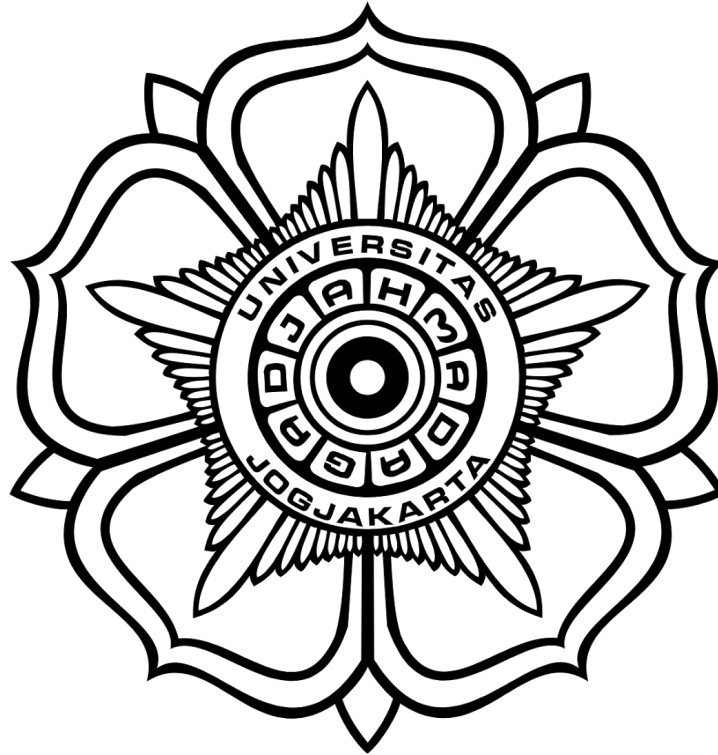


**Clinic Appointment System:  
Handle Patient Registration, Doctor Schedules, and  
Appointments.**



**Erdizah Ghodi Al Haidar (24/537670/PA/22787)**

**Giganius Revo (24/541359/PA/22965)**

**Nicholas Lim (24/543264/PA/23075)**

**DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS GADJAH MADA**

**2025**

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Pada masa sekarang, kebutuhan akan layanan kesehatan di masyarakat terus meningkat dengan cepat, oleh karena itu diperlukan layanan kesehatan yang cepat, efisien, dan terorganisir. Klinik menjadi salah satu penyedia layanan kesehatan yang dapat menjangkau masyarakat secara luas bahkan di daerah-daerah pelosok sekalipun. Namun, masih banyak klinik yang menggunakan sistem pencatatan dan pengelolaan janji temu pasien yang manual dan tradisional. Sistem yang tradisional ini biasanya menggunakan pencatatan secara fisik dengan buku penjadwalan atau komunikasi langsung melalui telepon.

Sistem yang manual dan tradisional ini tentunya memiliki kelemahan yaitu dalam hal kecepatan dan efisiensi. Pasien harus datang secara langsung untuk mengecek ketersediaan jadwal dokter dan melakukan janji temu. Hal ini bisa menimbulkan beberapa masalah seperti antrian yang panjang dan mengakibatkan penanganan pasien yang kurang cepat dan efisien. Sistem seperti ini juga rentan akan kesalahan pencatatan jadwal atau pencatatan janji temu yang dapat menimbulkan berbagai masalah lainnya.

Maka dari itu, diperlukan sebuah sistem basis data modern yang menyediakan informasi ketersediaan jadwal dokter dan pendaftaran serta pencatatan janji temu. Sistem ini akan menggunakan bahasa pemrograman SQL yang berfungsi untuk meningkatkan efisiensi pengelolaan data, memastikan keakuratan informasi, dan mempercepat proses administrasi klinik. Dengan adanya sistem basis data modern, klinik dapat mengurangi risiko kesalahan pencatatan, memberikan kemudahan bagi pasien untuk melihat ketersediaan jadwal dokter dan membuat janji temu dengan dokter serta membantu tenaga medis dan juga staf klinik untuk mengatur alur kerja mereka supaya lebih efektif. Pada akhirnya, sistem basis data modern ini diharapkan dapat meningkatkan kualitas pelayanan kesehatan, mempercepat proses pelayanan dalam sebuah klinik serta memberikan kemudahan bagi semua orang yang terlibat.

### **1.2 Rumusan Masalah**

1. Bagaimana cara merancang dan membangun sistem basis data yang dapat mengelola data pasien, dokter, dan jadwal temu secara terhubung, sekaligus mencegah terjadinya benturan jadwal antara satu pasien dengan dokter lain?
2. Bagaimana sistem ini dapat membantu proses pembuatan, pengubahan, dan pembatalan janji temu antara pasien dan dokter agar lebih mudah dan efisien?
3. Bagaimana sistem basis data dapat mempermudah kegiatan administrasi klinik, seperti pendaftaran pasien baru, konfirmasi janji temu, dan pencatatan hasil konsultasi, sehingga layanan menjadi lebih cepat, akurat, dan teratur?

