Nama : Rachel Amelia Lesmana

Kelas : 4IA14

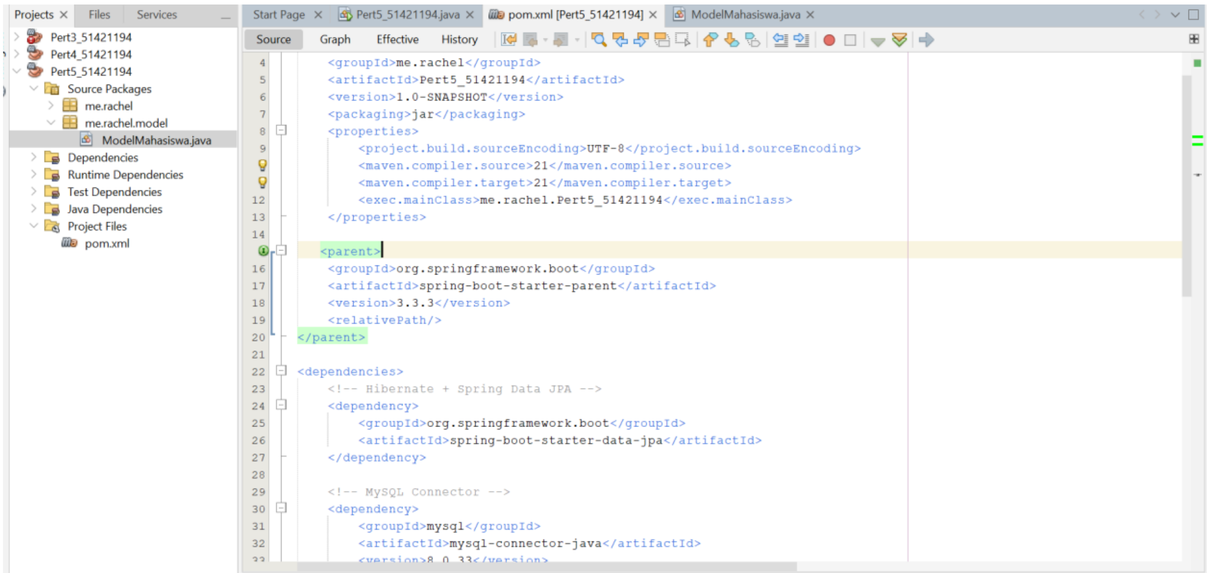
NPM : 51421194

Dependency Injection (DI) adalah sebuah pola desain (design pattern) dalam pemrograman yang digunakan untuk mengelola ketergantungan antara objek-objek dalam aplikasi. Dalam konteks Spring Framework, Dependency Injection adalah cara untuk mengelola objek-objek dalam aplikasi sehingga objek yang satu dapat memperoleh ketergantungan atau dependency dari objek lainnya, tanpa perlu membuat atau menginisialisasi objek tersebut secara manual.

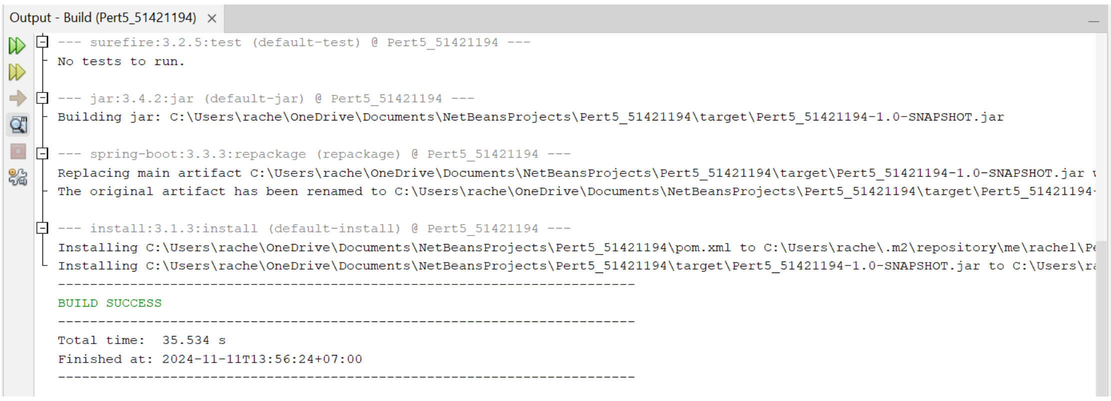
Mengapa penting:

- Mengurangi Keterikatan (Loose Coupling)
- Meningkatkan Testabilitas
- Mempermudah Pengelolaan Lifecycle
- Meningkatkan Keterbacaan Kode
- Fleksibilitas dalam Pengaturan

