LAPORAN AKHIR PRAKTIKUM

Mata Praktikum : Teknik Kompilasi

Kelas : RPL2

Praktikum ke- : 3 & 4

Tanggal : 26 Oktober 2024

Materi :

NPM : 51421476

Nama : Theodore Gabbelambok Situmorang

Ketua Asisten : Dimas Renaldy

Paraf Asisten :

Nama Asisten :

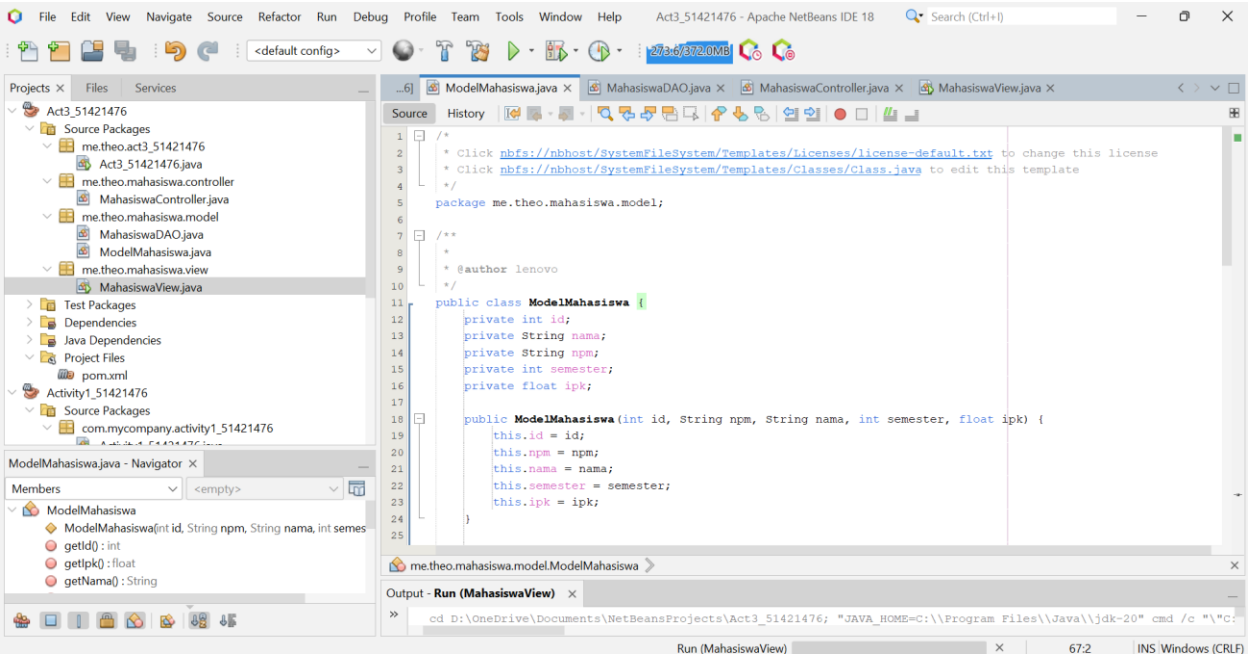
Jumlah Lembar :

