**APLIKASI PELAYANAN KESEHATAN 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 |
| Program Studi | : Diploma III Teknik Informatika |
| Judul | : Aplikasi Pelayanan Kesehatan Masyarakat berbasis Web menggunakan *Framework* VueJs |

|  |  |  |  |
| --- | --- | --- | --- |
| Pembimbing | : I. | Adi Suheryadi, S.ST., M.Kom. |  |
|  |  | NIP 199003222019031007 | ……………… |

Telah berhasil dipertahankan dihadapan dewan penguji pada tanggal …. Agustus 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 Penguji | : | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  NIP |  |
|  |  | .............. |
| Anggota Penguji I | : | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  NIP |  |
|  | .............. |
| Anggota Penguji II | : | Adi Suheryadi, S.ST., M.Kom.  NIP 199003222019031007 |  |
|  | .............. |

|  |
| --- |
| Indramayu, … Agustus 2023  Ketua Jurusan Teknik Informatika |
| Eka Ismantohadi, S.Kom., M.Eng. |
| NIP 198107092021211005 |

# **HALAMAN PERNYATAAN**

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

|  |
| --- |
| Indramayu, 3 Agustus 2023 |
| Yang menyatakan, |
|  |
|  |
| Muhamad Rafli Septian. |
| NIM. 2003018 |

# **ABSTRAK**

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

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

# ***ABSTRACT***

*The Indonesian government, in its e-health strategy, strives to improve healthcare services by incorporating information and communication technology. Health is of utmost importance in fulfilling human needs, and health issues are crucial factors for a nation's development. However, there are challenges in achieving comprehensive healthcare services, with one of the obstacles being distance and time accessibility. Based on this, a solution is required, which is a web-based application for healthcare services and public consultation. Through this application, the public can access various healthcare services, such as information on nearby healthcare facilities and health articles, purchase health products, and schedule appointments with doctors at hospitals. The development of this application employs the agile software development method, utilizing the Scrum framework. Agile is a software development methodology carried out iteratively. In determining the location of the nearest healthcare services, the Haversine theorem is used for calculations. The result of developing this healthcare service and public consultation application is expected to enhance the quality of life for the community, as they can access healthcare services without being constrained by distance and time.*

***Keywords****: Healthcare Service, Agile Scrum, Haversine*

# **MOTTO**

“Ngejar apa sih sampe ga solat?”

# **KATA PENGANTAR**

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

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

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

1. Bapak Eka Ismantohadi, M.Kom. selaku Ketua Jurusan Teknik Informatika Politeknik Negeri Indramayu.
2. Bapak Muhamad Mustamiin, S.Pd., M.Kom. selaku Wali Dosen kelas D-3 TI 3 A.
3. Bapak Adi Suheryadi, S.ST., M.Kom. selaku Dosen Pembimbing yang telah *member*i dukungan dan arahan.
4. Bapak Fachrul Pralienka Bani Muhamad., S.ST., M.Kom selaku koordinator Program Studi D3 Teknik Informatika.
5. Kedua orang tua dan keluarga yang telah memberi dukungan moril dan materil yang tidak pernah henti.
6. Teman-teman kelas D-3 TI 3 A yang selalu memberi dukungan moril selama menjalani perkuliahan.
7. Mohammad Ilham Teguhriyadi dan Nandang Eka Prasetya, selaku rekan seperjuangan dalam mengerjakan 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 Masyarakat berbasis Web menggunakan *Framework* VueJs” dapat menambah wawasan serta ilmu pengetahuan bagi pembaca.

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

# **DAFTAR ISI**

[HALAMAN PERNYATAAN iii](#_Toc141871440)

[ABSTRAK iv](#_Toc141871441)

[*ABSTRACT* v](#_Toc141871442)

[KATA PENGANTAR vi](#_Toc141871443)

[DAFTAR ISI vii](#_Toc141871444)

[DAFTAR GAMBAR x](#_Toc141871445)

[DAFTAR TABEL xiii](#_Toc141871446)

[DAFTAR LAMPIRAN xiv](#_Toc141871447)

[BAB I PENDAHULUAN 1](#_Toc141871448)

[1.1 Latar Belakang 1](#_Toc141871449)

[1.2 Rumusan Masalah 2](#_Toc141871450)

[1.3 Batasan Masalah 2](#_Toc141871451)

[1.4 Tujuan 3](#_Toc141871452)

[1.5 Manfaat 3](#_Toc141871453)

[1.6 Sistematika Penulisan 4](#_Toc141871454)

[BAB II LANDASAN TEORI 1](#_Toc141871455)

[2.1. Dokter 1](#_Toc141871456)

[2.2 Dokter Praktik Umum 1](#_Toc141871457)

[2.3 Dokter Spesialis 2](#_Toc141871458)

[2.4 Keperawatan 2](#_Toc141871459)

[2.5 Perawat 2](#_Toc141871460)

[2.5.1 Peran Perawat 3](#_Toc141871461)

[2.5.2 Peran sebagai Pelaksana Kesehatan 3](#_Toc141871462)

[2.5.3 Peran sebagai Pendidik 3](#_Toc141871463)

[2.5.4 Peran sebagai administrasi 3](#_Toc141871464)

[2.5.5 Peran sebagai konseling 3](#_Toc141871465)

[2.5.6 Peran sebagai peneliti 4](#_Toc141871466)

[2.6 Latitude 4](#_Toc141871467)

[2.7 Longitude 4](#_Toc141871468)

[2.8 Teorema Haversine 4](#_Toc141871469)

[2.9 API 5](#_Toc141871470)

[2.10 Vue.js 5](#_Toc141871471)

[2.11 Metode Scrum 6](#_Toc141871472)

[2.12 UML (Unified Modeling Language) 7](#_Toc141871473)

[2.12.1 Pengenalan UML 7](#_Toc141871474)

[2.12.2 *Use Case* Diagram 7](#_Toc141871475)

[2.12.3 *Activity* Diagram 8](#_Toc141871476)

[2.13 *Entity Relationship* Diagram(ERD) 9](#_Toc141871477)

[2.14 *Global Positioning System* (GPS) 10](#_Toc141871478)

[2.15 OpenStreetMap 11](#_Toc141871479)

[2.16 LeafletJS 11](#_Toc141871480)

[2.17 *Black Box Testing* 11](#_Toc141871481)

[BAB III METODE PELAKSANAAN 13](#_Toc141871482)

[3.1 Metode Pengembangan Perangkat Lunak 13](#_Toc141871483)

[3.2 Analisis Kebutuhan Sistem 14](#_Toc141871484)

[3.2.1 Kebutuhan Fungsional 14](#_Toc141871485)

[3.2.2 Kebutuhan Non Fungsional 17](#_Toc141871486)

[3.2.3 Kebutuhan *Hardware* 17](#_Toc141871487)

[3.2.4 Kebutuhan *Software* 18](#_Toc141871488)

[3.3 Perancangan Sistem 18](#_Toc141871489)

[3.3.1 Use Case Diagram 18](#_Toc141871490)

[3.3.2 Activity Diagram 23](#_Toc141871491)

[3.4 Flowchart 38](#_Toc141871492)

[3.4.1 Flowchart Registrasi Akun 39](#_Toc141871493)

[3.4.2 Flowchart Membaca Artikel 39](#_Toc141871494)

[3.4.3 Flowchart Melihat Rumah Sakit Terdekat 40](#_Toc141871495)

[3.4.4 Flowchart Memesan Produk Apotek 41](#_Toc141871496)

[3.4.5 Flowchart Membuat Janji Kunjungan Rumah Sakit 41](#_Toc141871497)

[3.5 Perancangan *Entity Relationship* Diagram (ERD) 42](#_Toc141871498)

[3.6 Perancangan Basis Data 44](#_Toc141871499)

[3.6.1 Tabel Master Data 44](#_Toc141871500)

[3.6.2 Tabel Proses Bisnis 49](#_Toc141871501)

[3.7 Perancangan Desain Tampilan Antarmuka 51](#_Toc141871502)

[3.7.1 Rancangan Antarmuka *Guest* 51](#_Toc141871503)

[3.7.2 Rancangan Antarmuka Admin 57](#_Toc141871504)

[3.7.3 Rancangan Antarmuka Dokter 59](#_Toc141871505)

[3.7.4 Rancangan Antarmuka Perawat 60](#_Toc141871506)

[3.7.5 Rancangan Antarmuka Owner Rumah Sakit 61](#_Toc141871507)

[3.7.6 Rancangan Antarmuka Owner Apotek 62](#_Toc141871508)

[3.7.7 Rancangan Antarmuka *Member* 64](#_Toc141871509)

[BAB IV HASIL DAN PEMBAHASAN 66](#_Toc141871510)

[4.1 Hasil 66](#_Toc141871511)

[4.2 Pembahasan 66](#_Toc141871512)

[4.2.1 Implementasi Basis Data 67](#_Toc141871513)

[4.2.2 *Sprint Review Product Backlog* 75](#_Toc141871514)

[4.2.3 Daily Activity 84](#_Toc141871515)

[4.2.4 Testing Endpoint API 85](#_Toc141871516)

[4.2.5 Testing Modul *Product Backlog* 85](#_Toc141871517)

[BAB V PENUTUP 86](#_Toc141871518)

[5.1 Kesimpulan 86](#_Toc141871519)

[5.2 Penutup 86](#_Toc141871520)

[DAFTAR PUSTAKA 87](#_Toc141871521)

[LAMPIRAN 88](#_Toc141871522)

# **DAFTAR GAMBAR**

[Gambar 3.1 Use Case Diagram 19](#_Toc141192940)

[Gambar 3.2 Activity Diagram Mengelola Data Rumah Sakit 24](#_Toc141192941)

[Gambar 3.3 Activity Diagram Mengelola Fasilitas Rumah Sakit 25](#_Toc141192942)

[Gambar 3.4 Acitivity Diagram Mengelola Data Spesialis Rumah Sakit 26](#_Toc141192943)

[Gambar 3.5 Activity Diagram Mengelola Jadwal Praktek Dokter Rumah Sakit 27](#_Toc141192944)

[Gambar 3.6 Activity Diagram Mengelola Data Apotek 28](#_Toc141192945)

[Gambar 3.7 Activity Diagram Mengelola Data Produk Apotek 29](#_Toc141192946)

[Gambar 3.8 Activity Diagram Mengubah Status Apotek 30](#_Toc141192947)

[Gambar 3.9 Activity Diagram Mengelola Data Grouping Produk 31](#_Toc141192948)

[Gambar 3.10 Activity Diagram Melihat Kategori Artikel 32](#_Toc141192949)

[Gambar 3.11 Activity Diagram Menetapkan Harga Konsultasi Dokter dan Perawat 32](#_Toc141192950)

[Gambar 3.12 Activity Diagram Melihat Dokter berdasarkan Spesialis 33](#_Toc141192951)

[Gambar 3.13 Activity Diagram Melakukan Pencarian Dokter dan Perawat 34](#_Toc141192952)

[Gambar 3.14 Activity Diagram Membuat Janji Kunjungan Rumah Sakit 35](#_Toc141192953)

[Gambar 3.15 Activity Diagram Membaca Artikel Kesehatan 36](#_Toc141192954)

[Gambar 3.16 Activity Diagram Registrasi Akun 37](#_Toc141192955)

[Gambar 3.17 Activity Diagram Mengubah Status Akun Dokter 38](#_Toc141192956)

[Gambar 3.18 Flowchart Register 39](#_Toc141192957)

[Gambar 3.19 Flowchart Membaca Artikel 40](#_Toc141192958)

[Gambar 3.20 Flowchart Melihat Rumah Sakit Terdekat 40](#_Toc141192959)

[Gambar 3.21 Flowchart Memesan Produk Apotek 41](#_Toc141192960)

[Gambar 3.22 Flowchart Membuat Janji Kunjungan Rumah Sakit 42](#_Toc141192961)

[Gambar 3.23 Rancangan ERD 42](#_Toc141192962)

[Gambar 3.24 Rancangan Antarmuka Halaman Home 52](#_Toc141192963)

[Gambar 3.25 Rancangan Antarmuka Halaman Artikel 53](#_Toc141192964)

[Gambar 3.26 Rancangan Antarmuka Halaman Detail Artikel 53](#_Toc141192965)

[Gambar 3.27 Rancangan Antarmuka Halaman Konsultasi Dokter dan Perawat 54](#_Toc141192966)

[Gambar 3.28 Rancangan Antarmuka Halaman Konsultasi 54](#_Toc141192967)

[Gambar 3.29 Rancangan Antarmuka Halaman Detail Dokter 55](#_Toc141192968)

[Gambar 3.30 Rancangan Antarmuka Halaman Detail Dokter berdasarkan Spesialis 55](#_Toc141192969)

[Gambar 3.31 Rancangan Antarmuka Halaman Perawat 56](#_Toc141192970)

[Gambar 3.32 Rancangan Antarmuka Halaman Apotek 57](#_Toc141192971)

[Gambar 3.33 Rancangan Antarmuka Halaman Rumah Sakit Terdekat 57](#_Toc141192972)

[Gambar 3.34 Rancangan Antarmuka Halaman Dashboard 58](#_Toc141192973)

[Gambar 3.35 Rancangan Antarmuka Mengelola Kategori Produk 58](#_Toc141192974)

[Gambar 3.36 Rancangan Antarmuka Kelola Dokter 59](#_Toc141192975)

[Gambar 3.37 Rancangan Antarmuka Kelola Artikel 59](#_Toc141192976)

[Gambar 3.38 Rancangan Antarmuka Ubah Profil Akun 60](#_Toc141192977)

[Gambar 3.39 Rancangan Antarmuka Halaman Ubah Profil Akun 60](#_Toc141192978)

[Gambar 3.40 Rancangan Antarmuka Halaman Kelola Rumah Sakit 61](#_Toc141192979)

[Gambar 3.41 Rancangan Antarmuka Halaman Kelola Spesialis 61](#_Toc141192980)

[Gambar 3.42 Rancangan Antarmuka Kelola Jadwal Praktek Dokter 62](#_Toc141192981)

[Gambar 3.43 Rancangan Antarmuka Halaman Detail Praktek Dokter 62](#_Toc141192982)

[Gambar 3.44 Rancangan Antarmuka Halaman Kelola Apotek 63](#_Toc141192983)

[Gambar 3.45 Rancangan Antarmuka Halaman Kelola Produk Apotek 63](#_Toc141192984)

[Gambar 3.46 Rancangan Antarmuka Halaman Kelola Produk Kategori 63](#_Toc141192985)

[Gambar 3.47 Rancangan antarmuka halaman detail rumah sakit 64](#_Toc141192986)

[Gambar 3.48 Rancangan Antarmuka Buat Janji Kunjungan Rumah Sakit 64](#_Toc141192987)

[Gambar 3.49 Rancangan Antarmuka Detail Kunjungan Rumah Sakit 65](#_Toc141192988)

[Gambar 3.50 Rancangan Antarmuka Halaman Ubah Profil Akun 65](#_Toc141192989)

[Gambar 4.1 Implementasi Basis Data 67](#_Toc141192990)

[Gambar 4.2 Implementasi Tabel Artikel 67](#_Toc141192991)

[Gambar 4.3 Implementasi Kategori Artikel 68](#_Toc141192992)

[Gambar 4.4 Implementasi Kategori Produk 68](#_Toc141192993)

[Gambar 4.5 Implementasi Tabel Spesialis Penyakit 68](#_Toc141192994)

[Gambar 4.6 Implementasi Tabel Spesialis Rumah Sakit 69](#_Toc141192995)

[Gambar 4.7 Implementasi Tabel Fasilitas Rumah Sakit 69](#_Toc141192996)

[Gambar 4.8 Implementasi Tabel Grouping Artikel 69](#_Toc141192997)

[Gambar 4.9 Implementasi Tabel Produk Kategori 70](#_Toc141192998)

[Gambar 4.10 Implementasi Tabel Profil Apotek 70](#_Toc141192999)

[Gambar 4.11 Implementasi Tabel Rumah Sakit 71](#_Toc141193000)

[Gambar 4.12 Implementasi Tabel Keahlian 71](#_Toc141193001)

[Gambar 4.13 Implementasi Tabel Join Keahlian 72](#_Toc141193002)

[Gambar 4.14 Implementasi Tabel Detail Praktek 72](#_Toc141193003)

[Gambar 4 15 Implementasi Tabel Jadwal Praktek 72](#_Toc141193004)

[Gambar 4.16 Implementasi Tabel Jadwal Antrian 73](#_Toc141193005)

[Gambar 4.17 Implementasi Tabel Transaksi 73](#_Toc141193006)

[Gambar 4.18 Implementasi Tabel Produk 74](#_Toc141193007)

[Gambar 4.19 Implementasi Tabel Users 74](#_Toc141193008)

[Gambar 4.20 Sprint Review Registrasi Akun 75](#_Toc141193009)

[Gambar 4.21 Sprint Review Login Aplikasi 76](#_Toc141193010)

[Gambar 4.22 Sprint Review Registrasi Akun Member 76](#_Toc141193011)

[Gambar 4.23 Sprint Review Modul Aktivasi Akun Pengguna 77](#_Toc141193012)

[Gambar 4.24 Sprint Review Melihat Produk 77](#_Toc141193013)

[Gambar 4.25 Sprint Review Keranjang Produk 78](#_Toc141193014)

[Gambar 4.26 Sprint Review Detail Checkout Produk 78](#_Toc141193015)

[Gambar 4.27 Sprint Review Riwayat Transaksi 79](#_Toc141193016)

[Gambar 4.28 Sprint Review Informasi Klinis Terdekat 80](#_Toc141193017)

[Gambar 4.29 Sprint Review Detail Informasi Klinis 80](#_Toc141193018)

[Gambar 4.30 Sprint Review Melihat Jadwal Praktek Dokter 80](#_Toc141193019)

[Gambar 4.31 Sprint Review Ubah Profil Akun 82](#_Toc141193020)

[Gambar 4.32 Sprint Review Ubah Kata Sandi Akun 82](#_Toc141193021)

# **DAFTAR TABEL**

[Tabel 3.1Tabel Kebutuhan Hardware 17](#_Toc139021680)

[Tabel 3.2 Tabel Kebutuhan Software 18](#_Toc139021681)

[Tabel 3.3 Tabel Penjelasan Use Case Diagram 19](#_Toc139021682)

[Tabel 3.4 Penjelasan ERD 43](#_Toc139021683)

[Tabel 3.5 Tabel Kategori Artikel 44](#_Toc139021684)

[Tabel 3.6 Tabel Kategori Produk 44](#_Toc139021685)

[Tabel 3.7 Tabel Spesialis 45](#_Toc139021686)

[Tabel 3.8 Tabel Spesialis Rumah Sakit 45](#_Toc139021687)

[Tabel 3.9 Tabel Fasilitas Rumah Sakit 45](#_Toc139021688)

[Tabel 3.10 Tabel Grouping Artikel 46](#_Toc139021689)

[Tabel 3.11 Tabel Produk Kategori 46](#_Toc139021690)

[Tabel 3.12 Tabel Profil Apotek 46](#_Toc139021691)

[Tabel 3.13 Tabel Rumah Sakit 47](#_Toc139021692)

[Tabel 3.14 Tabel Keahlian 47](#_Toc139021693)

[Tabel 3.15 Tabel Join Keahlian 48](#_Toc139021694)

[Tabel 3.16 Tabel Detail Praktek 48](#_Toc139021695)

[Tabel 3.17 Tabel Jadwal Praktek 48](#_Toc139021696)

[Tabel 3.18 Tabel Transaksi 49](#_Toc139021697)

[Tabel 3.19 Tabel Artikel 49](#_Toc139021698)

[Tabel 3.20 Tabel Jadwal Antrian 49](#_Toc139021699)

[Tabel 3.21 Tabel Produk 50](#_Toc139021700)

[Tabel 3.22 Tabel Users 50](#_Toc139021701)

# **DAFTAR LAMPIRAN**

Lampiran 1 Kode Program

Lampiran 2 *Product Backlog*

Lampiran 3 *Sprint Planning*

Lampiran 4 *Daily Activity*

Lampiran 5 *Testing Product Backlog*

Lampiran 6 *Testing Endpoint*

Lampiran 7 Biodata Penulis

# **BAB I PENDAHULUAN**

## Latar Belakang

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

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

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

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

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

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

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

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

## Rumusan Masalah

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

1. Bagaimana mengetahui informasi terkait kebutuhan kesehatan masyarakat serta lokasi pelayanan kesehatan terdekat?
2. Bagaimana mendapatkan produk kesehatan melalui web?
3. Bagaimana membuat janji kunjungan temu dokter di rumah sakit melalui web?

## Batasan Masalah

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

1. Aplikasi dirancang berbasis web menggunakan *framework* vuejs 3 dan NodeJs versi 18.
2. Rumah sakit dan apotek terdekat hanya menampilkan lokasi di peta dan menggunakan google maps untuk menuju titik lokasi.
3. Tidak terintegrasi sistem informasi rumah sakit, pada saat membuat janji kunjungan.
4. Hanya dapat membuat satu jadwal kunjungan dalam satu hari dengan dokter yang sama.
5. Tidak ada penggantian jadwal (*reschedule*) pada janji kunjungan, namun ada dapat dibatalkan 60 menit sebelum jadwal kunjungan berlangsung.

## Tujuan

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

1. Dapat mengetahui layanan kesehatan terdekat berdasarkan lokasi dan memberikan informasi informasi kesehatan berupa artikel, manfaat dan penggunaan obat, serta dokter dan rumah sakit sesuai spesialisasinya.
2. Dapat melakukan pemesanan obat, vitamin, dan produk kesehatan lainnya melalui website.
3. Dapat membuat janji kunjungan temu dokter di rumah sakit tanpa perlu mendaftarnya secara *offline.*

## Manfaat

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

1. Masyarakat dapat mengakses pelayanan kesehatan tanpa perlu meluangkan banyak waktu. Mereka dapat mengakses aplikasi kapan saja dan di mana saja.
2. Aplikasi ini menyediakan informasi tentang lokasi terdekat apotek dan rumah sakit. Masyarakat dapat mengetahui apotek terdekat untuk membeli obat dan rumah sakit terdekat untuk mendapatkan perawatan yang diperlukan.
3. 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.
4. Masyarakat dapat membuat janji kunjungan ke rumah sakit melalui aplikasi ini. Serta masyarakat dapat mengetahui jadwal praktek dokter yang ada pada rumah sakit.
5. Aplikasi ini menyediakan artikel kesehatan yang dapat meningkatkan pengetahuan dan wawasan masyarakat tentang kesehatan..

## Sistematika Penulisan

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

**BAB I PENDAHULUAN**

Pada bab pendahuluan dipaparkan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan masalah, serta manfaat penelitian aplikasi pelayanan kesehatan dmasyarakat berbasis web menggunakan *framework* vuejs.

**BAB II LANDASAN TEORI**

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

**BAB III METODE PELAKSANAAN**

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

**BAB IV HASIL DAN PEMBAHASAN**

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

**BAB V PENUTUP**

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

# **BAB II LANDASAN TEORI**

## Dokter

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

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

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

## Dokter 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.

## 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.

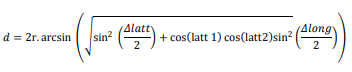
## Longitude

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

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

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



Keterangan:

*d*  : jarak

*r* : jari-jari bumi adalah 6371 km

Δlatt : Selisih latitude daerah asal dengan daerah tujuan

Δlong : Selisih longitude daerah asal dengan daerah tujuan

## API

API (Application Programming Interface) adalah 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 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

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

Agile development menekankan alur terjadi revisi maka akan dilakukan iterasi atau perulangan tanpa menunggu proses selesai terlebih dahulu. Berikut ini adalah tahapan dilakukannya pembuatan aplikasi pelayanan kesehatan masyarakat berbasis web menggunakan framework vue js dengan metode agile development:

## UML (Unified Modeling Language)

### Pengenalan UML

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

### *Use Case* Diagram

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

Tabel 2.1 Simbol *Use Case* Diagram

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

Sumber: Ahaddin, 2015.

### *Activity* Diagram

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

Tabel 2.2 Simbol *Activity* Diagram

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

Sumber: Idris, 2018.

## *Entity Relationship* Diagram(ERD)

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

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

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

Sumber: Setiadi, 2017.

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

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

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

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

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

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

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

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

## *Global Positioning System* (GPS)

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

## OpenStreetMap

OpenStreetMap (OSM) merupakan sebuah proyek berbasis 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 sederhana, performa dan penggunaan yang mudah. Leaflet dibuat agar dapat digunakan oleh sistem 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

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

1. *Product Backlog*

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

1. *Sprint Backlog*

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

1. *Daily Activity*

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

1. *Sprint Review*

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

## Analisis Kebutuhan Sistem

Dalam proses membangun aplikasi pelayanan kesehatan 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 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 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 oleh pengguna.

### Kebutuhan *Hardware*

Daftar kebutuhan *hardware* (perangkat keras) dalam proses pembangunan aplikasi pelayanan kesehatan 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 i5-6600U |
| 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 masyarakat berbasis web untuk analisis sentimen ini terdapat pada tabel 3.2.

