ACTIVITY PERTEMUAN 5

NAMA : Andi Purnama

NPM : 50422215

KELAS : 4IA18

MATERI : Spring Framework

MATA PRAKTIKUM : Rekayasa Perangkat Lunak 2

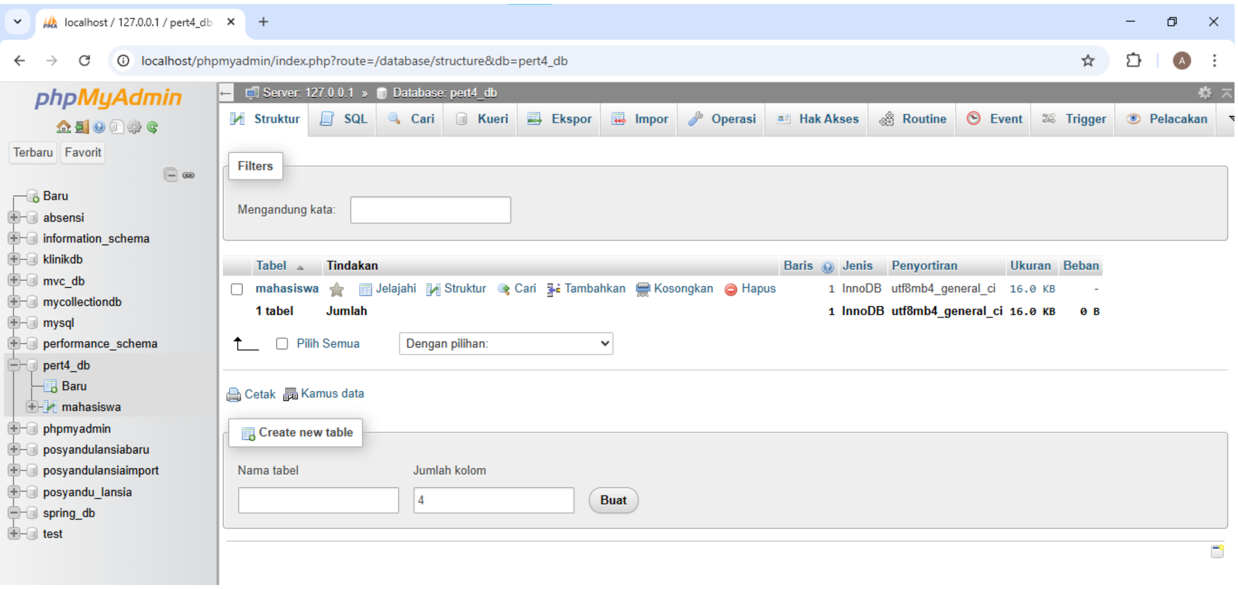
---

1. Dependency Injection (DI) adalah konsep penting dalam Spring Framework. Jelaskan apa itu Dependency Injection dan mengapa DI sangat penting dalam pengembangan aplikasi berbasis Spring Boot.