POM

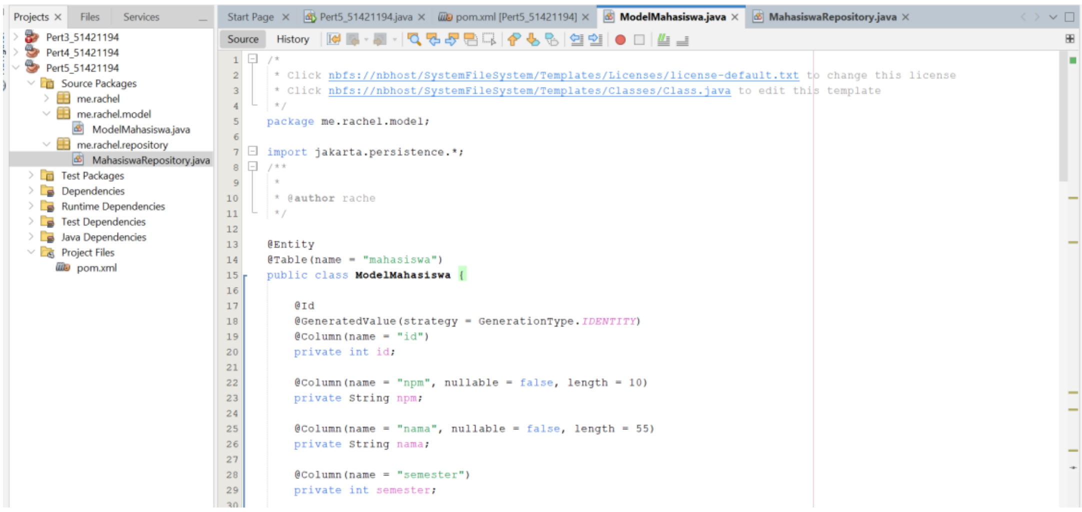


```xml
    <groupId>me.rachel</groupId>
    <artifactId>Pert5_51421194</artifactId>
    <version>1.0-SNAPSHOT</version>
    <packaging>jar</packaging>
    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <maven.compiler.source>21</maven.compiler.source>
        <maven.compiler.target>21</maven.compiler.target>
        <exec.mainClass>me.rachel.Pert5_51421194</exec.mainClass>
    </properties>

   <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>3.3.3</version>
    <relativePath/>
</parent>

<dependencies>
    <!-- Hibernate + Spring Data JPA -->
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>

    <!-- MySQL Connector -->
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <version>8.0.33</version>
```



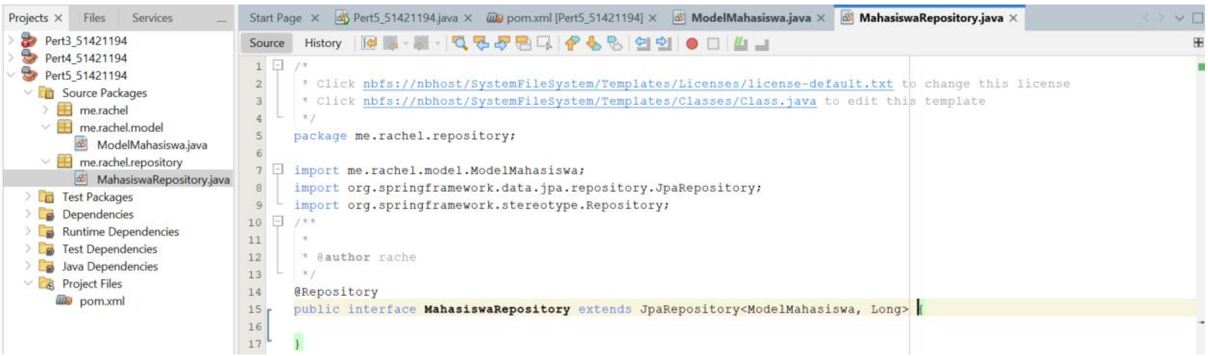
```
--- surefire:3.2.5:test (default-test) @ Pert5_51421194 ---
No tests to run.

--- jar:3.4.2:jar (default-jar) @ Pert5_51421194 ---
Building jar: C:\Users\rache\OneDrive\Documents\NetBeansProjects\Pert5_51421194\target\Pert5_51421194-1.0-SNAPSHOT.jar

--- spring-boot:3.3.3:repackage (repackage) @ Pert5_51421194 ---
Replacing main artifact C:\Users\rache\OneDrive\Documents\NetBeansProjects\Pert5_51421194\target\Pert5_51421194-1.0-SNAPSHOT.jar w
The original artifact has been renamed to C:\Users\rache\OneDrive\Documents\NetBeansProjects\Pert5_51421194\target\Pert5_51421194-

--- install:3.1.3:install (default-install) @ Pert5_51421194 ---
Installing C:\Users\rache\OneDrive\Documents\NetBeansProjects\Pert5_51421194\pom.xml to C:\Users\rache\.m2\repository\me\rachel\Pe
Installing C:\Users\rache\OneDrive\Documents\NetBeansProjects\Pert5_51421194\target\Pert5_51421194-1.0-SNAPSHOT.jar to C:\Users\ra
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  35.534 s
Finished at: 2024-11-11T13:56:24+07:00
------------------------------------------------------------------------
```

