

Tugas Quis 3

Nama : Tiurma Taqiyyahafizh

Kelas : TI-23-PA

NPM : 232310019

Proyek Akhir Pemrograman Web: FreeClass

Dalam proyek akhir pemrograman web saya, saya mengembangkan sebuah aplikasi bernama **FreeClass**. Ini adalah sistem yang dirancang untuk **mempermudah dan mengefisienkan proses pemesanan ruang kelas** di lingkungan akademik. Ide utamanya adalah menghilangkan kerumitan administrasi manual dan potensi konflik jadwal yang sering terjadi dalam peminjaman ruang.

Deskripsi Singkat Sistem

FreeClass adalah sistem yang membagi pengalaman pengguna berdasarkan peran:

- **Untuk admin**, ada **website desktop** yang berfungsi sebagai panel kontrol utama. Di sini, admin bisa mengelola semua aspek sistem, dari data ruangan hingga verifikasi pemesanan.
- **Untuk pengguna (user)** seperti mahasiswa dan dosen, tersedia **aplikasi mobile**. Melalui aplikasi ini, mereka bisa dengan mudah memesan ruang kelas kapan saja dan di mana saja.

Tujuan utama FreeClass adalah menciptakan proses peminjaman ruang yang **cepat, aman, transparan, dan fleksibel**, yang pada akhirnya akan meningkatkan produktivitas akademik.

Fitur-Fitur Utama

FreeClass dilengkapi dengan fitur-fitur esensial untuk mendukung alur kerja yang mulus:

Untuk Admin (via Website Desktop):

- **Manajemen Pengguna:** Admin dapat dengan aman masuk dan keluar dari sistem.
- **Manajemen Ruangan:** Admin memiliki kontrol penuh untuk menambah, mengedit, atau menghapus detail ruangan yang tersedia (misalnya, kapasitas, fasilitas, ketersediaan).
- **Manajemen Pemesanan:** Admin dapat melihat semua permintaan pemesanan dari user dan memiliki wewenang untuk menyetujui atau menolak pemesanan tersebut.
- **Laporan & Monitoring:** Tersedia rekap penggunaan ruang yang dapat difilter berdasarkan tanggal, nama ruangan, dan status pemesanan, membantu dalam analisis dan perencanaan.
- **Keamanan Akses:** Akses ke sistem admin dijamin melalui autentikasi login yang ketat.

Untuk User (Mahasiswa/Dosen via Aplikasi Mobile):

- **Pemesanan Ruangan Cepat:** User bisa langsung memesan ruangan tanpa perlu login, hanya dengan mengisi detail pemesanan seperti nama, keperluan, tanggal, waktu, dan pilihan ruangan.

- **Cek Status Pemesanan:** User dapat dengan mudah melacak status pemesanan mereka (disetujui/ditolak) menggunakan ID pemesanan atau nama.
- **Tampilan Status Instan:** Aplikasi memberikan notifikasi atau tampilan status langsung setelah pemesanan dikirim, dan status ini bisa dicek ulang kapan saja.
- **Antarmuka Sederhana & Responsif:** Aplikasi dirancang agar mudah digunakan, dengan navigasi intuitif, tampilan bersih, dan akses cepat tanpa proses login yang rumit.

Target Pengguna

Target pengguna utama dari aplikasi FreeClass adalah:

- **Mahasiswa:** Mereka akan menggunakan aplikasi untuk memesan ruang belajar kelompok, ruang diskusi, atau untuk kegiatan organisasi.
- **Dosen:** Dosen dapat memanfaatkan aplikasi untuk memesan ruang kelas tambahan, ruang bimbingan, atau pertemuan di luar jam perkuliahan reguler.
- **Admin/Staf Akademik:** Pihak yang bertanggung jawab atas pengelolaan fasilitas dan penjadwalan ruang akan menggunakan website desktop untuk memantau, mengelola, dan memverifikasi semua pemesanan.

Secara keseluruhan, FreeClass dibuat untuk seluruh civitas akademika yang membutuhkan akses mudah dan terorganisir ke fasilitas ruang kelas.

