

**DOKUMEN PANDUAN PENGGUNAAN APLIKASI E-KLINIK XYZ**  
**CAPSTONE TEAM C22-114**  
**KAMPUS MERDEKA SIB X DICODING CYCLE 3**  
**2022**

Team ID: C22-114

Nama Anggota:

1. A327X0880 - Firman Ferdiansyah
2. A009Y0083 - Reina Lamya
3. A246X0493 - R. Mahendra Agung Wicaksono
4. A246X0499 - Yuda Wahfiudin

Paket Pembelajaran: Pengembang Aplikasi Android

Tema yang dipilih: Solusi Terkait Kesehatan dan Kesejahteraan Lingkungan

Judul Proyek: Rancangan Aplikasi Sistem Informasi Pelayanan Klinik Kesehatan XYZ (E-Klinik) berbasis Android

**BAB I**  
**PENDAHULUAN**

**1.1 Latar Belakang**

Pada zaman sekarang ini hampir semua orang membutuhkan pelayanan publik yang efektif dan efisien. Termasuk didalamnya adalah pelayanan kesehatan pada Klinik Kesehatan. Klinik Kesehatan merupakan instansi pelayanan kesehatan publik yang sangat vital dan memiliki peran sangat penting bagi masyarakat, dimana mereka adalah pengguna layanan kesehatan dan tentunya harus mendapat pelayanan yang prima karena berkaitan erat dengan masalah kesehatan. Pengembangan Teknologi informasi juga mempengaruhi tingkat pelayanan Klinik Kesehatan. Para pasien menginginkan informasi tentang penjadwalan dokter berbasis system informasi mobile agar lebih mudah dalam proses informasi kesediaan dokter dan jadwal dokter di Klinik Kesehatan XYZ (E- Klinik).

Berdasarkan permasalahan di atas kami membuat aplikasi pelayanan kesehatan pada Klinik Kesehatan XYZ (E- Klinik) berbasis Android. Manfaat yang diperoleh pasien ketika menggunakan aplikasi ini adalah dapat mempermudah pasien dalam mendapatkan pelayanan dan informasi

seputar kebersihan, kesehatan lingkungan, dan pencegahan penyakit dengan menggunakan perangkat mobile berbasis Android. Aplikasi ini diperuntukan bagi masyarakat yang membutuhkan informasi tentang pelayanan di Klinik Kesehatan XYZ (E- Klinik). Pasien dapat menggunakan aplikasi ini ketika ada keluhan tentang kesehatannya yang akan dikonsultasikan ke dokter. Pasien dapat menggunakan aplikasi ini kapan saja dan dimana saja. Aplikasi ini dirancang dengan menggunakan bahasa pemrograman kotlin dan menggunakan Android Studio sebagai kode editor.

## **1.2 Identifikasi Masalah**

1. Tidak adanya informasi publik tentang jadwal praktek dokter poliklinik secara up to date, hal ini memicu pasien datang lebih awal tidak sesuai jadwal pelayanan dokter poliklinik.
2. Tidak adanya informasi tentang ruangan yang tersedia untuk rawat inap
3. Kurang Informasi akan ketersediaan stok obat-obatan
4. Pendaftaran pasien hanya bisa dilakukan di rumah sakit melalui antrian loket pendaftaran.

## **1.3 Tujuan**

Tujuan dari tugas ini adalah:

1. Meningkatkan kualitas dan mutu pelayanan Kesehatan Klinik.
2. Mengembangkan sistem yang lebih baik dari sistem informasi yang sudah ada.
3. Memberikan informasi up to date terhadap pasien
4. Menjadi referensi bagi peneliti berikutnya di bidang pengembangan sistem informasi manajemen antrian Klinik.

## **1.4 Manfaat**

Sedangkan manfaat yang dapat diperoleh penelitian ini adalah:

1. Meningkatkan daya minat pasien untuk berkunjung kembali dengan pemanfaatan aplikasi informasi pelayanan kesehatan berbasis Mobile
2. Dapat menurunkan waktu tunggu pasien dalam pendaftaran rawat jalan poliklinik pada Klinik.
3. Memberikan kemudahan bagi pasien untuk melakukan pendaftaran pada poliklinik tanpa

harus datang lebih awal atau dini hari untuk mendapatkan informasi dan pelayanan kesehatan di Klinik.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Application Programming Interface (API)**

