

**IMPLEMENTASI METODE RAD DALAM PERANCANGAN
APLIKASI ANDROID MANAJEMEN KETERSEDIAAN
TEMPAT TIDUR PASIEN**

SKRIPSI



**BELAAZAHWA
2022230029**

**FAKULTAS ILMU KOMPUTER
PROGRAM STUDI INFORMATIKA
UNIVERSITAS PRABUMULIH
2026**

IMPLEMENTASI METODE RAD DALAM PERANCANGAN APLIKASI ANDROID MANAJEMEN KETERSEDIAAN TEMPAT TIDUR PASIEN

Untuk Memenuhi Salah Satu Syarat Mendapatkan Gelar Sarjana Komputer
Program Studi Informatika Universitas Prabumulih



**BELAAZAHWA
2022230029**

**FAKULTAS ILMU KOMPUTER
PROGRAM STUDI INFORMATIKA
UNIVERSITAS PRABUMULIH
2026**

HALAMAN PENGESAHAN

**IMPLEMENTASI METODE *RAD* DALAM PERANCANGAN APLIKASI *ANDROID*
MANAJEMEN KETERSEDIAAN TEMPAT TIDUR PASIEN**

SKRIPSI

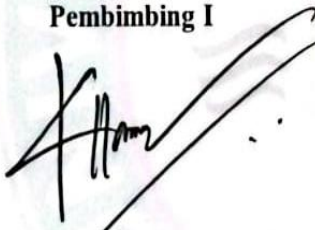
**Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Komputer
Program Studi Informatika Universitas Prabumulih**

Oleh :

**Belaazahwa
2022230029**

Prabumulih, 17 Juni 2026

Pembimbing I



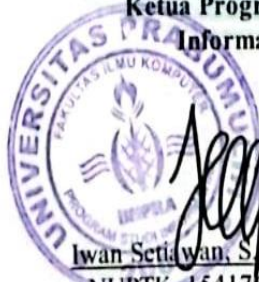
Khana Wijaya, S.Kom., M.Kom.
NUPTK. 7146768669130363

Pembimbing II



Riza Kartina, S.Kom., M.Kom.
NUPTK. 0549767668231073

**Ketua Program Studi
Informatika**

Iwan Setiawan, S.Kom., M.Kom.
NUPTK. 1541769670130303

HALAMAN PERSETUJUAN

Skripsi ini dengan judul “Implementasi Metode *RAD* dalam Perancangan Aplikasi *Android* Manajemen Ketersediaan tempat Tidur Pasien” telah dipertahankan dan disetujui Tim Penguji Skripsi Program Studi Informatika Fakultas Ilmu Komputer Universitas Prabumulih pada tanggal 17 Juni 2026.

Prabumulih, 17 Juni 2026

Tim Penguji Tugas Akhir
Ketua:

Khana Wijaya, S.Kom., M.Kom
NUPTK. 7146768669130363



Anggota:

1. Riza Kartina, S.Kom., M.Kom
NUPTK. 0549767668231073



2. Nur Aini Hutagalung S.Kom., M.Si., M.Kom..
NUPTK. 2434764664300022



Mengetahui,
Dekan Fakultas Ilmu Komputer



(Khana Wijaya, S.Kom., M.Kom)
NUPTK. 7146768669130363

Ketua Program Studi
Informatika



(Iwan Setiawan, S.Kom., M.Kom)
NUPTK. 1541769670130303

PERNYATAAN INTEGRITAS

Saya yang bertanda tangan di bawah ini

Nama : Belaazahwa

NIM : 2022230029

Program Studi : Informatika

Menyatakan dengan sebenarnya bahwa, skripsi ini benar-benar merupakan hasil karya saya dan tidak terdapat karya orang lain, apabila di kemudian hari terbukti atau dapat dibuktikan bahwa tugas akhir ini hasil *plagiasi*, maka saya bersedia menerima sanksi sesuai dengan ketentuan yang berlaku atas perbuatan tersebut.

Prabumulih, 17 Juni 2026

Yang membuat pernyataan,



Belaazahwa

NIM. 2022230029

ABSTRAK

Implementasi Metode *RAD* dalam Perancangan Aplikasi *Android* Manajemen Ketersediaan Tempat Tidur Pasien

Belaazahwa, 2022230029, 2026

Rumah sakit umum daerah kota prabumulih sering mengalami kendala dalam mengelola ketersediaan tempat tidur pasien karena proses pengecekan yang masih dilakukan secara manual. Hal ini dapat menyebabkan pelayanan menjadi lebih lambat dan data yang diperoleh menjadi kurang akurat. Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk merancang dan membantu pihak rumah sakit dalam memantau ketersediaan tempat tidur pasien secara *real-time*. Penelitian ini menggunakan metode penelitian deskriptif pendekatan kualitatif dan metode pengembangan sistem yang digunakan yaitu *Rapid Application Development (RAD)* karena metode ini memungkinkan proses pembuatan aplikasi yang dilakukan lebih cepat dengan melibatkan pengguna langsung dalam setiap tahap pengembangan. Aplikasi ini dibuat menggunakan *Framework Flutter* dengan Bahasa pemograman *Dart*. Hasil dari penelitian ini diharapkan dapat mempermudah petugas rumah sakit dalam mengecek ketersediaan tempat tidur, serta meningkatkan ketepatan dan efisiensi kerja di lingkungan rumah sakit. Serta memberikan kemudahan bagi pasien untuk memperoleh informasi terkait ketersediaan tempat tidur dan melakukan komunikasi dengan pihak rumah sakit tanpa perlu datang langsung.

Kata Kunci: Aplikasi *Android*, *Dart*, *Flutter*, Ketersediaan Tempat Tidur Pasien, *Rapid Application Development (RAD)*.

ABSTRACT

Implementation of the RAD Method in Designing an Android Application for Patient Bed Availability Management

Belaazahwa, 2022230029, 2026

Hospitals often face difficulties in managing patient bed availability because the checking process is still performed manually. This can lead to delays in service and inaccurate data. Based on this issue, this study aims to design and develop an Android application that helps hospitals monitor patient bed availability in real-time. The study uses a descriptive qualitative approach, and the system development method applied is Rapid Application Development (RAD), which allows faster application development by involving users in each stage of the process. The application is built using the Flutter framework and the Dart programming language. The results of this study are expected to help hospital staff easily check bed availability, speed up patient placement, and improve accuracy and efficiency in hospital operations, as well as provide convenience for patients in accessing bed availability information and communicating with hospital staff without having to visit the hospital directly.

Keywords: Android Application, Bed Availability, Dart, Flutter, Rapid Application Development (RAD).

KATA PENGANTAR

Puji dan syukur penulis ucapkan kehadirat Allah SWT atas limpahan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan Skripsi yang berjudul Implementasi Metode *RAD* dalam Manajemen ketersediaan Tempat Tidur Pasien, Skripsi ini merupakan salah satu syarat untuk mendapatkan gelar Sarjana di Program Studi S1 Informatika Fakultas Ilmu Komputer Universitas Prabumulih. Penulis banyak menerima bantuan, arahan, dan bimbingan dari berbagai pihak. Penulis mengucapkan terima kasih kepada:

1. Ibu Dr. Yuniar Pratiwi, S.Si., M.Si selaku Rektor Universitas Prabumulih.
2. Bapak Andi Christian, S.Kom.,M.Kom selaku Wakil Rektor Universitas Prabumulih.
3. Bapak Khana Wijaya, S.Kom., M.Kom selaku Dekan Fakultas Universitas Prabumulih sekaligus pembimbing I.
4. Ibu Suhartini, S.Kom., M.Kom selaku Wakil Dekan Bidang Keuangan Fakultas Ilmu Komputer Universitas Prabumulih.
5. Ibu Nur Aini Hutagalung, S.Kom.,M.Si.,M.Kom selaku Wakil Dekan Bidang Kemahasiswaan Fakultas Ilmu Komputer Universitas Prabumulih.
6. Bapak Iwan Setiawan, S.Kom.,M.Kom selaku Ketua Program Studi Informatika dan Pembimbing Akademik
7. Ibu Riza Kartina, S.Kom., M.Kom sebagai Pembimbing II.
8. Bapak dan Ibu Dosen Program Studi Informatika Fakultas Ilmu Komputer Universitas Prabumulih.
9. Bapak/Ibu staf administrasi Fakultas Ilmu Komputer Universitas Prabumulih.
10. Staf Administrasi RSUD Prabumulih yang telah memberikan izin dan membantu dalam penelitian ini.
11. Mama, Abang serta Seluruh Keluarga Tercinta yang telah memberikan doa, dukungan, motivasi selama proses penyusunan skripsi ini.
12. Teman – teman Program Studi Informatika Universitas Prabumulih Angkatan 2022 yang telah memberikan semangat, bantuan, dan dukungan selama penyusunan skripsi ini.

Penulisan menyadari bahwa Skripsi ini masih terdapat kekurangan. Oleh karena itu, penulis mengharapkan kritik dan saran yang bersifat membangun untuk kesempurnaan Skripsi. Semoga skripsi ini dapat memberikan manfaat bagi pembaca dan menjadi referensi bagi penelitian selanjutnya.

Prabumulih, 17 Juni 2026

Bul.

Belaazahwa
2022230029



DAFTAR ISI

	Halaman
COVER DALAM	ii
HALAMAN PENGESAHAN	iii
HALAMAN PERSETUJUAN.....	iv
HALAMAN INTEGRITAS	v
ABSTRAT	vi
ABSTRACT.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI	x
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR.....	xv
DAFTAR LAMPIRAN	xvi
DAFTAR SINGKATAN	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Sistematika Penulisan	4
1.7 BAB II TINJAUAN PUSTAKA	6
2.1 Pengertian Implementasi	6
2.2 Pengertian Metode <i>Rapid Application Development (RAD)</i>	6
2.3 Pengertian Perancangan	7
2.4 Pengertian Aplikasi.....	7
2.5 Pengertian <i>Android</i>	7
2.6 Pengertian Manajemen	8
2.7 Pengertian <i>Visual Studio Code (VS Code)</i>	8
2.8 Pengertian <i>Flutter</i>	9
2.9 Pengertian <i>Dart</i>	10
2.10 Pengertian <i>Firebase</i>	11

2.11 Penelitian Terdahulu.....	11
2.12 Kerangka Berpikir Penelitian.....	13
BAB III METODE PENELITIAN	16
3.1 Objek dan Jadwal Penelitian	16
3.1.1 Sejarah RSUD Prabumulih	16
3.1.2 Visi & Misi RSUD Prabumulih.....	17
3.1.3 Struktural Organisasi RSUD Prabumulih	18
3.1.4 Jadwal penelitian	21
3.2 Metode Penelitian	22
3.3 Sumber Data.....	22
3.4 Teknik Pengumpulan Data	23
3.5 Metode Pengembangan Sistem	24
3.6 Alat Bantu Perancangan	25
3.6.1 Alat Bantu Utama	28
3.6.2 Alat Bantu Desain	29
3.6.3 Alat Bantu <i>Debugging</i> dan Pengembangan.....	29
3.7 Pengujian Sistem	30
BAB IV ANALISA DAN PERANCANGAN	31
4.1 Analisa Sistem	31
4.1.1 Analisa dan Evaluasi Sistem Berjalan.....	31
4.1.2 Analisa Kebutuhan	32
4.2 Perancangan Sistem Desain yang diusulkan	37
4.2.1 Sistem (Logis).....	37
4.2.2 Perancangan Database.....	47
4.2.3 Perancangan Antarmuka	50
BAB V HASIL DAN PEMBAHASAN.....	54
5.1 Spesifikasi Perangkat.....	54
5.1.1 Perangkat Keras yang digunakan	54
5.1.2 Perangkat Lunak yang digunakan	54
5.2 Implementasi <i>Database</i>	55
5.3 Tampilan Antarmuka.....	60
5.4 Hasil Pengujian Sistem.....	69
BAB VI KESIMPULAN DAN SARAN	122

6.1 Kesimpulan	122
6.2 Saran	122
DAFTAR PUSTAKA.....	124
DAFTAR LAMPIRAN	128



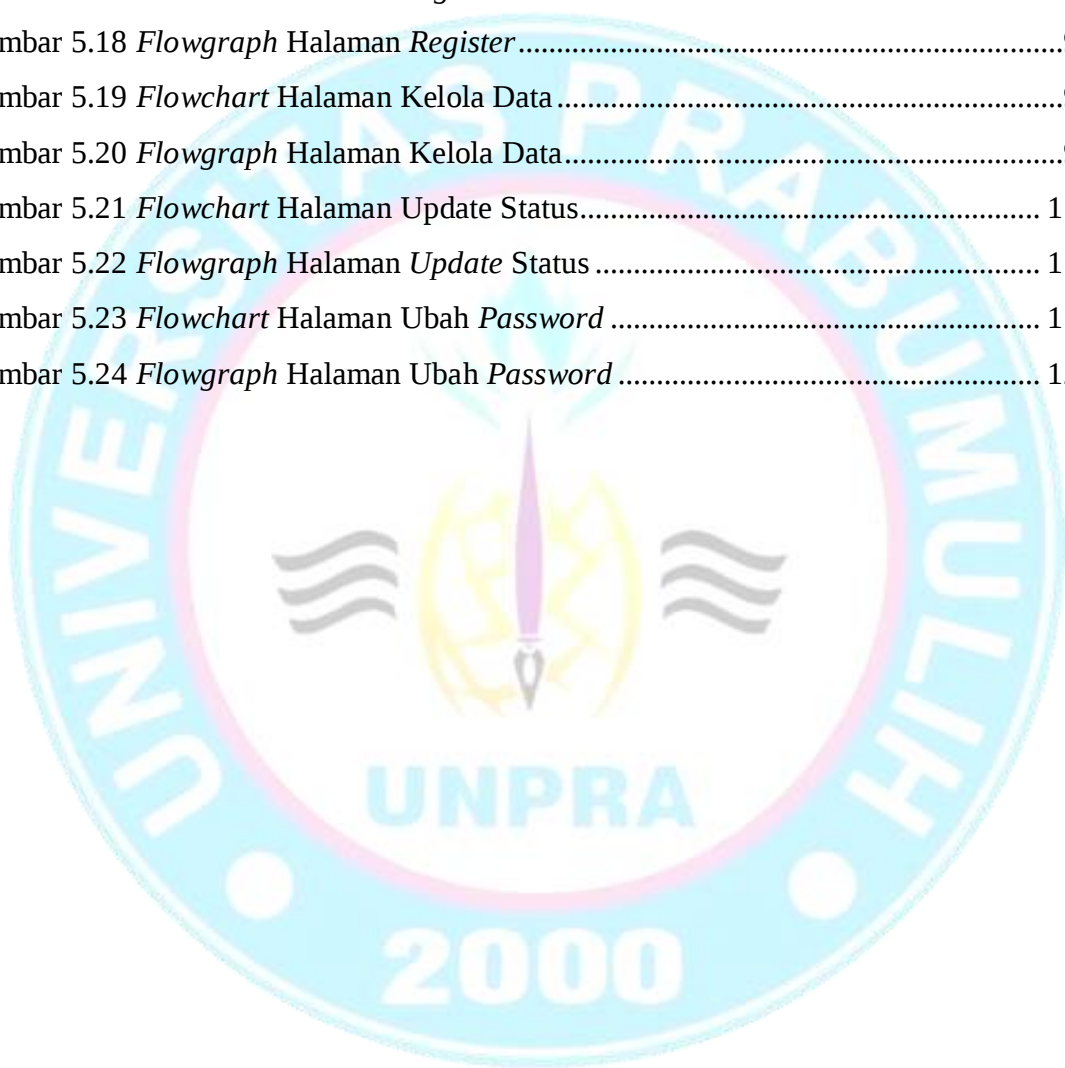
DAFTAR TABEL

	Halaman
Tabel 2.1 Penelitian Terdahulu	11
Tabel 3.1 Jadwal Penelitian	21
Tabel 3.2 <i>Use Case Diagram</i>	26
Tabel 3.3 <i>Activity Diagram</i>	27
Tabel 3.4 <i>Entity Relantionship Diagram</i>	28
Tabel 4,1 <i>Database</i> Pengguna	47
Tabel 4.2 <i>Database</i> Ruang	48
Tabel 4.3 <i>Database</i> Riwayat Status	48
Tabel 4.4 <i>Database</i> Obrolan.....	49
Tabel 4.5 <i>Database</i> Riwayat Password	49
Tabel 5.1 Spesifikasi Perangkat Keras.....	54
Tabel 5.2 Spesifikasi Perangkat lunak	55
Tabel 5.3 Identifikasi <i>Statement</i> Halaman <i>Login</i>	72
Tabel 5.4 <i>Test Case</i> Halaman <i>Login</i>	73
Tabel 5.5 Jalur Independent Halaman <i>Login</i>	77
Tabel 5.6 Identifikasi <i>Statement</i> Halaman <i>Register</i>	85
Tabel 5.7 <i>Test Case</i> Halaman <i>Register</i>	88
Tabel 5.8 Jalur Independent Halaman <i>Register</i>	92
Tabel 5.9 Identifikasi <i>Statement</i> Halaman Kelola Data.....	94
Tabel 5.10 <i>Test Case</i> Halaman Kelola Data.....	94
Tabel 5.11 Jalur Independent Halaman Kelola Data	98
Tabel 5.12 Identifikasi <i>Statement</i> Halaman <i>Update</i> Status.....	106
Tabel 5.13 <i>Test Case</i> Halaman <i>Update</i> Status	108
Tabel 5.14 Jalur Independent Halaman <i>Update</i> Status.....	112
Tabel 5.15 Identifikasi <i>Statement</i> Halaman Ubah <i>Password</i>	116
Tabel 5.16 <i>Test Case</i> Halaman Ubah <i>Password</i>	118
Tabel 5.17 Jalur Independent Halaman Ubah <i>Password</i>	121

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Kerangka Berpikir Penelitian.....	14
Gambar 3.1 Struktur Organisasi Struktur Organisasi	18
Gambar 3.2 Metode <i>Rapid Application Development (RAD)</i>	24
Gambar 4.1 <i>Use Case</i> Sistem yang diusulkan.....	38
Gambar 4.2 <i>Activity Diagram Login</i>	40
Gambar 4.3 <i>Activity Diagram Dashboard</i>	41
Gambar 4.4 <i>Activity Diagram</i> Kelola Data	42
Gambar 4.5 <i>Activity Diagram Update Status</i>	43
Gambar 4.6 <i>Activity Diagram</i> Kelola Akun	44
Gambar 4.7 <i>Activity Diagram</i> Pesan.....	45
Gambar 4.8 <i>Activity Diagram</i> Ubah Password	46
Gambar 4.9 <i>Database Entity Relationship Diagram (ERD)</i>	47
Gambar 4.10 Rancangan Halaman <i>Login dan Register</i>	50
Gambar 4.11 Rancangan Halaman <i>Dashboard</i>	50
Gambar 4.12 Rancangan Halaman Kelola Data.....	51
Gambar 4.13 Rancangan Halaman Kelola Akun	51
Gambar 4.14 Rancangan Halaman <i>Update Status</i>	52
Gambar 4.15 Rancangan Halaman pesan	52
Gambar 4.16 Rancangan Halaman Pengaturan Akun.....	53
Gambar 4.17 Rancangan Halaman Ubah Password	53
Gambar 5.1 Halaman <i>Database</i> Pengguna.....	55
Gambar 5.2 Halaman <i>Database</i> Ruang	56
Gambar 5.3 Halaman <i>Database</i> Obrolan.....	57
Gambar 5.4 Halaman <i>Database</i> Riwayat Status	58
Gambar 5.5 <i>Database</i> Riwayat Password	59
Gambar 5.6 Antarmuka Halaman <i>Login</i>	60
Gambar 5.7 Antarmuka Halaman <i>Register</i>	61
Gambar 5.8 Antarmuka Halaman <i>Dashboard</i>	62
Gambar 5.9 Antarmuka Halaman Kelola Data Ruangan.....	63
Gambar 5.10 Antarmuka Halaman Kelola Akun	64

Gambar 5.11 Antarmuka Halaman <i>Update Status</i>	65
Gambar 5.12 Antarmuka Halaman <i>Pesan</i>	66
Gambar 5.13 Antarmuka Halaman <i>Pengaturan Akun</i>	67
Gambar 5.14 Antarmuka Halaman <i>Ubah Password</i>	68
Gambar 5.15 <i>Flowchart</i> Halaman <i>Login</i>	74
Gambar 5.16 <i>Flowgraph</i> Halaman <i>Login</i>	76
Gambar 5.17 <i>Flowchart</i> Halaman <i>Register</i>	89
Gambar 5.18 <i>Flowgraph</i> Halaman <i>Register</i>	91
Gambar 5.19 <i>Flowchart</i> Halaman <i>Kelola Data</i>	95
Gambar 5.20 <i>Flowgraph</i> Halaman <i>Kelola Data</i>	97
Gambar 5.21 <i>Flowchart</i> Halaman <i>Update Status</i>	110
Gambar 5.22 <i>Flowgraph</i> Halaman <i>Update Status</i>	111
Gambar 5.23 <i>Flowchart</i> Halaman <i>Ubah Password</i>	119
Gambar 5.24 <i>Flowgraph</i> Halaman <i>Ubah Password</i>	120



DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Surat Izin Permohonan Pengambilan Data.....	128
Lampiran 2 Surat Balasan Pengambilan Data	129
Lampiran 3 Observasi Wawancara	130
Lampiran 4 Draft Pertanyaan Wawancara.....	131



DAFTAR SINGKATAN

RSUD	: Rumah Sakit Umum Daerah
RAD	: <i>Rapid Application Development</i>
VSCode	: <i>Visual Studio Code</i>
UML	: <i>Unified Modified Language</i>
ERD	: <i>Entity Relantionship Diagram</i>



BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi yang semakin canggih di era globalisasi, tentunya memberikan begitu banyak aspek perubahan yang mengharuskan setiap orang untuk menggunakan berbagai macam peralatan teknologi. Sebagai alat bantu dalam menjalankan berbagai aktivitas sebagai sarana dalam mendukung produktifitas. Jika kita perhatikan, teknologi menyeluruh yang saat ini banyak sekali digunakan oleh seluruh lapisan masyarakat di Indonesia adalah *smartphone* dengan sistem *Android*.(Hidayati, 2024)

Rumah sakit adalah tempat yang memberikan pelayanan kesehatan kepada masyarakat secara menyeluruh mulai dari pelayanan rawat jalan, rawat inap, hingga gawat darurat. Salah satu hal yang penting dalam kegiatan rawat inap adalah pengelolaan ketersediaan tempat tidur pasien. Informasi mengenai jumlah tempat tidur yang tersedia, terisi, atau sedang dalam perawatan. Sangat penting bagi pihak rumah sakit untuk mengatur kapasitas ruang, mengelola alur pasien, dan mengambil keputusan secara cepat dan tepat.(Sinta Sani et al., 2022)

Rumah Sakit Umum Daerah (RSUD) Kota Prabumulih merupakan rumah sakit rujukan utama di wilayah Kota Prabumulih dan sekitarnya. Sebagai fasilitas kesehatan milik Pemerintah Daerah RSUD Prabumulih memiliki peran penting dalam memberikan pelayanan kesehatan kepada masyarakat. Seiring dengan meningkatnya jumlah pasien setiap tahunnya, RSUD Prabumulih sering menghadapi tantangan besar dalam mengelola 167 tempat tidur yang tersebar dalam berbagai kelas serta berbagai departemen spesialisasi. Berdasarkan hasil wawancara, meskipun RSUD Prabumulih telah memiliki sistem operasional untuk mengecek ketersediaan tempat tidur namun mereka masih menggunakan alat komunikasi telepon ke setiap ruangan untuk mengetahui ketersediaan tempat tidur.

Hal ini mengakibatkan penempatan pasien menjadi lebih lambat dan sangat berpotensi menimbulkan ketidakakuratan data permasalahan tersebut bisa berdampak langsung kepada efisiensi operasional pihak rumah sakit dan kualitas dalam pelayanan terhadap pasien. Keterlambatan dalam mengetahui ketersediaan tempat tidur dapat menyebabkan

pasien mengantri lama, maka dari itu sangat dibutuhkan sebuah aplikasi *Android* untuk memudahkan petugas dalam mengecek ketersediaan tempat tidur pasien secara cepat.

Untuk mengatasi permasalahan tersebut, diperlukan sebuah solusi berbasis teknologi yang mampu menyediakan informasi ketersediaan tempat tidur secara cepat dan akurat. Oleh karena itu, peneliti mengusulkan untuk membangun sebuah aplikasi berbasis *Android* yang berfungsi untuk memantau status ketersediaan tempat tidur. Aplikasi ini dirancang untuk digunakan oleh petugas administrasi sebagai pengelola data ruangan, pengelola akun perawat ruangan dan bisa mengirim pesan ke pasien, perawat ruangan sebagai pihak yang akan memperbarui status ketersediaan tempat tidur secara langsung dari lapangan serta pasien sebagai pengguna yang dapat melihat informasi terkait ketersediaan tempat tidur dan dapat melakukan komunikasi dengan staf administrasi terkait informasi lebih lanjut.

Dalam proses pengembangan aplikasi, peneliti menggunakan metode *Rapid Application Development RAD* karena metode ini memungkinkan pengembangan sistem dilakukan secara cepat dengan melibatkan pengguna dalam proses perancangan dan pengujian aplikasinya. Dengan demikian aplikasi yang dibangun diharapkan benar-benar sesuai dengan kebutuhan RSUD Prabumulih serta dapat meningkatkan akurasi data dan mempercepat pelayanan kepada pasien.

1.2 Rumusan Masalah

1. Bagaimana merancang dan membangun aplikasi *Android* yang dapat membantu pihak RSUD Prabumulih dalam mengelola, memantau ketersediaan tempat tidur pasien?
2. Bagaimana penerapan metode *Rapid Application Development (RAD)* dalam proses pengembangan aplikasi tersebut?
3. Apakah aplikasi yang dibangun dapat membantu petugas administrasi, perawat ruangan serta pasien dalam mempercepat proses pengecekan untuk meningkatkan keakuratan informasi ketersediaan tempat tidur pasien?

1.3 Batasan Masalah

1. Penelitian ini hanya berfokus kepada perancangan dan pembangunan aplikasi *Android* untuk mengelola, memantau ketersediaan.
2. Aplikasi yang dibangun hanya menyediakan informasi status tempat tidur (kosong, terisi, dalam perbaikan).

3. Aplikasi ini hanya dapat digunakan oleh petugas administrasi, perawat ruangan dan pasien.

1.4 Tujuan Penelitian

1. Untuk merancang dan membangun aplikasi *Android* yang dapat membantu RSUD Prabumulih dalam mengelola, memantau ketersediaan tempat tidur secara cepat dan akurat.
2. Menyediakan sistem informasi secara langsung yang bisa digunakan oleh calon pasien agar mereka bisa mengetahui kondisi ruang rawat inap terlebih dahulu sebelum datang ke rumah sakit.
3. Mempercepat proses pengecekan serta meningkatkan keakuratan informasi ketersediaan tempat tidur.

1.5 Manfaat Penelitian

1. Bagi Penulis

Penelitian ini diharapkan dapat memberikan pengalaman langsung kepada penulis dalam menerapkan apa yang sudah dipelajari selama masa perkuliahan ke dalam praktik nyata, khususnya dalam merancang dan membangun sebuah aplikasi *Android*.

2. Bagi Rumah Sakit

Hasil dari penelitian ini diharapkan dapat membantu pihak rumah sakit dalam mengelola dan memantau ketersediaan tempat tidur pasien secara cepat dan akurat. Dengan adanya aplikasi ini proses dalam penempatan pasien dapat dilakukan secara lebih cepat, efisien dan akurat.

3. Bagi Akademik

Penelitian ini diharapkan dapat menjadi referensi dan bahan pembelajaran dalam Program Studi Informatika, khususnya yang berkaitan dengan pengembangan aplikasi *Android* dengan menggunakan metode *Rapid Application Development (RAD)*.

4. Bagi Pasien

Dengan adanya aplikasi ini diharapkan pasien dapat mengetahui kondisi ketersediaan ruang rawat inap terlebih dahulu sebelum datang ke rumah sakit sehingga

dapat mengurangi waktu tunggu dan membantu proses pelayanan menjadi lebih efektif.

1.6 Sistematika Penelitian

Sistematika penulisan ini membantu supaya skripsi lebih terarah pada masalah yang dibahas, sehingga tidak menyimpang dari ketentuan dan tujuan yang ditetapkan dengan sistematika penelitian, sebagai berikut :

BAB I PENDAHULUAN

Pada bab ini berisikan latar belakang, rumusan masalah, batasan masalah, tujuan penelitian dan sistematika penelitian.

BAB II TINJAUAN PUSTAKA

Pada bab ini umumnya berisikan teori yang menjelaskan tentang pemahaman-pemahaman yang berisi hasil kajian, referensi serta kutipan jurnal yang memiliki judul serupa, dalam bab ini juga membahas tentang pengertian, penelitian terdahulu dan kerangka pemikiran.

BAB III METODE PENELITIAN

Pada bab ini biasanya berisikan tentang objek/tempat dilakukannya penelitian dan penjelasan tentang penelitian apa yang dilakukan di tempat itu dari awal sampai akhir.

BAB IV ANALISA DAN PERANCANGAN

Pada bab ini akan dibahas mengenai proses awal pembuatan sebuah sistem dimulai dari analisis masalah pada sistem yang lama dan memberikan solusi yang tepat terkait masalah yang terjadi. Bab ini juga menjelaskan perancangan sistem yang dibutuhkan dengan menggunakan *UML*(*UseCase*, *Activity Diagram* dan *Entity Relationship Diagram*), serta perancangan antarmuka (*UI/UX*).

BAB V HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas mengenai hasil akhir dari sebuah sistem dimulai dari spesifikasi perangkat keras dan perangkat lunak yang digunakan hingga

implementasi basis data dan tampilan antarmuka aplikasi. Selain itu juga membahas mengenai hasil pengujian sistem untuk memastikan sebuah semua fitur telah berjalan dengan baik dan sesuai dengan tujuan yang direncanakan.

BAB VI KESIMPULAN DAN SARAN

Pada bab ini akan membahas tentang rangkuman dari hasil penelitian ke dalam poin-poin kesimpulan yang menjawab rumusan masalah serta saran-saran yang diberikan oleh penulis untuk pengembangan lebih lanjut kedepannya agar sistem dapat menjadi lebih sempurna dan bermanfaat bagi pengguna dan peneliti di masa depan.



BAB II

TINJAUAN PUSTAKA

2.1 Pengertian Implementasi

Implementasi adalah proses untuk menerapkan suatu sistem informasi yang telah dibuat agar pengguna dapat menggunakannya menggantikan sistem informasi yang lama (Pujilestari, 2023). Implementasi adalah suatu tindakan atau pelaksanaan dari sebuah rencana yang sudah disusun secara matang dan terperinci. Implementasi biasanya dilakukan setelah perencanaan sudah dianggap sempurna. (Martani, 2023)

Berdasarkan kedua penjelasan diatas mengenai implementasi bisa disimpulkan bahwa implementasi dapat diartikan sebagai proses penerapan hasil rancangan atau ide yang direncanakan dengan baik ke dalam bentuk nyata, sehingga dapat dijalankan dalam lingkungan operasional untuk mencapai tujuan yang telah ditetapkan.