LABORATORIUM TEKNIK INFORMATIKA

UNIVERSITAS GUNADARMA

2023

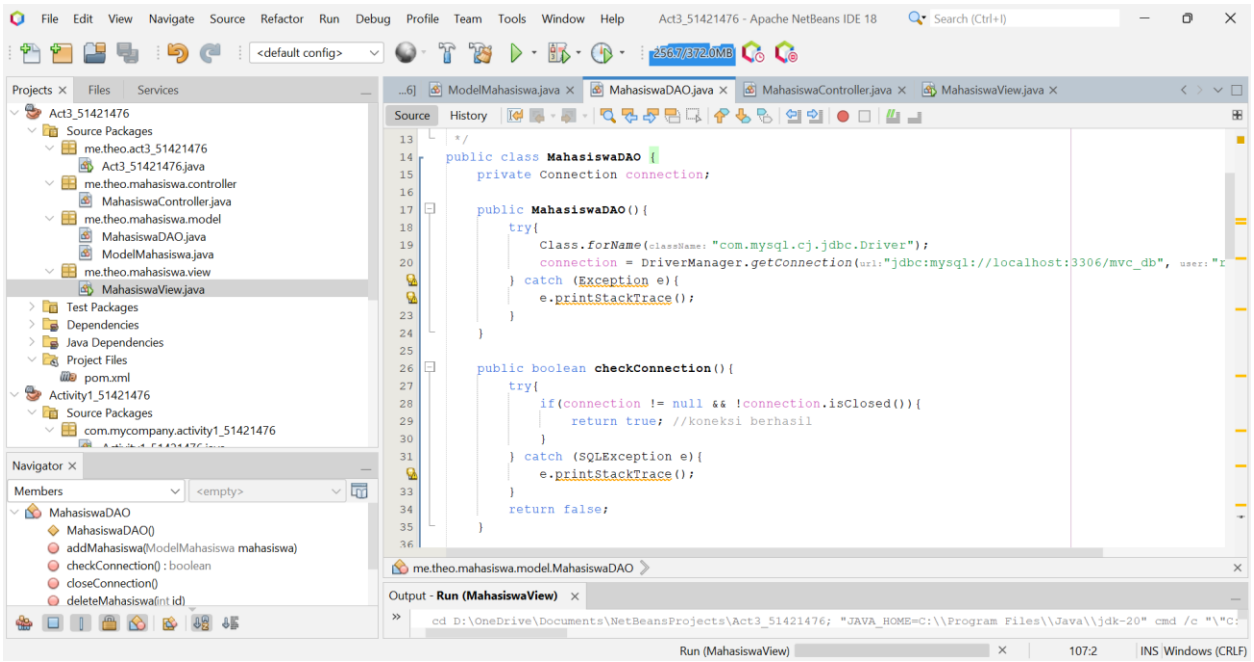
LAPORAN AKHIR 3



Kita mendeklarasikan kelas ModelMahasiswa dengan variabel id, nama, npm, semester, dan ipk.

Konstruktor adalah metode khusus yang dipanggil saat objek ModelMahasiswa dibuat, menerima lima parameter: id, npm, nama, semester, dan ipk. this digunakan untuk merujuk pada variabel kelas dan membedakannya dari parameter yang namanya sama.

Metode getter dan setter digunakan untuk mengakses dan mengubah nilai variabel seperti id mahasiswa. Getter mengembalikan nilai variabel, sementara setter mengubah nilainya, sehingga tetap menjaga prinsip enkapsulasi dalam kelas.



Kita melakukan import java.sql, java.util.ArrayList, dan java.util.List untuk mencegah terjadinya error dalam kode.

Mendeklarasikan kelas dengan variabel connection.

Melakukan koneksi ke database SQL yang sudah dibuat, dengan menyesuaikan localhost sesuai database yang kita buat.

Metode checkConnection (public boolean checkConnection()) mengembalikan nilai true jika koneksi berhasil, dan false jika gagal.

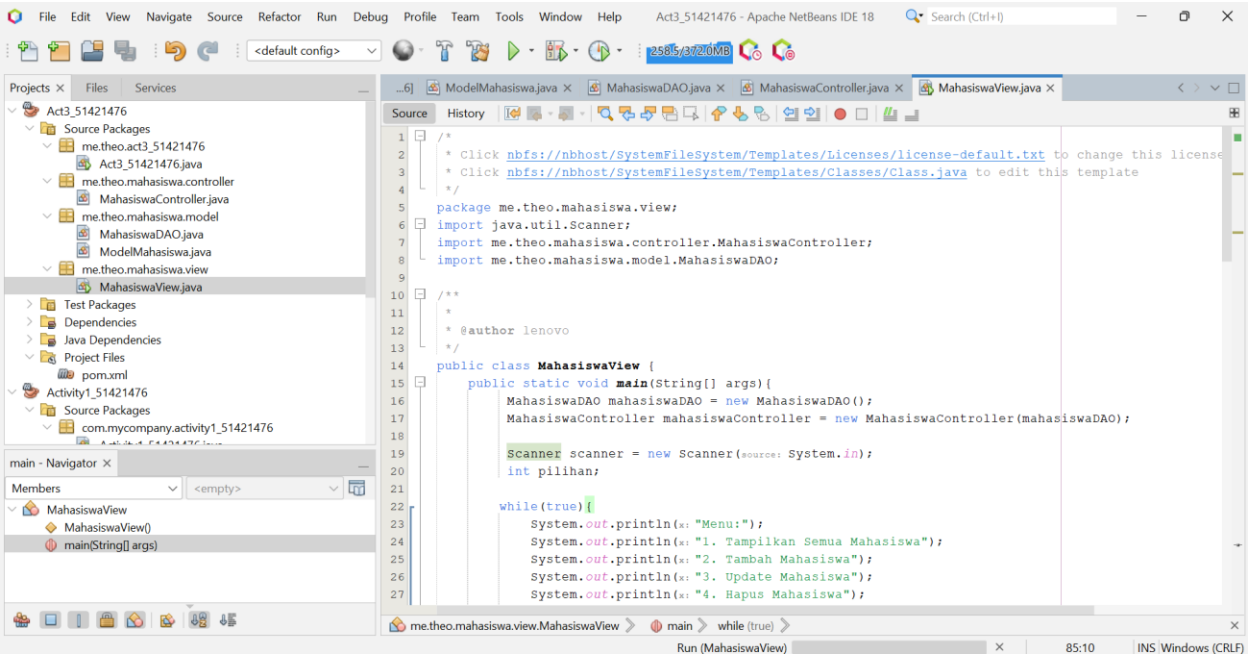
Metode addMahasiswa (public void addMahasiswa(ModelMahasiswa mahasiswa)) menambahkan data mahasiswa ke tabel mahasiswa dalam database. Pernyataan SQL menggunakan placeholder (?) untuk nilai yang akan dimasukkan.

