

# **BAB I**

## **PENDAHULUAN**

### **A. Latar Belakang**

Dalam kehidupan manusia saat ini perkembangan dunia teknologi berkembang sangat cepat khususnya dalam bidang teknologi informasi. Perkembangan teknologi informasi semakin memudahkan manusia dalam melakukan kegiatan pada setiap harinya serta memiliki fungsi untuk menciptakan sistem efektif dan efisien terlebih dalam proses pengolahan data, pencarian data dan penyajian informasinya secara *real time*. Salah satu contoh berkembangnya teknologi informasi adalah dalam dunia pelayanan kesehatan yaitu aplikasi rekam medis. Rekam Medis digunakan untuk menunjang tercapainya tertib administrasi pada suatu organisasi pelayanan kesehatan.

Pemerintah di bidang teknologi informasi seperti internet saat ini sudah dikembangkan dan tersebar di seluruh pulau Indonesia. Keberadaan KTP Elektronik sebagai implementasi *big data* sudah mulai merambah ke banyak aspek pelayanan publik. Tentunya hal tersebut sangat memungkinkan e-KTP dengan nomor NIK menjadi kunci bagi seseorang untuk mengakses dan diakses datanya oleh pihak siapa pun yang berkepentingan. Termasuk sebagai kunci identitas rekam medis.

Rekam medis merupakan berkas yang berisikan catatan dan dokumen tentang identitas pasien, pemeriksaan, pengobatan serta tindakan dan pelayanan lain kepada pasien selama mendapatkan perawatan di penyedia layanan kesehatan.

Rekam medis adalah data yang bersifat pribadi dan menjadi salah satu informasi yang penting bagi seorang pasien pada saat menjalani pelayanan kesehatan. Kepemilikan informasi tersebut merupakan kepentingan dasar seorang pasien dan data tersebut bersifat rahasia bagi orang lain yang tidak berhak. (Iriani, 2014)

Puskesmas Barukku yang menjadi salah satu pusat pelayanan kesehatan di Kabupaten Sidrap merupakan salah satu instansi pelayanan kesehatan tingkat I yang saat ini berjalan. Dengan jumlah pasien yang cukup banyak menyebabkan masalah yang cukup berat dalam mendapatkan informasi tentang pasiennya, kunjungan berobat pasien, rekam medis pasien dan juga data obat yang sudah digunakan oleh puskesmas tersebut. Selain masalah pendataan pasien dan pengarsipan catatan medis merupakan suatu hal penting yang perlu diperhatikan. Apalagi di Puskesmas tersebut pendataan pasien, dari mulai pendaftaran pasien dan pengarsipan catatan medis pasien masih dilakukan secara konvensional, artinya masih menuliskannya pada tumpukan-tumpukan kertas dan di simpan pada rak-rak penyimpanan, sehingga ketika pihak puskesmas membutuhkan data-data pasien, laporan kunjungan dan juga laporan data obat-obatan yang sudah digunakan, sehingga petugas perlu waktu yang cukup lama untuk mencari dimana data tersebut disimpan. Rekam medis sendiri berisi tentang identitas pasien, hasil pemeriksaan, pengobatan dan tindakan yang dilakukan kepada pasien tersebut.

Selain itu, proses bisnis untuk melakukan kegiatan rekam medis pasien pada Puskesmas Barukku dimulai dari datangnya pasien ke bagian administrasi, kemudian bagian administrasi menanyakan pada pasien tersebut, apakah pasien merupakan pasien baru atau pasien lama. Jika pasien tersebut merupakan pasien

baru, maka bagian administrasi akan memberikan formulir pendaftaran kepada pasien untuk diisi oleh pasien sesuai dengan identitas diri. Kemudian setelah pasien mengisi formulir pendaftaran, bagian administrasi melakukan pencatatan identitas pasien pada buku rekam medis dan kartu *member*. Selanjutnya kartu *member* diberikan kepada pasien yang telah melakukan pendaftaran, Jika pasien tersebut merupakan pasien lama maka bagian administrasi akan meminta kartu *member* pasien, kemudian bagian administrasi mencari kartu rekam medis pasien yang sesuai dengan nomer dan identitas pasien. Selanjutnya kartu tersebut diberikan pada dokter untuk melakukan konsultasi, setelah pasien masuk ke ruangan dokter, maka dokter akan melakukan anamnesis dan menanyakan keluhan serta riwayat penyakit yang diderita saat ini. Dokter akan mencatat keluhan dan diagnosis ke dalam kartu rekam medis pasien. Selanjutnya kartu rekam medis pasien diberikan kepada bagian petugas obat untuk pengambilan obat. Setelah petugas obat memberikan obat kepada pasien, maka pasien diperbolehkan pulang. Sesudah bagian obat akan memberikan kartu pasien ke bagian administrasi, bagian administrasi akan melakukan proses rekap data keseluruhan pasien yang berkunjung pada hari tersebut dari kartu pasien dan ditulis kembali di buku rekam medis. Pasien yang tidak memiliki atau yang kehilangan kartu berobat akan dibuatkan buku rekam medis baru. Sehingga pasien yang kehilangan kartu berobat akan memiliki dua buku rekam medis, hal ini dapat mengakibatkan penggandaan data rekam medis pasien pada lemari buku.

Salah satu teknologi informasi yang mendukung guna peningkatan dan pencapaian mutu pelayanan kesehatan adalah aplikasi rekam medis. Rekam Medis

digunakan untuk menunjang tercapainya tertib administrasi pada suatu organisasi pelayanan kesehatan, hal ini sejalan dengan UU RI No. 8 Th 1997 Tentang Dokumen Perusahaan. Yang berisi (1) ketentuan wajib simpan untuk catatan mengenai keadaan kekayaan dan yang mencerminkan hak dan kewajiban perusahaan selama 10 tahun; dan (2) keberadaan dokumen perusahaan yang dibuat dalam bentuk bukan berupa kertas, seperti dalam bentuk microfilm, elektronik dan atau hasil cetaknya, dapat dijadikan sebagai alat bukti yang sah.

Pelayanan kesehatan yang baik merupakan kewajiban yang harus dipenuhi agar kebutuhan pasien terpenuhi. Semakin tinggi pelayanan sebuah puskesmas maka proses pengelolaan data rekam medis pun semakin kompleks, dimana setiap data rekam medis yang ada harus mampu dikelola secara efektif dan efisien. Maka dari itu, untuk memenuhi kebutuhan tersebut sudah seharusnya pengelolaan rekam medis menggunakan teknologi informasi terbaru guna meningkatkan mutu pelayanan kesehatan,

Berdasarkan uraian latar belakang masalah yang telah dijelaskan, permasalahan diatas maka penelitian ini berfokus pada pemecahan masalah untuk merancang suatu sistem rekam medis pada pelayanan kesehatan, maka penulis mengusulkan judul **“Aplikasi *Mobile* Rekam Medis Pasien Berdasarkan NIK Pada Pelayanan Kesehatan Menggunakan *QR Code*”**. Dengan sistem yang baru ini dapat mengintegrasikan NIK sebagai implementasi *big data* sehingga penelusuran rekam medis pasien jadi lebih mudah diagnosis penyakit dan juga obatnya dengan menggunakan teknologi *QR Code Scanner*.

## B. Rumusan Masalah

Berdasarkan uraian latar belakang yang telah diuraikan, maka akan dibuat suatu penelitian pada masalah yang sedang dihadapi. Adapun latar belakang masalah yang dapat diidentifikasi sebagai berikut :

1. Bagaimana menganalisis, merancang, dan membangun suatu aplikasi rekam medis pasien berbasis *mobile*?
2. Bagaimana mengimplementasi *big data* berdasarkan NIK pada pelayanan kesehatan menggunakan *QR Code*?

## C. Batasan Masalah

Adapun agar pembahasan menjadi lebih terarah dan tidak menyimpang dan juga sesuai dengan latar belakang yang sudah diuraikan sebelumnya, maka penulis membatasi masalah sebagai berikut :

1. Ruang lingkup penelitian ini difokuskan atau terpusat pada pelayanan kesehatan tingkat I yaitu Puskesmas Barukku di Kabupaten Sidrap.
2. Aplikasi ini mengelola data-data pasien seperti mengubah data pasien, melakukan diagnosa, menyimpan data rekam medis pasien dan membuat laporan data rekam medis.
3. Aplikasi ini terdapat trafik pasien dan pemeriksaan dalam setiap bulannya yang ditampilkan dalam bentuk grafik kurva.
4. Pembuatan kartu pasien yang berkenaan mengenai data pasien.
5. Aplikasi dibangun dengan menggunakan bahasa pemrograman *HTML*, *Javascript*, *PHP*, dan *Android Studio* serta database *MySQL*.

#### **D. Tujuan Penelitian**

Berdasarkan rumusan masalah yang telah didefinisikan sebelumnya tujuan dari penelitian dan perancangan sistem dalam penelitian ini adalah :

1. Untuk menganalisis, merancang, dan membangun suatu aplikasi rekam medis pasien berbasis *mobile*.
2. Mengimplementasi *big data* berdasarkan NIK pada pelayanan kesehatan menggunakan *QR Code* yaitu petugas dan pasien dapat menggunakan dan mengakses aplikasi ini dengan mudah dan aman.

#### **E. Manfaat Penelitian**

Salah satu fungsi dari penelitian adalah bisa memberikan manfaat dan kegunaan yang secara nyata akan diterima oleh berbagai pihak yang terkait dengan obyek penelitian seperti aspek akademis dan praktis. Dalam penelitian kali ini peneliti berharap bisa memberikan kegunaan, antara lain :

1. Aspek Akademis
  - a) Untuk mengaplikasikan teori yang di dapat di bangku perkuliahan dengan kondisi real di lapangan, khususnya terkait dengan permasalahan yang dibahas terkait penelitian dan *tools* atau metode penyelesaian masalah.
  - b) Meningkatkan kemampuan berfikir mengenai langkah-langkah dan tata cara membuat sistem yang berguna dan dapat diukur melalui keefektifan dalam penggunaan dan keefisienan dalam pemanfaatan aplikasi.

- c) Dapat membuat perguruan tinggi dikenal lebih luas oleh Institusi swasta maupun negeri yang dijadikan objek penelitian.
- d) Diharapkan Sebagai bahan masukan bagi para pengkaji ilmu komunikasi di bidang *Public Relations* yang berminat untuk meneliti masalah yang sama.

## 2. Aspek Praktis

- a. Memiliki sistem yang baru berbasis *mobile* pada pengelolaan rekam medis pasien, sehingga mampu meningkatkan kualitas pelayanan terhadap pasien dan kemudahan serta keakuratan akses data bagi petugas rekam medis.
- b. Bagi peneliti diharapkan dapat memberikan pengalaman dan menerapkan ilmu pengetahuan yang diterima selama mengikuti perkuliahan.
- c. Sebagai tugas akhir untuk mendapatkan gelar sarjana strata 1 (S.1) pada Fakultas Teknik, Program Studi Informatika, Universitas Muhammadiyah Parepare.

## F. Sistematika Penulisan

Sistematika penulisan skripsi ini dibagi ke dalam enam bab yang masing-masing menjelaskan bagian-bagian yang berbeda namun tetap memiliki korelasi satu dengan yang lain. Setiap bab-nya terdiri dari beberapa sub tersendiri. Diawali dengan bab pendahuluan dan di akhiri dengan bab penutup yang berisi kesimpulan dan saran-saran. Secara singkat akan diuraikan sebagai berikut :

### **BAB I : PENDAHULUAN**

Pada bab ini akan dijelaskan tentang latar belakang penulisan, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

## **BAB II : TINJAUAN PUSTAKA**

Bab ini berisi uraian tentang landasan teori-teori dari perancangan “*Aplikasi Mobile Rekam Medis Pasien Berdasarkan NIK Pada Pelayanan Kesehatan Menggunakan QR Code*”. Bab ini juga membahas mengenai berbagai teori yang mendasari dari analisis permasalahan yang berhubungan dengan topik yang dibahas.

## **BAB III : METODOLOGI PENELITIAN**

Dalam bab ini membahas mengenai metode penelitian yang digunakan dalam merancang dan membangun sistem. Pada bab ini juga membahas tentang identifikasi masalah, analisis kebutuhan aplikasi yang meliputi kebutuhan masukan, proses dan keluaran aplikasi yang akan dibangun.

## **BAB IV : HASIL DAN PEMBAHASAN**

Pada bab ini membahas mengenai hasil dari tahap analisis dan perancangan sesuai dengan metode yang digunakan pada alat dan sistem yang dibuat serta hasil dari pengujian.

## **BAB V : PENUTUP**

Pada bab akhir ini berisi tentang kesimpulan dari penelitian yang telah dilakukan dan saran yang diberikan untuk pengembangan lebih lanjut agar tercapai hasil yang lebih maksimal.

## BAB II

### TINJAUAN PUSTAKA

#### A. Penelitian Terdahulu

Berbagai penelitian sebelumnya merupakan hal yang sangat perlu dan dapat dijadikan sebagai data pendukung. Salah satu data pendukung yang menurut peneliti perlu dijadikan bagian tersendiri adalah penelitian terdahulu yang relevan dengan permasalahan yang sedang dibahas dalam penelitian ini. Oleh karena itu, peneliti melakukan kajian terhadap beberapa hasil penelitian berupa tugas akhir/skripsi atau jurnal melalui *internet*.

1. Hussain Muhammad Irsyad (2020) dengan skripsi yang berjudul **“Rancang Bangun Sistem Pelayanan Pasien Rawat Jalan di Puskesmas Pallangga Kabupaten Gowa Berbasis Web”**. Berdasarkan hasil penelitian yang dilakukan oleh peneliti tersebut, tujuan dari penelitian ini adalah merancang sistem pelayanan pasien rawat jalan berbasis *web* agar memudahkan pihak puskesmas dalam melayani pasien di Puskesmas Pallangga. Jenis Penelitian yang digunakan penulis adalah deskriptif dan kualitatif dengan menggunakan metode pengumpulan data yaitu observasi, wawancara dan studi pustaka. Adapun metode yang digunakan untuk merancang sistem pelayanan rawat jalan ini menggunakan metode waterfall. Aplikasi ini dibuat dengan *Framework Codeigniter (CI)* dengan perangkat lunak database *MySQL* serta pengujian sistem aplikasi ini menggunakan metode *Black-box*.

2. Fauzi Megantara, (2019) dalam sebuah jurnal “**Implementasi *Big Data* Untuk Pencarian *Pattern Data* Gudang Pada PT. Bank Mandiri (Persero) TBK**”.

Peneliti berfokus akan mengembangkan aplikasi yang dapat mengolah dan melakukan pencarian pola data sebagai bahan pelaporan, implementasi teknologi *big data* akan sangat membantu proses pengelolaan data pada aplikasi tersebut, dikarenakan data yang di kelola dalam kurun waktu yang cepat akan terus bertambah, sehingga pengelolaan data menggunakan teknologi *big data* menjadi solusi untuk dapat mengolah data dalam melakukan pencarian *pattern* pada data gudang Bank Mandiri. Aplikasi yang akan di kembangkan tersebut akan menyajikan informasi-informasi yang di butuhkan seperti *pattern* barang yang paling banyak di pesan dan *pattern user* yang paling banyak melakukan pemesanan, sehingga *pattern* pada aplikasi tersebut akan membantu staff dalam melakukan pelaporan dan *Manegement* tingkat atas dalam melakukan analisa dalam pengambilan keputusan.

3. Arin Sri Alfiani (2017) dengan skripsi yang berjudul “**Aplikasi *Presensi* Pegawai Menggunakan *QR Code* Berbasis *Android* Di Puskesmas Cisaga**”.

Dengan aplikasi ini setiap pegawai akan mendapatkan *id card* untuk presensi yang didalamnya terdapat *QR Code*. Perancangan aplikasi yang dibangun meliputi perancangan aplikasi dengan menggunakan diagram *UML* dan perancangan arsitektur database menggunakan *Entity Reationship Diagram* (ERD) dan *Logical Record Structure* (LRS). Aplikasi yang dibuat diharapkan dapat membantu mempermudah pegawai puskesmas cisaga dalam melakukan *presensi*, dan dapat mengurangi kecurangan apabila pegawai tidak bisa hadir..

## B. Tinjauan Teori

### 1. Aplikasi (*Application*)

Aplikasi berasal dari kata *application* yaitu bentuk benda dari kata kerja to *apply* yang dalam Bahasa Indonesia berarti pengolah. Secara istilah, aplikasi komputer adalah suatu *sub* kelas perangkat lunak komputer yang menggunakan komputer langsung untuk melakukan suatu tugas yang diinginkan pemakai. Menurut Kamus Besar Bahasa Indonesia (KBBI, 2015) Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna. Sementara menurut Hartono (1999) aplikasi adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output*.

### 2. Rekam Medis

#### a. Definisi Rekam Medis

Rekam medis diartikan sebagai keterangan baik yang tertulis maupun yang terekam tentang identitas, anamnesis, pemeriksaan, laboratorium, diagnosa, segala pelayanan dan tindakan medis yang diberikan kepada pasien, dan pengobatan baik yang di rawat inap, rawat jalan, maupun yang mendapatkan pelayanan gawat darurat (Depkes RI, 1997). Rekam medis mempunyai pengertian yang sangat luas tidak hanya sekedar kegiatan pencatatan akan tetapi mempunyai pengertian sebagai satu sistem

penyelenggaraan rekam medis. Sedangkan kegiatan pencatatannya sendiri hanya merupakan kegiatan dari salah satu penyelenggara rekam medis.

Menurut Permenkes 269 Tahun 2008 tentang rekam medis, rekam medis adalah berkas yang berisikan catatan dan dokumen tentang identitas pasien, pemeriksaan, pengobatan, tindakan dan pelayanan lain yang telah diberikan kepada pasien. Rekam medis harus di buat secara tertulis, lengkap dan jelas atau secara elektronik. Rekam medis harus di buat segera setelah pasien menerima pelayanan.

#### b. Tujuan Rekam Medis

Menurut Depkes RI (1997), tujuan rekam medis adalah menunjang tercapainya tertib administrasi dalam rangka upaya peningkatan pelayanan kesehatan di rumah sakit. Tanpa di dukung suatu sistem pengolahan rekam medis yang baik dan benar, mustahil tertib administrasi rumah sakit akan berhasil sebagaimana yang diharapkan. Sedangkan tertib administrasi merupakan salah satu faktor yang menentukan dalam upaya pelayanan kesehatan di rumah sakit.

### 3. Manfaat Rekam Medis

Kegunaan rekam medis dapat dilihat dari beberapa aspek (Depkes RI, 1997), antara lain :

#### 1) Aspek Administrasi

Suatu berkas rekam medis mempunyai nilai-nilai administrasi karena isinya menyangkut tindakan berdasarkan wewenang dan tanggung jawab sebagai tenaga medis dan paramedik dalam mencapai tujuan pelayanan kesehatan.

## 2) Aspek Medis

Suatu berkas rekam medis mempunyai nilai medis karena catatan tersebut dipergunakan sebagai dasar untuk merencanakan pengobatan/perawatan yang harus diberikan kepada seorang pasien.

## 3) Aspek Hukum

Suatu berkas rekam medis mempunyai nilai hukum karena isinya menyangkut masalah adanya jaminan kepastian hukum atas dasar keadilan dalam rangka usaha menegakkan hukum serta penyediaan bahan tanda bukti untuk menegakkan keadilan.

## 4) Aspek Keuangan

Suatu berkas rekam medis mempunyai nilai uang dikarenakan isinya mengandung data/ informasi yang dapat dipergunakan sebagai aspek-aspek keuangan.

## 5) Aspek Penelitian

Suatu berkas rekam medis mempunyai nilai penelitian karena isinya menyangkut data/ informasi yang dapat dipergunakan sebagai aspek penelitian dan pengembangan ilmu pengetahuan di bidang kesehatan.

