# **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 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).

## 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).

## 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 praktek, 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

*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 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 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 praktek 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 merupakan garis yang menentukan lokasi berada di sebelah utara atau selatan ekuator. Garis lintang diukur mulai dari titik 0 derajat dari khatulistiwa sampai 90 derajat di kutub.

## Longitude

Longitude atau garis bujur merupakan digunakan untuk menentukan lokasi di wilayah barat atau timur dari garis utara selatan yang sering disebut juga garis meridian. Garis bujur diukur dari 0 derajat di wilayah Greenwich sampai 180 derajat di International Date Line.

## Teorema Haversine

Metode Haversine adalah sebuah persamaan yang penting dalam bidang navigasi, dan dapat digunakan untuk menghitung jarak antara dua titik, berdasarkan posisi garis lintang (*latitude*) dan posisi garis bujur (*longitude*) sebagai variabel masukan (Adiwilaga, 2014). *Haversine Formula member*ikan jarak lingkaran besar antara dua titik pada permukaan bumi berdasarkan garis bujur dan garis lintang. Dengan mengasumsikan bahwa bumi berbentuk bulat sempurna dengan jari-jari *R* 6.371 km.

Pertama ditentukan terlebih dahulu titik awal dan titik tujuan, titik awal berupa *latitude1* (lat1) dan *longitude* (long1). Titik awal dan titik tujuan tersebut berbentuk desimal derajat yang dilakukan perubahan menjadi nilai sudut radian, kemudian melakukan perhitungan dengan rumus *Haversine Formula* (Adiwilaga, 2014). contoh perhitungan dengan menggunakan rumus *Theorema Haversine* sebagai berikut:

=(6371,1\*((2\*ASIN(SQRT((SIN((RADIANS(-6.883675)–RADIANS(- 6.894896))/2)^2)+COS(RADIANS(-6.883675))\*COS(RADIANS(-6.894896))\* (SIN((RADIANS(107.595555)RADIANS(107.603720))/2)^2)))))) \* 1000

= 1539,256128 m

*Theorema Haversine Formula* digunakan sebagai metode yang membantu untuk mengetahui pengukuran jarak antara dua titik dengan mengasumsikan bahwa bumi bulat.

## API

API (Application Programming Interface) adalah representasi fungsi-fungsi pada logic basis data yang diakses oleh logic antarmuka secara programmatis. API digunakan sebagai cara dua bagian seperti aplikasi atau program untuk saling berkomunikasi. Penerapan API ini sifatnya sangat luas dan dapat digunakan dalam bahasa pemrograman, framework, sistem operasi, dan web service.

Metode yang digunakan web service dapat berupa REST (Representational State Transfer). REST merupakan gaya perancangan pengembangan API berbasis *mobile* dengan menggunakan HTTP dalam komunikasi data. Metode HTTP yang biasa digunakan adalah get, post, delete, dan put. REST API digunakan untuk interaksi antar mesin dengan HTTP. Klien akan mengakses URL yang di dalamnya terdapat resources yang diinginkan klien.

## 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 web 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).

## Metode Agile

Menurut Adi dan Permana (2015) dalam penelitiannya yang berjudul “Scrum Method Implementation in a Software Development Project Management” menyatakan bahwa Scrum pertama kali dikembangkan oleh Jeff Sutherland pada tahun 1993 dan tujuannya adalah menjadi metodologi pengembangan yang mengikuti prinsip-prinsip metodologi Agille. Scrum merupakan sebuah kerangka kerja untuk menyelesaikan pekerjaan-pekerjaan yang kompleks dan selalu berubah. Kerangka Scrum digunakan untuk menjawab persoalan adaptif yang kompleks, menghasilkan kreatifitas dan inovasi.

## 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, Hartanto: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

## 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 berupa artikel kesehatan.
      5. Mendapatkan informasi mengenai 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 ataupun klinik.

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. 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.

### 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 dengan mudah 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 i7-6600U |
| 2. | Memori | 8 GB |
| 3. | Hardisk | 256 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

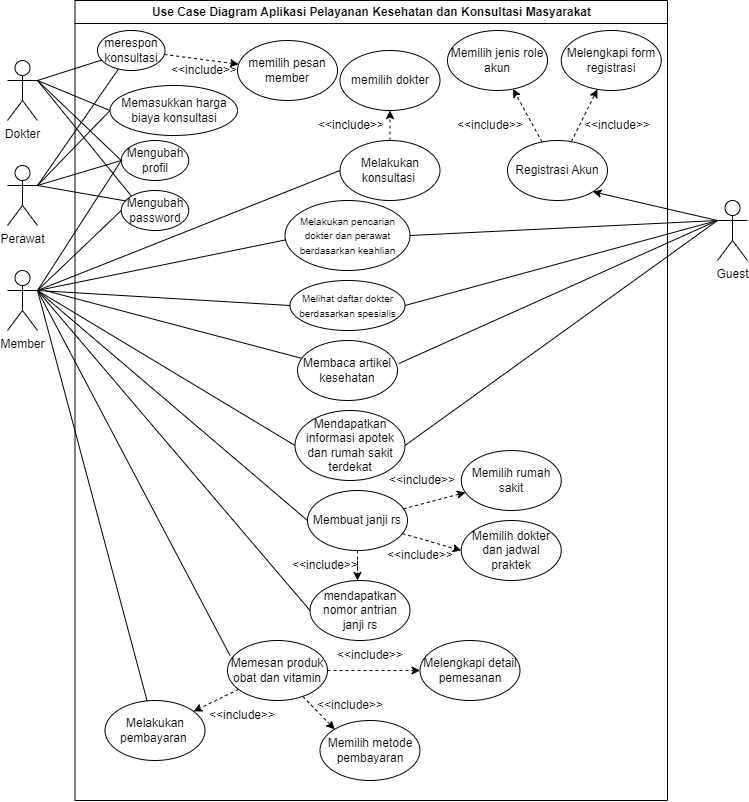
|  |  |  |
| --- | --- | --- |
| **No** | **Jenis *Software*** | **Kebutuhan *Software*** |
| 1. | Bahasa *Scripting* | JavaScript |
| 2. | *Software* Pengolah | VsCode |
| 3. | Penyimpanan Data | Basis data (MySql dan Firebase) |
| 4. | Jenis Ponsel | Oppo |
| 5 | Sistem Operasi | Android 12 |

## 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 actor dengan sistem. Dalam aplikasi ini menjelaskan tujuh aktor yang memiliki beberapa aksi. Adapun rancangan *Use Case* dari aplikasi pelayanan kesehatan dan konsultasi masyarakat dapat dilihat pada Gambar 3.1.