### **1.3 Tujuan**

1. Membangun sistem basis data terintegrasi yang dapat mengelola data pasien, jadwal praktek dokter, dan informasi janji temu yang efisien dan akurat.
2. Menyediakan mekanisme pencatatan, pembuatan, pengubahan dan pembatalan janji temu dengan lebih mudah, efisien, cepat, dan akurat sehingga memudahkan pasien dan tenaga medis.
3. Menyediakan data sebagai landasan yang akurat dan terstruktur untuk pengambilan keputusan di lingkungan klinik.

### **1.4 Batasan Masalah**

1. Membangun sistem basis data terintegrasi untuk pengelolaan layanan klinik.
2. Menyediakan mekanisme pembuatan, pengubahan, dan pembatalan janji temu secara efisien.
3. Menyediakan data yang terstruktur dan akurat sebagai dasar pengambilan keputusan di lingkungan klinik.

## **BAB II**

### **SYSTEM USERS DAN SYSTEM OBJECTIVES**

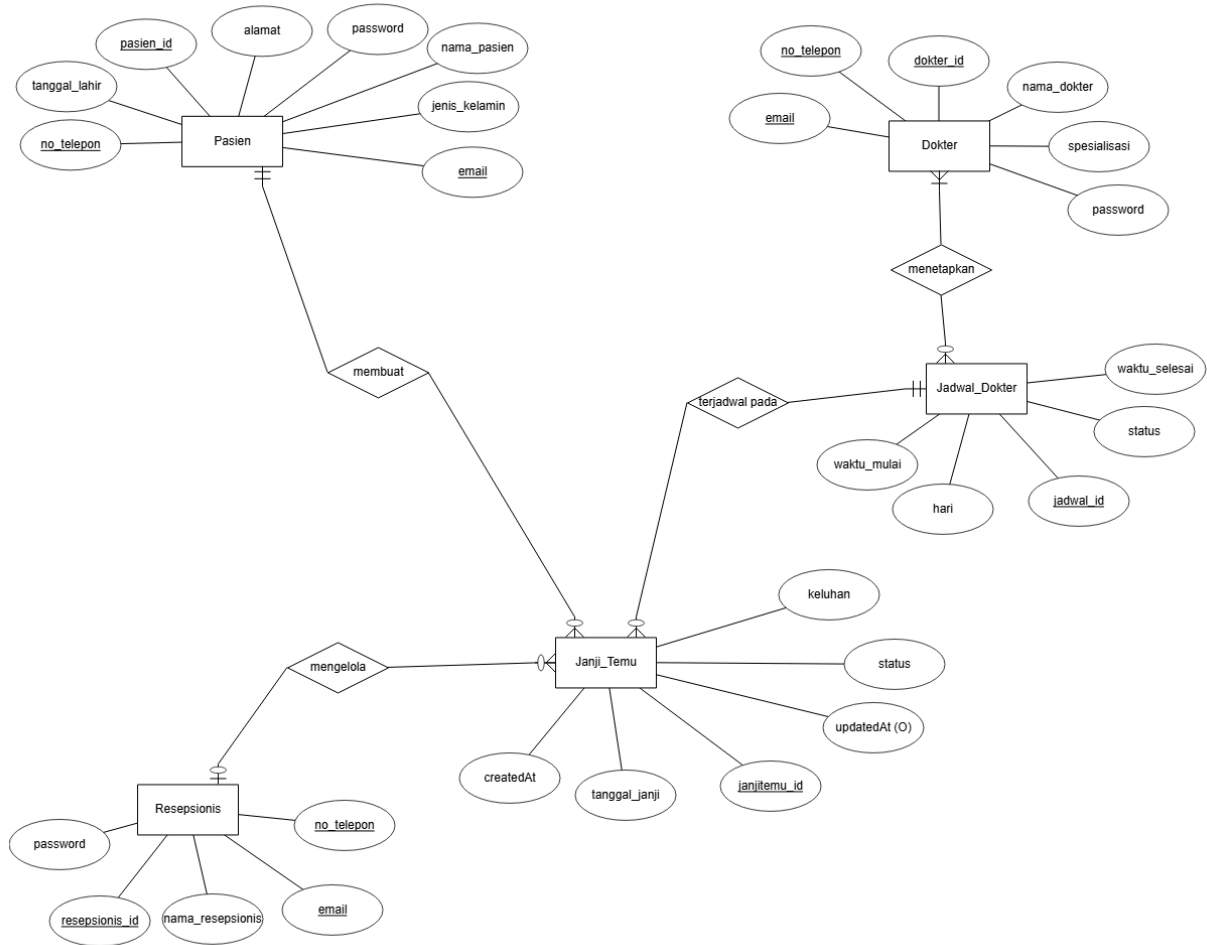
#### **2.1 System Users**

- a. Pasien:
  - i. Dapat mendaftar akun menggunakan nama, tanggal lahir, jenis kelamin, alamat, nomor telepon, email, dan password.
  - ii. Dapat melihat jadwal dokter yang tersedia berdasarkan data pada tabel Jadwal\_Dokter.
  - iii. Dapat membuat janji temu (booking appointment) melalui tabel Janji\_Temu.
  - iv. Dapat melihat riwayat janji temu yang telah dibuat beserta statusnya (Menunggu, Dikonfirmasi, Selesai, Dibatalkan).
  - v. Dapat mengubah atau memperbarui informasi profil pribadi seperti alamat, nomor telepon, dan email
- b. Dokter:
  - i. Dapat melihat daftar janji temu pasien yang terhubung dengan jadwal mereka melalui tabel Menetapkan dan Janji\_Temu.
  - ii. Dapat mengatur ketersediaan jadwal praktek (menandai jadwal sebagai Tersedia, Penuh, atau Libur).
  - iii. Dapat mengonfirmasi atau menolak janji temu pasien yang terjadwal.
- c. Resepsionis (Admin):
  - i. Dapat melihat seluruh data janji temu dari semua pasien dan dokter.
  - ii. Dapat memperbarui status janji temu secara manual (misalnya, dari Menunggu menjadi Dikonfirmasi atau Selesai).
  - iii. Dapat membantu pasien mendaftar atau menjadwalkan janji temu jika pasien datang langsung ke klinik.
  - iv. Dapat memantau jadwal dokter dan ketersediaan waktu praktek.

#### **2.2 System Objectives**

- a. Menyediakan sistem registrasi pasien digital yang memungkinkan penyimpanan data pribadi secara aman dan efisien di basis data (Pasien).
- b. Mengelola pembuatan dan pengaturan janji temu antara pasien dan dokter secara otomatis untuk meminimalkan bentrok jadwal (Janji\_Temu, Jadwal\_Dokter).
- c. Mempermudah dokter dalam mengelola jadwal praktek, dengan kemampuan mengubah status jadwal sesuai kondisi aktual.
- d. Meningkatkan efisiensi administrasi klinik dengan memberi akses bagi resepsionis untuk mengatur janji temu, memantau jadwal, dan mengonfirmasi status layanan.

- e. Menjamin transparansi dan keteraturan data, di mana setiap perubahan status atau informasi janji temu terekam otomatis melalui sistem (terbantu oleh trigger `trigger_updateJanji`).



Gambar 2.2 Entity Relationship Diagram

## **BAB III METODOLOGI**

### **3.1 Arsitektur Sistem**

Sistem Clinic Appointment Management System dikembangkan menggunakan arsitektur client-server yang terdiri dari tiga lapisan utama, yaitu frontend, backend, dan basis data. Frontend berfungsi sebagai antarmuka pengguna yang memungkinkan interaksi antara pengguna dan sistem. Backend berperan sebagai pengelola logika aplikasi dan penghubung antara frontend dan basis data. Basis data berfungsi sebagai media penyimpanan seluruh data sistem secara terpusat.

Alur kerja sistem dimulai dari permintaan yang dikirimkan oleh pengguna melalui frontend, kemudian diproses oleh backend, dan selanjutnya dilakukan pengambilan atau penyimpanan data pada basis data MySQL.

### **3.2 Entity Relationship Diagram (ERD)**

Entity Relationship Diagram (ERD) adalah representasi visual dari struktur basis data yang menunjukkan entitas, atribut, dan hubungan antar entitas. Pada sistem basis data *clinic appointment* ini, ERD digunakan untuk memodelkan hubungan antara pasien, dokter, resepsionis, jadwal dokter, dan janji temu.

### **3.3 Skema Relasional**

Skema relasional merupakan hasil translasi dari ERD ke dalam bentuk tabel-tabel relasional yang bisa diimplementasikan pada SQL. Pada skema relasional, setiap entitas diterjemahkan menjadi tabel, atribut dari entitas akan menjadi kolom dan relasi antar relasi akan direpresentasikan dalam bentuk foreign key pada tabel.

### **3.4 Implementasi Sistem**

Implementasi sistem dilakukan menggunakan MySQL sebagai basis data, Node.js dan Express.js sebagai backend, serta antarmuka berbasis web sebagai frontend. Backend menyediakan Application Programming Interface (API) untuk menangani proses autentikasi, pengelolaan data, serta komunikasi dengan basis data.

### **3.5 Aturan Bisnis dan Trigger**

Sistem menerapkan beberapa aturan bisnis utama, antara lain satu dokter tidak dapat memiliki lebih dari satu janji temu pada waktu yang sama. Selain itu, status janji temu dikelola secara terstruktur. Trigger pada basis data digunakan untuk memperbarui status jadwal dokter secara otomatis berdasarkan perubahan status janji temu.

## **BAB IV**

### **PENGUJIAN DAN PEMBAHASAN**

#### **4.1 Pengujian Sistem**

Pengujian sistem dilakukan dengan menjalankan beberapa skenario, seperti registrasi pengguna, pembuatan janji temu, konfirmasi janji temu oleh dokter, serta pengelolaan data oleh resepsionis. Hasil pengujian menunjukkan bahwa sistem dapat berjalan sesuai dengan fungsi yang dirancang.

