# **KLINIK PRATAMA KHAZANAH MEDIKA**

**Studi Kasus: Klinik**

# 

**LAPORAN PROYEK AKHIR**

Dibuat untuk memenuhi syarat menyelesaikan matakuliah Analisis Berorientasi Objek, Basis Data, Pemrograman Berorientasi Objek dan Pemrograman Web

Oleh:

1. Muhammad Jibril Satya P (22416255201243)
2. Nurul Latifah (22416255201239)
3. Iis Mayanti (22416255201189)

**TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS BUANA PERJUANGAN KARAWANG**

**2023**

# **DAFTAR ISI**

**DAFTAR ISI** i

**DAFTAR GAMBAR** iii

**DAFTAR TABEL** v

**KATA PENGANTAR** vi

**BAB I PENDAHULUAN** 1

**1.1** **Latar Belakang** 1

**1.2** **Rumusan Masalah** 2

**1.3** **Tujuan** 2

**1.4** **Batasan Masalah** 2

**BAB II TINJAUAN PUSTAKA** 3

**2.1 Konsep Analisis Desain Berorientasi Objek** 3

**2.2 Prinsip Basis Data** 4

**2.3 Pemrograman Web dan Teknologi Terkait** 4

**2.4 Konsep Pemrograman Berorientasi Objek** 6

**BAB III ANALISIS SISTEM** 7

**3.1 Spesifikasi Persyaratan Sistem** 7

**3.1.1** **Analisis Sistem** 7

**3.1.2** **Scope** 8

**3.1.3** **User Characteristic** 8

**3.1.4** **Constraints** 8

**3.1.5** **Assumptions and Depedencies** 8

**3.1.6** **Specific Requirements** 9

**3.1.7** **Fungcionality** 9

**3.1.8** **Hardware Interface** 10

**3.2 Diagram Use Case** 10

**3.2.1 Use Case Diagram Login Pasien** 10

**3.2.2 Use Case Diagram Dashboard Pasien** 12

**3.2.3 Use Case Diagram Resepsionis** 14

**3.2.4 Use Case Diagram Pemilik Klinik** 16

**3.3 Diagram Kelas** 18

**3.4 Diagram Activity** 19

**3.4.1 Diagram Activity Pasien** 19

**3.4.2 Diagram Activity Resepsionis** 22

**3.4.3 Diagram Activity Pemilik Klinik** 24

**3.5 Desain Sequence** 25

**3.5.1 Sequence Diagram Pasien** 25

**3.5.2 Sequence Diagram Resepsionis** 28

**3.5.2 Sequence Diagram Pemilik Klinik** 29

**3.6 Desain Basis Data** 30

**3.7 Arsitektur Pemrograman Web** 31

**3.8 Implementasi Pemrograman Berorientasi Objek** 32

**BAB IV HASIL DAN PEMBAHASAN** 34

**4.1 Pengujian Fungsional** 34

**4.2 Pengujian Performa** 35

**4.3 Implementasi Sistem** 37

**4.4 Analisis Hasil Pengujian** 42

**BAB V KESIMPULAN** 43

**5.1 Kesimpulan** 43

**5.2 Saran** 43

**DAFTAR PUSTAKA** 44

**LAMPIRAN** 45

**1.** **Kode Program** 45

**2.** **Hasil Uji Coba** 45

**3.** **Timeline** 45

**4.** **Angket Penelitian** 45

# **DAFTAR GAMBAR**

Gambar 1 Use Case Diagram Login Pasien 10

Gambar 2 Use Case Diagram Dashboard Pasien 12

Gambar 3 Use Case Diagram Resepsionis 14

Gambar 4 Use Case Diagram Pemilik Klinik 16

Gambar 5 Diagram Kelas 18

Gambar 6 Diagram Activity Login Pasien 19

Gambar 7 Diagram Activity Buat Akun Pasien 19

Gambar 8 Diagram Activity Registrasi Pasien 20

Gambar 9 Diagram Activity Memilih Janji Temu Dengan Dokter 20

Gambar 10 Diagram Activity Melihat Resep Obat 21

Gambar 11 Diagram Activity Edit Profile Pasien 21

Gambar 12 Diagram Activity Membuat Janji Temu Dengan Dokter 22

Gambar 13 Diagram Activity Menginput Rekam Medis Pasien 23

Gambar 14 Diagram Activity Memberi Resep Obat Kepada Pasien 23

Gambar 15 Diagram Activity Mengedit Profile 24

Gambar 16 Diagram Activity Melihat Rekam Medis Pasien 24

Gambar 17 Sequence Diagram Login Pasien 25

Gambar 18 Sequence Diagram Buat Akun Pasien 25

Gambar 19 Sequence Diagram Registrasi Pasien 26

Gambar 20 Sequence Diagram Memilih Janji Temu Dengan Dokter 26

Gambar 21 Sequence Diagram Melihat Resep Obat 27

Gambar 22 Sequence Diagram Edit Profile Pasien 27

Gambar 23 Sequence Diagram Membuat Janji Temu Dengan Dokter 28

Gambar 24 Sequence Diagram Menginput Rekam Medis 28

Gambar 25 Sequence Diagram Memberi Resep Obat Kepada Pasien 29

Gambar 26 Sequence Diagram Mengedit Profile 29

Gambar 27 Sequence Diagram Melihat Rekam Medis Pasien 29

Gambar 28 Desain Basis Data 30

Gambar 29 Arsitektur Pemrograman Web 31

Gambar 30 Implementasi PBO 1 32

Gambar 31 Implementasi PBO 2 32

Gambar 32 Implementasi PBO 3 33

Gambar 33 Pengujian Performa 35

Gambar 34 Pengujian Performa 35

Gambar 35 Pengujian Performa 35

Gambar 36 Pengujian Performa 36

Gambar 37 Pengujian Performa 36

Gambar 38 Pengujian Performa 36

Gambar 39 Pengujian Performa 37

Gambar 40 Pengujian Performa 37

Gambar 41 Implementasi Sistem Buat Akun 37

Gambar 42 Implementasi Sitem Login Klinik 38

Gambar 43 Implementasi Sistem Reset Password 38

Gambar 44 Implementasi Sitem Dashboard User 39

Gambar 45 Implementasi Sistem Form Register User 39

Gambar 46 Implementasi Sistem Membuat Janji Temu User 39

Gambar 47 Implementasi Sistem Menerima Resep Obat User 40

Gambar 48 Implementasi Sistem Dashboard My Profile User 40

Gambar 49 Implementasi Sistem Dashboard Admin 40

Gambar 50 Implementasi Sistem Rekam Medis Admin 41

Gambar 51 Implementasi Sistem Resep Obat Admin 41

Gambar 52 Implementasi Sistem Janji Temu Admin 41

Gambar 53 Implementasi Sistem Dashboard My Profile Admin 42

# 

# **DAFTAR TABEL**

Tabel 1 Fungsi Sistem 9

Tabel 2 Use Case Deskripsi Login 11

Tabel 3 Use Case Deskripsi Buat Akun 11

Tabel 4 Use Case Deskripsi Registrasi 12

Tabel 5 Use Case Deskripsi Memilih Janji Temu Dengan Dokter 13

Tabel 6 Use Case Deskripsi Melihat Resep Obat 13

Tabel 7 Use Case Deskripsi Edit Profile 14

Tabel 8 Use Case Deskripsi Membuat Janji Temu Dengan Dokter 15

Tabel 9 Use Case Deskripsi Menginput Rekam Medis 15

Tabel 10 Use Case Deskripsi Memberi Resep Obat Kepada Pasien 16

Tabel 11 Use Case Deskripsi Mengedit Profile 17

Tabel 12 Use Case Deskripsi Melihat Rekam Medis Pasien 17

Tabel 13 Pengujian Fungsionalitas 33

# **KATA PENGANTAR**

Dengan menyebut nama Allah SWT yang Maha Pengasih lagi Maha Panyayang, kami panjatkan puja dan puji syukur atas kehadirat-Nya, yang telah melimpahkan rahmat, hidayah, dan inayah-Nya kepada kami, sehingga kami dapat menyelesaikan Laporan Proyek Akhir kami tentang “Klinik”.

Laporan ini telah kami susun dengan maksimal dan mendapatkan bantuan dari berbagai pihak sehingga dapat memperlancar pembuatan laporan ini. Untuk itu kami menyampaikan banyak terima kasih kepada semua pihak yang telah berpartisipasi dalam pembuatan laporan ini.

Terlepas dari semua itu, kami menyadari sepenuhnya bahwa masih ada kekurangan baik dari segi susunan kalimat maupun tata bahasanya. Oleh karena itu dengan segala kekurangan dalam laporan ini, kami terima segala saran dan kritik dari pembaca agar saya dapat memperbaiki laporan ini.

Akhir kata kami berharap semoga Laporan Proyek Akhir tentang “Klinik” ini dapat memberikan manfaat dan berguna bagi para pembaca juga kami sebagai penyusun.

Karawang, 28 November 2023

Penyusun

# **BAB I PENDAHULUAN**

## **Latar Belakang**

Dalam era globalisasi dan kemajuan teknologi informasi, perkembangan teknologi pada saat ini telah berkembang pesat. Transformasi digital telah menciptakan peluang baru dalam berbagai sektor, termasuk pelayanan kesehatan. Pelayanan kesehatan yang efisien dan berintegrasi menjadi salah satu kunci dalam meningkatkan kualitas hidup masyarakat.

Salah satu aspek yang mendapat sorotan adalah pengembangan sistem kesehatan yang memungkinkan pasien untuk mengakses layanan kesehatan secara online. Dengan meningkatnya kebutuhan akan kenyamanan dan aksesibilitas, proyek ini bertujuan untuk mengembangkan sebuah aplikasi klinik berbasis online yang memungkinkan pasien untuk melakukan pendaftaran dan pembuatan janji temu melalui aplikasi, serta menerima resep obat dalam bentuk digital.

Saat ini klinik-klinik medis memainkan peran yang penting sebagai penyedia layanan kesehatan primer. Pelayanan kesehatan konvensional seringkali dihadapkan pada tantangan, seperti antrian panjang, keterbatasan waktu, dan jarak geografis yang menjadi hambatan bagi beberapa individu untuk mendapatkan perawatan medis dengan tepat waktu. Dalam konteks ini, aplikasi klinik online diharapkan dapat menjadi solusi inovatif yang tidak hanya memudahkan akses pasien terhadap layanan kesehatan, tetapi juga meningkatkan efisiensi operasional klinik.

Pada umumnya, proses pendaftaran dan pembuatan janji temu dengan dokter seringkali memerlukan waktu yang cukup lama, dengan potensi antrian panjang dan kesulitan administratif. Dalam kaitannya dengan itu, penggunaan sistem klinik online diharapkan dapat mempermudah dan mempercepat proses pendaftaran, serta memberikan kemudahan dalam membuat janji temu dengan dokter.

Dengan mengidentifikasi kebutuhan dan peluang dalam pelayanan kesehatan berbasis online, proyek pembuatan aplikasi untuk klinik ini diharapkan dapat memberikan kontribusi positif terhadap peningkatan kualitas hidup masyarakat melalui pemberian akses yang mudah, cepat, dan efisien terhadap layanan kesehatan.

## **Rumusan Masalah**

Berdasarkan latar belakang, terdapat beberapa permasalahan yang perlu dipecahkan melalui pengembangan aplikasi klinik berobat online ini. Rumusan masalah proyek ini dapat dirinci sebagai berikut :

1. Bagaimana meningkatkan efisiensi dan kualitas pelayanan kesehatan dengan mengimplementasikan sistem klinik online untuk mengatur jadwal temu dengan dokter secara online?
2. **Bagaimana membangun aplikasi web klinik untuk mengelola data klinik seperti data pasien, data jadwal dokter, data obat, pendaftaran dan rekam medis oleh pegawai klinik?**
3. Bagaimana memastikan keamanan data pasien dalam penggunaan sistem klinik online?

## **Tujuan**

Tujuan dari laporan proyek akhir ini adalah :

1. Meningkatkan efisiensi dan aksesibilitas layanan kesehatan dengan mengembangkan sistem klinik online yang memungkinkan pasien mengatur jadwal temu dengan dokter secara efisien.
2. Mengembangkan aplikasi web klinik yang memiliki fungsionalitas menyeluruh untuk mengelola data pasien, data jadwal dokter, dan data obat secara terpadu.
3. Menjamin keamanan data pasien melalui implementasi langkah-langkah keamanan yang ketat dalam sistem klinik online.

Dengan mencapai tujuan-tujuan ini, laporan ini diharapkan dapat memberikan wawasan mendalam tentang proses pengembangan aplikasi, efektivitas dalam meningkatkan pelayanan kesehatan, serta solusi untuk mengatasi tantangan yang mungkin timbul dalam implementasi teknologi ini.

## **Batasan Masalah**

1. Tidak menyediakan nomor antrean untuk pasien.

2. Tidak bisa menerima asuransi kesehatan.

3. Tidak bisa melihat rekam medis orang lain.

4. Pembatasan jumlah pasien setiap harinya.

# **BAB II TINJAUAN PUSTAKA**

## **2.1 Konsep Analisis Desain Berorientasi Objek**

Menurut (Pelawi, 2013)Analisis berorientasi objek mendefinisikan semua jenis objek yang digunakan dalam suatu sistem dan menunjukkan kepada user kebutuhan yang diperlukan untuk berinteraksi dengan sistem guna menyelesaikan tugas yang dilakukan. Teknik ini bertujuan untuk mempelajari objek yang ada untuk melihat apakah objek tersebut masih dapat digunakan kembali atau dipakai lagi untuk penggunaan yang baru. Ini juga digunakan untuk mendefinisikan objek baru atau objek-objek yang sudah dimodifikasi yang akan digabungkan dengan objek yang sudah ada untuk menghasilkan aplikasi komputer yang berguna.

Metode analisis dan desain berorientasi objek merupakan suatu metode untuk memeriksa kebutuhan dari sudut pandang kelas dan objek kemudian dipelajari permasalahannya dengan menspesifikan atau mengobservasi permasalahan tersebut sesuai dengan gabungan antara struktur data dan perilaku dalam satu entitas. Tujuan dari analisis berorientasi objek adalah untuk mengembangkan model yang menggambarkan perangkat lunak komputer sekaligus memenuhi serangkaian persyaratan yang ditentukan pelanggan.

Komponen utama dalam analisis desain berorientasi objek :

1. Objek : Objek adalah entitas dasar dalam analisis desain berorientasi objek. Mereka mendeskripsikan elemen-elemen nyata dalam sistem.
2. Kelas : Kelas adalah blueprint suatu objek. Kelas mendefinisikan atribut (sifat) dan metode (tindakan) yang akan dimiliki objek yang dibuat dari kelas tersebut.
3. Atribut : Atribut adalah data yang berkaitan dengan suatu objek atau kelas.
4. Metode : Metode adalah suatu tindakan atau operasi yang dapat dilakukan pada suatu objek.
5. **Pewarisan (Inheritance)** : Ini adalah konsep yang memungkinkan suatu kelas mewarisi atribut dan metode dari kelas lain.

Dengan memodelkan sistem sebagai kumpulan objek yang berinteraksi, pengembang dapat membuat perangkat lunak yang lebih akurat, mudah dikelola, dan dapat digunakan kembali. Analisis desain berorientasi objek memainkan peran penting dalam dunia pengembangan perangkat lunak modern dan terus meningkat seiring dengan meningkatnya kompleksitas sistem.

## **2.2 Prinsip Basis Data**

Menurut (Jantce TJ Sitinjak et al., 2020)Basis data atau database adalah sekumpulan tabel-tabel yang berisikan sekumpulan data dan fakta sebagai sumber informasi yang disimpan dalam media penyimpanan secara digital dan dapat diperiksa menggunakan suatu program komputer yang berguna untuk memudahkan aktivitas untuk memperoleh informasi.

Melalui database ini pengguna dapat dengan mudah mencari informasi, menyimpan informasi, ataupun membuang informasi. Database umumnya berada pada perangkat komputer dan dikelola secara otomatis oleh [Database Management System](https://www.dewaweb.com/blog/apa-itu-dbms/" \o "Database Management System) (DBMS). Database berperan penting dalam perangkat untuk mengumpulkan informasi, data, atau file secara terintegrasi.

Prinsip basis data adalah pengarsipan dan pengorganisasian data. Tujuan utamanya adalah untuk memudahkan pengambilan data dan arsip yang disimpan. Perbedaan database dengan lemari arsip hanya terletak pada media penyimpanannya. Jika database menggunakan media penyimpanan elektronik.

Operasi dasar basis data menurut (Sari, 2018) :

1. Pembuatan basis data baru (create database).
2. Penghapusan basis data (drop database).
3. Membuat tabel baru di database (create table).
4. Menghapus tabel dari suatu database (drop table).
5. Menambahkan atau mengisi data baru ke sebuah tabel di sebuah database (insert).
6. Pengambilan data dari sebuah tabel (query)
7. Pengubahan data dari sebuah tabel (update).
8. Hapus data dari sebuah tabel (delete).

## **2.3 Pemrograman Web dan Teknologi Terkait**

Pemrograman web adalah salah satu cabang dari pemrograman komputer. Menurut (Pembelajaran, 2013)Pemograman web dapat diartikan sebagai kegiatan pembuatan program atau aplikasi berbasis web menggunakan bahasa pemograman tertentu sehingga dapat memproses data dan menghasilkan informasi sesuai yang dikehendaki oleh pemilik website.

Pemrograman web merupakan proses pembuatan website untuk keperluan internet. Banyak orang mengenal web dengan istilah WWW atau World Wide Web*.* World Wide Webadalah halaman-halaman webyang dihubungkan satu sama lain atau disebut juga hyperlink yang membentuk lautan informasi dengan menggunakan protokol HyperText Transfer Protocol (HTTP).

Dalam semua ilmu pemrograman, selalu ada [bahasa pemrograman](https://www.dicoding.com/blog/sejarah-bahasa-pemrograman/) yang digunakan untuk membuat aplikasi yang berisi pernyataan, perintah, atau pun fungsi agar aplikasi yang dibuat sesuai dengan desain programmer*.* Oleh karena itu, banyak juga bahasa yang digunakan dalam pemrograman web, antara lain :

1. HyperText Markup Language **(HTML) :** HTML sebenarnya bukan bahasa pemrograman, melainkan bahasa penandaan yang terdiri dari sekumpulan tag. Pada dasarnya HTML hanya mendefinisikan bagian tertentu dari suatu halaman web yaitu isi yang perlu ditampilkan oleh browserdengan cara tertentu.
2. Cascading Style Sheet **(CSS) :** CSS adalah bahasa yang digunakan untuk mengontrol tampilan suatu dokumen yang ditulis dengan HTML. CSS juga memiliki kerangka css dan digunakan untuk menambahkan pola tertentu pada halaman web agar desain halaman menarik dilihat.
3. Hypertext Preprocessor **(PHP) :** Bahasa pemrograman PHP merupakan bahasa scriptingyang harus dikuasai oleh seorang web developer. Bahasa pemrograman PHP memerlukan web server untuk dapat dijalankan. Bahasa pemrograman PHP digunakan dengan file HTML untuk memungkinkan file melakukan berbagai fungsi.
4. **JavaScript :** [JavaScript adalah](https://www.dicoding.com/blog/apa-itu-javascript-fungsi-dan-contohnya/) bahasa scripting yang berjalan di sisi client*.* Artinya pemrosesan script dilakukan di komputer user. Secara umum, JavaScript digunakan untuk membuat animasi dan bentuk interaktif lainnya pada halaman web. Untuk menjalankan scriptyang ditulis dalam JavaScript, kita memerlukan browseryang mendukung dan dapat menjalankan JavaScript atau sering disebut javascript-enabled browser.
5. Structured Query Language**(SQL) :** SQL adalah bahasa khususdomain yang digunakan untuk mengolah data dalam Relational Database Management System (RDBMS). Aplikasi RDBMS yang banyak digunakan oleh pengembang aplikasi web untuk mengelola databasenya adalah MySQL. Secara umum fungsi bahasa pemrograman PHP digunakan untuk membuat, membaca, mengubah atau pun menghapus data pada SQL, yang kemudian dapat ditampilkan pada halaman web.

Semua bahasa pemrograman web diatas akan digunakan untuk membuat proyek akhir yang berkaitan dengan pembuatan aplikasi sistem klinik berbasis web ini. Dengan menggunakan teknologi pemrograman web ini diharapkan aplikasi bisa memenuhi segala kebutuhan yang dapat mempermudah kerja dan sistem pengelolaan klinik.

## **2.4 Konsep Pemrograman Berorientasi Objek**

Pemrograman Berorientasi Objek (OOP) adalah pemrograman yang berorientasi kepada objek, dimana semua data dan fungsi dikemas ke dalam kelas atau objek. Setiap objek dapat menerima pesan, memproses data, mengirim data, dan menyimpan serta memanipulasi data. Beberapa objek berinteraksi dengan saling memberikan informasi satu sama lain (Ristianti, 2019).

Model data berorientasi objek dapat membuat struktur program kita lebih rapih dan efektif untuk data yang besar dan kompleks, serta memudahkan untuk memodifikasi atau memelihara suatu program ketika terjadi bug atau error.

