

## ACTIVITY PERTEMUAN 5

**NAMA** : Leonhard Dominikus Adiarsa Fernandez

**NPM** : 50421746

**KELAS** : 3IA14

**MATERI** :

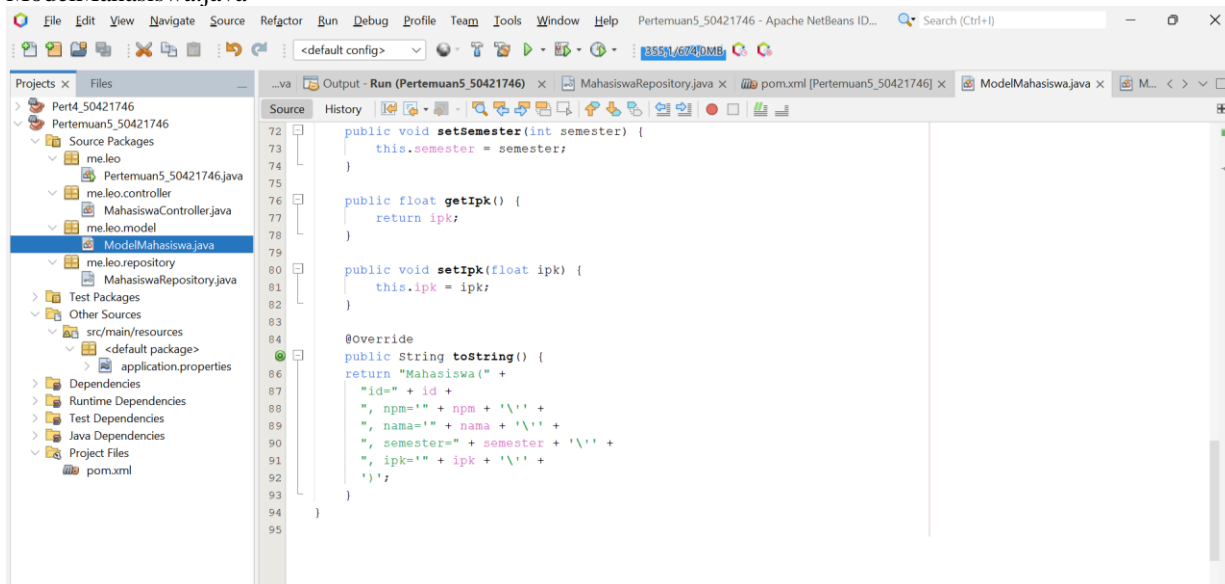
**MATA PRAKTIKUM** : RPL 2

**Dependency Injection adlaah konsep penting dalam framework. Jelaskan apa itu Dependency Injection dan mengapa DI sangat penting dalam pengembangan berbasis spring boot!**

Dependency Injection (DI) adalah salah satu implementasi dari konsep Inversion of Control (IoC). DI berfungsi untuk memungkinkan suatu kelas tidak bergantung secara langsung pada kelas lain, sehingga sebuah kelas dapat menerima dependensi dari luar. Dalam Spring Boot, DI sangat penting karena memudahkan dalam pengelolaan dan pengaturan dependensi, serta meningkatkan keterpisahan antar komponen dalam aplikasi. Dengan menggunakan DI, kode menjadi lebih modular dan mudah diuji, karena dependensi dapat diganti atau disuntikkan sesuai kebutuhan tanpa harus mengubah kode secara langsung.