ModelMahasiswa



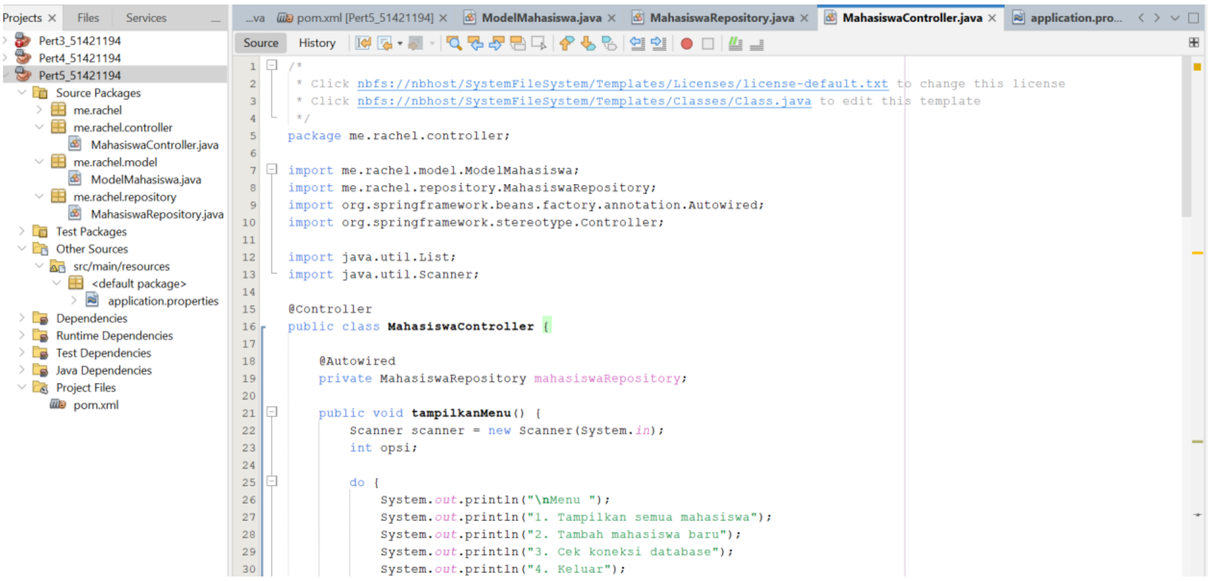
```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package me.rachel.model;

import jakarta.persistence.*;
/**
 *
 * @author rache
 */

@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;
```

MahasiswaRepository

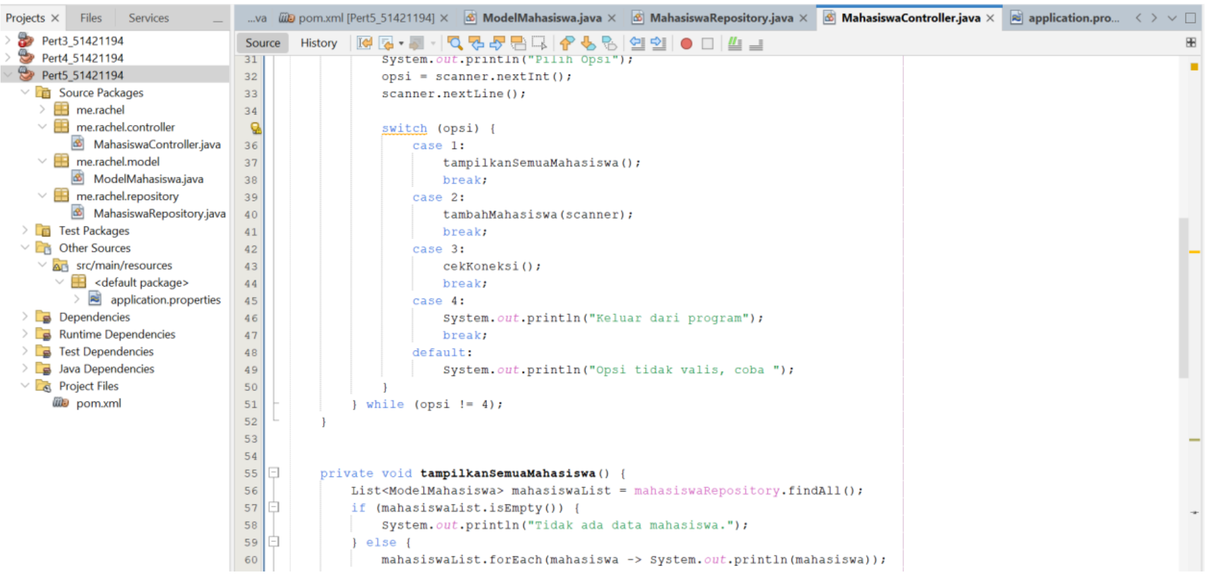


```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package me.rachel.repository;

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

}
```