2.2 Pengertian Metode *Rapid Application Development (RAD)*

Metode *Rapid application development (RAD)* atau pengembangan aplikasi cepat merupakan suatu proses yang dilakukan dalam membangun aplikasi dengan tujuan mempercepat proses pengerjaan dan memprioritaskan perangkat lunak yang terdiri dari tahap *requirement planning*, sistem desain, panduan dan implementasi. (Voni Besin, 2023)

Metode *Rapid Application Development (RAD)* adalah metode pengembangan sistem informasi dengan waktu yang singkat yang dianggap cocok untuk proyek perangkat lunak. Dalam *RAD* pendekatan yang digunakan adalah iteratif yang berarti bahwa model kerja sistem dibangun pada awal tahap pengembangan dan kemudian model tersebut dapat ditingkatkan atau diubah seiring perkembangan proyek. (Dalis, 2024)

Berdasarkan kedua penjelasan diatas bisa disimpulkan bahwa metode *Rapid Application Development (RAD)* adalah pendekatan pengembangan dengan melibatkan pengguna secara aktif sehingga setiap perubahan kebutuhan atau ide dari pengguna bisa dapat segera di implementasikan untuk menghasilkan sistem yang sesuai dengan keinginan dan kebutuhan pengguna.

2.3 Pengertian Perancangan

Perancangan adalah menganalisis secara mendalam terhadap kebutuhan pengguna yang nantinya hasil analisis tersebut akan dijadikan sebagai bahan untuk membuat desain awal sistem mulai dari rancangan database maupun diagram. (Dalis, 2024)

Perancangan adalah sebuah proses untuk mendefinisikan sesuatu yang akan dikerjakan dengan menggunakan teknik yang bervariasi serta didalamnya melibatkan deskripsi mengenai arsitektur serta detail komponen dan juga keterbatasan yang akan dialami dalam proses pengerjaannya. (Pujilestari, 2023)

Berdasarkan kedua penjelasan diatas perancangan berarti membuat gambaran atau rancangan yang jelas tentang bagaimana sistem tersebut akan bekerja, mulai dari alur kerja, data yang dibutuhkan hingga cara penggunaannya. Tahap ini penting karena dapat membantu memastikan sistem yang dibuat mudah digunakan dan sesuai dengan kebutuhan pengguna.

2.4 Pengertian Aplikasi

Aplikasi adalah perangkat lunak atau program yang telah dikembangkan untuk melakukan suatu tugas yang diinginkan oleh pengguna pada komputer, laptop maupun handphone (Bahat Nauli, 2024). Aplikasi merupakan suatu perangkat lunak (*software*) atau program komputer yang beroperasi pada sistem yang dibuat serta dikembangkan untuk melakukan perintah tertentu, Istilah aplikasi sendiri dalam dari bahasa Inggris disebut dengan “*Application*” yang bisa diartikan sebagai pemakaian atau penggunaan. (Syahputra et al., 2024)

Berdasarkan kedua penjelasan diatas aplikasi adalah perangkat lunak berbasis *Android* yang dirancang untuk mempermudah penggunaan dalam melakukan aktivitas sehari-hari. Dimulai dari mencari informasi, belajar, komunikasi dan hiburan. Dengan adanya aplikasi hal yang dilakukan secara manual tersebut bisa dilakukan dengan menggunakan perangkat *smartphone*.

2.5 Pengertian *Android*

Android adalah suatu platform masa depan karena android menjadi sebuah mobile platform pertama yang lengkap, terbuka dan bebas. (Hertantyo, 2021). *Android* adalah sistem operasi untuk perangkat *mobile* seperti *tablet* dan *smartphone* yang dikembangkan

oleh *google* dan *open handset alliance*. Sistem operasi ini berbasis linux yang telah dimodifikasi dan dikembangkan terutama untuk peralatan dengan antarmuka (*interface*) layar sentuh (*touch screen*). (Prasetio, 2022)

Menurut kedua pengertian diatas bisa disimpulkan *Android* merupakan sistem operasi terbuka dan lengkap yang dikembangkan oleh *Google* dan *Open Handset Alliance*, berbasis *Linux*, serta dirancang khusus untuk perangkat *mobile* seperti *smartphone* dan *tablet*. Sebagai platform yang bebas dan fleksibel, *Android* memiliki potensi besar untuk terus berkembang dan menjadi *platform* masa depan dalam dunia teknologi perangkat bergerak.

2.6 Pengertian Manajemen

Manajemen adalah proses pelaksanaan fungsi-fungsi manajemen meliputi perencanaan, pelaksanaan, pengendalian dan koordinasi yang dilakukan secara sistematis pada suatu proyek dengan menggunakan sumber daya yang tersedia secara efektif dan efisien untuk mencapai tujuan proyek secara optimal. (Putri, 2024)

Manajemen adalah proses merencanakan, mengorganisasikan, mengarahkan, dan mengendalikan kegiatan untuk mencapai tujuan organisasi secara efektif dan efisien dengan menggunakan sumber daya organisasi. (Habsy, 2024)

Berdasarkan kedua pengertian diatas manajemen ialah proses perencanaan, pengelolaan, pengorganisasian suatu sumber daya agar dapat memperoleh hasil yang sesuai optimal.

2.7 Pengertian Visual Studio Code (VS Code)

Visual Studio Code (VS Code) ini adalah sebuah teks editor ringan dan handal yang dibuat oleh *Microsoft* untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi *Linux*, *Mac*, dan *Windows*. *Teks editor* ini secara langsung mendukung Bahasa pemrograman *JavaScript*, *Typescript*, dan *Node.js*, serta bahasa pemrograman lainnya dengan bantuan plugin yang dapat dipasang via marketplace *Visual Studio Code* (seperti *C++*, *C#*, *Python*, *Go*, *Java*, dst). Banyak sekali fitur-fitur yang disediakan oleh *Visual Studio Code*, diantaranya *Intellisense*, *Git Integration*, *Debugging*, dan fitur ekstensi yang menambah kemampuan teks editor. Fitur-fitur tersebut akan terus bertambah seiring dengan bertambahnya versi *Visual Studio Code*. Pembaruan versi *Visual Studio Code* ini

juga dilakukan berkala setiap bulan, dan inilah yang membedakan *VS Code* dengan teks editor-teks editor yang lain. Teks editor *VS Code* juga bersifat *open source*, yang mana kode sumbernya dapat kalian lihat dan kalian dapat berkontribusi untuk pengembangannya. Kode sumber dari *VS Code* ini pun dapat dilihat di *link Github*. Hal ini juga yang membuat *VS Code* menjadi favorit para pengembang aplikasi, karena para pengembang aplikasi bisa ikut serta dalam proses pengembangan *VS Code* ke depannya (Aldi, 2025)

Visual Studio Code adalah sebuah teks editor ringan dan handal yang dibuat oleh *Microsoft* untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi *Linux*, *Mac*, dan *Windows*. *VS Code* adalah *software* yang sangat ringan namun kuat editor kode sumbernya yang berjalan dari *desktop* muncul dengan *built-in* dukungan untuk *JavaScript*, *naskah* dan *Node.js* dan memiliki *Array* beragam eksistensi yang tersedia untuk bahasa lain, termasuk *C++*, *C#*, *Phyton*, dan *PHP* (Nasution, 2024)

Berdasarkan kedua pengertian diatas *Visual Studio Code* adalah editor kode yang banyak digunakan oleh para pengembang karena kemampuannya yang dapat disesuaikan dengan berbagai kebutuhan pemograman. Karena kemudahan penggunaan, performa yang stabil serta sifatnya yang *open source* membuat *Visual Studio Code* menjadi pilahan utama bagi banyak pengembang aplikasi, baik pemula maupun profesional.

2.8 Pengertian *Flutter*

Flutter adalah teknologi *open source* yang di buat oleh *Google* untuk membuat aplikasi *mobile* pada *Android* dan *iOS*. Dikembangkan menggunakan bahasa pemrograman *Dart*, *Flutter* juga dapat digunakan untuk membuat aplikasi *multiplatform*. Kelebihan utamanya adalah penggunaan satu kode untuk berbagai platform, yang mempercepat dan menyederhanakan proses pengembangan aplikasi.(Kalsum, 2024).

Flutter adalah suatu *framework* yang bersifat *open source*, dikembangkan oleh *Google* untuk membangun antar muka (*user interface*). *Flutter* dapat dikembangkan untuk perangkat yang memiliki sistem operasi *Android* dan *iOS*. Banyak perusahaan besar dunia telah mengadopsi *Flutter* untuk mengembangkan aplikasi. Sebut saja *Google*, *Alibaba Group*, dan *Grab*. *Flutter* memiliki beberapa keunggulan yaitu pertama sebagai *cross platform* yang mana dapat dijalankan pada beberapa platform yang berbeda. Dengan menggunakan *flutter*, dapat membuat aplikasi *Android* dan *iOS* sekaligus. Selain *mobile*,

dapat juga membuat aplikasi *web* dan *desktop*. Tentunya ini akan menghemat waktu, dikarenakan tidak perlu mempelajari bahasa *native* masing – masing platform.(Fiorenza, 2023)

Berdasarkan kedua pengertian diatas *Flutter* Adalah sebuah *framework open source* yang dikembangkan oleh *google* untuk membangun aplikasi lintas platfotm (*multiplatform*) menggunakan satu basis kode. *Framework* ini menggunakan Bahasa pemograman *dart* yang dirancang untuk membuat antarmuka pengguna (*UI*) yang cepat, respontif dan konsisten dan dapat langsung digunakan pada sistem operas seperti *Android*, *Ios*, *web* dan *detoskop*. Dengan kemampuannya sebagai *cross-platform*, *flutter* memudahkan proses pengembangan karena pengembang tidak perlu membuat aplikasi berbeda untuk setiap *platfrom*.

2.9 Pengertian Dart

Dart adalah bahasa pemrograman tingkat tinggi yang dikembangkan oleh *Google*. Pertama kali diperkenalkan pada tahun 2011, *Dart* dirancang dengan tujuan memberikan solusi pemrograman modern untuk pengembangan aplikasi lintas *platform*, baik untuk *web*, *server*, maupun aplikasi *mobile*. Bahasa ini memiliki sintaks yang sederhana dan mudah dipelajari, serta mendukung berbagai paradigma pemrograman, seperti pemrograman berorientasi objek, pemrograman asinkron, dan pemrograman fungsional (Sulistya, 2025).

Dart merupakan *programming language* lintas *platform* atau *platform* independen yang artinya dapat dijalankan pada sistem operasi yang berbeda seperti *Windows*, *Linux*, *Unix* dan *MacOS*, dll yang awalnya dikembangkan oleh *Google* dan kemudian disetujui sebagai standar oleh Ecma, yang saat ini digunakan untuk membangun aplikasi *web*, *server*, *desktop*, dan seluler (Jamjani, 2024)

Berdasarkan dari kedua pengertian diatas *Dart* Adalah bahasa pemograman yang sederhana yang dapat memudahkan pengembang dalam membangun aplikasi dengan cepat, responsive, dan efisien pada berbagai perangkat lunak. Selain itu dukungan terhadap fitur asinkron, pemograman berioorientasi objek serta integrasinya dengan *Flutter* membuat *Dart* sangat cocok digunakan dalam pembuatan aplikasi *mobile*.

2.10 Pengertian *Firebase*

Firebase adalah platform pengembangan aplikasi yang dibeli oleh *Google* pada tahun 2014. Ini memungkinkan pengembang untuk mengintegrasikan fitur berbasis *cloud* ke dalam aplikasi *mobile* dan *web* dengan mudah. *Firebase* menyediakan berbagai fitur seperti otentikasi pengguna, basis data *real-time*, penyimpanan *file* berbasis *cloud*, pesan instan, dan analitik (Kalsum, 2024).

Firebase adalah suatu layanan dari *Google* untuk memberikan kemudahan bahkan mempermudah para *developer* aplikasi dalam mengembangkan aplikasinya. *Firebase* alias *Backend as a Service (BaaS)* merupakan Solusi yang ditawarkan oleh *Google* untuk mempercepat pekerjaan *developer*. (Suhendro, 2021)

Berdasarkan kedua pengertian diatas *firebase* Adalah sebuah platfrom yang dikembangkan dan dikelola oleh google untuk mempermudah pengembangan aplikasi *mobile* maupun *web*. *Firebase* menyediakan berbagai layanan seperti autentikasi pengguna, basis data *real-time*, penyimpanan *file*, dan berbagai fitur lainnya yang dapat membantu mempercepat proses pembuatan aplikasi.

2.11 Penelitian Terdahulu

Tabel 2.1 Penelitian Terdahulu

No.	Jurnal	Penulis	Tahun	Pembahasan
1	Analisis Ketersediaan Tempat Tidur untuk Pasien Covid-19 pada Rumah Sakit di Daerah Kota Bandung Berbasis Web Siranap.	Alfadli M.S, dkk	2021	Membahas penggunaan SIRANAP untuk menampilkan ketersediaan tempat tidur pasien Covid-19 secara <i>real-time</i> agar data akurat dan mudah diakses masyarakat dan rumah sakit.
2	Sistem informasi <i>inventory</i> aset dan logistik rumah sakit bina kasih berbasis website dengan menggunakan metode <i>Rapid Application Development (RAD)</i> .	Toni Limbong, dkk	2022	Membahas penerapan RAD untuk sistem <i>inventory</i> dan logistik rumah sakit, membantu pencatatan, pemantauan, dan pengendalian barang secara cepat.

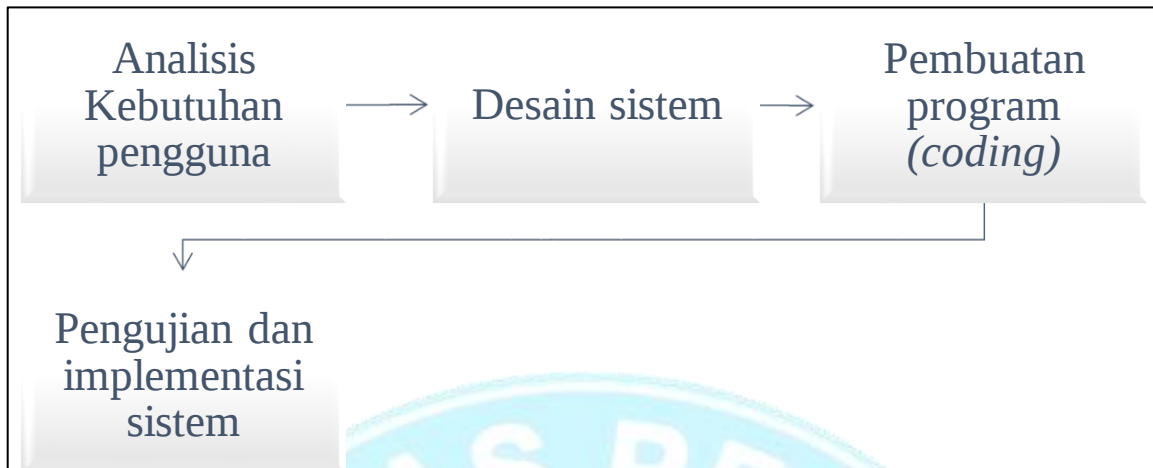
3	Perancangan sistem informasi tempat tidur berbasis <i>web</i> di rumah sakit TNI.AU.DR.M.Salamun.	Intan Pujiulestari,dkk	2023	Membahas sistem untuk mempermudah petugas mengecek ketersediaan tempat tidur secara <i>real-time</i> menggunakan metode <i>waterfall</i> , diuji dengan <i>UAT</i> .
4	Perancangan dan pembuatan sistem informasi rawat inap di RSUD Baesuki berbasis <i>website</i> .	Liana Ulfa,dkk	2023	Membahas sistem yang mempercepat pencatatan dan pengecekan ruangan kosong, memungkinkan input data pasien dan pemantauan ruang secara otomatis, serta informasi dapat diakses masyarakat secara <i>real-time</i> .
5	Sistem informasi keperawatan pada rumah sakit Dr. A.K.Gani dengan metode <i>Rapid Application Development (RAD)</i> .	Nyimas Sopiah,dkk	2023	Membahas pengembangan sistem keperawatan dengan <i>RAD</i> untuk mempermudah pencatatan dan pelaporan data pasien, pengembangan lebih cepat dengan keterlibatan pengguna.
6	Sistem informasi ketersediaan ruang rawat inap pada rumah sakit Bhakti Kasih Polewali.	Mustika,dkk	2024	Membahas aplikasi rawat inap berbasis <i>prototype</i> yang memudahkan pengelolaan data sesuai kebutuhan, meningkatkan efisiensi kerja perawat dan mengurangi kesalahan data.
7	Rancang bangun aplikasi rekam elektronik berbasis <i>website</i> menggunakan metode <i>RAD</i> pada klinik pratama rumah detensi surabaya.	Abdan Ifandy, dkk	2024	Membahas penggunaan <i>RAD</i> dan desain <i>UML</i> untuk membuat aplikasi rekam elektronik berbasis <i>web</i> , dikembangkan dengan <i>PHP</i> , <i>MySQL</i> , dan <i>Bootstrap</i> .
8	Pengembangan aplikasi <i>dashboard</i> pasien rawat inap pada rumah sakit		2024	Membahas aplikasi monitoring pasien rawat inap yang menampilkan status kamar dan tempat

	XYZ menggunakan <i>Framework laravel</i> .	Gusti Agung Ayu Searini, dkk		tidur real-time, membantu manajemen mengontrol kapasitas pasien
9	Hubungan efisiensi <i>Bed Occupancy Rate (BOR)</i> dengan kepuasan Pasien Rawat Inap di Koesnandi Bondowoso.	Sangkot,dkk	2024	Membahas efektivitas SIMRS dalam pengelolaan tempat tidur pasien, mempercepat pendaftaran dan penempatan pasien, data dapat digunakan untuk aplikasi <i>real-time</i> .
10	Pengelolaan data dengan metode <i>Rapid Application Development (RAD)</i> pada Posyandu pada Puskesmas Tanjung Ampalu Kabupaten Sijunjung.	Deri Putra,dkk	2024	Membahas sistem pengelolaan data ibu hamil, bayi, dan balita secara digital dengan RAD, mempermudah petugas dan memastikan sistem sesuai kebutuhan lapangan.

Sumber : dari penulis 2026

2.12 Kerangka Berfikir Penelitian

Kerangka berpikir penelitian merupakan suatu alur pemikiran yang tersusun secara rapi untuk menggambarkan proses pemikiran peneliti dalam memecahkan suatu masalah yang terjadi. Kerangka berpikir ini menjelaskan hubungan antara masalah yang diidentifikasi, teori atau konsep yang digunakan hingga solusi atau hasil yang ingin dicapai. Dengan adanya kerangka berpikir ini menjadikan penelitian ini memiliki arah yang jelas dan terstruktur karena setiap langkah yang dilakukan didasarkan pada logika dan dasar teori yang kuat. Melalui kerangka berpikir ini dapat dilihat bagaimana teori atau konsep yang diberikan sehingga menghasilkan suatu solusi yaitu adanya aplikasi *Android* terkait ketersediaan tempat tidur pasien di RSUD Prabumulih.



Sumber dari penulis, 2026

Gambar 2.1 Kerangka Berpikir Penelitian

Kerangka berpikir penelitian ini disusun melalui empat tahapan sesuai metode *Rapid Application Development (RAD)*:

1. Analisis kebutuhan sistem: pada tahap ini peneliti mengidentifikasi permasalahan yang terjadi di RSUD Prabumulih dalam mengelola ketersediaan tempat tidur pasien. Analisis ini dilakukan dengan melalui observasi langsung dan wawancara dengan petugas administrasi. Dari tahap ini diperoleh gambaran kebutuhan sistem seperti informasi status tempat tidur bisa dilakukan secara cepat dan dapat langsung di lihat oleh pasien, serta pembagian hak akses pengguna aplikasi yaitu petugas administrasi, perawat ruangan, dan calon pasien.
2. Membuat desain : berdasarkan dari hasil wawancara dan observasi peneliti mulai merancang antar muka aplikasi dengan menggunakan *prototype* awal aplikasi yang kemudian akan dikonsultasikan bersama dengan pengguna untuk memperoleh masukan dan masukan tersebut akan digunakan oleh peneliti sebagai dasar pembuatan dan perbaikan desain agar aplikasi benar-benar sesuai dengan kebutuhan mereka.
3. Membuat program (*coding*) : setelah desain *prototype* telah disetujui oleh pengguna maka peneliti mulai melakukan pengkodean. Pada tahap ini mulai dikembangkannya fitur-fitur utama yang dimiliki oleh aplikasi nantinya seperti pengelolaan data tempat tidur, pembaruan status data tempat tidur,serta tampilan informasi ketersediaan tempat tidur.

4. Pengujian dan implementasi sistem : yaitu proses menguji aplikasi yang telah dibuat dengan melibatkan langsung pengguna. Pengujian dilakukan untuk memastikan bahwa aplikasi berjalan sesuai fungsinya, mudah digunakan, tampilan sederhana, dan benar-benar dapat membantu proses pengelolaan ketersediaan tempat tidur. Jika saat pengujian aplikasi ditemukannya ada kekurangan maka peneliti akan melakukan perbaikan, setelah sistem yang dibangun sesuai dengan keinginan pihak RSUD Prabumulih maka sistem akan diimplementasikan di lingkungan RSUD Prabumulih untuk meningkatkan akurasi data serta mempercepat pelayanan di RSUD Prabumulih.



BAB III

METODE PENELITIAN

3.1 Objek dan Jadwal Penelitian

Objek penelitian adalah segala sesuatu yang menjadi fokus utama para peneliti dalam melakukan suatu kegiatan penelitian. Objek ini merupakan sesuatu yang dikaji untuk mendapatkan data dan informasi yang relevan guna menjawab rumusan masalah, batasan masalah, tujuan penelitian dan latar belakang.

3.1.1 Sejarah RSUD Prabumulih

Pada tahun 1947 sampai dengan 1955 berdirilah Balai Pengobatan yang merupakan cikal bakal Rumah Sakit Umum Daerah (RSUD) Prabumulih. Lokasi balai pengobatan tersebut adalah bangunan eks kantor Marga Kapak Tengah (Lokasi Lapangan tenis Dusun Prabumulih sekarang ini).

Pada tahun 1955 balai Pengobatan tersebut dikembangkan menjadi Rumah Sakit Umum Daerah Prabumulih, yang lokasinya dari tahun 1955 sampai akhir tahun 2008 di jalan AK. Gani No. 41 Lk. III kelurahan Tugu Kecil Prabumulih Timur, dengan luas tanah I adalah 5.940,56 m², luas tanah II adalah 892,50 m², luas tanah III adalah 354,51 m² dan luas tanah IV adalah 10.000 m². Jadi total seluruh luas tanah RSUD Prabumulih adalah 7.197,57 m². Sedangkan luas bangunannya adalah 1.508,4446 m². Sejak tahun 1955 sampai akhir tahun 2008 RSUD Prabumulih memiliki tempat tidur (TT) dari 51 buah menjadi 69 dan terakhir bertambah menjadi 82 buah.

Berdasarkan Keputusan Menteri Kesehatan Republik Indonesia No. 107/MENKES/SK/I/1955 tanggal 30 Januari 1955 tentang peningkatan kelas Rumah sakit Umum Daerah Prabumulih, maka status RSUD Prabumulih menjadi Rumah sakit tipe C, hal ini sudah di perdakan dengan Perda Kabupaten Muara Enim No. 31 tanggal 22 Januari 1996.

Pada tahun 2005 RSUD Prabumulih mulai melakukan pembangunan gedung baru yang berlokasi di jalan Lingkar Barat Gunung Ibul. Pembangunan ini bertujuan untuk menambah kualitas dan fasilitas yang ada di RSUD Prabumulih Pembangunan gedung baru dilakukan selama 3 tahun dan pada bulan Februari 2008, kantor RSUD mulai dipindahkan ke lokasi gedung baru.

Kemudian menyusul pelayanannya yang mulai dipindahkan ke gedung baru pada tanggal 20 Desember 2008. Gedung baru ini memiliki fasilitas yang jauh lebih baik dibanding dengan gedung yang lama, sebagai contohnya adalah area yang luas, peralatan medis yang baru, di dukung Tim IT (SIM-RS, SMS Gateway, Hotspot, Sistem antrian, CCTV) dan lain-lain. Fasilitas rawat inap yang tersedia yaitu Kelas I, Kelas II, Kelas III, VIP dan VVIP yang jumlah keseluruhannya 167 tempat tidur, tidak termasuk 8 tempat tidur ruangan neonatus. Kelas-kelas tersebut berada pada beberapa Departemen Pediatric (Anak), Departemen Medical (Penyakit Dalam), Departemen Surgical (Bedah), dan Departemen Maternitas (Kebidanan).

Pada tanggal 15 Nopember 2007, Menteri Kesehatan Republik Indonesia menetapkan bahwa RSUD Kota Prabumulih mendapat status Akreditasi “PENUH TINGKAT DASAR” dengan Nomor SK YM.01.10/III/1329/07. Yang berlaku dari tanggal 15 Nopember 2007 sampai dengan 15 Nopember 2010.

3.1.2 Visi & Misi RSUD Prabumulih

A. Visi

Visi RSUD Kota Prabumulih adalah “Menjadi Milik dan Kebanggaan Masyarakat Kota Prabumulih”

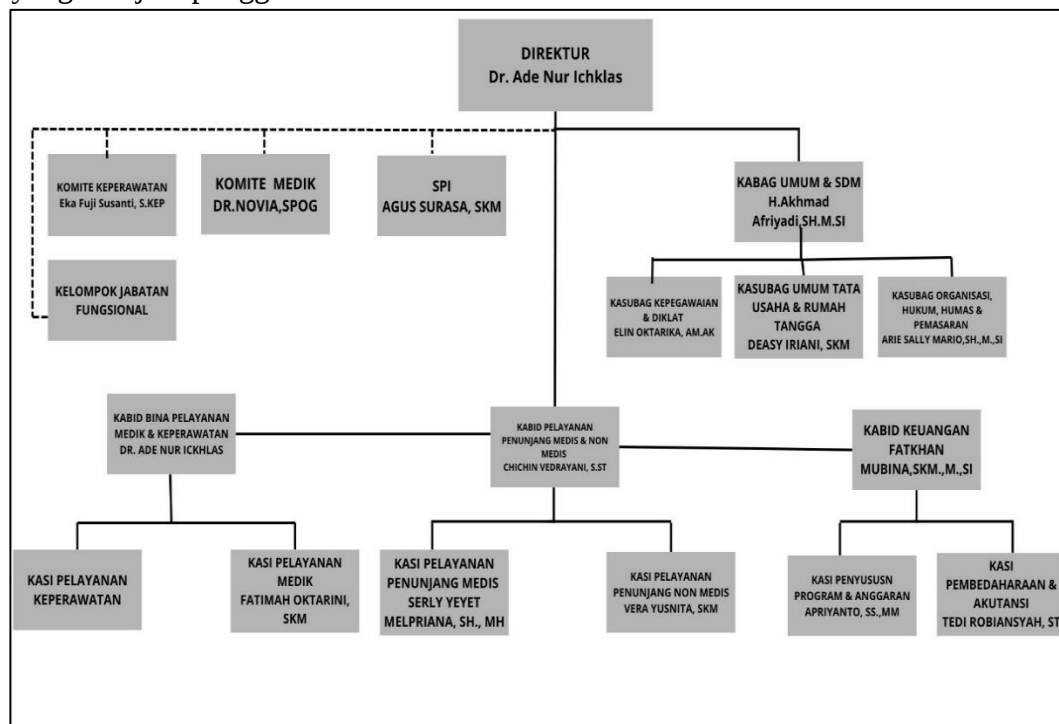
B. Misi

Misi RSUD Kota Prabumulih adalah:

- Meningkatkan pelayanan rumah sakit yang berkualitas dan terjangkau.
- Mewujudkan pegawai rumah sakit yang professional, beretika dan berakhlak.
- Pengembangan sarana prasarana dan kemitraan pelayanan rumah sakit.

3.1.3 Struktur Organisasi RSUD Prabumulih

Struktur organisasi adalah susunan penting yang menggambarkan pembagian tugas, wewenang, dan tanggung jawab dalam setiap organisasi. Dengan adanya struktur organisasi ini dapat membantu peneliti mengidentifikasi unit mana yang menjadi pengguna utama sistem.



Sumber dari RSUD Prabumulih, 2026

Gambar 3.1 Struktur Organisasi RSUD Prabumulih

Berdasarkan gambar diatas RSUD Prabumulih memiliki susunan organisasi yang terstruktur untuk mendukung kegiatan dalam RSUD Prabumulih, struktur organisasi dapat dijelaskan sebagai berikut:

1. Direktur yang berperan sebagai pemimpin, seluruh kegiatan pada RSUD Prabumulih seperti memastikan seluruh bidang bekerja sesuai dengan yang ditugaskan.
2. Komite keperawatan yang bertugas untuk mengatur, mengawasi dan meningkatkan mutu pelayanan keperawatan di RSUD Prabumulih.

3. Komite medik bertugas untuk melakukan pembinaan , pengawasan dan pengendalian mutu pelayanan medis serta menjaga profesionalisme tenaga medis.
4. SPI bertugas untuk mengawasi dan memsadikan seluruh kegiatan di RSUD Prabumulih berjalan sesuai prosedur yang berlaku.
5. Kabag umum & SDM bertugas untuk mengelola administrasi umum,kepegawaian dan sumber daya manusia di RSUD Prabumulih.
6. Kasubag tata usaha & rumah tangga bertugas untuk mengawasi kebersihan, keamanan serta pemeliharaan sarana dan prasarana RSUD Prabumulih.
7. Kasubag kepegawaian & diklat bertugas untuk menangani data pegawai, proses rekrutmen, promosi, serta penyusunan program pendidikan dan pelatihan bagi SDM RSUD Prabumulih.
8. Kasubag organisasi, hukum, humas & pemasaran bertugas untuk mengelola struktur organisasi urusan hukum, hubungan masyarakat, serta promosi dan kerja sama pihak RSUD Prabumulih dengan pihak lain.
9. Kabid bina pelayanan medik & keperawatan bertugas untuk mengkoordinasikan seluruh pelayanan medis dan keperawatan,seperti mengembangkan mutu layanan kesehatan, serta memastikan standar pelayanan medis dijalankan.
10. Kasi pelayanan keperawatan bertugas untuk mengatur dan mengawasi kegiatan pelayanan keperawatan agar sesuai dengan standar asuhan.
11. Kasi pelayanan medik bertugas untuk mengatur operasional pelayanan medis, jadwal dokter dan pelaksanaan pelayanan rawat jalan maupun rawat inap.
12. Kabag umum & SDM bertugas untuk mengelola administrasi umum,kepegawaian dan sumber daya manusia di RSUD Prabumulih.
13. Kasubag tata usaha & rumah tangga bertugas untuk mengawasi kebersihan, keamanan serta pemeliharaan sarana dan prasarana RSUD Prabumulih.
14. Kasubag kepegawaian & diklat bertugas untuk menangani data pegawai, proses rekrutmen, promosi, serta penyusunan program pendidikan dan pelatihan bagi SDM RSUD Prabumulih.