Tabel 3.2 Tabel Kebutuhan Software

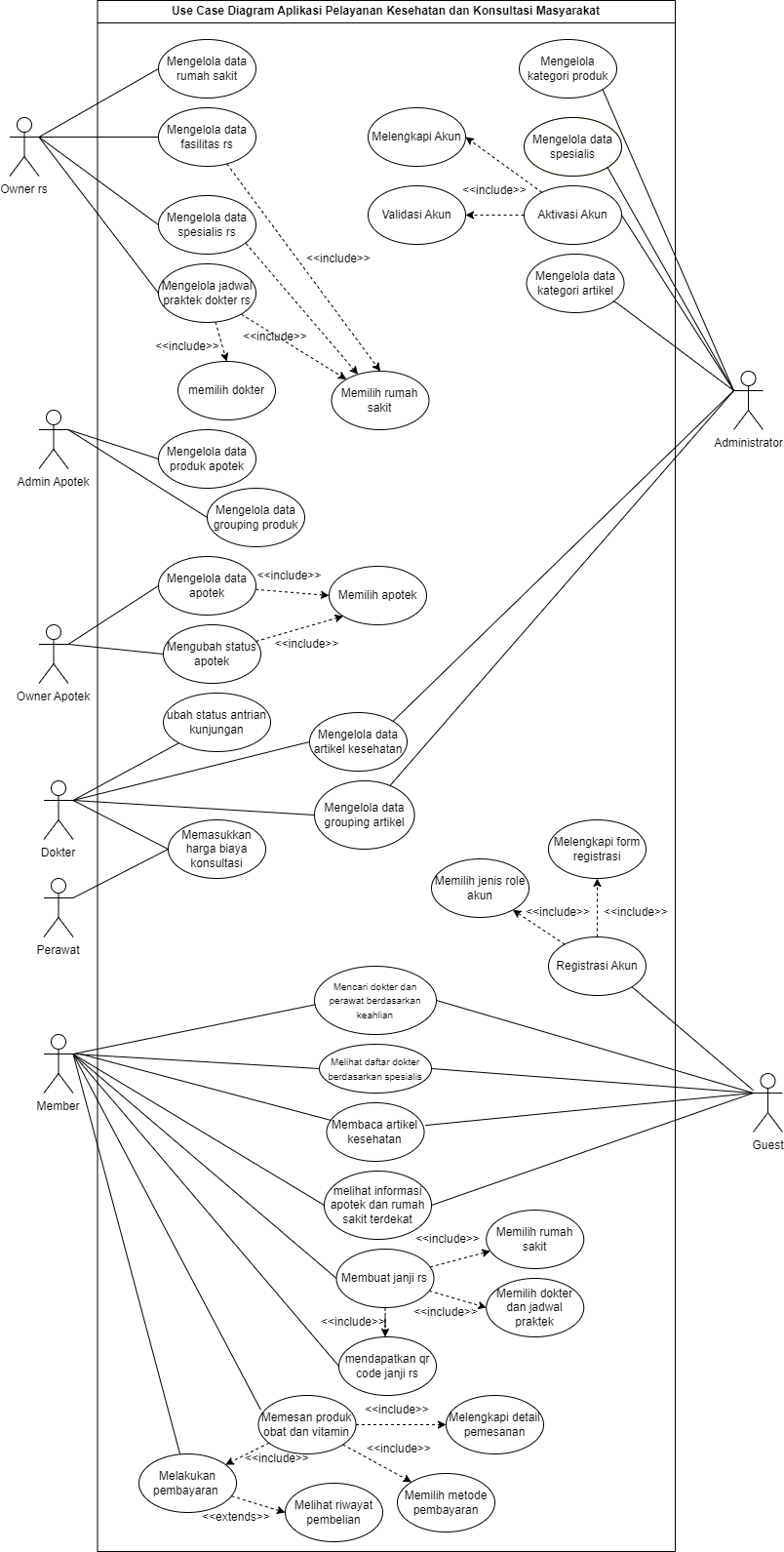
|  |  |  |
| --- | --- | --- |
| **No** | **Jenis *Software*** | **Kebutuhan *Software*** |
| 1. | Bahasa *Scripting* | PHP, JavaScript, CSS, VueJs |
| 2. | *Software* Pengolah | VsCode |
| 3. | Penyimpanan Data | Basis data (MySql) |
| 4. | Web Browser | Google Chrome, Microsoft Edge |
| 5 | Sistem 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 digunakan oleh pengguna sebagaimana mestinya yang tercantum dalam tujuan dan manfaat pembuatan aplikasi. Dalam penerapannya, sistem akan menggunakan *Unified Modelling Language* sebagai model pengembangan

### Use Case Diagram

*Use Case* Diagrammerupakan diagram yang menjelaskan hubungan antara actordengan sistem. Dalam aplikasi ini menjelaskan tujuh aktor yang memiliki beberapa aksi. Aktor tersebut adalah member, dokter, guest, perawat, pemilik rumah sakit, pemilik apotek, dan admin. Setiap aktor memiliki hak akses berbeda antar aktor lainnya. Adapun rancangan *Use Case* dari aplikasi pelayanan kesehatan 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 | Mengelola data apotek | Menu ini digunakan oleh owner apotek untuk melihat, menambahkan, mengubah, dan menghapus data apotek. |
| Mengubah status apotek | Menu ini digunakan oleh owner apotek untuk mengubah status dari apotek yang dimilikinya. |
| 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. |
| Mengelola data *grouping* produk | Menu ini digunakan oleh owner apotek untuk melihat, menambahkan, mengubah, dan menghapus data *grouping* produk. |
| 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. |
| Ubah status antrian kunjungan | Menu ini digunakan oleh dokter untuk mengubah status antrian konsumen ketika sudah menyelesaikan konsultasi secara tatap muka. |
| 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. |
| *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 | 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 spesialis. |
| Mengubah status akun dokter | Menu ini digunakan oleh admin untuk mengaktifkan akun dokter. Admin wajib validasi dokumen yang diunggah. Serta, menambahkan nomor STR. |
| 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

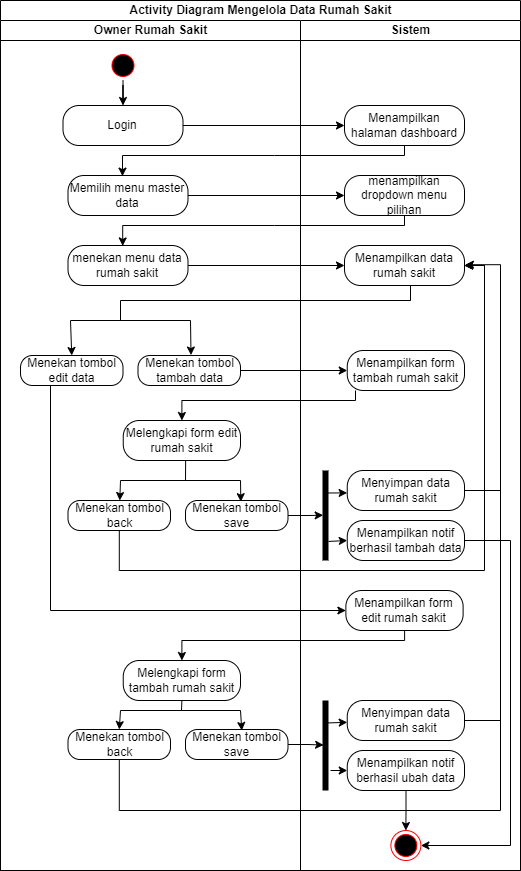
*Activity* Diagram menggambarkan aliran kerja, atau proses, dalam suatu sistem. Karena itu, mereka tergambar seperti flowchart. *Activity* Diagram mendokumentasikan perilaku internal yang ada dalam Use Case, antara Use Case, atau bisnis secara keseluruhan (Unhelkar, 2018).

Terdapat beberapa *activity* Diagramyang ada dalam proses perancangan sistem aplikasi pelayanan kesehatan 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. Owner rumah sakit melakukan tambah, edit, dan hapus data rumah sakit.

Dalam pengelolaan data rumah sakit, admin rumah sakit dapat mengisi nama, deskripsi, dan lokasi rumah sakit. Lokasi rumah sakit berupa latitude dan longitude.

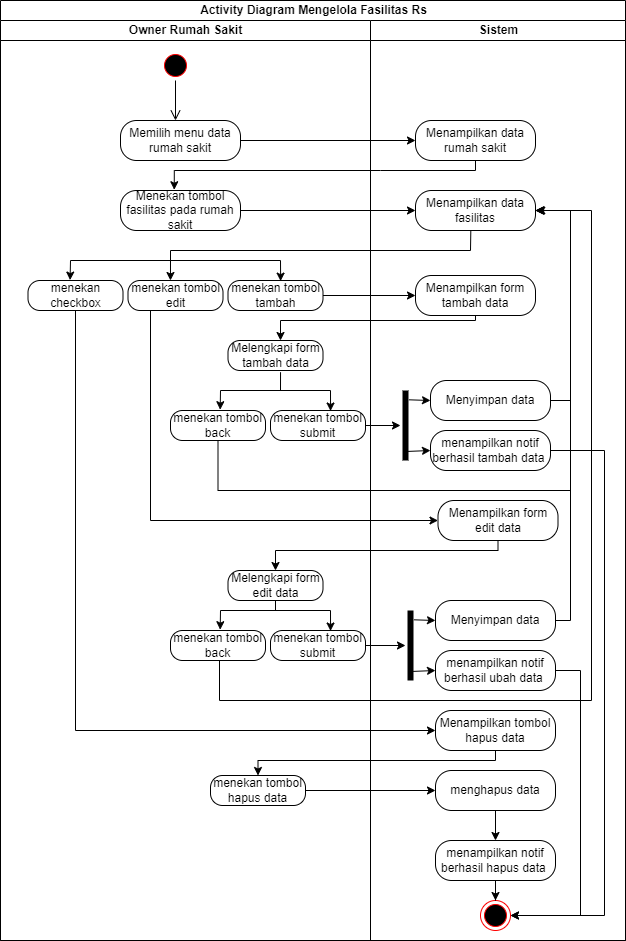


Gambar 3.2 Activity Diagram Mengelola Data Rumah Sakit

*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. Untuk mengelola data fasilitas, user harus memilih rumah sakit yang akan ditambahkan data fasilitasnya. Menu ini dapat menambah, mengubah, dan menghapus data fasilitas rumah sakit.

Admin rumah sakit melakukan login dengan akun pemilik rumah sakit. Sistem akan menampilkan data rumah sakit berdasarkan pemiliknya. Ketika akan memasukkan atau menambahkan fasilitas pada rumah sakit yang akan dipilih. User perlu melengkapi form fasilitas. Dalam form tersebut terdapat nama fasilitas dan id rumah sakit. Sesudahnya menekan tombol simpan, untuk menambahkan data.

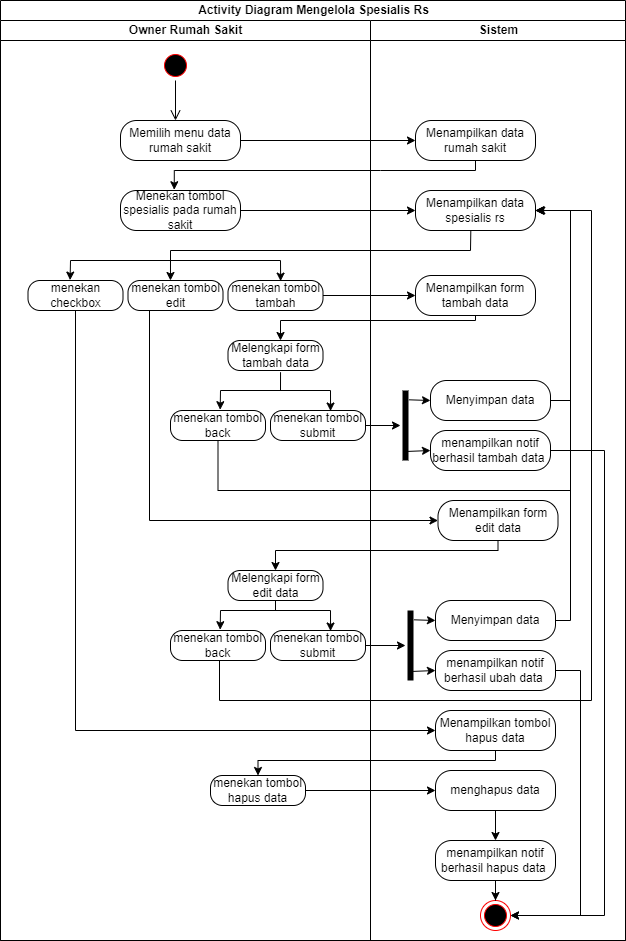


Gambar 3.3 Activity Diagram Mengelola Fasilitas Rumah Sakit

*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 ketika di-klik maka akan menampilkan data spesialis rumah sakit.

Pada halaman spesialis rumah sakit, terdapat tombol tambah, edit, dan hapus data. Ketika menekan tombol tambah data sistem 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. Ketika ingin menghapus data, owner rumah sakit perlu menekan *checkbox* lalu akan menampilkan tombol hapus data. Menekan tombol hapus data, maka data yang dipilih akan terhapus dari sistem.

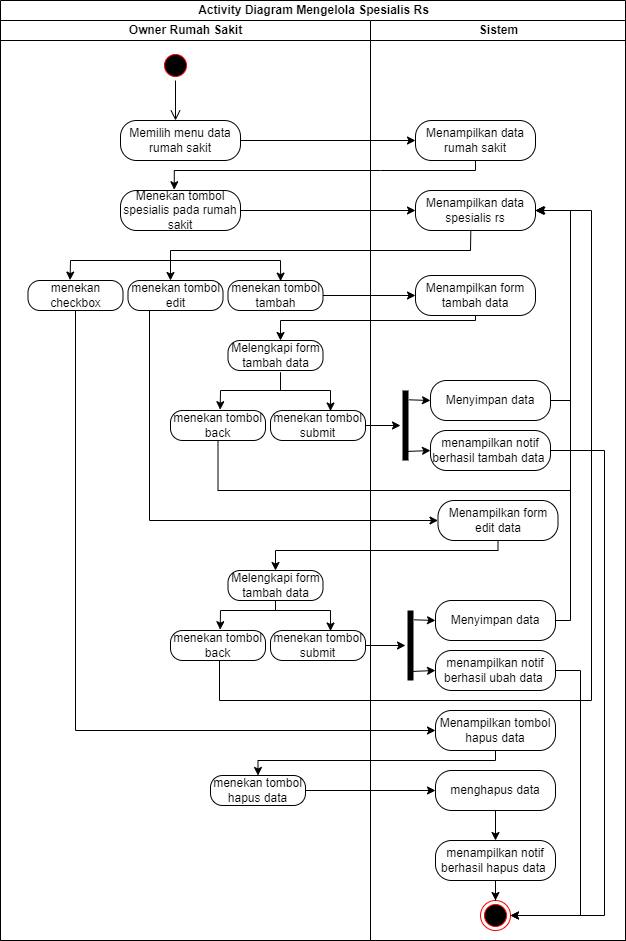


Gambar 3.4 Acitivity Diagram Mengelola Data Spesialis Rumah Sakit

*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. Untuk mengelola jadwal praktek dokter di rumah sakit, user perlu memilih menu rumah sakit terlebih dahulu.

Menu rumah sakit menampilkan data rumah sakit, ketika ingin mengelola jadwal praktek, user perlu menekan tombol lihat jadwal praktek. Pada halaman jadwal praktek terdapat nama dokter. Ketika akan melakukan tambah data atau edit data, user menekan tombol jadwal praktek. Di halaman tersebut terdapat jadwal praktek dokter tersebut tersedia di hari apa saja.

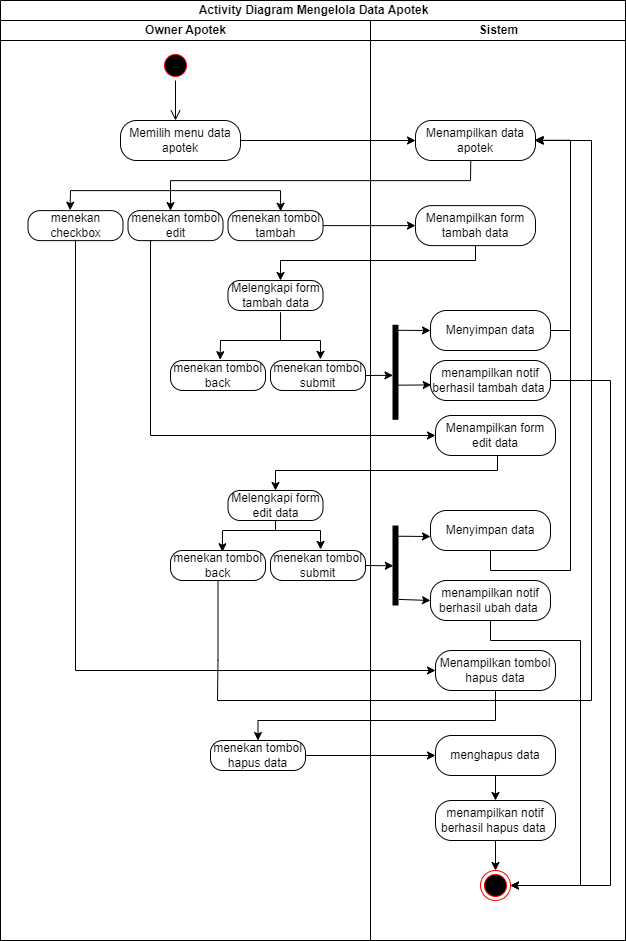


Gambar 3.5 Activity Diagram Mengelola Jadwal Praktek Dokter Rumah Sakit

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. Ketika aktor menekan tombol hapus data, maka sistem akan menghapus data apotek yang sudah dipilih.

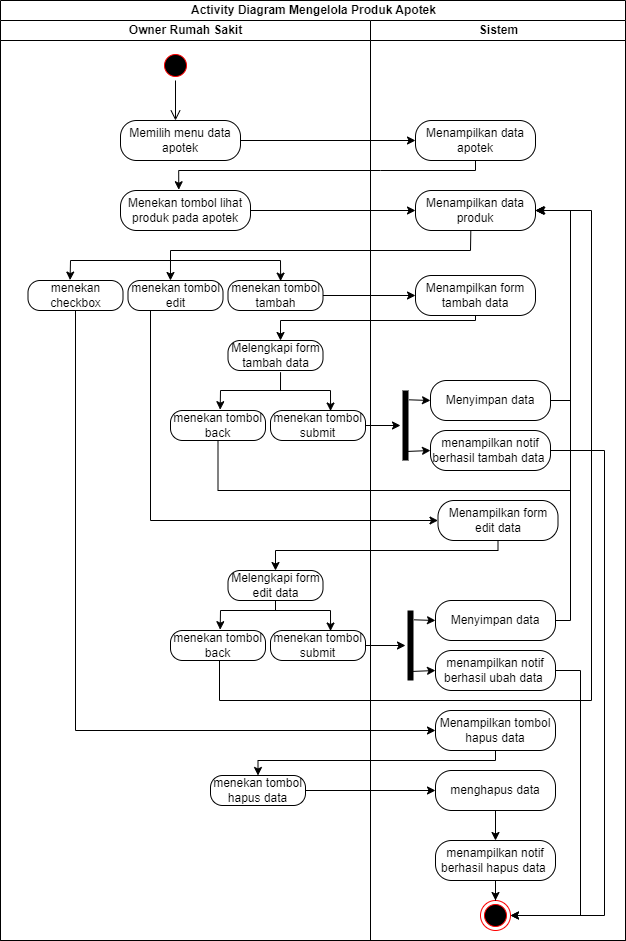


Gambar 3.6 Activity Diagram Mengelola Data Apotek

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.

Proses menambahkan data produk apotek, aktor perlu melengkapi form tambah data. Dan ketika hendak 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.

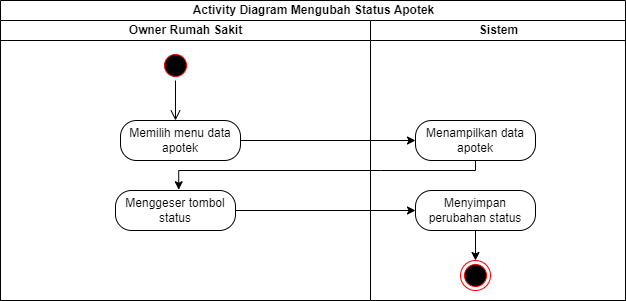


Gambar 3.7 Activity Diagram Mengelola Data Produk Apotek

1. *Activity* Diagram Mengubah Status Apotek

Pada *activity* diagram ini owner apotek dapat melakukan perubahan status apotek. Status apotek digunakan untuk mengetahui bahwa apotek tersebut dapat menerima pembelian atau tidak. Ketika 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.



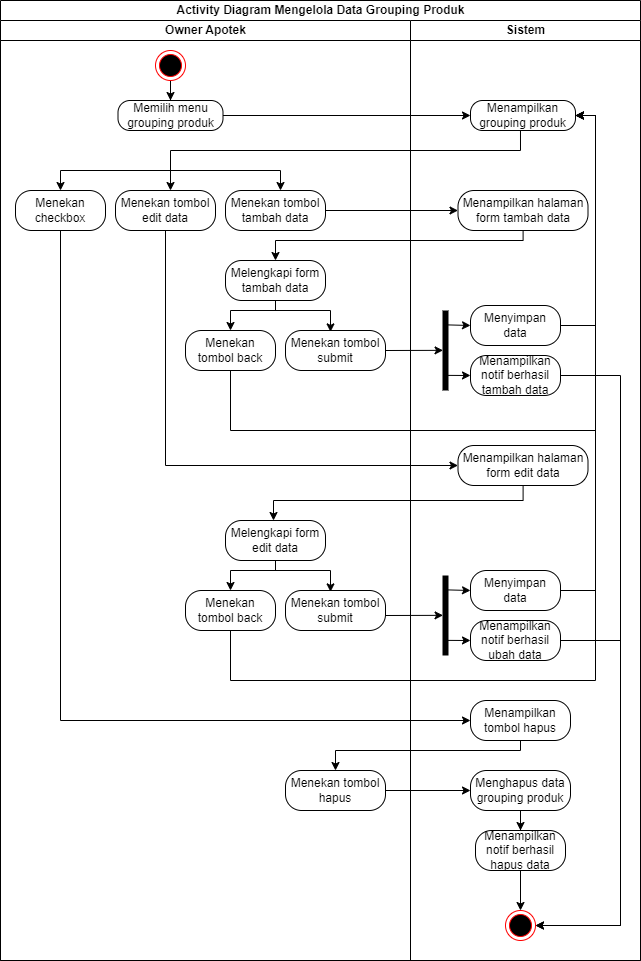
Gambar 3.8 Activity Diagram Mengubah Status Apotek

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.*

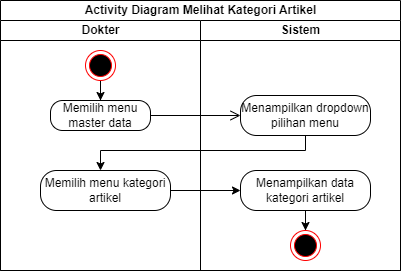
Data akan tersimpan di dalam sistem. Untuk edit, langkahnya hampir sama seperti tambah data. Dalam menghapus data, owner apotek harus menekan *checkbox* untuk menampilkan tombol hapus data. Kemudian dengan menekan tombol hapus data, maka data akan terhapus dari sistem. Adapun *activity* diagram mengelola data produk apotek terdapat dalam Gambar 3.9.



Gambar 3.9 Activity Diagram Mengelola Data Grouping Produk

1. *Activity* Diagram Melihat Kategori Artikel

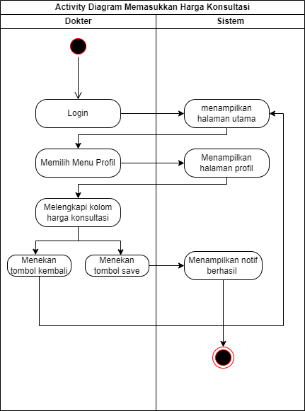
Pada *activity* diagram ini dokter dapat melihat kategori artikel. Dokter dapat melihat kategori artikel dengan melakukan aktivitas memilih menu master data pada dashboard, kemudian sistem akan menampilkan *dropdown* pilihan menu, selanjutnya aktor dokter memilih menu kategori artikel. Sesudahnya sistem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

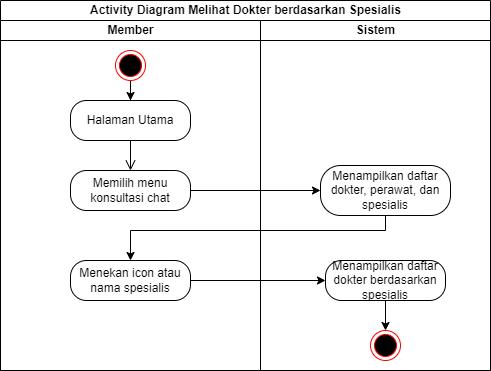
Pada *activity* diagram ini aktor dokter dan perawat menetapkan harga biaya konsultasi. Aktivitas yang pertama dilakukan adalah login aplikasi, sesudahnya memilih menu profil. Lalu pada kolom biaya, lengkapi dengan memasukkan nominal biaya. Adapun *activity* diagram terdapat dalam Gambar 3.11.



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.

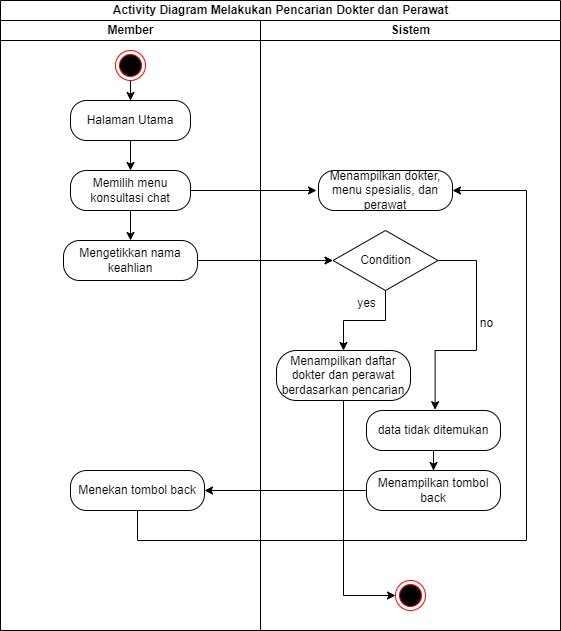


Gambar 3.12 Activity Diagram Melihat Dokter berdasarkan Spesialis

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 kemudian memilih menu konsultasi chat. Sesudahnya mengetikkan nama keahlian yang hendak dicari.