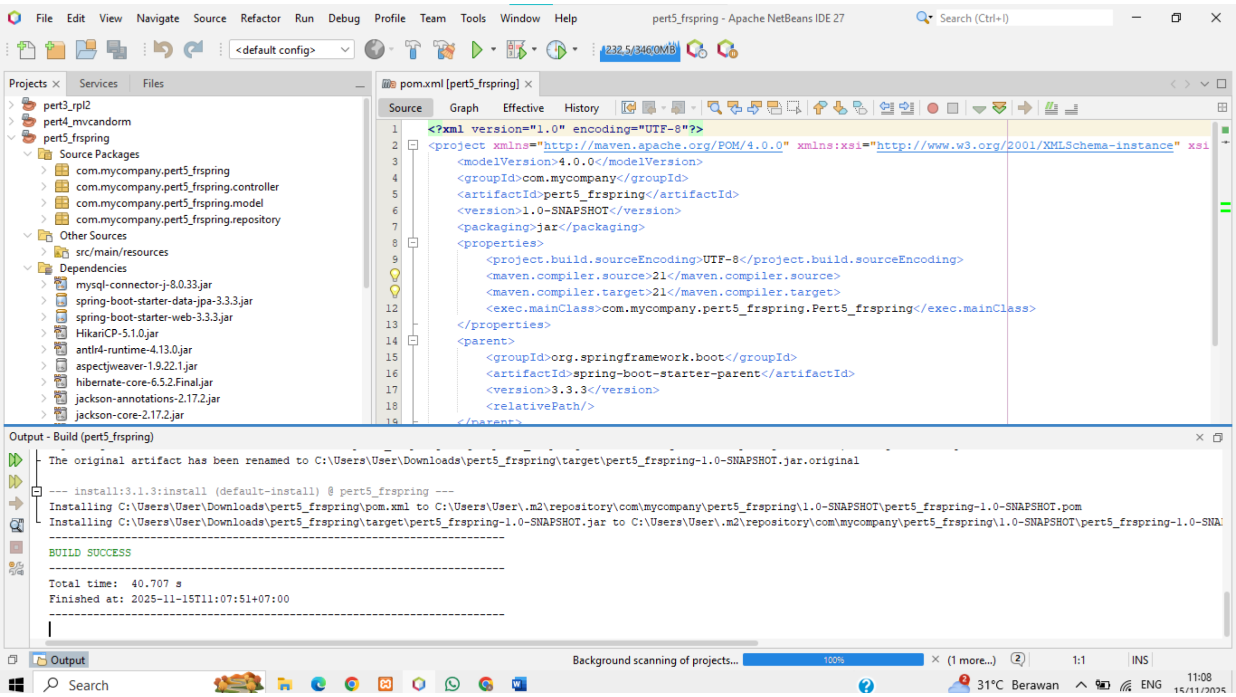
Jawab:

Dependency Injection (DI) adalah sebuah konsep di mana suatu objek tidak membuat sendiri objek lain yang dibutuhkannya, tetapi menerima (atau “disuntikkan”) dependensi tersebut dari luar oleh Spring melalui mekanisme Inversion of Control (IoC). Dengan pendekatan ini, objek hanya mendefinisikan apa yang dibutuhkannya, sementara Spring yang bertanggung jawab mengelola siklus hidup, pembuatan, dan penyuntikan objek-objek tersebut. DI sangat penting dalam pengembangan aplikasi berbasis Spring Boot karena mampu mengurangi tight coupling antar komponen sehingga kode menjadi lebih fleksibel dan mudah dikembangkan. Selain itu, DI mempermudah proses testing karena dependensi dapat diganti dengan mock secara sederhana tanpa harus mengubah logika utama. DI juga memungkinkan pengelolaan komponen secara efisien melalui container Spring, sehingga konfigurasi aplikasi menjadi lebih rapi, modular, serta mendukung prinsip clean architecture.

2. Screenshot Kode dan Output Program!



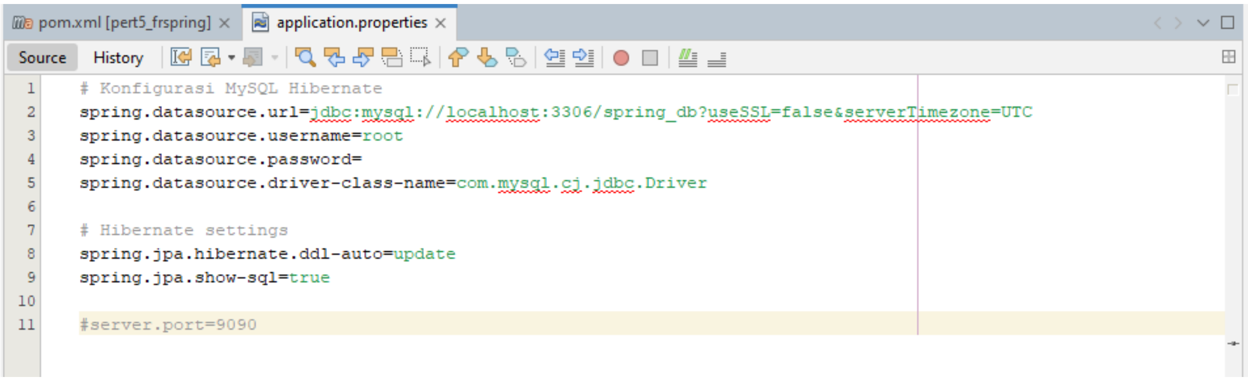
Tabel	Tindakan	Baris	Jenis	Penyortiran	Ukuran	Beban
mahasiswa	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	1	InnoDB	utf8mb4_general_ci	16.0 KB	-
1 tabel	Jumlah	1	InnoDB	utf8mb4_general_ci	16.0 KB	0 B



