

**FINAL REPORT MATA KULIAH MII212501
BASIS DATA KELAS KOMA 2025:**

WEBSITE “CLINIC APPOINTMENT SYSTEM”



Nama Anggota:

- | | |
|-----------------------------|----------------------|
| 1. Aliya Khairun Nisa | (24/543832/PA/23111) |
| 2. Farsya Nabila Tori | (24/543855/PA/23113) |
| 3. Gracella Wiendy Koesnadi | (24/541739/PA/22991) |

**DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS GADJAH MADA
DESEMBER 2025**

DAFTAR ISI

DAFTAR ISI.....	1
BAB I PENDAHULUAN.....	2
I. LATAR BELAKANG.....	2
II. TUJUAN SISTEM.....	3
IV. KASUS PENGGUNAAN.....	4
BAB II DESAIN BASIS DATA.....	5
I. ENTITY RELATIONSHIP DIAGRAM (ERD).....	5
II. NORMALISASI.....	10
BAB III IMPLEMENTASI BASIS DATA.....	13
I. PRIMARY KEY.....	13
II. FOREIGN KEY.....	13
III. CONTOH PERINTAH SQL.....	15
BAB IV IMPLEMENTASI APLIKASI.....	18
1. Akses Universal (Pasien, Dokter, dan Resepsionis).....	20
2. Akses Pasien.....	21
3. Akses Dokter.....	28
4. Akses Resepsionis.....	31
BAB V PENGUJIAN DAN HASIL.....	34
1. Pengujian dan Hasil Role Pasien.....	37
2. Pengujian dan Hasil Role Dokter.....	41
3. Pengujian dan Hasil Role Resepsionis.....	53
BAB VI KESIMPULAN DAN REFLEKSI.....	59
I. KESIMPULAN.....	59
II. REFLEKSI.....	59
BAB VII APPENDIX.....	61
DAFTAR PUSTAKA.....	68

BAB I

PENDAHULUAN

I. LATAR BELAKANG

Banyak klinik menggunakan sistem manual untuk pendaftaran dan antrean yang dianggap kurang efektif dan efisien. Sistem manual ini menyebabkan pemborosan waktu dan ketidaknyamanan bagi pasien, serta kesulitan dalam pengelolaan data pasien dan jadwal pelayanan. Dampak ketiadaan sistem *appointment* (jadwal janji) digital adalah antrean panjang, waktu tunggu lama, pengelolaan kurang efisien, serta potensi penurunan kepuasan pasien dan beban kerja tenaga medis yang tidak optimal. Penelitian di Klinik Medika Saintika menunjukkan bahwa tanpa sistem digital, proses pendaftaran dan antrean kurang optimal. Pengembangan sistem *online booking* berbasis *website* berhasil meningkatkan efektivitas pelayanan, memudahkan pasien dalam pemesanan antrian, melihat jadwal, dan mengelola data pasien secara efisien.

Namun, studi lain menyebutkan bahwa walaupun sudah banyak penggunaan pendaftaran *online* (misalnya via WhatsApp, SMS), masih ada kendala teknis dan rendahnya kesadaran masyarakat untuk menggunakan sistem digital. Sebagian besar pendaftaran masih menggunakan cara manual atau mengalami kegagalan *online* karena kesalahan data atau teknis. Hal ini menunjukkan bahwa penggunaan teknologi digital dalam pelayanan klinik masih perlu ditingkatkan dan dioptimalkan agar dapat memberikan hasil yang maksimal.

Seringkali, klinik yang masih menggunakan sistem tradisional menghadapi kesulitan dalam mengatur *appointment* pasien dengan dokter, yang mengakibatkan penumpukan pasien. Selain itu, pencatatan data pasien dan riwayat kunjungan secara manual memperlambat pekerjaan admin sebab mereka terkendala dalam pencarian, peninjauan kembali, atau memverifikasi riwayat pasien. Kondisi ini bertentangan dengan standar pelayanan yang diharapkan. Padahal, agar pelayanan kesehatan berkualitas, ia harus meliputi sejumlah aspek kunci menurut Peraturan Kesehatan Republik Indonesia

Nomor HK.01.07/MENKES/1128/2022. Aspek ini mencakup ketepatan waktu (reduksi waktu tunggu), adil (layanan tidak memiliki kualitas beragam), terintegrasi (perawatan berbagai layanan), dan efisien (optimalisasi sumber daya tersedia).

Oleh karena itu, pengelolaan *database* dalam sebuah klinik menjadi hal yang sangat penting. Teknologi hadir untuk memberikan kemudahan sehingga sinkronisasi dan pengolahan data menjadi lebih mudah. Dengan adanya sistem *database* yang baik, hal ini dapat meningkatkan kualitas pelayanan, efektivitas pengelolaan informasi medis, dan tingkat efisiensi operasional di klinik. Oleh karena itu, diperlukan sebuah pengaturan proses pembuatan janji temu (*Clinic Appointment System*) yang mampu mengatur proses janji temu antara pasien dan dokter secara digital, akurat, dan terintegrasi.

II. TUJUAN SISTEM

Tujuan dari sistem janji temu klinik (*Clinic Appointment System*) adalah:

- Memudahkan proses pendaftaran dan penjadwalan janji temu pasien secara online;
- Mengurangi kesalahan penjadwalan dan penumpukan pasien di klinik;
- Menyediakan sistem yang menyimpan dan mengelola data pasien serta dokter dengan aman;
- Mempermudah dokter dalam melihat daftar janji dan riwayat pasien; dan
- Meningkatkan efisiensi operasional klinik secara keseluruhan.

III. PENGGUNA SISTEM

Sistem ini dirancang untuk digunakan oleh empat pengguna utama dengan peran masing-masing, yaitu

- 1) Dokter: Tenaga medis yang menerima dan menangani *appointment* dari pasien serta memberikan catatan medis.
- 2) Pasien: Pengguna utama yang menggunakan sistem untuk membuat *appointment* dengan dokter.

- 3) Resepsonis: Petugas *front desk* yang membantu pasien yang datang langsung ke klinik tanpa registrasi *online*.

IV. KASUS PENGGUNAAN

Domain dalam sistem *database* yang dirancang mengatur *Clinic Appointment System* melibatkan proses pendaftaran pasien, penjadwalan janji temu, konfirmasi janji temu, dan penyimpanan data pemeriksaan pasien. Dalam sistem ini, proses dimulai ketika pasien melakukan pendaftaran dan mengisi data pribadi. Setelah terdaftar, pasien dapat memilih dokter dan menentukan waktu janji temu yang tersedia. Setelah pasien membuat janji, sistem akan mencatat data *appointment* serta statusnya (“dikonfirmasi”, atau “selesai”). Berikut adalah rincian kasus penggunaan (*use cases*) untuk setiap pengguna.

1) Dokter

- Melihat daftar *appointment* yang sudah dijadwalkan.
- Mengakses data pasien yang akan diperiksa.
- Menambah dan menghapus jadwal praktik.
- Melihat riwayat pasien untuk konsultasi berikutnya.

2) Pasien

- Registrasi akun dan mengisi data pribadi.
- Melihat daftar dokter dan jadwal ketersediaan mereka.
- Melakukan *booking appointment*.
- Melihat status dan riwayat *appointment*.

3) Resepsonis

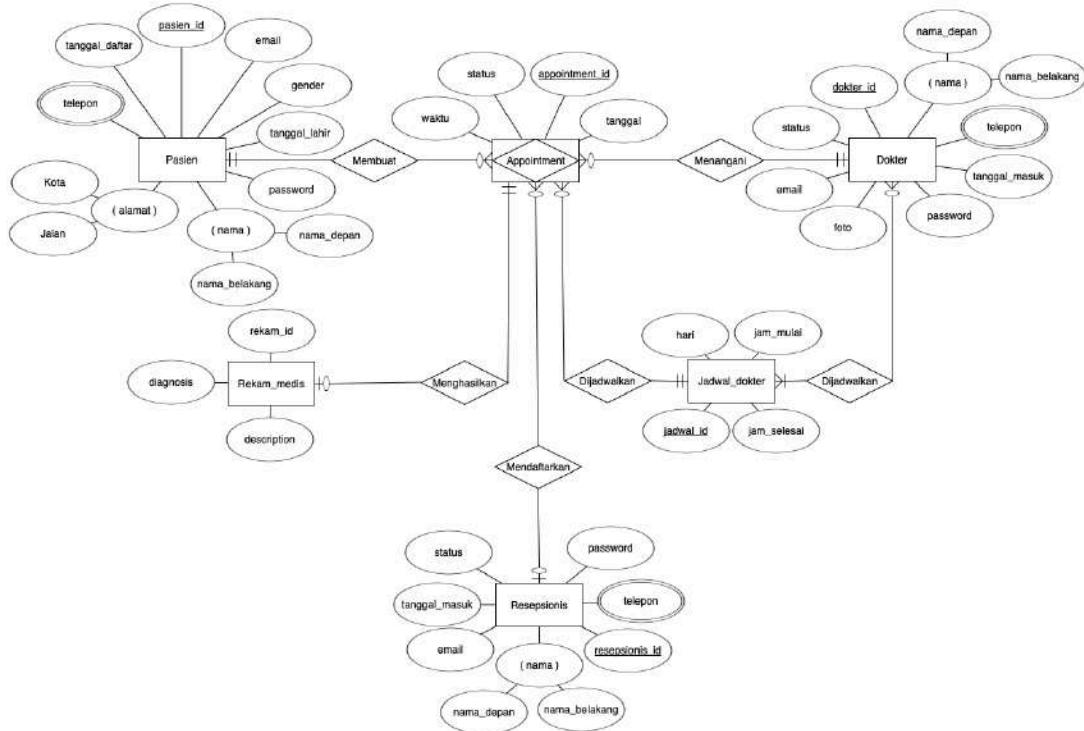
- Membantu pasien melakukan pendaftaran *appointment*.
- Membuat *appointment* atas nama pasien.

BAB II

DESAIN BASIS DATA

I. ENTITY RELATIONSHIP DIAGRAM (ERD)

Gambar 1 adalah rancangan hubungan antarentitas untuk *Clinic Appointment System* yang menjadi dasar struktur *database* agar semua proses terintegrasi.



Gambar 1. ERD *Clinic Appointment System*

Tabel 1. Daftar Entitas dan Atributnya

Entitas	Atribut
Pasien	Pasien_id, nama, alamat, telepon, email, gender, tanggal_daftar, tanggal_lahir, password
Dokter	Dokter_id, nama, email, jadwal_id, telepon, tanggal_masuk, status, password

Appointment	Appointment_id, resepsionis_id, pasien_id, dokter_id, tanggal, waktu, status
Rekam_medis	rekam_id, diagnosis, description, appointment_id
Resepsionis	Resepsionis_id, nama, telepon, email, tanggal_masuk, status, password
Jadwal_dokter	Jadwal_id, dokter_id, hari, jam_mulai, jam_selesai

PENJELASAN ENTITAS DAN ATRIBUT

1) Pasien

Entitas ini mewakili individu yang datang ke klinik untuk mendapatkan layanan kesehatan dan berfungsi untuk menyimpan identitas orang yang akan diperiksa atau diobati.

- Pasien_id digunakan sebagai *primary key* agar setiap pasien memiliki identitas unik dalam *database*. Hal ini dapat mencegah data ganda.
- Nama diperlukan untuk identifikasi dasar pasien saat pendaftaran dan pemeriksaan.
- Alamat berguna untuk administrasi dan pelacakan domisili pasien jika dibutuhkan (misal, untuk rujukan atau surat menyurat).
- Telepon diperlukan untuk komunikasi cepat, seperti konfirmasi janji temu atau perubahan jadwal.
- Email berguna sebagai alternatif lain jika pasien tidak dapat dihubungi melalui telepon.
- Gender diperlukan untuk pencatatan medis karena beberapa penyakit/obat berbeda untuk antarjenis kelamin.
- Tanggal_daftar digunakan untuk mencatat kapan pasien pertama kali terdaftar; membantu analisis data pasien baru.
- Password berguna untuk autentikasi akun pasien.

2) Dokter

Entitas ini merepresentasikan tenaga medis profesional yang bekerja di klinik. Fungsinya adalah sebagai penyedia layanan kesehatan yang akan memeriksa, mendiagnosis, dan merawat pasien.

- Dokter_id sebagai *primary key* untuk membedakan setiap dokter.
- Nama diperlukan untuk menampilkan identitas dokter di sistem janji temu dan rekam medis.
- Jadwal_id dapat menghubungkan dokter dengan jadwal praktiknya agar sistem tahu kapan dokter tersedia.
- Telepon untuk keperluan komunikasi internal (misalnya oleh admin atau resepsionis).
- Tanggal_masuk akan mencatat kapan dokter mulai bekerja, berguna untuk administrasi SDM.
- Status untuk menandai apakah dokter masih aktif praktik atau sedang cuti/pensiun.
- Email dan password berguna untuk autentikasi akun dokter.

3) Appointment (*Associative Entity*)

Entitas ini adalah pertemuan antara pasien dan dokter pada waktu yang telah ditentukan untuk pemeriksaan medis.

- Appointment_id sebagai *primary key* untuk membedakan setiap janji temu pasien dengan dokter.
- Resepsionis_id menandai siapa yang membuat *appointment* jika dilakukan secara manual (bisa *NULL* untuk pendaftaran online).
- Pasien_id menandakan siapa pasien yang membuat atau memiliki janji temu (relasi ke *Pasien*).
- Dokter_id menandakan dokter yang dituju oleh pasien (relasi ke *Dokter*).
- Tanggal & waktu diperlukan agar sistem bisa menampilkan kapan janji temu berlangsung.

- Status untuk memantau apakah janji temu masih dijadwalkan, telah selesai, atau dibatalkan.

4) Rekam_medis

Entitas ini mewakili laporan hasil pemeriksaan kesehatan dan dibuat setelah proses *appointment* selesai atau sedang berlangsung, yang berisi catatan klinis mengenai kondisi pasien dari hasil pemeriksaan dokter.

- Rekam_id sebagai *primary key* agar setiap rekam medis dapat dikenali secara unik.
- Diagnosis menyimpan hasil analisis dokter terhadap kondisi pasien.
- Description menyimpan keterangan tambahan seperti resep, tindakan, atau catatan lanjutan.
- Appointment_id sebagai *foreign key* untuk menghubungkan hasil diagnosis dengan *appointment* yang menghasilkan rekam medis tersebut.

5) Resepsonis

Entitas ini mewakili staf administrasi yang bertugas di bagian depan dan bertanggung jawab untuk mengelola pendaftaran pasien dan memfasilitasi proses administrasi awal klinik.

- Resepsionis_id sebagai *primary key* agar setiap petugas dapat dikenali secara unik.
- Nama untuk identifikasi petugas *front desk*.
- Telepon untuk komunikasi internal antarstaf.
- Jam_kerja mengatur pembagian waktu kerja dan tanggung jawab tiap resepsionis.
- Tanggal_masuk dicatat untuk kebutuhan administrasi kepegawaian.
- Status menandai apakah resepsionis masih aktif bekerja.

6) Jadwal_Dokter

Entitas ini merepresentasikan slot waktu atau sesi waktu yang tersedia di klinik yang memperjelas kapan layanan kesehatan (jadwal praktik dokter) tersedia.

- Jadwal_id sebagai *primary key* untuk setiap jadwal unik dokter.
- Dokter_id menghubungkan jadwal dengan dokter yang bersangkutan.
- Hari menunjukkan hari praktik dokter.
- Jam_mulai dan jam_selesai adalah waktu mulai dan selesai praktik.

Tabel 2. Hubungan Antarentitas

Hubungan	Entitas	Kardinalitas	Penjelasan
Membuat	Pasien - Appointment	1:N	Pasien bisa membuat banyak <i>appointment</i>
Mendaftarkan	Resepsionis - Appointment	1:N	Resepsionis mendaftarkan <i>appointment</i> pasien
Menghasilkan	Appointment - Rekam_medis	1:1	Tiap <i>appointment</i> menghasilkan satu rekam medis
Menangani	Dokter - Appointment	1:N	Dokter menangani banyak <i>appointment</i>
Dijdadwakan	Dokter - Jadwal_dokter	N:N	Dokter punya beberapa jadwal_dokter dan jadwal_dokter dapat mempunyai banyak dokter
Dijdadwakan	Jadwal_dokter - Appointment	1:N	Jadwal_dokter dapat memuat banyak <i>appointment</i>

II. NORMALISASI

- **First Normal Form (1NF)**

Relasi yang memenuhi 1NF adalah relasi yang setiap perpotongan baris dan kolomnya berisi satu dan hanya satu nilai. Maka dari itu, atribut yang *multivalued* akan didekomposisi menjadi tabel baru. Dalam *logical schema* kami, atribut yang di-set *multivalued* hanyalah atribut telepon yang ada di beberapa entitas. Sehingga atribut telepon pada tiap entitas (Pasien, Resepsonis, Dokter) dibuatkan tabel baru. Untuk atribut email dalam entitas pasien tidak *multivalued* karena kami berencana untuk menggunakan email tersebut untuk autentikasi akun pasien (pasien *login* ke akun dengan email). Jadi, email bukan atribut yang *multivalued*. Oleh karena itu, *database* kami sudah 1NF karena tidak ada atribut *multivalued* atau tidak ada grup atau kolom berulang dalam satu baris.

