**APLIKASI PELAYANAN KESEHATAN DAN KONSULTASI MASYARAKAT BERBASIS *MOBILE* MENGGUNAKAN**

**REACT NATIVE**

**TUGAS AKHIR**Sebagai salah satu syarat untuk menyelesaikan   
Program Diploma III Teknik Informatika  
Politeknik Negeri Indramayu  
  
  
  
  
**Oleh:  
MOHAMMAD ILHAM TEGUHRIYADI  
NIM. 2003077  
  
  
  
PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
POLITEKNIK NEGERI INDRAMAYU   
AGUSTUS 2023**

# **HALAMAN PENGESAHAN**

Tugas Akhir ini diajukan oleh:  
Nama : Mohammad Teguh Riyadi  
NIM : 2003077  
ProgramStudi : Diploma III Teknik Informatika  
Judul : Aplikasi Pelayanan Kesehatan dan Konsultasi Masyarakat berbasis   
 *Mobile* menggunakan React Native  
Pembimbing : Adi Suheryadi, S.ST., M.Kom.  
 NIP. - ………………….

Telah berhasil dipertahankan dihadapan dewan penguji pada tanggal 15 Juli 2023 dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Ahli Madya Program Studi Diploma III Teknik Informatika, Jurusan Teknik Informatika, Politeknik Negeri Indramayu.

DEWAN PENGUJI

Ketua Penguj : Hamdan, M.Kom.  
 NIP. 2003018 ………………….

|  |
| --- |
| Indramayu, 26 September 2023 |
| Ketua Jurusan Teknik Informatika |
|  |
|  |
| Dewa Buana, M.T. |
| NIP. 2003018 |

Anggota : Hamdan, M.T.  
Penguji I NIP. 2003019 ………………….

# **HALAMAN PERNYATAAN**

Dengan ini saya menyatakan dengan sesungguhnya bahwa Tugas Akhir ini adalah asli hasil karya saya sendiri serta **Tugas Akhir/Skripsi** ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar **Ahli Madya/Sarjana Terapan** di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dirujuk dalam naskah ini dan disebutkan dalam daftar Pustaka.

|  |
| --- |
| Indramayu, 3 Agustus 2023 |
| Yang menyatakan, |
|  |
|  |
| Mohammad Ilham Teguhriyadi. |
| NIM. 2003018 |

# **ABSTRAK**

Pemerintah Indonesia dalam strategi *e-health* mengupayakan untuk meningkatkan pelayanan kesehatan dengan melibatkan teknologi informasi dan komunikasi di dalemnya. Kesehatan merupakan hal utama dalam pemenuhan kebutuhan manusia. Masalah kesehatan adalah faktor untuk pembangunan suatu negara, namun terdapat tantangan dalam mewujudkan suatu pelayanan kesehatan yang menyeluruh. Salah satu tantangan tersebut adalah akses jarak dan waktu. Atas dasar tersebut, diperlukan solusi untuk mengatasinya, yaitu melalui aplikasi pelayanan kesehatan dan konsultasi masyarakat berbasis *mobile*. Masyarakat dapat menerima akses layanan kesehatan seperti konsultasi kesehatan dengan dokter dan perawat, dan jasa perawatan oleh perawat terdekat. Pengembangan aplikasi ini menggunakan metode pengembangan perangkat lunak agile dengan kerangka kerja scrum. Agile adalah metodologi pengembangan perangkat lunak yang dilakukan berulang-ulang. Dalam menentukan lokasi pelayanan kesehatan terdekat menggunakan teorema haversine sebagai perhitungannya. Hasil dari pengembangan aplikasi pelayanan kesehatan dan konsultasi masyarakat ini diharapkan kualitas hidup masyarakat akan meningkat karena mereka dapat mengakses pelayanan kesehatan kesehatan tanpa terkendala jarak dan waktu.

**Kata Kunci:** Pelayanan Kesehatan, Agile Scrum, Haversine

# ***ABSTRACT***

*The Indonesian government, through its e-health strategy, endeavors to enhance healthcare services by incorporating information and communication technology. Health is of paramount importance in fulfilling human needs. Health issues are crucial factors for a nation's development, but there are challenges in achieving comprehensive healthcare services. One of these challenges is accessibility in terms of distance and time. To address this, a solution is required, namely through a mobile-based healthcare and consultation application. The public can access healthcare services such as consultations with doctors and nurses, as well as home care services provided by nearby caregivers through this application. The development of this application employs the agile software development method using the Scrum framework. Agile is a software development methodology that is iterative in nature. To determine the nearest healthcare location, the Haversine theorem is utilized for its calculations. The expected outcome of this community healthcare application's development is an improvement in the quality of life for the public, as they can access healthcare services without being hindered by distance and time constraints.*

***Keywords****: Healthcare Services, Agile Scrum, Haversine.*

# **MOTTO**

“Kita pernah ada sebelum pada akhirnya menjadi masing-masing :v”

# **KATA PENGANTAR**

Dengan mengucapkan puji dan syukur atas kehadirat Allah SWT, karena berkat rahmat dan hidayah-Nya penulis dapat menyelesaikan Tugas Akhir dengan tepat waktu. Tugas akhir merupakan salah satu syarat dalam Jurusan Teknik Informatika untuk menyelesaikan Diploma III Program Studi Teknik Informatika pada Politeknik Negeri Indramayu.

Dalam melakukan pengerjaan Tugas Akhir, penulis tidak luput juga dari berbagai macam kendala. Namun, berkat karunia-Nya dan dukungan moril dari segala pihak yang tiada hentinya. Akhirnya, penulis dapat menyelesaikan Tugas Akhir dengan tepat waktu dan masih dalam keadaan sehat.

Pengerjaan Tugas Akhir ini tidak akan pernah selesai jika tanpa dukungan baik moril maupun materil dari berbagai pihak yang sudah terlibat dalam proses pengerjaannya. Maka dari itu, penulis mengucapkan terima kasih kepada;

1. Bapak Eka Ismantohadi, M.Kom. selaku Ketua Jurusan Teknik Informatika Politeknik Negeri Indramayu.
2. Bapak Adi Suheryadi, S.ST., M.Kom selaku Wali Dosen kelas D-3 TI 3 C.
3. Bapak Adi Suheryadi, S.ST., M.Kom. selaku Dosen Pembimbing yang telah *member*i dukungan dan arahan.
4. Bapak Fachrul Pralienka Bani Muhamad., S.ST., M.Kom selaku koordinator Program Studi D3 Teknik Informatika.
5. Kedua orang tua dan keluarga yang telah memberi dukungan moril dan materil yang tidak pernah henti.
6. Muhamad Rafli Septian, selaku rekan seperjuangan dalam mengerjakan Tugas Akhir.
7. Tas hijau

Penulis menyadari bahwa Tugas Akhir yang telah diselesaikan jauh dari kata sempurna. Karena keterbatasan waktu dan pengetahuan yang penulis punya. Oleh karena itu, penulis terbuka untuk menerima saran dan kritik. Penulis berharap bahwa Tugas Akhir yang berjudul “Aplikasi Pelayanan Kesehatan dan Konsultasi Masyarakat berbasis *Mobile* menggunakan React Native” dapat menambah wawasan serta ilmu pengetahuan bagi pembaca.

|  |
| --- |
| Indramayu, … Agustus 2023 |
|  |
| Penulis |

# **DAFTAR ISI**

[HALAMAN PENGESAHAN 2](#_Toc141897124)

[HALAMAN PERNYATAAN 3](#_Toc141897125)

[ABSTRAK 4](#_Toc141897126)

[*ABSTRACT* 5](#_Toc141897127)

[KATA PENGANTAR 6](#_Toc141897128)

[DAFTAR ISI 8](#_Toc141897129)

[DAFTAR LAMPIRAN 13](#_Toc141897130)

[BAB I PENDAHULUAN 14](#_Toc141897131)

[1.1 Latar Belakang 14](#_Toc141897132)

[1.2 Rumusan Masalah 15](#_Toc141897133)

[1.3 Batasan Masalah 16](#_Toc141897134)

[1.4 Tujuan 16](#_Toc141897135)

[1.5 Manfaat 16](#_Toc141897136)

[1.6 Sistematika Penulisan 17](#_Toc141897137)

[BAB II LANDASAN TEORI 19](#_Toc141897138)

[2.1. Dokter 19](#_Toc141897139)

[2.2 Dokter Spesialis 20](#_Toc141897140)

[2.3 Dokter Praktik Umum 20](#_Toc141897141)

[2.4 Keperawatan 20](#_Toc141897142)

[2.5 Perawat 20](#_Toc141897143)

[2.5.1 Peran Perawat 21](#_Toc141897144)

[2.5.2 Peran sebagai Pelaksana Kesehatan 21](#_Toc141897145)

[2.5.3 Peran sebagai Pendidik Administrasi 21](#_Toc141897146)

[2.5.4 Peran sebagai Pendidik 21](#_Toc141897147)

[2.5.5 Peran sebagai konseling 22](#_Toc141897148)

[2.5.6 Peran sebagai peneliti 22](#_Toc141897149)

[2.6 Konsultasi Kesehatan Daring 22](#_Toc141897150)

[2.7 Latitude 22](#_Toc141897151)

[2.8 Longitude 22](#_Toc141897152)

[2.9 Teorema Haversine 23](#_Toc141897153)

[2.10 API 23](#_Toc141897154)

[2.11 React Native 24](#_Toc141897155)

[2.12 Agile Development 24](#_Toc141897156)

[2.13 UML (Unified Modeling Language) 24](#_Toc141897157)

[2.13.1 Pengenalan UML 24](#_Toc141897158)

[2.13.2 *Use Case* Diagram 25](#_Toc141897159)

[2.13.2 *Activity* Diagram 26](#_Toc141897160)

[2.14 *Entity Relationship* Diagram(ERD) 27](#_Toc141897161)

[2.15 *Global Positioning System* (GPS) 28](#_Toc141897162)

[2.16 OpenStreetMap 28](#_Toc141897163)

[2.17 *Black Box Testing* 29](#_Toc141897164)

[BAB III METODE PELAKSANAAN 30](#_Toc141897165)

[3.1 Metode Pengembangan Perangkat Lunak 30](#_Toc141897166)

[3.2 Analisis Kebutuhan Sistem 31](#_Toc141897167)

[3.2.1 Kebutuhan Fungsional 31](#_Toc141897168)

[3.2.2 Kebutuhan Non Fungsional 33](#_Toc141897169)

[3.2.3 Kebutuhan *Hardware* 33](#_Toc141897170)

[3.2.4 Kebutuhan *Software* 34](#_Toc141897171)

[3.3 Perancangan Sistem 34](#_Toc141897172)

[3.3.1 Use Case Diagram 34](#_Toc141897173)

[3.3.2 Activity Diagram 38](#_Toc141897174)

[3.4 Flowchart 46](#_Toc141897175)

[3.4.1 Flowchart Registrasi Akun 46](#_Toc141897176)

[3.4.2 Flowchart Membaca Artikel 47](#_Toc141897177)

[3.4.3 Flowchart Melihat Rumah Sakit Terdekat 47](#_Toc141897178)

[3.4.4 Flowchart Memesan Produk Apotek 48](#_Toc141897179)

[3.4.5 Flowchart Membuat Janji Kunjungan Rumah Sakit 48](#_Toc141897180)

[3.4.6 Flowchart Melakukan Konsultasi Kesehatan 49](#_Toc141897181)

[3.5 Perancangan *Entity Relationship* Diagram (ERD) 50](#_Toc141897182)

[3.6 Perancangan Basis Data 52](#_Toc141897183)

[3.6.1 Tabel Master Data 52](#_Toc141897184)

[3.6.2 Tabel Proses Bisnis 57](#_Toc141897185)

[3.7 Perancangan Desain Tampilan Antarmuka 59](#_Toc141897186)

[3.7.1 Rancangan Antarmuka 59](#_Toc141897187)

[BAB IV HASIL DAN PEMBAHASAN 65](#_Toc141897188)

[4.1 Hasil 65](#_Toc141897189)

[4.2 Pembahasan 65](#_Toc141897190)

[4.2.1 Implementasi Basis Data 66](#_Toc141897191)

[4.2.2 *Sprint Review Product Backlog* 74](#_Toc141897192)

[4.2.3 *Daily Activity* 76](#_Toc141897193)

[4.2.4 *Testing* *Endpoint* API 77](#_Toc141897194)

[4.2.5 *Testing* Modul *Product Backlog* 77](#_Toc141897195)

[BAB V PENUTUP 78](#_Toc141897196)

[5.1 Kesimpulan 78](#_Toc141897197)

[5.2 Saran 78](#_Toc141897198)

[DAFTAR PUSTAKA 79](#_Toc141897199)

**DAFTAR GAMBAR**

[Gambar 3.1 Metode Agile Scrum 30](#_Toc141562895)

[Gambar 3.2 Use Case Diagram 34](#_Toc141562896)

[Gambar 3.3 Flowchart Register 45](#_Toc141562897)

[Gambar 3.4 Flowchart Membaca Artikel 46](#_Toc141562898)

[Gambar 3.5 Flowchart Melihat Rumah Sakit Terdekat 46](#_Toc141562899)

[Gambar 3.6 Flowchart Memesan Produk Apotek 47](#_Toc141562900)

[Gambar 3.7 Flowchart Membuat Janji Kunjungan Rumah Sakit 48](#_Toc141562901)

[Gambar 3.8 Rancangan ERD 48](#_Toc141562902)

[Gambar 3.9 Rancangan Antarmuka Halaman Intro dan Register 58](#_Toc141562903)

[Gambar 3.10 Rancangan Antarmuka Halaman Home dan Detail Artikel 59](#_Toc141562904)

[Gambar 3.11 Rancangan Antramuka Chat Dokter dan Perawat 60](#_Toc141562905)

[Gambar 3.12 Rancangan Antarmuka Menu Toko Kesehatan 61](#_Toc141562906)

[Gambar 3.13 Rancangan Antarmuka Pembayaran Produk dan Profil 62](#_Toc141562907)

**DAFTAR TABEL**

[Tabel 3.1Tabel Kebutuhan Hardware 31](#_Toc141611793)

[Tabel 3.2 Tabel Kebutuhan Software 32](#_Toc141611794)

[Tabel 3.3 Tabel Penjelasan Use Case Diagram 33](#_Toc141611795)

[Tabel 3.4 Penjelasan ERD 48](#_Toc141611796)

[Tabel 3.5 Tabel Kategori Artikel 49](#_Toc141611797)

[Tabel 3.6 Tabel Kategori Produk 50](#_Toc141611798)

[Tabel 3.7 Tabel Spesialis 50](#_Toc141611799)

[Tabel 3.8 Tabel Spesialis Rumah Sakit 50](#_Toc141611800)

[Tabel 3.9 Tabel Fasilitas Rumah Sakit 51](#_Toc141611801)

[Tabel 3.10 Tabel Grouping Artikel 51](#_Toc141611802)

[Tabel 3.11 Tabel Produk Kategori 51](#_Toc141611803)

[Tabel 3.12 Tabel Profil Apotek 52](#_Toc141611804)

[Tabel 3.13 Tabel Rumah Sakit 52](#_Toc141611805)

[Tabel 3.14 Tabel Keahlian 53](#_Toc141611806)

[Tabel 3.15 Tabel Join Keahlian 53](#_Toc141611807)

[Tabel 3.16 Tabel Detail Praktik 54](#_Toc141611808)

[Tabel 3.17 Tabel Jadwal Praktik 54](#_Toc141611809)

[Tabel 3.18 Tabel Transaksi 54](#_Toc141611810)

[Tabel 3.19 Tabel Artikel 55](#_Toc141611811)

[Tabel 3.20 Tabel Jadwal Antrian 55](#_Toc141611812)

[Tabel 3.21 Tabel Produk 56](#_Toc141611813)

[Tabel 3.22 Tabel Users 56](#_Toc141611814)

# **DAFTAR LAMPIRAN**

Lampiran 1 Kode Program

Lampiran 2 *Product Backlog*

Lampiran 3 *Sprint Planning*

Lampiran 4 *Daily Activity*

Lampiran 5 *Testing Product Backlog*

Lampiran 6 *Testing Endpoint*

Lampiran 7 Biodata Penulis

# 

# **BAB I PENDAHULUAN**

## Latar Belakang

Pemanfaatan teknologi semakin berkembang dengan pesat, tidak hanya untuk memenuhi kebutuhan informasi, namun juga untuk meningkatkan efektivitas kehidupan manusia. Teknologi digunakan untuk meningkatkan pelayanan aktivitas manusia. Salah satu penerapannya pada bidang kesehatan. Menurut Peraturan Menteri Kesehatan Republik Indonesia nomor 46 tahun 2017 tentang strategi e-health secara umum membahas adanya kapasitas teknologi dan komunikasi di bidang kesehatan untuk meningkatkan kualitas, aksebilitas, dan kesinambungan pelayanan kesehatan.

Menurut WorldHealth Organization (WHO) menyatakan bahwa kesehatan adalah keadaan sejahtera badan, jiwa dan sosial yang memungkinkan setiap orang hidup produktif secara sosial dan ekonomi. Kesehatan seseorang tidak hanya diukur dari segi aspek fisik, mental, dan sosial, tapi juga diukur dari aspek produktivitasnya dalam arti mempunyai pekerjaan atau menghasilkan secara ekonomi.

Masalah kesehatan merupakan satu dari berbagai faktor kualitas hidup yang mencerminkan pemenuhan kebutuhan dasar manusia. Bidang kesehatan sangat erat kaitannya dengan pembangunan suatu negara, khususnya pembangunan terhadap sumber daya manusia. Pembangunan kesehatan diarahkan untuk meningkatkan mutu pelayanan kesehatan yang dilaksanakan melalui peningkatan mutu pelayanan rumah sakit, puskesmas, dan lembaga pelayanan kesehatan lainnya.

Tantangan dalam sektor kesehatan adalah kemudahan masyarakat untuk mencapai pusat pelayanan kesehatan. Kondisi tersebut menjadi tantangan bagi pemerintah Indonesia untuk memberikan pelayanan yang optimal, dan ini menjadi fokus perhatian. Penyebabnya adalah keterbatasan pelayanan kesehatan yang belum memadai untuk memenuhi kebutuhan semua penduduk yang berada di daerah tertinggal, perbatasan dan kepulauan (DPTK).

Permasalahan kesehatan masyarakat perlu ditangani dengan tepat dan tanggap agar proses pemulihan tidak memakan waktu yang lama. Melalui penanganan yang tepat, kondisi tubuh akan lebih siap melakukan aktivitas seperti semula. Namun masyarakat mengalami kendala yaitu sulitnya akses dalam menerima pelayanan kesehatan.

Dalam pelayanan kesehatan, tidak hanya rumah sakit yang menjadi sarana dalam memberikan layanan kepada masyarakat yang membutuhkan penanganan kesehatan. Terdapat praktik dokter dan perawat secara pribadi, lalu apotek yang dapat memberikan penanganan kesehatan berupa kebutuhan kesehatan, seperti obat dan vitamin.

Untuk mengatasi permasalahan yang terjadi di masyarakat seperti memberikan pelayanan kesehatan yang optimal, maka terdapat sebuah solusi yaitu dengan dibangunnya sebuah aplikasi layanan kesehatan dan konsultasi masyarakat berbasis *mobile*. Layanan penunjang kesehatan tersedia dalam aplikasi ini seperti informasi layanan kesehatan berupa rumah sakit dan apotek terdekat dan artikel kesehatan, pemesanan produk kesehatan, membuat janji kunjungan temu dokter di rumah sakit, dan terdapat fitur lainnya.

Dengan adanya aplikasi ini, masyarakat dapat mengakses pelayanan kesehatan tanpa harus menghadapi kendala sulitnya akses. Aplikasi ini dapat menghasilkan solusi bagi masyarakat yang belum menerima pelayanan kesehatan secara optimal. Diharapkan kualitas hidup masyarakat akan meningkat karena mereka dapat mengakses pelayanan kesehatan kesehatan tanpa terkendala jarak dan waktu.

## Rumusan Masalah

Berdasarkan uraian latar belakang masalah di atas, terdapat beberapa rumusan masalah yang didapatkan, yaitu;

1. Bagaimana mengetahui informasi terkait kebutuhan kesehatan masyarakat serta lokasi pelayanan kesehatan terdekat?
2. Bagaimana mendapatkan produk kesehatan melalui *smartphone*?
3. Bagaimana membuat janji kunjungan temu dokter di rumah sakit melalui *smartphone*?
4. Bagaimana mendapatkan konsultasi kesehatan dengan dokter dan perawat?
5. Bagaimana menerima perawatan kesehatan oleh perawat terdekat?

## Batasan Masalah

Untuk menyelaraskan dan memfokuskan pembahasan, dapat diperoleh beberapa batasan masalah, di antaranya:

1. Aplikasi dirancang berbasis mobile menggunakan react native dan NodeJs versi 18.
2. Konsultasi kesehatan tidak menerapkan pembayaran biaya konsultasi.
3. Tidak terintegrasi sistem informasi rumah sakit, pada saat membuat janji kunjungan.
4. Hanya dapat membuat satu jadwal kunjungan dalam satu hari dengan dokter yang sama.
5. Tidak ada penggantian jadwal (*reschedule*) pada janji kunjungan, namun dapat dibatalkan sebelum jadwal kunjungan berlangsung.
6. Rumah sakit dan apotek terdekat hanya menampilkan jarak dan menggunakan google maps untuk menuju titik lokasi.

## Tujuan

Berdasarkan rumusan dan batasan masalah yang sudah dijelaskan, pembuatan aplikasi ini memiliki berbagai macam tujuan, yaitu;

1. Dapat mengetahui layanan kesehatan terdekat berdasarkan lokasi dan memberikan informasi informasi kesehatan berupa artikel, manfaat dan penggunaan obat, serta dokter dan rumah sakit sesuai spesialisasinya.
2. Dapat melakukan pemesanan obat, vitamin, dan produk kesehatan lainnya melalui *smartphone*.
3. Dapat membuat janji kunjungan temu dokter di rumah sakit tanpa perlu mendaftarnya secara *offline.*
4. Dapat melakukan konsultasi kesehatan melalui *chat* dengan dokter dan perawat.
5. Dapat menerima perawatan kesehatan berupa rawat jalan oleh perawat terdekat.