Sistem akan menampilkan dokter dan perawat berdasarkan keahlian yang diketik. Ketika data yang diketik tersedia atau ada di dalam database, maka akan menampilkan daftar dokter dan perawat. Ketika tidak ada data yang dapat ditampilkan pada halaman maka sistem menampilkan pesan data tidak ada lalu mengembalikan ke halaman sebelumnya jika menekan tombol *back.* Adapun *activity* diagram ini dapat dilihat pada Gambar 3.13.

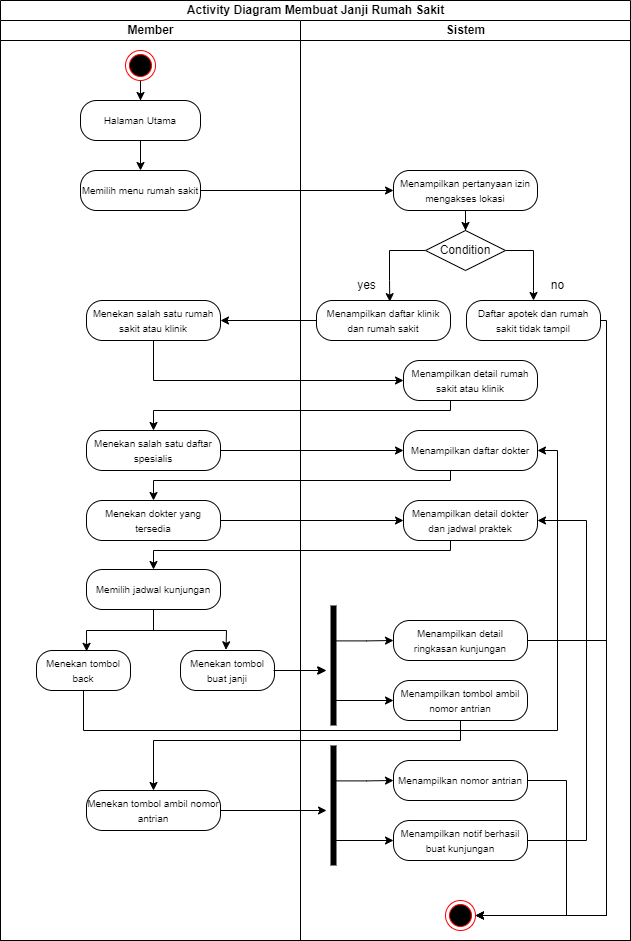


Gambar 3.13 Activity Diagram Melakukan Pencarian Dokter dan Perawat

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.

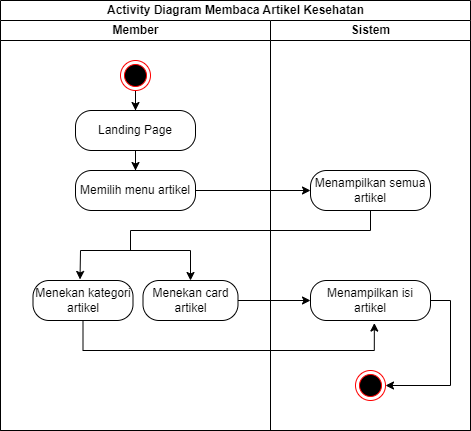
Selanjutnya, *member* menekan salah satu rumah sakit yang akan dikunjungi, setelahnya sistem akan menampilkan detail informasi dan data spesialis yang tersedia pada rumah sakit. Untuk membuat janji kunjungan, *member* perlu login ke dalam aplikasi terlebih dahulu. Lalu, *member* menekan salah satu nama 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.

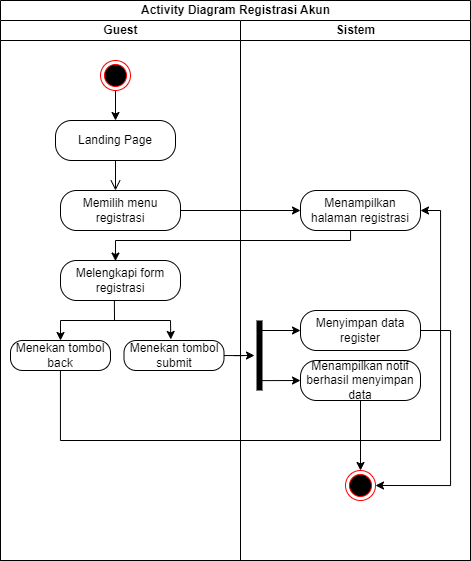


Gambar 3.15 Activity Diagram Membaca Artikel Kesehatan

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 masyarakat berbasis web. Aktivitas yang pertama dilakukan adalah membuka halaman utama aplikasi.

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.

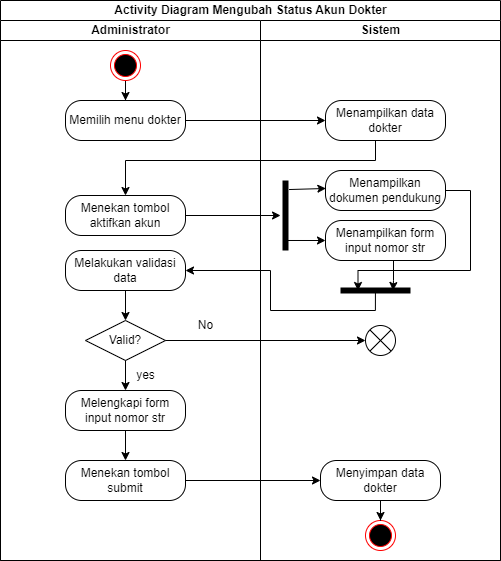
Pada *activity* diagram ini member melakukan pemesanan obat dan vitamin di apotek atau toko kesehatan yang tersedia dalam aplikasi. Aktivitas yang pertama dilakukan adalah login ke aplikasi, kemudian memilih menu toko kesehatan, kemudian sistem akan menampilkan seluruh obat dan vitamin.

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).



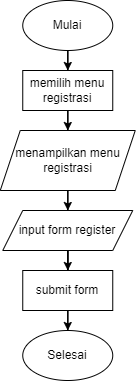
Gambar 3.17 Activity Diagram Mengubah Status Akun Dokter

## Flowchart

Flowchart yang akan dibangun pada aplikasi pelayanan kesehatan masyarakat berbasis web ini terdapat beberapa flowchart. Di antaranya adalah Flowchart registrasi akun, flowchart membaca artikel, flowchart melihat rumah sakit terdekat, flowchart memesan produk apotek, dan flowchart membuat janji kunjangan rumah sakit. Adapun rancangan flowchart dapat dilihat sebagai berikut.

### Flowchart Registrasi Akun

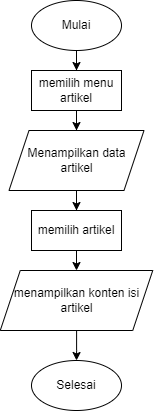
Flowchart berikut merupakan proses untuk melakukan pendaftaran akun. *Guest* akan memilih menu registrasi dan kemudian mengisi form registrasi. Di dalam form registrasi, *guest* memilih jenis akun, untuk jenis akun yang tersedia dalam aplikasi ini terdapat akun *member*, dokter, perawat, owner rumah sakit, dan owner apotek. Jika mendaftar sebagai *member, guest* tidak perlu untuk mengunggah dokumen pendukung. Jika mendaftar akun selain akun *member, guest* akan menunggu admin aplikasi untuk validasi dokumen pendukung. Setelahnya di-*submit*. Maka, data registrasi akan tersimpan.



Gambar 3.18 Flowchart Register

### Flowchart Membaca Artikel

Flowchart berikut merupakan proses untuk membaca artikel kesehatan. Proses yang pertama dilakukan adalah memilih menu artikel, sistem akan menampilkan keseluruhan artikel yang dapat dibaca oleh pengguna. Untuk dapat membaca artikel, user menekan salah satu artikel. Kemudian sistem akan menampilkan konten artikel tersebut setelah dipilih. Konten artikel berupa judul, deskripsi, dan foto artikel. Adapun flowchart membaca artikel dapat dilihat pada gambar 3.19.



Gambar 3.19 Flowchart Membaca Artikel

### Flowchart Melihat Rumah Sakit Terdekat

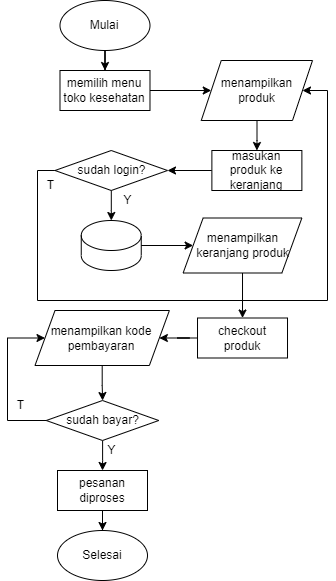
Flowchart berikut merupakan proses untuk melihat rumah sakit terdekat berdasarkan lokasi. *User* perlu mengizinkan aplikasi untuk mengakses lokasi pengguna. Hal tersebut dilakukan untuk mendapatkan lokasi *user* dan rumah sakit terdekat dari pengguna. Ketika pengguna tidak mengizinkan sistem untuk mengakses lokasi pada browser, maka daftar rumah sakit tidak akan tampil pada halaman. Adapun flowchart melihat rumah sakit terdekat dapat dilihat pada gambar 3.20.



Gambar 3.20 Flowchart Melihat Rumah Sakit Terdekat

### Flowchart Memesan Produk Apotek

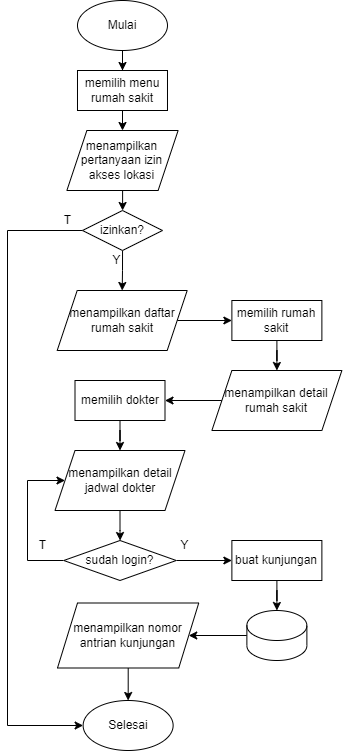
Flowchart berikut merupakan proses untuk memesan produk apotek. Proses yang pertama adalah memilih menu toko kesehatan, lalu menampilkan produk. *Member* dapat memilih produk yang akan dibeli. Terdapat seluruh produk dan kategorinya. Untuk membeli produk, *member* dapat memasukkannya ke dalam keranjang terlebih dahulu ataupun dapat membaca deskripsi produknya. Jika belum melakukan login, maka pengguna tidak akan bisa melakukan *checkout* produk. Adapun flowchart memesan produk apotek dapat dilihat pada gambar 3.21.



Gambar 3.21 Flowchart Memesan Produk Apotek

### Flowchart Membuat Janji Kunjungan Rumah Sakit

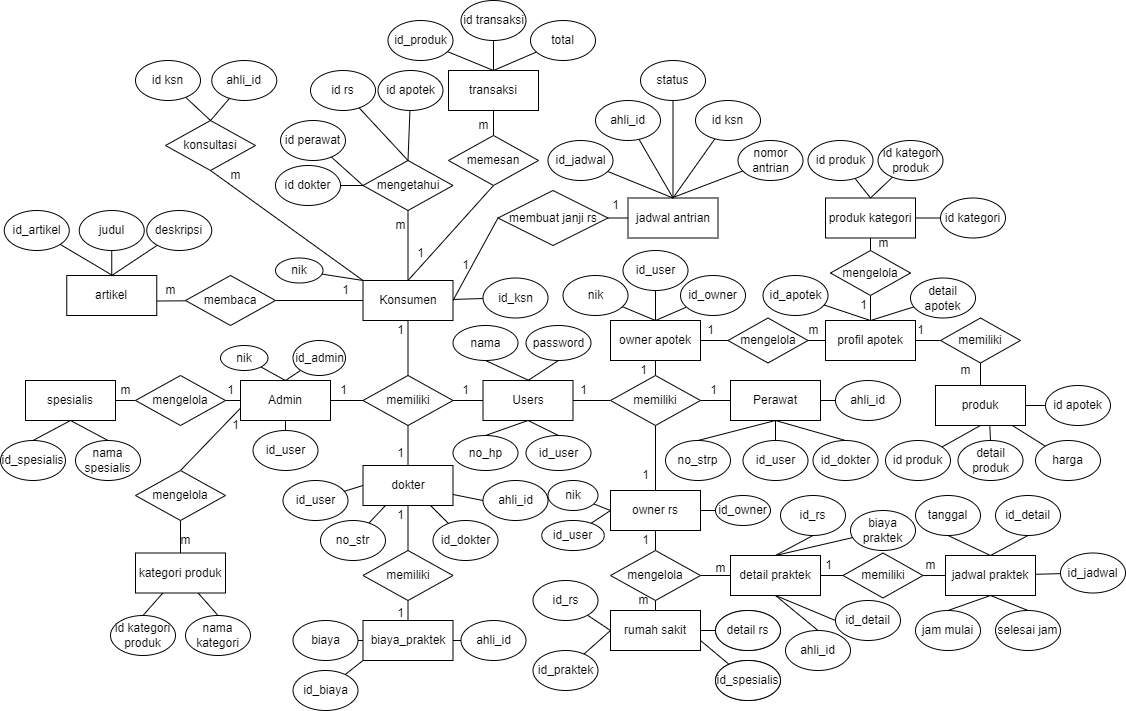
Flowchart berikut merupakan proses untuk membuat janji kunjungan rumah sakit. Proses dalam mebuat janji adalah dengan mengakses menu rumah sakit. Kemudian memilih dokter yang tersedia dalam halaman tersebut. Halaman buat janji menampilkan jadwal dan harga praktek dokter. Jadwal praktek dikelola oleh administrator 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 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. |
| 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 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 Penyakit

|  |  |  |  |
| --- | --- | --- | --- |
| **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 Produk Kategori

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. | longitude | 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. | Nama | varchar(50) | NOT NULL |
| 2. | Email | varchar(50) | NOT NULL |
| 3. | Password | varchar(50) | NOT NULL |
| 4. | Nomor\_hp | varchar(50) | NOT NULL |
| 5. | alamat | varchar(50) | NOT NULL |
| 6. | Id\_role | text | NOT NULL |
| 7. | token | double | NOT NULL |
| 8. | foto | varchar(255) | DEFAULT NULL |
| 9. | status | varchar(255 | NOT NULL |
| 10. | Jenis kelamin | varchar(255 | NOT NULL |

## Perancangan Desain Tampilan Antarmuka

Dalam Rancangan Antarmuka membuat aplikasi web untuk aplikasi pelayanan kesehatan 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:

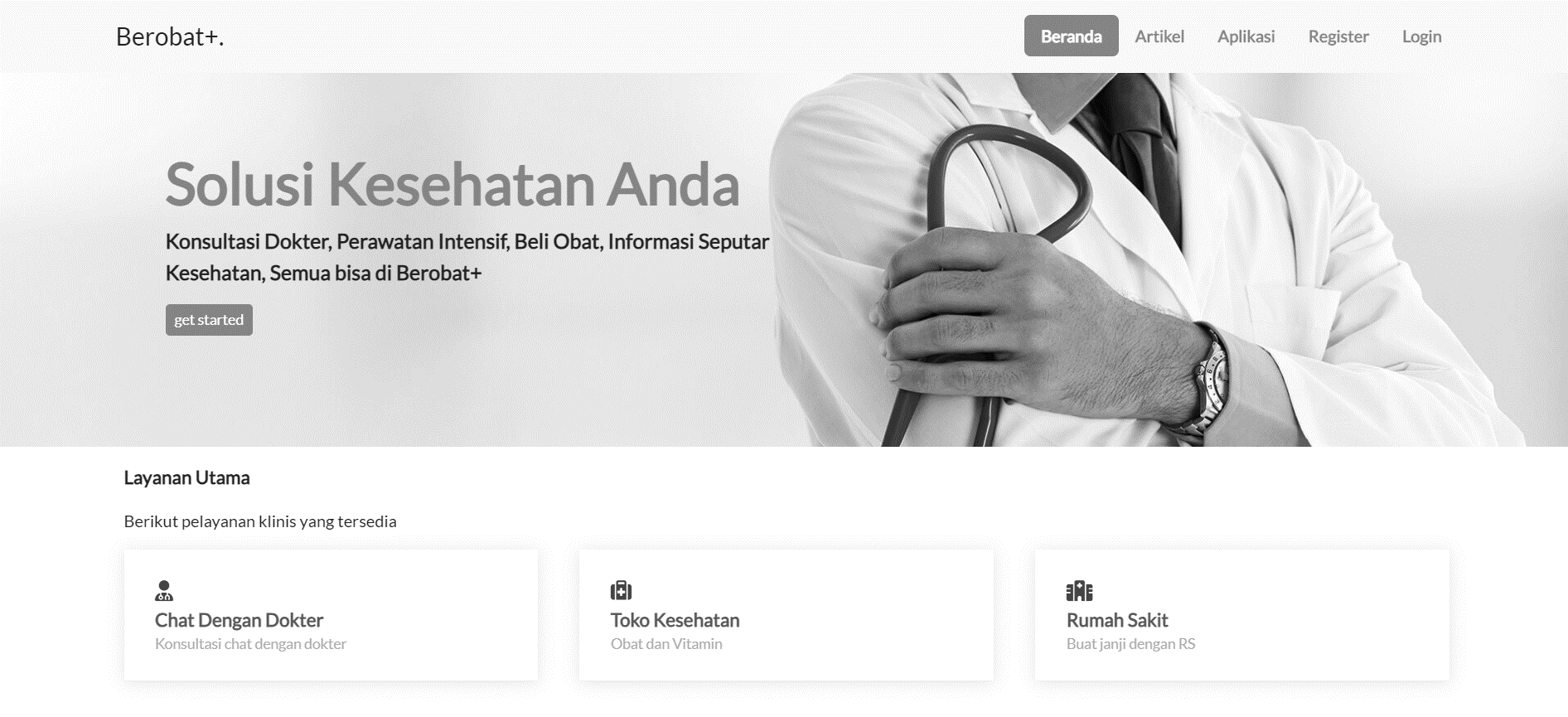
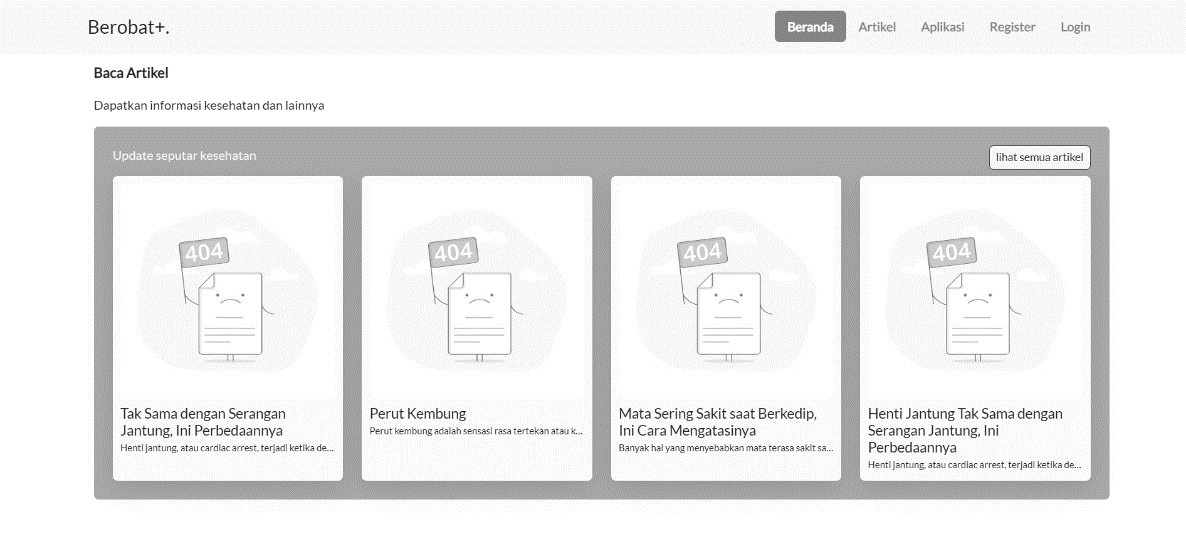
### Rancangan Antarmuka *Guest*

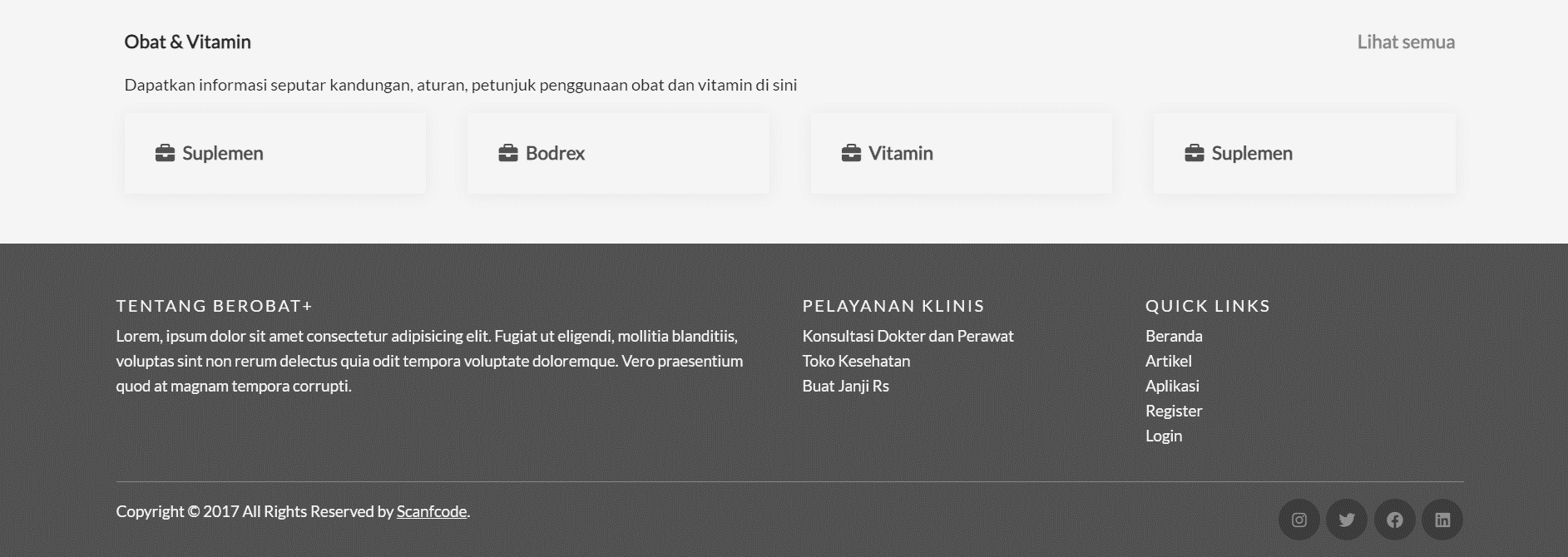
Rancangan antarmuka aplikasi pelayanan kesehatan masyarakat ini merupakan rancangan *user interface* dari sisi *guest.* Berikut adalah rancangan antarmuka *guest*:

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.

Section yang pertama adalah chat dokter, menu tersebut digunakan untuk menampilkan data dokter, perawat, dan spesialis. Section kedua adalah toko kesehatan, pengguna dapat membeli produk melalui menu toko kesehatan. Menu toko kesehatan menampilkan keseluruhan data produk. Pengguna dapat membeli obat berdasarkan kategorinya dan juga tidak. Selanjutnya adalah section janji kunjungan ke rumah sakit. Menu janji kunjungan rumah sakit digunakan oleh pengguna untuk membuat janji kunjungan. Pengguna juga dapat membaca artikel melalui halaman utama. Terdapat empat artikel yang dapat dibaca. Adapun rancangan antarmuka halaman *home* dapat dilihat pada gambar 3.19.