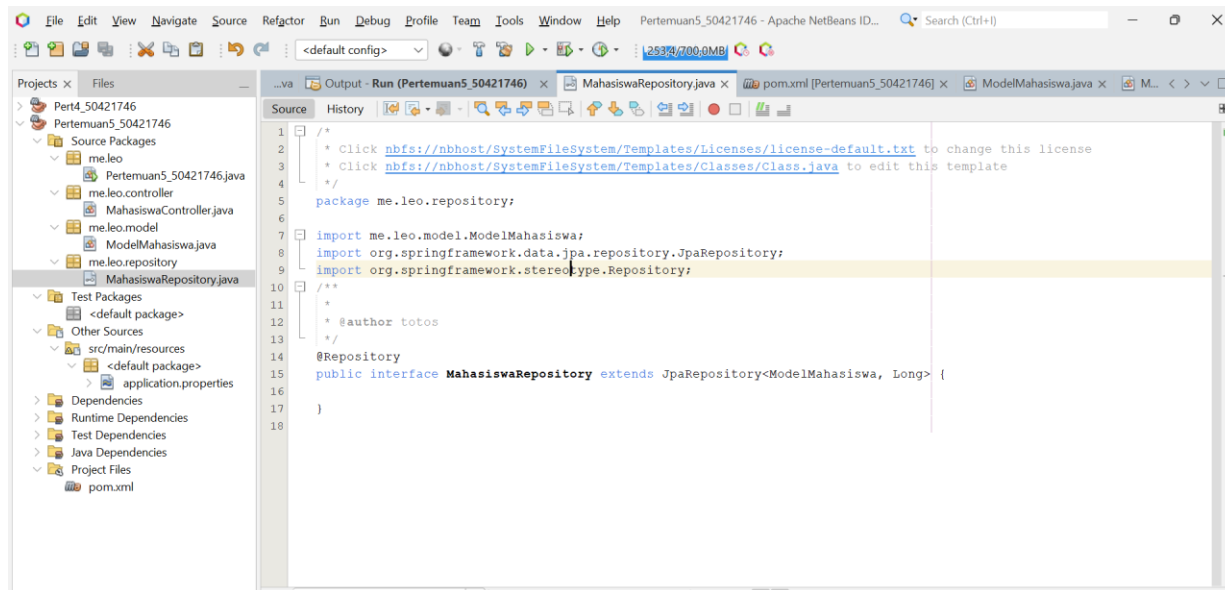
**Screenshot kode dan output program!**

ModelMahasiswa.java

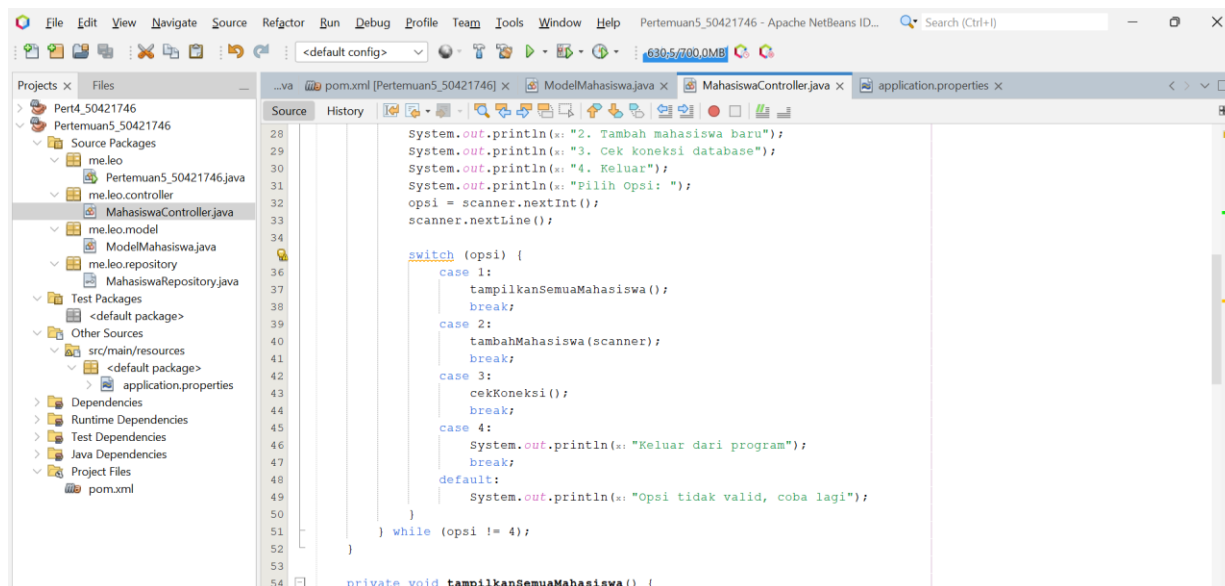
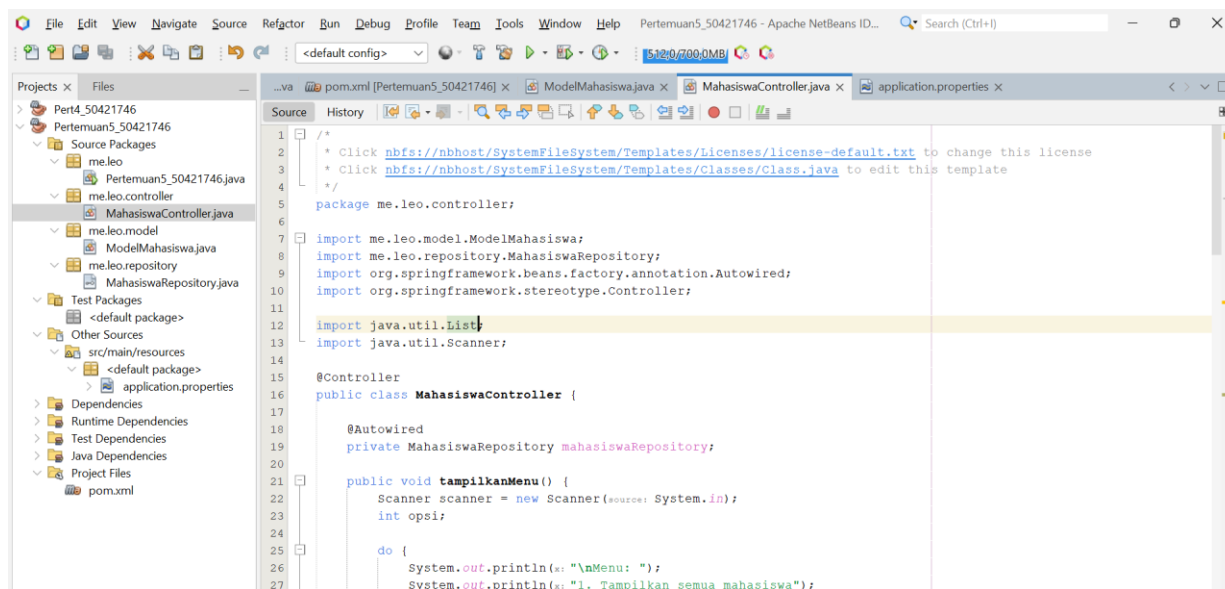


```
72 public void setSemester(int semester) {
73     this.semester = semester;
74 }
75
76 public float getIpk() {
77     return ipk;
78 }
79
80 public void setIpk(float ipk) {
81     this.ipk = ipk;
82 }
83
84 @Override
85 public String toString() {
86     return "Mahasiswa(" +
87         "id=" + id +
88         ", npm=" + npm + '\n' +
89         ", nama=" + nama + '\n' +
90         ", semester=" + semester + '\n' +
91         ", ipk=" + ipk + '\n' +
92         ')';
93 }
94
95 }
```