## 6) Aspek Pendidikan

Suatu berkas rekam medis mempunyai nilai pendidikan karena isinya menyangkut data/informasi tentang perkembangan kronologis dan kegiatan pelayanan medik yang diberikan kepada pasien. Informasi tersebut dapat dipergunakan sebagai bahan/ referensi pengajaran di bidang profesi pemakai.

#### 7) Aspek Dokumentasi

Suatu rekam medis mempunyai nilai dokumentasi karena isinya menyangkut sumber ingatan yang harus didokumentasikan dan dipakai sebagai bahan pertanggung jawaban dan laporan rumah sakit. Dari beberapa aspek di atas, rekam medis mempunyai kegunaan yang sangat luas, karena tidak hanya menyangkut antara pasien dengan pemberi pelayanan saja.

#### 4. *QR Code (Quick Response Code)*

*Quick Response Code* sering disebut *QR Code* atau Kode QR adalah semacam simbol dua dimensi yang dikembangkan oleh Denso Wave yang merupakan anak perusahaan dari Toyota sebuah perusahaan Jepang pada tahun 1994. Tujuan dari *QR Code* ini adalah untuk menyampaikan informasi secara cepat dan juga mendapat tanggapan secara cepat. Pada awalnya *QR Code* digunakan untuk pelacakan bagian kendaraan untuk *manufacturing*. Namun sekarang, telah digunakan untuk komersil yang ditujukan pada pengguna telepon seluler.

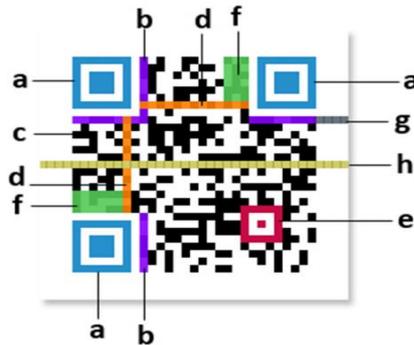


**Gambar 2.1** *QR Code*  
(Sumber : <https://id.qr-code-generator.com>)

*QR Code* biasanya berbentuk persegi putih kecil dengan bentuk geometris hitam (dapat dilihat di gambar 2.1), meskipun sekarang banyak yang telah berwarna dan digunakan sebagai brand produk. Informasi yang dikodekan dalam *QR Code* dapat berupa *URL*, nomor telepon, pesan *SMS*, *V-Card*, atau teks apapun. *QR Code*

telah mendapatkan standarisasi internasional ISO/IEC18004 dan Jepang JISX-0510.

a. Anatomi *QR Code*



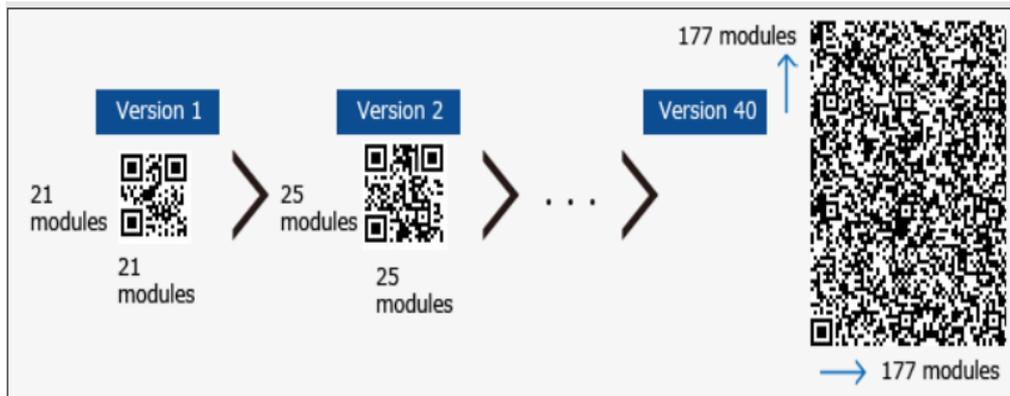
**Gambar 2.2** Anatomi *QR Code*  
(Sumber : <https://id.qr-code-generator.com>)

Adapun beberapa penjelasan dari anatomi *QR Code* antara lain, sebagai berikut:

- 1) *Finder Pattern* berfungsi untuk identifikasi letak *QR Code*.
- 2) *Format Information* berfungsi untuk informasi tentang *error correction level* dan *mask pattern*.
- 3) Data berfungsi untuk menyimpan data yang dikodekan.
- 4) *Timing Pattern* merupakan pola yang berfungsi untuk identifikasi koordinat pusat *QR Code*, berbentuk modul hitam putih.
- 5) *Alignment Pattern* merupakan pola yang berfungsi memperbaiki penyimpangan *QR Code* terutama distorsi *non linier*.
- 6) *Version Information* adalah versi dari sebuah *QR Code*.

- 7) *Quiet Zone* merupakan daerah kosong di bagian terluar *QR Code* yang mempermudah mengenali pengenalan *QR* oleh sensor CCD.
- 8) *QR Code version* adalah versi dari *QR Code* yang digunakan.

b. Versi *QR Code*



**Gambar 2.3** Versi *QR Code*  
(Sumber : <https://id.qr-code-generator.com>)

*QR Code* dapat menghasilkan 40 versi yang berbeda dari versi 1 (21 x 21 modul) sampai versi 40 (177 x 177 modul). Tingkatan Versi *QR Code* 1 dan 2 berbeda 4 modul berlaku sampai dengan versi 40. Setiap versi memiliki konfigurasi atau modul yang berbeda. Modul ini mengacu pada titik hitam dan putih yang membentuk suatu *QR Code*. Setiap versi *QR Code* memiliki kapasitas maksimum data, jenis karakter dan tingkat koreksi kesalahan.

### C. Tinjauan Perangkat Lunak

#### 1. *Android*

*Android* merupakan perangkat bergerak pada sistem operasi untuk telepon seluler yang Menggunakan *Linux*. Menurut (Hermawan, 2011), *Android* merupakan OS (*Operating System*) *Mobile* yang tumbuh ditengah OS lainnya yang

berkembang dewasa ini. Berdasarkan pendapat diatas, maka dapat ditarik kesimpulan bahwa *Android* adalah sistem operasi Menggunakan *Linux* yang sedang berkembang ditengah OS lainnya. (Arifianto, 2011)

Ada beberapa macam *android* dari yang tingkatan pertama hingga terbaru dengan beberapa fitur *software* yang melengkapinya sebagai berikut.

a) *Android Software Development Kit* (SDK)

*Android* SDK adalah *tools API* (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada *platform Android* menggunakan bahasa pemrograman *Java*. *Android* merupakan *subset* perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang release oleh *Google*. Saat ini di sediakan *Android* SDK (*Software Development Kit*) sebagai alat bantu dan *API* untuk mulai mengembangkan aplikasi pada *platform Android* menggunakan bahasa pemograman java (Nazruddin, 2011).

b) *Java SE Development Kit* (JDK)

JDK adalah perangkat lunak yang digunakan untuk melakukan proses kompilasi dari kode *java* ke *bytecode* yang dapat dimengerti dan dapat dijalankan oleh JRE (*Java Runtime Envirotment*). JDK wajib terinstall pada komputer yang akan melakukan proses pembuatan aplikasi Menggunakan *java*, namun tidak wajib terinstall di komputer yang akan menjalankan aplikasi yang dibangun dengan *java*. (Nazruddin, 2011).

c) *Android Development Tools* (ADT)

*Android Development Tools* (ADT) yang lebih dikenal *plugin Eclipse*. *Plugin* ini yang membuat *Eclipse* dapat membuat *project* yang menggunakan *Android*. ADT adalah *plugins* di *Eclipse* yang harus diinstall sehingga *Android SDK* dapat dihubungkan dengan IDE *Eclipse* yang digunakan sebagai tempat coding aplikasi *Android* nantinya. ADT adalah kepanjangan dari *Android Development Tolls* yang menghubungkan antara IDE *Eclipse* dengan *Android SDK* (Nazruddin, 2011). *Android Virtual Device* (AVD) yang merupakan emulator untuk menjalankan program aplikasi *Android* yang dibuat. AVD ini nantinya yang kita jadikan sebagai tempat tes dan menjalankan aplikasi *Android* yang dibuat. AVD berjalan di *virtual Machine* (Nazruddin, 2011).

## 2. PHP (*Hypertext Preprocessor*)

PHP (*Hypertext Prosesor*) adalah akronim dari *Hypertext Preprocessor*, yaitu suatu bahasa pemograman berbasis kode-kode (*script*) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke *browser* menjadi kode HTML. (Diar Puji Octavian 2010) Kode PHP mempunyai ciri-ciri khusus, yaitu :

- a) Hanya dapat dijalankan menggunakan *web server* misalnya : *Apache*.
- b) Kode PHP dapat diletakan dan dijalankan di *web server*.
- c) Kode PHP dapat digunakan untuk mengakses database, seperti: *MY SQL*, *PostgreSQL*, *Oracle*, dan lain-lain.
- d) Merupakan *software* yang bersifat *open source*.
- e) Gratis untuk di *download* dan digunakan.
- f) Memiliki sistem *multiplatform*, artinya dapat dijalankan menggunakan sistem operasi apapun, seperti *Linux*, *Unix*, *Windows*, dan lain-lain.

PHP merupakan *software Open-Source* yang disebar dan dilisensikan secara gratis serta dapat di-*download* secara bebas dari situs resminya <http://www.php.net>. PHP ditulis menggunakan bahasa C. PHP memiliki banyak kelebihan yang tidak dimiliki oleh bahasa *script* sejenis. PHP difokuskan pada pembuatan *script server-side*, yang bisa melakukan apa saja yang dapat dilakukan oleh CGI, seperti mengumpulkan data *form*, menghasilkan isi halaman dinamis, dan kemampuan mengirim serta menerima *cookies*, bahkan lebih dari kemampuan CGI.

### 3. JavaScript

*JavaScript* adalah bahasa pemrograman tingkat tinggi dan dinamis. *JavaScript* populer di internet dan dapat bekerja di sebagian besar penjelajah *web* populer seperti *Google Chrome*, *Internet Explorer (IE)*, *Mozilla Firefox*, *Netscape* dan *Opera*. Kode *JavaScript* dapat disisipkan dalam halaman *web* menggunakan tag *SCRIPT*. *JavaScript* merupakan salah satu teknologi inti *World Wide Web* selain *HTML* dan *CSS*. *JavaScript* membantu membuat halaman *web* interaktif dan merupakan bagian aplikasi *web* yang esensial.

Awalnya hanya diimplementasi sebagai *client-side* dalam penjelajah *web*, kini *engine JavaScript* disisipkan ke dalam perangkat lunak lain seperti dalam *server-side* dalam server *web* dan basis data, dalam program *non web* seperti perangkat lunak pengolah kata dan pembaca *PDF*, dan sebagai *runtime environment* yang memungkinkan penggunaan *JavaScript* untuk membuat aplikasi *desktop* maupun *mobile*. (Madcoms Madiun, 2012)

#### 4. HTTP (*Hypertext Transfer Protocol*)

HTTP (*Hypertext Transfer Protocol*) adalah *protocol* yang digunakan untuk mentransfer dokumen. *Protocol* ini adalah *protocol* ringan, tidak berstatus yang dapat dipergunakan berbagai macam *tipe* dokumen. Pengembangan HTTP dikoordinasikan oleh *Konsorsium World Wide Web* (W3C) dan grup bekerja *Internet Engineering Task Force* (IETF), bekerja dalam publikasi satu seri RFC, yang paling terkenal RFC yang menjelaskan HTTP/1,1, versi HTTP yang digunakan umum sekarang ini. HTTP adalah sebuah *protocol* meminta / menjawab antara *client* dan *server*.

Sebuah *client* HTTP seperti web *browser*, biasanya melalui permintaan dengan membuat hubungan *TCP/IP* ke port tertentu. Sebuah server HTTP yang mendengarkan di *port* tersebut menunggu *client* mengirim kode permintaan di ikuti dengan pesan *MIME* yang memiliki data tertentu. (Deka M Wildan, 2014).

#### 5. *PhpMyAdmin*

*PhpMyAdmin* adalah suatu perangkat lunak yang ditulis dalam bahasa pemrograman PHP digunakan untuk menangani administrasi *MySQL* dengan menggunakan protokol WWW secara langsung dari *browser*. *PhpMyAdmin* mendukung berbagai operasi *MySQL*, diantaranya mengelola basis data, tabel-tabel, bidang *fields*, relasi *relations*, indeks, pengguna *users*, perijinan *permissions*, dan lain-lain. Dengan menggunakan *PhpMyAdmin* diharapkan akan memudahkan dalam manajemen database *MySQL*. (Alan Nur Aditya, 2011)

## 6. XAMPP

*XAMPP* ialah perangkat lunak gratis yang mendukung banyak sistem operasi berbagai program (*XAMPP* 2019). Fungsi *XAMPP* adalah server yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP* server, *MySQL* database, dan penterjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP* Server, *MySQL* database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. Nama *XAMPP* merupakan singkatan dari X (tempat sistem operasi apapun), *Apache*, *MySQL*, *PHP* dan *Perl*.

*XAMPP* adalah perangkat lunak besar, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Program ini digunakan untuk melayani tampilan halaman sistem yang dinamis sehingga mampu menyajikan tampilan dengan baik dan dinamis. (Alan Nur Aditya, 2011).

### **D. Perancangan Basis Data**

Menurut (Jogiyanto Hartono, 2005:700). Basis data (*Database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di simpan luar komputer dan digunakan perangkat lunak tertentu untuk memanipulasi. Sedangkan menurut penadapat saya Basis Data adalah data data yang tersimpan dalam komputer dan masih harus di proses agar saling berhubungan satu sama lain. Database merupakan salah satu komponen yang penting di sistem informasi, karena berfungsi sebagai basis penyedia informasi bagi para pemakai.

Sistem basis data ini adalah suatu sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan lainnya dan membuatnya tersedia untuk beberapa aplikasi yang bermacam-macam di dalam suatu organisasi.

### 1. *Entity Relationship Diagram ( ERD )*

Menurut Andi, ERD berguna untuk mengorganisasi data-data (*hanya data*) dalam hubungannya dengan data-data yang lain. Sedangkan menurut saya ERD berfungsi untuk mengelompokkan data-data agar terjadi keselarasan hubungan antar data lainnya. ERD mempunyai koleksi objek-objek dasar yang dinamakan entitas serta hubungan (*relationship*) antara entitas – entitas itu.

Entitas adalah sesuatu atau objek pada dunia nyata yang dapat dibedakan satu terhadap yang lainnya, yang bermanfaat bagi aplikasi yang sedang kita kembangkan. Sedangkan *Relationship* adalah hubungan antara beberapa entitas.

### 2. Normalisasi.

Menurut Andi, Normalisasi adalah proses pengolahan data yang masih salah dan memiliki data ganda untuk kemudian di perbaiki sehingga menghasilkan relasi-relasi yang lebih baik dari sebelumnya. Normalisasi dapat dipahami sebagai tahapan yang masing-masing berhubungan dengan bentuk normal.

Bentuk normal adalah keadaan relasi yang dihasilkan dengan menerapkan aturan sederhana yang berkaitan dengan konsep kebergantungan fungsional pada relasi yang bersangkutan sebagai berikut.

a. Bentuk normal pertama.

Bentuk normal pertama adalah suatu bentuk relasi dimana atribut bernilai banyak (*Multivalued attribute*) telah dihilangkan sehingga kita akan menjumpai nilai tunggal (mungkin saja nilai *null*).

b. Bentuk normal kedua.

Suatu relasi adalah dalam bentuk normal kedua jika dia berada dalam bentuk normal pertama dan setiap atribut bukan kunci bergantung penuh pada kunci *primer*.

c. Bentuk normal ketiga.

Relasi adalah dalam bentuk normal ke tiga adalah jika berada dalam bentuk normal kedua dan tidak dijumpai kebergantungan *transitif*.

3. Tabel Relasi.

Tabel relasi adalah tabel hubungan antara suatu himpunan entitas dengan entitas yang lainnya.

4. Kunci.

Kunci (*key*) merupakan suatu atribut yang unik yang dapat digunakan untuk membedakan suatu entitas dengan entitas yang lainnya dalam suatu himpunan entitas.

## E. Tahap-tahap Perancangan Sistem

### 1. Use Case

*Use case diagram* adalah abstraksi dari interaksi antara sistem dan *actor*. *Use case* bekerja mendeskripsikan tipe interaksi antara *user* sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai.

### 2. Kamus Data

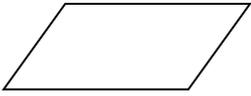
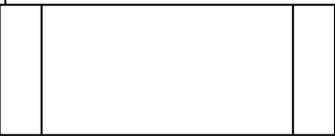
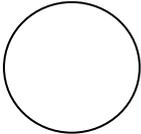
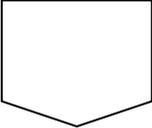
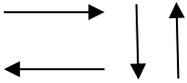
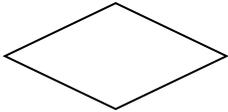
Kamus data (*data dictionary*) adalah katalog fakta tentang data dan kebutuhan informasi yang mengalir pada sistem dengan lengkap dari suatu sistem informasi. Kamus data di buat berdasarkan arus data yang mengalir pada *Data Flow Diagram* (DFD).

Kamus data merupakan tempat penyimpanan dari aliran-aliran data, *file-file* dan proses-proses dalam sebuah sistem dibangun. Bagian ini menjelaskan secara detail proses-proses yang terjadi disetiap proses, *file* dan struktur data mengenai model sistem yang digambarkan. Kamus data digunakan untuk menjelaskan semua data yang mengalir atau digunakan dalam sistem, yaitu mengenai arus data yang masuk ke dalam sistem dan tentang informasi yang dibutuhkan oleh pemakai sistem.

### 3. Bagan Alir Program

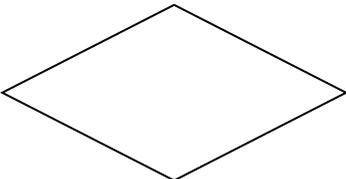
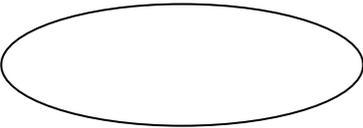
Bagan alir program (*program flowchart*) merupakan bagan yang menjelaskan rinci langkah-langkah dari proses program. Bagan alir program dibuat dari derifikasi bagan alir system. Bagan alir program dibuat dengan menggunakan simbol-simbol sebagai berikut :

**Tabel 2.1** Simbol *Flowchart*

Simbol	Keterangan
	Simbol <i>input/output</i> digunakan untuk mewakili data <i>input/output</i> .
	Simbol proses, digunakan untuk mewakili suatu proses.
	Simbol proses terdefinisi, digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan ditempat lain.
	Simbol titik terminal, digunakan untuk awal dan akhir suatu proses.
	Simbol penghubung, digunakan untuk menunjukkan sambungan dari alir yang terputus di halaman yang masih sama.
	Simbol penghubung, digunakan untuk menunjukkan sambungan dari alir yang terputus di halaman yang berbeda.
	Simbol garis alir, menunjukkan arus dari proses.
	Simbol keputusan, digunakan unuk suatu penyelesaian kondisi di dalam program .

#### 4. *Entity Relationship Diagram*

**Tabel 2.2** Simbol *Entity Relationship Diagram*

NOTASI	KETERANGAN
	Entitas, yaitu kumpulan dari objek yang diidentifikasi secara unik.
	Relasi, yaitu hubungan yang terjadi antara satu atau lebih entitas. Jenis hubungan antara lain: satu ke satu, satu ke banyak dan banyak ke banyak.
	Atribut, yaitu karakteristik dari <i>entity</i> atau relasi yang merupakan penjelasan detail tentang entitas.
	Garis, hubungan antara <i>entity</i> dengan atributnya dan himpunan entitas dengan himpunan relasi.
	<i>Input/Output</i> data, yaitu proses <i>input/output</i> data, parameter, informasi.