Gambar 3.1 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. |
| Perawat | Menetapkan harga konsultasi | Menu ini digunakan oleh perawat untuk menetapkan harga konsultasi perawat. |
| Mengubah Profil | Menu ini digunakan oleh dokter untuk mengubah profil. |
| Mengubah Password | Menu ini digunakan oleh dokter untuk mengubah password akun. |
| *Member*  *Member* | Melakukan pencarian dokter dan perawat berdasarkan keahlian | Menu ini digunakan oleh *member* untuk mencari dokter atau perawat berdasarkan keahliannya. |
| Melihat daftar dokter berdasarkan spesialis | Menu ini digunakan oleh *member* untuk mendapatkan informasi daftar dokter berdasarkan spesialisnya. |
| 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. |

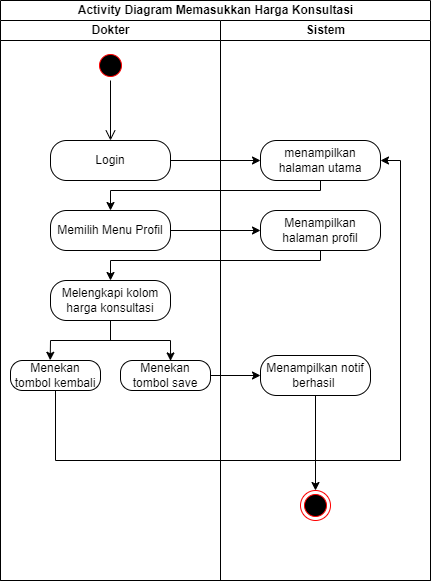
### Activity Diagram

*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).

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

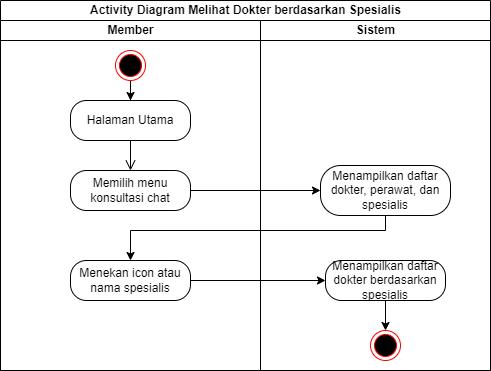
Pada *activity* diagram ini aktor dokter dan perawat menetapkan harga biaya konsultasi. Aktivitas yang pertama dilakukan adalah login aplikasi, sesudahnya memilih menu profil. Lalu pada kolom biaya, lengkapi dengan memasukkan nominal biaya. Adapun *activity* diagram terdapat dalam Gambar 3.2.



Gambar 3.2 Activity Diagram Menetapkan Harga Konsultasi Dokter dan Perawat

1. *Activity* Diagram Melihat Dokter berdasarkan Spesialis

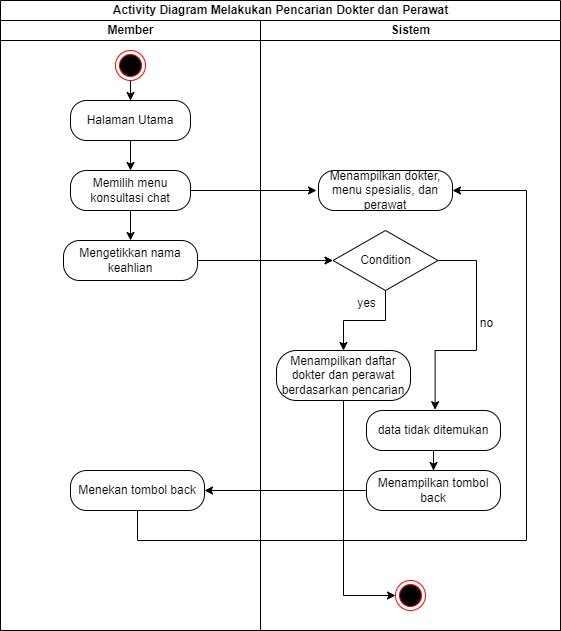
Pada *activity* diagram ini *member* melakukan aksi melihat dokter berdasarkan spesialis. Langkah yang dilakukan adalah dengan masuk ke halaman utama aplikasi, sesudahnya menekan menu konsultasi chat, sistem akan menampilkan daftar dokter, perawat, dan spesialis. Kemudian aktor *member* menekan menu spesialis. Sistem akan menampilkan dokter berdasarkan nama spesialis yang di-klik. Adapun *activity* diagram melihat dokter berdasarkan spesialis dapat dilihat dalam Gambar 3.3.



Gambar 3.3 Activity Diagram Melihat Dokter berdasarkan Spesialis

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

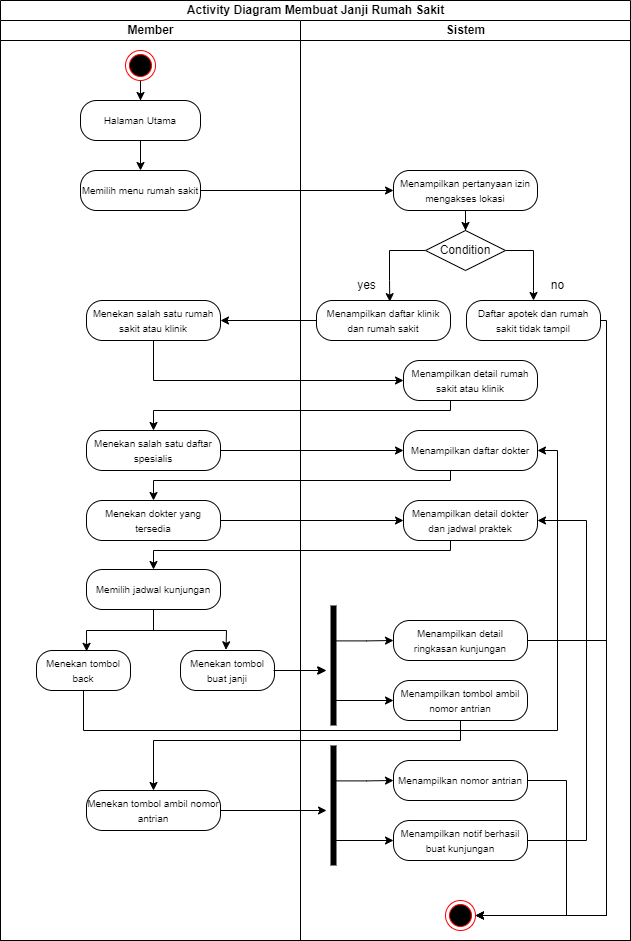
Pada *activity* diagram ini aktor *member* melakukan pencarian nama dokter dan perawat berdasarkan keahlian yang dimilikinya. Dalam aktivitas awalnya, *member* masuk ke halaman utama aplikasi kemudia memilih menu konsultasi chat. Sesudahnya mengetikkan nama keahlian yang hendak dicari. Sistem akan menampilkan dokter dan perawat berdasarkan keahlian yang diketik. Jika data yang diketik tersedia atau ada di dalam sistem, maka akan menampilkan daftar dokter dan perawat. Jika tidak ada maka akan menampilkan pesan data tidak ada lalu mengembalikan ke halaman sebelumnya jika menekan tombol *back.* 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