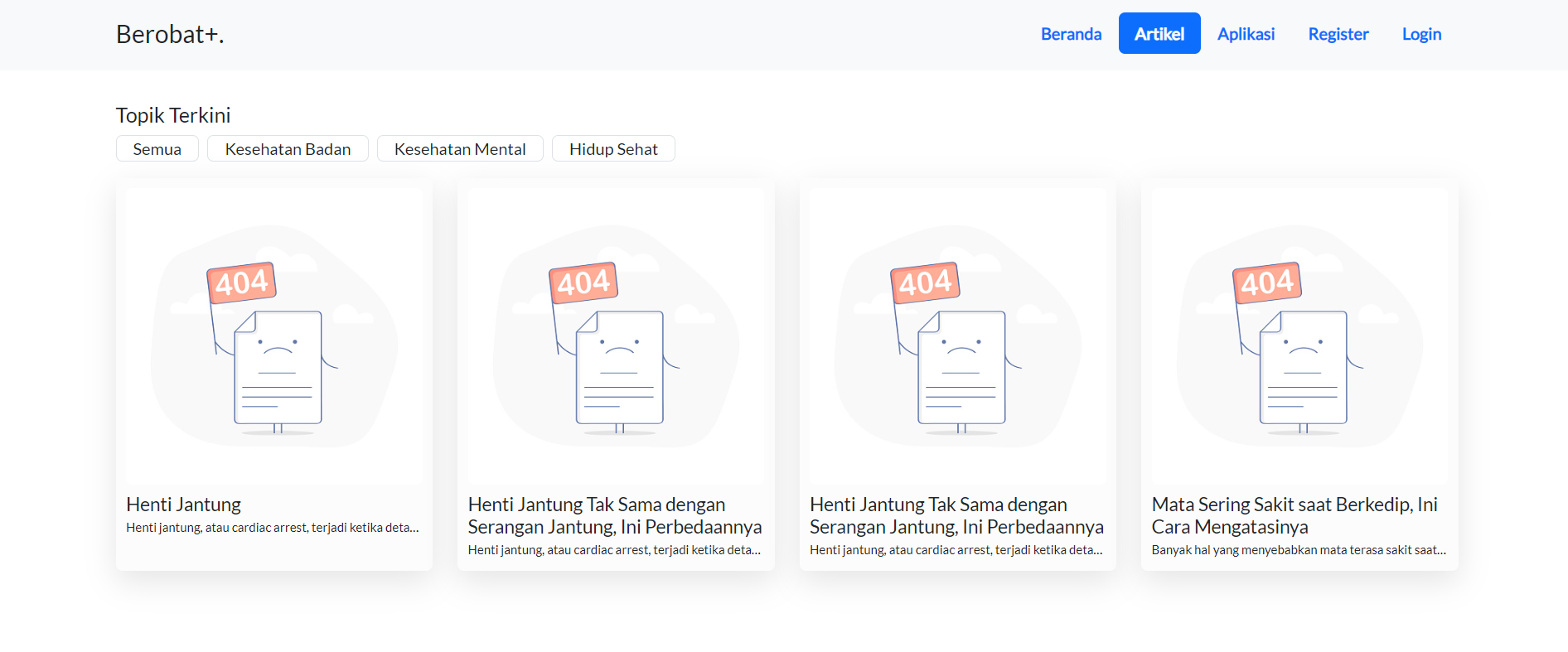




Gambar 3.24 Rancangan Antarmuka Halaman Home

1. Rancangan antarmuka halaman artikel

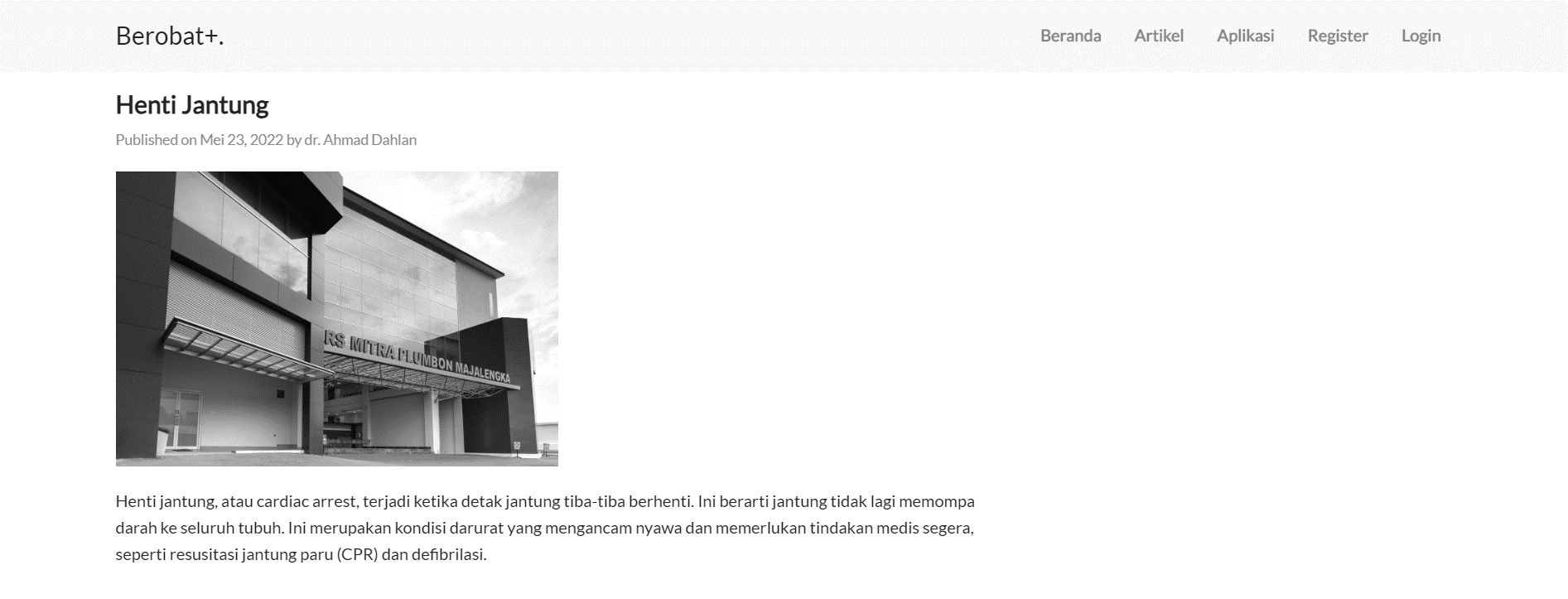
Pada halaman ini terdapat artikel yang dapat dibaca oleh *guest.* Dengan menekan teks artikel pada bagian *navbar,* maka akan menampilkan keseluruhan artikel kesehatan. Rancangan interface halaman artikel terdapat kategori atau topik yang dapat di-klik ketika *guest* hendak membaca artikel berdasarkan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

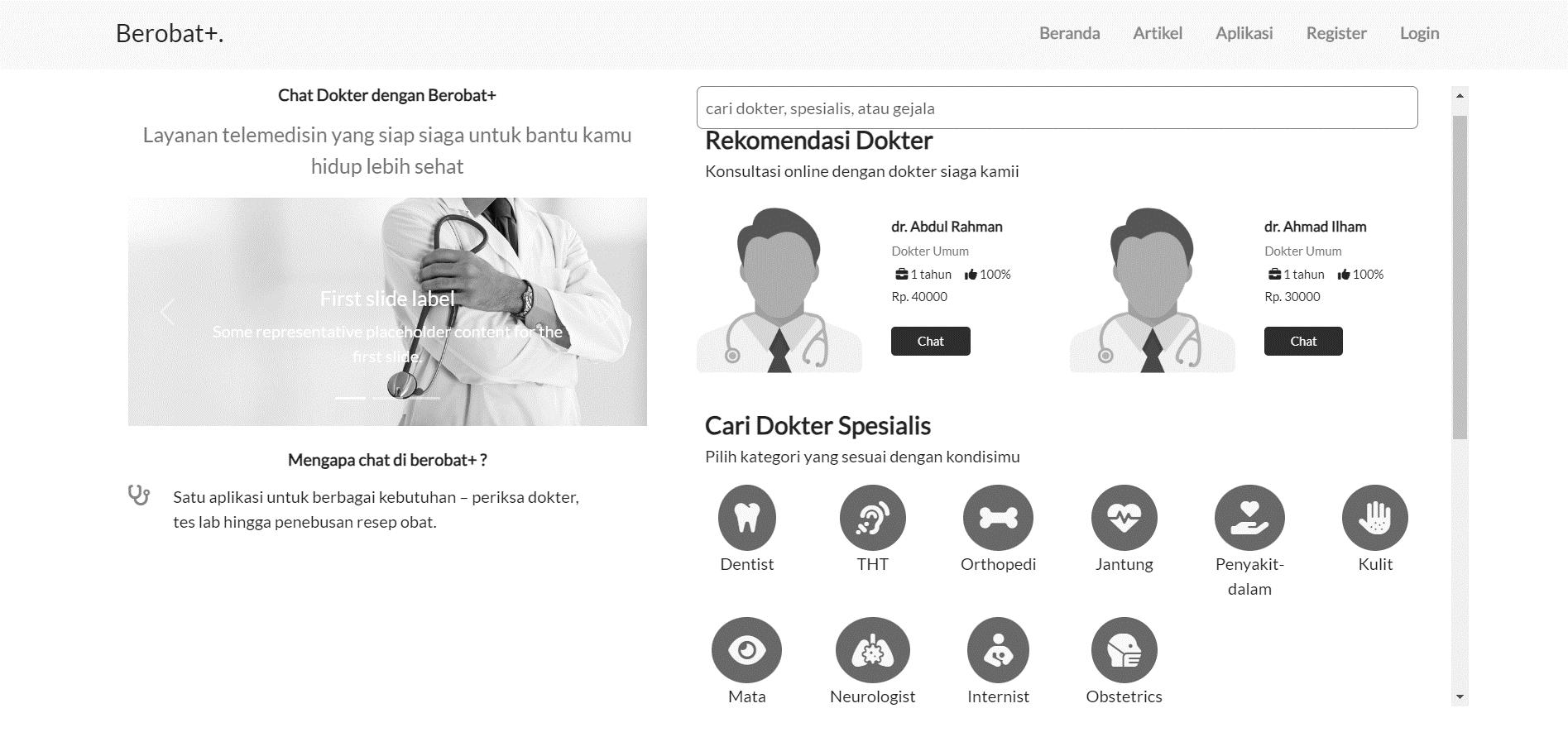
Pada rancangan antarmuka halaman detail artikel terdapat konten dari artikel yang di-klik oleh *guest.* Konten artikel terdapat judul artikel, deskripsi artikel, foto, dan *author*-nya. Adapun rancangan antarmuka halaman detail artikel dapat dilihat pada gambar 3.21.



Gambar 3.26 Rancangan Antarmuka Halaman Detail Artikel

1. Rancangan antarmuka konsultasi kesehatan dokter dan perawat

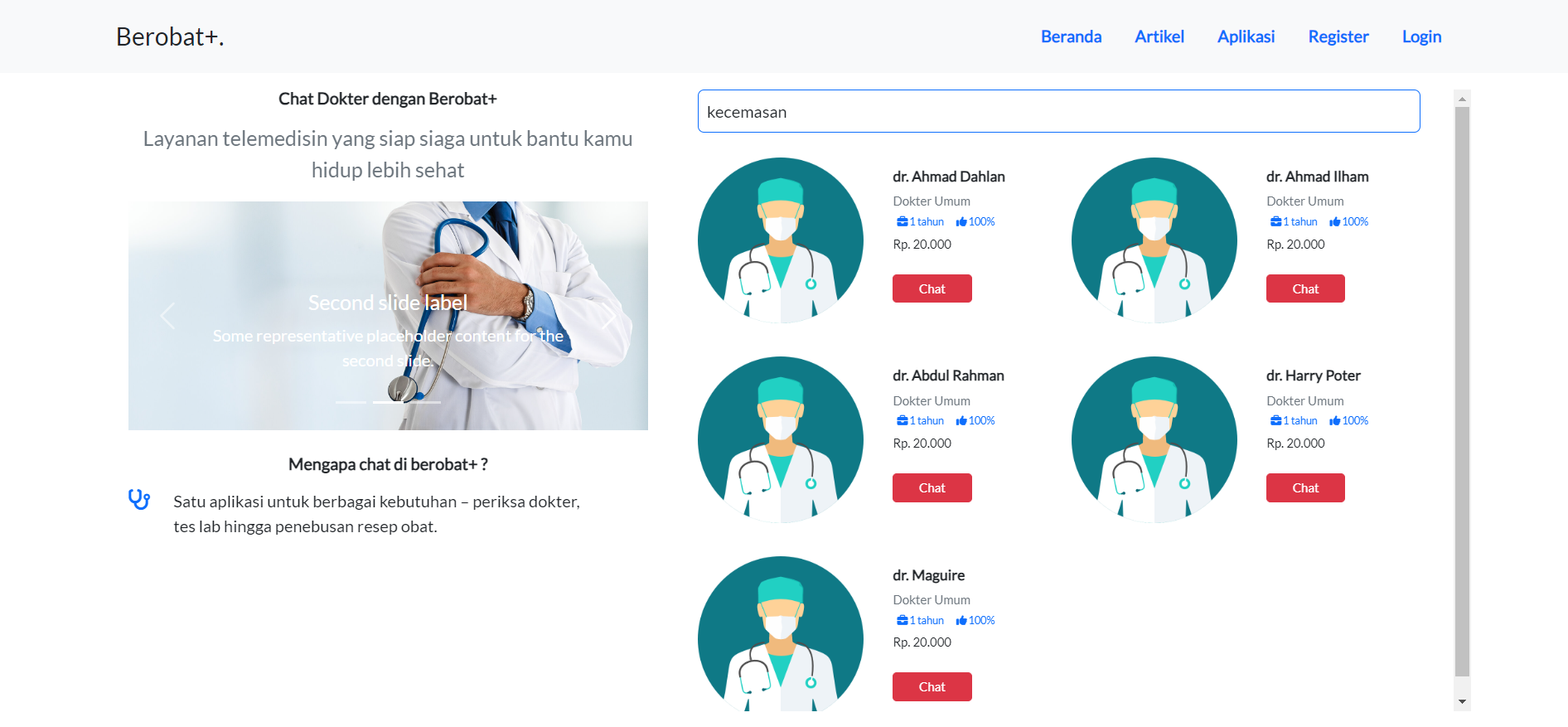
Rancangan antarmuka halaman konsultasi kesehatan dokter dan perawat terdapat konten berupa rekomendasi dokter, dan daftar spesialis yang tersedia pada aplikasi pelayanan kesehatan masyarakat. Dokter yang ditampilkan pada halaman ini dibatasi yaitu hanya dua dokter. Adapun rancangan antarmuka halaman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

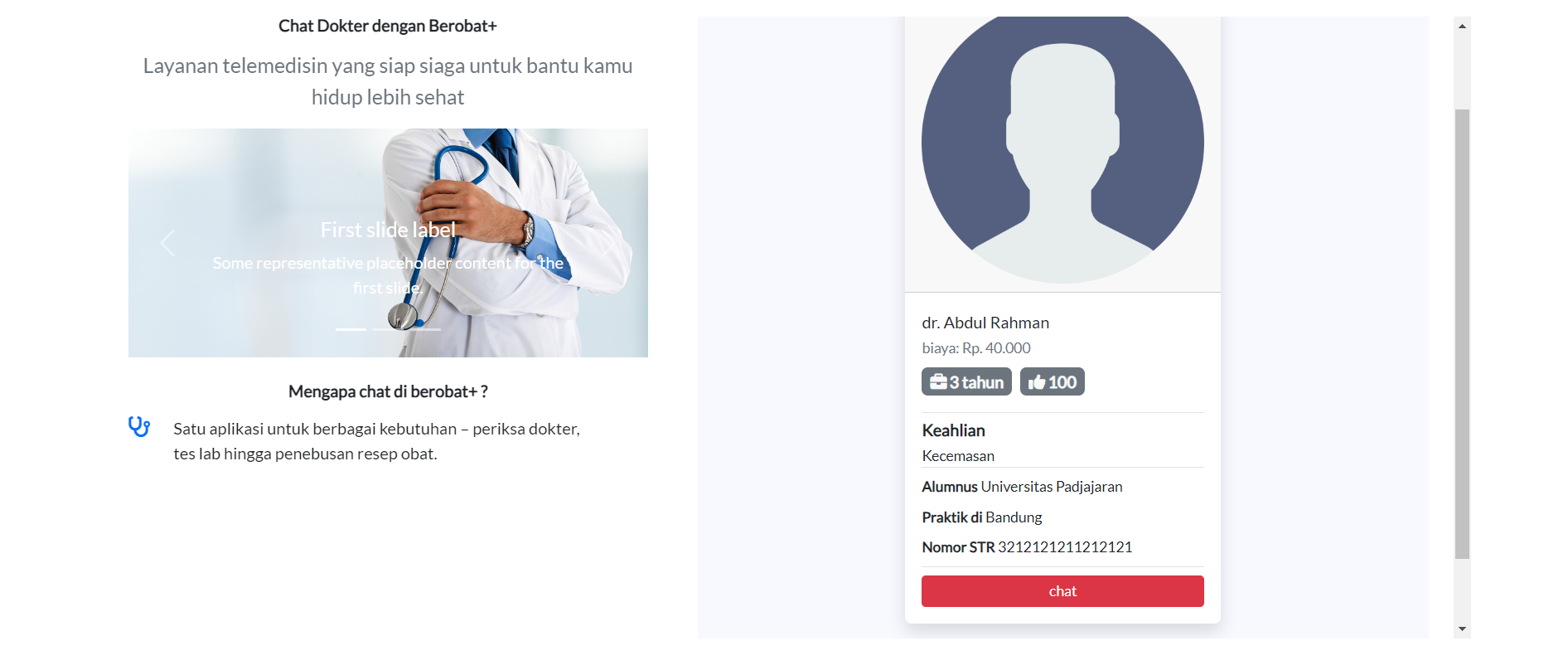
Rancangan *interface* ini merupakan halaman untuk melakukan pencarian dokter dan perawat melalui *form input* yang disediakan oleh aplikasi. *Guest* dapat mencari dokter atau perawat dengan mengetikkan nama tenaga medisnya. Adapun rancangan antarmuka halaman konsultasi kesehatan dokter dan perawat dapat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



Gambar 3.28 Rancangan Antarmuka Halaman Konsultasi

1. Rancangan antarmuka halaman detail dokter

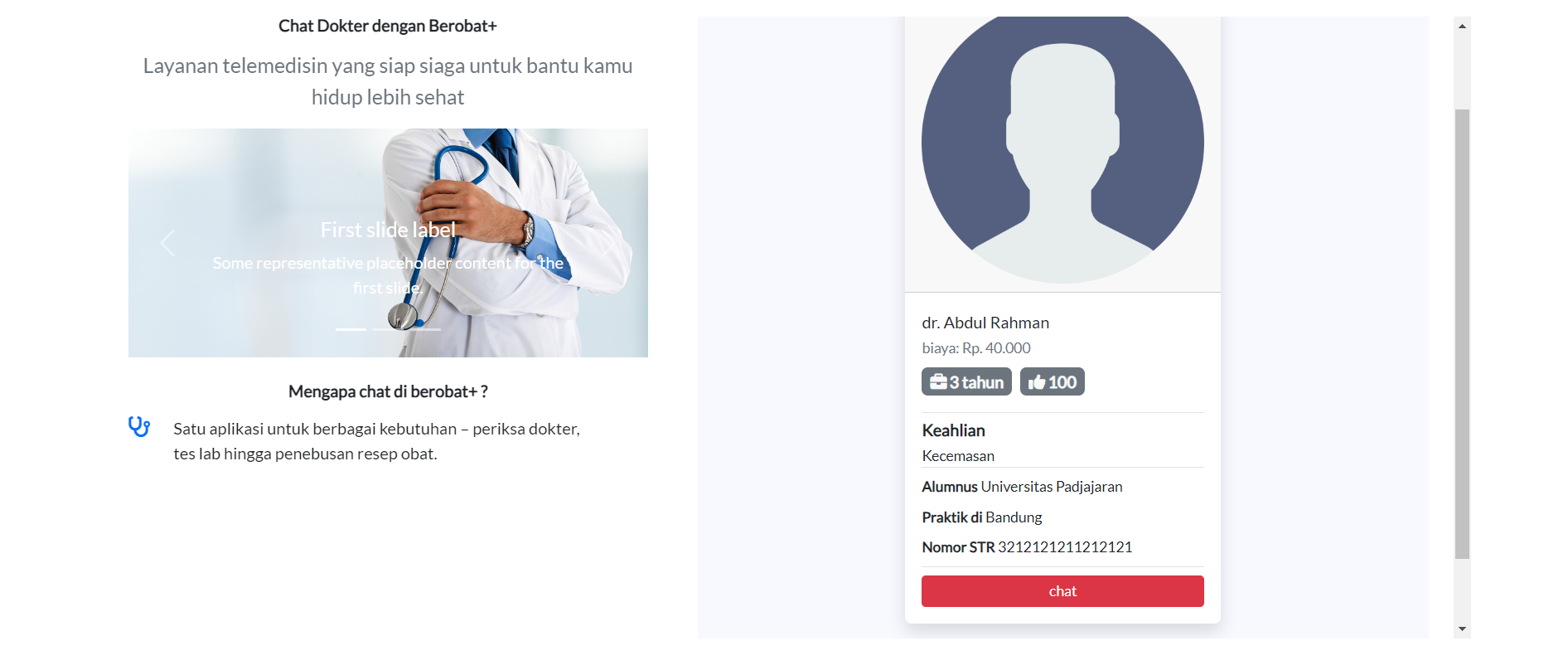
Rancangan antarmuka ini berisi detail dari dokter. Terdapat nama, biaya konsultasi, nama keahlian, dan nomor STR. Untuk mengetahui halaman ini *guest* perlu memilih dokter terlebih dahulu. Adapun rancangan antarmuka halaman detail dokter dapat dilihat pada gambar 3.24.



Gambar 3.29 Rancangan Antarmuka Halaman Detail Dokter

1. Rancangan antarmuka halaman detail dokter berdasarkan spesialis

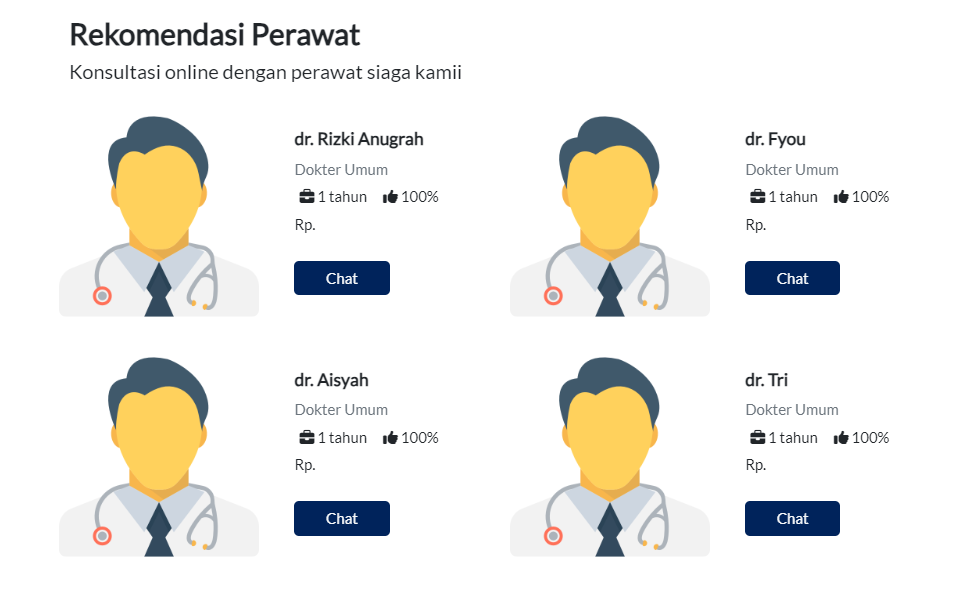
Rancangan antarmuka ini berisi detail dari dokter berdasarkan spesialisnya. Terdapat nama, biaya konsultasi, nama keahlian, dan nomor STR. Untuk mengetahui halaman ini *guest* perlu memilih dokter terlebih dahulu. Adapun rancangan antarmuka halaman detail dokter berdasarkan spesialis dapat dilihat pada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

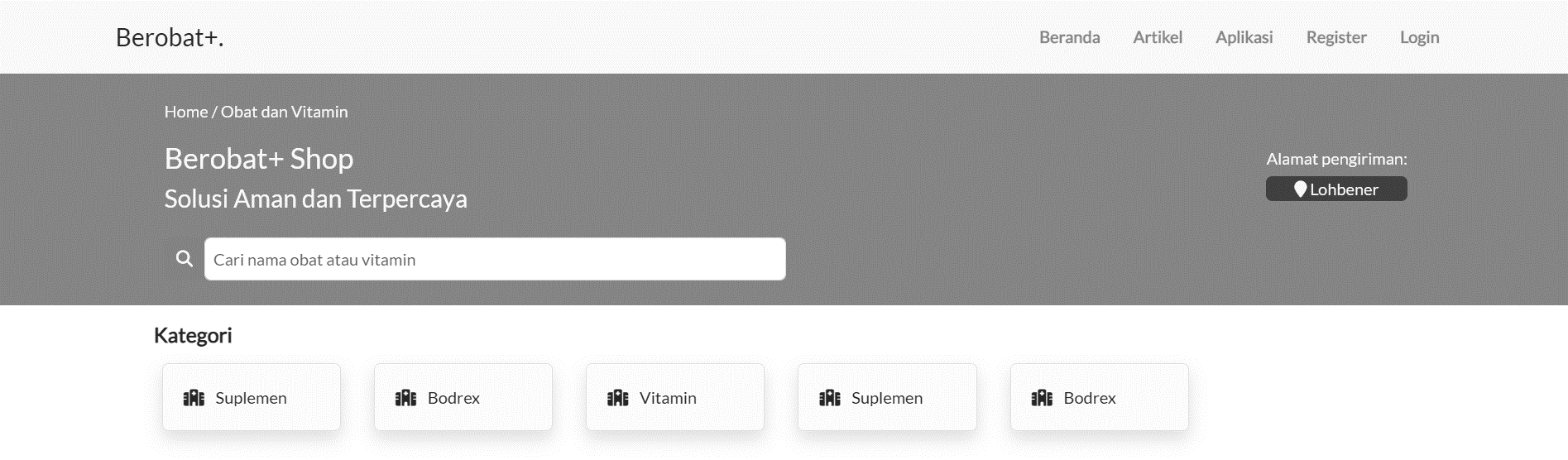


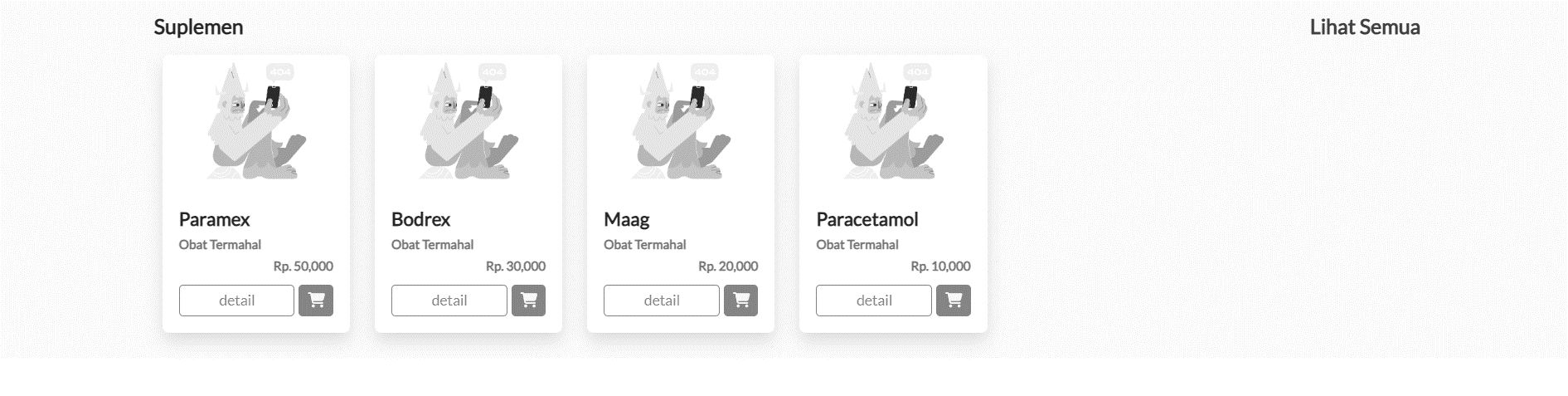
Gambar 3.31 Rancangan Antarmuka Halaman Perawat

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. Pada perancangan ini disediakan input untuk melakukan pencarian data produk. Lalu di halaman ini juga menampilkan lokasi pengguna. Pengguna atau member dapat membeli produk atau mencari produk berdasarkan kategori produknya. Adapun rancangan antarmuka halaman apotek dapat dilihat pada gambar 3.27.

