

Nama : Herdiansyah Suhendar

Kelas : 4IA17

NPM : 50421604

## **Activity M5 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 teknik dalam pengembangan perangkat lunak di mana suatu objek menerima objek lain yang dibutuhkannya (dependency) dari luar alih-alih menciptakannya sendiri. Dalam konteks Spring Framework dan Spring Boot, DI memungkinkan komponen atau objek untuk dikelola dan disuntikkan oleh Spring Container secara otomatis, sehingga mempermudah pengaturan dan pengelolaan dependency.

### **Cara Kerja Dependency Injection**

DI di Spring bekerja dengan cara *inversion of control* (IoC) di mana framework mengatur alur kontrol objek dan dependency-nya. Proses ini memungkinkan Spring untuk membuat objek, mengelola siklus hidupnya, dan menginisialisasi dependency yang dibutuhkan. DI biasanya diterapkan melalui tiga cara:

1. **Constructor Injection:** Dependency diberikan melalui konstruktor.
2. **Setter Injection:** Dependency diberikan melalui metode setter.
3. **Field Injection:** Dependency diberikan langsung melalui anotasi pada field, seperti `@Autowired`.

### **Mengapa Dependency Injection Penting dalam Spring Boot?**

1. **Mendukung Modularitas:** DI memungkinkan pemisahan kode ke dalam modul-modul yang lebih kecil. Setiap modul bisa memiliki dependensinya sendiri, yang disuntikkan oleh Spring. Ini membuat kode lebih mudah di-maintain dan di-reuse.
2. **Memudahkan Testing:** Dengan DI, komponen-komponen tidak bergantung pada instance tertentu. Ini memungkinkan developer untuk menyuntikkan *mock object* saat pengujian, sehingga pengujian unit lebih mudah dilakukan tanpa memerlukan ketergantungan aktual.
3. **Mengurangi Tight Coupling:** Dengan dependency disuntikkan dari luar, komponen-komponen dalam aplikasi menjadi loosely coupled (tidak tergantung secara langsung). Ini membuat komponen lebih fleksibel, mudah diubah, dan dapat digunakan ulang.
4. **Pengelolaan Siklus Hidup Objek yang Efisien:** Spring bertanggung jawab untuk mengelola siklus hidup objek dalam aplikasi, termasuk saat membuat, menginisialisasi, dan menghancurkannya. Ini mengurangi beban developer untuk mengelola siklus hidup objek secara manual.

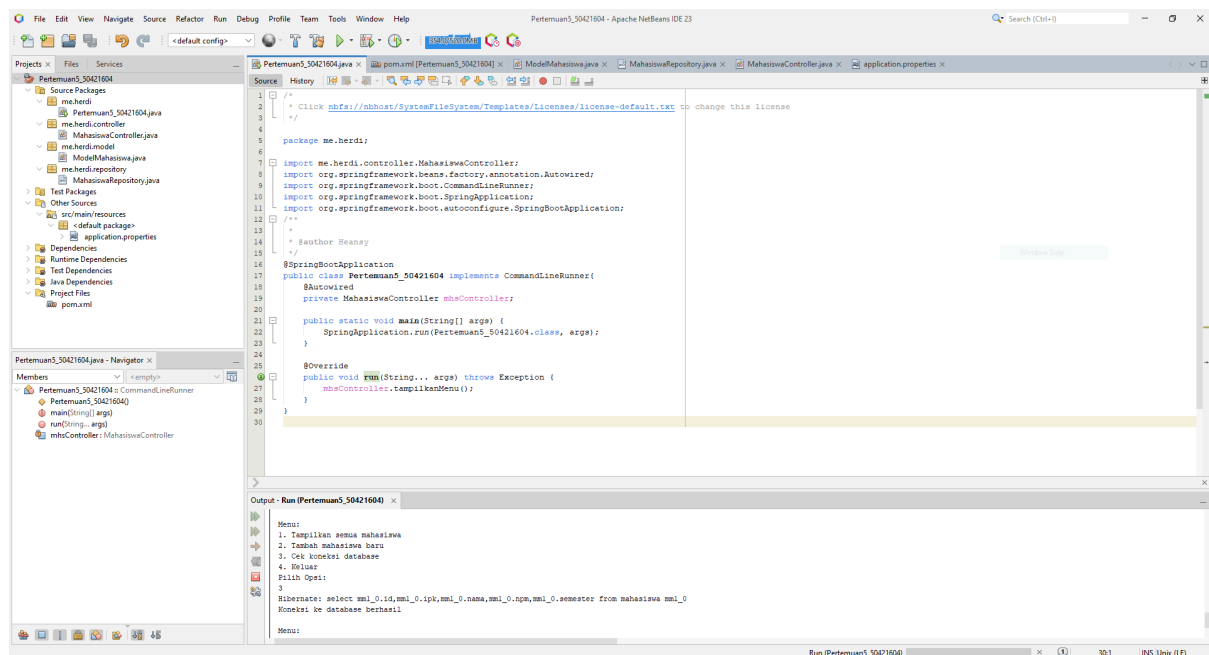
5. **Dukungan Konfigurasi yang Lebih Baik:** DI memungkinkan konfigurasi berbasis anotasi atau XML di Spring File Boot, yang memungkinkan untuk mengatur dependency dan komponen dengan mudah dalam satu tempat.

