

LAPORAN AKHIR PRAKTIKUM

Mata Praktikum : Rekayasa Perangkat Lunak 2

Kelas : 4IA14

Praktikum ke- : 4

Tanggal : 4 November 2024

Materi : Konsep Dasar Object Relational Mapping (ORM) dan Framework Hibernate

NPM : 50421651

Nama : Indah Dwi Apriliani

Ketua Asisten : Suryo Aji Widagdo

Paraf Asisten :

Nama Asisten :

Jumlah Lembar : 13 Lembar

LABORATORIUM TEKNIK INFORMATIKA

UNIVERSITAS GUNADARMA

2024

1. Jelaskan apa yang dimaksud dengan ORM dan keuntungannya!

Jawab :

ORM (Object-Relational Mapping) adalah teknik yang digunakan dalam pemrograman untuk menghubungkan dan memetakan objek dalam bahasa pemrograman dengan tabel dalam basis data relasional. Dengan ORM, pengembang dapat berinteraksi dengan basis data menggunakan objek, tanpa perlu menulis query SQL secara langsung. Keuntungan utama dari ORM termasuk abstraksi database yang menyembunyikan kompleksitas SQL dan memberikan antarmuka yang lebih intuitif, sehingga meningkatkan produktivitas pengembang. Selain itu, ORM menawarkan portabilitas yang memudahkan pemindahan aplikasi antara berbagai sistem basis data. Penggunaan ORM juga mengurangi risiko kesalahan sintaksis dalam penulisan query, karena objek lebih mudah dikelola dan sering dilengkapi dengan validasi otomatis untuk menjaga integritas data. Dukungan untuk prinsip pemrograman berorientasi objek memungkinkan pengembang memanfaatkan fitur-fitur OOP, seperti pewarisan dan polimorfisme, sementara pengelolaan relasi antar objek menjadi lebih sederhana. Dengan demikian, ORM adalah alat yang sangat berguna dalam pengembangan aplikasi yang memerlukan interaksi dengan basis data, menjadikannya cara yang efisien dan efektif untuk mengelola data.

2. Jelaskan Kode dan langkah-langkah program yang telah dibuat!

Jawab :

ModelMahasiswa.java

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package me.indah.model;

import jakarta.persistence.*;

/**
 *
 * @author ACER
 */
@Entity
@Table(name = "mahasiswa")
public class ModelMahasiswa {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id")
    private int id;

    @Column(name = "npm", nullable = false, length = 10)
    private String npm;

    @Column(name = "nama", nullable = false, length = 55)
    private String nama;

    @Column(name = "semester")
    private int semester;

    @Column(name = "ipk")
    private float ipk;

    public ModelMahasiswa() {
```

Penjelasan :

- Kelas ini merepresentasikan data mahasiswa dengan atribut seperti id, npm, nama, semester, dan ipk.
- Terdapat getter dan setter untuk mengakses atau mengubah data.
- `import jakarta.persistence.*;` digunakan untuk mengimpor anotasi dan fitur JPA yang akan membantu dalam pengelolaan entitas dan pemetaan ke database.

HibernateUtil.java

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package me.indah.model;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

public class HibernateUtil {
    private static SessionFactory sessionFactory;

    static {
        try {
            sessionFactory = new Configuration().configure().buildSessionFactory();
        } catch (Throwable ex) {
            System.err.println("Initial sessionFactory creation failed." + ex);
            throw new ExceptionInInitializerError(ex);
        }
    }

    public static SessionFactory getSessionFactory() {
        return sessionFactory;
    }

    public static void testConnection() {
        try (Session session = sessionFactory.openSession()) {
            System.out.println("Connection to the database was successful!");
        } catch (Exception e) {
            System.err.println("Failed to connect to the database.");
            e.printStackTrace();
        }
    }
}
```

Penjelasan :

Untuk mengelola koneksi Hibernate ke database. Kelas ini menggunakan pola Singleton untuk memastikan hanya satu instance SessionFactory yang digunakan selama aplikasi berjalan. Kode HibernateUtil di atas menyertakan beberapa elemen penting untuk pengelolaan koneksi Hibernate dalam sebuah aplikasi.

