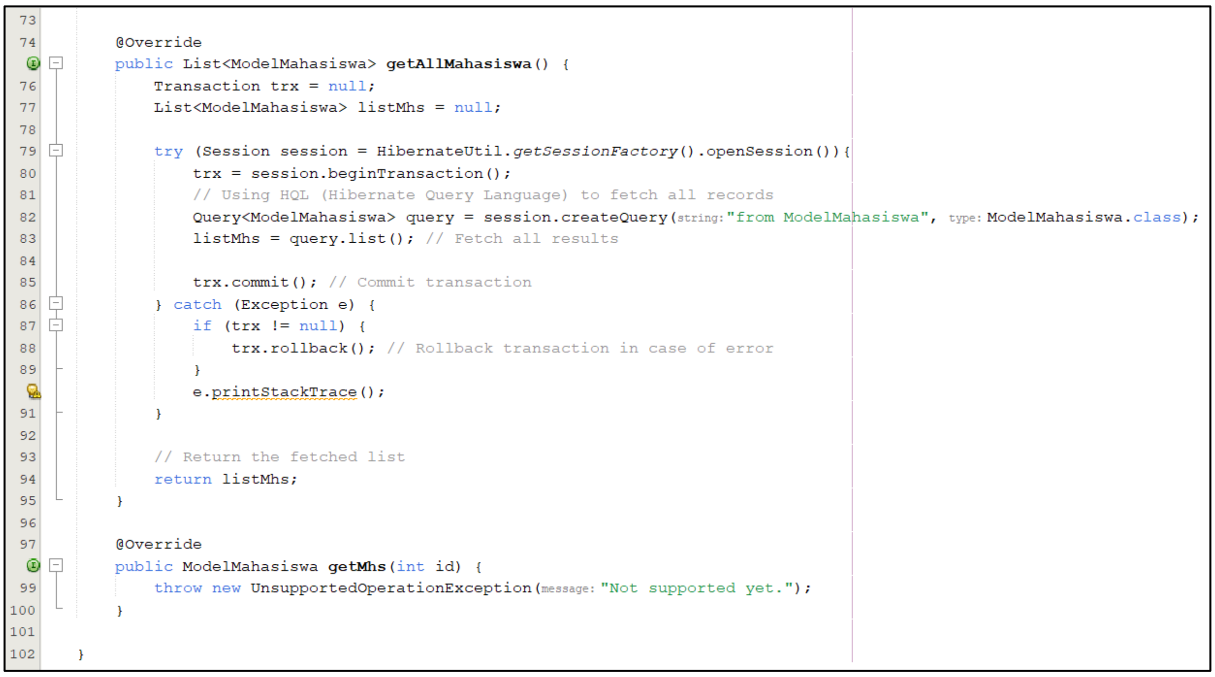
MahasiswaControllermpl.java



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

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

/**
 *
 * @author Maulana
 */
public class MahasiswaControllerImpl implements MahasiswaController{

    @Override
    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();
        }
    }