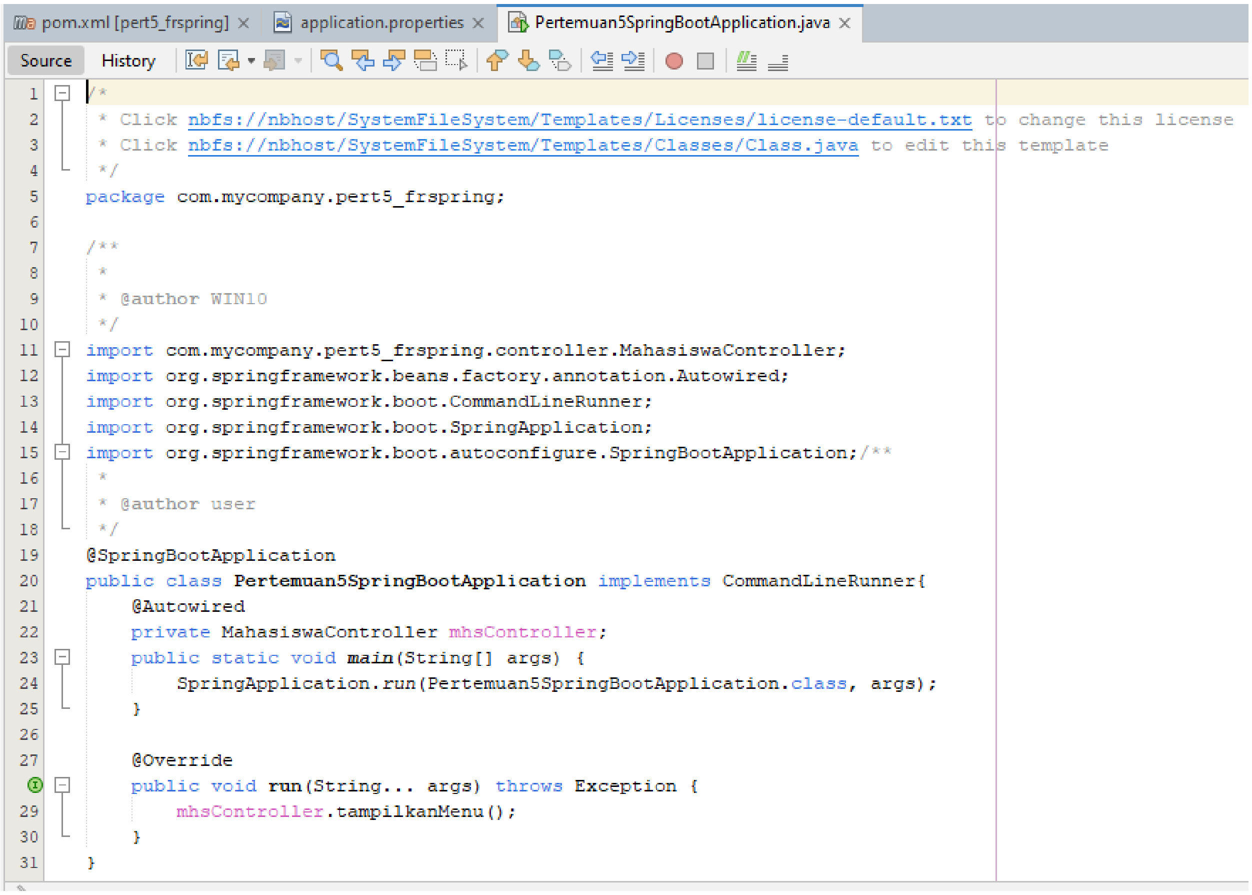
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
pert5_frspring - Apache NetBeans IDE 27
Search (Ctrl+I)
<default config>
232,5/346,0MB
Projects Services Files
pert3_rpl2
pert4_mvcandorm
pert5_frspring
Source Packages
com.mycompany.pert5_frspring
com.mycompany.pert5_frspring.controller
com.mycompany.pert5_frspring.model
com.mycompany.pert5_frspring.repository
Other Sources
src/main/resources
Dependencies
mysql-connector-j-8.0.33.jar
spring-boot-starter-data-jpa-3.3.3.jar
spring-boot-starter-web-3.3.3.jar
HikariCP-5.1.0.jar
antlr4-runtime-4.13.0.jar
aspectjweaver-1.9.22.1.jar
hibernate-core-6.5.2.Final.jar
jackson-annotations-2.17.2.jar
jackson-core-2.17.2.jar
pom.xml [pert5_frspring]
Source Graph Effective History
Output - Build (pert5_frspring)
Output
Background scanning of projects...
100%
(1 more...)
1:1
INS
Search
31°C Berawan
ENG
11:08
15/11/2025

```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
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.mycompany</groupId>
    <artifactId>pert5_frspring</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.pert5_frspring.Pert5_frspring</exec.mainClass>
    </properties>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>3.3.3</version>
        <relativePath/>
    </parent>
```