Tentu, saya akan jelaskan tujuan proyek akhir saya, FreeClass, dan bagaimana implementasi Laravel di dalamnya.

Tujuan Proyek Akhir: FreeClass

Proyek akhir saya, FreeClass, dikembangkan dengan tujuan utama untuk **menyelesaikan permasalahan yang ada dalam proses pemesanan ruang kelas tradisional** dan memberikan dampak positif bagi lingkungan akademik.

Permasalahan yang Ingin Diselesaikan

Sebelum adanya FreeClass, proses pemesanan ruang kelas sering kali diwarnai oleh berbagai kendala:

- **Proses Manual yang Rumit:** Pemesanan sering melibatkan formulir fisik, konfirmasi via telepon atau email, dan pencatatan manual, yang memakan waktu dan rentan kesalahan.
- **Kurangnya Transparansi Informasi:** Pengguna sulit mengetahui ketersediaan ruang secara *real-time*, sering kali harus menghubungi pihak administrasi berkali-kali.
- **Konflik Jadwal:** Tanpa sistem terpusat, sering terjadi bentrok jadwal atau pemesanan ganda untuk ruangan yang sama, menimbulkan kebingungan dan ketidaknyamanan.
- **Akses Terbatas:** Proses pemesanan hanya bisa dilakukan pada jam kerja atau di lokasi tertentu, membatasi fleksibilitas pengguna.
- **Pengelolaan Data yang Tidak Efisien:** Admin kesulitan melacak riwayat pemakaian ruang dan membuat laporan, karena data tersebar atau tidak terdigitalisasi dengan baik.

Tujuan Utama Pengembangan Aplikasi

Berdasarkan permasalahan di atas, tujuan utama pengembangan FreeClass adalah:

1. **Menciptakan Sistem Pemesanan Ruang yang Efisien dan Terstruktur:** Menyediakan platform digital yang memungkinkan pemesanan ruang kelas dilakukan dengan cepat, akurat, dan minim kesalahan.
2. **Meningkatkan Transparansi Informasi Ketersediaan Ruang:** Memastikan pengguna dapat melihat status ketersediaan ruang secara *real-time* sebelum melakukan pemesanan.
3. **Meminimalisir Konflik Jadwal Pemesanan:** Menerapkan logika sistem yang secara otomatis mencegah tumpang tindih pemesanan.
4. **Menyediakan Akses Fleksibel untuk Pengguna:** Memungkinkan mahasiswa dan dosen memesan ruang kapan saja dan di mana saja melalui aplikasi *mobile*.
5. **Mempermudah Pengelolaan dan Monitoring Ruang bagi Admin:** Memberikan alat yang komprehensif bagi admin untuk mengelola data ruangan, memverifikasi pemesanan, dan membuat laporan penggunaan.

Dampak/Manfaat Aplikasi terhadap Pengguna

Pengembangan FreeClass diharapkan membawa dampak dan manfaat signifikan bagi seluruh penggunanya:

- **Bagi Mahasiswa dan Dosen:**
 - **Kemudahan dan Kecepatan Pemesanan:** Mereka dapat memesan ruang dalam hitungan menit tanpa prosedur manual yang rumit, menghemat waktu dan tenaga.
 - **Fleksibilitas:** Dapat melakukan pemesanan dari mana saja, kapan saja, sesuai kebutuhan.
 - **Kepastian Informasi:** Langsung mengetahui status pemesanan mereka (disetujui/ditolak) dan ketersediaan ruang.
 - **Peningkatan Produktivitas Akademik:** Dengan proses yang mulus, mereka bisa lebih fokus pada kegiatan belajar mengajar tanpa terganggu urusan administrasi ruang.
- **Bagi Admin/Staf Akademik:**
 - **Efisiensi Pengelolaan:** Proses verifikasi dan pengelolaan data ruangan menjadi lebih cepat dan terorganisir.
 - **Pengurangan Beban Kerja Administratif:** Mengurangi *input* data manual dan penanganan konflik jadwal.
 - **Data yang Akurat dan Terpusat:** Memudahkan pembuatan laporan penggunaan ruang dan analisis data untuk pengambilan keputusan.
 - **Keamanan Data:** Sistem login admin memastikan hanya pihak yang berwenang yang dapat mengakses dan mengelola data penting.