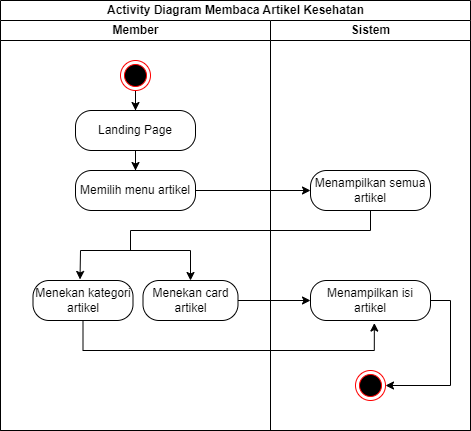
Pada *activity* diagram ini *member* membuat janji kunjungan rumah sakit. Aktivitas pertama yang dilakukan oleh *member* adalah memilih menu rumah sakit, lalu sistem akan menampilkan pertanyaan untuk meminta izin mengakses lokasi *member.* Jika diizinkan mengakses lokasi, maka sistem akan menampilkan daftar rumah sakit yang terdekat berdasarkan lokasi *member* berada. Selanjutnya, *member* menekan salah satu rumah sakit yang akan dikunjungi, setelahnya sistem akan menampilkan detail informasi dan data spesialis yang tersedia pada rumah sakit. Untuk membuat janji kunjungan, *member* perlu login ke dalam aplikasi terlebih dahulu. Lalu, *member* menekan salah satu nama spesialis yang ingin dikunjungi, kemudian sistem akan menampilkan daftar dokter berdasarkan spesialisnya. Untuk membuat kunjungan, *member* menekan tombol buat janji pada kolom dokter, kemudian sistem akan menampilkan detail kunjungan. Jika sudah sesuai, selanjutnya menekan tombol nomor antrian untuk mendapatkan nomor kunjungan. Setelahnya *member* akan mendapatkan nomor antrian yang nantinya dibawa ke rumah sakit tujuan.



Gambar 3.5 Activity Diagram Membuat Janji Kunjungan Rumah Sakit

1. *Activity* Diagram Membaca Artikel Kesehatan

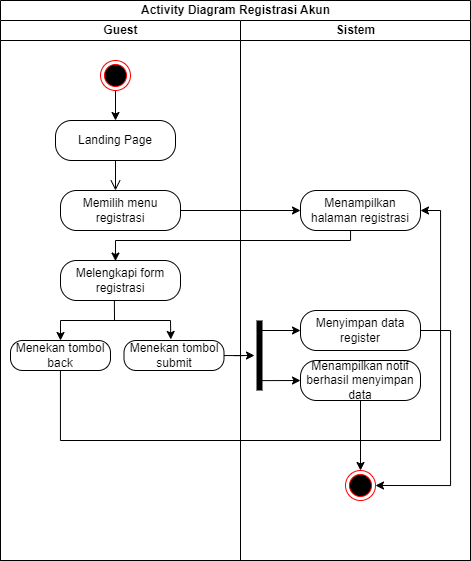
Pada *activity* diagram ini *member* dan *guest* dapat membaca artikel kesehatan. Untuk membaca artikel, aktivitas yang pertama dilakukan adalah masuk ke dalam aplikasi, dapat dengan login terlebih dahulu atau tidak. Pada halaman utama aplikasi, terdapat empat artikel terbaru yang di-*publish.* Aktor dapat langsung membaca artikel melalui halaman utama. Jika ingin membaca semua artikel dapat mengunjungi menu artikel yang tersedia pada bagian navbar aplikasi. *Member* juga 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. *Guest* wajib memilih mendaftar sebagai jenis *role* apa. *Role* yang tersedia pada halaman register ini terdapat *role* dokter, perawat, konsumen, owner apotek, dan owner rumah sakit. *Activity* diagram registrasi akun dapat dilihat pada Gambar 3.7.



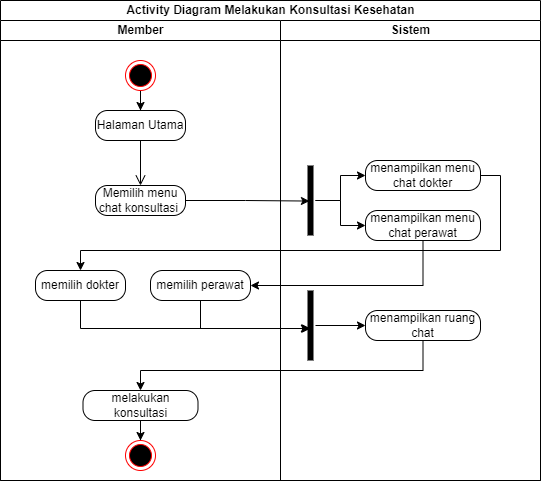
Gambar 3.7 Activity Diagram Registrasi Akun

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

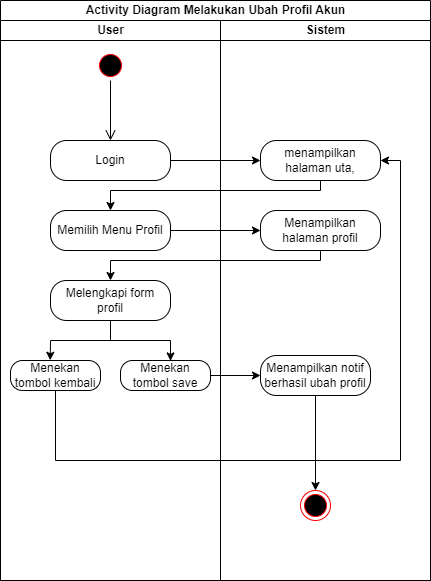
Pada *activity* diagram ini, *member* melakukan konsultasi kesehatan dengan masuk ke halaman utama, setelahnya memilih menu chat konsultasi, kemudian sistem akan menampilkan menu chat dokter dan perawat. Jika *member* ingin melakukan konsultasi, perlu untuk memilih dokter atau perawat, sesudahnya sistem 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 Mengubah Profil Akun

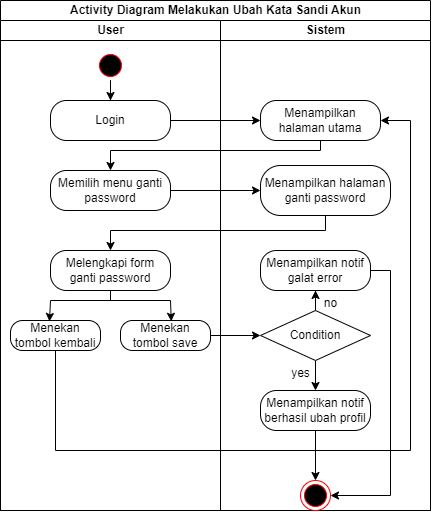
Pada *activity* diagram ini, *user* melakukan ubah profil dengan melakukan login terlebih dahulu. Setelahnya, memilih menu profil maka sistem akan menampilkan halaman profil. Untuk melakukan perubahan data akun, *user* perlu mengisi kolom-kolom profil yang akan diubah. Adapun *activity* diagram mengubah profil dapat dilihat pada gambar 3.10.