Selain itu, setiap tabel juga sudah memiliki *primary key* yang merupakan hal wajib pada aturan 1NF. Hal ini terutama terlihat pada relasi Rekam_medis yang termasuk *weak entity*, didefinisikan untuk memiliki *primary key* berupa gabungan antara diagnosis (*discriminator*) dan appointment_id (*primary key* dari *strong entity*).

- **Second Normal Form (2NF)**

Relasi 2NF adalah relasi yang memenuhi 1NF dan setiap atribut yang bukan *primary key* memiliki ketergantungan fungsional penuh (*full functional dependency*) pada *primary key*. Berikut adalah penjelasan 2NF pada *logical schema* kami.

Tabel 3. Second Normal Form

Tabel	Penjelasan
Pasien	Semua atribut <i>non-key</i> (nama_depan, email, dll.) bergantung penuh pada pasien_id sebagai <i>primary key</i>
Pasien_telepon	Semua atribut <i>non-key</i> bergantung penuh ke Pasien melalui <i>foreign key</i>
Resepsionis	Semua atribut <i>non-key</i> (nama_depan, status, dll.) bergantung penuh pada resepsionis_id sebagai <i>primary key</i>
Resepsionis_telepon	Semua atribut <i>non-key</i> bergantung penuh ke resepsionis melalui <i>foreign key</i>
Dokter	Semua atribut <i>non-key</i> (nama_depan, tanggal_masuk, dll.) bergantung penuh pada dokter_id sebagai <i>primary key</i>
Dokter_telepon	Semua atribut bergantung penuh ke Dokter melalui <i>foreign key</i>
Jadwal_dokter	Semua atribut <i>non-key</i> (hari, jam_mulai, jam_selesai) bergantung penuh pada jadwal_id sebagai <i>primary key</i>
Dijadwalkan	Dokter_id dan jadwal_id berperan sebagai <i>composite primary key</i> , tetapi tidak ada atribut <i>non-key</i> tambahan (hanya <i>foreign key pair</i>)
Appointment	Semua atribut <i>non-key</i> (waktu, status, tanggal) bergantung penuh pada appointment_id sebagai <i>primary key</i>
Rekam_medis	Diagnosis dan description bergantung penuh pada appointment_id sebagai <i>primary key</i>

Database kami sudah 2NF karena setiap atribut yang bukan *primary key* pada tabel mempunyai ketergantungan penuh pada *primary key* (*no partial dependencies*).

- **Third Normal Form (3NF)**

Relasi 3NF adalah relasi yang memenuhi 1NF, 2NF, dan atribut yang bukan *primary key* tidak memiliki ketergantungan transitif pada *primary key*. Dalam *logical schema* kami, atribut yang kemungkinan *transitive dependency* hanyalah diagnosis dan description pada tabel Rekam_medis (*appointment_id* → diagnosis → description). Setelah ditelusuri lebih lanjut, kedua atribut ini tidak *transitive dependency* karena:

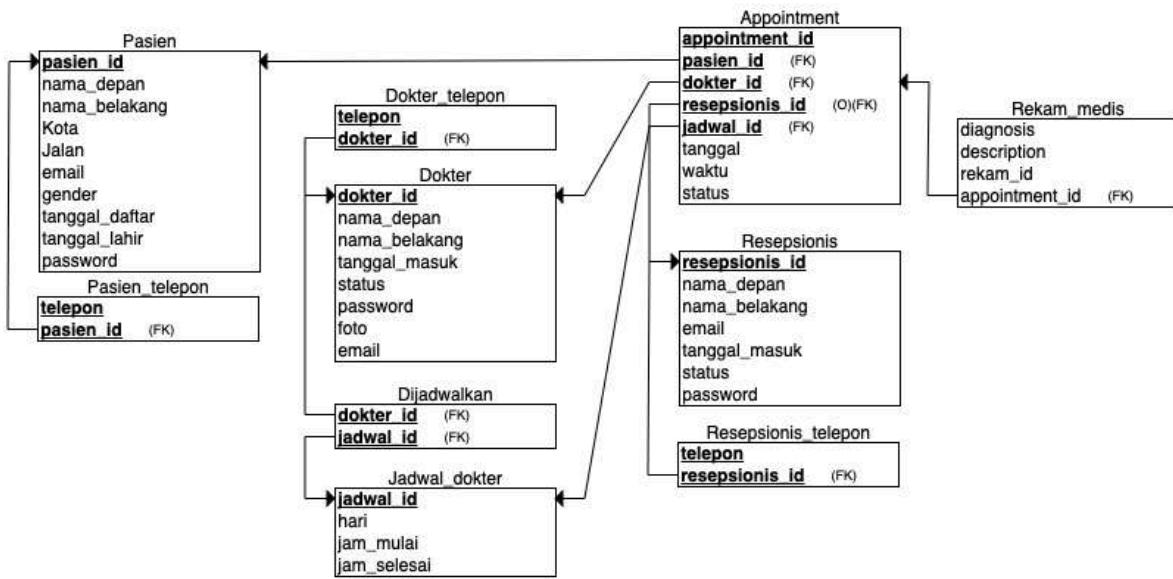
Tabel 4. Contoh Kasus

Diagnosis	Description
Flu	Batuk kering
Flu	Sakit tenggorokan
Flu	Pilek

Dapat dilihat bahwa nilai diagnosis sama, tetapi nilai *description*-nya dapat berbeda, ini membuktikan bahwa nilai *description* tidak bergantung pada diagnosis sehingga bukan *transitive dependency*. Dengan ini, *database* kami sudah 3NF karena nilai-nilainya atomik, tidak ada *partial dependencies*, dan tidak ada *transitive dependencies*.

BAB III

IMPLEMENTASI BASIS DATA



Gambar 2. Skema Relasional *Clinic Appointment System*

I. PRIMARY KEY

Primary key digunakan untuk memastikan setiap baris data dalam tabel bersifat unik dan dapat diidentifikasi.

- dokter_id: Identitas unik untuk setiap dokter dalam tabel Dokter.
- pasien_id: Identitas unik untuk setiap pasien dalam tabel Pasien.
- resepsionis_id: Identitas unik untuk setiap resepsionis dalam tabel Resepsionis.
- jadwal_id: Identitas unik untuk setiap slot jadwal dalam tabel Jadwal_dokter.
- appointment_id: Identitas unik untuk setiap transaksi janji temu dalam tabel Appointment.

II. FOREIGN KEY

Foreign key menghubungkan satu tabel dengan tabel lainnya dan menciptakan relasi antarentitas. Dalam sistem ini, penerapannya memastikan sinkronisasi data secara otomatis, seperti pada tabel *Appointment* yang akan langsung menghapus jadwal temu

jika data dokter dihapus (CASCADE) agar tidak ada antrean tanpa dokter. Sebaliknya, relasi ke resepsionis menerapkan aturan khusus (SET NULL) untuk mempertahankan riwayat transaksi medis pasien, yakni data janji temu tetap tersimpan utuh meskipun akun resepsionis yang memprosesnya telah dihapus dari sistem.

Tabel 5. Key Constraints

Tabel	Constraint	Penjelasan
Appointment	FOREIGN KEY (dokter_id) REFERENCES Dokter (dokter_id) ON DELETE CASCADE.	Menghubungkan <i>appointment</i> dengan dokter yang bertugas. Menggunakan ON DELETE CASCADE agar jika data seorang dokter dihapus dari sistem, maka semua jadwal <i>appointment</i> yang terkait dengan dokter tersebut akan otomatis terhapus.
	FOREIGN KEY (jadwal_id) REFERENCES Jadwal_dokter (jadwal_id) ON DELETE CASCADE.	Relasi ini menghubungkan <i>appointment</i> dengan jadwal dokter. Jika jadwal dokter dihapus, <i>appointment</i> yang ada pada slot tersebut juga akan terhapus.
	FOREIGN KEY (resepsionis_id) REFERENCES Resepsionis (resepsionis_id) ON DELETE SET NULL.	Relasi ini mencatat resepsionis mana yang memproses janji temu tersebut. Menggunakan ON DELETE SET NULL agar jika data resepsionis dihapus (misal <i>resign</i>), data <i>appointment</i> tidak hilang, melainkan kolom resepsionis_id akan menjadi NULL. Ini menjaga riwayat medis pasien tetap ada meskipun staf adminnya sudah tidak bekerja.
Dijadwalkan	dijadwalkan_ibfk_1 & dijadwalkan_ibfk_2	<i>Constraint</i> ini mereferensikan Dokter(dokter_id) dan Jadwal_dokter(jadwal_id). Keduanya menggunakan ON DELETE CASCADE. Jika dokter resign atau slot waktu dihapus, maka

		data penugasan di tabel ini akan otomatis terhapus.
Rekam_medis	FOREIGN KEY (appointment_id) REFERENCES Appointment (appointment_id) ON DELETE CASCADE.	Setiap rekam medis pasti berasal dari satu sesi janji temu. ON DELETE CASCADE menandakan bahwa rekam medis adalah entitas yang bergantung penuh pada <i>appointment</i> . Jika data <i>appointment</i> dihapus, catatan medisnya juga ikut terhapus.
X_Telepon	Masing-masing menjadi FOREIGN KEY ke tabel induknya (Dokter, Pasien, Resepsionis).	Untuk Dokter dan Resepsionis, menggunakan ON DELETE CASCADE. Jika dokternya dihapus, nomor teleponnya dihapus. Untuk pasien (pasien_telepon_ibfk_1), <i>constraint</i> hanya mendefinisikan relasi tanpa CASCADE eksplisit di baris tersebut, tetapi secara logika relasional, ini menghubungkan nomor telefon ke pemiliknya.

III. CONTOH PERINTAH SQL

- **Melihat Data *Appointment***

Menggabungkan tabel *Appointment*, Pasien, dan Dokter untuk menampilkan tanggal, waktu, status, nama lengkap pasien, dan nama lengkap dokter untuk *appointment* pada tanggal tertentu.

SQL

```

SELECT
    a.tanggal,
    a.waktu,
    a.status,
    CONCAT(p.nama_depan, ' ', p.nama_belakang) AS Nama_Pasien,
    CONCAT(d.nama_depan, ' ', d.nama_belakang) AS Nama_Dokter
FROM Appointment a
JOIN Pasien p ON a.pasien_id = p.pasien_id
JOIN Dokter d ON a.dokter_id = d.dokter_id

```

```
WHERE a.tanggal = '2025-11-23' ;
```

- **Melihat Jadwal Dokter**

Query ini digunakan untuk mengetahui hari dan jam berapa saja seorang dokter memiliki jadwal praktik di klinik dengan cara menghubungkan tabel Dokter dan Jadwal_dokter melalui tabel perantara Dijadwalkan. Ini diperlukan karena hubungan Dokter dan Jadwal adalah *Many-to-Many*.

SQL

```
SELECT  
    Dokter.nama_depan,  
    Dokter.nama_belakang,  
    Jadwal_dokter.hari,  
    Jadwal_dokter.jam_mulai,  
    Jadwal_dokter.jam_selesai  
FROM Dokter  
JOIN Dijadwalkan ON Dokter.dokter_id = Dijadwalkan.dokter_id  
JOIN Jadwal_dokter ON Dijadwalkan.jadwal_id = Jadwal_dokter.jadwal_id  
ORDER BY Jadwal_dokter.hari;
```

- **Melihat Riwayat Diagnosis Pasien**

Query ini menghubungkan tabel Rekam_medis ke Appointment, lalu ke Pasien untuk menampilkan riwayat rekam medis (diagnosis dan deskripsi) milik pasien tertentu berdasarkan data appointment yang pernah dilakukan.

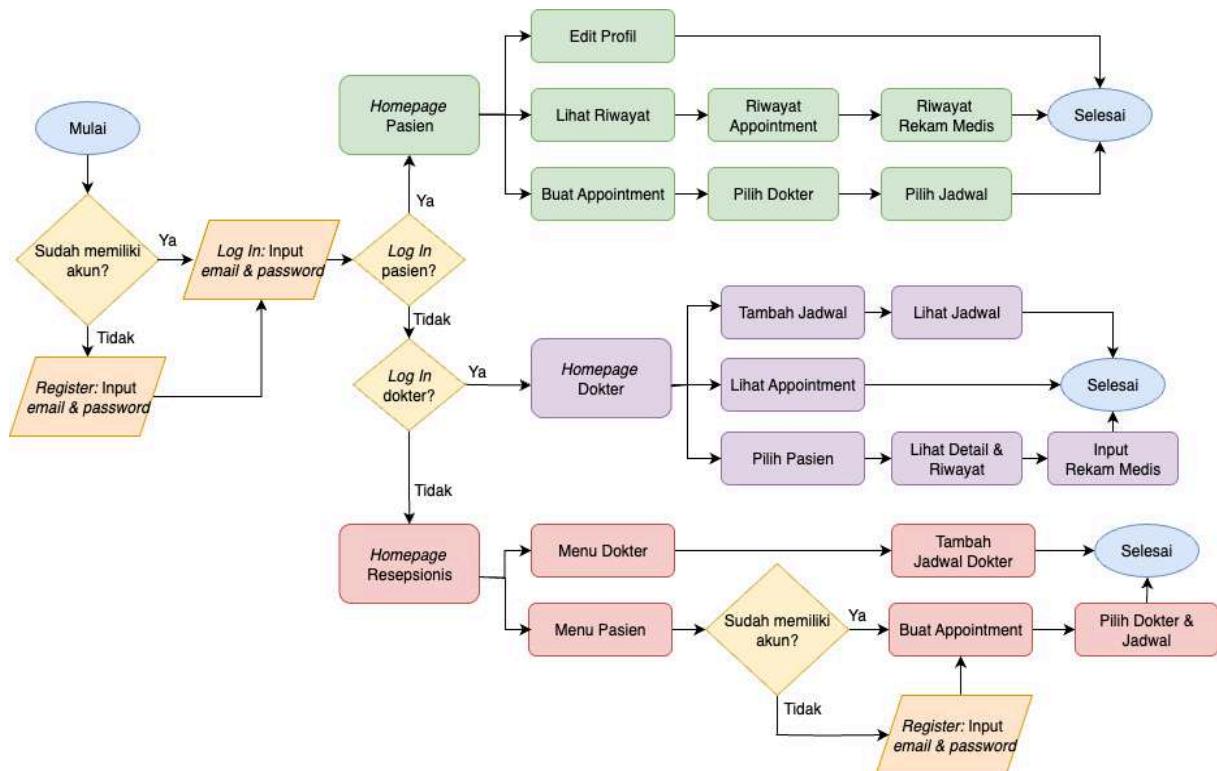
SQL

```
SELECT  
    Pasien.nama_depan,  
    Appointment.tanggal,  
    Rekam_medis.diagnosis,  
    Rekam_medis.description
```

```
FROM Rekam_medis
JOIN Appointment ON Rekam_medis.appointment_id = Appointment.appointment_id
JOIN Pasien ON Appointment.pasien_id = Pasien.pasien_id
WHERE Pasien.email = 'tori@gmail.com';
```

BAB IV

IMPLEMENTASI APLIKASI



Gambar 3. Diagram Alir (Flowchart) Penggunaan Website

Gambar di atas mengilustrasikan alur logika penggunaan laman atau *website Clinic Appointment System*. Alur ini menggambarkan langkah-langkah yang dilalui pengguna mulai dari mengakses situs web hingga menyelesaikan tujuan mereka berdasarkan peran (*role*) masing-masing. Proses tersebut dapat dijabarkan sebagai berikut.

1. **Autentikasi Pengguna:** Sistem dimulai dengan pengecekan kepemilikan akun. Pengguna yang belum terdaftar diarahkan ke halaman *Register* untuk mengisi data diri (*email* dan *password*). Pengguna yang sudah memiliki akun dapat langsung

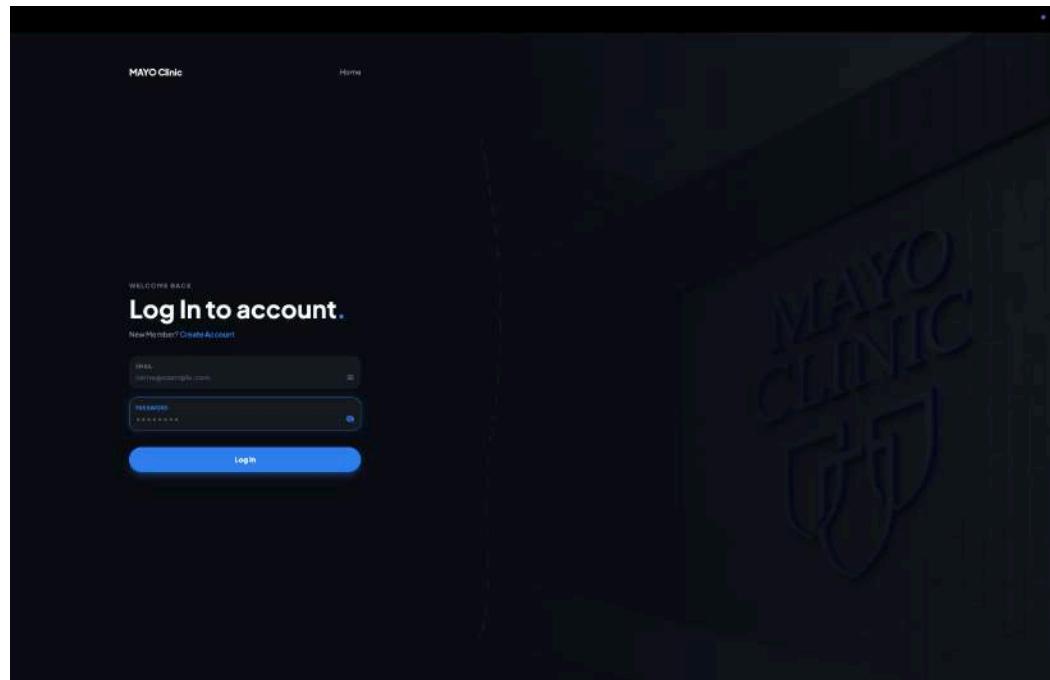
melakukan *Log In*. Sistem kemudian akan memvalidasi *role* pengguna tersebut apakah sebagai Pasien, Dokter, atau Resepsionis.