15. Kabag umum & SDM bertugas untuk mengelola administrasi umum,kepegawaian dan sumber daya manusia di RSUD Prabumulih.
16. Kasubag tata usaha & rumah tangga bertugas untuk mengawasi kebersihan, keamanan serta pemeliharaan sarana dan prasarana RSUD Prabumulih.
17. Kasubag kepegawaian & diklat bertugas untuk menangani data pegawai, proses rekrutmen, promosi, serta penyusunan program pendidikan dan pelatihan bagi SDM RSUD Prabumulih.
18. Kasubag organisasi, hukum, humas & pemasaran bertugas untuk mengelola struktur organisasi urusan hukum, hubungan masyarakat, serta promosi dan kerja sama pihak RSUD Prabumulih dengan pihak lain.
19. Kabid bina pelayanan medik & keperawatan bertugas untuk mengkoordinasikan seluruh pelayanan medis dan keperawatan,seperti mengembangkan mutu layanan kesehatan, serta memastikan standar pelayanan medis dijalankan.
20. Kasi pelayanan keperawatan bertugas untuk mengatur dan mengawasi kegiatan pelayanan keperawatan agar sesuai dengan standar asuhan.
21. Kasi pelayanan medik bertugas untuk mengatur operasional pelayanan medis, jadwal dokter dan pelaksanaan pelayanan rawat jalan maupun rawat inap.
22. Kabid pelayanan penunjang medis & non medis bertugas untuk mengelola layanan penunjang baik di bagian medis (laboratorium, dll) maupun bagian non medis (kebersihan, keamanan, dll).
23. Kasi pelayanan penunjang non medis bertugas sebagai pertanggung jawab atas kegiatan laboratorium,radiologi, farmasi, dan sarana medis pendukung pelayanan.
24. Kasi pelayanan penunjang medis bertugas sebagai mengelola dan mengawasi unit-unit penunjang pelayanan medis agar mendukung pelayanan pasien secara optimal.
25. Kasi bendahara & akuntansi bertugas untuk mengelola kas, pembukuan dan laporan keuangan, memastikan seluruh transaksi keuangan sesuai ketentuan dan dapat dipertanggungjawabkan.

26. Kasi penyusunan program dan anggaran bertugas untuk menyusun rencana kerja dan anggaran rumah sakit, serta memantau dan mengevaluasi pelaksanaannya.
27. Kelompok jabatan fungsional bertugas untuk melaksanakan kegiatan teknis sesuai bidang keahlian masing-masing (dokter, perawat, apoteker, analis, dan lain-lain) untuk mendukung pelayanan rumah sakit.
28. Kabid keuangan yang bertugas untuk mengelola seluruh aspek keuangan agar operasional rumah sakit berjalan efektif, transparan, dan akuntabel.

3.1.4 Jadwal Penelitian

Tabel 3.1 Jadwal Penelitian

No	Kegiatan Penelitian	Nov	Des	Jan	Feb	Mar	Apr	Mei	Jun
1	Penyusunan Proposal								
2	Studi Literatur								
3	Pengumpulan Data								
4	Analisis Kebutuhan Sistem (Requirement Planning)								
5	Perancangan Sistem (User Design)								
6	Pengembangan Aplikasi (Construction)								
7	Pengujian Aplikasi (Implementation)								
8	Penyusunan Skripsi								
9	Sidang Skripsi								
10	Revisi Skripsi & Sistem								

Jadwal penelitian dilaksanakan selama 7 bulan terhitung dari tanggal 13 November 2025 sampai dengan 13 Mei 2026. Dimulai dari penyusunan skripsi yang dilakukan dibulan november sampai desember dan diakhiri dengan penyusunan dan sidang skripsi di bulan januari sampai juni .

3.2 Metode Penelitian

Metode penelitian adalah suatu cara atau disebut juga metode ilmiah untuk mengumpulkan informasi mengenai pokok kajian. Dengan tujuan akhir untuk menyelesaikan masalah yang dihadapi. Metode yang dipakai dalam penelitian kali ini adalah metode deskriptif dengan pendekatan kualitatif.

Deskriptif secara karakteristik bermaksud membantu para peneliti untuk menggambarkan atau mempertajam penjelasan penelitian mereka agar nantinya dapat mempermudah dipahami oleh orang lain yang ingin mengetahui penelitian mereka (Manurung, 2021).

Penelitian kualitatif merupakan suatu teknik penelitian yang menggunakan narasi atau kata-kata dalam menjelaskan dan menjabarkan makna dari setiap fenomena, gejala, dan situasi sosial tertentu. Dalam penelitian kualitatif, peneliti adalah instrumen kunci untuk memaknai dan menginterpretasikan setiap fenomena, gejala dan situasi sosial tertentu. Karena itu peneliti perlu menguasai teori untuk menganalisis kesenjangan yang terjadi antara konsep teoritis dengan fakta yang terjadi (Waruwu, 2023).

Metode deksriptif pendekatan kualitatif dalam penelitian ini adalah metode yang digunakan untuk menggambarkan, memahami dan menganalisis fenomena yang terjadi secara mendalam berdasarkan data yang diperoleh dari lapangan. Berdasarkan metode ini peneliti berusaha memahami bagaimana proses pengecekan ketersediaan tempat tidur dilakukan, apa saja kendala yang dialami.

3.3 Sumber Data

1. Data primer

Data primer adalah data yang diperoleh secara langsung melalui interaksi antara peneliti dengan subjek penelitian. Dalam penelitian ini, data primer didapatkan melalui wawancara langsung dengan ibu Dian Anugrahni sebagai staf pada bagian administrasi pada RSUD Prabumulih untuk mendapatkan informasi sesuai dengan kebutuhan penelitian.

2. Data sekunder

Data sekunder adalah sumber data penelitian yang diperoleh secara tidak langsung melalui media perantara. Artinya, data ini tidak dikumpulkan langsung oleh peneliti melainkan dari sumber yang telah ada sebelumnya, seperti dokumen, literatur, atau data yang di kumpulkan oleh pihak lain. Contoh sumber data sekunder meliputi buku, jurnal akademis, artikel.

3.4 Teknik Pengumpulan Data

Teknik pengumpulan data adalah suatu hal yang penting dalam sebuah penelitian. pengumpulan data dalam penelitian dimaksudkan untuk memperoleh bahan-bahan, keterangan, kenyataan-kenyataan dan informasi yang dapat dipercaya. Teknik yang digunakan untuk mengumpulkan data adalah sebagai berikut:

1. Observasi

Observasi adalah mengamati secara langsung sistem yang sedang berjalan di RSUD Prabumulih terkait ketersediaan tempat tidur pasien.

2. Wawancara

Wawancara adalah melakukan tanya jawab dengan ibu Dian Anugrahni sebagai staf bagian administrasi pada RSUD Prabumulih untuk memperoleh informasi yang lebih dalam tentang data ketersediaan tempat tidur pasien di RSUD Prabumulih.

3. Dokumentasi

Dokumentasi adalah mengumpulkan dokumen-dokumen seperti struktur organisasi, visi & misi, dan lainnya sebagai bahan pendukung.

4. Studi Literatur

Studi literatur adalah kegiatan mengumpulkan dan mempelajari berbagai referensi yang berkaitan dengan topik penelitian dan sistem yang akan

dikembangkan. Referensi tersebut dapat berupa buku, jurnal artikel ilmiah maupun sumber informasi lain.

3.5 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam pengembangan aplikasi ini adalah metode *Rapid Application Development (RAD)*. *Rapid Application Development (RAD)* dipilih karena pendekatannya yang interatif dan fokus kepada kolaborasi antara pengembang dan pengguna dalam membuat desain. Dalam penelitian ini pengguna yang terlibat yaitu admin administrasi dan petugas/perawat dalam setiap tahap pengembangan mulai dari perancangan hingga pengujian aplikasi. Dengan demikian, aplikasi yang dihasilkan dapat lebih sesuai dengan kebutuhan pengguna dan lebih cepat dalam penyesuaian terhadap perubahan.



Sumber : Sondang, 2024

Gambar 3.2 Metode *Rapid Aplikasi Development*

Adapun tahapan dalam metode *Rapid Application Development (RAD)* sebagai berikut :

1. Perencanaan Kebutuhan

Pada tahap ini dilakukannya identifikasi kebutuhan sistem bersama pengguna untuk menentukan sistem seperti apa yang akan dikembangkan atau dirancang nanti agar sesuai dengan kebutuhan pengguna.

2. Desain Sistem

Pada tahap ini pengguna ikut terlibat dalam membuat *prototype* rancangan antarmuka sistem dan memberikan masukan langsung sehingga desain lebih sesuai dengan kebutuhan pengguna

3. Pengembangan

Tahap ini adalah tahap memulai membuat sistem yang sudah direncanakan. Memulai menyusun suatu kode program atau biasa disebut *coding*, untuk merubah desain sistem yang telah dibuat menjadi sebuah aplikasi yang telah direncanakan agar dapat digunakan.

4. Implementasi

Pada tahap ini sistem mulai diimplementasikan atau diuji langsung ke lingkungan RSUD Prabumulih. Pengguna akan di ajarkan gimana untuk menggunakan aplikasi tersebut.serta evaluasi apakah sistem berjalan sesuai dengan kebutuhan pengguna.

3.6 Alat Bantu Perancangan

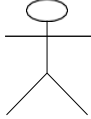
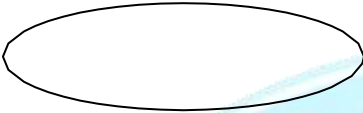
Alat bantu perancangan digunakan untuk membantu peneliti dalam membuat gambaran awal tentang bagaimana aplikasi akan bekerja sebelum dimulainya tahap pengkodean. Beberapa alat bantu yang digunakan ialah *UML (Unifed Modified Languange)* seperti *Use Case Diagram* yang berfungsi untuk menggambarkan peran dan aktivitas dari pengguna, dan *Activity Diagram* yang berfungsi untuk menjelaskan alur proses sebuah aplikasi yang akan dikembangkan.

UML (Unified Modelling Languange) adalah suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorinteasi objek. *UML* juga didefinisikan sebagai suatu bahasa standar visualisasi, perancangan, dan pendokumentasian sistem, atau dikenal juga sebagai bahasa standar penulisa *blueprint* sebuah *software*. *UML* diharapkan mampu mempermudah pengembangan perangkat lunak serta memnuhi semua kebutuhan pengguna dengan efektif, lengkap dan tepat.(Suharni, 2023).

1. Use Case Diagram

Use case diagram merupakan visualisasi dari beberapa komponen, seperti *actor*, *use case*, dan relasi antar komponen Melalui *use case diagram*, dapat membantu analis dalam penyusunan kebutuhan (*requirement*) pengembangan sistem. *Use case diagram* dipakai untuk menjelaskan perancangan sistem kepada userdan melakukan perancangan semua fitur yang ada pada sistem yang akan dibangun (Narulita et al., 2024)

Tabel 3.2 *Use Case Diagram*

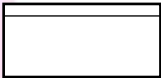
Simbol	Deskripsi
Aktor/Actor 	Digunakan untuk menjelaskan sesuatu atas seseorang yang sedang berinteraksi dengan sistem.
	Menggambarkan suatu perilaku dari sistem tanpa mengungkapkan struktur internal dari sistem tersebut.
UseCase	
association 	Jalur komunikasi antar actor dengan use case yang saling berpartisipasi.
Ekstensi/extend <<extend>> 	Penambahan perilaku ke dalam use case dasar yang tidak tahu tentang hal tersebut..
generalization 	Hubungan antara use case umum dengan use case yang lebih spesifik, yang mewarisi dan menambah fitur terhadapnya.
Include <<include>> 	Penambahan perilaku ke dalam use case dasar yang secara eksplisit menjelaskan penambahannya.

Sumber : suharni,2023

2. Activity Diagram

Activity diagram merupakan salah satu yang paling banyak digunakan di antara semua diagram *UML*. Alur *activity diagram* dapat digunakan untuk mengontrol urutan tindakan yang dieksekusi atau untuk merepresentasikan komunikasi data. Biasanya diagram ini digunakan untuk desain tingkat tinggi seperti memodelkan proses bisnis, dan desain tingkat rendah seperti untuk mendeskripsikan algoritma yang akan diimplementasikan (Wayahdi, 2023).

Tabel 3.3 Activity Diagram

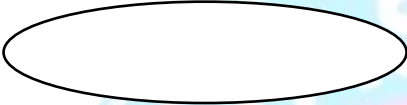
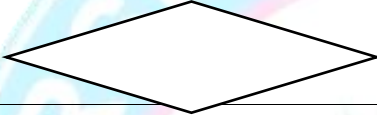
Simbol	Deskripsi
Status awal/ <i>Initial</i> 	Menunjukkan di mana aliran kerja dimulai.
<i>Actions</i> 	Langkah-langkah dalam sebuah activity.
<i>Decision</i> 	Menunjukkan di mana keputusan akan dibuat.
<i>Swimlane</i> 	Mengelompokkan activity berdasarkan actor.
Status akhir/ <i>Final</i> 	Menunjukkan di mana aliran kerja berakhir.

Sumber : Suharni, 2023

3. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah diagram berbentuk notasi grafis yang berada dalam pembuatan *database* yang menghubungkan antara data satu dengan yang lain. Fungsi *ERD* adalah sebagai alat bantu dalam pembuatan *database* dan memberikan gambaran bagaimana kerja *database* yang akan dibuat. Di dalam *ERD* terdapat 3 elemen dasar, yaitu entitas, atribut, dan relasi (Putri & Khairunnisa, 2025)

Tabel 3.4 *Entity Relationship Diagram (ERD)*

Simbol	Deskripsi
Entitas 	Sesuatu di dalam sistem atau di luar sistem yang saling berinteraksi.
Atribut 	Sifat, atau elemen dari setiap entitas.
Relationship 	Hubungan antara entitas1 dengan entitas lainnya.
Link 	Menghubungkan satu entitas dengan entitas lainnya.

Sumber : Ridho Satrianle Wicaksono, 2022

3.6.1 Alat Bantu Utama

Alat bantu utama ialah perangkat atau aplikasi yang digunakan oleh peneliti untuk mendukung proses pengembangan sistem. Dalam penelitian ini peneliti menggunakan alat bantu utama yaitu *Visual Studio Code* dengan *framework Flutter* dan *Dart* sebagai bahasa pemrogramannya.

Visual Studio Code dipilih karena mudah digunakan, memiliki tampilan yang sederhana serta aplikasi ini juga mendukung banyak bahasa pemrograman dan ekstensi yang dapat membantu dalam proses pengembangan aplikasi manajemen ketersediaan tempat tidur pasien.

Flutter digunakan sebagai *framework* untuk membangun tampilan aplikasi sedangkan *Dart* digunakan sebagai bahasa pemrograman utama yang digunakan peneliti untuk membuat logika dan fungsi serta alat kerja dalam aplikasi. Selain itu fitur *hot reload* pada *Flutter* memungkinkan peneliti melihat perubahan secara

langsung tanpa harus menjalankan ulang aplikasi dari awal. Hal ini membantu peneliti dalam proses pembuatan aplikasi berjalan dengan cepat.

3.6.2 Alat Bantu Desain

Alat bantu desain adalah perangkat atau aplikasi yang digunakan oleh peneliti untuk membuat gambaran awal mengenai tampilan antarmuka sistem sebelum proses pengkodean dilakukan. Proses ini dapat membantu peneliti dalam merancang tampilan, menyusun struktur halaman serta menentukan bagaimana alur interaksi dalam aplikasi. Dengan adanya rancangan tampilan antarmuka yang telah dibuat terlebih dahulu dapat membuat proses pengkodean menjadi lebih terarah karena peneliti sudah memiliki rancangan aplikasi yang jelas.

Dalam penelitian ini, peneliti menggunakan Figma sebagai alat bantu desain. Figma adalah salah satu *design tools* yang biasanya digunakan untuk membuat tampilan aplikasi *mobile*, *desktop*, *website* dan lain-lain. Figma bisa digunakan di sistem operasi *windows*, *linux* ataupun *mac* dengan terhubung ke internet. (Al-Faruq, 2022). Figma menyediakan fitur seperti pembuatan *layout*, pengaturan warna, penempatan *icon*, serta pembuat *prototype* interaktif agar peneliti memahami alur penggunaan aplikasi. Dengan dukungan Figma proses perancangan antarmuka menjadi lebih terstruktur, rapi dan mempermudah peneliti pada tahap pengkodean.

3.6.3 Alat Bantu *Debugging* dan Pengembangan

Alat bantu *debugging* dan pengembangan adalah salah satu tahapan yang dilakukan oleh peneliti untuk memeriksa apakah sistem berjalan sesuai yang diinginkan. Dalam penelitian ini peneliti menggunakan fitur bawaan dari *Visual Studio Code* untuk menjalankan aplikasi. Aplikasi ini akan dijalankan yang dimana hasil programnya akan dapat dilihat langsung pada perangkat *mobile* atau emulator. Dengan begitu peneliti dapat melihat langsung bagaimana aplikasi bekerja, serta dapat memeriksa apakah ada bagian kode yang bermasalah agar dapat langsung segera dilakukannya perbaikan pada kode program.

3.7 Pengujian Sistem

Pengujian sistem merupakan proses untuk menjalankan aplikasi yang dibuat untuk memastikan apakah aplikasinya berjalan sesuai dengan kebutuhan dari pengguna yang telah ditentukan. Pengujian ini dilakukan untuk melihat apakah aplikasi manajemen ketersediaan tempat tidur pasien dapat bekerja dengan baik atau tidak. Dalam penelitian ini, pengujian sistem dilakukan menggunakan *White Box Testing*.

White-box testing (juga dikenal sebagai *clear box testing*, *glass box testing*, *transparent box testing* dan *structural testing*) adalah metode pengujian *software* yang menguji struktur internal atau cara kerja suatu aplikasi, yang bertentangan dengan fungsinya (yaitu *black-box testing*). Dalam *white-box testing*, perspektif internal sistem, serta keterampilan pemrograman, digunakan untuk merancang kasus pengujian (Wintana & Pribadi, 2022).

White-box testing menguji lebih dalam dari *Black-box testing* jadi mengapa peneliti memilih metode ini untuk pengujian sistem. Karena ini menguji lebih dalam dan hasilnya dapat memeriksa lebih kompleks dengan pemrograman dan proses yang berjalan. Menguji lebih dalam juga berarti bisa lebih memeriksa seberapa efisien pemrograman itu sendiri dan memeriksa kesalahan dari dalam. Walaupun antarmuka pengguna tidak menunjukkan kesalahan.

BAB IV

ANALISA DAN PERANCANGAN

4.1 Analisa Sistem

4.1.1 Analisa dan Evaluasi Sistem Berjalan

A. Analisis Sistem Berjalan

Pada sistem yang berjalan saat ini, proses pengelolaan ketersediaan tempat tidur pasien masih dilakukan secara manual oleh pihak rumah sakit yaitu informasi terkait tempat tidur ada berapa kosong, terisi, atau dalam proses pembersihan dan informasi tersebut masih diperoleh melalui komunikasi telepon antara petugas administrasi dan perawat ruangan. Sehingga pasien yang ingin mengetahui ketersediaan kamar harus datang langsung ke rumah sakit dan menanyakan langsung kepada petugas dan hal ini membutuhkan waktu yang cukup lama. Selain itu, pembaruan status tempat tidur sering mengalami keterlambatan karena laporan terkait data tempat tidur masih bergantung dengan telepon ke perawat ruangan. Kondisi ini dapat menyebabkan informasi yang diterima oleh pasien maupun administrasi tidak sesuai dengan kondisi yang sebenarnya di lapangan. Oleh karena itu, sistem yang berjalan saat ini belum mampu memberikan informasi ketersediaan tempat tidur secara cepat dan akurat. Diperlukan suatu sistem informasi yang dapat mengintegrasikan pengelolaan data tempat tidur sehingga setiap perubahan status dapat diperbarui secara langsung dan informasi mengenai ketersediaan tempat tidur dapat ditampilkan secara lebih akurat. Dengan adanya sistem tersebut, petugas administrasi dapat memperoleh informasi tanpa harus selalu menghubungi perawat melalui telepon, sedangkan pasien dan keluarga pasien dapat mengetahui ketersediaan kamar dengan lebih mudah tanpa harus datang langsung ke rumah sakit. Penerapan sistem informasi ini diharapkan dapat meningkatkan efektivitas dan kinerja petugas rumah sakit dan informasi yang didapatkan bisa lebih cepat dan akurat.

B. Evaluasi Sistem Berjalan

Berdasarkan hasil dari analisis terhadap sistem yang sedang berjalan, dapat disimpulkan bahwa metode pengelolaan ketersediaan tempat tidur yang masih menggunakan telepon antara petugas administrasi dan perawat ruangan belum mampu memenuhi kebutuhan pelayanan yang cepat dan efisien. Sistem tersebut menyebabkan keterlambatan dalam penyampaian informasi baik kepada pasien maupun pihak administrasi dan membuat pasien harus menunggu lama untuk mendapatkan informasi terkait ketersediaan tempat tidur yang ada. Untuk mengatasi masalah tersebut maka dibutuhkan sebuah sistem yang mampu menampilkan informasi secara cepat dan akurat dan dapat diakses oleh petugas administrasi, perawat ruangan dan pasien. Sistem yang diusulkan memungkinkan pasien dapat melihat secara langsung mengenai ketersediaan tempat tidur yang ada dan pasien juga dapat melakukan komunikasi dengan pihak admin terkait informasi lebih lanjut. Sementara pembaruan status tempat tidur seperti berisi, kosong atau dalam perawatan dapat dilakukan oleh perawat ruangan agar data yang diperoleh sesuai dengan yang sebenarnya, dan petugas administrasi dapat mengelola data tempat tidur dan akun perawat ruangan. Dengan adanya sistem ini diharapkan proses pelayanan menjadi lebih efektif, cepat dan akurat.

4.1.2 Analisa Kebutuhan

A. Kebutuhan Pengguna

Pengguna pada aplikasi manajemen ketersediaan tempat tidur ini terdiri dari 3 aktor yang memiliki perannya masing masing. Adapun pengguna sistem pada aplikasi ini antara lain:

1. Petugas Administrasi

Petugas Administrasi ialah staff administrasi yang bertugas untuk mengelola data tempat tidur seperti menambah dan menghapus tempat tidur serta dapat mengelola akun perawat seperti membuat akun perawat dan menghapus akun perawat.

2. Perawat Ruangan

Perawat ruangan staff ruangan yang bertugas untuk memperbarui status tempat tidur, seperti kosong, terisi atau dalam perawatan sesuai dengan kondisi nyata di lapangan.

3. Pasien

Pasien bertugas untuk melihat informasi ketersediaan tempat tidur yang tersedia di RSUD Prabumulih, dan juga melakukan komunikasi *via chat* dengan pihak admin untuk mendapatkan informasi lebih lanjut.

B. Kebutuhan Data

Kebutuhan data pada sistem ini digunakan untuk mendukung proses pengelolaan informasi dan pemantauan ketersediaan tempat tidur. Data yang dibutuhkan dalam sistem ini disesuaikan dengan fungsi dan hak akses masing-masing pengguna. Adapun data yang diperlukan dalam sistem adalah sebagai berikut :

1. Data Pengguna

Data pengguna ialah data yang digunakan untuk mengelola akun pengguna dalam sistem Aplikasi Manajemen Ketersediaan Tempat Tidur. Data ini meliputi nama pengguna (*username*) dan *password* serta peran setiap pengguna dalam sistem, seperti staff administrasi yang mengelola data tempat tidur dan ruangan, perawat ruangan yang memperbarui status tempat tidur serta pasien tidak melewati sistem login karena dalam sistem pasien hanya menggunakan Aplikasi sebagai pusat informasi.

2. Data Ruangan

Data ruangan berisi informasi mengenai ruangan rawat inap di rumah sakit seperti nama ruangan, kelas perawatan dan jumlah tempat tidur.

Data ini digunakan untuk mengelompokkan tempat tidur berdasarkan ruangan dan kelas

C. Kebutuhan Fungsional

Kebutuhan fungsional adalah fungsi-fungsi yang harus ada dalam sistem agar aplikasinya dapat berjalan dengan baik. Adapun kebutuhan fungsionalitas dalam sistem ini adalah sebagai berikut :

1. Sistem dapat melakukan proses login dengan cara memasukkan *username* dan *password* pengguna sesuai dengan perannya masing-masing dalam sistem.
2. Sistem dapat melakukan proses *logout* untuk mengakhiri sistem.
3. Sistem dapat menampilkan data tempat tidur berdasarkan ruangan, kelas perawatan masing-masing.
4. Sistem dapat menampilkan status tempat tidur seperti kosong, terisi, dan dalam perawatan.
5. Sistem dapat mengelola tempat tidur seperti menambah tempat tidur atau menghapus tempat tidur.
6. Sistem dapat memperbarui status tempat tidur seperti kosong, terisi, sedang dalam perawatan sesuai dengan kondisi di lapangan.
7. Sistem dapat menyimpan setiap perubahan data tempat tidur yang telah dilakukan oleh perawat ruangan ke dalam basis data.
8. Sistem dapat mengelola akun perawat ruangan yang telah terdaftar dengan menambah dan menghapus akun perawat ruangan jika ada perawat ruangan yang telah *resign* dari RSUD Prabumulih.
9. Sistem dapat menampilkan daftar akun dari perawat ruangan yang telah terdaftar pada sistem pada dashboard staff administrasi.
10. Sistem dapat melakukan komunikasi yang bisa dilakukan oleh pasien ke admin untuk mengetahui informasi lebih lanjut.
11. Sistem dapat melakukan ubah *password* yang bisa dilakukan oleh pengguna.

D. Kebutuhan *Non-Fungsional*

Kebutuhan *non-fungsional* adalah kebutuhan yang berkaitan dengan kualitas yang dimiliki oleh sistem yang berpengaruh terhadap kenyamanan, keamanan dan kinerja aplikasi. Adapun kebutuhan *non-fungsional* yang dimiliki oleh sistem ini adalah :

1. Keamanan Data Pengguna

Sistem harus mampu menjaga keamanan data pengguna agar tidak dapat diakses oleh pihak yang tidak berwenang. Setiap pengguna diwajibkan melakukan proses login sebelum mengakses sistem, sehingga identitas pengguna dapat dikenali dengan baik. Selain itu, sistem harus membatasi hak akses pengguna sesuai dengan peran masing-masing, seperti pasien, petugas administrasi, dan perawat ruangan. Dengan adanya pengaturan hak akses ini, pengguna hanya dapat mengakses fitur dan data yang sesuai dengan tugas dan tanggung jawabnya, sehingga kerahasiaan dan keamanan data dapat terjaga.

2. Tampilan sederhana

Sistem harus memiliki tampilan antarmuka yang sederhana, rapi, dan mudah dipahami oleh pengguna. Desain tampilan sistem harus disesuaikan dengan kebutuhan pengguna, baik petugas administrasi, perawat ruangan, maupun pasien, sehingga sistem dapat digunakan tanpa memerlukan pelatihan yang rumit.

3. Akses Sistem Yang Cepat

Sistem harus mampu menampilkan informasi ketersediaan tempat tidur dengan cepat dan responsif. Kecepatan akses ini sangat penting agar pengguna tidak perlu menunggu lama saat membuka halaman sistem atau melihat data yang dibutuhkan. Dengan waktu respon yang cepat, sistem diharapkan dapat mendukung pelayanan rumah sakit secara efektif,

khususnya dalam menyampaikan informasi ketersediaan tempat tidur kepada pasien dan petugas terkait.

4. Keandalan Sistem

Sistem harus dapat berjalan secara stabil dan konsisten selama digunakan. Sistem diharapkan mampu meminimalkan terjadinya kesalahan, baik kesalahan sistem maupun kesalahan data, sehingga informasi yang ditampilkan tetap akurat. Keandalan sistem ini penting agar pengguna dapat mempercayai sistem sebagai sumber informasi utama dalam pengelolaan ketersediaan tempat tidur pasien.

5. Ketersediaan Sistem

Sistem harus bisa diakses kapan saja selama jaringan dan *server* berada dalam kondisi normal. Hal ini bertujuan agar pengguna dapat memperoleh informasi ketersediaan tempat tidur setiap saat tanpa terbatas oleh waktu. Dengan tingkat ketersediaan yang baik, sistem dapat mendukung proses pelayanan rumah sakit secara berkelanjutan.

E. Analisa Perangkat Keras & Lunak (*Hardware/Software*)

Analisa perangkat keras dan lunak apa saja yang digunakan untuk membantu dalam membangun aplikasi, menguji aplikasi ini.

1. Analisa Perangkat Keras (*Hardware*)

Hardware dalam bahasa Indonesia disebut juga dengan nama perangkat keras merupakan salah satu komponen dari sebuah komputer yang sifat alatnya dapat dilihat dan diraba secara langsung dengan kasat mata atau yang berbentuk nyata, yang berfungsi untuk mendukung proses komputerisasi. *Hardware* juga dikatakan bagian fisik komputer yang merupakan penghubung antara pengguna atau user dengan sistem yang terdapat pada

komponen komputer. (Trisna et al., 2022) Spesifik Perangkat Keras yang digunakan dalam pembuatan sistem ini adalah sebagai berikut ;

- A. *Laptop Acer Aspire 3 A314-33* dengan *RAM 4 GB*, digunakan sebagai alat bantu dalam merancang, mengembangkan sebuah aplikasi.
- B. *Smartphone Android vivo Y35* versi 14 dengan *RAM 8 GB*, digunakan untuk menjalankan aplikasi.

2. Analisa Perangkat Lunak (Software)

Software dalam bahasa Indonesia disebut juga dengan nama perangkat lunak merupakan sekumpulan instruksi atau perintah yang memungkinkan hardware atau perangkat keras untuk dapat memproses data. Perangkat lunak ini tidak berbentuk fisik, melainkan berupa program yang diciptakan melalui perangkat elektronik yang dikenal sistem operasi dan program aplikasi (Trisna et al., 2022). Spesifik dari Perangkat lunak yang digunakan pada pembuatan sistem ini adalah sebagai berikut :

- 1) Sistem Operasi *Windows 11*
- 2) *Figma version 126.4.11* sebagai alat bantu dalam pembuatan rancangan aplikasi.
- 3) *Visual Studio Code version 1.124.0*
- 4) *Framework Flutter version 3.24.4*
- 5) Bahasa Pemrograman *Dart version 3.5.4*
- 6) *Firebase* sebagai *Database*
- 7) Koneksi internet

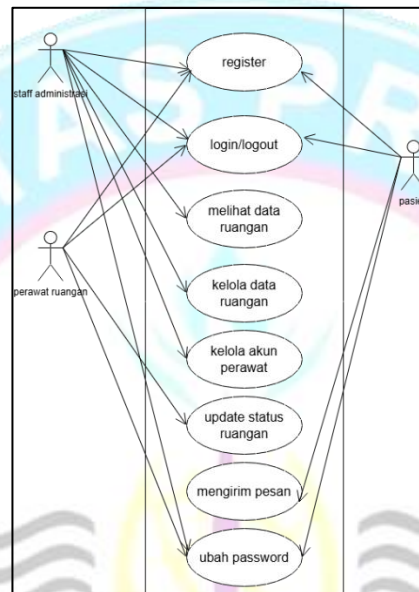
4.2 Perancangan Sistem Desain yang diusulkan

4.2.1 Perancangan Sistem (logis)

Perancangan sistem secara logis adalah tahapan dalam pengembangan sistem yang berfokus pada perencanaan cara kerja sistem secara keseluruhan tanpa membahas teknologi atau perangkat yang digunakan. Pada tahap ini, sistem dirancang berdasarkan kebutuhan pengguna dan tujuan sistem, sehingga dapat

menggambarkan alur proses, hubungan antar data, serta interaksi antara pengguna dan sistem secara jelas dan terstruktur. Perancangan sistem secara logis ini digambarkan melalui model-model perancangan seperti *Use Case Diagram*, *Activity Diagram* serta *Entity Relationship Diagram (ERD)* sehingga sebuah sistem yang akan di buat bisa tergambarkan dengan jelas.