## Manfaat

Manfaat yang didapatkan dari pembuatan Aplikasi Pelayanan Kesehatan dan Konsultasi Masyarakat Berbasis *Mobile* Menggunakan React Native ini adalah:

1. Masyarakat dapat mengakses pelayanan kesehatan tanpa perlu membutuhkan banyak waktu.
2. Masyarakat dapat mengetahui informasi layanan kesehatan berupa tempat pelayanan kesehatan terdekat, dokter dan perawat, dan artikel kesehatan.
3. Masyarakat dapat membeli produk kesehatan yang dibutuhkan melalui aplikasi pelayanan kesehatan dan konsultasi masyarakat ini.
4. Masyarakat dapat membuat janji kunjungan ke rumah sakit melalui aplikasi ini. Serta masyarakat dapat mengetahui jadwal praktik dokter yang ada pada rumah sakit.
5. Masyarakat dapat melakukan konsultasi kesehatan dengan dokter dan perawat, serta menerima jasa perawatan kesehatan oleh perawat terdekat.

## Sistematika Penulisan

Dalam sistematika penulisan laporan tugas akhir ini dibagi menjadi lima bab, dimana setiap bagian memiliki pembahasan yang berbeda-beda tetapi saling terkait antara satu dengan lainnya. Untuk memudahkan penulisan laporan tugas akhir ini, maka akan diurutkan dan dijabarkan setiap bagian secara sistematis. Adapun sistematika penulisan laporan ini sebagai berikut:

**BAB I PENDAHULUAN**

Pada bab pendahuluan dipaparkan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan masalah, serta manfaat penelitian aplikasi pelayanan kesehatan dan konsultasi masyarakat berbasis *mobile* menggunakan react native.

**BAB II LANDASAN TEORI**

Landasan teori merupakan seperangkat definsi maupun konsep yang telah disusun berdasarkan sumber referensi yang valid tentang teori-teori terkait penyusunan laporan tugas akhir serta beberapa literatur review yang berhubungan dengan penelitian. Sumber yang dijadikan referensi berasal dari buku yang berikaitan dengan penelitian dari sumber yang dapat dipercaya. Teori-teori tersebut mengenai dokter, perawat, dan sebagai lainnya.

**BAB III METODE PELAKSANAAN**

Metode Pelaksanaan menjelaskan tentang tahapan serta metode penelitian yang digunakan dalam membangun aplikasi pelayanan kesehatan dan konsultasi masyarakat berbasis *mobile*  menggunakan react native. Pada bagian ini terdapat perancangan aplikasi berupa flowchart, *Unified Modelling Language* (UML), *Entitiy Relationship Diagram* (ERD), rancangan basis data, dan rancangan antarmuka yang dibuat untuk menjelaskan tahapan alur proses serta gambaran nyata dari sistem yang akan dibuat.

**BAB IV HASIL DAN PEMBAHASAN**

Pada bab ini terdapat hasil dari tahapan penelitian, mulai dari tahap analisis, implementasi desain, hasil pengujian aplikasi serta implementasinya yang dipaparkan secara teoritik. Bab ini juga memastikan bahwa aplikasi yang dibuat telah sesuai atas tahapan-tahapan sebelumnya.

**BAB V PENUTUP**

Bab ini berisi kesimpulan dan saran yang berkaitan dengan analisa dan optimalisasi aplikasi berdasarkan yang telah diuraikan pada bab-bab sebelumnya. Dalam pembuatan kesimpulan, hal – hal yang perlu diperkuat berhubungan dengan pembuktian tidak diperkuat berhubungan dengan pembuktian tidak terdapat dari hasil penelitian. Saran yang penulis cantumkan merupakan pendapat untuk melaksanakan sesuatu yang belum diselesaikan secara maksimal.

# 

# **BAB II LANDASAN TEORI**

## Dokter

Dokter adalah seorang tenaga kesehatan yang menjadi tempat kontak pertama pasien dengan dokternya untuk menyelesaikan semua masalah kesehatan tanpa memandang jenis penyakit, organologi, golongan usia, dan jenis kelamin, sedini dan sedapat mungkin, secara menyeluruh, paripurna, bersinambungan, dan dalam koordinasi serta kolaborasi dengan profesional kesehatan lainnya, dengan menggunakan prinsip pelayanan yang efektif dan efisien serta menjunjung tinggi tanggung jawab profesional, hukum, etika dan moral. Layanan yang diselenggarakannya adalah sebatas kompetensi dasar kedokteran yang diperolehnya selama pendidikan kedokteran (KKI, 2011).

Menurut (WFME, 2012) kompetensi yang harus dicapai seorang dokter meliputi tujuh area kompetensi atau kompetensi utama yaitu:

1. Keterampilan komunikasi efektif.
2. Keterampilan klinik dasar.
3. Keterampilan menerapkan dasar-dasar ilmu biomedik, ilmu klinik, ilmu perilaku dan epidemiologi dalam praktik kedokteran.
4. Keterampilan pengelolaan masalah kesehatan pada individu, keluarga ataupun masyarakat dengan cara yang komprehensif, holistik, bersinambung, terkoordinasi dan bekerja sama dalam konteks Pelayanan Kesehatan Primer.
5. Memanfaatkan, menilai secara kritis dan mengelola informasi.
6. Mawas diri dan mengembangkan diri atau belajar sepanjang hayat.
7. Menjunjung tinggi etika, moral dan profesionalisme dalam praktik.

## Dokter Spesialis

Dokter spesialis adalah dokter yang memiliki ilmu pengetahuan, keahlian dan keterampilan akan suatu bidang kedokteran tertentu secara mendalam. Dokter spesialis bertugas menangani penyakit-penyakit spesifik sesuai dengan keahliannya. Adapun lama program pendidikan spesialis bervariasi, bergantung pada spesialisasi ilmunya, namun rata-rata memerlukan waktu selama 8 semester. Spesialisasi sangat diperlukan untuk mengembangkan pelayanan medis di bidangnya. Seorang dokter spesialis dianjurkan untuk menangani pasien setelah pasien dirujuk oleh Dokter Praktik Umum untuk berkonsultasi dengan dokter spesialis. Dengan demikian, dokter spesialis berfungsi menyediakan layanan kesehatan sekunder setelah Dokter Praktik Umum (Tono, 2012).

## Dokter Praktik Umum

Dokter praktik umum (DPU), sering kali dikenal sebagai dokter umum adalah lulusan fakultus kedokteran yang memiliki kewenangan pelayanan hanya sebatas layanan primer. Cakupan layanan yang diberikan tidak dibatasi oleh jenis penyakit, jenis kelamin, sistem organ, atau golongan manusia. Seorang dokter umum berhak *member*ikan pelayanan pengobatan kepada seluruh pasien dengan kisaran penyakit yang luas. Dokter praktik umum berperan sebagai layanan kesehatan primer pada masyarakat. Namun demikian, bila seorang dokter umum tidak memiliki kecakapan yang memadai untuk menangani pasien maka DPU dapat merujuk pasien untuk melanjutkan pengobatan pada dokter spesialis (Djuwanto, 2012).

## Keperawatan

Keperawatan adalah suatu bentuk pelayanan profesional yang merupakan bagian integral dari pelayanan kesehatan, didasarkan pada ilmu dan kiat keperawatan, berbentuk pelayanan bio-psiko-sosial-spiritual yang komprehensif, ditujukan pada individu, keluarga dan masyarakat baik sakit maupun sehat yang mencakup seluruh proses kehidupan manusia (Kusnano, 2006).

## Perawat

Perawat berasal dari bahasa latin yaitu kata nutrix yang berarti merawat atau memelihara. Menurut Kusnanto (2006), perawat adalah seorang profesional yang mempunyai kemampuan, tanggung jawab, dan kewenangan melaksanakan pelayanan atau asuhan keperawatan pada berbagai jenjang pelayanan keperawatan.

Perawat menurut Wardhono (2008) adalah orang yang telah menyelesaikan pendidikan profesional keperawatan, dan diberi kewenangan untuk melaksanakan peran serta fungsinya.

### Peran Perawat

Peran perawat adalah sebagai pelaksana pelayanan keperawatan, pengelola pelayanan keperawatan dan institusi pendidikan, sebagai pendidik dalam keperawatan, peneliti dan pengembangan keperawatan. atau peran adalah cara untuk menyatakan aktivitas perawat dalam praktik, di mana telah menyelesaikan pendidikan formalnya diakui dan diberi kewenangan oleh pemerintah untuk menjalankan tugas dan tanggung jawab keperawatan secara profesional, sesuai dengan kode etik profesinya (Asmadi, 2008:76).

Menurut Kusnanto, (2006:82) peran perawat dapat dibagi menjadi 5 bagian yaitu:

### Peran sebagai Pelaksana Kesehatan

Peran sebagai pelaksana yaitu keseluruhan kegiatan pelayanan masyarakat dalam mencapai tujuan kesehatan melalui kerja sama dengan tim kesehatan lainnya, dalam melaksanakan peran tersebut perawat bertindak selaku: pemberi rasa nyaman, pelindung dan pembela, *communicator,* mediator, dan rehabilitator.

### Peran sebagai Pendidik Administrasi

Perawat kesehatan yang diharapkan dapat mengelola kegiatan pelayanan kesehatan puskesmas dan masyarakat sesuai dengan beban tugas dan bertanggung jawab terhadap suatu permasalahan, mengambil keputusan dalam pemecah masalah, pengelolaan tenaga, membuat kualitas mekanis kontrol, dan bersosialisasi dengan masyarakat.

### Peran sebagai Pendidik

*Member*i pendidikan dan pemahaman kepada individu keluarga, kelompok dan masyarakat baik di rumah, puskesmas dan masyarakat dilakukan secara terorganisir dalam rangka menanamkan kualitas perilaku sehat, seperti yang diharapkan dalam mencapai tingkat kesehatan yang optimal.

### Peran sebagai konseling

Perawat kesehatan yang dapat dijadikan sebagai tempat bertanya individu, kelompok dan masyarakat untuk memecahkan berbagai permasalahan dalam bidang kesehatan dan keperawatan yang dihadapi dan akhirnya dapat membantu *member*ikan jalan keluar dalam mengatasi masalah dan keperawatan yang dengan melibatkan sumber-sumber lain, misalnya keluarga.

### Peran sebagai peneliti

Melakukan identifikasi terhadap fenomena yang terjadi di masyarakat yang dapat berpengaruh pada penurunan kesehatan bahkan mengancam kesehatan, selanjutnya penelitian dilaksanakan dalam kaitannya untuk menemukan faktor yang menjadi pencetus atau penyebab terjadinya permasalahan tersebut melalui kegiatan penelitian dalam praktik keperawatan.

## Konsultasi Kesehatan Daring

Menurut Peraturan Menteri Kesehatan (20:2019), telekonsultasi atau konsultasi daring klinis adalah pelayanan kesehatan yang dilakukan secara jarak jauh untuk membantu menegakkan diagnosis atau *member*i pertimbangam atau saran tata laksana. Konsultasi ini dapat dilaksanakan dalam bentuk tertulis suara dan video. serupa dengan konsultasi di rumah sakit, telekonsultasi klinis juga mesti masuk rekam medis pasien.

## Latitude

Latitude atau garis lintang adalah garis maya yang melingkari bumi ditarik dari arah barat hingga ke timur atau sebaliknya, sejajar dengan equator (garis khatulistiwa). Garis lintang terus melingkari bumi, dari equator hingga ke bagian kutub utara dan kutub selatan bumi. Titik yang dipakai dari 0 ke 90 derajat kearah kutub utara dan 0 ke -90 derajat ke kutub selatan.

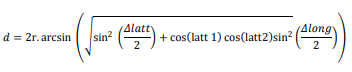
## Longitude

Longitude adalah garis lintang. Angka dari sudut bundar bumi horizontal. Titik diawali dari 0 ke 180 derajat, dan - ke arah sebaliknya. Titik 0 dimulai dari garis negara Inggris. Mengarah ke Indonesia akan menjadi angka positif. Kebalikannya koordinat Longitude minus adalah arah kebalikan.

## Teorema Haversine

Teorema Haversine digunakan untuk menghitung jarak antara titik di permukaan bumi menggunakan garis lintang (longitude) dan garis bujur (lattitude) sebagai variabel inputan. Haversine formula adalah persamaan penting pada navigasi, memberikan jarak lingkaran besar antara dua titik pada permukaan bola (bumi) berdasarkan bujur dan lintang (Dwi Prasetyo, Khafiizh Hastuti, 2015).

Teorema haversine merupakan metode untuk mengetahui jarak antar dua titik dengan memperhitungkan bahwa bumi bukanlah sebuah bidang datar namun adalah sebuah bidang yang memiliki derajat kelengkungan. Teorema haversine digunakan untuk menghitung jarak antara 2 titik dengan berdasarkan panjang garis lurus antara 2 titik pada garis bujur (lattitude) dan garis lintang (longitude). Dengan mengasumsikan bahwa bumi berbentuk bulat sempurna dengan jari-jari R 6.367, 45 km, dan lokasi dari 2 titik di koordinant bola (lintang dan bujur) masing-masing adalah lon1, lat1, dan lon2, lat2, maka rumus Haversine dapat ditulis dengan persamaan sebagai berikut:



Keterangan:

*d*  : jarak

*r* : jari-jari bumi adalah 6371 km

Δlatt : Selisih latitude daerah asal dengan daerah tujuan

Δlong : Selisih longitude daerah asal dengan daerah tujuan

## API

API (Application Programming Interface) adalah sekumpulan perintah, fungsi, class dan protocol yang memungkinkan sebuah perangkat lunak berhubungan dengan perangkat lunak lainnya. Tujuan utama dari API adalah untuk memudahkan programmer menggunakan kembali infrastruktur sebuah perangkat lunak tanpa membuang waktu untuk menulis dan membuat kembali infrastruktur tersebut.

## React Native

React Native merupakan sebuah Javascript framework untuk membangun sebuah aplikasi berbasis mobile untuk Android dan iOS dengan menggunakan React. React sendiri adalah sebuah framework yang dirilis oleh Facebook untuk membangun sebuah tampilan yang kaya dan interaktif. React Native akan melakukan kompilasi terhadap komponen React menjadi komponen Native iOS dan Android sehingga React Native akan melakukan render aplikasi dengan menggunakan komponen UI mobile, bukan *mobile* view. Selain itu, kode yang ditulis dengan menggunakan React Native dapat dijalankan kedua platform iOS dan Android. Biasanya, aplikasi berbasis iOS dan Android merupakan dua aplikasi yang terpisah. Aplikasi iOS dibuat menggunakan bahasa pemrograman Swift dan aplikasi Android dibuat menggunakan Kotlin atau Java. Dengan React Native, pembuatan aplikasi hanya perlu dibuat dengan satu Bahasa pemrograman saja dan dapat dibuat menjadi aplikasi iOS dan Android (Lim, 2020).

Sebuah halaman aplikasi React Native terdiri dari gabungan dari beberapa komponen. Ada dua cara pembuatan komponen React Native yaitu dengan menggunakan class dan function (Paul & Nalwaya, 2019).

## *Agile Development*

*Agile development* merupakan pendekatan lebih lanjut dari SDLC (*Software Development Lifecycle*) untuk memfasilitasi pengembangan aplikasi yang membutuhkan waktu yang singkat, dan memberikan tingkat keberhasilan pengembangan aplikasi lebih baik dari metode desain terstruktur (Permana, 2018).

*Agile development* menekankan alur terjadi revisi maka akan dilakukan iterasi atau perulangan tanpa menunggu proses selesai terlebih dahulu. Berikut ini adalah tahapan dilakukannya pembuatan aplikasi pelayanan kesehatan dan konsultasi masyarakat berbasis *mobile* menggunakan react native dengan metode agile development:

## UML (*Unified Modeling Language*)

### Pengenalan UML

*Unified Modeling Language* (UML) adalah sebuah bahasa pemodelan visual yang dirancang untuk khusus untuk pengembangan dan analisis sistem berorientasi objek dan desain. UML pertama kali dikembangkan oleh Grady Booch, Jim Rumbaugh, dan Ivars Jacobson pada pertengahan tahun 1990. (Journal of Database Management: Keng Siau and Qing Cao, 2001:26).

### *Use Case* Diagram

*Use Case* Diagram adalah model persyaratan sistem pada level tinggi. *Use Case* Diagram digunakan untuk memvisualisasikan *Use Case*, sektor yang terkait beserta dengan interaksinya. Model visual dari penggunaan *Use Case* memfasilitasi pemahaman proses bisnis dan bantuan dalam komunikasi dengan pemangku kepentingan. Spesifikasi dan dokumentasi *Use Case* yang ditunjukkan dalam bentuk diagram dapat membentuk inti dari pemodelan persyaratan (Unhelkar, 2018).

Tabel 2.1 Simbol *Use Case* Diagram

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Keterangan** |
|  | Aktor | Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi |
|  | *Use Case* | Mewakili apa yang sistem dapat lakukan. |
|  | *Association* | Merelasikan *actor* dengan *use case* |
|  | *System Boundary* | Menggambarkan batasan sistem terhadap lingkungannya |
|  | *Include* | Menunjukkan bahwa suatu *use case* seluruhnya merupakan fungsionalitas dari *use case* lainnya |
|  | *Extend* | Menunjukkan bahwa suatu *use case* merupakan tambahin fungsional dari *use case* lainnya jika suatu kondisi terpenuhi |
|  | *Generalization* | Menunjukkan spesialisasi actor untuk dapat berpartisipasi dengan *use case* |

Sumber: Ahaddin, 2015.

### *Activity* Diagram

*Activity* Diagram menggambarkan aliran kerja, atau proses, dalam suatu sistem. Karena itu, mereka tergambar seperti flowchart. *Activity* Diagram mendokumentasikan perilaku internal yang ada dalam Use Case, antara Use Case, atau bisnis secara keseluruhan (Unhelkar, 2018).

Tabel 2.2 Simbol *Activity* Diagram

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Keterangan** |
|  | *Start* | Status awal sebuah aktivitas |
|  | Aktivitas | Aktivitas yang dilakukan oleh sistem |
|  | Kondisi | Suatu kondisi atau sebuah pilihan |
|  | Penggabungan | Menggabungkan lebih dari satu aktivitas menjadi satu |
|  | *End* | Status akhir yang dilakukan sistem |
|  | *Swimlane* | *Swimlane* memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi |

Sumber: Idris, 2018.

## *Entity Relationship* Diagram(ERD)

Entity Relationship Diagram adalah suatu diagram untuk menggambarkan desain konseptual dari suatu basis data relasional. ERD juga merupakan gambaran yang menghubungkan antara objek satu dengan objek lain dalam dunia nyata. Dapat dikatakan bahwa bahan yang akan digunakan untuk membuat ERD adalah dari objek nyata (Utami, 2012).

Dalam ERD, hubungan (relasi) dapat terdiri dari sejumlah entitas yang disebut dengan derajat relasi. Derajat relasi maksimum disebut dengan kardinalitas, sehingga kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas lain. Kardinalitas yang terjadi antara dua himpunan entitas dapat berupa (Setiadi, 2017):

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Keterangan** |
|  | *Entity* | Suatu objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain. |
|  | *Relation* | Menunjukkan adanya hubungan di antara sejumlah *entity* yang berbeda. |
|  | *Atribute* | Mendeskripsikan karakter dari suatu *entity*. |
|  | *Atribute Key* | Satu atau gabungan dari beberapa atribute yang dapat membedakan semua baris data (*Row*/*Record*) dalam tabel secara unik. |
|  | *Line* | Sebagai penghubung antara *relation* dengan  *entity*. |

Sumber: Setiadi, 2017.

* + - 1. *One to One* (1-1)

Pada relasi ini, satu entitas dihubungkan dengan satu entitas lainnya. Setiap entitas pada himpunan entitas A dapat berelasi dengan paling banyak satu entitas pada himpunan B, demikian lalu sebaliknya.

* + - 1. *One to Many* (2-N)

Pada relasi ini, satu entitas dihubungkan dengan banyak entitas lainnya. Setiap entitas pada himpunan entitas A dapat berelasi dengan banyak entitas pada himpunan B, tetapi tidak sebaliknya.

* + - 1. *Many to One* (N-1)

Pada relasi ini, banyak entitas berkaitan dengan satu entitas. Setiap entitas pada himpunan A dapat berelasi dengan satu entitas pada himpunan B. tetapi tidak sebaliknya.

* + - 1. *Many to Many* (N-N)

Pada relasi ini, banyak entitas berkaitan dengan banyak entitas juga. Setiap entitas pada himpunan A dapat berelasi dengan banyak entitas pada himpunan B, demikian juga sebaliknya.

## *Global Positioning System* (GPS)

GPS adalah sistem satelit navigasi dan penentuan posisi yang berbasiskan pada pengamatan satelit-satelit *Global Positioning System.* Nama formalnya adalah NAVSTAR GPS, Kependekan dari “Navigation Satellite Timing and Ranging Global Position System”. Sistem yang dapat digunakan oleh banyak orang sekaligus dalam segala cuaca ini, didesain untuk *member*ikan posisi dan kecepatan tiga dimensi yang teliti dan juga informasi mengenai waktu, secara kontinyu di seluruh dunia (Abidin, 2006).

## OpenStreetMap

OpenStreetMap (OSM) merupakan sebuah proyek berbasis *mobile* untuk membuat peta seluruh dunia yang gratis dan terbuka, dibangun sepenuhnya oleh sukarelawan dengan melakukan survey menggunakan GPS, mendigitasi citra satelit, dan mengumpulan serta membebaskan data geografis yang tersedia di public.