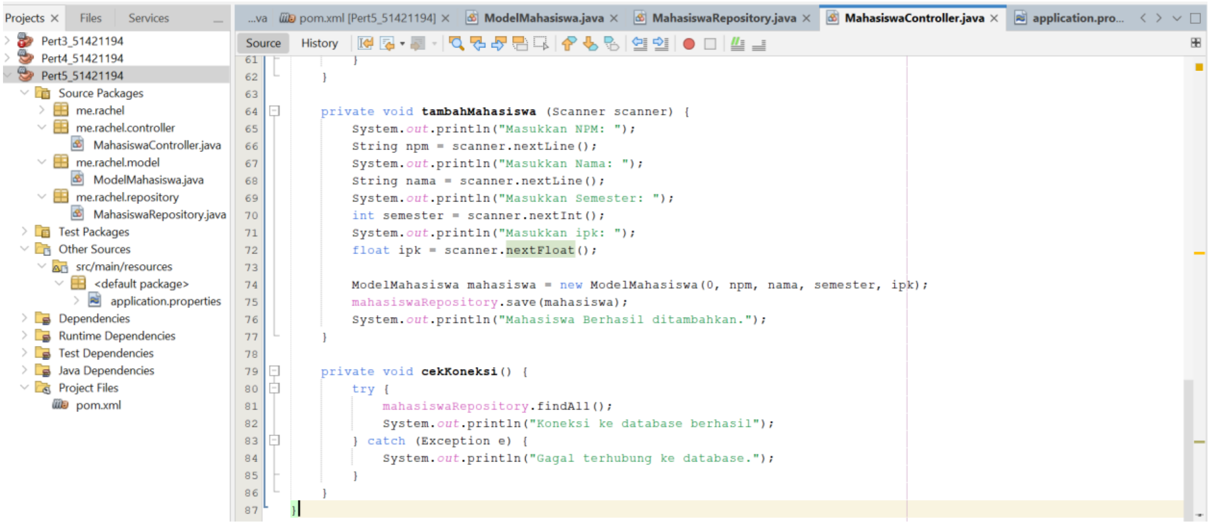
MahasiswaController



```
/*
 * 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.rachel.controller;

import me.rachel.model.ModelMahasiswa;
import me.rachel.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");
```

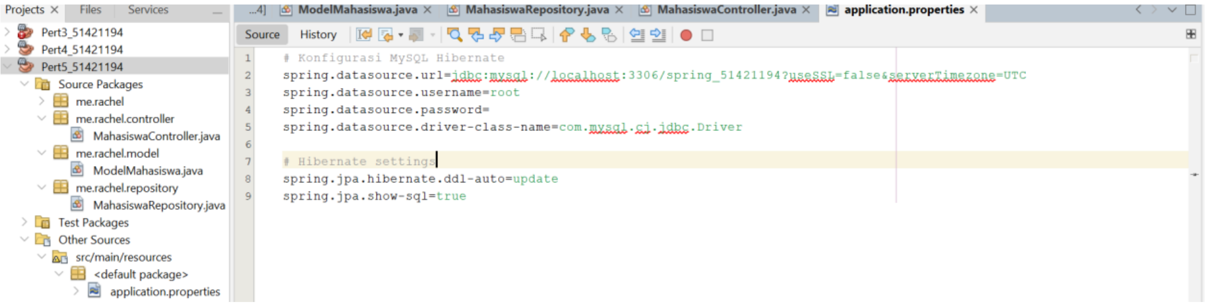


```java
            System.out.println("Pilih Opsi");
            opsi = scanner.nextInt();
            scanner.nextLine();

            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 valis, coba ");
            }
        } while (opsi != 4);
    }


    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));
```