MahasiswaRepository.java



## MahasiswaController.java



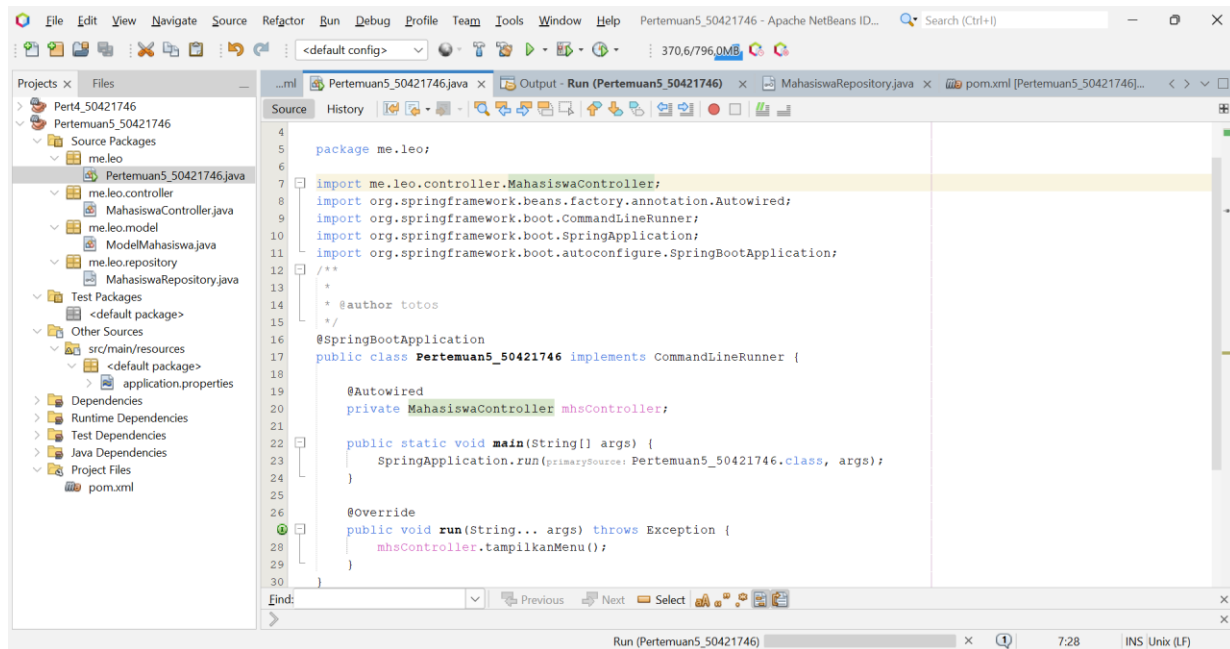
```
55 List<ModelMahasiswa> mahasiswaList = mahasiswaRepository.findAll();
56 if (mahasiswaList.isEmpty()) {
57     System.out.println(x: "Tidak ada data mahasiswa.");
58 } else {
59     mahasiswaList.forEach(mahasiswa -> System.out.println(x: mahasiswa));
60 }
61
62
63 private void tambahMahasiswa(Scanner scanner){
64     System.out.println(x: "Masukkan NPM: ");
65     String npm = scanner.nextLine();
66     System.out.println(x: "Masukkan Nama: ");
67     String nama = scanner.next();
68     System.out.println(x: "Masukkan Semester: ");
69     int semester = scanner.nextInt();
70     System.out.println(x: "Masukkan IPK: ");
71     float ipk = scanner.nextFloat();
72
73     ModelMahasiswa mahasiswa = new ModelMahasiswa(id: 0, npm, nama, semester, ipk);
74     mahasiswaRepository.save(entity: mahasiswa);
75     System.out.println(x: "Mahasiswa Berhasil ditambahkan.");
76 }
77
78 private void cekKoneksi() {
79     try {
80         mahasiswaRepository.findAll();
81         System.out.println(x: "Koneksi ke database berhasil");
82     } catch (Exception e) {
83         System.out.println(x: "Gagal terhubung ke database");
84     }
85 }
86
87
```

