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

***FRAMEWORK* VUEJS**

**TUGAS AKHIR**Sebagai salah satu syarat untuk menyelesaikan   
Program Diploma III Teknik Informatika  
Politeknik Negeri Indramayu  
  
  
  
  
**Oleh:  
MUHAMAD RAFLI SEPTIAN  
NIM. 2003018  
  
  
  
PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
POLITEKNIK NEGERI INDRAMAYU   
AGUSTUS 2023**

# **HALAMAN PENGESAHAN**

Tugas Akhir ini diajukan oleh:  
Nama : MUHAMAD RAFLI SEPTIAN  
NIM : 2003018  
ProgramStudi : Diploma III Teknik Informatika  
Judul : Aplikasi Pelayanan Kesehatan dan Konsultasi Masyarakat berbasis   
 Web menggunakan *Framework* VUE JS  
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, 26 September 2023 |
| Yang menyatakan, |
|  |
|  |
| Dewa Buana, M.T. |
| NIM. 2003018 |

# **ABSTRAK**

# ***ABSTRACT***

# **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 di dalam Jurusan Teknik Informatika untuk menyelesaikan Diploma III Program Studi Teknik Informatika pada Politeknik Negeri Indramayu.

Dalam penulisan dan 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 Muhamad Mustamiin, S.Pd., M.Kom. selaku Wali Dosen kelas D-3 TI 3 A.
3. Adi Suheryadi, S.ST., M.Kom. selaku Dosen Pembimbing yang telah *member*i dukungan dan arahan.
4. Kedua orang tua dan keluarga yang telah *member*i dukungan moril dan materil yang tidak pernah henti.
5. Mohammad Ilham Teguhriyadi dan Nandang Eka Prasetya, selaku rekan seperjuangan dalam mengerjakan Tugas Akhir.
6. Kedai kita, selaku tempat pengerjaan Tugas Akhir.

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 Web menggunakan *framework* VueJs” dapat menambah wawasan serta ilmu pengetahuan bagi pembaca.

# **DAFTAR ISI**

[**HALAMAN PENGESAHAN** 2](#_Toc138748142)

[**HALAMAN PERNYATAAN** 3](#_Toc138748143)

[**ABSTRAK** 4](#_Toc138748144)

[***ABSTRACT*** 5](#_Toc138748145)

[**KATA PENGANTAR** 6](#_Toc138748146)

[**DAFTAR ISI** 7](#_Toc138748147)

[**DAFTAR GAMBAR** 10](#_Toc138748148)

[**DAFTAR TABEL** 11](#_Toc138748149)

[**DAFTAR LAMPIRAN** 12](#_Toc138748150)

[**BAB I PENDAHULUAN** 13](#_Toc138748151)

[1.1 Latar Belakang 13](#_Toc138748152)

[1.2 Rumusan Masalah 14](#_Toc138748153)

[1.3 Batasan Masalah 15](#_Toc138748154)

[1.4 Tujuan 15](#_Toc138748155)

[1.5 Manfaat 15](#_Toc138748156)

[1.6 Sistematika Penulisan 16](#_Toc138748157)

[**BAB II LANDASAN TEORI** 17](#_Toc138748158)

[2.1. Dokter 17](#_Toc138748159)

[2.2 Dokter Praktik Umum 17](#_Toc138748160)

[2.3 Dokter Spesialis 18](#_Toc138748161)

[2.4 Keperawatan 18](#_Toc138748162)

[2.5 Perawat 18](#_Toc138748163)

[2.5.1 Peran Perawat 19](#_Toc138748164)

[2.5.2 Peran sebagai Pelaksana Kesehatan 19](#_Toc138748165)

[2.5.3 Peran sebagai Pendidik 19](#_Toc138748166)

[2.5.4 Peran sebagai administrasi 19](#_Toc138748167)

[2.5.5 Peran sebagai konseling 19](#_Toc138748168)

[2.5.6 Peran sebagai peneliti 20](#_Toc138748169)

[2.6 Konsultasi Kesehatan Daring 20](#_Toc138748170)

[2.7 Latitude 20](#_Toc138748171)

[2.8 Longitude 20](#_Toc138748172)

[2.9 Teorema Haversine 20](#_Toc138748173)

[2.10 API 21](#_Toc138748174)

[2.11 Vue.js 21](#_Toc138748175)

[2.12 Metode Scrum 22](#_Toc138748176)

[2.13 UML (Unified Modeling Language) 23](#_Toc138748177)

[2.13.1 Pengenalan UML 23](#_Toc138748178)

[2.13.2 *Use Case* Diagram 23](#_Toc138748179)

[2.13.2 *Activity* Diagram 24](#_Toc138748180)

[2.14 *Entity Relationship* Diagram(ERD) 25](#_Toc138748181)

[2.15 *Global Positioning System* (GPS) 26](#_Toc138748182)

[2.16 OpenStreetMap 26](#_Toc138748183)

[2.17 LeafletJS 26](#_Toc138748184)

[2.18 *Black Box Testing* 27](#_Toc138748185)

[**BAB III METODE PELAKSANAAN** 28](#_Toc138748186)

[3.1 Metode Pengembangan Perangkat Lunak 28](#_Toc138748187)

[3.2 Analisis Kebutuhan Sistem 28](#_Toc138748188)

[3.2.1 Kebutuhan Fungsional 29](#_Toc138748189)

[3.2.2 Kebutuhan Non Fungsional 29](#_Toc138748190)

[3.2.3 Kebutuhan *Hardware* 29](#_Toc138748191)

[3.2.4 Kebutuhan *Software* 29](#_Toc138748192)

[3.3 Perancangan Sistem 28](#_Toc138748193)

[3.3.1 Use Case Diagram 29](#_Toc138748194)

[3.3.2 Activity Diagram 29](#_Toc138748195)

[3.4 Perancangan Basis Data 28](#_Toc138748196)

[3.5 Perancangan Desain Tampilan Antarmuka 28](#_Toc138748197)

[**BAB IV HASIL DAN PEMBAHASAN** 29](#_Toc138748198)

[**BAB V PENUTUP** 30](#_Toc138748199)

[**DAFTAR PUSTAKA** 31](#_Toc138748200)

# **DAFTAR GAMBAR**

[Gambar 3.1 Use Case Diagram 33](#_Toc138746767)

[Gambar 3.2 Activity Diagram Mengelola Data Rumah Sakit 38](#_Toc138746768)

[Gambar 3.3 Activity Diagram Mengelola Fasilitas Rumah Sakit 39](#_Toc138746769)

[Gambar 3.4 Acitivity Diagram Mengelola Data Spesialis Rumah Sakit 40](#_Toc138746770)

[Gambar 3.5 Activity Diagram Mengelola Jadwal Praktek Dokter Rumah Sakit 41](#_Toc138746771)

[Gambar 3.6 Activity Diagram Mengelola Data Apotek 42](#_Toc138746772)

[Gambar 3.7 Activity Diagram Mengelola Data Produk Apotek 43](#_Toc138746773)

[Gambar 3.8 Activity Diagram Mengubah Status Apotek 44](#_Toc138746774)

[Gambar 3.9 Activity Diagram Mengelola Data Grouping Produk 45](#_Toc138746775)

[Gambar 3.10 Activity Diagram Melihat Kategori Artikel 46](#_Toc138746776)

[Gambar 3.11 Activity Diagram Menetapkan Harga Konsultasi Dokter dan Perawat 47](#_Toc138746777)

[Gambar 3.12 Activity Diagram Melihat Dokter berdasarkan Spesialis 48](#_Toc138746778)

[Gambar 3.13 Activity Diagram Melakukan Pencarian Dokter dan Perawat 49](#_Toc138746779)

[Gambar 3.14 Activity Diagram Membuat Janji Kunjungan Rumah Sakit 50](#_Toc138746780)

[Gambar 3.15 Activity Diagram Membaca Artikel Kesehatan 51](#_Toc138746781)

[Gambar 3.16 Activity Diagram Registrasi Akun 52](#_Toc138746782)

[Gambar 3.17 Activity Diagram Mengubah Status Akun Dokter 53](#_Toc138746783)

[Gambar 3.18 Rancangan ERD 54](#_Toc138746784)

# **DAFTAR TABEL**

[Tabel 3.1Tabel Kebutuhan Hardware 31](#_Toc138746785)

[Tabel 3.2 Tabel Kebutuhan Software 32](#_Toc138746786)

[Tabel 3.3 Tabel Penjelasan Use Case Diagram 33](#_Toc138746787)

[Tabel 3.4 Penjelasan ERD 54](#_Toc138746788)

# **DAFTAR LAMPIRAN**

# **BAB I PENDAHULUAN**

## Latar Belakang

Di era digitalisasi ini, pemanfaatan teknologi semakin berkembang dengan pesat, tidak hanya untuk memenuhi kebutuhan informasi, tetapi juga untuk meningkatkan efektifitas kehidupan manusia. Teknologi pada saat ini 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 Rochmawan (2008:4) kebutuhan manusia banyak dan beraneka ragam, bahkan tidak hanya beraneka ragam, juga bertambah terus tidak akan ada habisnya sejalan dengan perkembangan peradaban serta kemajuan ilmu pengetahuan dan teknologi. Faktor kebutuhan yang selalu bertambah seiring waktu, maka memaksa manusia untuk tetap bekerja. Bekerja dengan waktu yang padat membuat manusia tidak memiliki waktu yang cukup untuk beristirahat.

Dalam artikel yang dipublikasi oleh Halodoc(2018) mengatakan bahwa pekerja yang memiliki beban kerja berlebih cenderung memiliki tekanan darah tinggi dibanding yang memiliki beban kerja normal (8 jam per hari). Apabila terus-menerus terjadi, tekanan darah tinggi yang dialami dapat menimbulkan risiko penyakit kardiovaskular. Oleh karena itu, manusia harus meluangkan waktu dalam hidupnya untuk beristirahat.

Hal tersebut terjadi karena tuntutan dari tempat kerja, sebagai seorang pekerja tentunya harus taat dengan peraturan kerja. Dalam UU ketenagakerjaan nomor 13 tahun 2003, karyawan dapat mengajukan cuti untuk beristirahat apabila sakit. Dan, perusahaan tetap wajib membayar upah meskipun karyawan tidak masuk kerja. Seringkali, perusahaan mengabaikan peraturan tersebut. Alhasil, karyawan yang sedang dalam kondisi tidak prima tetap melakukan pekerjaan.

Kesehatan adalah hal utama dan suatu hal yang diinginkan oleh semua makhluk hidup di bumi. Karena tubuh yang tidak fit, akan mempengaruhi produktifitas atau bahkan bisa mendapatkan risiko kehilangan nyawa. Dengan segala kepadatan aktivitasnya membuat manusia tidak punya banyak waktu untuk memeriksa kondisi kesehatannya ke dokter. Penanganan yang tepat dan tanggap adalah hal yang terpenting ketika menghadapi tubuh yang sedang sakit. Dengan pemberian penanganan yang tepat dan tanggap, tentunya akan mempercepat proses pemulihan kondisi tubuh.

Untuk memfasilitasi masyarakat agar selalu memperhatikan kesehatannya di sela kesibukannya, Penulis memiliki kesempatan untuk membangun aplikasi pelayanan kesehatan dan konsultasi masyarakat berbasis web. Dengan aplikasi ini tentunya memudahkan masyarakat untuk mendapatkan pelayanan kesehatan dengan tepat dan tanggap.

Dalam inovasi ini, penulis akan menghadirkan beragam fitur unggulan bagi masyarakat seperti; konsultasi melalui *chat* dengan dokter serta perawat yang dapat diakses melalui *smartphone*, Pengguna aplikasi juga dapat melihat apotek dan rumah sakit dengan lokasi yang terdekat, Artikel kesehatan yang dapat menambah wawasan serta ilmu pengetahuan bagi pembaca, dan toko kesehatan yang menyediakan produk berupa obat dan vitamin.

## Rumusan Masalah

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

1. Bagaimana cara agar masyarakat mendapatkan pelayanan kesehatan tanpa meluangkan banyak waktu?
2. Bagaimana cara agar masyarakat dapat konsultasi dengan dokter tanpa harus menjumpai rumah sakit?
3. Bagaimana cara agar masyarakat mendapatkan perawatan tanpa mengunjungi rumah sakit?
4. Bagaimana cara agar masyarakat menambah ilmu pengetahuan tentang kesehatan dengan cepat dan kredibel?
5. Bagaimana cara agar masyarakat mengetahui lokasi terdekat apotek dan rumah sakit ?
6. Bagaimana cara agar masyarakat mendapatkan obat dan vitamin dengan efektif?

## Batasan Masalah

## Tujuan

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

1. Masyarakat dapat melakukan konsultasi langsung dengan dokter spesialis melalui chat.
2. Masyarakat yang membutuhkan rawat jalan dapat melakukan perawatan oleh perawat yang tersedia dalam aplikasi tanpa harus mengunjungi klinik dan rumah sakit.
3. Masyarakat dapat mengetahui apotek dan rumah sakit terdekat berdasarkan lokasinya.
4. Masyarakat dapat melakukan pemesanan obat dan vitamin melalui aplikasi.
5. Masyarakat dapat membaca artikel kesehatan untuk menambah wawasan dan juga ilmu pengetahuan.

## Manfaat

Manfaat yang didapatkan dari pembuatan Aplikasi Reservasi Pelayanan Kesehatan dan Konsultasi Masyarakat Berbasis Web Menggunakan *Framework* VueJS ini adalah:

1. Masyarakat dapat dengan mudah mengakses pelayanan kesehatan tanpa perlu meluangkan banyak waktu. Mereka dapat mengakses aplikasi melalui smartphone mereka kapan saja dan di mana saja.
2. Melalui fitur konsultasi chat, masyarakat dapat dengan mudah berkonsultasi dengan dokter tanpa harus pergi ke rumah sakit. Mereka dapat mengajukan pertanyaan, membagikan keluhan, dan mendapatkan saran medis langsung melalui aplikasi.
3. Masyarakat yang membutuhkan perawatan jarak jauh dapat mendapatkan perawatan yang mereka butuhkan melalui aplikasi. Mereka dapat berkomunikasi dengan perawat yang tersedia dalam aplikasi tanpa harus mengunjungi klinik atau rumah sakit secara fisik.
4. Aplikasi ini menyediakan informasi tentang lokasi terdekat apotek dan rumah sakit. Masyarakat dapat dengan mudah mengetahui apotek terdekat untuk membeli obat dan rumah sakit terdekat untuk mendapatkan perawatan yang diperlukan.
5. Masyarakat dapat melakukan pemesanan obat dan vitamin melalui aplikasi dengan efektif. Mereka dapat memilih produk yang mereka butuhkan, melakukan pembayaran, dan mendapatkan pengiriman obat ke lokasi mereka.
6. Aplikasi ini menyediakan artikel kesehatan yang dapat meningkatkan pengetahuan dan wawasan masyarakat tentang kesehatan. Mereka dapat membaca artikel-artikel ini untuk mendapatkan informasi terbaru tentang kondisi kesehatan, gaya hidup sehat, dan tips pencegahan penyakit.

## Sistematika Penulisan

# **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 web 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.

## Vue.js

Pada pemrograman *javascript* terdapat sejumlah kerangka kerja (framework) yang dapat digunakan, salah satunya adalah vue.js. Vue.js merupakan sebuah kerangka kerja bersifat progresif yang digunakan untuk membangun antarmuka bagi pengguna. Berbeda dengan *framework* javascript lainnya, vue.js dirancang dari dasar agar dapat digunakan secara bertahap sesuai dengan sifatnya yaitu progresif. Vue.js difokuskan pada tampilan saja dan mudah untuk diintegrasikan dengan kerangka kerja lain atau proyek lain. Vue.js juga mendukung *single page application* ketika dikombinasikan dengan alat yang modern dan dukungan *library.*

## Metode Scrum

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.

Sprint merupakan inti dari metode Scrum yang merupakan batasan waktu yang dalam 1 bulan atau kurang dimana sebuah ikremen yang selesai, berfungsi dan berpotensi untuk dikembangkan. Proses sprint biasanya memiliki durasi waktu yang konsisten. Jika proses sprint tahap pertama sudah selesai maka dilanjutkan dengan proses sprint selanjutnya.

Tahapan SCRUM ini terdiri dari product log, sprint backlog, sprint, working increment of the software. Tahapan – tahapan penelitian dalam metode agile SCRUM ini dibagi menjadi beberapa tahapan, yaitu:

1. *Product Backlog*

Merupakan sebuah list daftar dari kebutuhan – kebutuhan yang perlu diketahui dan dipahami. Pada tahapan ini dilakukan pendefinisian dari kebutuhan pelaku bisnis, pendeskripsian kebutuhan.

1. *Sprint Backlog*

Merupakan daftar – daftar dari backlog yang dipilih sesuai dengan kebutuhan dari pelaku usaha agar dapat mencapai sprint goal. Pada tahapan ini pekerjaan sprint backlog dilakukan oleh peneliti dan pelaku usaha agar pembangunan aplikasi dapat mencapai sprint goal dan memastikan agar ada peningkatan dan perbaikan berkelanjutan.

1. *Sprint*

Dilakukan untuk memonitoring pekerjaan – pekerjaan dalam product backlog.

1. *Working increment of the software*

Pada tahapan ini adalah proses penyelesaian sprint, dan pada tahapan ini bisa digunakan oleh pelaku usaha.

## 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* |

### *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 |
|  | *Exit* |  |
|  | *Swimlane* | *Swimlane* memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi |

## *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):

* + - 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 web 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.

## LeafletJS

Leaflet merupakan *Open source* javascript *library* untuk membuat peta yang bersahabat dan interaktif, yang di mana memiliki banyak fitur dalam pembuatan peta. Leaflet yang dibuat dengan simpel , performa dan penggunaan yang mudah. Leaflet dibuat agar dapat digunakan untuk banyak sistem baik dalam desktop maupun *mobile*, yang di mana menggunakan HTML 5 dan CSS3 pada modern web browser dan dapat digunakan untuk versi lama juga.

## *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 web 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 web 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. 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 Owner Apotek

Kebutuhan Owner Apotek dalam *member*ikan pelayanan kesehatan dan mengoptimalkan Apotek yang dimilikinya yaitu:

* + - 1. Melakukan login ke dalam aplikasi.
      2. Melakukan ubah profil akun.
      3. Melakukan ubah kata sandi akun.
      4. Melihat data kategori produk.
      5. Mengelola apotek yang dimiliki seperti menambah, mengubah, dan menghapus apotek.
      6. Mengelola data produk yang ada di dalam apotek.
      7. Mengaktifkan dan men-nonaktifkan status apotek.
      8. Mengelola data *grouping* produk.

1. Kebutuhan Owner Rumah Sakit

Dalam meningkatkan juga menunjang pelayanan kesehatan rumah sakit, owner rumah sakit memiliki kebutuhan yaitu:

Melakukan login ke dalam aplikasi.

Melakukan ubah profil akun.

Melakukan ubah kata sandi akun.

Melihat data nama spesialis yang tersedia dalam aplikasi.

Mengelola data rumah sakit yang dimiliki.

Mengelola data fasilitas rumah sakit yang dimiliki.

Mengelola data spesialis rumah sakit yang dimiliki.

Mengelola jadwal praktek dokter di rumah sakit.

Mengelola antrian *member* di rumah sakit.

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. Mengelola data artikel kesehatan.
5. Mengelola data kategori artikel kesehatan.
6. Mengelola data *grouping* artikel kesehatan.
7. Mengelola keahlian yang dimiliki.
8. Memasukkan nominal harga untuk biaya konsultasi chat.
9. 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. Mengelola keahlian yang dimiliki.
5. Memasukkan nominal harga untuk biaya konsultasi chat.
6. Kebutuhan Administrator Aplikasi
7. Melihat jumlah konsumen, dokter, perawat, owner rumah sakit, dan owner apotek.
8. Mengelola master data kategori artikel kesehatan.
9. Mengelola data artikel kesehatan.
10. Mengelola data *grouping* artikel kesehatan.
11. Melihat data rumah sakit berupa nama, alamat, dan nama pemilik rumah sakit.
12. Melihat data fasilitas rumah sakit.
13. Melihat data spesialis rumah sakit.
14. Melihat data dokter spesialis di dalam rumah sakit.
15. Mengaktifkan status akun owner rumah sakit.
16. Melihat dokumen pendukung pendaftaran akun.
17. Mengaktifkan status akun owner apotek.
18. Mengaktifkan status akun perawat.
19. Mengaktifkan status akun dokter.
20. Melihat data konsumen berupa nama, NIK, alamat, dan nomor telepon.
21. Mengelola master data kategori produk apotek.
22. Mengelola master data spesialis penyakit.
23. Melihat data dari keahlian dokter.
24. Melihat data dari keahlian perawat.
25. Melihat data apotek berupa nama, nomor telepon, dan alamat.
26. Melihat data produk yang tersedia dalam apotek.
27. Melihat data *grouping* produk.

### 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 web, yaitu:

1. Aplikasi dapat dijalankan pada *web browser.*
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 web 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 web untuk analisis sentimen ini terdapat pada tabel 3.2.