1. Perancangan *Use Case Diagram*



Sumber: data diolah oleh penulis (2026)

Gambar 4.1 *Use Case Diagram* yang diusulkan

Use Case Diagram menggambarkan interaksi antara aktor dengan sistem Aplikasi Manajemen Ketersediaan Tempat Tidur Pasien. Sistem ini memiliki tiga aktor utama yaitu:

1. Administrasi

Administrasi dapat melakukan *login* dan *logout*, mengelola akun pengguna seperti membuat akun pengguna dan menghapus akun pengguna, selain bisa

melihat ketersediaan tempat tidur petugas administrasi juga bisa mengelola data tempat tidur tersebut seperti menambah dan menghapus tempat tidur. selain itu, administrasi dapat melakukan ubah password.

2. Perawat Ruangan

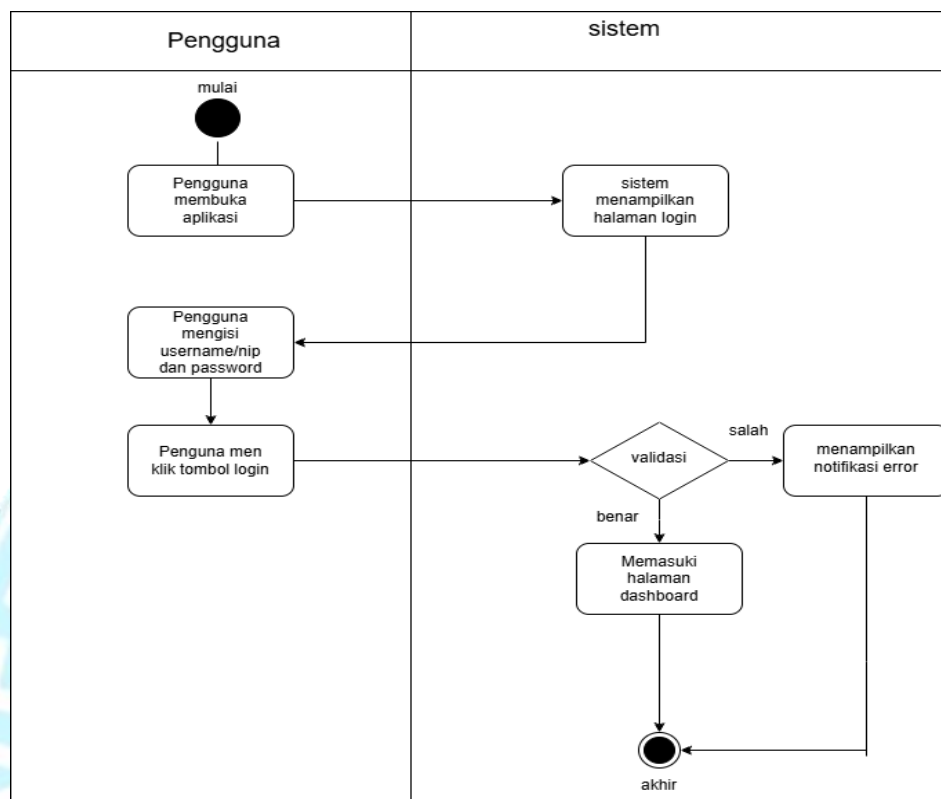
Perawat dapat melakukan login dan logout pada sistem, perawat ruangan Universitas Prabumulih melihat informasi ketersediaan tempat tidur yang terdapat pada ruangan yang menjadi tanggung jawabnya. Selain itu, perawat dapat memperbarui status data tempat tidur sesuai dengan kondisi ruangan yang ada seperti kosong, terisi, dan dalam perbaikan. Setiap perubahan status yang dirubah bisa dilihat langsung oleh petugas administrasi dan juga oleh pasien dengan adanya fitur tersebut, perawat tidak perlu lagi menyampaikan informasi dengan menelpon petugas administrasi terlebih dahulu lagi sehingga penyampaian informasi dapat dilakukan secara cepat dan akurat. dan perawat ruangan dapat melakukan perubahan password.

3. Pasien

Pasien berperan sebagai pengguna yang membutuhkan informasi mengenai ketersediaan tempat tidur atau kamar di rumah sakit. Pasien dapat melakukan login dan logout pada sistem untuk mengakses informasi yang ada. Setelah masuk pasien dapat melihat informasi mengenai ketersediaan tempat tidur yang sesuai dengan yang terjadi pada kondisi di dunia nyata. Pasien juga dapat melihat informasi ketersediaan tempat tidur berdasarkan ruangan yang dicari dan pasien juga dapat mengetahui informasi secara lebih lanjut mengenai harga atau informasi terkait ruangan yang dituju apakah fasilitasnya memenuhi standar ruangan yang dicari oleh pasien atau tidak dan apakah harga yang ada pada ruangan tersebut sesuai dengan harga yang diinginkan oleh pasien atau tidak. Dan yang terakhir pasien juga dapat melakukan perubahan password pada akun yang digunakan untuk menjaga keamanan informasi dan data pribadi pasien.

2. Perancangan Activity Diagram

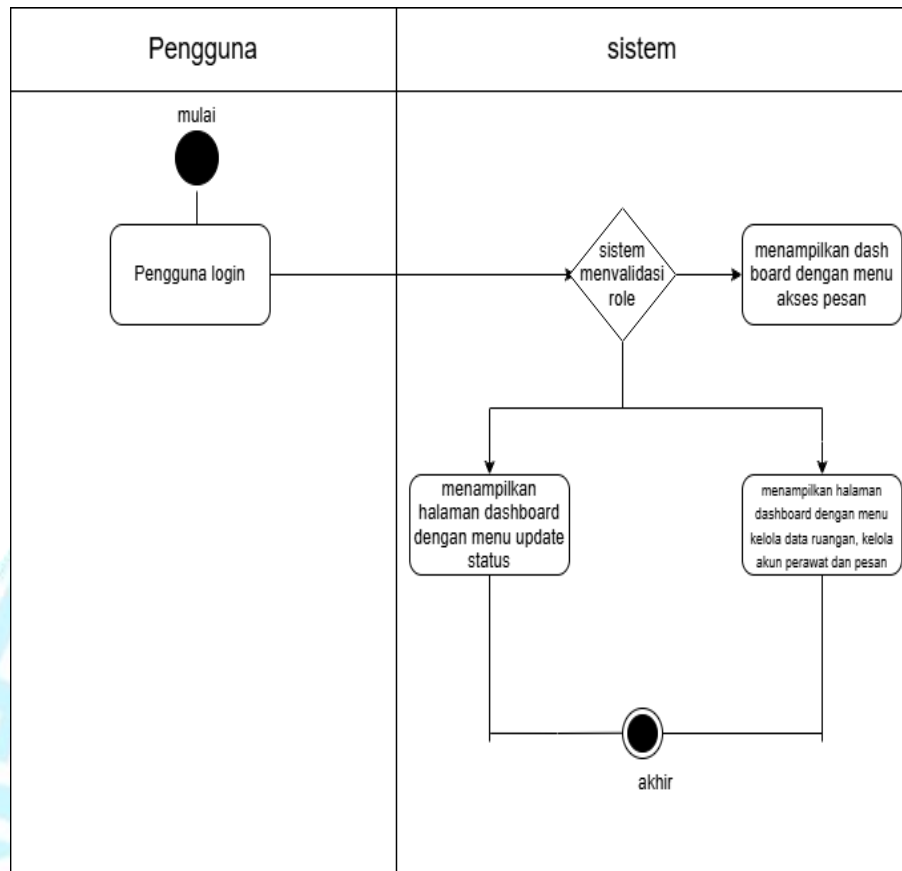
A. Activity Diagram Login



Sumber: data diolah oleh penulis (2026)

Gambar 4.2 Activity Diagram Login

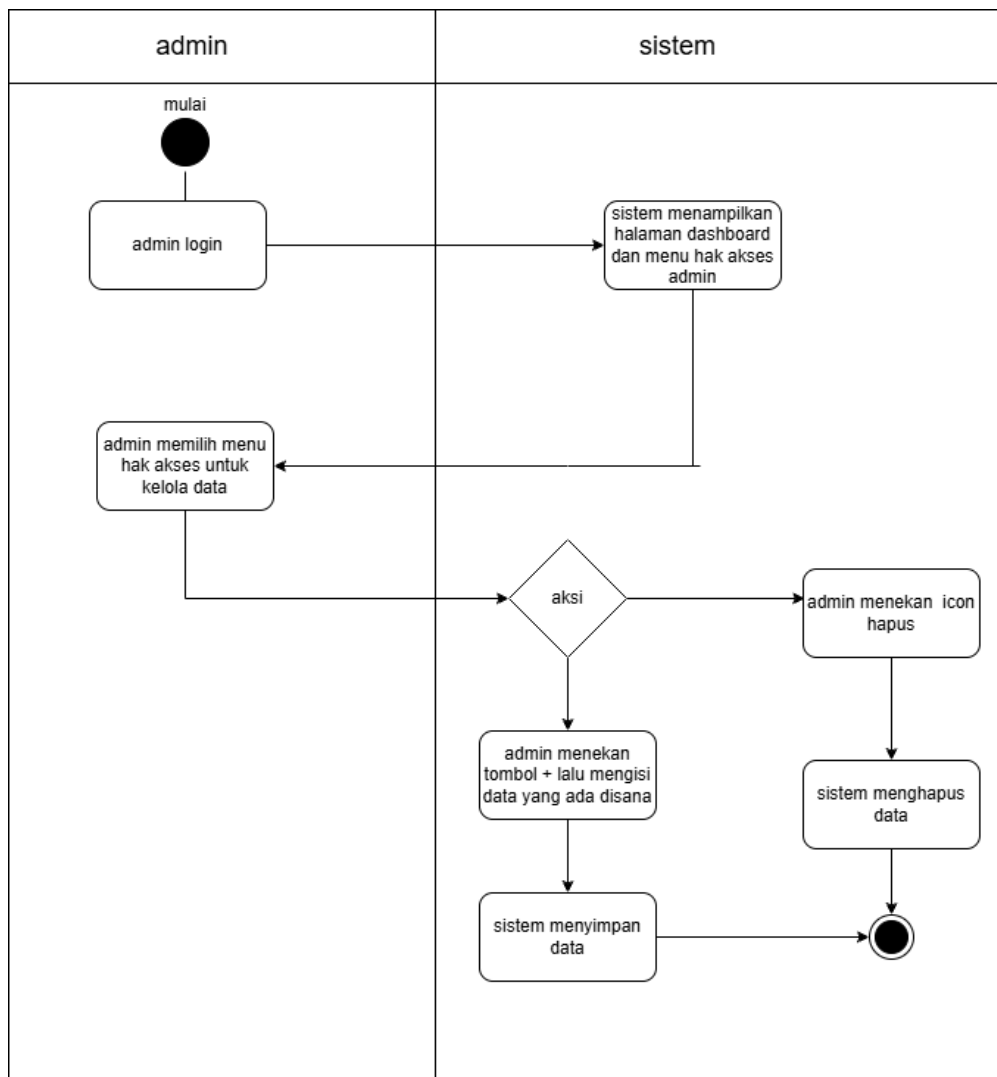
B. Activity Diagram Dashboard



Sumber: data diolah oleh penulis (2026)

Gambar 4.3 Activity Diagram dashboard

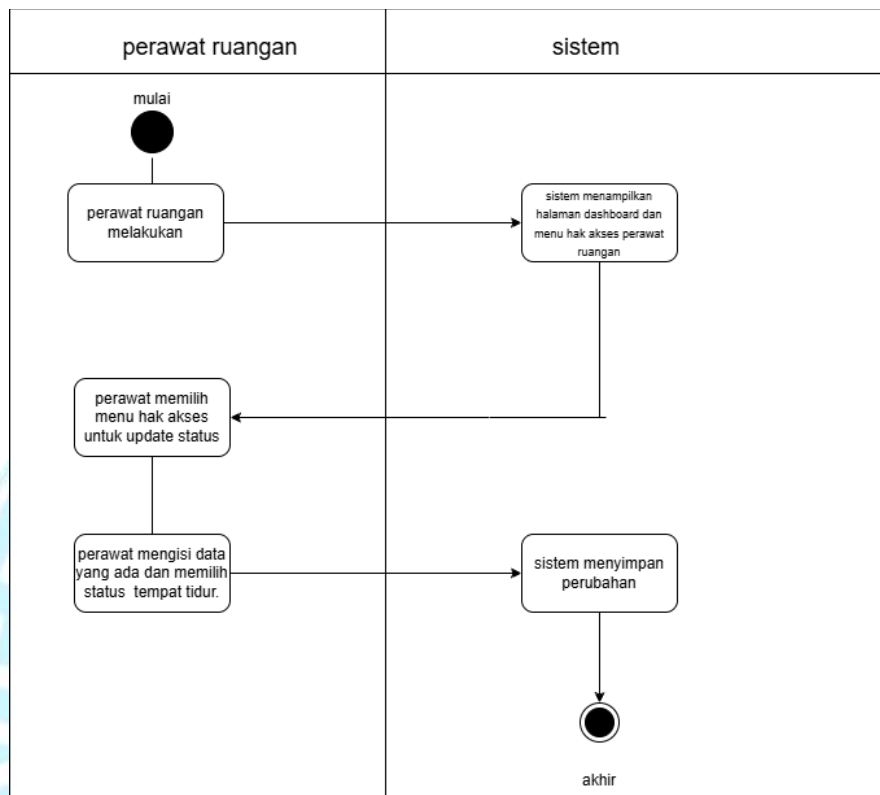
C. Activity Diagram Kelola data



Sumber: data diolah oleh penulis (2026)

Gambar 4.4 Activity Diagram Kelola Data

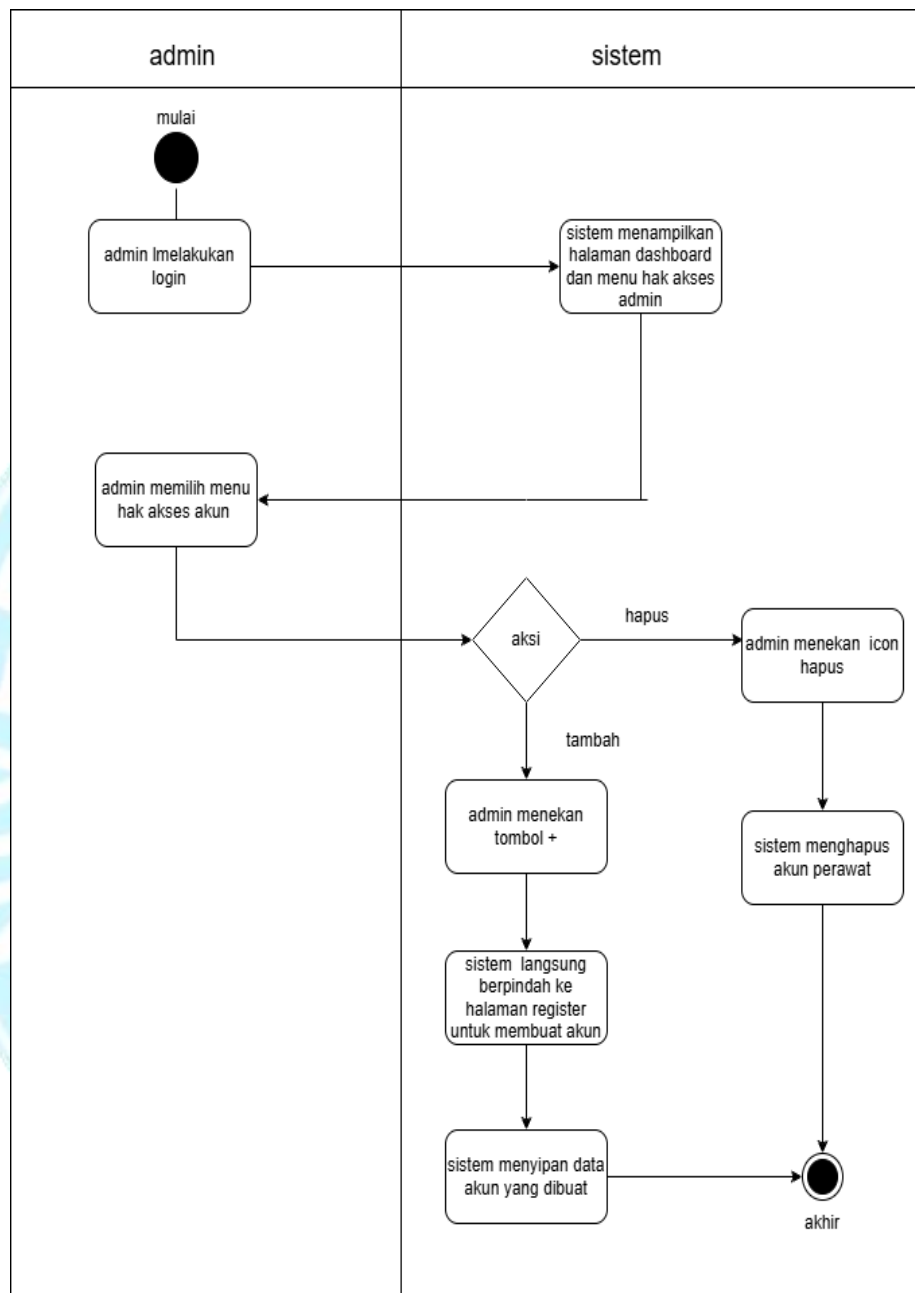
D. Activity Diagram Update Status



Sumber: data diolah oleh penulis (2026)

Gambar 4.5 Activity Diagram Update Status

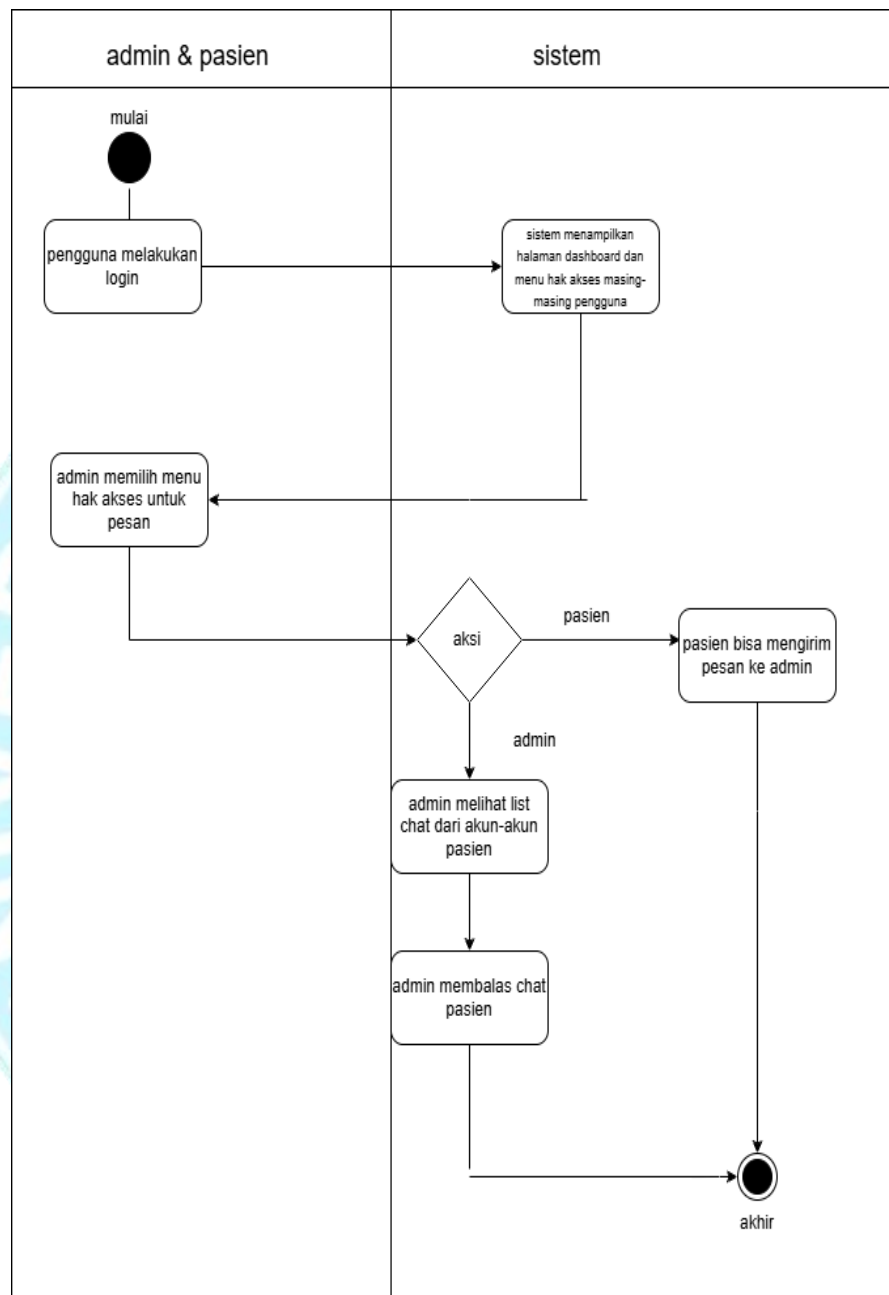
E. Activity Diagram Kelola Akun



Sumber: data diolah oleh penulis (2026)

Gambar 4.6 Activity Diagram Kelola Akun

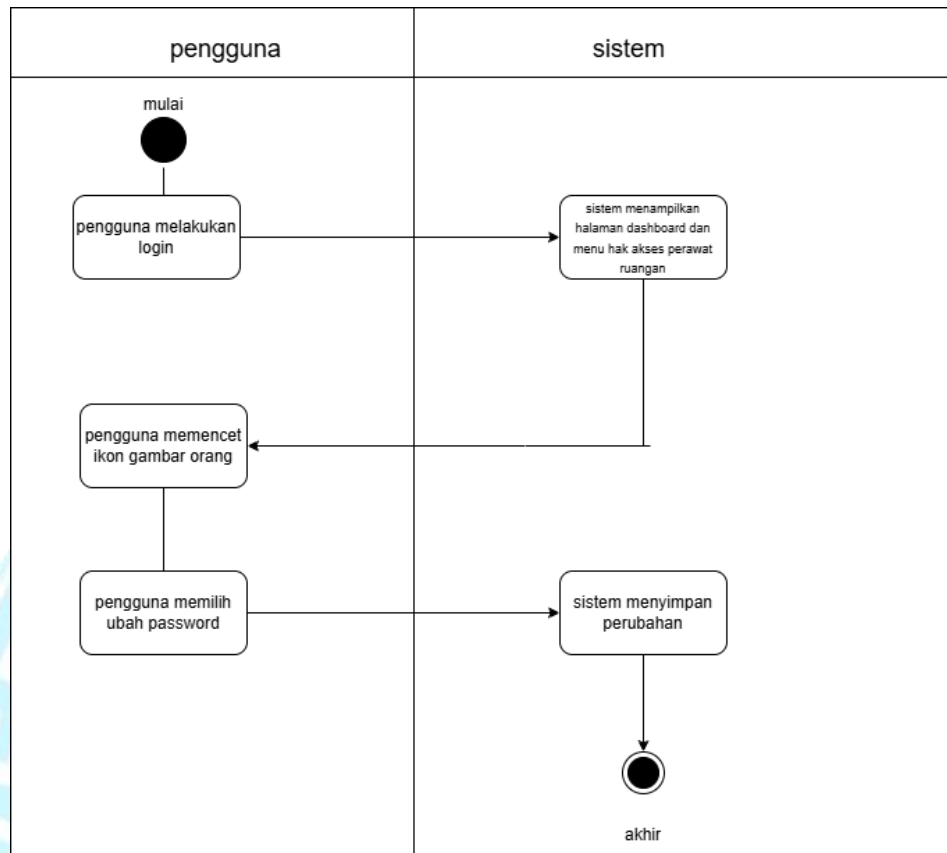
F. Activity Diagram Pesan



Sumber: data diolah oleh penulis (2026)

Gambar 4.7 Activity Diagram Pesan

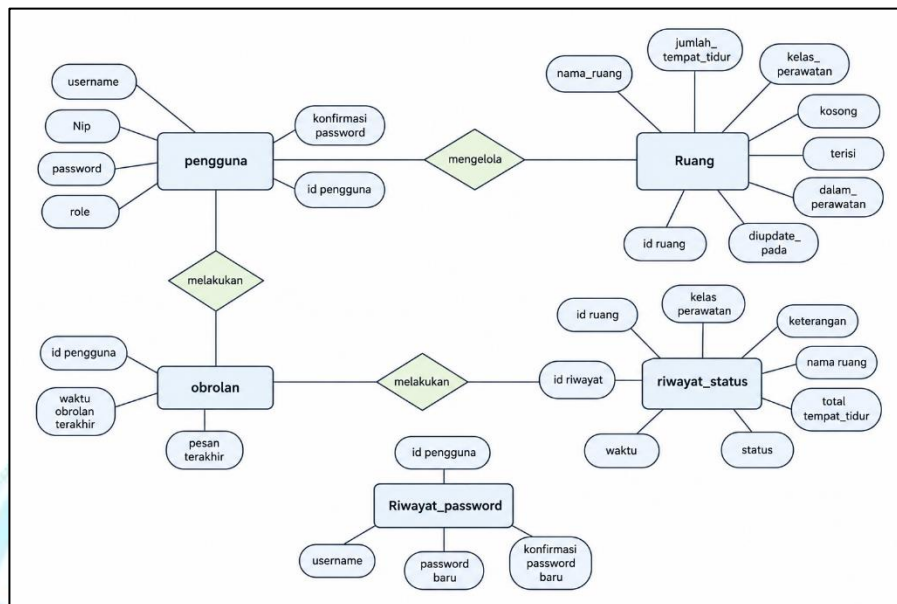
G. Activity Diagram ubah password



Sumber: data diolah oleh penulis (2026)

Gambar 4.8 Activity Diagram Ubah Password

3. Perancangan Entity Relationship Diagram (ERD)



Sumber: data diolah oleh penulis (2026)

Gambar 4.9 Entity Relationship Diagram (ERD)

4.2.2 Perancangan Database

A. Tabel Database Pengguna

Tabel 4.1 Database Pengguna

No	Nama Field	Type	Keterangan
1	<i>Id_user</i>	String (PK)	Id yang dimiliki setiap pengguna
2	<i>username</i>	String	Username pengguna
3	<i>NIP</i>	String	NIP perawat ruangan dan petugas administrasi
3	<i>password</i>	String	Password Pengguna
4	Konfirmasi <i>password</i>	String	konfirmasi password Pengguna
4	<i>Role</i>	String	Admin/perawat ruangan/Pasien

Sumber: data diolah oleh penulis (2026)

B. Tabel Database Ruang

Tabel 4.2 Database Ruang

No	Nama Field	Type	Keterangan
1	Id_Ruangan	String (PK)	Primary Key
2	Nama Ruang	String	Nama kamar
3	Kelas Perawatan	String	Contoh; Kelas I, Kelas II, dll
4	Jumlah Tempat Tidur	Int64	Jumlah tempat tidur dalam satu ruangan
5	kosong	String	Status dalam satu ruangan ada yang kosong
6	terisi	String	Status yang menandakan sudah terisi
7	Dalam perbaikan	String	Status jika ruangan sedang dalam perawatan/perbaikan
8	Di update pada	TimeStamp	Waktu saat diupdate

Sumber: data diolah oleh penulis (2026)

C. Tabel Database Riwayat Status

Tabel 4.3 Database Riwayat Status

No	Nama Field	Type	Keterangan
1	Id Riwayat	String	Id riwayat update
3	Nama Ruang	String	Nama ruangan
4	Kelas Perawatan	String	Kelas dalam satu ruangan
4	Status lama	String	Status sebelum diupdate
5	Status baru	String	Status setelah diupdate
5	Jumlah Tempat Tidur	Int64	Jumlah tempat tidur dalam satu ruangan
6	Diupdate oleh	TimeStamp	Siapa yang melakukan update
8	Diupdate pada	String	Kapan waktu melakukan update

Sumber: data diolah oleh penulis (2026)

D. Tabel Database Obrolan

Tabel 4.4 Database Obrolan

No	Nama Field	Type	Keterangan
1	Id obrolan	<i>String (PK)</i>	Id pesan masing-masing pasien dan admin
2	pengguna	<i>String</i>	Pengguna yang melakukan pesan (administrasi & pasien)
3	Waktu terakhir	<i>TimeStamp</i>	Kapan waktu saat obrolan berakhir
4	Pesan terakhir	<i>String</i>	Pesan terakhir obrolan

Sumber: data diolah oleh penulis (2026)

E. Tabel Database Riwayat password

Tabel 4.5 Database Riwayat Password

No	Nama Field	Type	Keterangan
1	Id pengguna	<i>string (PK)</i>	Id pesan masing-masing pasien dan admin
2	<i>username</i>	<i>String</i>	<i>Username</i> pengguna saat login.
3	<i>Password</i> baru	<i>String</i>	<i>Password</i> baru pengguna
4	konfirmasi <i>password</i> baru	<i>String</i>	Konfirmasi <i>password</i> baru pengguna

Sumber: data diolah oleh penulis (2026)

4.2.3 Perancangan Antarmuka

A. Halaman Login dan Register

LOGO

Sistem Monitoring Ketersediaan Tempat Tidur
pada RSUD Prabumulih

Belum punya akun? Register

LOGO

Sistem Monitoring Ketersediaan Tempat Tidur
pada RSUD Prabumulih

Udah punya akun? Login

Sumber: data diolah oleh penulis (2026)

Gambar 4.10 Halaman login dan register

B. Halaman Dashboard

No	Name Ruang	Kelas
1	Surgical-1	Kelas I
2	Pediatric-1	Kelas I
3	Medical-3	Kelas III
4	Medical-2	Kelas II
5	picu	PICU
6	Surgical-2	Kelas II

Sumber: data diolah oleh penulis (2026)

Gambar 4.11 Halaman Dashboard

C. Halaman Kelola Data

← Kelola Data LOGO

medical -1	Icon Sampah
kelas	kelas I
Jumlah Bed	7
Kosong	0
Terisi	0
Perbaikan	0

+

TAMBAH RUANGAN

Nama Ruangan

Kelas Perawatan

Jumlah bed

Batal Simpan

Sumber: data diolah oleh penulis (2026)

Gambar 4.12 Halaman Kelola Data

D. Halaman Kelola Akun

← Kelola Akun LOGO

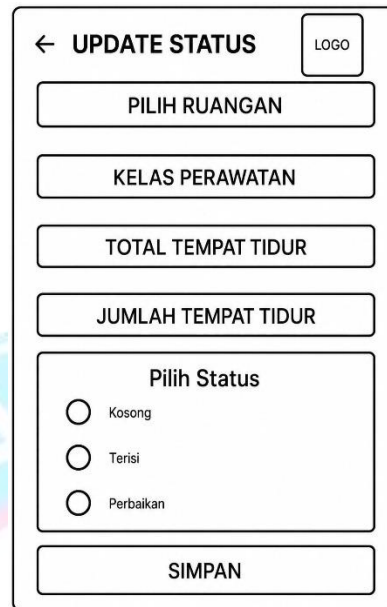
perawat4
 Nip : 123456
 Role : perawat Ruangan

+

Sumber: data diolah oleh penulis (2026)

Gambar 4.13 Halaman Kelola Akun

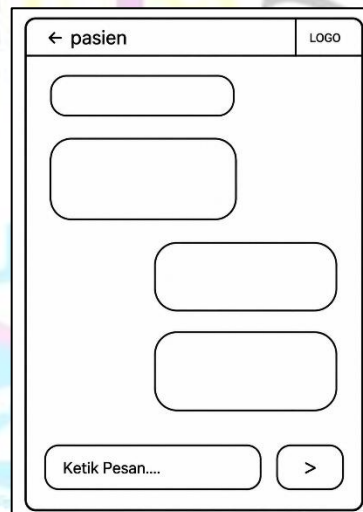
E. Halaman *Update Status*



Sumber: data diolah oleh penulis (2026)

Gambar 4.14 Halaman *Update status*

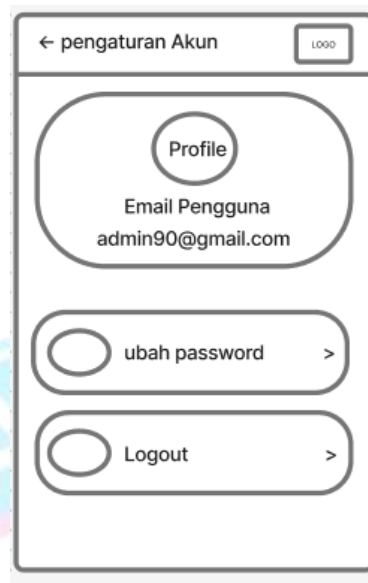
F. Halaman pesan



Sumber: data diolah oleh penulis (2026)

Gambar 4.15 Halaman Pesan

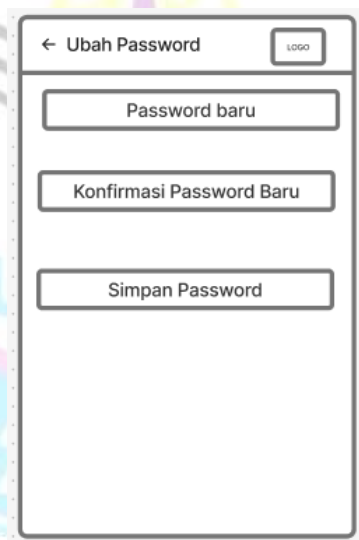
G. Halaman Pengaturan Akun



Sumber: data diolah oleh penulis (2026)

Gambar 4.16 Halaman Pengaturan Akun

H. Halaman Ubah Password



Sumber: data diolah oleh penulis (2026)

Gambar 4.17 Halaman Ubah Password

BAB V

HASIL DAN PEMBAHASAN

5.1 Spesifikasi Perangkat

Pada tahap pembuatan aplikasi dibutuhkan perangkat - perangkat yang memadai agar proses pada saat pembuatan program berjalan dengan lancar. Pada penelitian kali ini mempunyai dua spesifikasi perangkat yang digunakan untuk membuat aplikasi. Perangkat keras (*hardware*) adalah perangkat yang digunakan sebagai alat untuk membuat program dan menjalankan program, sedangkan perangkat lunak (*software*) adalah perangkat yang digunakan untuk membantu proses pembuatan program.

5.1.1 Perangkat Keras yang digunakan

Perangkat keras merupakan semua komponen atau peralatan fisik yang digunakan dalam pemrosesan informasi. Fungsi perangkat keras umumnya yaitu untuk mendukung kerja komputer, menampilkan atau menerima *input* (masukan) proses, mengolah data atau informasi, memberikan *output*, serta menyimpan data atau informasi yang terpasang di dalam maupun di luar *computer* (Dalimunthe.dkk, 2023). Perangkat keras yang saya gunakan pada pembuatan aplikasi *Android* ini adalah sebagai berikut:

Tabel 5.1 spesifikasi perangkat keras

No	Nama perangkat	Keterangan
1	Laptop Acer Aspire 3 A314-33 dengan RAM 4 GB	Digunakan untuk membuat program.
2	Smartphone vivo Y35 versi 14 dengan RAM 8 GB	Digunakan untuk menjalankan aplikasi.

5.1.2 Perangkat Lunak yang digunakan

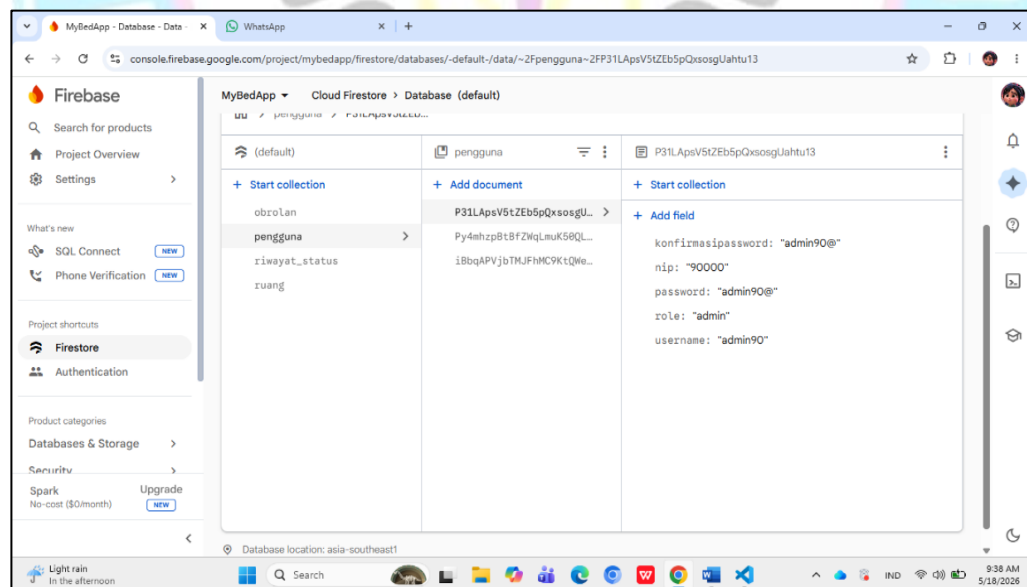
Perangkat lunak adalah program komputer yang terdiri dari serangkaian perintah atau fungsi yang dirancang untuk menjalankan tugas-tugas spesifik (Adiya, etc.2024). Perangkat lunak yang saya gunakan pada pembuatan aplikasi *Android* ini adalah sebagai berikut;

Tabel 5.2 spesifikasi perangkat lunak

No	Nama Perangkat	Keterangan
1	<i>Windows 11</i>	Sistem Operasi.
2	<i>Visual Studio Code version 1.124.0</i>	Tempat untuk membuat program.
3	<i>Framework Flutter version 3.24.4</i>	Framework yang dipakai untuk membuat program.
4	<i>Bahasa pemrograman Dart version 3.5.4</i>	Bahasa pemrograman yang dipakai.
5	<i>Firebase</i>	Database yang digunakan untuk menyimpan data.
6	Koneksi Internet	Untuk mendukung proses jalannya aplikasi.
7	<i>Figma version 126.4.11</i>	Alat bantu dalam pembuatan perancangan.

5.2 Implementasi Database

A. Database Pengguna

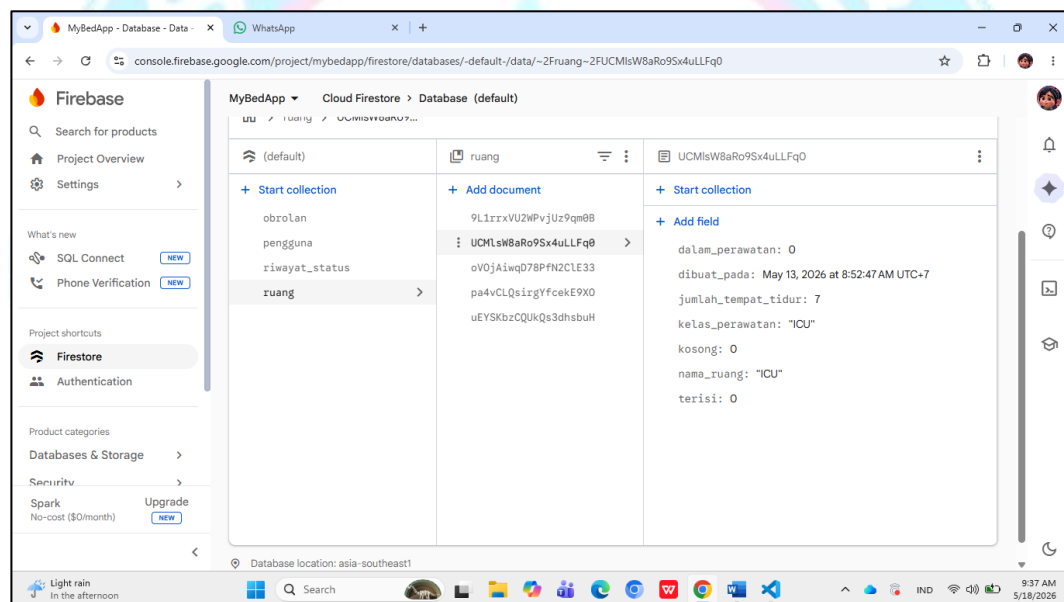


Sumber: data diolah oleh penulis (2026)

Gambar 5.1 Halaman *Database* Pengguna

Database pengguna yang berisi username, role, password, dan konfirmasi password digunakan untuk mengelola data akun pengguna dalam suatu sistem. Username berfungsi sebagai identitas unik untuk proses login, sedangkan role digunakan untuk menentukan hak akses atau wewenang pengguna, seperti admin, perawat ruangan atau pasien. Password berfungsi untuk menjaga keamanan akun agar tidak dapat diakses oleh pihak lain, dan konfirmasi password digunakan untuk memastikan bahwa password yang dimasukkan sudah benar dan sesuai. Dengan adanya data tersebut, sistem dapat mengatur akses pengguna serta menjaga keamanan informasi yang tersimpan.

B. Database Ruang



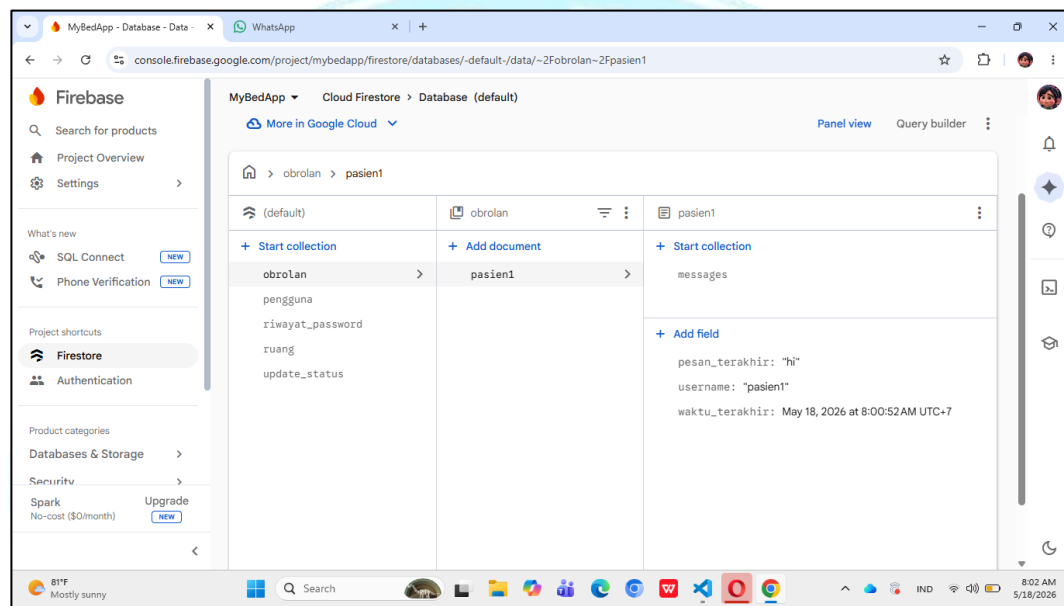
Sumber: data diolah oleh penulis (2026)

Gambar 5.2 Halaman *Database Ruang*

Database ruang pada gambar tersebut digunakan untuk menyimpan informasi mengenai ruangan perawatan di rumah sakit. Setiap dokumen pada koleksi ruang mewakili satu ruangan tertentu, seperti ruang ICU, yang memiliki beberapa atribut penting. Field *nama_ruang* berisi nama ruangan, *kelas_perawatan* menunjukkan jenis atau kelas perawatan yang tersedia, sedangkan *jumlah_tempat_tidur* menunjukkan kapasitas tempat tidur yang ada di ruangan tersebut. Field *terisi* digunakan untuk mencatat jumlah tempat tidur yang sedang digunakan dan kosong.

menunjukkan jumlah tempat tidur yang masih tersedia. Selain itu, dalam_perawatan menyimpan jumlah pasien yang sedang menjalani perawatan, dan dibuat_pada mencatat tanggal serta waktu data ruangan dibuat. Dengan adanya database ini, sistem dapat memantau kondisi dan ketersediaan ruang perawatan secara lebih mudah dan terorganisir.

C. Database Obrolan



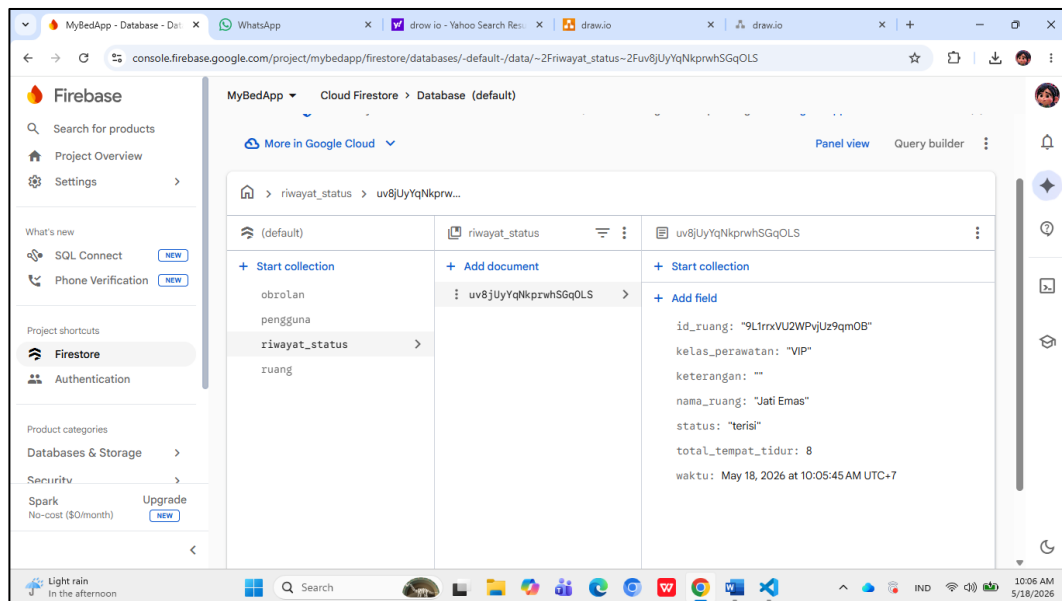
Sumber: data diolah oleh penulis (2026)

Gambar 5.3 Halaman *Database* Obrolan

Database obrolan digunakan untuk menyimpan data percakapan atau pesan antara pengguna dalam sistem. Pada koleksi obrolan, setiap dokumen mewakili satu ruang percakapan atau sesi chat, seperti dokumen pasien1. Di dalam dokumen tersebut terdapat field username yang menunjukkan identitas pengguna yang terlibat dalam percakapan, pesan_terakhir yang menyimpan isi pesan terakhir yang dikirim, serta waktu_terakhir yang mencatat tanggal dan waktu pesan terakhir dikirim. Selain itu, terdapat subkoleksi messages yang berfungsi untuk menyimpan seluruh riwayat pesan dalam percakapan tersebut. Dengan adanya database obrolan ini, sistem dapat

menampilkan daftar chat, menyimpan riwayat percakapan, dan mempermudah komunikasi antara pengguna secara *real-time*.

D. Database Riwayat Status

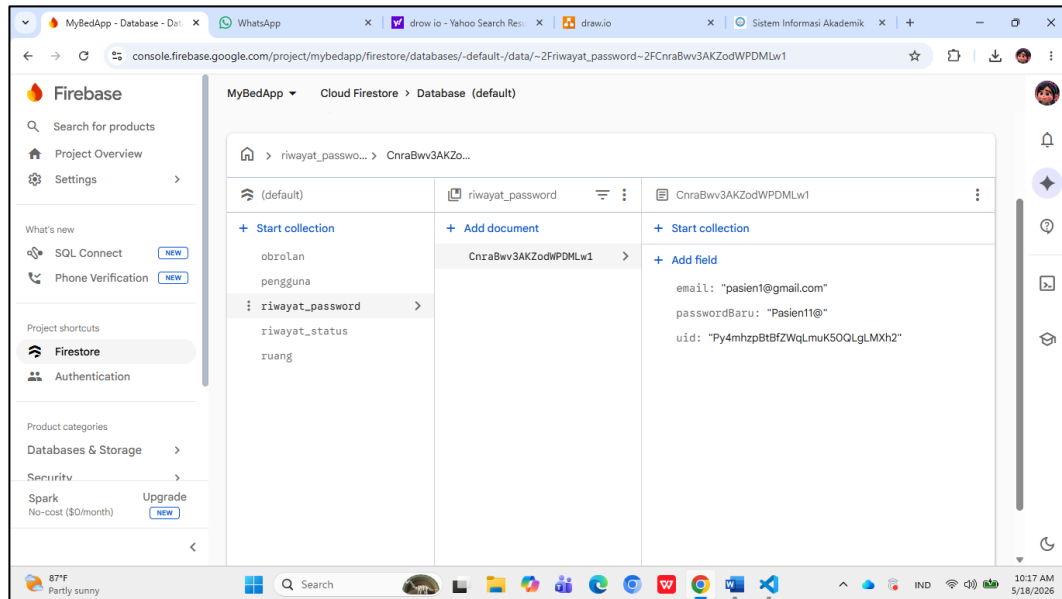


Sumber: data diolah oleh penulis (2026)

Gambar 5.4 Halaman *Database Riwayat Status*

Database riwayat_status digunakan untuk menyimpan riwayat perubahan status suatu ruangan perawatan. Setiap dokumen pada koleksi ini mencatat informasi mengenai kondisi ruangan pada waktu tertentu sehingga perubahan status ruangan dapat dipantau dan ditelusuri kembali. *Field id_ruang* berfungsi sebagai penghubung dengan data pada koleksi ruang, *nama_ruang* menyimpan nama ruangan, dan *kelas_perawatan* menunjukkan jenis kelas perawatan yang tersedia. *Field status* digunakan untuk mencatat kondisi ruangan, seperti terisi atau kosong, sedangkan *total_tempat_tidur* menunjukkan jumlah kapasitas tempat tidur pada ruangan tersebut. *Field keterangan* dapat digunakan untuk menambahkan informasi tambahan jika diperlukan, dan *waktu* mencatat tanggal serta waktu perubahan status terjadi. Dengan adanya database *riwayat_status*, sistem dapat menyimpan histori perubahan kondisi ruangan sehingga memudahkan proses pemantauan, pelaporan, dan evaluasi penggunaan ruang perawatan.

E. Database Riwayat Password



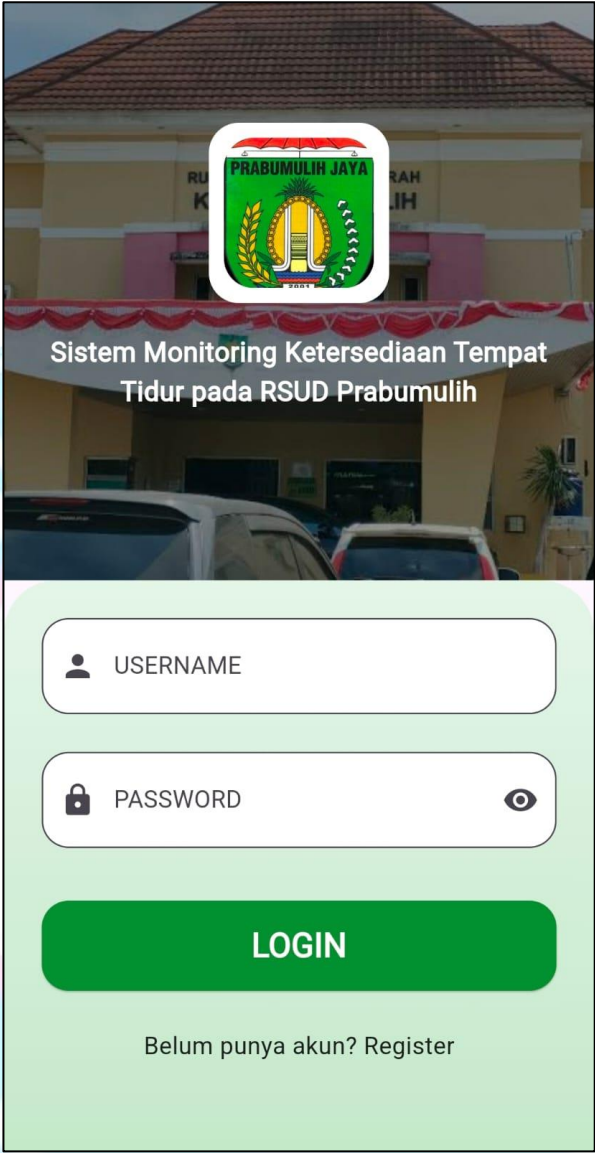
Sumber: data diolah oleh penulis (2026)

Gambar 5.5 Halaman *Database Riwayat Password*

Database riwayat_password digunakan untuk menyimpan data riwayat perubahan kata sandi pengguna. Setiap dokumen pada koleksi ini berisi informasi mengenai akun yang melakukan perubahan password sehingga perubahan tersebut dapat dicatat dan dikelola oleh sistem. *Field email* menyimpan alamat *email* pengguna yang terkait dengan akun, *passwordBaru* berisi kata sandi baru yang telah diubah, dan *uid* merupakan identitas unik pengguna yang dihasilkan oleh *Firebase Authentication*. Dengan adanya database *riwayat_password*, sistem dapat mencatat perubahan kata sandi yang dilakukan pengguna sebagai bagian dari pengelolaan akun dan pemantauan aktivitas keamanan.

5.3 Tampilan Antar Muka

A. Tampilan Antar Muka *Login*



PRABUMULIH JAYA

Sistem Monitoring Ketersediaan Tempat
Tidur pada RSUD Prabumulih

USERNAME

PASSWORD

LOGIN

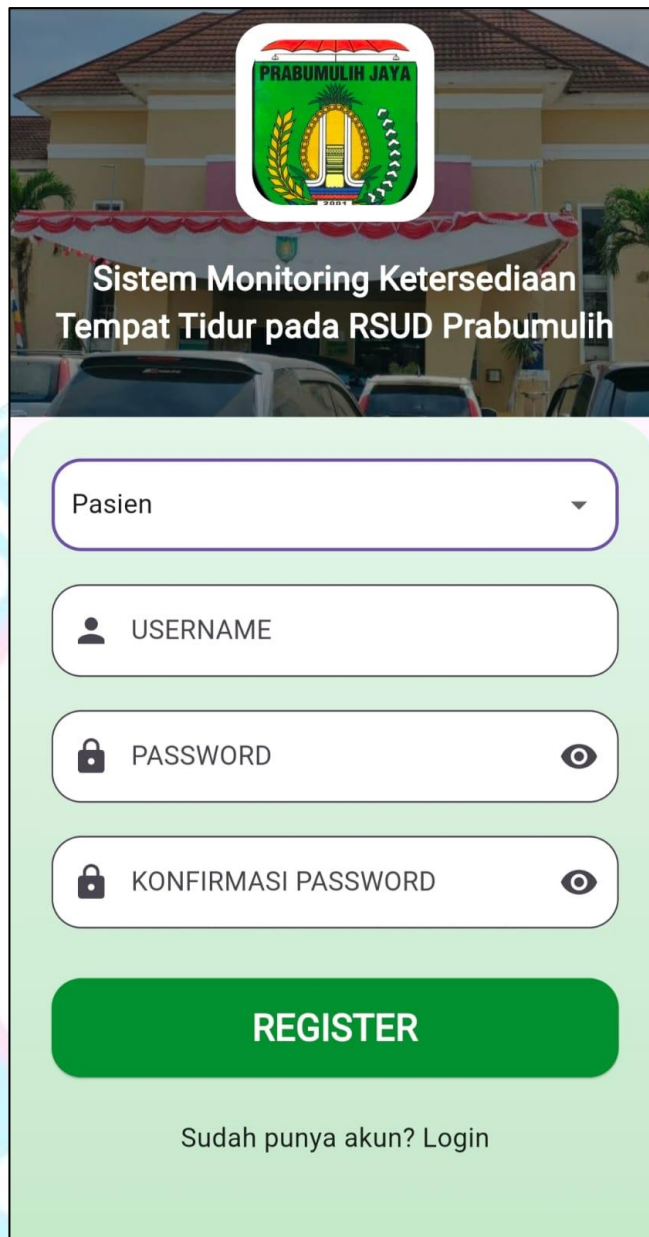
Belum punya akun? Register

Sumber: data diolah oleh penulis (2026)

Gambar 5.6 Antarmuka Halaman *login*

Halaman *login* ini adalah halaman awal pada sebuah aplikasi yang dimana para pengguna harus mengisi data seperti *username* dan *password* terlebih dahulu sesuai dengan yang telah terdaftar di *register*.

B. Tampilan Antarmuka *Register*



Sistem Monitoring Ketersediaan Tempat Tidur pada RSUD Prabumulih

Pasien

USERNAME

PASSWORD

KONFIRMASI PASSWORD

REGISTER

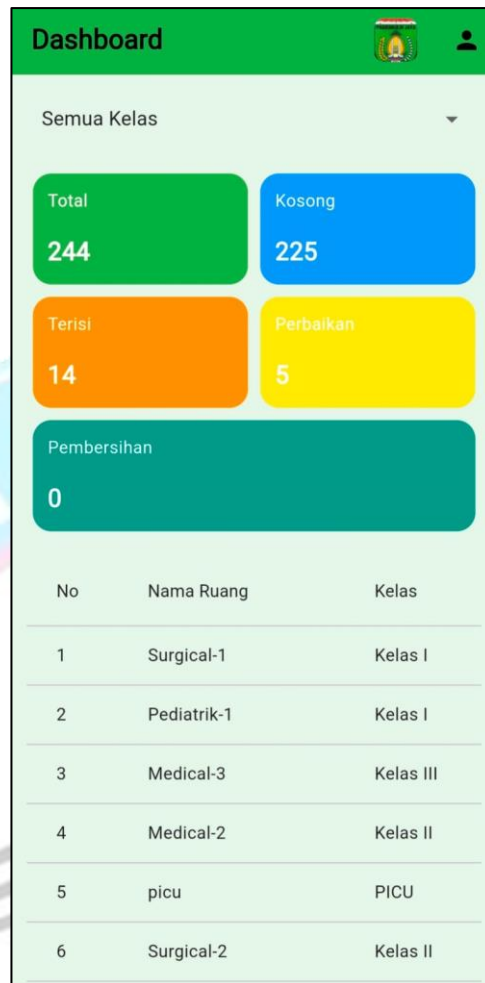
Sudah punya akun? [Login](#)

Sumber: data diolah oleh penulis (2026)

Gambar 5.7 Antarmuka Halaman *Register*

Halaman *register* adalah halaman yang dimana pengguna jika belum memiliki akun harus mendaftarkan akun terlebih dahulu di *register* dengan memilih *role* terlebih dahulu jika yang dipilih *role* pasien maka pasien harus mengisi *username*, *password* dan konfirmasi *password*. Dan jika yang dipilih adalah *role* admin atau perawat ruangan maka ada tambahan nip dalam kolom pengisian.

C. Tampilan Antarmuka *Dashboard*



Sumber: data diolah oleh penulis (2026)

Gambar 5.8 Antarmuka Halaman *Dashboard*

Halaman *dashboard* ini adalah halaman yang memperlihatkan berapa jumlah tempat tidur yang kosong, terisi, perbaikan, dan ada juga tabel yang berisi nama-nama ruang yang ada di rumah sakit RSUD Prabumulih. Halaman *dashboard* setiap pengguna itu sama yang membedakannya adalah menu yang ada di bawah untuk admin terdapat menu kelola data, kelola akun perawat dan pesan. Pada

dashboard perawat ruangan terdapat menu *update* status dan pada *dashboard* pasien hanya terdapat menu *chat* admin.

D. Tampilan Antarmuka Kelola Data Ruangan

Kelola Data

Surgical-1

Kelas	Kelas I
Jumlah Bed	4
Kosong	3
Terisi	0
Perbaikan	1

Pediatrik-1

Kelas	Kelas I
Jumlah Bed	4
Kosong	0
Terisi	0
Perbaikan	0

Medical-3

Kelas	Kelas III
Jumlah Bed	23

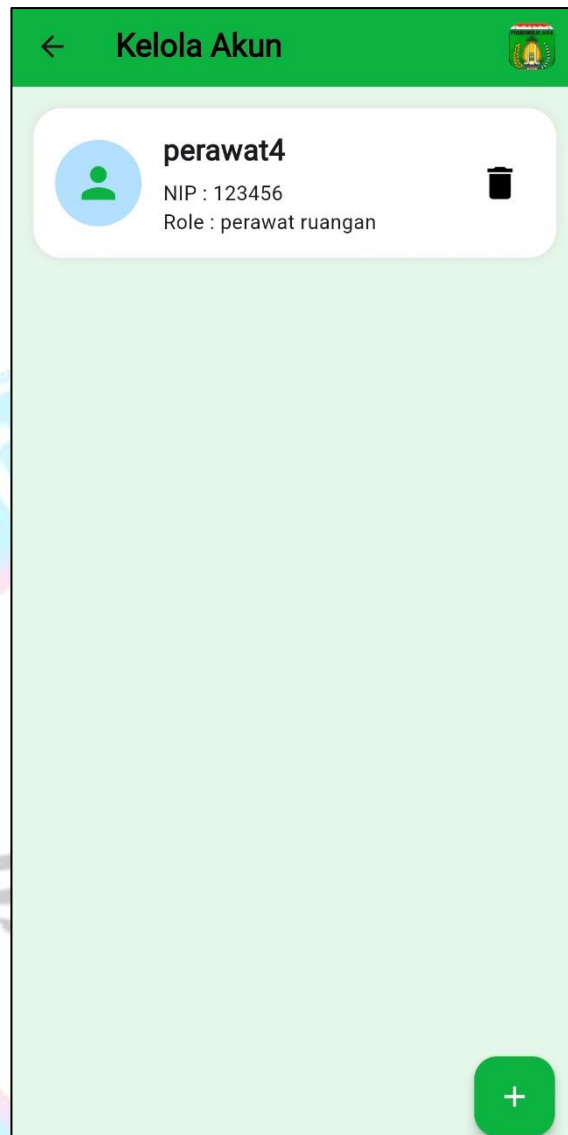
+

Sumber: data diolah oleh penulis (2026)

Gambar 5.9 Antarmuka Halaman Kelola Data Ruangan

Halaman kelola data ini adalah halaman yang digunakan oleh *admin* untuk menambah ruangan dengan menklik tombol '+' untuk menambah ruangan dan akan keluar data-data yang harus diisi oleh *admin* yaitu nama ruangan, kelas perawatan dan jumlah tempat tidur dan untuk menghapus ruangan admin hanya tinggal memencet ikon sampah untuk menghapus ruangan yang ada.

