

# **ANALISIS DAN DESAIN PERANGKAT LUNAK**

## **LAPORAN ANALISIS APLIKASI RUMAH SAKIT (UJIAN AKHIR SEMESTER)**

### **Kelompok 3 :**

- 1. Putra Satria Nagara Mahmud (Project Manager)**
- 2. Arphan Maulana Firdaus (Back-end)**
- 3. Rafi Sidiqamrullah (Front-end)**
- 4. Ramadi Haykal (UI/UX)**
- 5. Muhamad Lingga Sepria Perdana (DBA)**

## **1. Pendahuluan**

Pada Rumah Sakit Modern ini, sedang mengalami berbagai tantangan dalam pengelolaan dalam operasional dan pelayanan kesehatan, dan di era sekarang yaitu era digitalisasi, kebutuhan akan sistem sebuah informasi yang efisien menjadi semakin mendesak untuk memastikan kelancaran proses operasional, penyimpanan suatu data dan dari segi pelayanan yang cepat menentukan sebuah kualitas kepada pasien.

Untuk mengikuti alur di era digitalisasi ini. Yaitu membuat sebuah aplikasi untuk pendataan pasien, Menentukan Janji Temu antara Pasien dengan Dokter dll. Dengan implementasi pada aplikasi ini, diharapkan rumah sakit dapat meningkatkan beberapa hal seperti meningkatkan efisiensi kerja, mengurangi kesalahan dalam administratif dan memberikan pelayanan yang lebih baik kepada pasien.

Untuk aplikasi ini memiliki fokus utama, seperti :

- a. Meningkatkan pengalaman dalam user experience (UX) pada pasien melalui layanan pendaftaran dan beberapa fitur lainnya
- b. Memberikan beberapa kemudahan untuk mengakses baik dokter, pasien hingga admin untuk mengelola data
- c. memastikan dalam bentuk security (keamanan) dan kerahasiaan data pasien
- d. Mengoptimalkan beberapa manajemen keuangan dan inventaris rumah sakit.

## **2. Analisis Kebutuhan**

### **2.1 Kebutuhan Fungsional**

#### **● Pendaftaran Pasien**

- A.** Fitur Pendaftaran Pasien baru atau lama menggunakan metode **Login** dan **Register**
- B.** Kemampuan pencatatan informasi pasien seperti identitas, riwayat penyakit, diagnosa dll
- C. AKTOR : Pasien dan Admin**

#### **● Penjadwalan Dokter dan Janji Temu**

- A. Penjadwalan jam kerja dokter
- B. Fasilitas untuk pasien membuat, membatalkan, atau mengubah janji temu
- C. **AKTOR : Pasien, Dokter, Admin Rumah Sakit**

- **Manajemen Rekam Medis Elektronik (RME)**

- A. Penyimpanan dan Pengelolaan rekam medis pasien
- B. Aksesibilitas data oleh dokter sesuai otorisasi.
- C. Catatan perkembangan pasien oleh tim medis
- D. **AKTOR : Dokter dan Admin Rumah Sakit**

- **Manajemen Keuangan, Administrasi, Inventaris dan Logistik**

- A. Penagihan Otomatis untuk layanan kesehatan.
- B. Laporan pendapatan dan pengeluaran rumah sakit
- C. Integrasi dengan sistem pembayaran digital
- D. Pelacakan Stok Obat dan Peralatan Medis
- E. Pengingat Untuk Pengadaan Stok
- F. **AKTOR : Admin Rumah Sakit**

### **Diagram Use Case :**

Berikut adalah isi dari beberapa diagram use case pada aplikasi rumah sakit :

#### **1. Aktor**

- **Pasien**

Disini, pasien akan terhubung dari pendaftaran, penjadwalan janji temu dan notifikasi

- **Dokter**

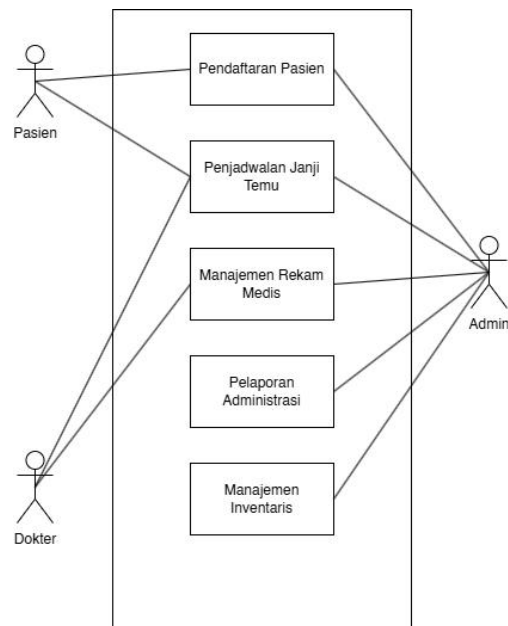
Pada Dokter akan terhubung ke beberapa kegiatan mulai dari penjadwalan janji temu, manajemen rekam medis dan notifikasi.

