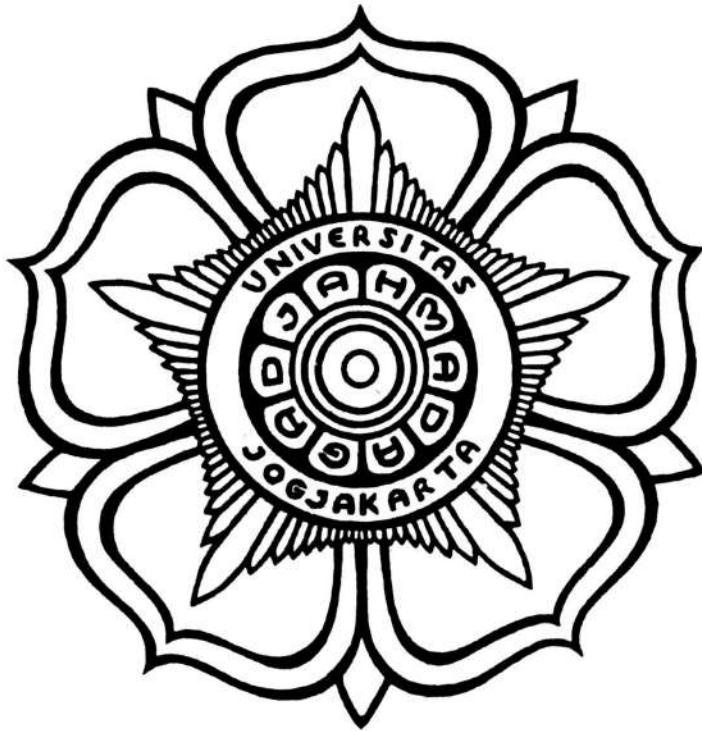


LAPORAN PROYEK AKHIR

**PENGEMBANGAN SISTEM INFORMASI MANAJEMEN KLINIK (CLINIC
APPOINTMENT SYSTEM)**



Disusun untuk Memenuhi Tugas Mata Kuliah Basis Data

Disusun Oleh - Kelompok 3:

Chelsea Natasja Jesslyne Sembiring (24/543571/PA/23097)

Miera Ardiyanti Farica (24/534436/PA/22665)

Universitas Gadjah Mada

Yogyakarta

November 2025

LAPORAN PROYEK AKHIR.....	1
PENGEMBANGAN SISTEM INFORMASI MANAJEMEN KLINIK (CLINIC APPOINTMENT SYSTEM).....	1
1. PENDAHULUAN (INTRODUCTION).....	3
1.1 Latar Belakang Masalah.....	3
1.2 Tujuan Proyek.....	3
1.3 Target Pengguna.....	3
2. PERANCANGAN BASIS DATA (DATABASE DESIGN).....	4
2.1 Entity Relationship Diagram (ERD).....	4
2.2 Normalisasi Data.....	5
3. IMPLEMENTASI BASIS DATA (DATABASE IMPLEMENTATION).....	8
3.1 Skema Relasional (DDL).....	8
3.2 Contoh Manipulasi Data (DML).....	9
4. IMPLEMENTASI APLIKASI (APPLICATION IMPLEMENTATION).....	11
4.1 Struktur Modul Aplikasi.....	11
4.2 Antarmuka Pengguna (User Interface).....	15
5. PENGUJIAN DAN HASIL (TESTING AND RESULTS).....	16
5.1 Skenario Pengujian.....	16
6. KESIMPULAN DAN REFLEksi (CONCLUSION AND REFLECTION).....	33
6.1 Kesimpulan.....	33
6.2 Tantangan.....	33
6.3 Saran Pengembangan.....	33

1. PENDAHULUAN (INTRODUCTION)

1.1 Latar Belakang Masalah

Dalam operasional fasilitas kesehatan, manajemen penjadwalan janji temu (*appointment*) merupakan komponen krusial. Berdasarkan analisis domain masalah, banyak klinik saat ini masih menghadapi inefisiensi akibat proses manual berbasis kertas. Hal ini menyebabkan risiko kesalahan penjadwalan (*double booking*), waktu tunggu pasien yang lama, serta kesulitan dalam pelacakan riwayat medis.

Selain itu, terdapat dorongan eksternal berupa perubahan ekspektasi pasien yang menginginkan akses layanan digital yang cepat, serta kewajiban regulasi pemerintah (Permenkes No. 24 Tahun 2022) yang mewajibkan fasilitas kesehatan untuk beralih ke Rekam Medis Elektronik (RME) dan integrasi data digital. Oleh karena itu, diperlukan sebuah sistem terpusat yang dapat mengelola siklus hidup janji temu secara efisien.

1.2 Tujuan Proyek

Tujuan utama dari pengembangan *Clinic Appointment System* ini adalah:

1. Sentralisasi Data: Menyediakan *single source of truth* untuk data pasien, dokter, klinik, dan jadwal.
2. Efisiensi Operasional: Mengotomatisasi proses pendaftaran dan penjadwalan untuk mengurangi beban administratif.
3. Integritas Data: Memastikan konsistensi data melalui penerapan batasan basis data (*constraints*) yang tepat.

1.3 Target Pengguna

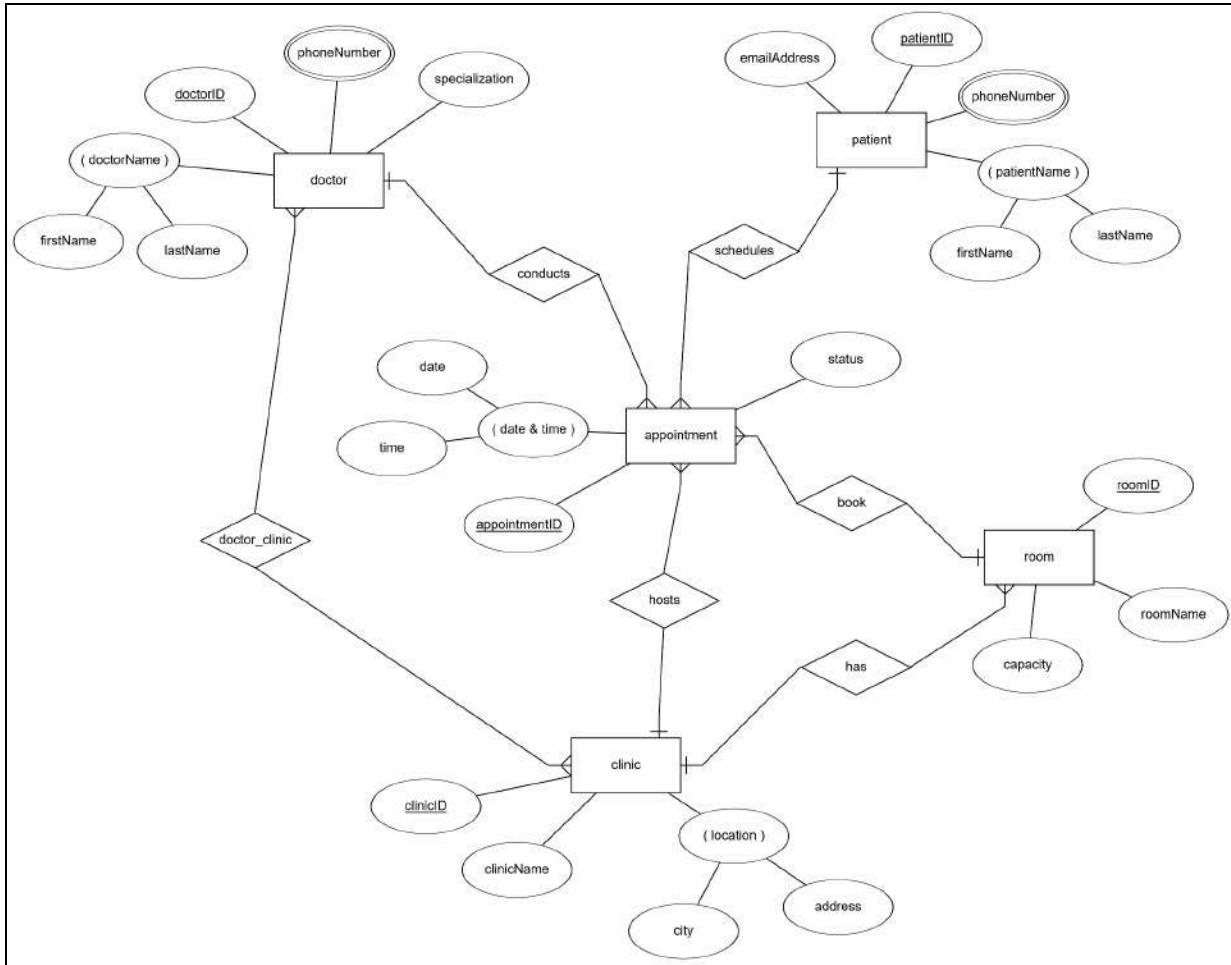
Sistem ini dirancang untuk tiga jenis pengguna utama:

1. Administrator: Mengelola data master (dokter, pasien, klinik, ruangan) dan memantau keseluruhan jadwal.
2. Dokter: Melihat jadwal praktik pribadi dan memperbarui status janji temu pasien.
3. Pasien: Melakukan pendaftaran janji temu secara mandiri dan melihat riwayat kunjungan.

2. PERANCANGAN BASIS DATA (DATABASE DESIGN)

2.1 Entity Relationship Diagram (ERD)

Berdasarkan analisis kebutuhan sistem, telah dirancang sebuah ERD yang mencakup entitas utama dan hubungannya.



Berikut adalah deskripsi elemen-elemen ERD tersebut:

- Entitas Utama:
 - Doctor: Menyimpan informasi tenaga medis.
 - Patient: Menyimpan data demografis pasien.
 - Clinic: Merepresentasikan lokasi fisik klinik.
 - Room: Merepresentasikan ruangan di dalam klinik tertentu.
 - Appointment: Entitas transaksional yang menghubungkan Pasien, Dokter, dan

Ruangan pada waktu tertentu.