E. Tampilan Antarmuka kelola Akun

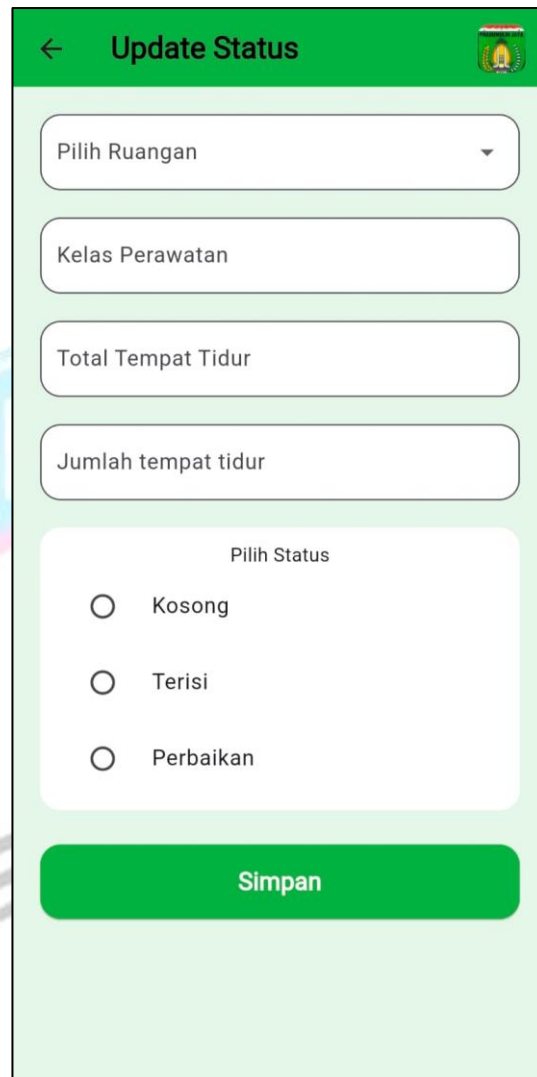


Sumber: data diolah oleh penulis (2026)

Gambar 5.10 Antarmuka Halaman Kelola Akun

Halaman kelola akun ini hanya digunakan oleh *admin* untuk menghapus akun perawat ruangan jika ada perawat ruangan yang keluar dan *admin* juga bisa membuatkan akun perawat ruangan.

F. Tampilan Antarmuka *Update Status*

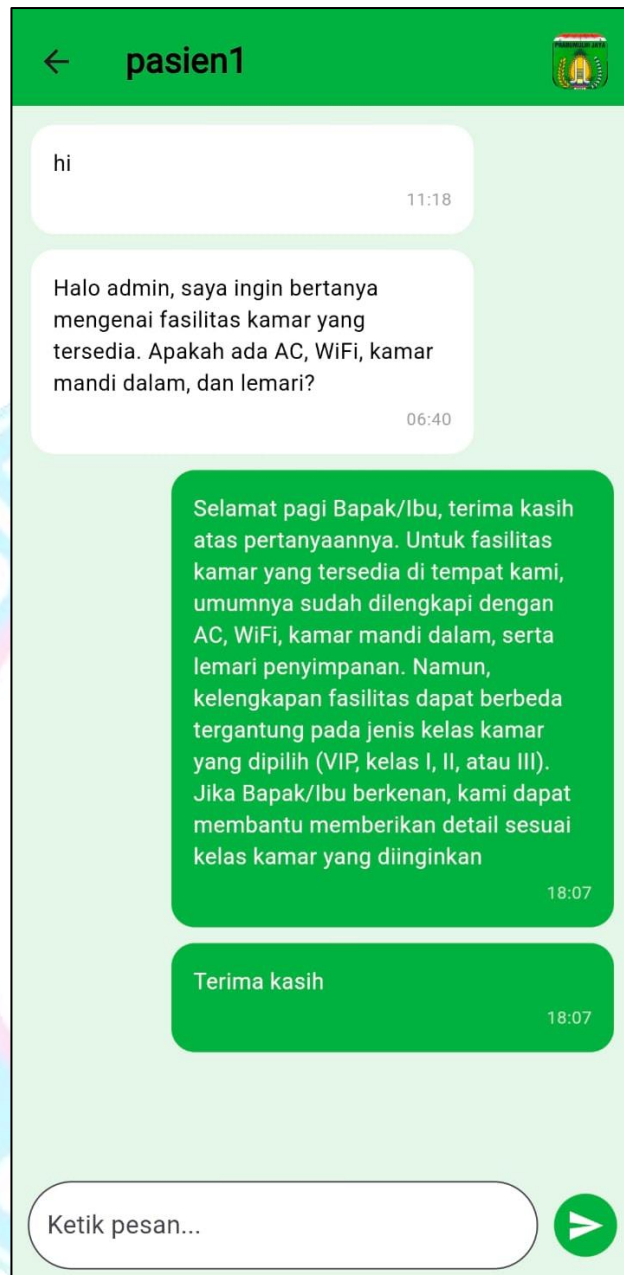


Sumber: data diolah oleh penulis (2026)

Gambar 5.11 Antarmuka Halaman Update Status

Halaman *update status* ini hanya digunakan oleh perawat ruangan untuk mengupdate status tempat tidur dengan mengisi data data yang dibutuhkan seperti pilih ruangan, kelas perawatan, total tempat tidur, pilih status dan simpan. Setelah simpan maka tabel yang ada di halaman *dashboard* akan berubah secara otomatis.

G. Tampilan Antarmuka Pesan

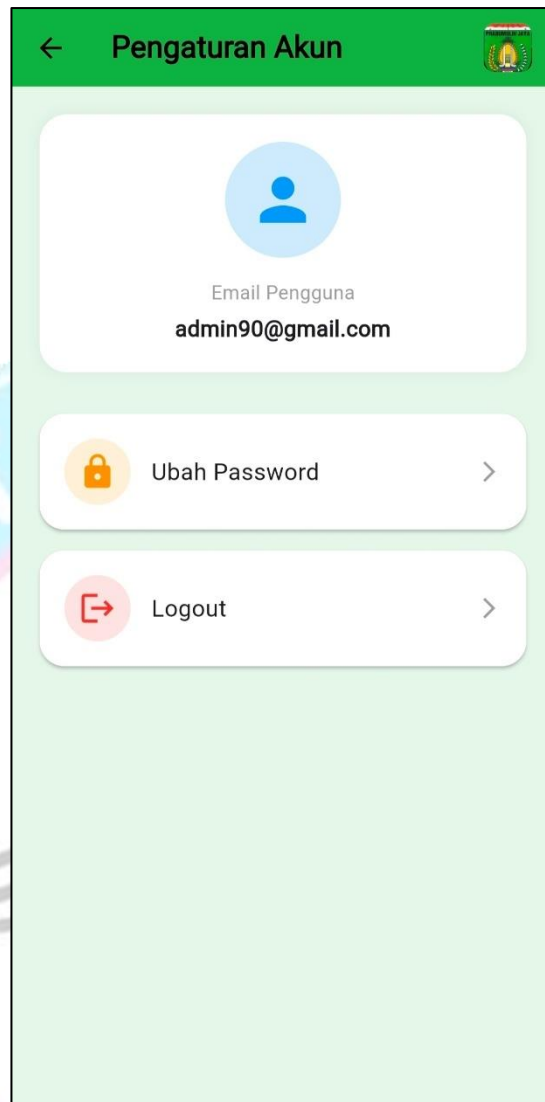


Sumber: data diolah oleh penulis (2026)

Gambar 5.12 Antarmuka Halaman Pesan

Halaman pesan ini hanya digunakan oleh *admin* dan pasien untuk berkomunikasi untuk mengetahui info lebih lanjut terkait fasilitas yang dimiliki oleh ruangan.

H. Tampilan Pengaturan Akun

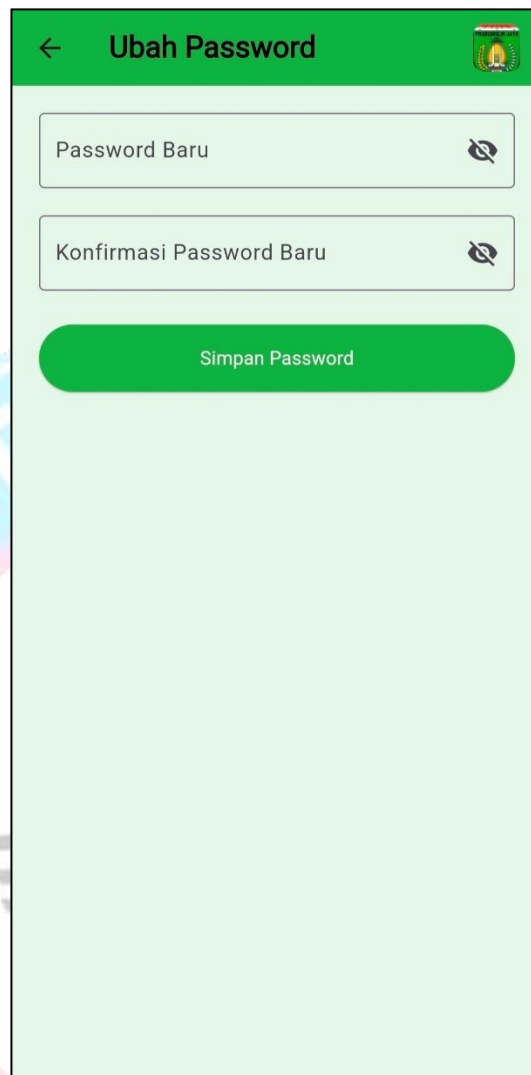


Sumber: data diolah oleh penulis (2026)

Gambar 5.13 Antarmuka Halaman Pengaturan Akun

Halaman pengaturan akun ini adalah halaman yang berisi *email* pengguna yang terdaftar dan pengguna bisa mengubah *password* nya dan juga bisa logout.

I. Tampilan Ubah Password



Sumber: data diolah oleh penulis (2026)

Gambar 5.14 Halaman Ubah Password

Halaman ubah password ini digunakan untuk pengguna jika ingin mengubah password yang lama menjadi password yang baru.

5.4 Hasil Pengujian Sistem

1. Pengujian Login

Berikut ini adalah kodingan *login* yang akan diuji menggunakan *whitebox testing*

```

if (username.isEmpty || password.isEmpty) {
    ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(
            content: Text("Semua kolom wajib diisi"),
        ),
    );
    return;
}

try {
    setState(() {
        isLoading = true;
    });

    String email = "$username@gmail.com";

    UserCredential userCredential =
        await FirebaseAuth.instance.signInWithEmailAndPassword(
            email: email,
            password: password,
        );

    String uid = userCredential.user!.uid;

    DocumentSnapshot userDoc = await FirebaseFirestore.instance
        .collection("pengguna")

```

```

        .doc(uid)
        .get();

    if (!userDoc.exists) {
        throw Exception("Data pengguna tidak ditemukan");
    }

    var userData = userDoc.data() as Map<String, dynamic>;

    String role = userData['role'] ?? "";
    String nama = userData['username'] ?? "";

    if (!mounted) return;

    Navigator.pushReplacement(
        context,
        MaterialPageRoute(
            builder: (context) => DashboardPage(
                role: role,
                username: nama,
            ),
        ),
    );

    ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(
            content: Text("LOGIN BERHASIL"),
        ),
    );
} on FirebaseAuthException catch (e) {
    String message = "";

```

```

if (e.code == "user-not-found") {

    message = "Username tidak ditemukan";
} else if (e.code == "wrong-password") {
    message = "Password salah";
} else if (e.code == "invalid-credential") {
    message = "Username atau password salah";
} else {
    message = "Login gagal";
}

ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(content: Text(message)),
);
} catch (e) {
    ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(
            content: Text("Terjadi kesalahan: $e"),
        ),
    );
} finally {
    });
}
}
}

```

A. Tujuan dilakukannya *Statement Coverage* pada halaman *login*

Statement Coverage dilakukan untuk memastikan bahwa setiap baris pernyataan dalam kode setidaknya diuji minimal satu kali. Pada halaman login ini dilakukannya pengujian bertujuan untuk:

1. Memastikan seluruh logika *login* berjalan dengan benar.
2. Memastikan tidak ada kode yang tidak dijalankan.

3. Menemukan kemungkinan *error* yang kemungkinan ada saat terjadinya pengujian.

B. Identifikasi *Statement* Halaman *Login*

Tabel 5.3 Identifikasi *Statement* Halaman *Login*

S1	<code>if (username.isEmpty password.isEmpty)</code>
S2	<code>ScaffoldMessenger.of(context).showSnackBar(const SnackBar(content: Text("Semua kolom wajib diisi"),</code>
S3	<code>setState() { isLoading = true;</code>
S4	<code>.signInWithEmailAndPassword(email: email, password: password,);</code>
S5	<code>if (!userDoc.exists) {</code>
S6	<code>throw Exception("Data pengguna tidak ditemukan"); }</code>
S7	<code>if (!mounted) return;</code>
S8	<code>Navigator.pushReplacement(ScaffoldMessenger.of(context).showSnackBar(const SnackBar(content: Text("LOGIN BERHASIL"),),);</code>
S9	<code>if (e.code == "user-not-found") { message = "Username tidak ditemukan";</code>
S10	<code>else if (e.code == "wrong-password") { message = "Password salah";</code>
S11	<code>else if (e.code == "invalid-credential") { message = "Username atau password salah";</code>

```

S12    else {
        message = "Login gagal";
S13    ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(S3-
        content: Text("Terjadi kesalahan: $e"),
S14    finally {
        });

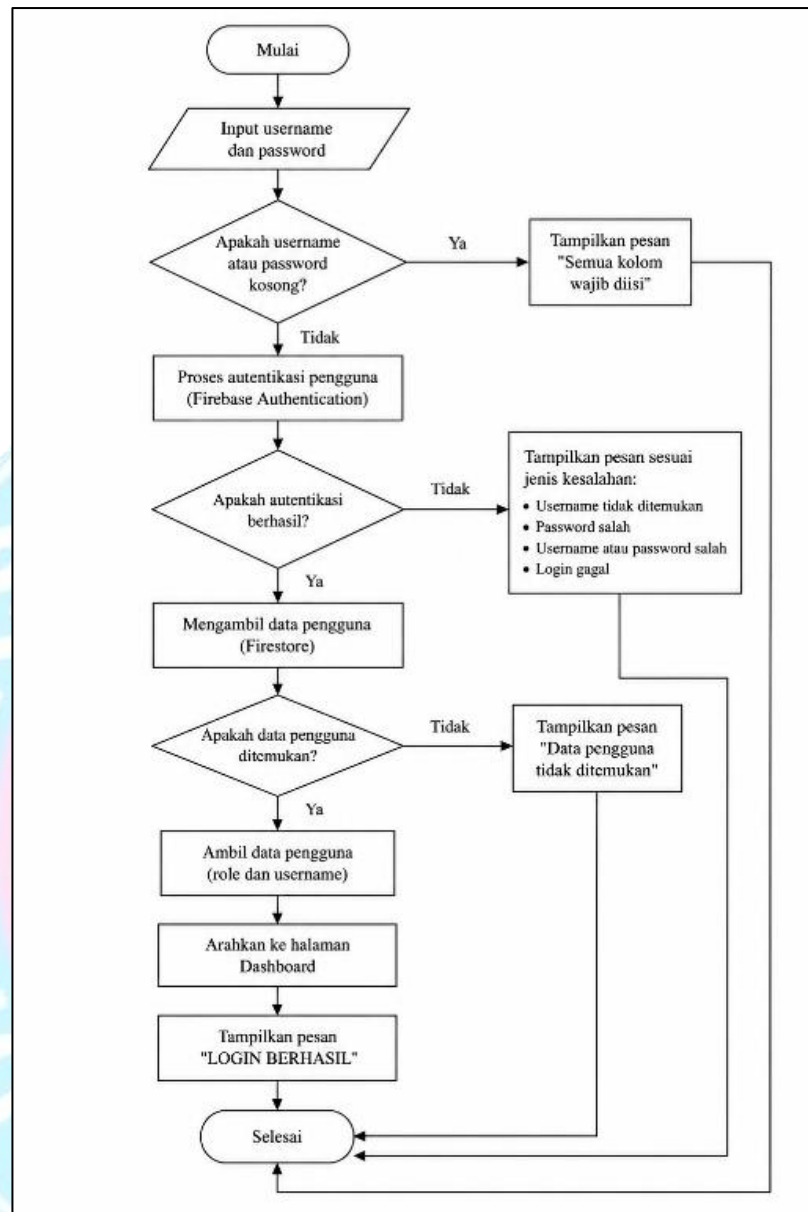
```

C. Test Case untuk Statement Coverage

Tabel 5.4 Test Case Halaman Login

Test Case	Input	Jalur Eksekusi	Output
TC1	Username dan password kosong	S1 – S2	Semua kolom wajib diisi
TC2	Username dan password benar, data firestore tersedia	S1-S3-S4-S5-S7-S8-S9-S14	Login Berhasil
TC3	Saat login data pengguna tidak ada di firestore	S1-S3-S4-S5-S6-S13	Data pengguna tidak ditemukan
TC4	Username tidak ditemukan	S1-S3-S4-S9-S14	Username tidak ditemukan
TC5	Password salah	S1-S3-S4-S10-S14	Password salah
TC6	Username dan password salah	S1-S3-S4-11-S14	Username dan password salah
TC7	Login gagal	S1-S3-S4-S12-S14	Login Gagal

D. Flowchart Login



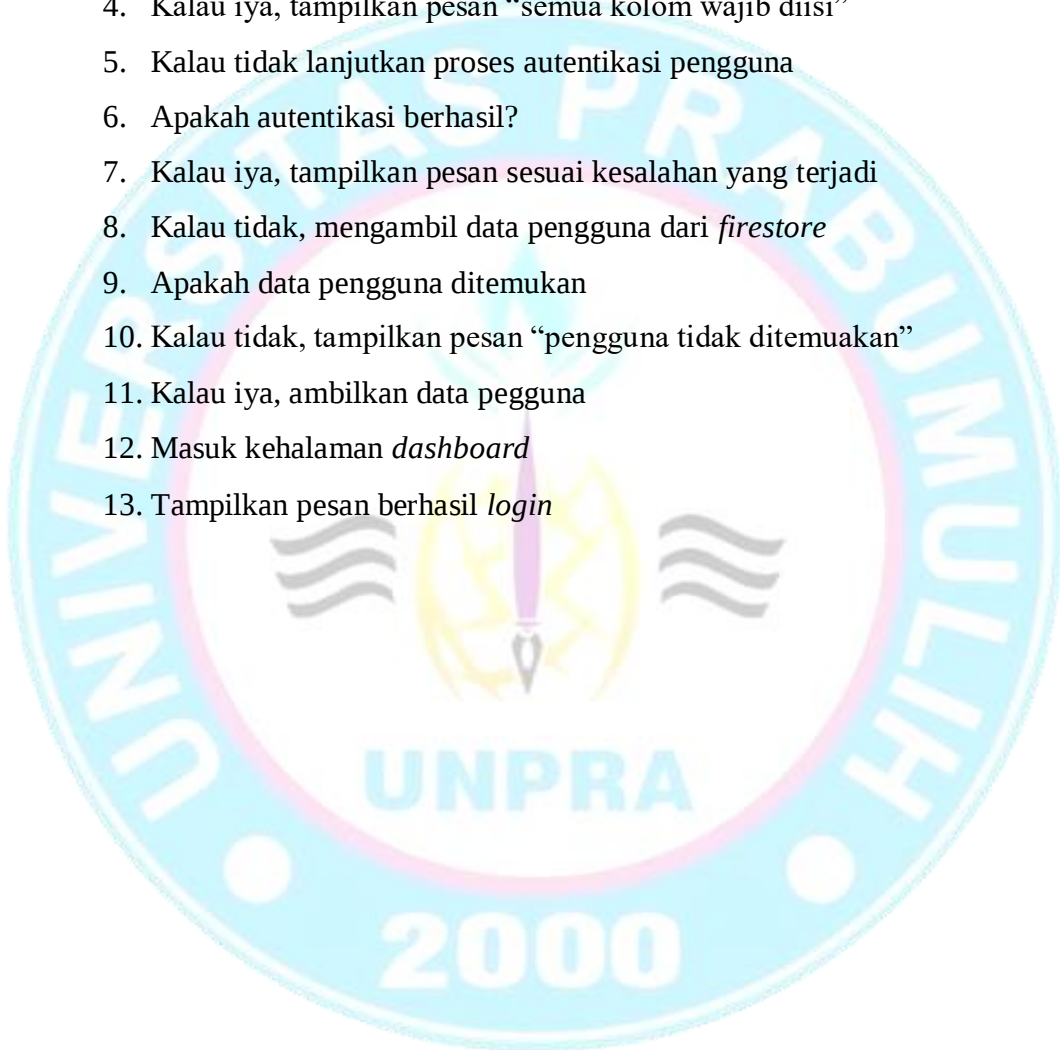
Sumber: data diolah oleh penulis (2026)

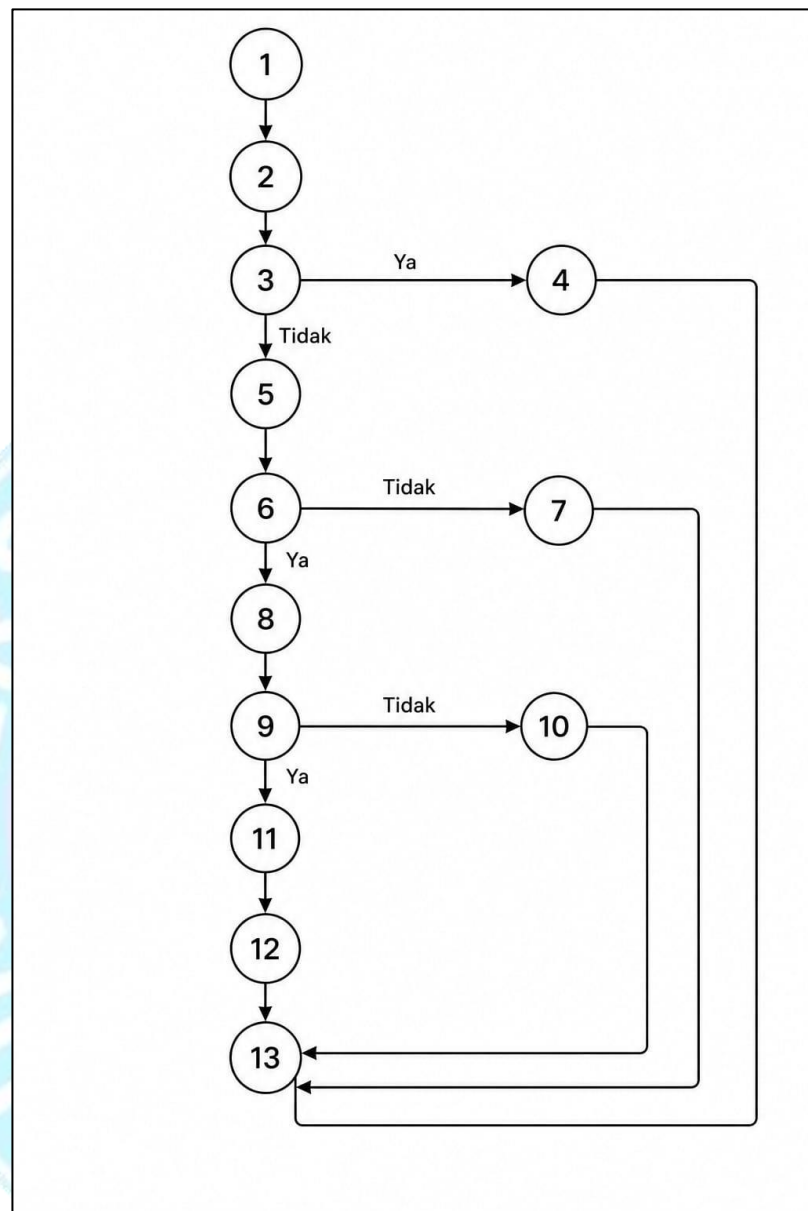
Gambar 5.15 Flowchart halaman login

E. Node dan Edge

Node

1. Mulai
2. *Input username dan password*
3. Apakah *username* dan *password* kosong?
4. Kalau iya, tampilkan pesan “semua kolom wajib diisi”
5. Kalau tidak lanjutkan proses autentikasi pengguna
6. Apakah autentikasi berhasil?
7. Kalau iya, tampilkan pesan sesuai kesalahan yang terjadi
8. Kalau tidak, mengambil data pengguna dari *firestore*
9. Apakah data pengguna ditemukan
10. Kalau tidak, tampilkan pesan “pengguna tidak ditemukan”
11. Kalau iya, ambilkan data pengguna
12. Masuk kehalaman *dashboard*
13. Tampilkan pesan berhasil *login*



F. Flowgraph Login

Sumber: data diolah oleh penulis (2026)

Gambar 5.16 *Flowgraph* halaman *login*

G. Jalur Independen (Path)

Tabel 5.5 Jalur independen halaman *login*

No	Jalur Independen	penjelasan
1	1-2-3-4-13	Jalur menunjukkan kondisi ketika kolom <i>username</i> dan <i>password</i> kosong.
2	1-2-3-4-5-6-7-13	Jalur yang menunjukkan kondisi ketika pengguna salah memasukkan <i>username</i> dan <i>password</i> .
3	1-2-3-4-5-6-7-8-9-10-13	Jalur yang menunjukkan kondisi data pengguna yang dimasukkan tidak ada di <i>firebase firestore</i> .
4	1-2-3-5-6-8-9-11-12-13	Jalur yang menunjukkan kondisi bahwa <i>login</i> berhasil dilakukan.

2. Pengujian Register

Berikut ini adalah kodingan *register* yang akan diuji menggunakan *whitebox testing*

```
if (username.isEmpty ||
    password.isEmpty ||
    konfirmasiPassword.isEmpty ||
    role == null) {
    ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(
            content: Text("Semua form wajib diisi"),
        ),
    );
};
```

```
return;

}

if (username.contains(" ")) {

    ScaffoldMessenger.of(context).showSnackBar(

        const SnackBar(

            content: Text(

                "Username tidak boleh mengandung spasi",

            ),

        ),

    );

    return;

}

if (username.length < 4) {

    ScaffoldMessenger.of(context).showSnackBar(

        const SnackBar(

            content: Text(

                "Username minimal 4 karakter",

            ),

        ),

    );

    return;
```

```
}
```

```
if (!validasiPassword(password)) {
```

```
  ScaffoldMessenger.of(context).showSnackBar(
```

```
    const SnackBar(
```

```
      content: Text(
```

```
        "Password minimal 8 karakter dengan huruf besar, huruf kecil, angka  
dan simbol",
```

```
      ),
```

```
    ),
```

```
  );
```

```
  return;
```

```
}
```

```
if (password != konfirmasiPassword) {
```

```
  ScaffoldMessenger.of(context).showSnackBar(
```

```
    const SnackBar(
```

```
      content: Text(
```

```
        "Konfirmasi password tidak cocok",
```

```
      ),
```

```
    ),
```

```
  );
```

```
  return;
```

```
}
```

```
if ((role == "admin" || role == "perawat ruangan") && nip.isEmpty) {
```

```
  ScaffoldMessenger.of(context).showSnackBar(
```

```
    const SnackBar(
```

```
      content: Text("NIP wajib diisi"),
```

```
    ),
```

```
  );
```

```
  return;
```

```
}
```

```
try {
```

```
  setState() {
```

```
    isLoading = true;
```

```
  });
```

```
QuerySnapshot cekUsername = await FirebaseFirestore.instance
```

```
  .collection("pengguna")
```

```
  .where(
```

```
    "username",
```

```
    isEqualTo: username,
```

```
  )
```

```
  .limit(1)
```

```

    .get();

    if (cekUsername.docs.isNotEmpty) {

        ScaffoldMessenger.of(context).showSnackBar(

            const SnackBar(

                content: Text(

                    "Username sudah digunakan",

                ),

            ),

        );

        setState() {

            isLoading = false;

        });

        return;

    }

    UserCredential userCredential =

        await FirebaseAuth.instance.createUserWithEmailAndPassword(

            email: email,

            password: password,

        );

```

```

await FirebaseFirestore.instance
    .collection("pengguna")
    .doc(userCredential.user!.uid)
    .set({
        "uid": userCredential.user!.uid,
        "username": username,
        "email": email,
        "nip": role == "pasien" ? "" : nip,
        "role": role,
        "dibuatPada": Timestamp.now(),
    });

if (!mounted) return;

ScaffoldMessenger.of(context).showSnackBar(
    const SnackBar(
        content: Text(
            "Register berhasil",
        ),
    ),
);

Navigator.pushReplacement(

```

```

context,

MaterialPageRoute(

  builder: (context) => const LoginPage(),

),

);

} on FirebaseAuthException catch (e) {

  String message = "";

  if (e.code == "email-already-in-use") {

    message = "Username sudah digunakan";

  } else if (e.code == "weak-password") {

    message = "Password terlalu lemah";

  } else if (e.code == "invalid-email") {

    message = "Email tidak valid";

  } else if (e.code == "network-request-failed") {

    message = "Koneksi internet bermasalah";

  } else {

    message = "Terjadi kesalahan";

  }

}

ScaffoldMessenger.of(context).showSnackBar(

  SnackBar(

    content: Text(message),

  ),

```

```

);

} catch (e) {

  ScaffoldMessenger.of(context).showSnackBar(

    SnackBar(

      content: Text(

        "Error: $e",

      ),

    ),

  );

} finally {

  if (mounted) {

    setState() {

      isLoading = false;

    });

  }

}

}

```

A. Tujuan dilakukannya *Statement Coverage* pada halaman *register*

Statement Coverage dilakukan untuk memastikan bahwa setiap baris pernyataan dalam kode setidaknya diuji minimal satu kali. Pada halaman login ini dilakukannya pengujian bertujuan untuk:

1. Memastikan seluruh proses validasi input berjalan dengan benar.
2. Menguji semua kondisi.
3. Memastikan proses registrasi ke *firebase auth* dan *firestore* berjalan dengan benar.

4. Menjamin bahwa semua alur berfungsi sesuai harapan.

A. Indentifikasi *Statement* Halaman *Register*

Tabel 5.6 Identifikasi *Statement* Halaman *Register*

S1	<i>if</i> (<i>username.isEmpty</i> <i>password.isEmpty</i> <i>konfirmasiPassword.isEmpty</i> <i>role == null</i>)
S2	<i>SnackBar</i> (<i>content: Text</i> ("Semua form wajib diisi"),
S3	<i>Return</i> ;
S4	<i>if</i> (<i>username.contains</i> (" "))
S5	<i>SnackBar</i> (<i>content: Text</i> ("Username tidak boleh mengandung spasi",),
S6	<i>Return</i> ;
S7	<i>if</i> (<i>username.length</i> < 4) {
S8	<i>SnackBar</i> (<i>content: Text</i> ("Username minimal 4 karakter",),
S9	<i>Return</i> ;
S10	<i>if</i> (! <i>validasiPassword(password)</i>) {

S11	<i>SnackBar</i> (<i>content: Text</i> ("Password minimal 8 karakter dengan huruf besar, huruf kecil, angka dan simbol",),
S12	<i>Return;</i>
S13	<i>if</i> (<i>password</i> != <i>konfirmasiPassword</i>) {
S14	<i>SnackBar</i> (<i>content: Text</i> ("Konfirmasi <i>password</i> tidak cocok",),
S15	<i>Return;</i>
S16	<i>if</i> ((<i>role</i> == "admin" <i>role</i> == "perawat ruangan") && <i>nip.isEmpty</i>)
	{
S17	<i>SnackBar</i> (<i>content: Text</i> ("NIP wajib diisi"),),
S18	<i>Return;</i>
S19	<i>try</i> { <i>setState</i> () { <i>isLoading</i> = <i>true</i> ; } };
S20	<i>cekUsername</i> = <i>await FirebaseFirestore.instance</i> <i>.collection</i> ("pengguna") <i>.where</i> ("username",

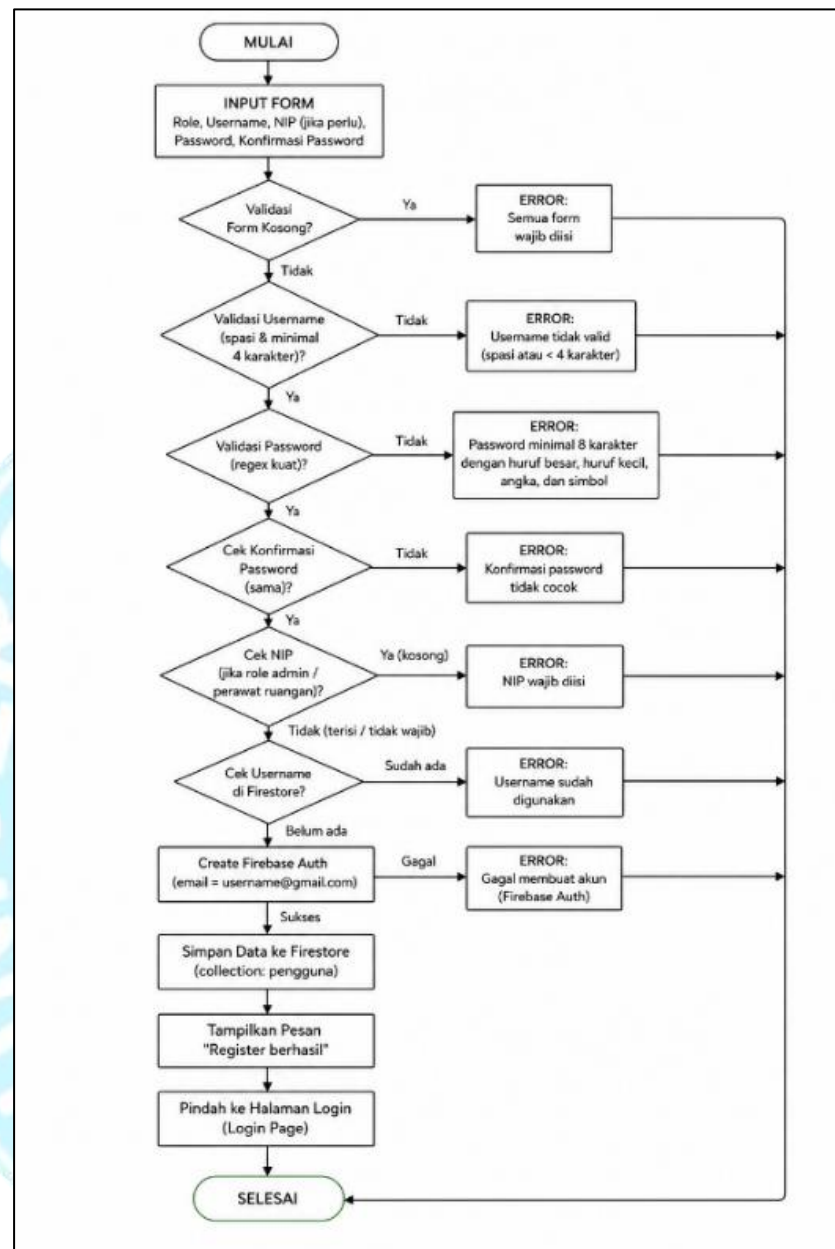
	<i>isEqualTo: username,</i>
	<i>)</i>
S21	<i>if (cekUsername.docs.isEmpty) {</i>
S22	<i>SnackBar(</i>
	<i>content: Text(</i>
	<i>"Username sudah digunakan",</i>
	<i>),</i>
S23	<i>setState() {</i>
	<i>isLoading = false;</i>
	<i>});</i>
S24	<i>Return;</i>
S25	<i>createUserWithEmailAndPassword(</i>
	<i>email: email,</i>
	<i>password: password,</i>
	<i>);</i>
S26	<i>FirebaseFirestore.instance</i>
	<i>.collection("pengguna")</i>
S27	<i>SnackBar(</i>
	<i>content: Text(</i>
	<i>"Register berhasil",</i>
	<i>),</i>
S28	<i>Navigator.pushReplacement(</i>
S29	<i>FirebaseAuthException catch (e) {</i>
S30	<i>Finally {</i>

B. Test Case untuk Statement Coverage

Tabel 5.7 Test Case Halaman Register

<i>Test case</i>	<i>Input</i>	Jalur eksekusi	<i>output</i>
<i>TC1</i>	Semua <i>form</i> kosong	S1-S2-S3-S29	Semua kolom wajib diisi
<i>TC2</i>	<i>Username</i> mengandung spasi	S4-S5-S6-S29	<i>Username</i> tidak boleh mengandung spasi
<i>TC3</i>	<i>Username</i> mengandung minimal 4 karakter	S7-S8-S9-S29	<i>Username</i> minimal 4 karakter
<i>TC4</i>	Validasi <i>password</i>	S10-S11-S12-S29	<i>Password</i> minimal 8 karakter dengan mengandung huruf,angka,simbol
<i>TC5</i>	Konfirmasi <i>password</i>	S13-S14-S15-S29	Konfirmasi <i>password</i> tidak sama
<i>TC6</i>	Nip wajib diisi	S16-S17-S18-S29	Nip wajib diisi
<i>TC7</i>	Cek <i>username</i>	S20-S21-S22-S29	<i>Username</i> sudah digunakan
<i>TC8</i>	Semua <i>form</i> diisi dan data disimpan ke <i>firestore</i>	S1-S4-S7-S10-S13-S14-S16-S19-S20-S25-S26-S27-S28-S29	<i>Register</i> berhasil
<i>TC9</i>	<i>Password</i> terlalu lemah	S1-S3-S5-S7-S9-S11-S20-S21-S22-S23-S24-S25-S29	<i>Password</i> terlalu lemah

C. Flowchart Register



Sumber: data diolah oleh penulis (2026)

Gambar 5.17 Flowchart halaman register

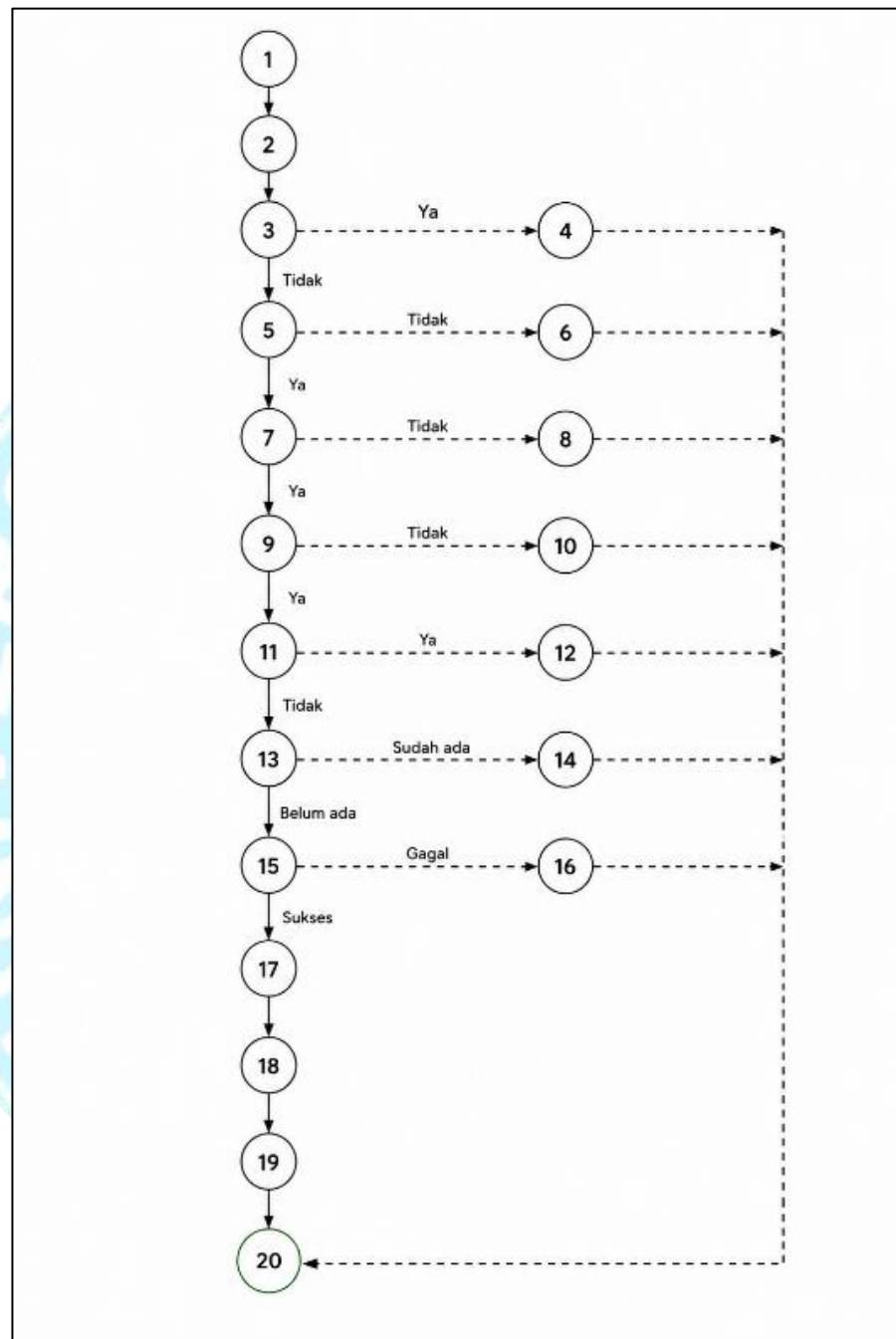
D. Node dan Edge

Node

1. Mulai

2. *Input form*
3. Validasi *form* kosong?
4. “semua *form* wajib diisi”
5. validasi *username*?
6. tampilkan pesan “*username* tidak boleh mengandung spasi dan *username* minimal 4 karakter”
7. melanjutkan ke validasi *password*?
8. tampilkan pesan “ *password* minimal 8 karakter dengan kombinasi huruf,angka, dan simbol”
9. lanjutkan ke cek konfirmasi *password*
10. Tampilkan “konfirmasi *password* tidak cocok”
11. Melanjutkan ke cek nip
12. Tampilkan pesan “Nip wajib diisi”
13. Cek *username* di *firestore*
14. Tampilkan pesan “*username* sudah digunakan”
15. Create *firebase auth* dengan (email=username@gmail.com)
16. Gagal membuat akun
17. Simpan data ke *firestore*
18. Tampilkan pesan “*register* berhasil”
19. Pindah ke halaman *login*
20. Selesai

E. Flowgraph Register



Sumber: data diolah oleh penulis (2026)

Gambar 5.18 Flowgraph halaman register

F. Jalur Independen (*Path*)

Tabel 5.8 jalur independen halaman *register*

No	Jalur Independen	penjelasan
1	1-2-3-4-20	Jalur menunjukkan kondisi ketika semua kolom kosong.
2	1-2-3-5-6-20	Jalur yang menunjukkan kondisi ketika pengguna salah memasukkan <i>username</i> salah.
3	1-2-3-4-5-7-8-20	Jalur yang menunjukkan kondisi ketika pengguna salah memasukkan <i>password</i> .
4	1-2-3-5-7-9-10-20	Jalur yang menunjukkan kondisi <i>password</i> dan konfirmasi <i>password</i> yang dimasukkan tidak sama.
5	1-2-3-5-7-9-11-12-20	Jalur yang menunjukkan kondisi ketika <i>admin</i> dan perawat belum mengisi nip.
6	1-2-3-5-7-9-11-13-14-20	Jalur yang menunjukkan kondisi bahwa <i>username</i> telah digunakan.
7	1-2-3-5-7-9-11-13-15-16-20	Jalur yang menunjukkan kondisi bahwa gagal membuat akun.
8	1-2-3-5-7-9-11-13-17-18-19-20	Jalur yang menunjukkan kondisi bahwa <i>register</i> berhasil dilakukan.

3. Pengujian kelola data

Berikut ini adalah kodingan kelola data yang akan diuji menggunakan whitebox testing

```
if (!mounted) return;
```

```
Navigator.pop(context);
```

```
ScaffoldMessenger.of(context).showSnackBar(
```

```

const SnackBar(
  content: Text("Ruangan berhasil ditambahkan"),
),
);
}

```

```

Future<void> hapusRuangan(String id) async {
  await firestore.collection("ruang").doc(id).delete();
  if (!mounted) return;
  ScaffoldMessenger.of(context).showSnackBar(
    const SnackBar(
      content: Text("Data berhasil dihapus"),
    ),
  );
}

```

A. Tujuan dilakukannya *Statement Coverage* pada halaman kelola data

Statement Coverage dilakukan untuk memastikan bahwa setiap baris pernyataan dalam kode setidaknya diuji minimal satu kali. Pada halaman login ini dilakukannya pengujian bertujuan untuk:

1. Mengetahui apakah seluruh fungsi pada halaman kelola data berfungsi dengan baik.
2. Memastikan tidak ada kode yang tidak diuji.
3. Menjamin alur logika aplikasi berjalan sesuai dengan yang diharapkan

B. Identifikasi *Statement* Halaman Kelola Data

Tabel 5.9 Identifikasi *Statement* Halaman Kelola Data

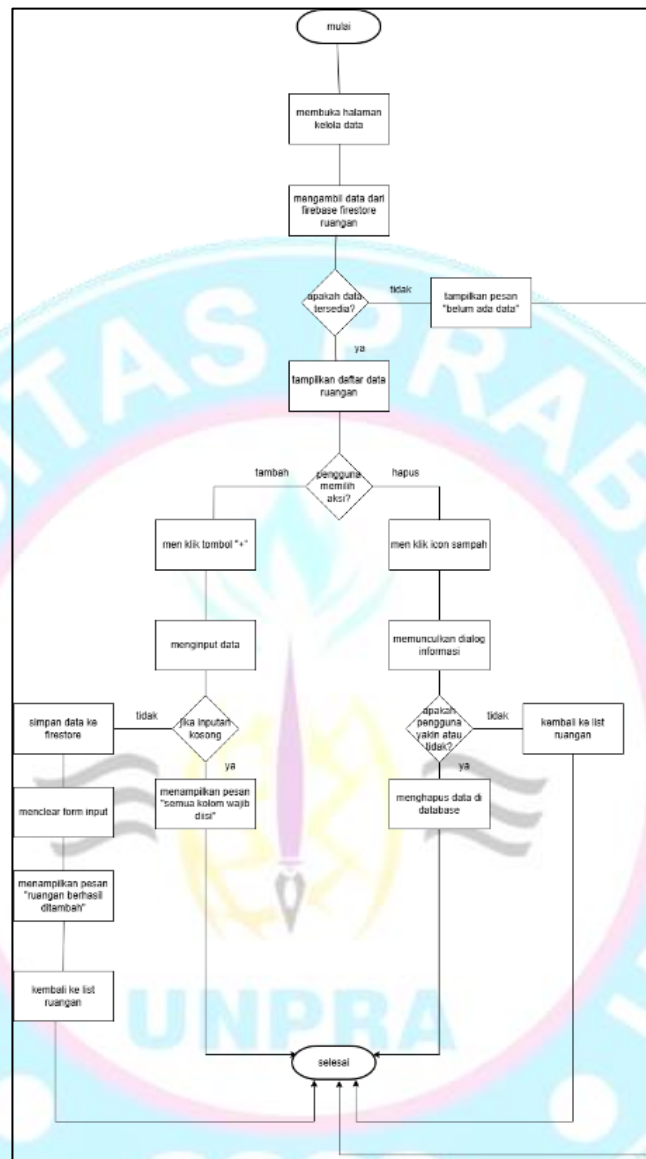
S1	<i>if (!mounted) return;</i>
S2	<i>ScaffoldMessenger.of(context).showSnackBar(const SnackBar(content: Text("Ruangan berhasil ditambahkan"),),</i>
S3	<i>await firestore.collection("ruang").doc(id).delete();</i>
S4	<i>if (!mounted) return;</i>
S5	<i>ScaffoldMessenger.of(context).showSnackBar(const SnackBar(content: Text("Data berhasil dihapus"),),</i>

C. Test Case untuk *Statement Coverage*

Tabel 5.10 *Test Case* Halaman Kelola Data

<i>Test Case</i>	<i>Input</i>	Jalur Eksekusi	<i>Output</i>
TC1	Tambah ruangan berhasil	S1-S2	Ruangan berhasil ditambahkan
TC2	Hapus ruangan berhasil	S3-S4-S5	Ruangan berhasil dihapus

D. Flowchart Kelola Data



Sumber: data diolah oleh penulis (2026)

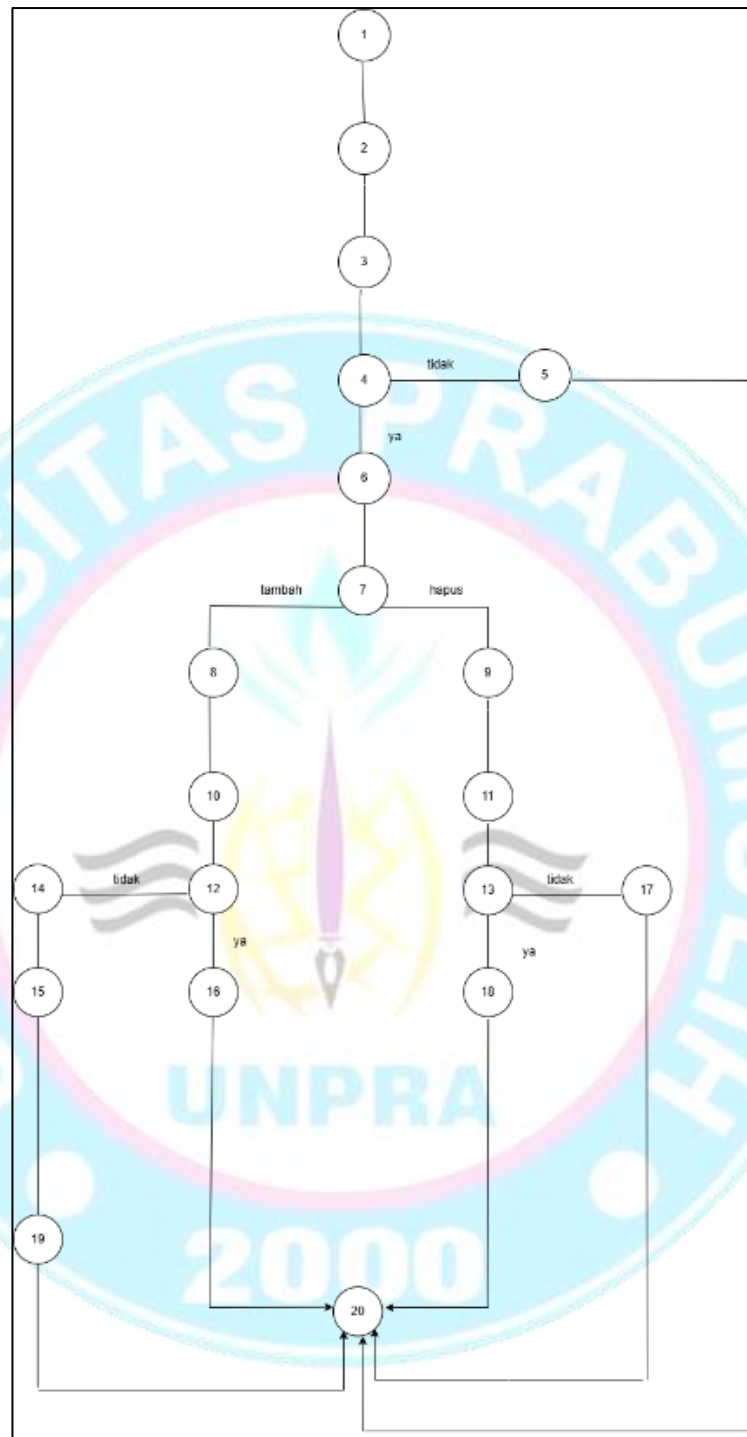
Gambar 5.19 *Flowchart* halaman kelola data

E. Node dan Edge

Node

1. Mulai
2. Membuka halaman kelola data
3. Mengambil data dari *firestore* ruang
4. Apakah data tersedia
5. Jika tambah, men klik tombol tambah
6. Menginput data
7. Jika inputan kosong
8. Jika iya, menampilkan pesan “semua kolom wajib diisi”
9. Jika tidak, Simpan data ke *firestore*
10. Menclear *form input*
11. Menampilkan pesan “ruangan berhasil ditambah”
12. Jika hapus, men klik *icon* sampah
13. Memunculkan dialog informasi
14. Apakah pengguna yakin atau tidak
15. Jika tidak, kembali ke list ruangan
16. Jika iya, menghapus data
17. selesai

F. *Flowgraph* Kelola Data



Sumber: data diolah oleh penulis (2026)

Gambar 5.20 *Flowgraph* halaman kelola data

G. Jalur Independen (*Path*)

Tabel 5.11 Jalur independen halaman kelola data

No	Jalur Independen	penjelasan
1	1-2-3-4-5-20	Jalur yang menunjukkan kondisi ketika belum ada data ruangan.
2	1-2-3-4-6-7-8-10-12-14-15-19-20	Jalur yang menunjukkan kondisi ketika melakukan tambah data ruangan.
3	1-2-3-4-6-7-8-10-12-16-20	Jalur yang menunjukkan kondisi ketika pengguna belum mengisi apapun di kolom tambah.
4	1-2-3-4-6-7-9-11-13-17-20	Jalur yang menunjukkan kondisi ketika <i>admin</i> batal melakukan penghapusan.
5	1-2-3-4-6-7-9-11-13-18-20	Jalur yang menunjukkan kondisi ketika <i>admin</i> setuju melakukan penghapusan.

4. Pengujian *update status*

Berikut ini adalah kodingan *update status* yang akan diuji menggunakan *whitebox testing*

```

if (selectedRuangId == null) {

  ScaffoldMessenger.of(context).showSnackBar(

    const SnackBar(

      content: Text(

        "Pilih ruangan terlebih dahulu",

      ),

    ),

  );
}

```

```

return;
}

```

```

if (status.isEmpty) {
    ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(
            content: Text(
                "Pilih status terlebih dahulu",
            ),
        ),
    );
    return;
}

int jumlah = int.tryParse(
    jumlahController.text,
) ??
0;

```

```

if (jumlah <= 0) {
    ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(
            content: Text(

```

```

        "Jumlah tidak valid",

    ),

    ),

);

return;
}

if (jumlah > totalTempatTidur) {
    ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(
            content: Text(
                "Jumlah melebihi total bed",
            ),
        ),
    );
    return;
}

```

```

if (status == "perbaikan" &&
    keteranganController.text.trim().isEmpty) {
    ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(

```

```
        content: Text(  
            "Keterangan perbaikan wajib diisi",  
        ),  
    ),  
);  
return;  
}  
  
try {  
    int kosong = 0;  
    int terisi = 0;  
    int perbaikan = 0;  
  
    if (status == "kosong") {  
        kosong = jumlah;  
        terisi = totalTempatTidur - kosong;  
    } else if (status == "terisi") {  
        terisi = jumlah;  
        kosong = totalTempatTidur - terisi;  
    } else if (status == "perbaikan") {  
        perbaikan = jumlah;  
        kosong = totalTempatTidur - perbaikan;  
    }  
}
```

```

await firestore.collection("riwayat_status").add({

    "id_ruang": selectedRuangId,

    "nama_ruang": selectedNamaRuang,

    "kelas_perawatan": kelasPerawatan,

    "total_tempat_tidur": totalTempatTidur,

    "status": status,

    "keterangan":

        keteranganController.text.trim(),

    "waktu": FieldValue.serverTimestamp(),

});

```

```

DocumentSnapshot ruangDoc = await firestore
    .collection("ruang")
    .doc(selectedRuangId)
    .get();

```

```

Map<String, dynamic> dataLama =
    ruangDoc.data() as Map<String, dynamic>;

```

```

if (status == "perbaikan") {

    await firestore
        .collection("ruang")

```

```

.doc(selectedRuangId)

.update({

"jumlah_tempat_tidur": totalTempatTidur,

"kosong": kosong,

"terisi": terisi,

"dalam_perbaikan": perbaikan,

"status_ruang": "Perbaikan",

"keterangan":

    keteranganController.text.trim(),

// waktu mulai perbaikan

"mulai_perbaikan":

    dataLama["mulai_perbaikan"] ??

        FieldValue.serverTimestamp(),

"waktu_diupdate":

    FieldValue.serverTimestamp(),

});

} else {

    await firestore

        .collection("ruang")

        .doc(selectedRuangId)

        .update({

```

```

    "jumlah_tempat_tidur": totalTempatTidur,

    "kosong": kosong,

    "terisi": terisi,

    "dalam_perbaikan": perbaikan,

    "status_ruang": "Aktif",

    // reset data perbaikan
    "keterangan": "",
    "mulai_perbaikan": null,

    "waktu_diupdate":
        FieldValue.serverTimestamp(),
    });
}

jumlahController.clear();
keteranganController.clear();

setState() {
    status = "";
}
};

```

```

ScaffoldMessenger.of(context).showSnackBar(

  const SnackBar(

    content: Text(

      "Status berhasil disimpan",

    ),

  ),

);
} catch (e) {
  ScaffoldMessenger.of(context).showSnackBar(
    SnackBar(
      content: Text(
        "Terjadi kesalahan: $e",
      ),
    ),
  );
}
}

```

A. Tujuan dilakukannya *Statement Coverage* pada Halaman *Update Status*

Statement Coverage dilakukan untuk memastikan bahwa setiap baris pernyataan dalam kode setidaknya diuji minimal satu kali. Pada halaman *login* ini dilakukannya pengujian bertujuan untuk:

1. Memastikan semua validasi *input* berjalan dengan benar.
2. Menguji seluruh kemungkinan kondisi.
3. Menjamin semua laur eksekusi bekerja dengan benar.

B. Identifikasi *Statement*Tabel 5.12 Identifikasi *Statement* Halaman *Update Status*

S1	<i>if</i> (<i>selectedRuangId</i> == <i>null</i>) {
S2	<i>ScaffoldMessenger.of(context).showSnackBar</i> (<i>const SnackBar</i> (<i>content: Text</i> ("Pilih ruangan terlebih dahulu",),
S3	<i>return</i> ; }
S4	<i>if</i> (<i>status.isEmpty</i>) {
S5	<i>ScaffoldMessenger.of(context).showSnackBar</i> (<i>const SnackBar</i> (<i>content: Text</i> ("Pilih status terlebih dahulu",),
S6	<i>return</i> ; }
S7	<i>int jumlah</i> = <i>int.tryParse</i> (<i>jumlahController.text</i> ,) ?? 0;
S8	<i>if</i> (<i>jumlah</i> <= 0) {
S9	<i>ScaffoldMessenger.of(context).showSnackBar</i> (

	<i>const SnackBar(</i>
	<i>content: Text(</i>
	<i>"Jumlah tidak valid",</i>
	<i>),</i>
S10	<i>if (jumlah > totalTempatTidur) {</i>
S11	<i>ScaffoldMessenger.of(context).showSnackBar(</i>
	<i>const SnackBar(</i>
	<i>content: Text(</i>
	<i>"Jumlah melebihi total bed",</i>
	<i>),</i>
S12	<i>if (status == "perbaikan" &&</i>
	<i>keteranganController.text.trim().isEmpty) {</i>
S13	<i>ScaffoldMessenger.of(context).showSnackBar(</i>
	<i>const SnackBar(</i>
	<i>content: Text(</i>
	<i>"Keterangan perbaikan wajib diisi",</i>
	<i>),</i>
S14	<i>Return;</i>
S15	<i>try {</i>
	<i>int kosong = 0;</i>
	<i>int terisi = 0;</i>
	<i>int perbaikan = 0;</i>
S16	<i>if (status == "kosong") {</i>
S17	<i>else if (status == "terisi") {</i>

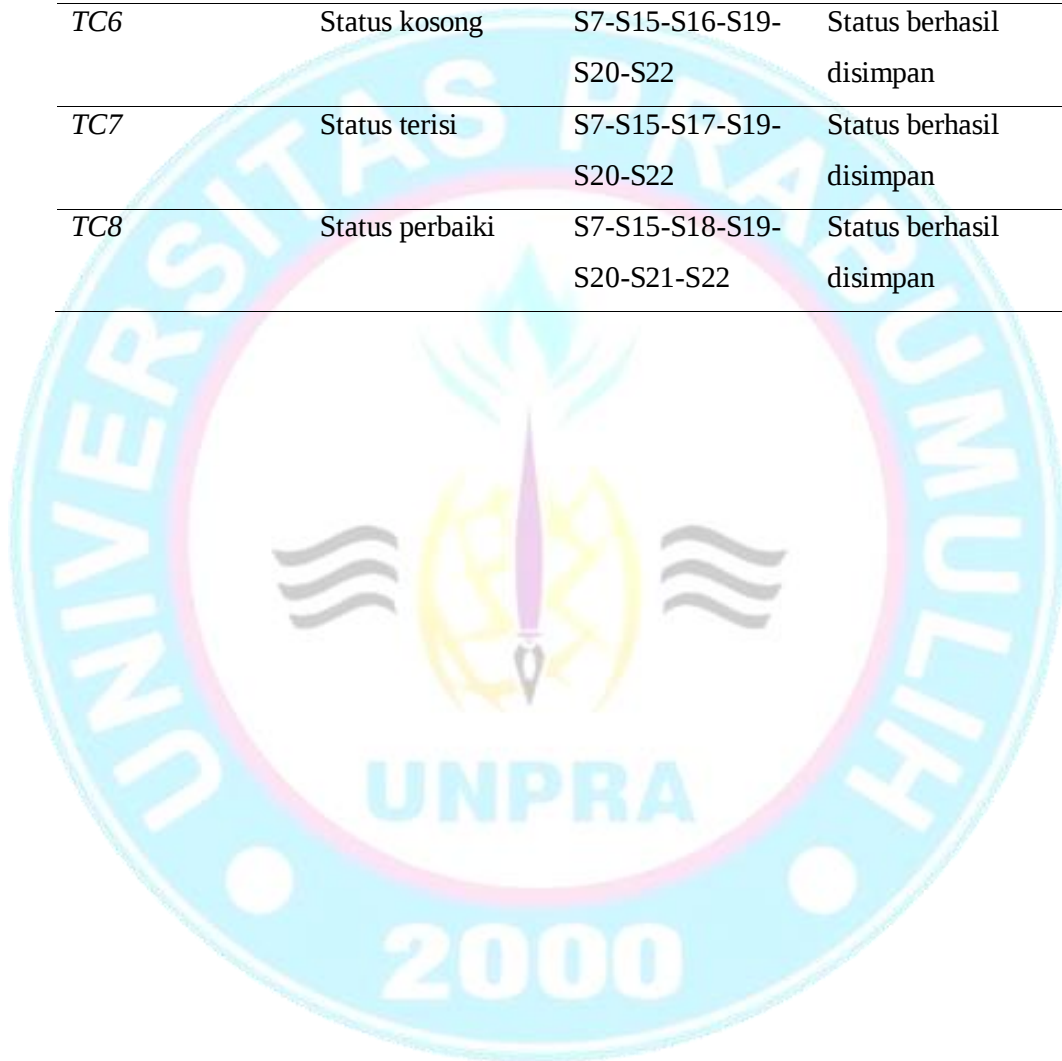
S18	<i>else if (status == "perbaikan") {</i>
S19	<i>firestore.collection("riwayat_status").add({</i>
S20	<i>DocumentSnapshot ruangDoc = await firestore</i> <i>.collection("ruang")</i> <i>.doc(selectedRuangId)</i> <i>.get());</i>
S21	<i>if (status == "perbaikan") {</i>
S22	<i>ScaffoldMessenger.of(context).showSnackBar(</i> <i>const SnackBar(</i> <i>content: Text(</i> <i>"Status berhasil disimpan",</i> <i>),</i>