OpenStreetMap dibuat oleh Steve Coast pada tahun 2004 untuk pemetaan di United Kingdom. Pada bulan April tahun 2006, OpenStreetMap Foundation berdiri, untuk membantu dalam perkembangannya, pengembangannya dan pendistribusiannya terhadap data Geospatial Gratis agar dapat digunakan dan dibagikan untuk semua orang.

## *Black Box Testing*

*Black box testing* adalah pengujian yang dilakukan dengan hanya mengamati hasil eksekusi melalui data uji dan dan cek fungsional perangkat lunak. Analoginya seperti melihat sebuah kotak hitam, di mana yang dapat terlihat hanya penampilan luar tanpa mengetahui apa di balik bungkus hitamnya. Sehingga *black box testing* hanya mengevaluasi penampilan eksternal (antarmuka) dan fungsional tanpa melihat *script coding*. (Kurniawati, 2018).

Tujuan *black box testing* adalah mencari kesalahan dan kegagalan dalam operasi tingkat tinggi, yang mencakup kemampuan dari perangkat lunak, operasional atau tata laksana maupun skenario pemakai. Fungsi dari pengujian ini berdasarkan kepada apa yang didapat oleh aplikasi. Untuk melakukan pengujian perilaku seseorang harus mengerti lingkup dari aplikasi, solusi bisnis yang diberikan oleh aplikasi, dan tujuan sistem dibuat. (Kurniawati, 2018).

# 

# **BAB III METODE PELAKSANAAN**

## Metode Pengembangan Perangkat Lunak

Pengembangan aplikasi pelayanan kesehatan dan konsultasi masyarakat berbasis *mobile* menggunakan metode pengembangan perangkat lunak Agile scrum. Agile adalah metodologi pengembangan perangkat lunak yang dilakukan berulang-ulang. Scrum merupakan sebuah prinsip pengembangan perangkat lunak dengan pendekatan Agile, yang dalam prosesnya berfungsi untuk menetapkan proses berjalannya proyek, kekuatan kolaborasi tim untuk mencapai tujuan akhir. Metode ini digunakan untuk membangun proyek yang dikerjakan secara cepat mengalami perubahan atau adaptif. Scrum terdiri atas beberapa aktivitas di dalamnya yaitu: product backlog, Sprint Backlog, daily scrum, dan Sprint Review.

1. *Product Backlog*

Product Backlog merupakan sebuah daftar fitur yang dibutuhkan sebagai bagian dari aplikasi yang akan menyelesaikan atas permasalahan yang telah dipaparkan pada bab satu. Pada tahap ini, penulis membuat daftar fitur yang akan dikerjakan untuk menunjang aplikasi pelayanan kesehatan dan konsultasi masyarakat berbasis *mobile*. Adapun daftar fitur dapat dilihat di halaman lampiran tabel.

1. *Sprint Backlog*

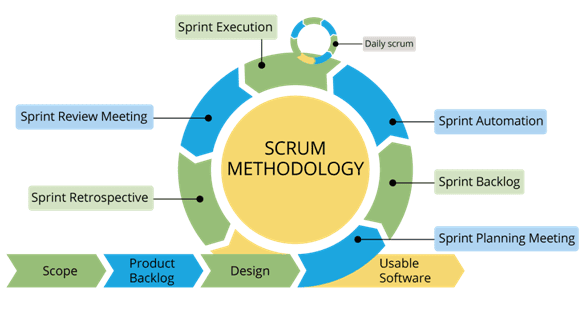
*Sprint Backlog* adalah aktivitas untuk menentukan berapa lama waktu yang dibutuhkan dalam mengerjakan *product backlog.* Tahapan ini juga menjelaskan *backlog* apa saja yang akan dikerjakan terlebih dahulu.Penulis merincikan berapa lama waktu yang dibutuhkan untuk mengerjakan *backlog*. Adapun *Sprint Backlog* aplikasi pelayanan kesehatan dan konsultasi masyarakat berbasis *mobile* dapat dilihat pada halaman lampiran tabel.

1. *Daily Activity*

Tahapan ini merupakan aktivitas untuk menyampaikan kegiatan yang telah dilakukan sebelumnya, seperti kendala apa saja yang dialami ketika mengerjakan *product backlog* dan *Sprint Backlog* aplikasi pelayanan kesehatan dan konsultasi masyarakat. Pada tahap ini, penulis menyampaikan kendala terhadap dosen pembimbing selaku *product owner.*

1. *Sprint Review*

*Sprint Review* merupakan tahap menyampaikan hasil dari *Sprint Backlog.* Pada tahapan ini *product* sudah siap untuk di-*review* oleh *product owner.* Adapun *sprint review* dapat dilihat pada bab 4.



Gambar 3.1 Metode Agile Scrum

## Analisis Kebutuhan Sistem

Dalam proses membangun aplikasi pelayanan kesehatan dan konsultasi masyarakat berbasis *mobile* ini terdapat kebutuhan fungsional dan kebutuhan non-fungsional. Berikut adalah kebutuhan-kebutuhan yang diperlukan:

### Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan akan fasilitas yang dibutuhkan serta fasilitas apa saja yang dilakukan oleh sistem secara umum. Kebutuhan fungsional adalah jenis kebutuhan yang berisi proses-proses apa saja yang nantinya dilakukan dalam aplikasi pelayanan kesehatan dan konsultasi masyarakat berbasis *mobile* ini.

1. Kebutuhan *Member*

Kebutuhan *Member* dalam mendapatkan pelayanan kesehatan yaitu:

* + - 1. Melakukan login ke dalam aplikasi.
      2. Melakukan ubah kata sandi akun.
      3. Melakukan ubah profil akun.
      4. Mendapatkan informasi kesehatan berupa artikel kesehatan.
      5. Mendapatkan informasi klinis berupa rumah sakit dan apotek terdekat dengan *member* berdasarkan lokasi.
      6. Memesan dan melakukan pembayaran produk obat dan vitamin.
      7. Mendapatkan informasi mengenai daftar dokter dan perawat yang tersedia dalam aplikasi.
      8. Melakukan pencarian dokter berdasarkan keahliannya.
      9. Melakukan pencarian perawat berdasarkan keahliannya.
      10. Melakukan pencarian rumah sakit dan klinik.
      11. Melakukan konsultasi kesehatan kepada dokter atau perawat.
      12. Membuat janji temu dokter di rumah sakit.
      13. Melihat riwayat transaksi janji kunjungan rumah sakit.

1. Kebutuhan *Guest*

Kebutuhan *Guest* untuk mendapatkan pelayanan kesehatan yaitu:

Melakukan registrasi akun.

Mendapatkan informasi berupa artikel kesehatan.

Mendapatkan informasi mengenai obat dan vitamin yang tersedia di dalam apotek.

Mendapatkan informasi mengenai daftar dokter dan perawat yang tersedia dalam aplikasi.

Melakukan pencarian dokter berdasarkan keahliannya.

Melakukan pencarian perawat berdasarkan keahliannya.

Melakukan pencarian rumah sakit dan apotek.

1. Kebutuhan Dokter

Kebutuhan fungsionalitas dokter dalam *member*ikan pelayanan kesehatan yang optimal kepada *member* dan *guest* membutuhkan hal berikut yaitu:

1. Melakukan login ke dalam aplikasi.
2. Melakukan ubah profil akun.
3. Melakukan ubah kata sandi akun.
4. Membalas pesan konsultasi *member.*
5. Memasukkan nominal harga untuk biaya konsultasi chat.
6. Membuat resep obat setelah konsultasi dengan member.
7. Kebutuhan Perawat

Kebutuhan fungsionalitas perawat dalam *member*ikan pelayanan kesehatan yang optimal kepada *member* dan *guest* membutuhkan hal berikut yaitu:

1. Melakukan login ke dalam aplikasi.
2. Melakukan ubah profil akun.
3. Melakukan ubah profil akun.
4. Membalas pesan konsultasi *member.*
5. Memasukkan nominal harga untuk biaya konsultasi chat.
6. Mengambil pemesanan jasa perawatan *member.*

### Kebutuhan Non Fungsional

Kebutuhan non-fungsional merupakan kebutuhan yang tidak terkait secara langsung dengan fitur tertentu dalam aplikasi. Namun, kebutuhan non-fungsional membantu jalannya kebutuhan fungsional. Berikut adalah kebutuhan non-fungsional dalam aplikasi pelayanan kesehatan dan konsultasi masyarakat berbasis *mobile*, yaitu:

1. Aplikasi dapat dijalankan pada *mobile device* (android)*.*
2. Aplikasi dapat digunakan apabila terhubung dengan internet.
3. Aplikasi memiliki tampilan antarmuka yang dapat dipahami oleh pengguna.

### Kebutuhan *Hardware*

Daftar kebutuhan *hardware* (perangkat keras) dalam proses pembangunan aplikasi pelayanan kesehatan dan konsultasi masyarakat berbasis *mobile* untuk analisis sentiment ini terdapat pada tabel 3.1.

Tabel 3.1Tabel Kebutuhan Hardware

|  |  |  |
| --- | --- | --- |
| **No** | **Jenis *Hardware*** | **Kebutuhan *Hardware*** |
| 1. | *Processor* | Intel Core i3 |
| 2. | Memori | 12 GB |
| 3. | Hardisk | 128 GB atau lebih |
| 4. | Tampilan | 12.5" FHD (1920 x 1080) IPS Touch, 300 nits |

### Kebutuhan *Software*

Daftar kebutuhan *software* (perangkat lunak) dalam proses membangun aplikasi pelayanan kesehatan dan konsultasi masyarakat berbasis *mobile* untuk analisis sentimen ini terdapat pada tabel 3.2.

Tabel 3.2 Tabel Kebutuhan Software

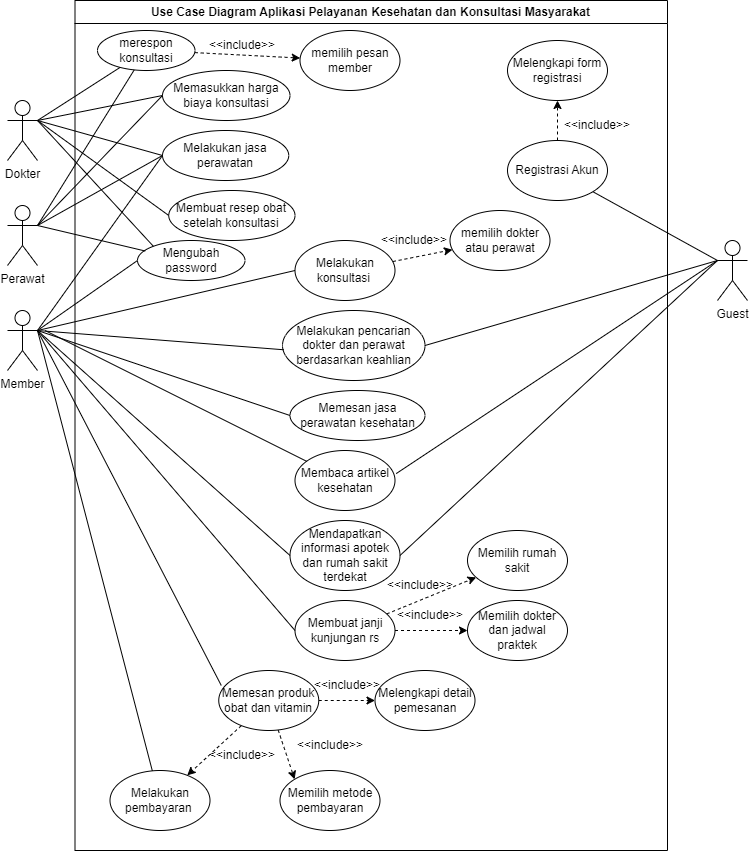
|  |  |  |
| --- | --- | --- |
| **No** | **Jenis *Software*** | **Kebutuhan *Software*** |
| 1. | Bahasa *Scripting* | JavaScript |
| 2. | *Software* Pengolah | VsCode |
| 3. | Penyimpanan Data | Basis data (MySql dan Firebase) |
| 4. | Jenis Ponsel | Oppo |
| 5 | Sistem Operasi | Android 12 |

## Perancangan Sistem

Perancangan sistem akan memaparkan mengenai rancangan dalam proses membangun aplikasi pelayanan kesehatan dan kesehatan masyarakat berbasis *mobile*. Perancangan sistem ini akan menjawab bagaimana mengatasi permasalahan yang terjadi agar sistem dapat berjalan dengan baik. Dalam penerapannya, sistem akan menggunakan *Unified Modelling Language* sebagai model.

### Use Case Diagram

*Use Case* Diagrammerupakan diagram yang menjelaskan hubungan antara aktor dengan sistem. Dalam aplikasi ini menjelaskan empat aktor yang memiliki beberapa aksi. Adapun rancangan *Use Case* dari aplikasi pelayanan kesehatan dan konsultasi masyarakat dapat dilihat pada Gambar 3.1.



Gambar 3.2 Use Case Diagram

Adapun penjelasan yang lebih rinci dari Gambar 3.1 *Use Case* Diagramdi atas dapat dilihat pada Tabel 3.3.

Tabel 3.3 Tabel Penjelasan Use Case Diagram

|  |  |  |
| --- | --- | --- |
| ***Actor*** | **Nama *Use Case*** | **Keterangan** |

|  |  |  |
| --- | --- | --- |
| Dokter | Menetapkan harga konsultasi | Menu ini digunakan oleh dokter untuk menetapkan harga konsultasi dokter. |
| Mengubah Profil | Menu ini digunakan oleh dokter untuk mengubah profil. |
| Mengubah Password | Menu ini digunakan oleh dokter untuk mengubah password akun. |
| Membuat resep obat setelah konsultasi | Menu ini digunakan dokter dalam membuat resep obat untuk member |
| Merespon konsultasi | Menu ini digunakan oleh dokter untuk membalas pesan konsultasi yang dikirimkan oleh *member* |
| Perawat | Menetapkan harga konsultasi | Menu ini digunakan oleh perawat untuk menetapkan harga konsultasi perawat. |
| Melakukan jasa perawatan | Menu ini digunakan perawat untuk melakukan perawatan ke member |
| Mengubah Password | Menu ini digunakan oleh perawat untuk mengubah password akun. |
| Merespon konsultasi | Menu ini digunakan oleh perawat untuk membalas pesan konsultasi yang dikirimkan oleh *member* |
| *Member* | Melakukan pencarian dokter dan perawat berdasarkan keahlian | Menu ini digunakan oleh *member* untuk mencari dokter atau perawat berdasarkan keahliannya. |
| Memesan jasa perawatan oleh perawat | Menu ini digunakan oleh member untuk memesan jasa perawatan oleh perawat terdekat |
| Membaca artikel kesehatan | Menu ini digunakan oleh *member* untuk membaca artikel kesehatan. |
| Mendapatkan informasi apotek dan rumah sakit terdekat | Menu ini digunakan oleh *member* untuk mengetahui rumah sakit dan apotek terdekat berdasarkan lokasi dari *member*. |
| Membuat janji rumah sakit | Menu ini digunakan oleh *member* untuk membuat janji kunjungan medis ke rumah sakit. *Member* wajib memilih rumah sakit dan jadwal dokter yang akan dikunjungi. |
| Mendapatkan nomor antrian janji rumah sakit | Setelah membuat janji, *member* akan mendapatkan nomor antrian kunjungan |
| Memesan obat dan vitamin | *Member* dapat melakukan pemesanan obat dan vitamin yang tersedi di dalam apotek. Setelah memesan, *member* diwajibkan untuk membayar biaya pemesanan. |
| Melakukan konsultasi | *Member* dapat melakukan konsultasi dengan dokter dan perawat melalui menu konsultasi. |
| Mengubah Profil | Menu ini digunakan oleh dokter untuk mengubah profil. |
| Mengubah Password | Menu ini digunakan oleh dokter untuk mengubah password akun. |
| *Guest* | Melakukan pencarian dokter dan perawat berdasarkan keahlian | Menu ini digunakan oleh *guest* untuk mencari dokter atau perawat berdasarkan keahliannya. |
| Melihat daftar dokter berdasarkan spesialis | Menu ini digunakan oleh *guest* untuk mendapatkan informasi daftar dokter berdasarkan spesialisnya. |
| Membaca artikel kesehatan | Menu ini digunakan oleh *guest* untuk membaca artikel kesehatan. |
| Mendapatkan informasi apotek dan rumah sakit terdekat | Menu ini digunakan oleh *member* untuk mengetahui rumah sakit dan apotek terdekat berdasarkan lokasi dari *member*. |
| Registrasi akun | *Guest* dapat melakukan registrasi akun. Dalam registrasi, *guest* diwajibkan untuk melengkapi form dan memilih *role* akun. |

### Activity Diagram

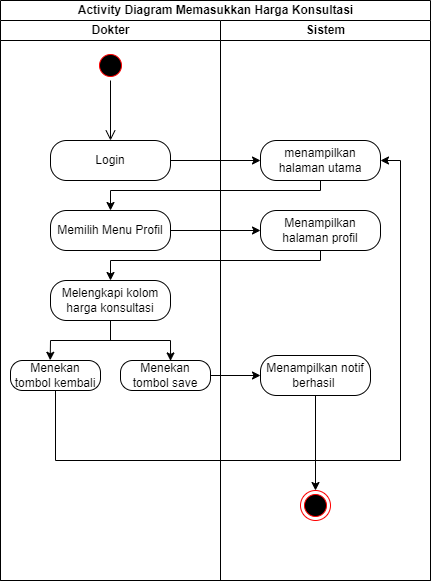
*Activity* Diagram menggambarkan aliran kerja, atau proses, dalam suatu sistem. Karena itu, mereka tergambar seperti flowchart. *Activity* Diagram mendokumentasikan perilaku internal yang ada dalam Use Case, antara Use Case, atau bisnis secara keseluruhan (Unhelkar, 2018).

Activity diagram ini menunjang penulis dalam membangun aplikasi, karena melalui activity diagram, penulis dapat mengetahui bagaimana alur sistem bekerja. Terdapat beberapa *activity* Diagramyang ada dalam proses perancangan sistem aplikasi pelayanan kesehatan dan konsultasi masyarakat berbasis *mobile* menggunakan react native. Berikut adalah *activity* Diagram:

1. *Activity* Diagram Menetapkan Harga Biaya Konsultasi Dokter dan Perawat

Pada *activity* diagram ini aktor dokter dan perawat menetapkan harga biaya konsultasi. Harga konsultasi ditentukan oleh dokter dan perawat itu sendiri. Aktivitas yang pertama dilakukan adalah login ke dalam aplikasi, sesudahnya memilih menu profil.

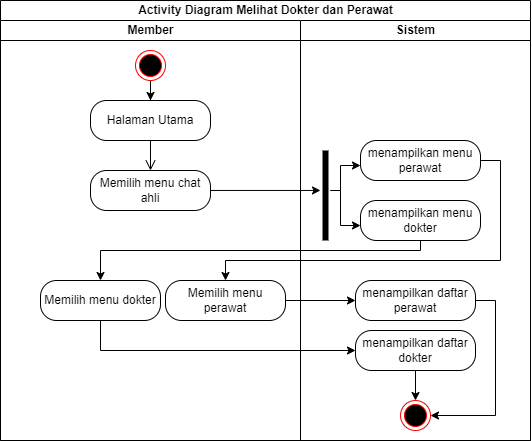
Pada halaman profil terdapat kolom harga biaya konsultasi, untuk menetapkan biaya konsultasi, pengguna akun dapat melengkapi kolom tersebut. Dokter dan Perawat dapat memasukkan biaya konsultasi dengan tanpa minimal dan maksimal biaya konsultasi. Adapun *activity* diagram menetapkan harga biaya konsultsai dokter dan perawat terdapat dalam Gambar 3.2.



Gambar 3.2 Activity Diagram Menetapkan Harga Konsultasi Dokter dan Perawat

1. *Activity* Diagram Melihat Dokter dan Perawat

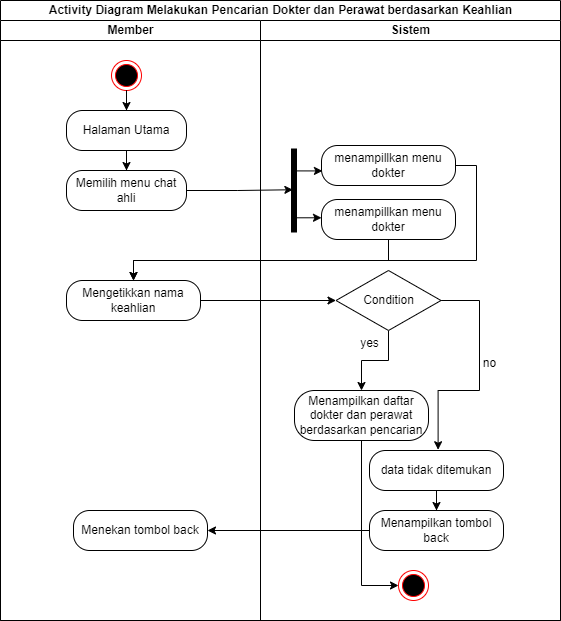
Pada *activity* diagram ini *member* melakukan aktivitas melihat dokter dan perawat. Langkah yang dilakukan adalah dengan masuk ke halaman utama aplikasi, sesudahnya menekan menu chat ahli, sistem akan menampilkan daftar dokter dan perawat. Adapun *activity* diagram melihat dokter berdasarkan spesialis dapat dilihat dalam Gambar 3.3.