   @Override
```

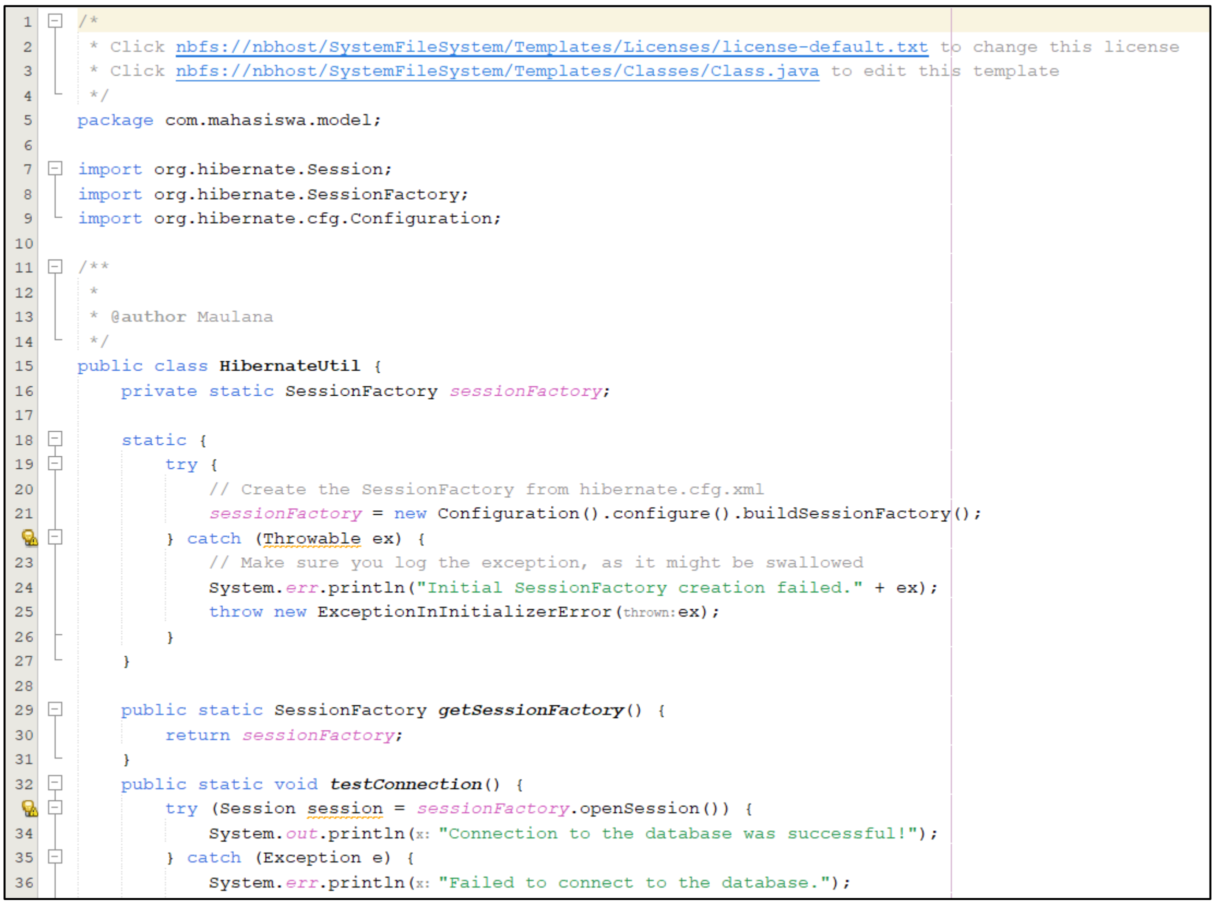


```java
    public void updateMhs(ModelMahasiswa mhs) {
        Transaction trx = null;

        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();
        }

    }

    @Override
    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();
        }

    }
```

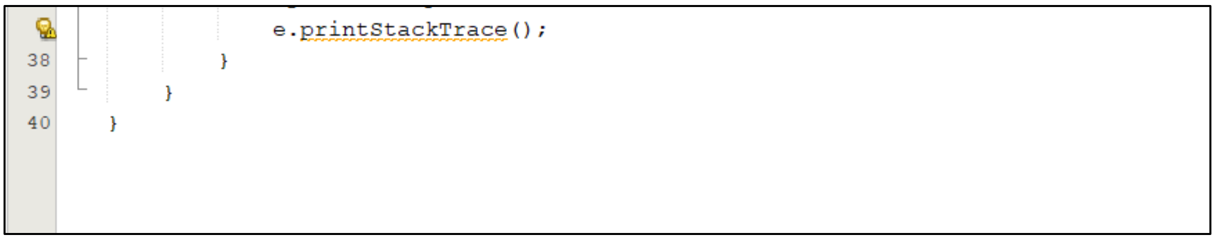


```java

    @Override
    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;
    }

    @Override
    public ModelMahasiswa getMhs(int id) {
        throw new UnsupportedOperationException(message: "Not supported yet.");
    }

