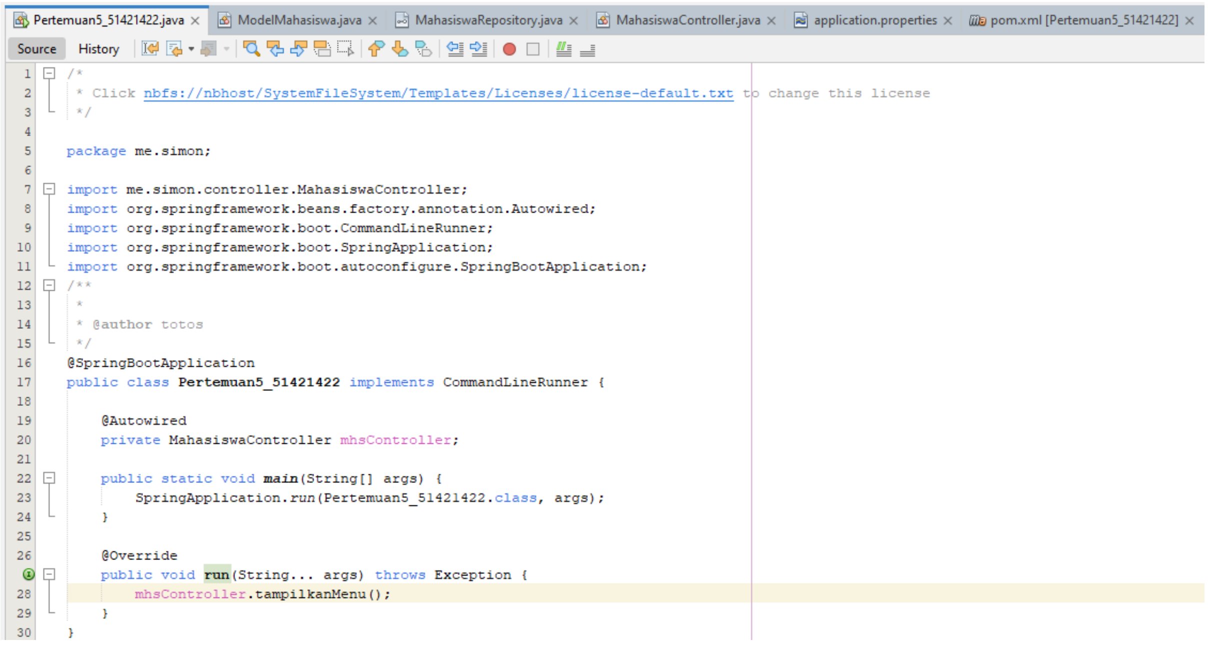
Activity Pertemuan 5

Nama : Simon Praja H P Pasaribu

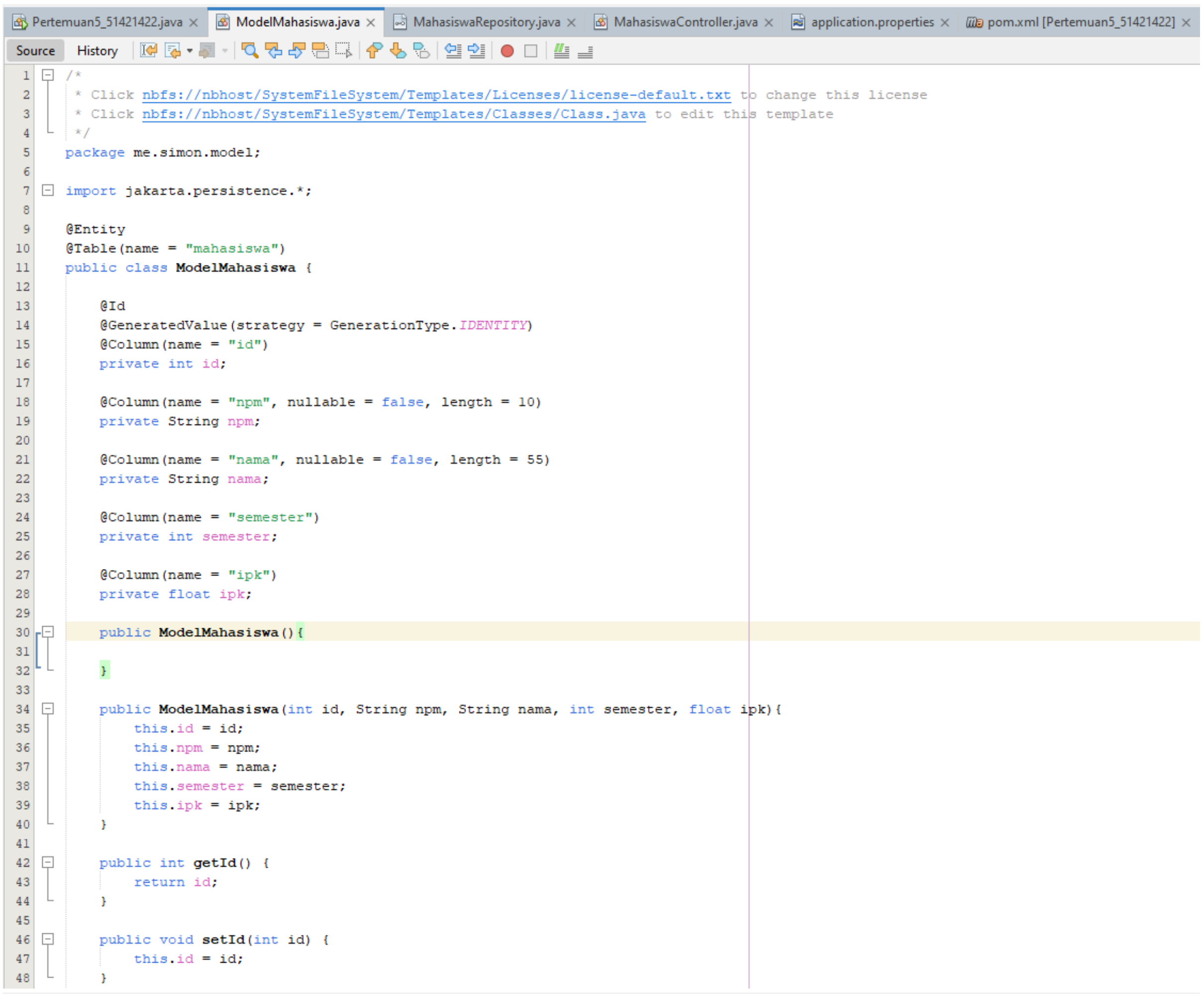
Kelas : 4IA14

NPM : 51421422

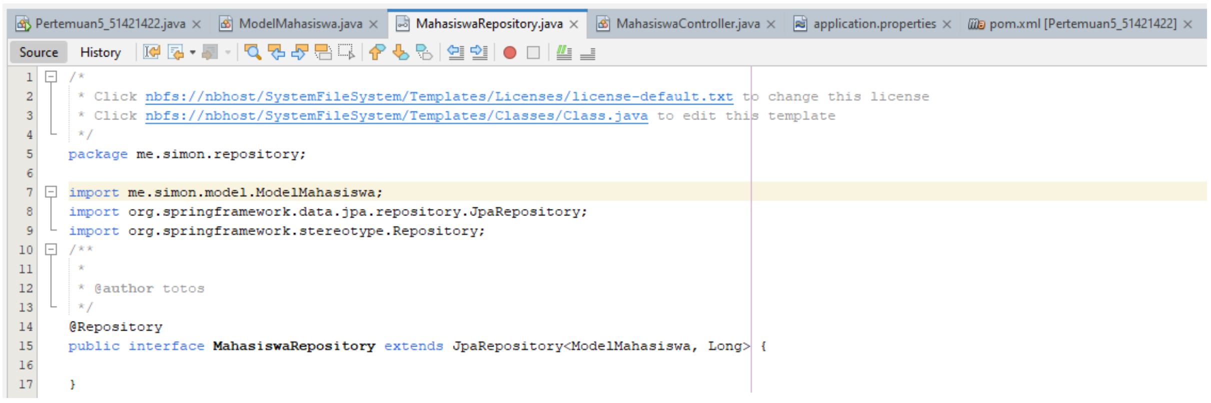
1. **Code:**



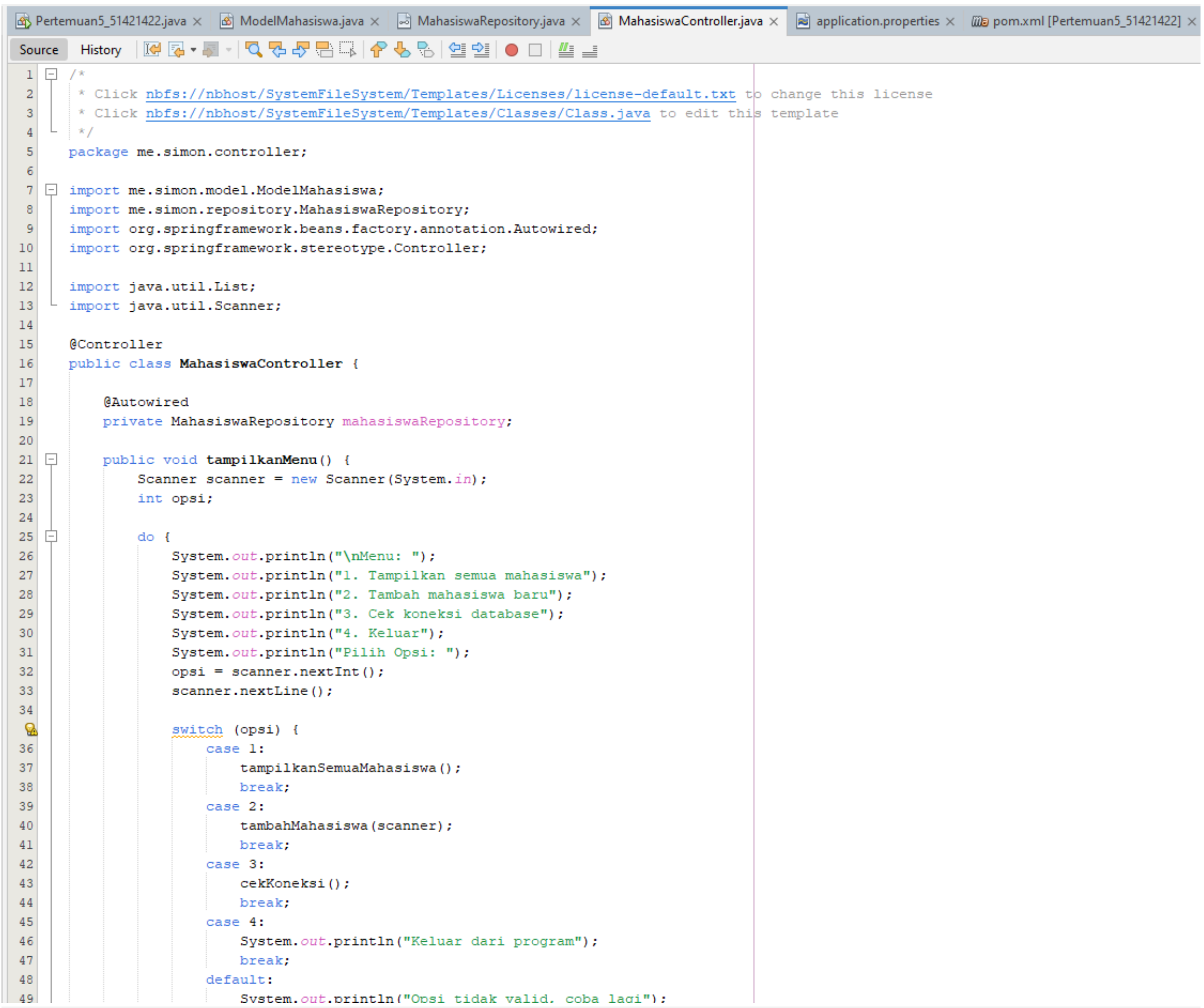
```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 */

package me.simon;

import me.simon.controller.MahasiswaController;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.boot.CommandLineRunner;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
/**
 *
 * @author totos
 */
@SpringBootApplication
public class Pertemuan5_51421422 implements CommandLineRunner {

    @Autowired
    private MahasiswaController mhsController;

    public static void main(String[] args) {
        SpringApplication.run(Pertemuan5_51421422.class, args);
    }

    @Override
    public void run(String... args) throws Exception {
        mhsController.tampilkanMenu();
    }
}
```



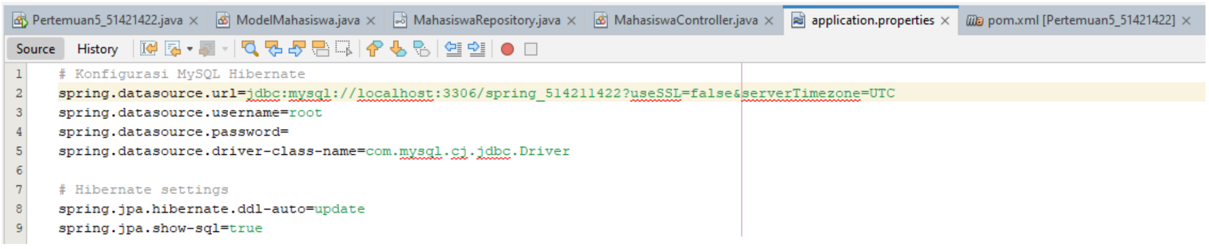
```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package me.simon.model;

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 = 10)
    private String npm;

    @Column(name = "nama", nullable = false, length = 55)
    private String nama;

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

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

    public ModelMahasiswa(){

    }

    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;
    }
```



```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 me.simon.repository;

import me.simon.model.ModelMahasiswa;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;
/**
 *
 * @author totos
 */
@Repository
public interface MahasiswaRepository extends JpaRepository<ModelMahasiswa, Long> {

}
```

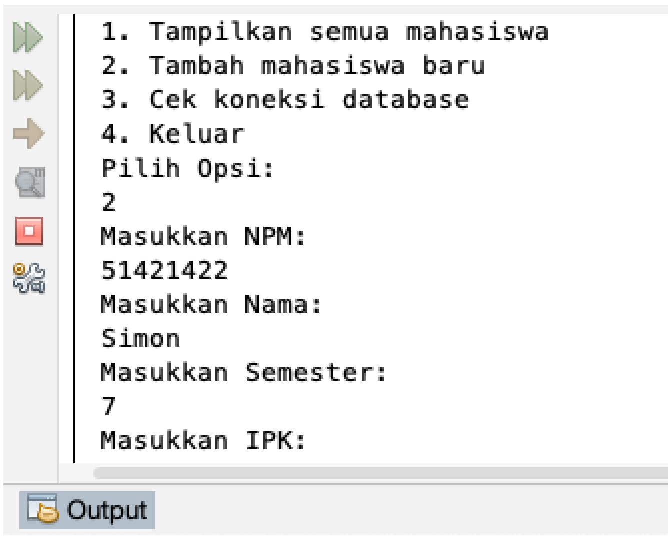


```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package me.simon.controller;

import me.simon.model.ModelMahasiswa;
import me.simon.repository.MahasiswaRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;

import java.util.List;
import java.util.Scanner;

@Controller
public class MahasiswaController {

    @Autowired
    private MahasiswaRepository mahasiswaRepository;

    public void tampilkanMenu() {
        Scanner scanner = new Scanner(System.in);
        int opsi;

        do {
            System.out.println("\nMenu: ");
            System.out.println("1. Tampilkan semua mahasiswa");
            System.out.println("2. Tambah mahasiswa baru");
            System.out.println("3. Cek koneksi database");
            System.out.println("4. Keluar");
            System.out.println("Pilih Opsi: ");
            opsi = scanner.nextInt();
            scanner.nextLine();

            switch (opsi) {
                case 1:
                    tampilkanSemuaMahasiswa();
                    break;
                case 2:
                    tambahMahasiswa(scanner);
                    break;
                case 3:
                    cekKoneksi();
                    break;
                case 4:
                    System.out.println("Keluar dari program");
                    break;
                default:
                    System.out.println("Opsi tidak valid, coba lagi");
```



```properties
# Konfigurasi MySQL Hibernate
spring.datasource.url=jdbc:mysql://localhost:3306/spring_51421422?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:



```
1. Tampilkan semua mahasiswa
2. Tambah mahasiswa baru
3. Cek koneksi database
4. Keluar
Pilih Opsi:
2
Masukkan NPM:
51421422
Masukkan Nama:
Simon
Masukkan Semester:
7
Masukkan IPK:
```

2. Dependency Injection (DI) adalah sebuah pola desain yang digunakan untuk mengelola dan menyuntikkan (inject) dependensi antar komponen dalam aplikasi. Dalam konteks Spring Framework, DI mengacu pada cara Spring mengelola objek-objek dalam aplikasi dan menyediakan objek-objek tersebut ke dalam komponen lain sesuai dengan kebutuhan.