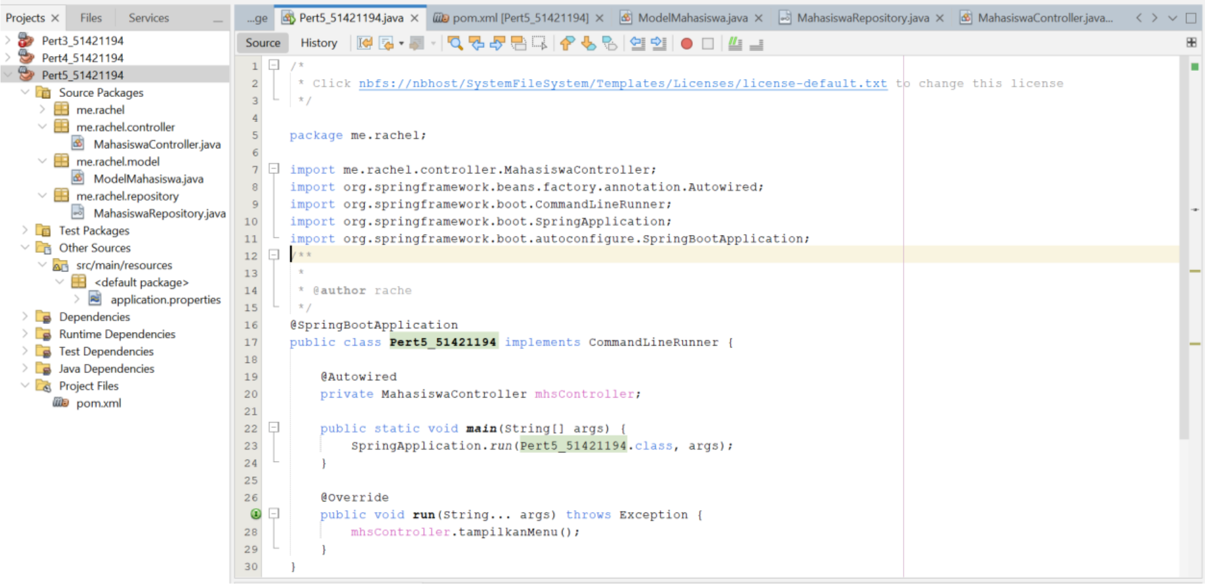
```java
        }
    }

    private void tambahMahasiswa (Scanner scanner) {
        System.out.println("Masukkan NPM: ");
        String npm = scanner.nextLine();
        System.out.println("Masukkan Nama: ");
        String nama = scanner.nextLine();
        System.out.println("Masukkan Semester: ");
        int semester = scanner.nextInt();
        System.out.println("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.");
        }
    }
}
```

Application.properties



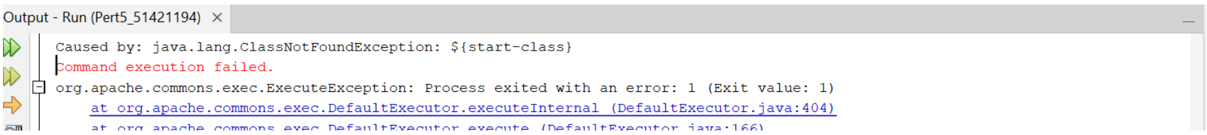
```properties
# Konfigurasi MySQL Hibernate
spring.datasource.url=jdbc:mysql://localhost:3306/spring_51421194?useSSL=false&serverTimezone=UTC
spring.datasource.username=root
spring.datasource.password=
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

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

Default file (Pert5_51421194)



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

package me.rachel;

import me.rachel.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 rache
 */
@SpringBootApplication
public class Pert5_51421194 implements CommandLineRunner {

    @Autowired
    private MahasiswaController mhsController;

    public static void main(String[] args) {
        SpringApplication.run(Pert5_51421194.class, args);
    }

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

Output



```
Caused by: java.lang.ClassNotFoundException: ${start-class}
Command execution failed.
org.apache.commons.exec.ExecuteException: Process exited with an error: 1 (Exit value: 1)
    at org.apache.commons.exec.DefaultExecutor.executeInternal (DefaultExecutor.java:404)
    at org.apache.commons.exec.DefaultExecutor.execute (DefaultExecutor.java:166)
```