

UJIAN AKHIR SEMESTER

Mata Kuliah: Praktikum Pengembangan Desain Web

Sifat: Take Home Project

Waktu: 3 Hari

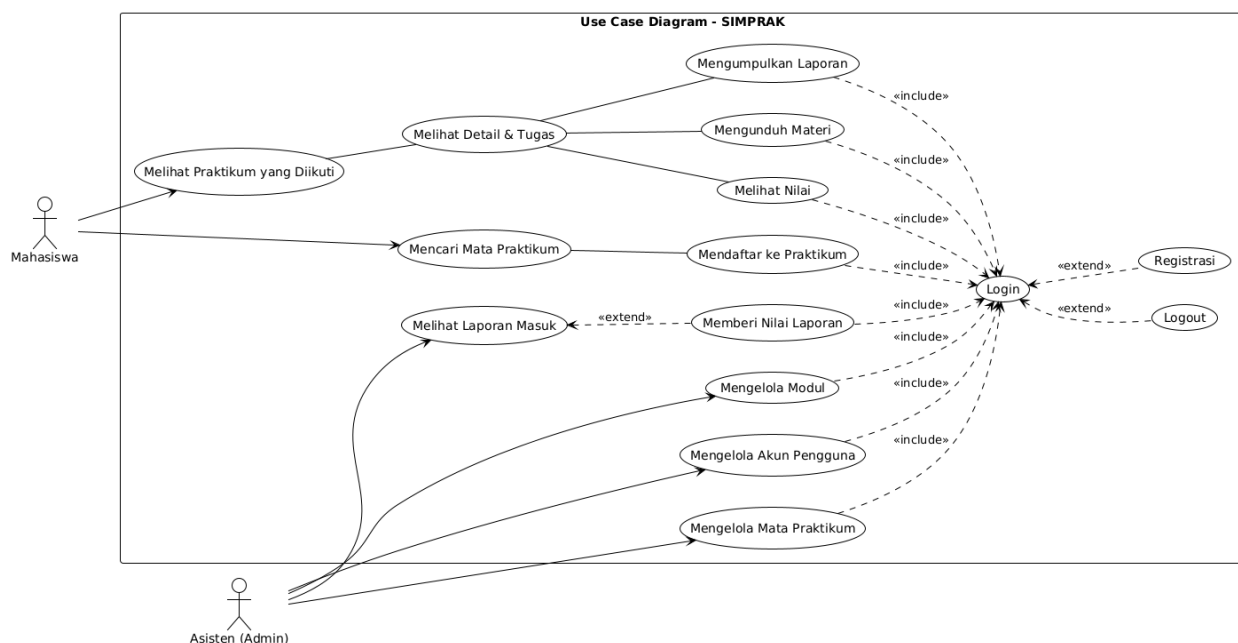
A. Latar Belakang

Manajemen kegiatan praktikum di sebuah institusi pendidikan seringkali masih dilakukan secara semi-manual. Proses pembagian materi, pengumpulan laporan, dan penilaian oleh asisten praktikum dapat menjadi tidak efisien jika tidak dikelola dalam satu sistem terpadu. Untuk mengatasi hal ini, diperlukan sebuah **Sistem Informasi Manajemen Praktikum (SIMPRAK)** berbasis web.

B. Tujuan Proyek

Mahasiswa diminta untuk merancang dan membangun sebuah aplikasi web fungsional dari awal (from scratch) yang mengimplementasikan semua fungsionalitas yang tertera pada diagram *Use Case* yang diberikan. Aplikasi ini harus dapat melayani dua jenis pengguna: **Mahasiswa** dan **Asisten**.

C. Diagram Use Case Acuan



D. Spesifikasi Fungsional Wajib

Aplikasi yang Anda bangun **wajib** memiliki semua fungsionalitas yang digambarkan dalam use case diagram. Rinciannya adalah sebagai berikut:

Modul 1: Fitur Umum & Autentikasi

Silakan lakukan *clone* terhadap proyek dari URL berikut:
<https://github.com/ramaravictor/SistemPengumpulanTugas>.

Proyek ini sudah memiliki fitur login, register, logout, serta halaman utama untuk mahasiswa dan asisten.