Metode getAllMahasiswa mengambil seluruh data mahasiswa dari tabel mahasiswa dalam database dan mengembalikannya dalam bentuk List berisi objek ModelMahasiswa. List mahasiswaList dibuat kosong dan diisi dengan data dari database menggunakan pernyataan SQL SELECT * FROM mahasiswa.

Metode updateMahasiswa memperbarui data mahasiswa di database berdasarkan id. SQL UPDATE digunakan untuk memperbarui kolom npm, nama, semester, dan ipk berdasarkan id.

Metode deleteMahasiswa menghapus data mahasiswa dari tabel berdasarkan id yang diberikan. Pernyataan SQL DELETE digunakan untuk menghapus data dari tabel berdasarkan id.

Metode closeConnection menutup koneksi ke database setelah operasi selesai untuk membebaskan sumber daya. connection.close() menutup koneksi, yang merupakan praktik baik untuk menghindari kebocoran memori.



Melakukan import java.util.Scanner dan folder yang telah dibuat sebelumnya.

Melakukan inisialisasi objek utama dengan:

MahasiswaDAO mahasiswaDAO = new MahasiswaDAO(); untuk mengelola akses data mahasiswa dalam database.

MahasiswaController mahasiswaController = new MahasiswaController(mahasiswaDAO); untuk membuat objek MahasiswaController yang dapat menggunakan fungsi dari MahasiswaDAO.

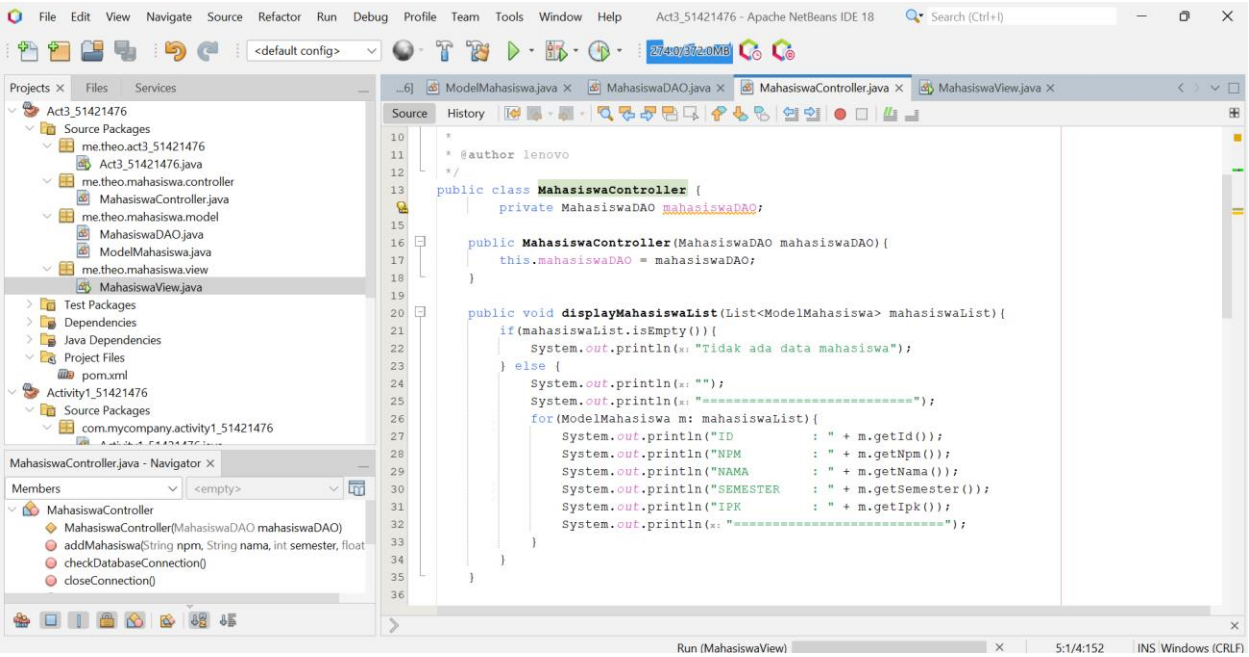
Scanner scanner = new Scanner(System.in); untuk menerima input dari pengguna melalui console.