Penggunaan pemrograman berorientasi objek mempunyai beberapa keuntungan, seperti : Modifikasi sistem yang dapat dilakukan tanpa banyak modul, proses maintenance lebih mudah walaupun data dalam jumlah yang besar dan kompleks, dapat mempercepat penyelesaian program, karna didukung banyak library objek, menghemat memori, dan mengurangi biaya yang dibutuhkan.

Pemrograman berorientasi objek memiliki beberapa istilah, seperti :

1. **Class** adalah blueprint dari object. Sebuah class berisi kode yang menjelaskan bagaimana object berperilaku dan berinteraksi satu sama lain.
2. **Method** adalah fungsi atau perilaku yang ada didalam class.
3. **Property** yaitu variabel yang dideklarasikan di dalam sebuah class, tetapi tidak terdapat dalam fungsi atau method dari suatu class.
4. **Object** merupakan bagian dari program yang berisi variabel dan method yang saling terhubung.
5. **Overloading** adalah pemisalan dalam sebuah class.
6. **Abstraksi** yaitu kemampuan suatu program dalam meyampaikan informasi yang akan diolah, yaitu kemampuan memfokuskan perhatian pada inti program.
7. **Enkapsulasi** adalah sebuah metode untuk mengatur struktur class dengan cara menyembunyikan alur kerja dari class tersebut.
8. **Polimorfisme** adalah konsep dimana terdapat banyak class yang memiliki signature method yang sama.

# **BAB III ANALISIS SISTEM**

## **3.1 Spesifikasi Persyaratan Sistem**

**SKENARIO :**

Untuk final project di semester 3 ini, kami diminta membuat sistem aplikasi untuk klinik. Klinik yang akan dibuat aplikasinya bernama Klinik Pratama Khazanah Medika, Klinik ini masih baru beroperasi sekitar 2 atau 3 bulan sehingga masih mengandalkan semuanya secara manual. Sistem yang akan dibuat untuk klinik ini antara lain sebagai berikut : Mengantur jadwal temu dengan dokter secara online. Melakukan pendaftaran atau registrasi, membuat janji temu dengan dokter, berobat ke klinik lalu mendapatkan resep obat di aplikasi.

### **Analisis Sistem**

* Identifikasi Objek

1. Resepsionis
2. Pasien
3. Pemilik klinik
4. Informasi medis

* Model Kamus Data

1. Resepsionis : {id\_karyawan, nama, alamat, no\_telp, jam\_kerja}
2. Pasien : {id\_rekam\_medis, nama, alamat, tanggal lahir, no\_telp, penyakit}
3. Pemilik klinik : {id\_pemilik, nama, alamat, no\_telp}
4. Informasi medis : {riwayat\_penyakit, riwayat\_alergi, obat\_diresepkan, diagnosa\_dokter}

* Kebutuhan Fungsional Sistem

1. Pasien dapat memilih waktu temu yang tersedia dan melakukan pemesanan online.
2. Sistem harus memungkinkan pasien untuk melihat jadwal dokter secara online.
3. Sistem ini membuat pasien dapat memperoleh resep obat secara online.

* Kebutuhan Non Fungsional Sistem

1. Sistem ini berbasis web aplikasi.
2. Menggunakan bahasa pemrograman : HTML, PHP dan CSS.
3. Sistem ini menggunakan database untuk pengarsipan data.
4. Sistem dapat berjalan pada computer spesifikasi minimal RAM 2GB.
5. Sistem dapat berjalan pada kecepatan 5 sampai 10 Mbps.

### **Scope**

Scope sistem klinik ini mencakup berbagai aspek yang perlu diperhatikan dalam perancangan, pengembangan, dan implementasi sistem. Berikut adalah gambaran umum tentang apa yang termasuk dalam scope :

* Sistem akan mencakup manajemen lengkap atas registrasi, membuat janji temu dengan dokter secara online dan juga mendapat resep obat di aplikasi.
* Pasien dapat membuat janji temu dengan dokter secara online.
* Obat dapat diresepkan secara online dari sistem.
* Resep obat dapat di beli di apotek atau di klinik secara langsung.

### **User Characteristic**

* Resepsionis : Dapat membuat janji temu antara dokter dengan pasien, menginput dan melihat rekam medis pasien dan meresepkan obat untuk pasien.
* Pasien : Dapat melakukan registrasi melalui aplikasi, membuat janji temu dengan dokter di aplikasi dan mendapatkan resep obat.
* Pemilik klinik **:** Bisa mengatur dan melihat jalannya sistem klinik dan dapat melakukan semua yang dilakukan aktor lain karena pemilik klinik merupakan super admin.

### **Constraints**

Batasan sistem klinik ini adalah sebagai berikut :

* Sistem dimana resepsionis dapat membuat janji temu antara dokter dan pasien, menginput rekam medis pasien dan juga meresepkan obat untuk pasien di aplikasi.
* Sistem pasien dapat registrasi online, membuat janji temu secara online, memberitahu keluhan penyakit awal dan di resepkan obat dalam aplikasi.
* Sistem dimana pemilik klinik dapat mengontrol dan melihat alur jalannya klinik.

### **Assumptions and Depedencies**

Asumsi dan ketergantungan yang digunakan dalam sistem ini adalah :

1. Asumsi bahwa pasien memiliki perangkat yang memadai (komputer atau smartphone) dan pengetahuan teknologi yang cukup untuk menggunakan platform online.
2. Asumsi bahwa informasi yang diberikan oleh pasien selama registrasi online dan deskripsi keluhan awal adalah akurat dan benar.
3. Ketergantungan pada integrasi sistem dengan database klinik untuk mengakses dan menyimpan data pasien, riwayat medis, dan informasi obat.
4. Ketergantungan pada implementasi kebijakan keamanan informasi yang efektif untuk melindungi data pasien dan transaksi online juga peraturan privasi dan keamanan data kesehatan yang berlaku.

### **Specific Requirements**

Dari kebutuhan yang telah disebutkan, berikut adalah beberapa persyaratan spesifik yang perlu diperhatikan dalam pengembangan sistem klinik :

* Sistem harus menyediakan formulir pendaftaran online yang mencakup informasi lengkap pasien, dan setiap pendaftaran harus menghasilkan akun pengguna untuk pasien.
* Informasi tentang jadwal dokter harus ditampilkan.
* Setiap obat harus memiliki informasi yang mencakup dosis, efek samping, dan instruksi penggunaan.
* Sistem harus dapat membuat janji temu antara dokter dan pasien.

### **Fungcionality**

|  |  |  |
| --- | --- | --- |
| **No** | **Fungsi** | **Deskripsi** |
| 1. | Login | Proses login admin maupun pasien ke dalam sistem klinik. |
| 2. | Dashboard | Terdapat beberapa menu yang bisa dilakukan. |
| 3. | Form register | Proses untuk pasien melakukan pengisian data diri dan keluhan. |
| 4. | Profile | Fungsi untuk menginput nama, tanggal lahir dan juga ganti password. |
| 5. | View rekam medis | Fungsi untuk menampilkan rekam medis pasien. |
| 6. | Membuat akun | Proses agar pasien dapat login ke dalam sistem. |
| 7. | Resep obat | Fungsi untuk menampilkan resep obat yang di dapat pasien. |
| 8. | Ganti Password | Fungsi untuk user mengganti password. |
| 9. | Janji Temu | Fungsi untuk pasien membuat janji temu dengan dokter. |
| 10. | Logout | Proses logout dari sistem klinik. |

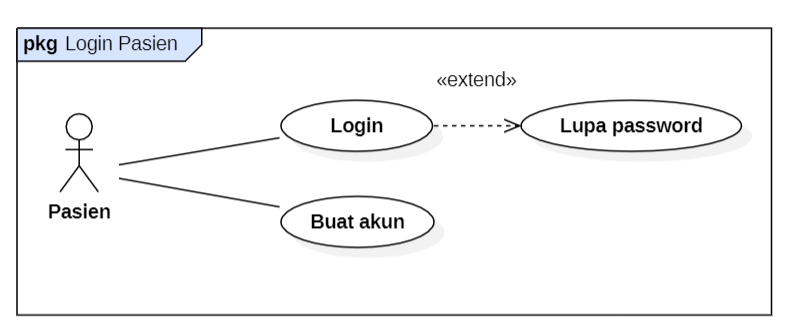
Tabel 1 Fungsi Sistem

### **Hardware Interface**

1. Memerlukan server untuk menyimpan data pasien, jadwal dokter, rekam medis, dan informasi lainnya.
2. Menerapkan sistem database di phpMyAdmin untuk menyimpan dan mengelola data pasien, resep obat, dan informasi kesehatan lainnya.
3. Koneksi internet yang stabil diperlukan untuk memastikan akses ke sistem secara online.
4. Menggunakan bahasa pemrograman HTML, PHP dan CSS.

## **3.2 Diagram Use Case**

### **3.2.1 Use Case Diagram Login Pasien**



Gambar 1 Use Case Diagram Login Pasien

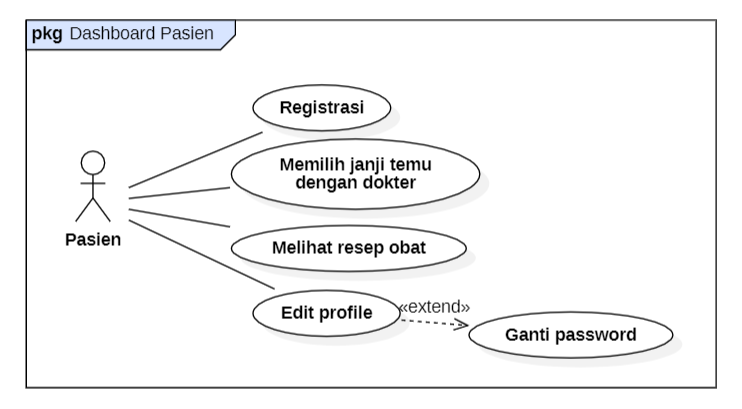
|  |  |  |
| --- | --- | --- |
| **Use Case Name** | : | Login Pasien |
| **Triger Event** | : | Aktor memilih menu login pada halaman utama |
| **Aktor** | : | Pasien |
| **Pre-condition** | : | Pasien membuat akun terlebih dahulu sebelum login |
| **Post-condition** | : | Pasien dapat login dan mengakses ke dalam sistem |
| **Normal Course** | : | |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. Mengisi data sesuai kolom yang tersedia |  | |  | 2. Validasi kelengkapan data pada form login | |  | 3. Menampilkan halaman dashboard setelah login | | 4. Lupa password |  | |
| **Alternative Course** | : | |  |  | | --- | --- | | **Aktor** | **Sistem** | |  | 3a. Muncul pesan karena data yang diisi tidak sesuai dengan form username dan password salah | | 3b. Aktor dapat memperbaiki data yang salah dan melanjutkan login |  | | 4. Melakukan reset password menggunakan tanggal lahir |  | |
| **Exception** | : | Aktor gagal login karena lupa password |

Tabel 2 Use Case Deskripsi Login

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | : | Buat Akun |
| **Triger Event** | : | Aktor memilih menu buat akun pada halaman utama |
| **Aktor** | : | Pasien |
| **Pre-condition** | : | Pasien membuat akun terlebih dahulu sebelum login |
| **Post-condition** | : | Pasien dapat login setelah selesai membuat akun |
| **Normal Course** | : | |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. Mengisi data sesuai kolom yang tersedia untuk mendaftar akun |  | |  | 2. Validasi password untuk memastikan password yang dimasukkan sudah benar | |  | 3. Menampilkan halaman login setelah berhasil membuat akun | |
| **Alternative Course** | : | |  |  | | --- | --- | | **Aktor** | **Sistem** | |  | 2a. Muncul pesan salah memasukkan validasi password | | 3b. Aktor dapat memperbaiki validasi password yang salah dan melanjutkan daftar akun |  | |
| **Exception** | : | Aktor tidak jadi mendaftar akun karena sudah memiliki akun |

Tabel 3 Use Case Deskripsi Buat Akun

### **3.2.2 Use Case Diagram Dashboard Pasien**



Gambar 2 Use Case Diagram Dashboard Pasien

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | : | Registrasi |
| **Triger Event** | : | Aktor melakukan registrasi untuk mengisi data diri dan keluhan |
| **Aktor** | : | Pasien |
| **Pre-condition** | : | Pasien mengisi form registrasi |
| **Post-condition** | : | Pasien telah mengisi registrasi dan telah tersimpan di database |
| **Normal Course** | : | |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. Mengisi formulir registrasi dan menuliskan keluhan |  | |  | 2. Menyimpan informasi data pasien ke dalam database | |
| **Alternative Course** | : | |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. Mengedit data dan keluhan |  | |  | 2. Mengupdate data yang telah di perbarui | |
| **Exception** | : | Aktor dapat membatalkan mengisi registrasi |

Tabel 4 Use Case Deskripsi Registrasi

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | : | Memilih janji temu dengan dokter |
| **Triger Event** | : | Aktor dapat membuat dan memilih janji temu dengan dokter |
| **Aktor** | : | Pasien |
| **Pre-condition** | : | Pasien memilih hari dan jam untuk membuat janji temu |
| **Post-condition** | : | Pasien mendapatkan jadwal janji temu dengan dokter |
| **Normal Course** | : | |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. Aktor memilih hari dan jam di aplikasi untuk konsul langsung dengan dokter |  | |  | 2. Menyimpan janji temu yang sudah dipilih aktor | |
| **Alternative Course** | : | |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. - |  | |  | 2. - | |
| **Exception** | : | Pasien batal membuat janji temu dengan dokter |

Tabel 5 Use Case Deskripsi Memilih Janji Temu Dengan Dokter

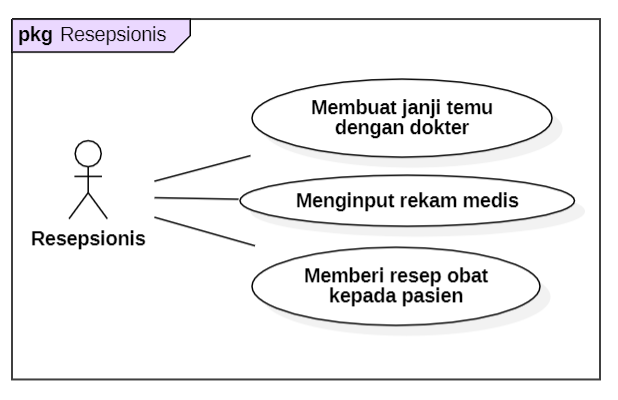
|  |  |  |
| --- | --- | --- |
| **Use Case Name** | : | Melihat Resep Obat |
| **Triger Event** | : | Aktor dapat melihat resep obat yang sudah diberikan resepsionis di aplikasi |
| **Aktor** | : | Pasien |
| **Pre-condition** | : | Pasien mendapat resep obat |
| **Post-condition** | : | Pasien memilih untuk menebus resep di apotek atau langsung di klinik secara offline |
| **Normal Course** | : | |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. Setelah selesai berobat di klinik aktor mendapatkan resep di aplikasi dan juga secara langsung |  | |  | 2. Menampilkan resep setelah aktor selesai berobat | |
| **Alternative Course** | : | |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. Memilih membeli resep obat langsung di klinik atau di apotek lain |  | |  | 2. - | |
| **Exception** | : | - |

Tabel 6 Use Case Deskripsi Melihat Resep Obat

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | : | Edit Profile |
| **Triger Event** | : | Aktor dapat mengedit profile dan ganti password |
| **Aktor** | : | Pasien |
| **Pre-condition** | : | Pasien memiliki data pribadi dalam profile |
| **Post-condition** | : | Pasien dapat mengubah data di dalam edit profile |
| **Normal Course** | : | |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. Aktor dapat mengedit data pribadi |  | |  | 2. Mengubah data aktor di dalam database | | 3. Aktor dapat mengubah password |  | |  | 4. Menyimpan password yang sudah di ganti aktor di dalam database | |
| **Alternative Course** | : | |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. Jika data masih tidak sesuai dapat diganti kembali |  | |  | 2. Mengubah kembali data dalam database | |
| **Exception** | : | Aktor tidak jadi mengedit profile dan password |

Tabel 7 Use Case Deskripsi Edit Profile

### **3.2.3 Use Case Diagram Resepsionis**



Gambar 3 Use Case Diagram Resepsionis

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | : | Membuat janji temu dengan dokter |
| **Triger Event** | : | Resepsionis memberi tahu dokter mengenai jadwal konsul yang sudah dipilih pasien |
| **Aktor** | : | Resepsionis |
| **Pre-condition** | : | Resepsionis menginput jadwal konsul |
| **Post-condition** | : | Janji temu antara dokter dan pasien terjadwal |
| **Normal Course** | : | |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. Aktor menginput jadwal janji temu yang sudah dibuat oleh pasien |  | |  | 2. Memproses ke dalam database | |
| **Alternative Course** | : | |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. Aktor memberitahu manual kepada dokter mengenai janji temu dengan pasien. |  | |  | 2. - | |
| **Exception** | : | Gagal menginput janji temu |

Tabel 8 Use Case Deskripsi Membuat Janji Temu Dengan Dokter

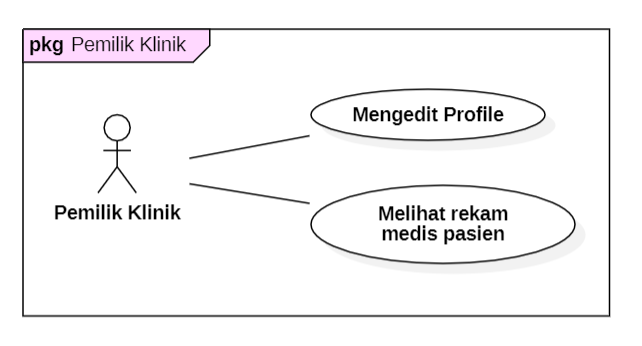
|  |  |  |
| --- | --- | --- |
| **Use Case Name** | : | Menginput Rekam Medis |
| **Triger Event** | : | Resepsionis menginput rekam medis ke dalam sistem/database |
| **Aktor** | : | Resepsionis |
| **Pre-condition** | : | Menginput atau mengedit rekam medis ke dalam sistem |
| **Post-condition** | : | Rekam medis berhasil di simpan di dalam database |
| **Normal Course** | : | |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. Menginput dan mengedit rekam medis pasien |  | |  | 2. Menyimpan rekam medis pasien ke dalam database | |
| **Alternative Course** | : | |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. - |  | |  | 2. - | |
| **Exception** | : | Gagal menginput rekam medis karena salah data |

Tabel 9 Use Case Deskripsi Menginput Rekam Medis

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | : | Memberi resep obat kepada pasien |
| **Triger Event** | : | Resepsionis memberi resep obat kepada pasien didalam sistem |
| **Aktor** | : | Resepsionis |
| **Pre-condition** | : | Menginput resep obat dan dosisnya |
| **Post-condition** | : | Resep obat berhasil di input ke dalam sistem |
| **Normal Course** | : | |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. Aktor menginputkan resep obat ke dalam sistem |  | |  | 2. Berhasil menginput resep obat | |
| **Alternative Course** | : | |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. Mengganti resep obat pasien |  | |  | 2. Mengedit resep obat dan meyimpannya | |
| **Exception** | : | - |

Tabel 10 Use Case Deskripsi Memberi Resep Obat Kepada Pasien

### **3.2.4 Use Case Diagram Pemilik Klinik**



Gambar 4 Use Case Diagram Pemilik Klinik

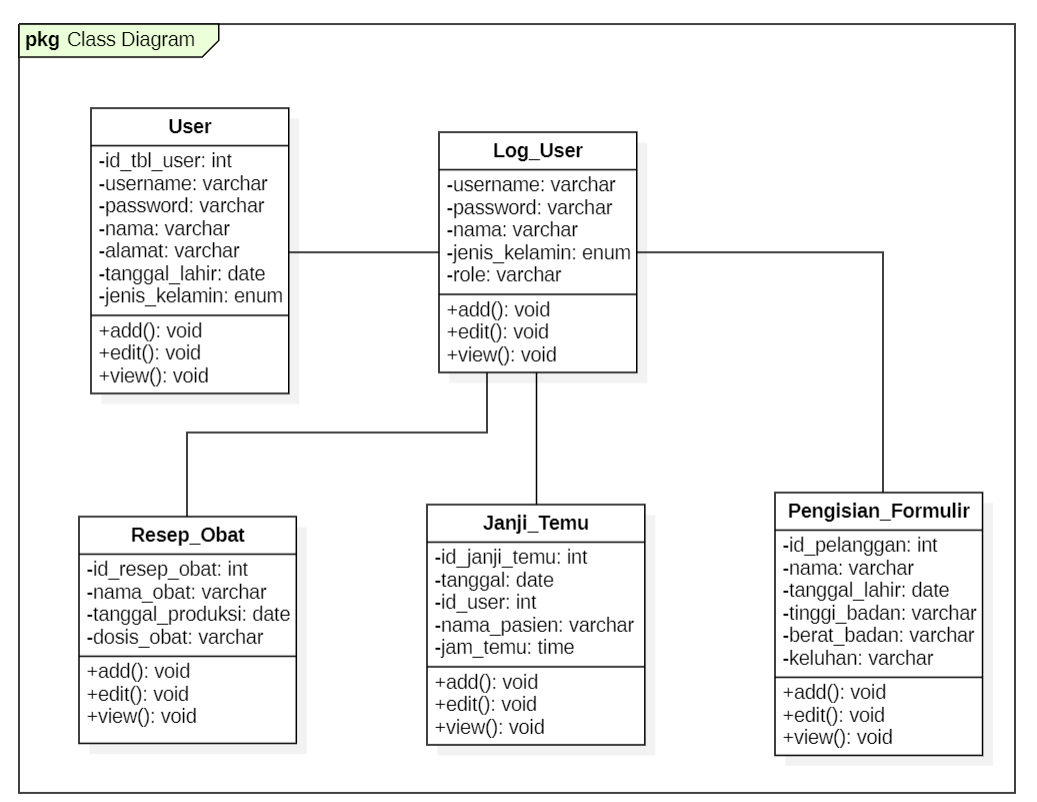
|  |  |  |
| --- | --- | --- |
| **Use Case Name** | : | Mengedit profile |
| **Triger Event** | : | Aktor dapat mengedit profile |
| **Aktor** | : | Pemilik klinik |
| **Pre-condition** | : | Aktor memiliki data pribadi dalam profile |
| **Post-condition** | : | Aktor dapat mengubah data di dalam edit profile |
| **Normal Course** | : | |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. Aktor dapat mengedit data pribadi |  | |  | 2. Mengubah data aktor di dalam database | | 3. Aktor dapat mengubah password |  | |  | 4. Mengubah password yang sudah di ganti aktor di dalam database | |
| **Alternative Course** | : | |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. Jika data masih tidak sesuai dapat diganti kembali |  | |  | 2. Mengubah kembali data dalam database | |
| **Exception** | : | - |