Gambar 3.3 Activity Diagram Melihat Dokter dan Perawat

1. *Activity* Diagram Melakukan Pencarian Dokter dan Perawat dari keahlian

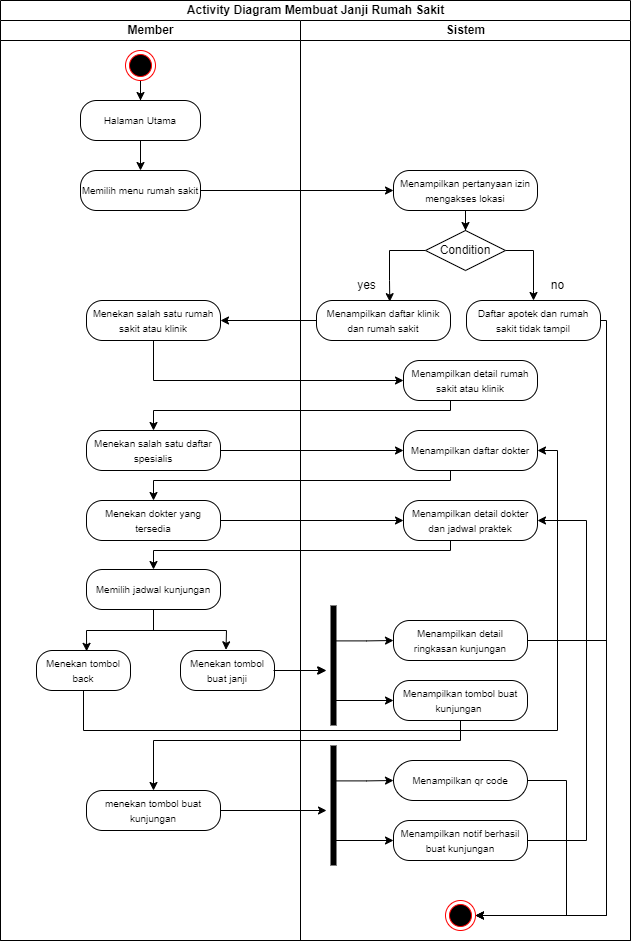
Pada *activity* diagram ini aktor *member* melakukan pencarian nama dokter dan perawat berdasarkan keahlian yang dimilikinya*.* Adapun *activity* diagram ini dapat dilihat pada Gambar 3.4.



Gambar 3.4 Activity Diagram Melakukan Pencarian Dokter dan Perawat

1. *Activiy* Diagram Membuat Janji Kunjungan Rumah Sakit

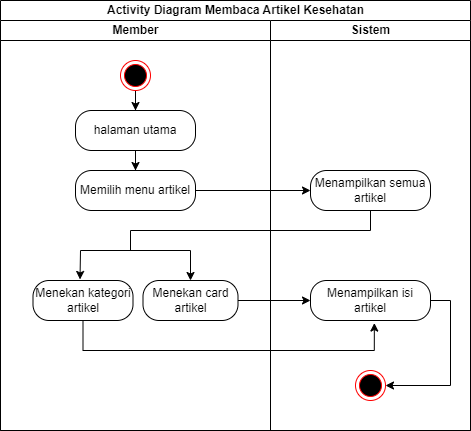
Member wajib melakukan login ke dalam aplikasi untuk membuat janji kunjungan. Member dapat memilih dokter dan jadwal praktiknya untuk melakukan kunjungan atau konsultasi secara langsung. Pembayaran biaya konsultasi kunjungan dilakukan di rumah sakit terkait. Adapun activity diagram ini dapat dilihat pada gambar 3.5.



Gambar 3.5 Activity Diagram Membuat Janji Kunjungan Rumah Sakit

1. *Activity* Diagram Membaca Artikel Kesehatan

Pada *activity* diagram ini *member* dan *guest* dapat membaca artikel kesehatan. Untuk membaca artikel, aktivitas yang pertama dilakukan adalah masuk ke dalam aplikasi, dapat dengan login terlebih dahulu atau tidak.. *Member* dapat membaca artikel berdasarkan kategori. Adapun *activity* diagram membaca artikel kesehatan dapat dilihat pada Gambar 3.6.

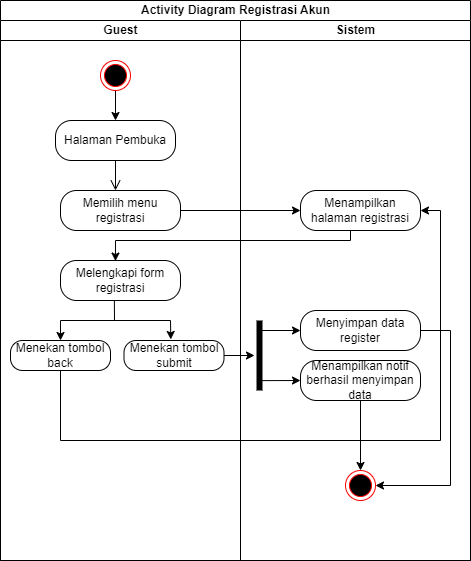


Gambar 3.6 Activity Diagram Membaca Artikel Kesehatan

1. *Activity* Diagram Registrasi Akun

Pada *activity* diagram ini *guest* melakukan registrasi akun. *Activity* diagram ini menggambarkan proses bagaimana mendapatkan jenis *role* akun aplikasi pelayanan kesehatan dan konsultasi masyarakat berbasis *mobile*. Aktivitas yang pertama dilakukan adalah membuka halaman utama aplikasi. Sesudahnya, aktor *guest* memilih menu registrasi. Kemudian sistem akan menampilkan halaman registrasi, aktor melengkapi form registrasi akun.. *Activity* diagram registrasi akun dapat dilihat pada Gambar 3.7.

Jenis akun selain *member* diwajibkan untuk mengunggah dokumen pendukung untuk aktivasi akun. Aktivasi akun dilakukan oleh admin aplikasi dan memakan waktu maksimal adalah tiga hari. Pendaftar akan diinformasikan melalui nomor telepon yang tersedia untuk memberi tahu bahwa proses validasi akun telah selesai dilaksanakan. Setelahnya, pendaftar dapat mengakses aplikasi.

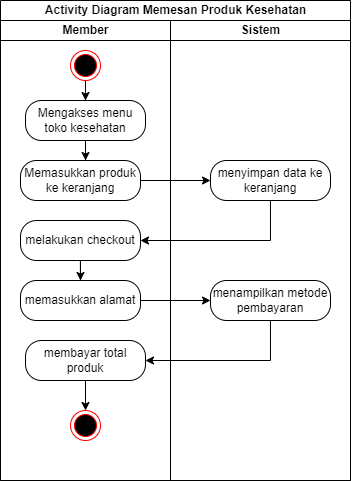


Gambar 3.7 Activity Diagram Registrasi Akun

1. *Activity* Diagram Memesan Produk Obat dan Vitamin.

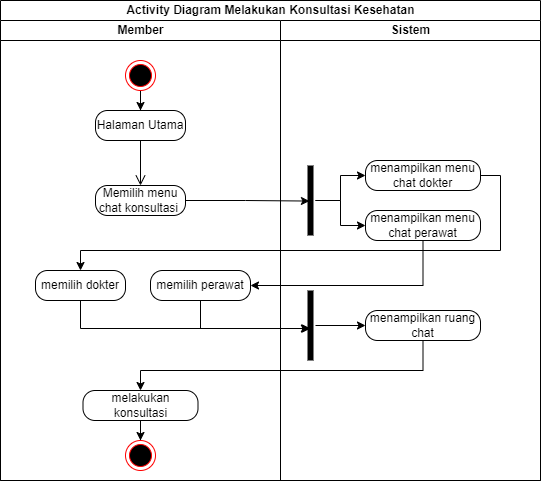
Pada *activity* diagram ini member melakukan pemesanan obat dan vitamin di apotek atau toko kesehatan yang tersedia dalam aplikasi. Aktivitas yang pertama dilakukan adalah login ke aplikasi, kemudian memilih menu toko kesehatan, kemudian sistem akan menampilkan seluruh obat dan vitamin.

Untuk membeli produk, *member* perlu menambahkan produk, ke dalam keranjang dengan menekan tombol keranjang yang tersedia di *card* produk. Setelahnya, *member* dapat melakukan pembayaran dengan memilih metode pembayaran yang tersedia di dalam aplikasi. Adapun activity diagram memesan obat dan produk dapat dilihat pada Gambar 3.8.



1. *Activity* Diagram Melakukan Konsultasi Kesehatan

Pada *activity* diagram ini, *member* melakukan konsultasi kesehatan dengan masuk ke halaman utama, setelahnya memilih menu chat konsultasi, kemudian sistem akan menampilkan menu chat dokter dan perawat. Jika *member* ingin melakukan konsultasi, perlu untuk memilih dokter atau perawat, sesudahnya sistem akan menampilkan ruang chat dengan dokter atau perawat yang dipilih. Adapun *activity* diagram melakukan konsultasi kesehatan dapat dilihat pada gambar 3.9.



Gambar 3.9 Activity Diagram Melakukan Konsultasi Kesehatan

1. *Activity* Diagram Memesan Jasa Rawat Jalan

Pada *activity* diagram memesan rawat jalan, *user* dapat menerima perawatan yang dilakukan oleh perawat. Untuk menggunakan fitur ini, pengguna atau member perlu melakukan konsultasi kesehatan terlebih dahulu dengan dokter. Hal tersebut perlu dikarenakan untuk mengetahui tindak lanjut penanganan yang sesuai.

Ketika konsultasi telah dilaksanakan, aplikasi akan menampilkan pilihan berupa lanjut rawat jalan atau tidak. Ketika memilih untuk lanjut rawat jalan, maka sistem akan mencari perawat terdekat dengan pengguna. Perawat terdekat akan menerima notifikasi jasa rawat jalan. Perawat dapat memproses rawat jalan ataupun tidak.

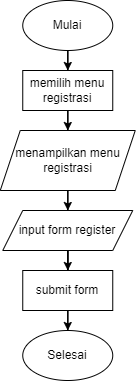


Gambar 3.11 Activity Diagram Mengubah Password

## Flowchart

### Flowchart Registrasi Akun

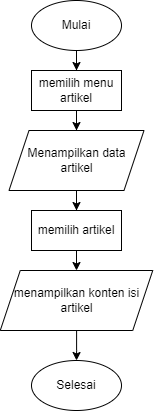
Flowchart berikut merupakan proses untuk melakukan pendaftaran akun. *Guest* akan memilih menu registrasi dan kemudian mengisi form registrasi. Setelahnya di-*submit*. Maka, data registrasi akan tersimpan.



Gambar 3.3 Flowchart Register

### Flowchart Membaca Artikel

Flowchart berikut merupakan proses untuk membaca artikel kesehatan. Proses utamanya adalah dengan memilih menu artikel kemudian menampilkan data artikel, ketika pengguna memilih artikel, sistem akan menampilkan konten berdasarkan artikel yang dipilih. KAdapun flowchart membaca artikel dapat dilihat pada gambar 3.19.



Gambar 3.4 Flowchart Membaca Artikel

### Flowchart Melihat Rumah Sakit Terdekat

Flowchart berikut merupakan proses untuk melihat rumah sakit terdekat berdasarkan lokasi. Adapun flowchart melihat rumah sakit terdekat dapat dilihat pada gambar 3.20.

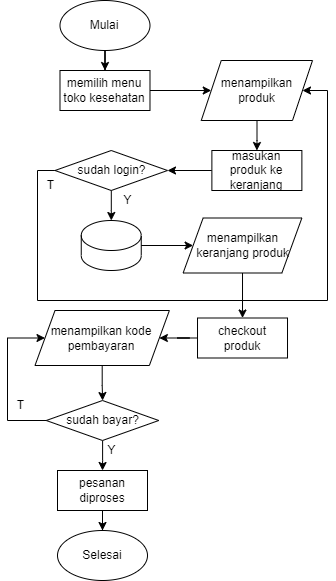


Gambar 3.5 Flowchart Melihat Rumah Sakit Terdekat

### Flowchart Memesan Produk Apotek

Flowchart berikut merupakan proses untuk memesan produk apotek. Proses yang pertama dilakukan adalah memilih menu toko kesehatan. Kemudian, sistem akan menampilkan produk. Data yang ditampilkan berupa nama produk, harga, dan jumlah ketersediaan produk. Untuk melakukan pembelian produk, pengguna aplikasi perlu melakukan login.

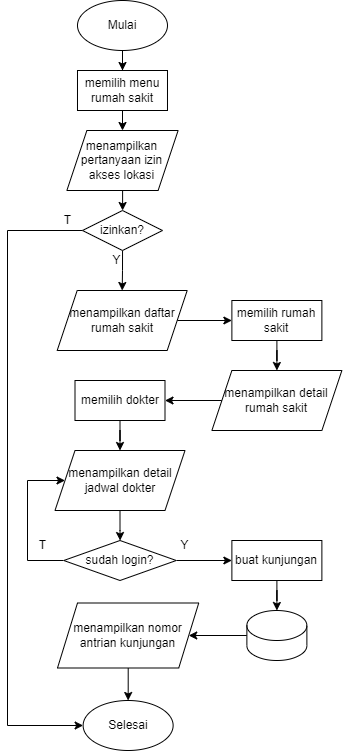
Member dapat memasukkan atau menampung produk yang akan dibeli ke dalam keranjang. Pengguna dapat membayar melalui berbagai jenis metode pembayaran yang telah disediakan oleh pihak *payment gateway.* Ketika sudah melakukan pembayaran, produk akan diproses oleh apotek. Adapun flowchart memesan produk apotek dapat dilihat pada gambar 3.21.



Gambar 3.6 Flowchart Memesan Produk Apotek

### Flowchart Membuat Janji Kunjungan Rumah Sakit

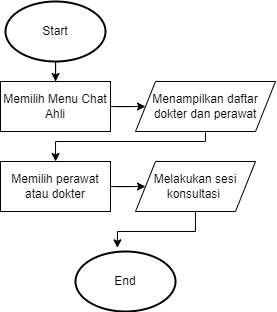
Flowchart berikut merupakan proses untuk membuat janji kunjungan rumah sakit. *Member*, memilih terlebih dahulu rumah sakit yang dituju, kemudian memilih dokter dan jadwal praktiknya. Setelahnya dapat membuat janji kunjungan temu dokter. Member akan mendapatkan nomor antrian kunjungan. Proses pembayaran biaya kunjungan dilakukan di rumah sakit yang dituju. Adapun flowchart membuat janji kunjungan rumah sakit dapat dilihat pada gambar 3.22.



Gambar 3.7 Flowchart Membuat Janji Kunjungan Rumah Sakit

### Flowchart Melakukan Konsultasi Kesehatan

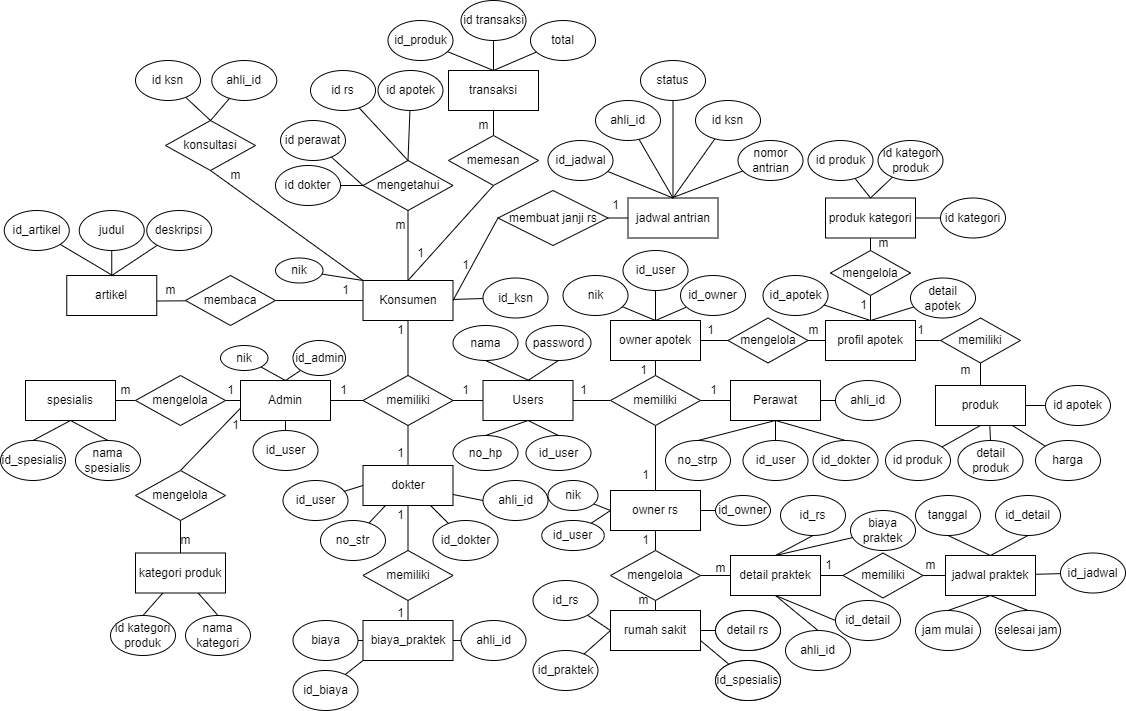
Flowchart melakukan konsultasi kesehatan merupakan proses untuk melakukan konsultasi kesehatan melalui *chat* dengan dokter dan perawat. *Member* yang telah memiliki akun dapat melakukan sesi konsultasi. Sesi konsultasi dapat dilakukan selama 30 menit secara gratis untuk akun yang baru melakukan pendaftaran. Adapun flowchart dapat dilihat pada gambar 3.8.



Gambar 3.8 Flowchart Melakukan Konsultasi Kesehatan

## Perancangan *Entity Relationship* Diagram (ERD)

Rancangan *Entity Relationship* Diagram (ERD) yang digunakan dalam pembangunan aplikasi pelayanan kesehatan dan konsultasi masyarakat berbasis *mobile* menggunakan react native terdapat pada gambar 3.18.



Gambar 3.9 Rancangan ERD

Adapun penjelasan dari Gambar 3.18 terdapat adalah pada Tabel 3.4 berikut ini:

Tabel 3.4 Penjelasan ERD

|  |  |
| --- | --- |
| **Nama tabel** | **Keterangan** |
| Users | Tabel yang digunakan untuk menyimpan data nama, nomor hp, dan password |
| Perawat | Tabel yang digunakan untuk menyimpan data perawat, juga berelasi dengan tabel users untuk melengkapi kolom lainnya. |
| Dokter | Tabel yang digunakan untuk menyimpan data dokter, juga berelasi dengan tabel users untuk melengkapi kolom lainnya. |
| Owner Rumah Sakit | Tabel yang digunakan untuk menyimpan data owner rumah sakit, juga berelasi dengan tabel users untuk melengkapi kolom lainnya. |
| Spesialis | Tabel ini berfungsi untuk menyimpan nama spesialis. |
| Artikel | Tabel ini berfungsi untuk menyimpan data artikel. |
| Kategori Produk | Tabel ini berfungsi untuk menyimpan kategori produk |
| Owner Apotek | Tabel yang digunakan untuk menyimpan data owner apotek, juga berelasi dengan tabel users untuk melengkapi kolom lainnya. |
| Admin | Tabel yang digunakan untuk menyimpan data admin rumah sakit, juga berelasi dengan tabel users untuk melengkapi kolom lainnya. |
| Konsumen | Tabel ini digunakan untuk menyimpan data *member* dan berelasi dengan tabel users. |
| Transaksi | Tabel ini digunakan untuk menyimpan data transaksi produk. Tabel transaksi berelasi dengan tabel produk. |
| Jadwal Antrian | Tabel ini digunakan untuk menyimpan data antrian kunjungan rumah sakit. |
| Apotek | Tabel ini digunakan untuk menyimpan detail dari apotek. |
| Rumah Sakit | Tabel ini digunakan untuk menyimpan detail dari rumah sakit. |
| Detail Praktik | Tabel ini digunakan untuk mengelola praktik dokter di dalam rumah sakit. |
| Jadwal Praktik | Tabel ini digunakan untuk mengelola jadwal praktik dokter di dalam rumah sakit. |
| Biaya Praktik | Tabel ini digunakan untuk mengatur harga konsultasi dokter. |
| Produk | Tabel ini digunakan untuk digunakan untuk menyimpan data produk di apotek. |
| Produk Kategori | Tabel ini digunakan untuk mengelompokkan produk dengan kategori produk |

## Perancangan Basis Data

Dalam membangun aplikasi pelayanan kesehatan dan konsultasi masyarakat berbasis *mobile* dibutuhkan perancangan basis data untuk menunjang aplikasi ini. Perancangan basis data ini dibagi menjadi 2 jenis tabel, terdapat tabel master data dan tabel proses bisnis. Tabel master data meliputi tabel-tabel yang berisi data penunjang dan tabel proses bisnis berisi tabel-tabel untuk proses bisnis. Berikut merupakan detail dari perancangan basis data dapat dilihat berikut:

### Tabel Master Data

* + - 1. Tabel kategori artikel

Tabel ini digunakan untuk menyimpan data kategori artikel yang memiliki id\_kategori\_artikel, slug\_kategori, dan nama\_kategori. Adapun detail dari tabel kategori artikel dapat dilihat pada tabel 3.5.

Tabel 3.5 Tabel Kategori Artikel

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1. | id\_kategori\_artikel | varchar(50) | NOT NULL |
| 2. | nama\_kategori | varchar(50) | NOT NULL |
| 3. | slug\_kategori | varchar(100) | NOT NULL |

* + - 1. Tabel Kategori Produk

Tabel ini digunakan untuk menyimpan data kategori produk yang memiliki id\_kategori\_produk, slug\_kategori\_produk, dan nama\_kategori\_produk. Adapun detail dari tabel kategori produk dapat dilihat pada tabel 3.6.

Tabel 3.6 Tabel Kategori Produk

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1. | id\_kategori\_produk | varchar(50) | NOT NULL |
| 2. | nama\_kategori\_produk | varchar(50) | NOT NULL |
| 3. | slug\_kategori\_produk | varchar(100) | NOT NULL |

* + - 1. Tabel Spesialis\_Penyakit

Tabel ini digunakan untuk menyimpan data spesialis yang memiliki id\_spesialis\_penyakit, nama\_spesialis, logo, dan slug\_spesialis. Adapun detail dari tabel spesialis dapat dilihat pada tabel 3.7.