Struktur while digunakan untuk Menu Utama yang menyediakan pilihan opsi bagi pengguna. Loop while(true) membuat program terus berjalan hingga opsi 6 (keluar) dipilih.

System.out.println(...) mencetak menu ke console untuk memandu pengguna.

pilihan = scanner.nextInt(); menerima input angka dari pengguna untuk menentukan opsi yang dipilih.

scanner.nextLine(); membersihkan buffer Scanner setelah menerima input angka.



Melakukan import ke folder yang sudah dibuat sebelumnya.

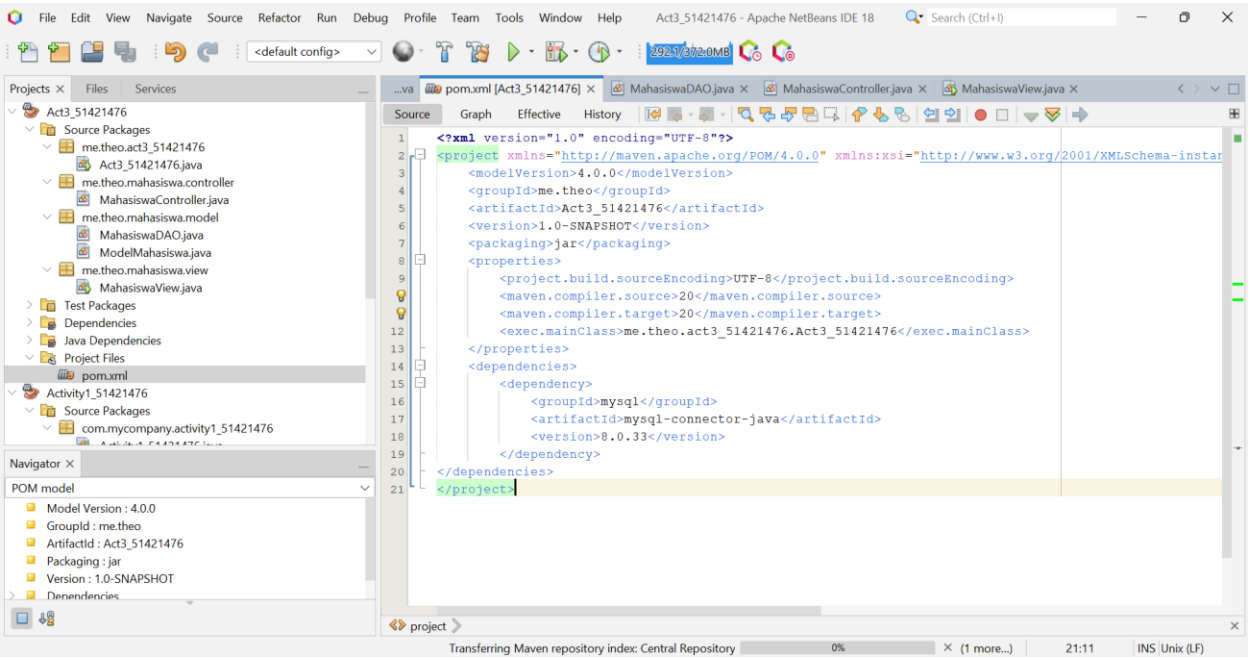
Mendeklarasikan kelas dan variabel.

Konstruktor digunakan untuk menginisialisasi objek MahasiswaController dan mengaitkan MahasiswaDAO untuk mengakses data. this.mahasiswaDAO = mahasiswaDAO; menetapkan parameter mahasiswaDAO ke variabel instance agar metode dalam MahasiswaController dapat mengakses data dari database.

Metode displayMahasiswaList menampilkan daftar mahasiswa dalam bentuk List<ModelMahasiswa> ke console. Jika mahasiswaList kosong, akan muncul pesan bahwa tidak ada data mahasiswa.

Metode displayMessage digunakan untuk menampilkan pesan teks ke console dengan System.out.println(message);.