Tabel 11 Use Case Deskripsi Mengedit Profile

|  |  |  |
| --- | --- | --- |
| **Use Case Name** | : | Melihat rekam medis pasien |
| **Triger Event** | : | Pemilik klinik dapat melihat rekam medis pasien karena super admin |
| **Aktor** | : | Pemilik klinik |
| **Pre-condition** | : | Melihat rekam medis untuk laporan klinik |
| **Post-condition** | : | Dapat melihat rekam medis pasien |
| **Normal Course** | : | |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. Melihat rekam medis pasien |  | |  | 2. Menampilkan rekam medis pasien dari database | |
| **Alternative Course** | : | |  |  | | --- | --- | | **Aktor** | **Sistem** | | 1. - |  | |  | 2. - | |
| **Exception** | : | - |

Tabel 12 Use Case Deskripsi Melihat Rekam Medis Pasien

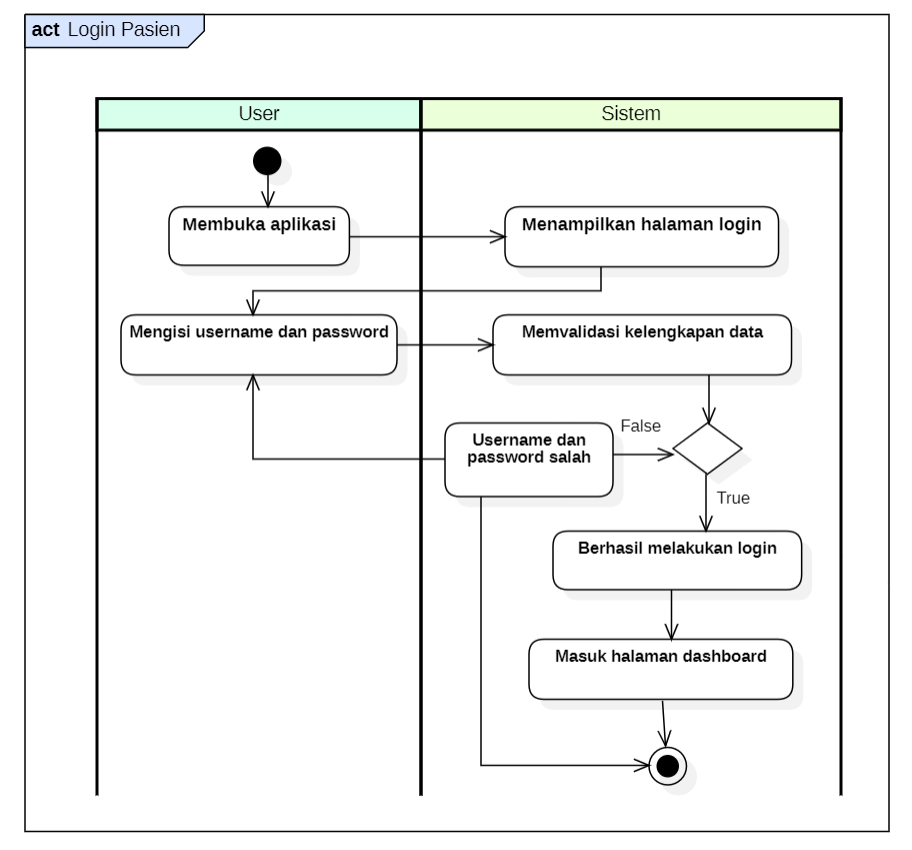
## **3.3 Diagram Kelas**



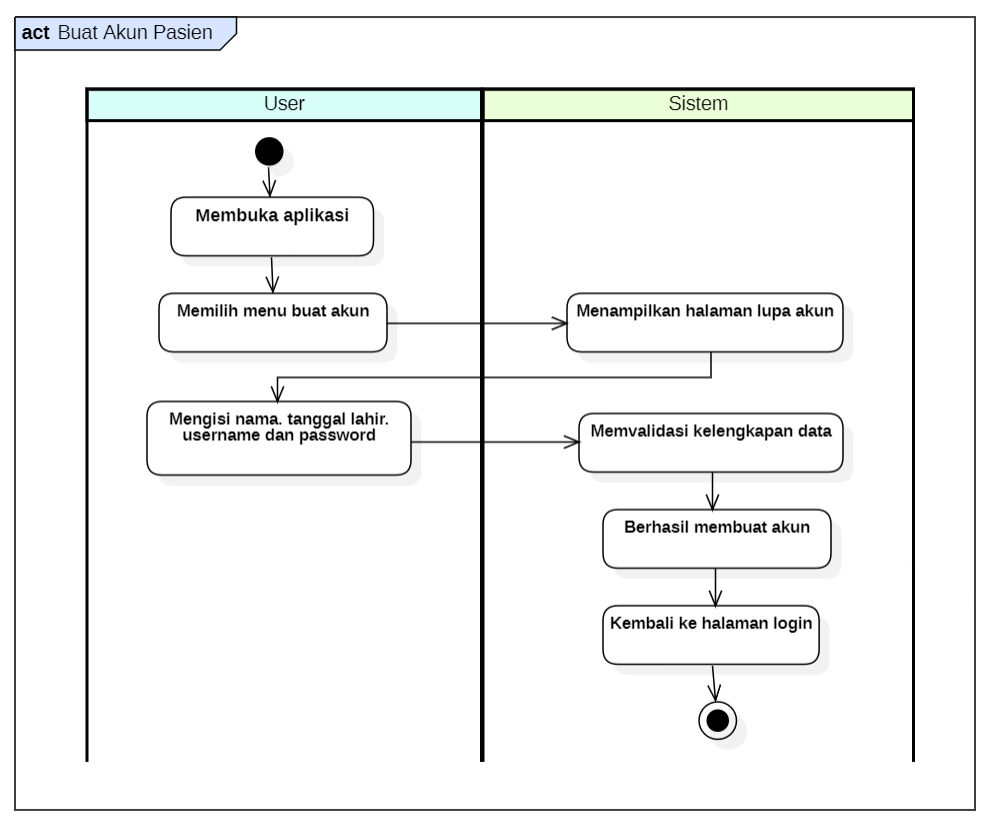
Gambar 5 Diagram Kelas

## **3.4 Diagram Activity**

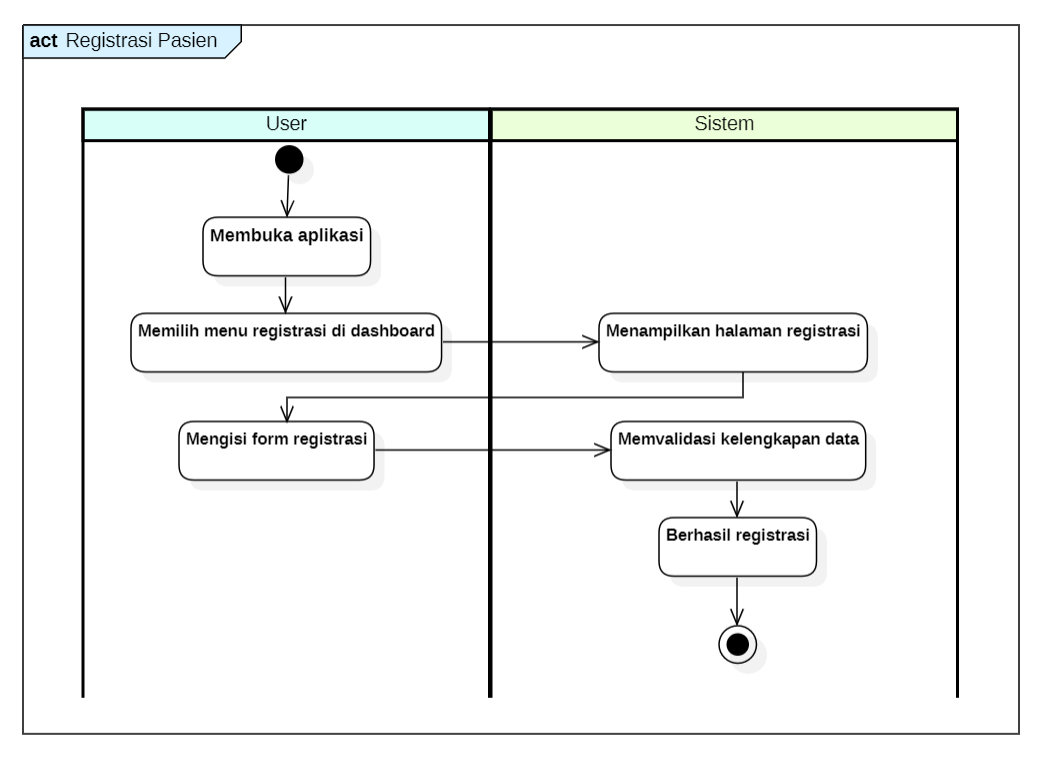
### **3.4.1 Diagram Activity Pasien**

****

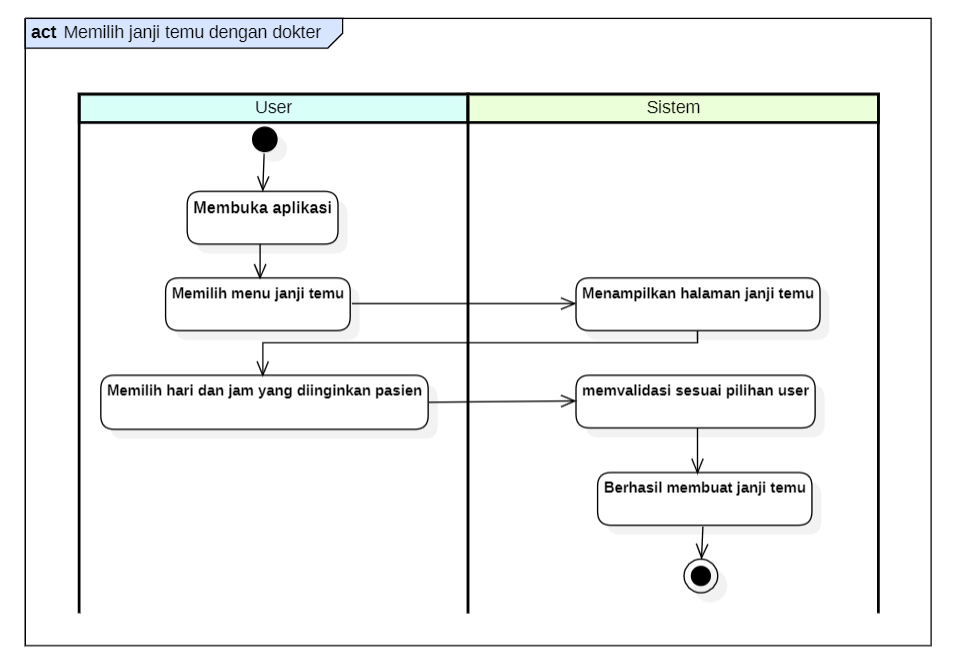
Gambar 6 Diagram Activity Login Pasien

****

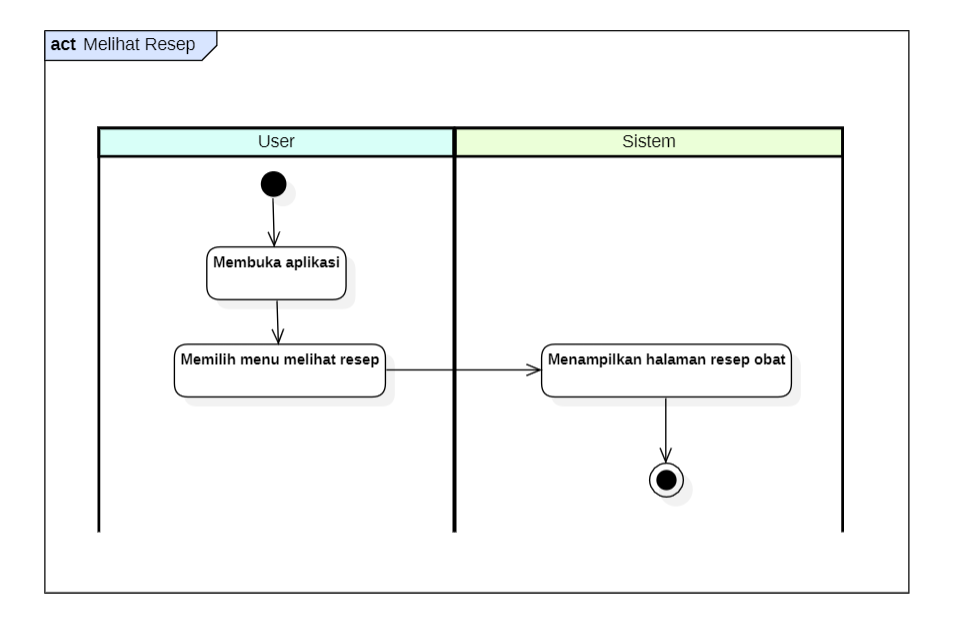
Gambar 7 Diagram Activity Buat Akun Pasien

****

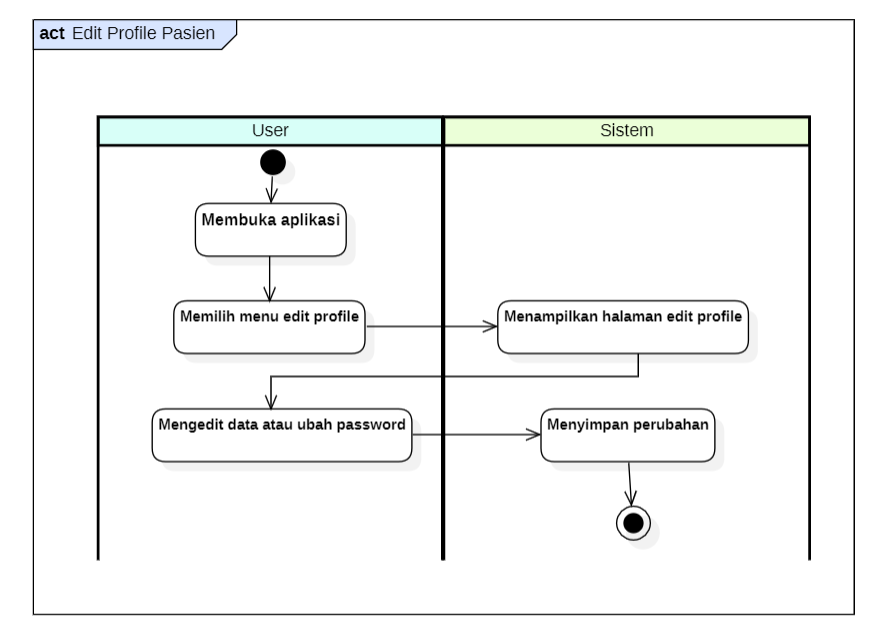
Gambar 8 Diagram Activity Registrasi Pasien

****

Gambar 9 Diagram Activity Memilih Janji Temu Dengan Dokter

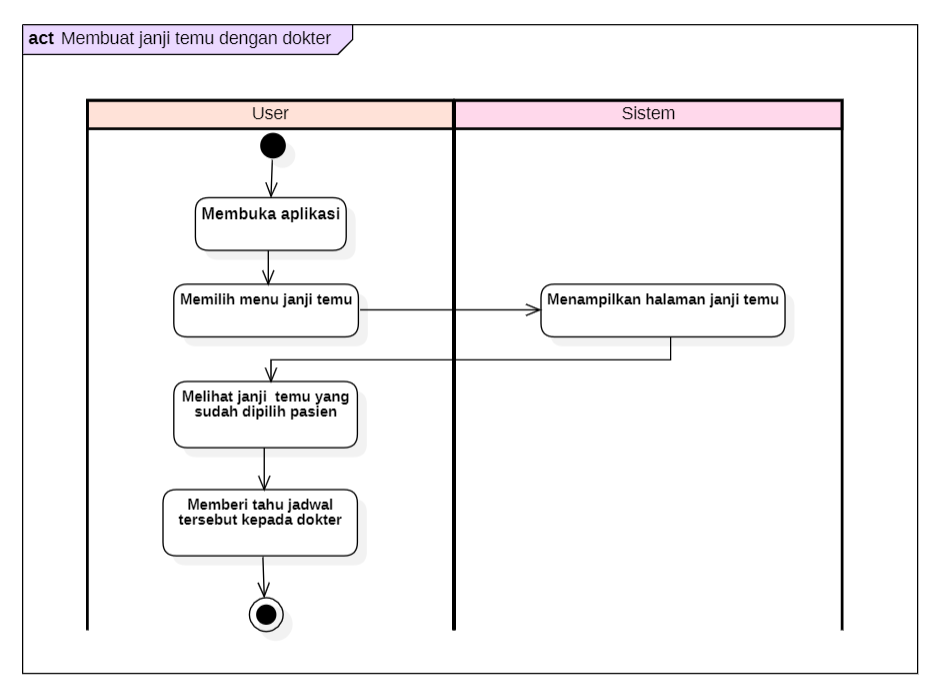
****

Gambar 10 Diagram Activity Melihat Resep Obat

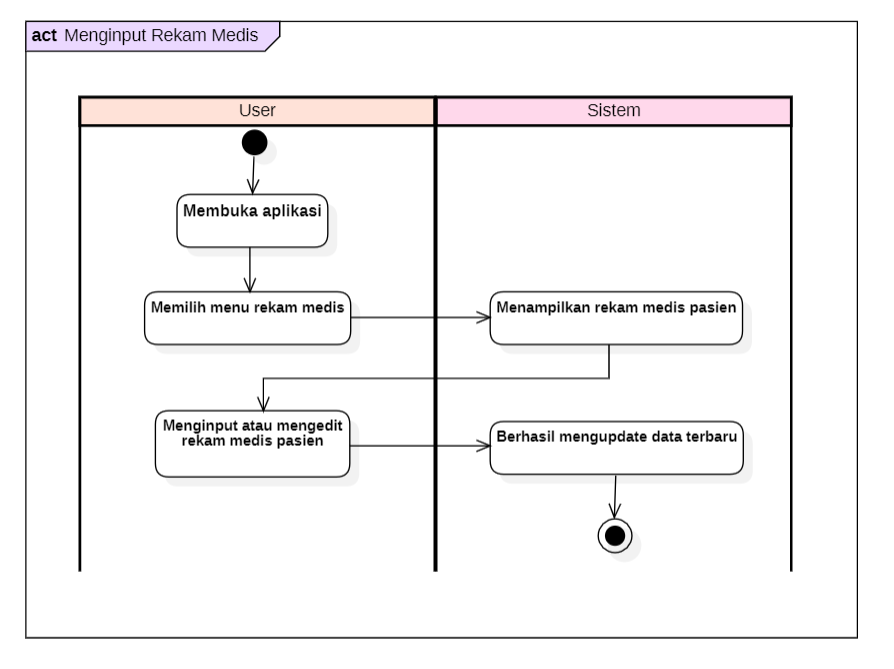
****

Gambar 11 Diagram Activity Edit Profile Pasien

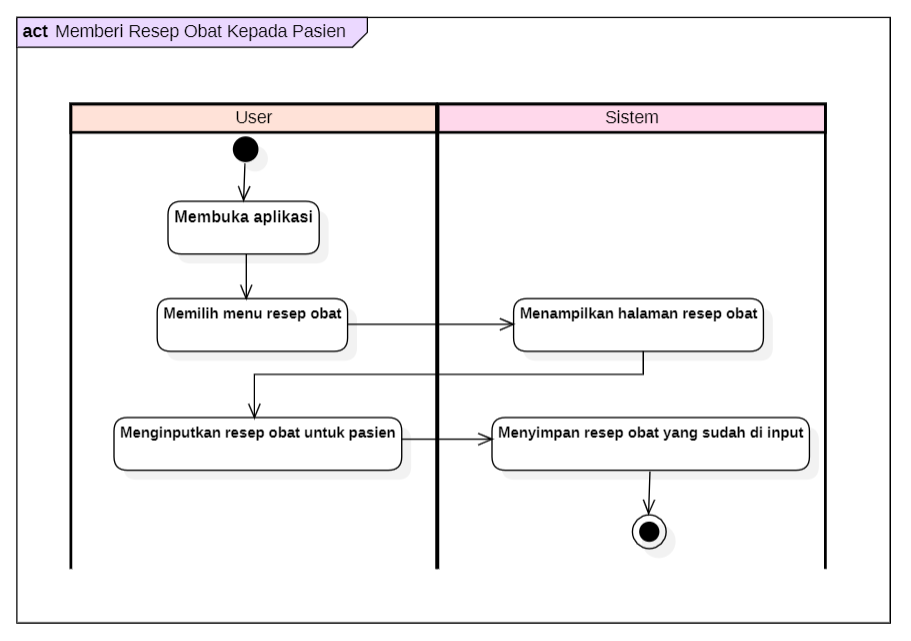
### **3.4.2 Diagram Activity Resepsionis**