}
```

HibernateUtil.java

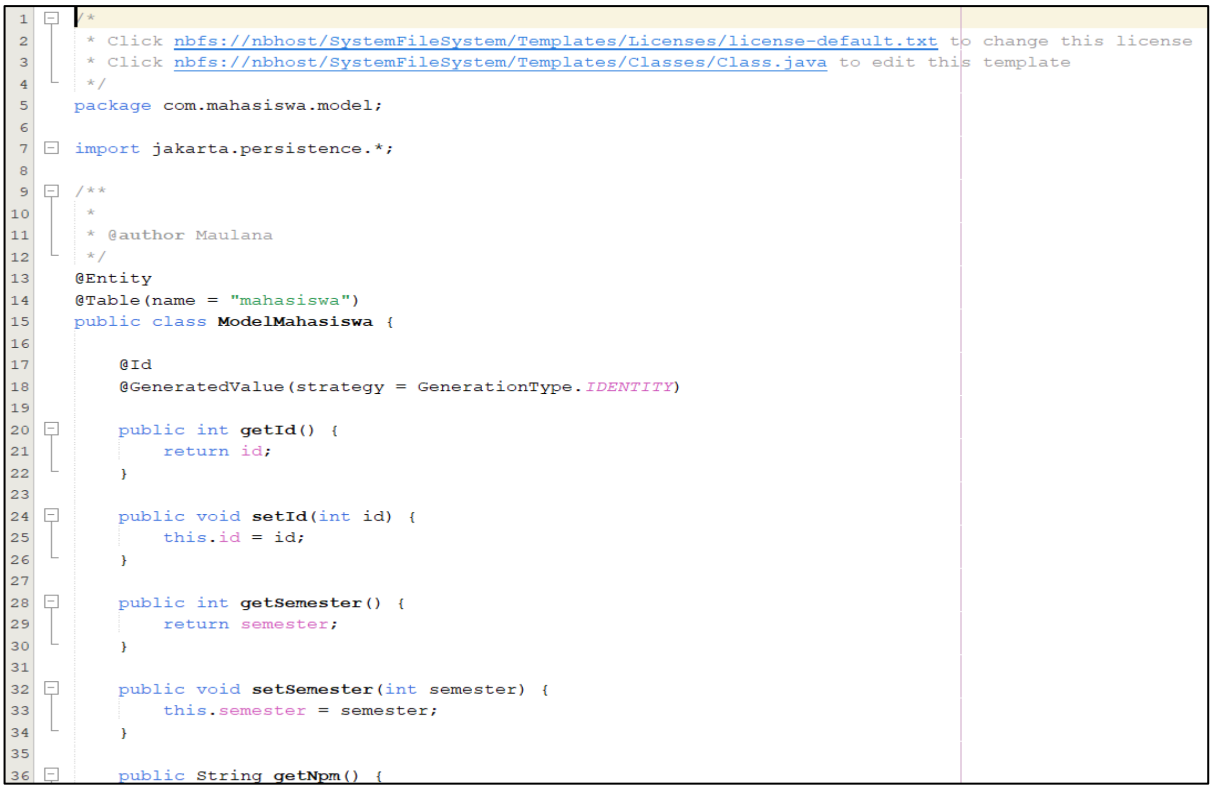


```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.mahasiswa.model;