- Atribut Multivalue:
 - Doctor_PhoneNumber dan Patient_PhoneNumber dipisahkan untuk memungkinkan satu entitas memiliki lebih dari satu nomor telepon.
- Relasi:
 - Seorang dokter dapat bekerja di banyak klinik (*Many-to-Many* melalui tabel asosiasi doctor_clinic).
 - Satu klinik memiliki banyak ruangan (*One-to-Many*).
 - Satu janji temu melibatkan satu pasien, satu dokter, dan satu ruangan (*Relationships* ke tabel Appointment).

2.2 Normalisasi Data

Untuk memastikan efisiensi penyimpanan dan mencegah anomali data, perancangan tabel telah melalui proses normalisasi hingga Bentuk Normal Ketiga (3NF). Langkah - langkah yang dilakukan untuk menormalisasikan database hingga 3NF adalah sebagai berikut:

a. Cek 1NF (First Normal Form):

- Aturan: Setiap sel tabel harus berisi nilai tunggal (atomik), dan tidak boleh ada grup berulang.
- Analisis: telah berhasil memenuhi 1NF.
 - Atribut phoneNumber yang multivalued telah dipisah menjadi tabel sendiri: doctor_phoneNumber dan patient_phoneNumber.
 - Atribut komposit (seperti doctorName dari ERD) telah dipecah menjadi firstName dan lastName.
 - Semua kolom lain (seperti date dan time yang dipisah) bersifat atomik.

b. Cek 2NF (Second Normal Form):

- Aturan: Harus dalam 1NF DAN tidak boleh ada partial dependency (ketergantungan parsial).
- Penjelasan: Partial dependency hanya terjadi jika Anda memiliki composite primary key (PK dengan banyak kolom). Ini berarti sebuah atribut non-key hanya bergantung pada sebagian dari PK tersebut, bukan seluruhnya.
- Analisis: telah berhasil memenuhi 2NF.
- Tabel dengan PK tunggal (doctor, patient, clinic, room, appointment) otomatis lolos 2NF.

- Kita perlu memeriksa tabel dengan PK komposit:
 - doctor_clinic (doctorID, clinicID): Tidak ada atribut non-key. Lolos 2NF.
 - doctor_phoneNumber (phoneNumber, doctorID): Tidak ada atribut nonkey. Lolos 2NF.
 - patient_phoneNumber (phoneNumber, patientID): Tidak ada atribut non-key. Lolos 2NF.

c. Cek 3NF (Third Normal Form):

- Aturan: Harus dalam 2NF DAN tidak boleh ada transitive dependency (ketergantungan transitif).
- Penjelasan: Ketergantungan transitif terjadi ketika atribut non-key (A) bergantung pada atribut non-key lainnya (B), yang bergantung pada Primary Key (PK).
 - Bentuknya: PK \rightarrow Atribut_NonKey_B \rightarrow Atribut_NonKey_A
- Analisis: doctor, patient, clinic: Lolos 3NF. Semua atribut (seperti firstName, city, address) bergantung langsung pada PK-nya (doctorID, clinicID), bukan pada atribut non-key lain.
 - room: Lolos 3NF. Atribut roomName dan capacity bergantung pada roomID. clinicID (FK) juga bergantung pada roomID. Tidak ada dependensi transitif.
 - doctor_clinic: Lolos 3NF. (Tidak ada atribut non-key).
 - doctor_phoneNumber: Lolos 3NF. (Tidak ada atribut non-key).
 - patient_phoneNumber: Lolos 3NF. (Tidak ada atribut non-key).
 - appointment: Melanggar 3NF.
- Penjelasan Pelanggaran 3NF pada Tabel appointment
 - Primary Key: appointmentID
 - Atribut Non-Key (termasuk FK): date, time, status, clinicID, patientID, doctorID, roomID Perhatikan dependensi (ketergantungan) berikut:
 1. appointmentID menentukan roomID (Benar. Janji temu ini ada di ruangan mana).
 2. roomID menentukan clinicID (Benar. Dilihat dari tabel room, setiap roomID pasti berlokasi di satu clinicID).

Ini menciptakan dependensi transitif: appointmentID \rightarrow roomID \rightarrow

clinicID

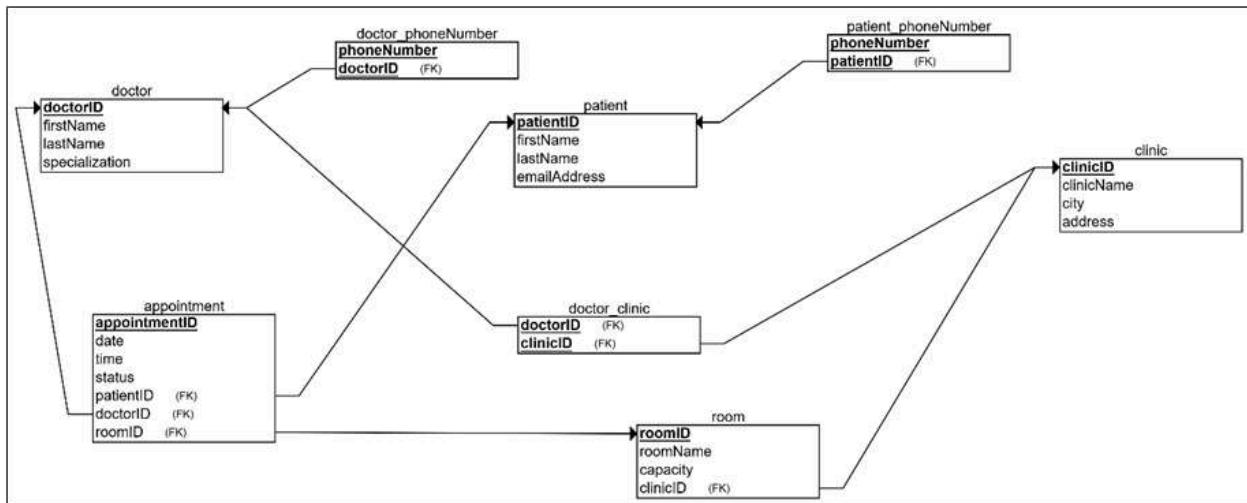
- Masalahnya: memiliki clinicID, sebuah atribut non-key, yang bergantung pada roomID, atribut non-key lainnya. Ini menyebabkan redundansi data.
 - Contoh: Jika ada 100 janji temu di roomID = 101, akan disimpan clinicID = "KLINIK-A" sebanyak 100 kali di 5 dalam tabel appointment, padahal informasi itu sudah ada di tabel room.
- Solusi untuk Mencapai 3NF

Untuk memperbaiki ini, harus dihapus kolom yang menyebabkan dependensi transitif dari tabel appointment. Hapus clinicID dari tabel appointment. Mengapa ini berhasil?

- Informasi clinicID tidak hilang.
- Jika perlu tahu di klinik mana sebuah janji temu diadakan, dapat ditemukan melalui roomID.

3. IMPLEMENTASI BASIS DATA (DATABASE IMPLEMENTATION)

Sistem basis data diimplementasikan menggunakan SQLite. Berikut adalah skema relasional final beserta *constraints* yang diterapkan.



3.1 Skema Relasional (DDL)

- Tabel Doctor

SQL

```
CREATE TABLE doctor (
    doctorID INT NOT NULL PRIMARY KEY,
    firstName VARCHAR(100) NOT NULL,
    lastName VARCHAR(100) NOT NULL,
    specialization VARCHAR(100) NOT NULL
);
```

- Tabel Appointment (Tabel Transaksi Utama)

Tabel ini menerapkan Foreign Key constraints untuk menjaga integritas referensial antar entitas.

SQL

```
CREATE TABLE appointment
(
    appointmentID INT NOT NULL,
    date DATE NOT NULL,
    time TIME NOT NULL,
    status VARCHAR(50) NOT NULL,
    patientID INT NOT NULL,
    doctorID INT NOT NULL,
    roomID INT NOT NULL,
    PRIMARY KEY (appointmentID),
    FOREIGN KEY (patientID) REFERENCES patient(patientID),
    FOREIGN KEY (doctorID) REFERENCES doctor(doctorID),
    FOREIGN KEY (roomID) REFERENCES room(roomID)
);
```

3.2 Contoh Manipulasi Data (DML)

Berikut adalah contoh operasi data yang digunakan dalam aplikasi:

Menambah Jadwal Baru (Create and Insert):

SQL

```
INSERT INTO appointment (date, time, status, patientID, doctorID, roomID)
VALUES ('2025-11-20', '10:00:00', 'Scheduled', 101, 1, 12);
```

- Mengambil Data Jadwal Dokter (Select with Join):

SQL

```
SELECT a.appointmentID, a.date, a.time, p.firstName, r.roomName  
FROM appointment a  
JOIN patient p  
ON a.patientID = p.patientID  
JOIN room r  
ON a.roomID = r.roomID  
WHERE a.doctorID = 1;
```

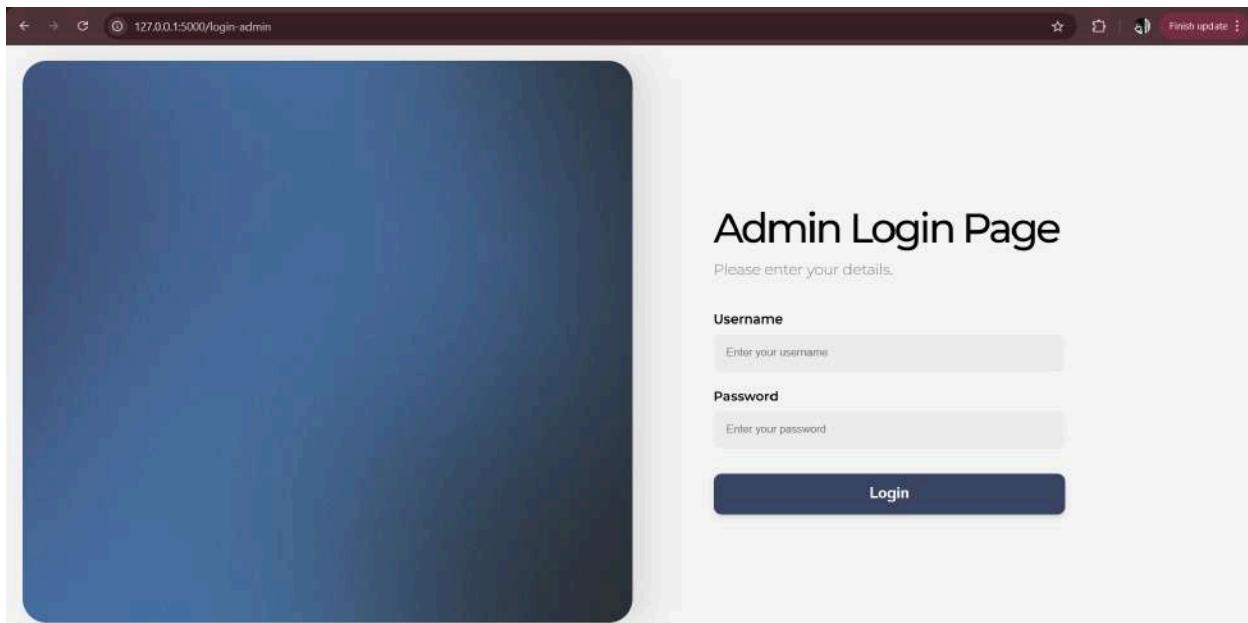
4. IMPLEMENTASI APLIKASI (APPLICATION IMPLEMENTATION)