Dependency Injection sangat penting karena memudahkan pengembangan aplikasi yang mudah di-maintain, scalable, dan memiliki struktur yang jelas dan terorganisir, serta mengikuti prinsip *solid programming* dalam OOP.

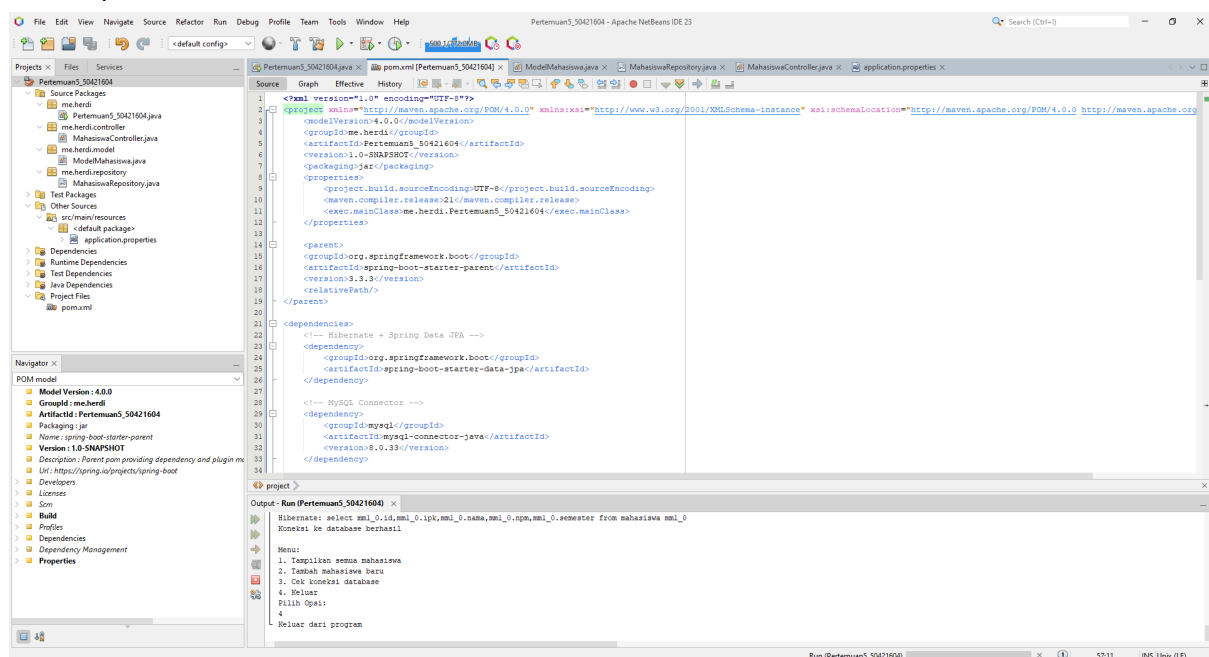
## 2. Screenshot kode dan output program!

Source code:

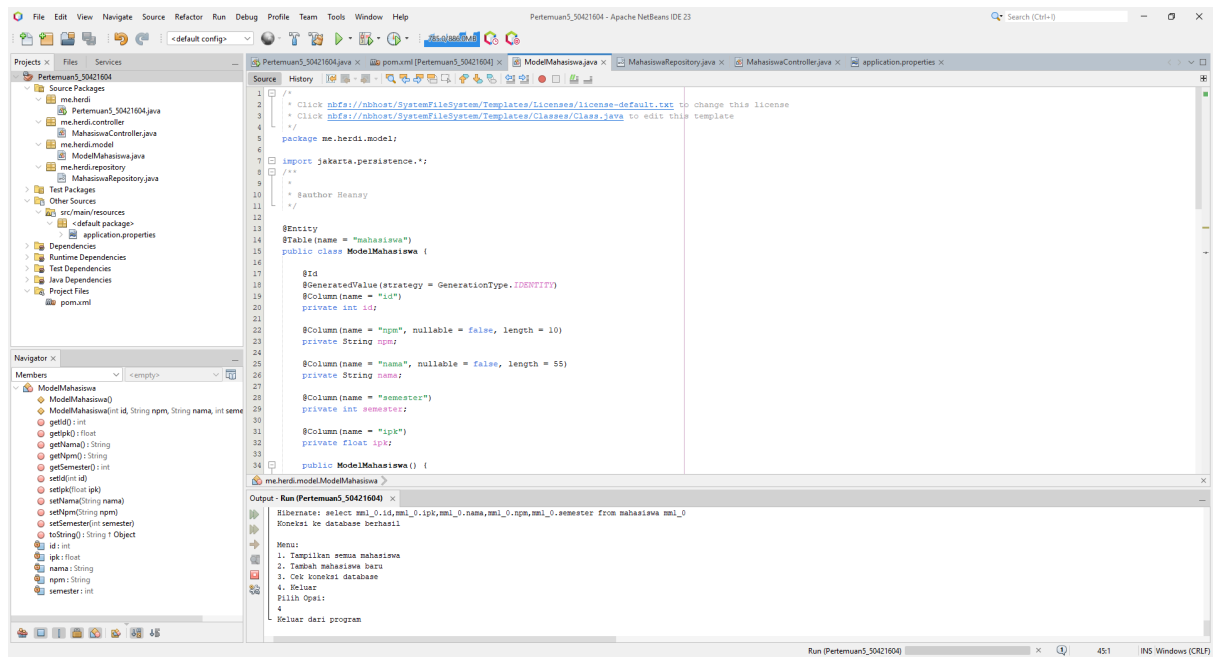
- Pertemuan5\_50421604.java



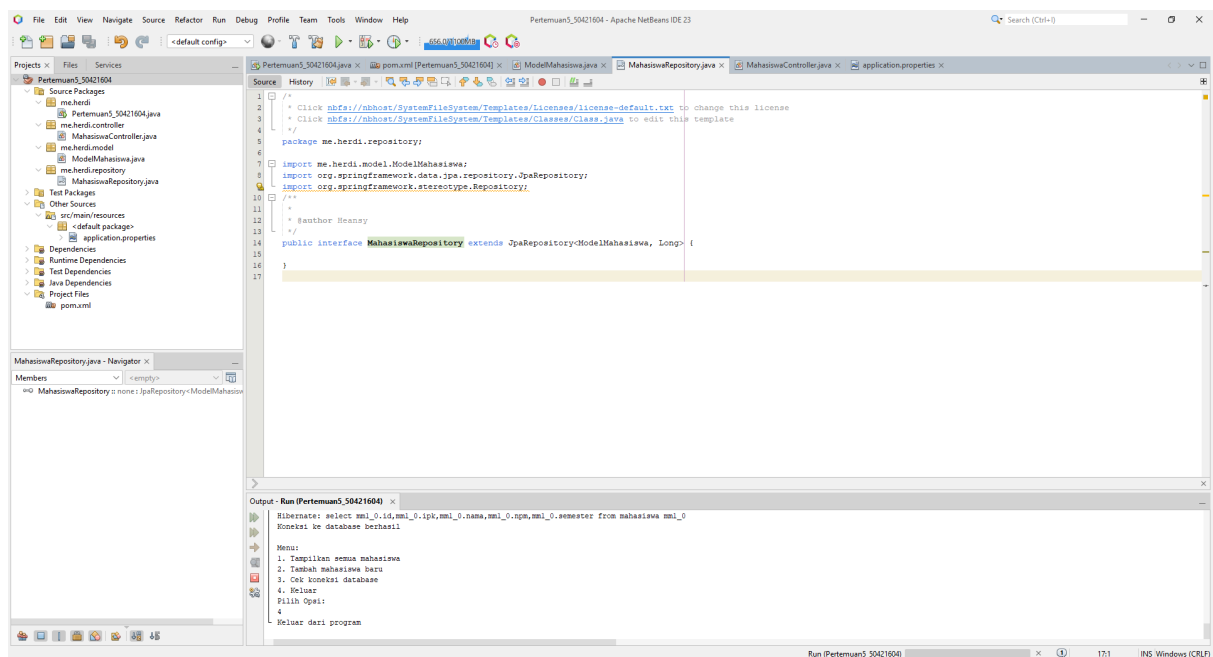
- pom.xml



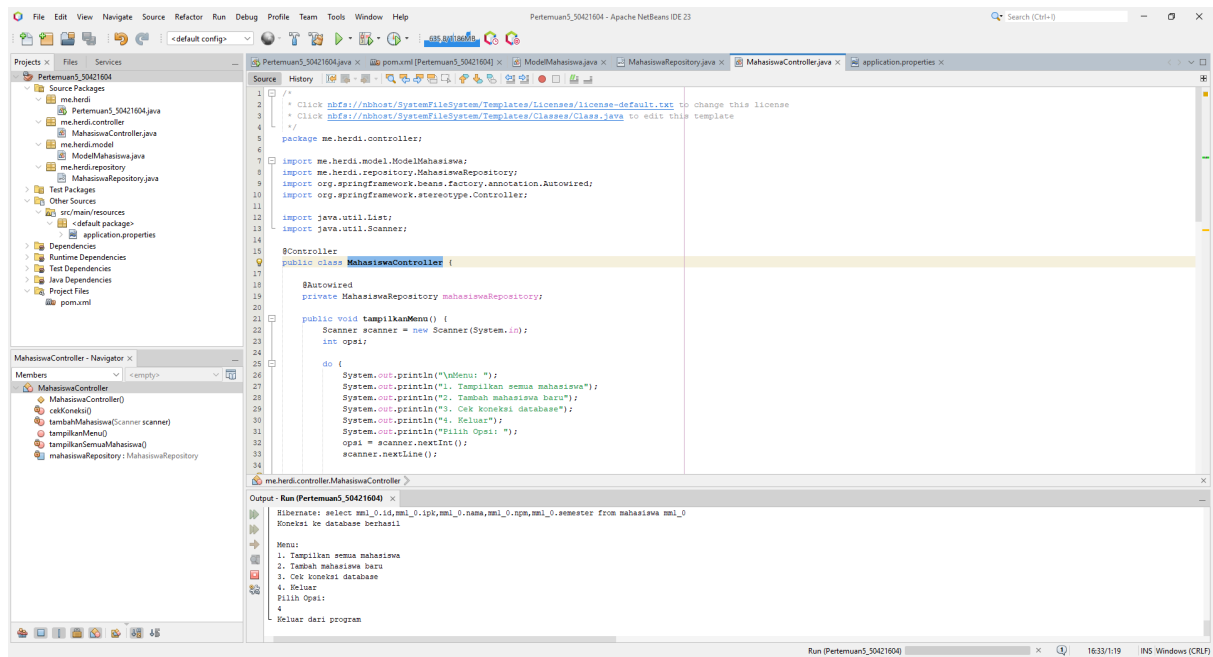