*Entity Relationship Diagram (ERD)* adalah adalah suatu pemodelan konseptual yang didesain secara khusus untuk mengidentifikasi entitas yang menjelaskan data dan hubungan antar data. *ERD* merupakan suatu model untuk menjelaskan hubungan antardata dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. *ERD* untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol. Pada dasarnya ada tiga komponen yang digunakan, yaitu :

- a. *Entity* merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. Simbol-simbol dari *entity* ini biasanya digambarkan dengan persegi panjang.
- b. Atribut Setiap entitas pasti mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasi isi elemen satu dengan yang lain. Gambar atribut diwakili oleh simbol *elips*.
- c. Hubungan / Relasi ialah hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda.

Relasi dapat digambarkan sebagai berikut : Relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dalam satu basis data yaitu:

1) Satu ke satu (*One to one*)

Hubungan relasi satu ke satu yaitu setiap entitas pada himpunan entitas A berhubungan paling banyak dengan satu entitas pada himpunan entitas B.

2) Satu ke banyak (*One to many*)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, tetapi setiap entitas pada entitas B dapat berhubungan dengan satu entitas pada himpunan entitas A.

3) Banyak ke banyak (*Many to many*)

Setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B. Mempunyai pengertian "Satu baris atau lebih data pada tabel pertama bisa dihubungkan ke satu atau lebih baris data pada tabel ke dua.

## 5. UML (*Unified Modelling Language*)

UML (*Unified Modelling Language*) menurut Munawar dalam buku “Pemodelan Visual dengan UML” (2005 : 17) menyatakan UML adalah salah satu alat bantu yang sangat handal didunia pengembangan sistem yang berorientasi obyek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

Sedangkan menurut menurut Adi nugroho (2010:6), “*Unified Modelling Language* adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berpradigma berorientasi objek”. Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami. Jurnal CCIT (2008:70) *Unified Modelling Language* (UML) didefinisikan oleh *Good Corporate* sebagai berikut :

*Unified Modelling Language* (UML) adalah “suatu alat bantu yang sangat handal dalam bidang pengembangan yang berorientasi objek. Hal ini disebabkan karena uml menyediakan bahasa pemodelan visual yang memungkinkan pengembang sistem untuk membuat cetak cetak biru atas visi mereka dalam bentuk baku”. *Metode Unified Modelling Language* (UML) menggunakan tiga bangunan dasar untuk mendeskripsikan sistem atau perangkat lunak yang akan dikembangkan, yaitu :

a. Sesuatu (*things*)

Ada empat *things* dalam *Unified Modelling Language* (UML) :

1. *Structural things*, bagian yang relative statis dapat berupa elemen-elemen yang bersifat fisik maupun konseptual.
2. *Behaviorial things*, bagian dinamis biasanya merupakan kata kerja dari model UML yang mencerminkan perilaku sepanjang ruang dan waktu.
3. *Grouping things*, bagian pengorganisasian dalam UML. Dalam penggambaran model UML yang rumit diperlukan penggambaran paket yang menyederhanakan model. Paket-paket ini kemudian dapat didekomposisi lebih lanjut. Paket berguna bagi pengelompokan sesuatu, misalnya model-model serta *subsistem-subsistem*.
4. *An notational things*, merupakan bagian yang meperjelas model UML. Dapat berisi komentar yang menjelaskan fungsi serta ciri-ciri tiap element dalam model UML.

b. Relasi (*relationship*)

Untuk upaya tersebut UML menyediakan 9 jenis diagram yang dapat dikelompokkan berdasarkan sifatnya *statis* atau *dinamis*. Ke 9 jenis diagram dalam UML itu adalah :

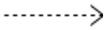
1. *Diagram kelas* Bersifat statis Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula *diagram kelas* memuat kelas-kelas aktif.

2. *Diagram* Objek Bersifat statis. *Diagram* ini memperlihatkan objek-objek serta relasi-relasi antar objek. Diagram objek memperlihatkan instalasi statis dari segala sesuatu yang dijumpai pada diagram kelas.
3. *Use-Case Diagram* Bersifat statis. *Diagram* ini memperlihatkan himpunan *use case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna.
4. *Sequence Diagram* Bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.
5. *Collaboration Diagram* Bersifat dinamis. Diagram kolaborasi adalah diagram interaksi yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan.
6. *Statechart Diagram* Bersifat dinamis. Diagram state ini memperlihatkan *state-state* pada sistem memuat *state*, *transisi*, *event*, serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka, kelas, kolaborasi dan terutama penting pada pemodelan sistem-sistem yang reaktif.
7. *Activity Diagram* Bersifat dinamis. *Diagram* aktivitas ini adalah tipe khusus dari *diagram state* yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama

penting dalam pemodelan fungsi-fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antar objek.

8. *Component Diagram* Bersifat statis. Dengan komponen ini memperlihatkan organisasi serta kebergantungan sistem atau perangkat lunak pada komponen-komponen yang telah ada sebelumnya. *Diagram* ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan kedalam satu atau lebih kelas-kelas, antarmuka-antarmuka serta kolaborasi-kolaborasi.
9. *Deployment Diagram* Bersifat statis. *Diagram* ini memperlihatkan konfigurasi saat aplikasi dijalankan. *Diagram* ini membuat simpul-simpul beserta komponen-komponen yang ada di dalamnya. *Diagram* ini sangat berguna pada banyak mesin. (Sholiq, 2006)

**Tabel 2.3** *Use Case Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung yang tidak mandiri ( <i>independent</i> ).
3		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek.
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .

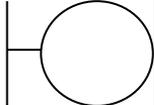
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan suatu paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan dalam sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerjasama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumberdaya komputasi.

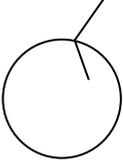
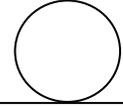
**Tabel 2.4** Simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan cara bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2		<i>Action</i>	State adalah suatu sistem yang mencerminkan eksekusi dari suatu aksi.
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.

4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan.
5		<i>Fork Node</i>	Satu aliran yang pada tahap-tahap tertentu berubah menjadi beberapa aliran.

**Tabel 2.5** Simbol *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Object</i> (Participant)	<i>Object</i> atau biasa juga disebut partisipan merupakan <i>instance</i> dari sebuah <i>class</i> dan dituliskan tersusun dari secara <i>horizontal</i> . Digambarkan ini sebagai sebuah <i>class</i> (kotak) dengan nama objek di dalamnya yang diawali dengan sebuah titik koma.
2		<i>Actor</i>	<i>Actor</i> juga dapat berkomunikasi dengan <i>object</i> , maka <i>actor</i> juga dapat diurutkan sebagai kolom.
3		<i>Lifeline</i>	<i>Lifeline</i> ini mengidentifikasi keberadaan sebuah <i>object</i> dalam basis waktu. Notasi untuk <i>lifeline</i> adalah garis putus-putus <i>vertical</i> yang di tarik oleh sebuah <i>object</i> .
4		<i>Collaborations</i>	<i>Activation</i> dinotasikan sebagai kotak segi empat yang digambar pada sebuah <i>lifeline action</i> mengidentifikasi dari sebuah <i>object</i> yang akan melakukan sebuah aksi.
5		<i>Boundary</i>	<i>Boundary</i> terletak dari diantara <i>system</i> dengan tema dunia di sekelilingnya. Semua dari <i>form</i> , laporan-laporan, antarmuka ke perangkat keras seperti <i>printer</i> atau <i>scanner</i> dan antara muka

			ke <i>system</i> lainnya adalah termasuk dalam kategori.
6		<i>Control</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
7		<i>Entity</i>	<i>Entity</i> ini digunakan menengani informasi yang mungkin akan disimpan secara suatu permanen. <i>Entity</i> ini biasa juga merupakan sebuah <i>table</i> .

## F. Metode Pengujian Perangkat Lunak

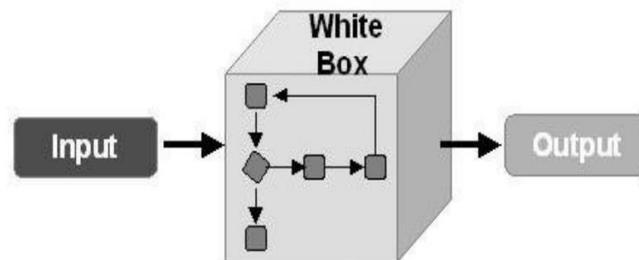
Beberapa *test-case* harus dilaksanakan dengan beberapa perbedaan strategi, *query*, atau jalur navigasi yang mewakili penggunaan sistem yang *typical*, kritis atau abnormal. Isu kunci pada pengembangan sistem adalah pemilihan sekelompok *test-case* yang cocok, sekecil dan secepat mungkin, untuk meyakinkan perilaku sistem secara detail.

Jika struktur kendali antar modul sudah terbukti bagus, maka pengujian yang tak kalah pentingnya adalah pengujian unit. Pengujian unit digunakan untuk menguji setiap modul untuk menjamin setiap modul menjalankan fungsinya dengan baik. Ada 2 metode untuk melakukan unit testing, yaitu :

### 1. *White Box Testing*

Uji coba *white box testing* merupakan metode perancangan *testcase* yang menggunakan struktural untuk mendapatkan *test case*, test ini digunakan untuk

meramal cara kerja perangkat lunak secara rinci kepada *logic path* (jalur logika), perangkat lunak di *test* dengan kondisi dan perulangan secara fisik.



**Gambar 2.4** Pengujian *White Box*

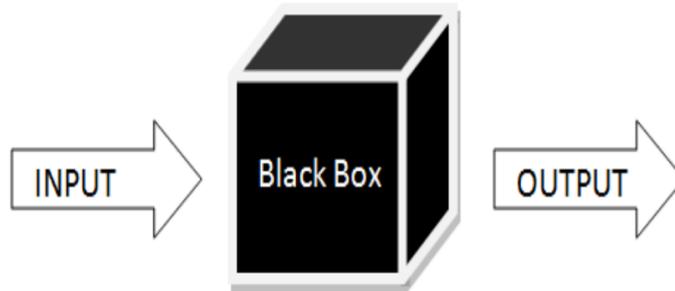
Contoh pengujian *white box* testing ini merupakan peringatan ketika user menginputkan *password user* yang salah, untuk kesalahan semacam ini akan memberikan suatu informasi kepada *user* mengenai kesalahan yang di lakukan.

## 2. *Black Box Testing*

*Black box testing* adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Jadi dianalogikan seperti kita melihat suatu kotak hitam, kita hanya bisa melihat penampilan luarnya saja, tanpa tau ada apa dibalik bungkus hitam nya. Sama seperti pengujian *black box*, mengevaluasi hanya dari tampilan luarnya (interfacenya).

Berdasarkan hasil uji coba yang dilakukan, seluruh navigasi dan tombol fasilitas program lainnya serta proses yang di jalankan tidak terjadi kesalahan, tetapi aplikasi mempunyai aturan-aturan yang sudah di tetapkan dan harus di ikuti karena apabila di hiraukan maka sistem akan menolak perintah yang tidak sesuai seperti kesalahan ketika *user* belum menginput data yang harusnya diinput sesuai ketentuan sistem yang di jalankan.

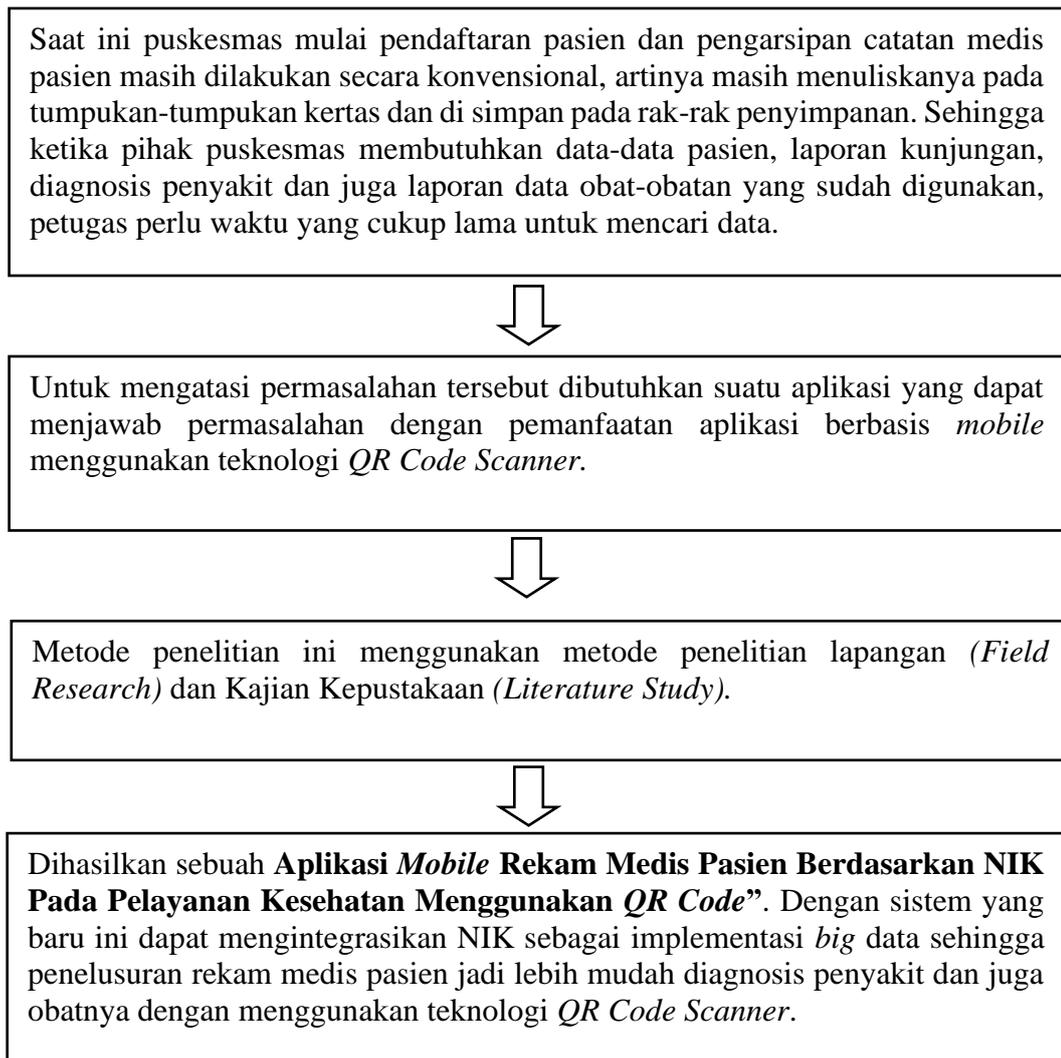
*Black box testing* terfokus pada apakah unit program memenuhi kebutuhan (*requirement*) yang disebutkan dalam spesifikasi. Pada *black box testing*, cara pengujian hanya dilakukan dengan menjalankan atau mengeksekusi unit atau modul, kemudian diamati apakah hasil dari unit itu sesuai dengan proses yang diinginkan.



**Gambar 2.5** Pengujian *Black Box*

## G. Kerangka Pikir

Untuk lebih memperjelas kerangka berpikir yang akan ditulis dalam penelitian akhir ini. Maka berikut ini digambarkan kerangka pikir tersebut pada kerangka berfikir dibawah:



Gambar 2.6 Kerangka Pikir

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **A. Tempat dan Waktu Penelitian**

1. Penelitian yang dilakukan oleh penulis bertempat di pelayanan kesehatan fasilitas tingkat I yaitu Puskesmas Barukku Jl. Poros Compong, Batu Kec. Pitu Riase, Kabupaten Sidrap (Sidenreng Rappang), Sulawesi Selatan 91691.
2. Pelaksanaan penelitian selama ( $\pm$ ) 2 bulan.

#### **B. Metode Penelitian**

Untuk membantu suatu kelancaran pengumpulan data, maka penulis menggunakan beberapa metode antara lain :

1. Metode Analisis
  - a) Tinjauan Pustaka (*Literature Review*), dilakukan dengan cara mengumpulkan data-data secara teoritis yang berhubungan dengan masalah yang menjadi objek penelitian dari literatur-literatur serta buku-buku dari perpustakaan yang digunakan sebagai landasan teori.
  - b) Pengamatan (*Observation*), proses mengumpulkan data-data yang diperlukan dengan mengadakan kunjungan secara langsung di Puskesmas Barukku sebagai pelayanan kesehatan tingkat I di Kabupaten Sidrap, mengamati langsung proses pencatatan rekam medis serta bertemu dengan

pihak terkait, data-data yang diperoleh kemudian dituangkan dalam bentuk kata-kata maupun skema.

## 2. Wawancara (*Interview*)

Wawancara merupakan teknik pengumpulan data yang dilakukan melalui tatap muka dan tanya jawab langsung antara pengumpul data terhadap narasumber/sumber data. Metode ini dengan cara mengumpulkan data-data yang diperlukan dengan mengadakan wawancara secara langsung dengan Kepala Puskesmas, Biro Humas dan Tenaga Kesehatan serta pihak-pihak yang terkait.

## 3. Kajian Pustaka

Tahapan penelitian yang dimaksud dalam penelitian ini ada beberapa tahapan yaitu persiapan penelitian, pengumpulan data analisis, perancangan, pengujian dan implementasi. Adapun Uraian dari tahapan tersebut adalah sebagai berikut :

### a) Persiapan Penelitian

Pada tahapan ini peneliti melakukan persiapan penelitian. Persiapan penelitian yang dimaksud adalah menyiapkan buku-buku, artikel-artikel tentang topik penelitian serta *software* yang digunakan selama penelitian.

### b) Studi Literature

Pada tahapan ini peneliti melakukan apa yang disebut dengan kajian pustaka, yaitu mempelajari buku-buku referensi dan hasil penelitian sejenis sebelumnya yang pernah dilakukan oleh orang lain. Tujuannya ialah untuk mendapatkan landasan teori mengenai masalah yang akan

diteliti. Teori merupakan pijakan bagi peneliti untuk memahami persoalan yang diteliti dengan benar dan sesuai dengan kerangka berpikir ilmiah.

### **C. Metode Pengumpulan Data**

Pada tahap ini peneliti melakukan wawancara dengan pihak yang terkait dengan penelitian, serta melakukan pencatatan dan pengamatan langsung di tempat penelitian.

#### 1. Analisis

Pada tahap analisis, peneliti melakukan analisa terhadap sistem yang di terapkan sekarang berdasarkan kemudian merumuskan masalah yang menjadi pokok penelitian sehingga dapat dibuat alternatif pemecahan masalah.

#### 2. Perancangan

Peneliti kemudian merancang aplikasi yang ingin dibuat berdasarkan alternatif pemecahan masalah.

#### 3. Pengujian

Setelah melakukan perancangan, peneliti kemudian menguji hasil perancangan yang telah dibuat. Jika hasil perancangan terdapat kekurangan atau kelemahan maka kembali ke tahap analisis.

#### 4. Implementasi

Setelah pada perancangan tidak terdapat kekurangan maka aplikasi siap untuk digunakan oleh *user*.

## D. Alat dan Bahan

Tahap desain dilakukan untuk menentukan ruang lingkup pembuatan aplikasi yang ditangani, yang selanjutnya dijabarkan dalam bentuk diagram alir. Dalam pembuatan program aplikasi tersebut penulis membutuhkan perangkat keras (*Hardware*) dan perangkat lunak (*Software*). Adapun *Hardware* dan *Software* yang digunakan adalah :

### 1. Perangkat keras (*Hardware*)

Perangkat keras yang digunakan untuk membuat aplikasi menggunakan Laptop atau PC (*Personal Komputer*) dengan spesifikasi sebagai berikut :

- a. *Processor Intel Core i7 – 7500U Processor (up to 4.0GHz)*
- b. *Kapasitas Memory 8 GB DDR 4*
- c. *Kapasitas Harddisk 500 GB*
- d. *Layar Monitor 14.0” LED backlit FHD*

### 2. Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan dalam pembuatan aplikasi adalah :

- a. *Sistem Operasi Windows 10*
- b. *XAMPP Control Panel*
- c. *PHP Sebagai Bahasa Program*
- d. *MySql untuk Database*
- e. *Adobe Dreamweaver CS6 dan bootstrap*
- f. *Android*

### 3. Bahan Penelitian

Berupa data-data rekam medis pasien yang di dapat langsung dari pelayanan fasilitas kesehatan tingkat I yaitu Puskesmas Barukku dalam bentuk observasi dan wawancara langsung yang berkaitan dengan penelitian ini.