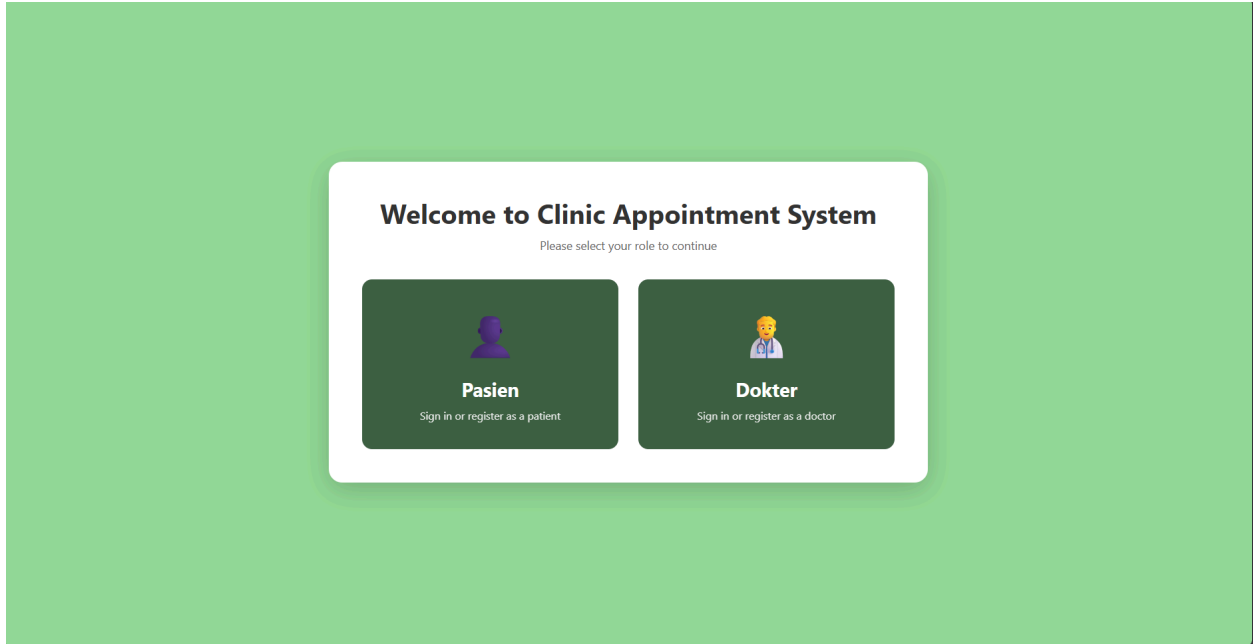
#### **4.2 Pembahasan**

Berdasarkan hasil pengujian, sistem mampu mendukung proses administrasi klinik secara lebih terstruktur dibandingkan metode manual. Namun, sistem masih memiliki keterbatasan, seperti belum diterapkannya mekanisme transaksi basis data secara penuh dan pengamanan tambahan terhadap akses langsung ke basis data.

## BAB V

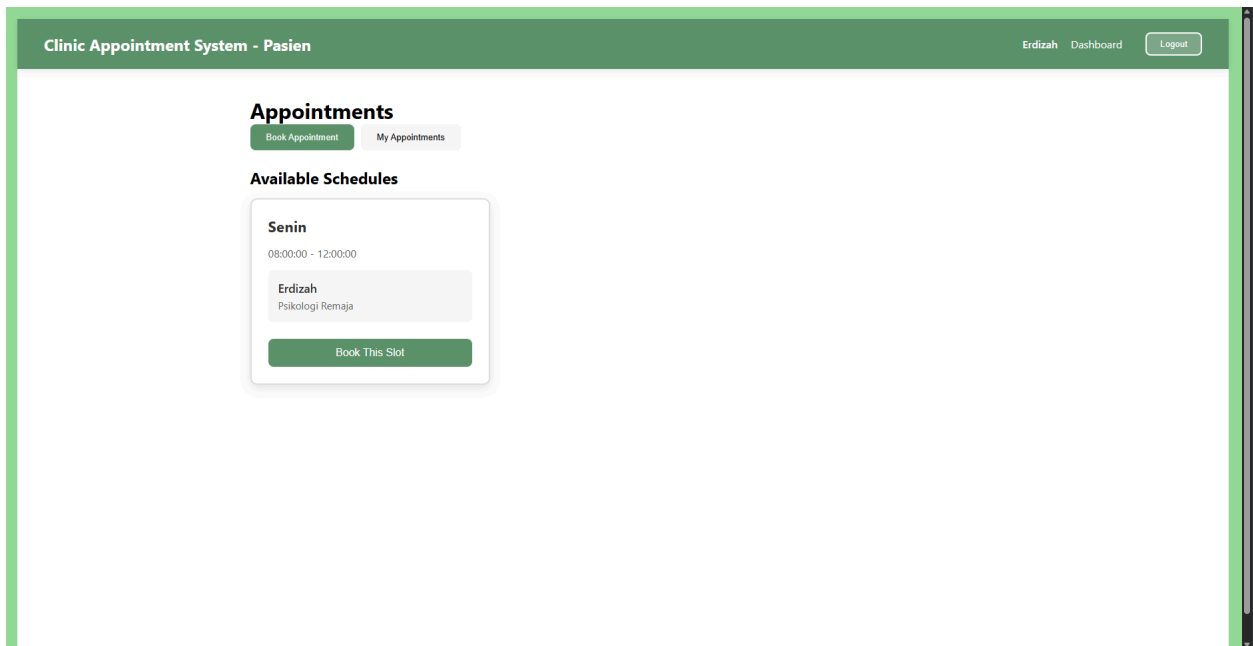
# DOKUMENTASI MODUL UTAMA

### 5.1 Modul Login



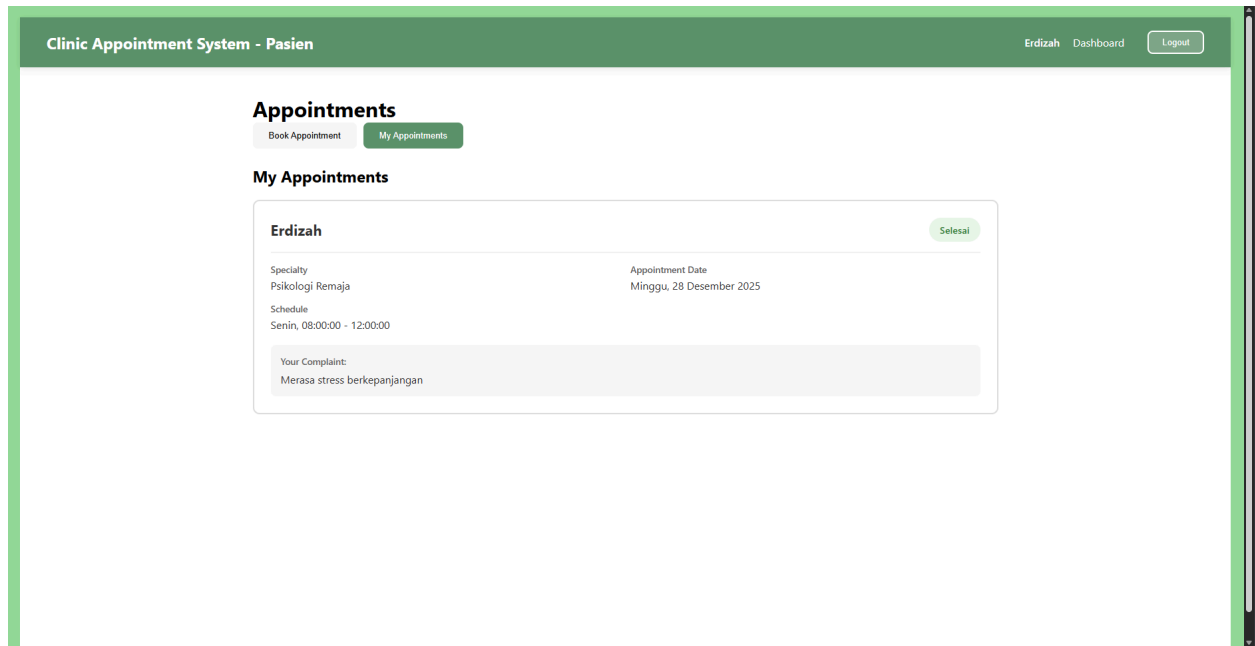
Gambar 5.1 Login Page

### 5.2 Modul Pasien



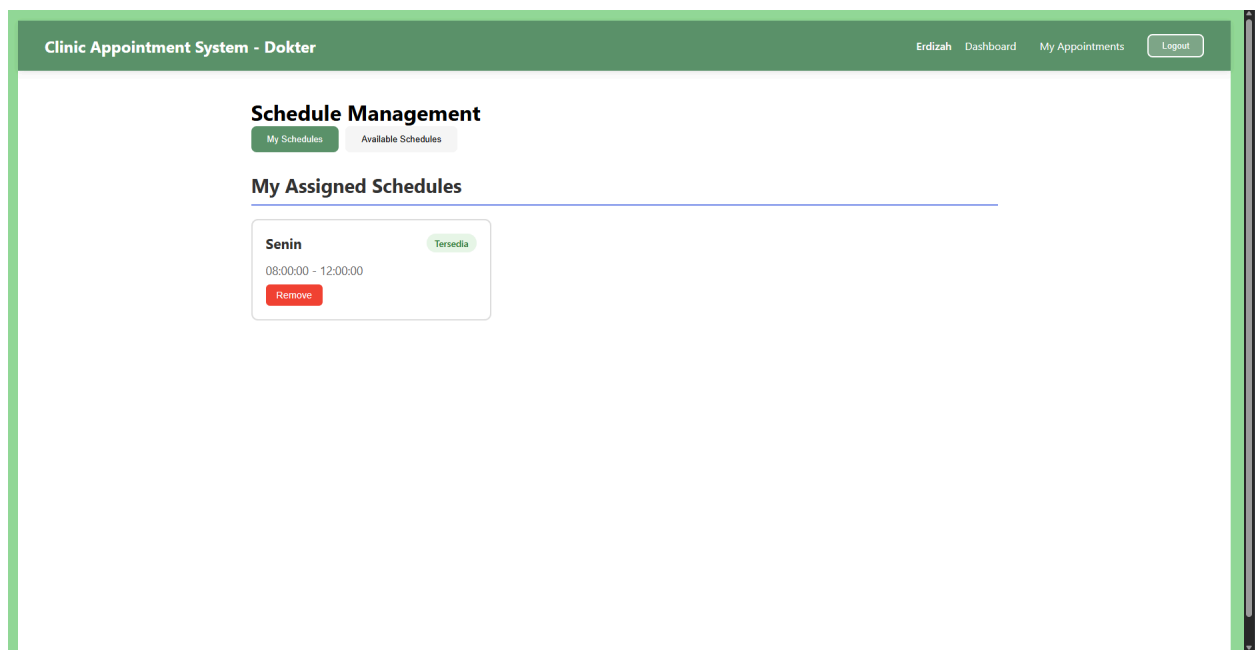
Gambar 5.2.1 Dashboard untuk Pasien



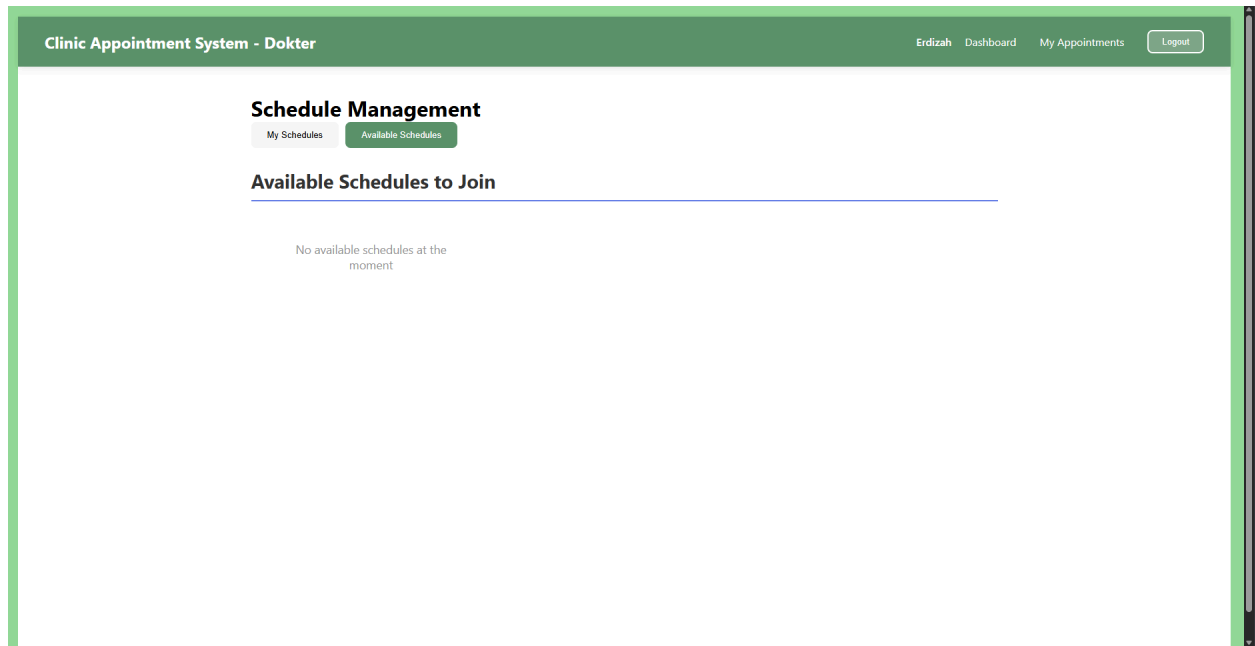


Gambar 5.2.2 Dashboard untuk Pasien

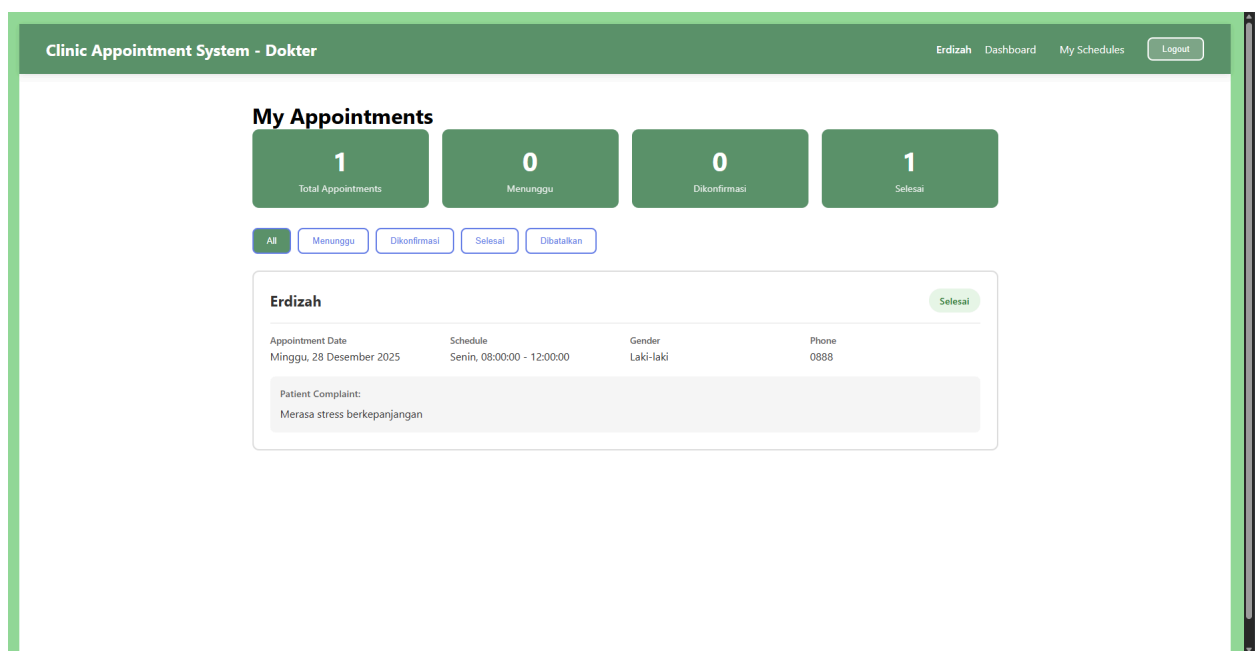
## 5.3 Modul Dokter



Gambar 5.3.1 Dashboard untuk Dokter

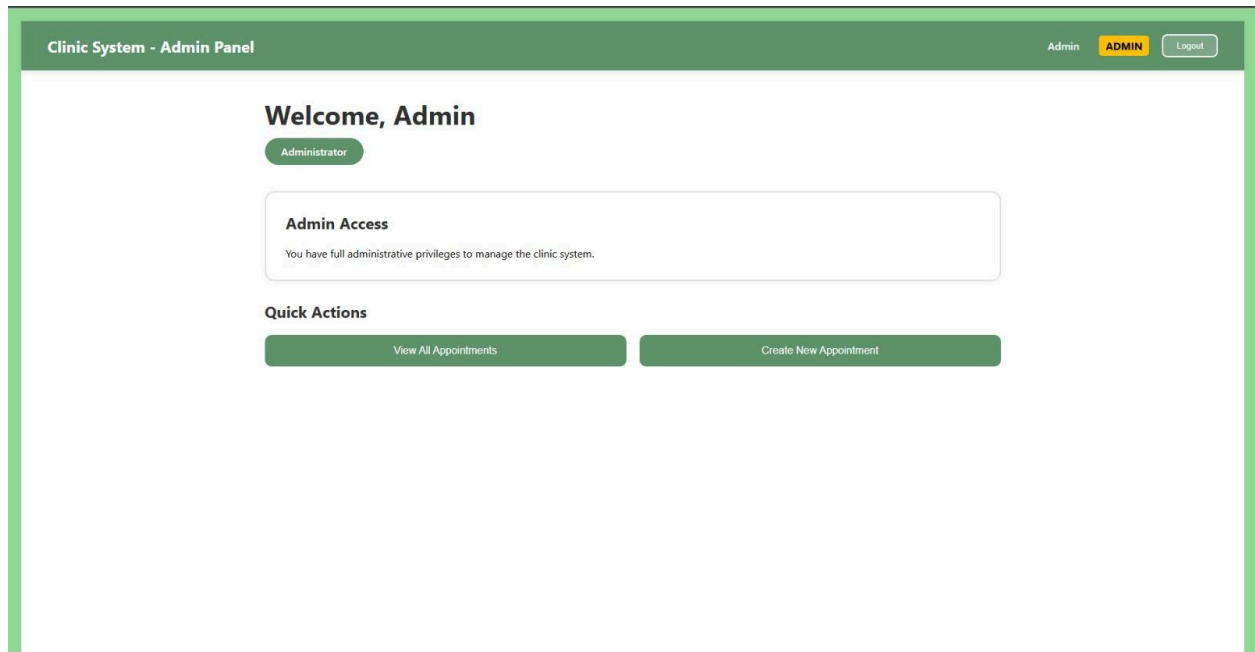


Gambar 5.3.2 Dashboard untuk Dokter

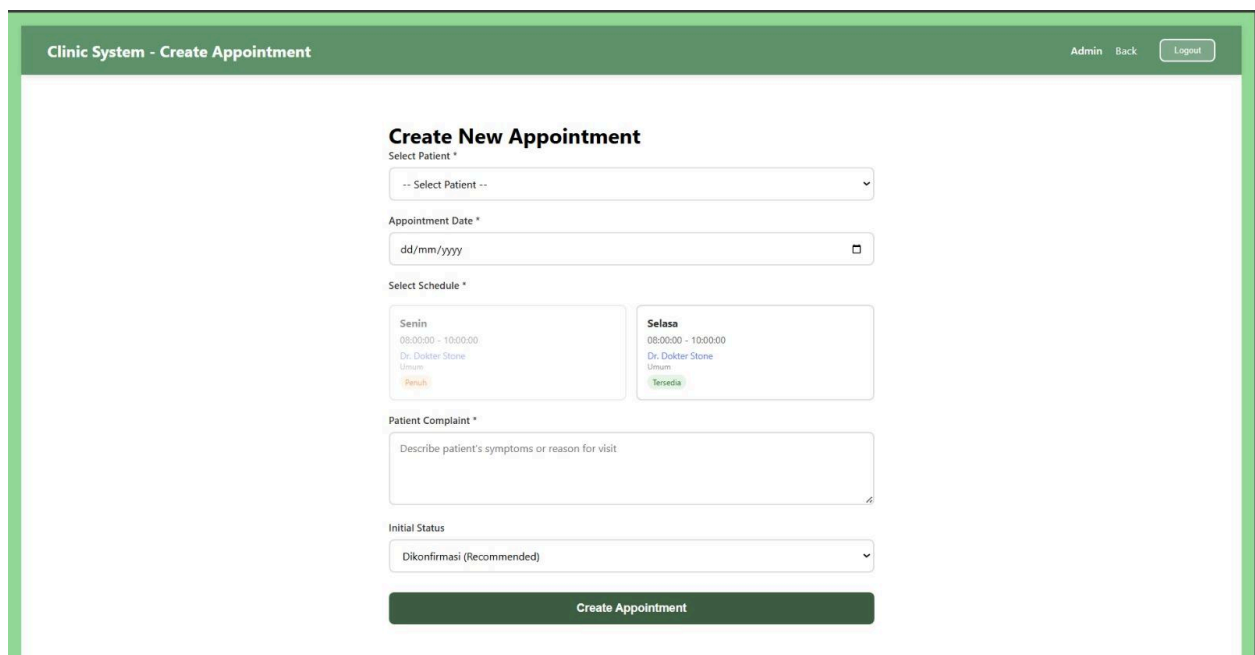


Gambar 5.3.4 Dashboard untuk Dokter

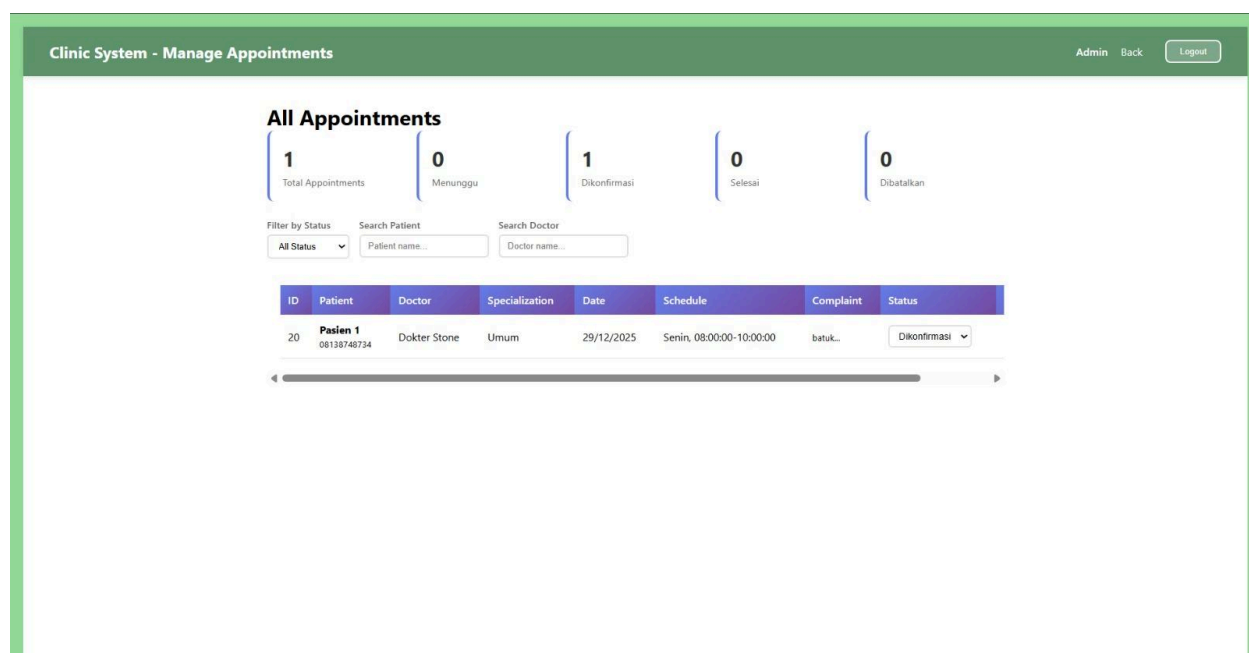
## 5.4 Modul Resepsionis (Admin)



Gambar 5.4.1 Dashboard untuk Resepsionis (Admin)



Gambar 5.4.2 Dashboard untuk Resepsionis (Admin)



Gambar 5.4.3 Dashboard untuk Resepsionis (Admin)

## **BAB VI**

### **KESIMPULAN DAN SARAN**

#### **6.1 Kesimpulan**

Berdasarkan hasil perancangan dan implementasi, dapat disimpulkan bahwa sistem Clinic Appointment Management System berhasil mengelola data klinik secara terintegrasi, meminimalkan potensi benturan jadwal dokter, serta meningkatkan efisiensi proses administrasi klinik.

#### **6.2 Saran**

Sebagai pengembangan selanjutnya, sistem dapat ditingkatkan dengan menambahkan mekanisme transaksi basis data, modul rekam medis pasien, serta peningkatan aspek keamanan sistem.

## **DAFTAR PUSTAKA**

Silberschatz, A., Korth, H. F., & Sudarshan, S. *Database System Concepts*.

Connolly, T., & Begg, C. *Database Systems: A Practical Approach to Design, Implementation, and Management*.