API adalah antarmuka yang digunakan untuk mengakses aplikasi atau layanan dari sebuah program. API memungkinkan pengembang untuk memakai fungsi yang sudah ada dari aplikasi lain sehingga tidak perlu membuat ulang dari awal. Pada konteks web, API merupakan pemanggilan fungsi lewat Hyper Text Transfer Protocol (HTTP) dan mendapatkan respon berupa Extensible Markup Language (XML) atau JavaScript Object Notation (JSON). Tujuan penggunaan dari API adalah untuk saling berbagi data antar aplikasi yang berbeda tersebut, Tujuan penggunaan API lainnya yaitu untuk mempercepat proses pengembangan aplikasi dengan cara menyediakan sebuah function yang terpisah sehingga para developer tidak perlu lagi membuat fitur yang serupa. API yang bekerja pada tingkat sistem operasi membantu aplikasi berkomunikasi dengan layer dasar dan satu sama lain mengikuti serangkaian protokol dan spesifikasi. Contoh yang dapat menggambarkan spesifikasi tersebut adalah POSIX (Portable Operating System Interface). Dengan menggunakan standar POSIX, aplikasi yang di-compile untuk bekerja pada sistem operasi tertentu juga dapat bekerja pada sistem lain yang memiliki kriteria yang sama. Software library juga memiliki peran penting dalam menciptakan compatibility antar sistem yang berbeda.

#### **2.2 Representational State Transfer (REST)**

REST merupakan seperangkat prinsip arsitektur yang melakukan transmisi data melalui antarmuka yang terstandarisasi seperti HTTP. REST API bekerja layaknya seperti aplikasi web biasa. Client dapat mengirimkan permintaan kepada server melalui protokol HTTP dan kemudian server memberikan respons balik kepada klien. REST dikembangkan oleh Roy Fielding yang merupakan co-founder dari Apache HTTP Server Project. Pada arsitektur REST, REST server menyediakan resources (sumber daya/data) dan REST client mengakses dan menampilkan resource tersebut untuk penggunaan selanjutnya. Setiap resource diidentifikasi oleh URIs (Universal Resource Identifiers) atau global ID. Resource tersebut direpresentasikan dalam bentuk format teks, JSON atau XML. Pada umumnya formatnya menggunakan JSON dan XML.

RESTFUL API memiliki 4 komponen penting yaitu:

- URL Design
- HTTP Verbs
- HTTP Response Code
- Format Response

4 komponen penting dari REST API tersebut.

### **2.2.1 URL Design**

RESTful API diakses menggunakan protokol HTTP. Penamaan dan struktur URL yang konsisten akan menghasilkan API yang baik dan mudah untuk dimengerti developer. URL API biasa disebut endpoint dalam pemanggilannya.

### **2.2.2 HTTP Verbs**

Setiap request yang dilakukan terdapat metode yang dipakai agar server mengerti apa yang sedang di request client:

a) GET.

GET adalah metode HTTP Request yang paling mudah, metode ini digunakan untuk membaca atau mendapatkan data dari sumber.

b) POST.

POST adalah metode HTTP Request yang digunakan untuk membuat data baru dengan menyisipkan data dalam body saat request dilakukan.

c) PUT.

PUT adalah metode HTTP Request yang biasanya digunakan untuk melakukan update data resource.

d) DELETE.

DELETE adalah metode HTTP Request yang digunakan untuk menghapus suatu data pada resource.

### **2.2.3 HTTP Response Code**

HTTP Response Code adalah kode standarisasi dalam menginformasikan hasil request kepada client. Secara umum terdapat 3 kelompok yang biasa kita jumpai pada RESTful API yaitu :

2XX : adalah response code yang menampilkan bahwa request berhasil.

4XX : adalah response code yang menampilkan bahwa request mengalami kesalahan pada sisi client.

5XX : adalah response code yang menampilkan bahwa request mengalami kesalahan pada sisi server.

### **2.2.4 Format Response**

Setiap request yang dilakukan client akan menerima data response dari server, response tersebut biasanya berupa data XML ataupun JSON. Setelah mendapatkan data response tersebut barulah client bisa menggunakannya dengan cara memarsing data tersebut dan diolah sesuai kebutuhan.

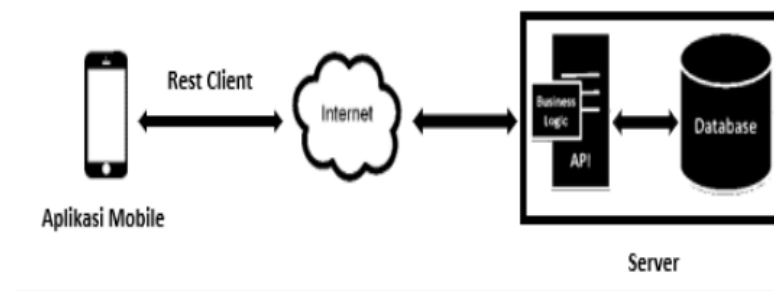
### **2.2.5 Cara REST APIs bekerja**

API secara eksplisit memanfaatkan metodologi HTTP yang ditentukan oleh protokol RFC 2616. Permintaan ke API bisa menggunakan GET untuk mengambil sumber daya, PUT untuk mengubah status atau memperbarui sumber daya, yang dapat berupa objek, file, atau blok, POST untuk membuat sumber daya itu, dan DELETE untuk menghapusnya.