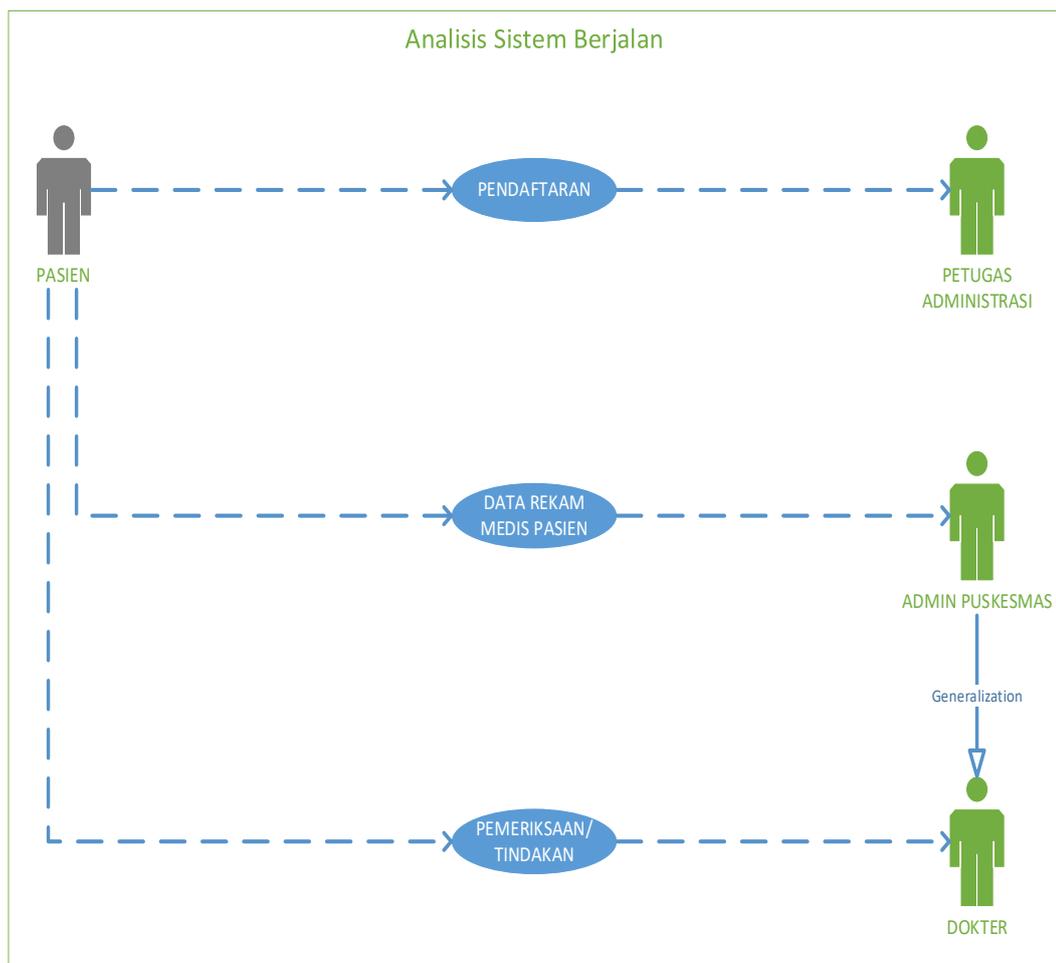
## BAB IV

### HASIL DAN PEMBAHASAN

#### A. Rancangan Sistem

##### 1. Sistem yang berjalan.

Berdasarkan pengamatan dan observasi yang dilakukan oleh penulis, adapun gambaran alur kerja sistem yang sedang berjalan di Puskesmas Barukku adalah sebagai berikut :



**Gambar 4.1** Use Case Diagram Sistem Berjalan

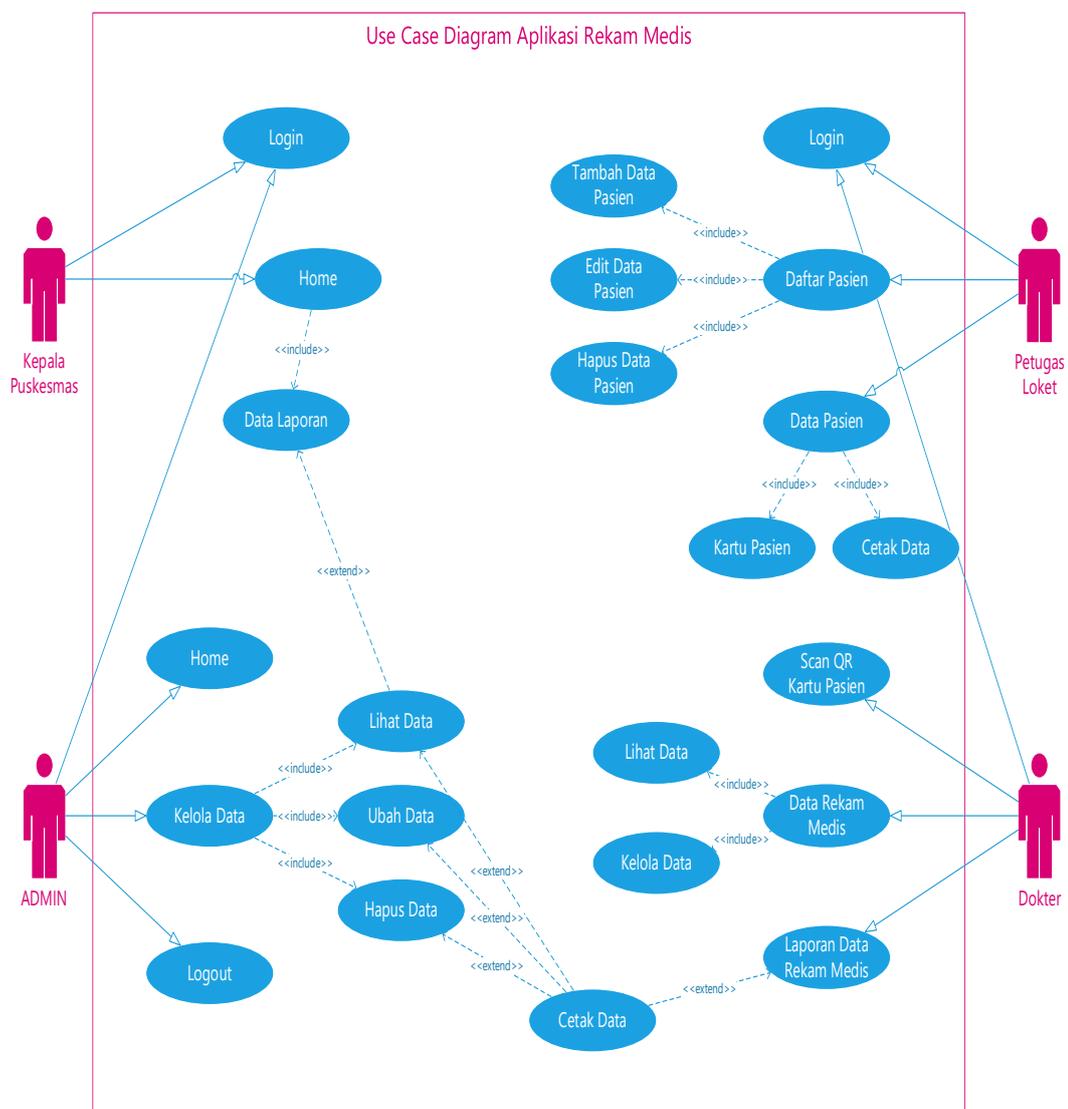
Proses bisnis untuk melakukan kegiatan rekam medis pasien pada Puskesmas Barukku dimulai dari datangnya pasien ke bagian administrasi, kemudian bagian administrasi menanyakan pada pasien tersebut, apakah pasien tersebut merupakan pasien baru atau pasien lama. Jika pasien merupakan pasien baru, maka bagian administrasi akan memberikan formulir pendaftaran kepada pasien untuk diisi oleh pasien sesuai dengan identitas diri. Kemudian setelah pasien mengisi formulir pendaftaran, bagian administrasi melakukan pencatatan identitas pasien pada buku rekam medis dan kartu *member*. Selanjutnya kartu *member* diberikan kepada pasien yang telah melakukan pendaftaran.

Jika pasien tersebut merupakan pasien lama maka bagian administrasi akan meminta kartu *member* pasien, kemudian bagian administrasi mencari kartu rekam medis pasien yang sesuai dengan nomer dan identitas pasien. Selanjutnya kartu tersebut diberikan pada dokter untuk melakukan pemeriksaan atau tindakan, setelah pasien masuk ke ruangan dokter, maka dokter akan melakukan anamnesis dan menanyakan keluhan serta riwayat penyakit yang di derita saat ini. Dokter akan mencatat keluhan dan *diagnosis* ke dalam kartu rekam medis pasien.

Pasien yang tidak memiliki atau yang kehilangan kartu berobat akan dibuatkan buku rekam medis baru. Sehingga pasien yang kehilangan kartu berobat akan memiliki dua buku rekam medis, hal ini dapat mengakibatkan penggandaan data rekam medis pasien pada lemari buku.

## 2. Sistem yang diusulkan.

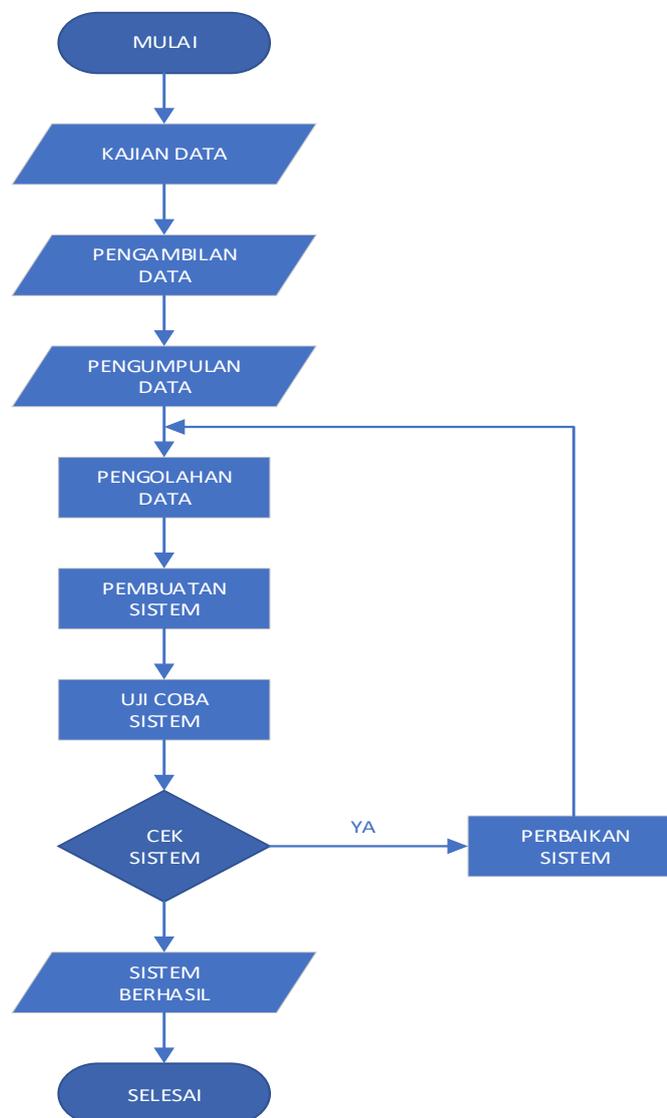
Sistem yang diusulkan menggunakan *Use Case Diagram*. *Use case diagram* menggambarkan interaksi antara actor dengan proses atau sistem yang dibuat. *Use case* meliputi semua hal yang ada pada sistem, sedangkan actor meliputi semua hal yang ada diluar sistem. *Actor* termasuk seseorang atau apa saja yang berhubungan dengan sistem yang dibangun, Adapun sistem yang penulis usulkan dapat dilihat pada gambar berikut :



**Gambar 4.2** Diagram Sistem Diusulkan

### 3. Flowchart

*Flowchart* merupakan suatu bagan yang menunjukkan alur kerja atau apa yang sedang dikerjakan di dalam sistem secara keseluruhan dan menjelaskan urutan dari prosedur-prosedur yang ada di dalam alur penelitian. Alur ini menjelaskan tentang perjalanan atau langkah-langkah dari aplikasi. Berikut merupakan *flowchart* alur system dapat dilihat pada gambar berikut.



**Gambar 4.3** *Flowchart* Sistem

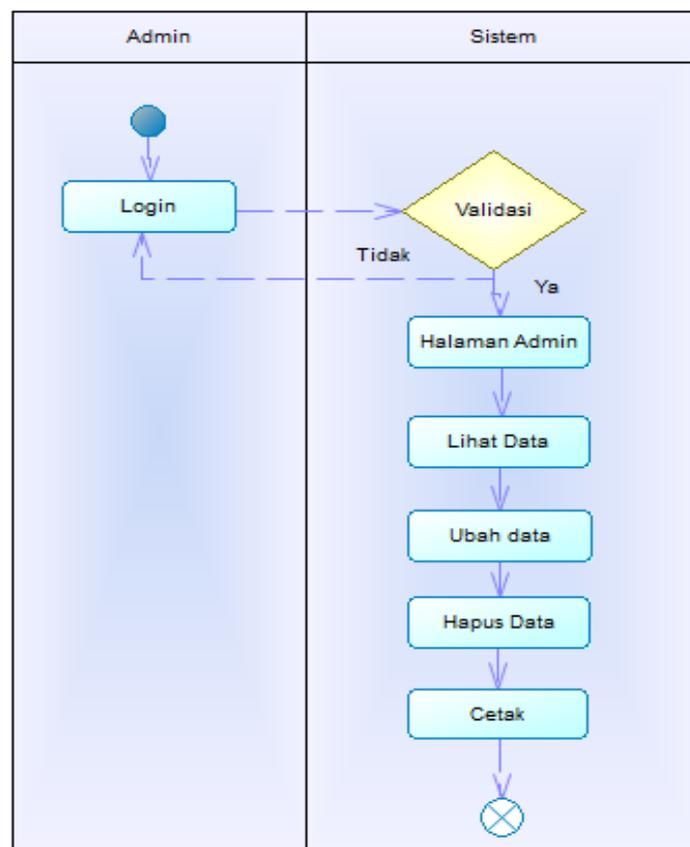
## B. Analisis Aliran Data UML

Analisis aliran data yang diusulkan ini bertujuan mengetahui suatu proses informasi, dalam analisis sistem penulis menggunakan pengembangan orientasi objek sehingga menggunakan UML.

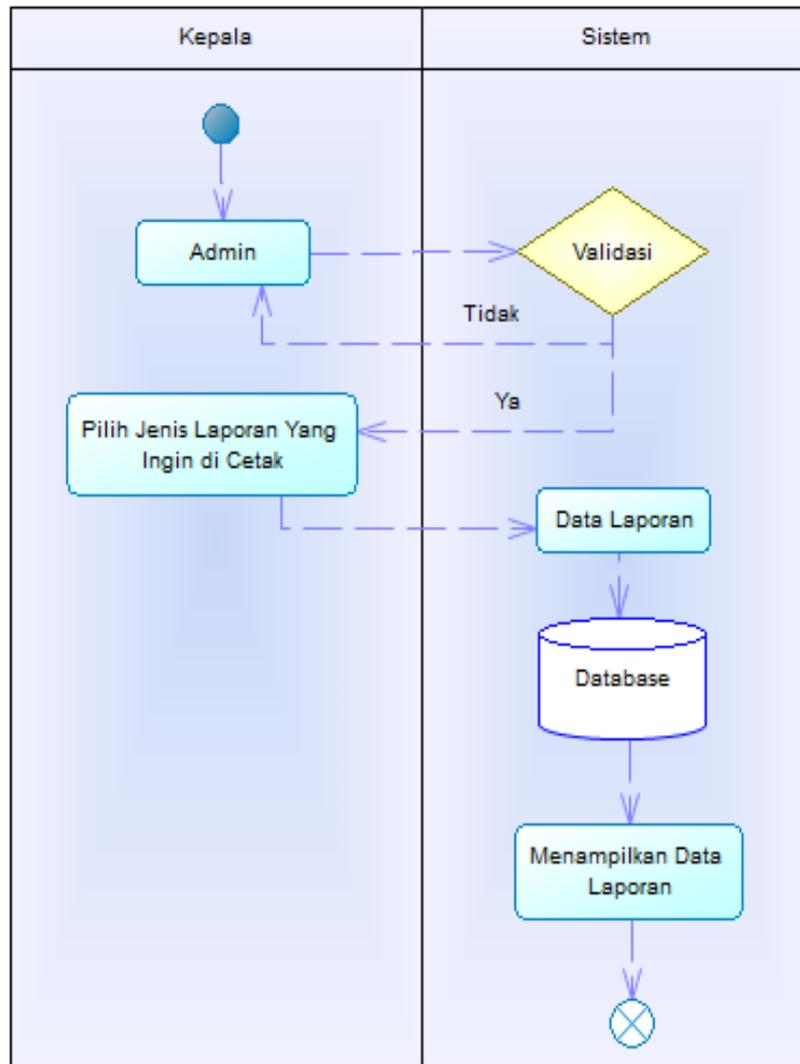
### 1. Activity Diagram

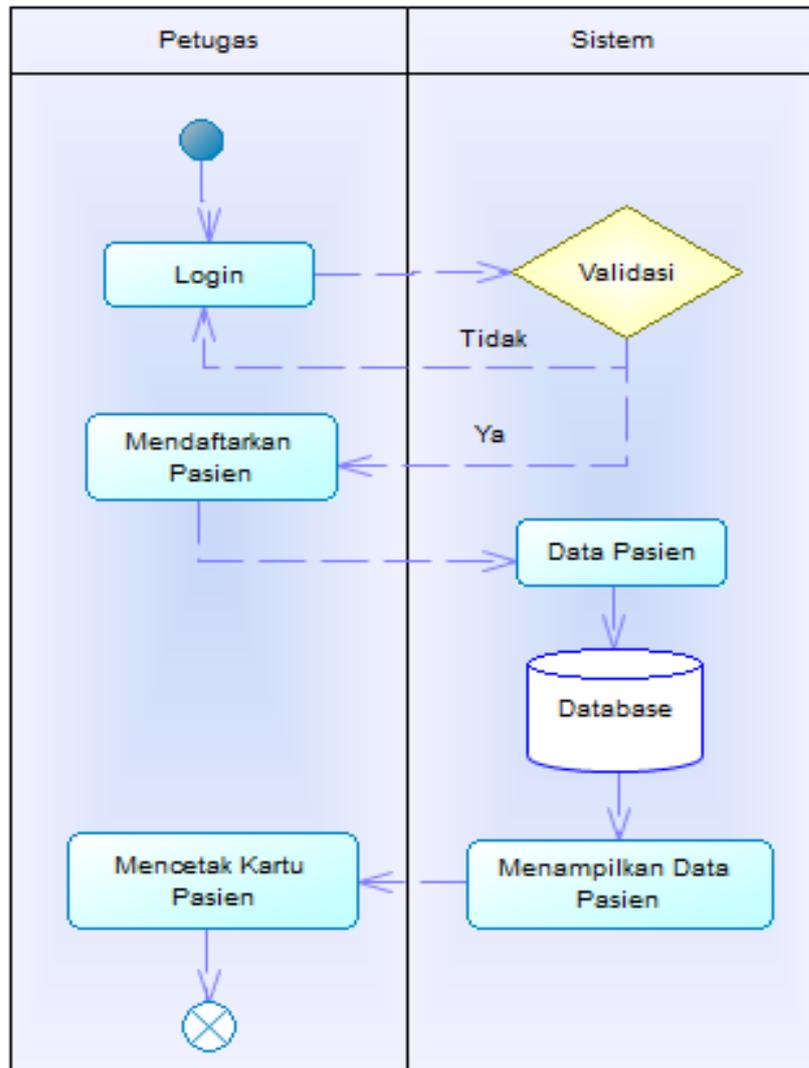
*Activity diagram* adalah adalah suatu bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak *Activity diagram* ini menjelaskan tentang aktifitas-aktifitas yang terjadi dalam sebuah aliran proses pada sebuah *system*.