Gambar 3.10 Activity Diagram Mengubah Profil Akun

1. *Activity* Diagram Mengubah Password Akun

Pada *activity* diagram ini, *user* melakukan ubah password dengan melakukan login terlebih dahulu. Setelahnya, memilih menu ubah password maka sistem akan menampilkan ubah password. Untuk melakukan perubahan password akun, *user* perlu mengisi kolom password. Adapun *activity* diagram mengubah password dapat dilihat pada gambar 3.11.

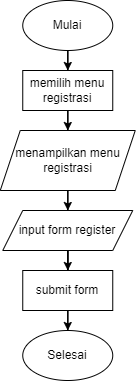


Gambar 3.11 Activity Diagram Mengubah Password

## Flowchart

### Flowchart Registrasi Akun

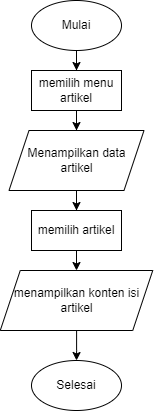
Flowchart berikut merupakan proses untuk melakukan pendaftaran akun. *Guest* akan memilih menu registrasi dan kemudian mengisi form registrasi. Setelahnya di-*submit*. Maka, data registrasi akan tersimpan.



Gambar 3.2 Flowchart Register

### Flowchart Membaca Artikel

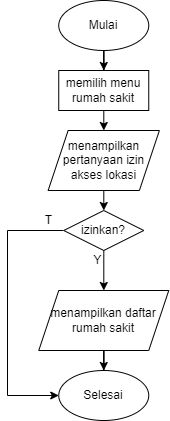
Flowchart berikut merupakan proses untuk membaca artikel kesehatan. Adapun flowchart membaca artikel dapat dilihat pada gambar 3.19.



Gambar 3.3 Flowchart Membaca Artikel

### Flowchart Melihat Rumah Sakit Terdekat

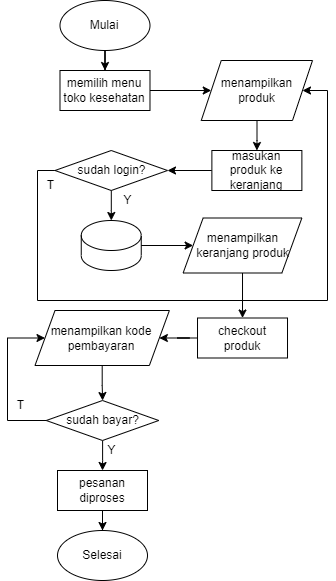
Flowchart berikut merupakan proses untuk melihat rumah sakit terdekat berdasarkan lokasi. Adapun flowchart melihat rumah sakit terdekat dapat dilihat pada gambar 3.20.



Gambar 3.4 Flowchart Melihat Rumah Sakit Terdekat

### Flowchart Memesan Produk Apotek

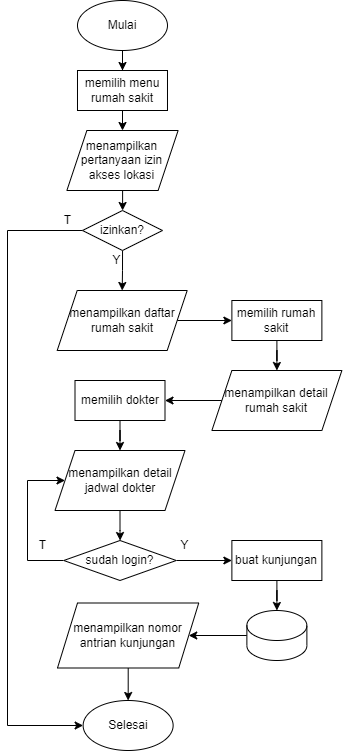
Flowchart berikut merupakan proses untuk memesan produk apotek. Adapun flowchart memesan produk apotek dapat dilihat pada gambar 3.21.



Gambar 3.5 Flowchart Memesan Produk Apotek

### Flowchart Membuat Janji Kunjungan Rumah Sakit

Flowchart berikut merupakan proses untuk membuat janji kunjungan rumah sakit. Adapun flowchart membuat janji kunjungan rumah sakit dapat dilihat pada gambar 3.22.

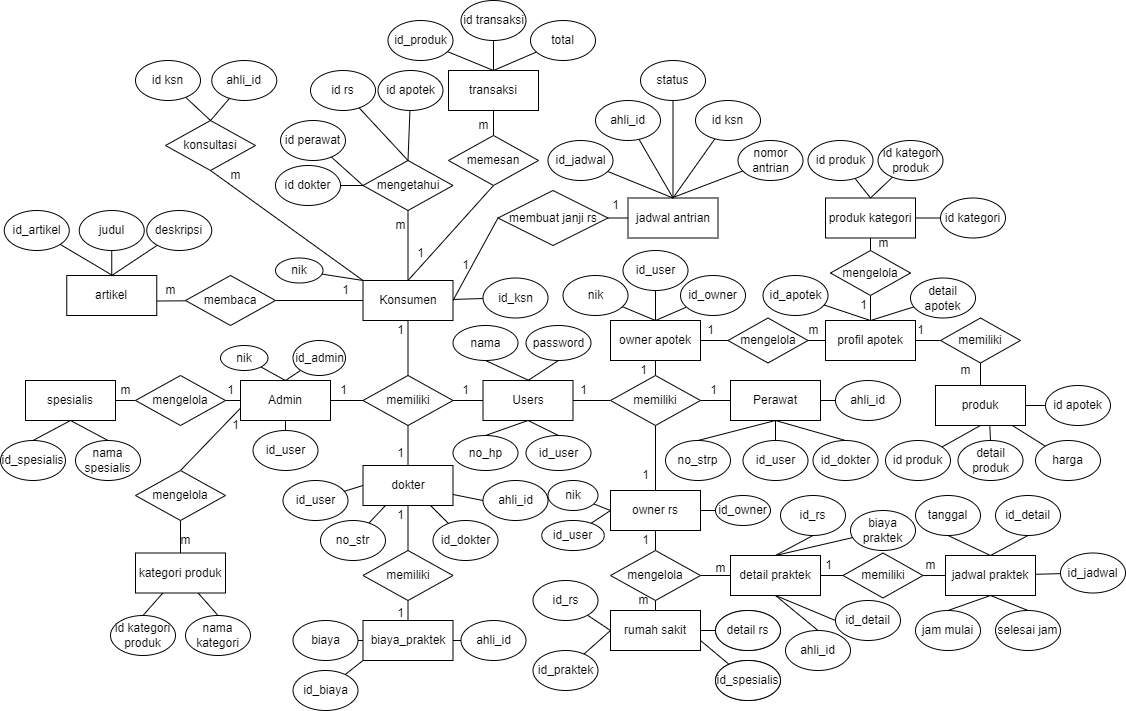


Gambar 3.6 Flowchart Membuat Janji Kunjungan Rumah Sakit

### 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 3.7 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 |
| **Nama tabel** | **Keterangan** |
| 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 Praktek | Tabel ini digunakan untuk mengelola praktek dokter di dalam rumah sakit. |
| Jadwal Praktek | Tabel ini digunakan untuk mengelola jadwal praktek dokter di dalam rumah sakit. |
| Biaya Praktek | 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 Praktek