Tabel 3.2 Tabel Kebutuhan Software

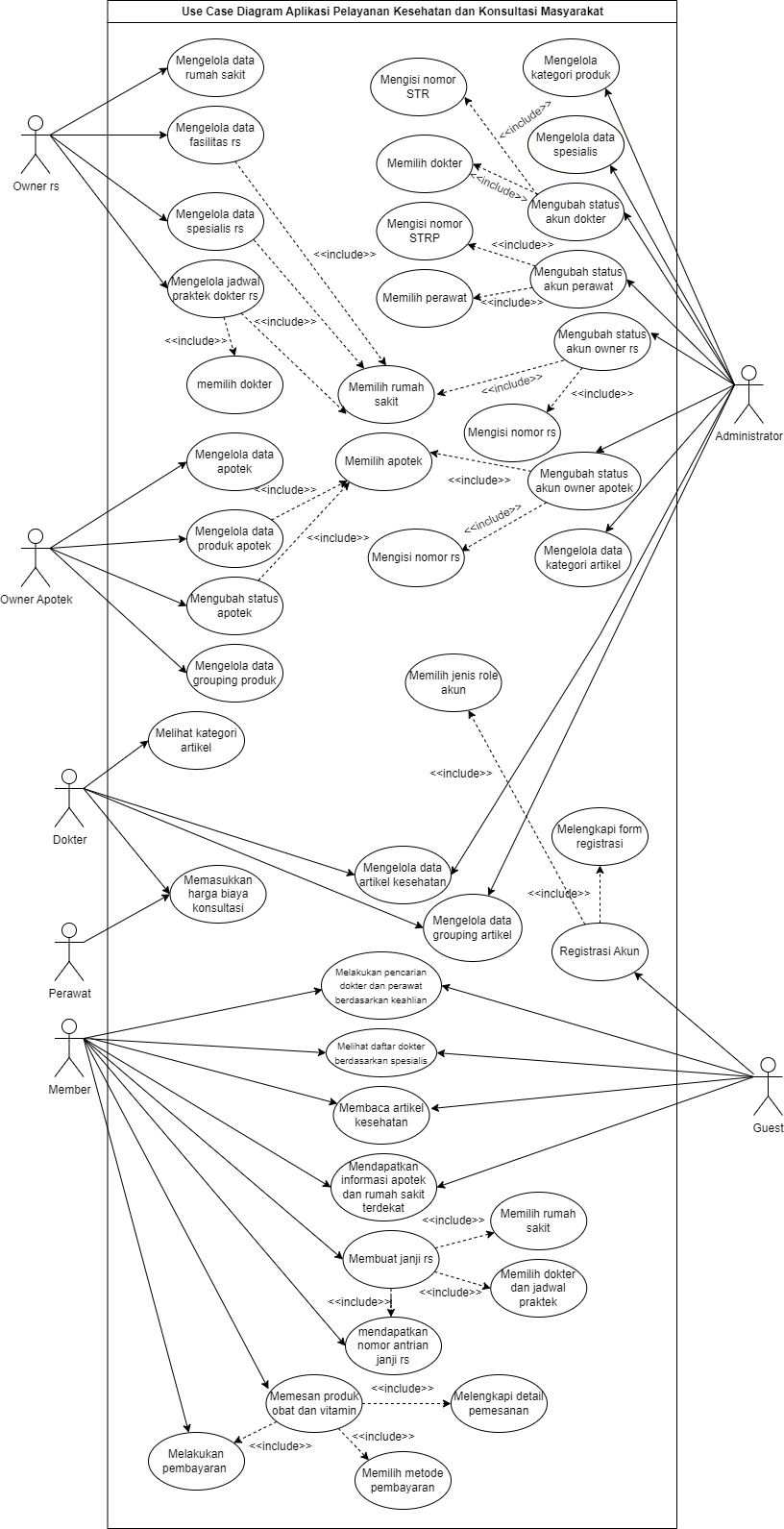
|  |  |  |
| --- | --- | --- |
| **No** | **Jenis *Software*** | **Kebutuhan *Software*** |
| 1. | Bahasa *Scripting* | PHP, JavaScript, CSS, VueJs |
| 2. | *Software* Pengolah | VsCode |
| 3. | Penyimpanan Data | Basis data (MySql) |
| 4. | Web Browser | Google Chrome, Microsoft Edge |
| 5 | Sistem Operasi | Windows 10 |

## Perancangan Sistem

Perancangan sistem akan memaparkan mengenai rancangan dalam proses membangun aplikasi pelayanan kesehatan dan kesehatan masyarakat berbasis web. 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 3.2.



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** |
| Owner Rumah Sakit | Mengelola data rumah sakit | Menu ini digunakan oleh owner rumah sakit untuk melihat, menambahkan, mengubah, dan menghapus data rumah sakit |
| Mengelola data fasilitas rumah sakit | Menu ini digunakan oleh owner rumah sakit untuk melihat, menambahkan, mengubah, dan menghapus data fasilitas yang tersedia di rumah sakit. Untuk mengelolanya, aktor perlu memilih rumah sakit terlebih dahulu. |
| Mengelola data spesialis rumah sakit | Menu ini digunakan oleh owner rumah sakit untuk melihat, menambahkan, mengubah, dan menghapus data spesialis di rumah sakit. Untuk mengelolanya, aktor perlu memilih rumah sakit terlebih dahulu. |
| Mengelola jadwal praktek dokter rumah sakit. | Menu ini digunakan oleh owner rumah sakit untuk melihat, menambahkan, mengubah, dan menghapus data praktek dokter di rumah sakit. Untuk mengelolanya, aktor perlu memilih rumah sakit terlebih dahulu. |
| Owner Apotek  ***Actor*** | Mengelola data apotek | Menu ini digunakan oleh owner apotek untuk melihat, menambahkan, mengubah, dan menghapus data apotek. |
| Mengelola data produk apotek | Menu ini digunakan oleh owner apotek untuk melihat, menambahkan, mengubah, dan menghapus data produk di apotek. Untuk mengelolanya, aktor perlu memilih apotek terlebih dahulu. |
| Mengubah status apotek | Menu ini digunakan oleh owner apotek untuk mengubah status dari apotek yang dimilikinya. |
| Mengelola data *grouping* produk  **Nama *Use Case*** | Menu ini digunakan oleh owner apotek untuk melihat, menambahkan, mengubah, dan menghapus data *grouping* produk.  **Keterangan** |
| Dokter | Melihat kategori artikel | Menu ini digunakan oleh dokter untuk melihat kategori artikel |
| Mengelola data artikel kesehatan | Menu ini digunakan oleh dokter untuk melihat, menambahkan, mengubah, dan menghapus data artikel. |
| Mengelola *grouping* artikel | Menu ini digunakan oleh dokter untuk melihat, menambahkan, mengubah, dan menghapus data *grouping* artikel. |
| Menetapkan harga konsultasi | Menu ini digunakan oleh dokter untuk menetapkan harga konsultasi dokter. |
| Perawat | Menetapkan harga konsultasi | Menu ini digunakan oleh perawat untuk menetapkan harga konsultasi perawat. |
| *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. |
| ***Actor*** | **Nama *Use Case*** | **Keterangan** |
| *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. |
| Admin  ***Actor*** | Mengelola kategori produk | Menu ini digunakan oleh admin untuk menambahkan, melihat, mengubah, dan menghapus data kategori produk. |
| Mengelola data artikel kesehatan | Menu ini digunakan oleh admin untuk melihat, menambahkan, mengubah, dan menghapus data artikel. |
| Mengelola *grouping* artikel | Menu ini digunakan oleh admin untuk melihat, menambahkan, mengubah, dan menghapus data *grouping* artikel. |
| Mengelola kategori artikel | Menu ini digunakan oleh admin untuk menambahkan, melihat, mengubah, dan menghapus data kategori artikel. |
| Mengelola data spesialis | Menu ini digunakan oleh admin untuk menambahkan, melihat, mengubah, dan menghapus data spesial. |
| Mengubah status akun dokter  **Nama *Use Case*** | Menu ini digunakan oleh admin untuk mengaktifkan akun dokter. Admin wajib validasi dokumen yang diunggah. Serta, menambahkan nomor STR.  **Keterangan** |
| Admin | Mengubah status akun perawat | Menu ini digunakan oleh admin untuk mengaktifkan akun perawat. Admin wajib validasi dokumen yang diunggah. Serta, menambahkan nomor STRP. |
| Mengubah status akun owner apotek | Menu ini digunakan oleh admin untuk mengaktifkan akun owner apotek. Admin wajib validasi dokumen yang diunggah. Serta, menambahkan nomor apotek. |
| Mengubah status akun owner rumah sakit | Menu ini digunakan oleh admin untuk mengaktifkan akun owner rumah sakit. Admin wajib validasi dokumen yang diunggah. Serta, menambahkan nomor rs. |

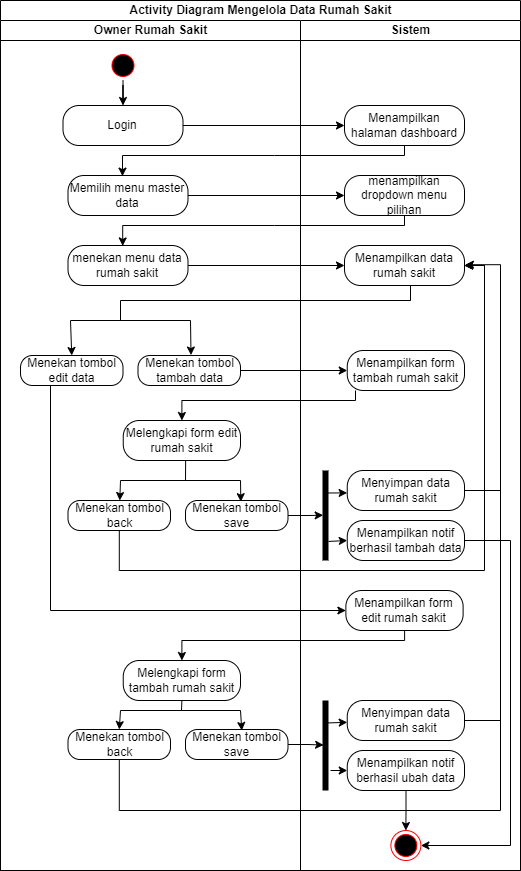
### Activity Diagram

*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 web menggunakan *framework* vuejs. Berikut adalah *activity* Diagram:

*Activity* DiagramMengelola Data Rumah Sakit

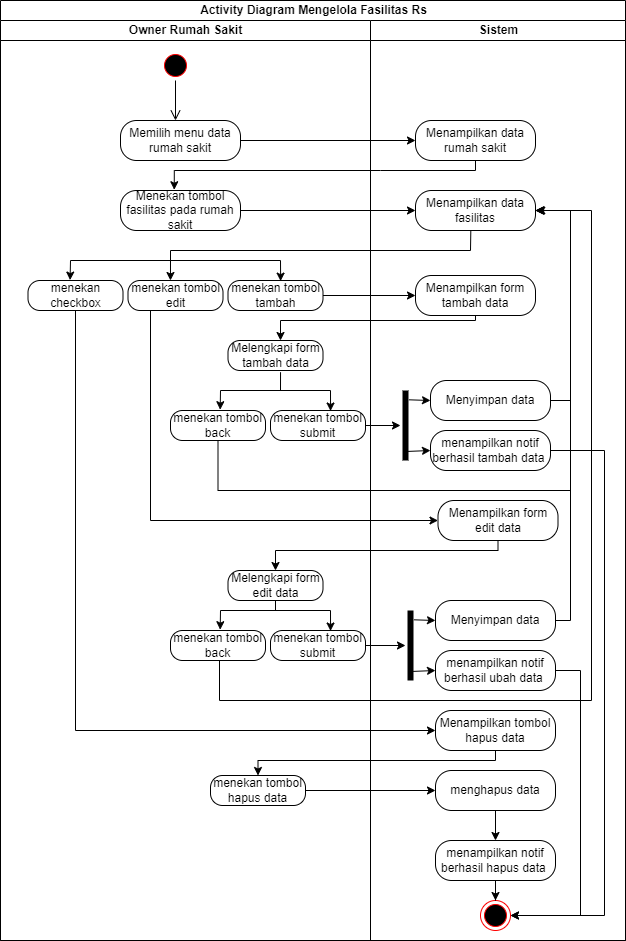
*Activity* Diagrammengelola data rumah sakit menggambarkan urutan aktivitas yang dilakukan oleh owner rumah sakit agar dapat mengelola data rumah sakit. Adapun *activity* Diagrammengelola data rumah sakit dapat dilihat pada Gambar 3.2. Pada *activity* Diagramini dimulai dari melakukan login, lalu akan menampilkan halaman dashboard admin. Selanjutnya, memilih menu master, sesudahnya memilih menu data rumah sakit. Lalu, akan menampilkan data rumah sakit, aktor owner rumah sakit akan dihadapkan oleh opsi tambah dan edit data. Jika menekan tambah data, sistem akan menampilkan form tambah data. Kemudian, owner rumah sakit akan melengkapi form, jika menekan submit, maka data akan tersimpan dan jika menekan tombol *back* maka sistem akan mengembalikan ke halaman data rumah sakit.



Gambar 3.2 Activity Diagram Mengelola Data Rumah Sakit

*Activity* Diagram Mengelola Fasilitas Rumah Sakit

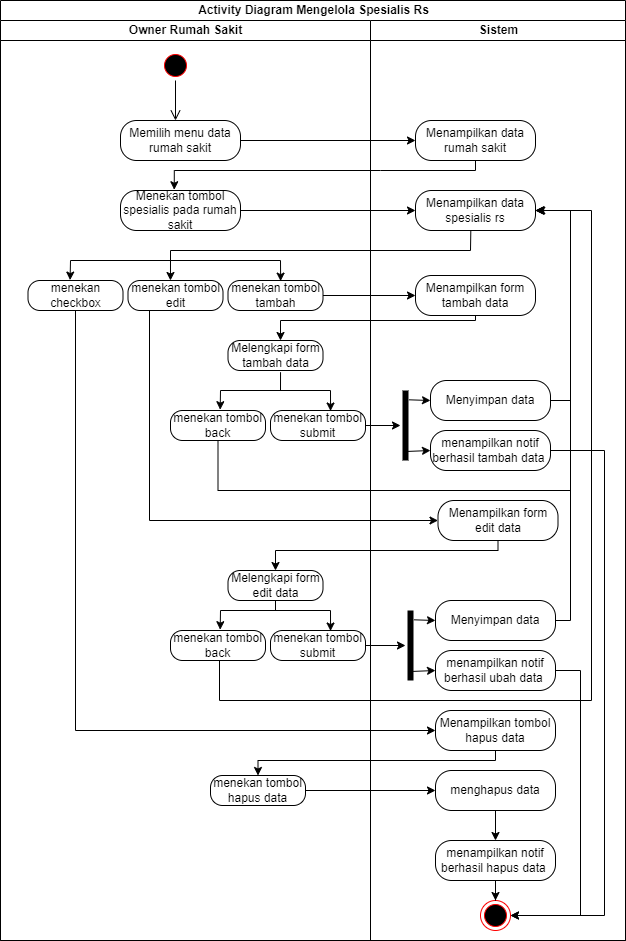
*Activity* Diagram ini menggambarkan alur sistem dalam mengelola fasilitas rumah sakit. Adapun penjelasan *activity* ini dapat dilihat pada Gambar 3.3. aktivitas yang dilakukan untuk pertama adalah memilih menu data rumah sakit, lalu sistem akan *member*ikan reaksi yaitu menampilkan data rumah sakit. Pada tabel data rumah sakit terdapat tombol fasilitas rumah sakit, sesudahnya jika di-klik maka akan menampilkan data fasilitas rumah sakit. Pada halaman fasilitas rumah sakit, terdapat tombol tambah, edit, dan hapus data. Jika menekan tombol tambah data akan menampilkan form tambah data, aktor diwajibkan untuk melengkapi form tambah data, lalu menekan tombol *submit* maka data akan tersimpan dalam sistem dan menampilkan notif berhasil tambah data. Pada proses edit data, prosesnya hampir sama dengan tambah data. Jika ingin menghapus data, owner rumah sakit perlu menekan *checkbox* maka akan menampilkan tombol hapus data. Jika menekan tombol hapus data, maka data yang dipilih akan terhapus dari sistem.



Gambar 3.3 Activity Diagram Mengelola Fasilitas Rumah Sakit

*Acitivity* Diagram Mengelola Data Spesialis Rumah Sakit

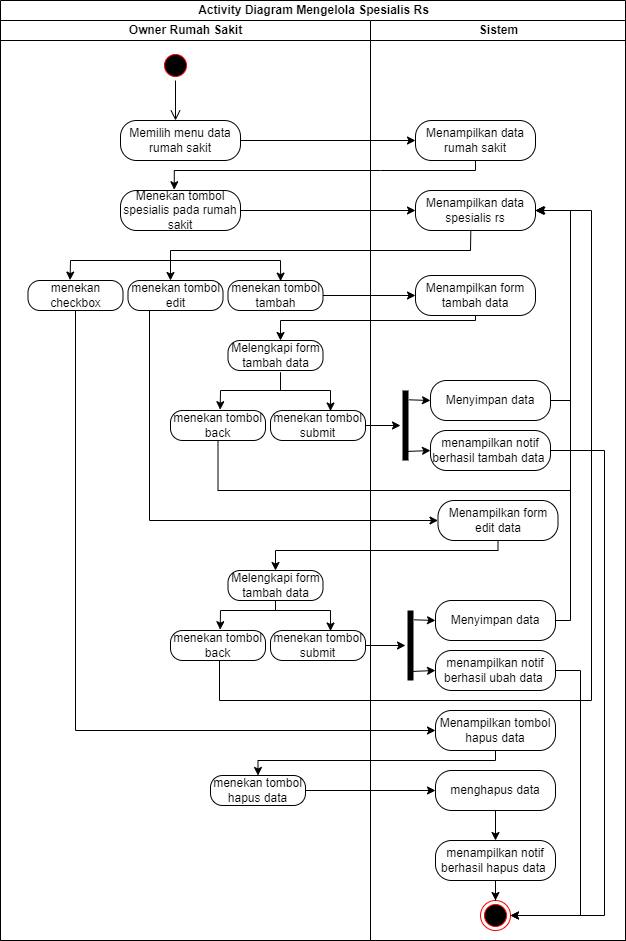
*Activity* Diagram ini menggambarkan alur sistem dalam mengelola spesialis rumah sakit. Adapun penjelasan *activity* ini dapat dilihat pada Gambar 3.4. aktivitas yang dilakukan untuk pertama adalah memilih menu data rumah sakit, lalu sistem akan *member*ikan reaksi yaitu menampilkan data rumah sakit. Pada tabel data rumah sakit terdapat tombol spesialis rumah sakit, sesudahnya jika di-klik maka akan menampilkan data spesialis rumah sakit. Pada halaman spesialis rumah sakit, terdapat tombol tambah, edit, dan hapus data. Jika menekan tombol tambah data akan menampilkan form tambah data, aktor diwajibkan untuk melengkapi form tambah data, lalu menekan tombol *submit* maka data akan tersimpan dalam sistem dan menampilkan notif berhasil tambah data. Pada proses edit data, prosesnya hampir sama dengan tambah data. Jika ingin menghapus data, owner rumah sakit perlu menekan *checkbox* maka akan menampilkan tombol hapus data. Jika menekan tombol hapus data, maka data yang dipilih akan terhapus dari sistem.



Gambar 3.4 Acitivity Diagram Mengelola Data Spesialis Rumah Sakit

*Activity* Diagram Mengelola Jadwal Praktek Dokter Rumah Sakit

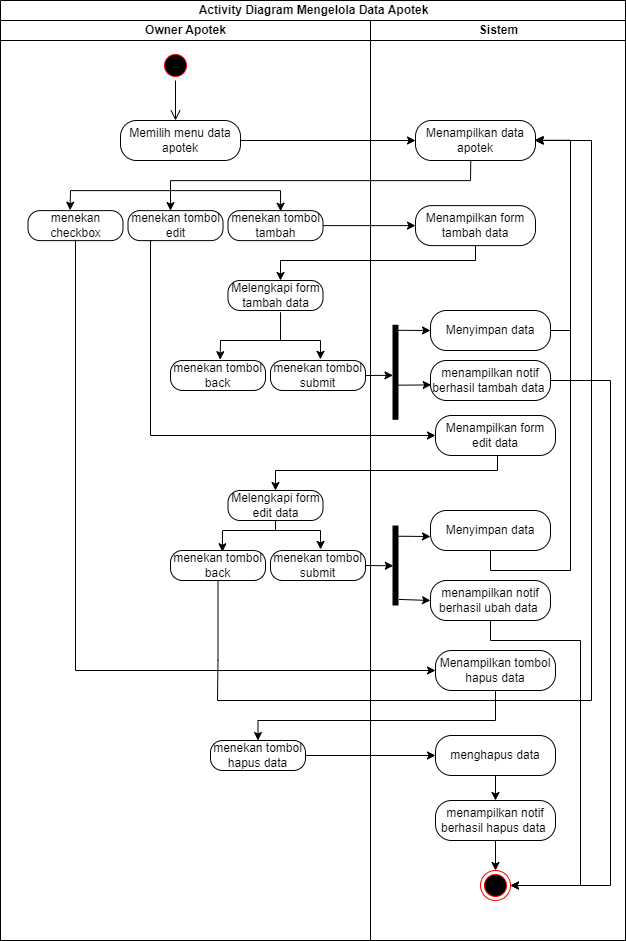
*Activity* Diagram ini menguraikan alur sistem dalam pengelolaan jadwal praktek dokter pada rumah sakit. Adapun *activity* diagram mengelola jadwal praktek dokter rumah sakit dapat dilihat pada Gambar 3.5. Aktivitas pada *activity* diagram ini di awali oleh aktor owner rumah sakit memilih menu rumah sakit, lalu sistem akan menampilkan data rumah sakit. Dalam halaman data rumah sakit, terdapat tombol lihat praktek, jika menekannya maka akan dialihkan ke halaman praktek yang berisi daftar jadwal praktek dokter. Owner rumah sakit dapat menambahkan data praktek di dalam halaman ini dengan menekan tombol tambah data, dan kemudian melengkapi form tambah data, sesudahnya menekan tombol *submit* maka data akan tersimpan di sistem dan menampilkan notif berhasil tambah data.