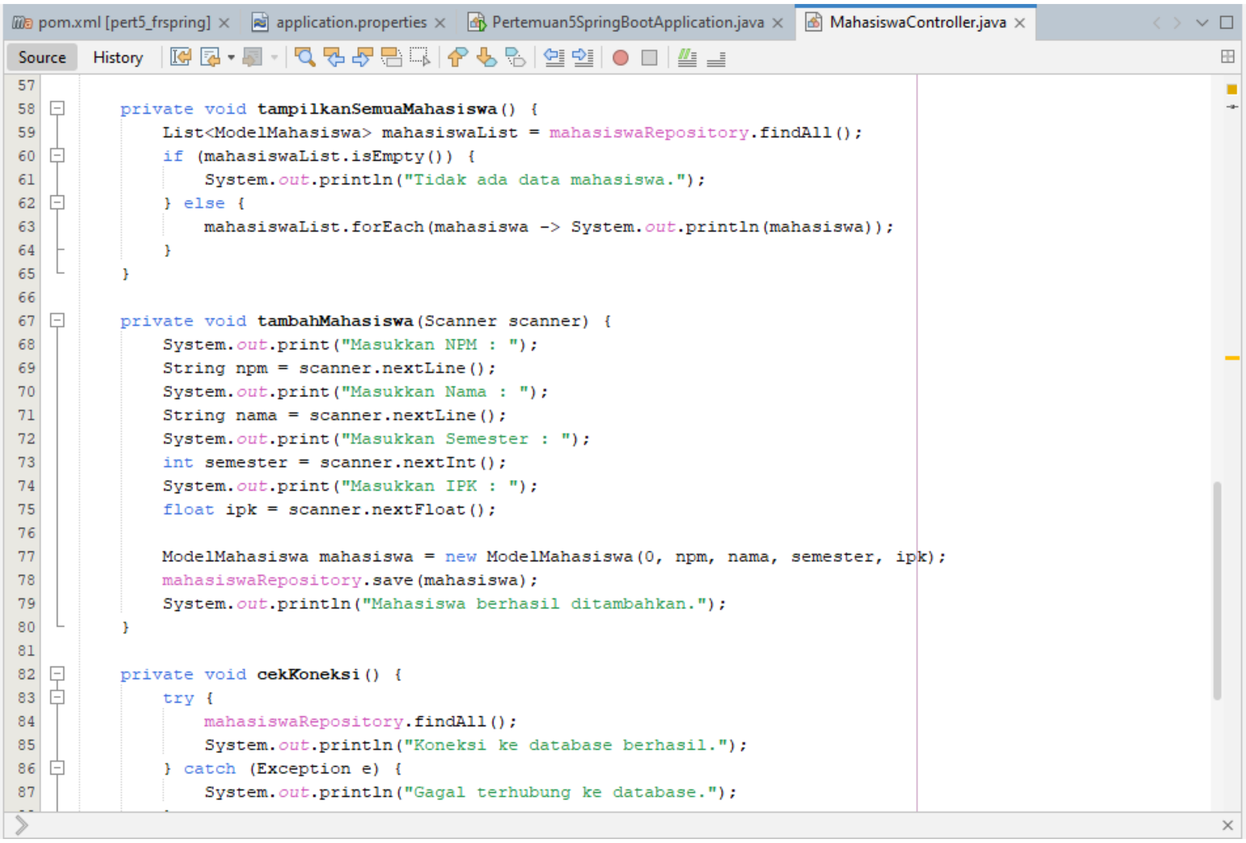
```
The original artifact has been renamed to C:\Users\User\Downloads\pert5_frspring\target\pert5_frspring-1.0-SNAPSHOT.jar.original

--- install:3.1.3:install (default-install) @ pert5_frspring ---
Installing C:\Users\User\Downloads\pert5_frspring\pom.xml to C:\Users\User\.m2\repository\com\mycompany\pert5_frspring\1.0-SNAPSHOT\pert5_frspring-1.0-SNAPSHOT.pom
Installing C:\Users\User\Downloads\pert5_frspring\target\pert5_frspring-1.0-SNAPSHOT.jar to C:\Users\User\.m2\repository\com\mycompany\pert5_frspring\1.0-SNAPSHOT\pert5_frspring-1.0-SNA
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  40.707 s
Finished at: 2025-11-15T11:07:51+07:00
------------------------------------------------------------------------
```