****

Gambar 12 Diagram Activity Membuat Janji Temu Dengan Dokter

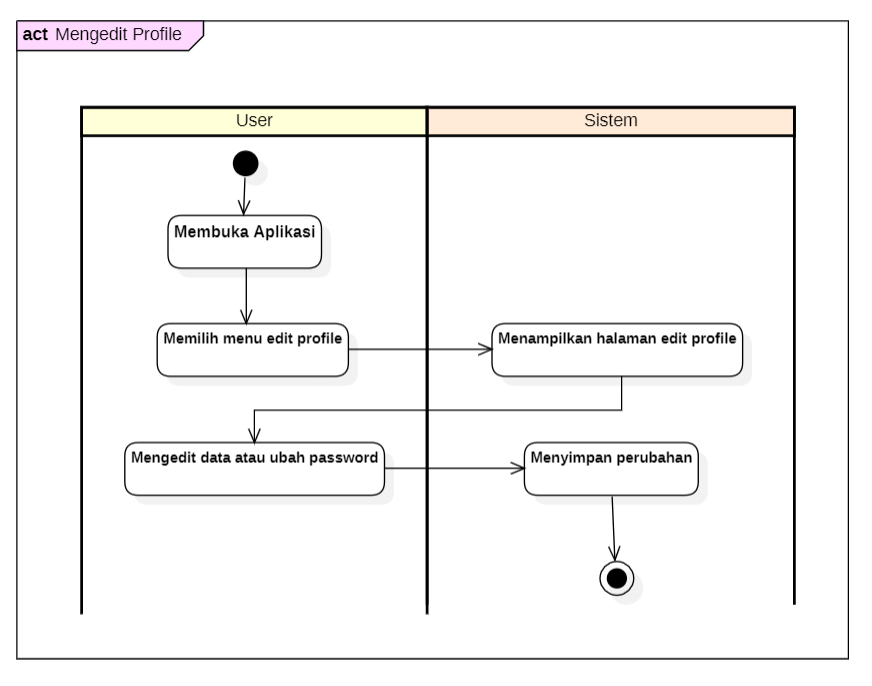
****

Gambar 13 Diagram Activity Menginput Rekam Medis Pasien

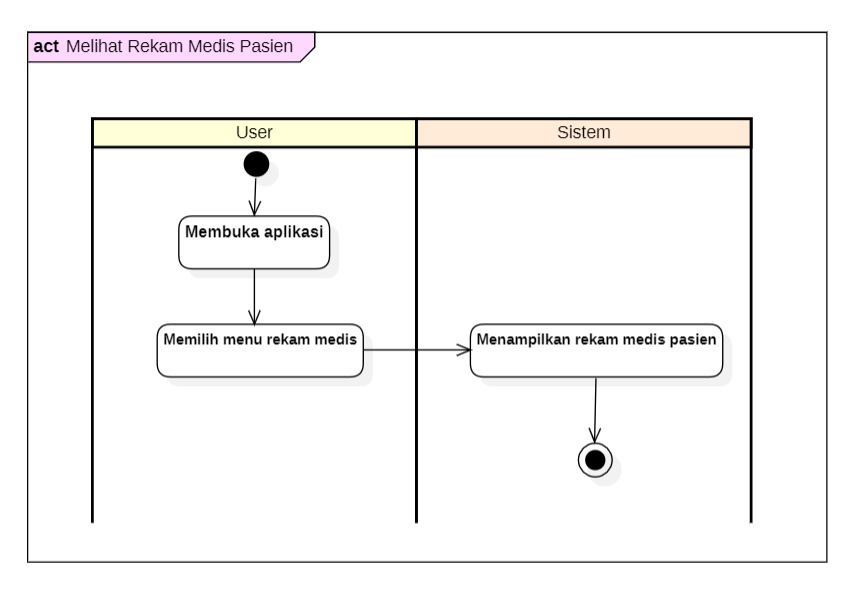
****

Gambar 14 Diagram Activity Memberi Resep Obat Kepada Pasien

### **3.4.3 Diagram Activity Pemilik Klinik**

****

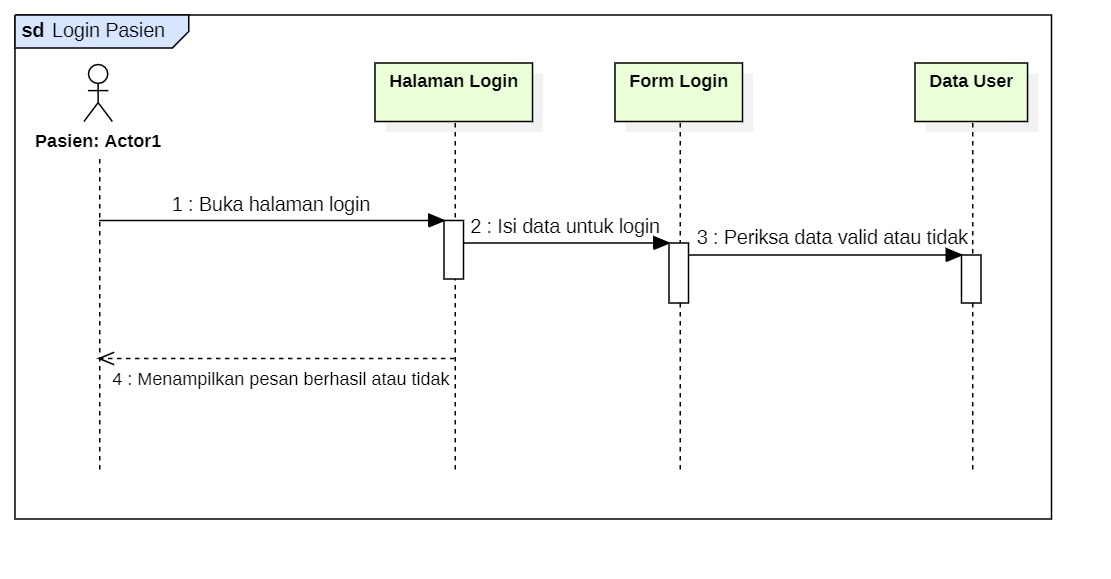
Gambar 15 Diagram Activity Mengedit Profile

****

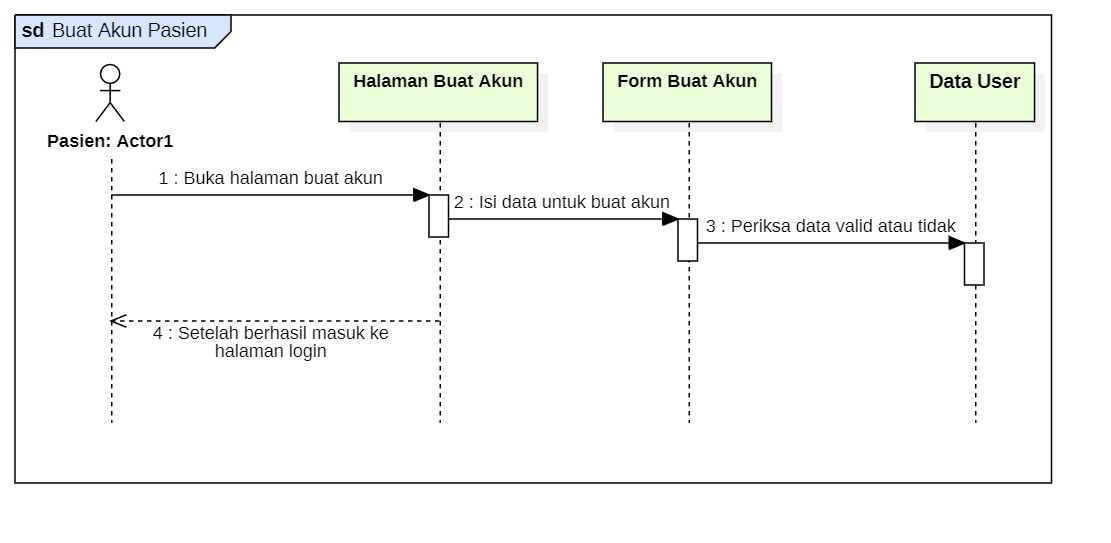
Gambar 16 Diagram Activity Melihat Rekam Medis Pasien

## **3.5 Desain Sequence**

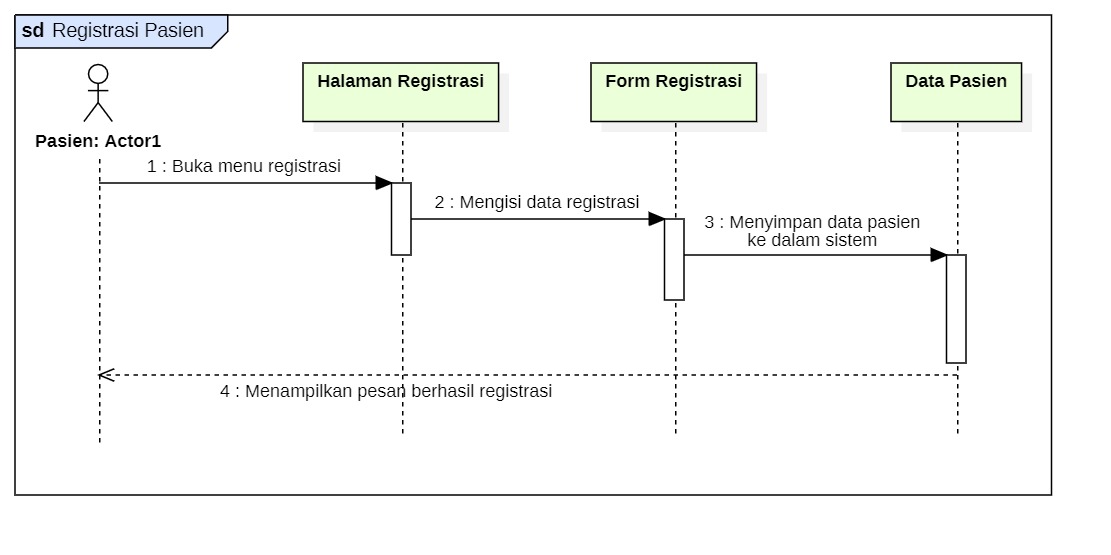
### **3.5.1 Sequence Diagram Pasien**



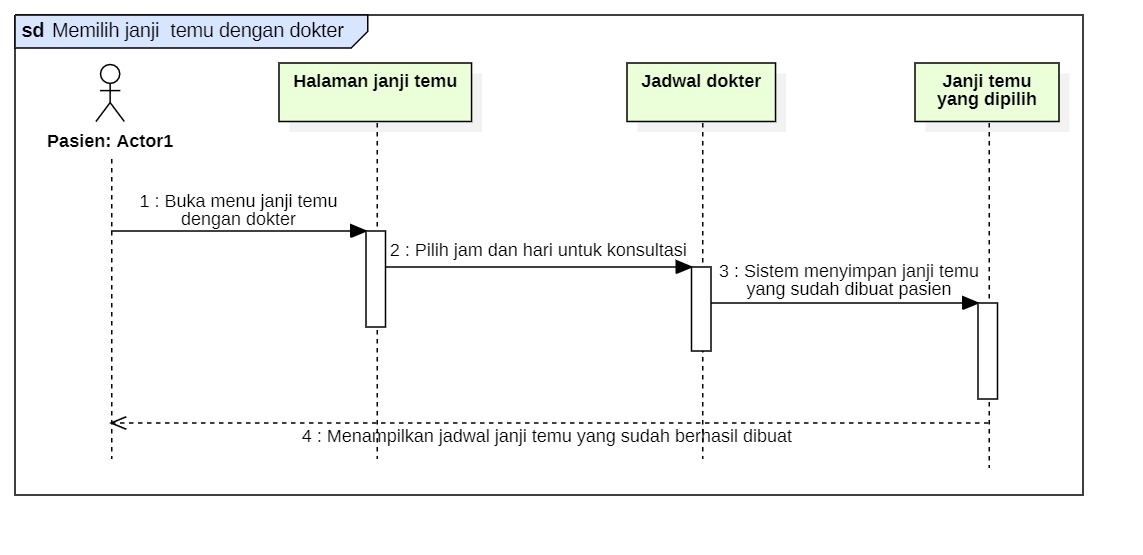
Gambar 17 Sequence Diagram Login Pasien



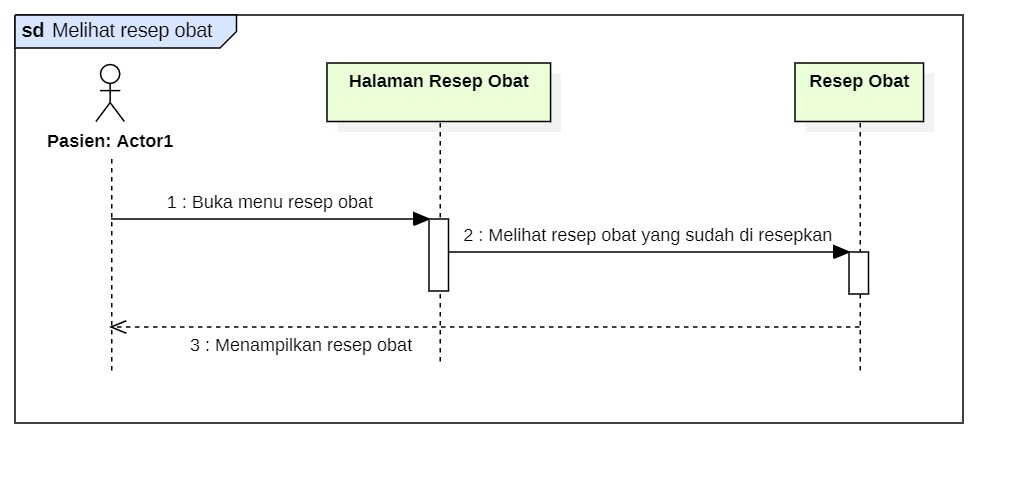
Gambar 18 Sequence Diagram Buat Akun Pasien



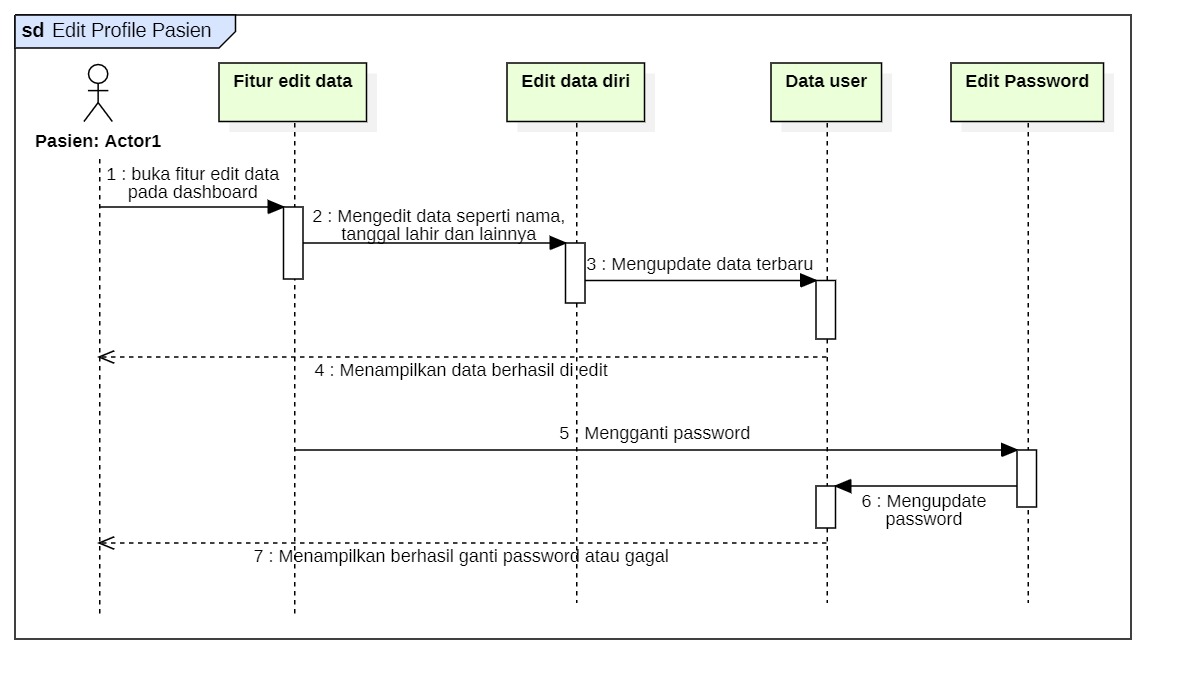
Gambar 19 Sequence Diagram Registrasi Pasien



Gambar 20 Sequence Diagram Memilih Janji Temu Dengan Dokter

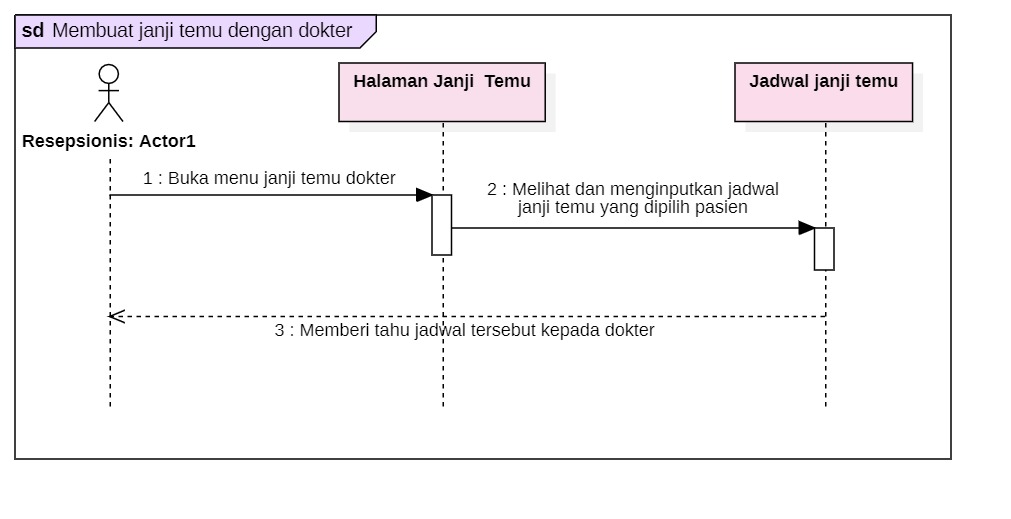


Gambar 21 Sequence Diagram Melihat Resep Obat

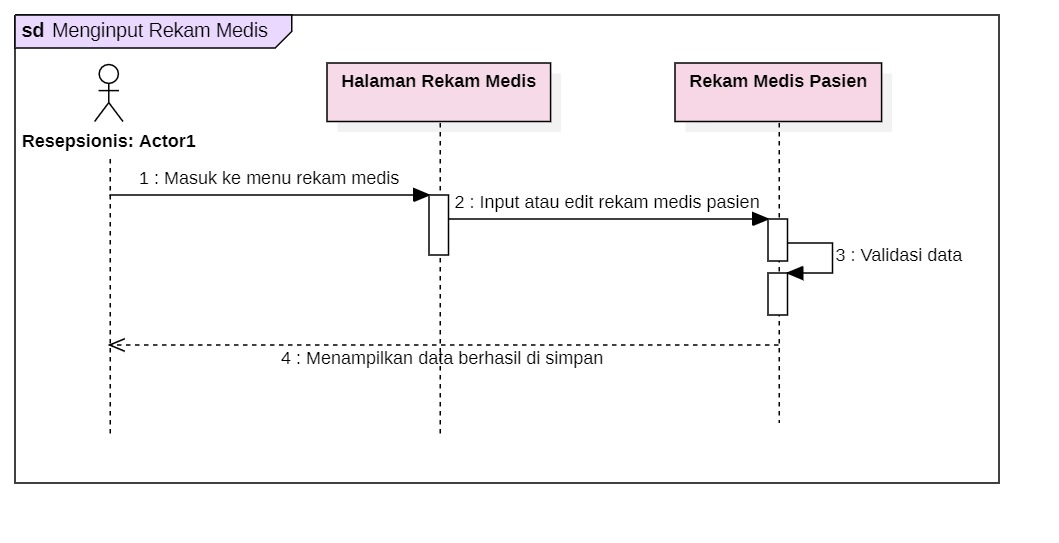


Gambar 22 Sequence Diagram Edit Profile Pasien

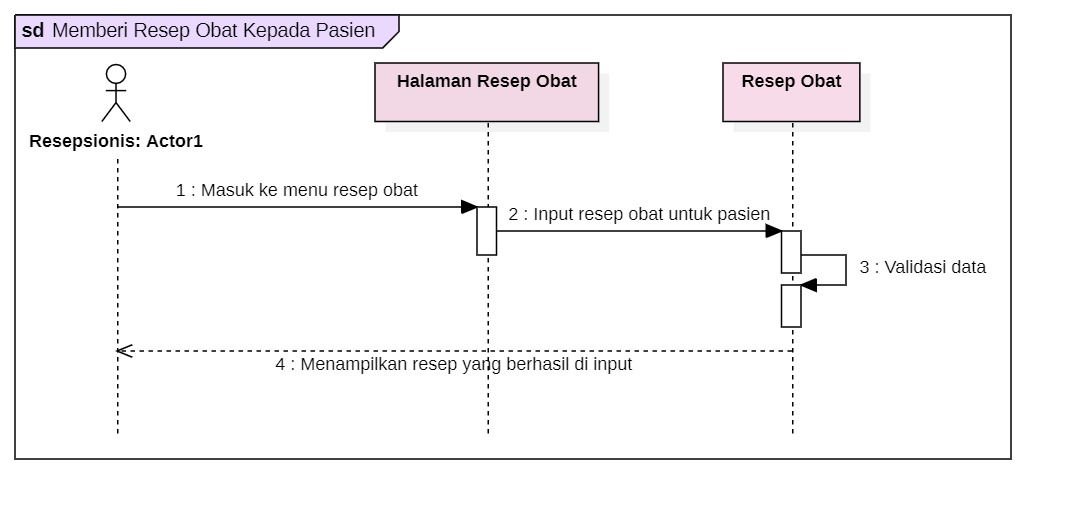
### **3.5.2 Sequence Diagram Resepsionis**