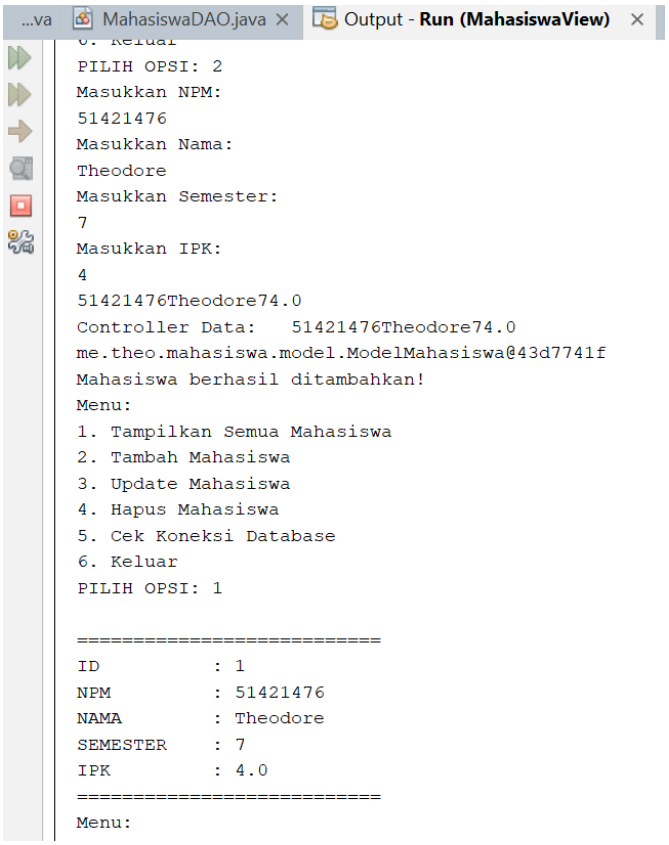
Metode CheckDataBaseConnection memeriksa status koneksi ke database melalui mahasiswaDAO dan menampilkan apakah koneksi berhasil atau gagal. boolean isConnected = mahasiswaDAO.checkConnection(); memanggil metode checkConnection dari mahasiswaDAO dan menyimpan hasilnya dalam isConnected.



Selanjutnya, buka file pom.xml yang ada di folder project files. File pom.xml berfungsi untuk menghubungkan proyek dengan database SQL yang telah dibuat.

OUTPUT


Output - Run (MahasiswaView)
cd D:\OneDrive\Documents\NetBeansProjects\Act3_51421476; "JAVA_HOME=C:\\Program Files\\Java\\jdk-20" cmd /c "\"C:
Scanning for projects...
----------------------< me.theo:Act3_51421476 >-----------------------
Building Act3_51421476 1.0-SNAPSHOT
from pom.xml
--------------------------------[jar]---------------------------------
The artifact mysql:mysql-connector-java:jar:8.0.33 has been relocated to com.mysql:mysql-connector-j:jar:8.0.33:
--- resources:3.3.0:resources (default-resources) @ Act3_51421476 ---
skip non existing resourceDirectory D:\OneDrive\Documents\NetBeansProjects\Act3_51421476\src\main\resources
--- compiler:3.10.1:compile (default-compile) @ Act3_51421476 ---
Changes detected - recompiling the module!
Compiling 5 source files to D:\OneDrive\Documents\NetBeansProjects\Act3_51421476\target\classes
--- exec:3.1.0:exec (default-cli) @ Act3_51421476 ---
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI:


...va
MahasiswaDAO.java
Output - Run (MahasiswaView)
6. Keluar
PILIH OPSI: 2
Masukkan NPM:
51421476
Masukkan Nama:
Theodore
Masukkan Semester:
7
Masukkan IPK:
4
51421476Theodore74.0
Controller Data: 51421476Theodore74.0
me.theo.mahasiswa.model.ModelMahasiswa@43d7741f
Mahasiswa berhasil ditambahkan!
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI: 1
==========
ID : 1
NPM : 51421476
NAMA : Theodore
SEMESTER : 7
IPK : 4.0
==========
Menu:

LAPORAN AKHIR PRAKTIKUM

Mata Praktikum : Teknik Kompilasi

Kelas : RPL2

Praktikum ke- : 3 & 4

Tanggal : 26 Oktober 2024

Materi :

NPM : 51421476

Nama : Theodore Gabbelambok Situmorang

Ketua Asisten : Dimas Renaldy

Paraf Asisten :

Nama Asisten :