Gambar 3.5 Activity Diagram Mengelola Jadwal Praktek Dokter Rumah Sakit

1. *Activity* Mengelola Data Apotek

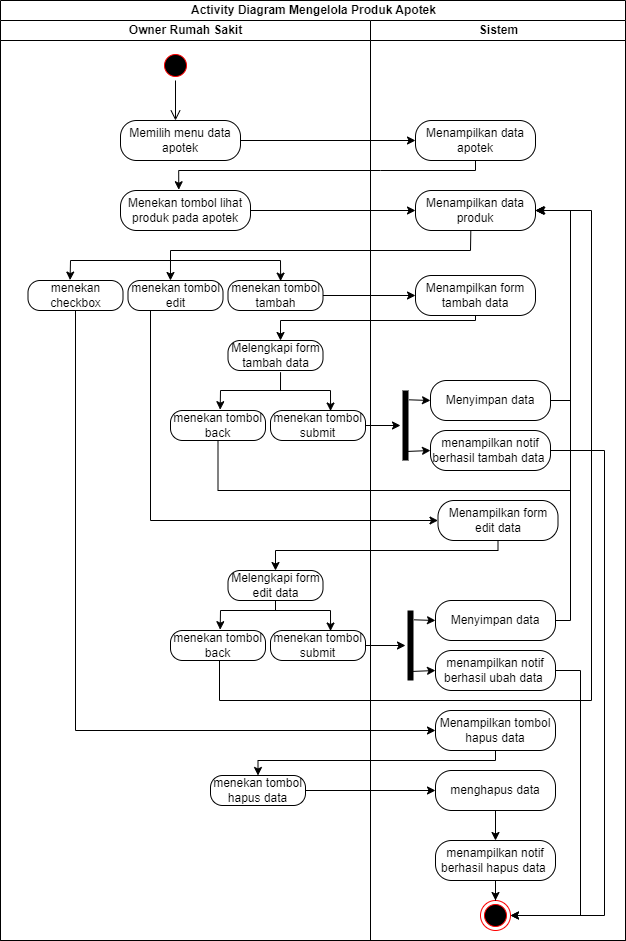
Pada *activity* diagram ini owner apotek mengelola data apotek. Owner apotek memulai aktivitas pertamanya dengan memilih menu apotek, setelahnya sistem akan menampilkan data apotek. Dalam halaman data apotek, terdapat beberapa aksi yang dapat dilakukan seperti menambah, mengubah, dan menghapus data. Untuk menambah data apotek, owner apotek wajib untuk melengkapi form tambah data. Owner apotek dapat menghapus data apotek dengan memilih apotek mana yang ingin dihapus dengan menekan kolom checkbox, maka sistem akan menampilkan tombol hapus data. Jika aktor menekan tombol hapus data, maka sistem akan menghapus data apotek yang sudah dipilih.



Gambar 3.6 Activity Diagram Mengelola Data Apotek

1. *Activity* Diagram Mengelola Data Produk Apotek

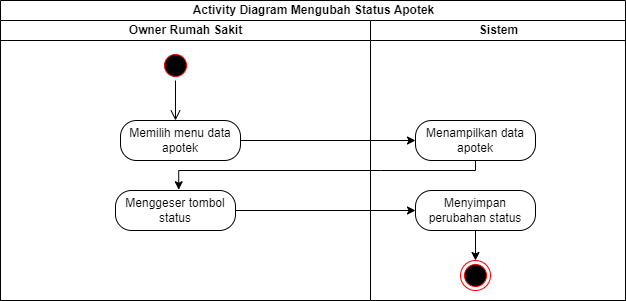
Pada *activity* diagram ini owner apotek mengelola data produk apotek. Owner apotek memilih menu apotek, lalu sistem menampilkan data apotek. Untuk mengelola produk apotek, aktor perlu menekan tombol lihat produk. Sistem akan menampilkan semua produk yang ada di dalam apotek yang di-klik. Jika ingin menambahkan data produk apotek, aktor perlu melengkapi form tambah data. Dan jika ingin menghapus data produk, perlu memilih data yang akan dihapus melalui *checkbox,* sistem akan menampilkan tombol hapus data ketika aktor sudah memilih data produk yang akan dihapus. Adapun *activity* diagram mengelola data produk apotek terdapat dalam Gambar 3.7.



Gambar 3.7 Activity Diagram Mengelola Data Produk Apotek

1. *Activity* Diagram Mengubah Status Apotek

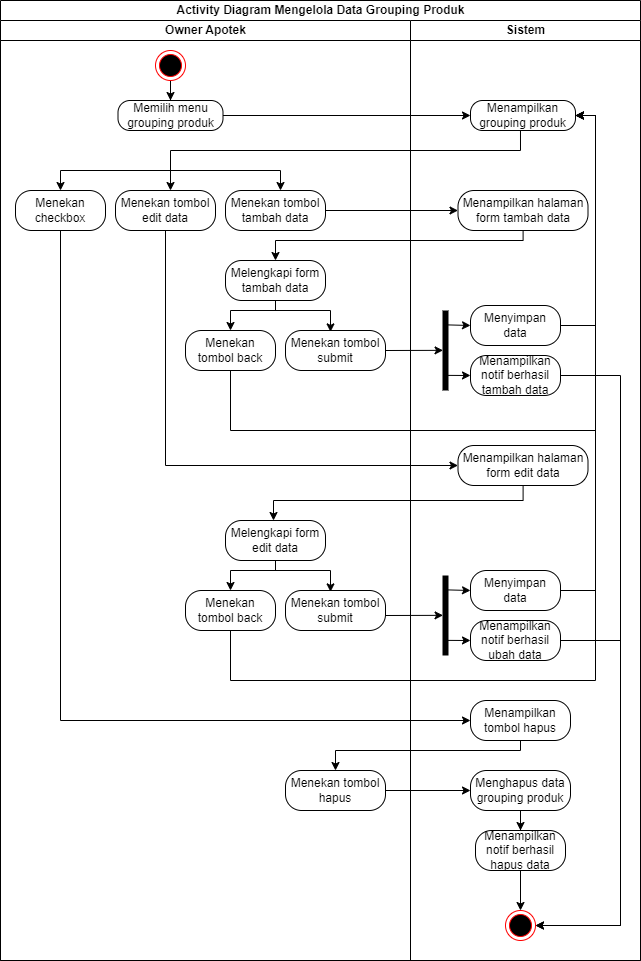
Pada *activity* diagram ini owner apotek dapat melakukan perubahan status apotek. Jika apotek yang dimilikinya sudah tidak aktif, maka owner dapat merubah statusnya. Aktivitas yang dilakukan pertama kali oleh owner apotek adalah memilih menu apotek pada dashboard, sesudahnya sistem akan menampilkan data apotek, lalu menggeser tombol *slider* ke kanan untuk mengaktifkan dan ke kiri untuk menon-aktifkan status apotek. Adapun *activity* diagram mengubah status apotek terdapat dalam Gambar 3.8.



Gambar 3.8 Activity Diagram Mengubah Status Apotek

1. *Activity* Diagram Mengelola Data *Grouping* Produk

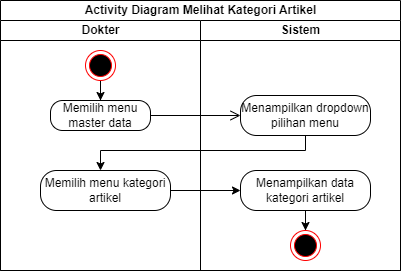
Pada *activity* diagram ini owner apotek melakukan mengelola data *grouping* produk. *Grouping* produk dilakukan untuk mengelompokkan produk dengan kategori produk. Saat menambahkan produk, produk belum memiliki kategori produk. Aktivitas pertama dalam mengelola data *grouping* produk adalah dengan memilih menu *grouping* produk pada apotek. Sistem akan menampilkan keseluruhan data produk dan kategorinya dalam satu tabel. Dalam mengelolanya, aktor dapat melakukan berupa aksi tambah, edit, dan hapus data *grouping* produk. Owner apotek menekan tombol tambah data untuk menampilkan form tambah data, dan kemudian melengkapi form tambah data tersebut, jika sudah maka selanjutnya adalah menekan tombol *submit.* Data akan tersimpan di dalam sistem. Untuk edit, langkahnya hampir sama seperti tambah data. Dalam menghapus data, owner apotek harus menekan *checkbox* untuk menampilkan tombol hapus data. Kemudian dengan menekan tombol hapus data, maka data akan terhapus dari sistem. Adapun *activity* diagram mengelola data produk apotek terdapat dalam Gambar 3.9.



Gambar 3.9 Activity Diagram Mengelola Data Grouping Produk

1. *Activity* Diagram Melihat Kategori Artikel

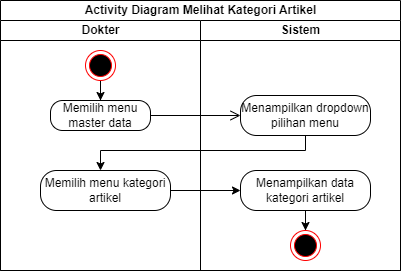
Pada *activity* diagram ini dokter dapat melihat kategori artikel. Dokter dapat melihat kategori artikel dengan melakukan aktivitas memilih menu master data pada dashboard, kemudian sistem akan menampilkan *dropdown* pilihan menu, selanjutnya aktor dokter memilih menu kategori artikel. Sesudahnya sistem akan menampilkan data kategori artikel. Adapun *activity* diagram melihat kategori artikel terdapat dalam Gambar 3.10.



Gambar 3.10 Activity Diagram Melihat Kategori Artikel

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

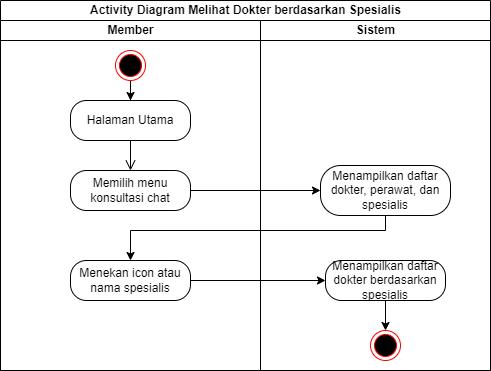
Pada *activity* diagram ini aktor dokter dan perawat menetapkan harga biaya konsultasi. Aktivitas yang pertama dilakukan adalah login aplikasi, sesudahnya memilih menu profil. Lalu pada kolom biaya, lengkapi dengan memasukkan nominal biaya. Adapun *activity* diagram terdapat dalam Gambar 3.11.



Gambar 3.11 Activity Diagram Menetapkan Harga Konsultasi Dokter dan Perawat

1. *Activity* Diagram Melihat Dokter berdasarkan Spesialis

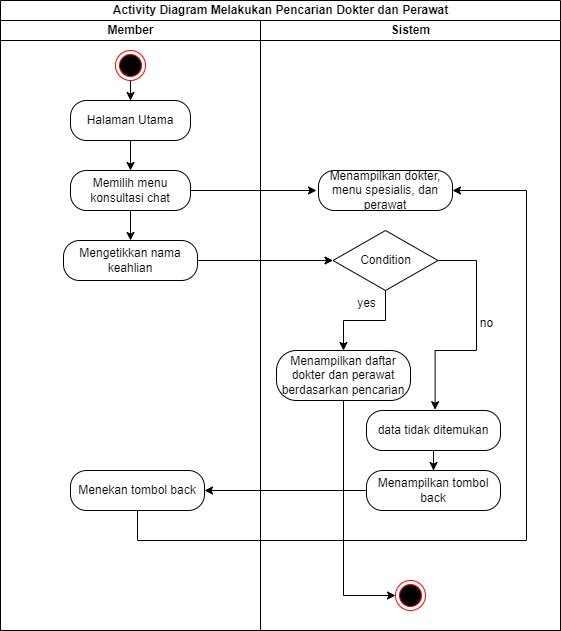
Pada *activity* diagram ini *member* melakukan aksi melihat dokter berdasarkan spesialis. Langkah yang dilakukan adalah dengan masuk ke halaman utama aplikasi, sesudahnya menekan menu konsultasi chat, sistem akan menampilkan daftar dokter, perawat, dan spesialis. Kemudian aktor *member* menekan menu spesialis. Sistem akan menampilkan dokter berdasarkan nama spesialis yang di-klik. Adapun *activity* diagram melihat dokter berdasarkan spesialis dapat dilihat dalam Gambar 3.12.



Gambar 3.12 Activity Diagram Melihat Dokter berdasarkan Spesialis

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

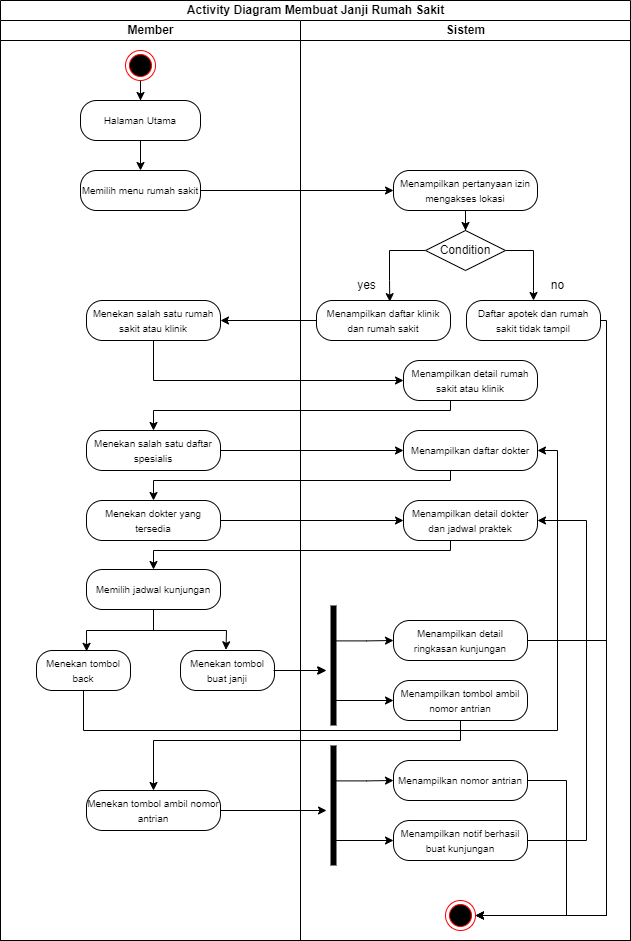
Pada *activity* diagram ini aktor *member* melakukan pencarian nama dokter dan perawat berdasarkan keahlian yang dimilikinya. Dalam aktivitas awalnya, *member* masuk ke halaman utama aplikasi kemudia memilih menu konsultasi chat. Sesudahnya mengetikkan nama keahlian yang hendak dicari. Sistem akan menampilkan 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 3.13.



Gambar 3.13 Activity Diagram Melakukan Pencarian Dokter dan Perawat

1. *Activiy* Diagram Membuat Janji Kunjungan Rumah Sakit

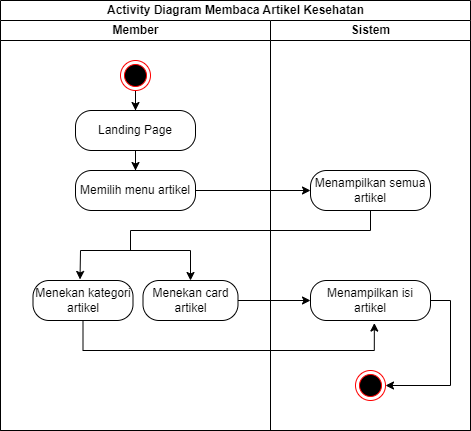
Pada *activity* diagram ini *member* membuat janji kunjungan rumah sakit. Aktivitas pertama yang dilakukan oleh *member* adalah memilih menu rumah sakit, lalu sistem akan menampilkan pertanyaan untuk meminta izin mengakses lokasi *member.* Jika diizinkan mengakses lokasi, maka sistem akan menampilkan 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 3.14 Activity Diagram Membuat Janji Kunjungan Rumah Sakit

1. *Activity* Diagram Membaca Artikel Kesehatan

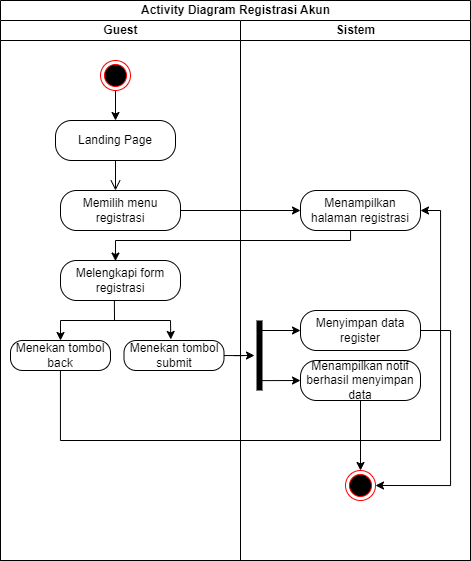
Pada *activity* diagram ini *member* dan *guest* dapat membaca artikel kesehatan. Untuk membaca artikel, aktivitas yang pertama dilakukan adalah masuk ke dalam aplikasi, dapat dengan login terlebih dahulu atau tidak. Pada halaman utama aplikasi, terdapat empat artikel terbaru yang di-*publish.* Aktor 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 3.15.



Gambar 3.15 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 web. 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 3.16.



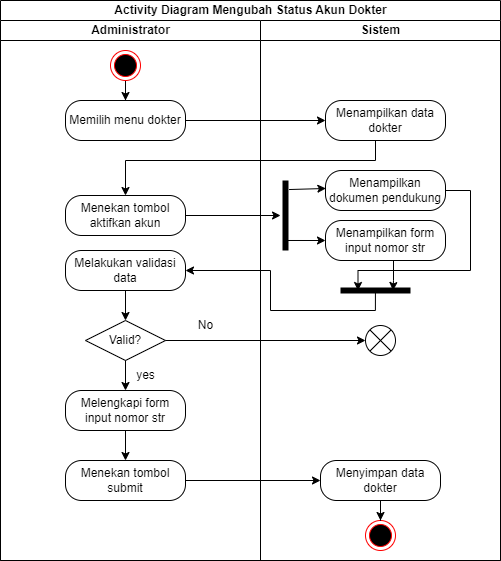
Gambar 3.16 Activity Diagram Registrasi Akun

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 3.17.

1. *Activity* Diagram Mengubah Status Akun Dokter

Pada *activity* diagram ini admin aplikasi melakukan aktivasi akun dokter. Setelah mendaftarkan akun, dokter belum mendapatkan izin untuk mengakses aplikasi pelayanan kesehatan dan konsulatasi masyarakat. Seorang admin perlu memvalidasi data registrasi dokter tersebut melalui menu ubah status akun dalam dashboard admin. Aktivitas yang dilakukan adalah memilih menu dokter, kemudian sistem akan menampilkan data dokter, peran aktor admin adalah validasi dokumen yang diunggah oleh dokter dengan cara melihat dokumen yang di-*submit,* jika datanya valid maka admin akan menginputkan nomor Surat Tanda Registrasi (STR).

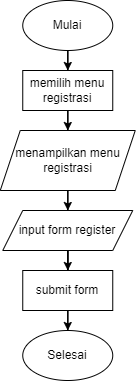


Gambar 3.17 Activity Diagram Mengubah Status Akun Dokter

## Flowchart

### Flowchart Registrasi Akun

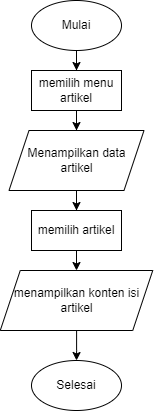
Flowchart berikut merupakan proses untuk melakukan pendaftaran akun. *Guest* akan memilih menu registrasi dan kemudian mengisi form registrasi. Setelahnya di-*submit*. Maka, data registrasi akan tersimpan.



