

## LAPORAN AKHIR PRAKTIKUM

Mata Praktikum : Rekayasa Perangkat Lunak 2  
Kelas : 4IA18  
Praktikum ke- : 4  
Tanggal : 08 November 2025  
Materi : Konsep dasar ORM dan Framework Hibernate  
NPM : 50422215  
Nama : Andi Purnama  
Ketua Asisten : Haikal Abizar  
Paraf Asisten : -  
Nama Asisten : 1.  
               2.  
               3.  
Jumlah Lembar : 5 Lembar



**LABORATORIUM INFORMATIKA**

**UNIVERSITAS GUNADARMA**

**2025**

## **1. Jelaskan apa yang dimaksud dengan ORM dan keuntungannya!**

### **Jawab:**

ORM (Object Relational Mapping) adalah teknik atau pendekatan dalam pemrograman yang digunakan untuk menghubungkan antara objek-objek dalam bahasa pemrograman berorientasi objek (seperti Java, Python, atau C#) dengan tabel-tabel dalam database relasional (seperti MySQL, PostgreSQL, atau Oracle). Dengan ORM, pengembang tidak perlu lagi menulis query SQL secara langsung untuk melakukan operasi seperti menyimpan, mengambil, memperbarui, atau menghapus data. Sebaliknya, semua operasi tersebut dilakukan melalui objek dalam kode program yang secara otomatis dipetakan ke tabel dan kolom di dalam database.

Sebagai contoh, dalam konteks Hibernate (ORM di Java), sebuah kelas seperti ModelMahasiswa dapat dihubungkan langsung ke tabel mahasiswa di database, dan atribut-atribut kelas seperti npm, nama, atau ipk akan mewakili kolom-kolom dalam tabel tersebut. Ketika objek ModelMahasiswa disimpan melalui Hibernate, ORM akan secara otomatis mengubahnya menjadi perintah SQL INSERT tanpa perlu programmer menulis query tersebut secara manual.

Keuntungan utama penggunaan ORM adalah efisiensi dan kemudahan dalam pengembangan aplikasi. Pertama, ORM menghemat waktu dan mengurangi kode SQL manual, karena semua operasi database dilakukan melalui objek dan metode yang lebih mudah dipahami. Kedua, ORM meningkatkan portabilitas aplikasi, karena jika suatu saat sistem berpindah dari satu jenis database ke database lain, perubahan yang dibutuhkan sangat minimal. Ketiga, ORM meningkatkan keamanan dengan mengurangi risiko serangan seperti SQL Injection, sebab query dihasilkan otomatis oleh framework dan tidak melibatkan penyusunan string SQL secara langsung. Keempat, ORM mempermudah pemeliharaan kode karena struktur program menjadi lebih terorganisir dan konsisten antara objek di kode dengan tabel di database.

Secara singkat, ORM adalah jembatan antara dunia pemrograman berorientasi objek dan database relasional, yang memberikan kemudahan, efisiensi, keamanan, dan portabilitas dalam pengembangan aplikasi berbasis data.

## 2. Jelaskan kode dan langkah-langkah program yang telah dibuat!

**Jawab:**

**Program yang dibuat merupakan** aplikasi sederhana berbasis Java Swing dan Hibernate yang berfungsi untuk mengelola data mahasiswa melalui operasi CRUD (Create, Read, Update, Delete). Aplikasi ini menggunakan arsitektur Model-View-Controller (MVC) agar struktur kode lebih teratur dan mudah dikembangkan. Komponen utama yang terlibat meliputi kelas model, controller, view, serta utilitas Hibernate untuk menghubungkan aplikasi dengan database.

**Kelas ModelMahasiswa.java** berperan sebagai entitas yang merepresentasikan tabel mahasiswa di dalam database. Kelas ini menggunakan anotasi JPA seperti @Entity, @Table, @Id, dan @GeneratedValue untuk mendefinisikan pemetaan antara atribut di dalam program dengan kolom di tabel database. Setiap mahasiswa memiliki atribut seperti id, npm, nama, semester, dan ipk, lengkap dengan metode getter dan setter untuk mengakses serta memodifikasi data tersebut. Dengan kata lain, kelas ini menjadi representasi dari data mahasiswa yang akan disimpan atau diambil dari database.