Implementasi Laravel dalam Proyek FreeClass

Ya, proyek akhir saya **menggunakan framework Laravel** untuk pengembangan sisi *backend* website admin dan API untuk aplikasi *mobile*. Laravel dipilih karena kemudahan pengembangan, fitur-fitur lengkap, dan keamanan yang terjamin.

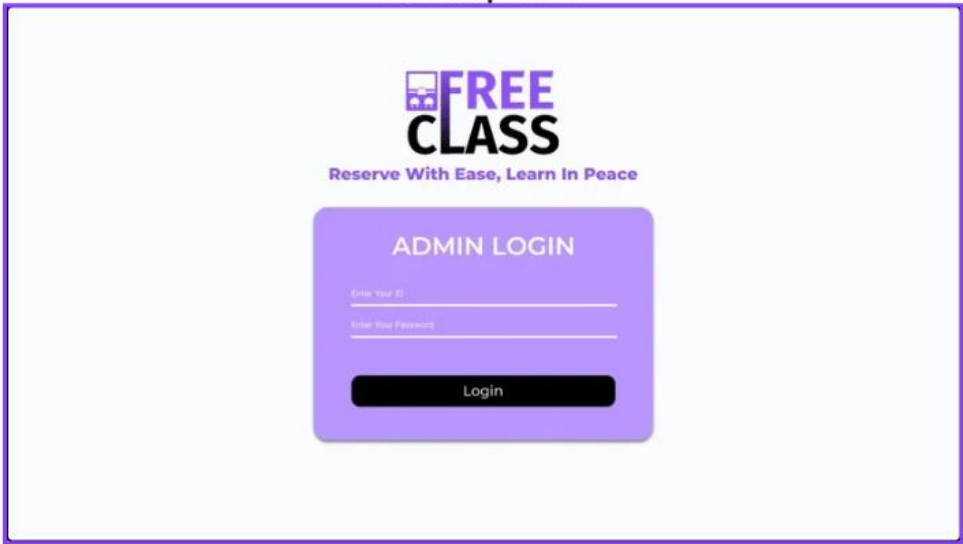
Berikut adalah beberapa implementasi Laravel dalam proyek FreeClass:

- **Routing:** Laravel digunakan untuk mendefinisikan semua rute (URL) yang diakses oleh website admin dan API. Misalnya, ada rute untuk halaman *login* admin (`/admin/login`), rute untuk melihat daftar ruangan (`/admin/rooms`), atau rute API untuk pemesanan ruangan (`/api/bookings/store`).
- **Controller:** Setiap rute akan diarahkan ke metode spesifik dalam **Controller**. Contohnya, ada `AuthController` untuk menangani proses *login* dan *logout* admin, `RoomController` untuk logika tambah, edit, hapus ruangan, dan `BookingController` untuk memproses permintaan pemesanan dari aplikasi *mobile* dan verifikasi oleh admin.
- **Model:** Laravel menyediakan **Eloquent ORM (Object-Relational Mapper)** yang memudahkan interaksi dengan *database*. Setiap tabel *database* diwakili oleh sebuah Model. Misalnya, ada `Room Model` untuk tabel `rooms`, `User Model` untuk tabel `users` (admin), dan `Booking Model` untuk tabel `bookings`. Model ini juga mendefinisikan relasi antar tabel (misalnya, satu ruangan bisa memiliki banyak pemesanan).
- **Migration:** Struktur *database* FreeClass dibuat dan dikelola menggunakan **Migrations** Laravel. Ini memungkinkan saya untuk membuat tabel seperti `rooms`, `users`, dan `bookings` dengan kolom-kolom yang diperlukan, serta mengelola perubahan skema *database* secara terstruktur dan dapat direvisi. Ini sangat membantu dalam kolaborasi tim dan *deployment*.
- **Blade Template:** Untuk tampilan *interface* website admin, saya menggunakan **Blade Template Engine** milik Laravel. Blade memungkinkan penulisan *view* yang bersih dan efisien dengan *syntax* yang sederhana untuk *looping*, *conditional statements*, dan *layout inheritance*. Misalnya, ada *template* untuk *layout* dasar admin, *view* untuk daftar ruangan, *form* tambah/edit ruangan, dan daftar pemesanan.
- **Authentication & Authorization:** Laravel menyediakan fitur **Authentication** bawaan yang sangat robust. Saya mengimplementasikannya untuk sistem *login* admin, memastikan hanya admin yang terautentikasi yang bisa mengakses fitur manajemen. Untuk **Authorization**, saya juga bisa menggunakan *middleware* atau *gates* Laravel untuk membatasi akses ke fitur tertentu hanya untuk peran admin.
- **API Resources:** Untuk komunikasi antara *backend* Laravel dan aplikasi *mobile*, saya mengembangkan **API** menggunakan Laravel. Saya memanfaatkan **API Resources** untuk memformat data yang dikirim dari *backend* ke aplikasi *mobile* menjadi bentuk JSON yang konsisten dan mudah dikonsumsi, misalnya ketika aplikasi *mobile* meminta daftar ruangan yang tersedia.
- **Validation:** Laravel memiliki fitur validasi *input* yang kuat. Saya menggunakannya untuk memvalidasi data yang masuk, baik dari *form* admin maupun dari API aplikasi *mobile*, memastikan data yang disimpan ke *database* selalu valid dan konsisten (misalnya, memastikan semua kolom wajib diisi, format tanggal benar, dll.).