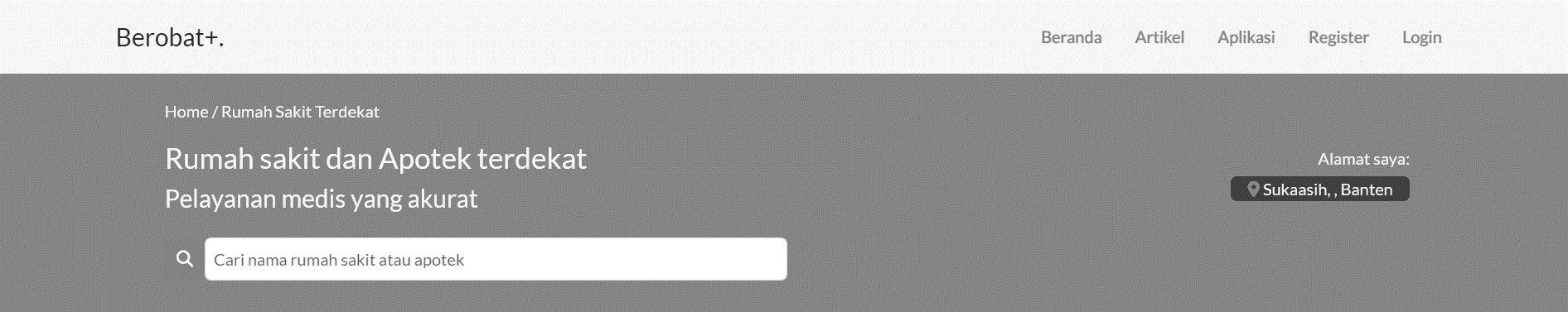


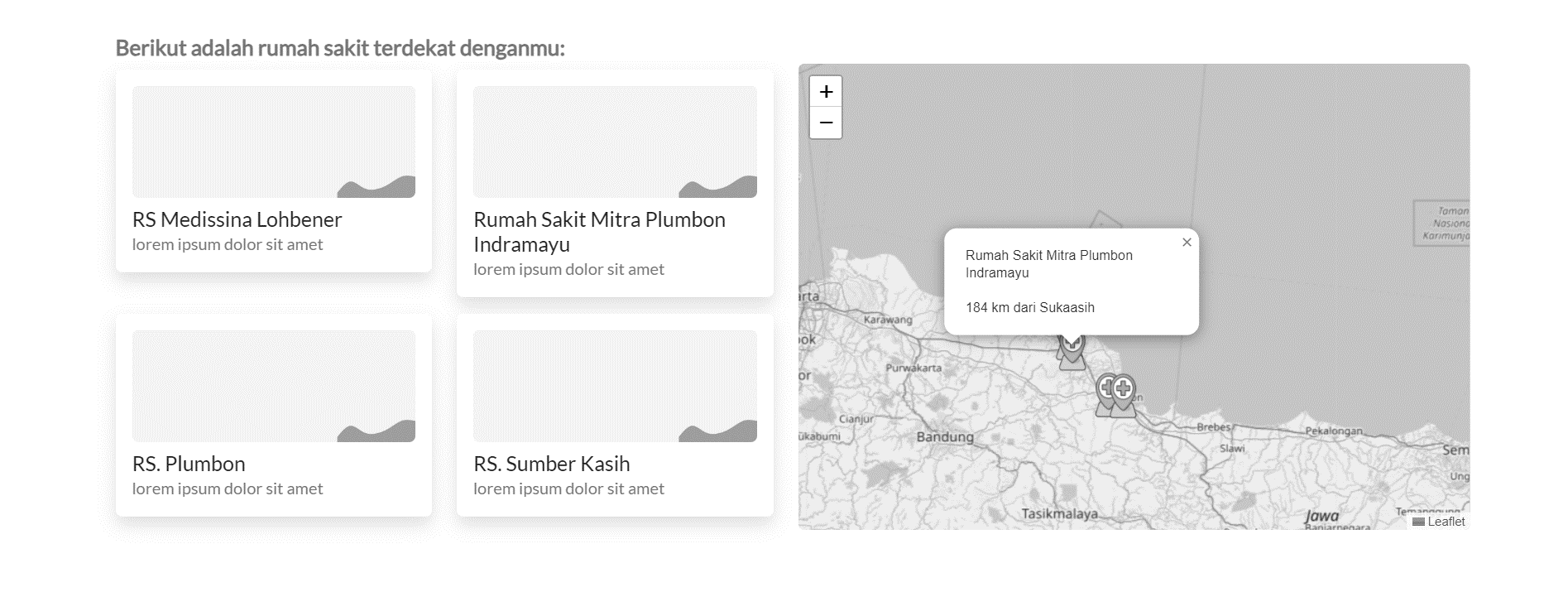


Gambar 3.32 Rancangan Antarmuka Halaman Apotek

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.





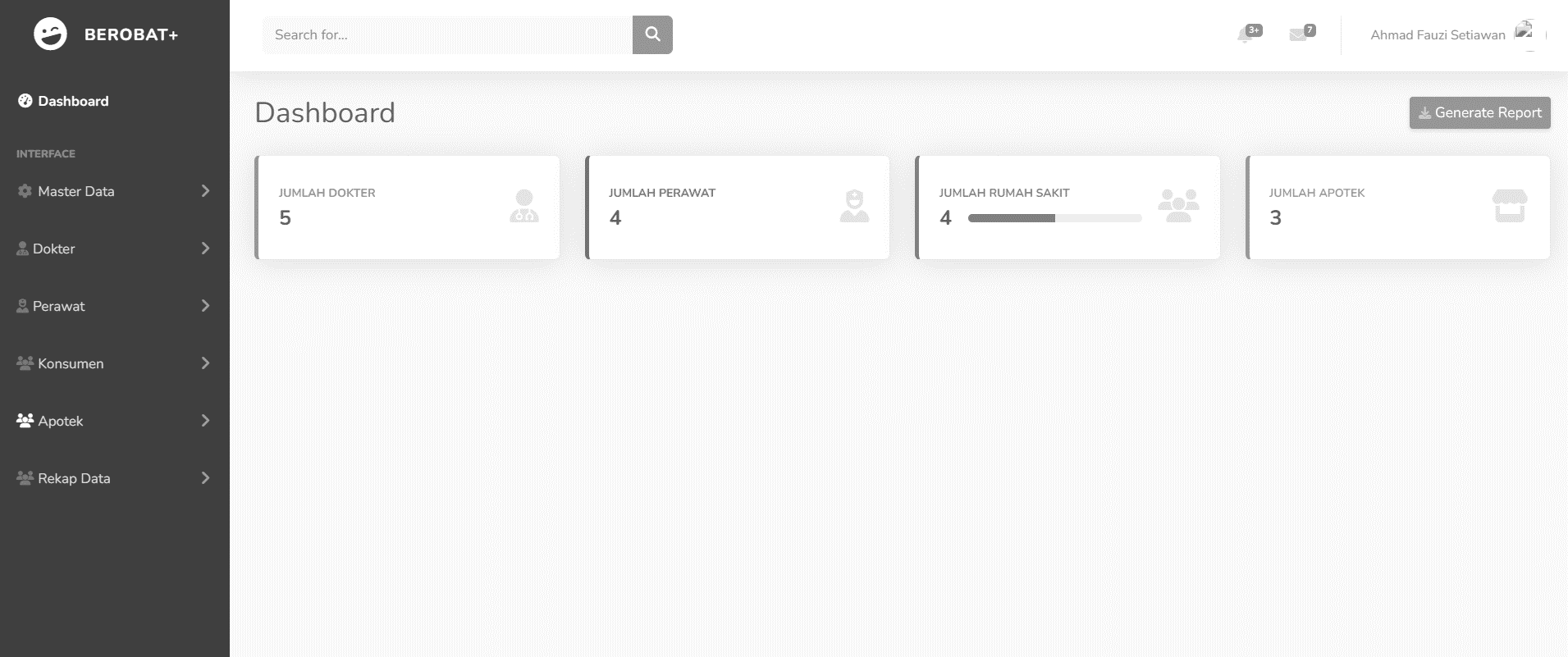
Gambar 3.33 Rancangan Antarmuka Halaman Rumah Sakit Terdekat

### Rancangan Antarmuka Admin

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

1. Rancangan Antarmuka Halaman Dashboard Utama

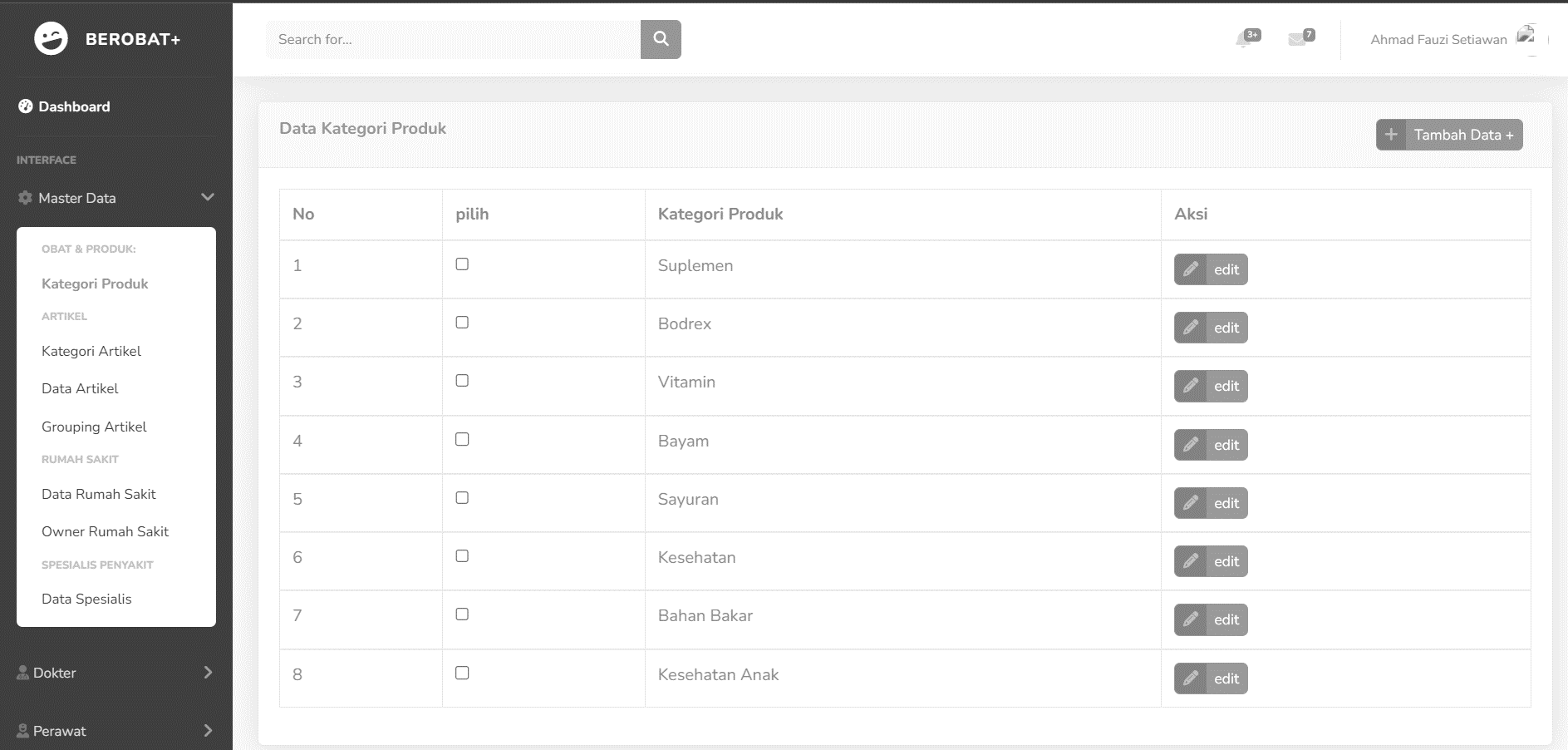
Rancangan antarmuka ini merupakan tampilan halaman dashboard utama. Yang berisi konten jumlah data. Adapun rancangan antarmuka dapat dilihat pada gambar 3.29.



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.



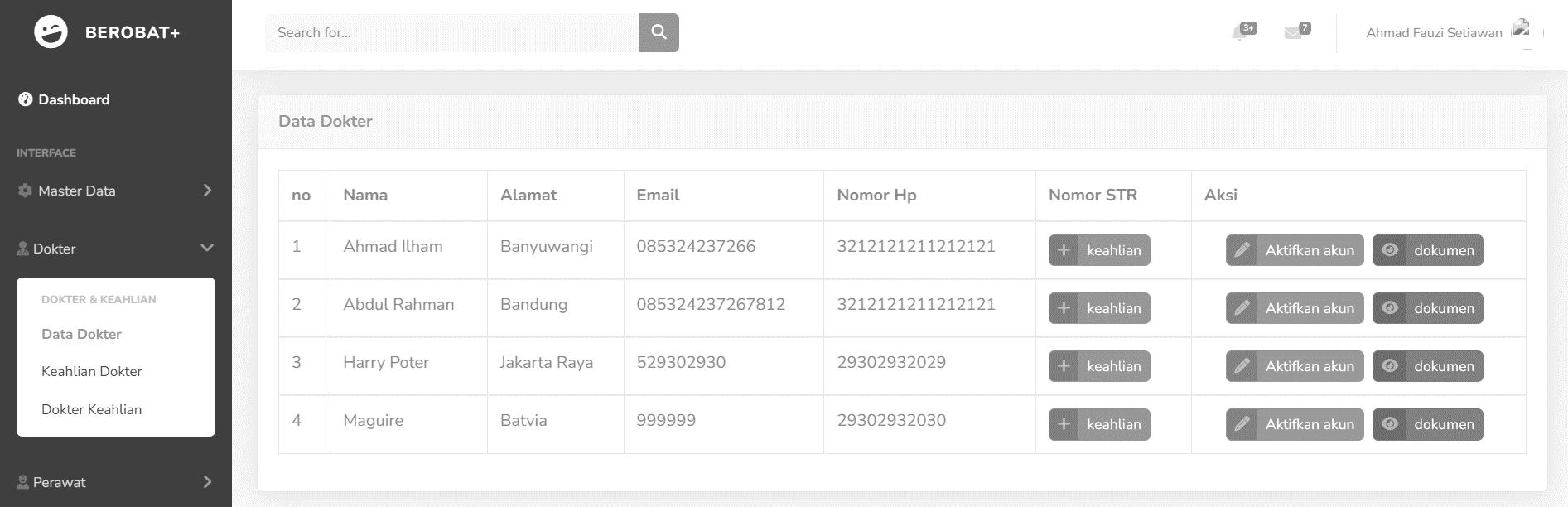
Gambar 3.35 Rancangan Antarmuka Mengelola Kategori Produk

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

pada gambar 3.31.



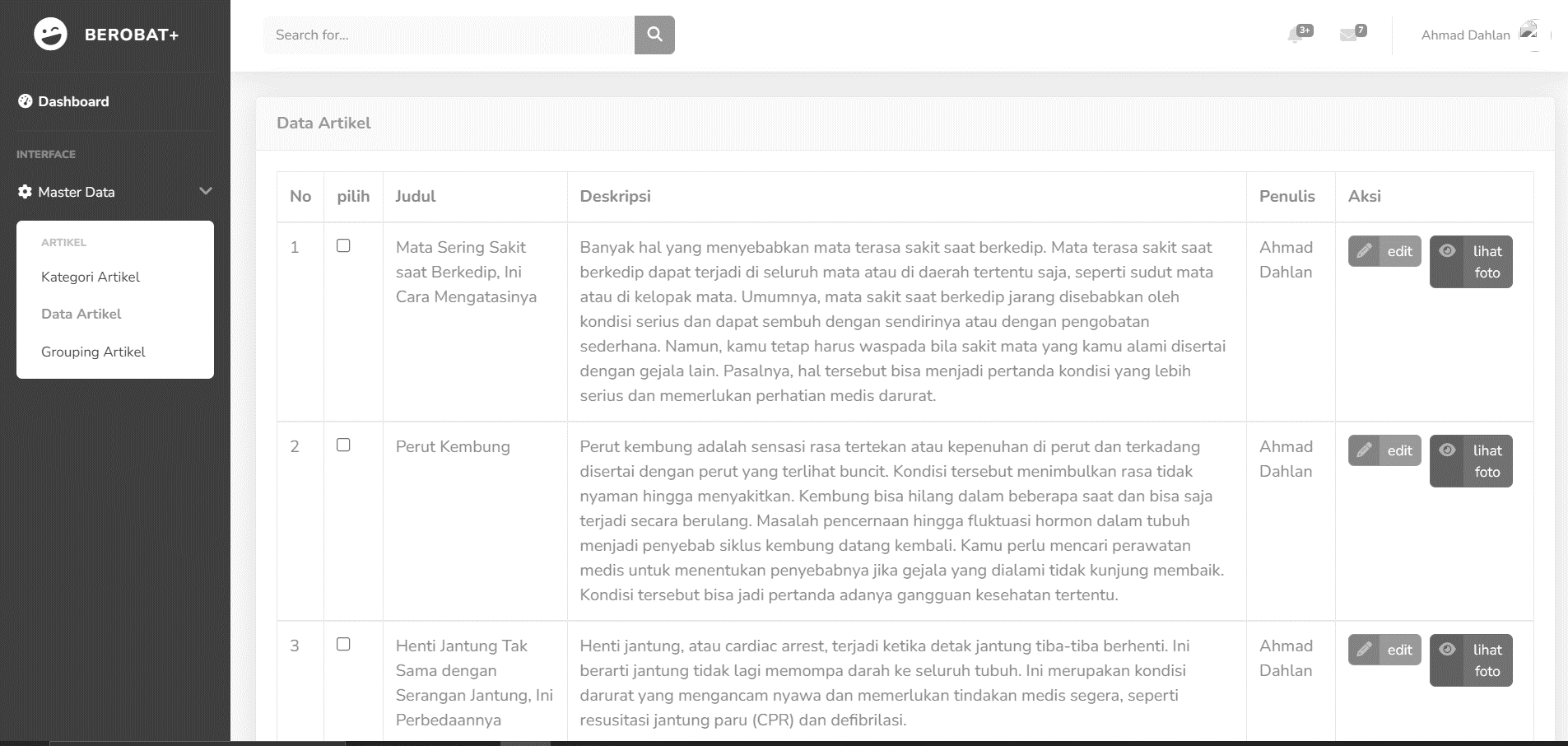
Gambar 3.36 Rancangan Antarmuka Kelola Dokter

### Rancangan Antarmuka Dokter

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

1. Rancangan antarmuka halaman kelola artikel

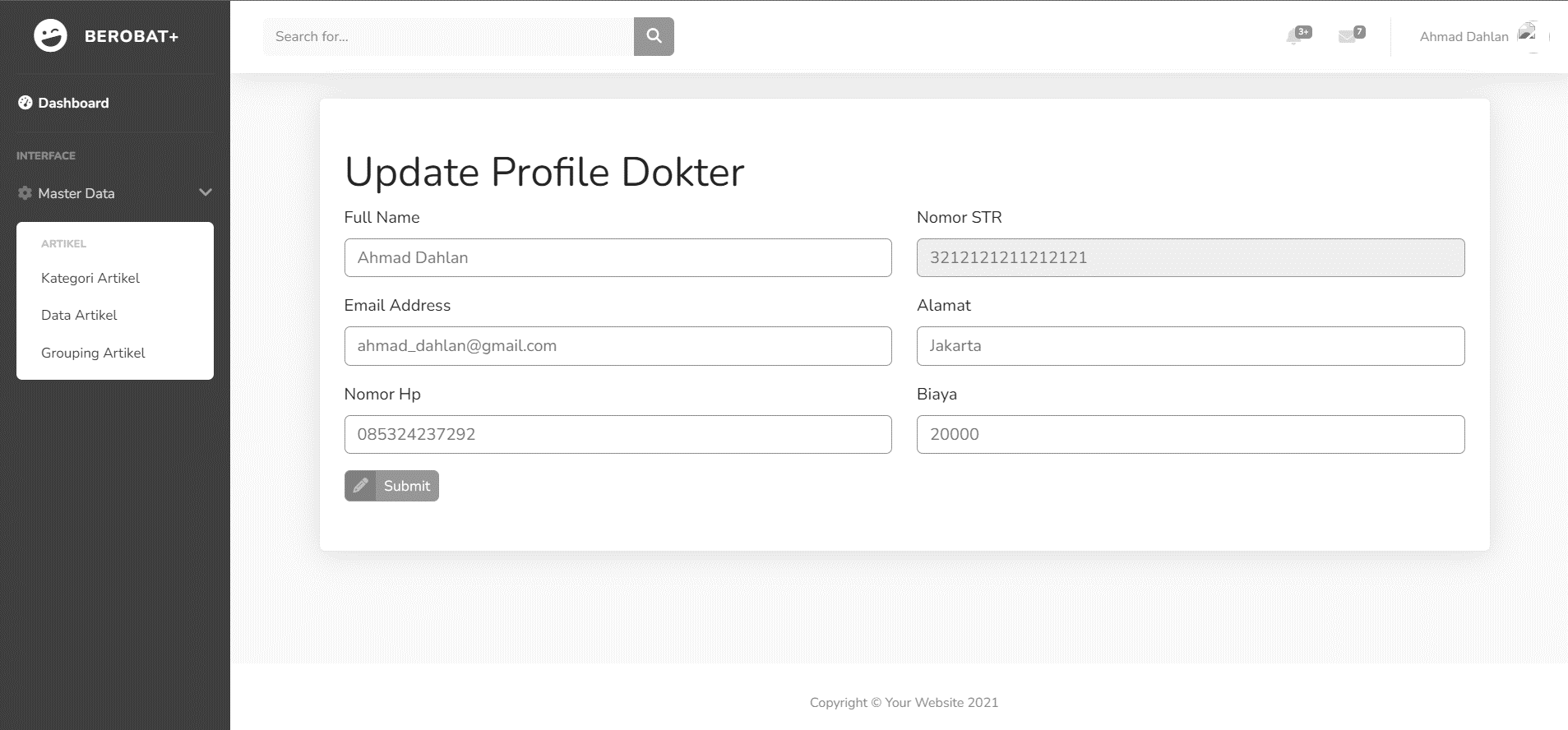
Rancangan antarmuka ini merupakan tampilan halaman untuk mengelola artikel. Adapun rancangan antarmuka dapat dilihat pada gambar 3.32.



Gambar 3.37 Rancangan Antarmuka Kelola Artikel

1. Rancangan antarmuka ubah profil akun

Rancangan antarmuka ini merupakan tampilan halaman untuk mengubah data akun. Terdapat kolom nama, email, alamat, nomor strp, nomor hp, dan biaya. Untuk mengelola biaya konsultasi dapat menggunakan menu profil. Dokter dan perawat dapat memasukkan nominal harga konsultasinya. Adapun rancangan antarmuka dapat dilihat pada gambar 3.33.



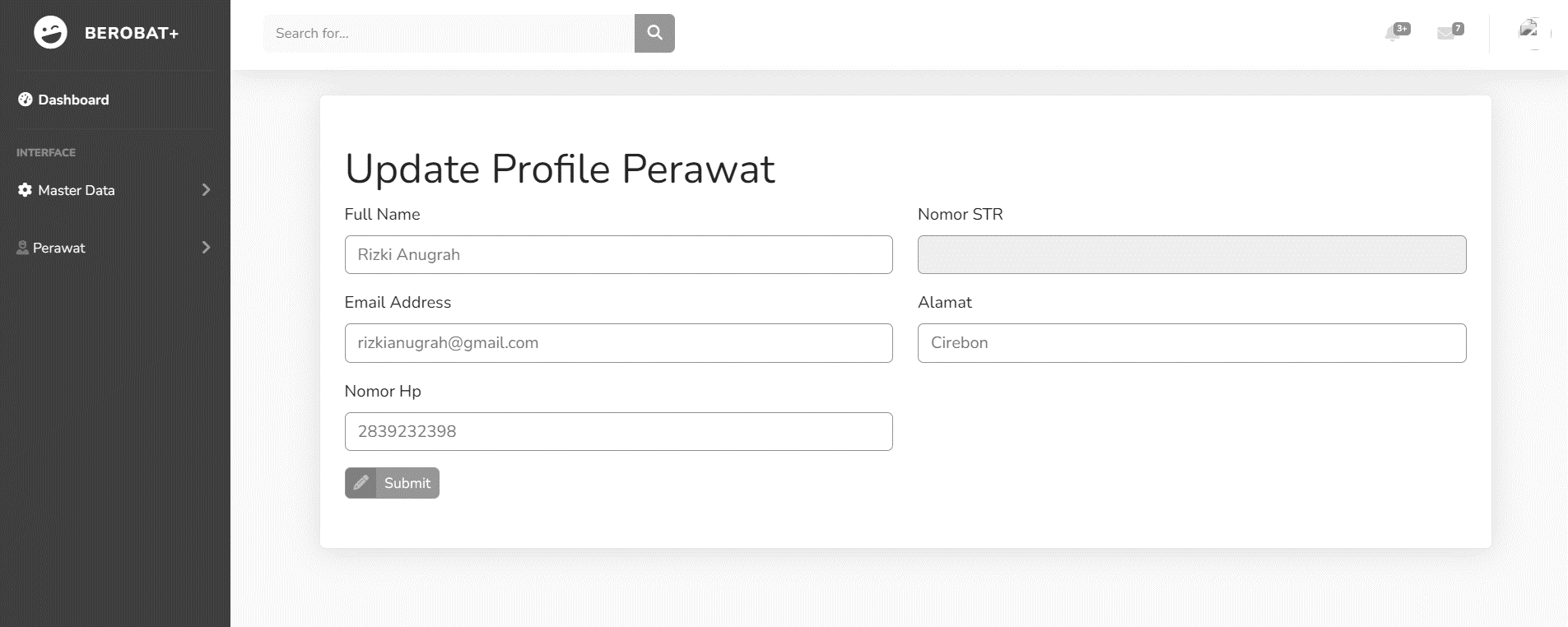
Gambar 3.38 Rancangan Antarmuka Ubah Profil Akun

### 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. Terdapat kolom nama, email, alamat, dan biaya. Untuk mengelola biaya konsultasi dapat menggunakan menu profil. Perawat dapat memasukkan nominal harga konsultasinya. 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

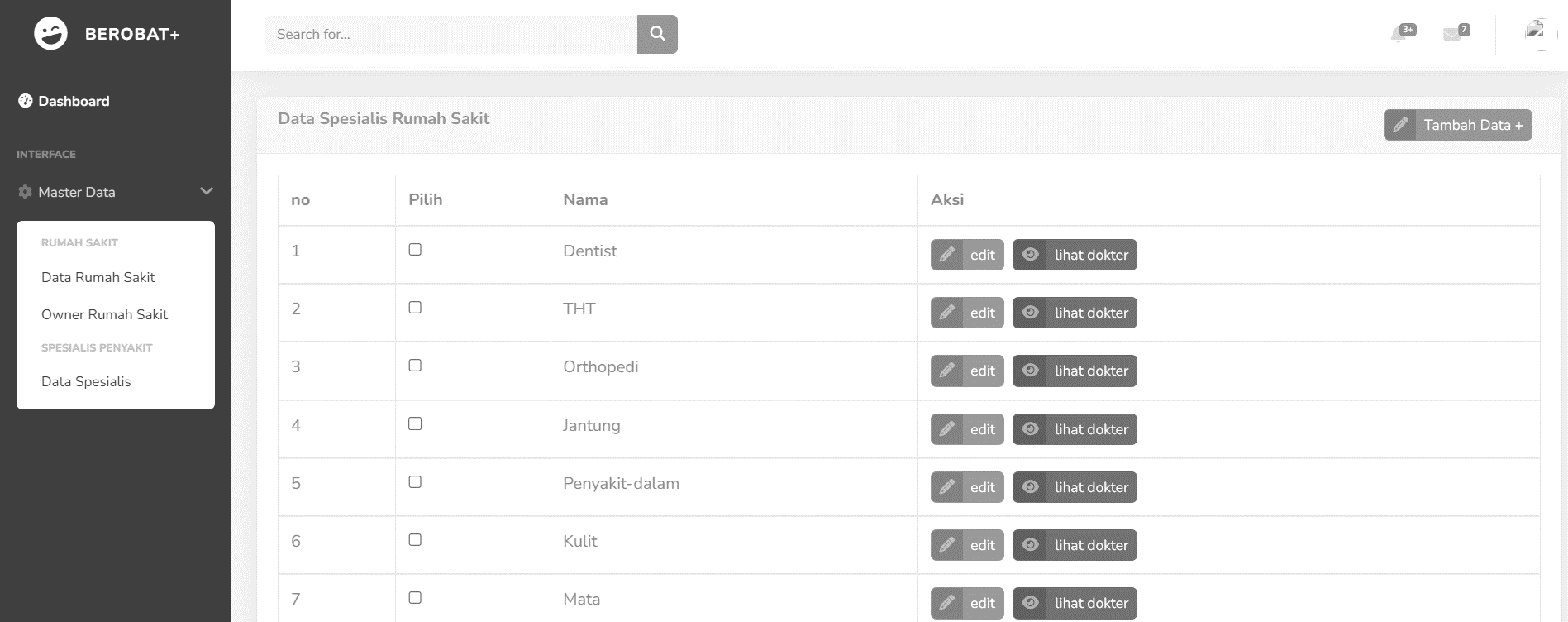
Rancangan antarmuka ini merupakan tampilan halaman untuk mengelola data rumah sakit. Adapun rancangan antarmuka dapat dilihat pada gambar 3.35.



Gambar 3.40 Rancangan Antarmuka Halaman Kelola Rumah Sakit

1. Rancangan antarmuka halaman kelola spesialis rumah sakit

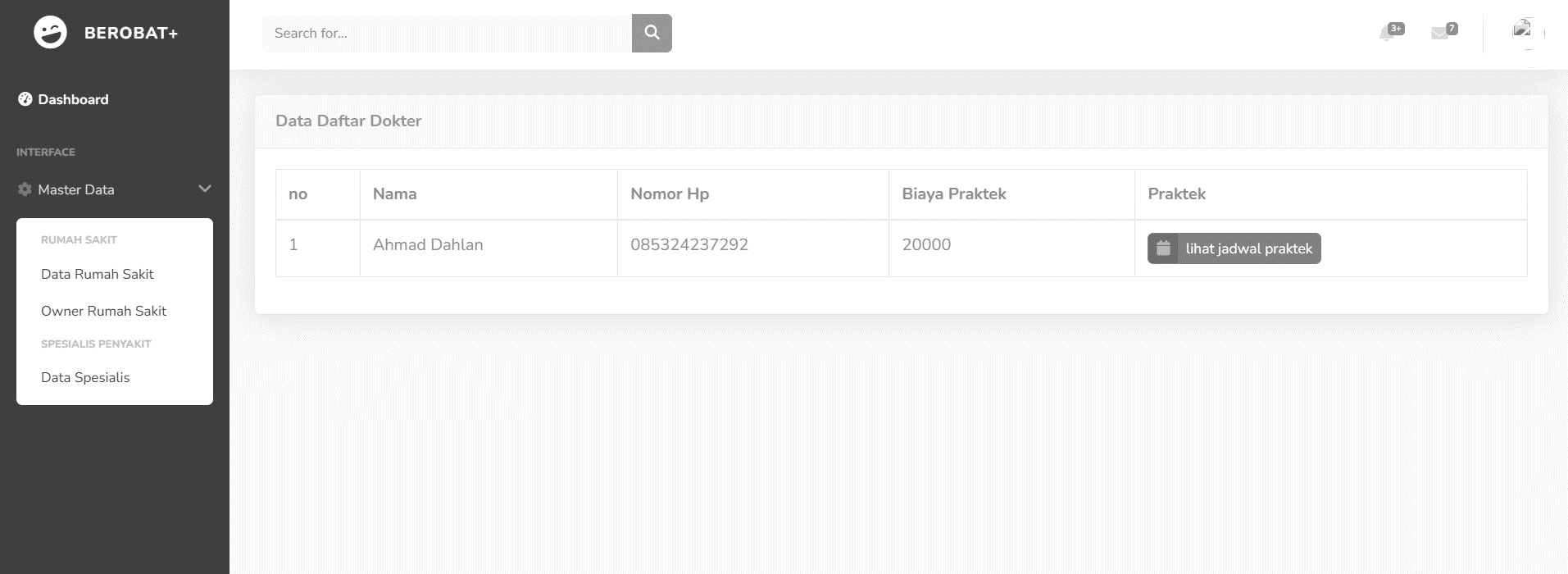
Rancangan antarmuka ini merupakan tampilan halaman untuk mengelola spesialis di rumah sakit. Adapun rancangan antarmuka dapat dilihat pada gambar 3.36.



Gambar 3.41 Rancangan Antarmuka Halaman Kelola Spesialis

1. Rancangan antarmuka kelola jadwal praktek dokter

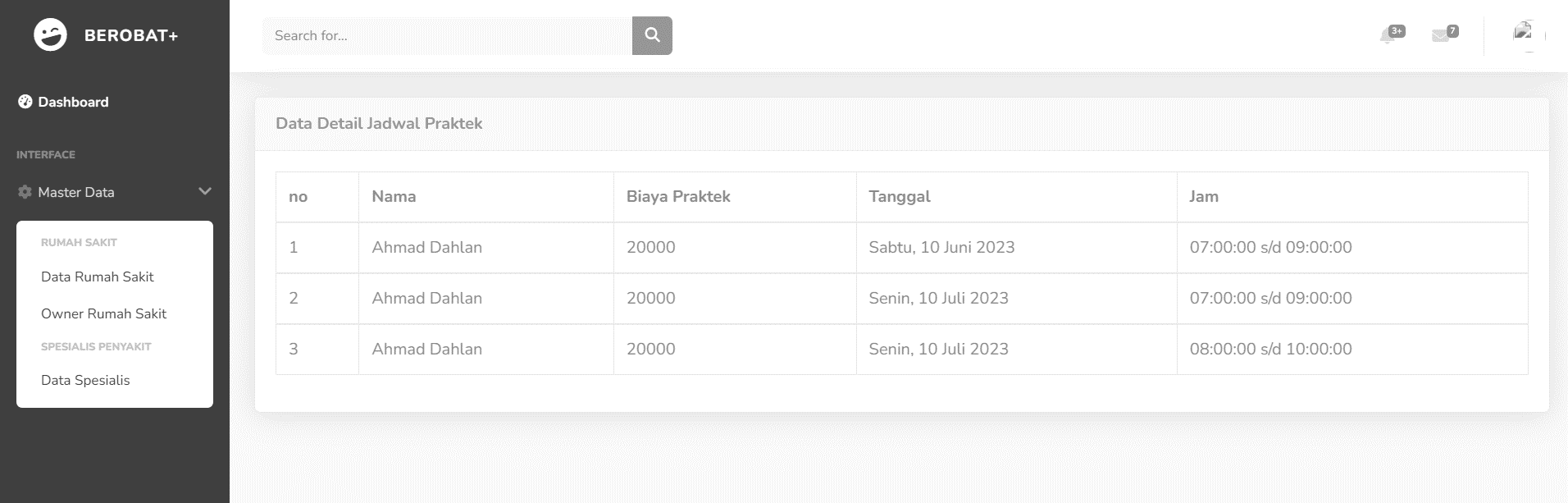
Rancangan antarmuka ini merupakan tampilan halaman untuk mengelola jadwal praktek di rumah sakit. Adapun rancangan antarmuka dapat dilihat pada gambar 3.36.



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. Terdapat tabel yang berisi nama, biaya praktek, tanggal, dan jam praktek. Rancangan antarmuka halaman detail praktek dokter dapat dilihat pada gambar 3.37.



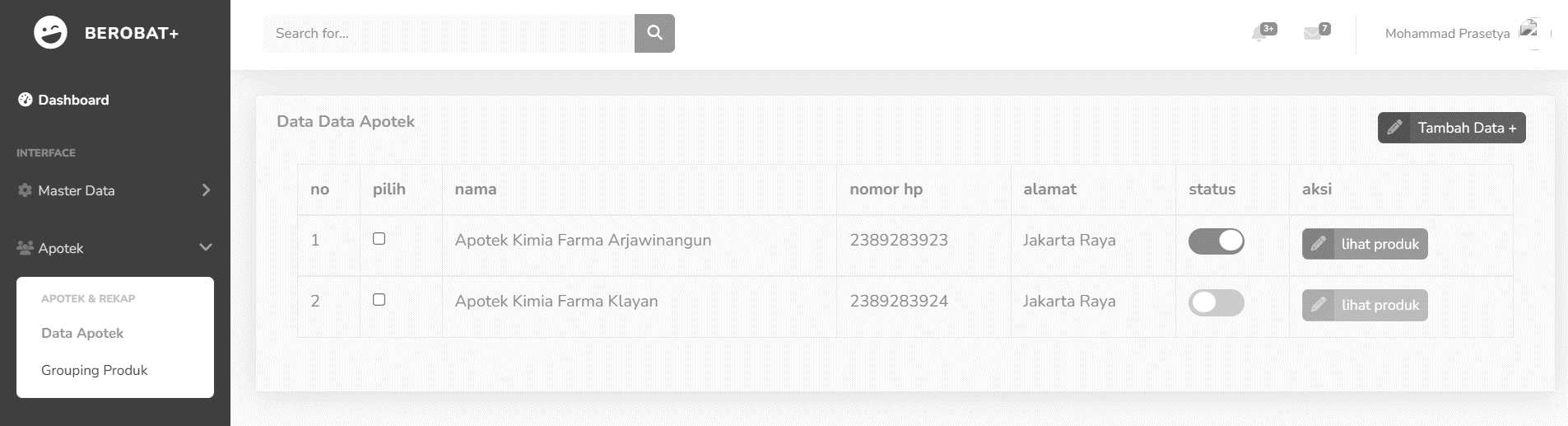
Gambar 3.43 Rancangan Antarmuka Halaman Detail Praktek Dokter

### 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

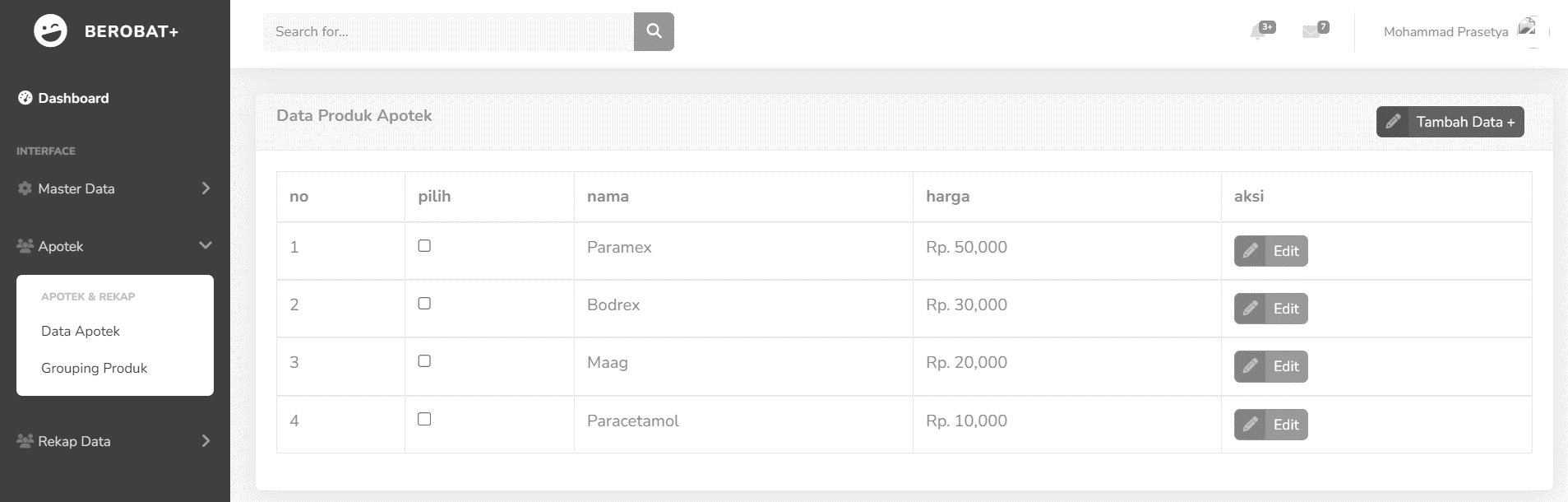
Rancangan antarmuka ini merupakan tampilan untuk mengelola apotek yang dimiliki owner apotek. Owner apotek dapat menambah, melihat, mengubah, dan menghapus apotek. Owner juga dapat mengubah status apotek melalui *slider button.* Adapun rancangan antarmuka halaman kelola apotek dapat dilihat pada gambar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



Gambar 3.44 Rancangan Antarmuka Halaman Kelola Apotek

1. Rancangan antarmuka halaman kelola produk apotek

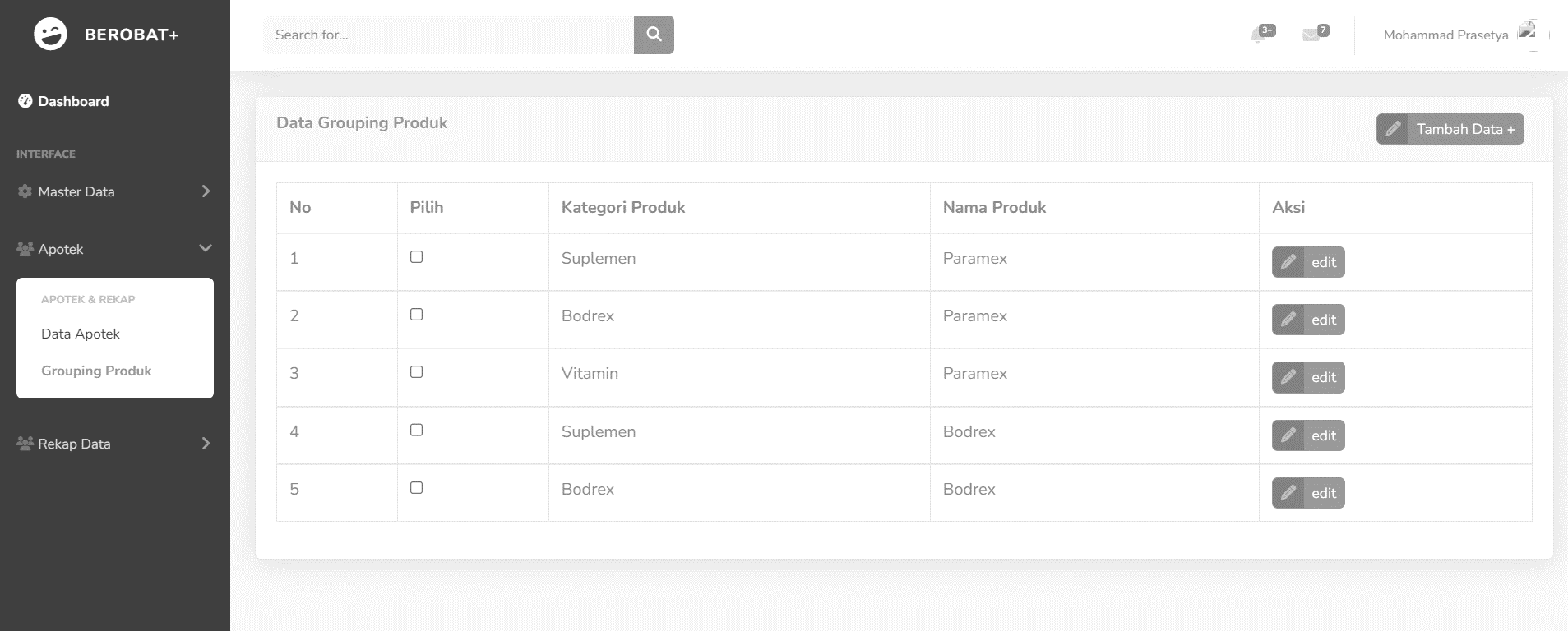
Rancangan antarmuka ini merupakan tampilan halaman untuk mengelola produk apotek. Adapun rancangan antarmuka dapat dilihat pada gambar 3.40.



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.



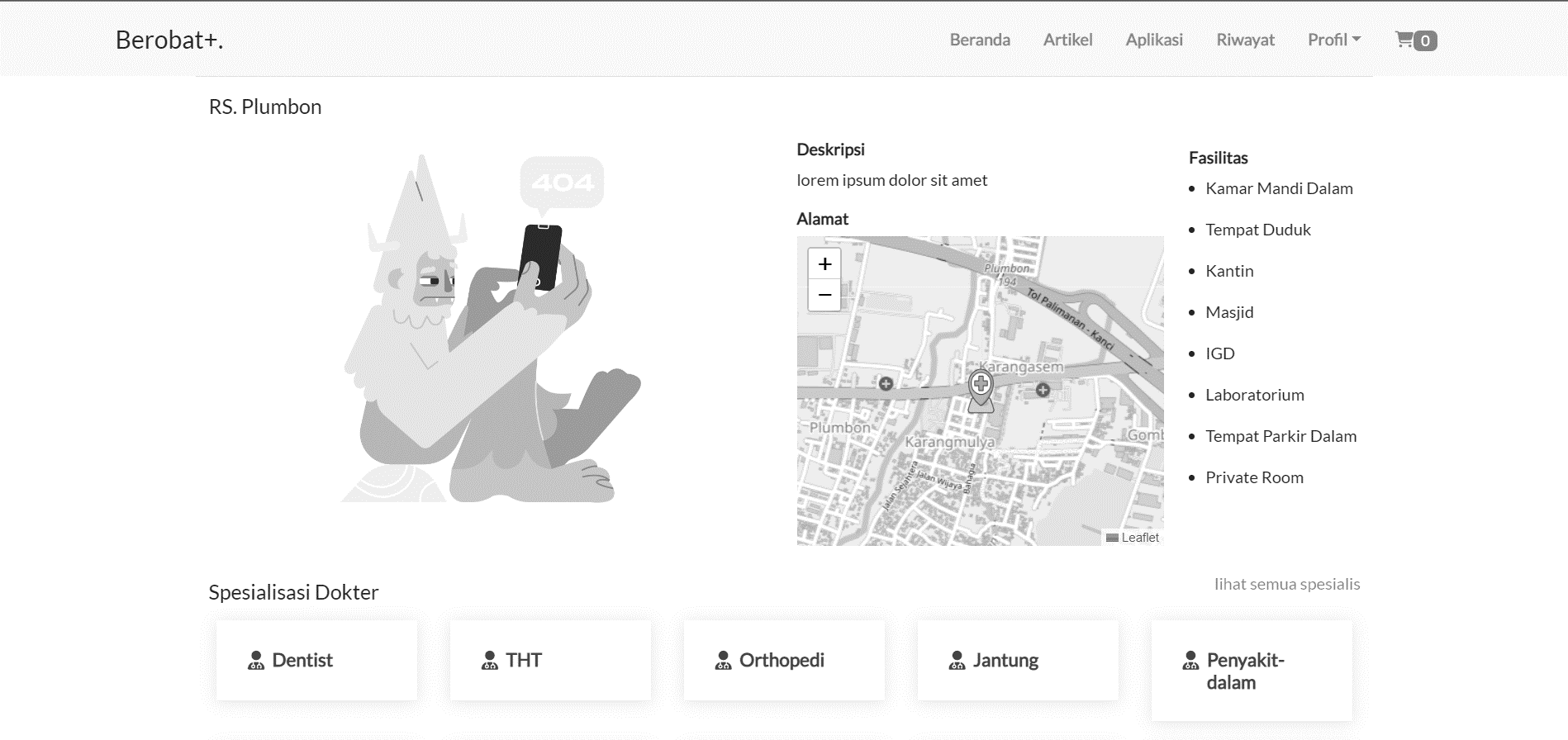
Gambar 3.46 Rancangan Antarmuka Halaman Kelola Produk Kategori

### Rancangan Antarmuka *Member*

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

1. Rancangan antarmuka detail rumah sakit

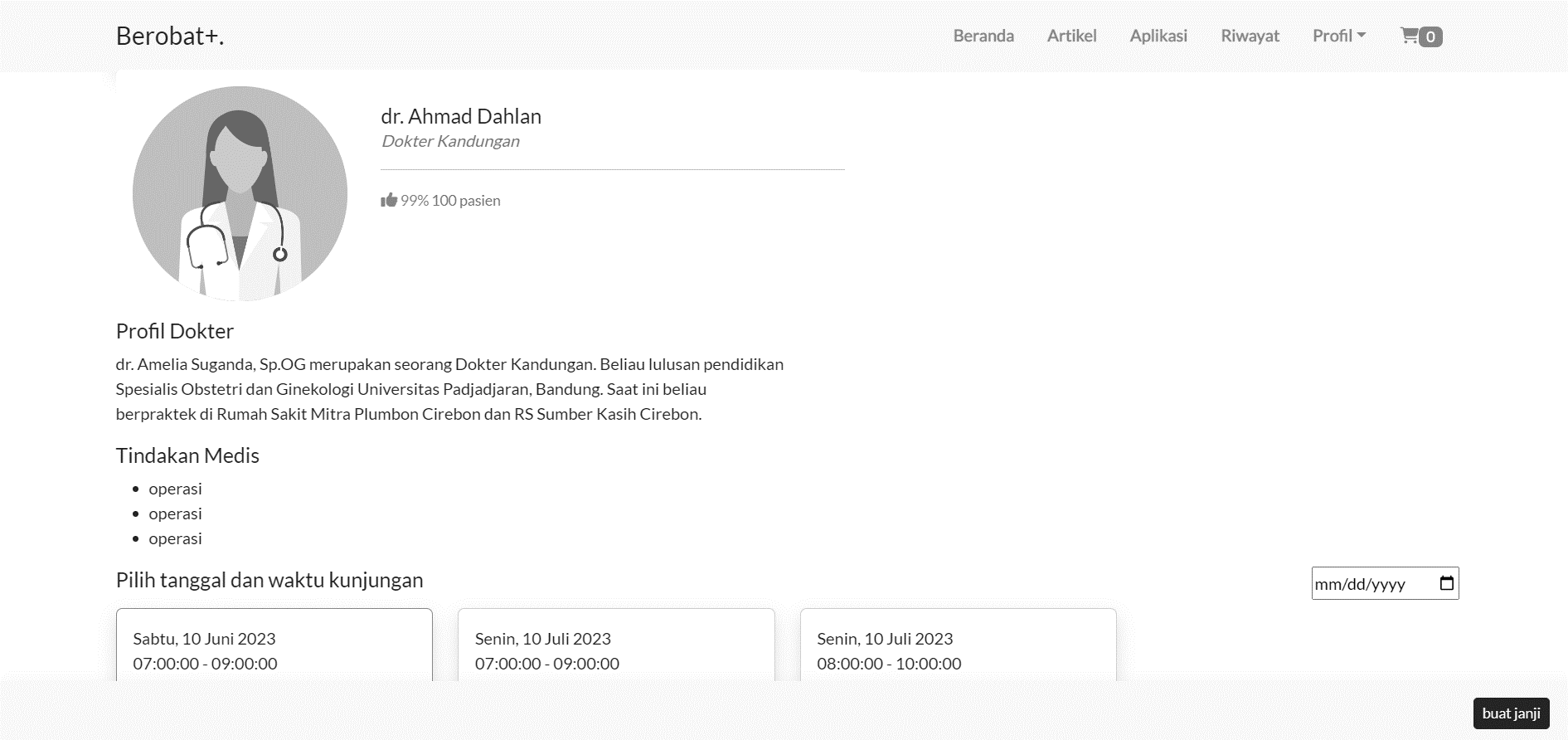
Rancangan antarmuka ini digunakan untuk menampilkan detail rumah sakit. Adapun rancangan antarmuka dapat dilihat pada gambar 3.42.