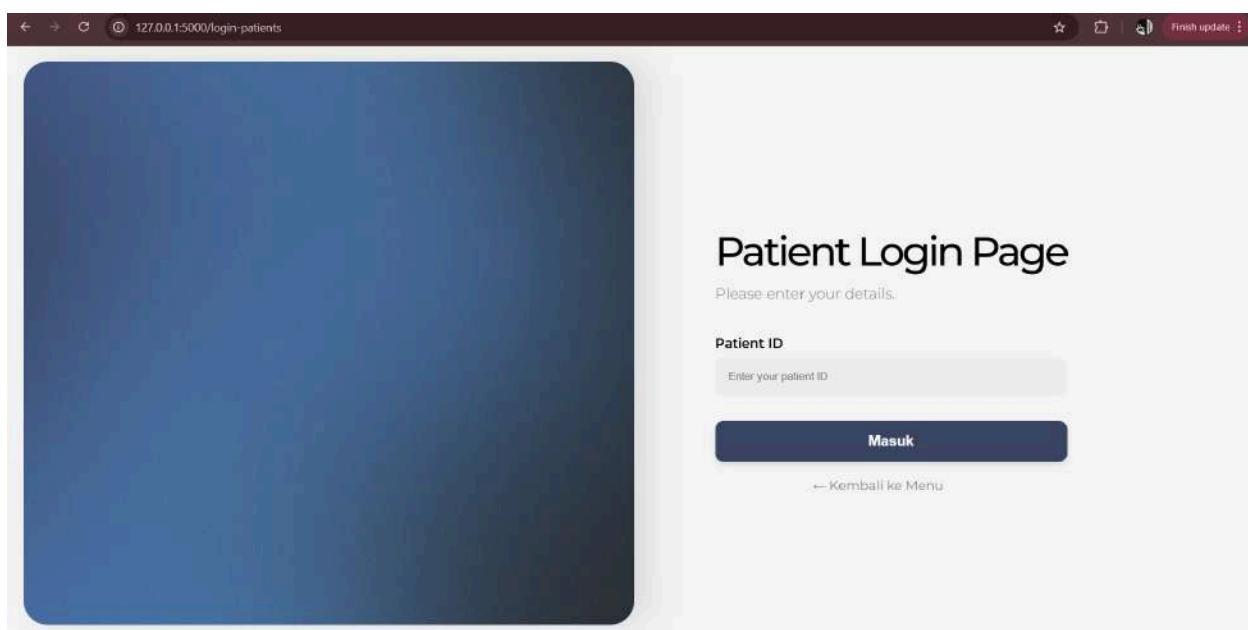
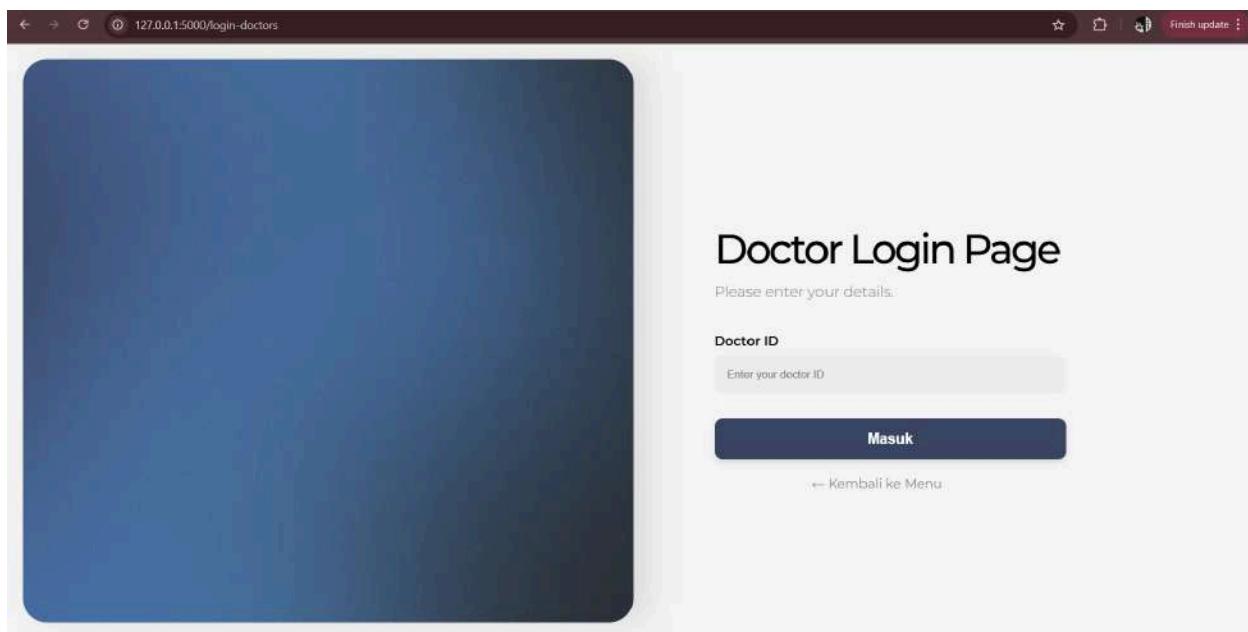
Aplikasi dibangun menggunakan bahasa pemrograman Python dengan framework Flask sebagai *backend* dan SQLite sebagai media penyimpanan data. Antarmuka pengguna (*Frontend*) dibangun menggunakan HTML dan CSS.

4.1 Struktur Modul Aplikasi

Aplikasi dibagi menjadi beberapa modul rute (*routes*) berdasarkan peran pengguna:

1. Modul Autentikasi (/login-admin, /login-doctors, /login-patients):
 - o Memisahkan pintu masuk bagi Admin, Dokter, dan Pasien.
 - o Verifikasi kredensial sederhana (Admin menggunakan *hardcoded password*, Dokter/Pasien menggunakan ID).





2. Modul Administrator (/admin):
 - o Fungsi: Dashboard pusat untuk CRUD (Create, Read, Update, Delete) seluruh entitas.
 - o Fitur: Menambah dokter baru, mendaftarkan klinik, dan mengelola seluruh jadwal janji temu.

Main Admin Page

Doctor's Table

Add a new doctor

<input type="text" value="Input doctor ID..."/>	<input type="text" value="Input first name..."/>	<input type="text" value="Input last name..."/>	<input type="text" value="Input specialization..."/>	<input type="button" value="ADD"/>
---	--	---	--	------------------------------------

ID	First Name	Last Name	Specialization	Action
1	celsi	miera	pip	<input type="button" value="Edit"/> <input type="button" value="HAPUS"/>

Patient's Table

Add a new patient

<input type="text" value="Input patient ID..."/>	<input type="text" value="Input first name..."/>	<input type="text" value="Input last name..."/>	<input type="text" value="Input email address..."/>	<input type="button" value="ADD"/>
--	--	---	---	------------------------------------

ID	First Name	Last Name	Email Address	Action
1	celsi	miera	blablabla@gmail.com	<input type="button" value="Edit"/> <input type="button" value="HAPUS"/>

Clinic's Table

Add a new clinic

<input type="text" value="Input clinic name..."/>	<input type="text" value="Input city..."/>	<input type="text" value="Input address..."/>	<input type="button" value="ADD"/>
---	--	---	------------------------------------

ID	Name	City	Address	Action
1	rparamita	jogja	pgung	<input type="button" value="Edit"/> <input type="button" value="HAPUS"/>
2	hermina	yogya	sleman	<input type="button" value="Edit"/> <input type="button" value="HAPUS"/>

Room's Table

Add a new room

<input type="text" value="Input room ID..."/>	<input type="text" value="Input room name..."/>	<input type="text" value="Input room capacity..."/>	-- Choose a clinic --	<input type="button" value="ADD"/>
---	---	---	-----------------------	------------------------------------

The screenshot shows two main sections of the application:

- Room's Table:** A section for managing rooms. It includes a form to "Add a new room" with fields for room ID, name, capacity, and clinic selection, followed by a table listing one room entry.
- Appointment's Table:** A section for managing appointments. It includes a form to "Book a new appointment" with fields for date, time, patient, doctor, room, and status, followed by a table listing two scheduled appointments.

ID	Name	Capacity	In Clinic	Action
1	ruang operasi	10	rparamita	<button>Edit</button> <button>HAPUS</button>

Date	Time	Patient	Doctor	Room	Status	Action
2025-11-28	02:35	celsi	clesi	ruang operasi	Scheduled	<button>Edit</button> <button>HAPUS</button>
2025-11-27	02:01	celsi	clesi	ruang operasi	Scheduled	<button>Edit</button> <button>HAPUS</button>

3. Modul Pasien (/patients/<id>):

- Fungsi: Memungkinkan pasien melihat jadwal yang tersedia dan melakukan booking.
- Fitur: Formulir "Book a new appointment" yang terhubung langsung dengan ketersediaan dokter dan ruangan.

The screenshot shows the "Halo, Patient celsi" page, which is part of the patient module. It displays a booking form and a list of scheduled appointments.