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

Tabel 3.16 Tabel Detail Praktek

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

* + - 1. Tabel Jadwal Praktek

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

Tabel 3.17 Tabel Jadwal Praktek

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Nama Kolom** | **Tipe Data** | **Keterangan** |
| 1. | id\_detail\_praktek | varchar(50) | NOT NULL |
| 2. | id\_jadwal\_praktek | 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 yang memiliki id\_jadwal\_antrian, konsumen\_id, ahli\_id, id\_jadwal\_praktek, nomor\_antrian, status, dan tanggal. 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\_praktek | 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 web 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 *Member*

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 Pembuka

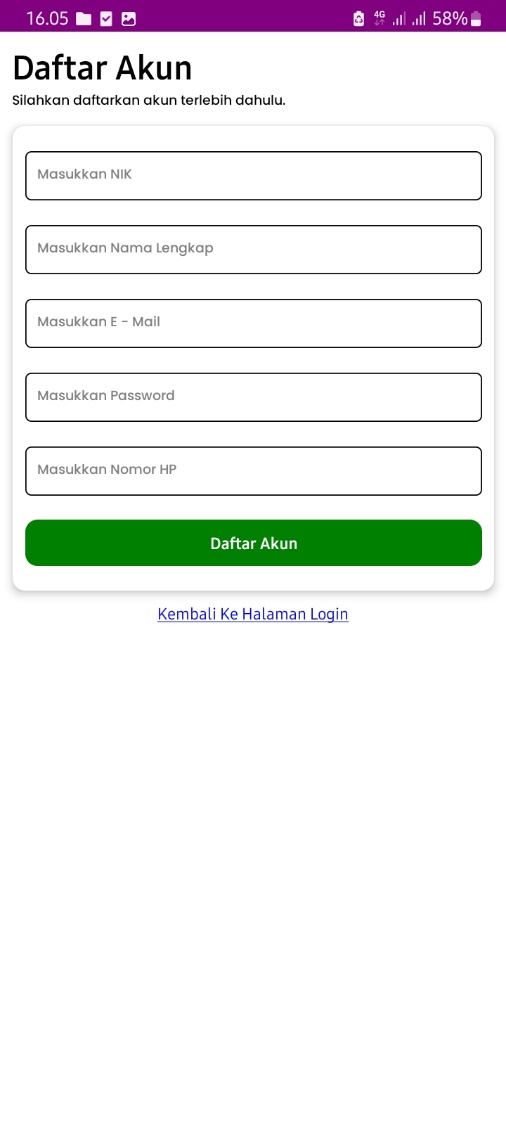
Antarmuka ini merupakan rancangan *user interface* dari sisi *guest* dan *member,* rancangan *interface* pada halaman pembuka terdapat pilihan yaitu login dan register . Adapun rancangan antarmuka halaman pembukadapat dilihat pada gambar 3.19.



Gambar 3.8 Rancangan Antarmuka Halaman Pembuka

1. Rancangan Antarmuka Halaman Registrasi

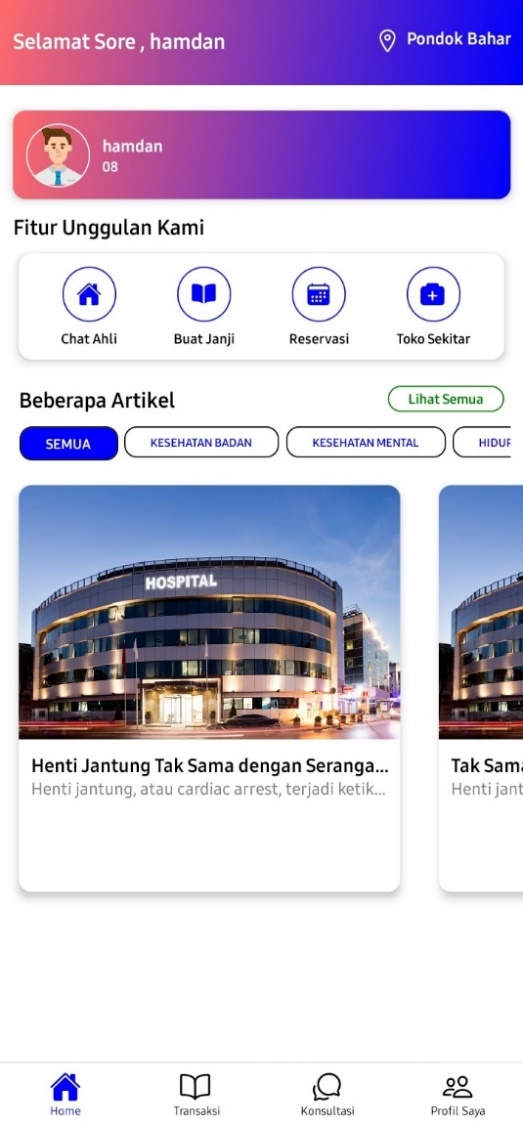
Rancangan antarmuka ini merupakan rancangan *user interface* untuk *guest* dalam membuat akun *member.* Adapun rancangan antarmuka halaman registrasi dapat dilihat pada gambar 3.9.



Gambar 3.9 Rancangan Antarmuka Halaman Registrasi

1. Rancangan Antarmuka Halaman Utama

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 Berikut dapat dilihat pada gambar.



Gambar 3.10 Rancangan Antramuka Halaman Utama

1. Rancangan Antarmuka Halaman Detail Artikel

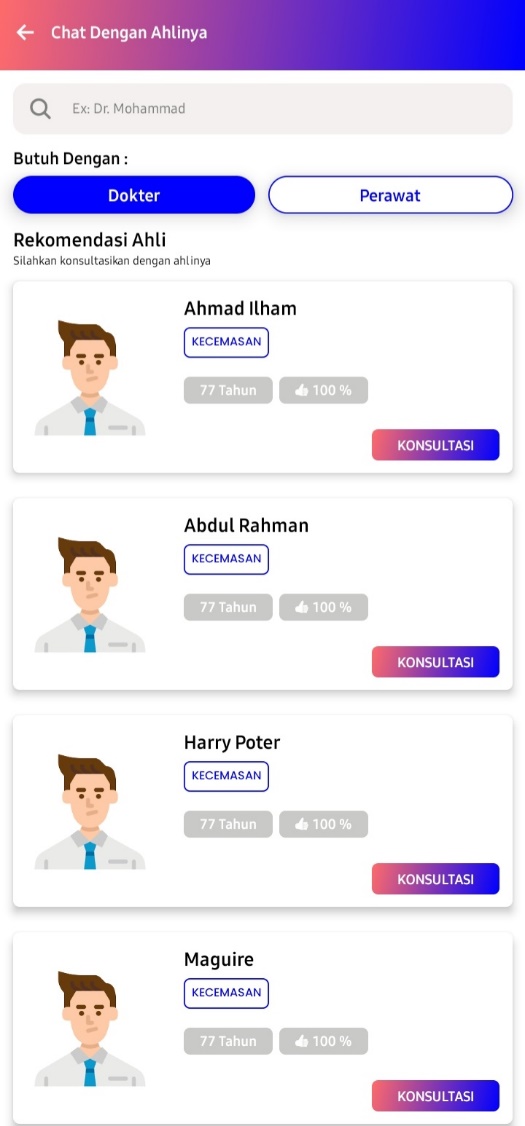
Rancangan antarmuka ini digunakan untuk menampilkan data konten dari artikel. Di dalam antarmuka ini terdapat gambar dan deskripsi artikel.



Gambar 3.11 Rancangan Antarmuka Halaman Detail Artikel

1. Rancangan Antarmuka Menu *Chat* Dokter

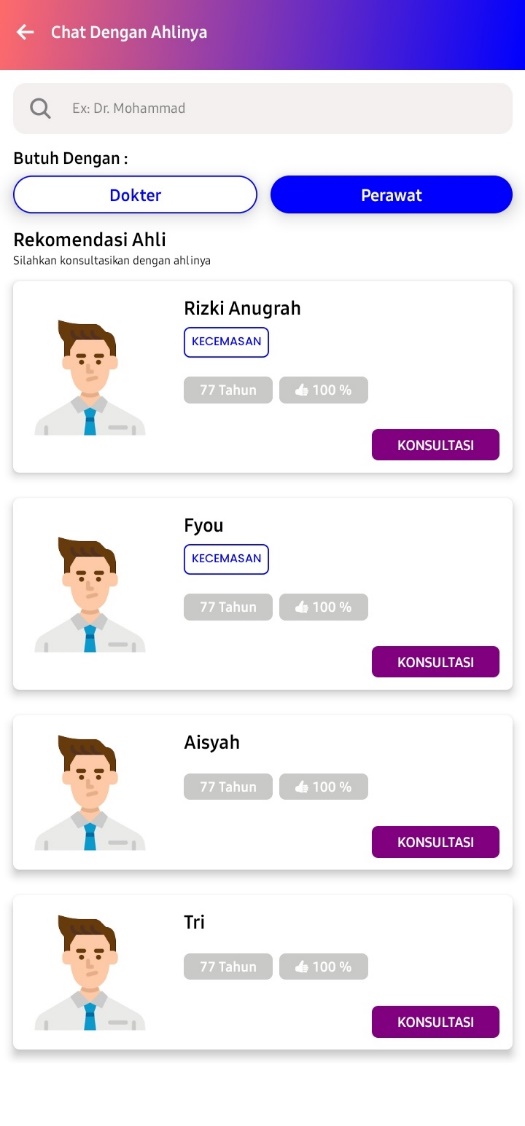
Rancangan antarmuka ini digunakan *member* untuk melakukan konsultasi kesehatan dengan dokter. Terdapat daftar dokter dan kolom *input* untuk mencari keahlian dokter. Adapun rancangan antarmuka menu *chat* dengan dokter dapat dilihat pada gambar 3.12.



Gambar 3.12 Rancangan Antarmuka Menu Chat Dokter

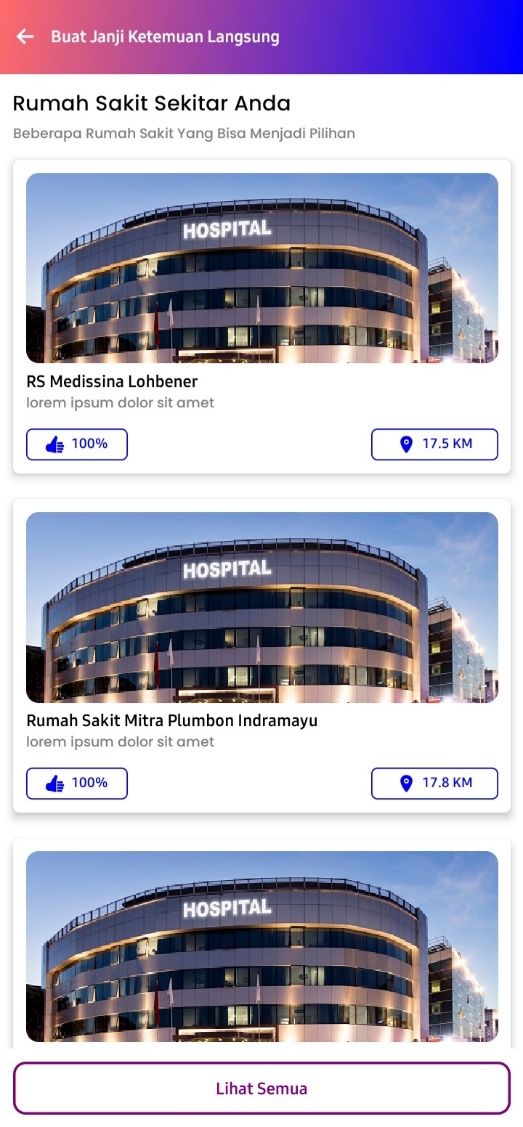
1. Rancangan Antarmuka Menu *Chat* Perawat

Rancangan antarmuka ini digunakan *member* untuk melakukan konsultasi kesehatan dengan perawat. Terdapat daftar perawat dan kolom *input* untuk mencari keahlian perawat. Adapun rancangan antarmuka menu *chat* dengan perawat dapat dilihat pada gambar 3.13.



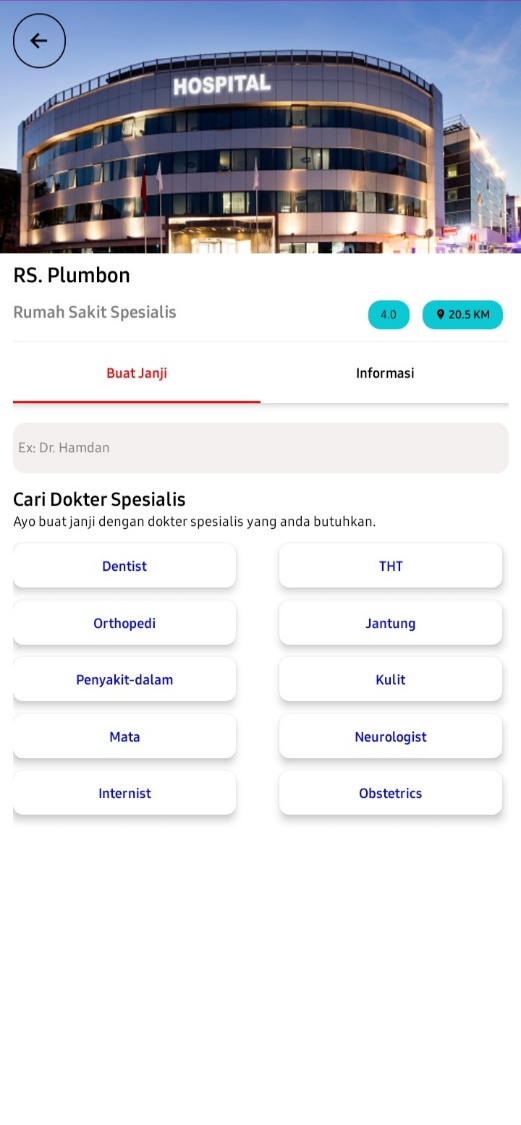
1. Rancangan Antarmuka Menu Buat Janji Rumah Sakit

Rancangan antarmuka ini merupakan menu untuk *member* membuat janji kunjungan. Berikut rancangan antarmuka menu buat janji rumah sakit dapat dilihat pada gambar.



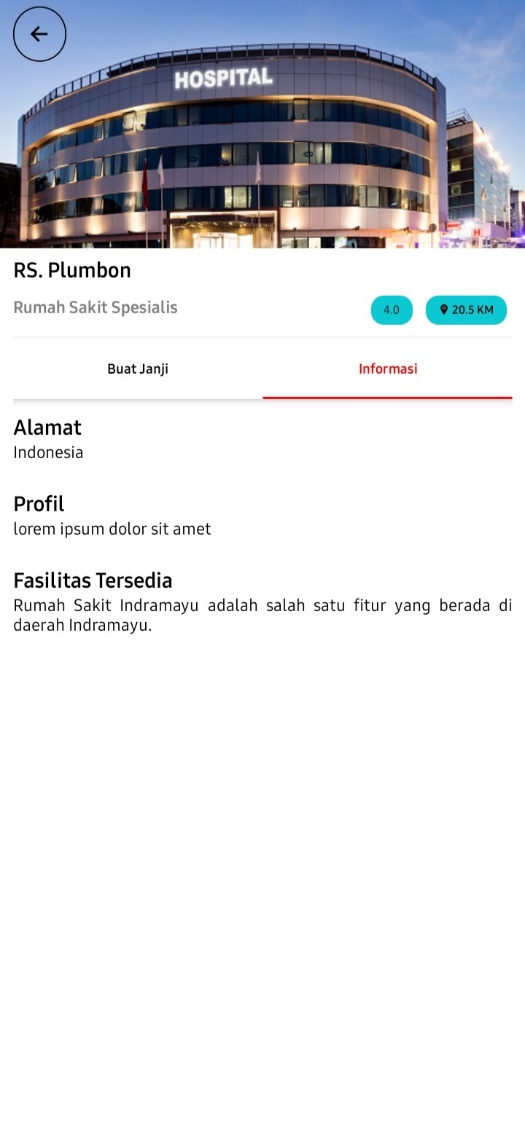
1. Rancangan Antarmuka Rumah Sakit

Rancangan antarmuka rumah sakit merupakan menu detail dari rumah sakit yang menampilkan detail rumah sakit yang berisi daftar spesialis dan informasi terkait rumah sakit. Berikut rancangan antarmuka menu rumah sakit dapat dilihat pada gambar.



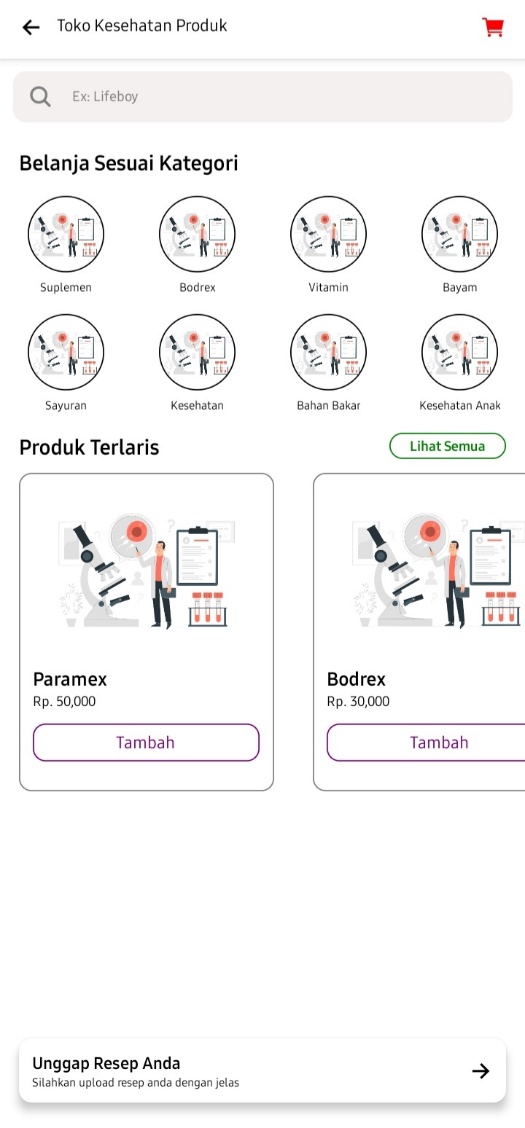
1. Rancangan Antarmuka Informasi Rumah Sakit

Rancangan antarmuka informasi rumah sakit merupakan tampilan dalam rumah sakit yang menampilkan informasi rumah sakit. Adapun rancangan antarmuka informasi rumah sakit dapat dilihat pada gambar berikut.