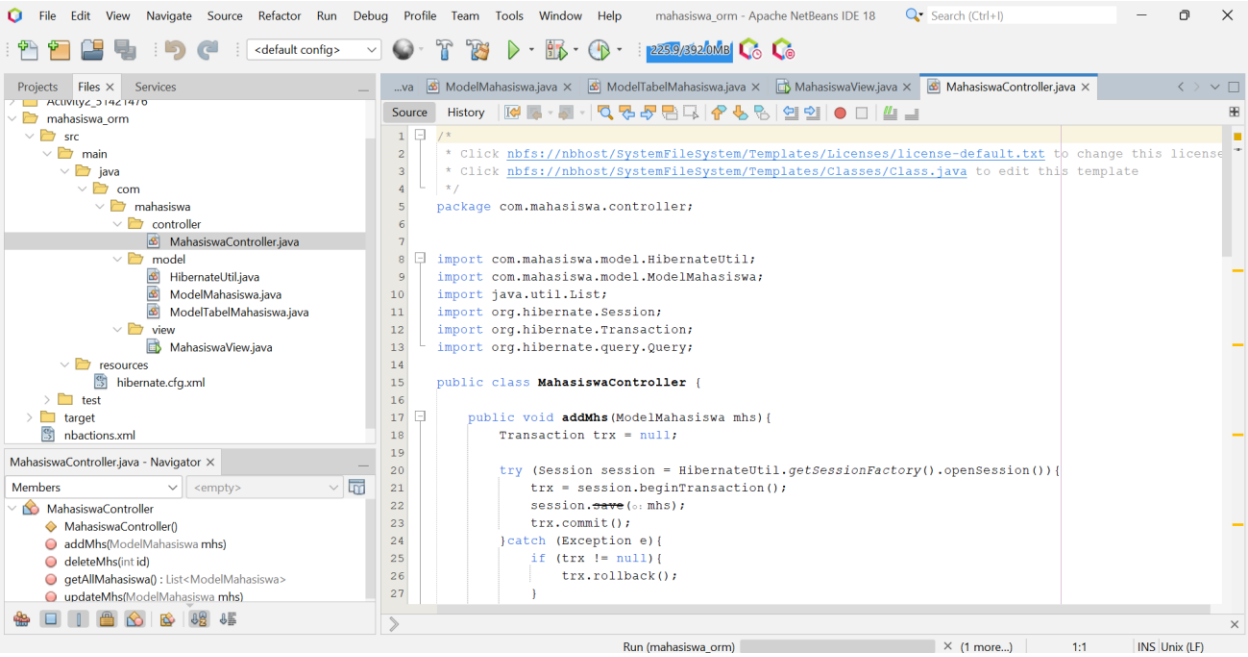
Jumlah Lembar :

LABORATORIUM TEKNIK INFORMATIKA

UNIVERSITAS GUNADARMA

2023

LAPORAN AKHIR 4



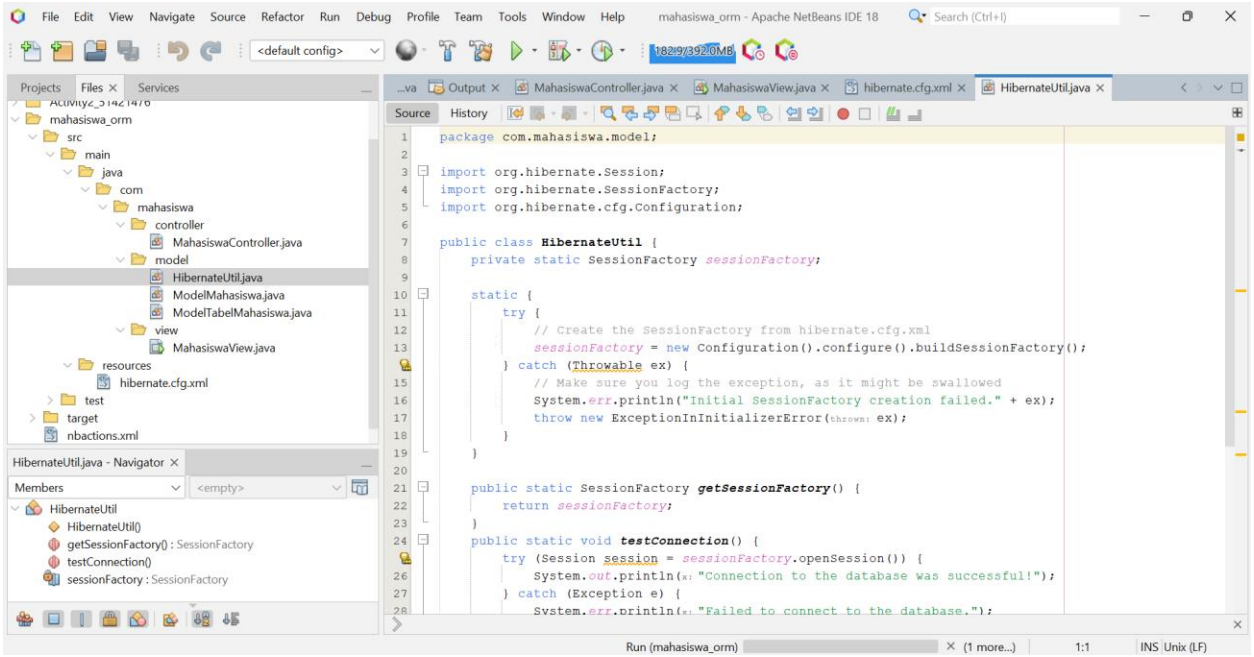
Melakukan import.