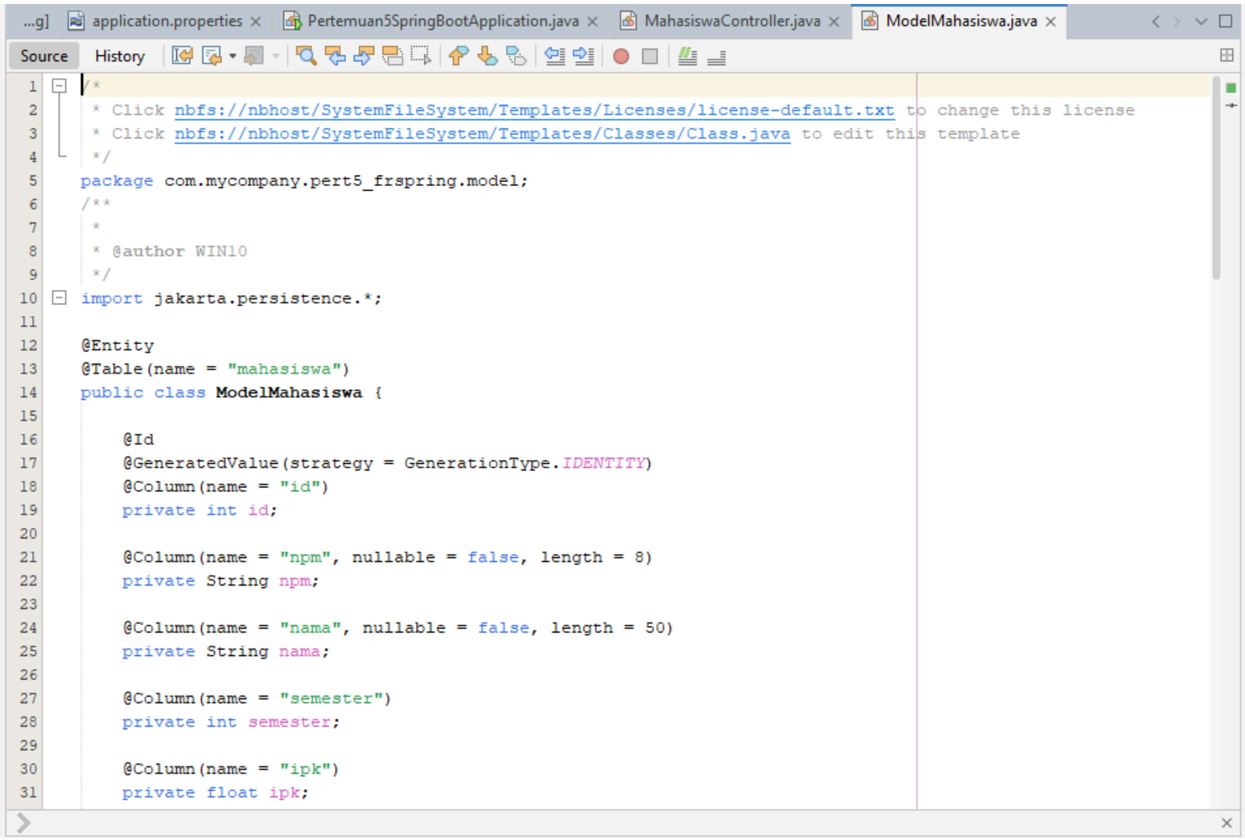
```
# Konfigurasi MySQL Hibernate
spring.datasource.url=jdbc:mysql://localhost:3306/spring_db?useSSL=false&serverTimezone=UTC
spring.datasource.username=root
spring.datasource.password=
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

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

#server.port=9090
```