MahasiswaController.java

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package me.indah.controller;

import me.indah.model.HibernateUtil;
import me.indah.model.ModelMahasiswa;
import java.util.List;
import org.hibernate.Session;
import org.hibernate.Transaction;
import org.hibernate.query.Query;

/**
 *
 * @author ACER
 */
public class MahasiswaController {

    public void addMhs (ModelMahasiswa mhs) {
        Transaction trx = null;
        try (Session session = HibernateUtil.getSessionFactory().openSession()) {
            trx = session.beginTransaction();
            session.save(mhs);
            trx.commit();
        } catch (Exception e) {
            if (trx != null) trx.rollback();
            e.printStackTrace();
        }
    }
}
```

Penjelasan :

MahasiswaController yang bertanggung jawab untuk mengelola operasi CRUD (Create, Read, Update, Delete) pada entitas ModelMahasiswa menggunakan Hibernate. Kelas ini menyediakan beberapa metode untuk menambahkan, memperbarui, menghapus, dan mengambil data mahasiswa dari database.

Setiap metode memastikan data diproses dengan aman, menggunakan transaksi untuk memastikan data tidak tersimpan atau berubah secara parsial jika terjadi kesalahan.

Mahasiswa View.java

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
package me.indah.view;

import me.indah.controller.MahasiswaController;
import me.indah.model.HibernateUtil;
import me.indah.model.ModelMahasiswa;
import me.indah.model.ModelTabelMahasiswa;
import java.util.List;
import javax.swing.*;

/**
 *
 * @author ACER
 */
public class MahasiswaView extends javax.swing.JFrame {

    /**
     * Creates new form MahasiswaView
     */
    private MahasiswaController controller;

    public MahasiswaView() {
        initComponents();
        HibernateUtil.testConnection();
        this.controller = new MahasiswaController();
        loadMahasiswaTable();
    }
}
```

Penjelasan :

Mendefinisikan antarmuka GUI (Graphical User Interface) untuk mengelola data mahasiswa menggunakan JFrame dari Java Swing, dalam kelas MahasiswaView. Kelas ini bekerja sama dengan MahasiswaController untuk melakukan operasi pada data mahasiswa, seperti menampilkan, menambah, dan menghapus data mahasiswa dari database.

Design

Source Design History

The Navigator window displays a tree hierarchy of components in the opened form.

The form contains the following elements:

- Four input fields labeled NPM, Nama, Semester, and IPK.
- Three buttons labeled Simpan, Buang, and Refresh.
- A table with four columns labeled Title 1, Title 2, Title 3, and Title 4.

Penjelasan :

➤ Label dan Field Input

Di sebelah kiri atas terdapat empat label (NPM, Nama, Semester, IPK) yang menunjukkan jenis data yang diinputkan oleh pengguna.

➤ Tabel Data

Di bagian bawah terdapat tabel yang menampilkan data mahasiswa dalam format kolom dan baris. Tabel ini memiliki header dengan label "Title 1", "Title 2", "Title 3", dan "Title 4", yang seharusnya diubah agar sesuai dengan kolom data mahasiswa seperti NPM, Nama, Semester, dan IPK.

➤ ScrollBar

Tabel ini juga dilengkapi dengan scrollbar di sebelah kanan yang mempermudah pengguna untuk menggulir data, terutama jika data yang ditampilkan lebih banyak dari tinggi area tabel.

➤ Tombol Aksi

Simpan: Untuk menyimpan data mahasiswa yang diinputkan ke dalam database.

Buang: (Kemungkinan) untuk menghapus data mahasiswa yang dipilih dari tabel.

Refresh: Untuk memperbarui tampilan tabel dengan data terbaru yang ada di database.

Model Tabel Mahasiswa

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit th
 */
package me.indah.model;

import javax.swing.table.AbstractTableModel;
import java.util.List;

public class ModelTabelMahasiswa extends AbstractTableModel{
    private List<ModelMahasiswa> mahasiswaList;
    private String[] columnNames = {"ID", "NPM", "Nama", "Semester", "IPK"};

    public ModelTabelMahasiswa(List<ModelMahasiswa> mahasiswaList) {
        this.mahasiswaList = mahasiswaList;
    }

    @Override
    public int getRowCount() {
        return mahasiswaList.size();
    }