## BAB III HASIL DAN PEMBAHASAN

### 3.1 Arsitektur Sistem

Aplikasi client (android) yang di bangun pada penelitian ini terintegrasi oleh Application Programming Interface yang menggunakan metode REST. Dapat dilihat pada gambar 1 merupakan skema pengambilan data menggunakan REST API. Data jasa yang sudah dipesan oleh pengguna nantinya akan digabungkan dan dikirim dalam bentuk json. Kemudian data tersebut akan dikirim ke server API. Server API akan melakukan proses permintaan ke database.



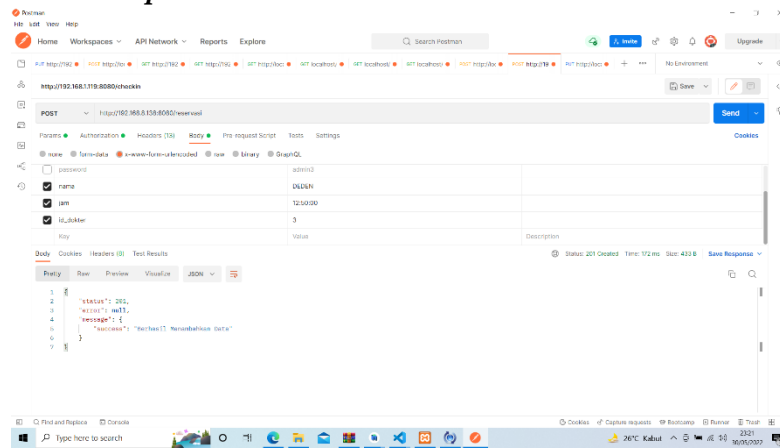
Gambar.1

Yang kemudian hasil dari proses permintaan akan dikembalikan lagi ke pengguna dalam bentuk json dan diuraikan oleh aplikasi sehingga data hasil proses bisa digunakan dan ditampilkan. Perancangan dimulai dari perancangan Application Programming Interface (API) dengan REST sebagai arsitektur yang dipakai untuk menghubungkan antara frontend dalam hal ini adalah aplikasi mobile berbasis android dengan database yang memuat segala data yang digunakan. API yang sudah dibuat akan dilakukan uji coba terlebih dahulu dan dites menggunakan aplikasi Postman yang merupakan sebagai salah satu tools penguji API, lalu dibangun content management system (CMS) sebagai website admin yang bertujuan untuk mengelola content yang akan dipakai untuk aplikasi nantinya, kemudian desain antarmuka untuk aplikasi sesuai dengan konsep API yang telah dibuat sebelumnya.

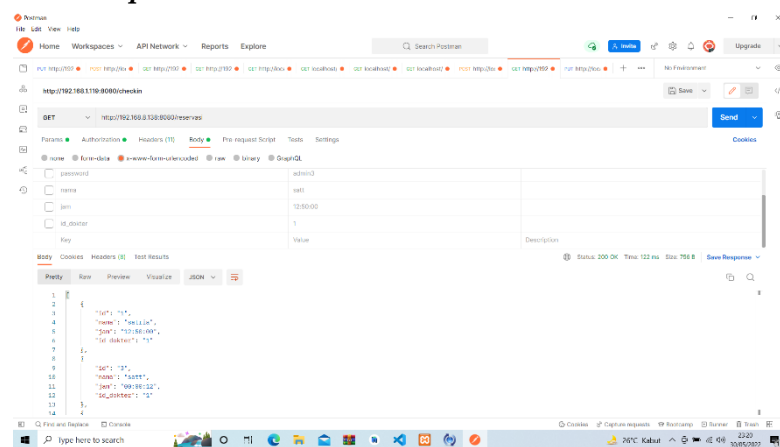
## 3.2 Proses Pembuatan Api

Berikut cara kerja api pada aplikasi pelayanan Kesehatan:

### 3.2.1 Menambahkan data pada reservasi

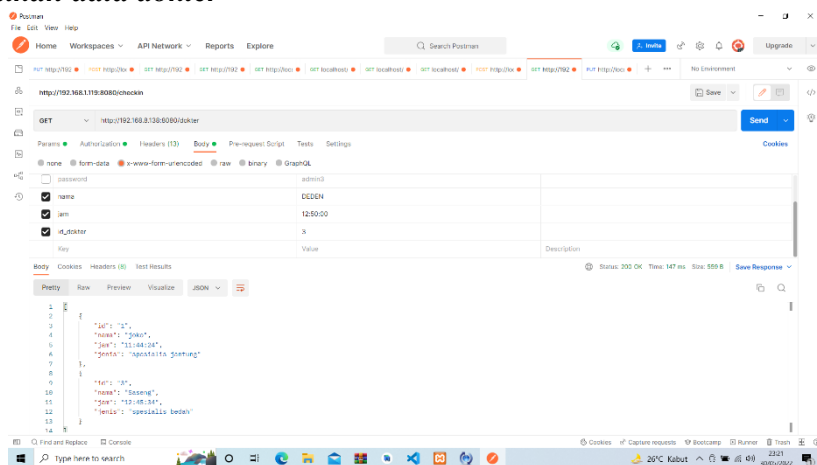


### 3.2.2 Menampilkan data pada reservasi

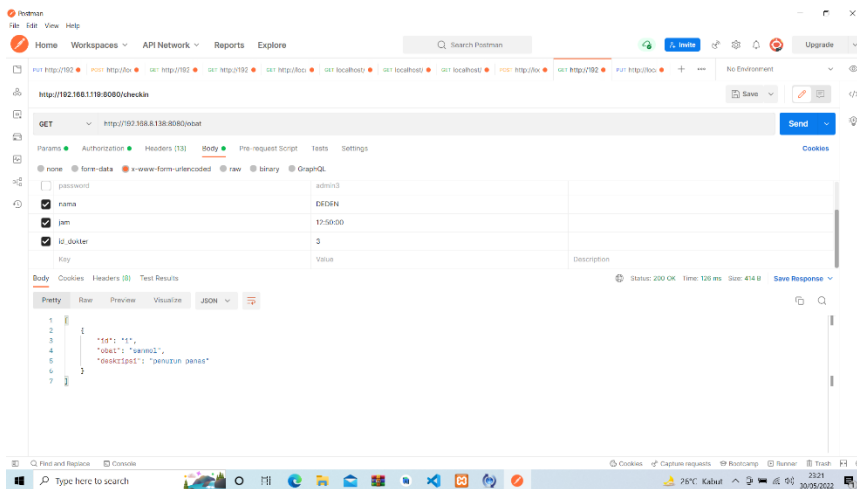




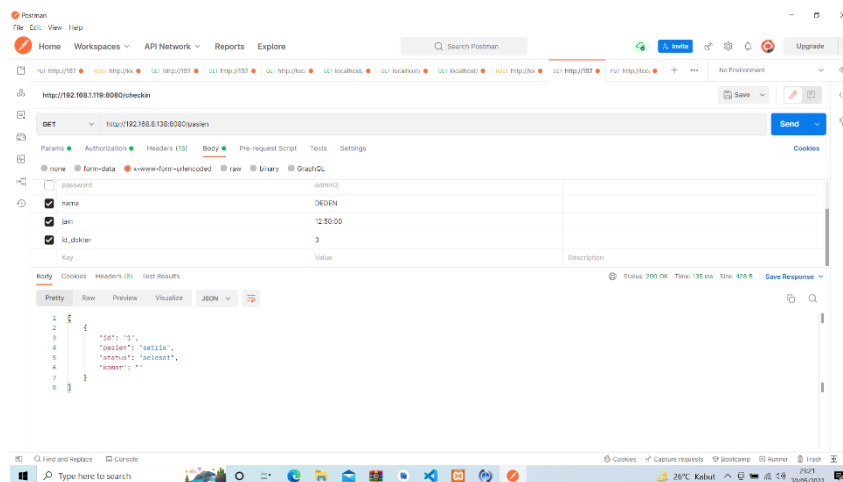
### 3.2.3 Menampilkan data dokter



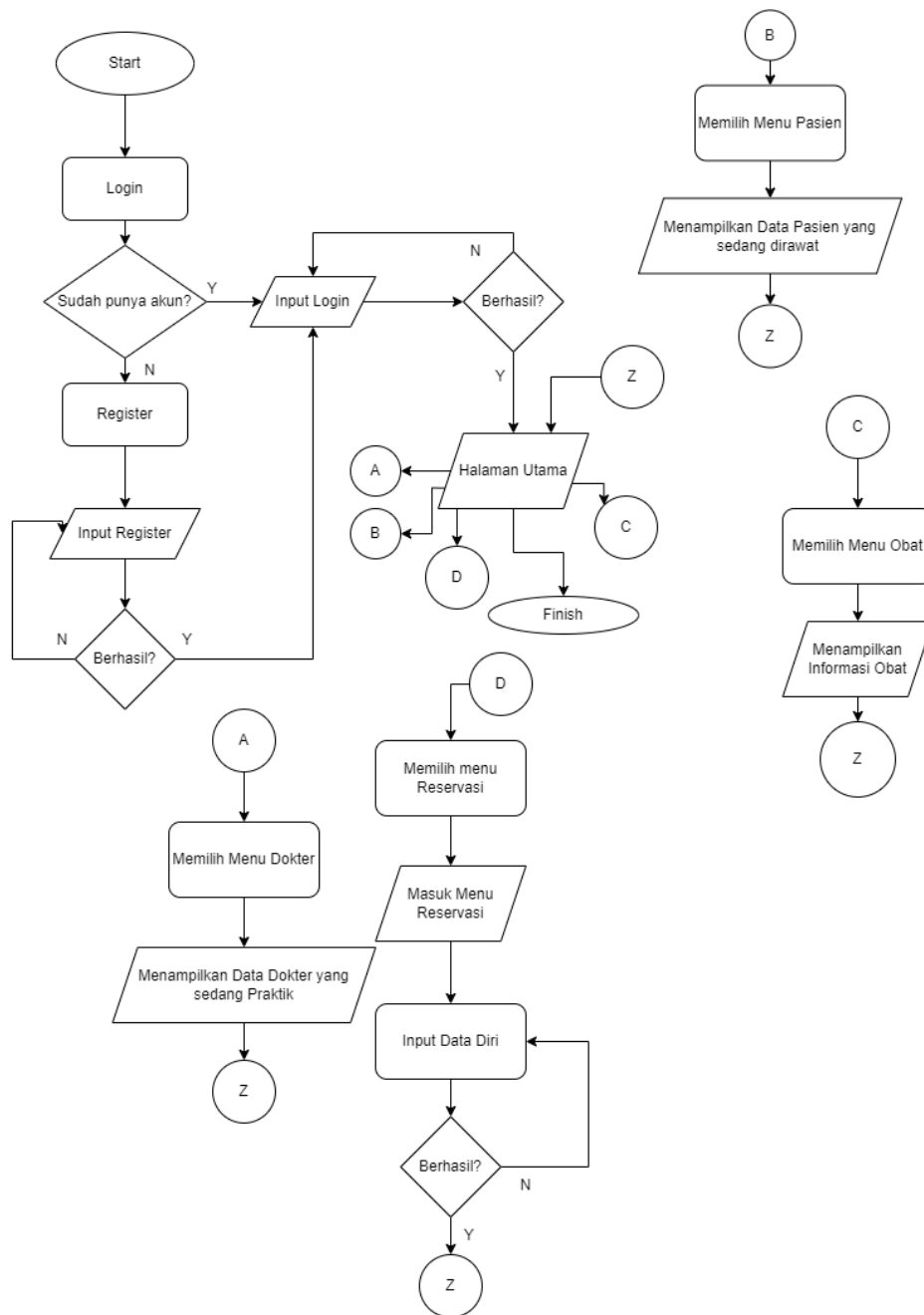
### 3.2.4 Menampilkan data obat



### 3.2.5 Menampilkan data pasien



### 3.3 Perancangan Sistem



Gambar 2

Berikut Merupakan Rancangan Flowchart aplikasi informasi pelayanan Kesehatan RSUD berbasis mobile.

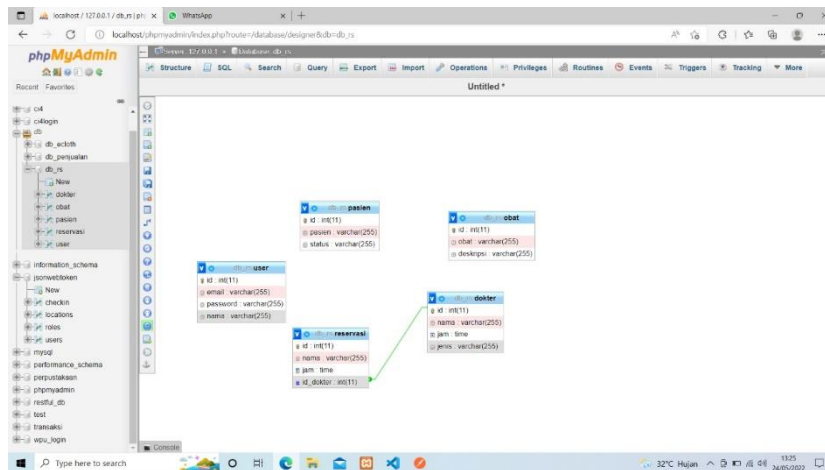
Dalam Use Case Flowchart di bawah ini memiliki scenario:

1. Aktor/Pasien melakukan asosiasi login, Ketika login berhasil pasien dapat langsung ke halaman utama, tetapi jika login gagal pasien kembali ke halaman login untuk melakukan login ulang.
2. Aktor/Pasien terhubung halaman utama setelah proses login terpenuhi
3. Dari halaman utama aktor dapat mengakses jadwal dokter untuk informasi jadwal dokter dan informasi ketersediaan stok obat-obatan
4. Dari halaman utama pasien juga dapat melakukan reservasi untuk input data diri pasien.

### 3.4 Relasi Antar Tabel

Relasi antar tabel menggambarkan keterhubungan antar tabel / entitas dalam sistem.