Gambar 3.18 Flowchart Register

### Flowchart Membaca Artikel

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



Gambar 3.19 Flowchart Membaca Artikel

### Flowchart Melihat Rumah Sakit Terdekat

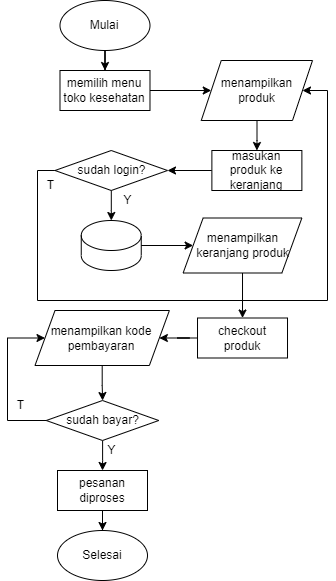
Flowchart berikut merupakan proses untuk melihat rumah sakit terdekat berdasarkan lokasi. Adapun flowchart melihat rumah sakit terdekat dapat dilihat pada gambar 3.20.



Gambar 3.20 Flowchart Melihat Rumah Sakit Terdekat

### Flowchart Memesan Produk Apotek

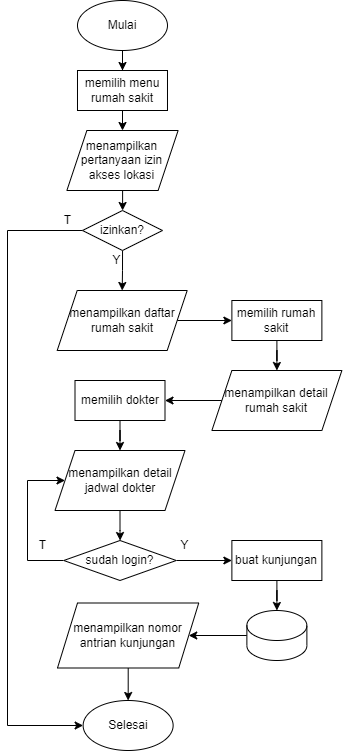
Flowchart berikut merupakan proses untuk memesan produk apotek. Adapun flowchart memesan produk apotek dapat dilihat pada gambar 3.21.



Gambar 3.21 Flowchart Memesan Produk Apotek

### Flowchart Membuat Janji Kunjungan Rumah Sakit

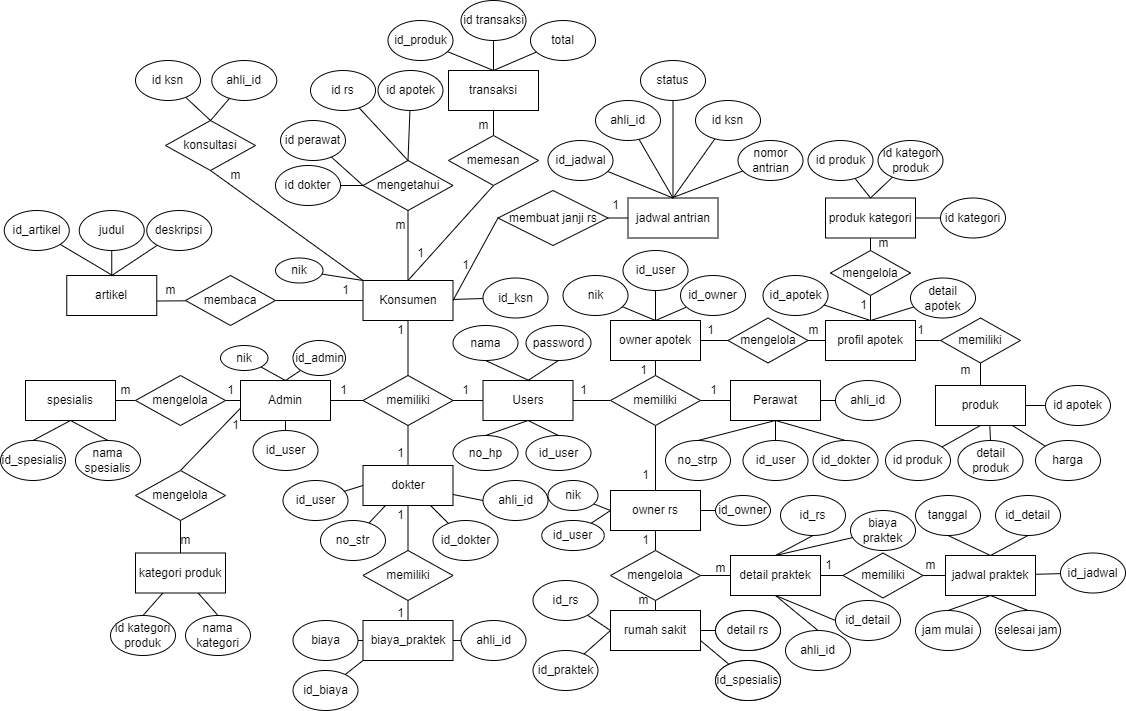
Flowchart berikut merupakan proses untuk membuat janji kunjungan rumah sakit. Adapun flowchart membuat janji kunjungan rumah sakit dapat dilihat pada gambar 3.22.



Gambar 3.22 Flowchart Membuat Janji Kunjungan Rumah Sakit

## Perancangan *Entity Relationship* Diagram (ERD)

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



Gambar 3.23 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. |
| **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. |
| Kategori Produk | Tabel ini berfungsi untuk menyimpan kategori produk |
| Spesialis | Tabel ini berfungsi untuk menyimpan nama spesial. |
| Artikel | Tabel ini berfungsi untuk menyimpan data artikel. |
| 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 web 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

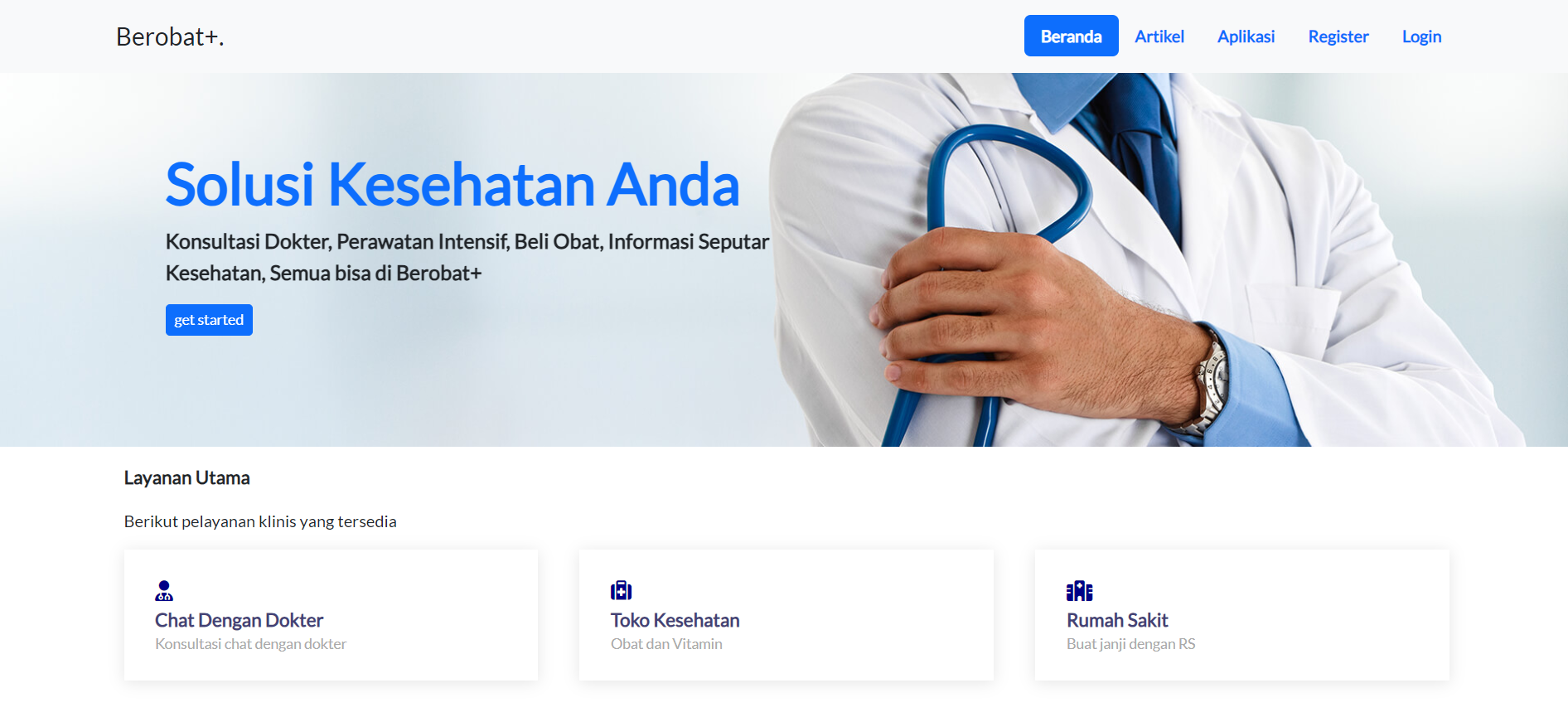
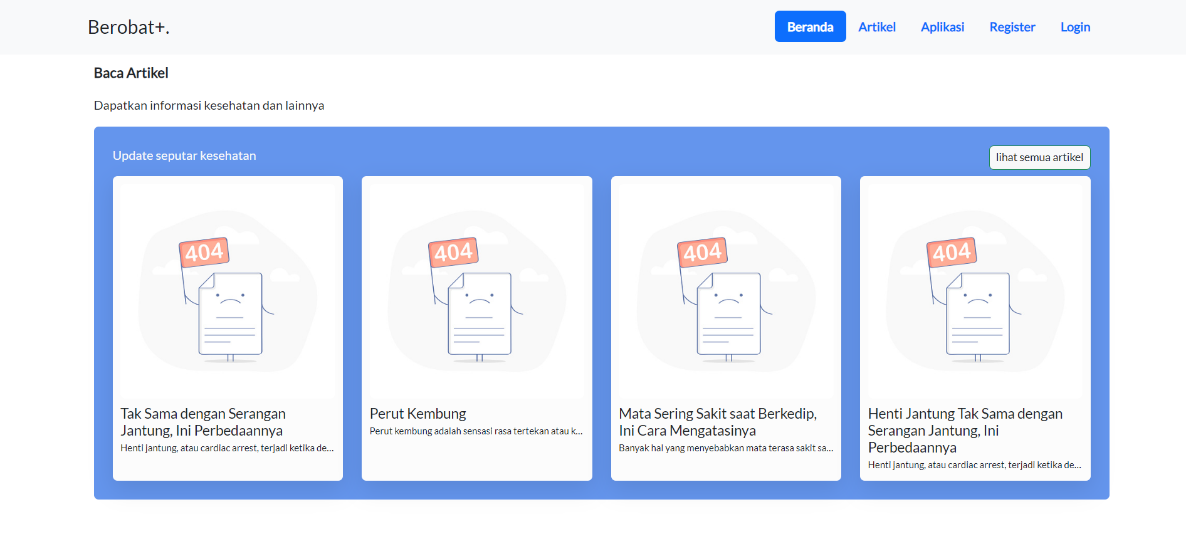
Dalam Rancangan Antarmuka membuat aplikasi web untuk aplikasi pelayanan kesehatan dan konsultasi masyarakat ini dibuat dengan menggunakan *software* Figma. Beberapa rancangan antarmuka dibagi menjadi beberapa bagian, yaitu: rancangan antarmuka *guest, member,* admin, owner rumah sakit, dokter, perawat,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

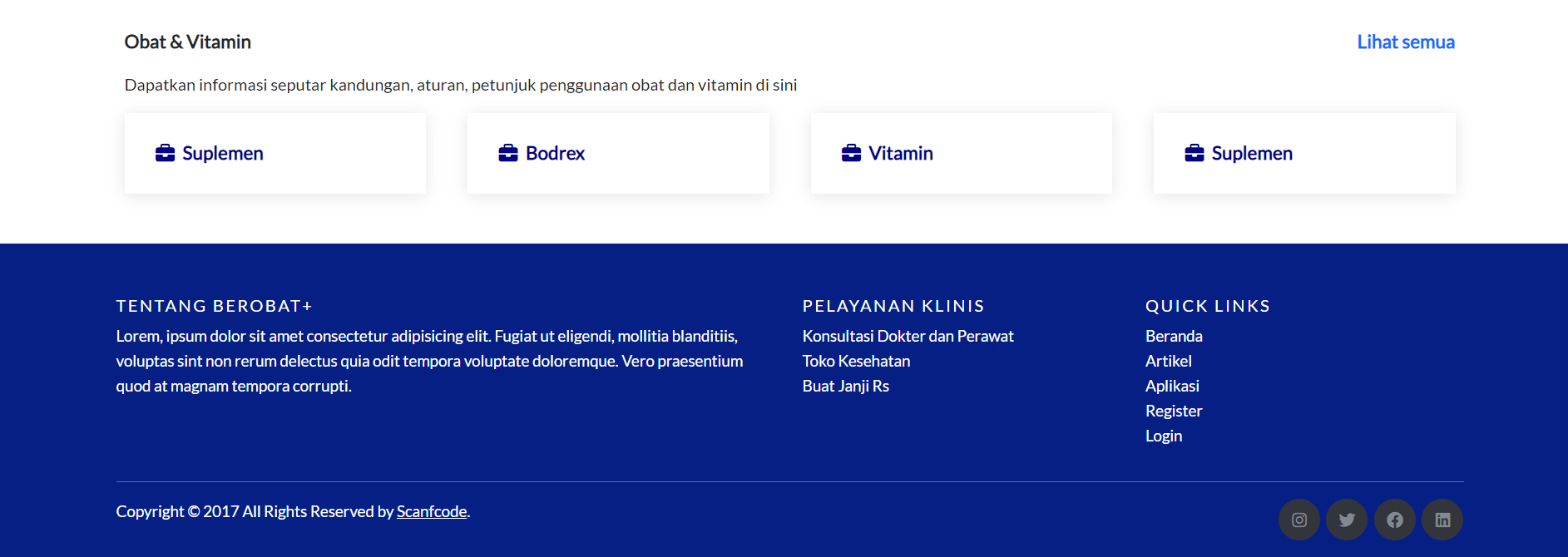
### Rancangan Antarmuka *Guest*

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

1. Rancangan antarmuka halaman *home*

Antarmuka ini merupakan rancangan *user interface* dari sisi *guest,* rancangan *interface* pada halaman *home* dibagi menjadi 3 *section* terdapat layanan utama yaitu konsultasi kesehatandengan dokter atau perawat, toko kesehatan, dan janji kunjungan ke rumah sakit. Adapun rancangan antarmuka halaman *home* dapat dilihat pada gambar 3.19.

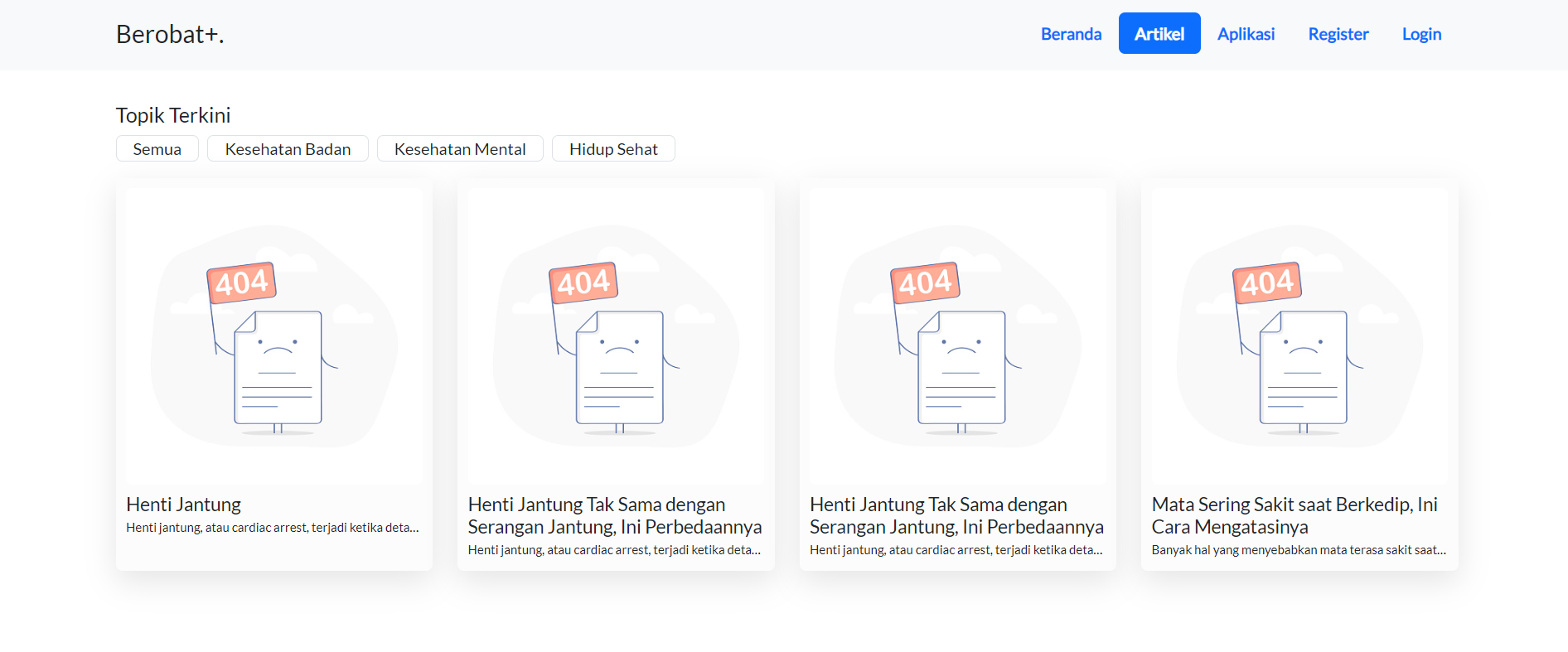




Gambar 3.24 Rancangan Antarmuka Halaman Home

1. Rancangan antarmuka halaman artikel

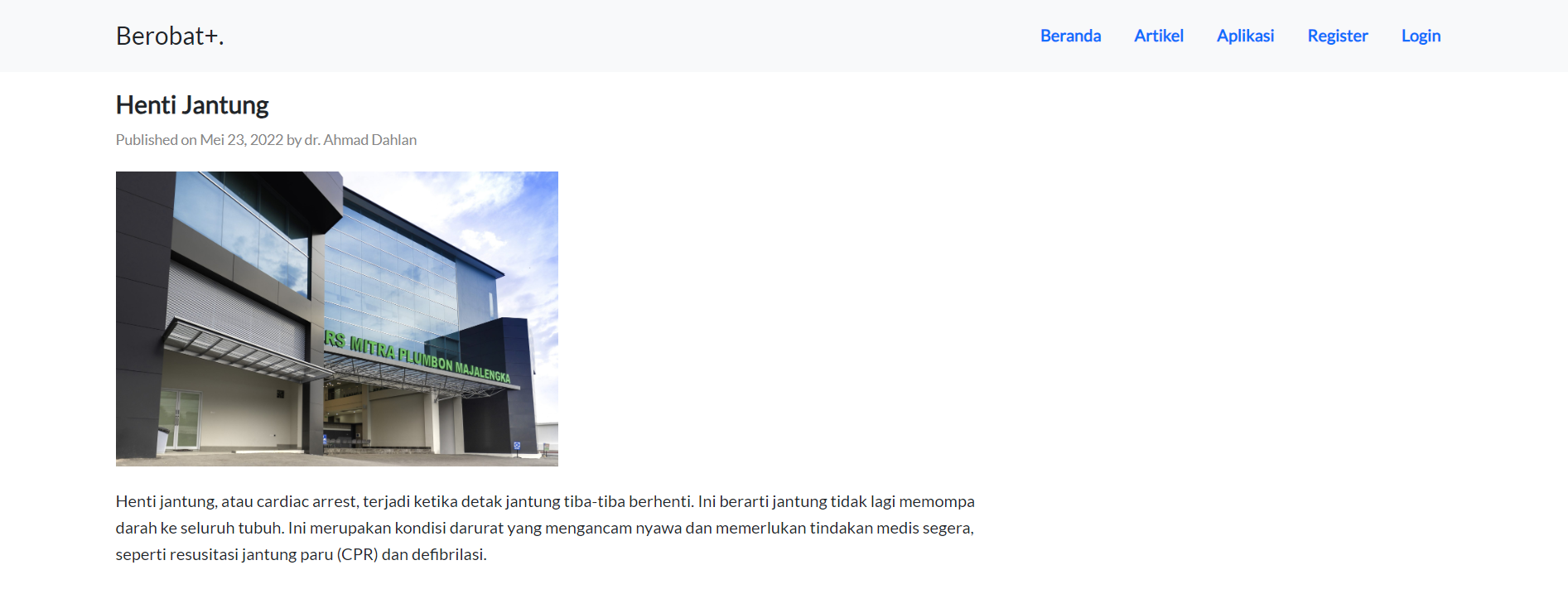
Pada halaman ini terdapat artikel yang dapat dibaca oleh *guest.* Dengan menekan teks artikel pada bagian *navbar,* maka akan menampilkan keseluruhan artikel kesehatan. Rancangan interface halaman artikel terdapat kategori atau topik yang dapat di-klik ketika *guest* hendak membaca artikel berdasarkan kategorinya. Adapun rancangan antarmuka halaman artikel dapat dilihat pada gambar 3.20.