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

/**
 *
 * @author Maulana
 */
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.");
```

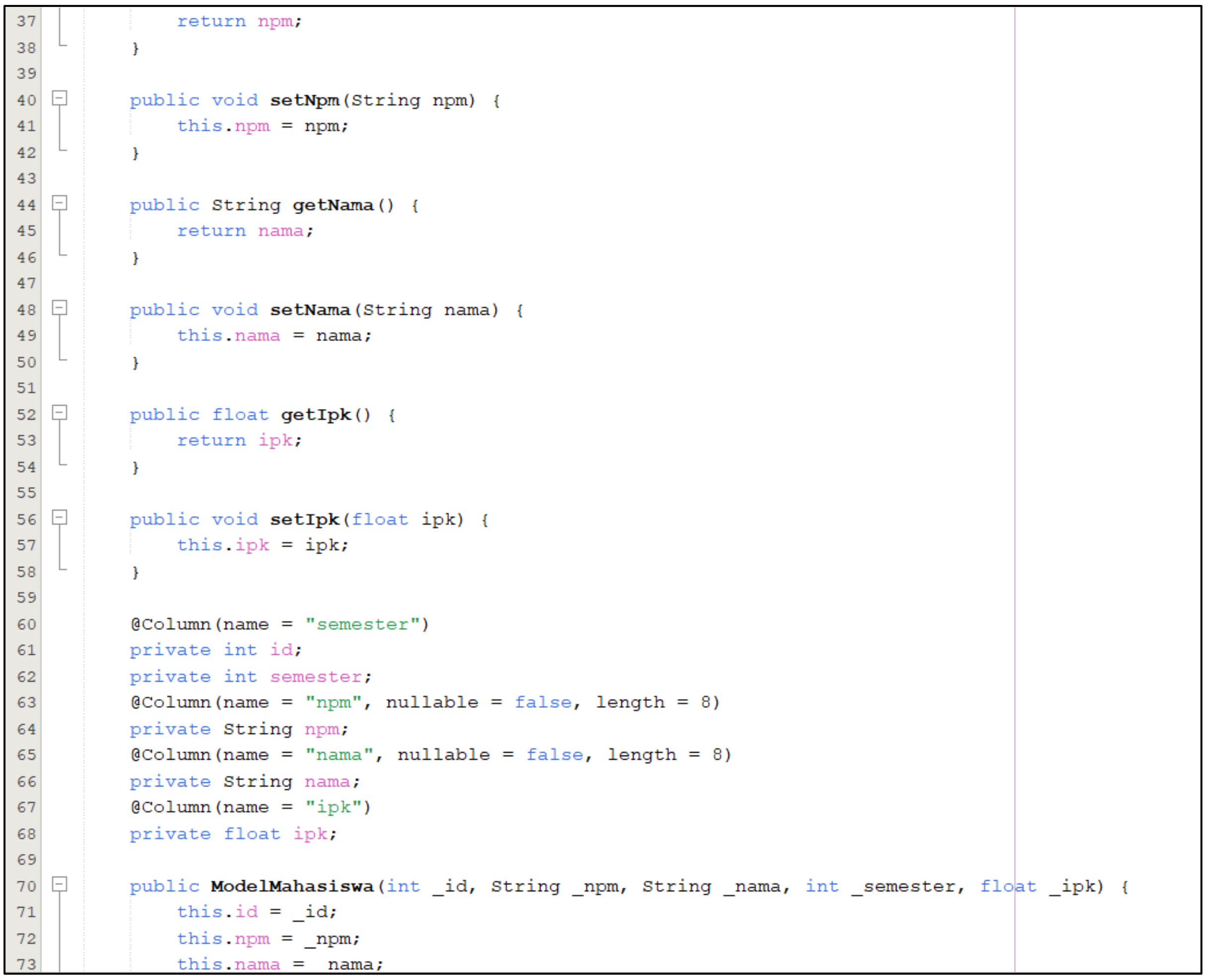