Lanjutkan pengembangan proyek ini dengan menambahkan fitur-fitur lainnya sesuai dengan *use case diagram* dan penjelasan yang terdapat pada Modul 2 dan Modul 3.

Modul 2: Fungsionalitas Mahasiswa

1. **Mencari Mata Praktikum:** Halaman katalog yang menampilkan semua mata praktikum yang tersedia di sistem. Halaman ini bisa diakses publik.
2. **Mendaftar ke Praktikum:** Mahasiswa yang sudah login dapat mendaftar ke mata praktikum yang ada di katalog.
3. **Melihat Praktikum yang Diikuti:** Halaman personal yang hanya menampilkan daftar mata praktikum yang sudah diikuti oleh mahasiswa tersebut.
4. **Melihat Detail & Tugas:** Dari halaman "Praktikum Saya", mahasiswa bisa mengklik salah satu mata praktikum untuk melihat detailnya, yang berisi:
 - **Mengunduh Materi:** Terdapat daftar modul/pertemuan dengan link untuk mengunduh file materi yang diunggah oleh asisten.
 - **Mengumpulkan Laporan:** Untuk setiap modul, terdapat form untuk mengunggah file laporan/tugas.
 - **Melihat Nilai:** Menampilkan nilai yang sudah diberikan oleh asisten untuk laporan yang telah dikumpulkan.

Modul 3: Fungsionalitas Asisten (Admin)

1. **Mengelola Mata Praktikum:**
 - Tampilan Fitur CRUD (Create, Read, Update, Delete) penuh untuk data master mata praktikum (misal: "Pemrograman Web", "Jaringan Komputer").
2. **Mengelola Modul:**
 - Asisten dapat menambahkan, mengubah, atau menghapus modul/pertemuan untuk setiap mata praktikum yang mereka ampu.
 - Saat menambah/mengubah modul, Asisten dapat mengunggah file materi (PDF/DOCX).
3. **Melihat Laporan Masuk:**
 - Menampilkan daftar semua laporan yang telah dikumpulkan oleh mahasiswa dari berbagai modul.

- Wajib ada fitur filter (berdasarkan modul, mahasiswa, atau status).
- 4. **Memberi Nilai Laporan:**
 - Dari halaman laporan masuk, asisten dapat melihat detail pengumpulan per mahasiswa.
 - Asisten dapat mengunduh file laporan yang dikumpulkan mahasiswa.
 - Asisten dapat memasukkan nilai (angka) dan feedback (teks) untuk laporan tersebut.
- 5. **Mengelola Akun Pengguna:**
 - Asisten (Admin) dapat melihat semua akun yang terdaftar.
 - Fitur CRUD (Create, Read, Update, Delete) untuk semua akun pengguna (Mahasiswa dan Asisten lainnya).

E. Kebutuhan Teknis

- **Bahasa & Framework:** PHP Native (tanpa framework).
- **Database:** MySQL atau MariaDB.
- **Frontend:** HTML dan CSS. Sangat disarankan menggunakan **Tailwind CSS** untuk styling agar sesuai dengan desain modern.

F. Kriteria Penilaian

- **Kelengkapan Fungsionalitas (60%):**
 - **Wajib** ada **minimal 1 fitur yang menjalankan backend CRUD** (Create, Read, Update, Delete).
 - **Bonus poin** jika **seluruh alur kerja aplikasi** (dari registrasi hingga penilaian) **berjalan lancar tanpa error**.
- **Kualitas Kode dan Struktur (20%):**
 - Kode terstruktur rapi (misal: pemisahan file untuk template, logika, dll).
 - Tidak semua kode ditulis dalam satu file.
- **Desain Antarmuka & Pengalaman Pengguna (UI/UX) (20%):**
 - **Semua tampilan dari use case diagram harus ada dan bisa diakses.**
→ Jadi **semua halaman UI harus dibuat**, walaupun tidak semua punya backend.
 - Tampilan bersih, modern, dan responsif.
 - Navigasi mudah: pengguna **bisa pindah antar halaman dengan jelas**.

Selamat Mengerjakan!