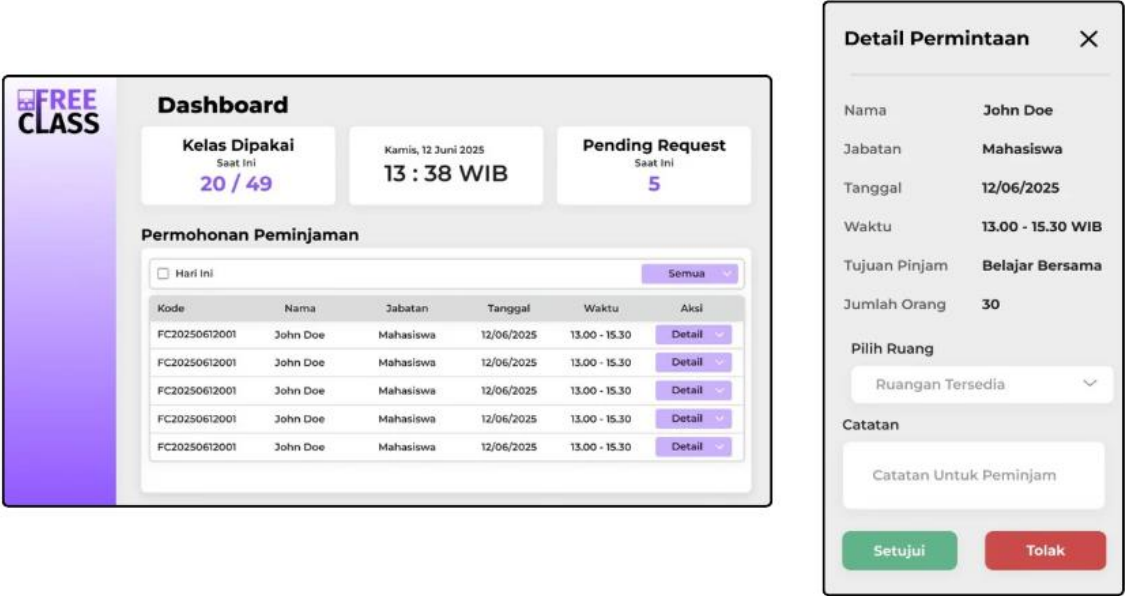
Dengan Laravel, pengembangan FreeClass menjadi lebih cepat, terstruktur, dan mudah dikelola, baik untuk sisi website admin maupun sebagai *backend* untuk API aplikasi *mobile*.

WEB

Mockup Web



Mockup Web



MOBILE