Gambar 23 Sequence Diagram Membuat Janji Temu Dengan Dokter



Gambar 24 Sequence Diagram Menginput Rekam Medis

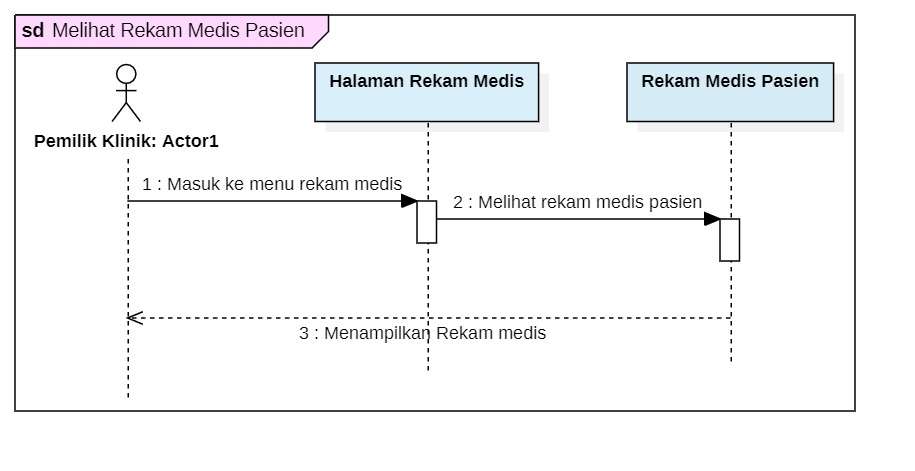


Gambar 25 Sequence Diagram Memberi Resep Obat Kepada Pasien

### **3.5.2 Sequence Diagram Pemilik Klinik**

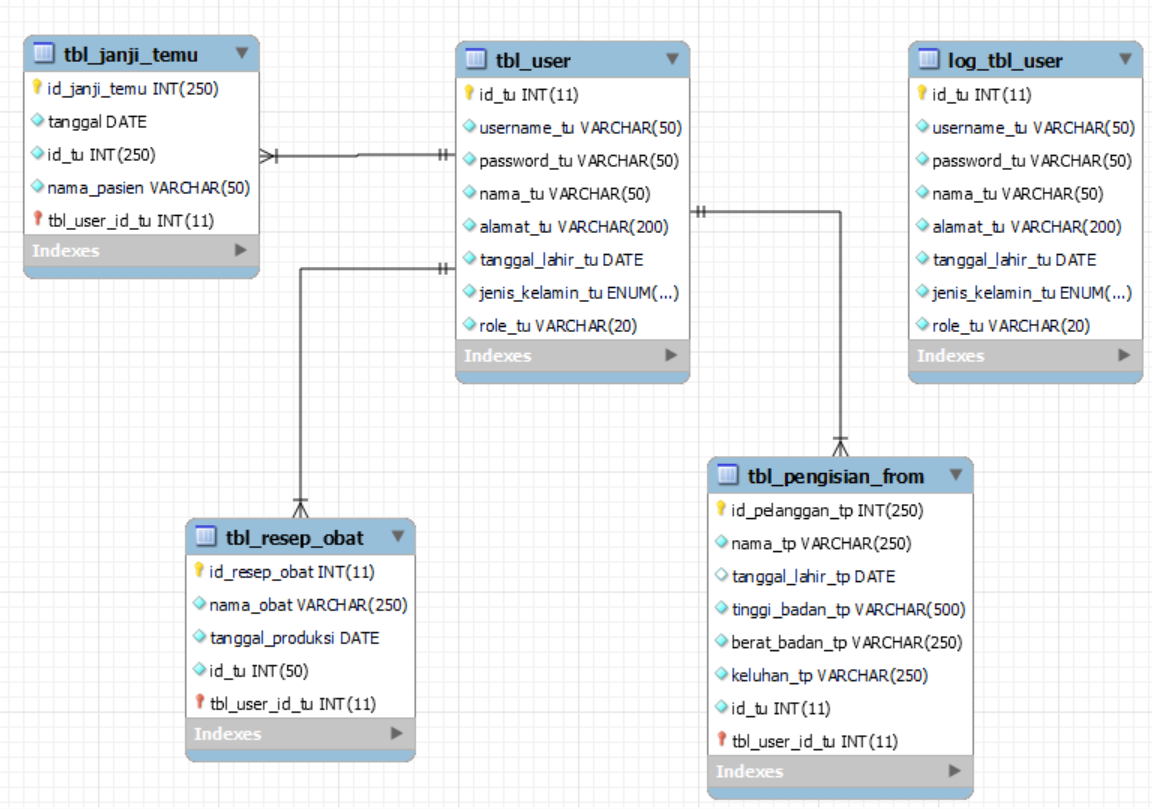


Gambar 26 Sequence Diagram Mengedit Profile



Gambar 27 Sequence Diagram Melihat Rekam Medis Pasien

## **3.6 Desain Basis Data**



Gambar 28 Desain Basis Data

Sesuai gambar desain ERD di atas terdapat 5 tabel yang digunakan untuk menjalankan sistem klinik ini. Terdapat tabel janji temu untuk mengatur janji temu antara dokter dan pasien, tabel user untuk menyimpan data pasien dan rekam medis pasien, tabel resep obat untuk menginputkan resep obat pasien ke dalam sistem, tabel pengisian form untuk mengisi formulir data dan keluhan awal pasien dan terakhir ada log tabel user yang tidak disambungkan hubungan kardinalitasnya karena tabel tersebut hanya berfungsi untuk mengambil data baru dari trigger.

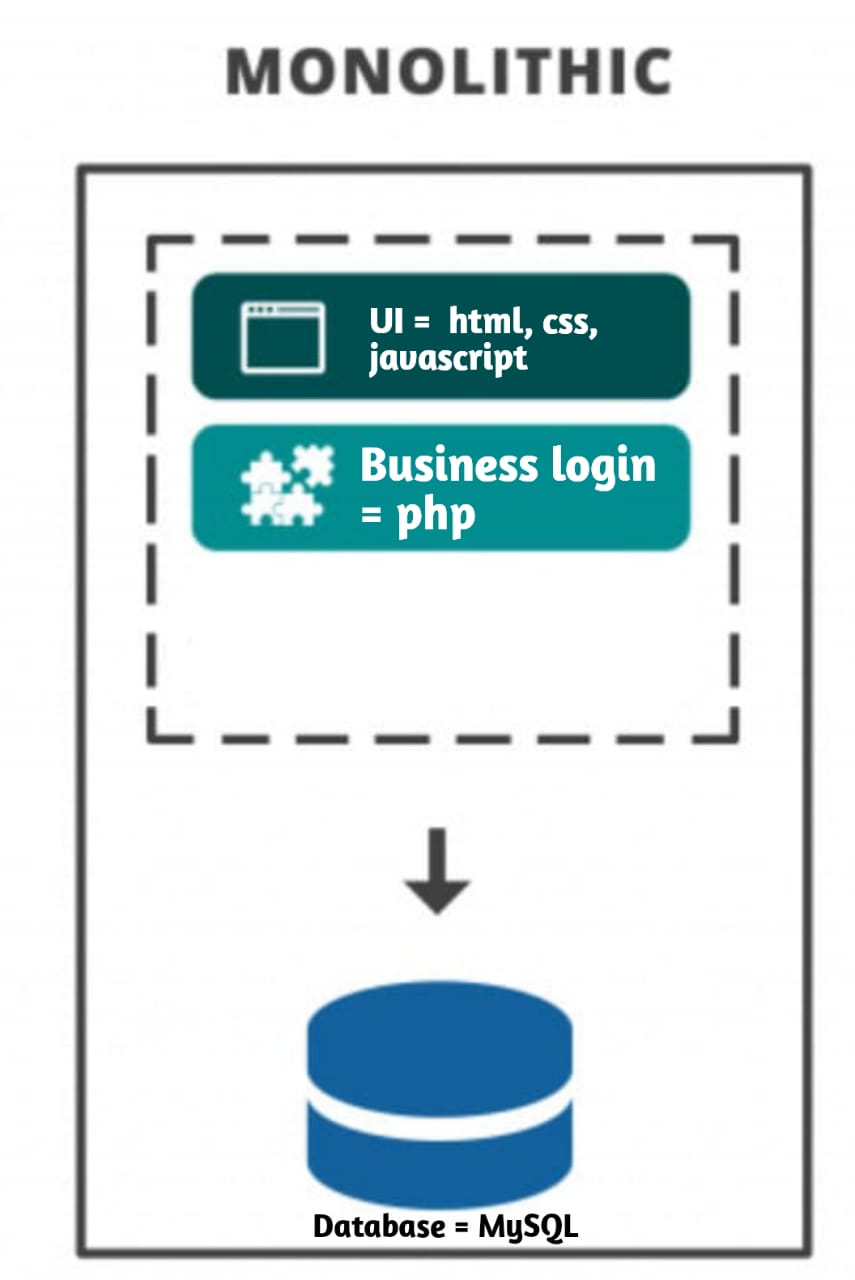
Kardinalitas dari desain basis data sistem klinik ini ialah :

1. Tabel user ke tabel janji temu : one to many (1:M) karena seorang pasien bisa membuat banyak janji temu.
2. Tabel user ke tabel resep obat : one to many (1:M) karena seorang pasien bisa mendapat beberapa resep obat.
3. Tabel user ke tabel pengisian form : one to many (1:M) karena seorang pasien dapat mengisi banyak formulir.

## **3.7 Arsitektur Pemrograman Web**

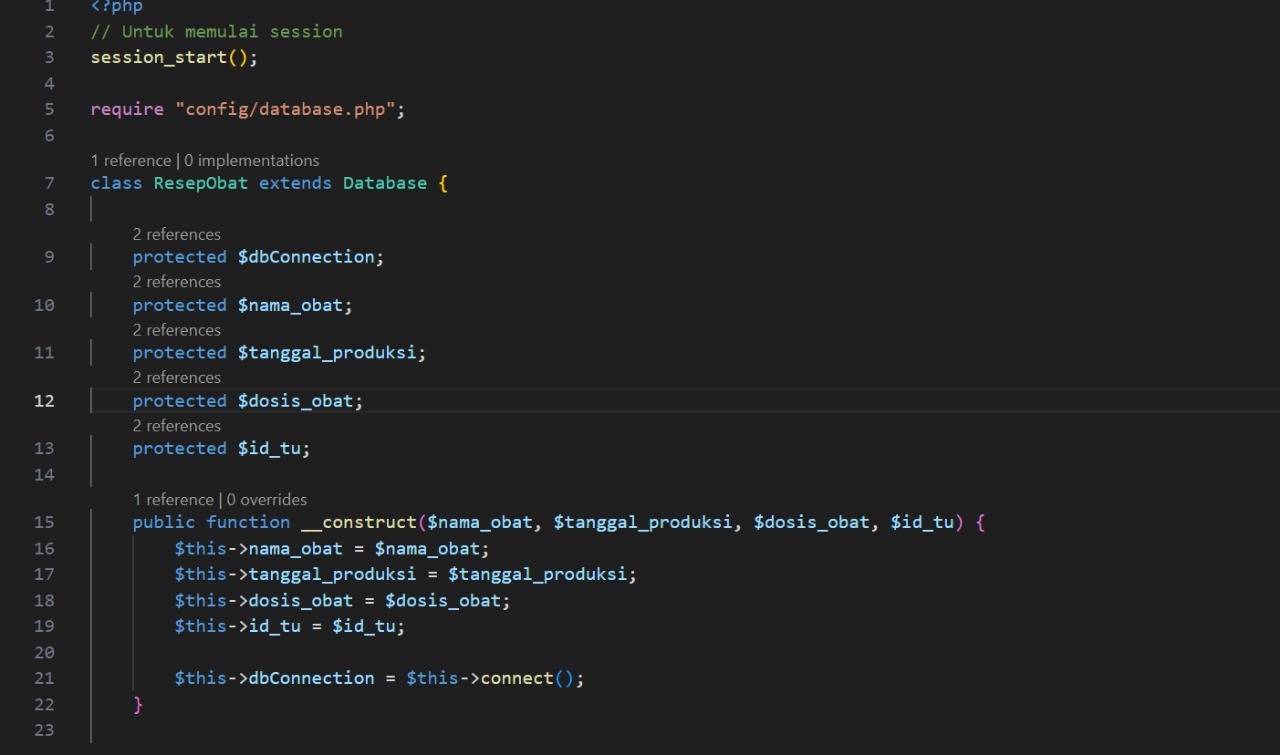
Aplikasi klinik ini di desain menggunakan arsitektur monolitik. Arsitektur monolitik adalah pendekatan pengembangan perangkat lunak di mana seluruh aplikasi dibangun sebagai satu kesatuan tunggal, yang terdiri dari komponen-komponen yang saling terkait dan berbagi kode sumber. Dalam arsitektur monolitik, seluruh basis kode aplikasi dikelola dalam satu repositori, memastikan proses pengembangan yang mudah.

Untuk UI/UX sistem ini menggunakan html, css dan juga JavaScript. Kemudian untuk Business login sistem ini menggunakan PHP, lalu untuk menyimpan database menggunakan MySQL.

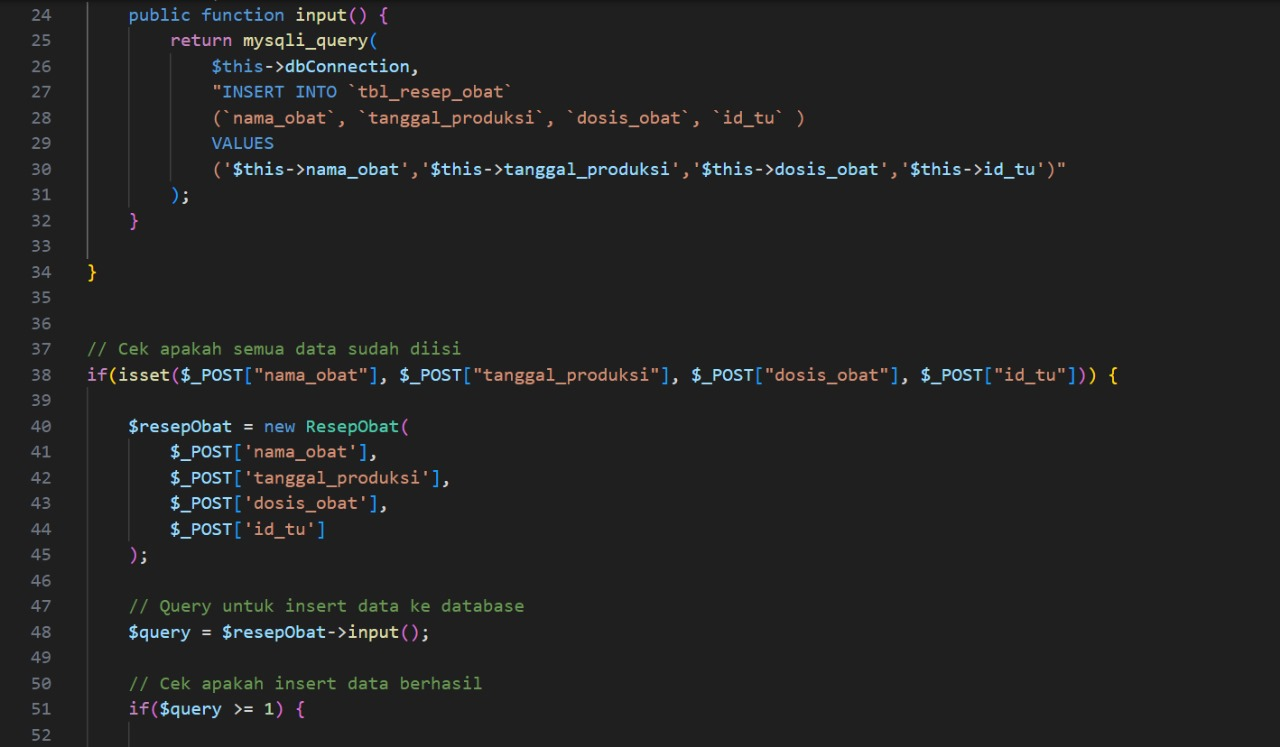


Gambar 29 Arsitektur Pemrograman Web

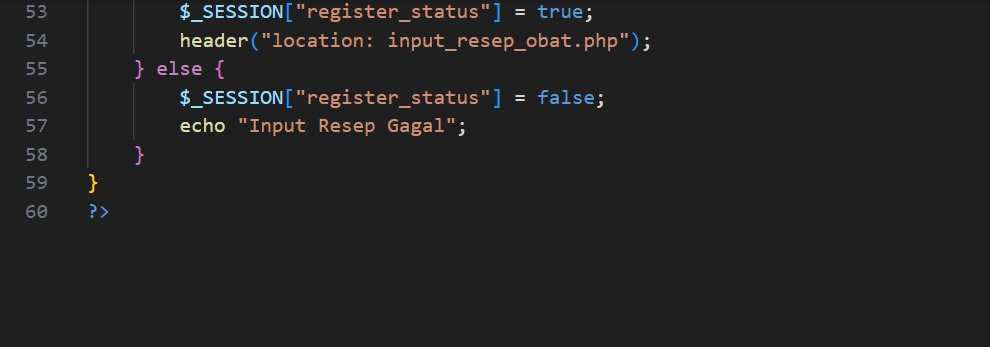
## **3.8 Implementasi Pemrograman Berorientasi Objek**



Gambar 30 Implementasi PBO 1



Gambar 31 Implementasi PBO 2



Gambar 32 Implementasi PBO 3

# **BAB IV HASIL DAN PEMBAHASAN**

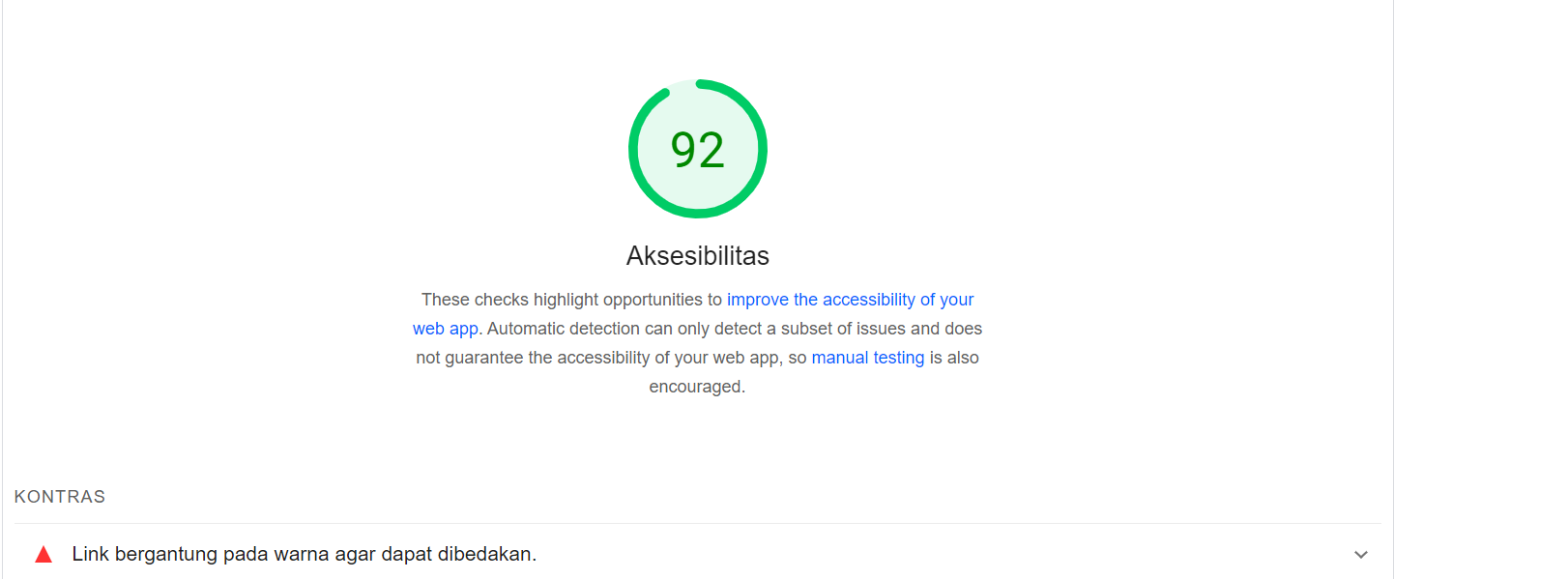
## **4.1 Pengujian Fungsional**

Pengujian fungsionalitas ini sudah kami tanyakan kepada klien kami, yaitu klinik pratama khazanah medika.