    @Override
    public int getColumnCount() {
        return columnNames.length;
    }
}
```

Penjelasan :

Pada sebuah model tabel untuk data mahasiswa dalam aplikasi berbasis Java yang menggunakan Swing. Kelas ModelTabelMahasiswa mengextends AbstractTableModel, yang memungkinkan kelas ini digunakan sebagai model untuk komponen tabel seperti JTable. Di dalamnya, terdapat variabel mahasiswaList, yang menyimpan daftar objek dari kelas ModelMahasiswa, dan array columnNames, yang berisi nama-nama kolom yang akan ditampilkan di tabel, seperti "ID", "NPM", "Nama", "Semester", dan "IPK". Konstruktor kelas ini

menerima list mahasiswa dan menginisialisasi variabel mahasiswaList dengan data tersebut.

Hibernate.cfg.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD 3.0//EN" "http://hibernate.org/dtd/hibernate-configuration-3.0.dtd" [
<hibernate-configuration>
  <session-factory>
    <!-- Database connection settings -->
    <property name="hibernate.connection.driver_class">com.mysql.cj.jdbc.Driver</property>
    <property name="hibernate.connection.url">jdbc:mysql://localhost:3306/hibernate_50421651</property>
    <property name="hibernate.connection.username">root</property>
    <property name="hibernate.connection.password"></property>

    <!-- JDBC connection pool settings -->
    <property name="hibernate.c3p0.min_size">5</property>
    <property name="hibernate.c3p0.max_size">20</property>
    <property name="hibernate.c3p0.timeout">300</property>
    <property name="hibernate.c3p0.max_statements">50</property>
    <property name="hibernate.c3p0.idle_test_period">3000</property>

    <!-- SQL dialect -->
    <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>

    <!-- Echo all executed SQL to stdout -->
    <property name="hibernate.show_sql">true</property>

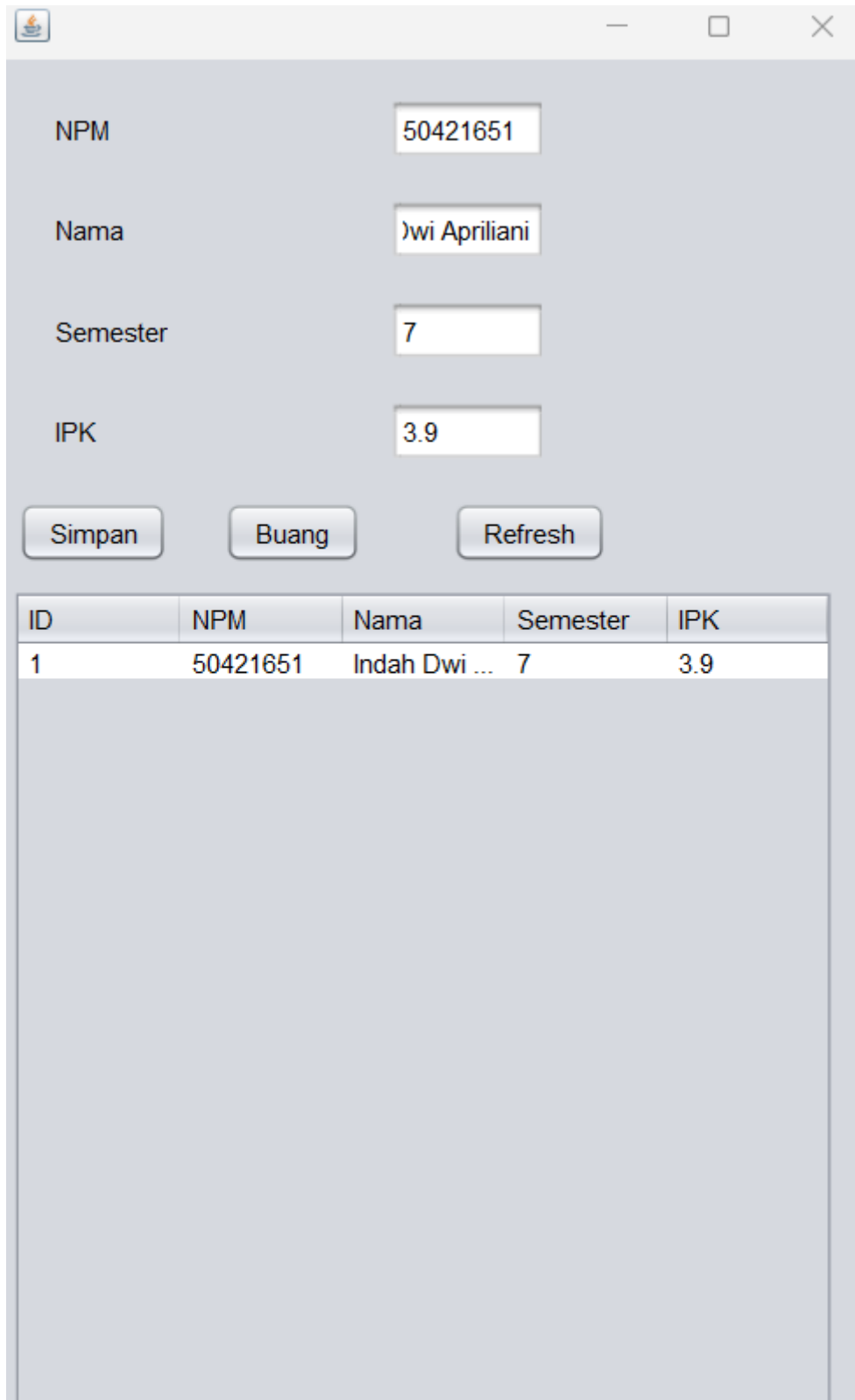
    <!-- Drop and re-create the database schema on startup -->
    <property name="hibernate.hbm2ddl.auto">update</property>

    <!-- Mapping class -->
    <mapping class="me.azwar.model.ModelMahasiswa"/>
  </session-factory>
</hibernate-configuration>
```

Penjelasan :

Pada konfigurasi untuk Hibernate, sebuah framework pemrograman Java yang digunakan untuk mengelola database melalui pendekatan Object-Relational Mapping (ORM). File konfigurasi ini dimulai dengan deklarasi XML yang menunjukkan versi dan encoding, diikuti oleh elemen `<hibernate-configuration>` yang menjadi akar dokumen. Di dalamnya, terdapat elemen `<session-factory>` yang mendefinisikan pengaturan untuk pabrik sesi, yang mengelola sesi komunikasi antara aplikasi dan database. Pengaturan koneksi database mencakup driver JDBC yang digunakan, URL untuk koneksi, serta kredensial autentikasi dengan username root dan password kosong. Selain itu, ada pengaturan pool koneksi JDBC yang mengatur ukuran minimum dan maksimum pool, waktu timeout, dan caching pernyataan SQL. Dialek SQL yang digunakan ditentukan dengan mengatur `hibernate.dialect` ke `MySQLDialect`.

OUTPUT :



The screenshot shows a web application window with a light gray background. At the top, there is a header bar with a small icon on the left and standard window controls (minimize, maximize, close) on the right. Below the header, the form contains four input fields, each with a label to its left: 'NPM' with the value '50421651', 'Nama' with the value 'Dwi Apriliani', 'Semester' with the value '7', and 'IPK' with the value '3.9'. Below these fields are three buttons: 'Simpan', 'Buang', and 'Refresh'. At the bottom of the form is a table with five columns: 'ID', 'NPM', 'Nama', 'Semester', and 'IPK'. The table contains one data row with the values: '1', '50421651', 'Indah Dwi ...', '7', and '3.9'.

| | |
|----------|---------------|
| NPM | 50421651 |
| Nama | Dwi Apriliani |
| Semester | 7 |
| IPK | 3.9 |

Simpan Buang Refresh

| ID | NPM | Nama | Semester | IPK |
|----|----------|---------------|----------|-----|
| 1 | 50421651 | Indah Dwi ... | 7 | 3.9 |

NPM

Nama

Semester

IPK

Simpan

Buang

Refresh

| ID | NPM | Nama | Semester | IPK |
|----|----------|---------------|----------|-----|
| 1 | 50421651 | Indah Dwi ... | 7 | 3.9 |


Hapus Mahasiswa

Masukkan ID yang ingin dihapus

1

OK

Cancel



—

□

×

NPM

Nama

Semester

IPK


Simpan

Buang

Refresh


| ID | NPM | Nama | Semester | IPK |
|----|----------|---------------|----------|-----|
| 1 | 50421651 | Indah Dwi ... | 7 | 3.9 |

sukses



Data berhasil dihapus

OK



—

□

×

NPM

Nama

Semester

IPK

Simpan

Buang

Refresh

| ID | NPM | Nama | Semester | IPK |
|----|-----|------|----------|-----|
|----|-----|------|----------|-----|