```java
            e.printStackTrace();
        }
    }
}
```

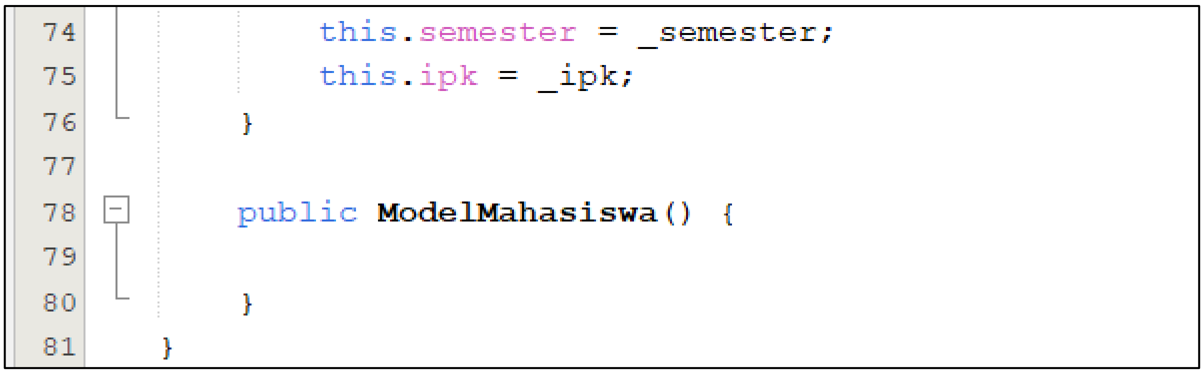
ModelMahasiswa.java



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

import jakarta.persistence.*;

/**
 *
 * @author Maulana
 */
@Entity
@Table(name = "mahasiswa")
public class ModelMahasiswa {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)

    public int getId() {
        return id;
    }

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

    public int getSemester() {
        return semester;
    }

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

    public String getNpm() {
```

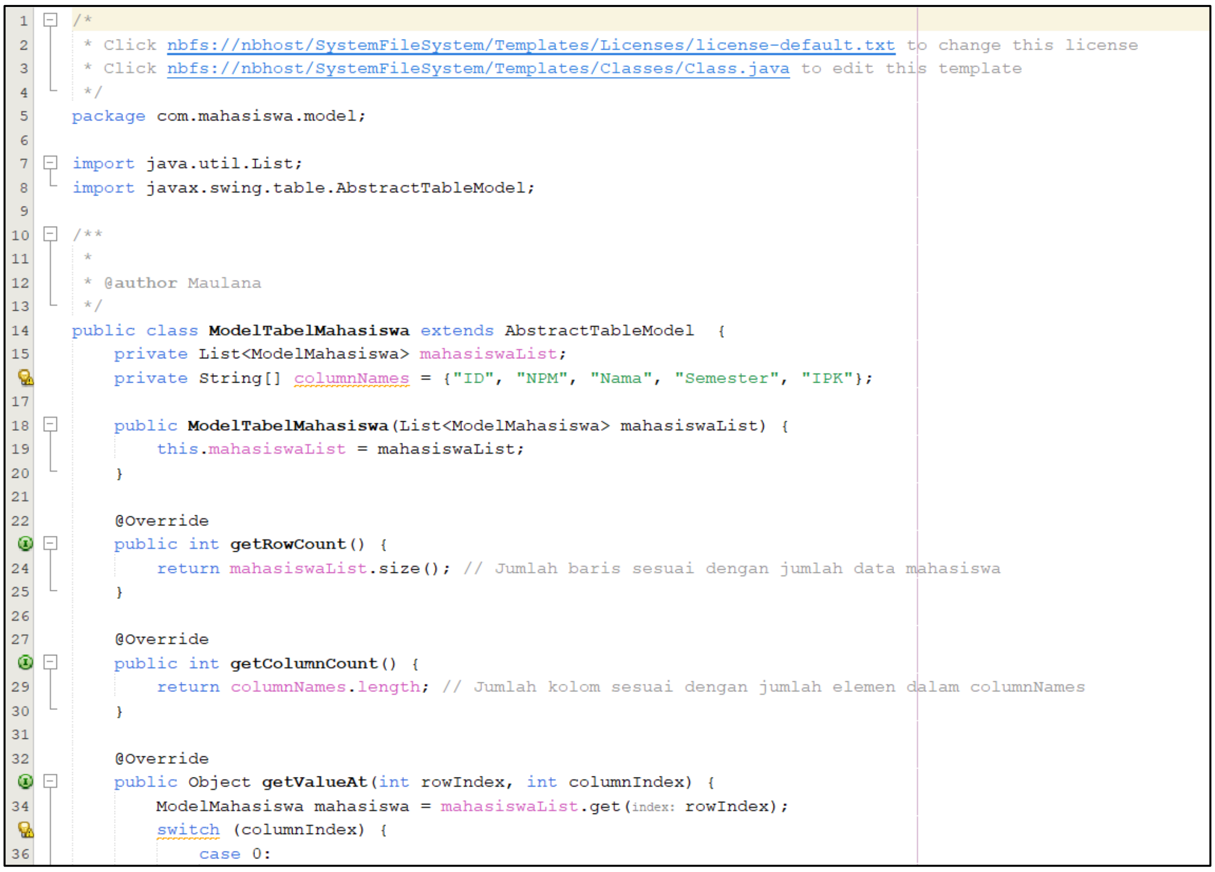


```java
        return npm;
    }

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

    public String getNama() {
        return nama;
    }

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

    public float getIpk() {
        return ipk;
    }

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

    @Column(name = "semester")
    private int id;
    private int semester;
    @Column(name = "npm", nullable = false, length = 8)
    private String npm;
    @Column(name = "nama", nullable = false, length = 8)
    private String nama;
    @Column(name = "ipk")
    private float ipk;

    public ModelMahasiswa(int _id, String _npm, String _nama, int _semester, float _ipk) {
        this.id = _id;
        this.npm = _npm;
        this.nama = _nama;
```