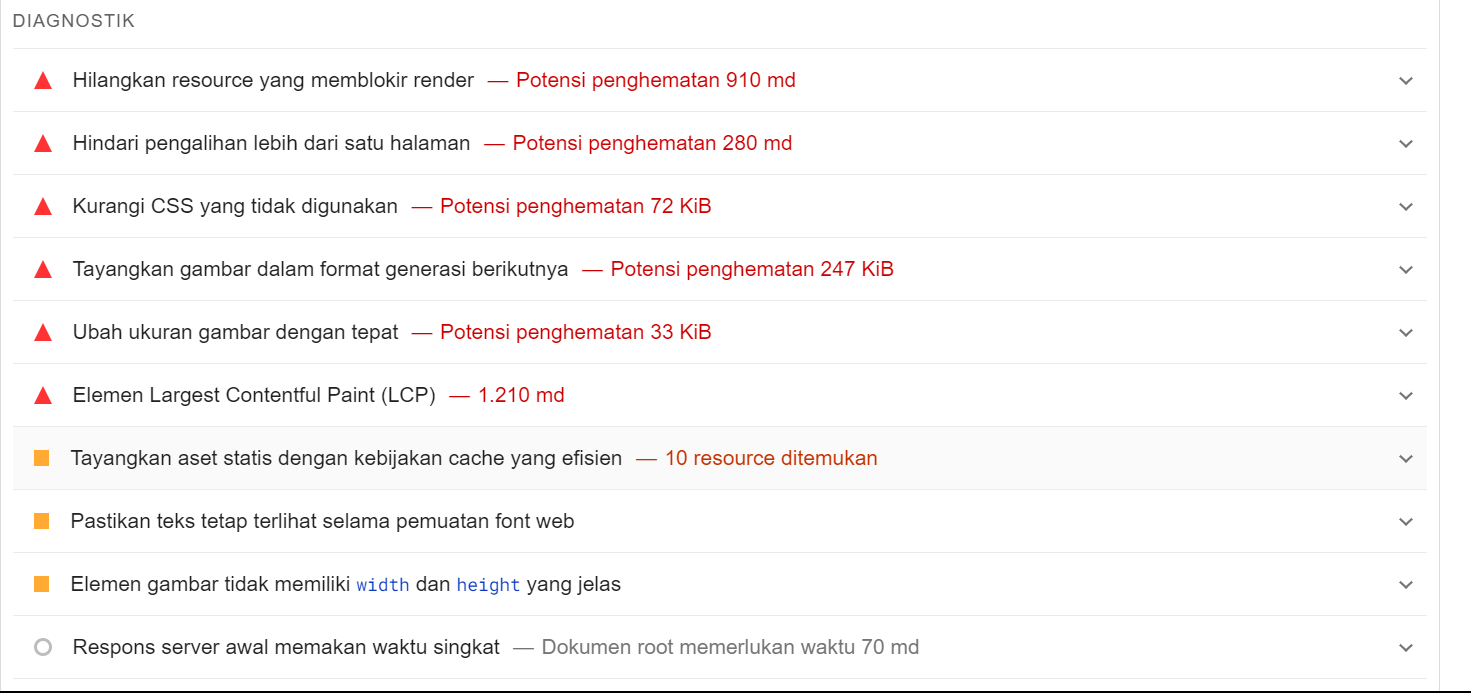
Tabel 13 Pengujian Fungsionalitas

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Fungsi | Dekripsi | Cara kerja fitur | Hasil pengujian | |
| Berhasil | Tidak |
| 1. | Login | Fungsi user dan admin untuk masuk ke sistem klinik | User harus mengisi form login seperti username dan password |  |  |
| 2. | Dashboard | Fungsi menampilkan beberapa menu yang tersedia | User membuka halaman dashboard dan memilih menu di dalamnya |  |  |
| 3. | Form Register | Fungsi untuk pasien mengisi data diri dan keluhan | User mengisi data sesuai form dan otomatis terinput rekam medisnya |  |  |
| 4. | Janji Temu | Fungsi untuk pasien membuat janji temu dengan dokter | Pasien mengisi tanggal dan jam untuk berobat ke klinik secara langsung |  |  |
| 5 | Melihat Resep Obat | Fungsi agar pasien dapat melihat resep obat di aplikasi | Pasien memilih menu resep obat di dashboard dan melihat resep obat |  |  |
| 6 | Menginput Resep Obat | Fungsi untuk resepsionis menginputkan resep obat untuk pasien | Resepsionis dapat menginput resep obat yang di dapat dari dokter ke dalam sistem |  |  |
| 7 | View Rekam Medis | Proses resepsionis melihat rekam medis pasien | Resepsionis melihat rekam medis pasien di dalam menu rekam medis |  |  |
| 8 | Profile | Fungsi untuk semua user mengganti data diri dan juga password | User mengganti data diri dan juga password di dalam fitur profile |  |  |
| 9 | Membuat Akun | Proses pasien membuat akun agar bisa login | User mengisi form buat akun |  |  |
| 10 | Ganti Password | Proses user untuk mengganti password | User mengganti password di dalam fitur profile |  |  |
| 11 | Logout | Proses untuk semua user keluar dari sistem klinik | User memilih menu logout |  |  |