Booking Form:

- Date: mm/dd/yyyy
- Time: 02:35
- Patient: celsi miera
- Doctor: clesi miera
- Room: ruang operasi
- Status: Scheduled
- Action: BOOK NOW

Scheduled Appointments:

Date	Time	Doctor	Room	Status	Action
2025-11-28	02:35	clesi miera	ruang operasi	Scheduled	<button>Edit</button> <button>HAPUS</button>
2025-11-27	02:01	clesi miera	ruang operasi	Scheduled	<button>Edit</button> <button>HAPUS</button>

4. Modul Dokter (/doctors/<id>):
- Fungsi: Dashboard operasional dokter.
 - Fitur: Melihat daftar pasien hari ini dan mengubah status janji temu (misal: menjadi "Selesai").

The screenshot shows a dark-themed dashboard for a doctor. At the top left is the text 'Klinik Kelompok 3'. At the top right is a 'Logout' link. In the center, it says 'Halo, Dr. clesi'. Below this is a table with the following data:

Tanggal	Jam	Nama Pasien	Ruangan	Status	Aksi
2025-11-28	02:35	celsi miera	ruang operasi	Scheduled	Selesai
2025-11-27	02:01	celsi miera	ruang operasi	Scheduled	Selesai

4.2 Antarmuka Pengguna (User Interface)

- Halaman Utama (Landing Page): Menyajikan informasi layanan klinik dan navigasi menuju halaman login.
(Placeholder: Screenshot Halaman Utama - Menampilkan menu navigasi dan banner klinik)
- Dashboard Admin: Menampilkan tabel data Dokter, Pasien, Klinik, dan Appointment dalam satu halaman terintegrasi dengan tombol aksi Edit/Hapus.
(Placeholder: Screenshot Admin Dashboard - Menampilkan tabel CRUD)
- Dashboard Pasien: Menampilkan riwayat janji temu pribadi dan formulir pendaftaran baru.
(Placeholder: Screenshot Dashboard Pasien - Menampilkan form booking)

5. PENGUJIAN DAN HASIL (TESTING AND RESULTS)

5.1 Skenario Pengujian

1. Homepage



2. Login

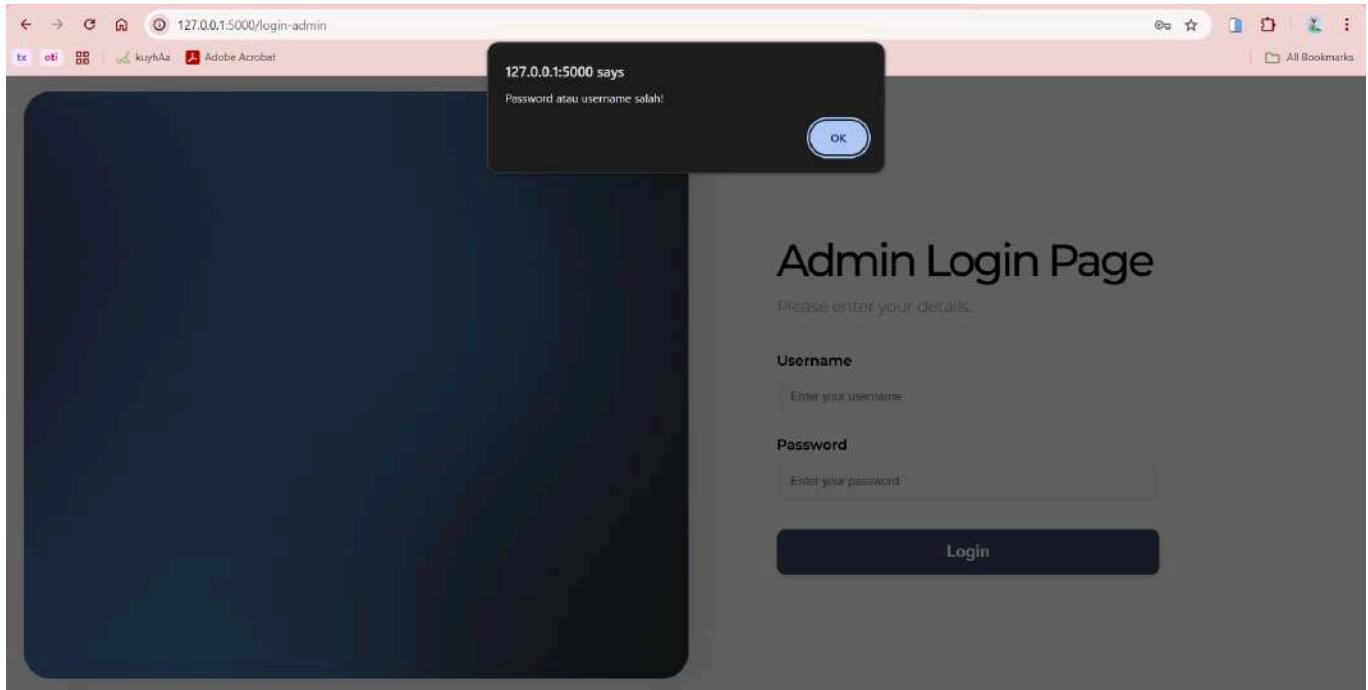
- Login Admin

Sistem menerapkan mekanisme keamanan di mana akses ke antarmuka administrator dibatasi hanya untuk pengguna yang memiliki kredensial valid. Untuk memverifikasi fungsionalitas ini, pengujian dapat dilakukan menggunakan data berikut:

- Username: admin
- Password: admin123

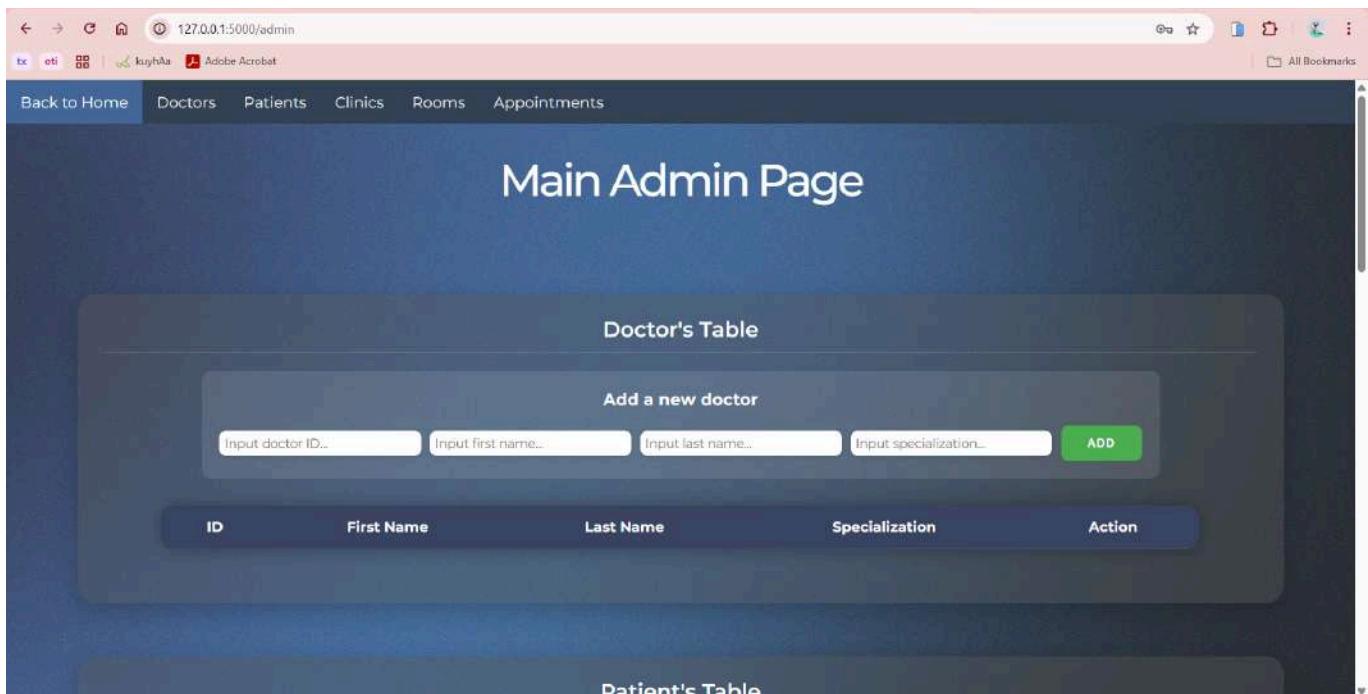
► Kasus 1:

Popup muncul saat user memasukkan password atau username yang salah.



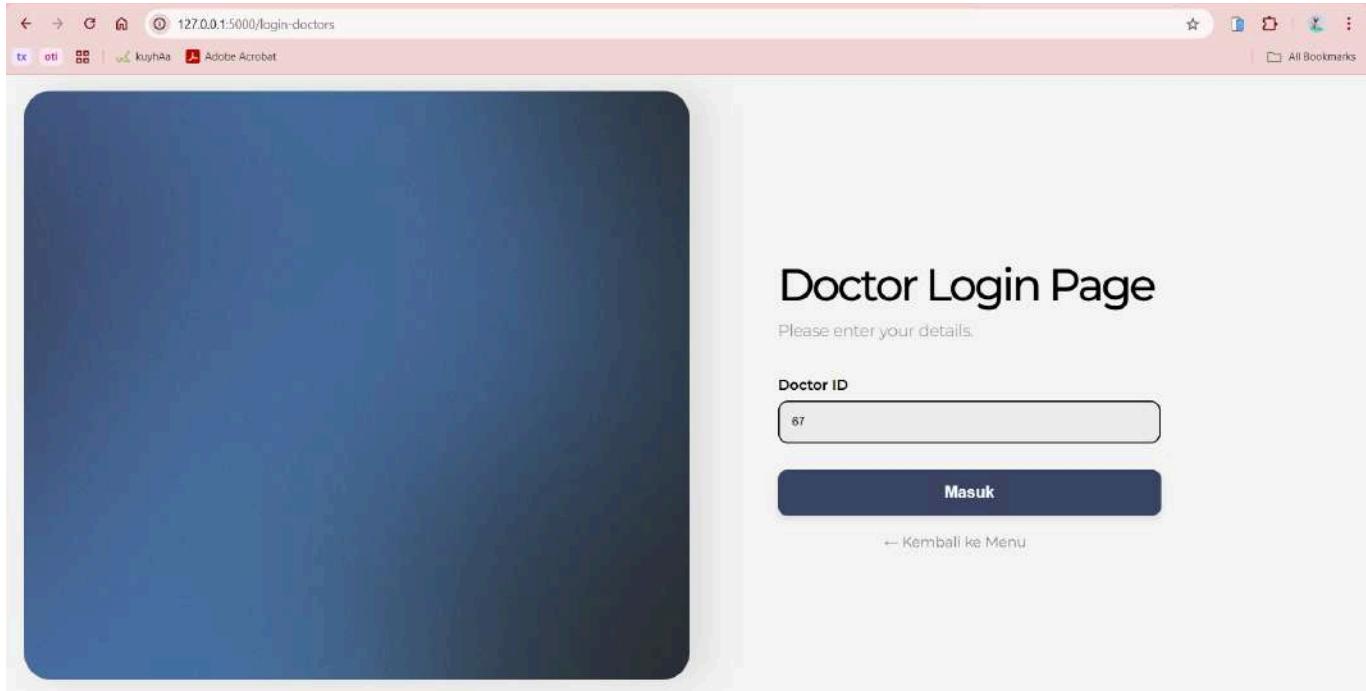
► Kasus 2:

User dapat masuk ke Page Admin jika memasukkan password dan username yang valid.