## ● ModelMahasiswa.java



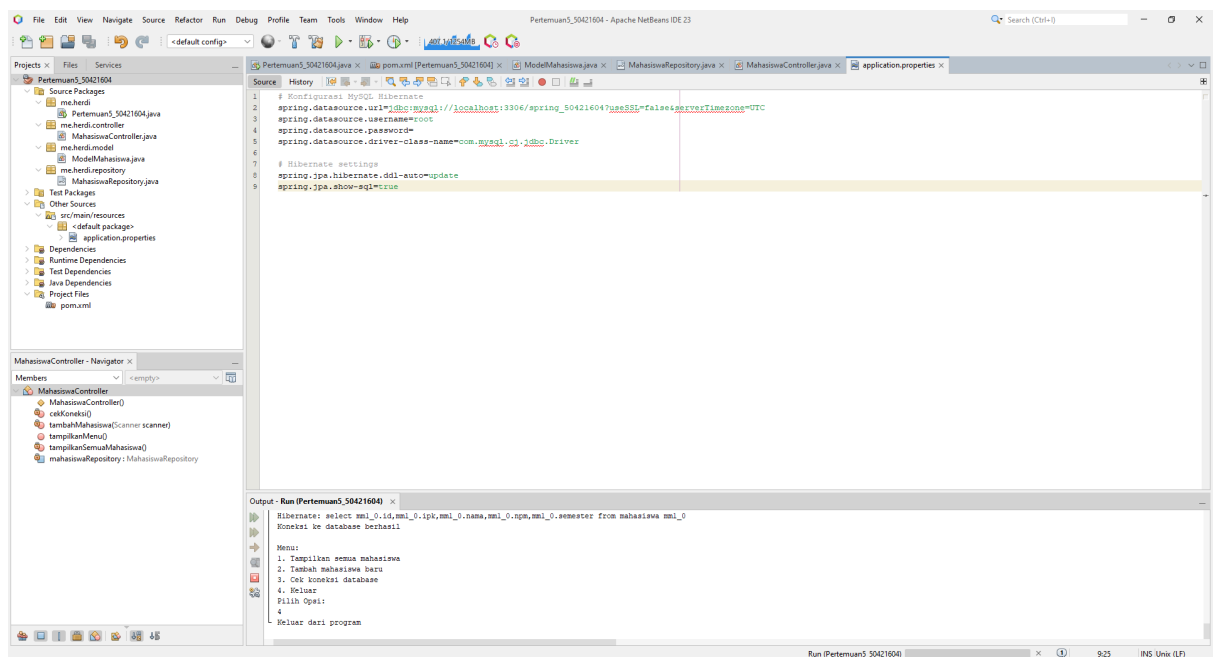
## ● MahasiswaRepository.java



## • MahasiswaController.java



## • application.properties



Output:

