**LAPORAN AKHIR PRAKTIKUM**

Mata Praktikum : Rekayasa Perangkat Lunak 2

Kelas : 4IA06

Praktikum ke- : 6

Tanggal : 19 November 2024

Materi : Implementasi Aspect Oriented Programming (AOP) dan Dependency Injection pada Project Spring dan Hibernate

NPM : 51421517

Nama : William Devin Septianus Pranggono

Ketua Asisten :

Paraf Asisten :

Nama Asisten : Gilbert Jefferson Faozato Mendrofa

Jumlah Lembar : 15 lembar

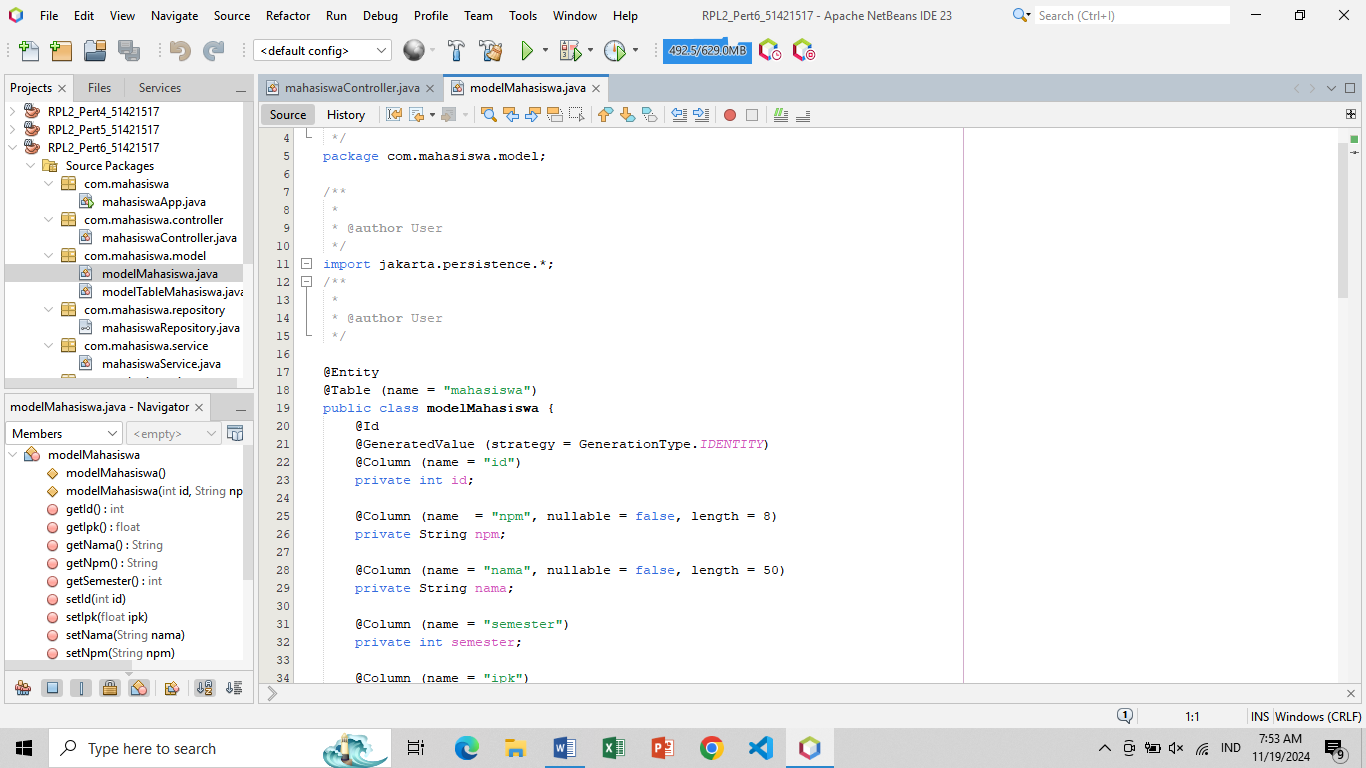
**LABORATORIUM TEKNIK INFORMATIKA**

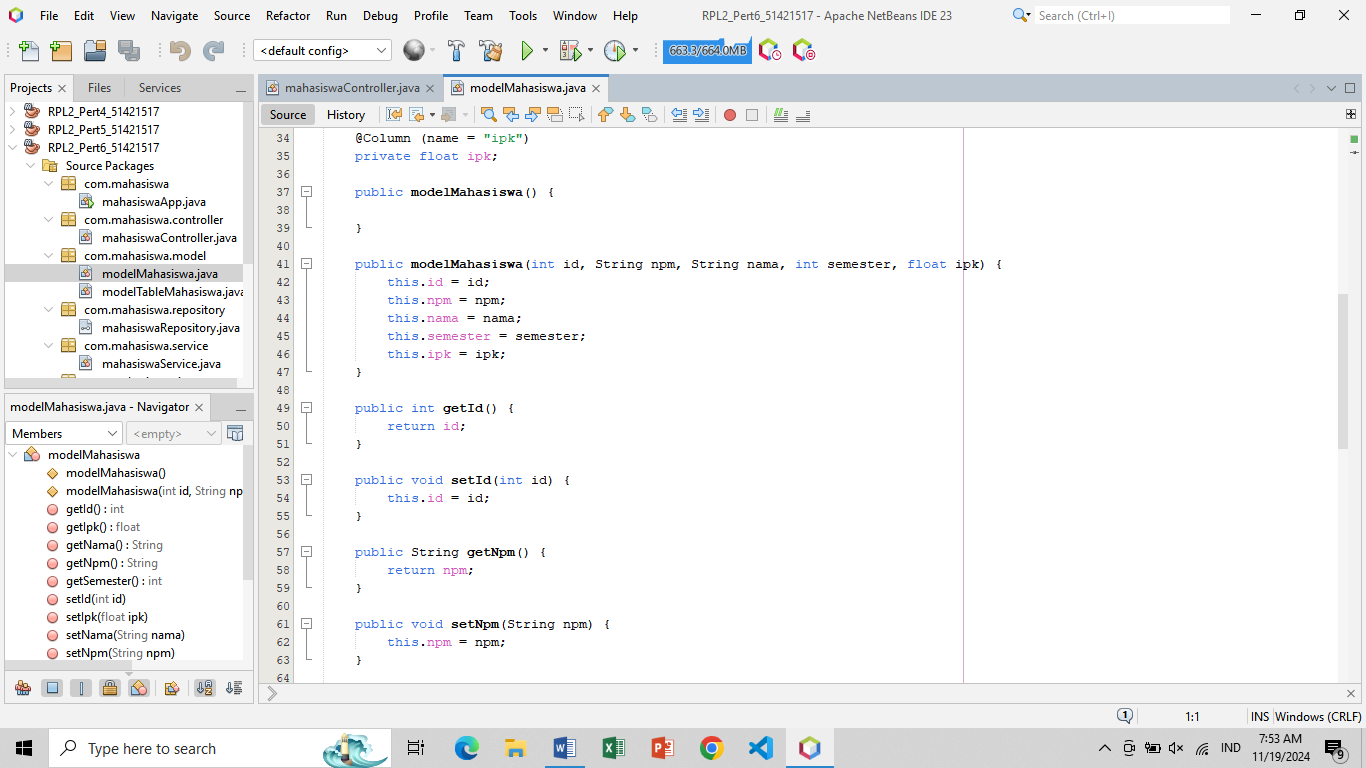
**UNIVERSITAS GUNADARMA**

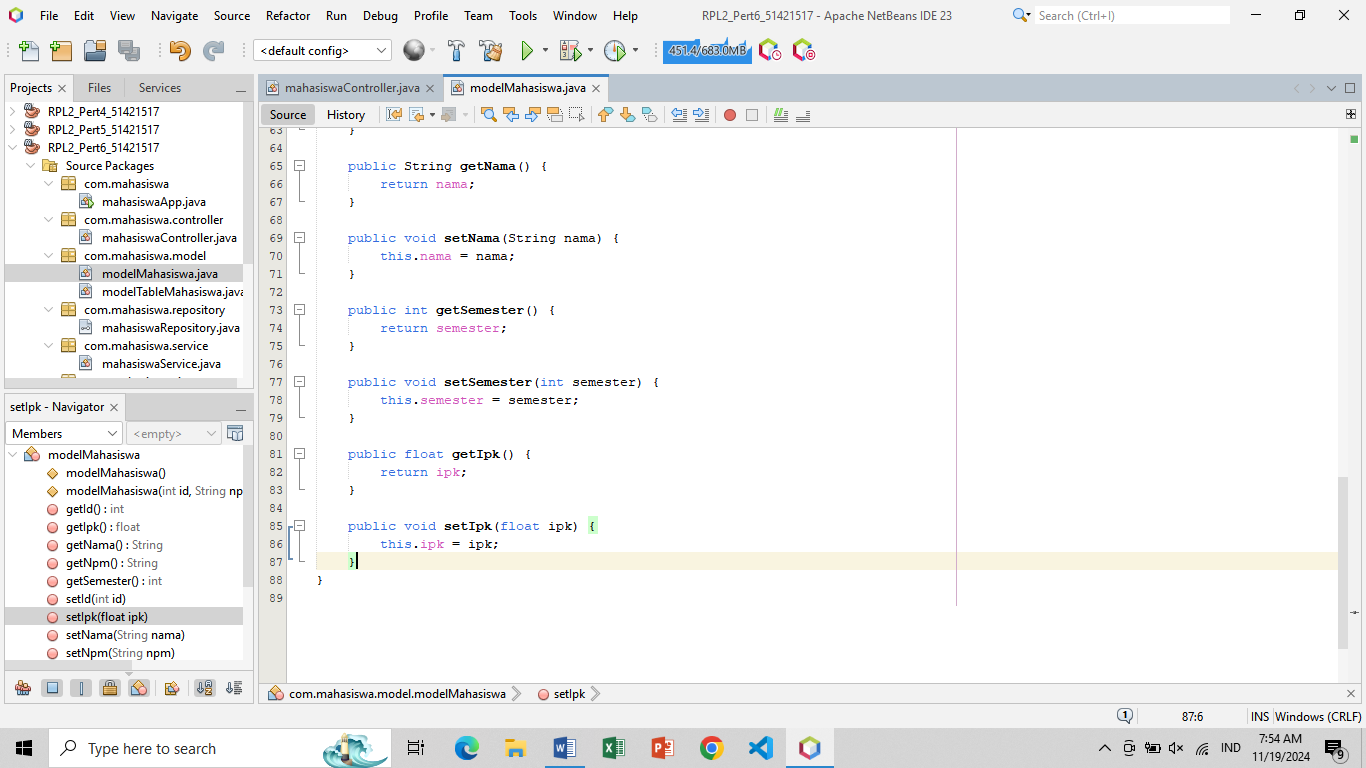
**2024**

**LISTING DAN LOGIKA PROGRAM**

1. modelMahasiswa.java

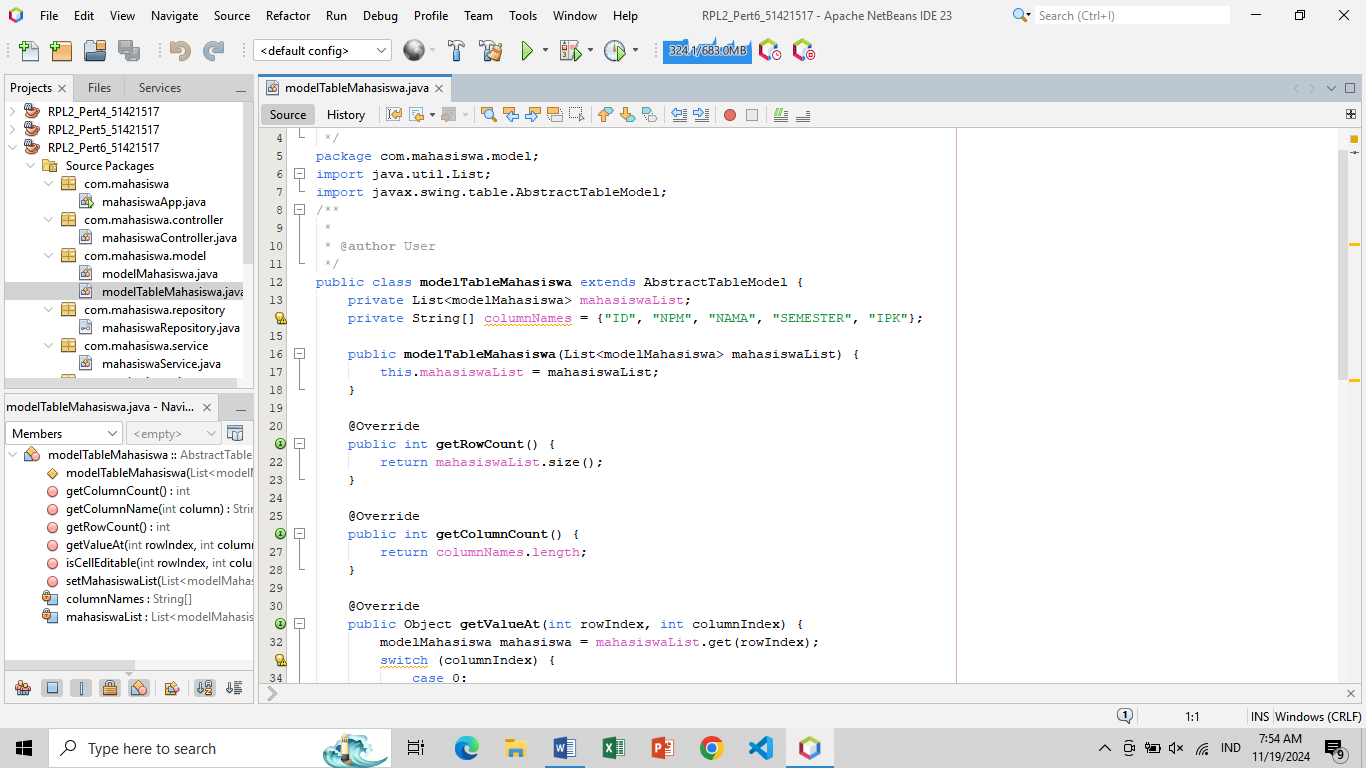


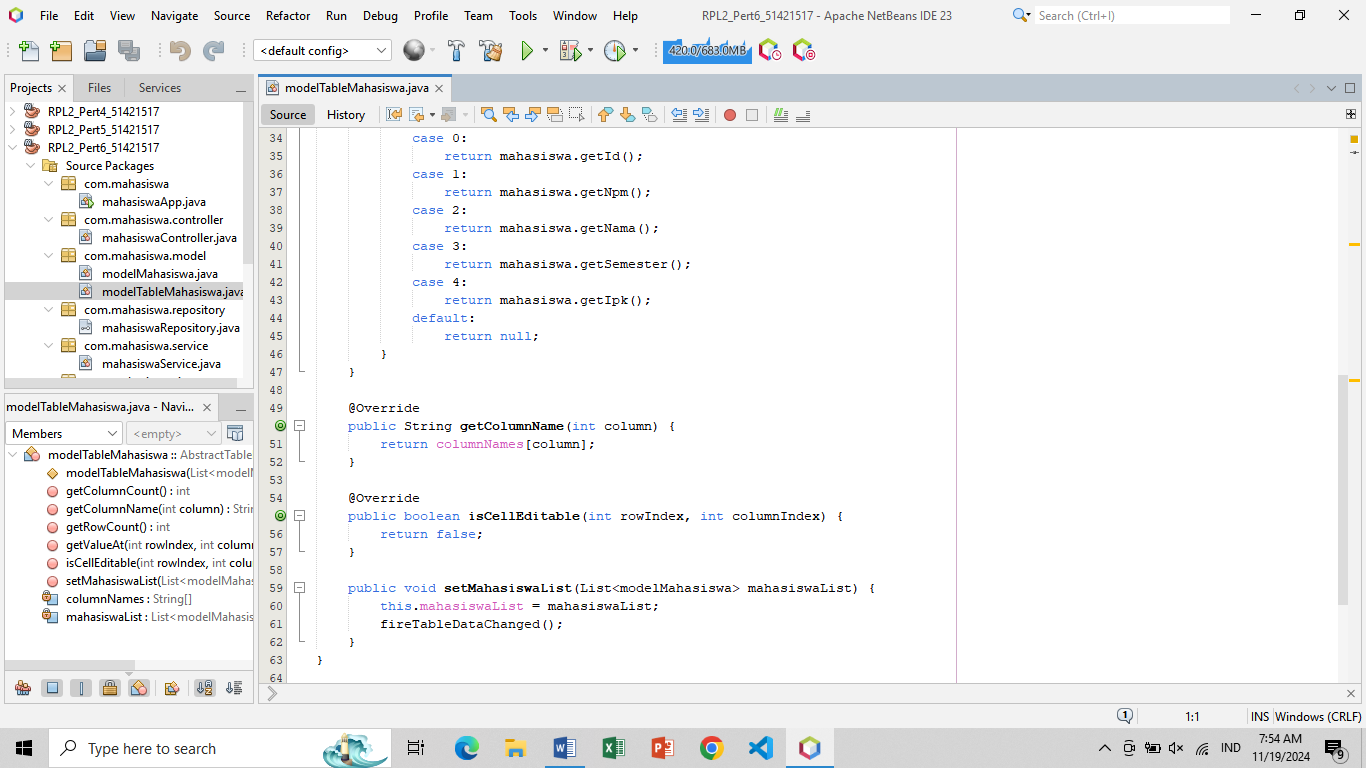




Class modelMahasiswa mendefinisikan struktur dari objek/tabel mahasiswa pada database, yaitu atribut serta tipe data yang dipakai. Karena atribut-atribut tersebut bersifat private, maka diperlukan method getter dan setter untuk membaca dan memanipulasi atribut.

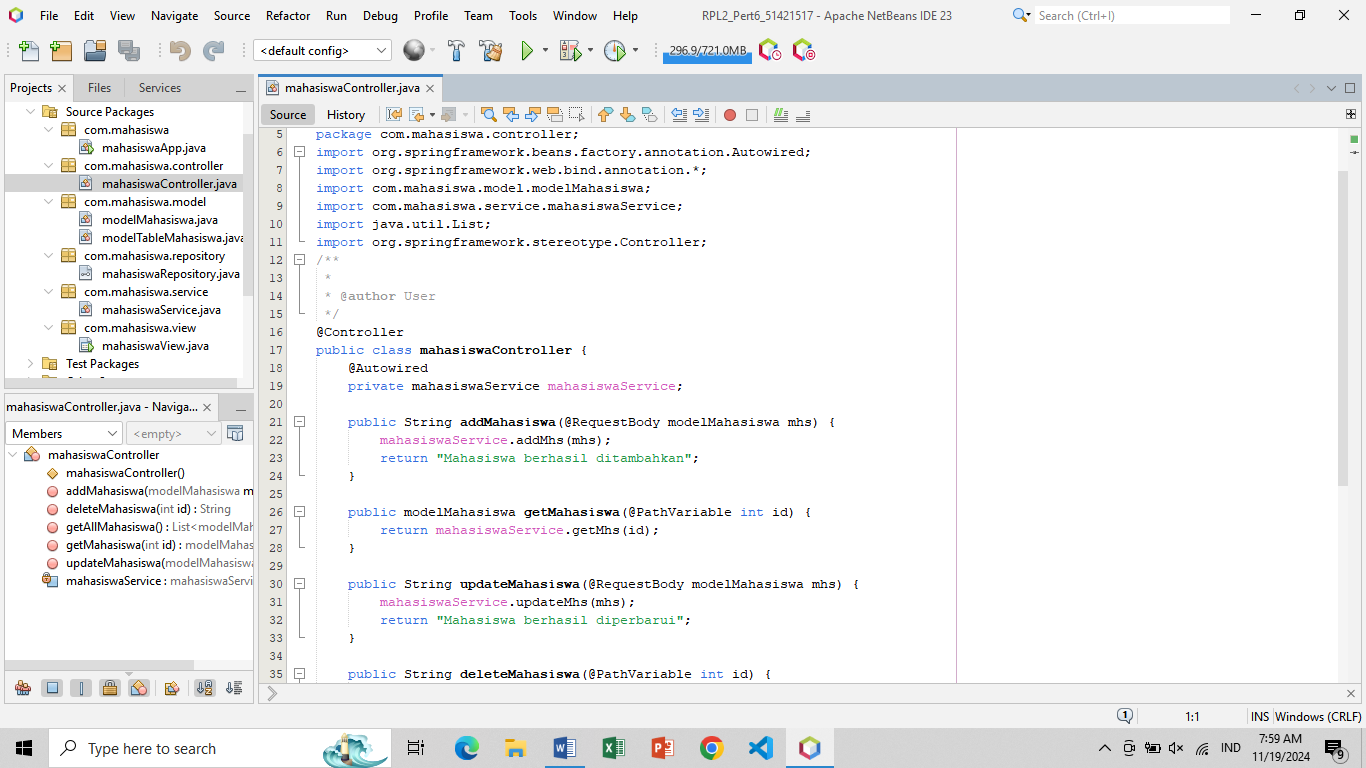
1. modelTableMahasiswa.java

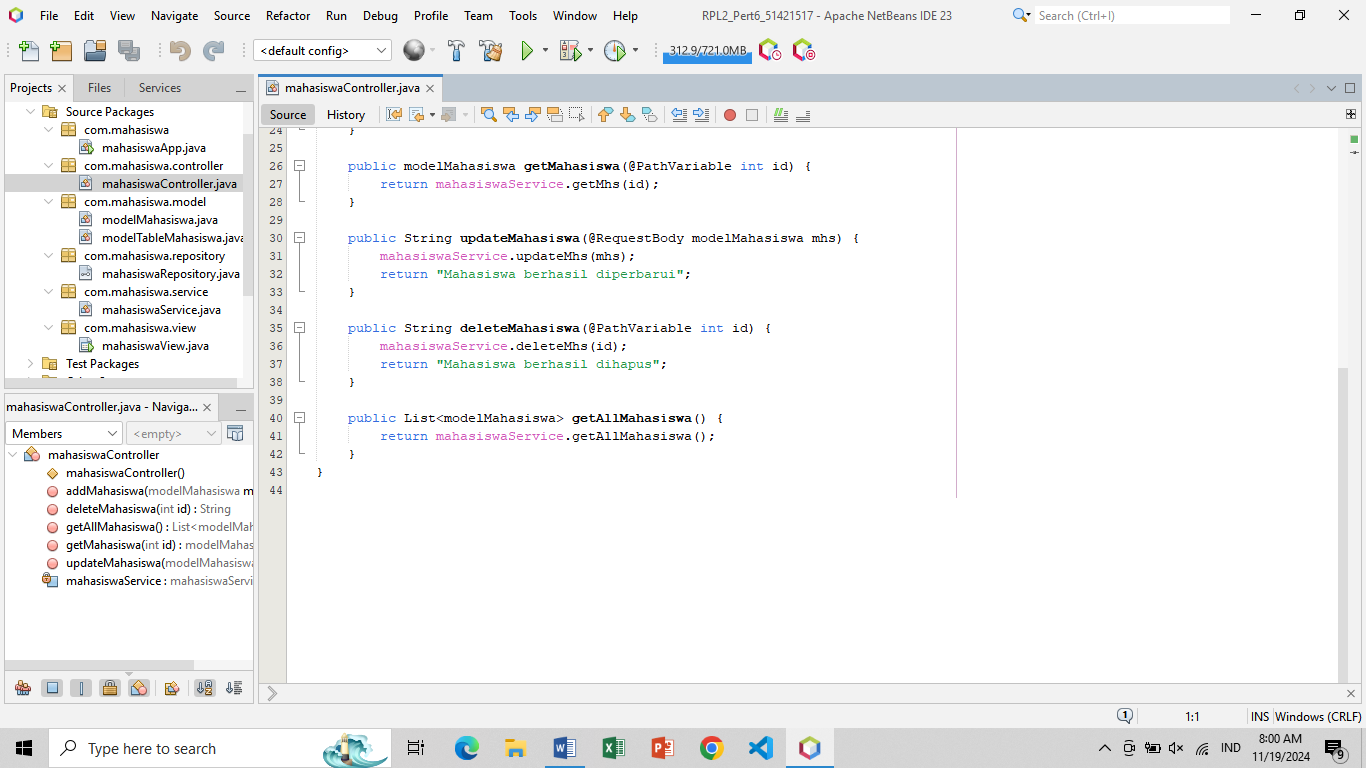




Class modelTableMahasiswa adalah subclass dari superclass AbstractTableModel, model tabel pada package javax.swing. Class ini berfungsi sebagai blueprint dari model tabel yang akan ditampilkan pada Jtable. Struktur tabel adalah List bertipe data modelMahasiswa. Selain itu, terdapat beberapa method superclass yang di-override oleh class ini untuk menyesuaikan dengan tipe data yang digunakan, beserta method setMahasiswaList untuk memodifikasi atribut mahasiswaList milik objek.

1. mahasiswaController.java





Class Mahasiswa Controller dianotasi dengan @Controller, yang berarti merupakan class yang berperan sebagai Controller pada Spring MVC. Objek mahasiswaService akan di-inject secara otomatis dengan bean mahasiswaService.

Method addMahasiswa menerima objek modelMahasiswa dan akan menambahkan objek tersebut ke tabel dengan memanggil method addMhs milik objek mahasiswaService.

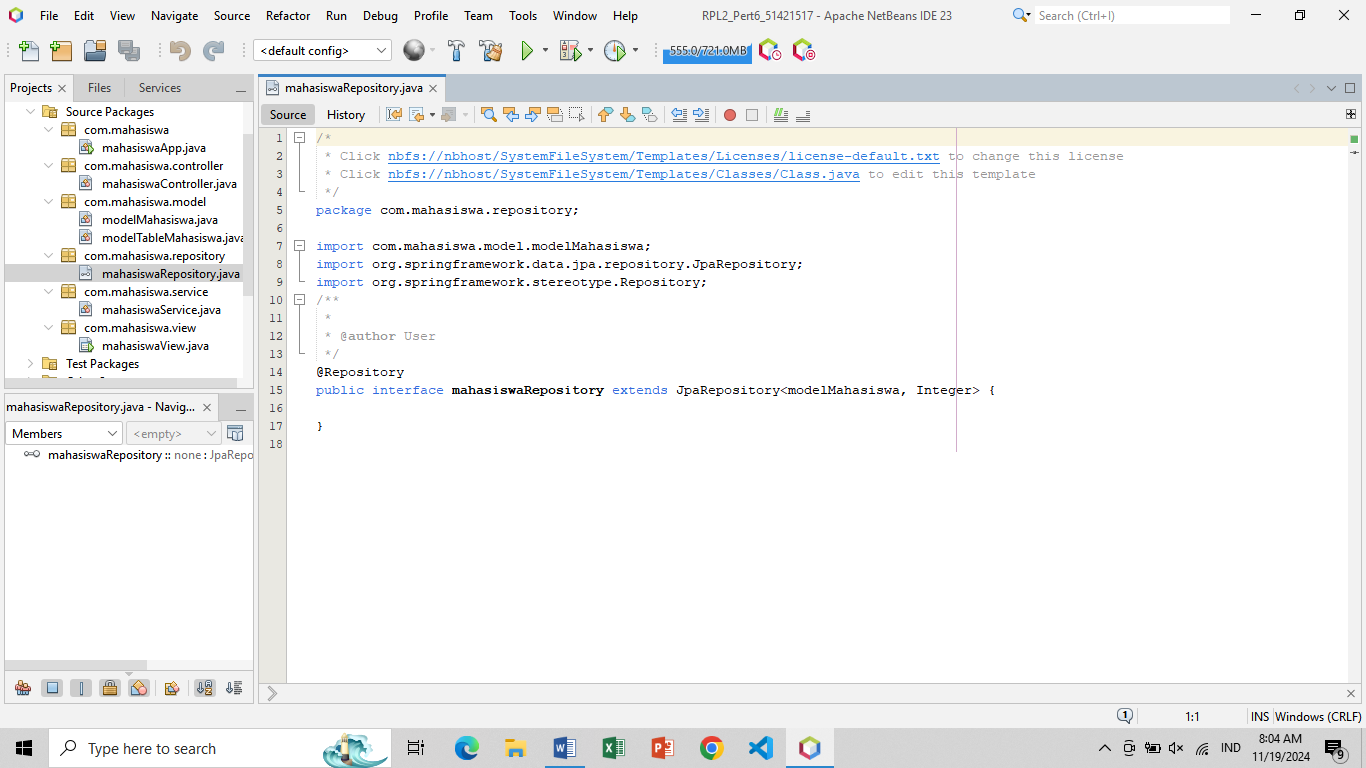
Method getMahasiswa menerima parameter id berupa integer dan akan mendapatkan record sesuai id yang diberikan dari tabel dengan memanggil method getMhs milik objek mahasiswaService.

Method updateMahasiswa menerima objek modelMahasiswa dan akan mengupdate data pada tabel dengan memanggil method updateMhs milik objek mahasiswaService.

Method deleteMahasiswa menerima parameter id berupa integer dan akan menghapus record sesuai id yang diberikan pada tabel dengan memanggil method deleteMhs milik objek mahasiswaService.

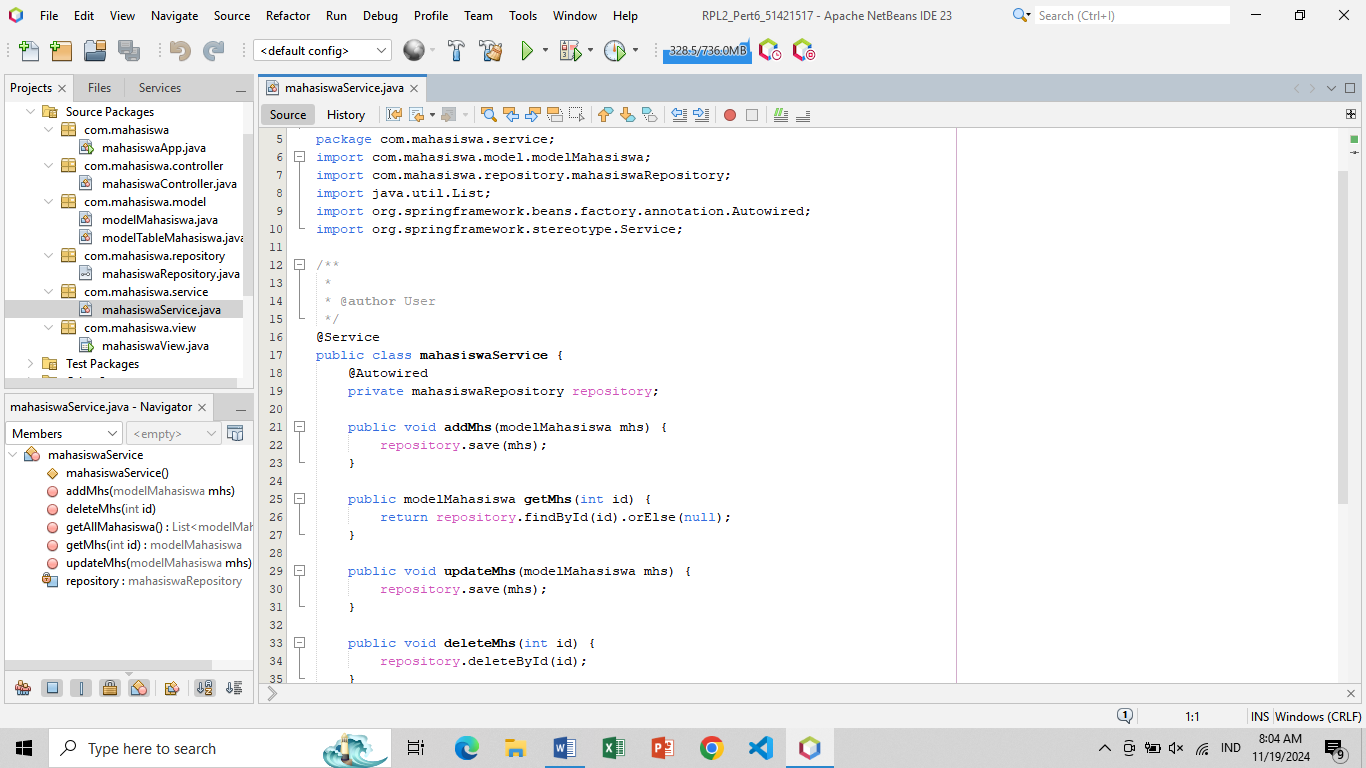
Method getAllMahasiswa akan membaca seluruh data pada tabel dengan memanggil method getAllMahasiswa milik objek mahasiswaService.

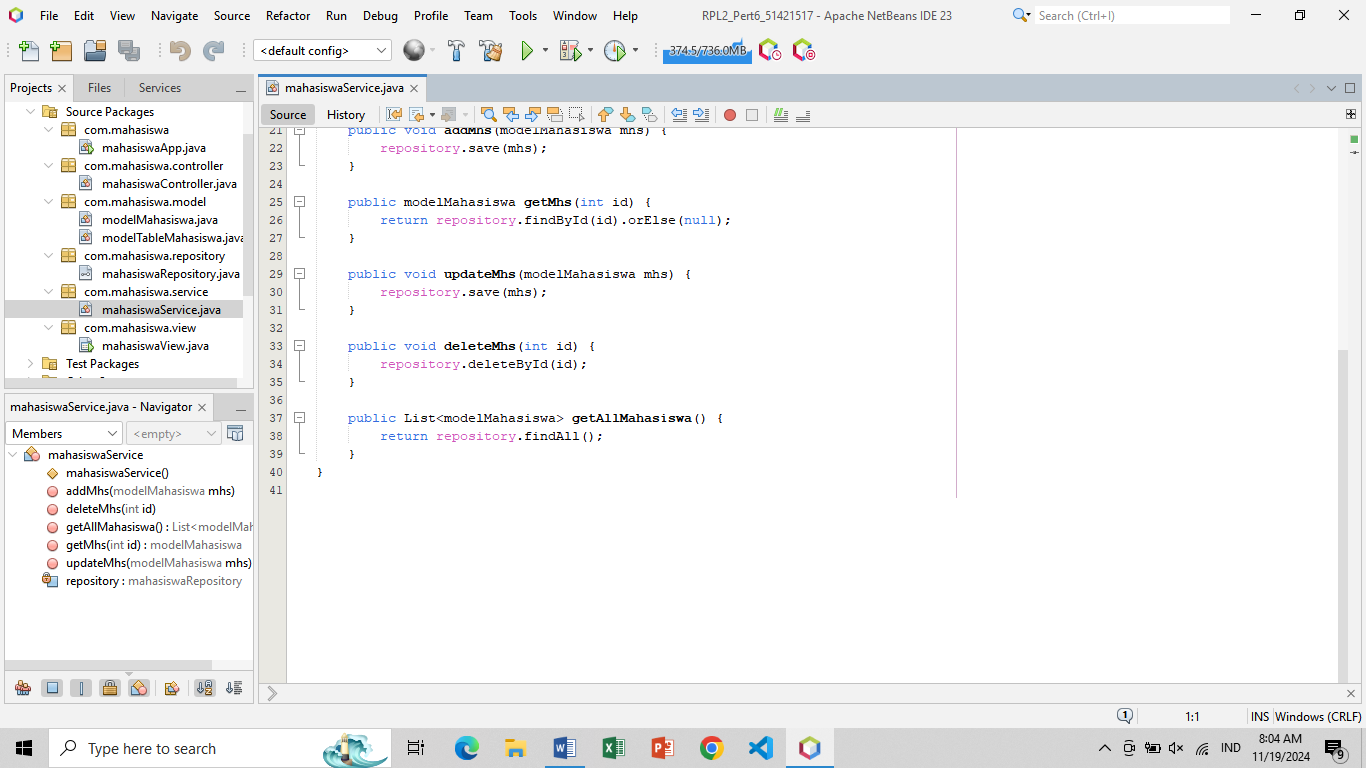
1. mahasiswaRepository.java



Class mahasiswaRepository berperan sebagai Repository atau penyimpanan data pada program, dengan tipe data yang disimpan adalah modelMahasiswa.

1. mahasiswaService.java





Class mahasiswaService menyediakan service untuk menambahkan dan memanipulasi data pada database melalui objek mahasiswaRepository yang di-inject.

Method addMhs menerima parameter objek modelMahasiswa dan akan menyimpan data tersebut pada tabel melalui objek mahasiswaRepository.

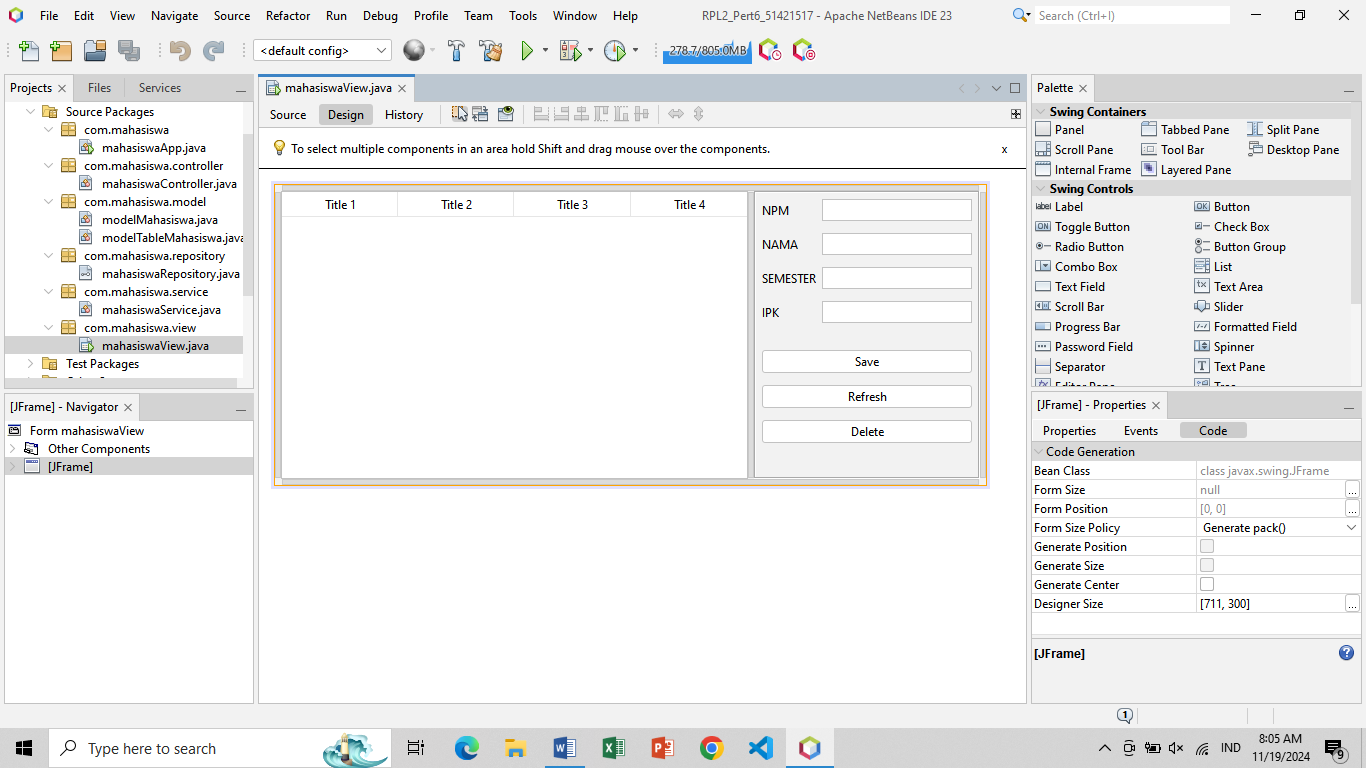
Method getMhs menerima parameter id berupa integer dan akan mencari data dengan id tersebut, dan jika data dengan id tersebut tidak ditemukan, maka akan mengembalikan null.

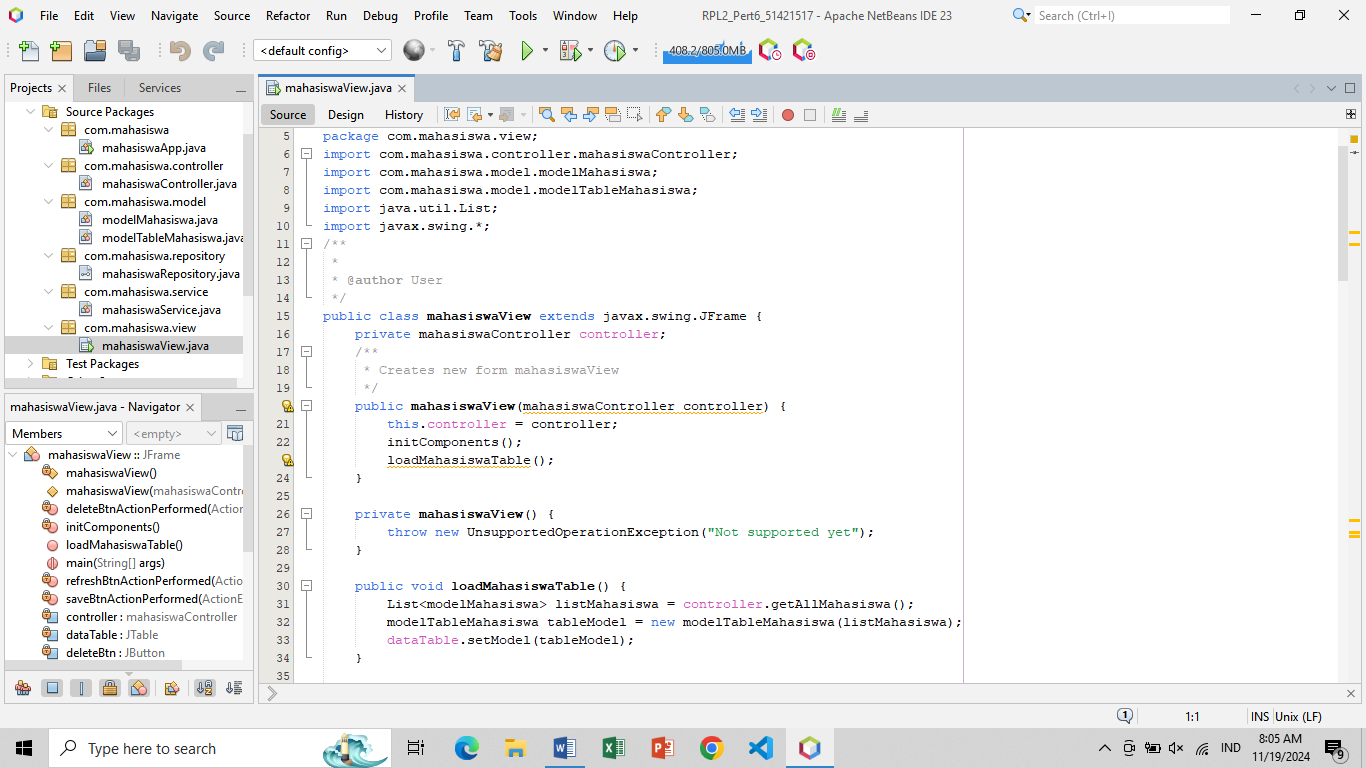
Method updateMhs menerima parameter objek modelMahasiswa dan akan mengupdate data tersebut pada tabel melalui objek mahasiswaRepository.

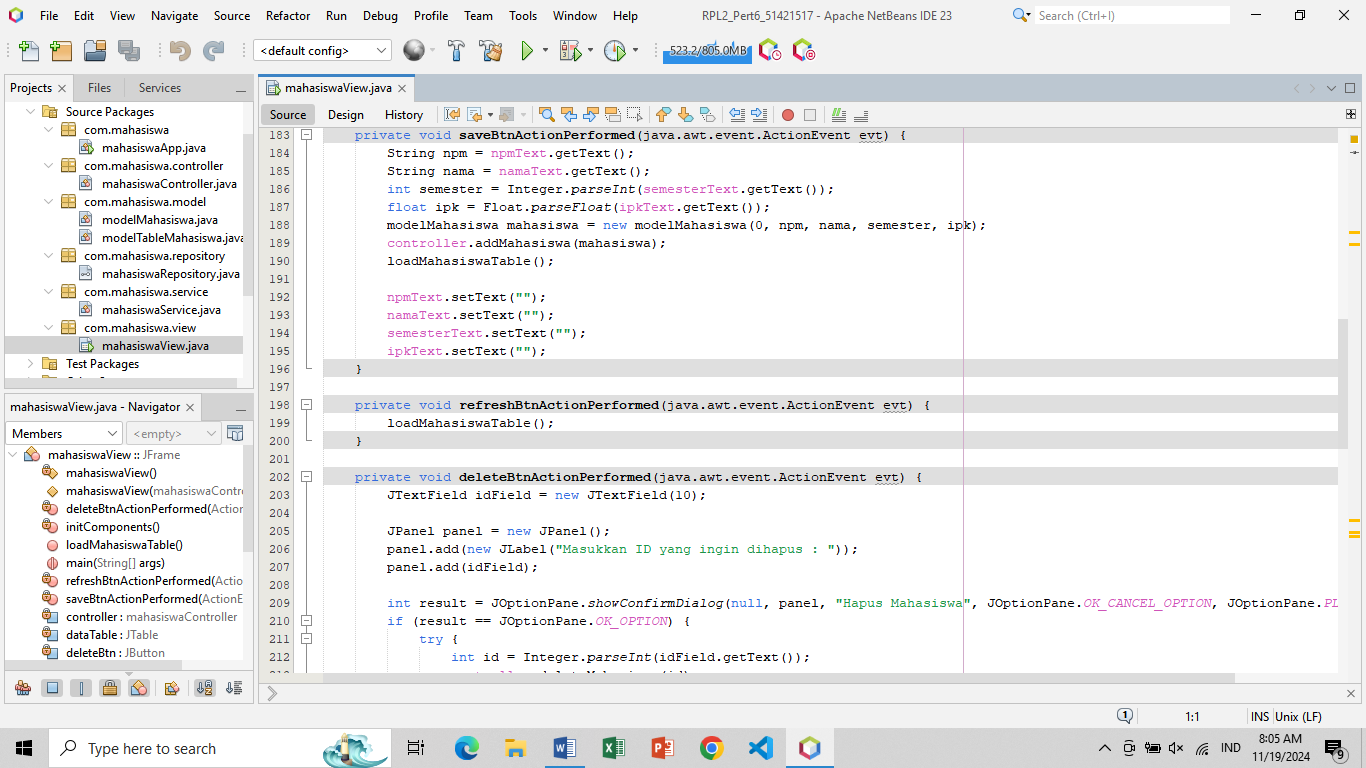
Method deleteMhs menerima parameter id berupa integer dan akan menghapus data dengan id tersebut.

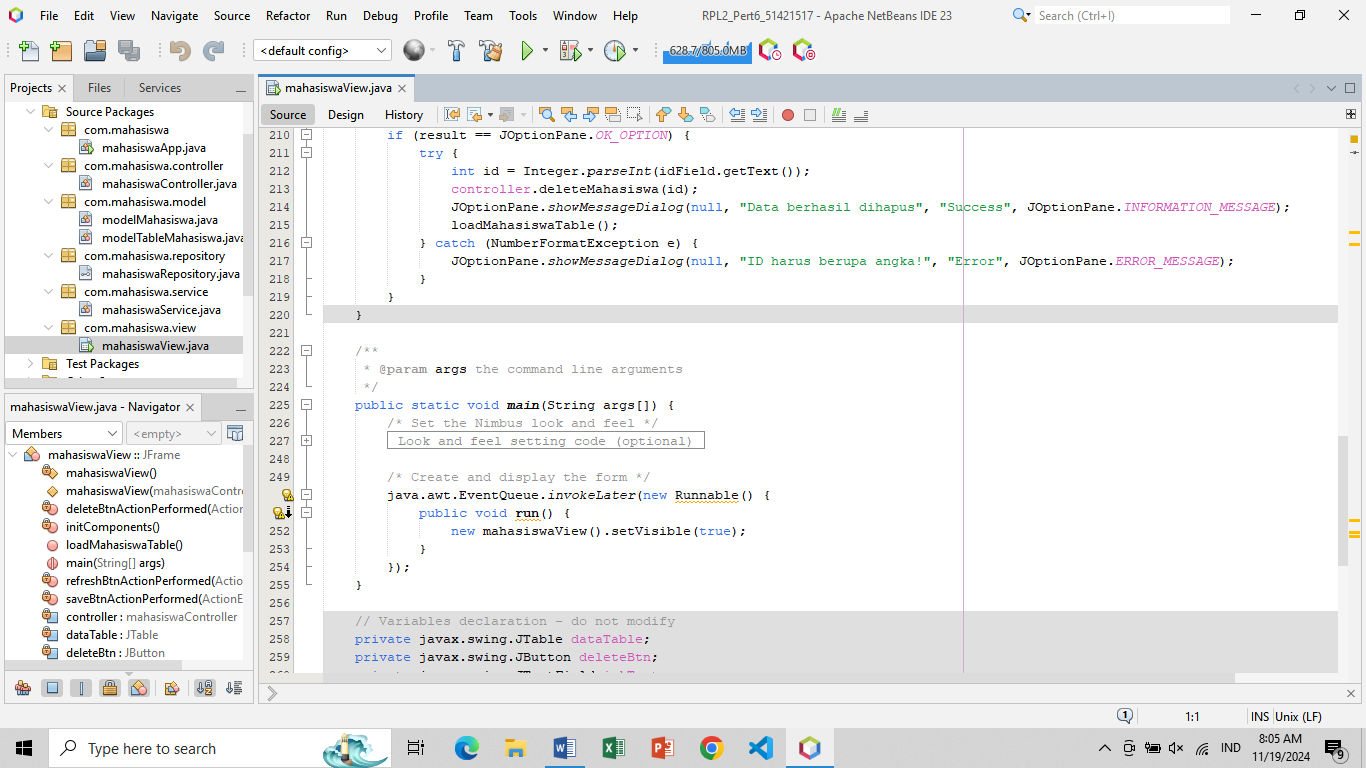
Method getAllMahasiswa akan mengembalikan seluruh data pada tabel dengan memanggil method findAll milik objek repository.

1. mahasiswaView.java









Class ini berperan sebagai View, yang akan menampilkan program dalam bentuk Java JFrame. Class ini memiliki atribut controller yang merupakan objek mahasiswaController.

Constructor mahasiswaView akan menginisialisasi komponen widget yang digunakan pada Jframe, menginisiasi controller, dan memuat tabel mahasiswa dengan memanggil method loadMahasiswaTable.

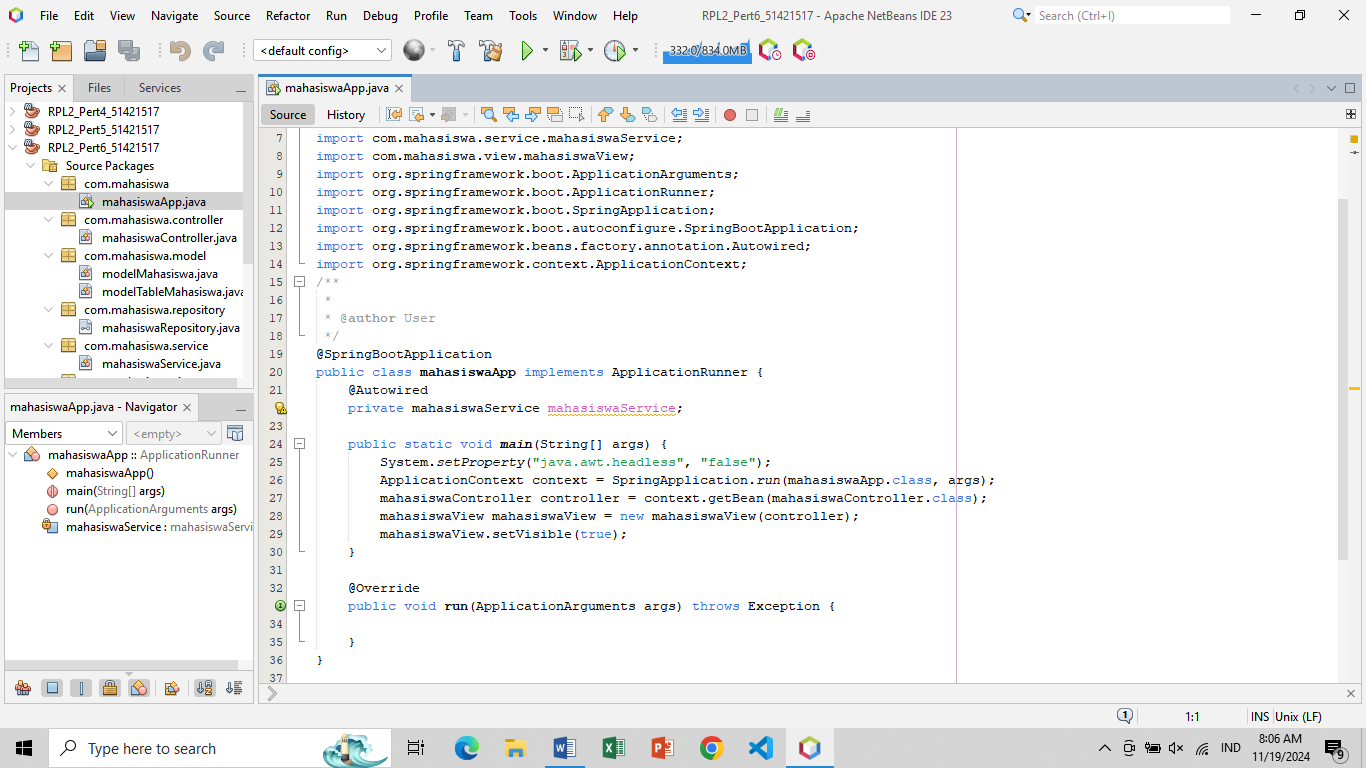
Method loadMahasiswaTable berfungsi untuk membaca seluruh data mahasiswa dari tabel mahasiswa. Hal tersebut dapat dilakukan dengan memanggil method getAllMahasiswa dari objek controller. Kemudian, struktur tabel didefinisikan menggunakan List bertipe data modelMahasiswa. Struktur tabel ini dijadikan sebagai argumen yang diberikan ke method setModel milik objek dataTable (Jtable).

Method saveBtnActionPerformed akan dieksekusi ketika tombol Simpan diklik. Hal yang dilakukan pertama-tama adalah mendapatkan data input mahasiswa, kemudian membuat objek modelMahasiswa baru dengan parameter-parameter tersebut. Objek controller kemudian menambahkan data tersebut ke database dengan memanggil method addMahasiswa. Kemudian, tabel dimuat ulang, dan text field dikosongkan kembali.

Method refreshBtnActionPerformed berfungsi untuk memuat ulang tabel dengan memanggil fungsi loadMahasiswaTable.

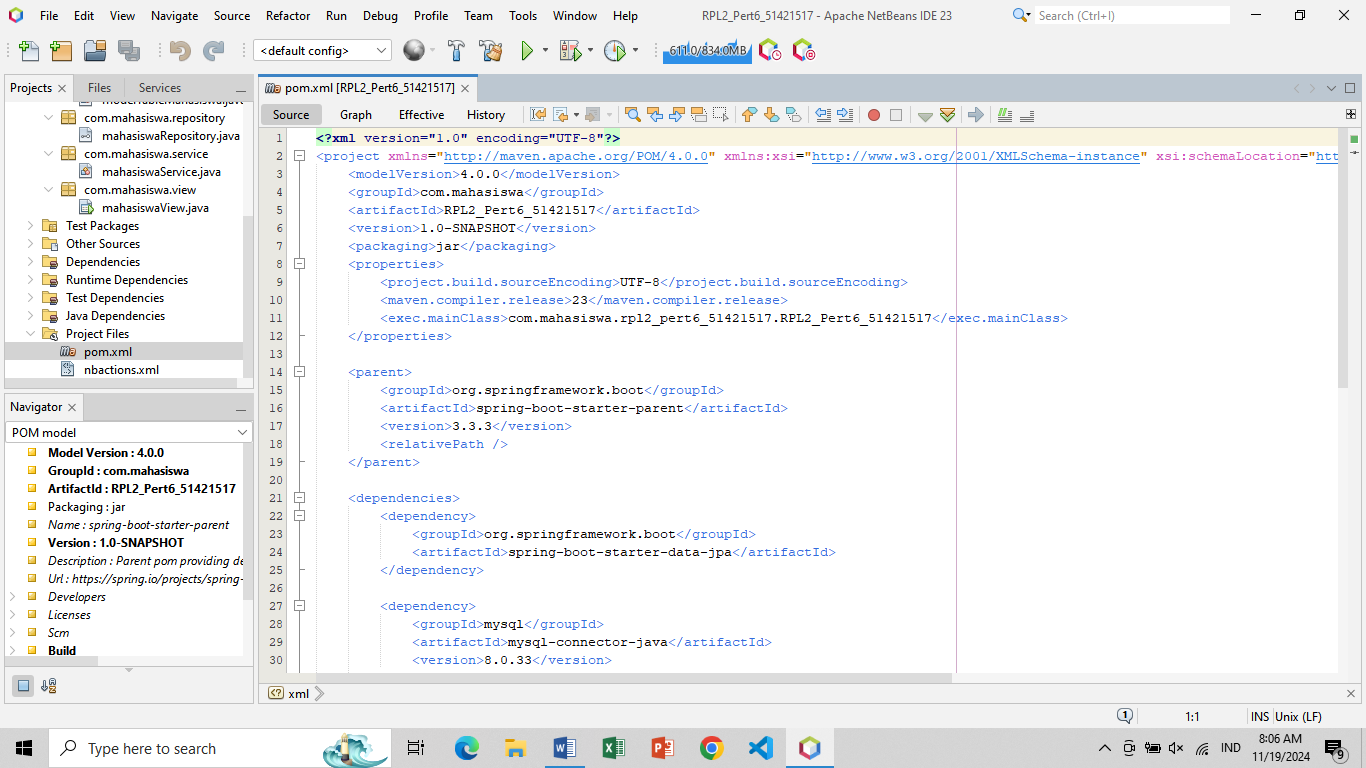
Method deleteBtnActionPerformed akan dieksekusi ketika tombol Delete diklik. Ketika tombol tersebut diklik, maka program akan menampilkan JPanel baru untuk meminta input ID mahasiswa yang ingin dihapus. Jika input berhasil dikonversi ke dalam tipe data integer, maka data dengan ID tersebut akan dihapus dari database dengan memanggil method deleteMahasiswa milik objek controller. Setelah berhasil dihapus dan menampilkan dialog konfirmasi, tabel akan dimuat ulang.

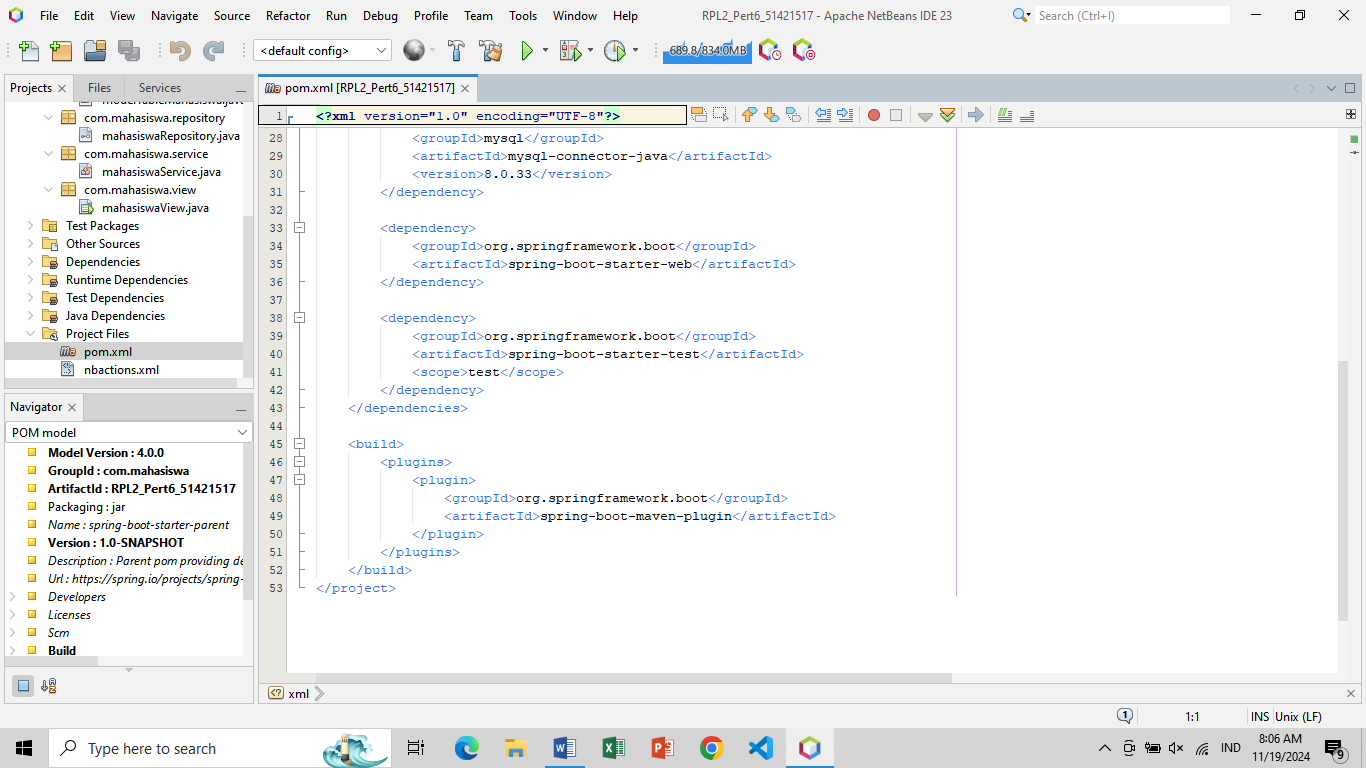
1. mahasiswaApp.java



Class tersebut adalah class main yang akan dipanggil ketika project dijalankan. Class tersebut mengimplementasi interface ApplicationRunner dari Spring. Objek dari MahasiswaController dan mahasiswaView akan diinisialisasi, kemudian program akan dijalankan.

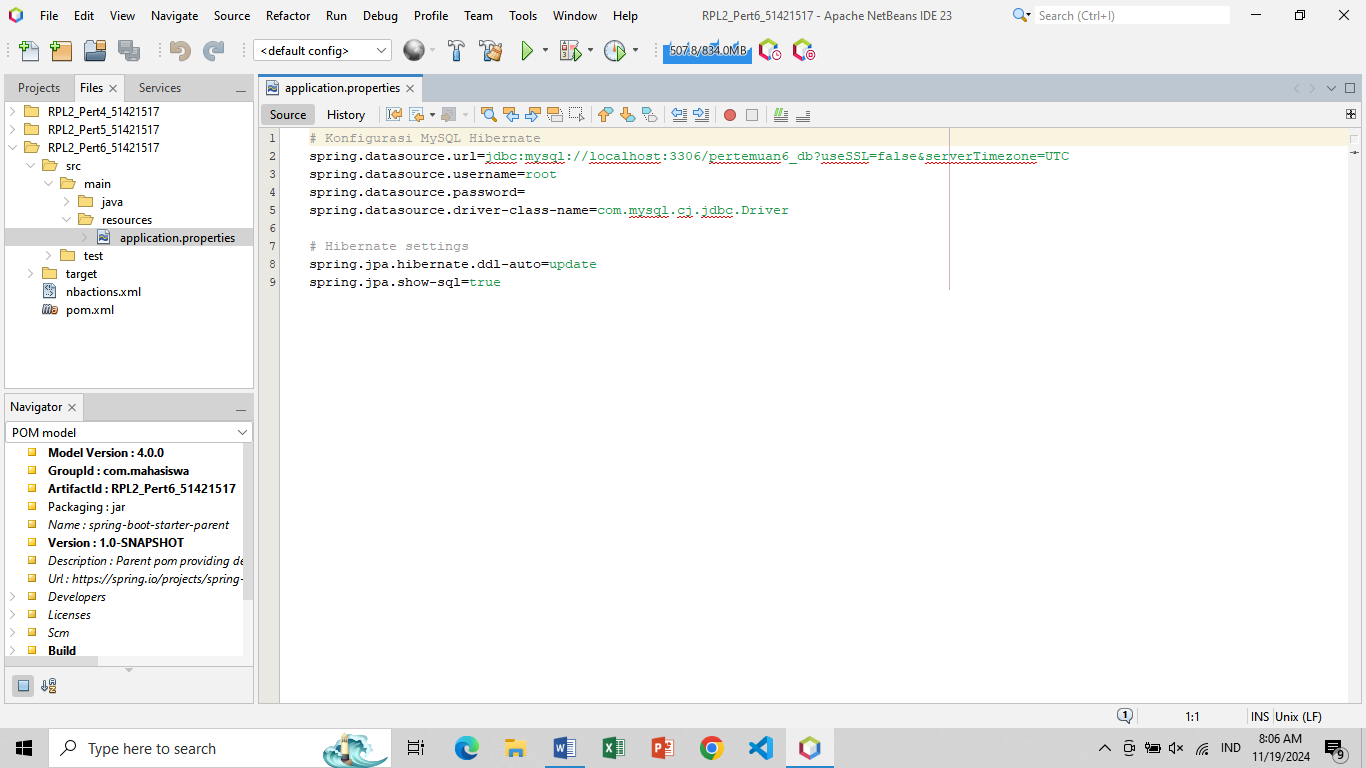
1. pom.xml





File pom.xml berisi dependencies yang digunakan pada program.

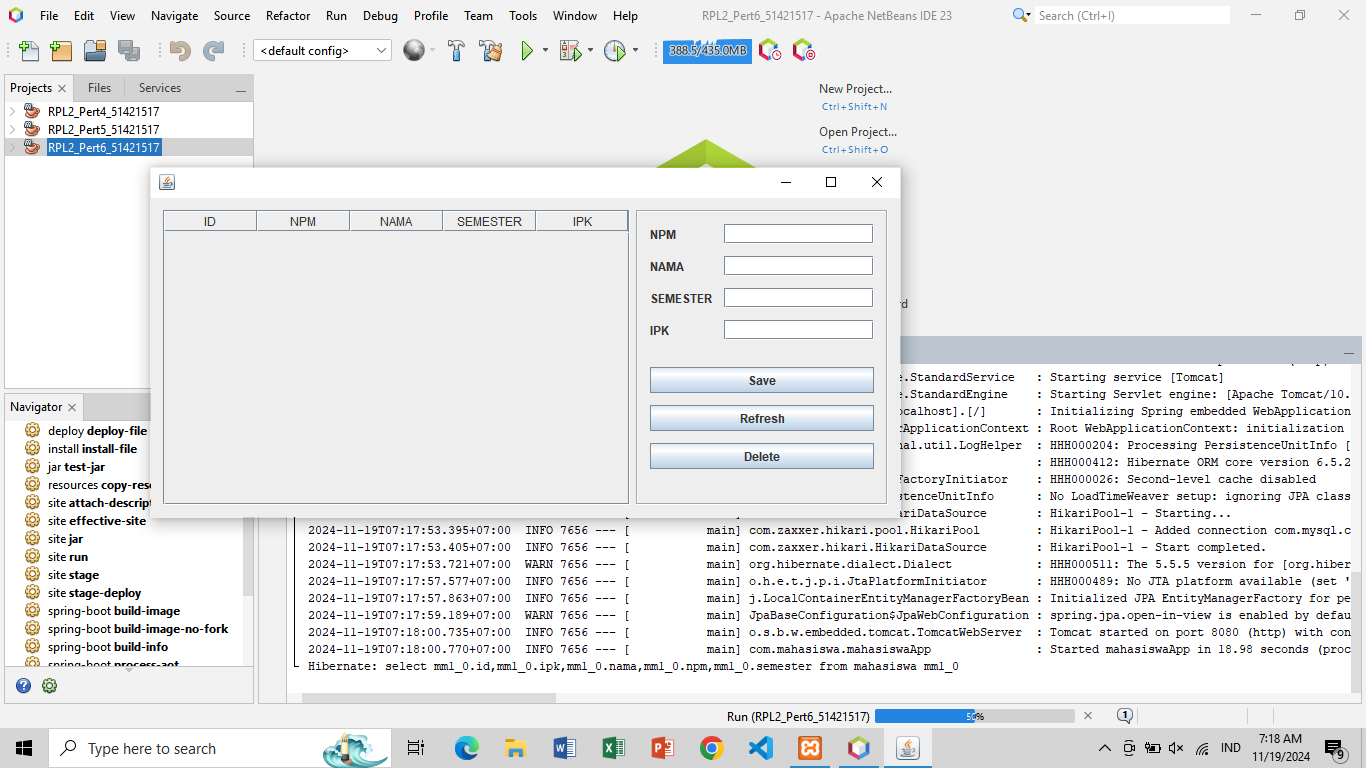
1. application.properties



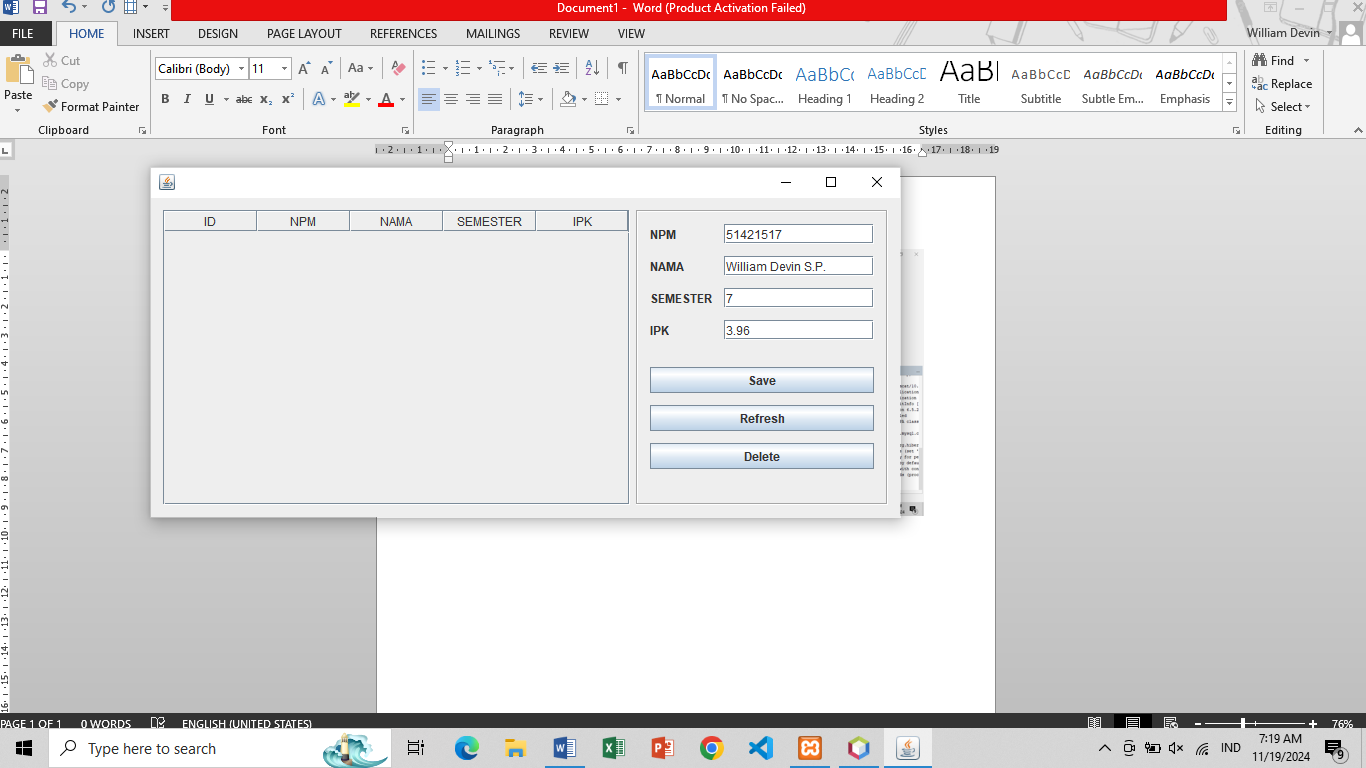
File application.properties mendefinisikan URL, username, password, dan nama Driver yang diperlukan untuk terhubung ke database. Selain itu, terdapat konfigurasi Hibernate pada file ini, yaitu ddl-auto dan show-sql yang diatur ke true.