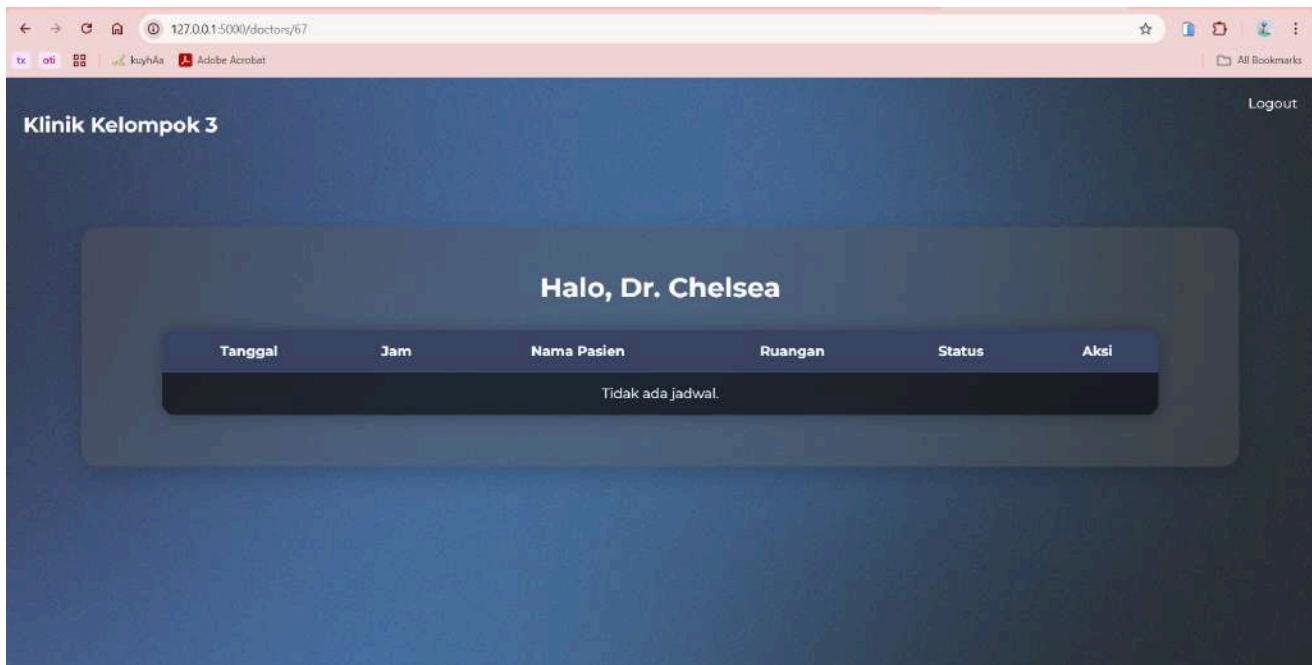


- Login Doctor

Untuk masuk ke dalam sistem, Dokter cukup memasukkan ID Dokter. Sistem akan mencocokkan ID yang diinput dengan data yang ada di database. Login hanya akan berhasil jika ID tersebut sudah didaftarkan sebelumnya oleh Admin.



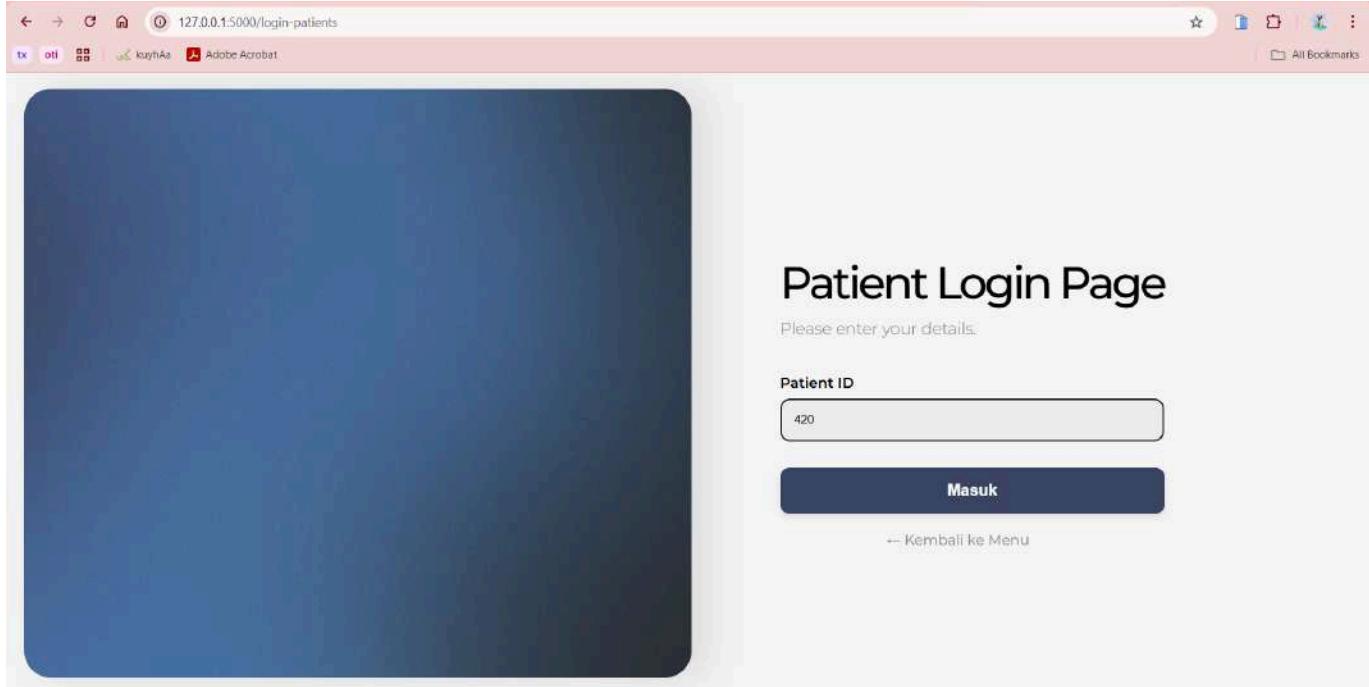
The screenshot shows a web browser window with the URL 127.0.0.1:5000/login-doctors. The page has a dark blue header and a light gray footer. The main content area is white and features a large dark blue rounded rectangle on the left. On the right, the text "Doctor Login Page" is displayed in a large, bold, black font. Below it, a smaller text says "Please enter your details.". There is a form field labeled "Doctor ID" containing the number "67". A dark blue button labeled "Masuk" is positioned below the input field. At the bottom of the page, there is a link "← Kembali ke Menu".



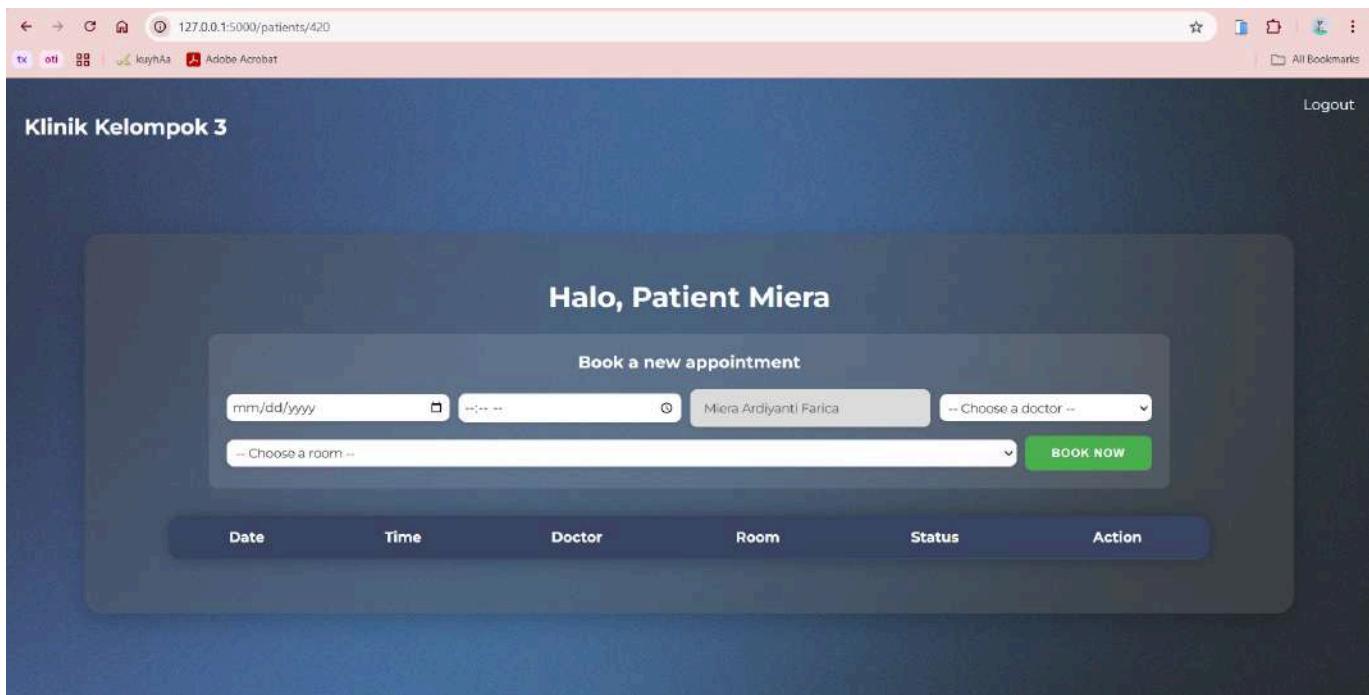
The screenshot shows a web browser window with the URL 127.0.0.1:5000/doctors/67. The header includes the text "Klinik Kelompok 3" and "Logout". The main content area has a dark blue background. At the top, the text "Halo, Dr. Chelsea" is displayed in a white font. Below it is a table with the following columns: Tanggal, Jam, Nama Pasien, Ruangan, Status, and Aksi. The table currently displays the message "Tidak ada jadwal." (No schedule). The overall design is clean and modern.