Tabel 3.7 Tabel Spesialis

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1. | id\_spesialis | varchar(50) | NOT NULL |
| 2. | nama\_spesialis | varchar(50) | NOT NULL |
| 3. | slug\_spesialis | varchar(100) | NOT NULL |
| 4. | logo | varchar(255) | NOT NULL |

* + - 1. Tabel Spesialis Rumah Sakit

Tabel ini digunakan untuk menyimpan data spesialis rumah sakit yang memiliki id\_spesialis, id\_rumah\_sakit, id\_penyakit. id\_rumah\_sakit merupakan relasi dari tabel rumah sakit dan id\_spesialis relasi dari tabel spesialis. Adapun detail dari tabel spesialis rumah sakit dapat dilihat pada tabel 3.8.

Tabel 3.8 Tabel Spesialis Rumah Sakit

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1. | id\_spesialis | varchar(50) | NOT NULL |
| 2. | id\_rumah\_sakit | varchar(50) | NOT NULL |
| 3. | id\_penyakit | varchar(50) | NOT NULL |

* + - 1. Tabel Fasilitas Rumah Sakit

Tabel ini digunakan untuk menyimpan data fasilitas rumah sakit yang memiliki id\_fasilitas, id\_rumah\_sakit, nama\_fasilitas. Adapun detail dari tabel fasilitas rumah sakit dapat dilihat pada tabel 3.9.

Tabel 3.9 Tabel Fasilitas Rumah Sakit

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1. | id\_fasilitas | varchar(50) | NOT NULL |
| 2. | id\_rumah\_sakit | varchar(50) | NOT NULL |
| 3. | nama\_fasilitas | varchar(50) | NOT NULL |

* + - 1. Tabel *grouping* artikel

Tabel ini digunakan untuk menyimpan data *grouping* artikel yang memiliki id\_grouping\_artikel, id\_artikel, dan id\_kategori\_artikel. Adapun detail dari tabel *grouping* artikel dapat dilihat pada tabel 3.10.

Tabel 3.10 Tabel Grouping Artikel

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1. | id\_grouping\_artikel | varchar(50) | NOT NULL |
| 2. | id\_artikel | varchar(50) | NOT NULL |
| 3. | id\_kategori\_artikel | varchar(50) | NOT NULL |

* + - 1. Tabel *Grouping* Produk

Tabel ini digunakan untuk menyimpan data produk kategori yang memiliki id\_kategori\_produk, kode\_produk, id\_kategori\_produk. Adapun detail dari tabel produk kategori dapat dilihat pada tabel 3.11.

Tabel 3.11 Tabel Produk Kategori

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1. | id\_produk\_kategori | varchar(50) | NOT NULL |
| 2. | kode\_produk | varchar(50) | NOT NULL |
| 3. | id\_kategori\_produk | varchar(50) | NOT NULL |

* + - 1. Tabel Profil Apotek

Tabel ini digunakan untuk menyimpan data apotek yang memiliki id\_profil\_apotek, detail apotek, dan id\_user. Adapun detail dari tabel profil apotek dapat dilihat pada tabel 3.12.

Tabel 3.12 Tabel Profil Apotek

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1. | id\_profil\_apotek | varchar(50) | NOT NULL |
| 2. | nama\_apotek | varchar(100) | NOT NULL |
| 3. | slug\_apotek | varchar(255) | NOT NULL |
| 4. | deskripsi\_apotek | text | NOT NULL |
| 5. | nomor\_hp | varchar(30) | NOT NULL |
| 6. | foto\_apotek | varchar(30) | DEFAULT NULL |
| 7. | status | tinyint(4) | NOT NULL |
| 8. | id\_user | varchar(255) | NOT NULL |
| 9. | latitude | varchar(100) | DEFAULT NULL |
| 10. | ;ongitude | varchar(100) | DEFAULT NULL |

* + - 1. Tabel Rumah Sakit

Tabel ini digunakan untuk menyimpan data rumah sakit yang memiliki id\_rumah\_sakit, detail\_rumah\_sakit, dan id\_user. Adapun detail dari tabel rumah sakit dapat dilihat pada tabel 3.13.

Tabel 3.13 Tabel Rumah Sakit

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1. | id\_rumah\_sakit | varchar(50) | NOT NULL |
| 2. | nama\_rs | varchar(100) | NOT NULL |
| 3. | slug\_rs | varchar(255) | NOT NULL |
| 4. | deskripsi\_rs | text | NOT NULL |
| 5. | nomor\_hp | varchar(30) | NOT NULL |
| 6. | foto\_rs | varchar(30) | DEFAULT NULL |
| 7. | id\_owner\_rumah\_sakit | varchar(30) | NOT NULL |
| 8. | id\_user | varchar(255) | NOT NULL |
| 9. | latitude | varchar(100) | DEFAULT NULL |
| 10. | longitude | varchar(100) | DEFAULT NULL |

* + - 1. Tabel Keahlian

Tabel ini digunakan untuk menyimpan data keahlian dokter dan perawat yang memiliki kolom nama\_keahlian, id\_keahlian, id\_spesialis\_penyakit, dan logo. Adapun detail dari tabel keahlian dapat dilihat pada tabel 3.14.

Tabel 3.14 Tabel Keahlian

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1. | id\_keahlian | varchar(50) | NOT NULL |
| 2. | id\_spesialis\_penyakit | varchar(100) | NOT NULL |
| 3. | nama\_keahlian | varchar(255) | NOT NULL |
| 4. | logo | varchar(255) | DEFAULT NULL |

* + - 1. Tabel Join Keahlian

Tabel ini digunakan untuk menyimpan data dari proses menggabungkan keahlian dengan dokter atau perawat. Adapun detail dari tabel join keahlian dapat dilihat pada tabel 3.15.

Tabel 3.15 Tabel Join Keahlian

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1. | id\_master\_join\_keahlian | varchar(50) | NOT NULL |
| 2. | user\_ahli\_id | integer(11) | NOT NULL |
| 3. | keahlian\_id | varchar(50) | NOT NULL |
| 4. | logo | varchar(255) | DEFAULT NULL |

* + - 1. Tabel Detail Praktik

Tabel ini digunakan untuk menyimpan data detail praktik yang memiliki id\_detail\_praktik, ahli\_id, id\_rumah\_sakit, dan biaya\_praktik. Adapun detail tabel detail praktik dapat dilihat pada tabel 3.16.

Tabel 3.16 Tabel Detail Praktik

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1. | id\_detail\_praktik | varchar(50) | NOT NULL |
| 2. | ahli\_id | tinyint(11) | NOT NULL |
| 3. | id\_rumah\_sakit | varchar(255) | NOT NULL |
| 4. | biaya\_praktik | varchar(255) | NOT NULL |

* + - 1. Tabel Jadwal Praktik

Tabel ini digunakan untuk menyimpan data jadwal praktik yang memiliki id\_detail\_praktik, id\_jadwal\_praktik, tanggal, mulai\_jam, dan selesai\_jam. Adapun detail tabel detail praktik dapat dilihat pada tabel 3.16.

Tabel 3.17 Tabel Jadwal Praktik

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1. | id\_detail\_praktik | varchar(50) | NOT NULL |
| 2. | id\_jadwal\_praktik | varchar(50) | NOT NULL |
| 3. | tanggal | date | NOT NULL |
| 4. | mulai\_jam | time | NOT NULL |
| 5. | selesai\_jam | time | NOT NULL |

### Tabel Proses Bisnis

Tabel Transaksi

Tabel ini digunakan untuk menyimpan data transaksi produk yang memiliki id\_transaksi, user\_id, total, dan pembayaran. Adapun penjelasan terkait tabel transaksi dapat dilihat pada tabel 3.18.

Tabel 3.18 Tabel Transaksi

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1. | id\_transaksi | varchar(50) | NOT NULL |
| 2. | Nama\_konsumen | varchar(50) | NOT NULL |
| 3. | User\_id | varchar(50) | NOT NULL |
| 4. | total | varchar(50) | NOT NULL |
| 5. | pembayaran | varchar(50) | NOT NULL |

Tabel Artikel

Tabel ini digunakan untuk menyimpan data artikel yang memiliki id\_artikel, judul\_artikel, slug\_artikel, deskripsi, foto, dan user\_id. Adapun penjelasan terkait tabel artikel dapat dilihat pada tabel 3.19.

Tabel 3.19 Tabel Artikel

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1. | id\_artikel | varchar(50) | NOT NULL |
| 2. | judul\_artikel | varchar(100) | NOT NULL |
| 3. | slug\_artikel | varchar(100) | NOT NULL |
| 4. | foto | varchar(255) | DEFAULT NULL |
| 5. | deskripsi | text | NOT NULL |
| 6. | user\_id | varchar(255) | NOT NULL |

Tabel Jadwal Antrian

Tabel ini digunakan untuk menyimpan data antrian janji rumah sakit. Adapun penjelasan terkait tabel jadwal antrian dapat dilihat pada tabel 3.20.

Tabel 3.20 Tabel Jadwal Antrian

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1. | id\_jadwal\_antrian | varchar(50) | NOT NULL |
| 2. | konsumen\_id | varchar(50) | NOT NULL |
| 3. | ahli\_id | varchar(50) | NOT NULL |
| 4. | id\_jadwal\_praktik | varchar(50) | NOT NULL |
| 5. | nomor\_antrian | tinyint(4) | NOT NULL |
| 6. | status | enum | NOT NULL |
| 7. | tanggal | date | NOT NULL |

Tabel Produk

Tabel produk digunakan untuk menyimpan data produk yang terdapat di apotek. Adapun penjelasan mengenai tabel produk dapat dilihat pada tabel 3.21.

Tabel 3.21 Tabel Produk

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1. | id\_produk | varchar(50) | NOT NULL |
| 2. | kode\_produk | varchar(50) | NOT NULL |
| 3. | id\_owner\_apotek | varchar(50) | NOT NULL |
| 4. | id\_profil\_apotek | varchar(50) | NOT NULL |
| 5. | Nama\_produk | Varchar(50) | NOT NULL |
| 6. | deskripsi | text | NOT NULL |
| 7. | harga | Double | NOT NULL |
| 8. | Foto | Varchar(255) | DEFAULT NULL |

Tabel Users

Tabel ini digunakan untuk menyimpan data pengguna aplikasi. Adapun penjelasan mengenai tabel user dapat dilihat pada tabel 3.22.

Tabel 3.22 Tabel Users

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1. | id\_produk | varchar(50) | NOT NULL |
| 2. | kode\_produk | varchar(50) | NOT NULL |
| 3. | id\_owner\_apotek | varchar(50) | NOT NULL |
| 4. | id\_profil\_apotek | varchar(50) | NOT NULL |
| 5. | nama\_produk | varchar(50) | NOT NULL |
| 6. | deskripsi | text | NOT NULL |
| 7. | harga | double | NOT NULL |
| 8. | foto | varchar(255) | DEFAULT NULL |

## Perancangan Desain Tampilan Antarmuka

Dalam Rancangan Antarmuka membuat aplikasi *mobile* untuk aplikasi pelayanan kesehatan dan konsultasi masyarakat ini dibuat dengan menggunakan *software* Figma. Beberapa rancangan antarmuka dibagi menjadi beberapa bagian, yaitu: rancangan antarmuka *guest, member*, dokter, dan perawat. Sebagai berikut adalah rancangan antarmuka:

### Rancangan Antarmuka

Rancangan antarmuka aplikasi pelayanan kesehatan dan konsultasi masyarakat ini merupakan rancangan *user interface* dari sisi *member.* Berikut adalah rancangan antarmuka *member*:

1. Rancangan antarmuka halaman Intro dan Register

Rancangan antarmuka ini merupakan halaman intro dan register. Halaman intro merupakan halaman awal ketika buka aplikasi. Ketika pengguna sudah memiliki akun, dapat memilih tombol masuk, jika belum dapat melakukan register. Guest diwajibkan untuk melengkapi form untuk mendapatkan hak akses login. Adapun rancangan antarmuka halaman intro dan register dapat dilihat pada gambar 3.10.

1. Rancangan Antarmuka Halaman Home dan Detail Artikel

Antarmuka ini digunakan untuk menampilkan halaman utama aplikasi. Terdapat konten *chat* ahli, buat janji, resevasi, dan toko sekitar. Menu chat ahli digunakan untuk melakukan konsultasi chat dengan dokter dan perawat. Menu buat janji digunakan untuk membuat janji kunjungan rumah sakit, dan menu toko sekitar merupakan menu untuk melakukan pemesanan produk obat dan vitamin. Member atau Guest dapat membaca artikel melalui artikel yang tersedia dalam halaman home. Berikut dapat dilihat pada gambar 10.

1. Rancangan Antarmuka Chat Dokter dan Perawat

Rancangan antarmuka ini digunakan *member* untuk melakukan konsultasi kesehatan dengan dokter atau perawat. Terdapat daftar dokter juga perawat dan kolom *input* untuk mencari keahlianya. Member dapat mengetikkan keluhan yang dirasakan seperti “pusing” di kolom pencarian, kemudian sistem akan menampilkan daftar dokter dan perawat yang memiliki keahlian dalam menangani kondisi tersebut. untuk melakukan konsultasi kesehatan dengan perawat. Terdapat daftar perawat dan kolom *input* untuk mencari keahlian perawat. Pengguna dapat mencari dokter ataupun berdasarkan keahliannya.

*Member* baru dapat melakukan konsultasi secara gratis selama 30 menit. Sesi konsultasi ini bebas dapat dengan dokter atau perawat. Setelah waktu berakhir, untuk melanjutkan konsultasi maka diperlukan membayar sesi konsultasi. Adapun rancangan antarmuka menu *chat* dengan perawat dapat dilihat pada gambar 3.12.

1. Rancangan Antarmuka Menu Rumah Sakit

Rancangan antarmuka rumah sakit menampilkan daftar rumah sakit terdekat. Aplikasi pelayanan kesehatan dan konsultasi masyarakat berbasis *mobile* menggunakan react native ini terdapat informasi klinis berupa rumah sakit dan apotek terdekat. Daftar tersebut dalam bentuk list menampilkan berapa jauh jarak untuk menuju pusat pelayanan klinis.

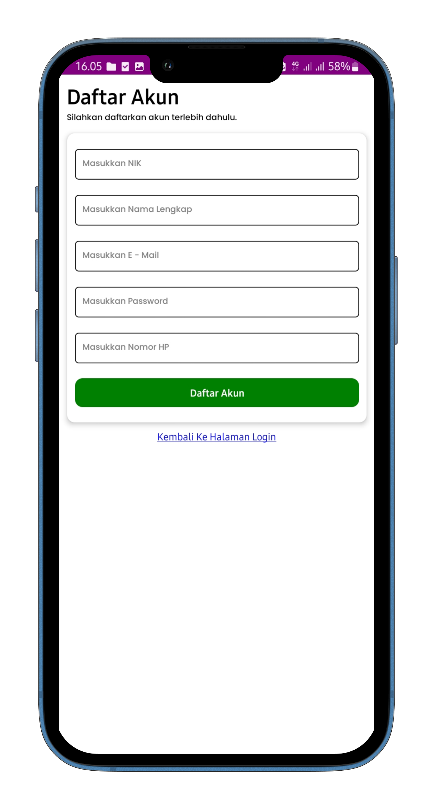
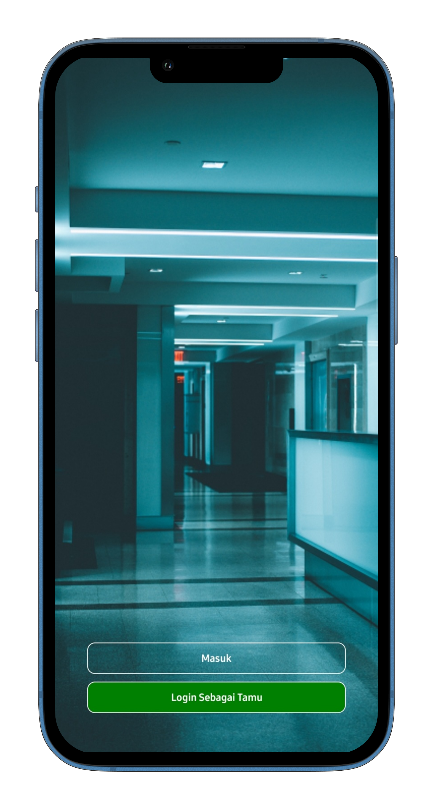
Member dapat melihat lebih detail pelayanan klinis dengan menekan salah satu rumah sakit yang ingin dituju. Detail pelayanan berisi seperti deskripsi rumah sakit atau apotek. Ketika memilih salah satu rumah sakit dapat menampilkan daftar spesialis yang tersedia pada rumah sakit tersebut.

1. Rancangan Antarmuka Toko Kesehatan

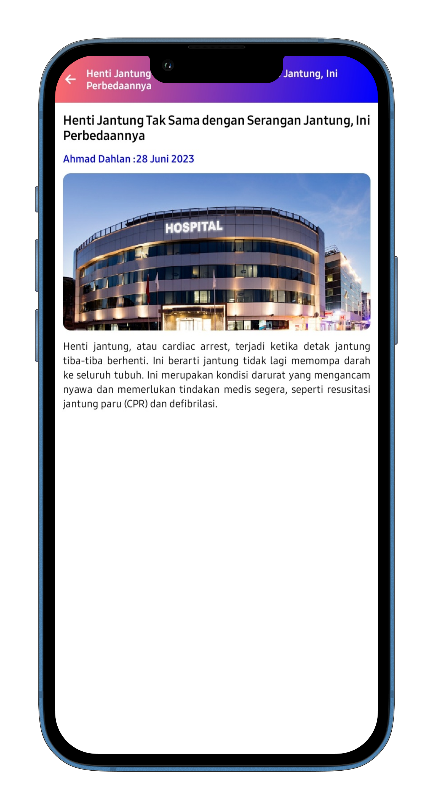
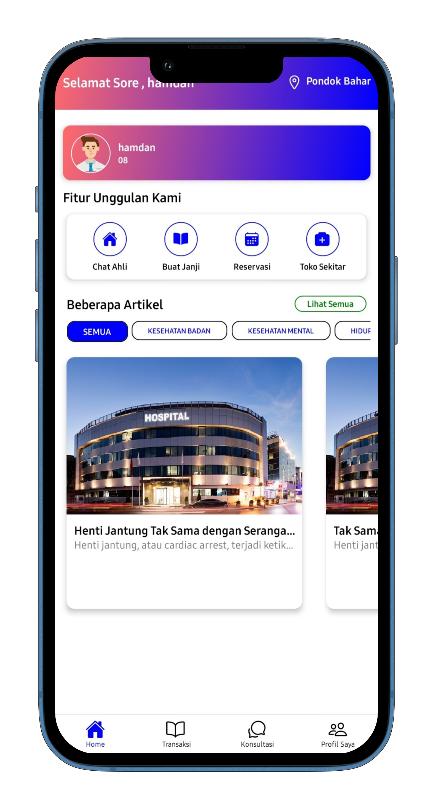
Menu ini digunakan untuk member membeli produk kesehatan. Member dapat membeli berbagai macam produk kesehatan yang tersedia di dalam apotek. Member dapat memasukkan produk ke dalam keranjang sebelum melakukan checkout produk. Toko kesehatan yang ada dalam aplikasi pelayanan kesehatan dan konsultasi masyarakat berbasis *mobile* memfasilitasi member mencari produk berdasarkan kategori. Adapun rancangan antarmuka toko kesehatan dapat dilihat pada gambar 3.13.

1. Rancangan Antarmuka Pembayaran Produk dan Profil

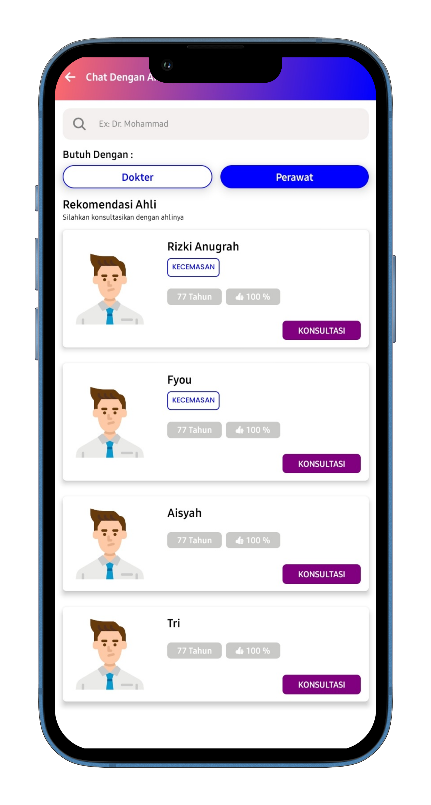
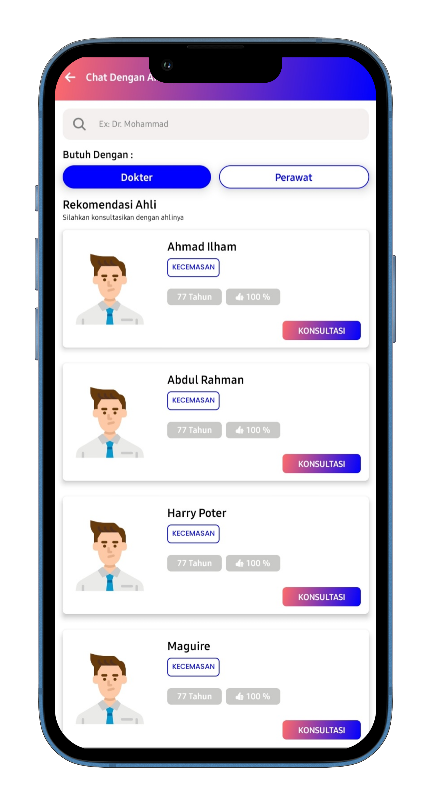
Rancangan antarmuka ini menampilkan rincian dari pembelanjaan, member dapat memilih pembayaran yang disediakan oleh *payment gateway.* Member dapat melakukan perubahan profil melalui menu profil.