Gambar 3.47 Rancangan antarmuka halaman detail rumah sakit

1. Rancangan antarmuka buat janji kunjungan rumah sakit

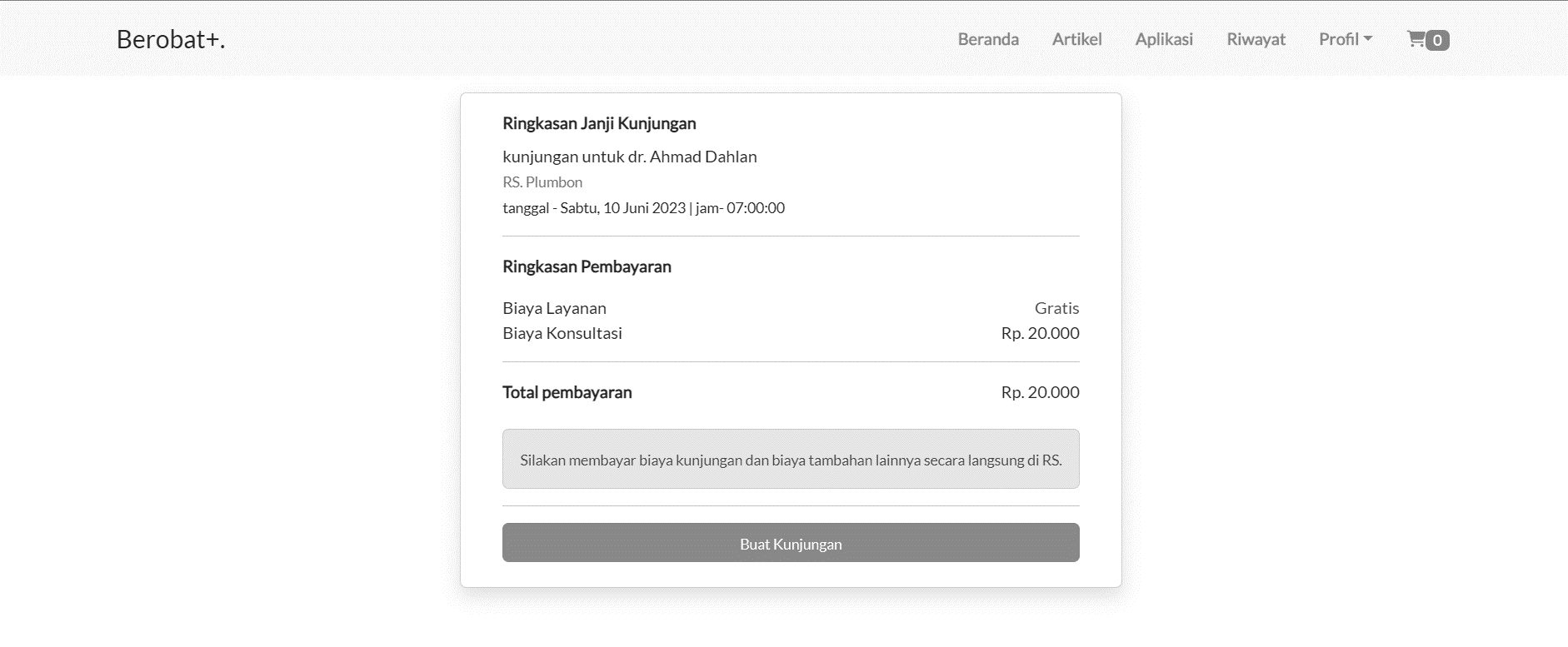
Rancangan antarmuka ini digunakan untuk menampilkan halaman untuk melakukan buat janji. Member perlu memilih jadwal praktek terlebih dahulu. Adapun rancangan antarmuka dapat dilihat pada gambar 3.43.



Gambar 3.48 Rancangan Antarmuka Buat Janji Kunjungan Rumah Sakit

1. Rancangan antarmuka detail kunjungan rumah sakit

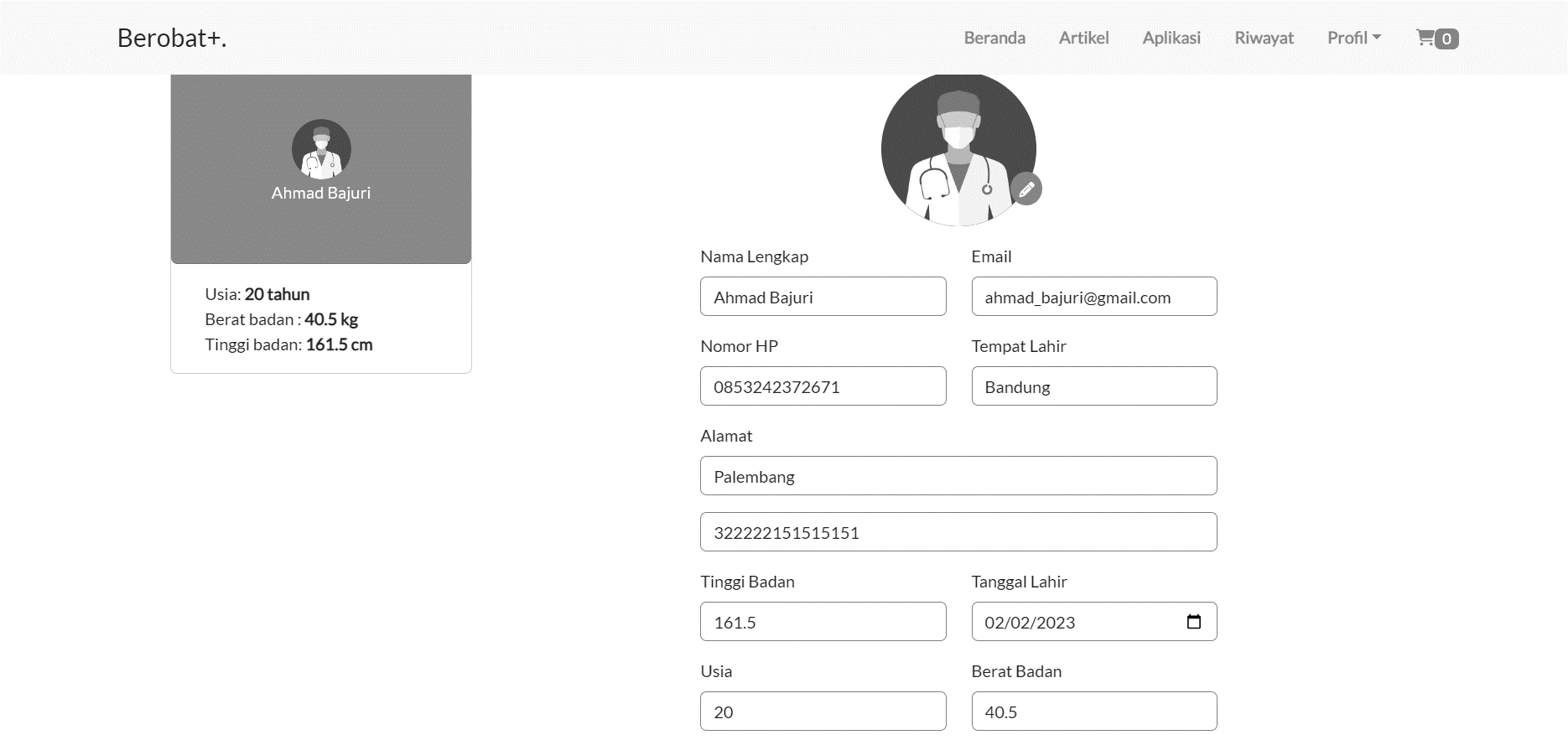
Rancangan antarmuka ini merupakan *user interface* untuk melihat detail kunjungan rumah sakit. Adapun rancangan antarmuka buat janji kunjungan rumah sakit dapat dilihat pada gambar 3.44.



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

Penelitian ini menghasilkan sebuah aplikasi pelayanan kesehatan masyarakat berbasis web. Aplikasi ini digunakan untuk menunjang pelayanan kesehatan bagi pengguna. Terdapat solusi yang dihadirkan dalam mengatasi permasalahan pada sektor kesehatan. Penulis menghadirkan fitur seperti informasi layanan kesehatan berupa rumah sakit dan apotek terdekat dan artikel kesehatan, pemesanan obat, vitamin, dan produk kesehatan lainnya, dan membuat antrian untuk kunjungan temu dokter di rumah sakit. Adapun fitur lainnya penulis bahas dalam bagian pembahasan.

Penulis menggunakan teorema haversine untuk memberikan hasil lokasi terdekat berdasarkan lokasi keberadaan pengguna. Haversine adalah perhitungan jarak antara dua titik berdasarkan koordinat lintang dan bujur pada permukaan bumi. Hasil dari perhitungan teorema tersebut digunakan untuk menentukan rumah sakit dan apotek terdekat,

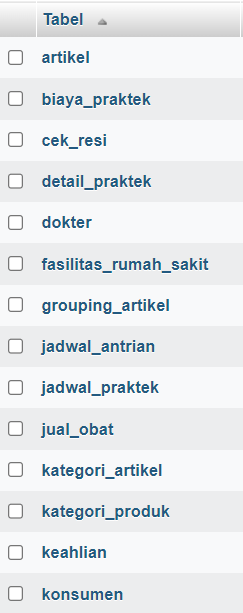
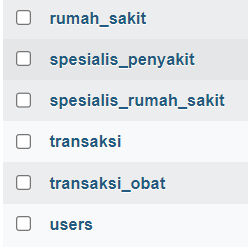
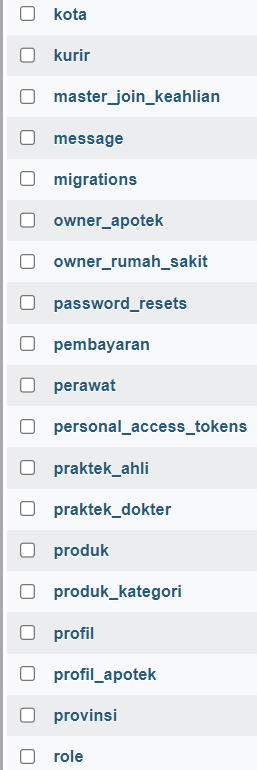
Dapat disimpulkan bahwa aplikasi pelayanan kesehatan masyarakat berbasis webmerupakan aplikasi yang dibangun untuk menunjang aktivitas sektor kesehatan dalam meningkatkan pelayanan kesehatan kepada masyarakat. Salah satu upaya dalam meningkatkan pelayanan kesehatan yaitu informasi layanan kesehatan seperti rumah sakit dan apotek terdekat, pemesanan produk kesehatan, dan buat janji kunjungan temu dokter di rumah sakit. Dengan upaya tersebut diharapkan akan mengatasi sulitnya akses masyarakat dalam mendapatkan layanan kesehatan.

## Pembahasan

Tahapan ini membahas tahapan implementasi analisis kebutuhan sistem, perancangan sistem, perancangan basis data, dan perancangan desain antarmuka. Pada tahapan ini menjelaskan tentang kesesuaian antara rancangan dengan hasil implementasi. Dan, pada bagian ini juga membahas tentang pengujian aplikasi, sehingga terciptanya aplikasi pelayanan kesehatan masyarakat berbasis web.

### Implementasi Basis Data

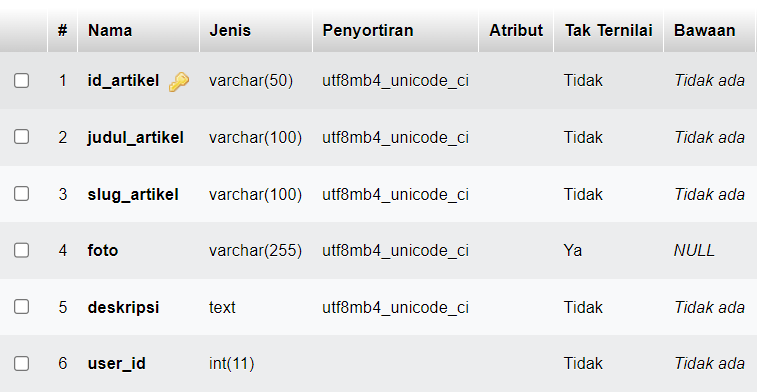
Basis data yang digunakan dalam membangun aplikasi pelayanan kesehatan masyarakat berbasis webadalah berobat\_api. Pada basis data berobat\_api terdapat 39 tabel yang digunakan untuk mengimplementasikan perancangan dan membangun aplikasi ini. Adapun basis data dapat dilihat pada gambar 4.1.



Gambar 4.1 Implementasi Basis Data

1. Implementasi Tabel Artikel

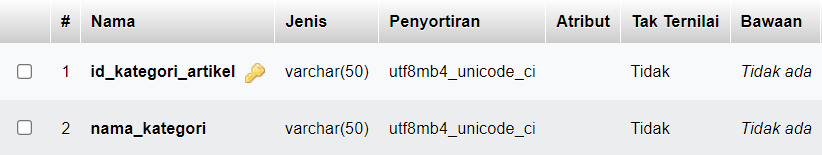
Tabel artikel digunakan untuk menyimpan data artikel yang dikelola oleh dokter dan admin. Adapun struktur tabel dapat dilihat pada gambar 4.2



Gambar 4.2 Implementasi Tabel Artikel

1. Implementasi Tabel Kategori Artikel

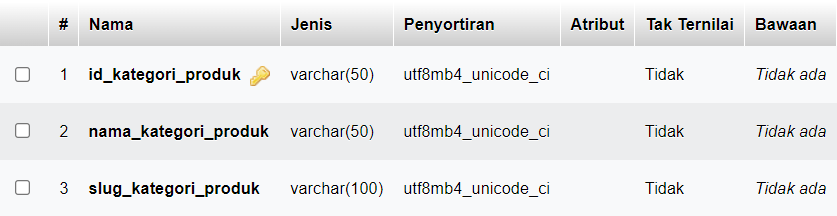
Tabel kategori artikel merupakan tabel untuk mengelola data kategori artikel. Kategori artikel digunakan untuk mengelola artikel. Adapun struktur tabel kategori artikel dapat dilihat pada gambar 4.3



Gambar 4.3 Implementasi Kategori Artikel

1. Implementasi Tabel Kategori Produk

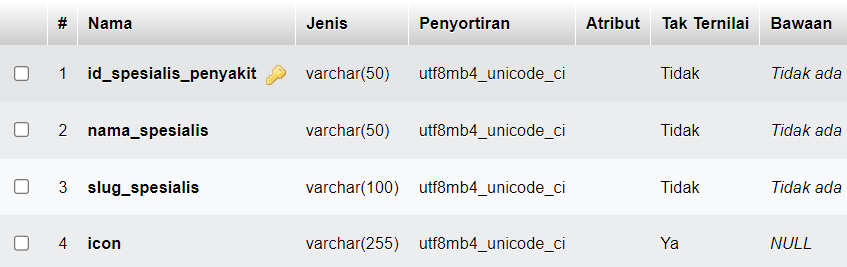
Tabel kategori produk digunakan untuk mengelola data kategori produk. Kategori produk digunakan ketika akan menambahkan sebuah produk, maka perlu menambahkan kategori produk terlebih dahulu.



Gambar 4.4 Implementasi Kategori Produk

1. Implementasi Tabel Spesialis Penyakit

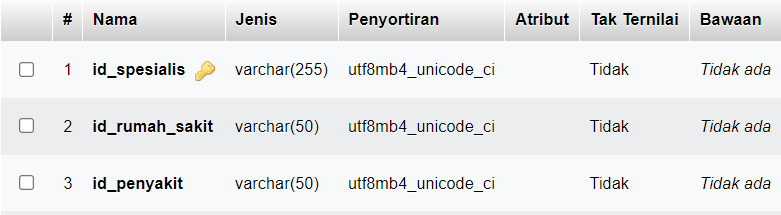
Table spesialis penyakit digunakan untuk menampilkan spesialis yang tersedia dalam aplikasi ini. Adapun struktur tabel spesialis penyakit dapat dilihat pada gambar 4.5.



Gambar 4.5 Implementasi Tabel Spesialis Penyakit

1. Implementasi Tabel Spesialis Rumah Sakit

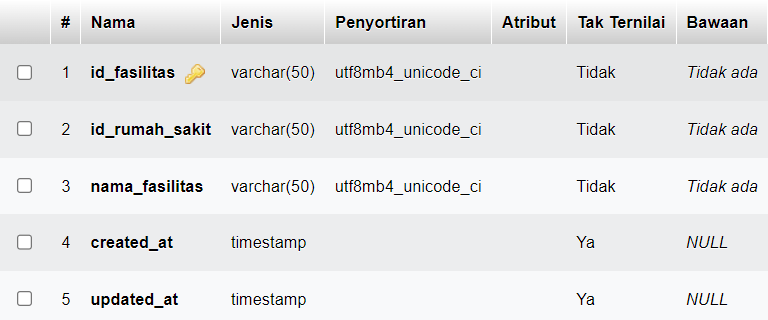
Tabel spesialis rumah sakit digunakan untuk mengelola spesialis penyakit yang tersedia pada rumah sakit. Adapun struktur dapat dilihat pada gambar 4.6.



Gambar 4.6 Implementasi Tabel Spesialis Rumah Sakit

1. Implementasi Tabel Fasilitas Rumah Sakit

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



Gambar 4.7 Implementasi Tabel Fasilitas Rumah Sakit

1. Implementasi Tabel Grouping Artikel

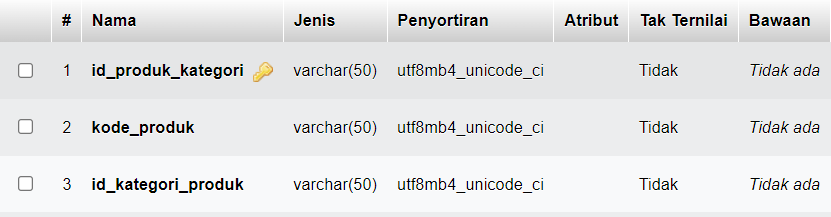
Tabel grouping digunakan untuk mengelompokkan artikel dengan kategori artikel. Adapun struktur tabel grouping artikel dapat dilihat pada gambar 4.8.



Gambar 4.8 Implementasi Tabel Grouping Artikel

1. Implementasi Tabel Produk Kategori

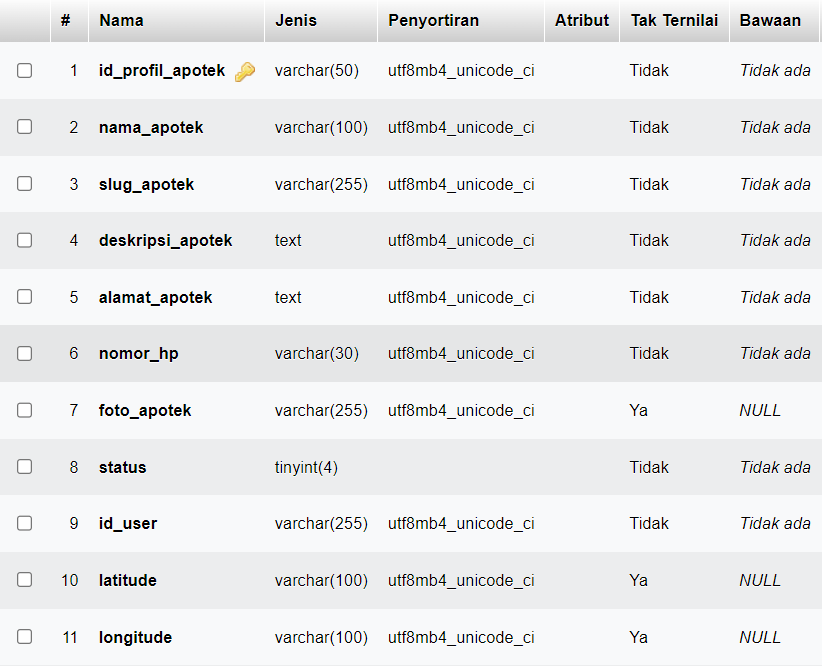
Tabel produk kategori digunakan untuk mengelompokkan produk dengan kategori produk. Adapun tabel produk kategori dapat dilihat pada gambar 4.9



Gambar 4.9 Implementasi Tabel Produk Kategori

1. Implementasi Tabel Profil Apotek

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



Gambar 4.10 Implementasi Tabel Profil Apotek

1. Implementasi Tabel Rumah Sakit

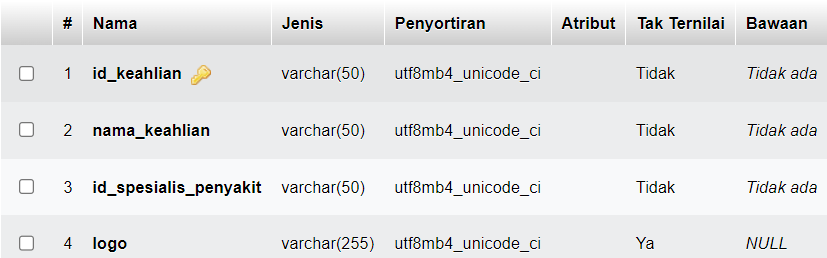
Tabel rumah sakit digunakan untuk mengelola data rumah sakit. Pemilik rumah sakit dapat mengelola rumah sakit. Adapun tabel rumah sakit dapat dilihat pada gambar 4.11.



Gambar 4.11 Implementasi Tabel Rumah Sakit

1. Implementasi Tabel Keahlian

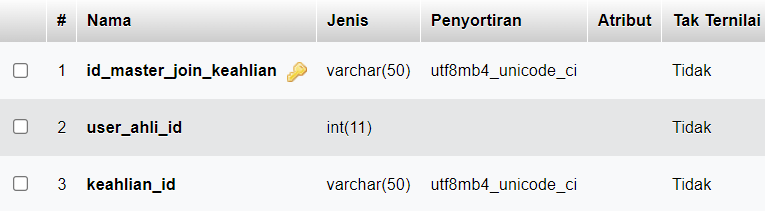
Tabel keahlian digunakan untuk mengelola keahlian yang dimiliki oleh dokter dan perawat. Adapun struktur tabel keahlian dapat dilihat pada gambar 4.12.