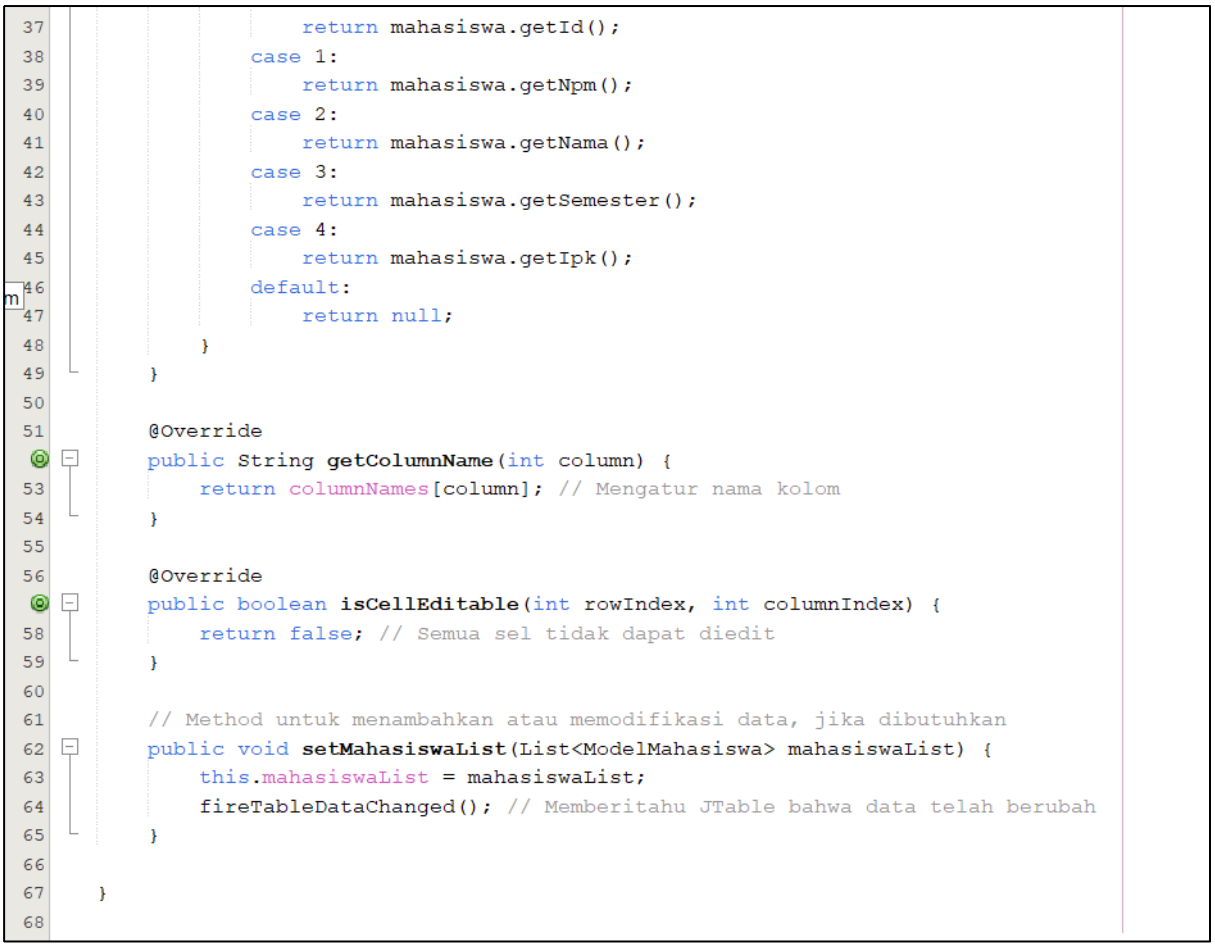


```java
            this.semester = _semester;
            this.ipk = _ipk;
        }

        public ModelMahasiswa()  {

        }
    }
```