**Kelas HibernateUtil.java** bertugas mengatur koneksi antara aplikasi dan database melalui konfigurasi Hibernate. Di dalamnya terdapat SessionFactory yang dibuat hanya sekali ketika program dijalankan, dan digunakan untuk membuka sesi (Session) setiap kali aplikasi ingin berinteraksi dengan database. Selain itu, disediakan juga metode testConnection() untuk memastikan koneksi berjalan dengan baik. Kelas ini menjadi fondasi utama agar Hibernate dapat beroperasi dengan benar dan melakukan komunikasi antara objek Java dan tabel database.

**Interface MahasiswaController.java** berfungsi mendefinisikan kontrak atau aturan tentang operasi yang dapat dilakukan terhadap data mahasiswa. Di dalamnya terdapat deklarasi metode seperti addMhs, getMhs, updateMhs, deleteMhs, dan getAllMahasiswa. Interface ini tidak berisi logika program, tetapi hanya menetapkan struktur agar implementasinya nanti lebih fleksibel. Tujuan penggunaan interface ini adalah untuk memisahkan antara definisi fungsi dan implementasinya sehingga program mudah diperluas dan dimodifikasi.

**Kelas MahasiswaControllerImpl.java** merupakan implementasi nyata dari interface MahasiswaController. Kelas ini menggunakan Hibernate untuk menjalankan operasi CRUD ke database. Metode addMhs digunakan untuk menyimpan data baru, updateMhs untuk memperbarui data yang sudah ada, deleteMhs untuk menghapus data berdasarkan ID, dan getAllMahasiswa untuk mengambil semua data dari tabel mahasiswa menggunakan perintah HQL (Hibernate Query Language). Setiap operasi dibungkus dalam transaksi (Transaction) agar data tetap konsisten. Jika terjadi kesalahan, transaksi akan di-rollback untuk mencegah kerusakan data. Kelas ini menjadi jembatan antara data model dengan tampilan pengguna.

**Kelas ModelTabelMahasiswa.java** digunakan untuk menampilkan data mahasiswa dalam bentuk tabel pada GUI. Kelas ini meng-extend AbstractTableModel sehingga dapat menghubungkan daftar objek ModelMahasiswa dengan komponen JTable di Java Swing. Di dalamnya terdapat pengaturan nama kolom, jumlah baris dan kolom, serta bagaimana data setiap mahasiswa akan ditampilkan di sel tabel melalui metode getValueAt. Selain itu, terdapat metode setMahasiswaList yang berfungsi memperbarui isi tabel ketika data diubah, ditambah, atau dihapus. Dengan demikian, kelas ini menjadi penghubung antara data yang ada di database dan tampilan tabel di aplikasi.

**Kelas MahasiswaView.java** merupakan tampilan antarmuka pengguna (GUI) yang dirancang menggunakan NetBeans GUI Builder. Tampilan ini menyediakan kolom isian untuk data mahasiswa seperti NPM, Nama, Semester, dan IPK, serta tiga tombol utama yaitu SAVE, REFRESH, dan DELETE. Tombol SAVE digunakan untuk menambahkan data mahasiswa baru, REFRESH untuk memperbarui tampilan tabel dengan data terbaru dari database, dan DELETE untuk menghapus data mahasiswa yang dipilih. Di bagian bawah terdapat sebuah JTable yang menampilkan daftar mahasiswa. View ini berinteraksi dengan controller untuk menjalankan operasi CRUD, di mana pengguna cukup berinteraksi melalui antarmuka tanpa perlu mengetahui proses yang terjadi di belakang layar.

Secara keseluruhan, program ini bekerja dengan alur ketika aplikasi dijalankan, Hibernate akan melakukan koneksi ke database melalui HibernateUtil. Pengguna dapat mengisi data mahasiswa melalui tampilan MahasiswaView dan menekan tombol SAVE untuk menyimpan data. Data yang dimasukkan akan dikirim ke MahasiswaControllerImpl untuk disimpan ke database sebagai objek ModelMahasiswa. Setelah itu, data akan dimuat kembali dan ditampilkan di tabel menggunakan ModelTabelMahasiswa. Tombol REFRESH digunakan untuk memuat ulang data dari database, dan tombol DELETE digunakan untuk menghapus data tertentu. Dengan kombinasi ini, aplikasi berhasil memisahkan antara logika bisnis, pengelolaan data, dan tampilan sehingga menjadi aplikasi CRUD berbasis Java yang terstruktur dengan baik.