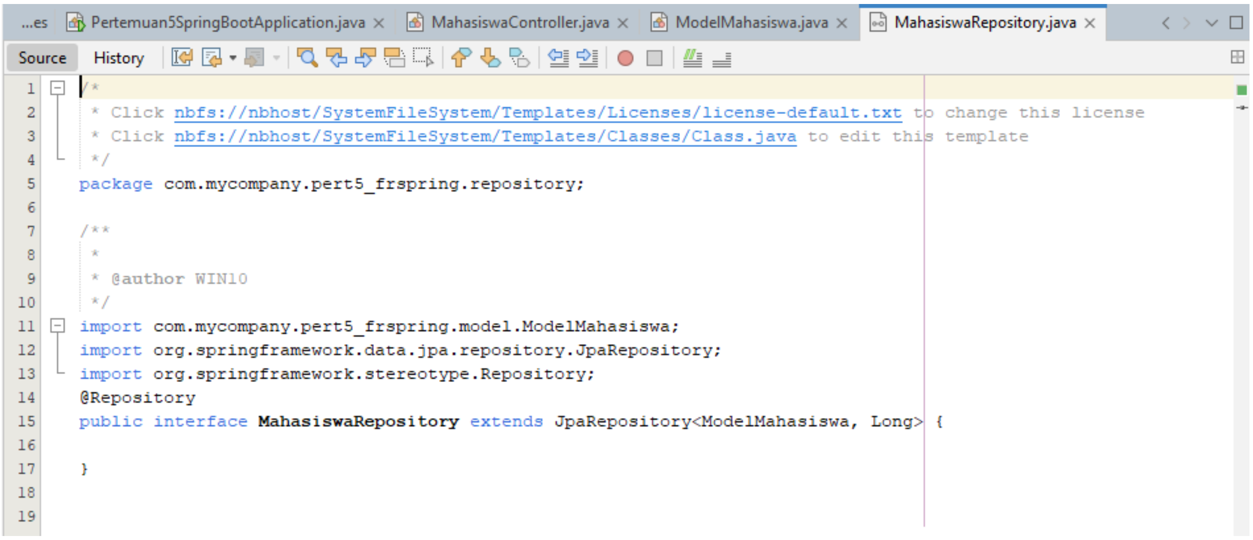
```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.mycompany.pert5_frspring;

/**
 *
 * @author WIN10
 */
import com.mycompany.pert5_frspring.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 user
 */
@SpringBootApplication
public class Pertemuan5SpringBootApplication implements CommandLineRunner{