ModelTabelMahasiswa.java

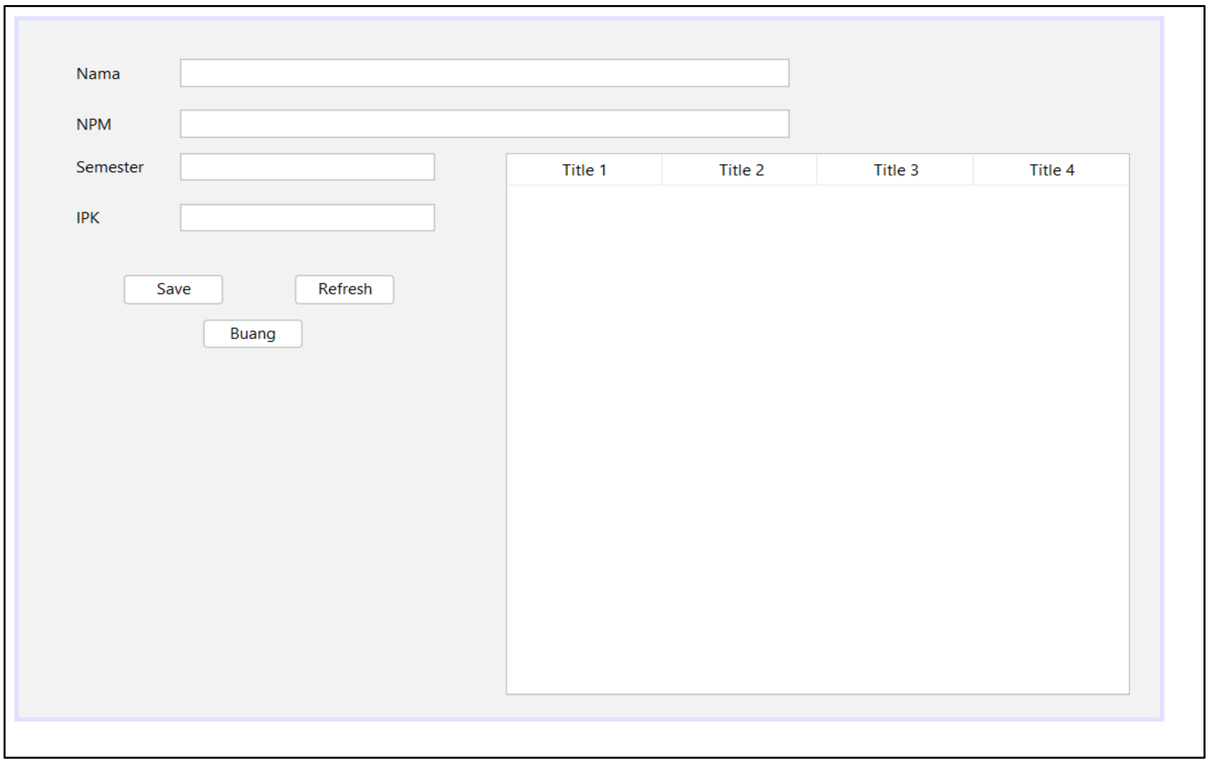


```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.mahasiswa.model;

import java.util.List;
import javax.swing.table.AbstractTableModel;

/**
 *
 * @author Maulana
 */
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:
```