- Login Patient

Sama seperti Doctor, untuk masuk ke dalam sistem, Patient cukup memasukkan ID Patient. Sistem akan mencocokkan ID yang diinput dengan data yang ada di database. Login juga hanya akan berhasil jika ID tersebut sudah didaftarkan sebelumnya oleh Admin.



The screenshot shows a web browser window titled "127.0.0.1:5000/login-patients". The page has a dark blue header and a white body. In the center, it says "Patient Login Page" and "Please enter your details.". Below this is a "Patient ID" input field containing "420", followed by a "Masuk" button. At the bottom right is a link "← Kembali ke Menu".



The screenshot shows a web browser window titled "127.0.0.1:5000/patients/420". The header includes "Klinik Kelompok 3" and "Logout". The main content area has a dark blue background. At the top, it says "Halo, Patient Miera". Below this is a "Book a new appointment" section with fields for Date (mm/dd/yyyy), Time, Doctor (Miera Ardiyanti Farica), Room, and Status, along with a "BOOK NOW" button. At the bottom, there is a table header with columns: Date, Time, Doctor, Room, Status, and Action.

3. Input Data

- Page Admin
- Table Doctor

The screenshot shows a web browser window with the URL `127.0.0.1:5000/admin`. The page title is "Main Admin Page". At the top, there is a navigation bar with links: "Back to Home", "Doctors", "Patients", "Clinics", "Rooms", and "Appointments". Below the navigation bar, there is a section titled "Doctor's Table" containing a form to "Add a new doctor" with fields for ID (67), First Name (Chelsea), Last Name (Natasja), and Specialization (Dentist). A green "ADD" button is present. Below the form is a table header row with columns: "ID", "First Name", "Last Name", "Specialization", and "Action".

This screenshot is identical to the one above, showing the "Doctor's Table" section. However, the table now contains a single row of data: ID 67, First Name Chelsea, Last Name Natasja, Specialization Dentist, and Action buttons (Edit and HAPUS).

○ Table Patient

The screenshot shows a web application interface for managing patients. At the top, there is a navigation bar with links for 'Back to Home', 'Doctors', 'Patients', 'Clinics', 'Rooms', and 'Appointments'. Below the navigation bar, a header displays the patient's ID (67), name (Chelsea), and role (Dentist). There are 'Edit' and 'HAPUS' buttons in the header.

The main area is titled 'Patient's Table' and contains a sub-section titled 'Add a new patient'. It features four input fields: 'ID' (420), 'First Name' (Miera), 'Last Name' (Ardiyanti Farica), and 'Email Address' (mieraardiyantifarica@gmail.com). A green 'ADD' button is located to the right of these fields.

Below the 'Add a new patient' section is a table header with columns: 'ID', 'First Name', 'Last Name', 'Email Address', and 'Action'. A single row of data is shown, corresponding to the patient details entered in the 'Add a new patient' form.

At the bottom of the page, there is another section titled 'Clinic's Table'.

This screenshot shows the same web application after a new patient has been added. The 'Patient's Table' section now displays a list of patients. The first patient listed is Miera Ardiyanti Farica, with ID 420, email mieraardiyantifarica@gmail.com, and the 'Action' column showing 'Edit' and 'HAPUS' buttons.

The 'Add a new patient' form is still present at the top of the 'Patient's Table' section, with the same input fields and 'ADD' button.

At the bottom of the page, there is another section titled 'Clinic's Table'.

○ Table Clinic

The screenshot shows a web-based administration interface for a medical clinic. At the top, there is a navigation bar with links for 'Back to Home', 'Doctors', 'Patients', 'Clinics', 'Rooms', and 'Appointments'. Below the navigation bar, the main content area has a title 'Clinic's Table'. A sub-section titled 'Add a new clinic' contains three input fields: 'Name' (Pramita), 'City' (Yogyakarta), and 'Address' (Jl. Cik Di Tiro No.17, Terban, Kec. Gondokusuman). A green 'ADD' button is located to the right of the address field. Below this is a table with columns: ID, Name, City, Address, and Action. One row is visible, showing ID 5, Name Pramita, City Yogyakarta, and Address Jl. Cik Di Tiro No.17, Terban, Kec. Gondokusuman, Kota Yogyakarta. To the right of the address are 'Edit' and 'HAPUS' buttons. At the bottom of the page, there is another section titled 'Room's Table'.

This screenshot shows the same web application after a new clinic has been added. The 'Clinic's Table' section now displays a single row in the table, corresponding to the data entered in the previous screenshot. The table columns are ID, Name, City, Address, and Action. The row shows ID 5, Name Pramita, City Yogyakarta, and Address Jl. Cik Di Tiro No.17, Terban, Kec. Gondokusuman, Kota Yogyakarta. The 'Edit' and 'HAPUS' buttons are also present. The 'Room's Table' section at the bottom of the page is still visible.

○ Table Room

The screenshot shows a web application interface for managing rooms. At the top, there is a navigation bar with links: Back to Home, Doctors, Patients, Clinics, Rooms, and Appointments. The main content area has a title "Room's Table". Below it, there is a form titled "Add a new room" with fields for ID (13), Name (Ruang Operasi), Capacity (8), and In Clinic (dropdown menu showing "Choose a clinic" and "Pramita"). A green "ADD" button is located to the right of the form. Below the form is a table with columns: ID, Name, Capacity, In Clinic, and Action. One row is visible, showing ID 13, Name Ruang Operasi, Capacity 8, In Clinic Pramita, and Action buttons for Edit and HAPUS.

This screenshot is identical to the one above, showing the "Room's Table" section of the web application. It displays the same form for adding a new room and the same table listing an existing room (ID 13, Name Ruang Operasi, Capacity 8, In Clinic Pramita). The "Edit" and "HAPUS" buttons are present in the Action column.

○ Table Appointment

The screenshot shows the 'Appointment's Table' section of a web application. At the top, there is a header with navigation links: 'Back to Home', 'Doctors', 'Patients', 'Clinics', 'Rooms', and 'Appointments'. Below the header, there is a row of cards with the following details:

13	Ruang Operasi	8	Pramita	<button>Edit</button>	<button>HAPUS</button>
----	---------------	---	---------	-----------------------	------------------------

In the center, there is a 'Book a new appointment' form with the following fields:

- Date: 12/20/2025
- Time: 06:30 AM
- Patient: Miera Ardiyanti Farica
- Doctor: Dr. Chelsea (Dentist)
- Room: Ruang Operasi
- Action: ADD

Below the form is a table titled 'Appointment's Table' with the following columns: Date, Time, Patient, Doctor, Room, Status, and Action. A single row is shown:

Date	Time	Patient	Doctor	Room	Status	Action
2025-12-20	06.30	Miera	Chelsea	Ruang Operasi	Scheduled	<button>Edit</button> <button>HAPUS</button>

The screenshot shows the 'Appointment's Table' section of a web application. At the top, there is a header with navigation links: 'Back to Home', 'Doctors', 'Patients', 'Clinics', 'Rooms', and 'Appointments'. Below the header, there is a row of cards with the following details:

13	Ruang Operasi	8	Pramita	<button>Edit</button>	<button>HAPUS</button>
----	---------------	---	---------	-----------------------	------------------------

In the center, there is a 'Book a new appointment' form with the following fields:

- Date: mm/dd/yyyy
- Time: --:--
- Patient: -- Choose a patient --
- Doctor: -- Choose a doctor --
- Room: -- Choose a room --
- Action: ADD

Below the form is a table titled 'Appointment's Table' with the following columns: Date, Time, Patient, Doctor, Room, Status, and Action. A single row is shown:

Date	Time	Patient	Doctor	Room	Status	Action
2025-12-20	06.30	Miera	Chelsea	Ruang Operasi	Scheduled	<button>Edit</button> <button>HAPUS</button>

○ Implementasi Edit dan Delete

Sistem menyediakan fasilitas bagi Admin untuk mengubah atau menghapus entitas data yang sudah terdaftar pada database.

The screenshot shows a web browser window with the URL `127.0.0.1:5000/admin?type=doctor&id=67`. The page title is "Main Admin Page". At the top, there is a navigation bar with links: "Back to Home", "Doctors", "Patients", "Clinics", "Rooms", and "Appointments". Below the navigation bar, the main content area has a title "Doctor's Table". A sub-section titled "Add a new doctor" contains four input fields: "Input doctor ID...", "Input first name...", "Input last name...", and "Input specialization...". A green "ADD" button is located to the right of these fields. Below this is a table with the following data:

ID	First Name	Last Name	Specialization	Action
67	Chelsea	Natasja Jesslyne	Dentist	<button>Simpan</button> <button>Batal</button>

This screenshot is similar to the previous one, showing the same web browser interface and "Main Admin Page" title. The "Doctors" section is active in the navigation bar. The "Doctor's Table" section shows the same table data as before, but the "Action" column for the row with ID 67 now contains two buttons: "Edit" and "HAPUS" (Delete). The "Edit" button is green, while the "HAPUS" button is white with a red border.

The screenshot shows the main admin interface with a dark blue header bar. The header includes navigation links: Back to Home, Doctors, Patients, Clinics, Rooms, and Appointments. The title "Main Admin Page" is centered above two main sections.

Doctor's Table

Add a new doctor

ID	First Name	Last Name	Specialization	Action

Patient's Table

Tindakan ini secara otomatis akan menghapus seluruh data pada tabel **Appointment** yang terkait dengan Dokter Chelsea.