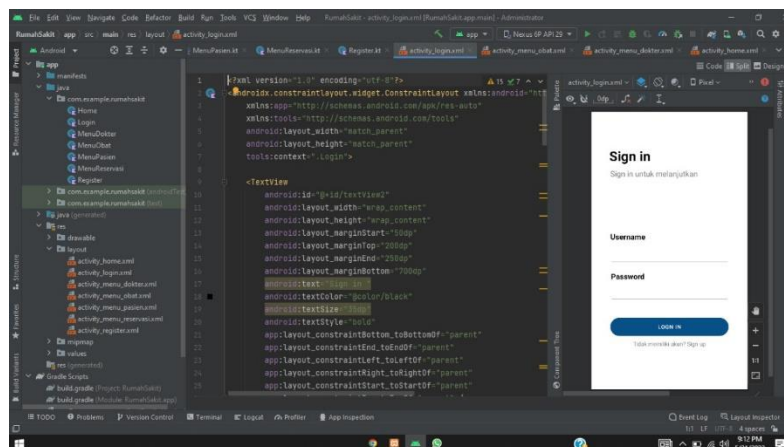
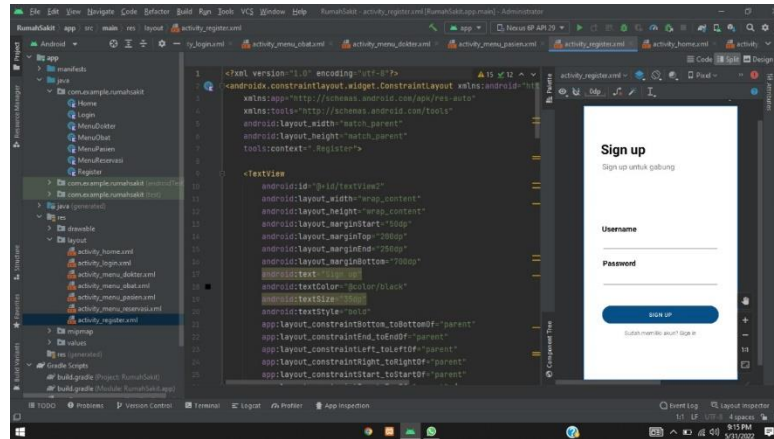
Berikut relasi tabel dari sistem yang diusulkan :



## 3.5 Perancangan Tampilan

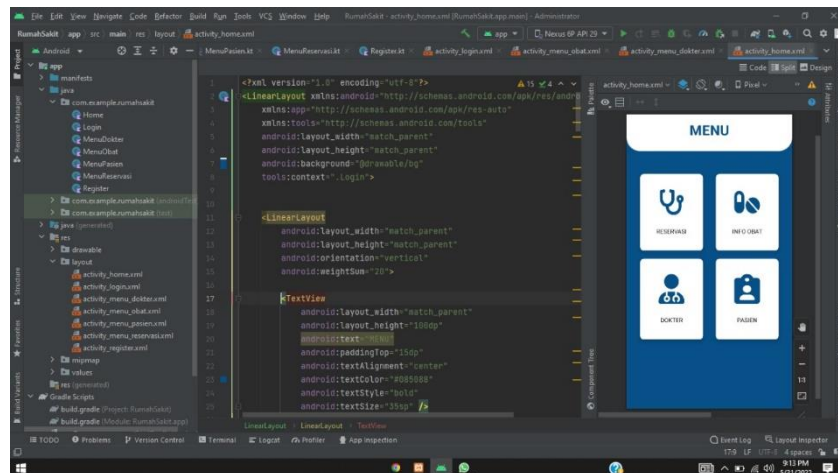
### 3.5.1 Halaman Login

Halaman Login saat aplikasi dibuka, terdapat logo judul dan deskripsi di bagian atas tengah, di bagian bawah terdapat isian username dan password, dan serta terdapat dua tombol sign in untuk proses login dan batal. Berikut gambar rancangan:



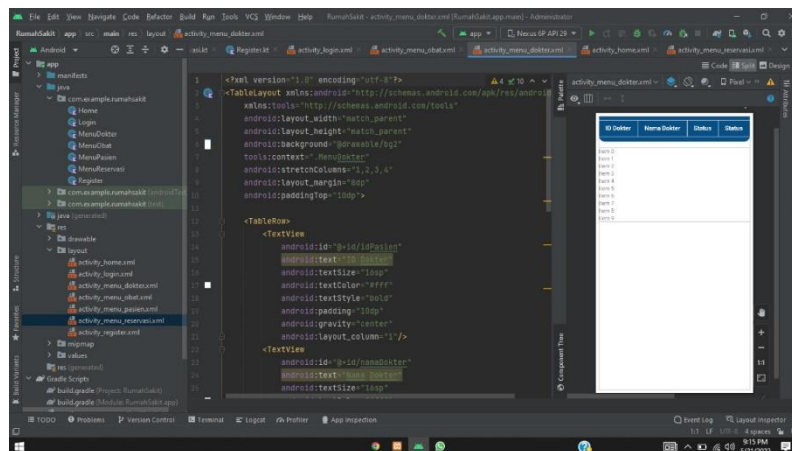
### 3.5.2 Halaman Utama

Halaman Utama dapat diakses setelah melakukan login pada aplikasi, halaman utama terdiri dari menu-menu yang dapat diakses oleh user untuk melakukan pendaftaran secara mandiri, data informasi kamar, jadwal dokter, info ketersediaan obat, dan di antara menu tersebut juga ada menu akun untuk melihat data pribadi dan riwayat user, Berikut gambar rancangan:



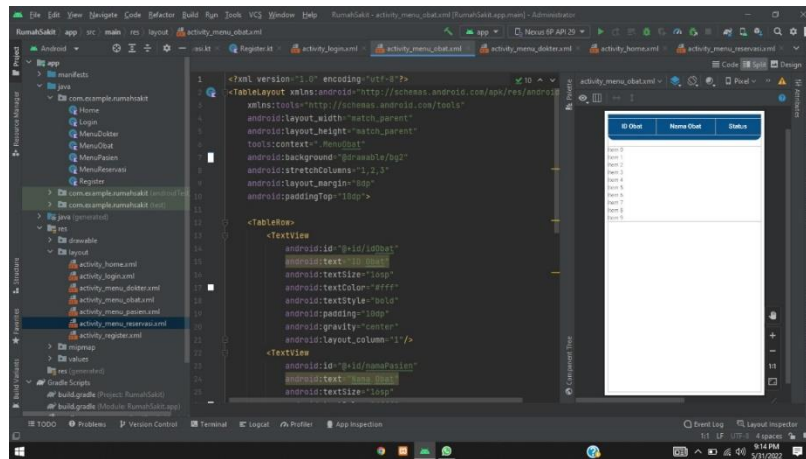
### 3.5.3 Halaman Dokter

Halaman dokter memiliki tombol kalender untuk memilih tanggal yang ingin dilihat, maka akan menampilkan list nama poliklinik, nama dokter dan jam praktek sesuai pilihan tanggal. Berikut gambar rancangan:



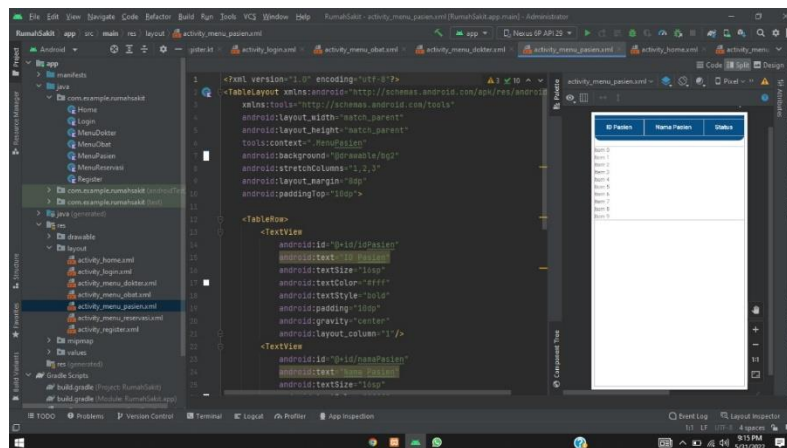
### 3.5.4 Halaman Obat

Halaman obat terdapat id obat, nama obat dan deskripsi. Halaman ini untuk informasi ketersediaan stok obat-obatan dan juga info nama obat serta kegunaannya.



### 3.5.5 Halaman Pasien

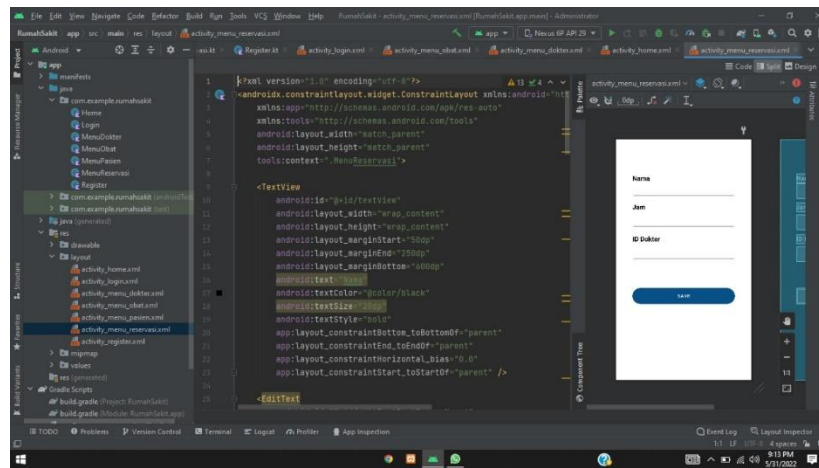
Menu master pasien yaitu tampilan menu yang digunakan untuk menambah data pasien baru, menyimpan data pasien, mengubah data pasien,, menampilkan data dengan listview dan keluar dari menu. Berikut tampilan menu master pasien adalah sebagai berikut :



### 3.5.6 Halaman Reservasi

Halaman form Pendaftaran terdapat nama pasien, jadwal kunjungan, dan id dokter. Halaman reservasi ini memudahkan pasien melakukan pendaftaran dengan tidak mengantri pada loket pendaftaran di RSUD

