**Judul Tugas:**

Pembuatan Aplikasi Web CRUD Menggunakan Spring Boot: Sistem Manajemen Proyek dengan Relasi Many-to-One

**Tujuan Pembelajaran:**

Mahasiswa diharapkan mampu memahami dan mengimplementasikan dasar-dasar pengembangan aplikasi web berbasis Java menggunakan Spring Boot dan Spring Data JPA, dengan fokus pada:

* Membangun aplikasi web berbasis Java menggunakan Spring Boot.
* Membuat entitas dan relasi antar model menggunakan JPA dan anotasi @ManyToOne.
* Mengimplementasikan operasi CRUD melalui REST API.
* Menggunakan database MySQL sebagai penyimpanan data.
* Menambahkan fitur pencarian (search) dan pengurutan data (sorting) di beberapa endpoint.

**Struktur Folder**

Gunakan struktur package seperti berikut:

* controller → berisi semua controller untuk endpoint
* model → berisi entity kelas
* repository → berisi interface repository
* resources → konfigurasi seperti application.properties

**Langkah-Langkah Pembuatan**

1. Buat Proyek Spring Boot Baru

Gunakan Spring Initializr dengan dependencies: Spring Web, Spring Data JPA, MySQL Driver, Lombok, dan Spring DevTools.

1. Buat 5 Entity (Model)

* Company: atribut id, name, dan address.
* Department: memiliki name dan relasi ke Company (Many-to-One).
* Employee: memiliki fullName, role, dan relasi ke Department (Many-to-One).
* Project: memiliki title, description, dan relasi ke Department (Many-to-One).
* Task: memiliki name, status, dan relasi ke Project (Many-to-One).

Setiap entity menggunakan anotasi @Entity, memiliki @Id dan @GeneratedValue, serta getter-setter otomatis dengan Lombok.

1. Buat Repository untuk Setiap Entity

* Buat interface yang extend JpaRepository untuk Company, Department, Employee, Project, dan Task.
* Tambahkan method custom untuk fitur search dan sorting di EmployeeRepository dan ProjectRepository.

1. Buat Controller untuk Masing-Masing Entity

* Setiap controller memiliki endpoint:
* GET /all untuk mendapatkan semua data
* GET /by-id/{id} untuk mendapatkan satu data
* POST /create untuk menambahkan data baru
* PUT /update/{id} untuk memperbarui data
* DELETE /delete/{id} untuk menghapus data
* Tambahan:
* EmployeeController dan ProjectController dilengkapi endpoint pencarian berdasarkan nama/title (/search/{keyword}).
* Ditambahkan juga endpoint sorting secara ascending dan descending.

1. Konfigurasi Database

* Edit file application.properties agar terhubung ke database MySQL.
* Gunakan spring.jpa.hibernate.ddl-auto=update agar tabel dibuat otomatis.

1. Testing Endpoint

* Gunakan Postman untuk menguji semua endpoint.
* Lakukan testing terhadap semua operasi CRUD, fitur pencarian dan sorting.
* Gunakan format JSON pada request POST dan PUT.