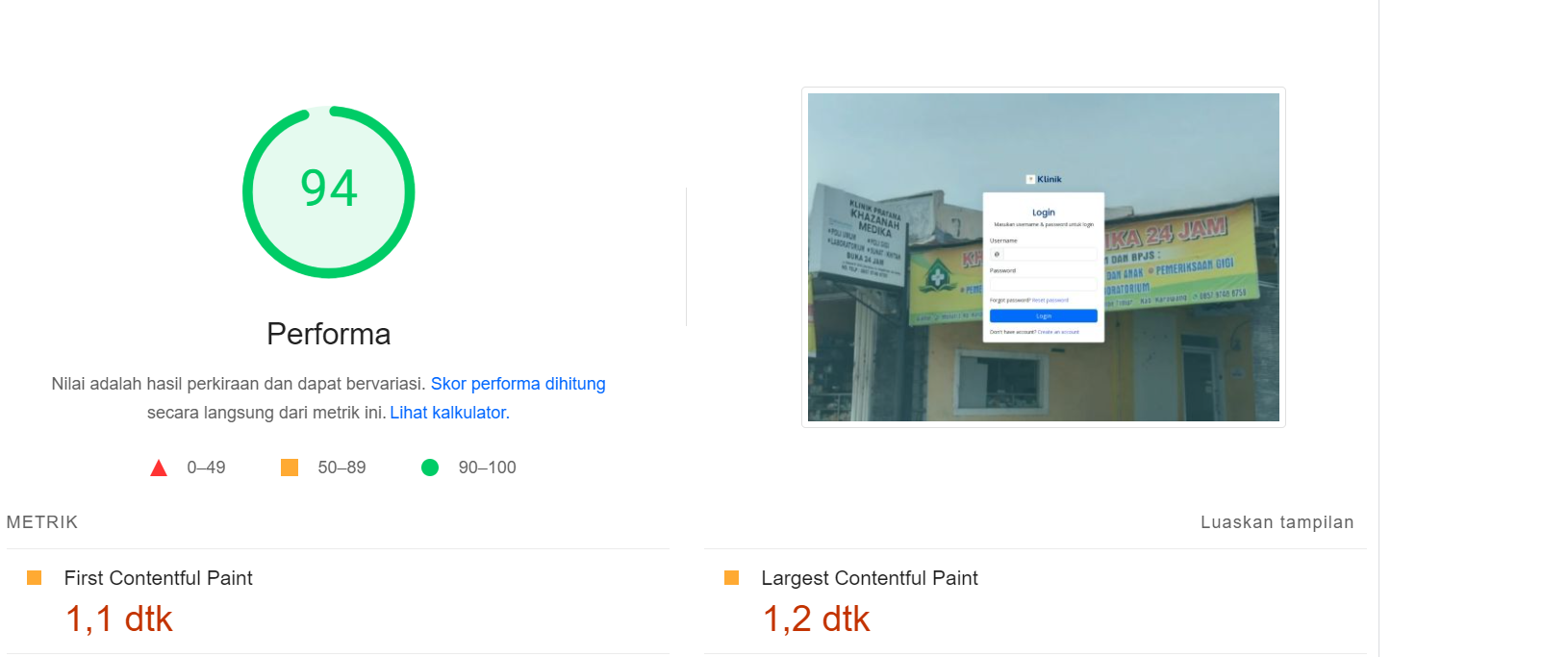
## **4.2 Pengujian Performa**

****

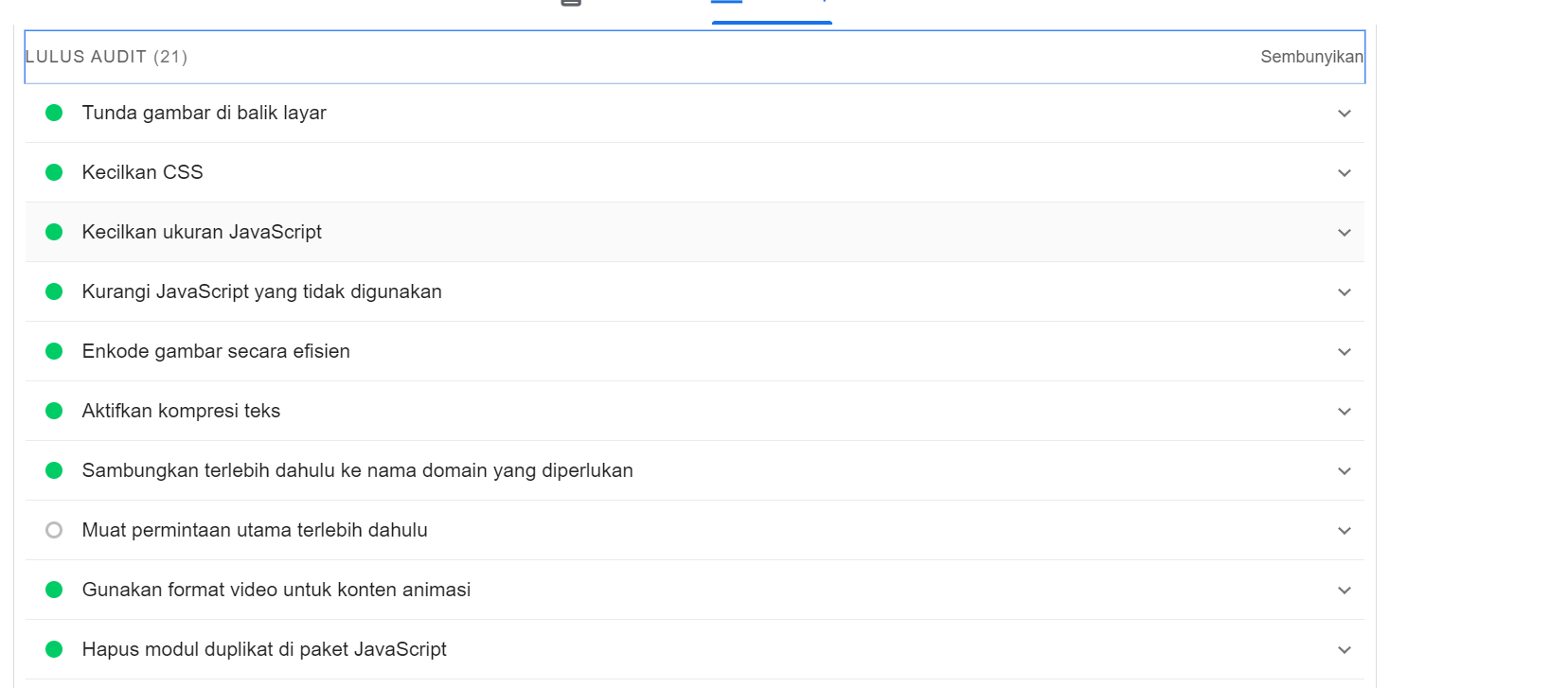
Gambar 33 Pengujian Performa



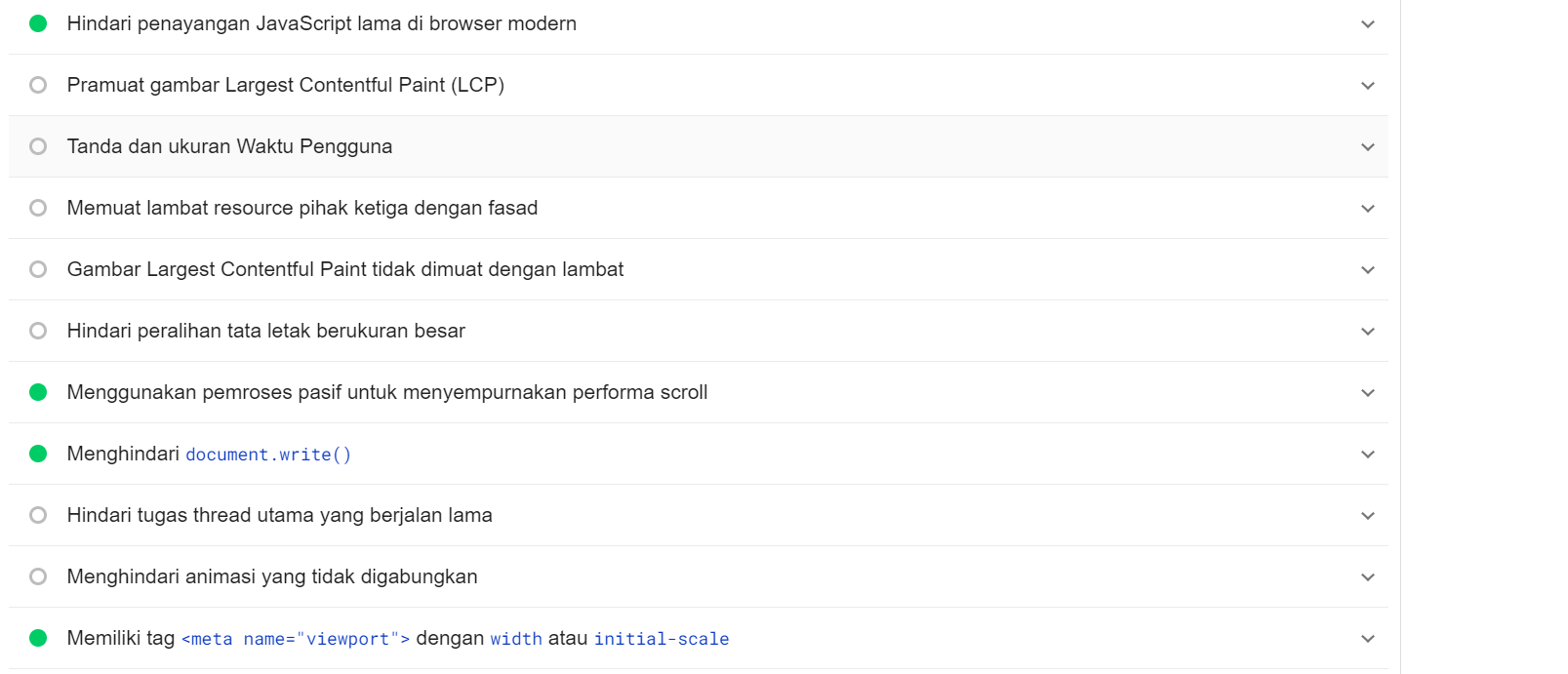
Gambar 34 Pengujian Performa



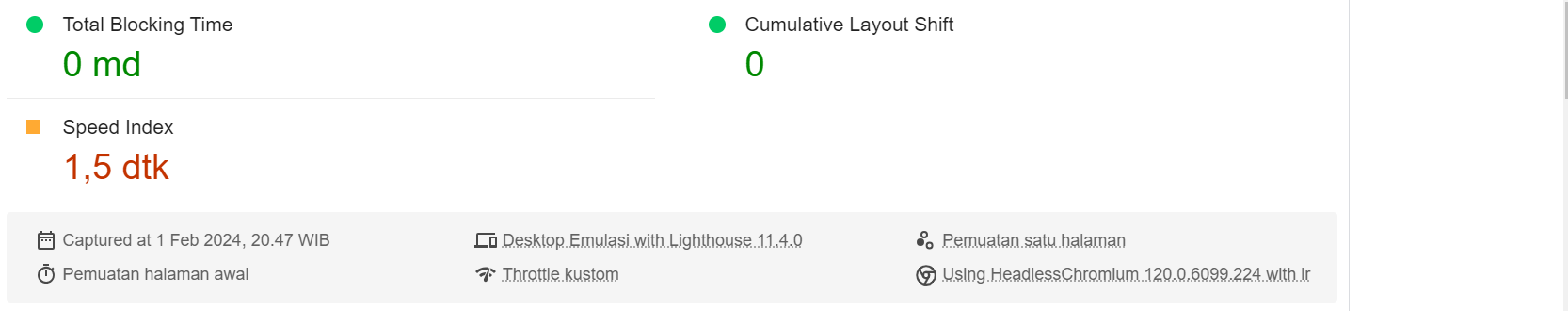
Gambar 35 Pengujian Performa



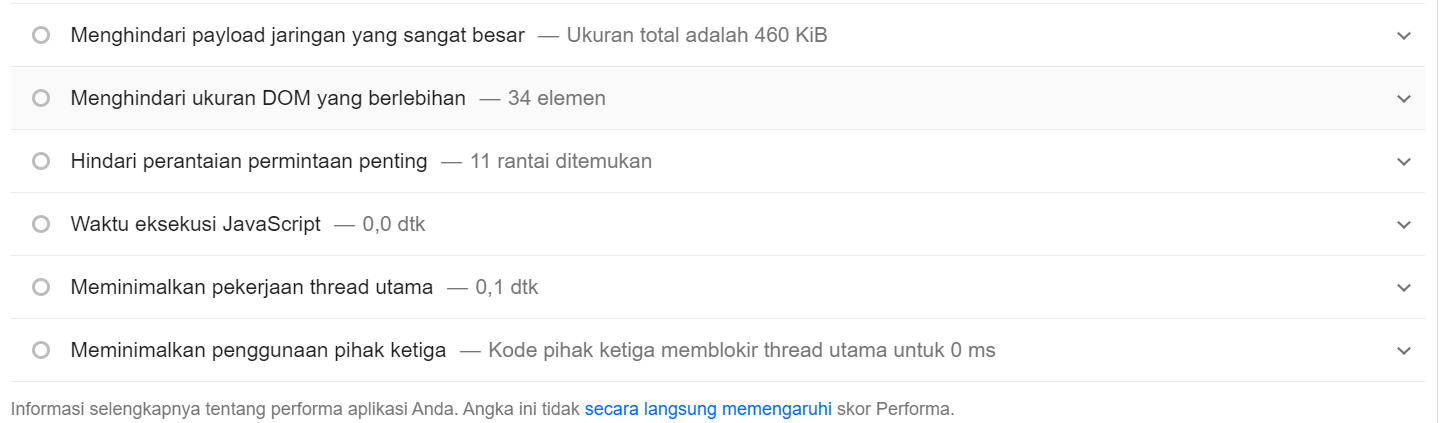
Gambar 36 Pengujian Performa



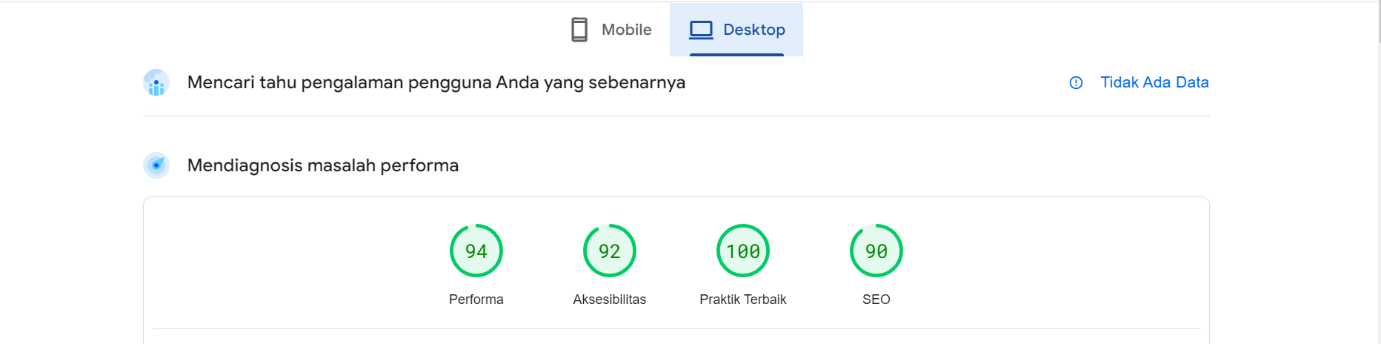
Gambar 37 Pengujian Performa



Gambar 38 Pengujian Performa

****

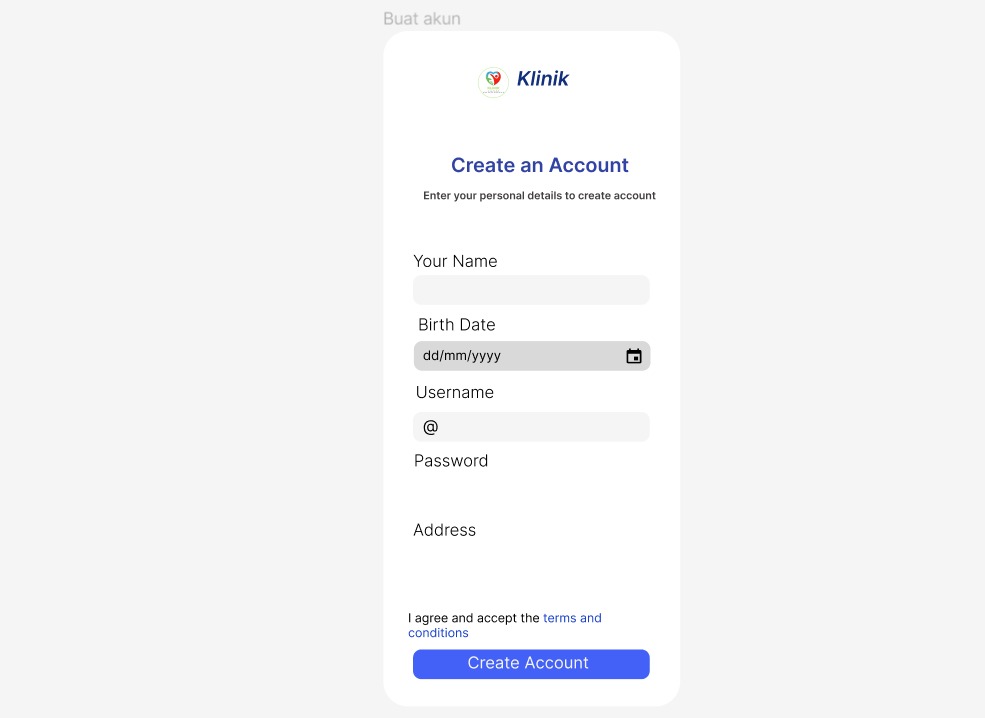
Gambar 39 Pengujian Performa

****

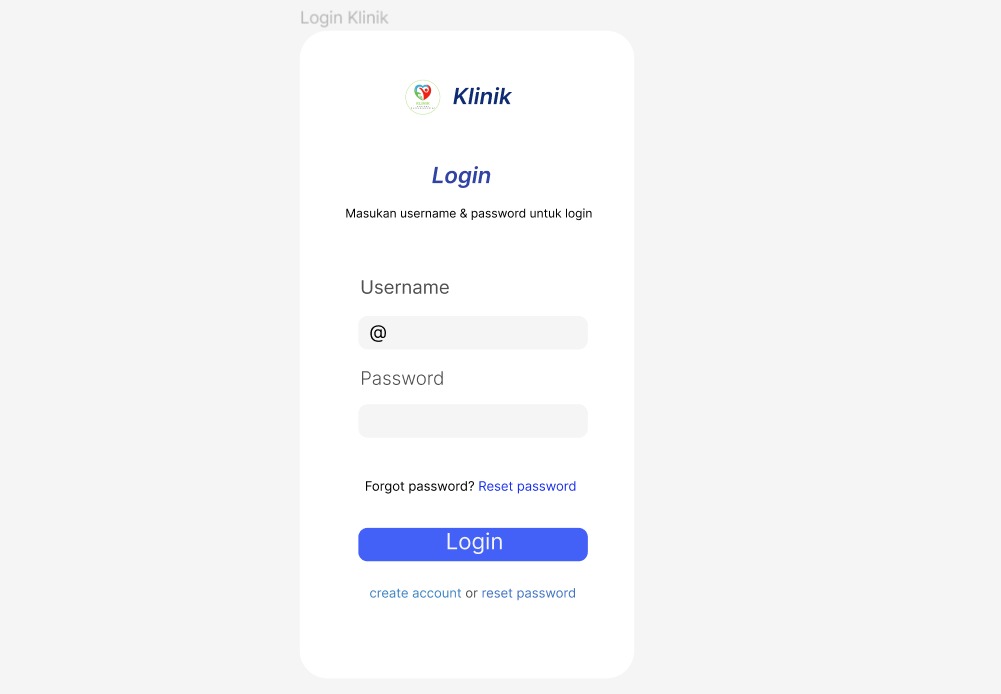
Gambar 40 Pengujian Performa

## **4.3 Implementasi Sistem**

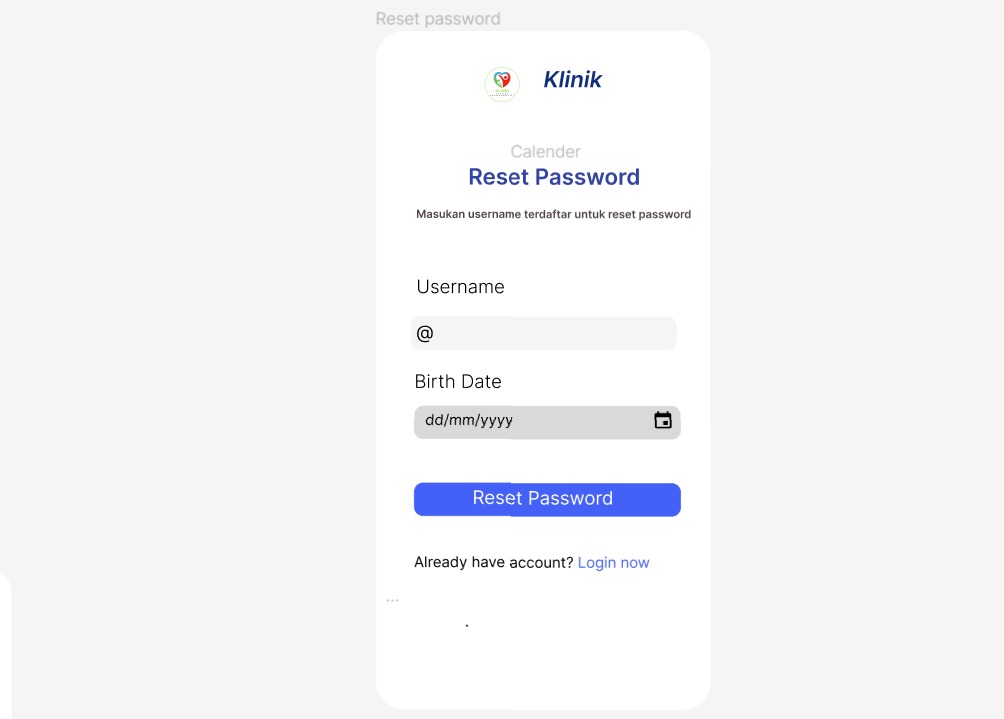
Implementasi Sistem ini berisi gambaran UI dari sistem web klinik ini.



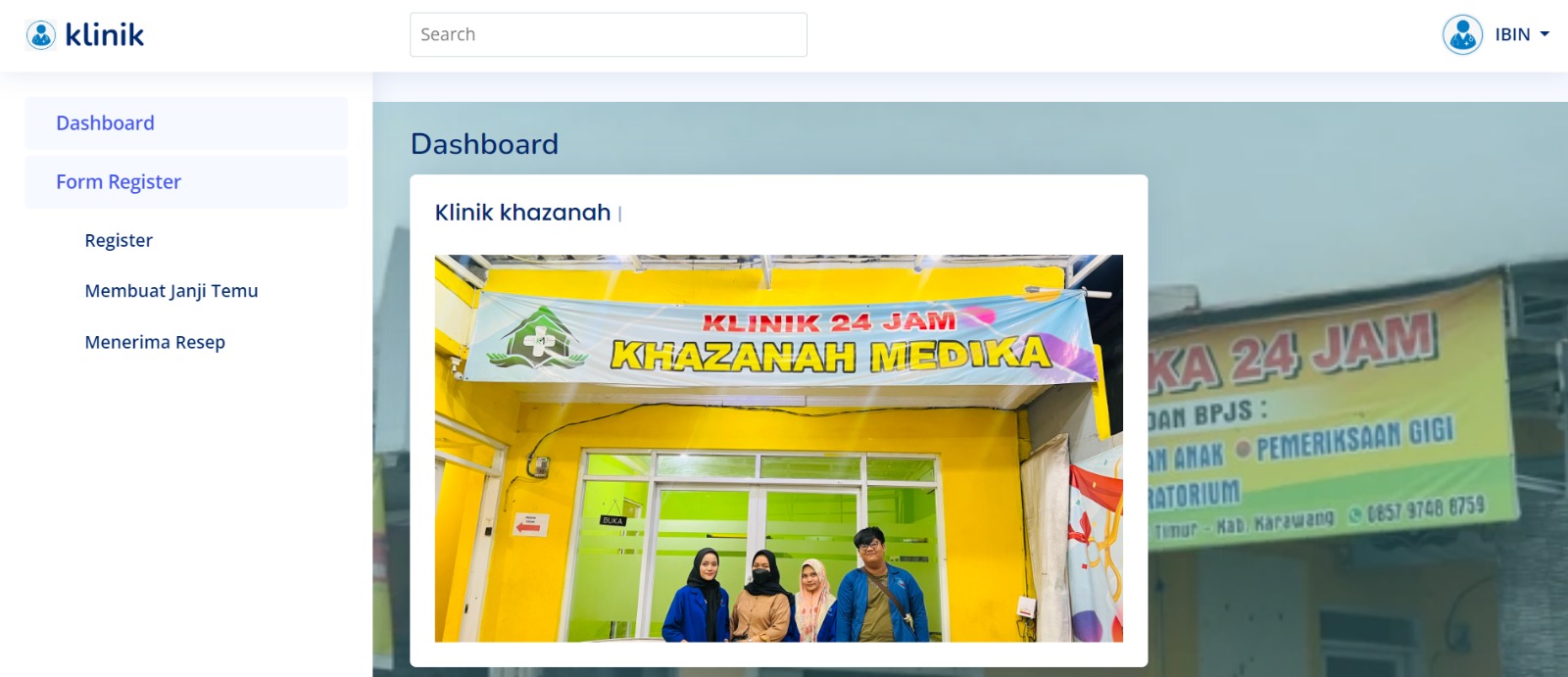
Gambar 41 Implementasi Sistem Buat Akun



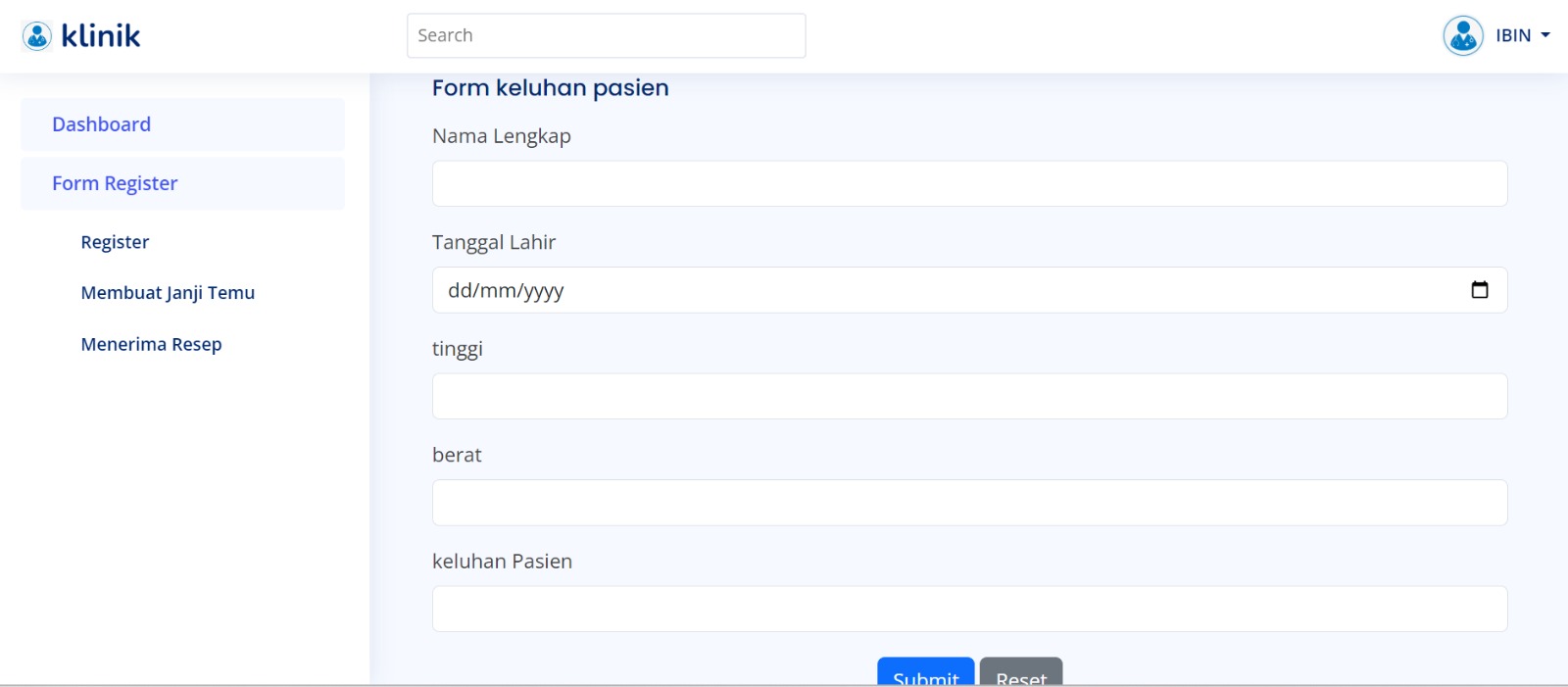
Gambar 42 Implementasi Sitem Login Klinik



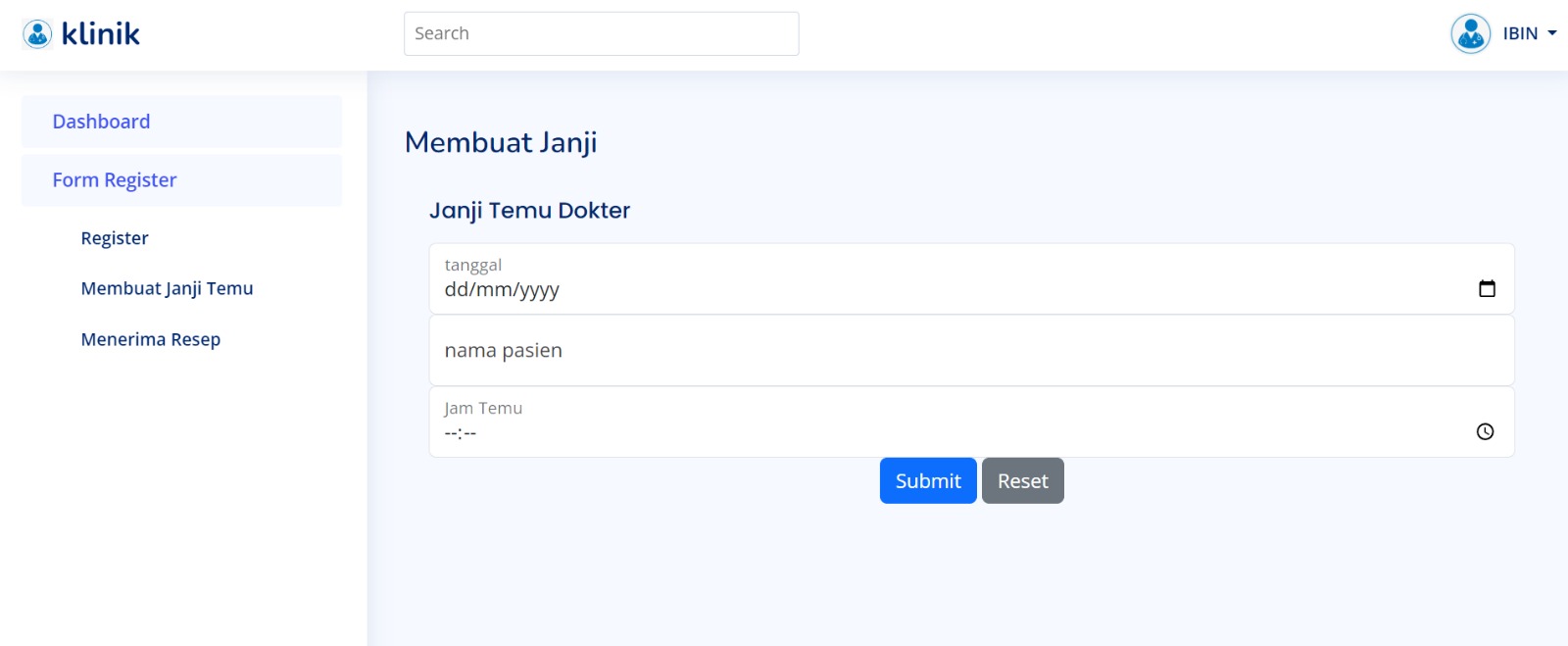
Gambar 43 Implementasi Sistem Reset Password



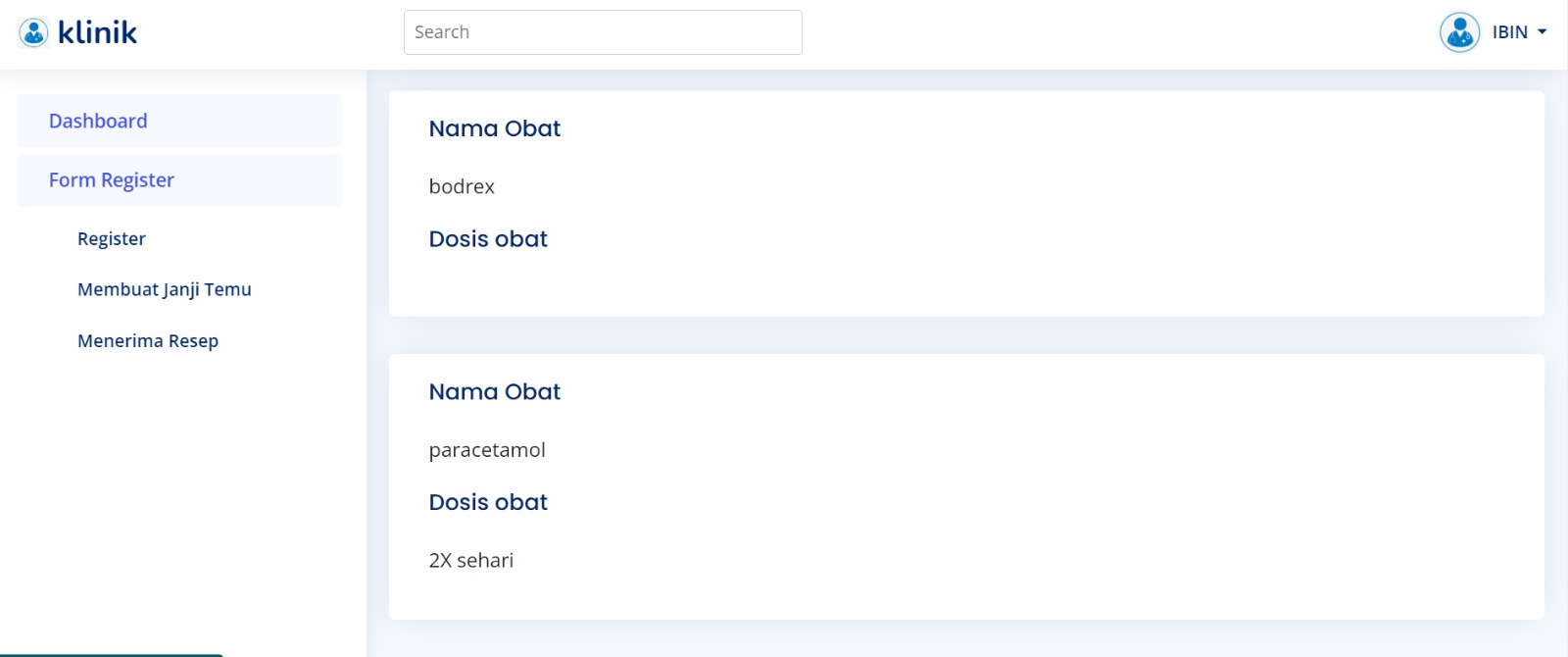
Gambar 44 Implementasi Sitem Dashboard User