**OUTPUT PROGRAM**

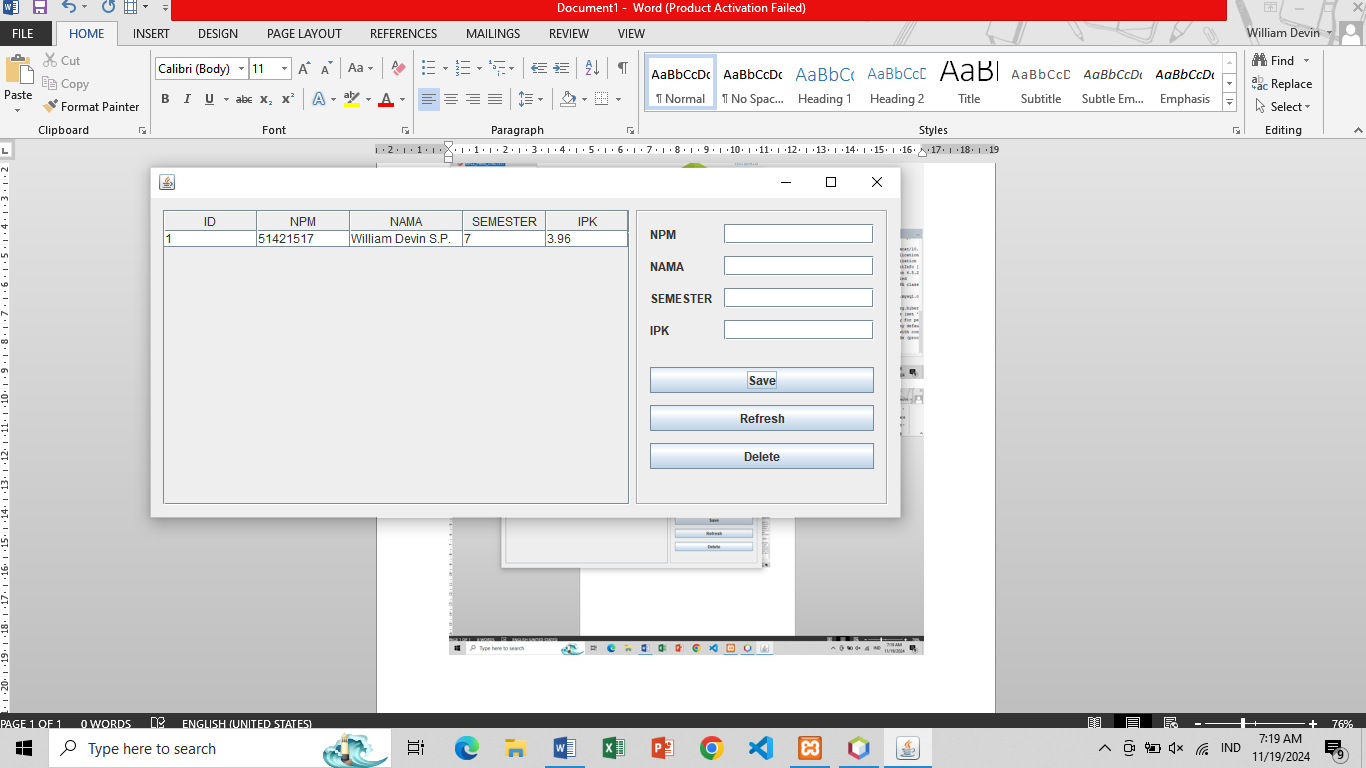
Tampilan awal



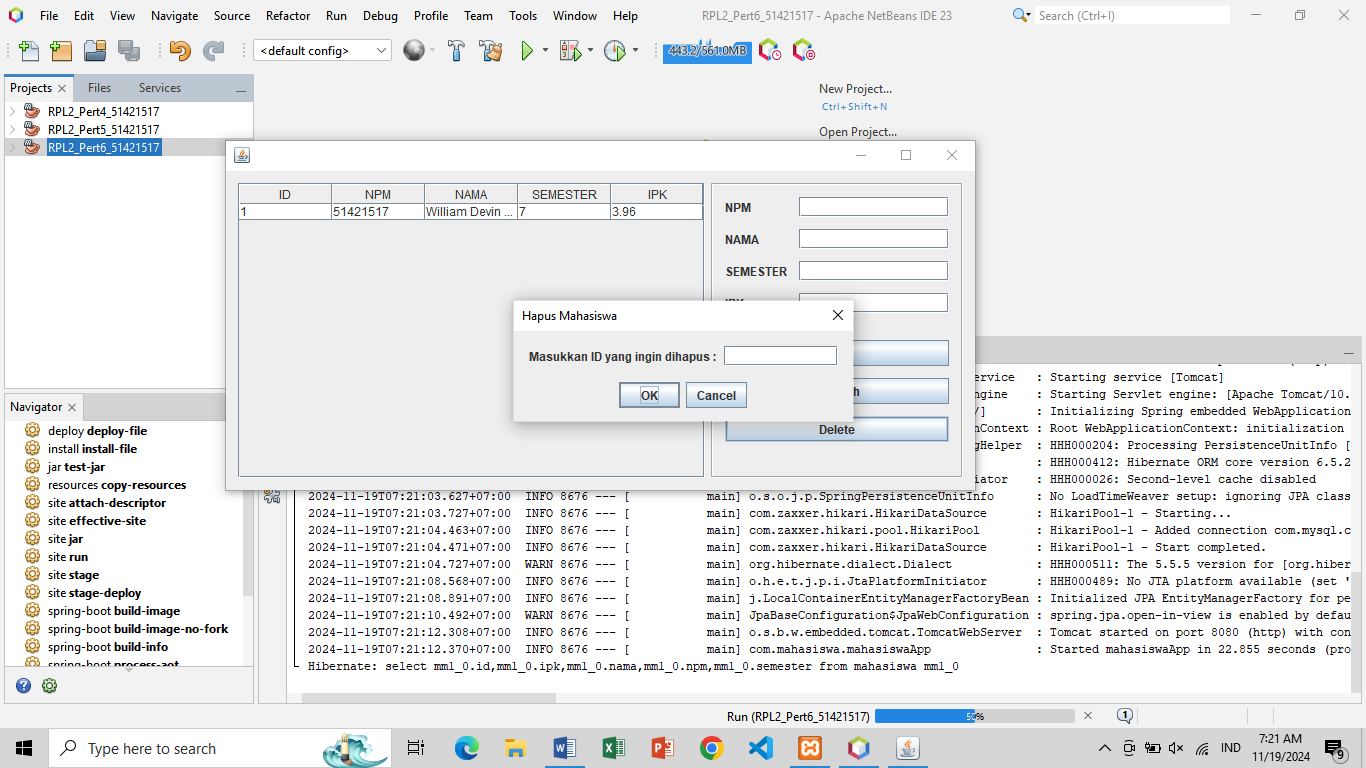
Tampilan memasukkan record



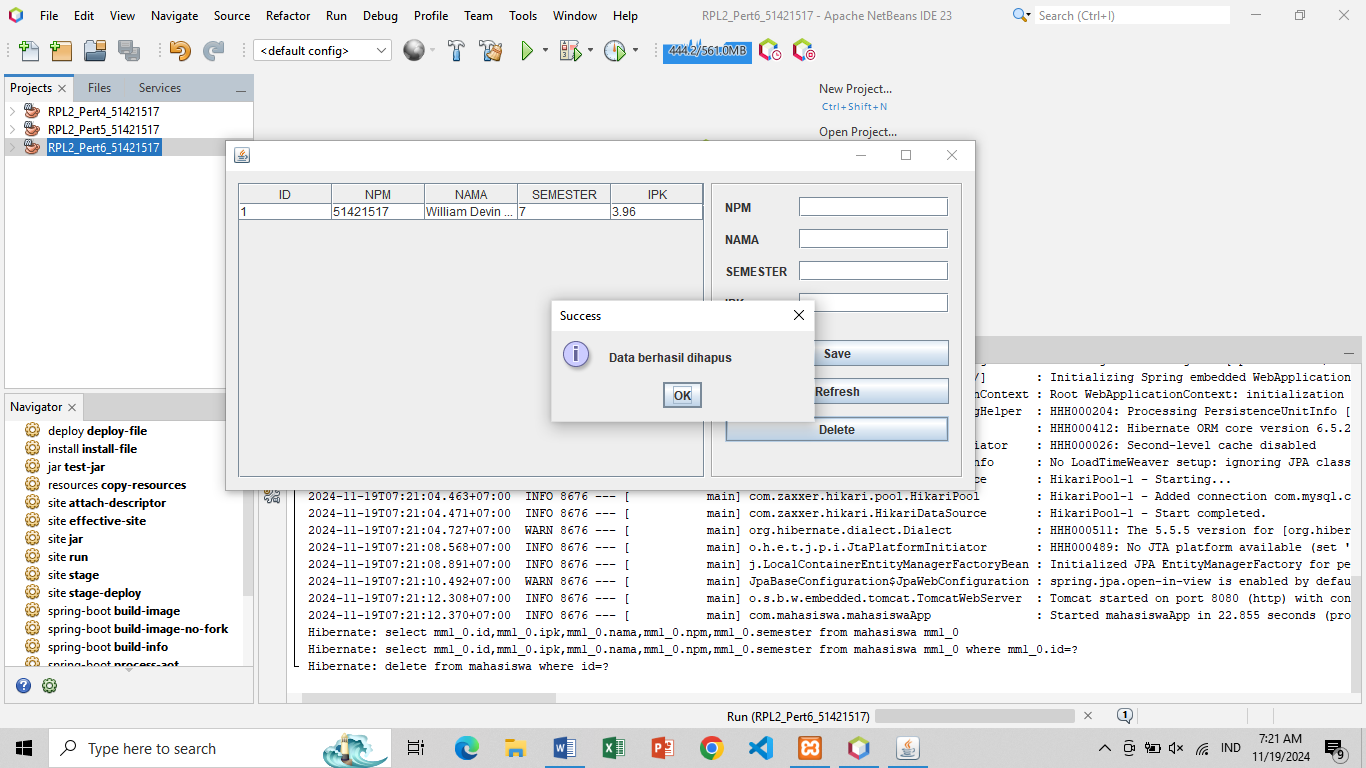
Tampilan ketika record berhasil ditambahkan



Tampilan dialog box setelah klik tombol Delete



Tampilan dialog box setelah menghapus record



Tampilan setelah record dihapus