Gambar 4.12 Implementasi Tabel Keahlian

1. Implementasi Tabel Join Keahlian

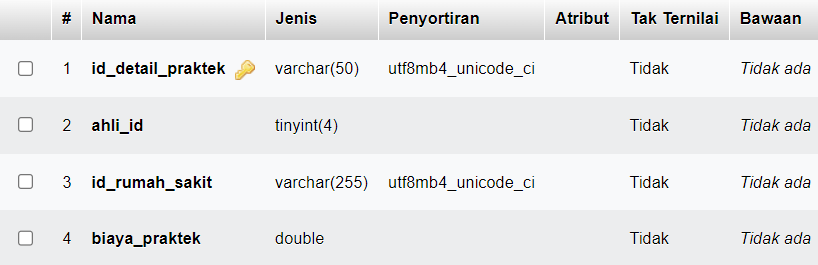
Tabel join keahlian digunakan untuk mengelompokkan antara dokter atau perawat dengan nama keahliannya. Adapun struktur tabel join keahlian dapat dilihat pada gambar 4.13.



Gambar 4.13 Implementasi Tabel Join Keahlian

1. Implementasi Tabel Detail Praktek

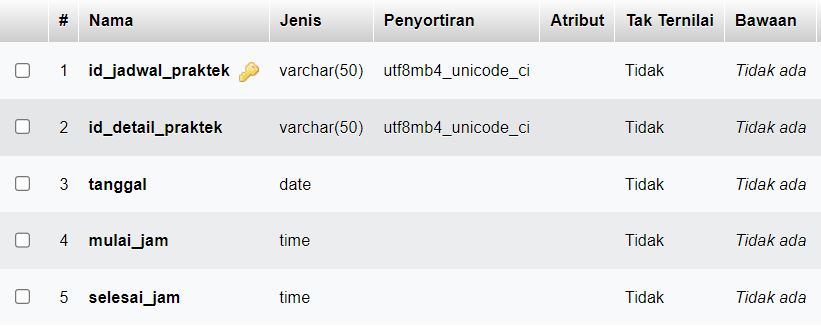
Tabel detail praktek merupakan tabel untuk melihat detail dari dokter yang melakukan praktek di rumah sakit. Adapun struktur tabel detail praktek dapat dilihat pada gambar 3.14.



Gambar 4.14 Implementasi Tabel Detail Praktek

1. Implementasi Tabel Jadwal Praktek

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

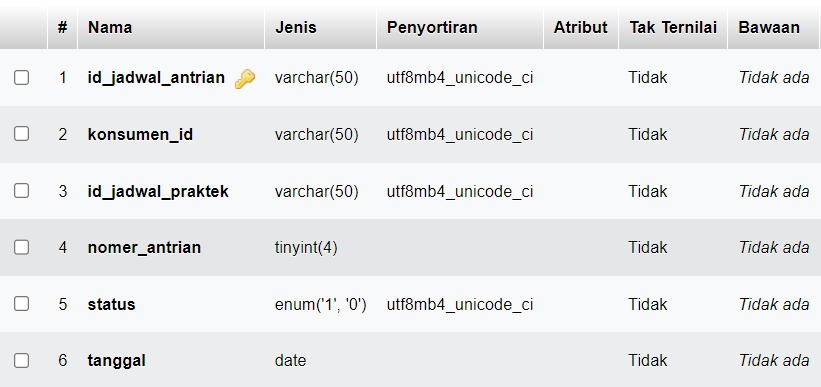


Gambar 4.15 Implementasi Tabel Jadwal Praktek

1. Implementasi Tabel Jadwal Antrian

Tabel jadwal antrian merupakan tabel untuk member dalam membuat antrian

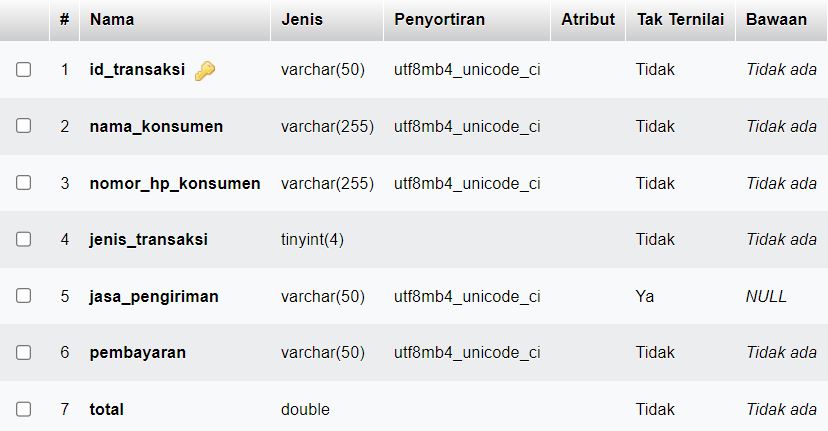
kunjungan temu dokter di rumah sakit. Adapun tabel jadwal antrian dapat dilihat pada gambar 3.16.



Gambar 4.16 Implementasi Tabel Jadwal Antrian

1. Implementasi Tabel Transaksi

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



Gambar 4.17 Implementasi Tabel Transaksi

1. Implementasi Tabel Produk

Tabel produk digunakan untuk mengelola data produk di setiap apotek. Pemilik

apotek dapat mengelola data produk yang tersedia di apotek.



Gambar 4.18 Implementasi Tabel Produk

1. Implementasi Tabel Users

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



Gambar 4.19 Implementasi Tabel Users

### *Sprint Review Product Backlog*

Bagian ini akan memaparkan implementasi metode agile scrum. Sprint review merupakan aktivitas untuk memaparkan pengerjaan product backlog yang telah dibuat sebelumnya. Adapun sprint review dapat dilihat berikut.

1. *Sprint Review* Modul Login dan Register

Modul ini digunakan oleh pemilik akun untuk mengakses aplikasi. Sebelum mengakses aplikasi, diperlukan akun pengguna. *Guest* dapat memilih jenis akun apa yang akan didaftarkan. Berobatplus menyediakan akses akun bagi *member,* dokter, perawat, pemilik apotek, dan pemilik rumah sakit.

Setiap akun terdapat perbedaan hak akses. *Member* dapat melakukan berbagai aktivitas seperti membaca artikel, memesan produk kesehatan, membuat janji temu dokter, dan mengetahui informasi layanan kesehatan berupa rumah sakit dan apotek terdekat. Akun dokter dan perawat dapat membuat artikel untuk dipublikasikan kepada pengguna. Informasi kesehatan yang kredibel dan terpercaya.

Jenis akun selain *member* perlu menunggu proses aktivasi terlebih dahulu. Proses aktivasi akun dilakukan oleh admin aplikasi. Setelah melewati proses aktivasi, akun dapat mengakses aplikasi. Adapun hasil *sprint review* modul login dan register dapat dilihat pada gambar 4.20.

1. *Sprint Review* Modul Aktivasi Akun

Sprint *review* ini menjelaskan bagaimana proses admin melakukan aktivasi akun. Proses aktivasi ini dilakukan oleh admin aplikasi. Selain pendaftaran akun dengan role *member* perlu tahap verifikasi. Tahapan tersebut verifikasi meliputi validasi dokumen yang diunggah. Ketika mendaftar akun sebagai dokter, dokumen yang diunggah adalah Surat Tanda Register (STR). Lalu, ketika mendaftar akun sebagai perawat, dokumen pendukung yang diunggah adalah Surat Tanda Registrasi Perawat.

Pemilik apotek dan rumah sakit juga dapat mengakses layanan aplikasi. Syarat untuk mengaksesnya adalah dengan mengunggah dokumen kepemilikan apotek atau rumah sakit. Tugas seorang administrator pada modul ini adalah melakukan aktivasi akun. Admin perlu melakukan pengecekkan seluruh data pendaftar. Ketika data telah sesuai, maka admin dapat mengaktifkan akun pendaftar. Adapun gambar *sprint review* modul aktivasi akun dapat dilihat pada gambar 4.21. Tombol aktifkan akun digunakan untuk mengatifkan status akun.

1. *Sprint Review* Modul Transaksi

Aplikasi Pelayanan Kesehatan Masyarakat ini memfasilitasi pengguna untuk melakukan pembelian obat, vitamin, dan produk kesehatan lainnya. Untuk menggunakan fitur ini, pengguna atau *member* diwajibkan untuk melakukan login. Hal tersebut diperlukan untuk menyimpan data pembeli. Member dapat memasukkan produk kesehatan ke dalam keranjang sebelum melakukan pembayaran, proses ini dapat dilihat pada gambar 4.23.

Produk kesehatan yang tersedia dalam aplikasi pelayanan kesehatan Masyarakat ini merupakan produk yang berasal dari apotek. Apotek memiliki hak akses untuk mengelola transaksi dan produk. Pembayaran yang disediakan oleh aplikasi ini adalah melalui pihak ketiga, *payment gateway*. Adapun moduk transaksi dapat dilihat pada gambar 4.22.

1. *Sprint Review* Modul Buat Janji Temu Dokter

*Sprint Review* ini merupakan fitur untuk melakukan registrasi kunjungan temu dokter di rumah sakit. Fitur ini digunakan bagi masyarakat yang akan melakukan konsultasi secara langsung di rumah sakit. Member juga diwajibkan untuk melakukan login, agar data kunjungan registrasi dapat disimpan.

Member dapat memilih rumah sakit, serta dokter yang akan dikunjungi. Tahap yang dilakukan untuk menggunakan fitur ini adalah dengan memilih rumah sakit, dokter, dan jadwal prakteknya. Setelahnya dapat menekan tombol buat janji, kemudian member akan mendapatkan qr code. Qr code digunakan sebagai bukti bahwa telah melakukan registrasi kunjungan.

Aplikasi pelayanan kesehatan Masyarakat ini tidak memfasilitasi pembayaran biaya konsultasi yang dilakukan di rumah sakit. Segala jenis pembayaran yang terdapat pada modul ini dipertanggungjawabkan oleh pihak rumah sakit. Aplikasi ini hanya menyediakan fitur registrasi janji kunjungan. Adapun sprint review buat janji temu dokter dapat dilihat pada gambar 4.24.

1. *Sprint Review* Modul Profil Akun

*Sprint Review* ini merupakan fitur untuk profil akun. Seluruh jenis akun dapat melakukan ubah profil, kata sandi, dan alamat. Ubah profil berupa nama, email, dan lain hal. Lalu kata sandi digunakan untuk mengakses aplikasi. Pengguna dapat mengganti kata sandi sewaktu-waktu ketika dibutuhkan. Alamat digunakan untuk melakukan pengiriman produk.

Menambahkan alamat, terdapat 2 cara, yaitu: secara manual dan melalui pin map. Dengan cara manual, pengguna memasukkan label alamat, lokasi, dan detail. Pin map, pengguna tidak perlu menamakan lokasi dengan mengetiknya, cukup dengan melihat map dan diarahkan dengan sesuai, maka lokasi akan terbuat secara otomatis.

1. *Sprint Review* Modul *Search*

*Member* dan *Guest* dapat mencari dokter dan berdasarkan keahlian yang dimilikinya. Hal ini digunakan untuk memudahkan pengguna dalam mengatasi permasalahan kesehatannya. Ketika pengguna merasakan gejala seperti: pusing. Pengguna hanya perlu mengetikkan pusing pada kolom *search* yang disediakan di aplikasi. Member juga dapat mencari rumah sakit melalui kolom input yang tersedia pada halaman rumah sakit. Adapun sprint review modul *search* dapat dilihat pada gambar.

1. *Sprint Review* Modul Informasi Layanan Kesehatan

*Member* dapat mengakses layanan kesehatan berupa informasi tempat pelayanan kesehatan seperti apotek dan rumah sakit terdekat. Fitur ini digunakan oleh masyarakat yang membutuhkan penanganan segera untuk menerima perawatan kesehatan. Dalam penerapannya, teorema haversine dibutuhkan dalam menentukan lokasi layanan kesehatan terdekat dengan pengguna.

*Member* atau *guest* dapat melihat lokasi rumah sakit dan apotek di map yang tersedia di halaman rumah sakit. Ketika titik lokasi rumah sakit atau apotek ditekan, maka akan menampilkan jarak antara titik lokasi tersebut dengan lokasi pengguna. Aplikasi ini tidak menampilkan rute menuju titik lokasi tersebut. Ketika menekan rute, maka akan dialihkan menuju laman google maps. Adapun sprint review modul layanan kesehatan dapat dilihat pada gambar.

### Daily Activity

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

* + - 1. Analisis kebutuhan

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

* + - 1. Penentuan *tools*

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

* + - 1. Penentuan Hak Akses

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

Tabel 4.1 Daily Activity

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

### Testing Endpoint API

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

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

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

### Testing Modul *Product Backlog*

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

Pengujian test case positif meliputi pengujian modul product backlog, terdapat tujuh product backlog dengan masing-masing *test case.* Pengujian aplikasi ini dilakukan oleh ….. Test case positif dan negatif memperoleh hasil sesuai dengan yang diharapkan, yaitu product backlog yang diuji dapat digunakan oleh pengguna. Hasil dari pengujian ini dapat disimpulkan bahwa aplikasi pelayanan kesehatan dan konsultasi masyarakat siap digunakan oleh Masyarakat dalam mengakses pelayanan kesehatan. Adapun hasil dari pengujian dapat dilihat pada gambar.

# **BAB V PENUTUP**

## Kesimpulan

Berdasarkan hasil dari pengerjaan *product backlog*, *sprint planning*, *sprint review*, dan pengujian aplikasi pelayanan kesehatan masyarakat berbasis web menggunakan *framework* vuejs maka dapat ditarik kesimpulan sebagai berikut.

1. Masyarakat dapat mengakses dan menerima berbagai pelayanan kesehatan melalui aplikasi ini.
2. Aplikasi pelayanan kesehatan masyarakat ini menghasilkan informasi kesehatan untuk menunjang pengetahuan pengguna. Informasi kesehatan berupa artikel yang dibuat oleh perawat dan dokter.
3. Aplikasi pelayanan kesehatan masyarakat ini menghasilkan informasi klinis berupa rumah sakit dan apotek terdekat. *Member* dan *Guest* dapat melihat lokasi pelayanan klinis di map yang tersedia dalam aplikasi.

## Penutup

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

1. Dapat dibuat konsultasi kesehatan melalui *chat* berbasis web
2. Memperbaiki *user interface* dan *user experience* aplikasi.

# **DAFTAR PUSTAKA**

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Javina, Yana Krisna. 2020. *RESTful API Untuk Menampilkan List Berita Menggunakan Arsitektur Microservices untuk ditampilkan di front-end.* Skripsi S-1. Daerah Istimewa Yogyakarta: Universitas Teknologi Digital Indonesia.

You, Evan. (2023). Progressive Javascript Framework. <https://vuejs.org/>. Diakses pada tanggal 12 Juli 2023.

# **LAMPIRAN**

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

**Menampilkan Informasi Klinis Terdekat**

|  |
| --- |
| getNearest() {  let type = 'postData';  let url = ['master/rumah\_sakit/data/find\_nearest', {  latitude: this.latitude,  longitude: this.longitude,  }];  this.isLoading = true;  this.$store  .dispatch(type, url)  .then(result => {  this.isLoading = false;  this.nearestResults = result.data;  this.calculateBounds();  })  .catch(err => {  console.log(err);  });  },  fetchLocationDetails() {  axios  .get('https://nominatim.openstreetmap.org/reverse', {  params: {  lat: this.latitude,  lon: this.longitude,  format: 'jsonv2',  },  })  .then(response => {  // Extract the desired location details from the response  this.locationName = response.data;  const { address } = response.data;  })  .catch(error => {  console.error('Error occurred while fetching location details:', error);  });  },  calculateBounds() {  if (this.nearestResults.length === 0) {  this.bounds = null;  return;  }  const minLat = Math.min(...this.nearestResults.map(result => result.latitude));  const maxLat = Math.max(...this.nearestResults.map(result => result.latitude));  const minLng = Math.min(...this.nearestResults.map(result => result.longitude));  const maxLng = Math.max(...this.nearestResults.map(result => result.longitude));  this.bounds = [[minLat, minLng], [maxLat, maxLng]];  }, |

**Melakukan Transaksi Produk**

|  |
| --- |
| buyProduct() {  let type = "postData";  let idKeranjangDetail = this.detail.map(detailItem => detailItem.idKeranjangDetail);  let url = [  "master/pembelian/transaksi", {  id\_keranjang: this.items.idKeranjang,  id\_keranjang\_detail: idKeranjangDetail,  }  ];  this.$store.dispatch(type, url)  .then((result) => {  this.$swal({  icon: 'success',  title: 'berhasil melakukan pembayaran'  }).then(() => {  this.$router.push({ name: 'Riwayat Pembelian' })  })  })  .catch((err) => {  console.log(err);  });  }, |

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



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



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





|  |
| --- |
| **LAMPIRAN 5. *TESTING PRODUCT BACKLOG* (Case Positif)** |





|  |
| --- |
| **LAMPIRAN 6. *TESTING PRODUCT BACKLOG* (Case Negatif)** |



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



|  |  |  |  |
| --- | --- | --- | --- |
| Testing Endpoint Api | | | |
| Modul Backlog | Fitur | Endpoint | Status Code Api |
| Register dan Login | Register Akun | https://berobatplus.shop/api/autentikasi/register | Success: 200 |
| Login Admin, Dokter, dan Perawat | https://berobatplus.shop/api/autentikasi/login | Success: 200 |
| Register Member | https://berobatplus.shop/api/akun/konsumen | Success: 200 |
| Login Owner Rumah Sakit dan Apotek | https://berobatplus.shop/api/autentikasi/login | Success: 200 |
| Aktivasi Akun | Aktivasi Akun Dokter | https://berobatplus.shop/api/akun/active\_account/{id\_user}/account | Success: 200 |
| Aktivasi Akun Perawat | [https://berobatplus.shop/api/akun/active\_account/{id\_user}/account](https://berobatplus.shop/api/akun/active_account/) | Success: 200 |
| Aktivasi Akun Owner Rumah Sakit | [https://berobatplus.shop/api/akun/active\_account/{id\_user}/account](https://berobatplus.shop/api/akun/active_account/) | Success: 200 |
| Aktivasi Akun Owner Apotek | https://berobatplus.shop/api/akun/active\_account/{id\_user}/account | Success: 200 |
| Transaksi | Memasukkan Produk ke Keranjang | https://berobatplus.shop/api/keranjang | Success: 200 |
| Melihat Produk | https://berobatplus.shop/api/apotek/produk/data\_produk | Success: 200 |
| Melakukan Pembayaran Produk | https://berobatplus.shop/api/pembelian/transaksi | Success: 200 |
| Melihat Riwayat Transaksi | https://berobatplus.shop/api/pembelian/transaksi | Success: 200 |
| Melihat Detail Transaksi Pembelian | https://berobatplus.shop/api/pembelian/transaksi | Success: 200 |
| Buat Janji Temu Dokter | Melihat Jadwal Praktek Dokter | [https://berobatplus.shop/api/master/ahli/jadwal\_praktek/{id\_detail\_praktek}](https://berobatplus.shop/api/konsumen/checkout) | Success: 200 |
| Melihat Rumah Sakit dan Apotek Terdekat | <https://berobatplus.shop/api/master/rumah_sakit/data/find_nearest> | Success: 200 |
| Melihat Dokter berdasarkan Spesialisnya di RS | https://berobatplus.shop/api/master/ahli/praktek/praktek\_rs | Success: 200 |
| Melihat Detail Rumah Sakit dan Apotek | https://berobatplus.shop/api/master/rumah\_sakit/data | Success: 200 |
| Membuat Janji Temu Dokter | https://berobatplus.shop/api/master/ahli/jadwal\_antrian | Success: 200 |
| Melihat Riwayat Kunjungan | https://berobatplus.shop/api/master/ahli/jadwal\_antrian | Success: 200 |
| Profil | Mengubah Profil Akun | <https://berobatplus.shop/api/akun/profil/nama_role/profil> | Success: 200 |
| Mengubah Kata Sandi Akun | https://berobatplus.shop/api/akun/change\_password | Success: 200 |
| Menambahkan Alamat Pengiriman | https://berobatplus.shop/api/master/alamat\_user | Success: 200 |
| Modul Search | Mencari Dokter dan Perawat terdekat | https://berobatplus.shop/api/akun/perawat | Success: 200 |
| Mencari Dokter dan Perawat berdasarkan Keahlian | https://berobatplus.shop/api/master/cari/keahlian | Success: 200 |
| Mencari Dokter berdasarkan Spesialisnya | [https://berobatplus.shop/api/master/spesialis/{id\_spesialis}/get\_dokter](https://berobatplus.shop/api/master/ahli/keahlian/master/) | Success: 200 |
| Mencari Rumah Sakit | <https://berobatplus.shop/api/master/cari/rumah_sakit> | Success: 200 |

|  |  |  |  |
| --- | --- | --- | --- |
| Testing Endpoint Api | | | |
| Modul Backlog | Fitur | Endpoint | Status Code Api |
| Register dan Login | Register Akun | https://berobatplus.shop/api/autentikasi/register | Success: 200 |
| Login Admin, Dokter, dan Perawat | https://berobatplus.shop/api/autentikasi/login | Success: 200 |
| Register Member | https://berobatplus.shop/api/akun/konsumen | Success: 200 |
| Login Owner Rumah Sakit dan Apotek | https://berobatplus.shop/api/autentikasi/login | Success: 200 |
| Aktivasi Akun | Aktivasi Akun Dokter | https://berobatplus.shop/api/akun/active\_account/{id\_user}/account | Success: 200 |
| Aktivasi Akun Perawat | [https://berobatplus.shop/api/akun/active\_account/{id\_user}/account](https://berobatplus.shop/api/akun/active_account/) | Success: 200 |
| Aktivasi Akun Owner Rumah Sakit | [https://berobatplus.shop/api/akun/active\_account/{id\_user}/account](https://berobatplus.shop/api/akun/active_account/) | Success: 200 |
| Aktivasi Akun Owner Apotek | https://berobatplus.shop/api/akun/active\_account/{id\_user}/account | Success: 200 |
| Transaksi | Memasukkan Produk ke Keranjang | https://berobatplus.shop/api/keranjang | Success: 200 |
| Melihat Produk | https://berobatplus.shop/api/apotek/produk/data\_produk | Success: 200 |
| Melakukan Pembayaran Produk | https://berobatplus.shop/api/pembelian/transaksi | Success: 200 |
| Melihat Riwayat Transaksi | https://berobatplus.shop/api/pembelian/transaksi | Success: 200 |
| Melihat Detail Transaksi Pembelian | https://berobatplus.shop/api/pembelian/transaksi | Success: 200 |
| Buat Janji Temu Dokter | Melihat Jadwal Praktek Dokter | [https://berobatplus.shop/api/master/ahli/jadwal\_praktek/{id\_detail\_praktek}](https://berobatplus.shop/api/konsumen/checkout) | Success: 200 |
| Melihat Rumah Sakit dan Apotek Terdekat | <https://berobatplus.shop/api/master/rumah_sakit/data/find_nearest> | Success: 200 |
| Melihat Dokter berdasarkan Spesialisnya di RS | https://berobatplus.shop/api/master/ahli/praktek/praktek\_rs | Success: 200 |
| Melihat Detail Rumah Sakit dan Apotek | https://berobatplus.shop/api/master/rumah\_sakit/data | Success: 200 |
| Membuat Janji Temu Dokter | https://berobatplus.shop/api/master/ahli/jadwal\_antrian | Success: 200 |
| Melihat Riwayat Kunjungan | https://berobatplus.shop/api/master/ahli/jadwal\_antrian | Success: 200 |
| Profil | Mengubah Profil Akun | <https://berobatplus.shop/api/akun/profil/nama_role/profil> | Success: 200 |
| Mengubah Kata Sandi Akun | https://berobatplus.shop/api/akun/change\_password | Success: 200 |
| Menambahkan Alamat Pengiriman | https://berobatplus.shop/api/master/alamat\_user | Success: 200 |
| Modul Search | Mencari Dokter dan Perawat terdekat | https://berobatplus.shop/api/akun/perawat | Success: 200 |
| Mencari Dokter dan Perawat berdasarkan Keahlian | https://berobatplus.shop/api/master/cari/keahlian | Success: 200 |
| Mencari Dokter berdasarkan Spesialisnya | [https://berobatplus.shop/api/master/spesialis/{id\_spesialis}/get\_dokter](https://berobatplus.shop/api/master/ahli/keahlian/master/) | Success: 200 |
| Mencari Rumah Sakit | <https://berobatplus.shop/api/master/cari/rumah_sakit> | Success: 200 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

|  |  |  |
| --- | --- | --- |
| Nama | : Muhamad Rafli Septian |  |
| Nomor Induk Mahasiswa | : 2003018 |
| Jenis Kelamin | : Laki-Laki |
| Tempat, Tanggal Lahir | : Tangerang, 26 September 2002 |
| Status | : Belum Menikah |
| Agama | : Islam |
| Asal Ijazah Sekolah | : Nama Sekolah | Tahun Ijazah |
| SD | : SDN Pondok Bahar 03 | * 2014 |
| SMP | : SMPN 18 Tangerang | * 2017 |
| SMA | : SMAN 3 Tangerang | * 2020 |
| Program Studi/Jurusan | : Teknik Informatika |  |
| Alamat Rumah Asal | : Pondok Bahar, Tangerang |  |
| No. Handphone | : 081411126356 |  |
| Email | : [rafliseptiannn25@gmail.com](mailto:rafliseptiannn25@gmail.com) |  |
| Nama Orang Tua | : Katna Suteja |  |
| Pekerjaan Orang Tua | : Karyawan Swasta |  |
| Judul Tugas Akhir | : Aplikasi Pelayanan Kesehatan Masyarakat berbasis Web menggunakan *Framework* VueJs |  |
| Pembimbing | : Adi Suheryadi, S.ST., M.Kom. |  |