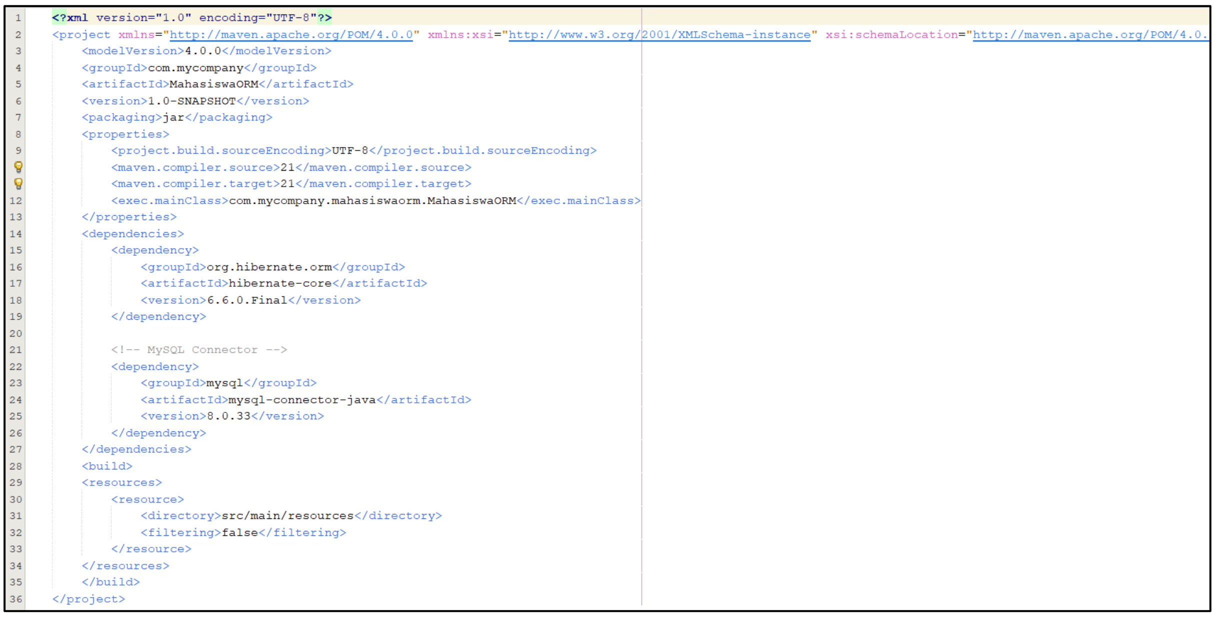


```
                    return mahasiswa.getId();
                case 1:
                    return mahasiswa.getNpm();
                case 2:
                    return mahasiswa.getNama();
                case 3:
                    return mahasiswa.getSemester();
                case 4:
                    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
        }

    }
```

MahasiswaView.java



pom.xml



```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.
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.mycompany</groupId>
    <artifactId>MahasiswaORM</artifactId>
    <version>1.0-SNAPSHOT</version>
    <packaging>jar</packaging>
    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <maven.compiler.source>21</maven.compiler.source>
        <maven.compiler.target>21</maven.compiler.target>
        <exec.mainClass>com.mycompany.mahasiswaorm.MahasiswaORM</exec.mainClass>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.hibernate.orm</groupId>
            <artifactId>hibernate-core</artifactId>
            <version>6.6.0.Final</version>
        </dependency>

        <!-- MySQL Connector -->
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
            <version>8.0.33</version>
        </dependency>
    </dependencies>
    <build>
    <resources>
        <resource>
            <directory>src/main/resources</directory>
            <filtering>false</filtering>
        </resource>
    </resources>
    </build>
</project>
```