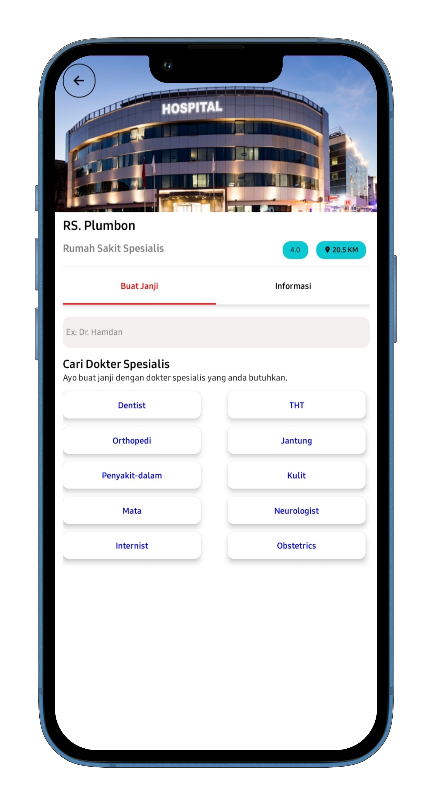
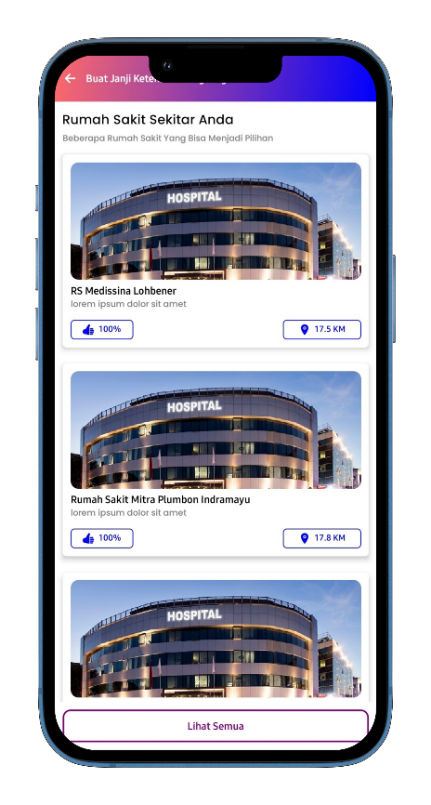
Gambar 3.10 Rancangan Antarmuka Intro dan Register



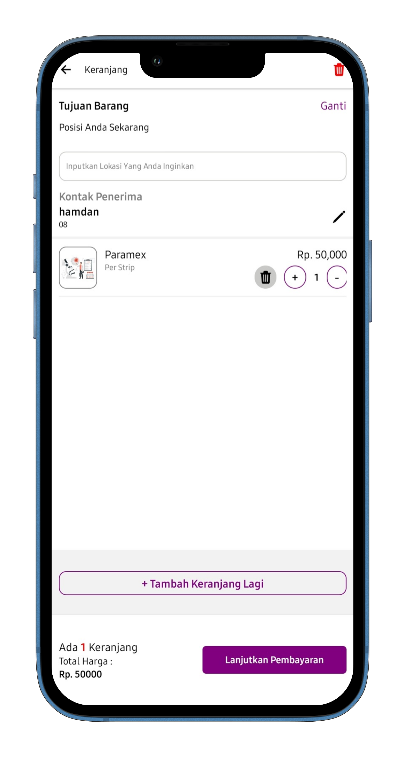
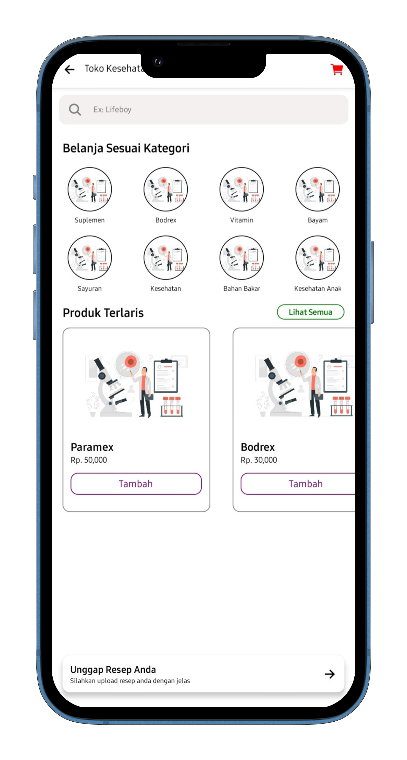
Gambar 3.11 Rancangan Antarmuka Halaman Home dan Detail Artikel



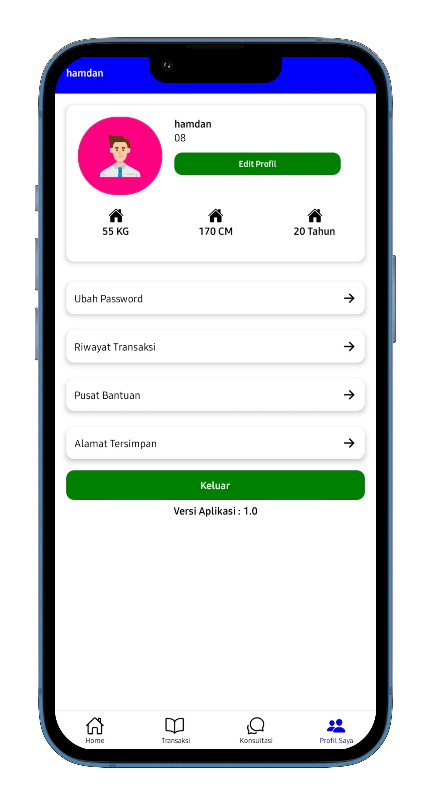
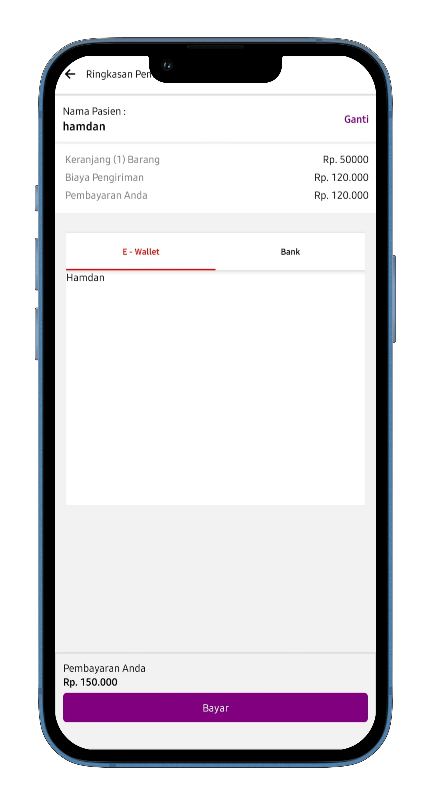
Gambar 3.12 Rancangan Antarmuka Chat Dokter dan Perawat



Gambar 3.13 Rancangan Antarmuka Menu Rumah Sakit



Gambar 3.14 Rancangan Antarmuka Toko Kesehatan



Gambar 3.15 Rancangan Antarmuka Pembayaran Produk dan Profil

# 

# **BAB IV HASIL DAN PEMBAHASAN**

## Hasil

Penelitian ini menghasilkan sebuah aplikasi pelayanan kesehatan dan konsultasi masyarakat berbasis *mobile* menggunakan react native. Aplikasi ini digunakan untuk menunjang pelayanan kesehatan bagi pengguna. Terdapat solusi yang dihadirkan dalam mengatasi permasalahan pada sektor kesehatan. Penulis menghadirkan fitur seperti informasi klinis berupa rumah sakit dan apotek terdekat, konsultasi melalui *chat* dengan dokter dan perawat, jasa perawatan kesehatan oleh perawat terdekat, dan membuat antrian untuk kunjungan temu dokter di rumah sakit. Adapun fitur lainnya penulis bahas dalam bagian pembahasan.

Penulis menggunakan teorema haversine untuk memberikan hasil lokasi terdekat berdasarkan lokasi keberadaan pengguna. Haversine adalah perhitungan jarak antara dua titik berdasarkan koordinat lintang dan bujur pada permukaan bumi. Hasil dari penerapan teorema tersebut digunakan untuk menentukan rumah sakit dan apotek terdekat, juga perawat terdekat untuk melakukan jasa perawatan kepada masyarakat yang membutuhkan penanganan yang tepat dan tanggap.

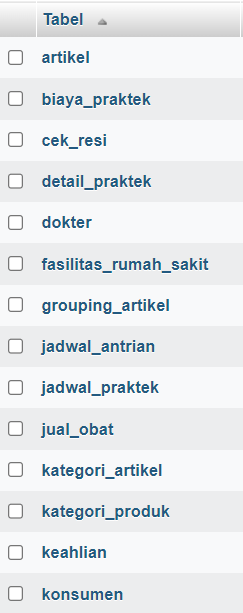
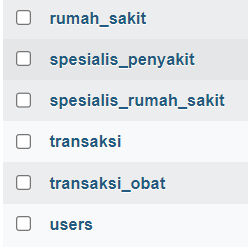
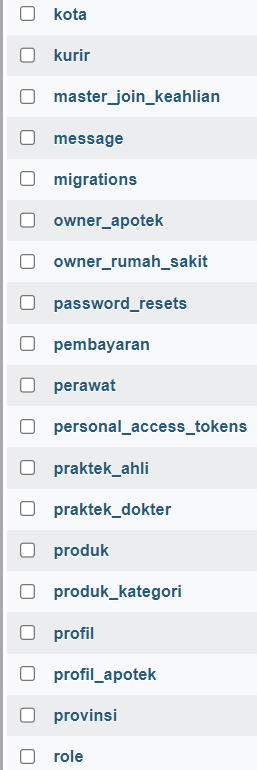
Dapat disimpulkan bahwa aplikasi pelayanan kesehatan dan konsultasi masyarakat berbasis *mobile* merupakan aplikasi yang dibangun untuk menunjang aktivitas sektor kesehatan dalam meningkatkan pelayanan kesehatan kepada masyarakat. Salah satu upaya dalam meningkatkan pelayanan kesehatan yaitu berupa konsultasi kesehatan dengan dokter dan perawat yang dapat memangkas waktu, karena dengan konsultasi kesehatan melalui *chat,* masyarakat tidak perlu mengunjungi dokter secara *offline* dan jasa perawatan kesehatan oleh perawat terdekat.

## Pembahasan

Tahapan ini membahas tahapan implementasi analisis kebutuhan sistem, perancangan sistem, perancangan basis data, serta penerapan metodologi agile scrum. Pada tahapan ini menjelaskan tentang kesesuaian antara rancangan dengan hasil implementasi, Dan, pada bagian ini juga membahas mengenasi pengujian modul *product backlog* dan *testing endpoint.*

### Implementasi Basis Data

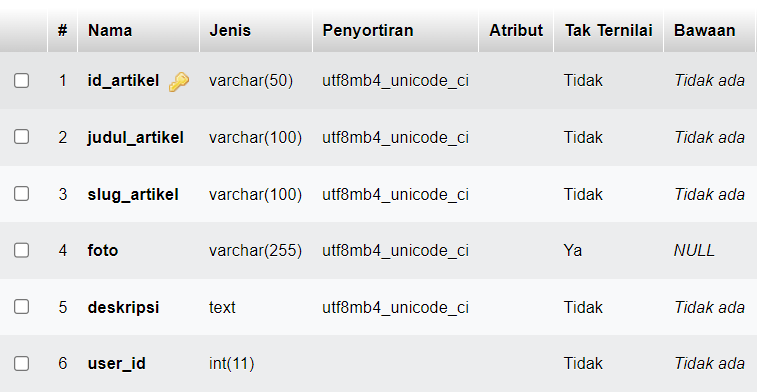
Basis data yang digunakan dalam membangun aplikasi pelayanan kesehatan dan konsultasi masyarakat berbasis *mobile* adalah berobat\_api. Pada basis data berobat\_api terdapat 39 tabel yang digunakan untuk mengimplementasikan perancangan dan membangun aplikasi ini. Adapun basis data dapat dilihat pada gambar 4.1.



Gambar 4.1 Implementasi Basis Data

1. Implementasi Tabel Artikel

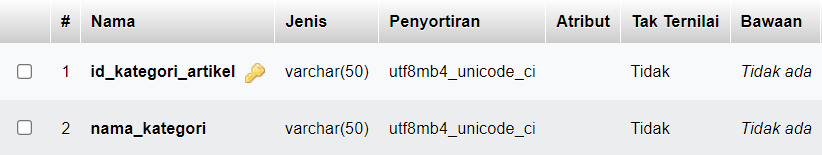
Tabel artikel digunakan untuk menyimpan data artikel yang dikelola oleh dokter dan admin. Adapun struktur tabel dapat dilihat pada gambar 4.2



Gambar 4.2 Implementasi Tabel Artikel

1. Implementasi Tabel Kategori Artikel

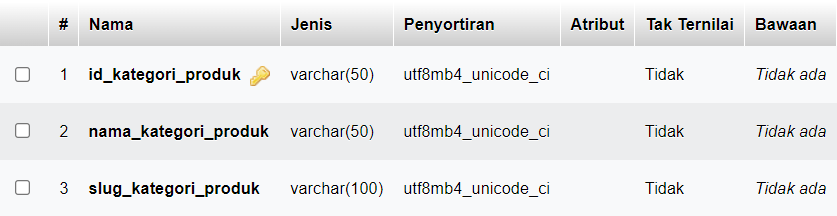
Tabel kategori artikel merupakan tabel untuk mengelola data kategori artikel. Kategori artikel digunakan untuk mengelola artikel. Adapun struktur tabel kategori artikel dapat dilihat pada gambar 4.3



Gambar 4.3 Implementasi Kategori Artikel

1. Implementasi Tabel Kategori Produk

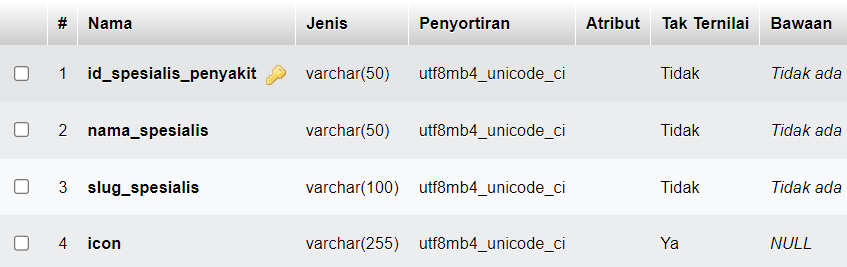
Tabel kategori produk digunakan untuk mengelola data kategori produk. Kategori produk digunakan ketika akan menambahkan sebuah produk, maka perlu menambahkan kategori produk terlebih dahulu.



Gambar 4.4 Implementasi Kategori Produk

1. Implementasi Tabel Spesialis Penyakit

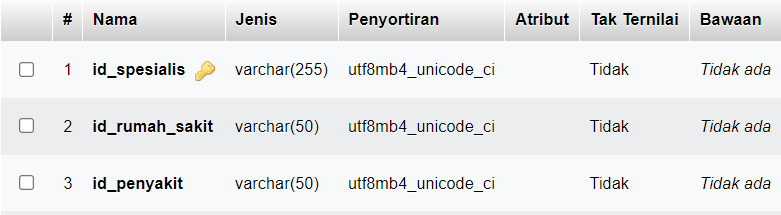
Table spesialis penyakit digunakan untuk menampilkan spesialis yang tersedia dalam aplikasi ini. Adapun struktur tabel spesialis penyakit dapat dilihat pada gambar 4.5.



Gambar 4.5 Implementasi Tabel Spesialis Penyakit

1. Implementasi Tabel Spesialis Rumah Sakit

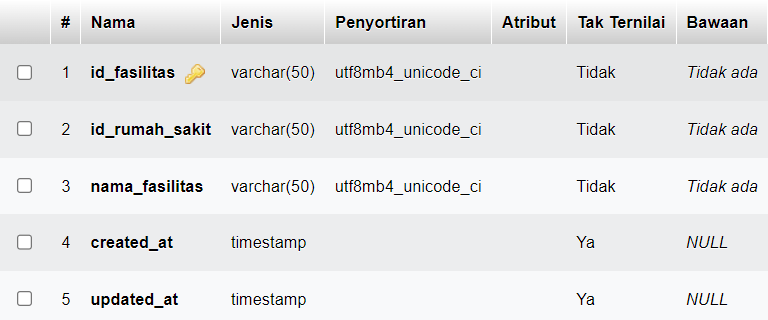
Tabel spesialis rumah sakit digunakan untuk mengelola spesialis penyakit yang tersedia pada rumah sakit. Adapun struktur dapat dilihat pada gambar 4.6.



Gambar 4.6 Implementasi Tabel Spesialis Rumah Sakit

1. Implementasi Tabel Fasilitas Rumah Sakit

Tabel fasilitas rumah sakit digunakan untuk mengelola data fasilitas yang tersedia pada rumah sakit. Adapun struktur tabel dapat dilihat pada gambar 4.7



Gambar 4.7 Implementasi Tabel Fasilitas Rumah Sakit

1. Implementasi Tabel Grouping Artikel

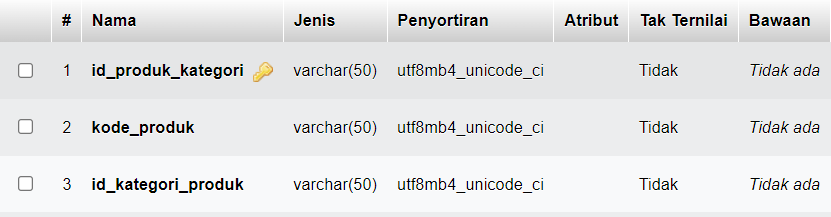
Tabel grouping digunakan untuk mengelompokkan artikel dengan kategori artikel. Adapun struktur tabel grouping artikel dapat dilihat pada gambar 4.8.



Gambar 4.8 Implementasi Tabel Grouping Artikel

1. Implementasi Tabel Produk Kategori

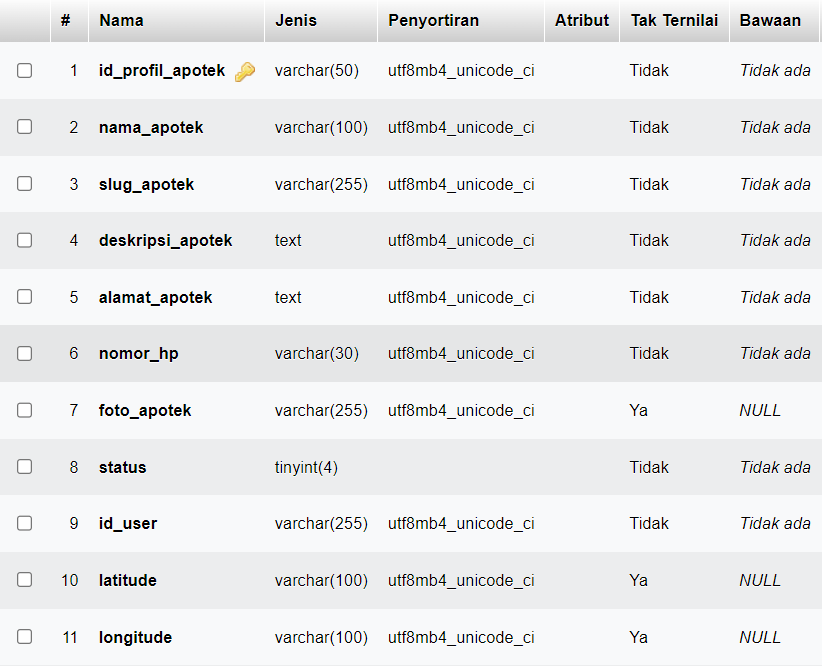
Tabel produk kategori digunakan untuk mengelompokkan produk dengan kategori produk. Adapun tabel produk kategori dapat dilihat pada gambar 4.9



Gambar 4.9 Implementasi Tabel Produk Kategori

1. Implementasi Tabel Profil Apotek

Implementasi tabel profil apotek pada aplikasi pelayanan kesehatan dan konsultasi masyarakat dapat dilihat pada gambar.



Gambar 4.10 Implementasi Tabel Profil Apotek

1. Implementasi Tabel Rumah Sakit

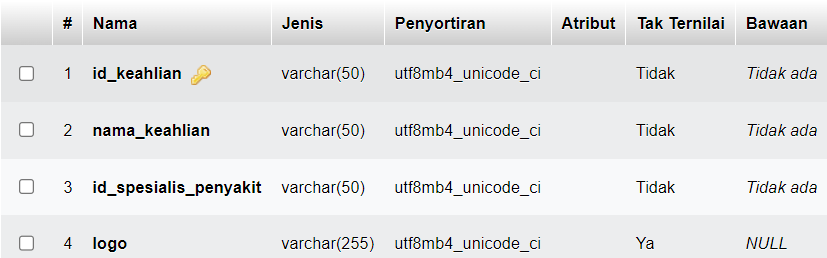
Tabel rumah sakit digunakan untuk mengelola data rumah sakit. Pemilik rumah sakit dapat mengelola rumah sakit. Adapun tabel rumah sakit dapat dilihat pada gambar 4.11.



Gambar 4.11 Implementasi Tabel Rumah Sakit

1. Implementasi Tabel Keahlian

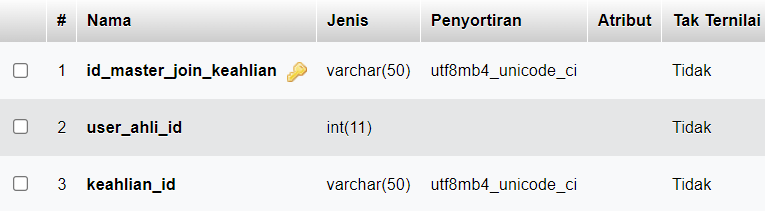
Tabel keahlian digunakan untuk mengelola keahlian yang dimiliki oleh dokter dan perawat. Adapun struktur tabel keahlian dapat dilihat pada gambar 4.12.