Metode addmhs digunakan untuk menambahkan data mahasiswa baru ke database. trx adalah objek Transaction untuk memastikan perubahan dijalankan dalam satu transaksi. session membuka koneksi ke database, dan session.save(mhs); menyimpan objek mahasiswa baru ke database. Jika berhasil, trx.commit(); menyimpan perubahan secara permanen; jika terjadi kesalahan, transaksi akan di-rollback.

Metode updatemhs memperbarui data mahasiswa di database. Setelah membuka koneksi dan memulai transaksi, session.update(mhs); memperbarui data mahasiswa. Jika berhasil, transaksi dikonfirmasi dengan commit, atau di-rollback jika ada kesalahan.

Metode deletemhs menghapus data mahasiswa berdasarkan id. Membuka koneksi ke database dan memulai transaksi, kemudian mengambil data mahasiswa dengan session.get(ModelMahasiswa.class, id);. Jika data ditemukan, session.delete(mhs); menghapusnya. Sistem menampilkan "Berhasil hapus" jika sukses, atau di-rollback jika gagal.

Metode getAllMahasiswa mengambil semua data mahasiswa dari database. Membuka koneksi dan memulai transaksi, lalu menggunakan HQL (Hibernate Query Language) dengan query from ModelMahasiswa untuk mengambil semua data ModelMahasiswa. Hasil disimpan dalam listMhs dan dikembalikan sebagai output. Transaksi dikonfirmasi jika berhasil, atau di-rollback jika tidak.



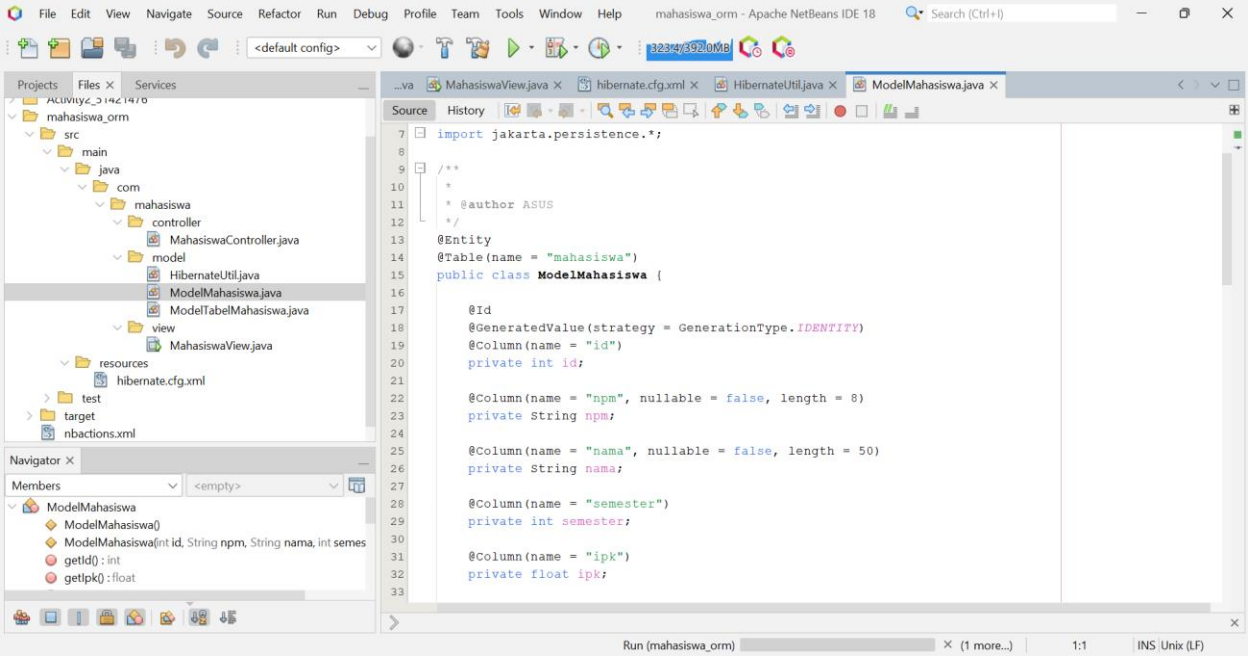
Melakukan import.

Mendeklarasikan kelas dengan variabel static sessionFactory.