The screenshot shows the admin interface with a dark blue header bar. The header includes navigation links: Back to Home, Doctors, Patients, Clinics, Rooms, and Appointments. A specific appointment record is displayed at the top:

13	Ruang Operasi	8	Pramita	<button>Edit</button> <button>HAPUS</button>
----	---------------	---	---------	--

Appointment's Table

Book a new appointment

Date	Time	Patient	Doctor	Room	Status	Action

- Page Doctor

Dokter dapat melihat jadwal temu (*appointment*) pribadi serta melakukan *logout* melalui halaman *dashboard* dokter.

Klinik Kelompok 3

Logout

Halo, Dr. Chelsea

Tanggal	Jam	Nama Pasien	Ruangan	Status	Aksi
2025-12-20	06:30	Miera Ardiyanti Farica	Ruang Operasi	Scheduled	Selesai

- Page Patient

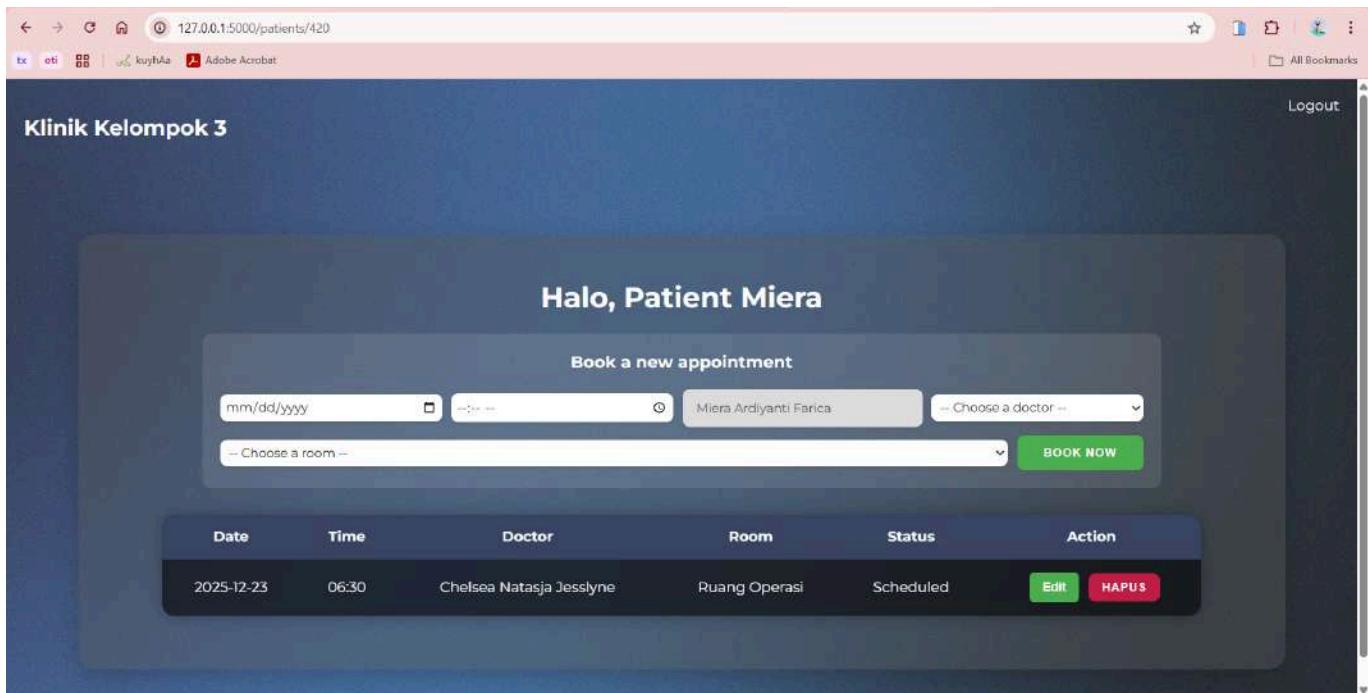
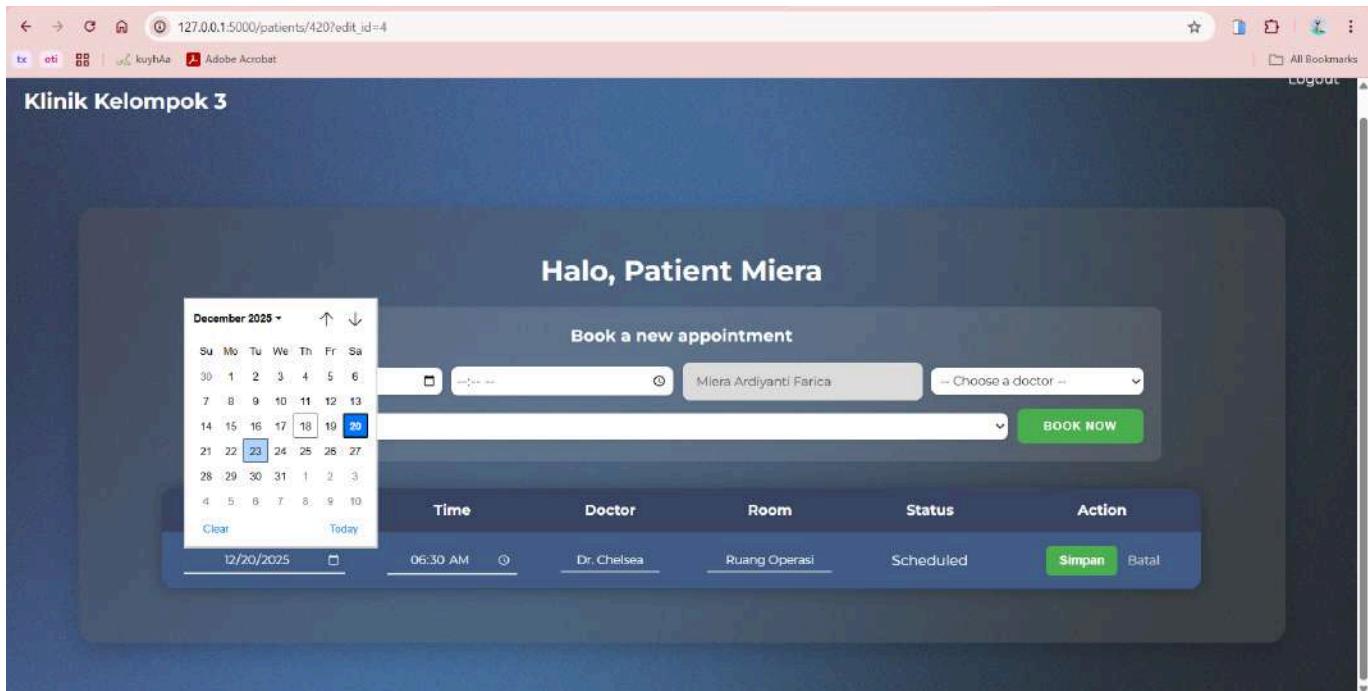
Sistem memfasilitasi Pasien untuk mengajukan jadwal temu baru, melihat riwayat jadwal pribadi, serta melakukan pembaruan atau pembatalan terhadap jadwal yang ada. Selain itu, tersedia pula opsi *logout* untuk mengakhiri sesi.

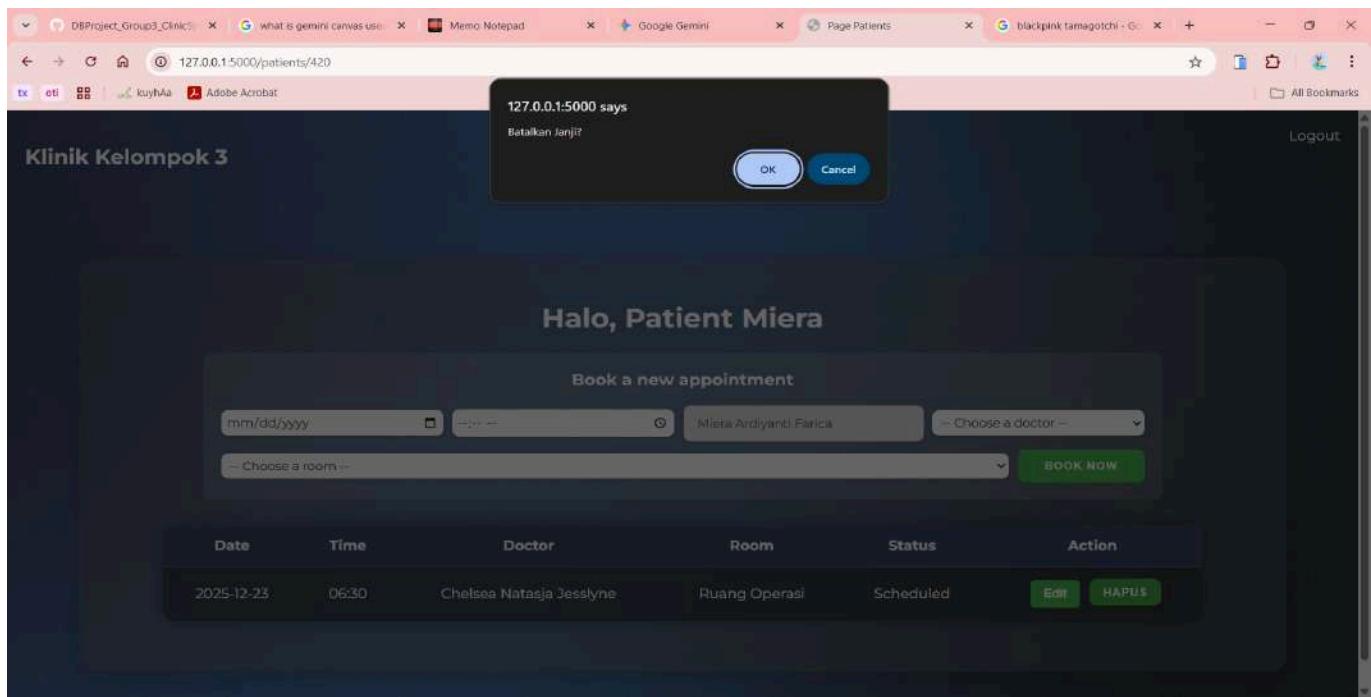
The screenshot shows a web browser window for 'Klinik Kelompok 3' at the URL 127.0.0.1:5000/patients/420. The page title is 'Halo, Patient Miera'. A central modal window titled 'Book a new appointment' contains fields for Date (mm/dd/yyyy), Time, Doctor (Miera Ardiyanti Farica), Room (dropdown placeholder 'Choose a room'), and a 'BOOK NOW' button. Below this, a table lists a single appointment: Date 2025-12-20, Time 06:30, Doctor Chelsea Natasja Jesslyne, Room Ruang Operasi, Status Scheduled, with 'Edit' and 'HAPUS' buttons. The top right corner of the main page has a 'Logout' link.

Date	Time	Doctor	Room	Status	Action
2025-12-20	06:30	Chelsea Natasja Jesslyne	Ruang Operasi	Scheduled	<button>Edit</button> <button>HAPUS</button>

○ Fitur Edit dan Delete Appointment

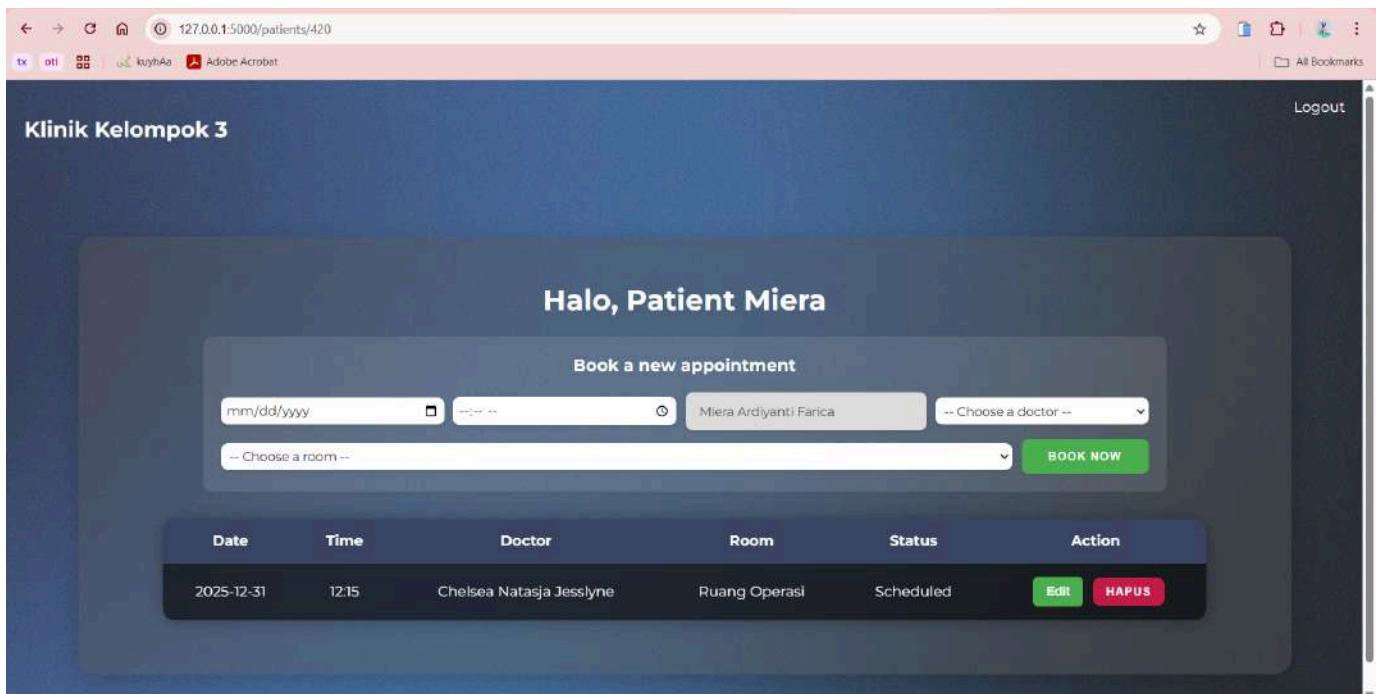
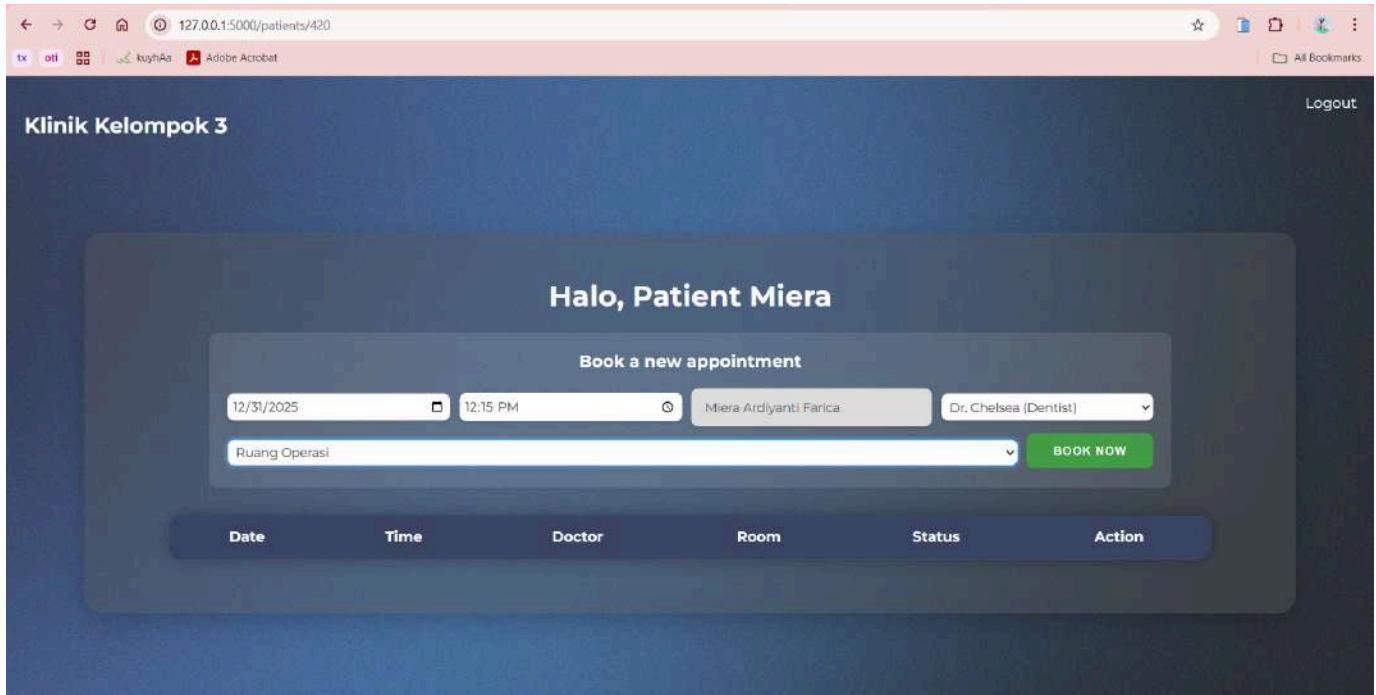
Karena sistem menggunakan basis data yang terintegrasi, setiap jadwal temu yang diedit atau dibatalkan oleh Pasien akan secara otomatis diperbarui pada daftar jadwal di halaman Dokter maupun Admin.





○ Fitur Add Appointment

Sama halnya dengan pembaruan atau penghapusan data, jadwal temu yang baru didaftarkan oleh Pasien akan secara otomatis muncul pada daftar jadwal di halaman Dokter dan Administrator.



► Page Admin

The screenshot shows a web-based administration tool for managing appointments. At the top, there is a navigation bar with links for 'Back to Home', 'Doctors', 'Patients', 'Clinics', 'Rooms', and 'Appointments'. The main content area is titled 'Appointment's Table' and contains a form for 'Booking a new appointment'. The form includes fields for Date (mm/dd/yyyy), Time, Patient (dropdown menu showing 'Choose a patient'), Doctor (dropdown menu showing 'Choose a doctor'), Room (dropdown menu showing 'Choose a room'), and an 'ADD' button. Below the form is a table listing a single appointment entry:

Date	Time	Patient	Doctor	Room	Status	Action
2025-12-31	12:15	Miera	Chelsea	Ruang Operasi	Scheduled	<button>Edit</button> <button>HAPUS</button>

► Page Doctor

The screenshot shows a web-based application for doctors. At the top, there is a header with the text 'Klinik Kelompok 3' and a 'Logout' link. The main content area displays a greeting message 'Halo, Dr. Chelsea' and a table showing the details of the appointment:

Tanggal	Jam	Nama Pasien	Ruangan	Status	Aksi
2025-12-31	12:15	Miera Ardiyanti Farica	Ruang Operasi	Scheduled	<button>Selesai</button>

6. KESIMPULAN DAN REFLEKSI (CONCLUSION AND REFLECTION)

6.1 Kesimpulan

Proyek *Clinic Appointment System* berhasil mengimplementasikan sistem informasi berbasis web yang mengintegrasikan manajemen data pasien, dokter, dan operasional klinik. Desain basis data yang telah dinormalisasi terbukti mampu menangani relasi kompleks seperti hubungan dokter dengan banyak klinik serta penjadwalan ruang. Implementasi menggunakan Flask memungkinkan interaksi yang responsif antara pengguna dan basis data.

6.2 Tantangan

Tantangan utama yang dihadapi selama pengembangan adalah:

1. Mapping Relasi SQL ke Objek Python: Memastikan data hasil query JOIN dapat ditampilkan dengan benar di template HTML.
2. Manajemen Constraint: Menangani error ketika data dihapus (misalnya menghapus ruangan yang memiliki jadwal aktif) memerlukan penanganan *Foreign Key* yang hati-hati.

6.3 Saran Pengembangan

Untuk pengembangan selanjutnya, disarankan:

1. Keamanan: Mengimplementasikan *hashing* password dan sesi login yang aman (bukan sekadar ID).
2. Validasi Bisnis: Menambahkan logika untuk mencegah bentrok jadwal (*conflict detection*) agar satu ruangan tidak bisa dipesan dua kali di waktu yang sama.
3. Rekam Medis: Mengintegrasikan fitur rekam medis pasien sesuai dengan standar RME.