- **Admin**

Untuk admin itu sendiri akan mengelola beberapa fitur yang ada pada aplikasi seperti pendaftaran, penjadwalan, manajemen rekam, manajemen keuangan dan administrasi baik inventaris maupun pelaporan.

Berikut adalah Lampiran dari Gambar Use Case :

#### **1. Use Case Diagram :**



Disini ada 3 Aktor seperti yang sudah dijelaskan, berikut hasil dari koneksi antara 3 aktor.

#### A. Pasien

Pada pasien menggunakan One-To-Many yaitu interaksi antara Janji Temu (Penjadwalan) dan RekamMedis (Memajemen Fitur Rekam Medis)

#### B. Dokter

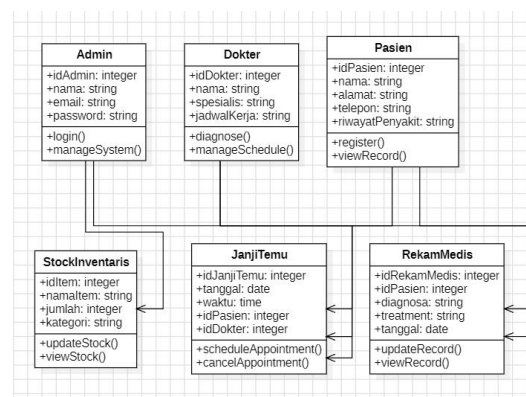
Sama Seperti Pasien, Menggunalam One-To-Many dan untuk interaksi sama seperti pasien yaitu memiliki hubungan dengan fitur JanjiTemu dan RekamMedis

#### C. Admin

Untuk Admin bertugas mengontrol beberapa fitur seperti : Pendaftaran, Memanajemen Inventaris Mengelola Beberapa Data pada RekamMedis dan JanjiTemu.

Untuk Rekam Medis dan Janji Temu, Menggunakan Many to One dikarenakan terhubung baik pasien dan dokter.

## 2. Class Diagram



Berikut ini merupakan penjelasan untuk Class Diagram :

**Admin** : Bertanggung jawab atas pengelolaan sistem.

**Pasien** : Terhubung dengan Class JanjiTemu dan Rekam Medis.

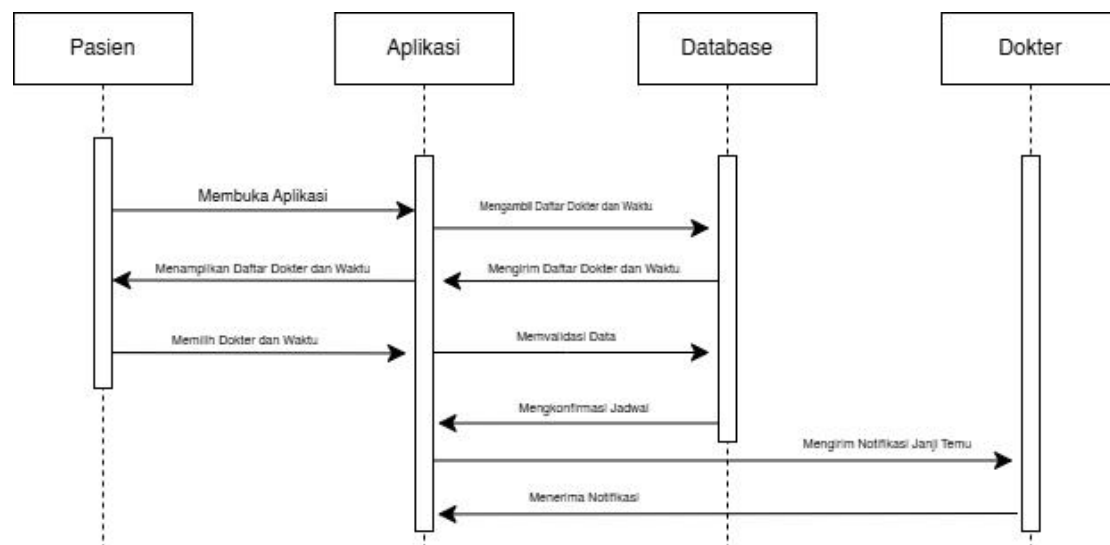
**Dokter** : Mengelola RekamMedis dan JanjiTemu

**JanjiTemu** : Menyimpan Data Jadwal Pasien dan Dokter

**RekamMedis** : Berisi Informasi Medis Pasien

**StokInventaris** : Mengelola data Inventaris Pada Aplikasi Rumah Sakit

### 3. Sequence Diagram



Untuk Sequence Diagram ini, Pasien Membuka aplikasi dan memilih fitur penjadwalan janji temu pada dokter, otomatis aplikasi akan mengirimkan pendataan ke database untuk mengambil sebuah data dan waktu yang tersedia. Untuk Database akan mengembalikan data ke aplikasi dan aplikasi akan menampilkan data kepada pasien. Setelah itu Pasien akan memilih dokter, waktu dan mengkonfirmasi janji temu pada dokter.