    @Autowired
    private MahasiswaController mhsController;
    public static void main(String[] args) {
        SpringApplication.run(Pertemuan5SpringBootApplication.class, args);
    }

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



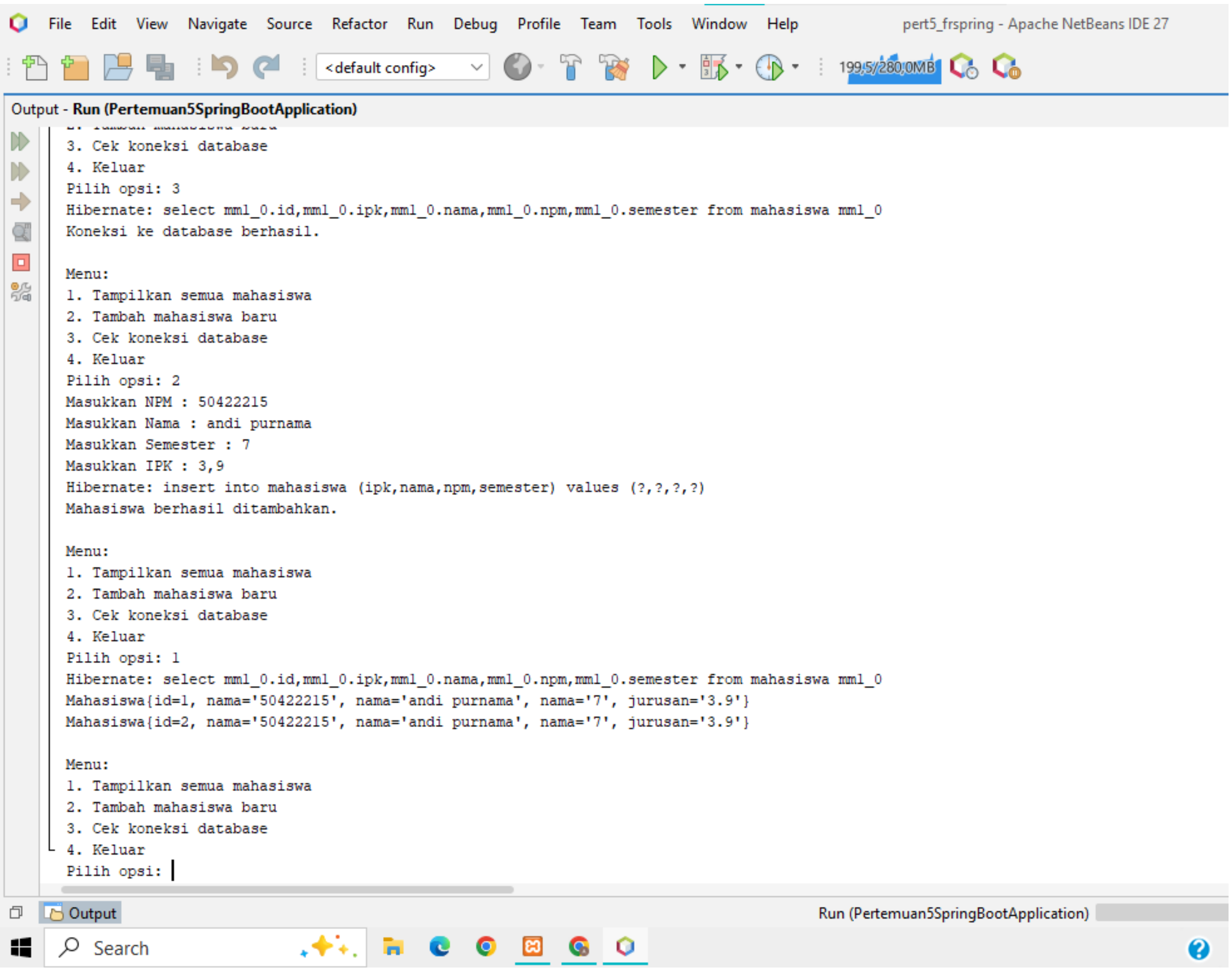
```java

    private void tampilkanSemuaMahasiswa() {
        List<ModelMahasiswa> mahasiswaList = mahasiswaRepository.findAll();
        if (mahasiswaList.isEmpty()) {
            System.out.println("Tidak ada data mahasiswa.");
        } else {
            mahasiswaList.forEach(mahasiswa -> System.out.println(mahasiswa));
        }
    }

    private void tambahMahasiswa(Scanner scanner) {
        System.out.print("Masukkan NPM : ");
        String npm = scanner.nextLine();
        System.out.print("Masukkan Nama : ");
        String nama = scanner.nextLine();
        System.out.print("Masukkan Semester : ");
        int semester = scanner.nextInt();
        System.out.print("Masukkan IPK : ");
        float ipk = scanner.nextFloat();

        ModelMahasiswa mahasiswa = new ModelMahasiswa(0, npm, nama, semester, ipk);
        mahasiswaRepository.save(mahasiswa);
        System.out.println("Mahasiswa berhasil ditambahkan.");
    }

    private void cekKoneksi() {
        try {
            mahasiswaRepository.findAll();
            System.out.println("Koneksi ke database berhasil.");
        } catch (Exception e) {
            System.out.println("Gagal terhubung ke database.");
```



```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.mycompany.pert5_frspring.model;
/**
 *
 * @author WIN10
 */
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;
```



```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.mycompany.pert5_frspring.repository;

/**
 *
 * @author WIN10
 */
import com.mycompany.pert5_frspring.model.ModelMahasiswa;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;
@Repository
public interface MahasiswaRepository extends JpaRepository<ModelMahasiswa, Long> {

}
```


File
Edit
View
Navigate
Source
Refactor
Run
Debug
Profile
Team
Tools
Window
Help
pert5_frspring - Apache NetBeans IDE 27
<default config>
199,5/280,0MB
Output - Run (Pertemuan5SpringBootApplication)
Output
Run (Pertemuan5SpringBootApplication)
Search

```
3. Cek koneksi database
4. Keluar
Pilih opsi: 3
Hibernate: select mml_0.id,mml_0.ipk,mml_0.nama,mml_0.npm,mml_0.semester from mahasiswa mml_0
Koneksi ke database berhasil.

Menu:
1. Tampilkan semua mahasiswa
2. Tambah mahasiswa baru
3. Cek koneksi database
4. Keluar
Pilih opsi: 2
Masukkan NPM : 50422215
Masukkan Nama : andi purnama
Masukkan Semester : 7
Masukkan IPK : 3,9
Hibernate: insert into mahasiswa (ipk,nama,npm,semester) values (?,?,?,?)
Mahasiswa berhasil ditambahkan.

Menu:
1. Tampilkan semua mahasiswa
2. Tambah mahasiswa baru
3. Cek koneksi database
4. Keluar
Pilih opsi: 1
Hibernate: select mml_0.id,mml_0.ipk,mml_0.nama,mml_0.npm,mml_0.semester from mahasiswa mml_0
Mahasiswa{id=1, nama='50422215', nama='andi purnama', nama='7', jurusan='3.9'}
Mahasiswa{id=2, nama='50422215', nama='andi purnama', nama='7', jurusan='3.9'}

Menu:
1. Tampilkan semua mahasiswa
2. Tambah mahasiswa baru
3. Cek koneksi database
4. Keluar
Pilih opsi: 
```