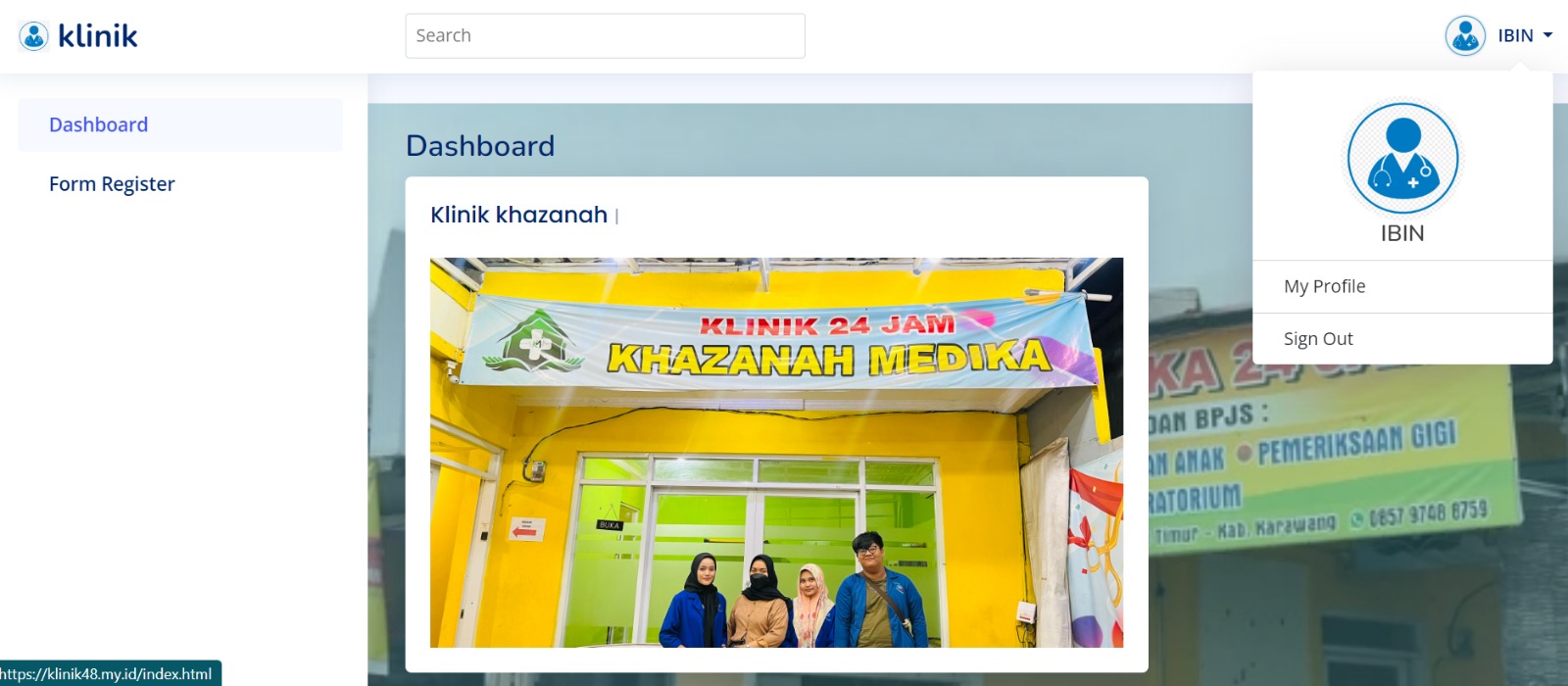
Gambar 45 Implementasi Sistem Form Register User



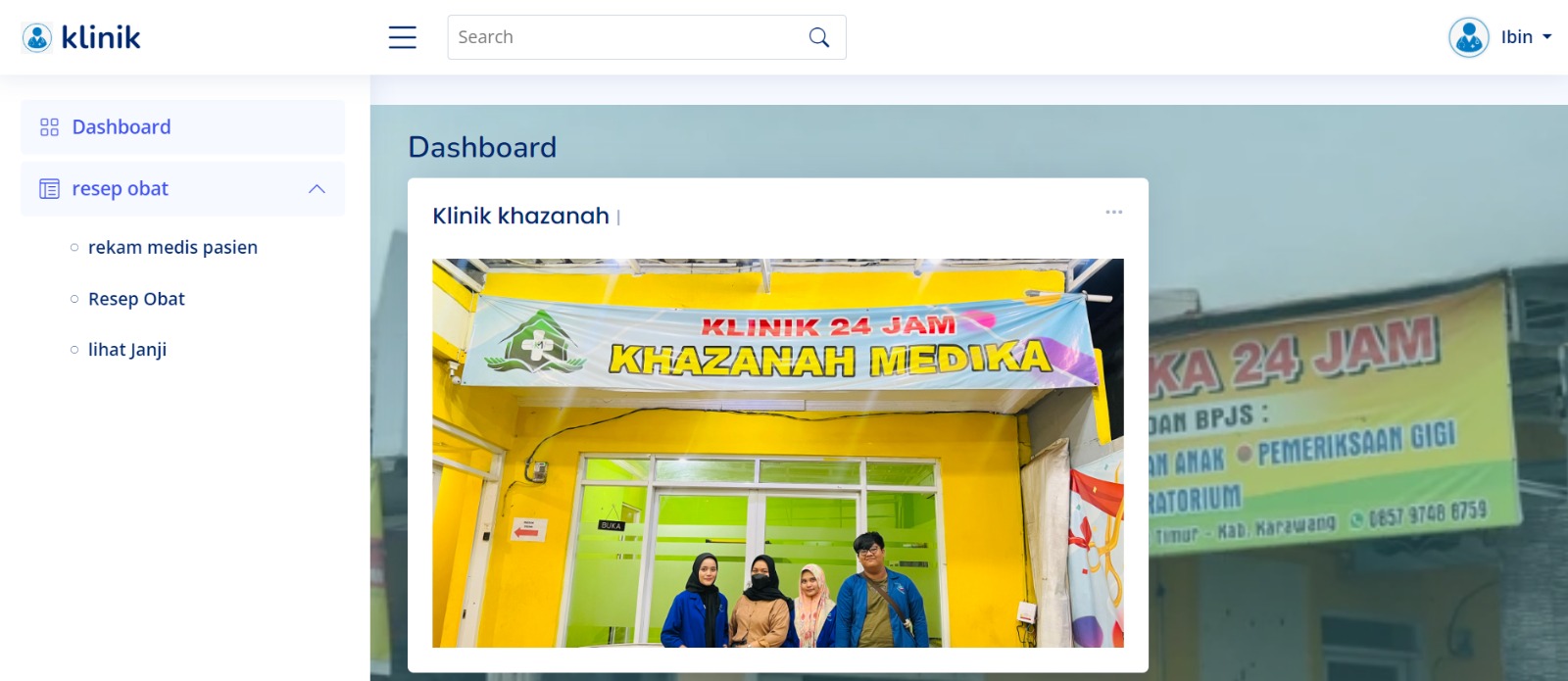
Gambar 46 Implementasi Sistem Membuat Janji Temu User



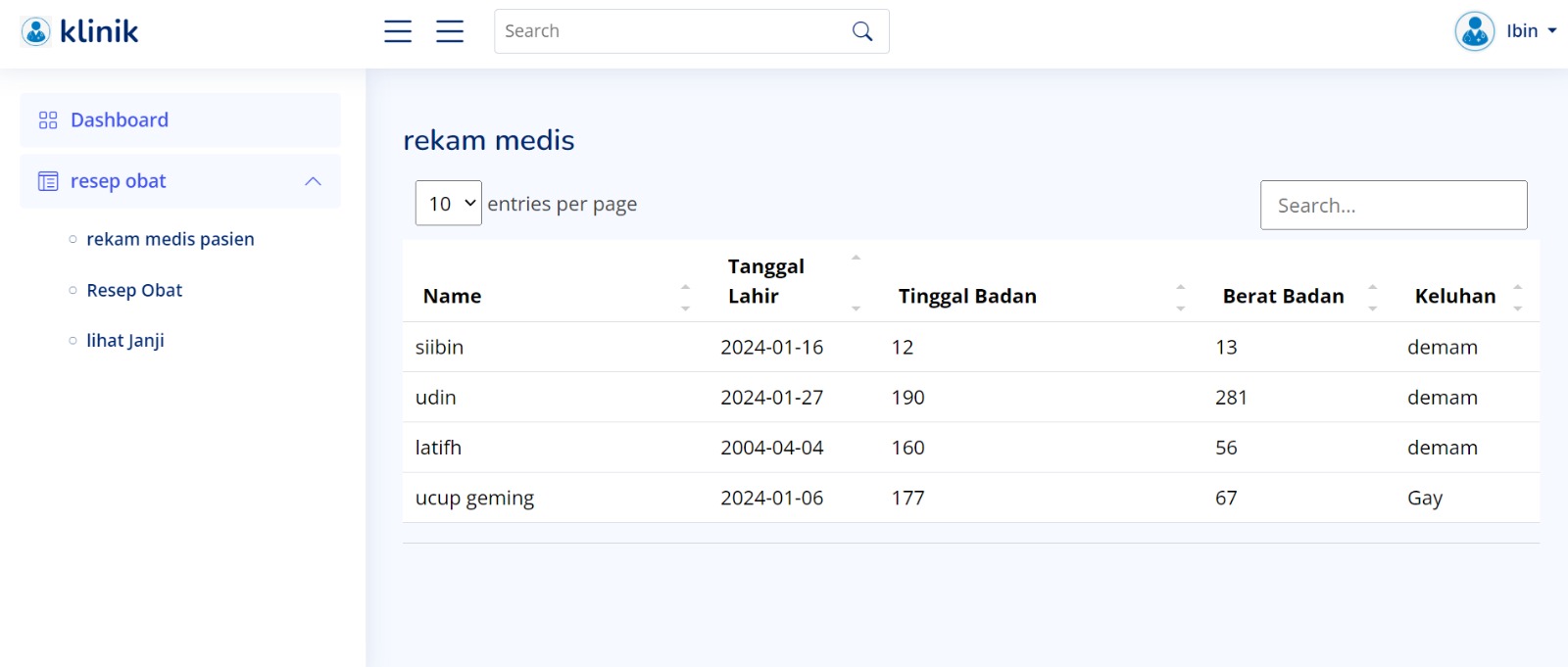
Gambar 47 Implementasi Sistem Menerima Resep Obat User



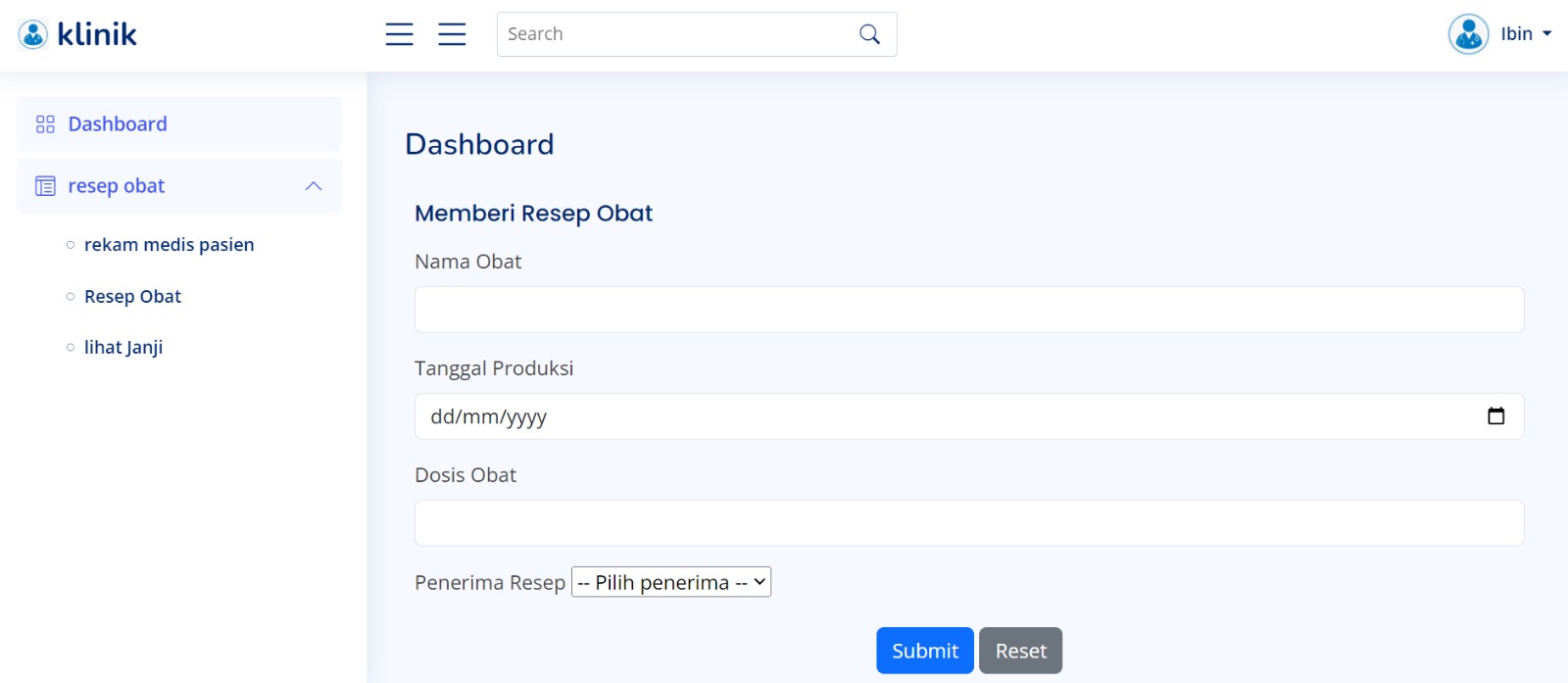
Gambar 48 Implementasi Sistem Dashboard My Profile User



Gambar 49 Implementasi Sistem Dashboard Admin



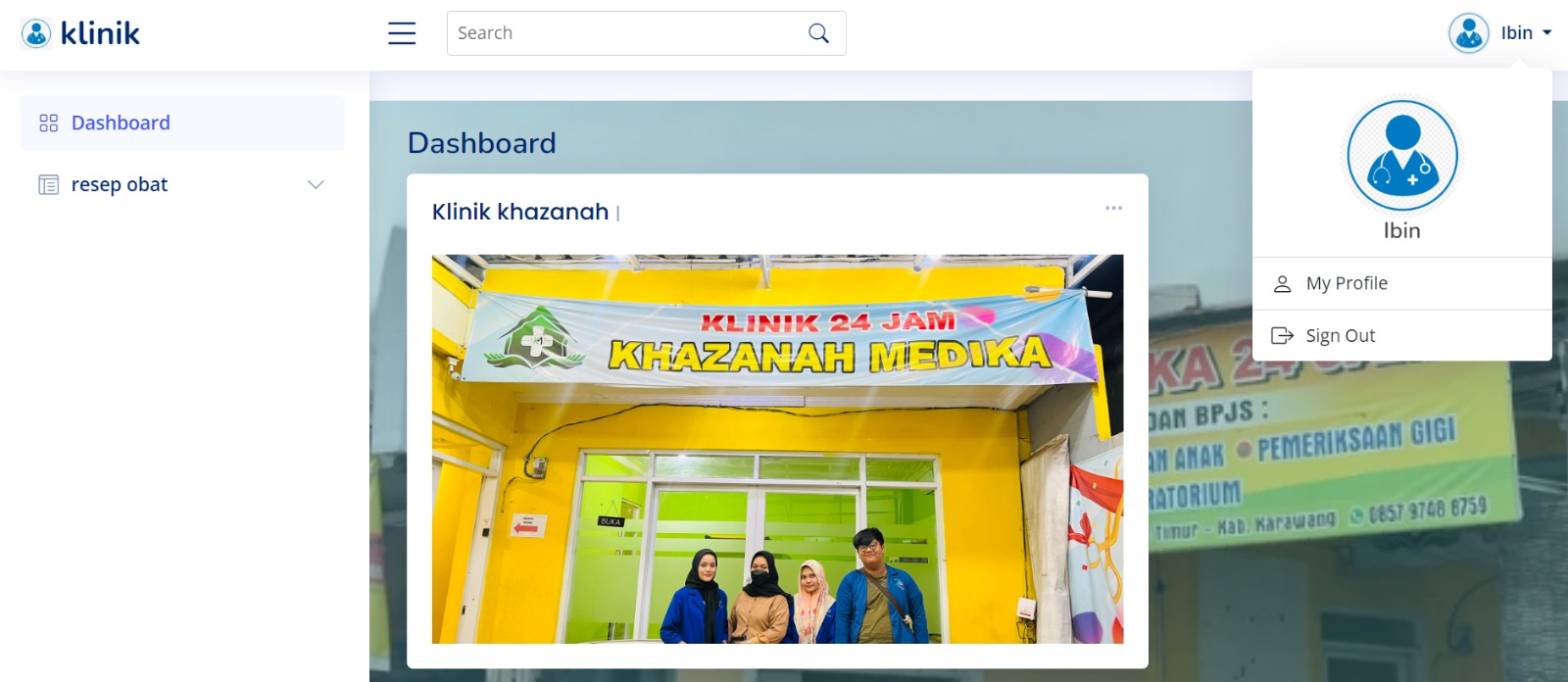
Gambar 50 Implementasi Sistem Rekam Medis Admin



Gambar 51 Implementasi Sistem Resep Obat Admin



Gambar 52 Implementasi Sistem Janji Temu Admin

Gambar 53 Implementasi Sistem Dashboard My Profile Admin

## **4.4 Analisis Hasil Pengujian**

Analisis hasil pengujian dari pengujian performa, yaitu :

1. Pengujian performa mendapatkan 94%.
2. Pengujian aksesibilitas mendapatkan 92%.
3. Pengujian Praktik Terbaik mendapatkan 100%.
4. Pengujian SEO mendapatkan 90%.

Dari hasil pengujian diatas, dapat disimpulkan bahwa sistem :

1. Aplikasi ini mampu menambah data ke dalam sistem.
2. Registrasi, janji temu dan resep obat dapat dijalankan di dalam sistem.
3. Melihat rekam medis hanya dapat dilakukan oleh resepsionis dan pemilik klinik.
4. Membuat sistem dimana dapat membuat janji temu dengan dokter secara online.

# **BAB V KESIMPULAN**

## **5.1 Kesimpulan**

Dalam pengembangan sistem aplikasi untuk Klinik Pratama Khazanah Medika, kami berhasil merancang dan mengimplementasikan sejumlah fitur yang mempermudah dan meningkatkan efisiensi operasional klinik. Proses pengembangan melibatkan pemodelan dan analisis kebutuhan pengguna, desain sistem, serta implementasi fitur-fitur utama. Berikut adalah beberapa kesimpulan utama dari proyek ini :

1. Dapat mengatur jadwal janji temu dengan dokter secara online melalui aplikasi.
2. Penggunaan sistem ini membantu klinik mengelola jadwal dokter, pendaftaran pasien, dan rekam medis secara lebih terstruktur dan efisien.
3. Sistem ini dirancang dengan memperhatikan keamanan data pasien dan privasi informasi medis.

Dengan implementasi sistem aplikasi ini, Diharapkan akan memberikan manfaat besar bagi klinik dan memberikan dasar untuk kemajuan teknologi dalam bidang pelayanan kesehatan.

## **5.2 Saran**

Saran kami tidaklah banyak, hanya saja kami meminta masukan atau kritikan yang membangun, karena kami tahu bahwa dalam laporan ini masih terdapat banyak kesahan dan kekurangan.

# **DAFTAR PUSTAKA**

Jantce TJ Sitinjak, D. D., Maman, ., & Suwita, J. (2020). Analisa Dan Perancangan Sistem Informasi Administrasi Kursus Bahasa Inggris Pada Intensive English Course Di Ciledug Tangerang. *Insan Pembangunan Sistem Informasi Dan Komputer (IPSIKOM)*, *8*(1). https://doi.org/10.58217/ipsikom.v8i1.164

Pelawi, D. (2013). *Pembuatan Sistem Informasi… (Dewan Pelawi)*. *9*, 213–224.

Pembelajaran, H. (2013). *Bab 10*. 283–320.

Ristianti, E. M. (2019). Konsep Pemrograman Berorientasi Object (Pbo) Pada Delphi. *Fakultas Komputer*, 18.

Sari, E. D. (2018). *Tinjauan Pustaka Penelitian terkait dengan Metode algortma*. *2015*, 5–11.

# **LAMPIRAN**

## **Kode Program**

<https://drive.google.com/file/d/1Ypfx9qN5ZNhl0exK0Uc-ixPhQ8rY7mUP/view?usp=drive_link>

## **Hasil Uji Coba**

User : <https://drive.google.com/file/d/1Y5mlvoNexuLiicMjXkRrdg0G-aWh0LN3/view?usp=drive_link>

Admin : <https://drive.google.com/file/d/1Y-dyycqwVJlbe6iZcGkRLv0IExkzpGTp/view?usp=drive_link>

## **Timeline**

<https://docs.google.com/spreadsheets/d/1e50yfJ54rhu5aid431AR9y83U_lIuO3F/edit?usp=drive_link&ouid=103587776532242897847&rtpof=true&sd=true>

## **Angket Penelitian**

<https://docs.google.com/document/d/1tMfa_cnuS7CAhvR1saWJ-sHdwevr2IC-/edit?usp=drive_link&ouid=103587776532242897847&rtpof=true&sd=true>