Gambar 4.12 Implementasi Tabel Keahlian

1. Implementasi Tabel Join Keahlian

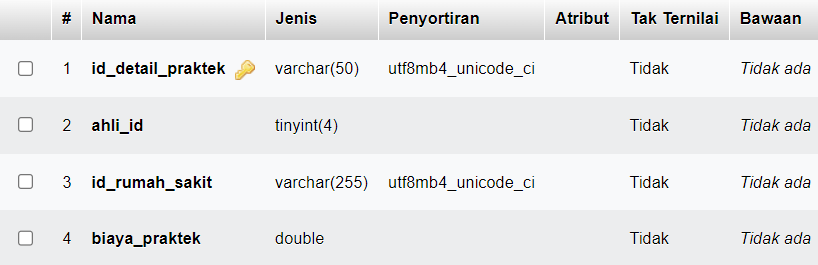
Tabel join keahlian digunakan untuk mengelompokkan antara dokter atau perawat dengan nama keahliannya. Adapun struktur tabel join keahlian dapat dilihat pada gambar 4.13.



Gambar 4.13 Implementasi Tabel Join Keahlian

1. Implementasi Tabel Detail Praktik

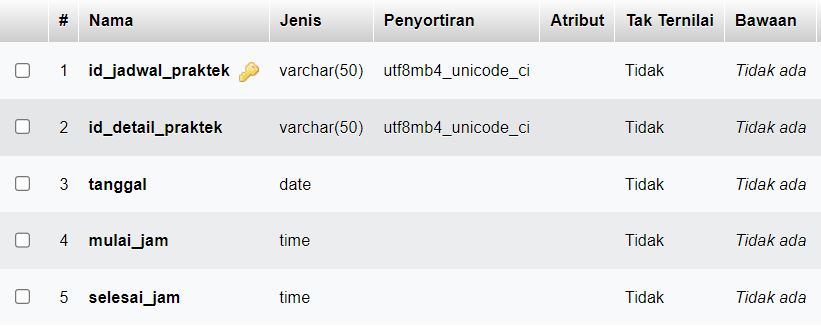
Tabel detail praktik merupakan tabel untuk melihat detail dari dokter yang melakukan praktik di rumah sakit. Adapun struktur tabel detail praktik dapat dilihat pada gambar 3.14.



Gambar 4.14 Implementasi Tabel Detail Praktik

1. Implementasi Tabel Jadwal Praktik

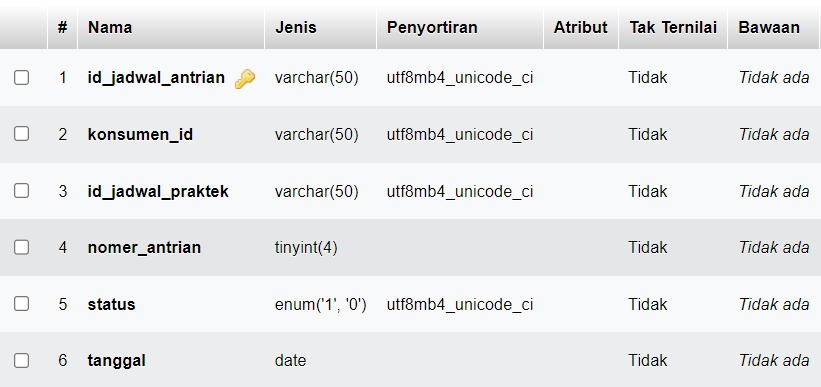
Tabel jadwal praktik merupakan tabel yang digunakan untuk mengelola jadwal praktik dokter di rumah sakit. Adapun tabel jadwal praktik pada gambar 3.15.



Gambar 4.15 Implementasi Tabel Jadwal Praktik

1. Implementasi Tabel Jadwal Antrian

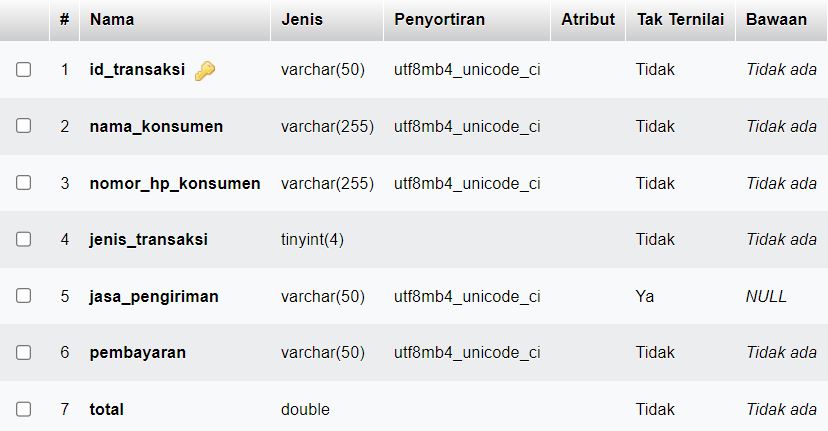
Tabel jadwal antrian merupakan tabel untuk member dalam membuat antrian kunjungan temu dokter di rumah sakit. Adapun tabel jadwal antrian dapat dilihat pada gambar 3.16.



Gambar 4.16 Implementasi Tabel Jadwal Antrian

1. Implementasi Tabel Transaksi

Implementasi tabel keahlian pada aplikasi pelayanan kesehatan dan konsultasi masyarakat dapat dilihat pada gambar.



Gambar 4.17 Implementasi Tabel Transaksi

1. Implementasi Tabel Produk

Tabel produk digunakan untuk mengelola data produk di setiap apotek. Pemilik apotek dapat mengelola data produk yang tersedia di apotek.



Gambar 4.18 Implementasi Tabel Produk

1. Implementasi Tabel Users

Tabel users merupakan tabel dari pengguna yang menampung seluruh data akun. Adapun tabel users dapat dilihat pada gambar 3.19



Gambar 4.19 Implementasi Tabel Users

### *Sprint Review Product Backlog*

Dalam pembahasan ini, penulis akan menjelaskan hasil dari pengerjaan *sprint.* Pemaparan sprint review merupakan bagian dari tahapan dalam pengerjaan aplikasi pelayanan kesehatan dan konsultasi masyarakat berbasis *mobile* menggunakan react native. Adapun sprint review yang dapat dilihat sebagai berikut.

Sprint Review Modul Login dan Registrasi Akun

Modul ini digunakan oleh member, guest, dokter, dan perawat. Setiap jenis akun terdapat perbedaan fitur di dalemnya. Sebelum mengakses fitur-fitur di dalamnya, guest dapat melakukan registrasi akun. Tujuan dari registrasi akun adalah untuk mendapatkan hak akses penggunaan fitur. Guest dapat mengakses aplikasi tanpa harus melakukan registrasi, namun terdapat keterbatasan dalam mengakses fitur.

Dokter dan Perawat dapat melakukan registrasi akun melalui aplikasi dalam versi *mobile*. Untuk versi *mobile* dokter dan perawat langsung diarahkan ke halaman login. Dokter dan perawat dapat melakukan sesi konsultasi kesehatan ketika sudah melakukan login. Adapun hasil dari sprint review dapat dilihat pada gambar 4.20.

Sprint Review Modul Transaksi Produk

Member dapat melakukan pembelian produk kesehatan dengan modul transaksi produk. Member dapat mencari produk sesuai kebutuhan dan kategorinya. Terdapat beragam jenis kategori produk dan produk kesehatan. Produk kesehatan yang tersedia dalam modul ini dikelola oleh apotek.

Sebelum melakukan pembelian produk, member dapat menampung produk yang akan dibeli melalui menu keranjang. Cukup tekan icon keranjang yang terdapat di sebelah produk, setelahnya produk akan tertampung di halaman keranjang. Member dapat membaca deskripsi produk dalam modul transaksi. Deskripsi produk digunakan untuk menjelaskan informasi mengenai produk tersebut, mencegah kesalahan dalam pembelian.

Member dapat melakukan pembayaran melalui beberapa jenis pembayaran yang disediakan oleh payment gateway. Sebelum melakukan pembayaran, member diwajibkan untuk memasukkan alamat pengiriman. Pembelian produk terdapat 2 jenis, yaitu umum dan atas resep dokter. Untuk resep dokter didapatkan ketika selesai melakukan sesi konsultasi kesehatan, dokter akan menanyakan bahwa member ingin membeli produk secara sendiri atau dicarikan oleh sistem berdasarkan apotek terdekat. Ketika dicarikan oleh sistem, produk yang dibutuhkan akan ditampilkan pada halaman produk. Adapun dapat dilihat pada gambar 4.21.

Sprint Review Modul buat Janji Kunjungan Temu Dokter

Modul ini digunakan oleh member untuk membuat janji kunjungan temu dokter di rumah sakit. Hasil dari buat janji adalah qr code. Qr code digunakan untuk proses validasi data member dari pihak rumah sakit. Member dapat memilih rumah sakit dan dokter yang akan dikunjungi melalui menu rumah sakit.

Modul ini digunakan dengan cara memilih rumah sakit, dokter, dan jadwal praktiknya. Sesudahnya member mendapatkan nomor antrian. Adapun hasil dari sprint modul backlog kunjungan temu dokter dapat dilihat pada gambar 4.23. Dokter dapat mengakhiri sesi kunjungan setelah melakukan konsultasi dengan member secara langsung melalui *smartphone.* Dapat dilihat pada gambar 4.24.

Sprint Review Modul Konsultasi Kesehatan

Modul Backlog ini digunakan untuk fitur konsultasi kesehatan dengan dokter dan perawat melalui *chat.* Member dapat melakukan konsultasi dengan tenaga medis yang kompeten di bidangnya. Untuk memulai sesi konsultasi, pengguna perlu melakukan login ke dalam aplikasi. Setelah login, member dapat memilih menu chat ahli. Di dalam menu chat ahli terdapat daftar dokter dan perawat.

Member dapat mencari dokter dan perawat berdasarkan keahlian atau keluhan yang dirasakan. Ketika member memilih dokter atau perawat, maka akan menampilkan detail data diri dari tenaga medis tersebut. Untuk memulai sesi konsultasi, member dapat menekan tombol chat, kemudian akan diarahkan ke halaman ruang konsultasi. Sesi konsultasi diberi waktu 30 menit bagi member yang baru mendaftar akun, untuk melakukan konsultasi kembali, member diwajibkan untuk membayar sesi konsultasi.

Setelah melakukan sesi konsultasi, member akan diberikan opsi dari dokter, apakah ingin menerima rawat jalan oleh perawat terdekat. Ketika member memilih untuk menerima rawat jalan oleh perawat terdekat, maka sistem akan mencari perawat yang terdekat berdasarkan lokasi pengguna. Adapun sprint review modul konsultasi kesehatan dapat dilihat pada gambar 4..

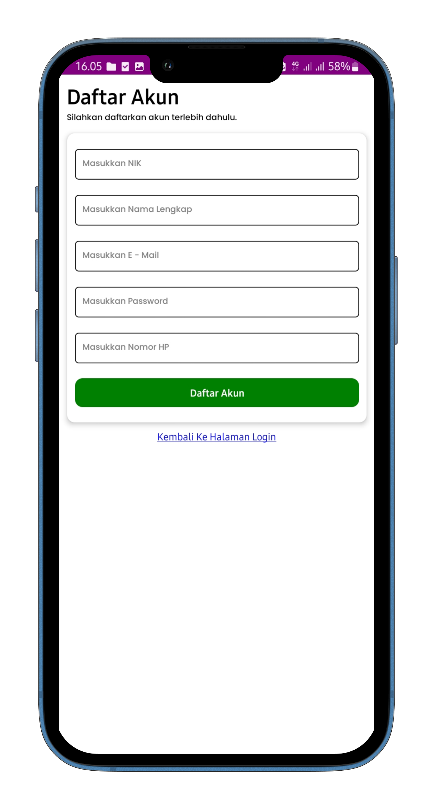
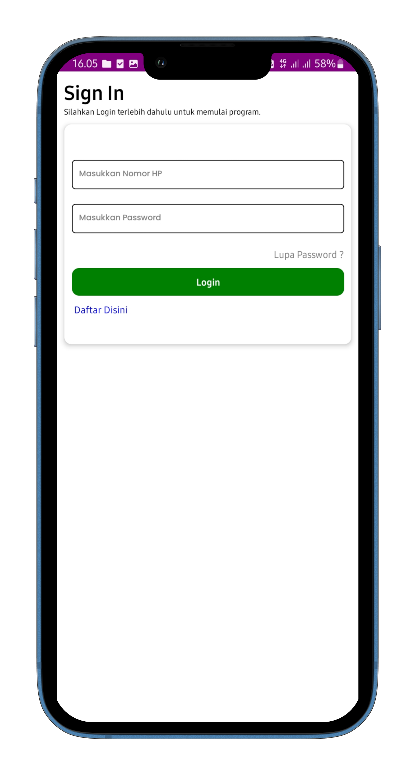
Sprint Review Modul Informasi Pelayanan Klinis

Modul ini digunakan oleh *member* untuk mengetahui tempat pelayanan klinis terdekat berdasarkan lokasi dari member. Informasi pelayanan klinis ini ditujukan kepada pengguna untuk memberikan informasi mengenai tempat pelayanan klinis yang terdekat dan sesuai dengan kondisi kesehatan yang dibutuhkan.

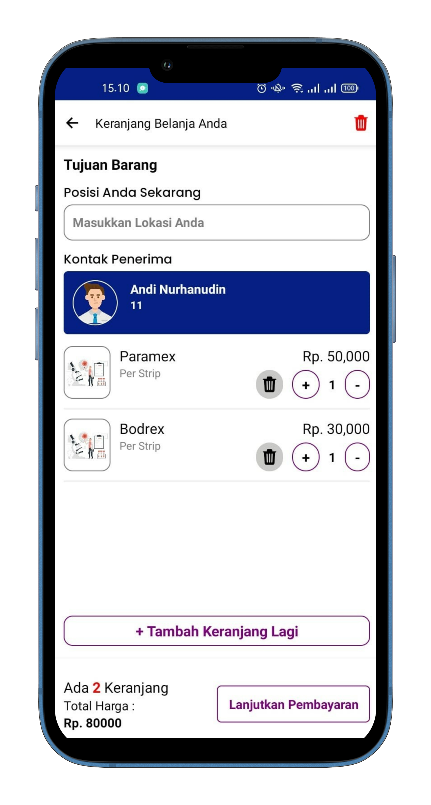
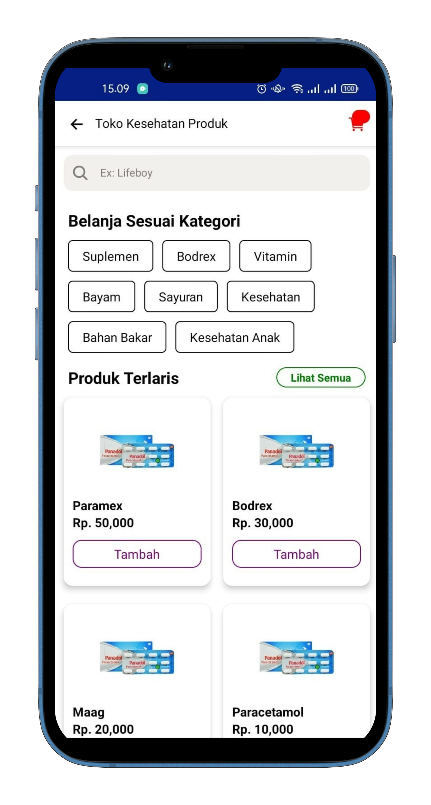
Guest dan Member dapat melihat detail dari rumah sakit atau apotek. Dalam detail informasi tersebut, menampilkan seperti alamat, deskripsi, dokter yang tersedia, dan lain hal. Metode yang digunakan untuk menghasilkan keputusan lokasi terdekat menggunakan teorema haversine. Adapun sprint review dapat dilihat pada gambar.

Sprint Review Modul Search

Modul ini digunakan oleh pengguna untuk mencari dokter dan perawat berdasarkan keahlian. Pengguna difasilitasi untuk melakukan konsultasi berdasarkan keluhan yang dirasakan. Pengguna cukup mengetikkan keluhan yang sedang dirasa di kolom pencarian, kemudian akan menampilkan dokter dan perawat berdasarkan keahlian. Adapun sprint review dapat dilihat pada gambar.



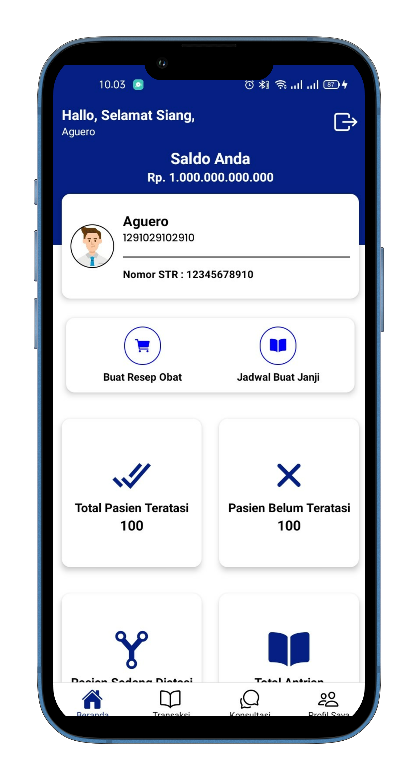
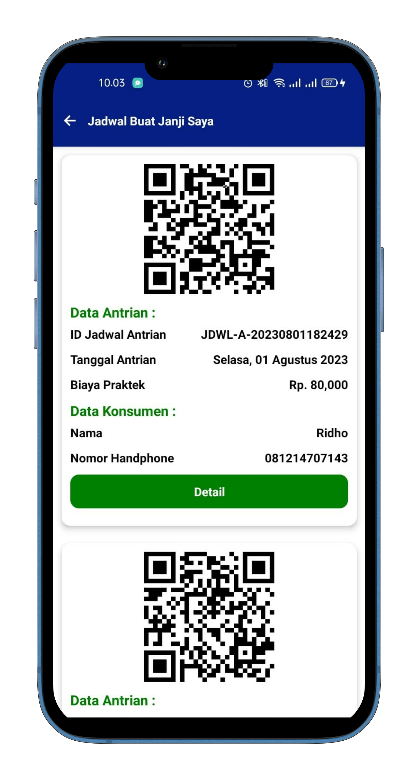
Gambar 4.20 Sprint Review Modul Login dan Register



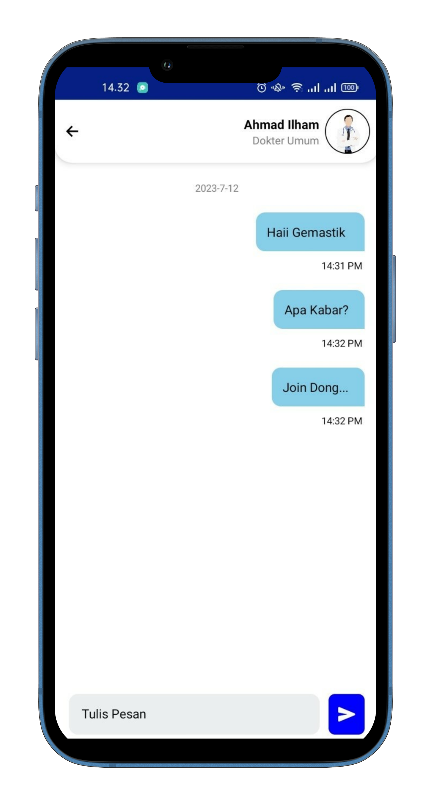
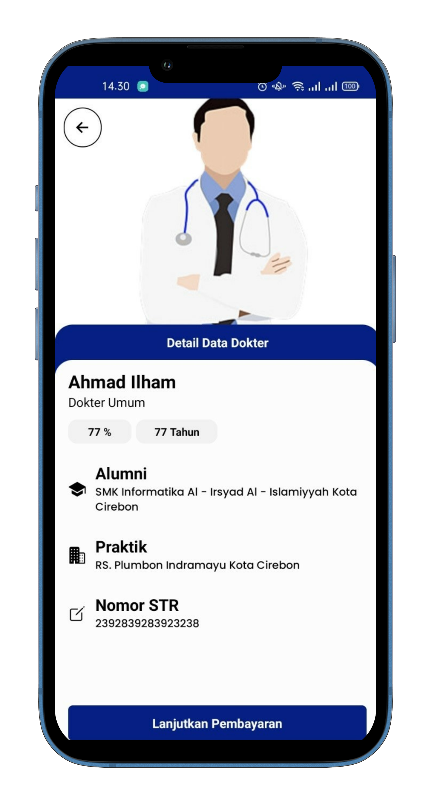
Gambar 4.21 Sprint Review Modul Transaksi Produk



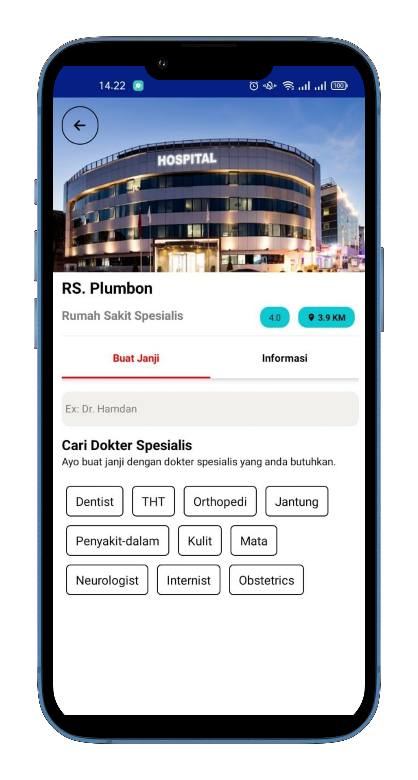
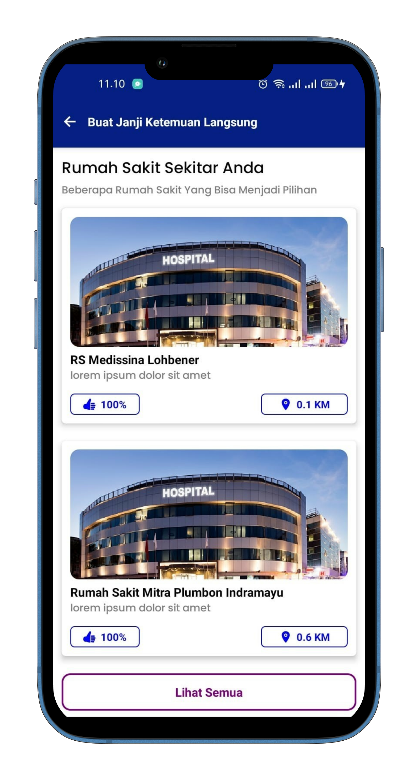
Gambar 4.22 Sprint Review Modul Buat Janji Dokter