Find: onald No matches

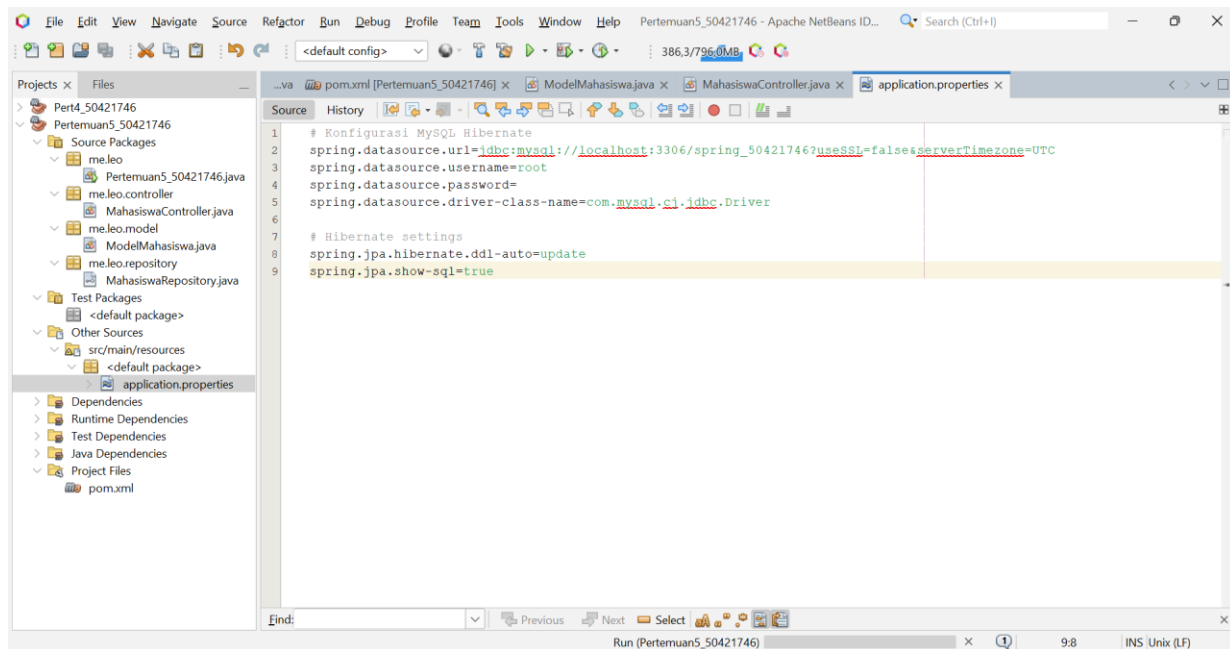
```
70 System.out.println(x: "Masukkan IPK: ");
71 float ipk = scanner.nextFloat();
72
73 ModelMahasiswa mahasiswa = new ModelMahasiswa(id: 0, npm, nama, semester, ipk);
74 mahasiswaRepository.save(entity: mahasiswa);
75 System.out.println(x: "Mahasiswa Berhasil ditambahkan.");
76 }
77
78 private void cekKoneksi() {
79     try {
80         mahasiswaRepository.findAll();
81         System.out.println(x: "Koneksi ke database berhasil");
82     } catch (Exception e) {
83         System.out.println(x: "Gagal terhubung ke database");
84     }
85 }
86
87
```

Find: Previous Next Select

Pertemuan5\_50421746



## Application.properties



Output:

