

ACTIVITY PERTEMUAN 5

NAMA : Yolanda Pangemanan

NPM : 51421536

KELAS : 4IA14

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 teknik pemrograman di mana sebuah objek tidak perlu lagi membuat dependensi (objek lain yang dibutuhkannya) secara langsung. Sebaliknya, dependensi tersebut "disuntikkan" ke dalam objek saat objek tersebut diinstansiasi. Dengan kata lain, alih-alih objek yang secara aktif mencari dependensi yang dibutuhkannya, dependensi tersebut secara pasif diberikan kepada objek.

Mengapa DI Penting dalam Spring Boot?

DI memainkan peran yang sangat krusial dalam pengembangan aplikasi berbasis Spring Boot karena beberapa alasan berikut:

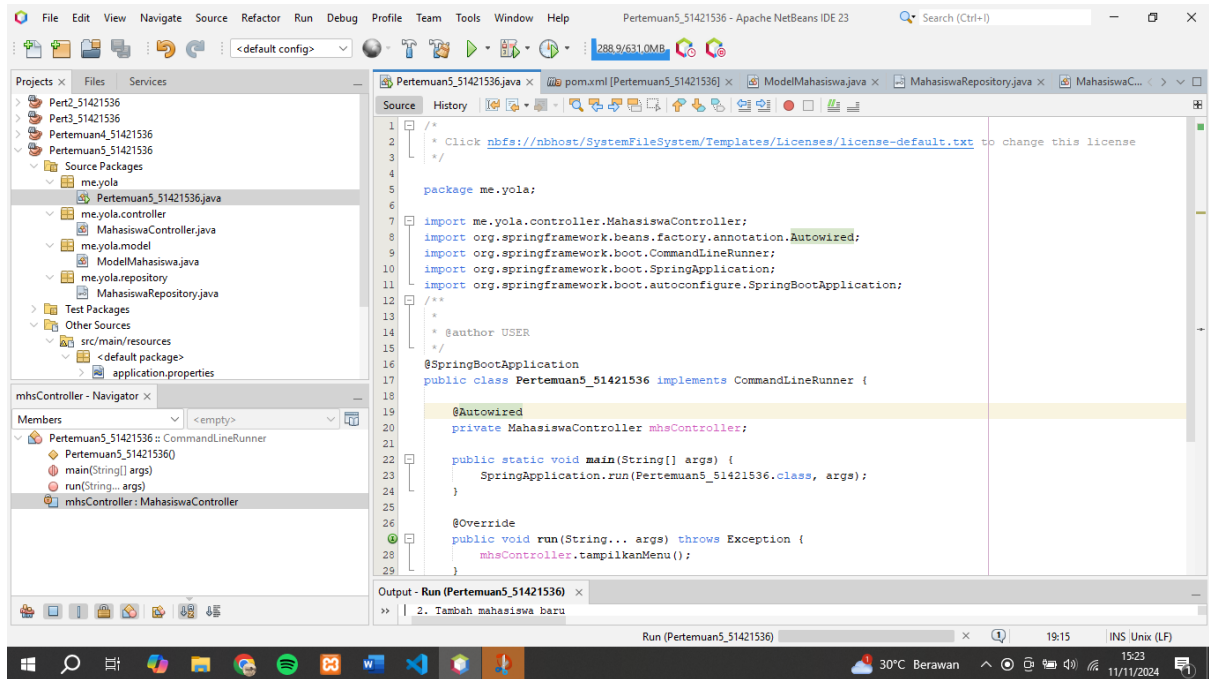
- Loose Coupling, DI membantu menciptakan hubungan yang longgar (loose coupling) antara kelas-kelas dalam aplikasi.
- Testabilitas, dapat lebih mudah menguji unit kelas-kelas dalam aplikasi dengan dapat menyuntikkan mock object atau stub untuk menggantikan dependensi yang sulit diuji atau memiliki efek samping.
- Reusabilitas, memungkinkan untuk membuat komponen-komponen yang lebih reusable.
- Keterbacaan Kode, kode yang menggunakan DI cenderung lebih mudah dibaca dan dipahami.
- Konfigurasi, memungkinkan untuk mengkonfigurasi aplikasi Anda secara fleksibel.

2. **Screenshot kode dan output program!**

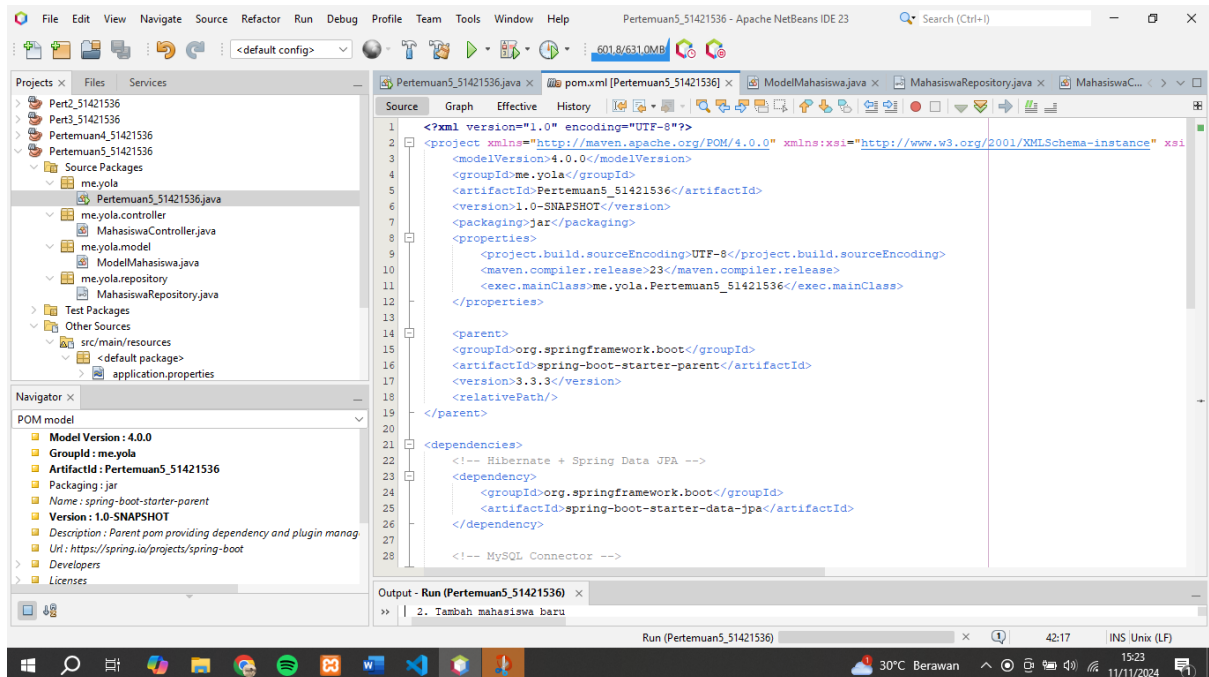
Jawab:

Program

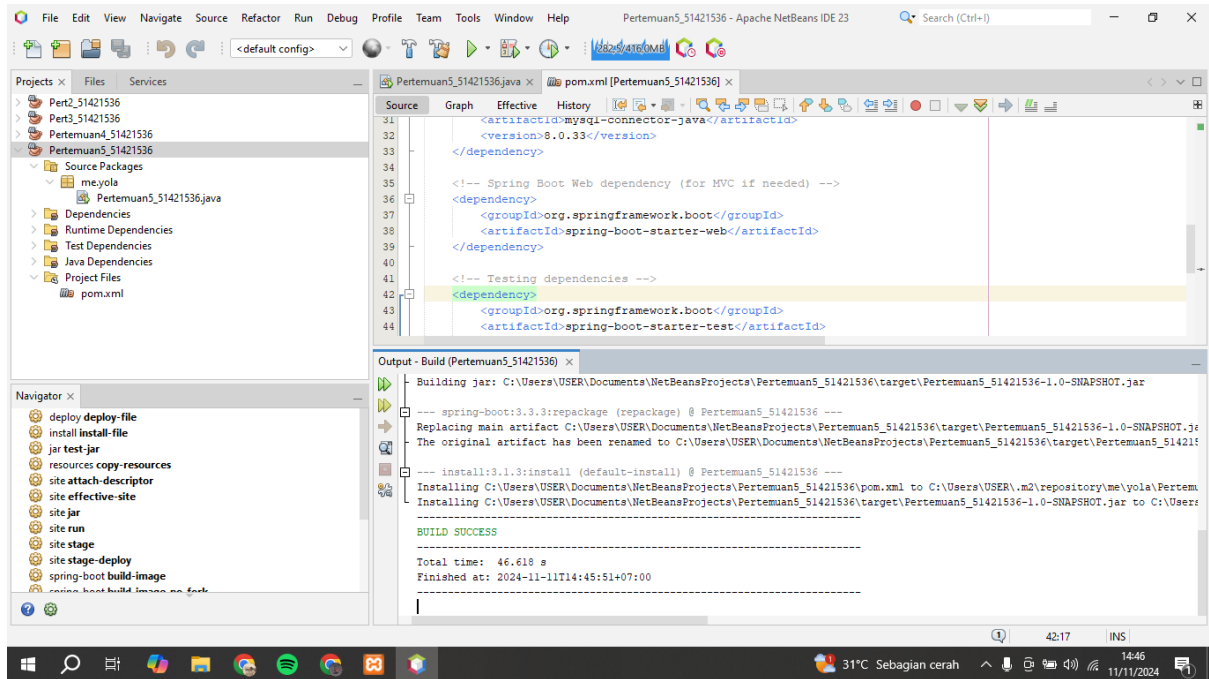
Pertemuan5_51421536.java



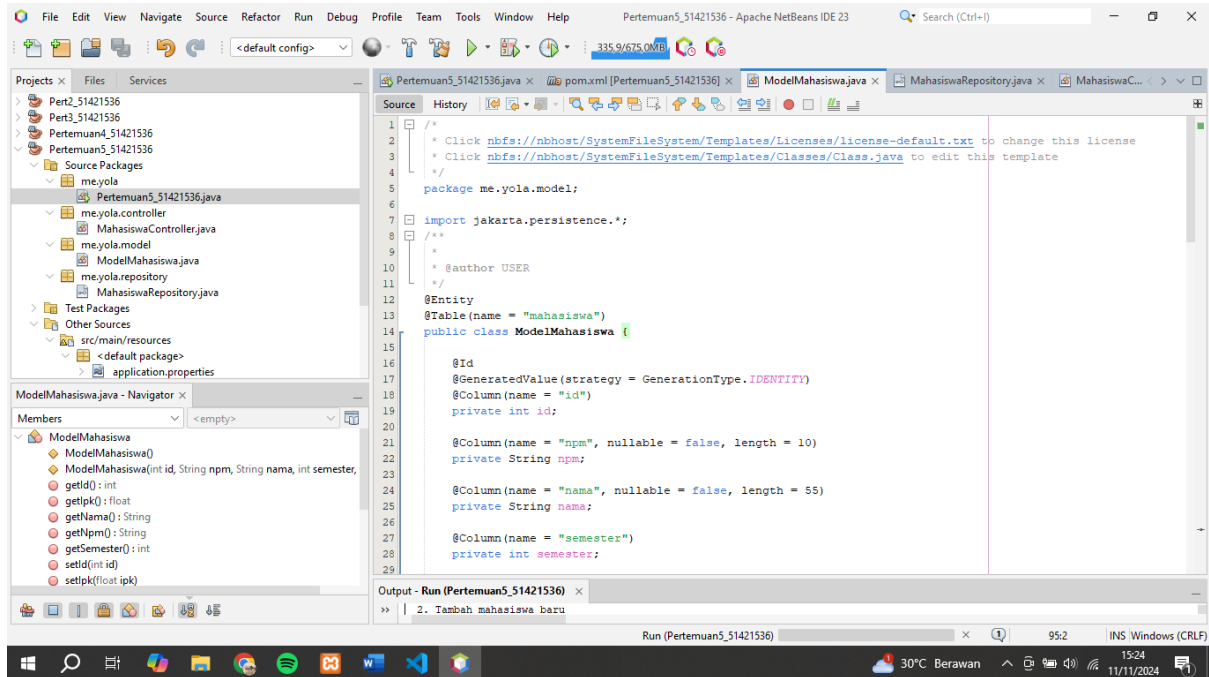
Pom.xml



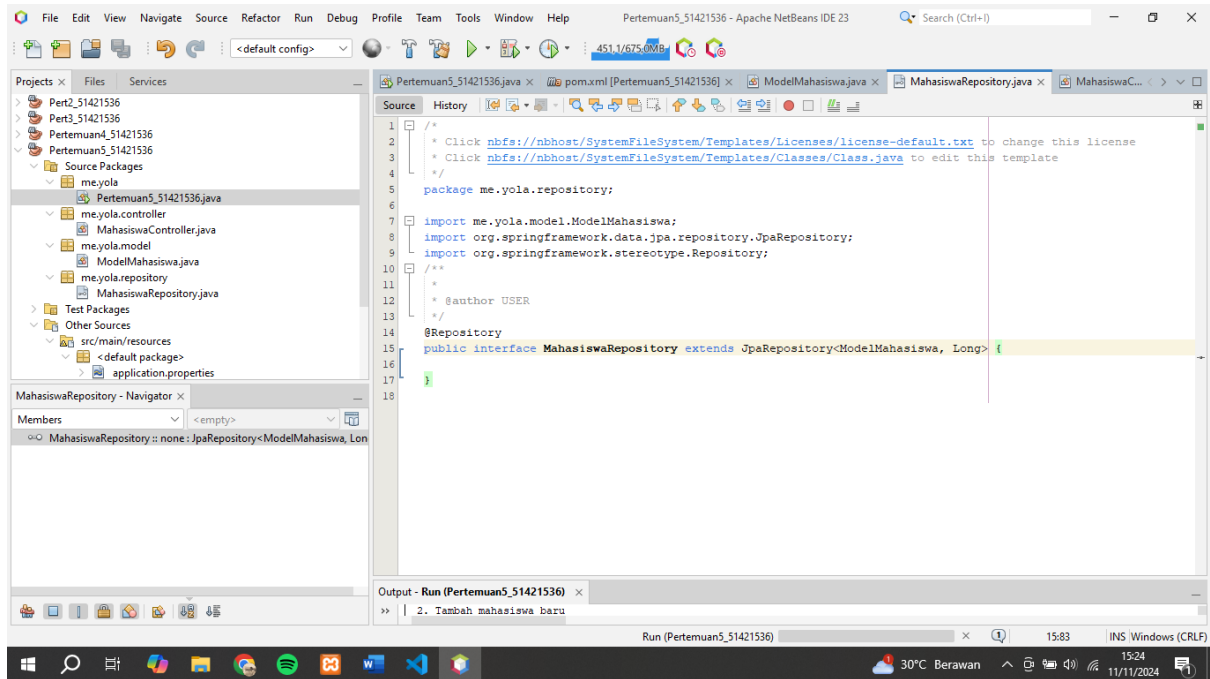
Lalu diclean and build



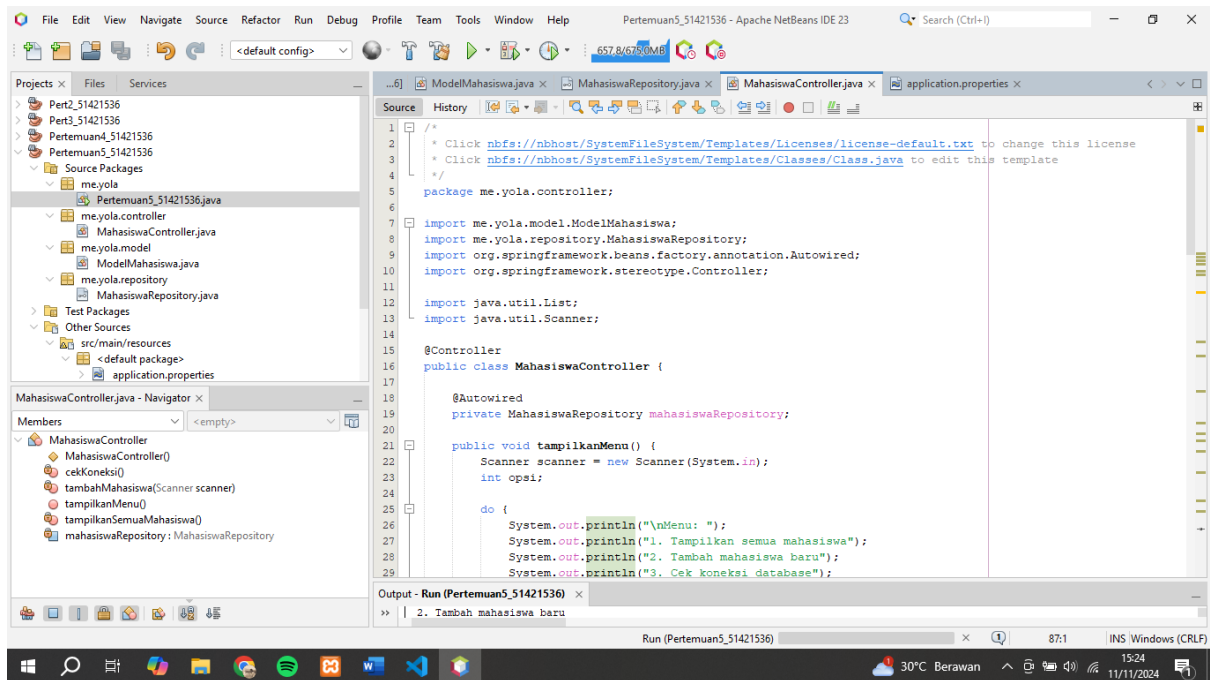
ModelMahasiswa.java



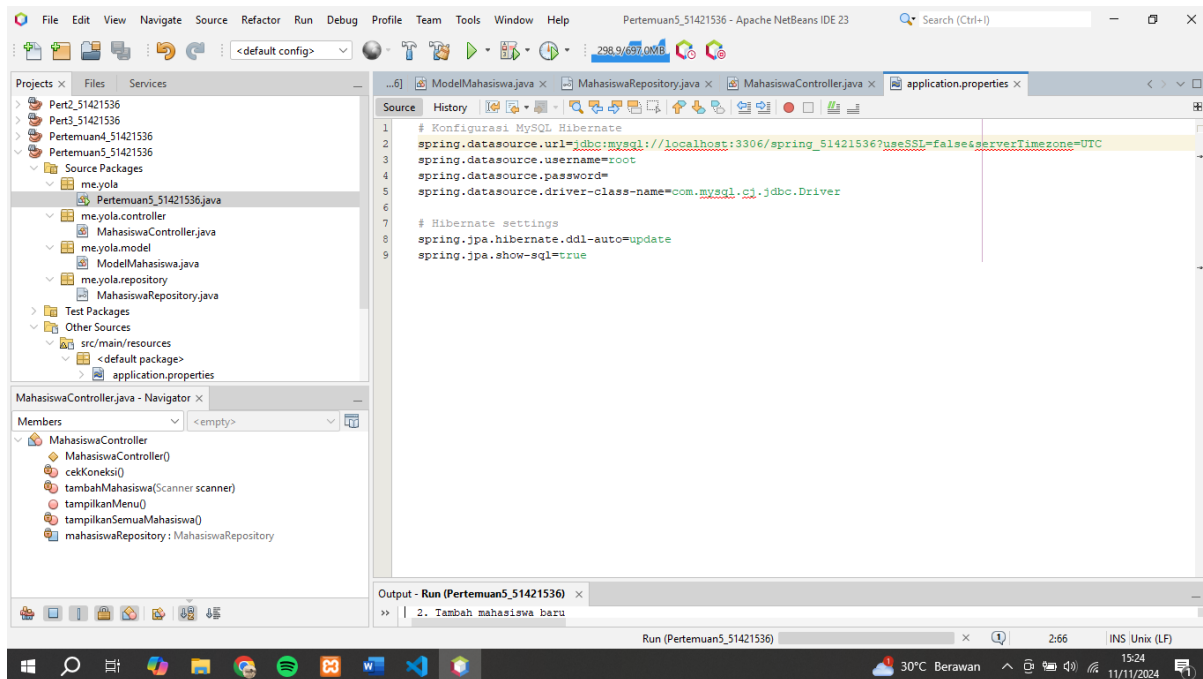
MahasiswaRepository.java



MahasiswaController.java



Application.properties



Output

Tambah Mahasiswa Baru

```
Output - Run (Pertemuan5_51421536) x
2024-11-11T15:15:43.868+07:00 INFO 13196 --- [          main] me.yola.Pertemuan5_51421536

Menu:
1. Tampilkan semua mahasiswa
2. Tambah mahasiswa baru
3. Cek koneksi database
4. Keluar
Pilih Opsi:
2
Masukkan NPM:
51421536
Masukkan Nama:
yolanda
Masukkan Semester:
7
Masukkan IPK:
4
Hibernate: insert into mahasiswa (ipk,nama,npm,semester) values (?, ?, ?, ?)
Mahasiswa Berhasil ditambahkan.
```

Tampilkan semua mahasiswa

```
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, npm='51421536', nama='yolanda', semester=7, ipk='4.0')
```

Cek Koneksi database

```
Menu:
1. Tampilkan semua mahasiswa
2. Tambah mahasiswa baru
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
```

Keluar dari program

Menu:

1. Tampilkan semua mahasiswa
2. Tambah mahasiswa baru
3. Cek koneksi database
4. Keluar

Pilih Opsi:

4

- Keluar dari program