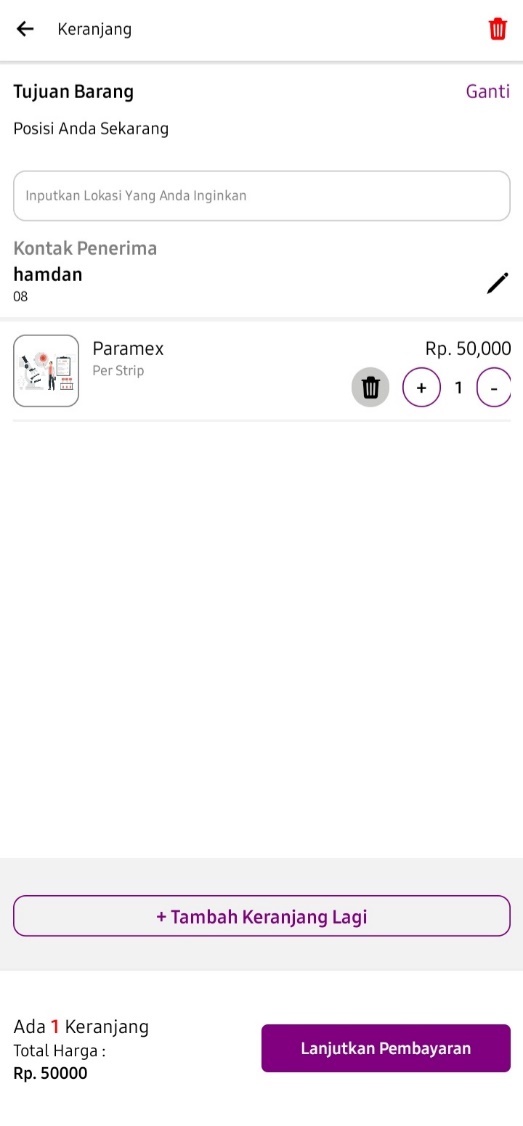
1. Rancangan Antarmuka Toko Kesehatan

Rancangan antarmuka ini merupakan *user interface* untuk menampilkan data produk yang tersedia dalam aplikasi. Pada menu ini juga, member dapat melakukan pemesanan produk dengan menekan tombol tambah. Adapun perancangan antarmuka toko kesehatan dapat dilihat pada gambar.



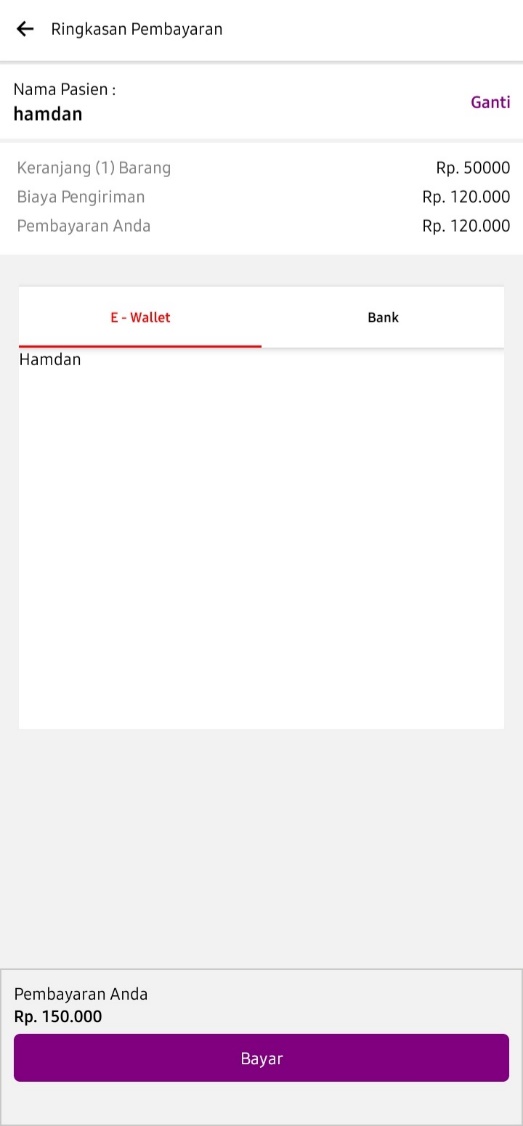
1. Rancangan Antarmuka Detail Keranjang Produk

Rancangan antarmuka detail keranjang produk merupakan *user interface* untuk menampilkan halaman rincian keranjang. Adapun rancangan antarmuka dapat dilihat pada gambar.



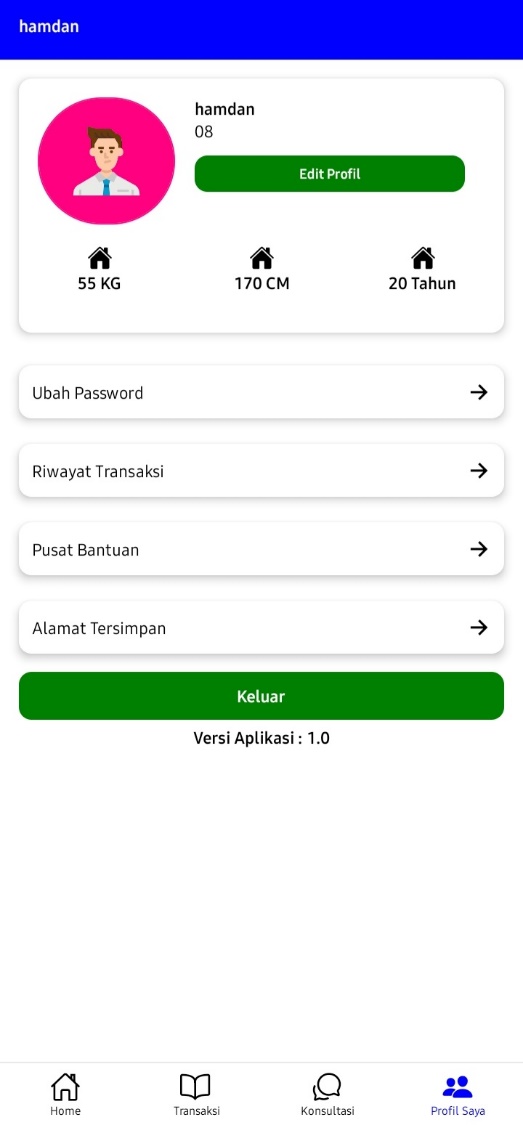
1. Rancangan Antarmuka Detail Pembayaran Produk

Rancangan antarmuka detail pembayaran produk merupakan *user interface* untuk menampilkan halaman ringkasan total pembayaran. Adapun rancangan antarmuka detail pembayaran produk dapat dilihat pada gambar.



1. Rancangan Antarmuka Halaman Profil

Rancangan antarmuka halaman profil merupakan *user interface* untuk menampilkan halaman profil *user.* Rancangan antarmuka halaman profil dapat dilihat pada gambar.



### Rancangan Antarmuka Dokter

Rancangan antarmuka admin merupakan rancangan *user interface* dari sisi dokter. Berikut adalah rancangan antarmuka admin:

1. Rancangan Antarmuka Halaman Dashboard Utama

### Rancangan Antarmuka Perawat

Rancangan antarmuka perawat merupakan rancangan *user interface* dari sisi dokter. Berikut adalah rancangan antarmuka perawat:

1. Rancangan antarmuka halaman kelola artikel