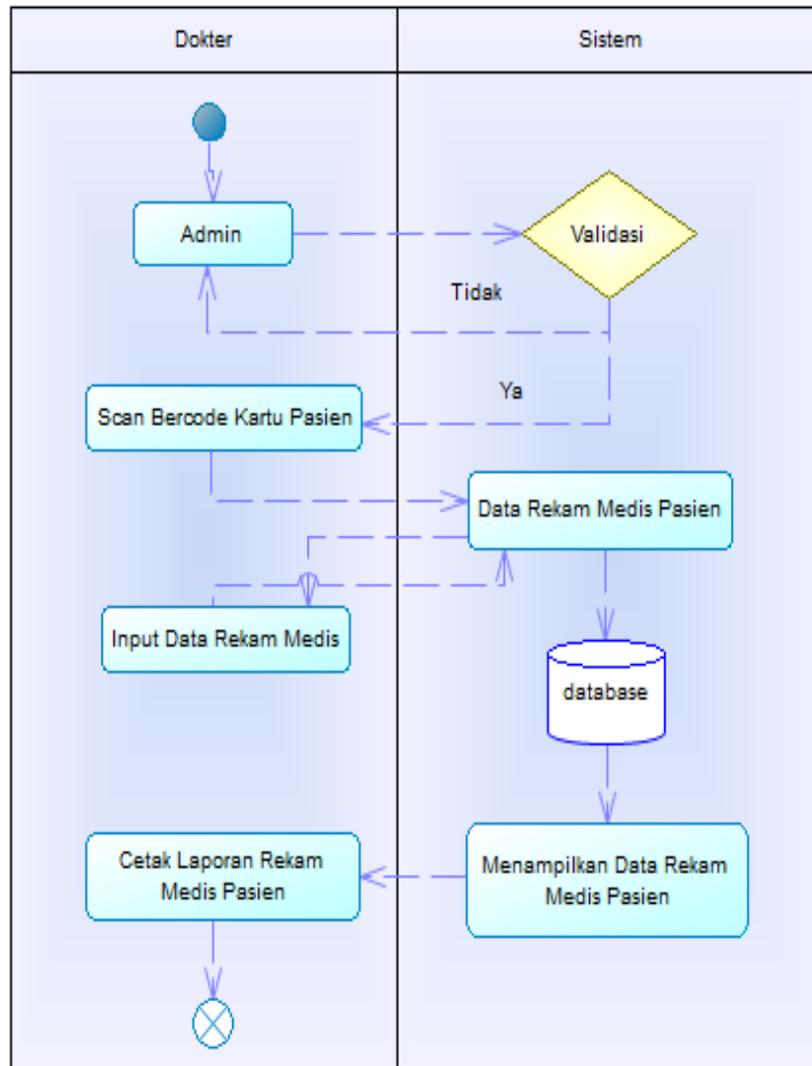
#### a. Activity Diagram Admin



**Gambar 4.4** Activity Diagram Admin

b. *Activity Diagram* Kepala Puskesmas**Gambar 4.5** *Activity Diagram* Kepala Puskesmas

c. *Activity Diagram* Petugas Loker**Gambar 4.6** *Activity Diagram* Petugas Loker

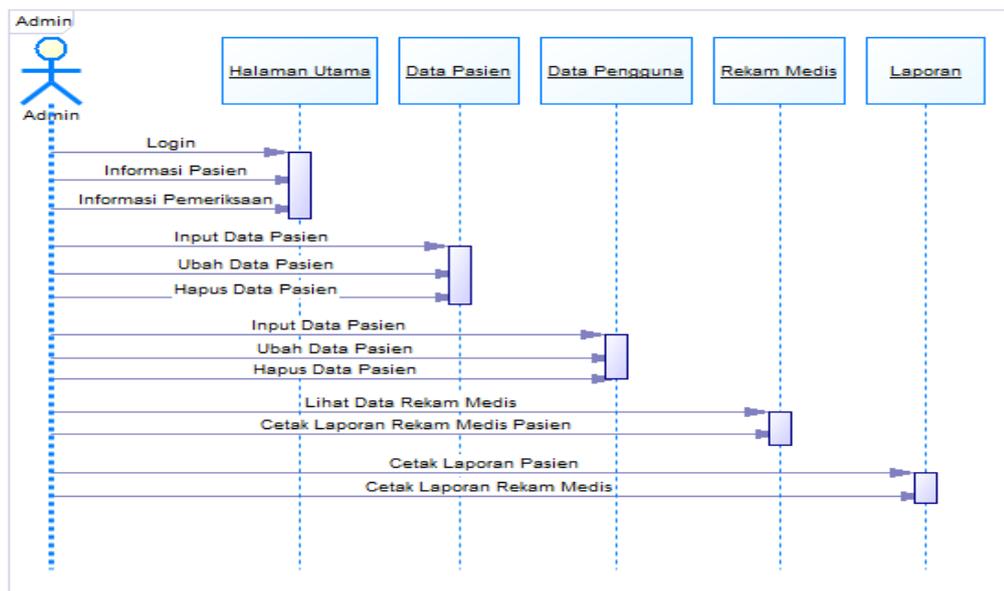
d. *Activity Diagram Dokter*

**Gambar 4.7** *Activity Diagram Dokter*

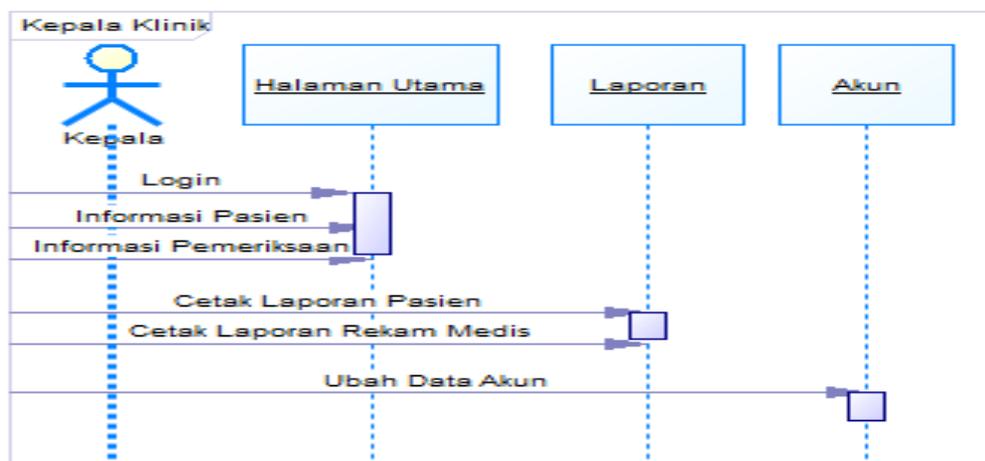
## 2. Sequence Diagram

*Sequence Diagram* merupakan salah satu *diagram interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan *message* (pesan) apa yang dikirim dan kapan pelaksanaannya.

### a. Sequence Diagram Admin

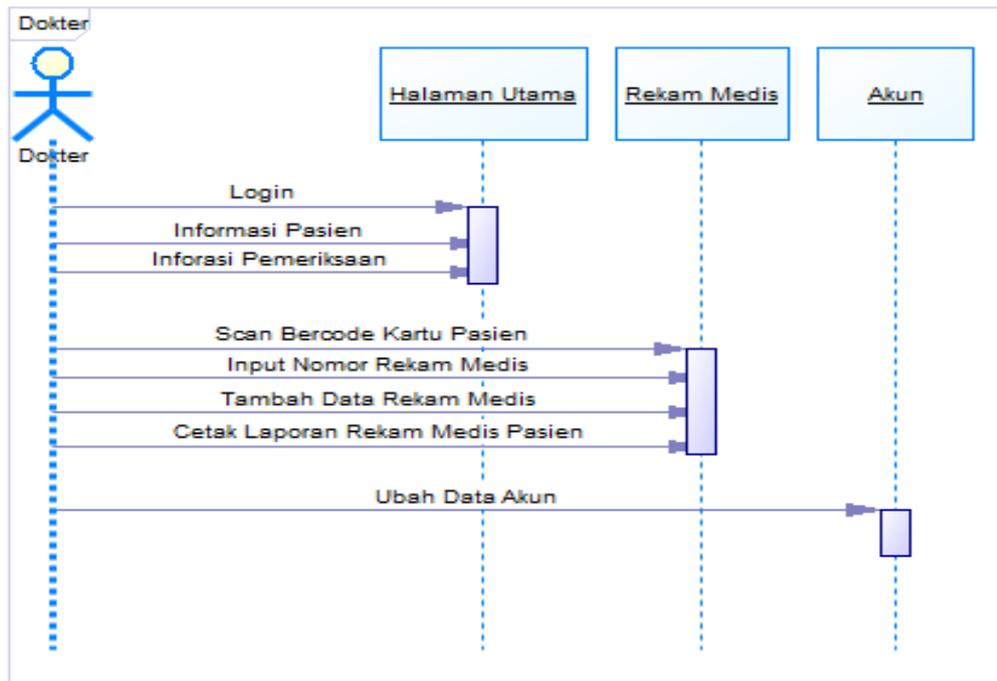


**Gambar 4.8** *Sequence Diagram Admin*



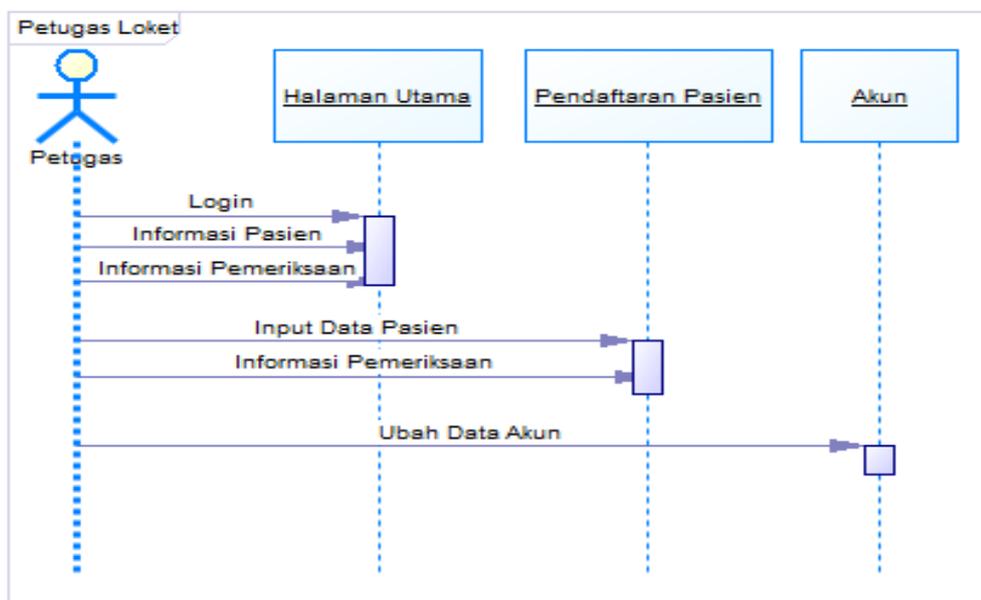
**Gambar 4.9** *Sequence Diagram Kepala Puskesmas*

b. *Sequence Diagram Dokter*



**Gambar 4.10** *Sequence Diagram Dokter*

c. *Sequence Diagram Petugas Loket*



**Gambar 4.11** *Sequence Diagram Petugas Loket*

### C. Implementasi

Pembuatan basis data yang akan dirancang berfungsi sebagai media penyimpanan yang dibutuhkan *system*. Rincian tabel dibawah ini merupakan *table* yang digunakan oleh *system* yang diusulkan:

**Tabel 4.1** Tabel Pasien

Nama Tabel		tbl_pasien		
No	Nama	Type	Size	Keterangan
1	Id	<i>Int</i>	11	<i>AUTO_INCREMENT</i>
2	Nama	<i>Varchar</i>	50	
3	Tanggal lahir	<i>Date</i>	-	
4	Alamat	<i>Varchar</i>	50	
5	Jenis Kelamin	<i>Varchar</i>	10	
6	Golongan Darah	<i>Char</i>	2	
7	No Rekam Medis	<i>Varchar</i>	10	
8	Tgl Pendaftaran	<i>Datetime</i>	-	

**Tabel 4.2** Tabel Rekam Medis

Nama Tabel		tbl_rekam_medis		
No	Nama	Type	Size	Keterangan
1	Id	<i>Int</i>	11	<i>AUTO_INCREMENT</i>
2	No_Rekam_Medis	<i>Varchar</i>	10	
3	Id_Dokter	<i>Varchar</i>	10	

4	Keterangan	<i>Longtext</i>	-	
5	TanggalPemeriksaan	<i>Datetime</i>	-	

**Tabel 4.3** Tabel *User*

Nama Tabel		tbl_user		
No	Nama	Type	Size	Keterangan
1	Id	<i>Int</i>	11	<i>AUTO_INCREMENT</i>
2	Nama	<i>Varchar</i>	25	
3	<i>Username</i>	<i>Varchar</i>	25	
4	<i>Password</i>	<i>Varchar</i>	25	
5	<i>Email</i>	<i>Varchar</i>	50	
6	Level	<i>Varchar</i>	25	

Berikut adalah detail dari proses normalisasi untuk tabel-tabel yang ada, dengan penjelasan Langkah demi langkah untuk mencapai 1NF, 2NF, dan 3NF. Setiap tahap menunjukkan bagaimana tabel disederhanakan untuk menghilangkan redundansi dan ketergantungan yang tidak perlu.

**Tabel 4.4** Pasien Struktur Awal

Id	Nama	Tanggal_lahir	Alamat	Jenis_Kelamin	Golongan_Darah	No_Rekam_Medis
PK						

a) 1NF (*First Normal Form*)

- 1) Kriteria: Semua kolom harus memiliki nilai atomik, dan tidak ada grup nilai berulang.

2) Status: Tabel sudah dalam 1NF karena setiap kolom memiliki nilai atomik.

b) 2NF (*Second Normal Form*)

1) Kriteria: Tabel harus berada dalam 1NF dan semua kolom non-primer harus bergantung sepenuhnya pada primary key.

2) Status: Tabel sudah dalam 2NF karena semua kolom non-primer (Nama, Tanggal\_lahir, Alamat, Jenis\_Kelamin, Golongan\_Darah, No\_Rekam\_Medis, Tgl\_Pendaftaran) bergantung sepenuhnya pada primary key (Id).

c) 3NF (*Third Normal Form*)

1) Kriteria: Tabel harus berada dalam 2NF dan tidak ada ketergantungan transitif, yaitu kolom non-primer tidak bergantung pada kolom non-primer lainnya.

2) Status: Tabel sudah dalam 3NF karena tidak ada kolom non-primer yang bergantung transitif pada Id. Semua kolom non-primer bergantung langsung pada Id.

**Tabel 4.5** Rekam Medis Struktur Awal

Id	No_Rekam_Medis	Id_Dokter	Keterangan	Tanggal_Pemeriksaan
PK		FK		

d) 1NF (*First Normal Form*)

3) Kriteria: Semua kolom harus memiliki nilai atomik, dan tidak ada grup nilai berulang.

4) Status: Tabel sudah dalam 1NF karena setiap kolom memiliki nilai atomik.

e) 2NF (*Second Normal Form*)

- 3) Kriteria: Tabel harus berada dalam 1NF dan semua kolom non-primer harus bergantung sepenuhnya pada primary key.
  - 4) Status: Tabel sudah dalam 2NF karena semua kolom non-primer (Nama, Tanggal\_lahir, Alamat, Jenis\_Kelamin, Golongan\_Darah, No\_Rekam\_Medis, Tgl\_Pendaftaran) bergantung sepenuhnya pada primary key (Id).
- f) 3NF (*Third Normal Form*)
- 1) Kriteria: Tabel harus berada dalam 2NF dan tidak ada ketergantungan transitif, yaitu kolom non-primer tidak bergantung pada kolom non-primer lainnya.
  - 2) Status: Tabel sudah dalam 3NF karena tidak ada kolom non-primer yang bergantung transitif pada Id. Semua kolom non-primer bergantung langsung pada Id.

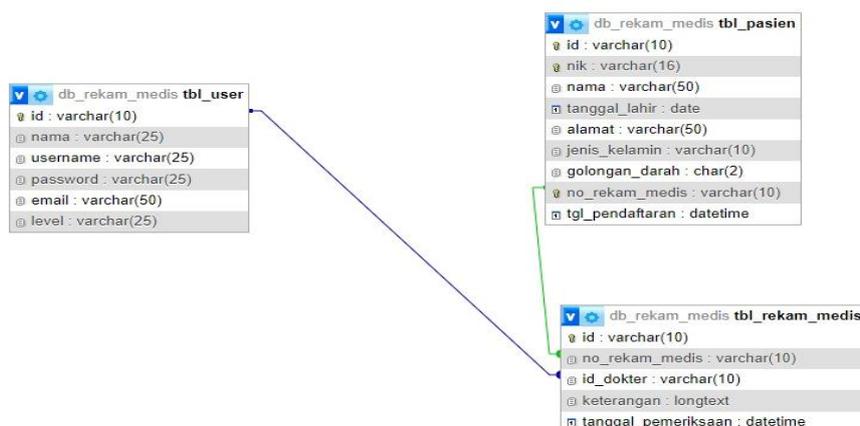
**Tabel 4.6** *User* Struktur Awal

Id	Nama	Username	Password	Email	Level
PK					

- a) 1NF (*First Normal Form*)
- 1) Kriteria: Semua kolom harus memiliki nilai atomik, dan tidak ada grup nilai berulang.
  - 2) Status: Tabel sudah dalam 1NF karena setiap kolom memiliki nilai atomik.
- b) 2NF (*Second Normal Form*)

- 1) Kriteria: Tabel harus berada dalam 1NF dan semua kolom non-primer harus bergantung sepenuhnya pada primary key.
  - 2) Status: Tabel sudah dalam 2NF karena semua kolom non-primer (Nama, Username, Password, Email, Level) bergantung sepenuhnya pada primary key (Id).
- c) 3NF (*Third Normal Form*)
- 1) Kriteria: Tabel harus berada dalam 2NF dan tidak ada ketergantungan transitif, yaitu kolom non-primer tidak bergantung pada kolom non-primer lainnya.
  - 2) Status: Tabel sudah dalam 3NF karena tidak ada kolom non-primer yang bergantung transitif pada Id. Semua kolom non-primer bergantung langsung pada Id.

Berikut adalah diagram ERD yang menggambarkan relasi antar tabel yang sudah dinormalisasi:



**Gambar 4.12** Entity Relationship Diagram (ERD) Rekam Medis

Diagram ERD ini memberikan gambaran umum tentang struktur data dan bagaimana tabel-tabel saling berhubungan. Ini membantu dalam perancangan database dan memastikan bahwa data disimpan dengan cara yang terorganisir dan efisien. Diagram ini akan menggambarkan hubungan antar tabel dengan kunci utama (*primary key*) dan kunci asing (*foreign key*). Berikut detail gambaran diagram relasi.

a) Tabel Pasien (tbl\_pasien)

1) Primary Key (PK): 'id'

2) Attributes:

(Nama, Tanggal\_lahir, Alamat, Jenis\_Kelamin, Golongan\_Darah,  
No\_Rekam\_MedisTgl\_Pendaftaran)

b) Tabel Rekam Medis (tbl\_rekam\_medis)

1) Primary Key (PK): 'id'

2) Foreign Key (FK); 'No\_Rekam\_Medis (merujuk ke No\_Rekam\_Medis di  
tbl\_pasien)'

3) Attributes:

(No\_Rekam\_Medis, Id\_Dokter, Keterangan, Tanggal\_Pemeriksaan)

c) Tabel *User* (tbl\_user)

1) Primary Key (PK): 'id'

2) Attributes:

(Nama, Username, Password, Email, Level)

d. Relasi antara tbl\_pasien dan tbl\_rekam\_medis:

- 1) No\_Rekam\_Medis di tbl\_rekam\_medis adalah foreign key yang merujuk ke kolom No\_Rekam\_Medis di tbl\_pasien.
- 2) Ini menunjukkan bahwa setiap rekam medis berhubungan dengan satu pasien tertentu melalui nomor rekam medis.

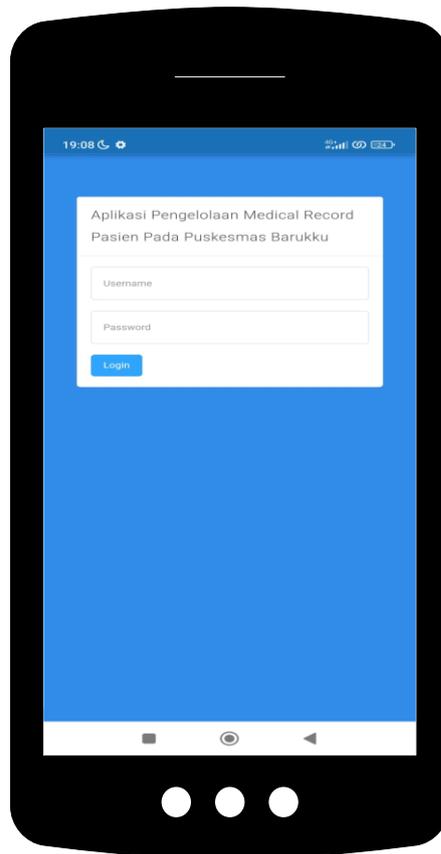
e. Tabel tbl\_user:

- 1) Tabel ini tidak memiliki relasi langsung dengan tabel lain berdasarkan informasi yang diberikan.
- 2) Tabel ini digunakan untuk mengelola data pengguna sistem.

Diagram relasi ini menggambarkan bagaimana data pasien dihubungkan dengan data rekam medis melalui nomor rekam medis, sementara data pengguna dikelola secara terpisah tanpa hubungan langsung dengan entitas lain dalam sistem.

## D. Detail Aplikasi

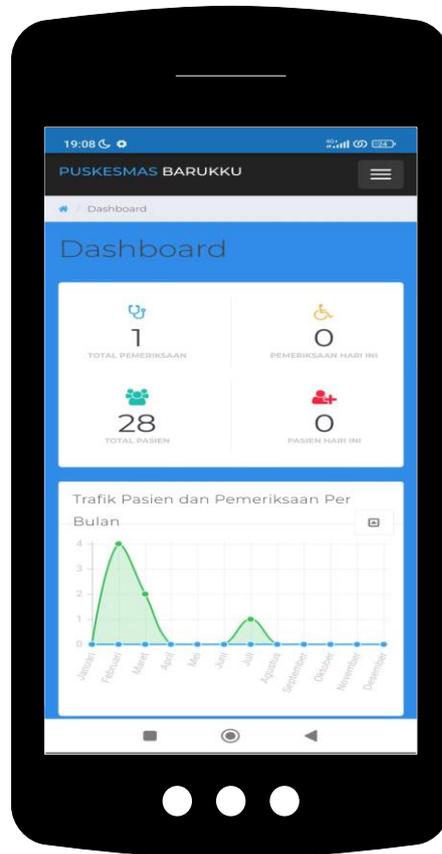
1. *Admin*
  - a. Halaman *Login Admin*



**Gambar 4.13** Halaman *Login Admin*

*Login* merupakan tampilan awal program yang digunakan untuk masuk ke akun *admin*.

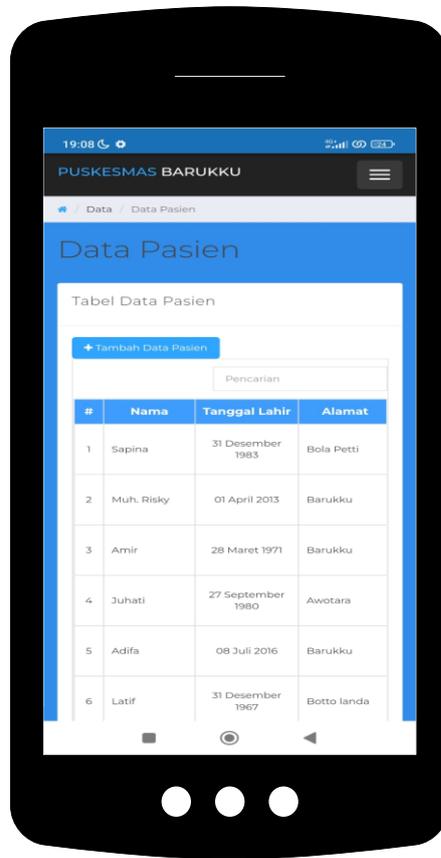
b. Halaman *Home Admin*



**Gambar 4.14** Halaman *Home Admin*

*Dashboard* yang berfungsi menampilkan informasi pemeriksaan dan pasien yang telah di *input*.

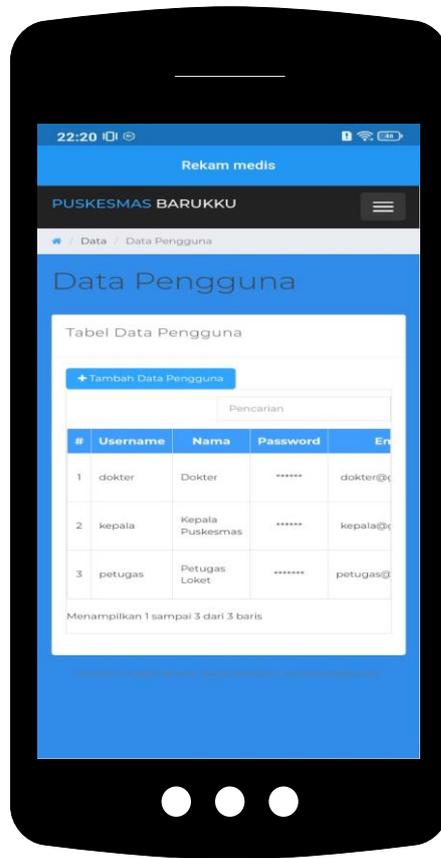
c. Halaman Data Pasien



**Gambar 4.15** Halaman Data Pasien

Halaman Data Pasien berfungsi menampilkan informasi dan mengelola data pasien.

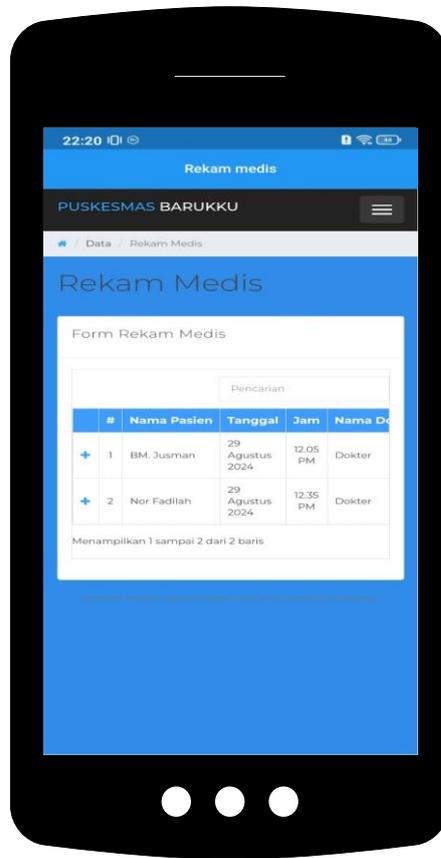
## d. Halaman Data Pengguna



**Gambar 4.16** Halaman Data Pengguna

Halaman Data Pengguna berfungsi menampilkan informasi dan mengelola data pengguna (*user*).

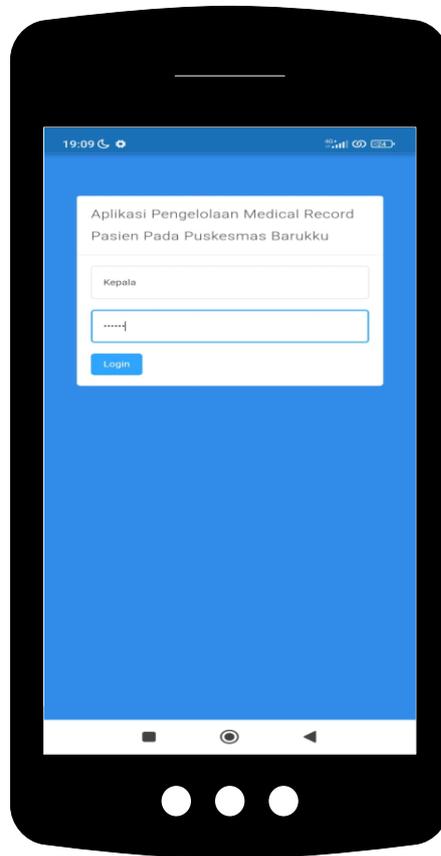
## e. Halaman Rekam Medis



**Gambar 4.17** Halaman Rekam Medis

Halaman Rekam Medis berfungsi untuk melihat rekam medis pasien yang telah di *input* oleh dokter dan dapat mencetak laporan rekam medis pasien.

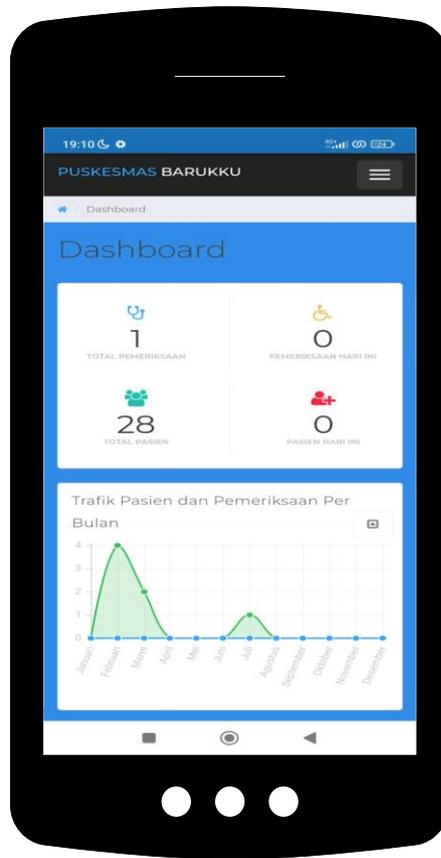
2. Kepala Puskesmas
  - a. Halaman *Login* Kepala Puskesmas



**Gambar 4.18** Halaman *Login* Kepala Puskesmas

*Login* merupakan tampilan awal program yang digunakan untuk masuk ke akun kepala puskesmas.

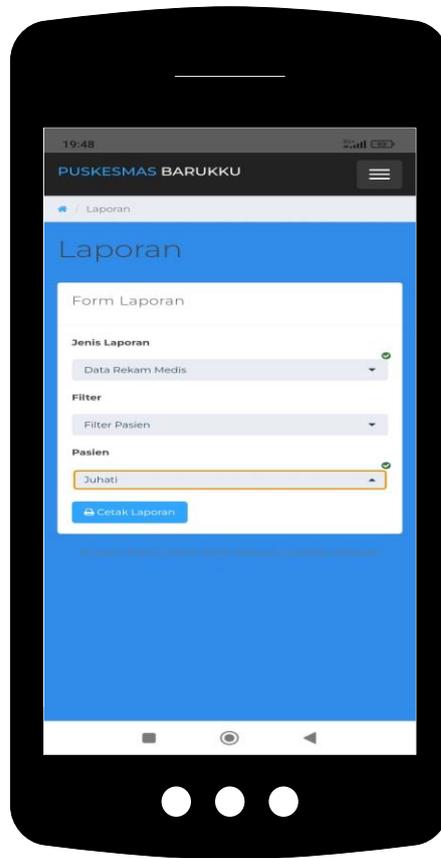
b. Halaman *Home* Kepala Puskesmas



**Gambar 4.19** Halaman *Home* Kepala Pukesmas

*Dashboard* yang berfungsi menampilkan informasi pemeriksaan dan pasien yang telah di *input*.

c. Halaman Laporan



**Gambar 4.20** Halaman Laporan

Halaman Laporan berfungsi untuk mencetak laporan data pasien dan laporan data rekam medis.

d. Halaman Akun Kepala Puskesmas

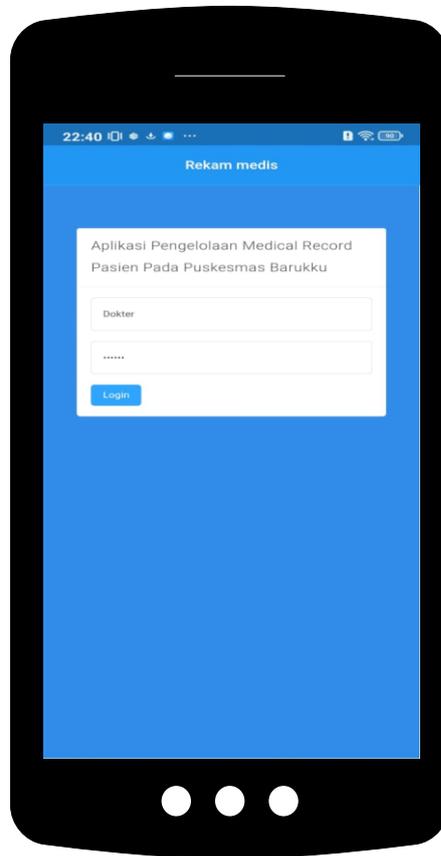


**Gambar 4.21** Halaman Akun Kepala Puskesmas

Halaman Akun berfungsi menampilkan informasi dan mengelola data akun pengguna (*user*).

### 3. Dokter

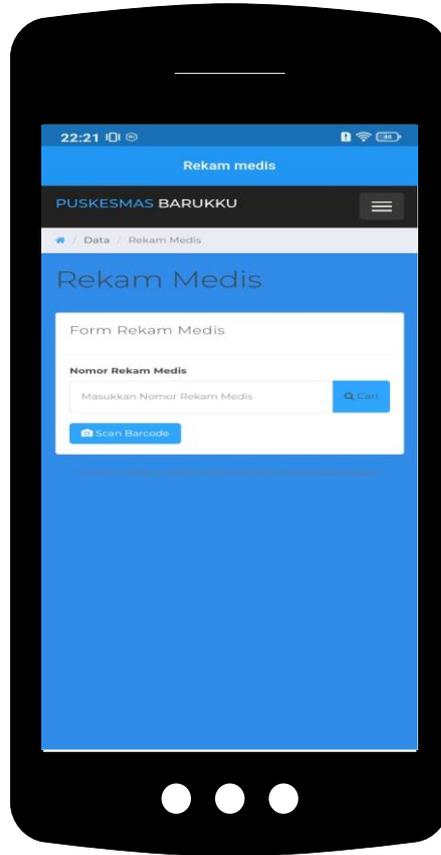
#### a. Halaman *Login* Dokter



**Gambar 4.22** halaman *Login* Dokter

*Login* merupakan tampilan awal program yang digunakan untuk masuk ke akun Dokter.

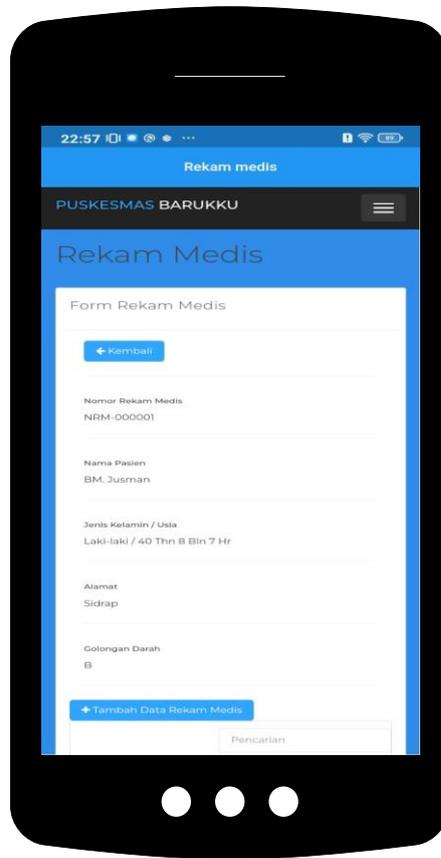
b. Halaman *Form* Pencarian Rekam Medis Dokter



**Gambar 4.23** Halaman *Form* Rekam Medis Dokter

Halaman *form* pencarian rekam medis dokter yang berfungsi menampilkan informasi pemeriksaan pasien dan scan *QR Code* kartu pasien.

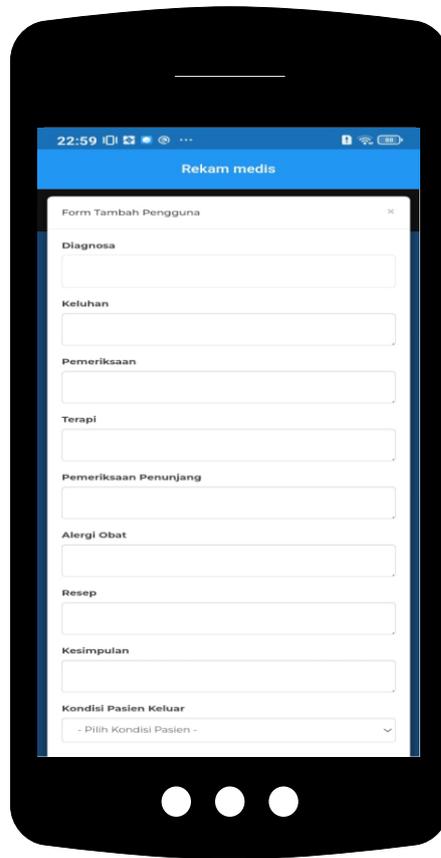
c. Halaman Hasil Pencarian Rekam Medis Dokter



**Gambar 4.24** Halaman Hasil Pencarian Rekam Medis Dokter

Halaman menu hasil pencarian rekam medis berfungsi menampilkan informasi data pasien berdasarkan diagnosis penyakit.

d. Halaman Menu Tambah Data Pemeriksaan Dokter



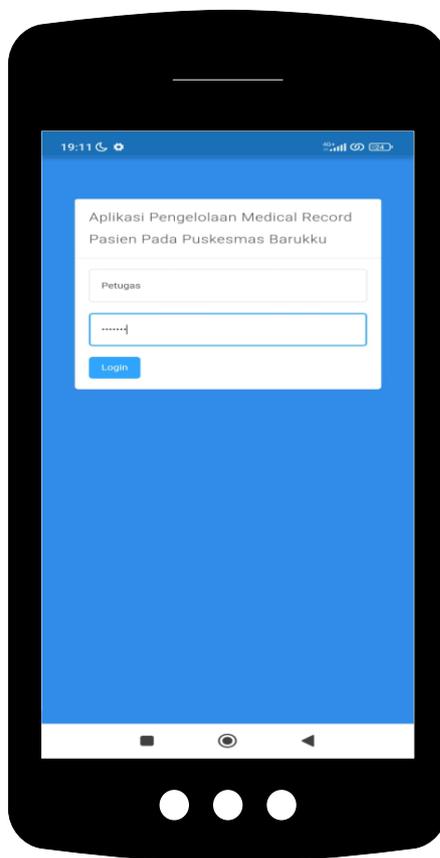
The image shows a mobile application interface for adding doctor examination data. The screen is titled "Rekam medis" and displays a form titled "Form Tambah Pengguna". The form contains several input fields for "Diagnosa", "Keluhan", "Pemeriksaan", "Terapi", "Pemeriksaan Penunjang", "Alergi Obat", "Resep", and "Kesimpulan". At the bottom, there is a dropdown menu for "Kondisi Pasien Keluar" with the option "- Pilih Kondisi Pasien -".

**Gambar 4.25** Halaman Tambah Data Pemeriksaan Dokter

Halaman tambah data pemeriksaan dokter berfungsi menambahkan rekam medis pemeriksaan pasien berdasarkan hasil pemeriksaan dokter.

#### 4. Petugas Locket

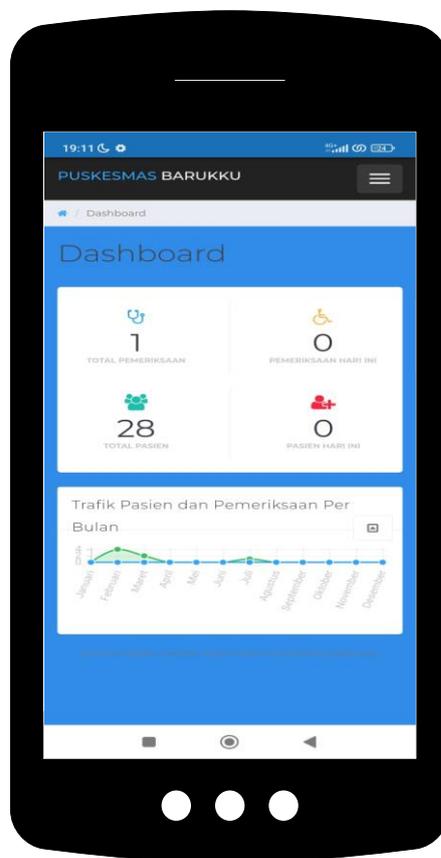
##### a. Halaman *Login* Petugas Locket



**Gambar 4.26** Halaman *Login* Petugas Locket

*Login* merupakan tampilan awal program yang digunakan untuk masuk ke akun Petugas Locket.

b. Halaman *Home* Petugas Loker



**Gambar 4.27** Halaman *Home* Petugas Loker

*Dashboard* yang berfungsi menampilkan informasi pemeriksaan dan pasien yang telah di *input*.

c. Halaman Pendaftaran Pasien



The image shows a mobile application interface for patient registration. The screen displays the following fields and options:

- Nomor Rekam Medis:** NRM-000031
- NIK:** NIK Pasien
- Nama:** Nama Pasien
- Tanggal Lahir:** [Date Picker]
- Alamat:** Alamat
- Jenis Kelamin:** - Pilih Jenis Kelamin -
- Golongan Darah:** - Pilih Golongan Darah -

**Gambar 4.28** Halaman Pendaftaran Pasien

Halaman Pendaftaran Pasien berfungsi untuk mendaftarkan pasien baru dan mencetak kartu pasien.

d. Halaman Akun Petugas Locket

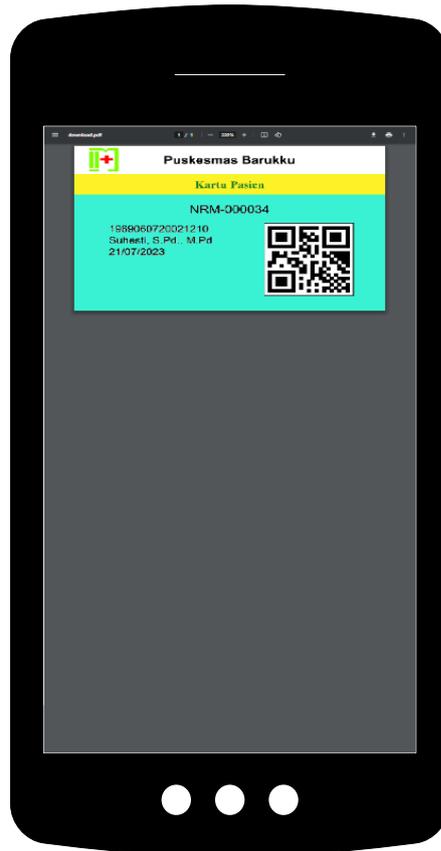


**Gambar 4.29** Halaman Petugas Locket

Halaman Akun berfungsi menampilkan informasi dan mengelola data akun pengguna (*user*).

#### D. Rancangan *Output* Program

Perancangan *output* program merupakan perancangan yang dilakukan untuk mengetahui informasi berupa suatu Aplikasi *Mobile* Rekam Medis Pasien Berdasarkan NIK Pada Pelayanan Kesehatan Menggunakan *QR Code*.



**Gambar 4.30** Halaman Kartu Pasien

### E. Alat dan Bahan

Adapun kebutuhan yang digunakan untuk Membangun sebuah sistem dengan judul “Aplikasi *Mobile* Rekam Medis Pasien Berdasarkan NIK Pada Pelayanan Kesehatan Menggunakan *QR Code*”.

#### 1. Kebutuhan *Hardware*

Spesifikasi *hardware* pengguna aplikasi ini adalah sebagai berikut:

**Tabel 4.7** Spesifikasi *Hardware* Komputer

Jenis	Spesifikasi
<i>Laptop/Komputer</i>	<i>Acer</i>
<i>Processor</i>	<i>Processor Intel Core i7 – 7500U Processor (up to 4.0GHz)</i>
<i>Memory</i>	<i>8 GB DDR 4 2400MHz</i>
<i>Hardisk</i>	<i>500 GB</i>

**Tabel 4.8** Spesifikasi *Smartphone Admin*

Jenis	Spesifikasi
Jenis <i>Smartphone</i>	Vivo Y69
Layar	5 inch'
Ram	4 GB

**Tabel 4.9** Spesifikasi *Smartphone User*

<b>Jenis</b>	<b>Spesifikasi</b>
<i>Smartphone</i>	<i>Xiaomi note 5 pro</i>
Layar	6 inch'
<i>Ram</i>	4 GB

## 2. Kebutuhan *Software*

Spesifikasi *software* pengguna aplikasi ini sebagai berikut :

**Tabel 4.10** Spesifikasi *Software* Komputer

<b>Jenis</b>	<b>Spesifikasi</b>
Sistem Operasi	<i>Windows 10 Ultimate</i>

**Tabel 4.11** Spesifikasi *Software Smartphone Admin*

<b>Jenis</b>	<b>Spesifikasi</b>
Versi <i>Android</i>	<i>7.1 Nougat</i>
Versi perangkat lunak	PD1705F_EX_A_1.18.0

**Tabel 4.12** Spesifikasi *Software Smartphone User*

<b>Jenis</b>	<b>Spesifikasi</b>
Versi <i>Android</i>	<i>Android 9 Pie</i>
Versi <i>MIUI</i>	<i>MIUI 10 Global 9.6.27 Beta</i>

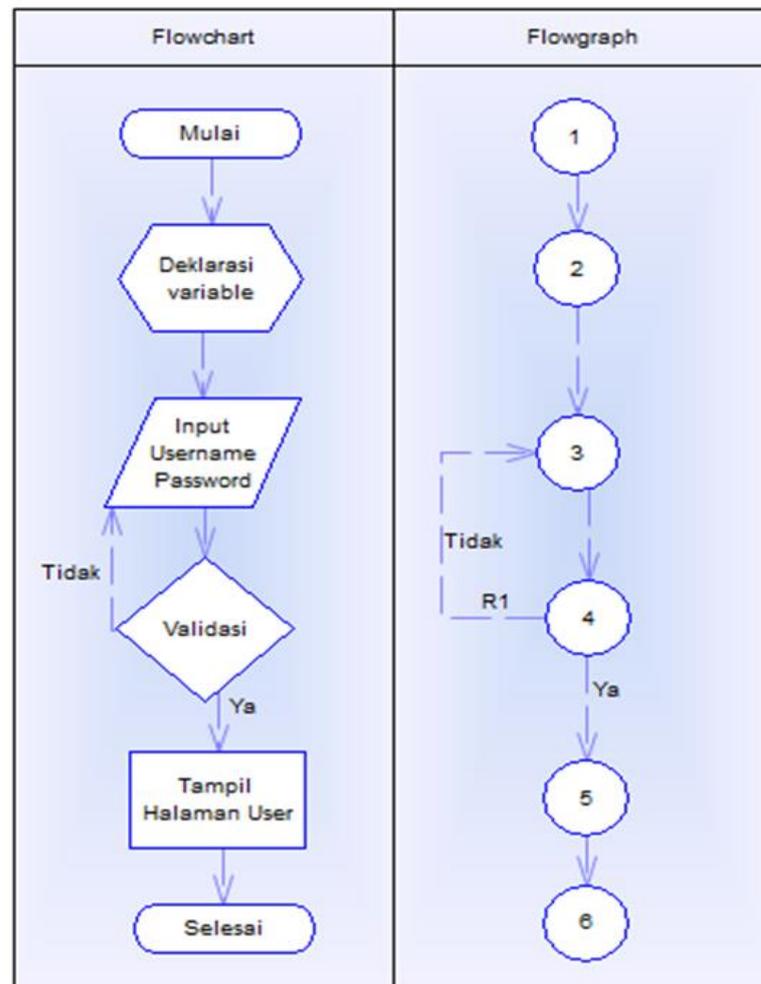
## **F. Metode pengujian**

Pengujian sistem merupakan proses eksekusi suatu program atau sistem secara keseluruhan. Pengujian menunjukkan bahwa fungsi dari perangkat lunak bekerja sesuai dengan spesifikasi dan bahwa persyaratan kinerja telah dipenuhi.

Metode pengujian yang digunakan ada *black-box*. Pengujian blackbox perangkat lunak dilakukan untuk memperlihatkan bahwa masing-masing fungsi telah beroperasi sepenuhnya dan secara bersamaan pula mencari kesalahan pada setiap fungsi.

## **G. Teknik Pengujian**

Pengujian *black-box* berkaitan dengan pengujian yang dilakukan pada interface perangkat lunak. Meskipun didesain untuk mengungkapkan kesalahan, pengujian *black-box* digunakan untuk memperlihatkan bahwa fungsi-fungsi perangkat lunak adalah operasional, bahwa *input* diterima dengan baik dan *output* dihasilkan dengan tepat, dan integritas informasi *external* (seperti file data) dipelihara. Pengujian *black-box* menguji beberapa aspek dasar suatu sistem dengan sedikit memperhatikan struktur logika *internal* perangkat lunak tersebut.

1. Pengujian *White Box*a) *Login*

**Gambar 4.31** Pengujian *Login*

Dari gambar *flowgraph* Menu *Login* diatas dapat dilakukan proses perhitungan sebagai berikut:

- 1) Menghitung *Cyclomatic Complexity*  $V(G)$  dari *Edge* dan *Node*:

Dengan Rumus :  $V(G) = E - N + 2$

Dengan  $E$  (*edge*) = 6

$N$  (*node*) = 6

$$\text{Predikat Node (P)} = 1$$

$$\begin{aligned} \text{Penyelesaian : } V(G) &= E - N + 2 \\ &= 6 - 6 + 2 \\ &= 2 \end{aligned}$$

$$\begin{aligned} \text{Predikat (P)} &= P + 1 \\ &= 1 + 1 \\ &= 2 \end{aligned}$$

2) Berdasarkan perhitungan *Cyclomatic Complexity* dari *Flowgraph* diatas memiliki *Region* = 2

3) *Independent Path* pada *flowgraph* diatas adalah :

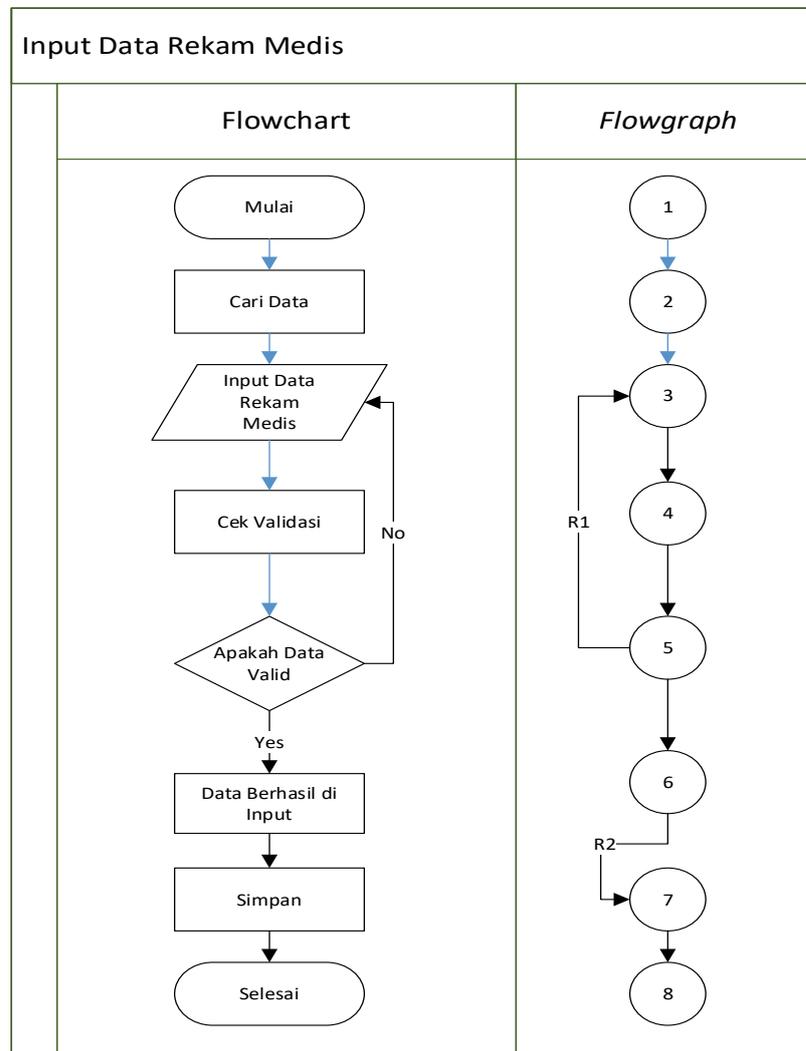
$$\text{Path 1} = 1 - 2 - 3 - 4 - 3$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 5 - 6$$

4) *Grafis Matriks*

**Tabel 4.13** *Grafik Matriks* dari *Flowchart Login*

	1	2	3	4	5	6	E-1
1		1					1-1=0
2			1				1-1=0
3				1			1-1=0
4			1		1		2-1=1
5						1	1-1=0
6							0
Sum (E) + 1							1+1=2

b) *Input Data*

**Gambar 4.32** Pengujian *Input Data* Rekam Medis

Dari gambar *flowgraph* *Input Data* diatas dapat dilakukan proses perhitungan sebagai berikut:

- 1) Menghitung *Cyclomatic Complexity*  $V(G)$  dari *Edge* dan *Node*:

Dengan Rumus :  $V(G) = E - N + 2$

Dengan  $E$  (*edge*) = 7

$N$  (*node*) = 7

$$\text{Predikat Node (P)} = 1$$

$$\begin{aligned} \text{Penyelesaian : } V(G) &= E - N + 2 \\ &= 7 - 7 + 2 = 2 \end{aligned}$$

$$\begin{aligned} \text{Predikat (P)} &= P + 1 \\ &= 1 + 1 = 2 \end{aligned}$$

2) Berdasarkan perhitungan *Cyclomatic Complexcity* dari *Flowgraph* diatas memiliki *Region* = 2

3) *Independent Path* pada *flowgraph* diatas adalah :

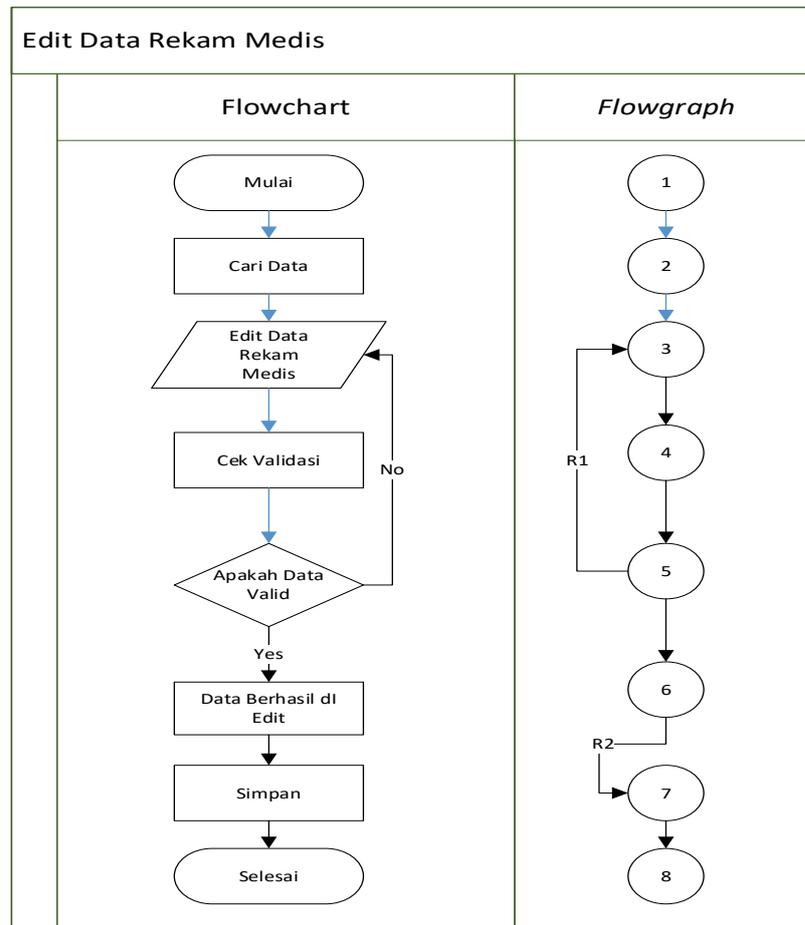
$$\text{Path 1} = 1 - 2 - 3 - 4 - 2$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 5 - 6 - 7$$

4) Grafik *Matriks*

**Tabel 4.14** Grafik *Matriks* pada *Flowgraph Input Data Rekam Medis*

	1	2	3	4	5	6	7	$E - 1 = 0$
1		1						$1 - 1 = 0$
2			1					$1 - 1 = 0$
3				1				$1 - 1 = 0$
4		1			1			$2 - 1 = 1$
5						1		$1 - 1 = 0$
6							1	$1 - 1 = 0$
7								0
Sum (E) + 1								$1 + 1 = 2$

c) *Edit Data*Tabel 4.17 Pengujian *Edit Data* Rekam Medis**Gambar 4.33** Pengujian *Edit Data* Rekam Medis

Dari gambar *flowgraph* *Edit Data* diatas dapat dilakukan proses perhitungan sebagai berikut:

- 1) Menghitung *Cyclomatic Complexity*  $V(G)$  dari *Edge* dan *Node*:

Dengan Rumus :  $V(G) = E - N + 2$

Dengan  $E$  (*edge*) = 8

$N$  (*node*) = 8

$$\text{Predikat Node (P)} = 1$$

$$\begin{aligned} \text{Penyelesaian : } V(G) &= E - N + 2 \\ &= 8 - 8 + 2 = 2 \end{aligned}$$

$$\begin{aligned} \text{Predikat (P)} &= P + 1 \\ &= 1 + 1 = 2 \end{aligned}$$

2) Berdasarkan perhitungan *Cyclomatic Complexity* dari *Flowgraph* diatas memiliki *Region* = 2