Membuat blok static yang dijalankan sekali saat kelas HibernateUtil dimuat ke memori. Dalam blok ini, sessionFactory diinisialisasi dengan memuat konfigurasi dari file hibernate.cfg.xml menggunakan new Configuration().configure().buildSessionFactory();, yang berisi pengaturan koneksi database dan konfigurasi Hibernate. Jika terjadi kesalahan saat membuat sessionFactory, pengecualian akan ditangkap, pesan error akan dicetak, dan aplikasi akan dihentikan.

Metode getSessionFactory berfungsi untuk menyediakan akses ke sessionFactory dan merupakan metode static sehingga dapat dipanggil tanpa membuat objek HibernateUtil. Kelas lain menggunakan metode ini untuk mendapatkan sessionFactory yang telah diinisialisasi.

Metode testConnection digunakan untuk menguji koneksi ke database dengan membuka sesi melalui sessionFactory.openSession(). Jika koneksi berhasil, akan muncul pesan "Connection to the database was successful!" dan jika gagal, akan mencetak pesan "Failed to connect to the database." bersama detail pengecualian menggunakan e.printStackTrace();.



Melakukan import.

Anotasi @Entity menandakan bahwa kelas ModelMahasiswa adalah entitas JPA yang merepresentasikan tabel di database.

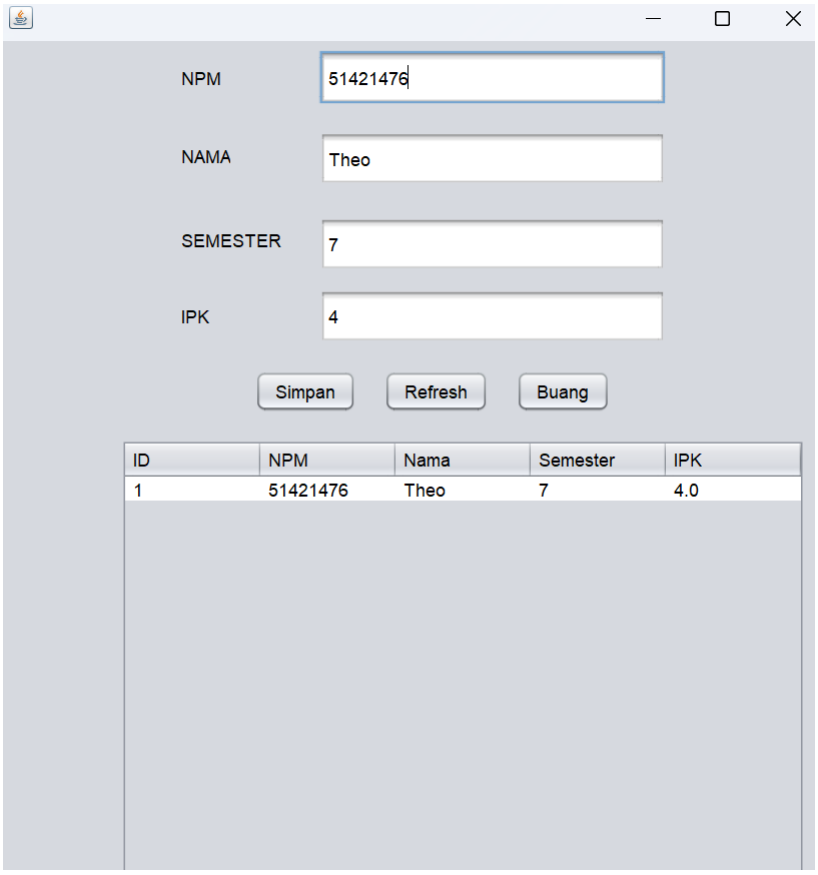
Anotasi @Table(name = "mahasiswa") digunakan untuk menentukan nama tabel database yang dipetakan ke entitas ini, yaitu mahasiswa. Jika tidak ditentukan, Hibernate menggunakan nama kelas sebagai nama tabel secara default.

Pendeklarasian variabel seperti id, npm, nama, semester, dan ipk.

Konstruktor ModelMahasiswa() (default) diperlukan oleh Hibernate untuk membuat instance kelas saat memuat data dari database. Konstruktor ModelMahasiswa(int id, String npm, String nama, int semester, float ipk) dengan parameter memudahkan inisialisasi instance ModelMahasiswa dengan nilai-nilai yang diperlukan.

Metode getter, seperti getId(), digunakan untuk mengambil nilai variabel, misalnya nilai id.

OUTPUT


NPM
51421476
NAMA
Theo
SEMESTER
7
IPK
4
Simpan
Refresh
Buang
ID
NPM
Nama
Semester
IPK
1
51421476
Theo
7
4.0