Gambar 3.25 Rancangan Antarmuka Halaman Artikel

1. Rancangan antarmuka halaman detail artikel

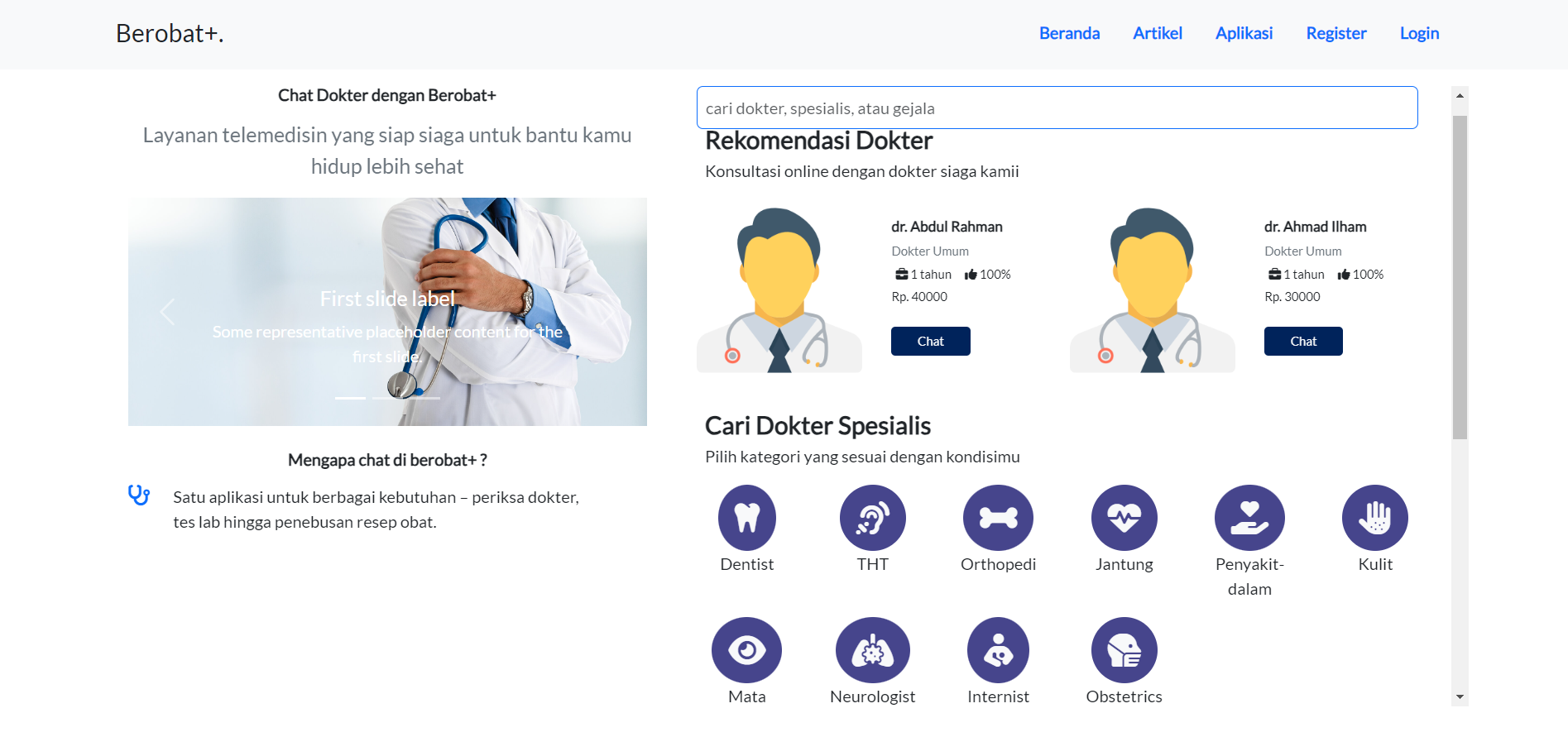
Pada rancangan antarmuka halaman detail artikel terdapat konten dari artikel yang di-klik oleh *guest.* Konten artikel terdapat judul artikel, deskripsi artikel, foto, dan *author*-nya. Adapun rancangan antarmuka halaman detail artikel dapat dilihat pada gambar 3.21.



Gambar 3.26 Rancangan Antarmuka Halaman Detail Artikel

1. Rancangan antarmuka konsultasi kesehatan dokter dan perawat

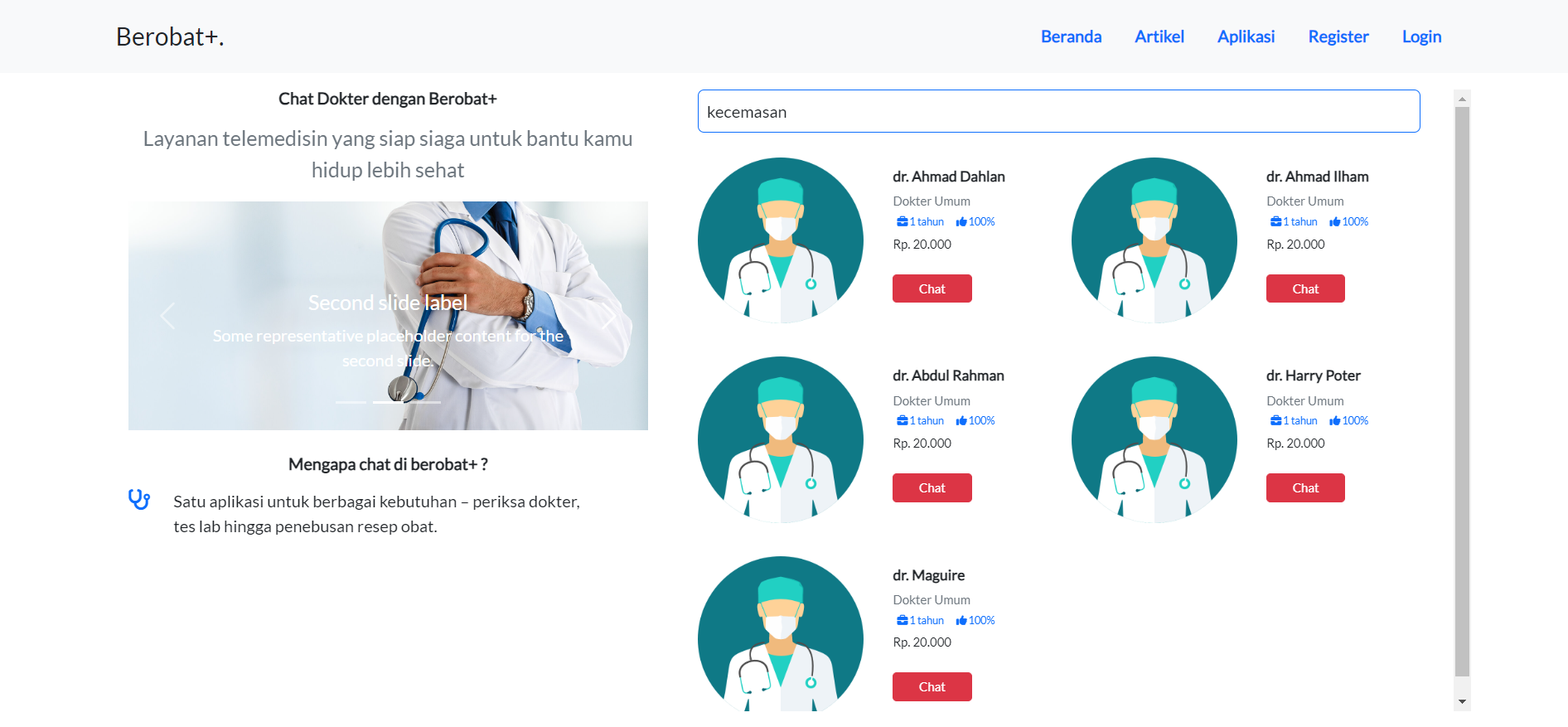
Rancangan antarmuka halaman konsultasi kesehatan dokter dan perawat terdapat konten berupa rekomendasi dokter, dan daftar spesialis yang tersedia pada aplikasi pelayanan kesehatan dan konsultasi masyarakat. Dokter yang ditampilkan pada halaman ini dibatasi yaitu hanya dua dokter. Adapun rancangan antarmuka halaman konsultasi kesehatan dokter dan perawat terdapat pada gambar 3.22.



Gambar 3.27 Rancangan Antarmuka Halaman Konsultasi Dokter dan Perawat

1. Rancangan antarmuka halaman pencarian dokter berdasarkan keahlian

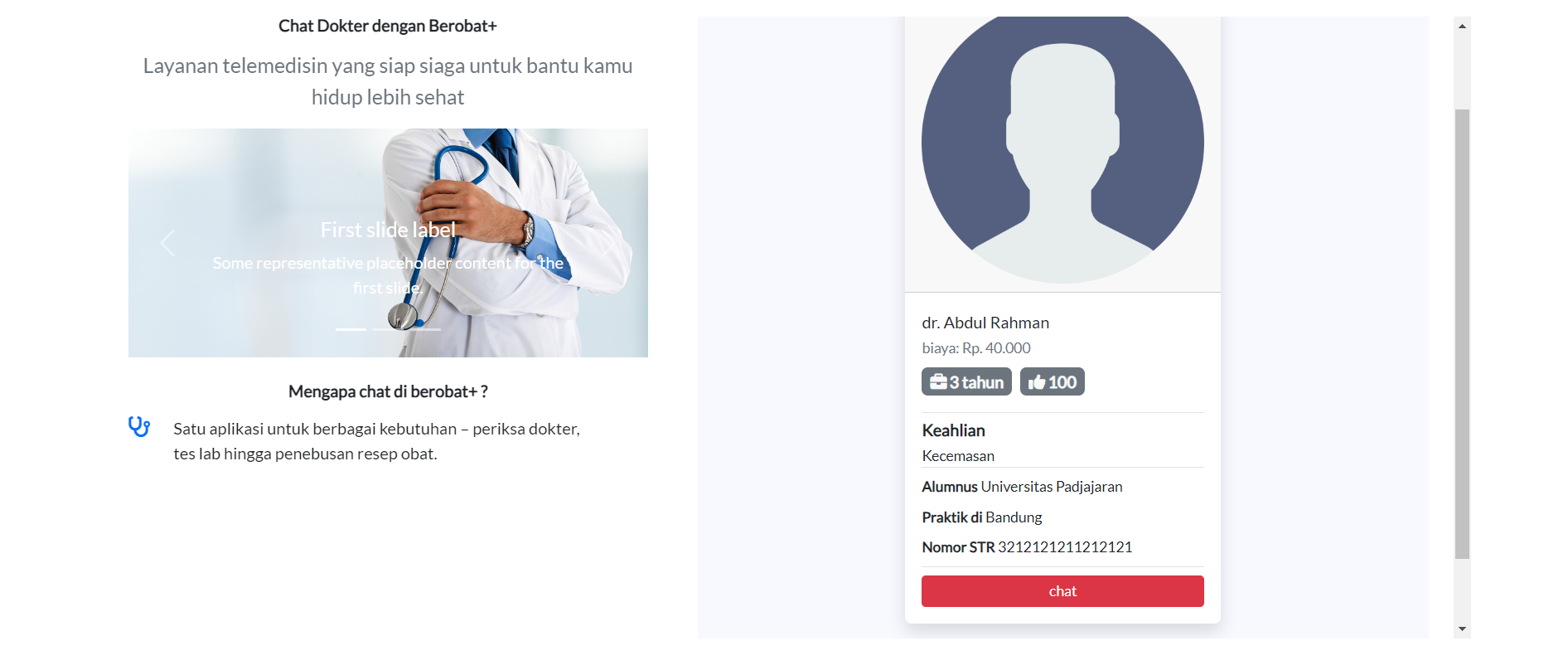
Rancangan *interface* ini merupakan halaman untuk melakukan pencarian dokter dan perawat melalui *form input* yang disediakan oleh aplikasi. *Guest* dapat mencari dokter atau perawat dengan mengetikkan nama tenaga medisnya. Adapun rancangan antarmuka halaman konsultasi kesehatan dokter dan perawat dapat dilihat pada gambar 3.23.



Gambar 3.28 Rancangan Antarmuka Halaman Konsultasi

1. Rancangan antarmuka halaman detail dokter

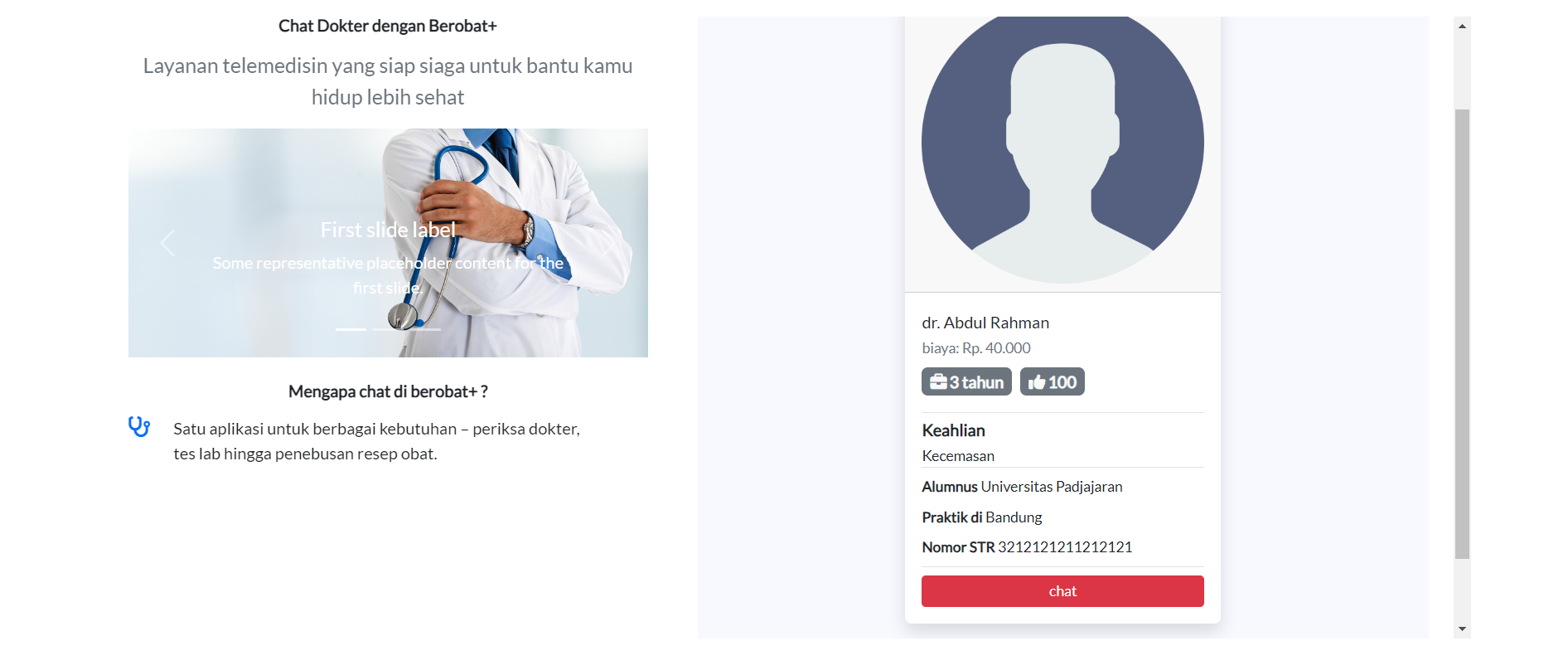
Rancangan antarmuka ini berisi detail dari dokter. Terdapat nama, biaya konsultasi, nama keahlian, dan nomor STR. Untuk mengetahui halaman ini *guest* perlu memilih dokter terlebih dahulu. Adapun rancangan antarmuka halaman detail dokter dapat dilihat pada gambar 3.24.



Gambar 3.29 Rancangan Antarmuka Halaman Detail Dokter

1. Rancangan antarmuka halaman detail dokter berdasarkan spesialis

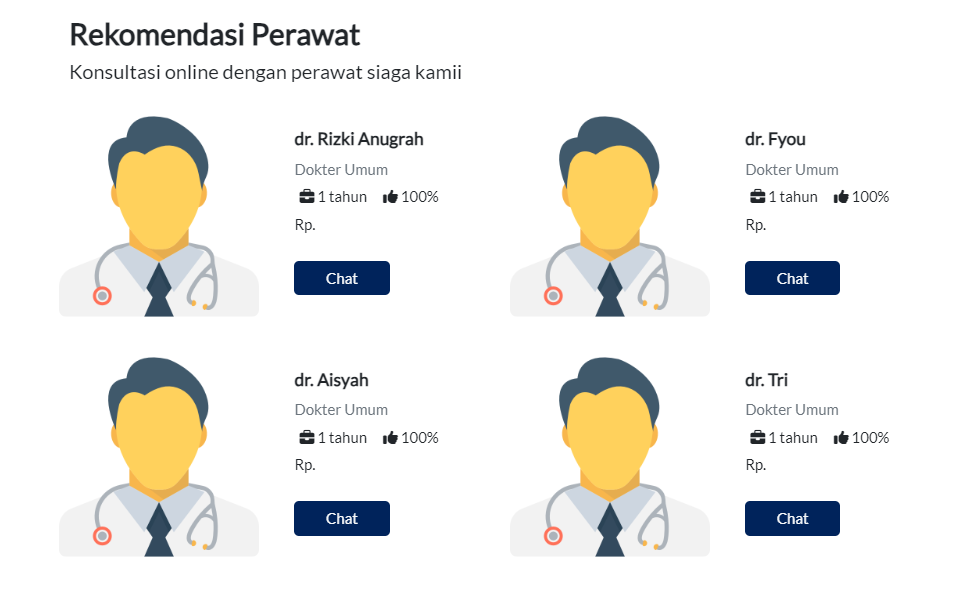
Rancangan antarmuka ini berisi detail dari dokter berdasarkan spesialisnya. Terdapat nama, biaya konsultasi, nama keahlian, dan nomor STR. Untuk mengetahui halaman ini *guest* perlu memilih dokter terlebih dahulu. Adapun rancangan antarmuka halaman detail dokter berdasarkan spesialis dapat dilihat pada gambar 3.24.



Gambar 3.30 Rancangan Antarmuka Halaman Detail Dokter berdasarkan Spesialis

1. Rancangan antarmuka halaman perawat

Rancangan antarmuka ini terdapat pada halaman konsultasi kesehatan dokter dan perawat. Terdapat daftar perawat yang tersedia pada aplikasi. Adapun rancangan antarmuka halaman perawat dapat dilihat pada gambar 3.26.

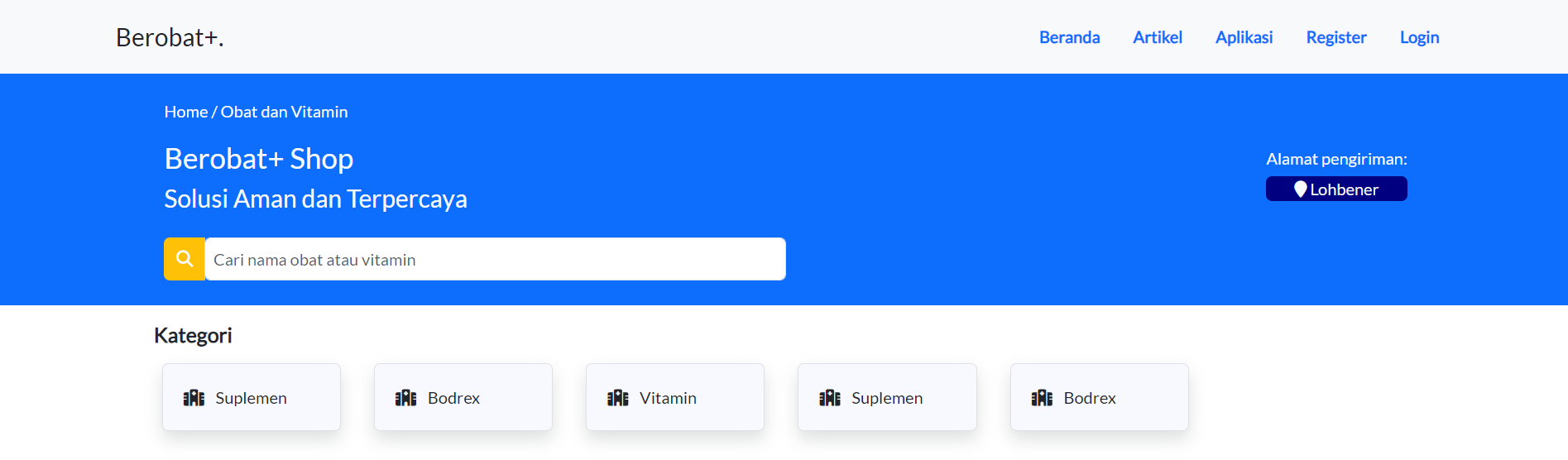


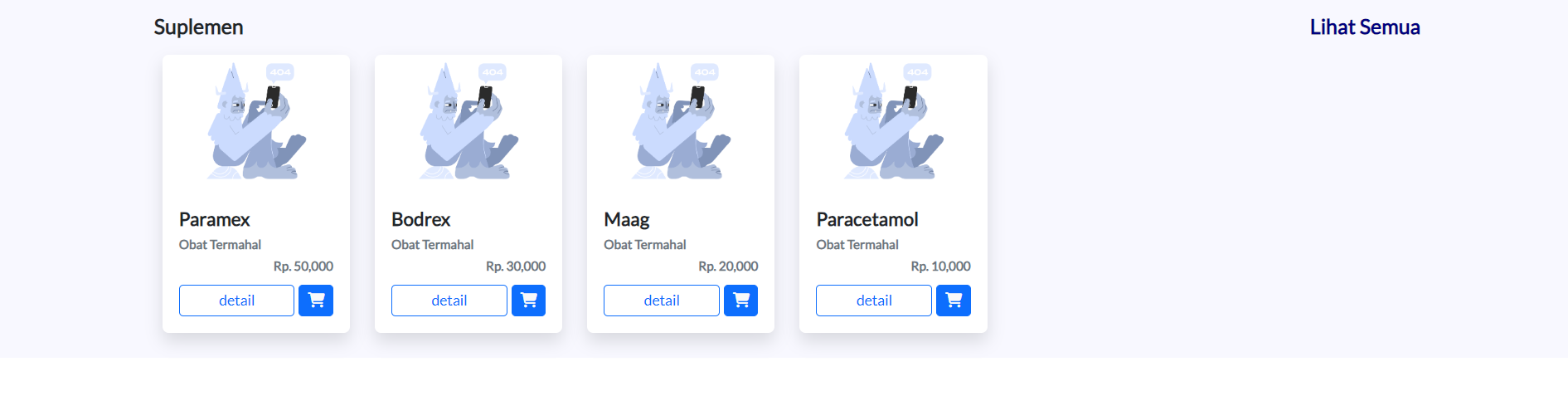
Gambar 3.31 Rancangan Antarmuka Halaman Perawat

1. Rancangan antarmuka halaman apotek

Rancangan antarmuka ini menampilkan daftar obat dan vitamin, juga apotek

terdekat berdasarkan lokasi *guest.* Untuk memesan obat dan vitamin, *guest* perlu mendaftar akun sebagai konsumen agar mendapatkan hak akses login ke dalam aplikasi. Adapun rancangan antarmuka halaman apotek dapat dilihat pada gambar 3.27.

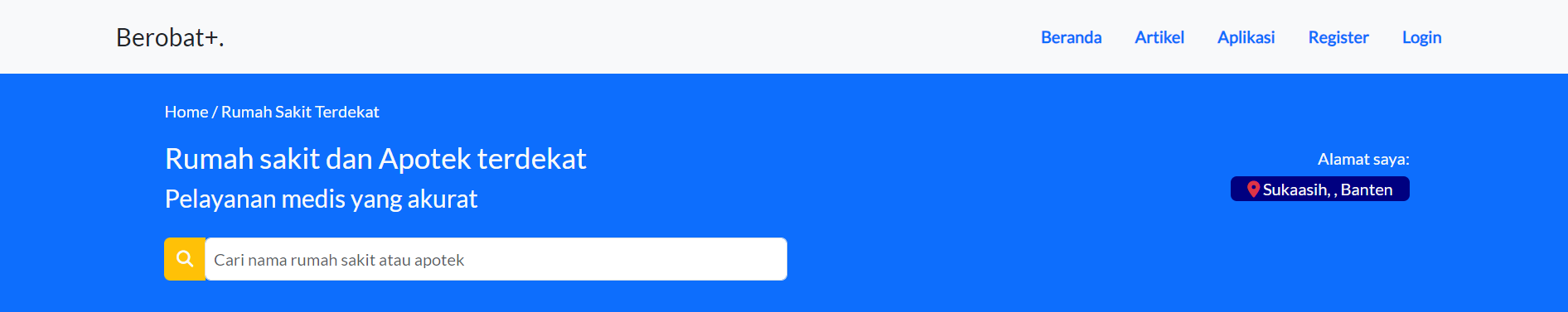


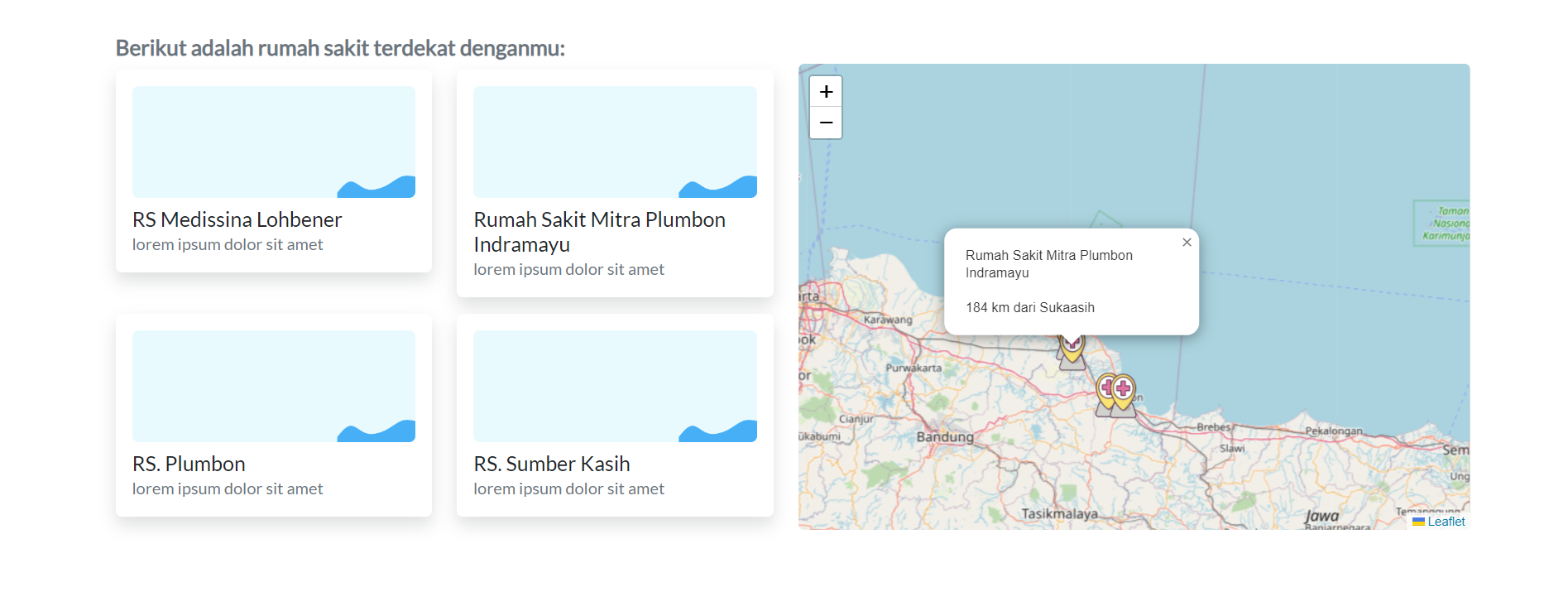


Gambar 3.32 Rancangan Antarmuka Halaman Apotek

1. Rancangan antarmuka halaman rumah sakit terdekat

Rancangan antarmuka ini digunakan untuk menampilkan daftar rumah sakit yang terdekat berdasarkan lokasi dari *guest.* Adapun rancangan antarmuka halaman rumah sakit terdekat dapat dilihat pada gambar 3.28.





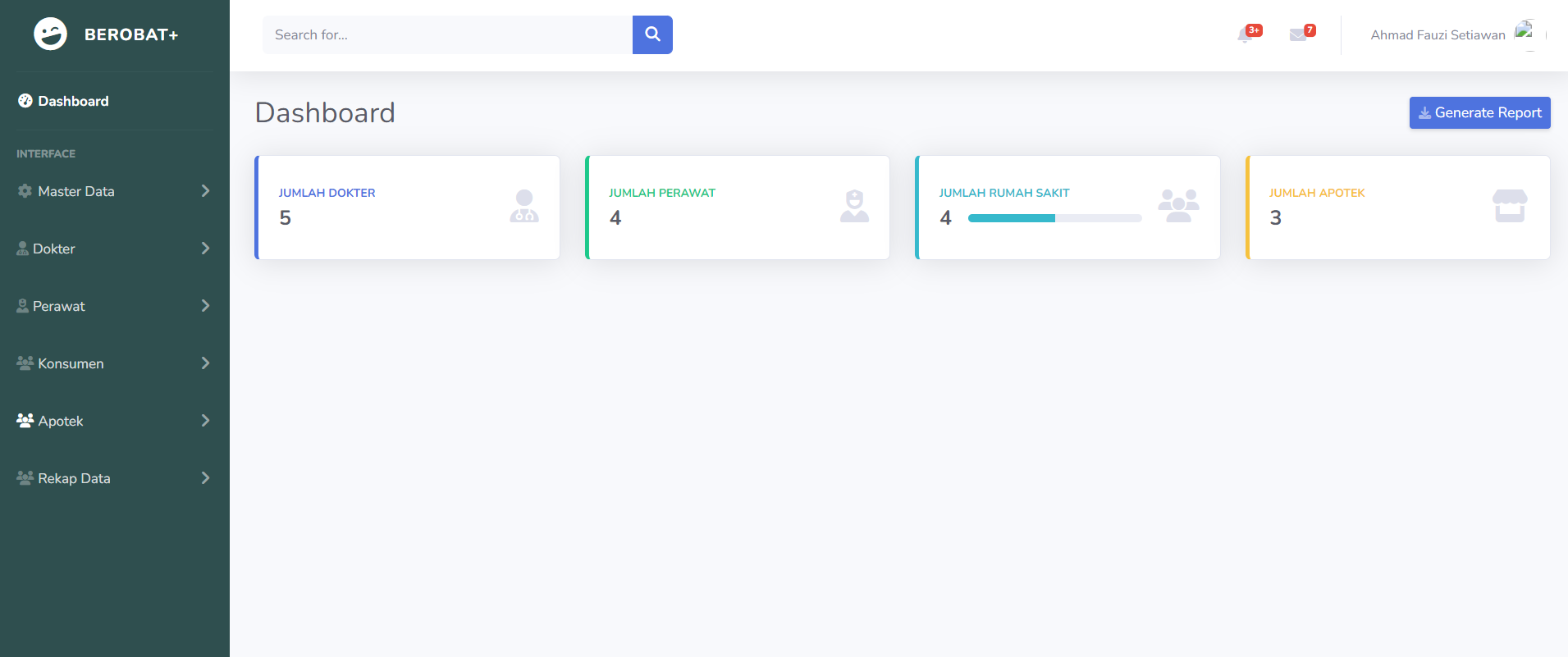
Gambar 3.33 Rancangan Antarmuka Halaman Rumah Sakit Terdekat

### Rancangan Antarmuka Admin

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

1. Rancangan Antarmuka Halaman Dashboard Utama

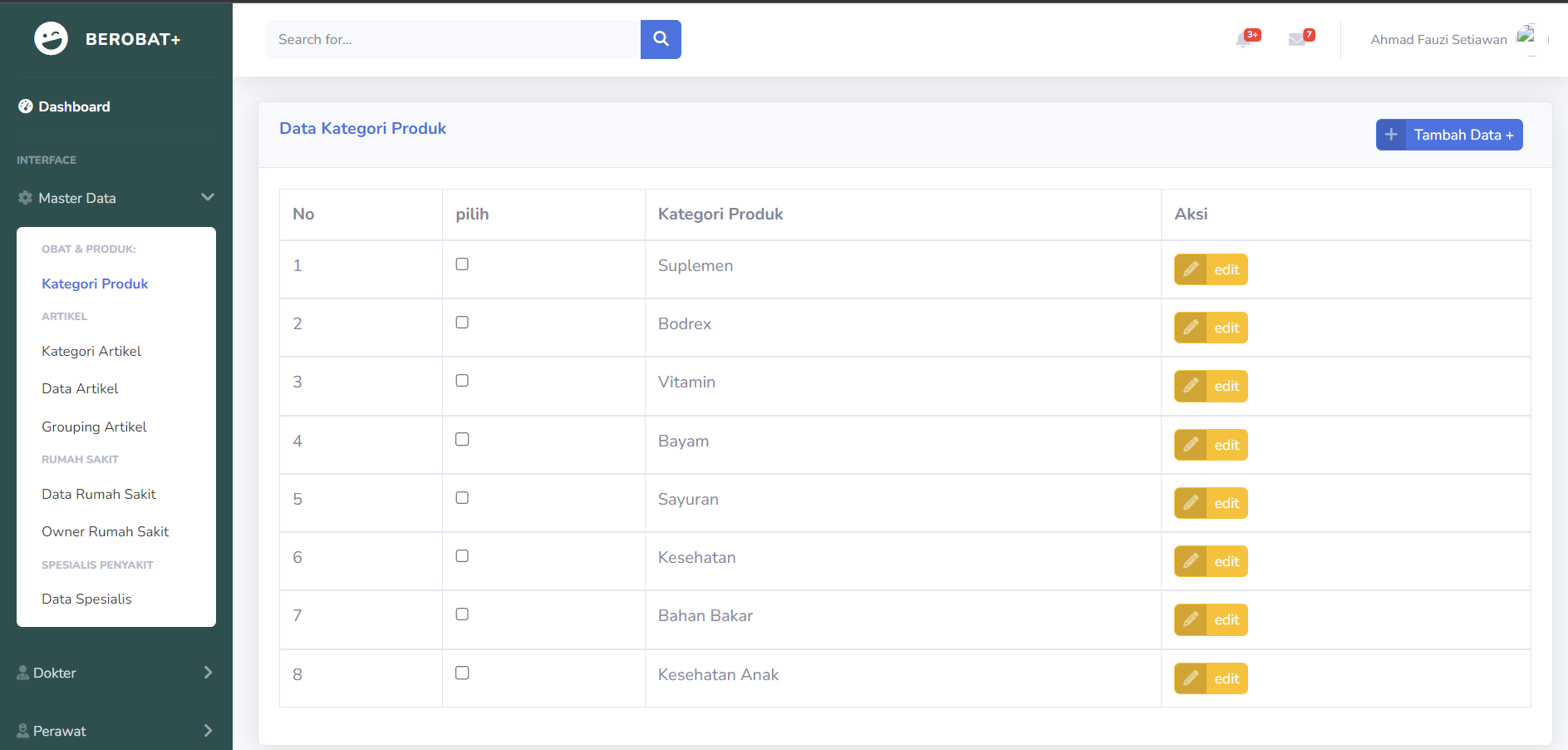
Rancangan antarmuka ini merupakan tampilan halaman dashboard utama. Yang berisi konten jumlah data. Adapun rancangan antarmuka dapat dilihat pada gambar 3.29.



Gambar 3.34 Rancangan Antarmuka Halaman Dashboard

1. Rancangan Antarmuka Halaman Mengelola Kategori Produk

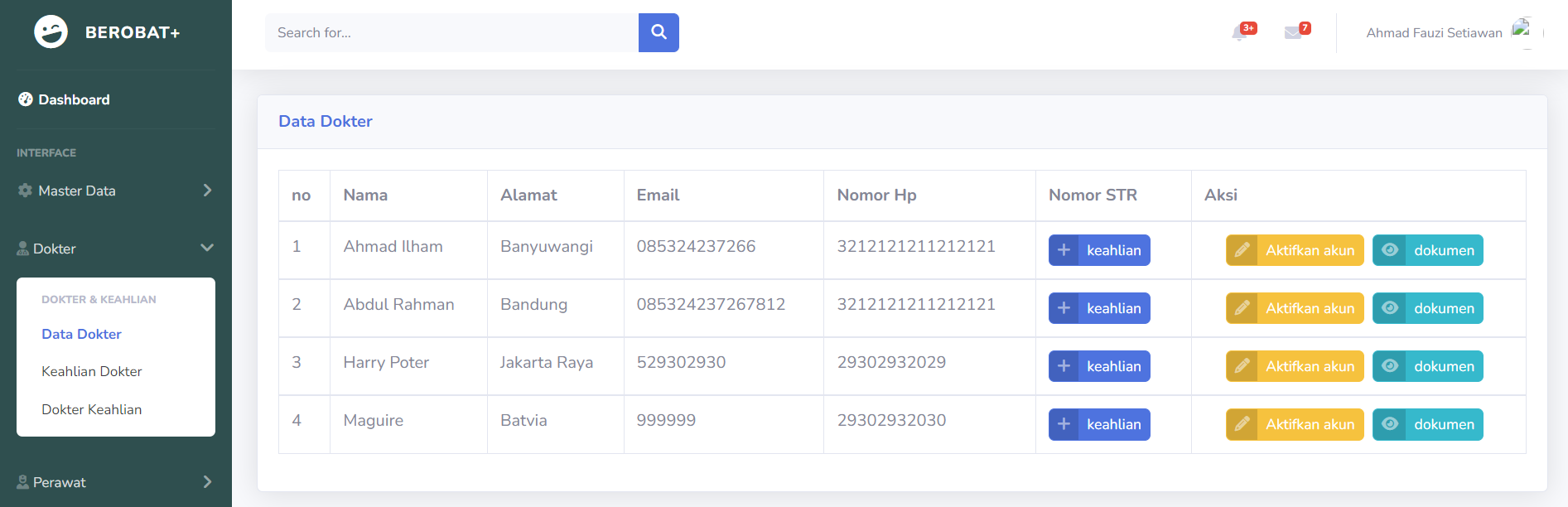
Rancangan antarmuka ini merupakan tampilan halaman untuk mengelola kategori produk. Adapun rancangan antarmuka dapat dilihat pada gambar 3.30.



Gambar 3.35 Rancangan Antarmuka Mengelola Kategori Produk

1. Rancangan Antarmuka Halaman Kelola Dokter

Pada perancangan halaman ini, admin dapat melakukan aktivisi akun dan melihat dokumen pendukung yang diunggah oleh dokter. Adapun rancangan antarmuka dapat dilihat pada gambar 3.31.



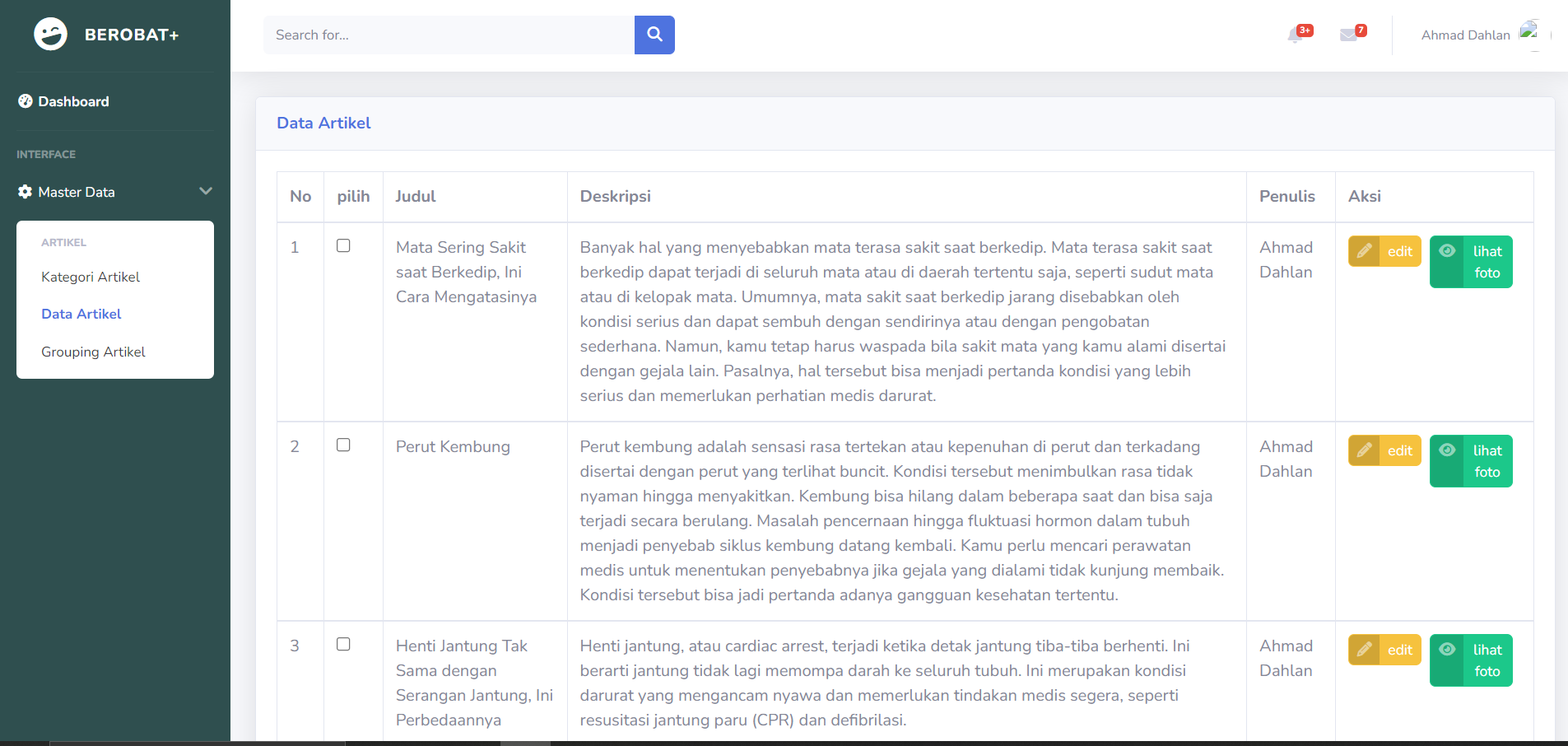
Gambar 3.36 Rancangan Antarmuka Kelola Dokter

### Rancangan Antarmuka Dokter

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

1. Rancangan antarmuka halaman kelola artikel

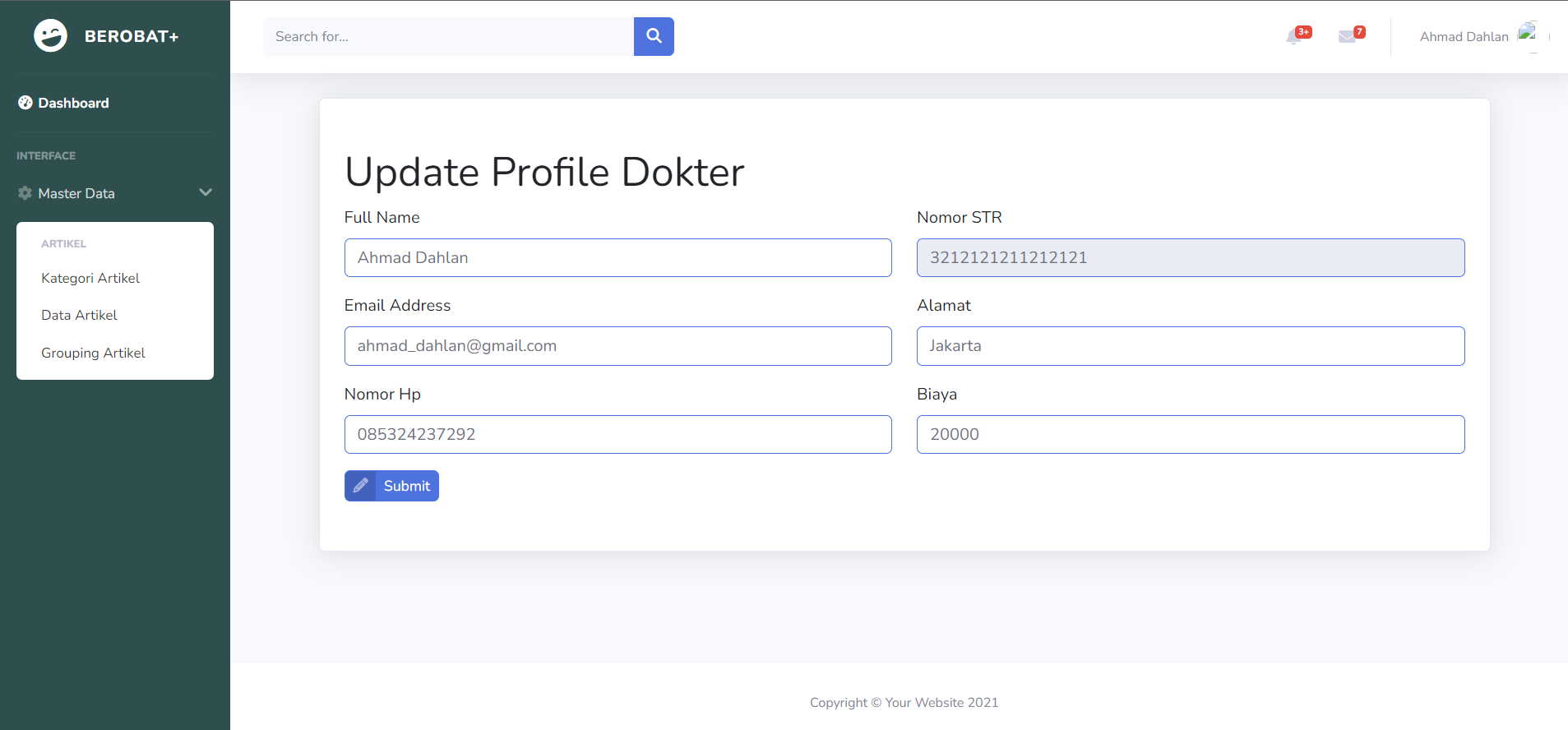
Rancangan antarmuka ini merupakan tampilan halaman untuk mengelola artikel. Adapun rancangan antarmuka dapat dilihat pada gambar 3.32.



Gambar 3.37 Rancangan Antarmuka Kelola Artikel

1. Rancangan antarmuka ubah profil akun

Rancangan antarmuka ini merupakan tampilan halaman untuk mengubah data akun. Adapun rancangan antarmuka dapat dilihat pada gambar 3.33.



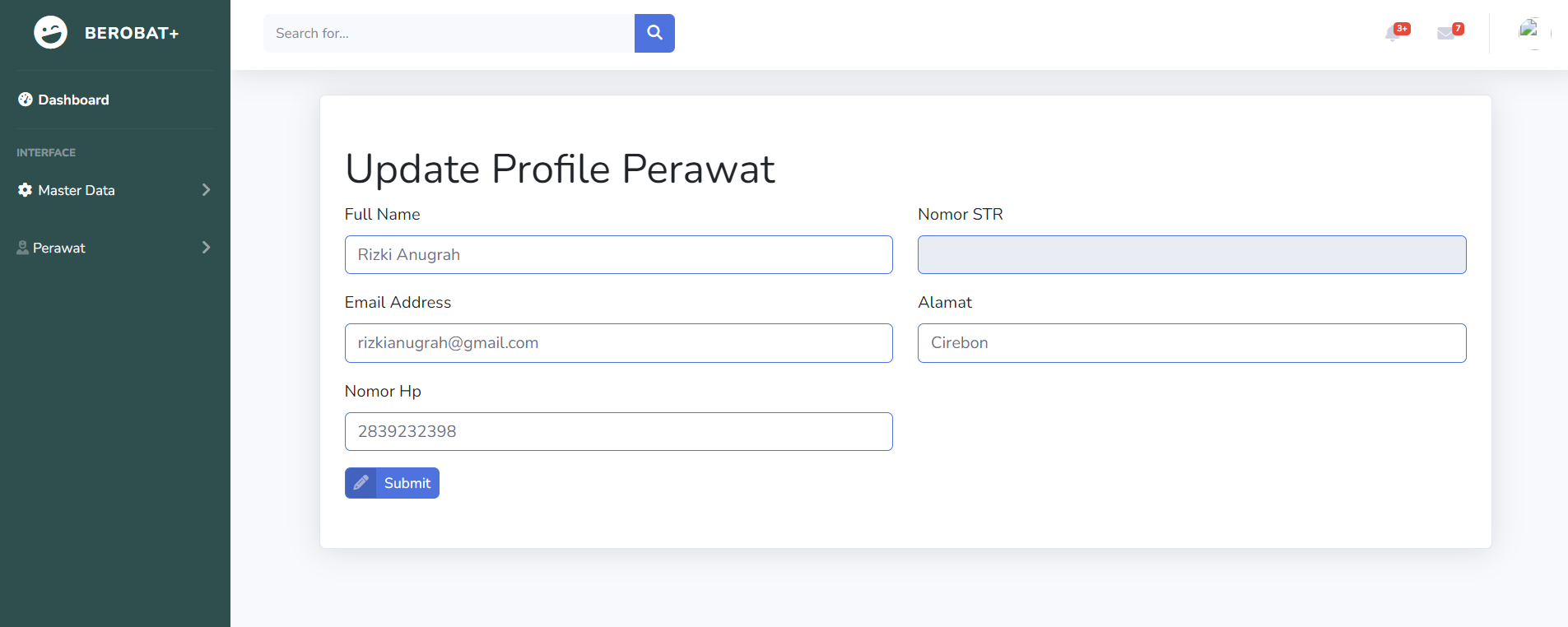
Gambar 3.38 Rancangan Antarmuka Ubah Profil Akun

### Rancangan Antarmuka Perawat

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

1. Rancangan antarmuka ubah profil akun

Rancangan antarmuka ini merupakan tampilan halaman untuk mengubah data akun. Adapun rancangan antarmuka dapat dilihat pada gambar 3.34.



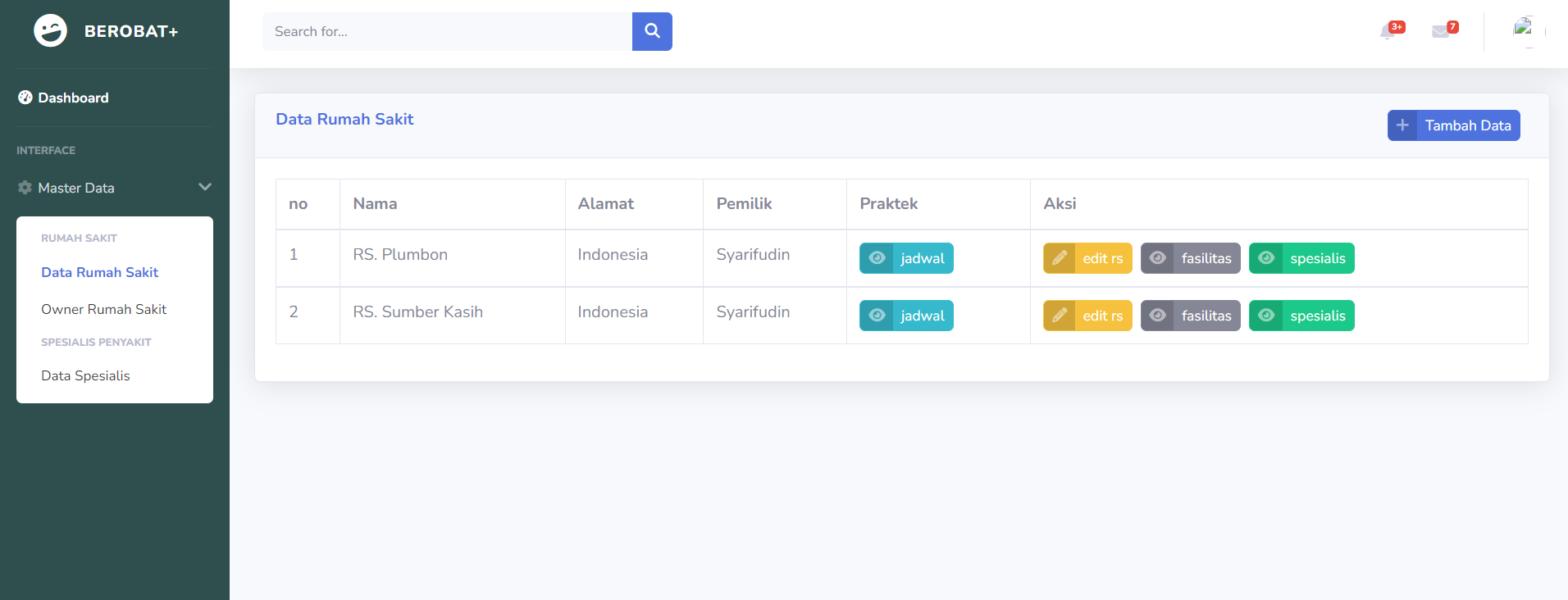
Gambar 3.39 Rancangan Antarmuka Halaman Ubah Profil Akun

### Rancangan Antarmuka Owner Rumah Sakit

Rancangan antarmuka owner rumah sakit merupakan rancangan *user interface* dari sisi owner rumah sakit. Berikut adalah rancangan antarmuka owner rumah sakit:

1. Rancangan antarmuka halaman kelola rumah sakit

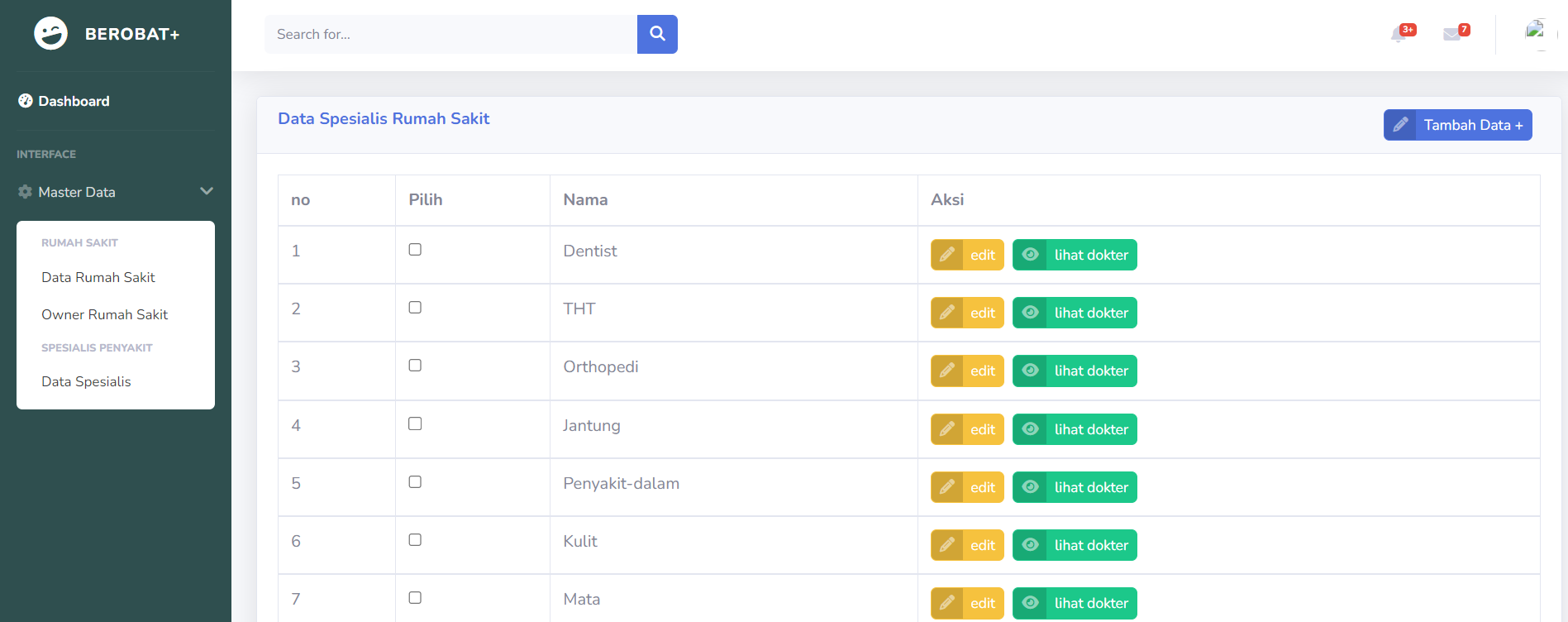
Rancangan antarmuka ini merupakan tampilan halaman untuk mengelola data rumah sakit. Adapun rancangan antarmuka dapat dilihat pada gambar 3.35.



Gambar 3.40 Rancangan Antarmuka Halaman Kelola Rumah Sakit

1. Rancangan antarmuka halaman kelola spesialis rumah sakit

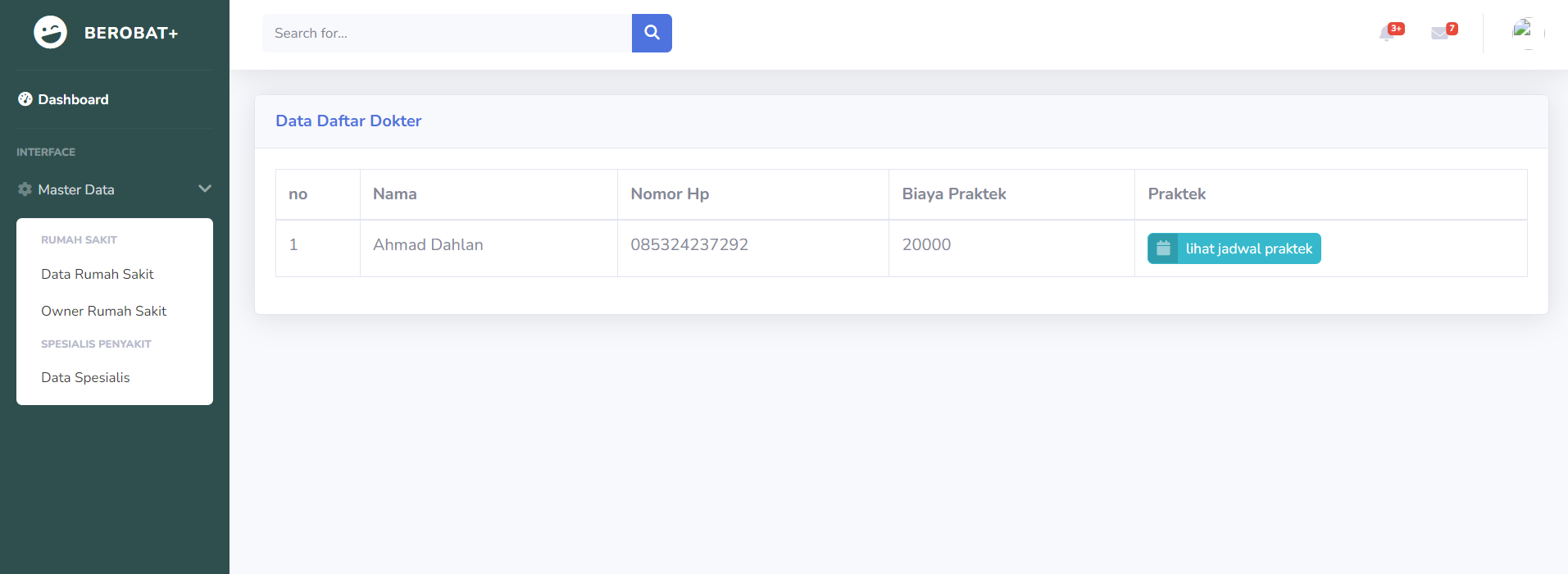
Rancangan antarmuka ini merupakan tampilan halaman untuk mengelola spesialis di rumah sakit. Adapun rancangan antarmuka dapat dilihat pada gambar 3.36.



Gambar 3.41 Rancangan Antarmuka Halaman Kelola Spesialis

1. Rancangan antarmuka kelola jadwal praktek dokter

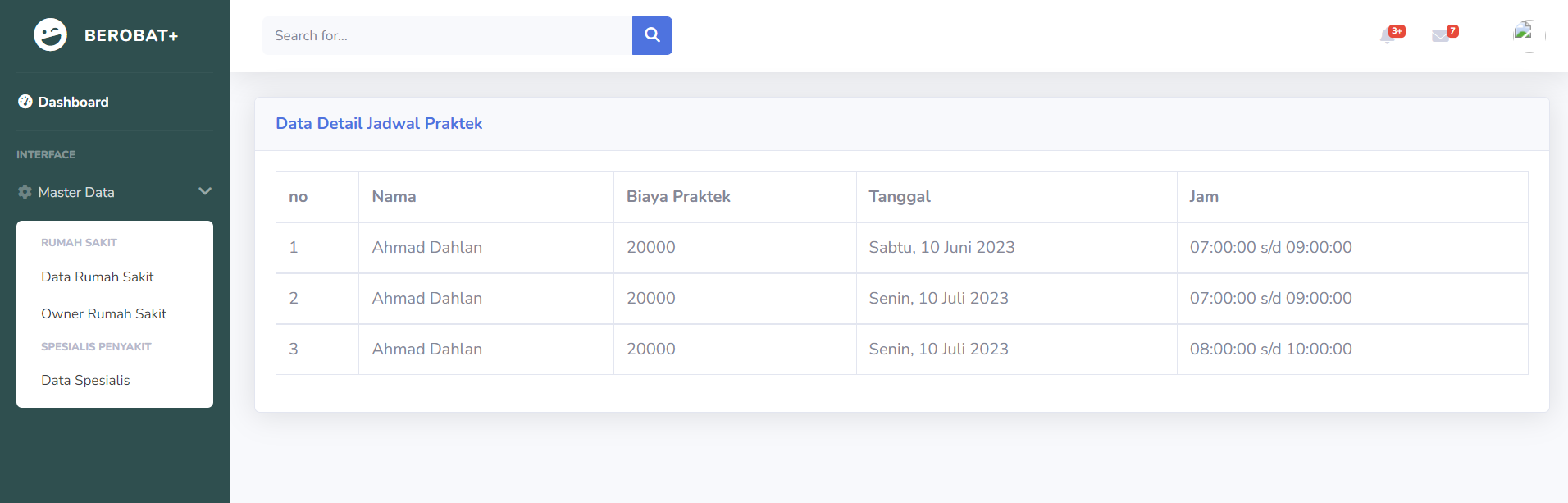
Rancangan antarmuka ini merupakan tampilan halaman untuk mengelola jadwal praktek di rumah sakit. Adapun rancangan antarmuka dapat dilihat pada gambar 3.36.



Gambar 3.42 Rancangan Antarmuka Kelola Jadwal Praktek Dokter

1. Rancangan antarmuka halaman detail praktek dokter

Rancangan antarmuka ini merupakan *user interface* untuk menampilkan detail praktek dokter. Rancangan antarmuka halaman detail praktek dokter dapat dilihat pada gambar 3.37.