Setelah itu aplikasi akan memvalidasi data dan menyimpan ke database dan akan memunculkan notifikasi pada dokter.

### 2.2 Kebutuhan Non-Fungsional

- **Keamanan Data :**

Enkripsi data pasien untuk menjaga kerahasiaan dan mengimplementasikan sistem otentikasi berlapis untuk akses

- **Kinerja :**

Untuk Waktu Respon dari sebuah aplikasi kurang dari 2 detik dan untuk sistem hanya mampu menangani hingga 10.000 pengguna aktif secara bersamaan

- **Skabilitas :**

Pada Sistem ini, akan dirancang untuk dikembangkan seiring pertumbuhan rumah sakit dalam arti lain yaitu mengupgrade

### 3. Identifikasi Tim Pelaksana

#### 3.1. Programmer

**Deskripsi:** Programmer bertanggung jawab untuk mengembangkan dan mengintegrasikan fitur-fitur aplikasi sesuai dengan kebutuhan fungsional yang telah ditentukan. Tim programmer terdiri dari developer frontend dan backend yang bekerja secara kolaboratif.

Untuk tugas dari Programmer adalah :

- Mengembangkan sebuah antarmuka atau interface pada aplikasi secara intuitif dan menggunakan beberapa framework modern seperti untuk aplikasi mobile menggunakan react atau untuk web yaitu laravel
- Membuat sebuah API dan Backend untuk mendukung sebuah pengelolaan data
- Mengintegrasikan beberapa fitur seperti pendaftaran pasien, manajemen janji temu dan pelaporan
- Melakukan Debugging dan pengujian aplikasi

#### 3.3. Database Administrator

**Deskripsi:** Database Administrator (DBA) bertanggung jawab untuk mendesain, mengimplementasikan, dan memelihara basis data yang efisien, aman, dan dapat diandalkan untuk aplikasi rumah sakit.

**Tugas:**

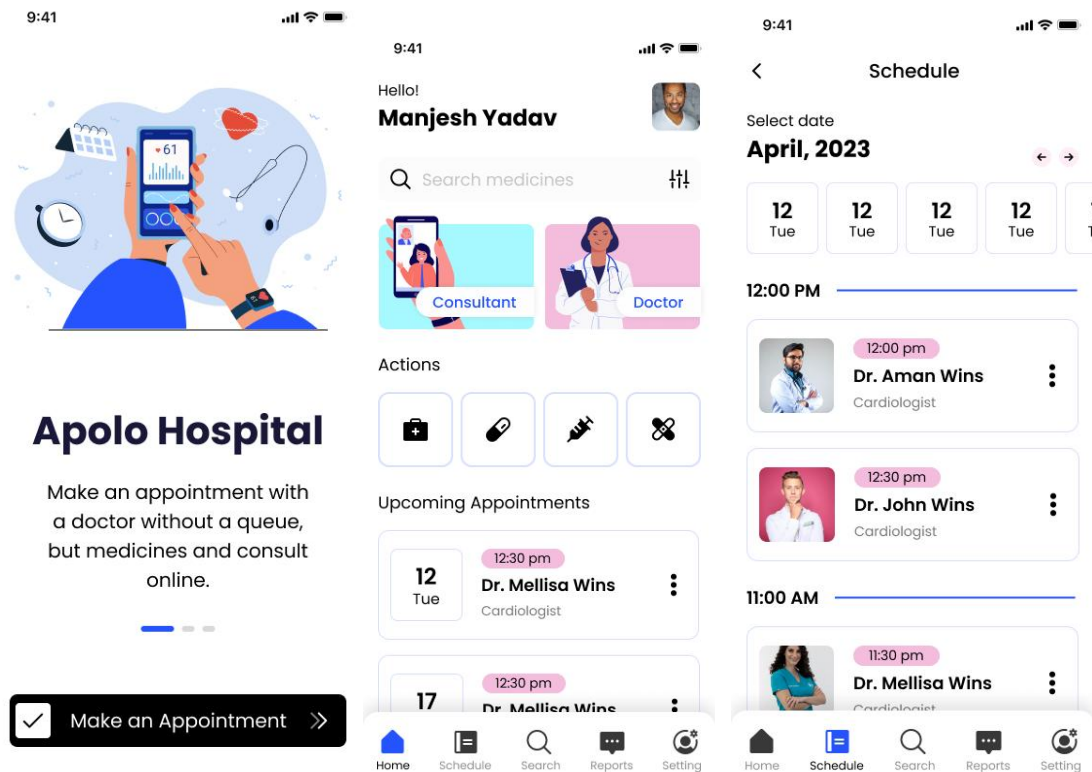
- Mendesain struktur basis data yang optimal untuk mendukung fitur aplikasi (menggunakan ERD).
- Mengelola data pasien, dokter, rekam medis, dan inventaris dengan aman dan efisien.
- Melakukan optimasi query untuk meningkatkan performa database.
- Memastikan keamanan data melalui backup rutin, enkripsi, dan pengaturan hak akses.
- Memantau kinerja basis data dan menangani masalah teknis (troubleshooting)