Sleman



## 3.6 Simulasi Berjalanannya Aplikasi

### 3.6.1 User Melakukan Registrasi / Sign Up menggunakan Username dan Password di halaman Sign in



### 3.6.2 User Melakukan Login menggunakan Username dan Password yang terdaftar di halaman Login

The screenshot displays the login interface of the E-Klinik mobile application. At the top, there is a status bar showing 'vivo 1918' and various system icons. Below this is a blue header bar with the text 'E-Klinik'. The main content area is white and features the title 'Sign in' in bold. Underneath the title is the subtitle 'Sign in untuk melanjutkan'. There are two input fields: 'Username' with the text 'admin2' and 'Password' with masked characters '\*\*\*\*\*'. A blue button labeled 'LOGN IN' is positioned below the password field. At the bottom of the login section, there is a link that reads 'Tidak memiliki akun? Sign Up'. The entire screen is framed by a blue border at the bottom, which contains standard Android navigation icons (back, home, and app drawer).



### 3.6.3 User memilih pilihan Reservsi pada menu

The screenshot displays the E-Klinik mobile application interface. At the top, there is a status bar for a Vivo 1918 phone showing the time 23:23 and various icons. Below the status bar is a dark blue header with the text "E-Klinik" and a circular icon. The main content area is white and contains a form with three labeled input fields: "Nama" with the text "Wisnu", "Jam" with the time "16:30", and "ID Dokter" with the number "3". Each field has a light blue underline. Below these fields is a dark blue button with the text "SAVE" in white. At the bottom of the screen is a dark blue navigation bar with three white icons: a back arrow, a circle, and a hamburger menu.

vivo 1918 23:23

E-Klinik

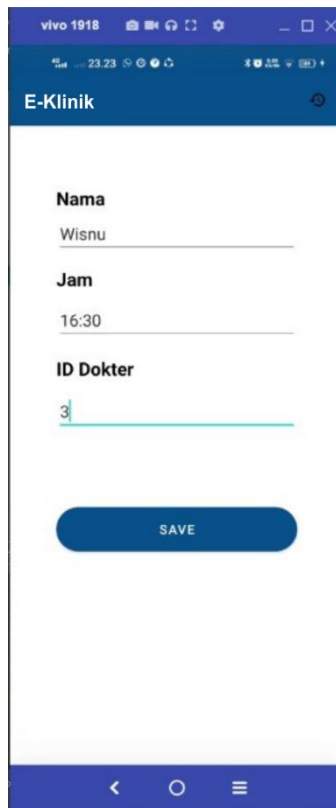
**Nama**  
Wisnu

**Jam**  
16:30

**ID Dokter**  
3

SAVE

### 3.6.4 User menginput Nama Pasien, Jam, dan ID Dokter pada Reservasi



The screenshot displays the 'E-Klinik' mobile application interface. At the top, there is a status bar for a 'vivo 1918' device showing the time as 23:23 and various system icons. Below the status bar is a dark blue header with the text 'E-Klinik' and a small circular icon on the right. The main content area is white and contains three input fields, each with a label above it: 'Nama' (Name) with the value 'Wisnu', 'Jam' (Time) with the value '16:30', and 'ID Dokter' (Doctor ID) with the value '3'. Each input field has a light blue underline. Below these fields is a dark blue rounded rectangular button with the text 'SAVE' in white. At the bottom of the screen is a dark blue navigation bar with three white icons: a back arrow, a circle, and a hamburger menu.

vivo 1918 23:23

E-Klinik

**Nama**  
Wisnu

**Jam**  
16:30

**ID Dokter**  
3

SAVE

### 3.6.5 *User memilih pilihan Pasien pada menu*



### 3.6.3 User memilih pilihan Dokter pada menu



### 3.6.3 *User memilih pilihan Info Obat pada menu*



### 3.7 Source Code

Link github : <https://github.com/YudaWahfiudin/E-Klinik-C22-114.git>

## **BAB IV**

### **PENUTUP**

#### **4.1 Kesimpulan**

Berdasarkan hasil tugas akhir dan pembahasan, maka didapatkan kesimpulan sebagai berikut:

1. Pendaftaran rawat jalan dapat dilakukan oleh pasien secara mandiri, dapat dilakukan dimanapun dan kapanpun, tanpa harus datang ke pendaftaran rawat jalan di Klinik.
2. Tersedianya informasi jadwal dokter, dan ketersediaan stok obat-obatan yang dapat diakses oleh pasien dengan menggunakan aplikasi pendaftaran rawat jalan pasien mandiri berbasis mobile.
3. Kepastian waktu kedatangan pasien dapat diperkirakan maksimal tiga puluh menit sebelum jam pelayanan dilakukan, tanpa harus datang pagi-pagi ke pendaftaran rawat jalan yang ada di Klinik. Hal ini dapat mengurai pasien datang secara bersamaan dalam satu waktu, melainkan sesuai jadwal pelayanan dokter spesialis.