3) *Independent Path* pada *flowgraph* diatas adalah :

$$\text{Path 1} = 1 - 2 - 3 - 4 - 5 - 3$$

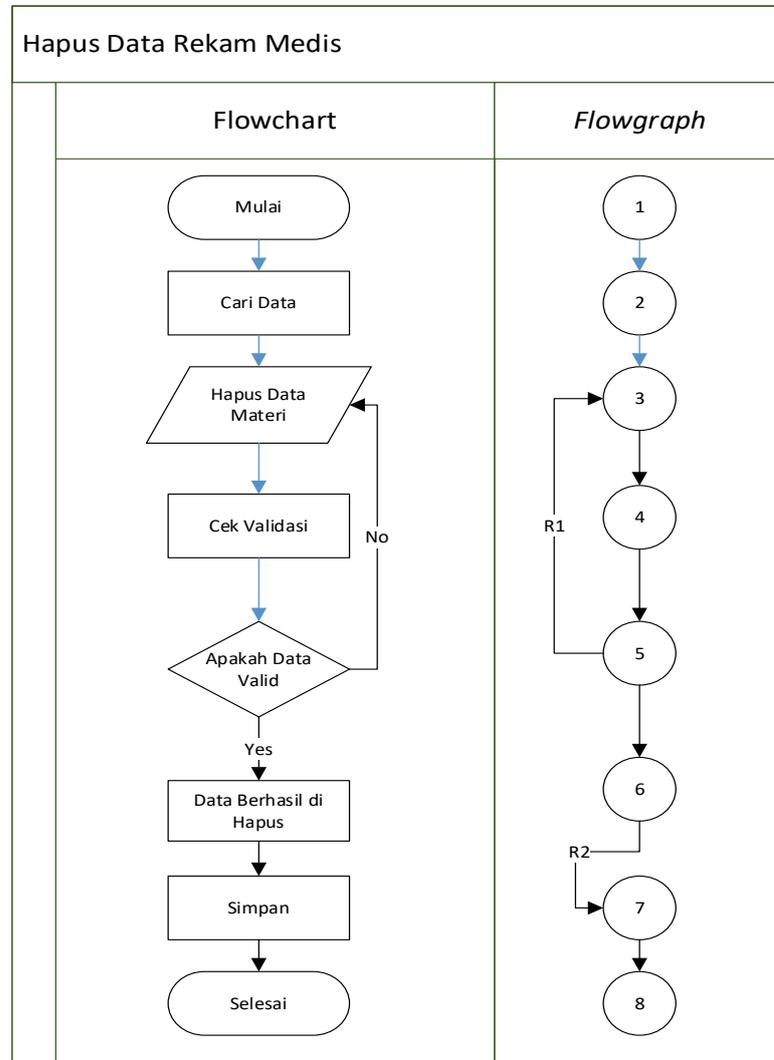
$$\text{Path 2} = 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8$$

4) Grafik *Matriks*

**Tabel 4.15** Grafik *Matriks* pada *Flowgraph Edit Data Rekam Medis*

	1	2	3	4	5	6	7	8	$E - 1 = 0$
1		1							$1 - 1 = 0$
2			1						$1 - 1 = 0$
3				1					$1 - 1 = 0$
4					1				$1 - 1 = 0$
5			1			1			$2 - 1 = 1$
6							1		$1 - 1 = 0$
7								1	$1 - 1 = 0$
8									0
Sum (E) + 1									$1 + 1 = 2$

## d) Hapus Data Rekam Medis



**Gambar 4.34** Pengujian Hapus Data Rekam Medis

Dari gambar *flowgraph* Hapus Data diatas dapat dilakukan proses perhitungan sebagai berikut:

- 1) Menghitung *Cyclomatic Complexity*  $V(G)$  dari *Edge* dan *Node*:

Dengan Rumus :  $V(G) = E - N + 2$

Dengan  $E$  (*edge*) = 8

$N$  (*node*) = 8

$$\text{Predikat Node (P)} = 1$$

$$\begin{aligned} \text{Penyelesaian : } V(G) &= E - N + 2 \\ &= 8 - 8 + 2 = 2 \end{aligned}$$

$$\begin{aligned} \text{Predikat (P)} &= P + 1 \\ &= 1 + 1 = 2 \end{aligned}$$

2) Berdasarkan perhitungan *Cyclomatic Complexity* dari *Flowgraph* diatas memiliki *Region* = 2

3) *Independent Path* pada *flowgraph* diatas adalah :

$$\text{Path 1} = 1 - 2 - 3 - 4 - 5 - 3$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8$$

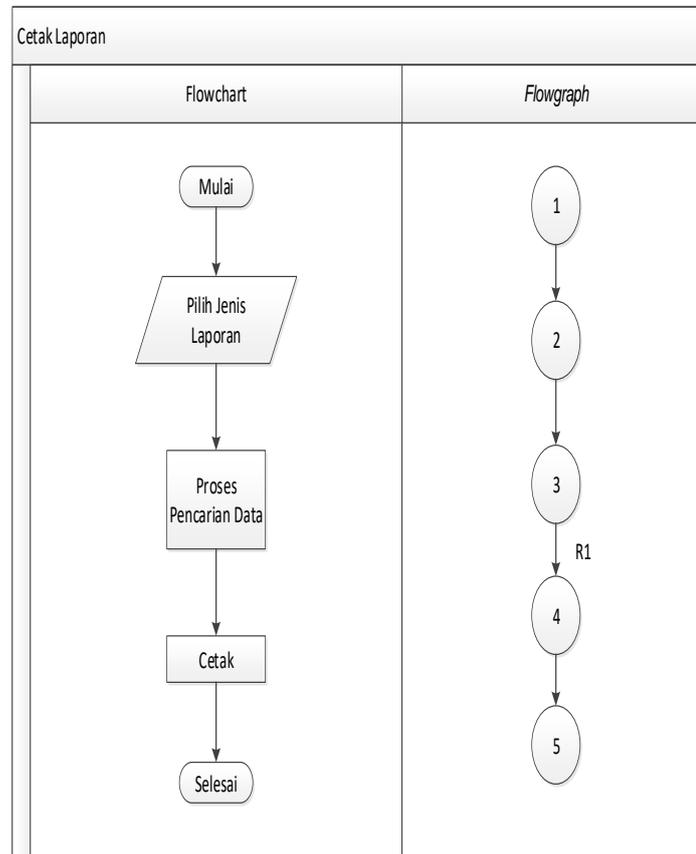
4) Grafik *Matriks*

**Tabel 4.16** Grafik *Matriks* pada *Flowgraph* Hapus Data Rekam Medis

	1	2	3	4	5	6	7	8	$E - 1 = 0$
1		1							$1 - 1 = 0$
2			1						$1 - 1 = 0$
3				1					$1 - 1 = 0$
4					1				$1 - 1 = 0$
5			1			1			$2 - 1 = 1$
6							1		$1 - 1 = 0$
7								1	$1 - 1 = 0$
8									0
Sum (E) + 1									$1 + 1 = 2$

## e) Cetak Laporan

Tabel 4.21 Pengujian Cetak Laporan Rekam Medis

**Gambar 4.35** Pengujian Cetak Laporan Rekam Medis

Dari gambar *flowgraph* Cetak Data Laporan Rekam Medis di atas dapat dilakukan proses perhitungan sebagai berikut:

- 1) Menghitung *Cyclomatic Complexity*  $V(G)$  dari *Edge* dan *Node*:

Dengan Rumus :  $V(G) = E - N + 2$

Dengan  $E$  (*edge*) = 4

$N$  (*node*) = 5

Predikat *Node* ( $P$ ) = 0

$$\begin{aligned}
 \text{Penyelesaian : } V(G) &= E - N + 2 \\
 &= 4 - 5 + 2 \\
 &= 1
 \end{aligned}$$

$$\begin{aligned}
 \text{Predikat (P)} &= P + 1 \\
 &= 0 + 1 \\
 &= 1
 \end{aligned}$$

2) Berdasarkan perhitungan *Cyclomatic Complexity* dari *Flowgraph* diatas memiliki *Region* = 1

3) *Independent Path* pada *flowgraph* diatas adalah :

$$\text{Path 1} = 1 - 2 - 3 - 4 - 5$$

4) Grafik *Matriks*

**Tabel 4.17** Grafik *Matriks* pada *Flowgraph* Laporan Rekam Medis

	1	2	3	4	5	$E - 1 = 0$
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3				1		$1 - 1 = 0$
4					1	$1 - 1 = 0$
5						0
Sum ( E ) + 1						$0 + 1 = 1$

## 2. Pengujian *Black Box*

Pada *fase* ini dilakukan pengujian terhadap sistem dengan menggunakan kotak hitam (*black-box testing*). Pengujian sistem ini bertujuan untuk mengetahui apakah fungsi-fungsi, masukan dan keluaran dari sistem telah sesuai dengan spesifikasi yang dibutuhkan. Cara pengujian *black-box testing* dilakukan dengan menjalankan sistem dan melakukan *input* data serta melihat inputnya apakah sudah sesuai yang diharapkan. Berdasarkan rencana hasil pengujian, maka dapat dilakukan pengujian sebagai berikut.

### a) Skenario *Admin*

**Tabel 4.18** Hasil Pengujian *Black Box Form Admin*

<b>Kasus dan Hasil Uji</b>		
<b>Aksi/ data masukan</b>	<b>Yang diharapkan</b>	<b>Kesimpulan</b>
<i>Login</i>	Berpindah ke Menu Utama	Berhasil ✓
Halaman Utama	<i>Admin</i> dapat mengakses halaman <i>home</i> dan semua fitur yang ada di halaman utama berfungsi dengan baik	Berhasil ✓
Halaman <i>Form</i> Data Pasien	<i>Admin</i> dapat mengakses halaman <i>form</i> data pasien dan semua fitur berfungsi dengan baik	Berhasil ✓
Halaman <i>Form</i> Data Pengguna	<i>Admin</i> dapat mengakses halaman data pengguna dan semua fitur berfungsi dengan baik	Berhasil ✓
Halaman <i>Form</i> Rekam Medis	<i>Admin</i> dapat mengakses halaman rekam medis dan semua fitur berfungsi dengan baik	Berhasil ✓
Halaman <i>Form</i> Laporan	<i>Admin</i> dapat mengakses halaman <i>form</i> laporan dan semua fitur berfungsi dengan baik	Berhasil ✓

<i>Logout</i>	<i>Admin</i> keluar dari halaman sistem	Berhasil ✓
---------------	---	------------

## b) Skenario Petugas

**Tabel 4.19** Hasil Pengujian *Black Box Form* Petugas

<b>Kasus dan Hasil Uji</b>		
<b>Aksi/ data masukan</b>	<b>Yang diharapkan</b>	<b>Kesimpulan</b>
<i>Activity Splash Screen</i>	<i>User</i> dapat mengakses halaman utama dan menampilkan <i>form login</i>	Berhasil ✓
<i>Login</i>	Berpindah ke Menu Utama	Berhasil ✓
Halaman Utama	<i>User</i> dapat mengakses halaman utama dan menampilkan gambaran informasi pasien	Berhasil ✓
Halaman <i>Form</i> Pendaftaran Pasien	<i>User</i> dapat mengakses <i>form</i> pendaftaran pasien tanpa ada <i>error</i> dan menampilkan data pasien	Berhasil ✓
Tambah Data Pasien	<i>User</i> dapat melakukan penambahan data pasien dengan menekan <i>button</i> “Tambah Data”	Berhasil ✓
<i>Edit</i> Data Pasien	<i>User</i> dapat melakukan perubahan data pasien dengan menekan <i>button</i> “ <i>Edit</i> ”	Berhasil ✓
Hapus Data Pasien	<i>User</i> dapat melakukan menghapus data pasien dengan menekan <i>button</i> “Hapus”	Berhasil ✓
Halaman <i>Form</i> Pengaturan Akun	<i>User</i> dapat mengakses <i>form</i> pengaturan akun tanpa ada <i>error</i>	Berhasil ✓
<i>Logout</i>	<i>User</i> keluar dari halaman sistem	Berhasil ✓

## c) Skenario Kepala Puskesmas

**Tabel 4.20** Hasil Pengujian *Black Box Form* Kepala Puskesmas

<b>Kasus dan Hasil Uji</b>		
<b>Aksi/ data masukan</b>	<b>Yang diharapkan</b>	<b>Kesimpulan</b>
<i>Activity Splash Screen</i>	<i>User</i> dapat mengakses halaman utama dan menampilkan <i>form login</i>	Berhasil ✓
<i>Login</i>	Berpindah ke Menu Utama	Berhasil ✓
Halaman Utama	<i>User</i> dapat mengakses halaman utama dan menampilkan informasi pasien	Berhasil ✓
Halaman <i>Form</i> Laporan	<i>User</i> dapat mengakses <i>form</i> pendaftaran pasien tanpa ada <i>error</i> dan menampilkan data rekam medis	Berhasil ✓
Halaman <i>Form</i> Pengaturan Akun	<i>User</i> dapat mengakses <i>form</i> pengaturan akun tanpa ada <i>error</i>	Berhasil ✓
<i>Logout</i>	<i>User</i> keluar dari halaman sistem	Berhasil ✓

## d) Skenario Dokter

**Tabel 4.21** Hasil Pengujian *Black Box Form* Dokter

<b>Kasus dan Hasil Uji</b>		
<b>Aksi/ data masukan</b>	<b>Yang diharapkan</b>	<b>Kesimpulan</b>
<i>Activity Splash Screen</i>	<i>User</i> dapat mengakses halaman utama dan menampilkan <i>form login</i>	Berhasil ✓
<i>Login</i>	Berpindah ke Menu Utama	Berhasil ✓
Halaman Utama	<i>User</i> dapat mengakses halaman utama dan menampilkan gambaran informasi pasien	Berhasil ✓

Halaman <i>Form</i> Rekam Medis	<i>User</i> dapat mengakses <i>form</i> rekam medis tanpa ada <i>error</i> dan menampilkan data rekam medis	Berhasil ✓
Halaman <i>Form</i> Pengaturan Akun	<i>User</i> dapat mengakses <i>form</i> pengaturan akun tanpa ada <i>error</i>	Berhasil ✓
<i>Logout</i>	<i>User</i> keluar dari halaman sistem	Berhasil ✓

## **BAB V**

### **PENUTUP**

#### **A. Kesimpulan**

Dihasilkannya sebuah Aplikasi Mobile Rekam Medis Pasien Berdasarkan NIK Pada Pelayanan Kesehatan Menggunakan *QR Code* dengan menggunakan bahasa pemrograman PHP untuk *Database*-nya menggunakan *MySQL*. Pengelolaan rekam medis pasien pada Puskesmas Barukku dengan cara pembuatan kartu pasien dengan *QR Code* akan lebih efektif dan efisien. Aplikasi yang dibuat sudah bebas dari kesalahan fungsi logika karena diuji dengan menggunakan dua teknik pengujian yaitu pengujian *Black Box* dan pengujian *White Box*.

1. Penelitian ini berhasil membangun Aplikasi Mobile Rekam Medis Pasien Berdasarkan NIK Pada Pelayanan Kesehatan Menggunakan *QR Code*.
2. Aplikasi rekam medis dapat meningkatkan efisiensi dan kualitas pelayanan kesehatan serta aplikasi ini memudahkan identifikasi pasien, mengurangi risiko kesalahan administratif, dan mempercepat akses data medis dan data NIK sudah terintegrasi dengan sistem *online*.

Pembuatan kartu pasien dengan *QR Code*, dapat mempermudah dokter dan pasien ketika ingin melakukan pemeriksaan. Dokter dapat melakukan scan *QR Code* kartu pasien untuk dapat melihat data rekam medis pasien dan dapat langsung menambahkan hasil rekam medis terbaru pasien.

## **B. Saran**

Dalam suatu Aplikasi Rekam Medis Pasien Pada Pelayanan Kesehatan Menggunakan *QR Code*. Berdasarkan hasil kesimpulan yang dijabarkan pada kesimpulan diatas, aplikasi ini masih jauh dari kesempurnaan. Apabila ada yang berniat untuk mengembangkan sistem ini, maka disarankan hal sebagai berikut.

1. Hendaknya sistem yang penulis rancang dapat dikembangkan sehingga dalam pengolahan datanya dapat lebih dijelaskan secara mendetail, agar pengolahan data dan pembuatan laporan dapat dilakukan lebih efektif dan efisien.
2. Sistem yang telah dikembangkan penulis sebaiknya dipelihara dengan sebaik-baiknya dan diperbaharui sesuai dengan kebutuhan yang ada serta melakukan *backup* data pada jangka waktu tertentu untuk menghindari kemungkinan data hilang atau rusak.

## DAFTAR PUSTAKA

- Adi Nugroho, 2005, *Konsep Pengembangan Sistem Basis Data Informatika*, Bandung.
- Adi Nugroho. 2010. *Rekayasa Perangkat Lunak Berbasis Objek dengan Metode USDP*. Andi. Yogyakarta
- Aditya, Alan Nur. 2011. *Jago PHP & MySQL Dalam Hitungan Menit*. Jakarta: Dunia Komputer.
- Arifianto, Teguh. (2011). *Membuat Interface Aplikasi Android Lebih Keren dengan LWUIT*. Yogyakarta: Andi Publisher.
- Arin Sri Alfiani (2017). “*Aplikasi Presensi Pegawai Menggunakan QR Code Berbasis Android Di Puskesmas Cisaga*”.
- Deka M Wildan, 2014. “*HTTP*”. Available at: <https://www.tensorflow.org/> [Accessed 29 Januari2022].
- Departemen Kesehatan RI. 1997. *Sistem Kearsipan Rekam Medis*.
- Fauzi Megantara, (2019). “*Implementasi Big Data Untuk Pencarian Pattern Data Gudang Pada PT. Bank Mandiri (Persero) TBK*”.
- Hartono, Jogiyanto. 1999. *Pengenalan Komputer*. Andi:Yogyakarta.
- Hussain Muhammad Irsyad, (2020). “*Rancang Bangun Sistem Pelayanan Pasien Rawat Jalan di Puskesmas Pallangga Kabupaten Gowa Berbasis Web*”.
- Jogiyanto Hartono, 2005. *Analisis dan Desain Sistem Informasi. Pendekatan terstruktur teori dan praktis aplikasi bisnis*. Andi. Yogyakarta.
- Kadir, A. 2008. *Dasar Pemograman dan Implementasi Database Relasional*. Yogyakarta: ANDI.
- KBBI, 2015. *Kamus Besar Bahasa Indonesia (KBBI)*. [Online, Diakses pada tanggal 7Februari 2022].
- Madcoms. 2012. *Manajemen Sistem Jaringan Komputer.Ed I*. Yogyakarta: Andi, Madiun: Madcoms.
- MenkesRI. 2008. *Peraturan Menteri Kesehatan (Permenkes) RI Nomor 269/MENKES/PER/III/2008 tentang Rekam Medis*.
- Munawar. (2005), *Pemodelan Visual dengan UML*.Grahallmu : Yogyakarta, 17-100.

- Nazruddin, Safaat H. 2012. (EdisiRevisi) *Pemograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Informatika, Bandung.
- Oktavian, Diar Puji. 2010. *Menjadi Programmer Jempolan Menggunakan PHP*. Yogyakarta: Penerbit MediaKom.
- Sholiq. 2006. *Pemodelan Sistem Informasi Berorientasi Objek dengan UML*. Graha Ilmu. Yogyakarta.
- Simarmata, Janner. 2009. *Rekayasa Perangkat Lunak*, Medan : Andi.
- Sugiyono. 2001. *Metode Penelitian kuantitatif kualitatif dan R&D*. Bandung: Alfabeta.
- Undang-Undang Nomor 8 Tahun 1997 tentang Dokumen Perusahaan. Tersedia dalam: <<https://www.jogloabang.com/pustaka/>>(diakses pada tanggal 28 Januari 2022).
- Universitas Muhammadiyah Parepare. (2021). *Panduan Akademik 2020-2021*, Parepare : UMPAR Press.