Untuk Sebuah Database Administrator menggunakan beberapa teknologi seperti phpMyAdmin, MySQL dan PostgreSQL

#### 3.2. UI/UX Designer

**Deskripsi:** UI/UX designer bertugas untuk merancang antarmuka pengguna yang ramah dan efisien, memastikan aplikasi mudah digunakan oleh semua pihak, termasuk pasien, dokter, dan admin rumah sakit.

Untuk Contoh UI/UX nya seperti ini :



Berikut ini merupakan penjelasan sekaligus saran pada Tampilan UI/UX yang ada pada desain ini :

- **Onboarding Screen:**

**Alur Penggunaan :** Saat pengguna pertama kali membuka aplikasi, layar onboarding memperkenalkan fitur utama seperti pembuatan janji temu tanpa antrian dan konsultasi online. Pengguna dapat memahami manfaat utama aplikasi sebelum melanjutkan.

**Kelebihan :** Ilustrasi menarik membantu menyampaikan pesan secara visual. Tombol aksi "Make an Appointment" mengarahkan pengguna secara intuitif ke langkah selanjutnya.

**Saran :** Tambahkan tombol "Skip" agar pengguna berpengalaman dapat langsung menuju layar utama. Tampilkan progres langkah onboarding untuk memberikan konteks kepada pengguna.

- **Home Screen:**

**Alur Penggunaan :** Pengguna disambut dengan tampilan profil, pencarian obat, dan daftar janji temu mendatang. Tindakan cepat seperti menambahkan catatan medis atau membuat janji temu tersedia di bagian tengah layar.

**Kelebihan :** Tata letak yang responsif dan terorganisir mempermudah akses fitur utama. Adanya ikon tindakan cepat memberikan efisiensi dalam navigasi.

**Saran :** Tambahkan pengelompokan janji temu berdasarkan status ("Upcoming", "Completed", "Cancelled") untuk memberikan informasi lebih terstruktur. Tambahkan notifikasi visual untuk fitur penting, seperti janji temu yang segera berlangsung.

### ● **Schedule Screen**

**Alur Penggunaan:** Pengguna memilih tanggal dari kalender di bagian atas layar. Setelah itu, daftar dokter yang tersedia pada waktu tertentu akan ditampilkan dengan spesialisasi dan waktu yang bisa dipilih.

**Kelebihan :** Tampilan kalender di bagian atas memudahkan navigasi. Desain kartu dokter memberikan informasi lengkap, seperti nama, spesialisasi, dan waktu janji temu.

**Saran :** Tambahkan warna atau label status untuk jadwal dokter ("Available", "Full") agar lebih informatif. Berikan fitur pencarian atau filter berdasarkan spesialisasi dokter untuk pengalaman yang lebih personal.

Dengan alur penggunaan yang dirancang untuk kenyamanan pengguna dan tampilan visual yang menarik, aplikasi ini diharapkan mampu memberikan pengalaman yang intuitif dan efisien. Desain responsif serta akses cepat ke fitur utama harus tetap menjadi prioritas dalam pengembangan UI/UX ini.

## **3. Kesimpulan**

Pengembangan aplikasi Rumah Sakit membutuhkan kolaborasi erat antara programmer, UI/UX designer, dan database administrator untuk memastikan solusi yang memenuhi kebutuhan rumah sakit. Fitur-fitur utama seperti manajemen pasien, rekam medis, dan keuangan harus diprioritaskan, dengan tetap memperhatikan keamanan dan kemudahan penggunaan. Langkah berikutnya adalah menyusun timeline pengembangan dan memulai proses desain prototipe.

Aplikasi ini diharapkan menjadi solusi digital yang komprehensif untuk rumah sakit modern, mengintegrasikan layanan kesehatan dengan teknologi untuk memberikan manfaat yang signifikan kepada pasien, tenaga medis, dan manajemen rumah sakit. Dengan implementasi yang tepat, aplikasi ini dapat meningkatkan efisiensi, menekan biaya operasional, dan memberikan pengalaman layanan yang unggul.