2. **Akses Pasien:** Apabila pengguna masuk sebagai Pasien, ia akan diarahkan ke *Homepage Pasien*. Fitur utama yang dapat diakses meliputi:
 - **Manajemen Profil:** Mengubah data diri (*Edit Profil*).
 - **Riwayat:** Melihat riwayat janji temu dan rekam medis sebelumnya.
 - **Pembuatan Janji Temu (Appointment):** Pasien dapat memilih dokter yang tersedia, menentukan jadwal, dan menyimpan janji temu.
3. **Akses Dokter:** Apabila pengguna masuk sebagai Dokter, sistem mengarahkan ke *Homepage Dokter*. Aktivitas utama dokter meliputi:
 - **Manajemen Jadwal:** Menambah jadwal praktik baru atau melihat jadwal yang ada.
 - **Penanganan Pasien:** Dokter dapat melihat daftar *appointment*, memilih pasien tertentu, melihat detail riwayat, dan melakukan *Input Rekam Medis* setelah pemeriksaan selesai.
4. **Akses Resepsionis:** Resepsionis memiliki hak akses administratif untuk membantu operasional klinik. Melalui *Homepage Resepsionis*, pengguna dapat mengakses menu manajemen Dokter (*Tambah Jadwal Dokter*) dan Pasien. Resepsionis juga dapat membantu pasien yang datang langsung (*walk-in*) untuk mendaftar akun atau membuatkan janji temu (*booking*) secara manual jika pasien tersebut belum memiliki akun.

Berikut terlampir implementasi antarmuka (*User Interface*) dari modul-modul yang telah dijelaskan dalam diagram alir di atas.

1. Akses Universal (Pasien, Dokter, dan Resepsonis)

1.1. Login Page

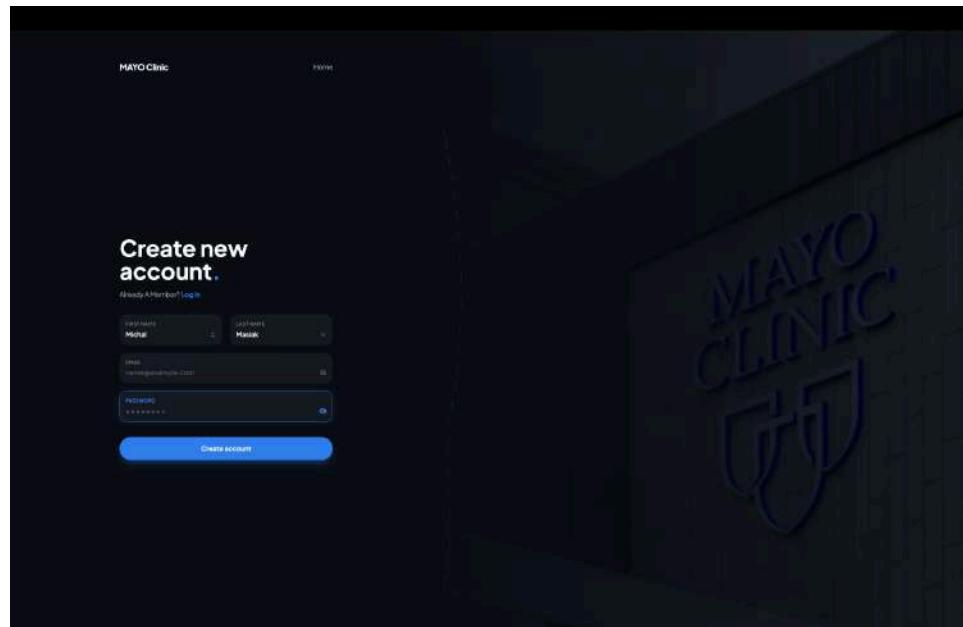


Gambar 4. Login Page

Sistem menerapkan mekanisme *Role-Based Access Control*. Saat pengguna melakukan *Log In*, sistem akan memindai kredensial (email dan *password* yang dilengkapi fitur *toggle visibility*) ke *database*. Apabila teridentifikasi sebagai Resepsonis, sistem otomatis mengarahkan ke *Dashboard Resepsonis*, begitu pula untuk Dokter dan Pasien.

2. Akses Pasien

2.1. Register Page



Gambar 5. Register Page

Pada modul registrasi, keamanan data diprioritaskan dengan menerapkan fungsi *hashing* menggunakan *library* Werkzeug Security. Hal ini memastikan kata sandi pengguna tersimpan dalam format terenkripsi di *database*.

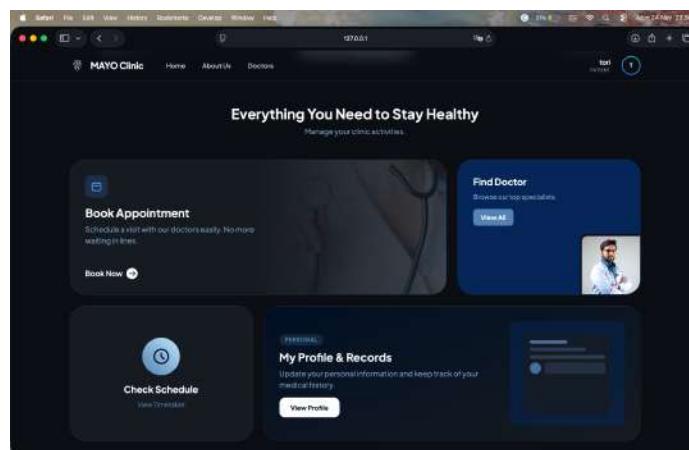
2.2. Homepage Pasien

Berikut adalah *homepage* yang mempunyai 5 *container*, *container* pertama sebagai pembuka, *container* kedua berisi *utilities*/fitur yang ada pada *website*, *container* ketiga adalah penjelasan singkat mengenai klinik. Selanjutnya, *container* keempat berisi daftar dokter beserta fotonya dan *container* terakhir berisi informasi tentang kontak klinik.



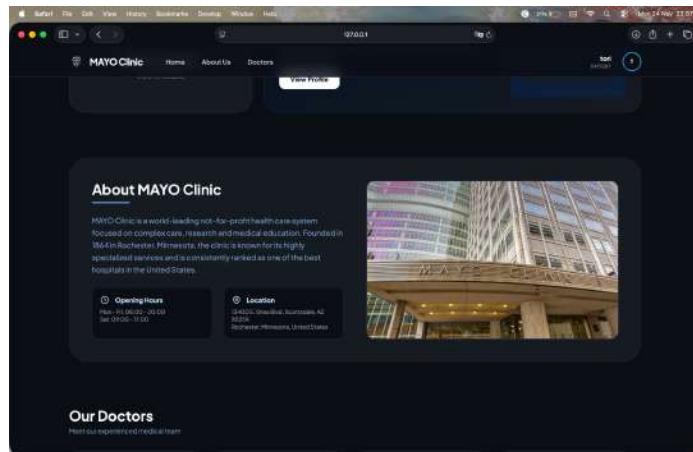
Gambar 6. Homepage Pasien Container 1

Container kedua berisi 4 *subcontainer*. *Subcontainer* pertama berisi tombol untuk membuat janji temu yang jika diklik akan dialihkan ke *bookAppointment.html*. *Subcontainer* kedua berisi tombol untuk melihat semua daftar dokter yang jika diklik akan dialihkan ke ke *container* keempat. Lalu *subcontainer* ketiga mempunyai tombol untuk melihat jadwal dokter sedangkan *subcontainer* terakhir memiliki tombol *view profile* yang jika diklik, pasien akan dialihkan ke *page profile.html*



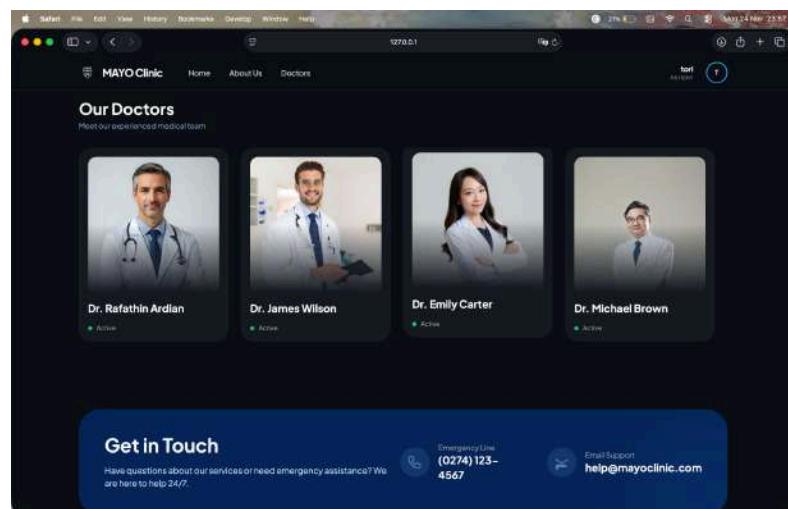
Gambar 7. Homepage Pasien Container 2

Berikut adalah *container* ketiga dari home.html yang berisi informasi singkat mengenai klinik. Informasi tersebut mencakup sejarah singkat, jam operasional, alamat, dan foto gedung.



Gambar 8. *Homepage Pasien Container 3*

Container keempat berisi list dokter yang bertugas di klinik beserta fotonya. *List* ini bisa di-swipe ke kanan agar dapat melihat semua dokter. *Container* terakhir yaitu *container* 5, berisi kontak *emergency* klinik yang dapat dihubungi serta *email* klinik jika pasien ingin memberi *feedback*.



Gambar 9. *Homepage Pasien Container 4 dan 5*

2.3. Profile

Halaman ini menyajikan data pribadi pasien. Terdapat *profile picture* dengan inisial yang dapat menyesuaikan dengan inisial 2 kata pertama dari nama pengguna. Dalam *page* ini, terdapat 2 tombol, yaitu tombol *appointment history* yang jika diklik akan dialihkan ke appointmentHistory.html. Tombol kedua yaitu *edit profile* yang dapat mengalihkan ke editProfile.html.



Gambar 10. Profile Pasien

2.4. Edit Profile

Halaman ini berbentuk form yang mempunyai 7 field yaitu first name, last name, gender, date of birth, city, street address, dan phone numbers. Field phone numbers dapat mempunyai banyak nilai. Selain itu, terdapat 2 tombol yaitu cancel yang berguna untuk kembali ke halaman profile, dan tombol save changes yang akan commit perubahan tersebut ke database yang ada di railway.

The screenshot shows the 'Edit Profile' page from the Mayo Clinic website. At the top, there's a navigation bar with links for 'Home', 'About Us', and 'Doctors'. On the right side of the header, there's a 'PATIENT' section with a user icon and the number '1'. The main title 'Edit Profile' is centered above a sub-instruction 'Update your personal information and manage your account.' Below this, there are several input fields: 'FIRSTNAME' (John), 'LASTNAME' (Doe), 'GENDER' (Select Gender), 'DATE OF BIRTH' (25/11/2025), 'CITY' (New York), 'STREET ADDRESS' (123 Main St), and a 'PHONE NUMBERS' section containing the number 08123456789. A button at the bottom right says 'Save Changes'.

(a)

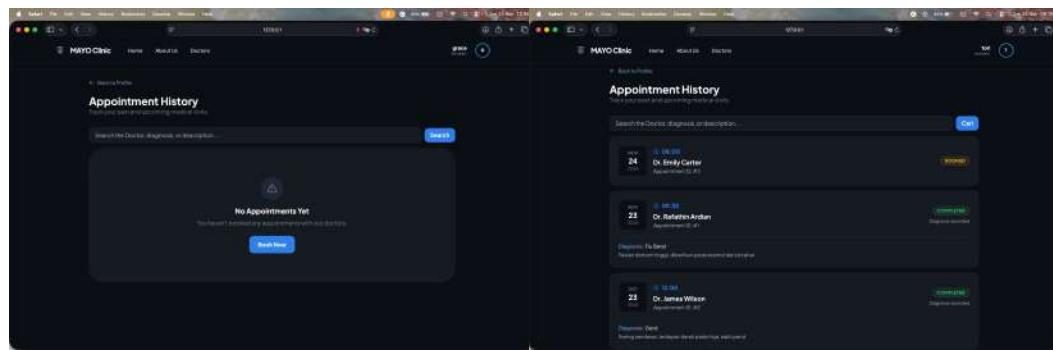
This screenshot shows the 'PHONE NUMBERS' section of the profile edit page. It lists four existing phone numbers: 848234738473, 4238974829347, 329847384734, and 08123456789. Each number is preceded by a small phone icon and followed by a red trash can icon for deletion. Below the list is a button labeled '+ Add Another Number'.

(b)

Gambar 11. Edit Profile Pasien (a) Data Diri (b) Nomor Telepon

2.5. Appointment History

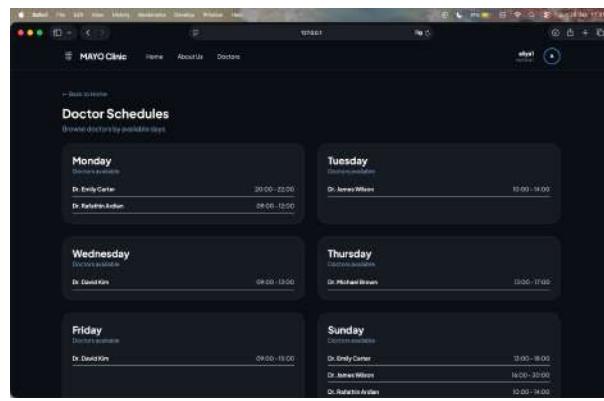
Halaman ini mempunyai fitur *search* yang dapat menghilangkan *card appointment history* jika tidak cocok dengan apa yang dicari. *Card* berisi informasi tanggal *appointment*, status, beserta rekam medis yang ada di bawahnya.



Gambar 12. Appointment History

2.6. Jadwal Dokter

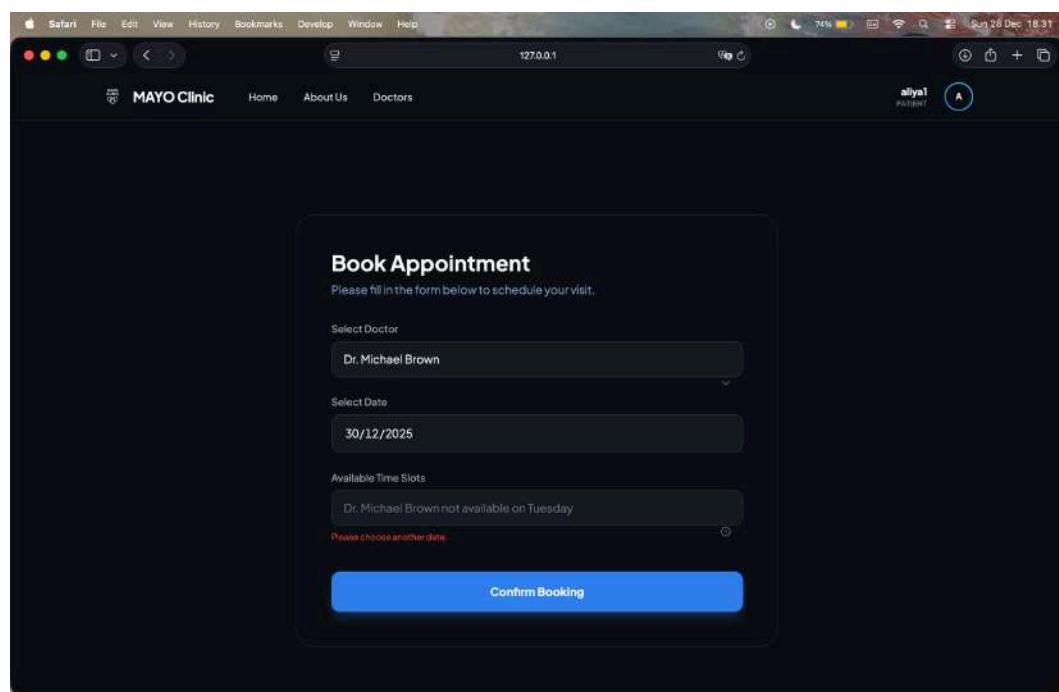
Halaman ini berfungsi sebagai papan informasi bagi pasien untuk melihat ketersediaan waktu praktik dokter spesialis di klinik sebelum mereka melakukan pemesanan (*booking*) janji temu.



Gambar 13. Jadwal Dokter

2.7. Booking Page

Halaman ini dapat diakses dari memencet tombol “book now” pada *container* kedua di *homepage*. Halaman ini berupa *form dropdown* yang dapat mempermudah pasien memilih dokter, jadwal, serta tanggalnya. Pasien diharapkan memilih dokter terlebih dahulu dan *dropdown schedule* otomatis akan menampilkan jadwal dari dokter yang dipilih. Pada dropdown tanggal, tanggal yang sudah lewat akan berwarna abu dan tidak bisa dipilih dan jika pasien memilih tanggal yang harinya bukan hari dokter tersebut praktek maka akan muncul pesan berwarna merah di bawah.



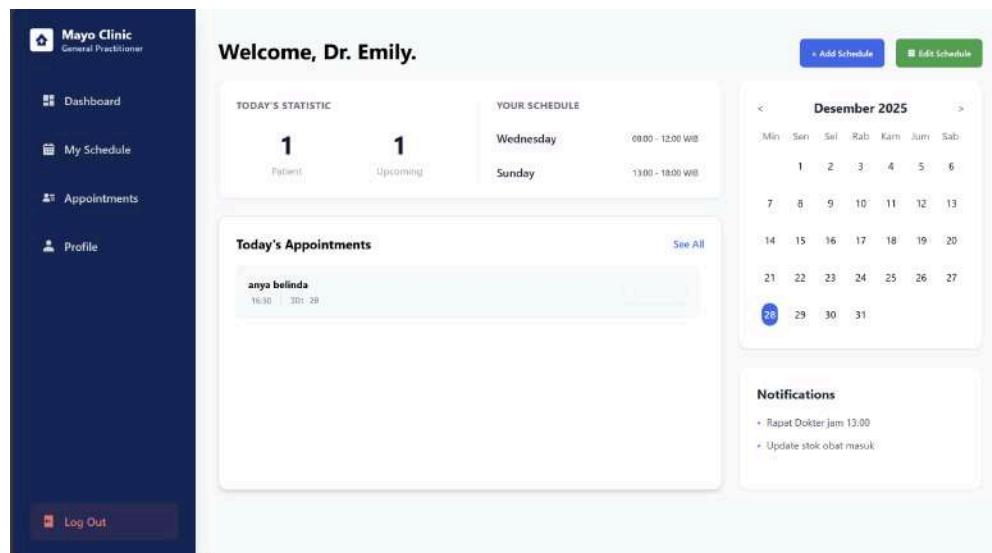
Gambar 14. *Booking Page*

3. Akses Dokter

3.1. Homepage Dokter

Homepage dokter terdiri atas beberapa fitur, antara lain *Dashboard* yang berisi ringkasan statistik, *My Schedules* untuk manajemen jadwal, *Appointments* untuk manajemen janji temu, dan *Profile* untuk mengubah data pribadi.

Pada tampilan *dashboard*, sistem menyajikan statistik jumlah pengunjung yang dihitung berdasarkan data janji temu. Selain itu, ditampilkan juga jadwal praktik dokter terkait serta informasi janji temu yang dialokasikan pada tanggal saat ini. Tampilan *dashboard* juga menyajikan dua tombol yang berfungsi untuk manajemen jadwal dokter.

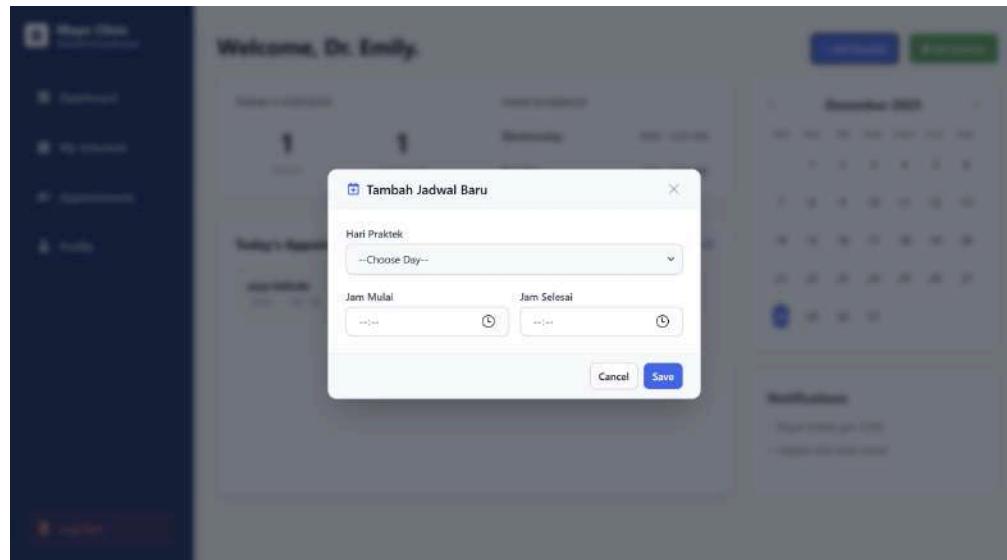


Gambar 15. Dashboard Dokter

3.2. Manajemen Jadwal Dokter

Fitur manajemen jadwal dokter mencakup dua fungsi utama, yaitu untuk menambahkan jadwal baru dan menghapus jadwal yang telah ada. Pada halaman

Dashboard, terdapat *pop up* yang berfungsi untuk menambahkan jadwal dokter (Gambar 16).



Gambar 16. Pop Up untuk Menambahkan Jadwal

Pada halaman *My Schedule*, dokter dapat menghapus jadwal yang sekiranya sudah tidak aktif.

A screenshot of the 'Manage Schedule' page. On the left, a sidebar menu includes 'Dashboard', 'My Schedule' (which is selected and highlighted in blue), 'Appointments', 'Profile', and 'Log Out'. The main content area is titled 'Manage Schedule' with the sub-instruction 'View and manage your weekly practice hours.' Below this is a table titled 'Active Schedule' with three rows. The table columns are 'DAY', 'PRACTICE TIME', and 'ACTION'. The first row shows 'Wednesday' with '08:00 - 12:00' and a 'Delete' button. The second and third rows both show 'Sunday' with '13:00 - 18:00' and a 'Delete' button. The 'ACTION' column has a small red icon next to each 'Delete' button.

Gambar 17. Halaman *My Schedule* untuk Manajemen Jadwal Dokter

3.3. Manajemen Appointment

Halaman ini berfungsi untuk melihat janji temu (*appointment*) yang sudah terjadwal, janji temu (*appointment*) yang sudah selesai sekaligus melihat catatan rekam medis pada janji temu (*appointment*) yang sudah selesai. Selain itu, terdapat fitur-fitur yang ditambahkan guna membantu dalam pengelolaan janji temu (*appointment*), yaitu *search bar* yang berfungsi untuk mencari nama pasien atau ID pasien, dan *filter status* untuk mencari janji temu (*appointment*) berdasarkan status.

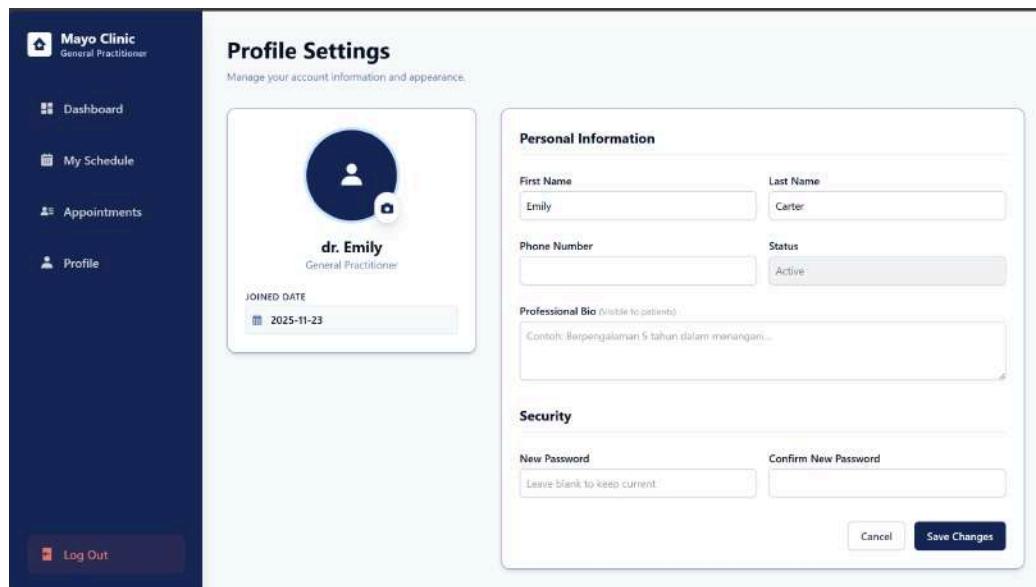
The screenshot shows the 'Patient Queue List' page. On the left is a dark sidebar with the Mayo Clinic logo, 'General Practitioner', and navigation links: Dashboard, My Schedule, Appointments (selected), Profile, and Log Out. The main area has a title 'Patient Queue List' and a subtitle 'Manage examination schedules and medical records for today.' It features three search/filter sections: 'SELECT DATE' (set to 28/12/2025), 'SEARCH PATIENT' (empty), and 'FILTER STATUS' (set to 'All Status'). Below these is a table with columns: TIME, PATIENT, STATUS, and ACTION. The data is as follows:

TIME	PATIENT	STATUS	ACTION
15:30	a anya belinda ID:11	Selesai	History
16:30	a anya belinda ID:11	Selesai	History
17:00	F Farsya Nabila Tori ID:4	On Going	Input Record
19:00	L Lulu Farasya ID:9	Terjadwal	Not Started

Gambar 18. Halaman *Appointments*

3.4. Manajemen *profile*

Halaman ini berfungsi untuk mengubah data seperti nama, nomor telepon, *password*, dan deskripsi profil.

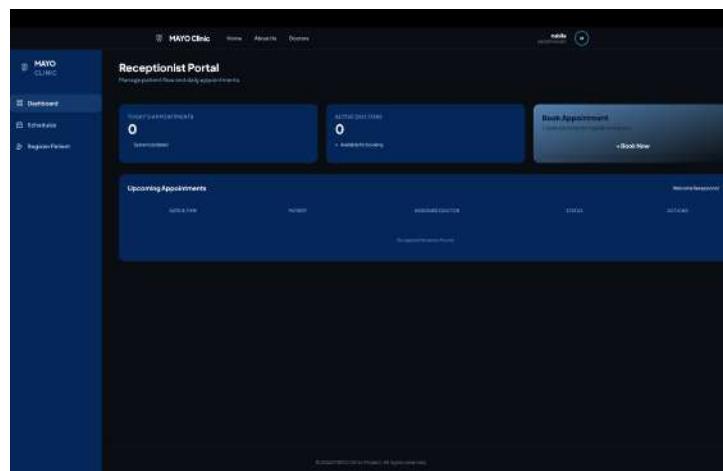


Gambar 19. Edit Profile

4. Akses Resepsonis

4.1. Homepage Resepsonis

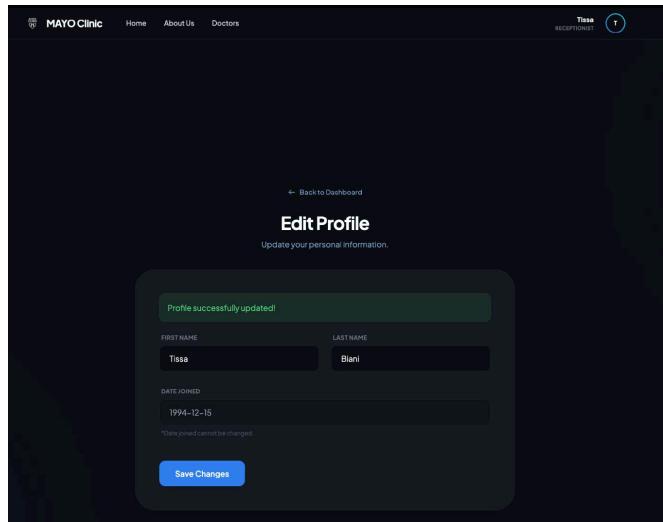
Tampilan *dashboard* resepsonis menyediakan akses menu melalui *sidebar*, yaitu *Dashboard* (beranda), *Schedules* untuk memantau jadwal praktik dokter, dan *Register Patient* untuk membantu pasien yang datang langsung (*walk-in*).



Gambar 20. Homepage Resepsonis

4.2. Manajemen Profil

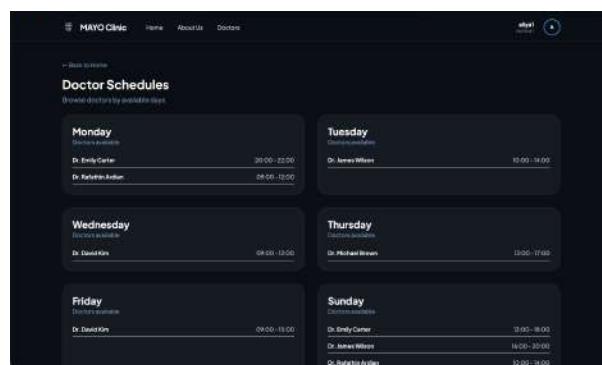
Resepsionis dapat memperbarui nama akun melalui fitur *Edit Profile*. Perubahan data akan otomatis tersinkronisasi pada tampilan utama *dashboard*.



Gambar 21. Edit Profile Resepsionis

4.3. Manajemen Jadwal Dokter

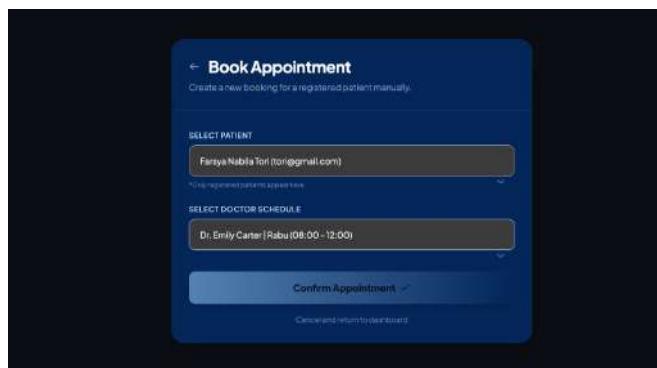
Fitur Manajemen Jadwal menyajikan informasi ketersediaan dokter yang terorganisir berdasarkan hari kerja secara visual. Tampilan ini memungkinkan resepsionis untuk mengidentifikasi nama dokter dan jam praktik yang tersedia secara efisien.



Gambar 22. Jadwal Dokter dari *Homepage* Resepsionis

4.4. Pembuatan *Appointment*

Fitur ini memfasilitasi resepsionis untuk mendaftarkan janji temu pasien secara manual (*walk-in*) dengan cepat. Melalui formulir *Book Appointment*, resepsionis cukup memilih data pasien dan jadwal dokter yang tersedia dari menu *dropdown*, lalu mengonfirmasinya untuk memperbarui sistem penjadwalan.



Gambar 23. Pembuatan *Appointment* Melalui Resepsionis

BAB V

PENGUJIAN DAN HASIL

Berdasarkan metode Black Box Testing, pengujian dilakukan dengan membuat skenario uji untuk fitur utama. Berikut adalah hasil pengujian fungsionalitas sistem yang disajikan sebagai berikut.

Tabel 5. Pengujian Fitur

Pasien				
No	Fitur	Skenario	Hasil yang Diharapkan	Status
1	Registrasi	Mendaftar akun pasien baru	Data pasien masuk ke tabel pasien dan bisa digunakan untuk login	Berhasil
2	Login	Masukkan email dan password yang benar	Berhasil masuk ke halaman home dan session nama muncul di navbar	Berhasil
3	Booking	Pasien memilih dokter, tanggal, dan jam yang tersedia	Data masuk ke tabel appointment dengan status waiting	Berhasil
4	Search	Pasien mencari riwayat appointment dengan kata kunci ‘david’	Sistem menampilkan daftar appointment yang mengandung kata ‘david’	Berhasil
Dokter				
No	Fitur	Skenario	Hasil yang Diharapkan	Status
1	Login	Dokter memasukkan email dan password yang sudah terdaftar	Dokter masuk ke halaman <i>Dashboard</i> dokter.	Berhasil
2	<i>Statistic Update</i>	Pasien membuat janji temu (<i>appointment</i>)	Data statistik pada halaman <i>Dashboard</i> berubah seiring bertambahnya janji temu	Berhasil

			(appointment).	
3	Menambahkan Jadwal	Dokter memilih hari dan jam untuk waktu praktik	Pada halaman <i>Dashboard</i> , kolom <i>Your Schedule</i> berhasil diperbarui dengan adanya data baru dan pada halaman <i>My Schedule</i> , data jadwal yang ditambahkan berhasil ditampilkan.	Berhasil
4	Menghapus Jadwal	Menekan tombol “ <i>delete</i> ”	Data jadwal yang ingin dihapus hilang dari halaman <i>Dashboard</i> dan <i>My Schedules</i>	Berhasil
5	Status Janji Temu (<i>Appointment</i>)	Menambahkan 2 pasien baru dengan jadwal yang berbeda	Status janji temu (<i>appointment</i>) berhasil menyesuaikan dengan jam janji temu (<i>appointment</i>).	Berhasil
6	Menambahkan Rekam Medis	Menekan tombol “ <i>input record</i> ” dan memasukkan data-data yang diperlukan	Data rekam medis tersimpan	Berhasil
7	History Rekam Medis	Dokter menekan tombol “ <i>history</i> ”	Menampilkan data hasil rekam medis yang diinput untuk setiap pasien yang melakukan janji temu (<i>appointment</i>)	Berhasil

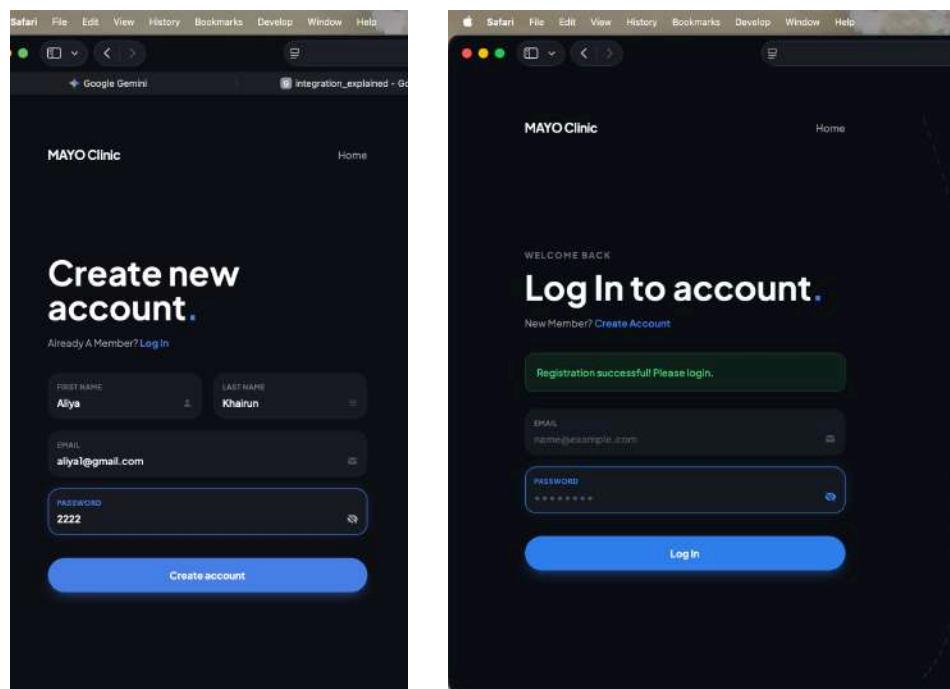
Resepsionis

No	Fitur	Skenario	Hasil yang Diharapkan	Status
1	Login	Resepsionis masuk menggunakan email dan <i>password</i> yang valid	Sistem mengarahkan ke <i>Dashboard</i> Resepsionis dan menampilkan nama pengguna pada <i>header</i>	Berhasil
2	Registrasi	Resepsionis mendaftarkan akun untuk pasien baru	Data pasien baru tersimpan di <i>database</i> dan nama pasien langsung muncul saat pencarian	Berhasil

		melalui menu <i>Register Patient</i>	untuk <i>booking appointment</i>	
3	Manajemen Profil	Resepsionis mengubah data diri (nama depan) dan menyimpan perubahan	Data tersimpan di database dan nama pada <i>header</i> langsung diperbarui	Berhasil
4	Manajemen Jadwal	Resepsionis mengakses menu <i>Schedules</i> untuk melihat ketersediaan dokter	Menampilkan visualisasi kartu jadwal praktik dokter yang terorganisir berdasarkan hari	Berhasil
5	<i>Walk-in Appointment</i>	Resepsionis mendaftarkan janji temu manual untuk pasien terdaftar	Data <i>appointment</i> berhasil disimpan ke sistem dan slot waktu dokter terisi	Berhasil

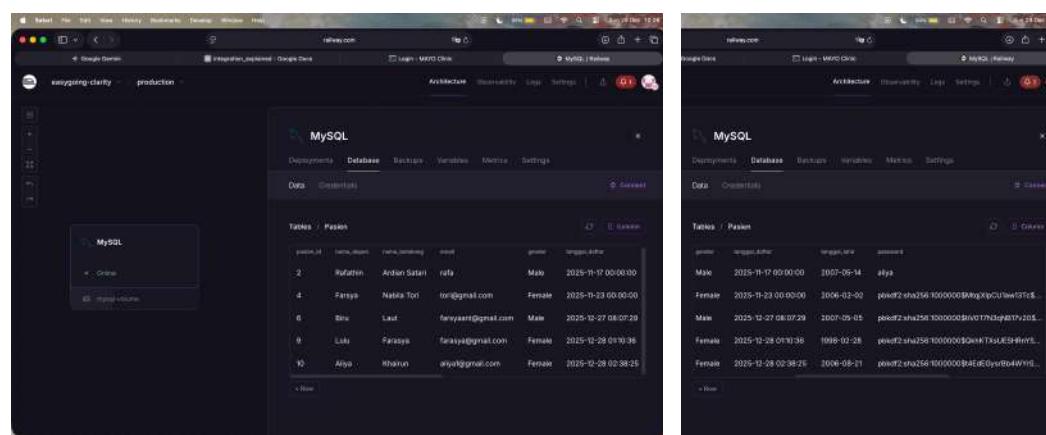
1. Pengujian dan Hasil *Role* Pasien

1.1. Register dan Login



Gambar 24. Proses *Register*

Testing dilakukan dengan menambahkan akun dengan email aliya1@gmail.com dan password 2222. Setelah tombol create account diklik, pengguna akan langsung diarahkan ke halaman login yang terdapat notif '*Registration successful! Please login.*'



The image shows two screenshots of the Railway MySQL interface. The left screenshot displays the 'Users' table with the following data:

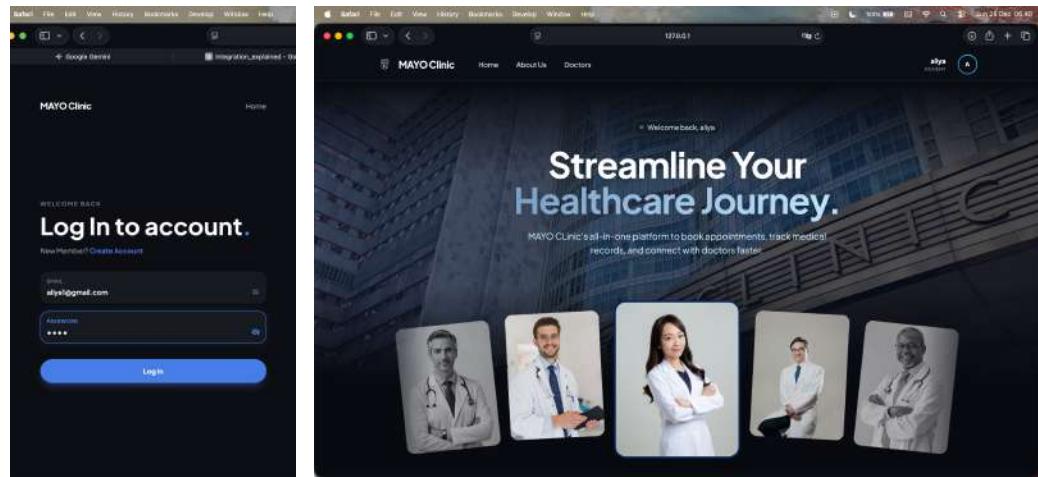
user_id	name	email	password	register_date
1	Rufatthin	Arifan Satri	rufa	Male 2025-11-17 00:00:00
4	Parasya	Natalia Tuti	tori@gmail.com	Female 2025-11-23 00:00:00
6	Bima	Lauh	baryant@gmail.com	Male 2025-12-27 08:07:29
8	Luks	Parasya	farisyah@gmail.com	Female 2025-12-28 01:10:38
10	Aliya	Kharun	aliya@gmail.com	Female 2025-12-28 02:08:25

The right screenshot displays the 'Sessions' table with the following data:

session_id	register_date	email	password
Male	2025-11-17 00:00:00	1007-06-14	aliya
Female	2025-11-23 00:00:00	1006-03-02	password2sha2561000000\$WngXgCuIew1SIC5...
Male	2025-12-27 08:07:29	1007-06-05	password2sha2561000000\$WngXgCuIew1SIC5...
Female	2025-12-28 01:10:38	1008-02-28	password2sha2561000000\$QmKHTKUfSHWhYt...
Female	2025-12-28 02:08:25	1006-03-21	password2sha2561000000\$4EAE0sys04Wrr...

Gambar 25. Hasil *Database* setelah Proses Registrasi

Pada *database* di Railway, pengguna dengan email aliya1@gmail.com berhasil ditambahkan serta passwordnya sudah dienkripsi. Hal ini membuktikan mekanisme register telah berhasil.

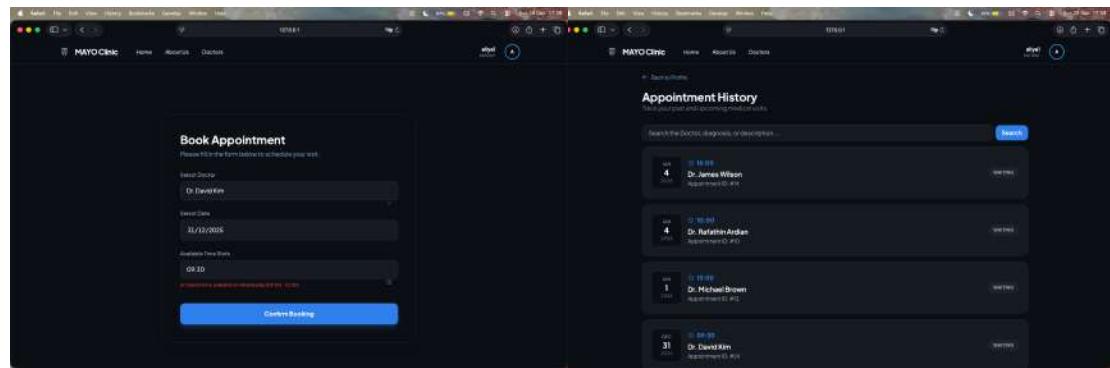


Gambar 26. Proses Login

Testing dilanjut dengan mengisi data akun baru yang sebelumnya baru didaftarkan. Setelah menekan tombol *login*, *tester* diarahkan ke home page pasien dengan session nama di bagian *navigation bar*. Ini membuktikan mekanisme *login* berhasil.

1.2. Booking

Testing dilakukan dengan menekan tombol “*book appointment*” pada *homepage* pasien. Setelah memilih dokter dan jadwal yang tersedia, *tester* diarahkan ke halaman *home page*. Untuk melihat *appointment* yang sudah dibuat, *tester* pergi ke halaman *profile* dan klik tombol *appointment history*. Di sana terdapat *appointment* yang baru saja dibuat. (Sebelumnya *tester* telah membuat beberapa *appointment* lain jadi pada halaman *appointment history* terdapat banyak *appointment*).



Gambar 27. Proses Book Appointment

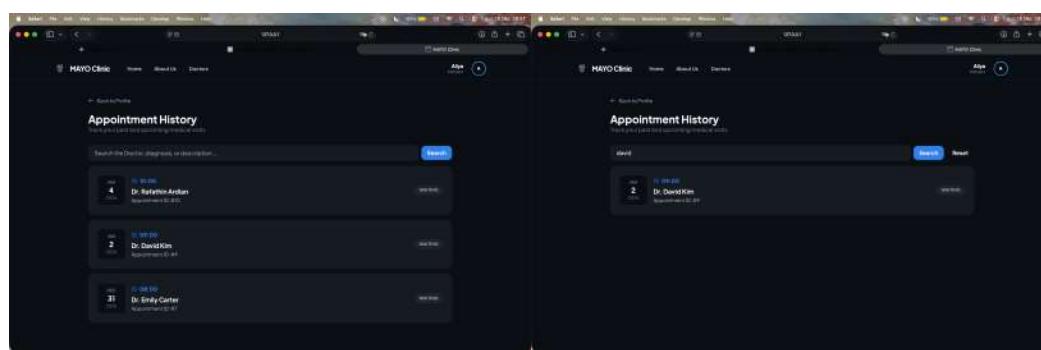
Pada database di Railway, sudah terdapat data booking appointment yang baru saja dibuat dengan dokter James Wilson (id:2) yang dibuat oleh pengguna aliya1@gmail.com (id:10). Hal ini membuktikan proses booking telah berhasil.

The image shows the Railway platform interface for a MySQL database named 'easygoing-clarity'. On the left, there's a sidebar with icons for deployments, databases, backups, variables, metrics, and settings. A central panel shows a 'MySQL' icon with a green 'Online' status and a 'mysql-volume' label. To the right, a detailed view of the 'Appointment' table is displayed. The table has columns: appointment_id, tanggal, waktu, status, pasien_id, doktor_id, recepcionis_id, and jadwal_. The data is as follows:

Gambar 28. Tabel *Appointment* pada Railway

1.3. Search

Testing ini dilakukan dengan menambah beberapa *appointment* dari dokter yang berbeda kemudian mengetik nama ‘david’ pada kolom pencarian. Setelah tombol *search* dilakukan, *website loading* sesaat dan hanya menampilkan container riwayat *appointment* dari dr. David. Hal ini membuktikan fitur search berhasil dan berfungsi dengan baik.

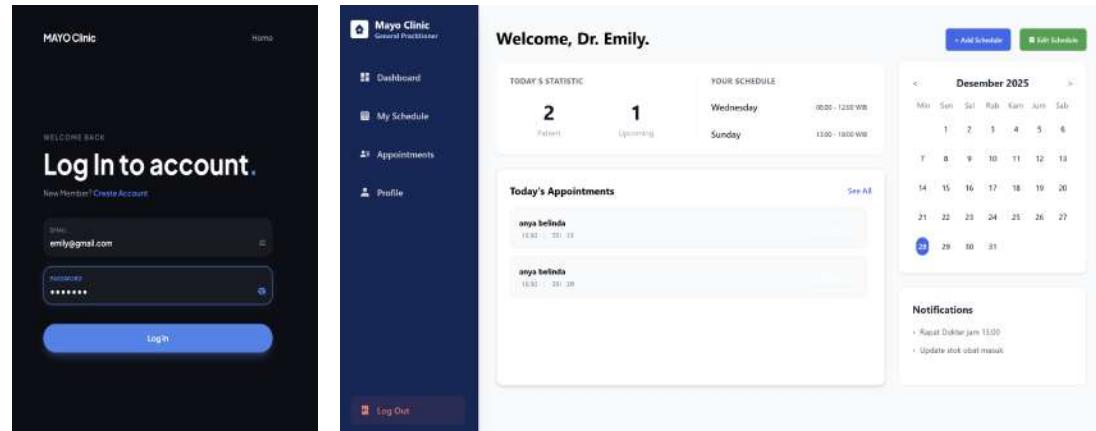


Gambar 29. Perbandingan Tampilan Setelah Melakukan *Search*

2. Pengujian dan Hasil *Role* Dokter

2.1. Login Dokter

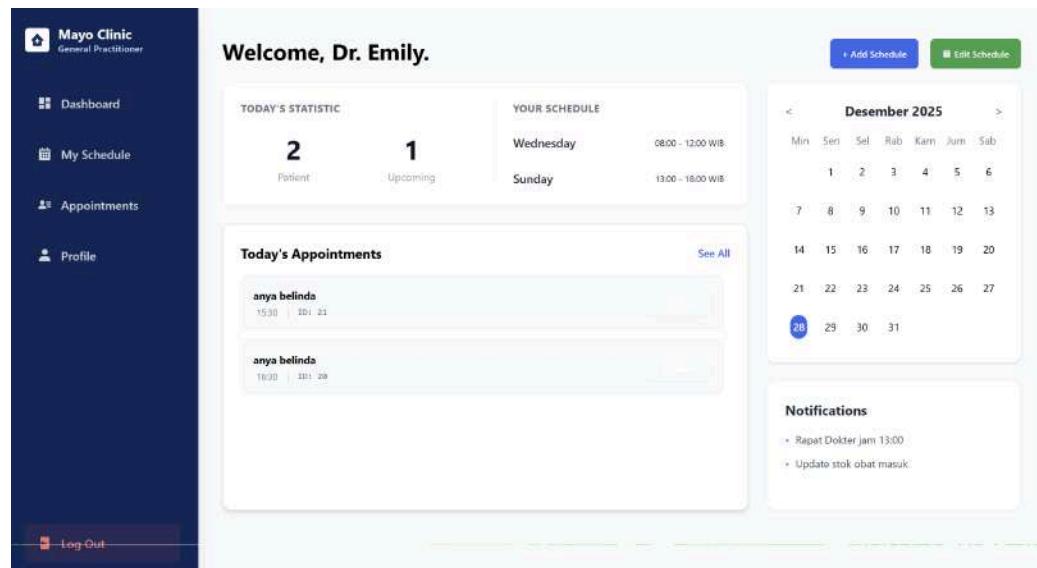
Pengujian dilakukan dengan data dokter yang sudah terdaftarkan, yaitu dokter Emily dengan email emily@gmail.com dan password 111111. Gambar 30 memperlihatkan dokter memasukkan data yang sudah didaftarkan tersebut dan ketika role berhasil diidentifikasi, sistem akan mengarahkan ke *Dashboard* dokter.



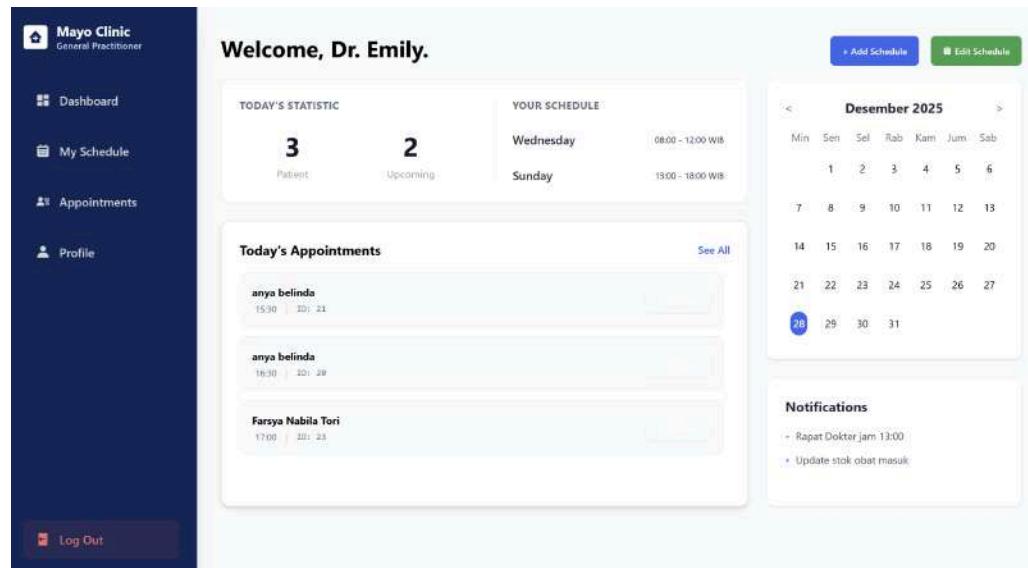
Gambar 30. Testing Login Dokter

2.2. Statistic Update

Statistik pengunjung akan berubah seiring bertambahnya pasien yang membuat janji temu (*appointment*).



(a)

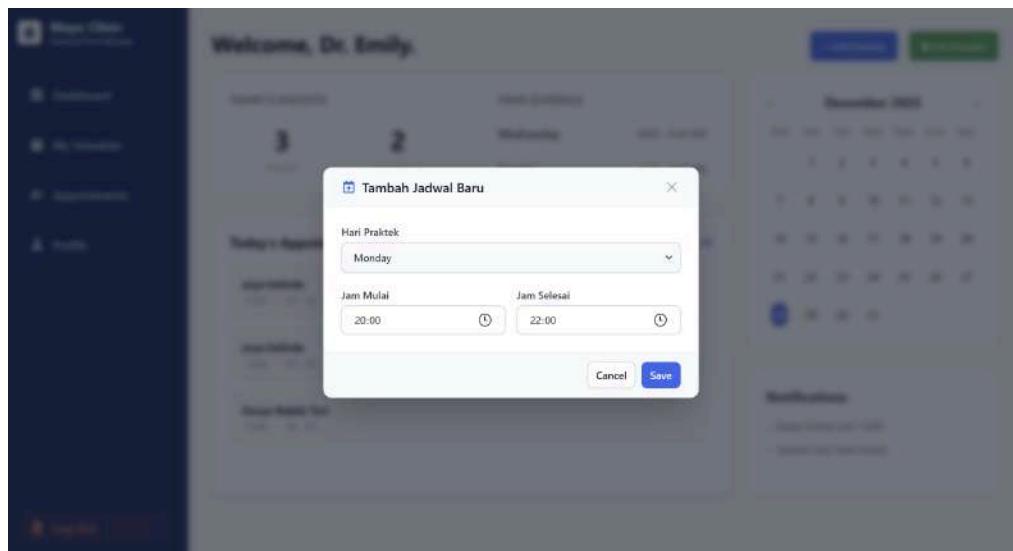


(b)

Gambar 31. (a) Data Statistik Awal (b) Data Statistik setelah *Appointment* Dibuat

2.3. Menambahkan Jadwal

Pada pengujian fitur menambahkan jadwal, akan dilakukan penambahan jadwal untuk Dr. Emily pada hari Senin pukul 20.00–22.00. Pada Gambar 31, dapat dilihat bahwa tidak terdapat jadwal pada hari Senin, baik di *Dashboard* maupun di halaman *My Schedule*.



Gambar 32. Proses Penambahan Jadwal

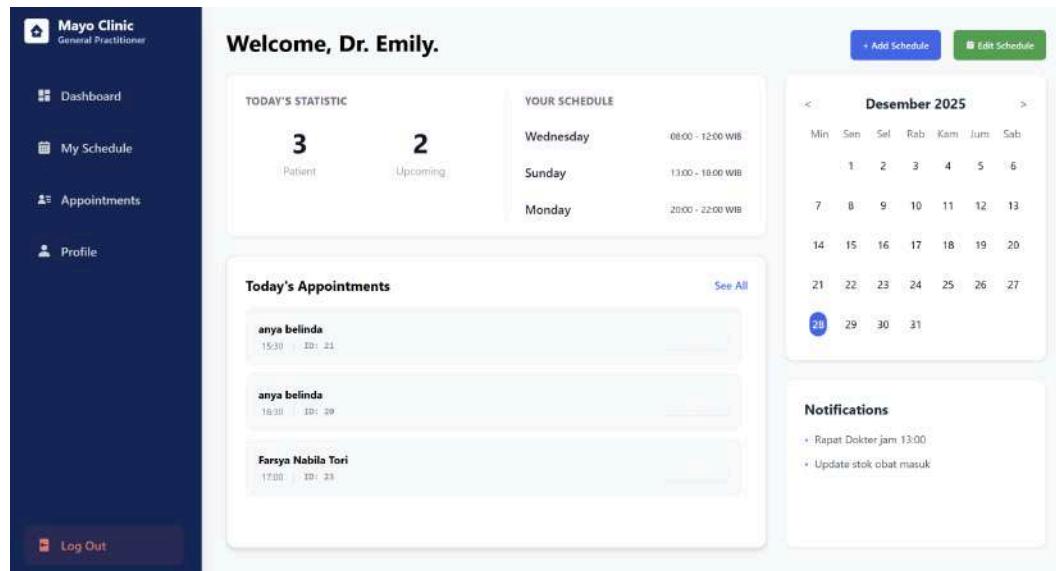
Ketika tombol ‘Save’ ditekan, sistem berhasil melakukan perintah INSERT pada tabel jadwal_dokter. Hasil INSERT dapat dilihat pada Gambar 33.

jadwal_id	hari	jam_mulai	jam_selesai
11	Monday	20:00	22:00

(a) (b)

Gambar 33. Hasil INSERT pada Tabel Jadwal_dokter (a) Sebelum INSERT (b)
Sesudah INSERT

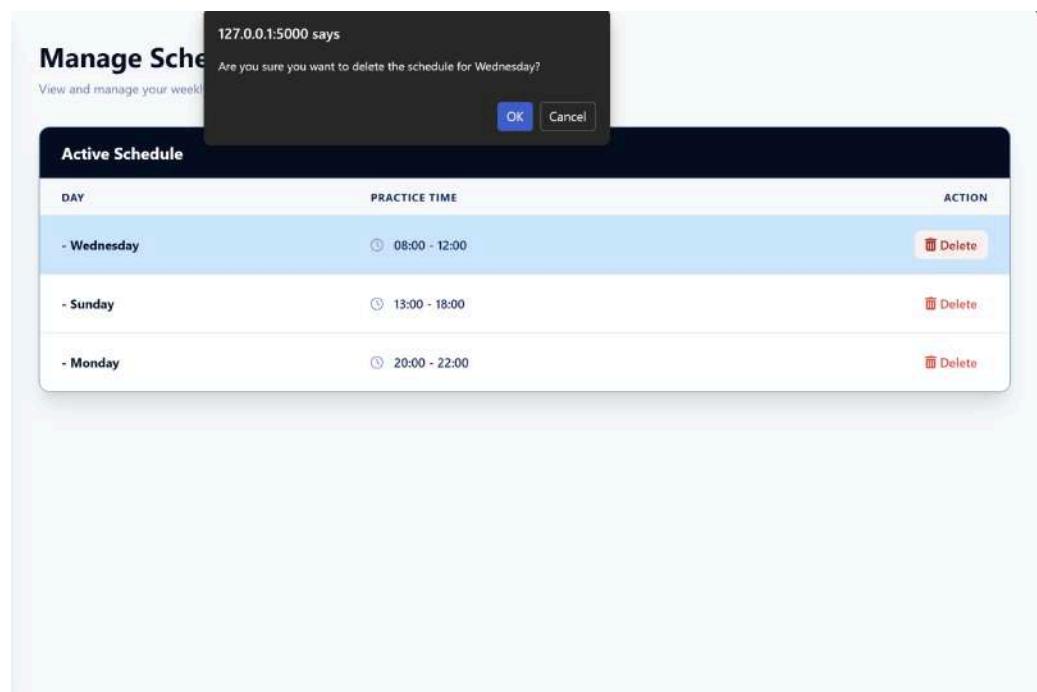
Penambahan jadwal juga memengaruhi tampilan pada halaman *Dashboard* (ditunjukkan oleh Gambar 34), di mana jadwal dokter akan diperbarui dan ditampilkan pada kolom *Your Schedule*.



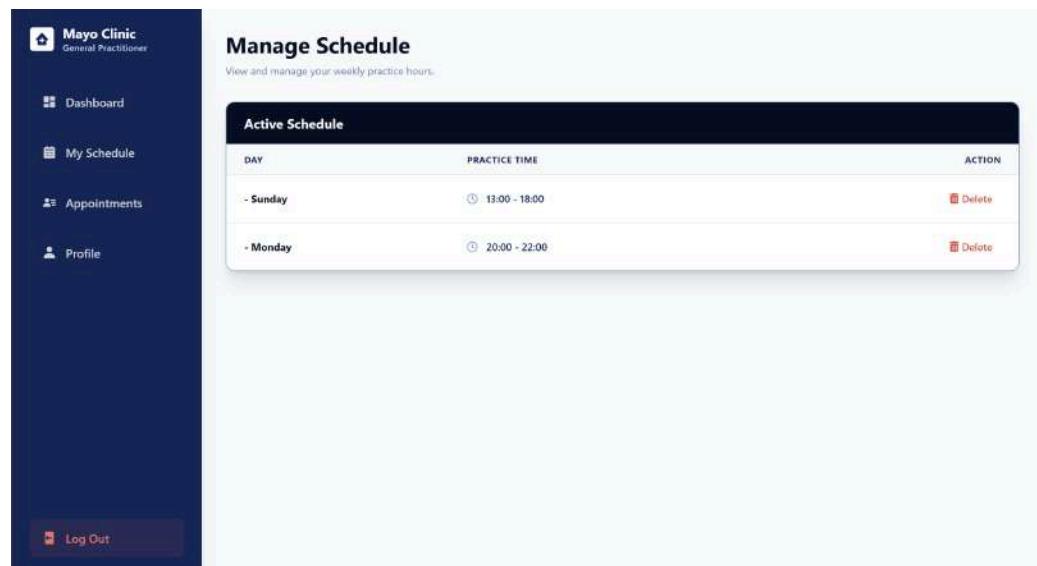
Gambar 34. Halaman *Dasboard* setelah Penambahan Jadwal

2.4. Menghapus Jadwal

Pada pengujian fitur penghapusan jadwal, jadwal yang dipilih untuk dihapus adalah jadwal hari Rabu. Ketika tombol “Delete” ditekan, sistem menampilkan notifikasi konfirmasi untuk memastikan kembali apakah pengguna yakin bahwa jadwal tersebut akan dihapus.



Gambar 35. Proses Menghapus Jadwal



Gambar 36. Tampilan setelah Jadwal Dihapus

1	Monday	09:00	12:00
2	Tuesday	10:00	14:00
4	Thursday	13:00	17:00
5	Friday	09:00	15:00
6	Wednesday	09:00	12:00
7	Sunday	10:00	14:00
8	Sunday	16:00	20:00
9	Sunday	13:00	18:00
11	Monday	20:00	22:00

Gambar 37. Hasil Database setelah Proses Penghapusan Jadwal Berhasil

Gambar 37 menunjukkan bahwa proses DROP untuk jadwal hari Rabu dengan jadwal_id adalah 3 berhasil. Dapat dilihat bahwa jadwal_id 3 sudah tidak ada dalam database.

2.5. Status Appointment

Status janji temu (*appointment*) akan berubah menyesuaikan waktunya. Jika belum waktunya, statusnya adalah terjadwal. Jika sudah waktunya, status berubah menjadi *on-going*. Ketika rekam medis sudah selesai dimasukkan, status akan berubah menjadi *done*.

Untuk pengujian ini, akan ditambahkan 2 pasien baru dengan waktu yang berbeda untuk memperlihatkan perubahan status. Pasien yang ditambahkan adalah Biru Laut dan Lulu Farasya. Dapat dilihat pada Gambar 38 bahwa status Biru laut adalah *on-going* dan Lulu Farasya masih terjadwal.

TIME	PATIENT	STATUS	ACTION
15:30	a anya belinda ID-11	Selesai	History
16:30	a anya belinda ID-11	Selesai	History
17:00	F Farsya Nabila Tori ID-4	On Going	Input Record
18:25	B Biru Laut ID-6	On Going	Input Record
19:00	L Lulu Farasya ID-11	Terjadwal	Not Started
20:00	L Lulu Farasya ID-9	Terjadwal	Not Started

Gambar 38. Status Janji Temu (*Appointment*)

2.6. Menambahkan Rekam Medis

Penambahan rekam medis dapat dilakukan dengan menekan tombol “Input Record” pada halaman *Appointments*. Dokter akan diarahkan ke halaman rekam medis untuk mengisi rekam medis lebih lanjut seperti pada Gambar 39. Untuk pengujian ini, akan digunakan data pasien dengan ID-4.

Patient Information

Farsya Nabila Tori
ID: 4

Gender: Female
Age: 19 Years Old
Appointment Date: 2025-12-28

Diagnosis & Treatment

Please fill in the medical record accurately.

EXAMINATION DATE

28 December 2025

DIAGNOSIS *

Demam Berdarah

MEDICAL NOTES / PRESCRIPTION *

Pusing, Jemas, demam hingga 40 derajat
Tindakan: Rawat Inap

Cancel Save & Finish

Gambar 39. Halaman untuk Mengisi Rekam Medis

Setelah selesai, dokter akan menekan tombol “Save & Finish”. Data rekam medis akan tersimpan dan dapat dilihat kembali pada *history* rekam medis. Dengan berhasilnya fitur ini, proses INSERT ke dalam database juga berhasil. Hasil proses INSERT pada database dapat dilihat pada Gambar 40 dan ditunjukkan oleh rekam_id 6 dimana rekam_id tersebut menyimpan data yang tadi dimasukkan.

The screenshot shows a table titled 'Rekam_Medis' with columns: 'rekam_id', 'diagnosis', 'description', and 'appointment_id'. The data includes:

rekam_id	diagnosis	description	appointment_id
1	Flu Berat	Pasien demam tinggi, diberikan paracetamol d...	1
2	Flu	batuk berdahak, pilek	21
3	flu	batuk batuk	21
4	flu	batuk berdahak, radang, pilek	21
5	diabetes tipe 2	pantau gula darah	20
6	Demam Berdarah	Pusing, lemas, demam hingga 40 derajat\nTin...	23

Gambar 40. Hasil INSERT pada *Database*

Status pasien juga akan berubah menjadi done, seperti yang ditunjukkan oleh Gambar 41.

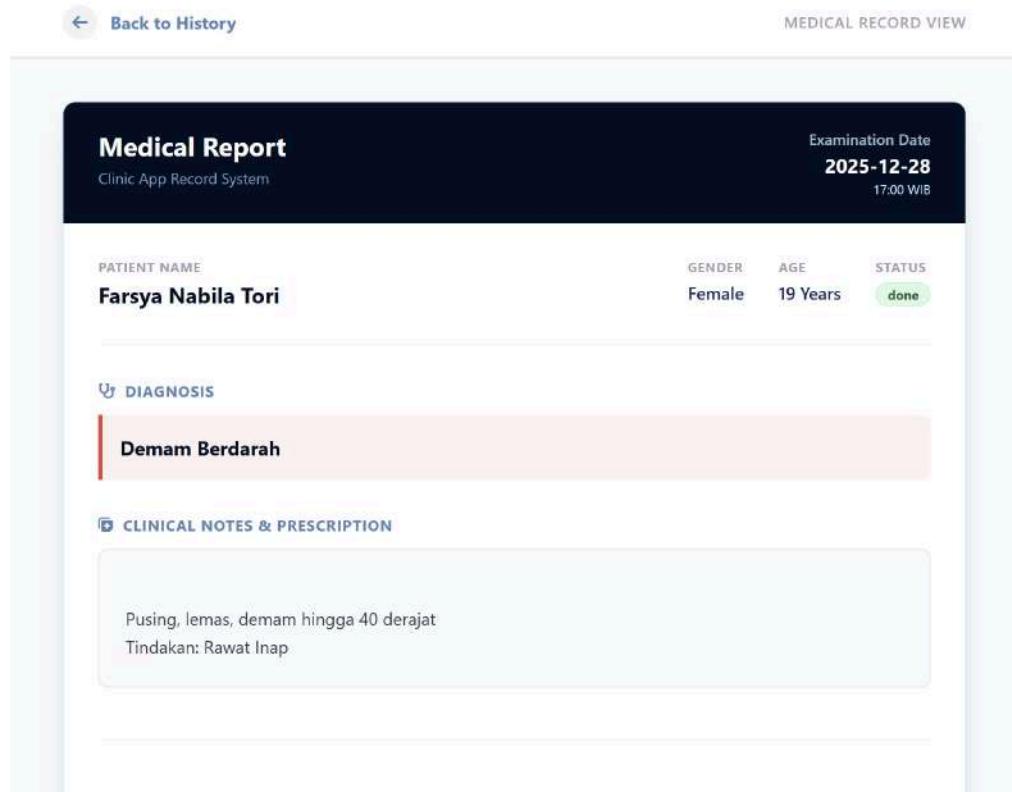
The screenshot shows the 'Patient Queue List' interface with the following details:

- Header:** Mayo Clinic General Practitioner
- Left Sidebar:**
 - Dashboard
 - My Schedule
 - Appointments
 - Profile
 - Log Out
- Top Bar:**
 - SELECT DATE: 28/12/2025
 - SEARCH PATIENT: Type Name or Patient ID...
 - FILTER STATUS: All Status
- Table:**| TIME | PATIENT | STATUS | ACTION |
| --- | --- | --- | --- |
| 15:30 | a anya belinda ID-11 | Sellesai | History |
| 16:30 | a anya belinda ID-11 | Sellesai | History |
| 17:00 | F Farsya Nabila Tori ID-4 | Sellesai | History |
| 18:25 | B Biru Laut ID-6 | On Going | Input Record |
| 19:00 | L Lulu Farasya ID-9 | Terjadwal | Not Started |
| 20:00 | L Lulu Farasya ID-9 | Terjadwal | Not Started |

Gambar 41. Perubahan Status Janji Temu (*Appointment*)

2.7. Melihat *History* Rekam Medis

History rekam medis dapat dilihat dengan menekan tombol “*History*” pada halaman Appointment. Dokter akan diarahkan ke halaman *medical history*. Pada tahapan ini digunakan proses SELECT untuk mengambil data dari *database*. Gambar 42 menunjukkan hasil rekam medis pada pasien ID-4.



Gambar 42. Rekam Medis Pasien ID-4

2.8. Edit Profile

Pada pengujian ini, beberapa data dari Dr. Emily akan diubah, seperti nama belakang menjadi Huston dan nomor telepon menjadi 08123456789088.

The screenshot shows the 'Profile Settings' page for 'dr. Emily'. On the left sidebar, there are links for Dashboard, My Schedule, Appointments, and Profile, with 'Profile' being the active tab. The main area displays a placeholder profile picture for 'dr. Emily' and her joined date as '2025-11-23'. The 'Personal Information' section contains fields for First Name ('Emily') and Last Name ('Huston'), Phone Number ('08123456789088'), and Status ('Active'). The 'Professional Bio' field contains placeholder text: 'Contohnya: Berpengalaman 5 tahun dalam menangani...'. The 'Security' section includes fields for New Password and Confirm New Password, with a note that leaving them blank keeps the current password. At the bottom right are 'Cancel' and 'Save Changes' buttons.

Gambar 43. Proses *Edit Profile*

Pada fitur ini, digunakan proses UPDATE ke *database* dan sistem ini dinyatakan berhasil karena pada database berhasil diubah. Dapat dilihat pada Gambar 44 dimana data Dr. Emily berhasil diubah dari Emily Carter menjadi Emily Huston, demikian juga dengan nomor telepon.

dokter_id	nama_depan	nama_belakang	tanggal_masuk	status	email	password	t
1	Rafathin	Ardian	2025-11-23	Active	NULL	NULL	c
2	James	Wilson	2025-11-23	Active	NULL	NULL	c
3	Emily	Huston	2025-11-23	Active	emily@gmail.com	1111111	c
4	Michael	Brown	2025-11-23	Active	NULL	NULL	c
5	David	Kim	2025-11-23	Active	NULL	NULL	c
6	Aliya	Rahman	2025-11-23	Active	aliya@gmail.com	11111111	c

(a)

Tables / Dokter_telepon		Column
telepon	dokter_id	
08123456789088	3	⋮
+ Row		

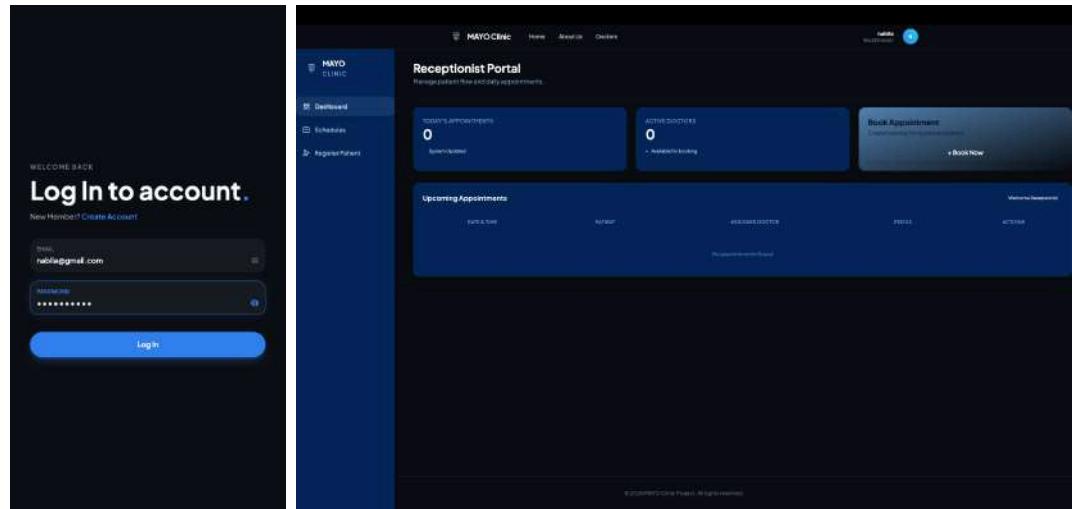
(b)

Gambar 44. Hasil proses UPDATE pada (a) Tabel Dokter (b) Tabel Telepon_dokter

3. Pengujian dan Hasil *Role* Resepsionis

3.1. Pengujian Login Resepsionis

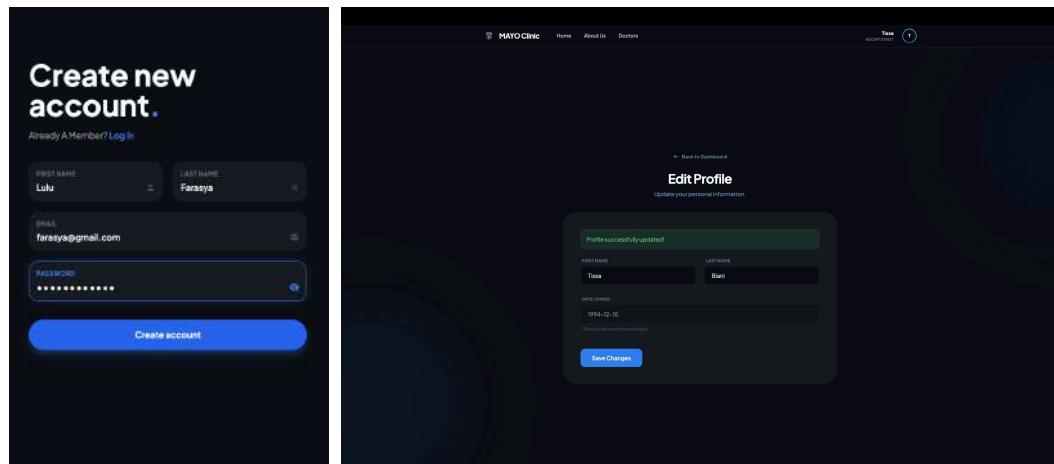
Gambar 45. memperlihatkan hasil pengujian fitur autentikasi untuk peran Resepsionis. Pengguna memasukkan kredensial yang valid, dan sistem memberikan respon sukses dengan mengarahkan pengguna ke halaman utama (*Dashboard*). Pada bagian *header* (pojok kanan atas), sistem berhasil menampilkan identitas pengguna yang sedang aktif, yang menandakan bahwa sesi (*session*) telah berhasil dibuat dan disimpan dengan aman.



Gambar 45. Pengujian Login Resepsionis

3.2. Pengujian Registrasi Pasien

Gambar 46 menunjukkan formulir pendaftaran pasien baru yang dijalankan oleh resepsionis. Pengujian dilakukan dengan mengisi seluruh *field* wajib (nama, *email*, *password*). Hasilnya, sistem berhasil memvalidasi input dan menambahkan data pasien baru ke *database* sehingga nama pasien tersebut langsung dapat ditemukan pada fitur pencarian saat pembuatan janji temu.



Gambar 46. Pengujian Registrasi Pasien

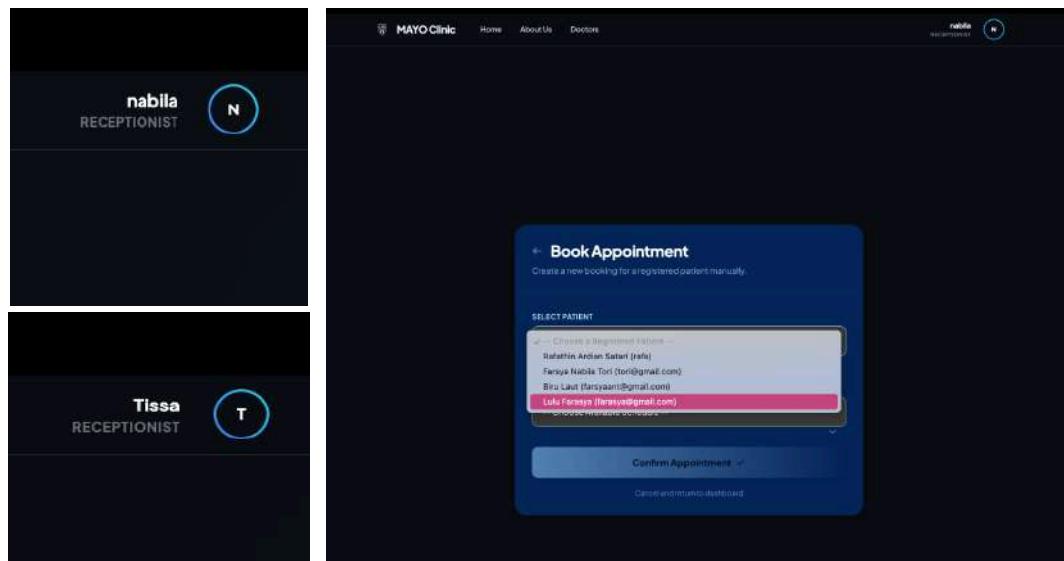
Sinkronisasi data dikonfirmasi melalui tabel Pasien. Sistem berhasil melakukan perintah INSERT data baru, termasuk melakukan enkripsi otomatis pada kolom *password* untuk keamanan.

#	pasien_id	nama_depan	nama_belakang	email	gender	tanggal_daftar	tanggal_lahir	password	kota	jalan
	varchar(50)	varchar(50)	varchar(50)	varchar(100)	enum('Male','Female')	datetime	date	varchar(256)	varchar(50)	varchar(256)
> 2	1	Rifathin	Ardian Satri	rifa	Male	2025-11-17 00:00:00	2007-05-14	pbkdf2:sha256:100000\$9tgc	Makassar	Jl. Nikel 3
> 4	2	Farsya	Nabila Tori	tori@gmail.com	Female	2025-11-23 00:00:00	2008-02-02	pbkdf2:sha256:100000\$9tgc	Medan	Jl. Kalurang KM. 20
> 6	3	Biyu	Laut	farsyant@gmail.com	Male	2015-12-27 08:07:39	2007-05-06	pbkdf2:sha256:100000\$9tgc	Medan	Jl. Tuscan
> 9	4	Lulu	Farsyata	farsyata@gmail.com	Female	2025-12-28 01:03:06	1999-02-28	pbkdf2:sha256:100000\$9tgc	Jakarta	Jl. Sudirman
> 10	5	Aliza	Kharun	aliza@gmail.com	Female	2025-12-28 02:08:28	2006-06-21	pbkdf2:sha256:100000\$9tgc	Tangerang Selatan	Separut
> 11	6	anya	Iulinda	anya@gmail.com	Female	2025-12-28 05:27:33	2009-12-31	pbkdf2:sha256:100000\$44c	Toronto	124 Main St

Gambar 47. Validasi Tabel Pasien pada Database

3.3. Pengujian Manajemen Profil

Pada pengujian fitur pembaruan profil, dilakukan skenario perubahan nama depan pengguna dari data lama menjadi data baru ('Tissa'). Seperti yang terlihat pada Gambar 48, sistem berhasil menyimpan perubahan ke dalam *database*. Selain itu, mekanisme pembaruan sesi (*session update*) juga terbukti berfungsi dengan baik, ditandai dengan berubahnya nama tampilan pada *navigation bar* secara otomatis tanpa mengharuskan pengguna untuk melakukan *logout* dan *login* ulang.



Gambar 48. Pengujian Manajemen Profil

Perubahan data profil diverifikasi pada tabel Resepsonis. Kolom nama_depan terbukti telah diperbarui dari nilai lama menjadi nilai baru ('Tissa') secara *real-time*.

Properties							DATA	Log	ER	Monitor	0	1
SELECT * FROM 'Resepsonis' LIMIT 100												
	resepsonis_id	nama_depan	nama_belakang	tanggal_masuk	status	email	password					
>	1	Aliya	Rahman	2025-11-23	Active	aliya@gmail.com	11111111					
>	2	Tissa	Biani	1994-12-15	Aktif	nabila@gmail.com	pbkdf2:sha256:100000\$1Mf					

Gambar 49. Validasi Tabel Resepsonis pada Database

3.4. Pengujian Manajemen Jadwal Dokter

Gambar 50 menunjukkan antarmuka fitur *Doctor's Schedule* yang diakses oleh resepsionis. Hasil pengujian menunjukkan bahwa sistem mampu menarik data jadwal dari *database* dan memvisualisasikannya dalam format kartu harian (nama dokter dan jam praktik), yang memudahkan resepsionis dalam memberikan informasi ketersediaan dokter kepada pasien secara cepat dan akurat.

The screenshot shows a dark-themed web application interface titled "Doctor's Schedule". At the top, there is a navigation bar with links for "Home", "About Us", and "Doctors". Below the title, a sub-header reads: "Check our specialists' availability here. Find the perfect time for your health consultation." The main content area is a grid of seven columns, each representing a day of the week from Monday to Sunday. Each column contains two rows of appointment slots, each with a doctor's name and a time range. A "Book Appointment" button is located at the bottom of each slot. The days of the week are labeled as follows: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday.

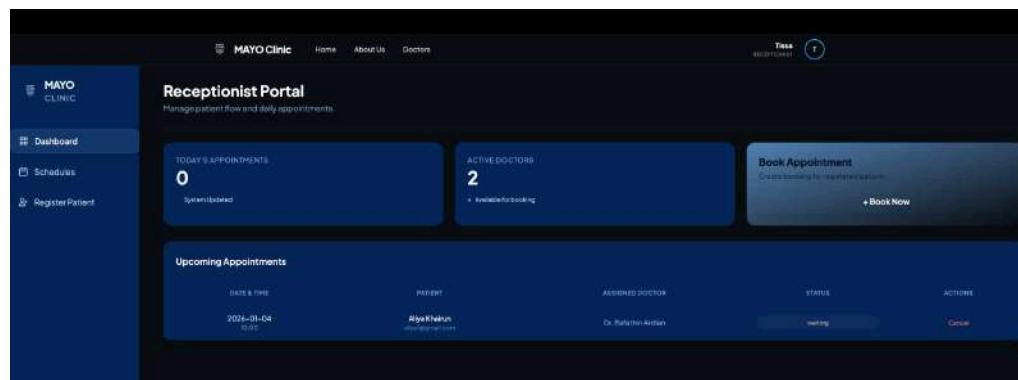
Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Dr. Rafathin Ardian 08:00 – 11:00 Book Appointment	Dr. James Wilson 10:00 – 12:00 Book Appointment	Dr. Emily Carter 08:00 – 11:00 Dr. David Kim 09:00 – 12:00 Book Appointment	Dr. Michael Brown 11:00 – 14:00 Book Appointment	Dr. David Kim 08:00 – 11:00 Book Appointment	No appointment available	No appointment available

Gambar 50. Pengujian Manajemen Jadwal Dokter

3.5. Pengujian *Walk-in Appointment*

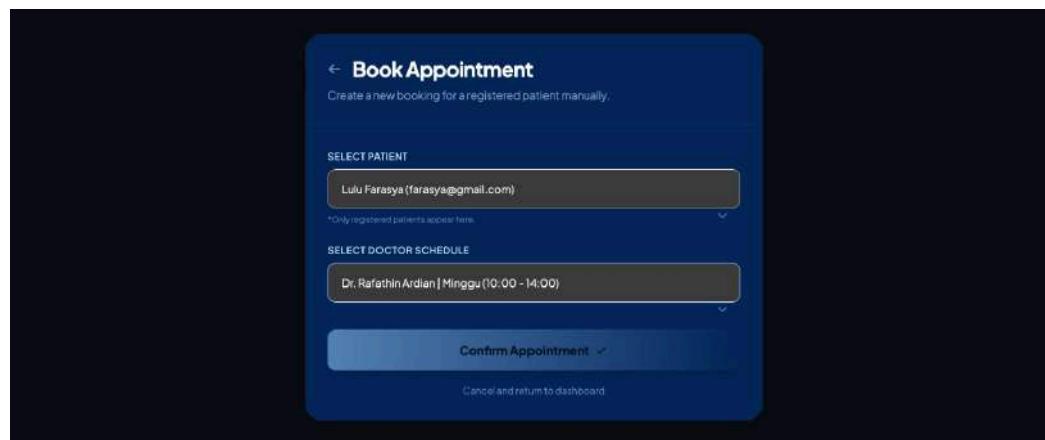
Gambar 51 menunjukkan pengujian fitur *walk-in* dengan memilih data pasien dan jadwal dokter melalui menu *dropdown*. Hasilnya berhasil, sistem berhasil memproses input dan menyimpan janji temu baru ke tabel *appointments* tanpa kendala teknis.

Skenario pengujian ini memverifikasi logika penjadwalan dinamis pada *dashboard* resepsionis yang diuji pada hari Minggu, di mana sistem pada awalnya mendeteksi ketersediaan dua dokter aktif.



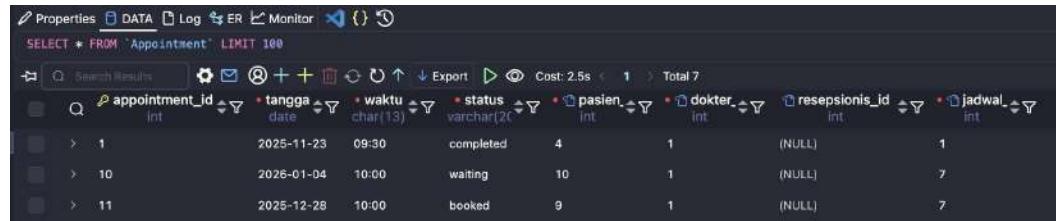
Gambar 51. Deteksi Ketersediaan Dokter pada Hari Minggu

Resepsionis kemudian melakukan proses pendaftaran manual (*walk-in*) untuk pasien 'Lulu Farasya' dengan memilih salah satu slot waktu yang tersedia.



Gambar 52. Proses Pendaftaran Manual (*Walk-In*)

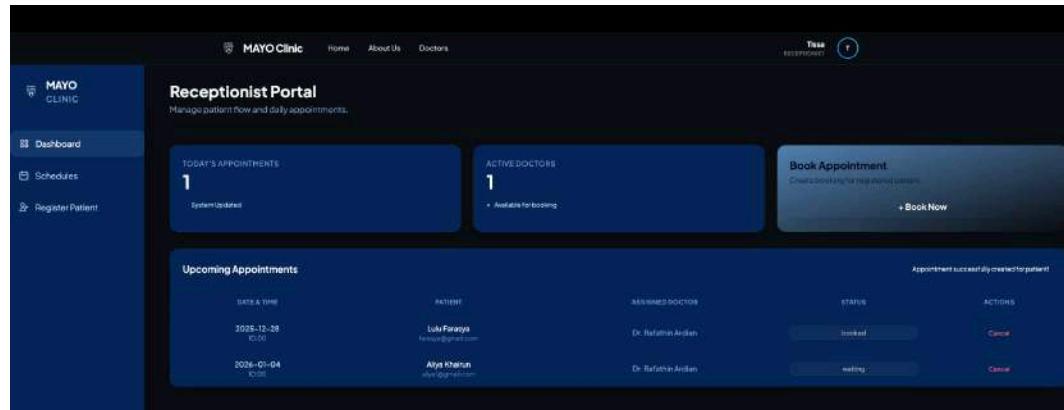
Validasi penyimpanan data dilakukan dengan memeriksa tabel Appointment pada database. Terlihat pada Gambar 53 di baris ketiga, data janji temu baru berhasil *ter-generate* dengan ID unik, serta terhubung dengan ID pasien dan ID dokter yang sesuai dengan input pada antarmuka.



	appointment_id	tanggal	waktu	status	pasien	dokter	resepsionis_id	jadwal
	1	2025-11-23	09:30	completed	4	1	(NULL)	1
	10	2026-01-04	10:00	waiting	10	1	(NULL)	7
	11	2025-12-28	10:00	booked	9	1	(NULL)	7

Gambar 53. Validasi Tabel Appointment pada *Database*

Kemudian, setelah konfirmasi berhasil, sistem merespons secara *real-time* dengan mengurangi jumlah '*Active Doctors*' menjadi satu (karena slot telah terpakai), menambah indikator '*Today's Appointments*', serta secara akurat menampilkan data janji temu pasien tersebut ke dalam tabel '*Upcoming Appointments*'.



Gambar 54. Pendaftaran *Walk-In* Berhasil

BAB VI

KESIMPULAN DAN REFLEKSI

I. KESIMPULAN

Berdasarkan seluruh tahapan pengembangan yang telah dilakukan, kami menyimpulkan bahwa proyek sistem janji temu klinik ini telah berhasil diimplementasikan dengan mengintegrasikan basis data relasional MySQL yang *di-deploy* di Railway dan aplikasi web berbasis Flask. Sistem ini mampu menangani proses bisnis utama, mulai dari manajemen jadwal dokter, proses *booking* oleh pasien, hingga pencatatan rekam medis yang terintegrasi secara otomatis. Penggunaan *constraints* seperti *Foreign Key* dengan aksi *CASCADE* dan *SET NULL* terbukti sangat krusial dalam menjaga integritas data antartabel, seperti pada hubungan antara tabel *Appointment* dengan Pasien serta Dokter. Secara keseluruhan, aplikasi ini dapat menjadi solusi digital yang efektif untuk meningkatkan efisiensi administrasi di lingkungan klinik.

II. REFLEKSI

Selama proses pengerjaan tugas besar ini, kami mendapatkan banyak pelajaran berharga mengenai kompleksitas pengembangan sistem informasi. Salah satu tantangan terbesar yang kami hadapi sebagai kelompok adalah memastikan sinkronisasi antara logika di sisi *backend* dengan batasan (*constraints*) yang ada di dalam basis data. Kami menyadari bahwa perancangan skema relasional yang matang, seperti penanganan atribut *multivalued* untuk nomor telepon, sangat memudahkan proses pengembangan fitur di tahap selanjutnya. Selain aspek teknis, kerja kelompok ini juga melatih kemampuan komunikasi dan pembagian tugas kami dalam menyelesaikan kendala yang muncul, seperti penanganan *conflict fetch data* pada kode program atau validasi *login* pengguna. Pengalaman ini memberikan gambaran nyata bagi kami sebagai mahasiswa mengenai bagaimana teori basis data diaplikasikan dalam sebuah produk perangkat lunak yang

fungsional dan responsif. Kami menyadari masih terdapat ruang untuk pengembangan lebih lanjut, tetapi pencapaian saat ini telah memberikan fondasi yang kuat bagi kami dalam memahami manajemen data yang sistematis.

BAB VII

APPENDIX

SQL

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

CREATE TABLE `Appointment` (
    `appointment_id` int(11) NOT NULL,
    `tanggal` date NOT NULL,
    `waktu` char(13) NOT NULL,
    `status` varchar(20) NOT NULL,
    `pasien_id` int(11) NOT NULL,
    `dokter_id` int(11) NOT NULL,
    `resepionis_id` int(11) DEFAULT NULL,
    `jadwal_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

INSERT INTO `Appointment` (`appointment_id`, `tanggal`, `waktu`, `status`,
    `pasien_id`, `dokter_id`, `resepionis_id`, `jadwal_id`) VALUES
(1, '2025-11-23', '09:30', 'completed', 4, 1, NULL, 1);

CREATE TABLE `Dijadwalkan` (
    `dokter_id` int(11) NOT NULL,
    `jadwal_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

INSERT INTO `Dijadwalkan` (`dokter_id`, `jadwal_id`) VALUES
(1, 1),
(2, 2),
(3, 3),
(4, 4),
(5, 5);

CREATE TABLE `Dokter` (
```

```

`dokter_id` int(11) NOT NULL,
`nama_depan` varchar(50) NOT NULL,
`nama_belakang` varchar(50) NOT NULL,
`tanggal_masuk` date NOT NULL,
`status` varchar(20) NOT NULL,
`email` varchar(100) DEFAULT NULL,
`password` varchar(255) DEFAULT NULL,
`foto` varchar(255) DEFAULT 'default.jpg'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

INSERT INTO `Dokter` (`dokter_id`, `nama_depan`, `nama_belakang`,
`tanggal_masuk`, `status`, `email`, `password`, `foto`) VALUES
(1, 'Rafathin', 'Ardian', '2025-11-23', 'Active', NULL, NULL,
'doctor1.jpg'),
(2, 'James', 'Wilson', '2025-11-23', 'Active', NULL, NULL, 'doctor2.jpg'),
(3, 'Emily', 'Carter', '2025-11-23', 'Active', NULL, NULL, 'doctor3.jpg'),
(4, 'Michael', 'Brown', '2025-11-23', 'Active', NULL, NULL, 'doctor4.jpg'),
(5, 'David', 'Kim', '2025-11-23', 'Active', NULL, NULL, 'doctor5.jpg');

CREATE TABLE `Dokter_telepon` (
`telepon` varchar(50) NOT NULL,
`dokter_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

CREATE TABLE `Jadwal_dokter` (
`jadwal_id` int(11) NOT NULL,
`hari` varchar(10) NOT NULL,
`jam_mulai` char(5) NOT NULL,
`jam_selesai` char(5) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

INSERT INTO `Jadwal_dokter` (`jadwal_id`, `hari`, `jam_mulai`,
`jam_selesai`) VALUES
(1, 'Senin', '09:00', '12:00'),
(2, 'Selasa', '10:00', '14:00'),
(3, 'Rabu', '08:00', '12:00'),
(4, 'Kamis', '13:00', '17:00'),

```

```

(5, 'Jumat', '09:00', '15:00');

CREATE TABLE `Pasien` (
  `pasien_id` int(11) NOT NULL,
  `nama_depan` varchar(50) DEFAULT NULL,
  `nama_belakang` varchar(50) DEFAULT NULL,
  `email` varchar(100) NOT NULL,
  `gender` enum('Male','Female') DEFAULT NULL,
  `tanggal_daftar` date DEFAULT current_timestamp(),
  `tanggal_lahir` date DEFAULT NULL,
  `password` varchar(255) DEFAULT NULL,
  `kota` varchar(50) DEFAULT NULL,
  `jalan` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

INSERT INTO `Pasien` (`pasien_id`, `nama_depan`, `nama_belakang`, `email`,
  `gender`, `tanggal_daftar`, `tanggal_lahir`, `password`, `kota`, `jalan`)
VALUES
(2, 'Rafathin', 'Ardian Satari', 'rafa', 'Male', '2025-11-17', '2007-05-14',
  'aliya', 'Makassar', 'Jl Nikel 3'),
(4, 'Farsya', 'Nabila Tori', 'tori@gmail.com', 'Female', '2025-11-23',
  '2006-02-02',
  'pbkdf2:sha256:100000$MtqjXlpCU1aw13Tc$1ece0052c0020d8a78951a7b0b7d7d955
  05aed57d6a0325699e33c4b5f94e64e', 'Medan', 'JL Kaliurang KM. 200 No.
  28');

CREATE TABLE `Pasien_telepon` (
  `telepon` varchar(50) NOT NULL,
  `pasien_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

INSERT INTO `Pasien_telepon` (`telepon`, `pasien_id`) VALUES
('0838238974328', 2),
('2938472892323', 4),
('3874823473847', 2),
('3928473284732', 2);

```

```

CREATE TABLE `Rekam_medis` (
  `rekam_id` int(11) NOT NULL,
  `diagnosis` varchar(100) NOT NULL,
  `description` varchar(255) NOT NULL,
  `appointment_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

INSERT INTO `Rekam_medis` (`rekam_id`, `diagnosis`, `description`,
`appointment_id`) VALUES
(1, 'Flu Berat', 'Pasien demam tinggi, diberikan paracetamol dan
istirahat.', 1);

CREATE TABLE `Resepsonis` (
  `resepsonis_id` int(11) NOT NULL,
  `nama_depan` varchar(50) NOT NULL,
  `nama_belakang` varchar(50) NOT NULL,
  `tanggal_masuk` date NOT NULL,
  `status` varchar(20) NOT NULL,
  `email` varchar(100) DEFAULT NULL,
  `password` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

CREATE TABLE `Resepsonis_telepon` (
  `telepon` varchar(50) NOT NULL,
  `resepsonis_id` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

ALTER TABLE `Appointment`
  ADD PRIMARY KEY (`appointment_id`),
  ADD KEY `pasien_id` (`pasien_id`),
  ADD KEY `dokter_id` (`dokter_id`),
  ADD KEY `resepsonis_id` (`resepsonis_id`),
  ADD KEY `jadwal_id` (`jadwal_id`);

ALTER TABLE `Dijadwalkan`
  ADD PRIMARY KEY (`dokter_id`, `jadwal_id`),
  ADD KEY `jadwal_id` (`jadwal_id`);

```

```

ALTER TABLE `Dokter`
ADD PRIMARY KEY (`dokter_id`),
ADD UNIQUE KEY `email` (`email`);

ALTER TABLE `Dokter_telepon`
ADD PRIMARY KEY (`telepon`, `dokter_id`),
ADD KEY `dokter_id` (`dokter_id`);

ALTER TABLE `Jadwal_dokter`
ADD PRIMARY KEY (`jadwal_id`);

ALTER TABLE `Pasien`
ADD PRIMARY KEY (`pasien_id`);

ALTER TABLE `Pasien_telepon`
ADD PRIMARY KEY (`telepon`, `pasien_id`),
ADD KEY `pasien_id` (`pasien_id`);

ALTER TABLE `Rekam_medis`
ADD PRIMARY KEY (`rekam_id`),
ADD KEY `appointment_id` (`appointment_id`);

ALTER TABLE `Resepsionis`
ADD PRIMARY KEY (`resepsionis_id`),
ADD UNIQUE KEY `email` (`email`);

ALTER TABLE `Resepsionis_telepon`
ADD PRIMARY KEY (`telepon`, `resepsionis_id`),
ADD KEY `resepsionis_id` (`resepsionis_id`);

ALTER TABLE `Appointment`
MODIFY `appointment_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;

ALTER TABLE `Dokter`
MODIFY `dokter_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;

```

```

ALTER TABLE `Jadwal_dokter`
MODIFY `jadwal_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=6;

ALTER TABLE `Pasien`
MODIFY `pasien_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;

ALTER TABLE `Rekam_medis`
MODIFY `rekam_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;

ALTER TABLE `Resepsionis`
MODIFY `resepsionis_id` int(11) NOT NULL AUTO_INCREMENT;

ALTER TABLE `Appointment`
ADD CONSTRAINT `appointment_ibfk_1` FOREIGN KEY (`pasien_id`) REFERENCES
`Pasien`(`pasien_id`) ON DELETE CASCADE,
ADD CONSTRAINT `appointment_ibfk_2` FOREIGN KEY (`dokter_id`) REFERENCES
`Dokter`(`dokter_id`) ON DELETE CASCADE,
ADD CONSTRAINT `appointment_ibfk_3` FOREIGN KEY (`resepsionis_id`)
REFERENCES `Resepsionis`(`resepsionis_id`) ON DELETE SET NULL,
ADD CONSTRAINT `appointment_ibfk_4` FOREIGN KEY (`jadwal_id`) REFERENCES
`Jadwal_dokter`(`jadwal_id`) ON DELETE CASCADE;

ALTER TABLE `Dijadwalkan`
ADD CONSTRAINT `dijadwalkan_ibfk_1` FOREIGN KEY (`dokter_id`) REFERENCES
`Dokter`(`dokter_id`) ON DELETE CASCADE,
ADD CONSTRAINT `dijadwalkan_ibfk_2` FOREIGN KEY (`jadwal_id`) REFERENCES
`Jadwal_dokter`(`jadwal_id`) ON DELETE CASCADE;

ALTER TABLE `Dokter_telepon`
ADD CONSTRAINT `dokter_telepon_ibfk_1` FOREIGN KEY (`dokter_id`)
REFERENCES `Dokter`(`dokter_id`) ON DELETE CASCADE;

ALTER TABLE `Pasien_telepon`
ADD CONSTRAINT `pasien_telepon_ibfk_1` FOREIGN KEY (`pasien_id`)
REFERENCES `Pasien`(`pasien_id`);

ALTER TABLE `Rekam_medis`

```

```
ADD CONSTRAINT `rekam_medis_ibfk_1` FOREIGN KEY (`appointment_id`)
REFERENCES `Appointment` (`appointment_id`) ON DELETE CASCADE;

ALTER TABLE `Resepsionis_telepon`
ADD CONSTRAINT `resepsionis_telepon_ibfk_1` FOREIGN KEY (`resepsionis_id`)
REFERENCES `Resepsionis` (`resepsionis_id`) ON DELETE CASCADE;

COMMIT;
```

DAFTAR PUSTAKA

- Ikhyana, F. K., Setyawan, F. E. B., Pratama, P., & Iswanti, Y. (2023). Keefektifan sistem pendaftaran dan antre online terhadap pelayanan kesehatan. *CoMPHI Journal: Community Medicine and Public Health of Indonesia Journal*, 4(1).
- Rohman, H., & Marsilah, M. (2022). Pemanfaatan sistem pendaftaran online melalui aplikasi WhatsApp di rumah sakit umum daerah. *Indonesian of Health Information Management Journal (INOHIM)*, 10(1), 18-26.
- Susilo, H., Abdillah, N., Ihksan, M., Morika, H. D., & Darma, I. Y. (2023). Analisa Dan Perancangan Sistem Informasi Booking Antrian Pelayan Pada Klinik Medika Saintika Berbasis Website. *Jurnal Kesehatan Medika Saintika*, 14(1), 344-352.
- Syahidin, Y., & Trioktafiani, Y. (2019). Perancangan Sistem Informasi Pasien Appointment Rawat Jalan Berbasis Web. *Jurnal Cendikia*, 18(1), 338-342.