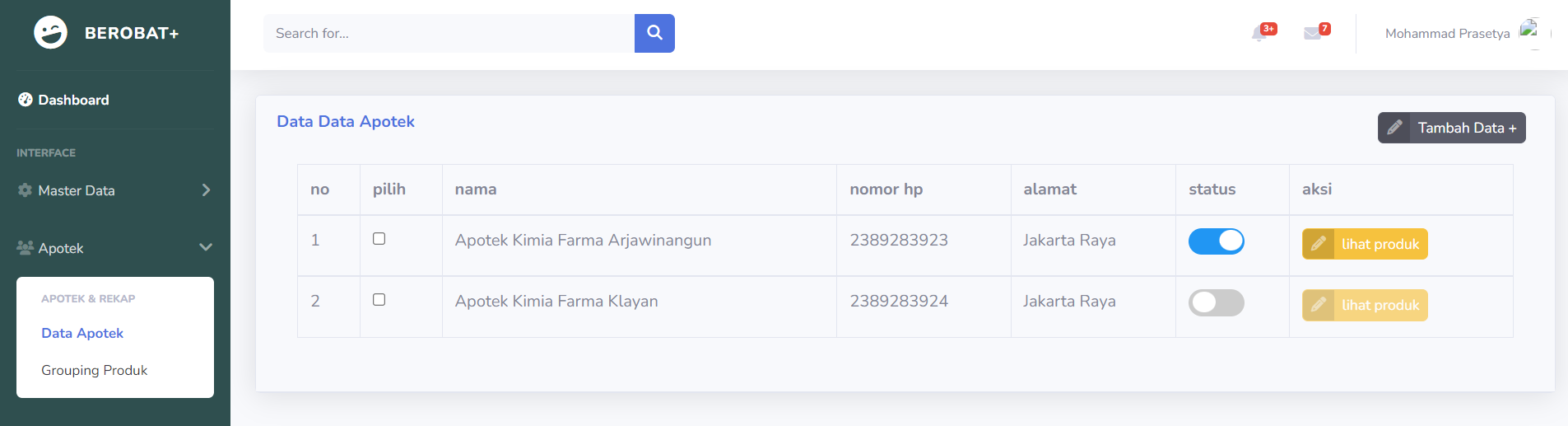
Gambar 3.43 Rancangan Antarmuka Halaman Detail Praktek Dokter

### Rancangan Antarmuka Owner Apotek

Rancangan antarmuka owner apotek merupakan rancangan *user interface* dari sisi owner apotek. Berikut adalah rancangan antarmuka owner apotek:

1. Rancangan antarmuka halaman kelola apotek

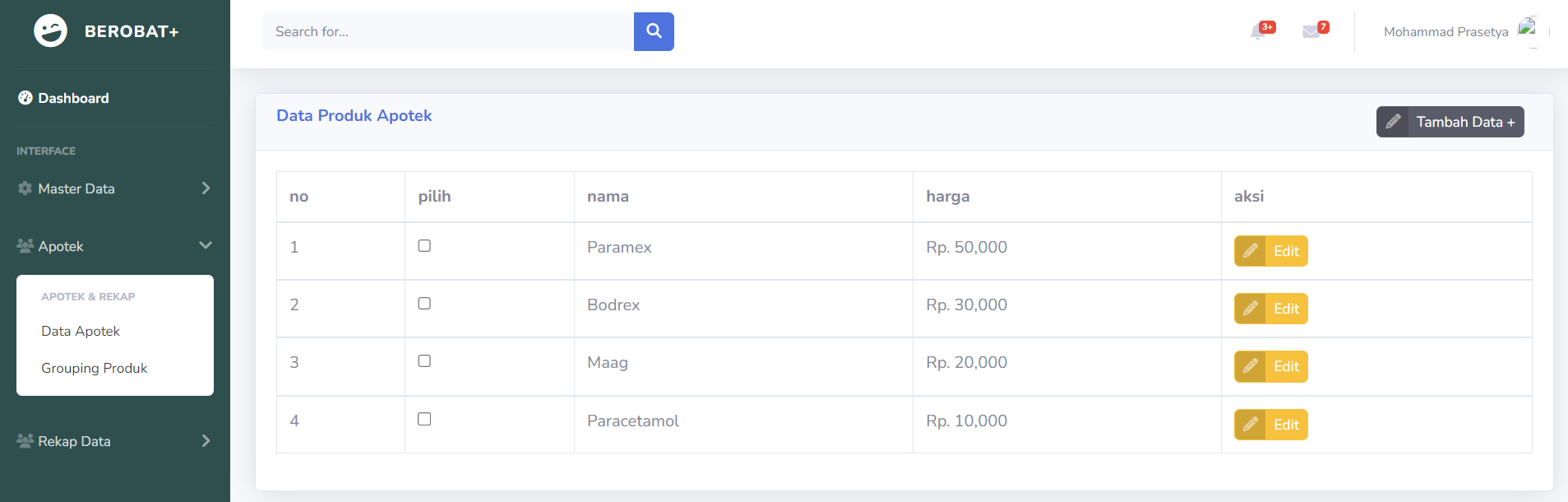
Rancangan antarmuka ini merupakan tampilan untuk mengelola apotek yang dimiliki owner apotek. Owner apotek dapat menambah, melihat, mengubah, dan menghapus apotek. Owner juga dapat mengubah status apotek melalui *slider button.* Adapun rancangan antarmuka halaman kelola apotek dapat dilihat pada gambar 3.39.



Gambar 3.44 Rancangan Antarmuka Halaman Kelola Apotek

1. Rancangan antarmuka halaman kelola produk apotek

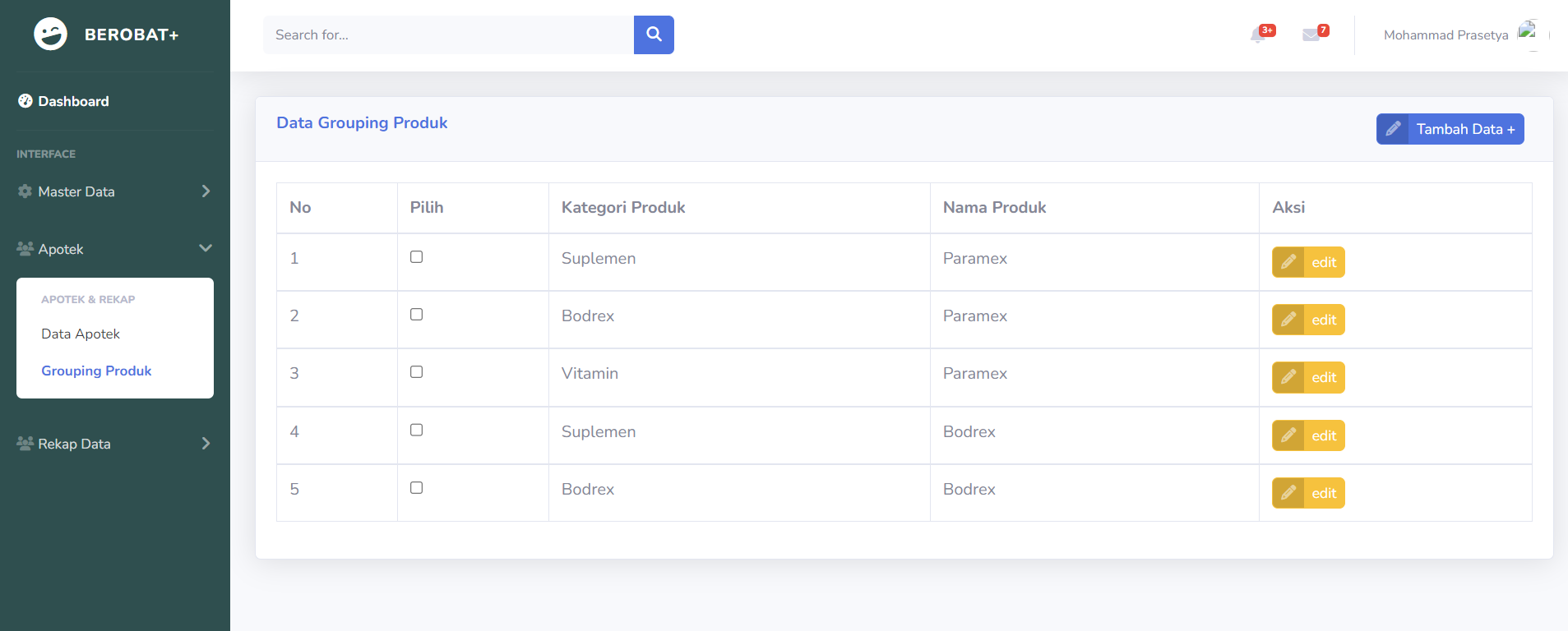
Rancangan antarmuka ini merupakan tampilan halaman untuk mengelola produk apotek. Adapun rancangan antarmuka dapat dilihat pada gambar 3.40.



Gambar 3.45 Rancangan Antarmuka Halaman Kelola Produk Apotek

1. Rancangan antarmuka halaman kelola produk kategori

Rancangan antarmuka ini digunakan pada halaman kelola produk apotek. Adapun rancangan antarmuka dapat dilihat pada gambar 3.41.



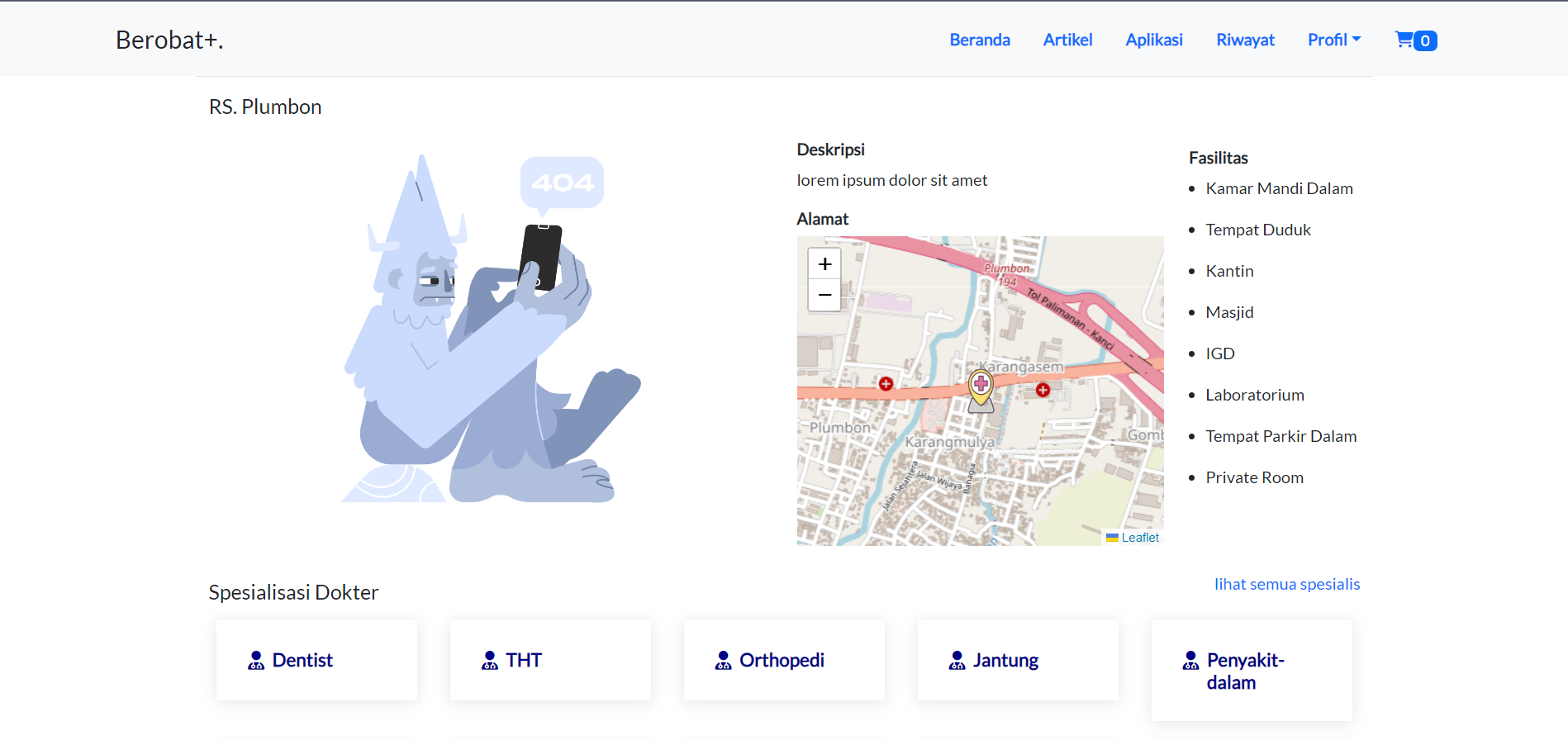
Gambar 3.46 Rancangan Antarmuka Halaman Kelola Produk Kategori

### Rancangan Antarmuka *Member*

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

1. Rancangan antarmuka detail rumah sakit

Rancangan antarmuka ini digunakan untuk menampilkan detail rumah sakit. Adapun rancangan antarmuka dapat dilihat pada gambar 3.42.



Gambar 3.47 Rancangan antarmuka halaman detail rumah sakit

1. Rancangan antarmuka buat janji kunjungan rumah sakit

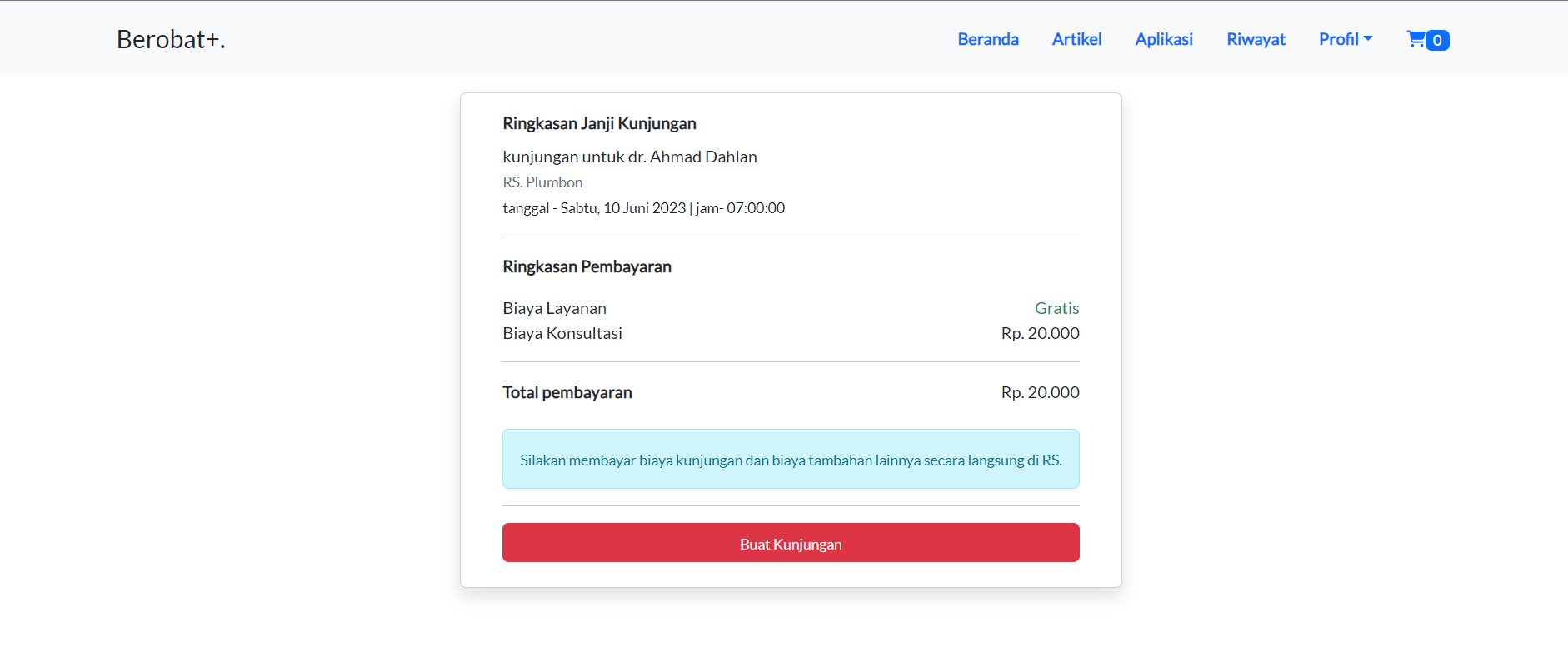
Rancangan antarmuka ini digunakan untuk menampilkan halaman untuk melakukan buat janji. Member perlu memilih jadwal praktek terlebih dahulu. Adapun rancangan antarmuka dapat dilihat pada gambar 3.43.



Gambar 3.48 Rancangan Antarmuka Buat Janji Kunjungan Rumah Sakit

1. Rancangan antarmuka detail kunjungan rumah sakit

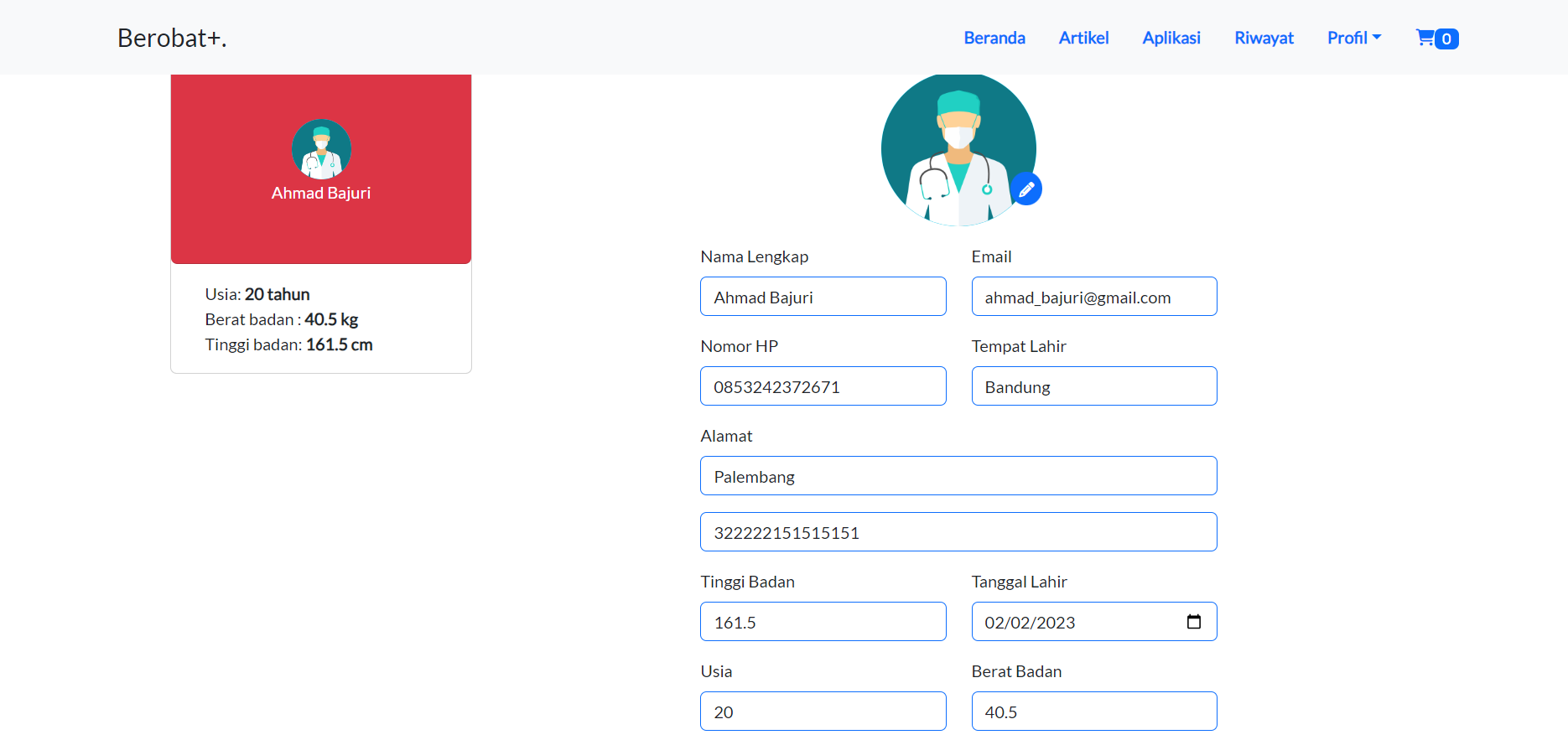
Rancangan antarmuka ini merupakan *user interface* untuk melihat detail kunjungan rumah sakit. Adapun rancangan antarmuka buat janji kunjungan rumah sakit dapat dilihat pada gambar 3.44.



Gambar 3.49 Rancangan Antarmuka Detail Kunjungan Rumah Sakit

1. Rancangan antarmuka ubah profil akun

Rancangan antarmuka ini merupakan tampilan halaman untuk mengubah data akun. Adapun rancangan antarmuka dapat dilihat pada gambar 3.45.



Gambar 3.50 Rancangan Antarmuka Halaman Ubah Profil Akun

# **BAB IV HASIL DAN PEMBAHASAN**

## Hasil

## Pembahasan

# **BAB V PENUTUP**

# **DAFTAR PUSTAKA**