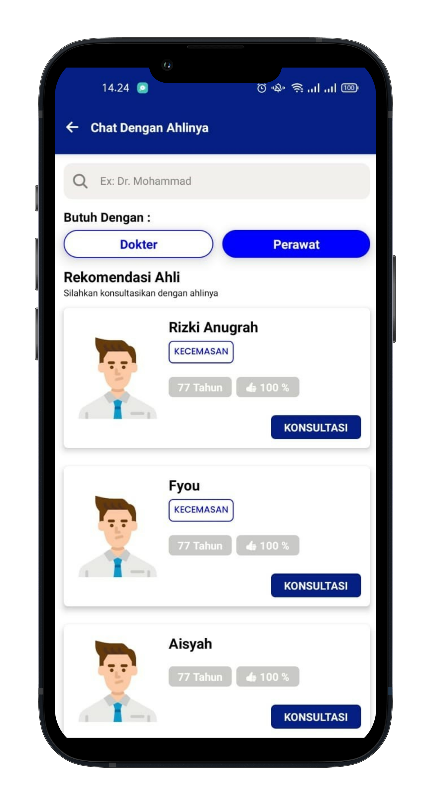
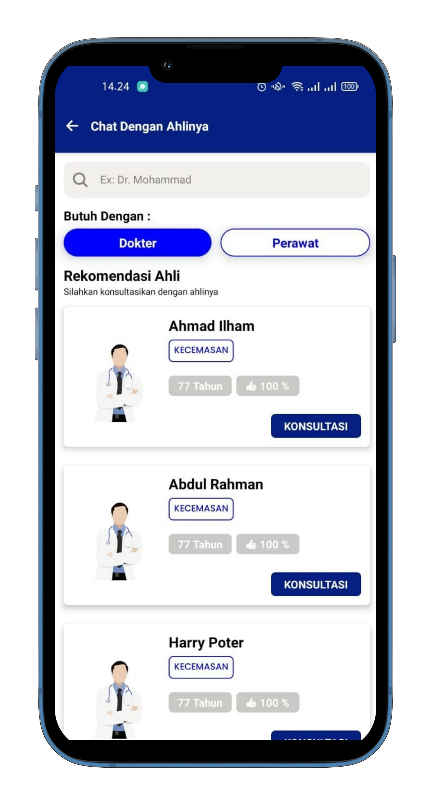
Gambar 4.23 Sprint Review Modul Buat Janji dari Sisi Dokter



Gambar 4.24 Sprint Review Modul Konsultasi Kesehatan



Gambar 4.25 Sprint Review Modul Infromasi Pelayanan Klinis



Gambar 4.26 Sprint Review Modul Search

### *Daily Activity*

*Daily Activity* merupakan bagian dari tahapan agile scrum. Pada tahap ini, anggota tim pengembang, memaparkan progress pengerjaan sprint kepada tim. Anggota tim akan menjelaskan kendala yang terjadi selama pengerjaan aplikasi. Anggota tim juga menyampaikan hasil pengerjaan sprint dari setiap modul product backlog. Berikut kendala yang penulis hadapi dalam membangun aplikasi ini.

* + - 1. Analisis kebutuhan

Pada *activity* ini penulis melakukan pembedahan kebutuhan yang digunakan dalam membangun aplikasi ini. Pembedahan berupa permasalahan yang akan diselesaikan.

* + - 1. Penentuan *tools*

Kendala yang dihadapkan oleh penulis adalah penentuan *tools* seperti Bahasa pemrograman yang efektif dalam membangun aplikasi ini. Setelah melakukan diskusi, penulis memilih javascript dengan react native untuk pengerjaan aplikasi pelayanan kesehatan dan konsultasi masyarakat berbasis *mobile.*

* + - 1. Penentuan Hak Akses

Setiap akun memiliki perbedaan hak akses, karena kepentingan dalam aplikasi memiliki tujuannya masing-masing. Pada tahap ini penulis terkendala dalam membagi hak akses akun. Adapun daily activity dapat dilihat pada tabel berikut.

Tabel 4.1 Daily Activity

|  |  |
| --- | --- |
| Daily Activity | Kendala |
| Daily Activity 1 | Penentuan tools untuk aplikasi |
| Daily Activity 2 | Analisis Kebutuhan |
| Daily Activity 3 | Penentuan Hak Akses |
| Daily Activity 4 | Pembatasan Fitur |
| Daily Activity 5 | Pembayaran Produk dan Jasa Konsultasi |
| Daily Activity 6 | Teknik Penentuan Jarak Terdekat |

### *Testing* *Endpoint* API

Dalam pembahasan ini, penulis telah melakukan pengujian pada endpoint API. Endpoint tersebut terdapat dalam sebuah aplikasi pelayanan kesehatan dan konsultasi masyarakat berbasis *mobile*. Fungsi endpoint ini dibutuhkan dalam menunjang pengerjaan berbagai fitur yang ada di dalam aplikasi tersebut. Secara keseluruhan, aplikasi ini memiliki total 80 endpoint API aktif yang beragam.

Dalam konteks ini, pembahasan lebih difokuskan pada pengujian endpoint utama yang menjadi bagian dari pengerjaan product backlog aplikasi. Sebanyak 26 endpoint utama telah diuji. Proses pengujian endpoint API ini melibatkan berbagai tahap penting, termasuk verifikasi keberfungsiannya, identifikasi dan penanganan potensi masalah atau bug, serta memastikan bahwa setiap endpoint beroperasi dengan baik sesuai dengan kebutuhan fungsional dan spesifikasi yang telah ditetapkan.

Hasil dari pengujian endpoint ini memiliki dampak signifikan dalam memastikan kualitas keseluruhan dari aplikasi pelayanan kesehatan dan konsultasi masyarakat ini. Dengan menjamin bahwa endpoint berfungsi dengan baik, 26 endpoint ini memiliki status kode 200 yang memiliki arti bahwa endpoint siap untuk digunakan. Adapun testing endpoint API dapat dilihat pada lampiran.

### *Testing* Modul *Product Backlog*

Pada pembahasan ini, penulis akan memaparkan kesimpulan hasil dari pengujian aplikasi pelayanan kesehatan dan konsultasi masyarakat berbasis *mobile* menggunakan react native. Pengujian dilakukan dengan dua *test* *case* yaitu negatif dan positif. *Test case* positif merupakan skenario pengujian aplikasi yang dilaksanakan untuk memperoleh hasil yang diinginkan dengan kondisi valid. Sedangkan kondisi *test case* negatif dirancang untuk menguji kemampuan sistem dalam menangani kondisi yang tidak valid.

Pengujian *test case* positif meliputi pengujian modul product backlog, terdapat tujuh product backlog dengan masing-masing *test case.* Pengujian aplikasi ini dilakukan oleh ….. Test case positif dan negatif memperoleh hasil yang sebagian besar adalah sesuai rencana, yaitu product backlog yang diuji dapat digunakan oleh pengguna. Namun terdapat test case yang menunjukkan hasil *not passed.*

Hasil *test case* yang tidak lolos pengujian menunjukkan bahwa terdapat bug atau keterbatasan yang dilakukan oleh aplikasi ini. Bug dan keterbatasan ini dapat dijadikan sebagai bahan evaluasi penulis untuk meningkatkan kinerja aplikasi pelayanan kesehatan dan konsultasi masyarakat berbasis *mobile* agar aplikasi dapat digunakan oleh masyarakat sebagai hal yang menunjang kebutuhan kesehatannya.

# 

# **BAB V PENUTUP**

## Kesimpulan

Berdasarkan hasil dari pengerjaan *product backlog*, *sprint planning*, *sprint review*, dan pengujian aplikasi pelayanan kesehatan dan konsultasi masyarakat berbasis *mobile* menggunakan react native maka dapat ditarik kesimpulan sebagai berikut.

1. Masyarakat dapat mengetahui informasi terkait kebutuhan kesehatannya dan lokasi pelayanan kesehatan terdekat.
2. Melalui aplikasi ini, Masyarakat dapat membeli produk kesehatan yang dibutuhkan.
3. Masyarakat dapat membuat janji kunjungan temu dokter di rumah sakit tanpa perlu mendaftar kunjungan secara *offline* di rumah sakit.
4. Melalui aplikasi ini, Masyarakat dapat melakukan konsultasi kesehatan dengan dokter dan perawat melalui *chat* menggunakan *smartphone.*
5. Masyarakat dapat menerima perawatan kesehatan oleh perawat terdekat.

## Saran

Aplikasi ini belum sepenuhnya sempurna, oleh karena itu dipaparkan beberapa saran untuk pengembangan selanjutnya agar dapat dipertimbangkan, sehingga aplikasi dapat digunakan dengan manfaat yang lebih luas. Saran-saran yang diberikan yaitu sebagai berikut:

1. Dapat dibuat notifikasi saat pesan konsultasi masuk.
2. Memperbaiki *user interface* dan *user experience* aplikasi.

# **DAFTAR PUSTAKA**

Abidin. (2006). *Penentuan Posisi GPS dan Aplikasinya .* Jakarta: PT. Pradnya Paramita.

Akbar, M. A. (2019). *Buku Ajar Konsep-Konsep Dasar Dalam Keperawatan Komunitas.* Sleman: DEEPUBLISH.

Asmadi. (2008). *Konsep Dasar Keperawatan.* Jakarta: EGC.

Djuwanto, T. (2012). Peran Dokter Spesialis Dalam Pelayanan Kesehatan Masyarakat. *IDI*, 4.

Fahribi, M. A. (2017). Sistem Pencarian Jarak Terdekat Menggunakan Metode Haversine berbasis Android. *Teknik Informatika*, 3-4.

Fahruroji, A. (2013). Sarana Bantu Dokter Keliling Studi Lokasi Desa Mata Gara. *Inosains*, 38.

Kusnanto. (2004). *Pengantar Profesi dan Praktik Keperawatan Profesional.* Jakarta: EGC.

Rianto, I., & Pratasik , S. (2020). Perancangan Aplikasi E-DUK Dalam Pengelolaan SDM Menggunakan Metode Agile Development. *CogITo Smart*, 204-216.

Yunandar, R. T. (2020). Aplikasi Manajemen Surat Masuk dan Surat Keluar berbasis Mobile di Perum Bulog Subdrive. *Jiska*, 66-90.

Menkes. (2019). Peraturan Menteri Kesehatan RI Nomor 20 Tahun 2019 tentang Penyelenggaraan Pelayanan Telemedicine Antar Fasilitas Pelayanan Kesehatan.

Aditya, Renanta. (2020). *Pembangunan Aplikasi Area Traffic Control System Menggunakan GPS dan CCTV berbasis Android di Kota Bandung.* Skripsi S-1. Bandung: Universitas Komputer Indonesia.

Wowiling, Marvel Christevan. (2020). *Pembangunan Aplikasi Peminjaman Buku Pada Wilayah Sekitar Pengguna*. Skripsi-1: Universitas Komputer Indonesia.

Pratama, Aditya. (2019). Belajar-Uml-Diagram. https://www.codepolitan.com/belajar-uml-sequence-diagram. Diakses pada tanggal 6 Juli 2023.

Wise. (2023). Tentang OpenStreetMap. https://openstreetmap.or.id/tentang-openstreetmap/. Diakses pada tanggal 6 Juli 2023.

Dewi, Ni Putu. (2021). *Gambaran Tingkat Pengetahuan Tentang Cara Menyikat Gigi pada Kepala Keluarga di Wilayah KKN IPE Kabupaten Jembrana 1 Tahun 2021.* Diploma. Denpasar: Politeknik Negeri Kesehatan.

Farahsati, Winda, Studi Independen, and Studi Independen. (2022). Apa Itu Agile? - Vocasia. Vocasia. <https://vocasia.id/blog/apa-itu-agile/>. Diakses pada tanggal 10 Juli 2023.

Ependi, U. (2012). Pengembangan E-Trace Alumni Dengan Menggunakan Pendekatan Metode Agile. *Scopus*, D237-D244.

Sudomo, Manurung. (2018). Mengenal Leaflet Js, Alternatif Membangun Peta Interaktif berbasis Web tanpa Google Maps. https://www.sgtgeomedia.com/ detailpost/mengenal-leaflet-js-alternatif-membangun-peta-interaktif-berbasis-web-tanpa-google-maps-api. Diakses pada tanggal 10 Juli 2023.

# **LAMPIRAN**

|  |
| --- |
| **LAMPIRAN 1. KODE PROGRAM** |

**Konsultasi Chat**

|  |
| --- |
| import React, { useState, useEffect } from 'react';  import { View, Text, Image, TouchableOpacity, ScrollView } from 'react-native';  import StatusBarComponent from '../../../components/StatusBar/StatusBarComponent';  import ChatItem from '../../../components/ChatItem';  import InputChat from '../../../components/InputChat';  import { configfirebase } from '../../../firebase/firebaseConfig';  import {  getChatTime,  getData,  setDateChat,  useForm,  } from '../../../utils';  import { useTime } from '../../../components/Time/TimeContext';  import Navigasi from '../../../partials/navigasi';  import { useNavigation } from '@react-navigation/native';  import AsyncStorage from '@react-native-async-storage/async-storage';  const Chating = ({route }) => {  const getDokter = route.params;  const navigation = useNavigation();  const [form, setForm] = useForm({  id\_dokter: '',  uid\_partner: '',  });  const { countdown, finished, formatTime, setCurrentPerUser } = useTime();  const durasiKonsultasi = 1800;  const isKonsultasiTerlewat = countdown <= 0;  const [chatContent, setChatContent] = useState('');  const [user, setUser] = useState({});  const [dataPribadi, setDataPribadi] = useState({});  const [chatData, setChatData] = useState([]);  useEffect(() => {  getDataUserLocal();  if (isKonsultasiTerlewat) {  AsyncStorage.removeItem("countdown");  }  const urlFirebase = `chatting/${user.uid}\_${getDokter.data.uid}/allChat/`;  configfirebase.database()  .ref(urlFirebase)  .on('value', snapshot => {  console.log(snapshot.val());  if (snapshot.val()) {  const dataSnapshot = snapshot.val();  const semuaDataChat = [];  Object.keys(dataSnapshot).map(key => {  const dataChat = dataSnapshot[key];  const newDataChat = [];  Object.keys(dataChat).map(itemChat => {  newDataChat.push({  id: itemChat,  data: dataChat[itemChat],  });  });  semuaDataChat.push({  id: key,  data: newDataChat,  });  });  setChatData(semuaDataChat);  }  });    }, [  user.uid,  dataPribadi.idx,  dataPribadi.token,  isKonsultasiTerlewat  ]);  const getDataUserLocal = () => {  getData('user').then(res => {  console.log(res);  setUser(res);  });  getData('dataUser').then(res => {  console.log(res);  setDataPribadi(res);  });  };  const chatSend = () => {  const today = new Date();  const data = {  sendBy: user.uid,  chatDate: today.getTime(),  chatTime: getChatTime(today),  chatContent: chatContent,  };  const chatID = `${user.uid}\_${getDokter.data.uid}`;  const urlFirebase = `chatting/${chatID}/allChat/${setDateChat(today)}`;  const urlMessageUser = `messages/${user.uid}/${chatID}`;  const urlMessageDokter = `messages/${getDokter.data.uid}/${chatID}`;  const dataHistoryChatForUser = {  lastContentChat: chatContent,  lastChatDate: today.getTime(),  uidPartner: getDokter.data.uid,  status: 1  };  const dataHistoryChatForDokter = {  lastContentChat: chatContent,  lastChatDate: today.getTime(),  uidPartner: user.uid,  };  configfirebase.database()  .ref(urlFirebase)  .push(data)  .then(response => {  setChatContent('');  configfirebase.database().ref(urlMessageUser).set(dataHistoryChatForUser);  configfirebase.database().ref(urlMessageDokter).set(dataHistoryChatForDokter);  })  .catch(error => {  console.log(error);  });  // Kirim ke Firebase  };  return (  <View style={{ flex: 1, backgroundColor: 'white' }}>  <StatusBarComponent />  <View style={{ backgroundColor: 'white', elevation: 5, height: 70, borderBottomRightRadius: 20, borderBottomLeftRadius: 20, flexDirection: 'row', justifyContent: 'center', alignItems: 'center' }}>  <View style={{ flex: 3, marginLeft: 10, flexDirection: 'row', marginHorizontal: 10 }}>  <Image source={require("../../../assets/images/background-doctor.png")} resizeMode='cover' style={{ height: 50, width: 50, borderRadius: 50, borderColor: 'black', borderWidth: 1 }} />  <View style={{ color: 'black', marginHorizontal: 10 }}>  <Text style={{ color: 'black', fontFamily: 'Poppins-Medium', fontWeight: 'bold', fontSize: 14 }}>  {getDokter.data.nama}  </Text>  <Text style={{ color: 'gray', fontFamily: 'Poppins-Medium', fontWeight: 'bold', fontSize: 12 }}>  Dokter  </Text>  </View>  </View>  <View style={{flex: 1, marginRight: 20}}>  <TouchableOpacity onPress={() => {  navigation.navigate(Navigasi.MAIN\_APP)  }} style={{backgroundColor: 'red', paddingVertical: 5, borderRadius: 5}}>  <Text style={{color: 'white', fontFamily: 'Poppins-Medium', fontSize: 12, fontWeight: 'bold', textAlign: 'center'}}>  Kembali  </Text>  </TouchableOpacity>  </View>  </View>  <View style={{ flex: 1 }}>  {isKonsultasiTerlewat ? (  <Text style={{ color: 'red', fontFamily: 'Poppins-Medium', fontSize: 14, fontWeight: 'bold', textAlign: 'center', marginHorizontal: 10, marginTop: 10, marginBottom: 5 }}>  Waktu Konsultasi Selesai  </Text>  ) : (  <Text style={{ color: 'red', fontFamily: 'Poppins-Medium', fontSize: 14, fontWeight: 'bold', textAlign: 'center', marginHorizontal: 10, marginTop: 10, marginBottom: 5 }}>  Waktu Konsultasi: {formatTime(countdown)}  </Text>  )}  <ScrollView showsVerticalScrollIndicator={false}>  {chatData.map(chat => {  return (  <View key={chat.id}>  <Text  style={{  fontSize: 11,  color: 'gray',  marginVertical: 20,  textAlign: 'center',  }}>  {chat.id}  </Text>  {chat.data.map(itemChat => {  return (  <ChatItem  key={itemChat.id}  isMe={itemChat.data.sendBy === user.uid}  text={itemChat.data.chatContent}  date={itemChat.data.chatTime}  />  );  })}  </View>  );  })}  </ScrollView>  </View>  {isKonsultasiTerlewat && countdown != 0 ? (  <View style={{borderColor: 'red', borderWidth: 1, paddingVertical: 10, marginHorizontal: 10, marginVertical: 10, borderRadius: 10}}>  <Text style={{color: 'red', fontFamily: 'Poppins-Medium', fontWeight: 'bold', textAlign: 'center'}}>  Konsultasi Anda Sudah Selesai  </Text>  </View>  ) : (  <InputChat  value={chatContent}  onChangeText={value => setChatContent(value)}  onButtonPress={chatSend}  />  ) }  </View>  );  };  export default Chating; |

|  |
| --- |
| **LAMPIRAN 2. *PRODUCT BACKLOG*** |



|  |
| --- |
| **LAMPIRAN 3. *SPRINT PLANNING*** |



|  |
| --- |
| **LAMPIRAN 4. *DAILY ACTIVITY*** |





|  |
| --- |
| **LAMPIRAN 5. *TESTING PRODUCT BACKLOG (TEST CASE POSITIF)*** |





|  |
| --- |
| **LAMPIRAN 6. *TESTING PRODUCT BACKLOG (CASE NEGATIF)*** |





|  |
| --- |
| **LAMPIRAN 6. *TESTING ENDPOINT API*** |



|  |
| --- |
| **LAMPIRAN 7. BIODATA PENULIS** |

|  |  |  |
| --- | --- | --- |
| Nama | : Mohammad Ilham Teguhriyadi |  |
| Nomor Induk Mahasiswa | : 2003077 |
| Jenis Kelamin | : Laki-Laki |
| Tempat, Tanggal Lahir | : Cirebon, 29 September 2002 |
| Status | : Belum Menikah |
| Agama | : Islam |
| Asal Ijazah Sekolah | : Nama Sekolah | Tahun Ijazah |
| SD | : SDN Kebon Baru 6 Kota Cirebon | * 2014 |
| SMP | : SMPN 2 Kota Cirebon | * 2017 |
| SMA | : SMKS Informatika Al-Irsyad Al-Islamiyah kota Cirebon | * 2020 |
| Program Studi/Jurusan | : Teknik Informatika |  |
| Alamat Rumah Asal | : Gunung Jati, Cirebon |  |
| No. Handphone | : 081214707143 |  |
| Email | : [ilham.teguh55@gmail.com](mailto:ilham.teguh55@gmail.com) |  |
| Nama Orang Tua | : Suripah |  |
| Pekerjaan Orang Tua | : Wirausaha/Ibu Rumah Tangga |  |
| Judul Tugas Akhir | : Aplikasi Pelayanan Kesehatan dan Konsultasi Masyarakat berbasis *Mobile* menggunakan React Native |  |
| Pembimbing | : Adi Suheryadi, S.ST., M.Kom. |  |