C. Test Case untuk Statement Coverage

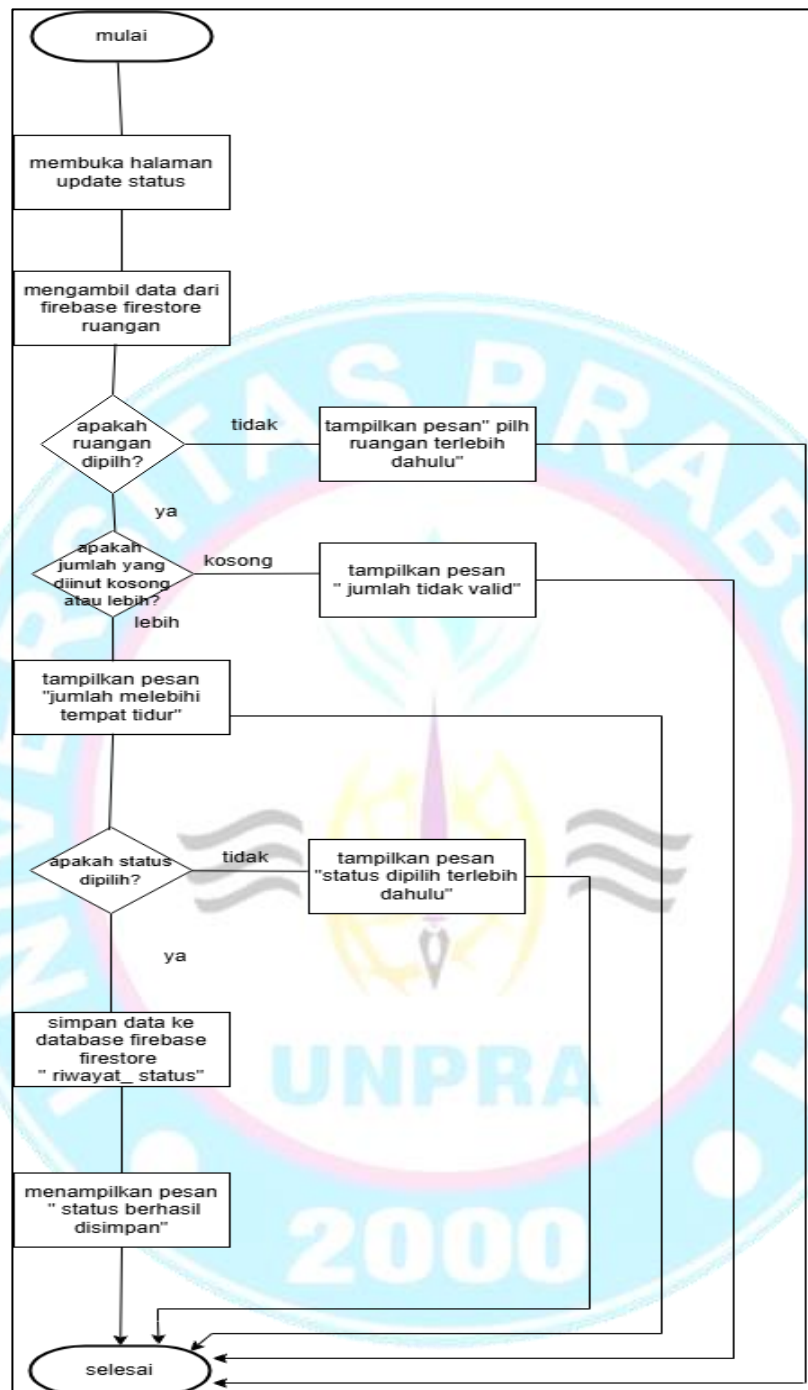
Tabel 5.13 Test Case Halaman Update Status

Test Case	Input	Jalur eksekusi	Output
TC1	Ruang belum dipilih	S1-S2-S3-S22	Pilih ruangan terlebih dahulu
TC2	Status belum dipilih	S4-S5-S6-S22	Pilih status terlebih dahulu
TC3	Ketika form jumlah yang diinput masih kosong	S7-S8-S9-S22	Jumlah tidak valid
TC4	Ketika jumlah yang diinput melebihi kapasitas	S10-S11-S22	Jumlah melebihi kapasitas

	yang dipunya di setiap ruang		
<i>TC5</i>	Ketika status yang dipilih adalah dalam perbaikan formnya masih kosong	S12-S13-S14-S22	Keterangan wajib diisi
<i>TC6</i>	Status kosong	S7-S15-S16-S19- S20-S22	Status berhasil disimpan
<i>TC7</i>	Status terisi	S7-S15-S17-S19- S20-S22	Status berhasil disimpan
<i>TC8</i>	Status perbaiki	S7-S15-S18-S19- S20-S21-S22	Status berhasil disimpan



D. Flowchart Update Status



Sumber: data diolah oleh penulis (2026)

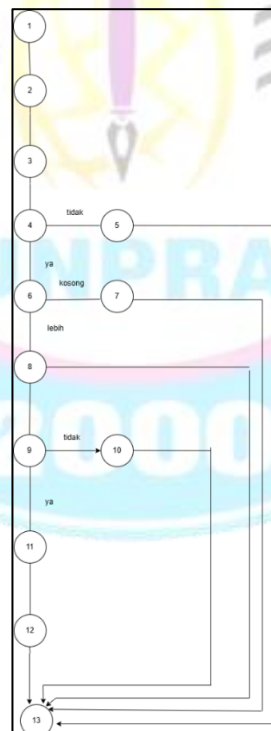
Gambar 5.21 Flowchart Halaman Update status

E. Node dan Edge

Node

1. Mulai
2. Membuka halaman *update* status
3. Mengambil data dari *firebase* ruang
4. Apakah ruangan dipilih?
5. Jika tidak, menampilkan pesan “pilih ruangan terlebih dahulu
6. Jika iya, apakah jumlah yang diinput kosong atau lebih?
7. Jika kosong, tampilkan pesan “jumlah tidak valid”
8. Jika lebih, tampilkan pesan “jumlah melebihi tempat tidur”
9. Apakah status dipilih?
10. Jika tidak, pilih status terlebih dahulu
11. Jika iya, simpen data ke *firestore* riwayat_status
12. Menampilkan pesan “status berhasil disimpan”

F. Flowgraph Update Status



Sumber: data diolah oleh penulis (2026)

Gambar 5.22 Flowgraph Halaman Update Status

G. Jalur Independen (*Path*)

Tabel 5.14 Jalur Independen Halaman *Update Status*

No	Jalur Independen	penjelasan
1	1-2-3-4-5-13	Jalur yang menunjukkan kondisi ketika ruangan belum dipilih.
2	1-2-3-4-6-7-13	Jalur yang menunjukkan kondisi jumlah tempat tidur belum diisi.
3	1-2-3-4-6-8-13	Jalur yang menunjukkan kondisijumlah tempat tidur yang diisi melebihi kapasitas tempat tidur di satu ruangan.
4	1-2-3-4-6-8-9-10-13	Jalur yang menunjukkan kondisi ketika status belum dipilih.
5	1-2-3-4-6-8-9-11-12-13	Jalur yang menunjukkan kondisi bahwa status berhasil dilakukan.

5. Pengujian ubah *password*

Berikut ini adalah kodingan ubah *password* yang akan diuji menggunakan *whitebox testing*

```
if (passwordBaru.isEmpty || konfirmasiPassword.isEmpty) {
    ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(content: Text("Password tidak boleh kosong")),
    );
    return;
}
```

```
if (passwordBaru != konfirmasiPassword) {
    ScaffoldMessenger.of(context).showSnackBar(
```

```

const SnackBar(
  content: Text("Password baru dan konfirmasi tidak sama"),
),
);

return;
}

if (!validasiPassword(passwordBaru)) {
  ScaffoldMessenger.of(context).showSnackBar(
    const SnackBar(
      content: Text(
        "Password minimal 8 karakter dengan huruf besar, kecil, angka,
simbol",
      ),
    ),
  );
  return;
}

try {
  setState(() => isLoading = true);

  User? user = auth.currentUser;

```

```

    if (user == null) {

        throw FirebaseAuthException(code: "user-not-found");

    }

```

```

    await user.updatePassword(passwordBaru);

```

```

    DocumentSnapshot userData = await FirebaseFirestore.instance

```

```

        .collection("pengguna")

```

```

        .doc(user.uid)

```

```

        .get();

```

```

    String username = userData["username"] ?? "Tidak diketahui";

```

```

    await FirebaseFirestore.instance.collection("riwayat_password").add({

```

```

        "uid": user.uid,

```

```

        "username": username,

```

```

        "passwordBaru": passwordBaru,

```

```

        "konfirmasiPasswordBaru": konfirmasiPassword,

```

```

        "waktu": Timestamp.now(),

```

```

    });

```

```

    _passwordBaruController.clear();

```

```

    _konfirmasipasswordBaruController.clear();

```

```

ScaffoldMessenger.of(context).showSnackBar(
    const SnackBar(content: Text("Password berhasil diubah")),
);

Navigator.pop(context);
} catch (e) {
    ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(content: Text("Error: $e")),
    );
} finally {
    setState(() => isLoading = false);
}
}

```

A. Tujuan dilakukannya *Statement Coverage* pada Halaman Ubah *Password*

Statement Coverage dilakukan untuk memastikan bahwa setiap baris pernyataan dalam kode setidaknya diuji minimal satu kali. Pada halaman login ini dilakukannya pengujian bertujuan untuk:

1. Memastikan semua validasi *password* berjalan dengan benar.
2. Menguji setiap kondisi yang dimiliki.
3. Memastikan proses ubah *password* berhasil
4. Memastikan data riwayat perubahan *password* berhasil tersimpan dalam *firestore*.
5. Menjamin tidak ada statement yang tidak dieksekusi.

B. Identifikasi *Statement*Tabel 5.15 Identifikasi *Statement* Halaman Ubah *Password*

S1	<i>if (passwordBaru.isEmpty konfirmasiPassword.isEmpty) {</i>
S2	<i>ScaffoldMessenger.of(context).showSnackBar(</i> <i>const SnackBar(content: Text("Password tidak boleh kosong")),</i> <i>);</i>
S3	<i>return;</i> <i>}</i>
S4	<i>if (passwordBaru != konfirmasiPassword) {</i>
S5	<i>ScaffoldMessenger.of(context).showSnackBar(</i> <i>const SnackBar(</i> <i>content: Text("Password baru dan konfirmasi tidak sama"),</i> <i>),</i>
S6	<i>return;</i> <i>}</i>
S7	<i>if (!validasiPassword(passwordBaru))</i>
S8	<i>ScaffoldMessenger.of(context).showSnackBar(</i> <i>const SnackBar(</i> <i>content: Text(</i> <i>"Password minimal 8 karakter dengan huruf besar, kecil, angka, simbol",</i>

```

        ),
S9      return;

        }
S10     setState(() => isLoading = true);
S11     User? user = auth.currentUser;
S12     if (user == null) {
S13       throw FirebaseAuthException(code: "user-not-found");
        }
S14     await user.updatePassword(passwordBaru);
S15     DocumentSnapshot userData = await FirebaseFirestore.instance
           .collection("pengguna")
           .doc(user.uid)
           .get();
S16     String username = userData["username"] ?? "Tidak diketahui";
S17     await
           FirebaseFirestore.instance.collection("riwayat_password").add({
S18     ScaffoldMessenger.of(context).showSnackBar(
           const SnackBar(content: Text("Password berhasil
           diubah")),
           );
S19     Navigator.pop(context);
S20     finally {

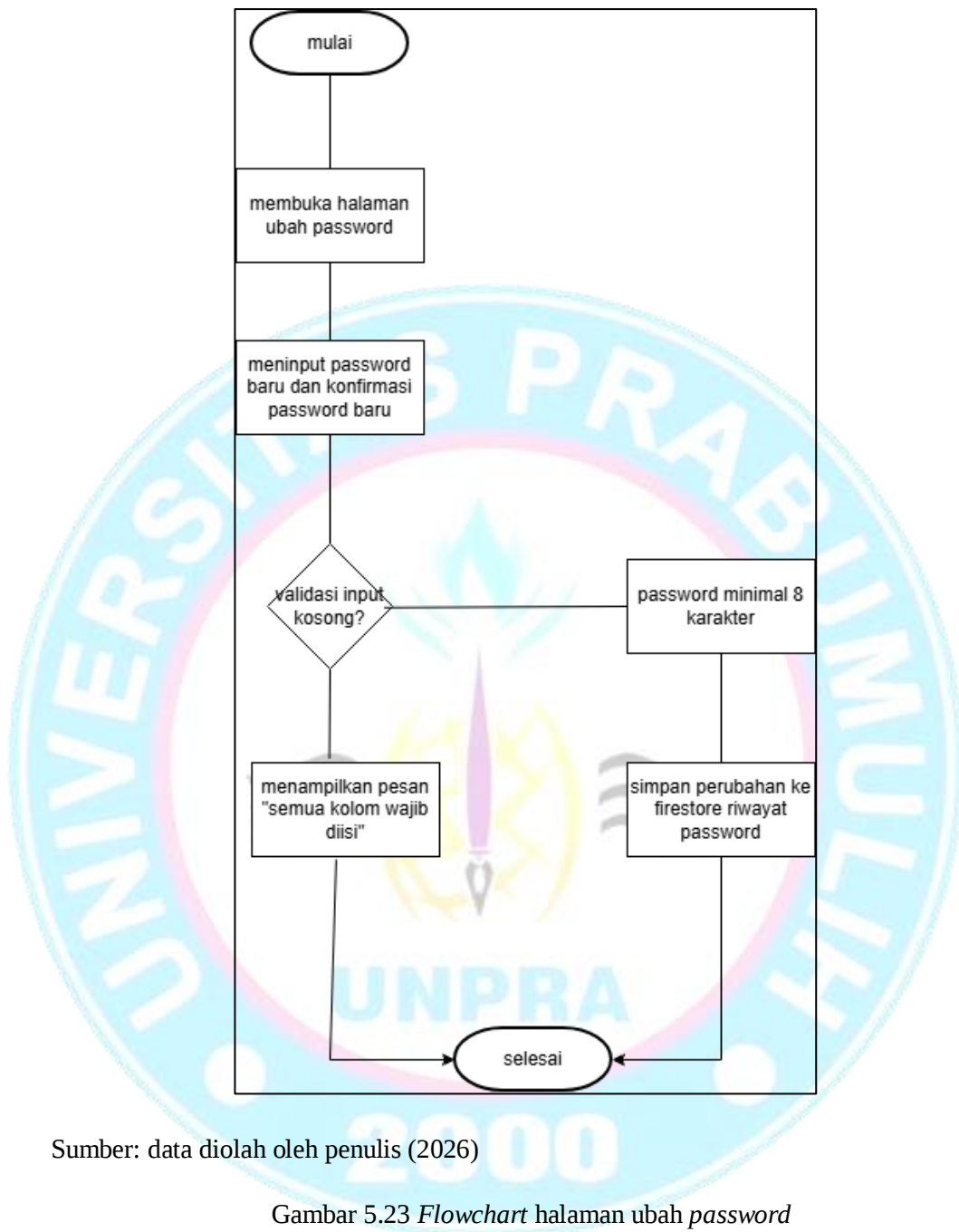
```

C. Test Case untuk Statement Coverage

Tabel 5.16 Test Case Halaman Ubah Password

<i>Test Case</i>	<i>Input</i>	<i>Jalur eksekusi</i>	<i>Output</i>
<i>TC1</i>	<i>Password kosong</i>	S1-S2-S3	<i>Password tidak boleh kosong</i>
<i>TC2</i>	<i>Password tidak sama</i>	S4-S5-S6	<i>Password baru dan konfirmasi tidak sama</i>
<i>TC3</i>	<i>Password lemah</i>	S7-S8-S9	<i>Password minimal 8 karakter</i>
<i>TC4</i>	<i>User belum login</i>	S10-S11-S12-S13-S16-S20	<i>Pengguna tidak ditemukan</i>
<i>TC5</i>	<i>Ubah password berhasil</i>	S10-S11-S14-S15-S17-S18-S19-S20	<i>Password baru berhasil dirubah</i>

D. Flowchart Ubah Password



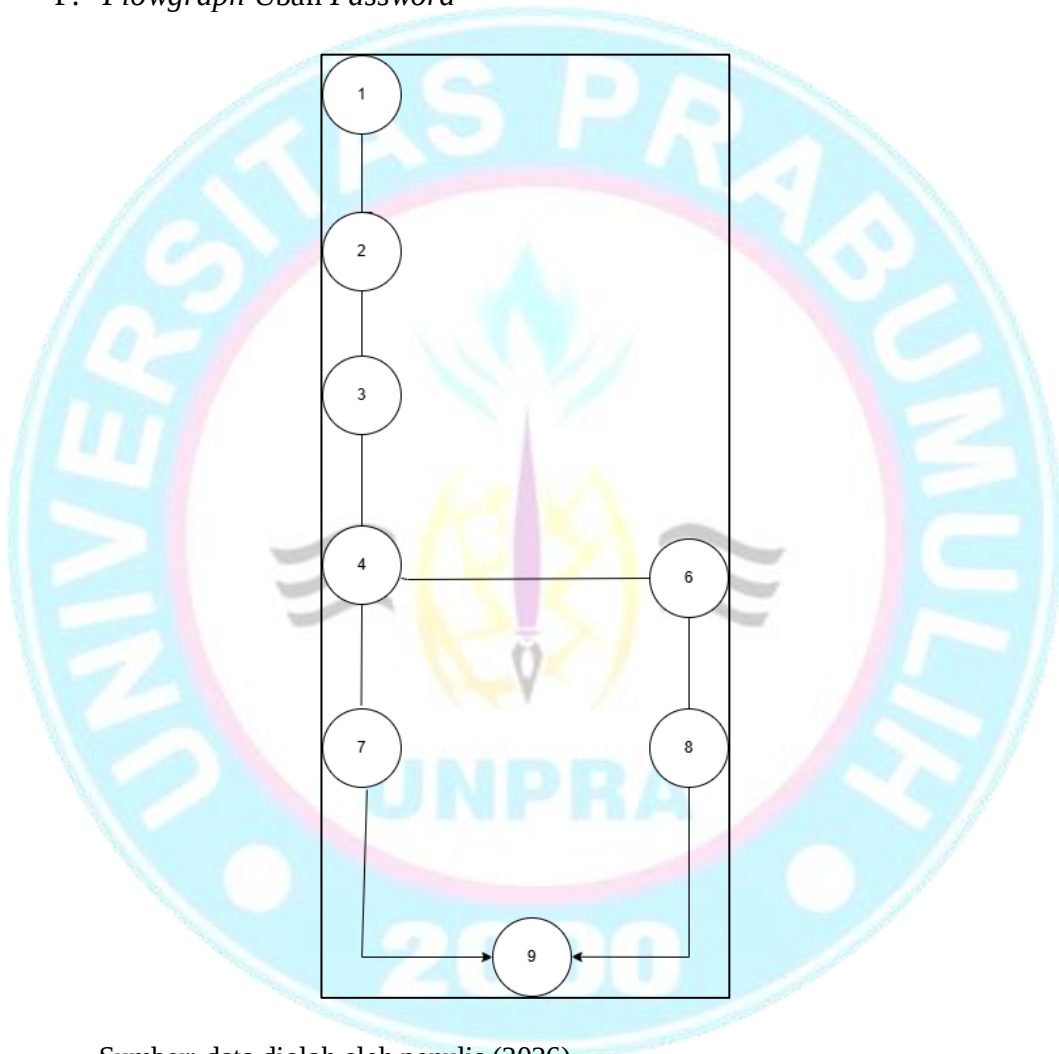
E. Node dan Edge

Node

1. Mulai
2. Membuka halaman ubah password
3. Validasi input kosong?

4. Jika iya, password minimal 8 karakter
5. Jika tidak, menampilkan pesan semua kolom wajib diisi
6. Simpan perubahan ke firestore riwayat password
7. Selesai

F. *Flowgraph* Ubah Password



Sumber: data diolah oleh penulis (2026)

Gambar 5.24 *Flowgraph* Halaman Ubah Password

G. Jalur Independen (*Path*)

Tabel 5.17 Jalur Independen Halaman Ubah Password

No	Jalur Independen	penjelasan
1	1-2-3-4-6-7-8	Jalur yang menunjukkan kondisi form input belum diisi.
2	1-2-3-4-5-7-8	Jalur yang menunjukkan kondisi form input password minimal 8 karakter.



BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan mengenai implementasi metode *Rapid Application Development (RAD)* dalam perancangan aplikasi Android manajemen ketersediaan tempat tidur pasien di RSUD Prabumulih, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Penelitian ini berhasil merancang dan membangun aplikasi *Android* manajemen ketersediaan tempat tidur pasien yang dapat membantu pihak RSUD Prabumulih dalam mengelola serta memantau informasi ketersediaan tempat tidur secara lebih cepat dan akurat.
2. Penerapan metode *Rapid Application Development (RAD)* pada proses pengembangan aplikasi dapat membantu mempercepat proses pembuatan sistem karena melibatkan pengguna secara langsung dalam tahap perencanaan, desain, pengembangan, dan pengujian sistem sehingga aplikasi yang dihasilkan lebih sesuai dengan kebutuhan pengguna.
3. Aplikasi yang dibangun mampu memberikan informasi ketersediaan tempat tidur secara langsung kepada petugas administrasi, perawat ruangan, dan pasien sehingga proses pengecekan tempat tidur tidak lagi dilakukan secara manual melalui komunikasi telepon antar ruangan.

6.2 Saran

Adapun beberapa saran yang dapat digunakan untuk pengembangan aplikasi di masa mendatang adalah sebagai berikut:

1. Pengembangan aplikasi dapat ditambahkan dengan fitur *booking* tempat tidur pasien sehingga proses pemesanan kamar rawat inap dapat dilakukan secara lebih mudah dan terstruktur.

2. Sistem dapat dikembangkan dengan fitur pemindahan kamar pasien untuk mempermudah pengelolaan perpindahan pasien antar ruangan sesuai kebutuhan pelayanan rumah sakit.
3. Tampilan antarmuka aplikasi dapat terus dikembangkan agar menjadi lebih menarik, interaktif, dan mudah digunakan oleh seluruh pengguna sistem.



DAFTAR PUSTAKA

- Al-Faruq, M. N. M. (2022). Perancangan UI/UX Semarang Virtual Tourism dengan Figma. *Walisongo Journal of Information Technology*, 4(1), 43–52.
- Aldi, M. (2025). Perancangan Sistem Digital Surat Masuk Dan Surat. *Jurnal Multidisiplin Ilmu Akademik*, 2(2), 65–77.
- Bahat Nauli, S. (2024). *Perancangan Aplikasi Untuk Mengelola Asset Pada PT Transmarco*.
- Dalis, S. (2024). Model Rapid Application Development (RAD) Untuk Perancangan Sistem Informasi Pengelolaan Surat (SIPERA) Pada Kelurahan. In *IMTechno: Journal of Industrial Management and Technology* (Vol. 5, Issue 1). <http://jurnal.bsi.ac.id/index.php/imtechno>
- Fiorenza. (2023). Pengembangan Aplikasi Mobile sebagai Media Edukasi Kesehatan Gigi menggunakan Teknologi Firebase serta Metode Prototyping (Studi Kasus RSGM Universitas Brawijaya). *Pengembangan Aplikasi Mobile Sebagai Media Edukasi Kesehatan Gigi Menggunakan Teknologi Firebase Serta Metode Prototyping (Studi Kasus RSGM Universitas Brawijaya)*, 7(1), 258–266. <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/12161>
- Fransiscus, B. P. (2022). Perancangan dan impementasi aplikasi android untuk layanan informasi pariwisata. *Journal Penerapan Teknologi Inforamasi Dan Komunikasi*, 1(2), 114–132.
- Habsy, B. (2024). Analisis Konsep Dasar Organisasi dan Hubungannya dengan Manajemen. In *Journal of Comprehensive Science* (Vol. 3, Issue 12).
- Hertantyo, G. (2021). Development Of An Android Application-Based Academic Information System At Immigration Polytechnic Pengembangan Sistem Informasi Akademik Berbasis Aplikasi Android Pada Politeknik Imigrasi. *Tematics: Technology Management and Informatics Research Journals*, 3(2), 23–53. <https://doi.org/10.52617/tematics.v3i2.331>
- Hidayati, L. (2024). Implementation of the Prototype Method in Designing an Android-Based Pramubakti Attendance Application with Geolocation

- Features. *Jurnal IPTEK Bagi Masyarakat*, 3(3), 92–106.
- Jamjani, D. W. (2024). Sistem Aplikasi Tabungan Siswa pada SMP Taruna Terpadu Berbasis Android dan Website. *Jurnal Rekayasa Komputasi Terapan*, 4(04), 346–352. <https://doi.org/10.30998/jrkt.v4i04.11136>
- Kalsum. (2024). Implementasi Framework Flutter untuk Chatbot Informasi Sistem Berbasis Aplikasi Mobile. *INNOVATIVE: Journal Of Social Science Research*, 4(2), 6272–6286. <https://j-innovative.org/index.php/Innovative>
- Manurung, K. (2021). Mencermati Penggunaan Metode Kualitatif Di Lingkungan Sekolah Tinggi Teologi. *FILADELFIA: Jurnal Teologi Dan Pendidikan Kristen*, 3(1), 285–300. <https://doi.org/10.55772/filadelphia.v3i1.48>
- Martani, A. (2023). Implementasi global positioning system GPS pada kendaraan menggunakan mikrotik lt ap mini. *Jurnal Satya Informatika*, 8(2).
- Narulita, S., Nugroho, A., & Abdillah, M. Z. (2024). Diagram Unified Modelling Language (UML) untuk Perancangan Sistem Informasi Manajemen Penelitian dan Pengabdian Masyarakat (SIMLITABMAS) Universitas Nasional Karangturi Semarang , Indonesia (deskripsi) dan perancangan sistem , khususnya pada pemrogr. *Jurnal Publikasi Sistem Informasi Dan Telekomunikasi*, 2(3), 244–256. <https://doi.org/10.62951/bridge.v2i3.174%0AAvailable>
- Nasution, A. S. (2024). Perancangan Aplikasi Pengajuan Cuti Karyawan Berbasis Web di Pusat Dukungan Operasional Layanan Pendapatan Daerah Pemerintah Provinsi Sumatera Utara. *Jurnal Ilmu Komputer Dan Bisnis*, 15(1), 8–19. <https://doi.org/10.47927/jikb.v15i1.691>
- Pujilestari, I. (2023). PERANCANGAN SISTEM INFORMASI TEMPAT TIDUR BERBASIS WEB DI RUMAH SAKIT TNI. AU. DR. M. SALAMUN. In *Novita LA TEDC* (Vol. 17, Issue 2).
- Putri. (2024). Analisis Dan Perancangan Aplikasi Manajemen Proyek Berbasis Mobile dan Web. *Jurnal SIFO Mikroskil*, 25(2), 39–54. <https://doi.org/10.55601/jsm.v25i2.1111>

- Putri, P. E., & Khairunnisa, K. (2025). Perancangan data base sistem pembelajaran sekolah dasar menggunakan ERD. *Jurnal Sistem Informasi (TEKNOFILE)*, 3(5), 314–326.
- Sinta Sani, R., Wulandari, S., Sukmaningsih, W. R., Surakarta, P. I., & Samanhudi, J. K. H. (2022). Analisis Kebutuhan Tempat Tidur Berdasarkan Standar Depkes Di Rumah Sakit PKU Muhammadiyah Surakarta. *Journal Health Information Management Indonesian (JHIMI) Jurnal JHIMI*, 01(02).
- Suharni. (2023). Perancangan Website Rumah Makan Ninik Sebagai Media Promosi Menggunakan UML. *Jurnal Rekayasa Informasi*, 12(1), 1–12.
- Suhendro, J. M. (2021). Rancang Bangun Aplikasi Seluler Penyedia Jasa Perawatan dan Kecantikan Menggunakan Framework Flutter. *Jurnal Spektrum*, 8(2), 68–82.
- Sulistya, Y. I. (2025). *Buku Pemrograman Dart Panduan Cepat Untuk Pemula*.
- Syahputra, F., Sabrina, E., Mediansyah, T., Armando, G., Sinaga, A., & PeranginAngin, Y. (2024). Meta Analisis: Pengembangan Media Pembelajaran Dasar-Dasar Pemrograman Berbasis Desktop Fahmy. *AR RUMMAN - Journal of Education and Learning Evaluation*, 1(2).
- Voni Besin, M. (2023). Implementasi Metode Rapid Application Development (RAD) dalam Membangun Aplikasi Boarding House di Sekitar Kampus Unimor Berbasis Android. *Journal Restikom ; Riset Teknik Informatika Dan Komputer*, 5(3), 412–423. <https://restikom.nusaputra.ac.id>
- Waruwu, M. (2023). Pendekatan Penelitian Pendidikan: Metode Penelitian Kualitatif, Metode Penelitian Kuantitatif dan Metode Penelitian Kombinasi (Mixed Method). *Jurnal Pendidikan Tambusai*, 7(1), 2896–2910. <https://doi.org/10.36706/jbti.v9i2.18333>
- Wayahdi, M, R., & Ruziq, F. (2023). Pemodelan Sistem Penerimaan Anggota Baru dengan Unified Modeling Language (UML) (Studi Kasus : Programmer Association of Battuta). *Jurnal Minfo Polgan Volume*, 12(1), 1514–1521. <https://doi.org/10.62951/bridge.v2i3.174%0AAvailable>

Wintana, D., & Pribadi, D. (2022). Analisis Perbandingan Efektifitas White-Box Testing dan Black-Box Testing. *Jurnal Larik: Ladang Artikel Ilmu Komputer*, 2(1), 8–16. <https://doi.org/10.31294/larik.v2i1.1382>



LAMPIRAN

Lampiran 1 Surat Izin Penelitian



**FAKULTAS ILMU KOMPUTER
UNIVERSITAS PRABUMULIH**

Prabumulih, 07 November 2025

Nomor : 036 /FASILKOM/UNPRA/Inf-S1/XI/2025
 Lampiran : -
 Perihal : Permohonan Penelitian dan Pengambilan Data

Kepada Yth,
 Pimpinan Rumah Sakit Umum Daerah (RSUD) Prabumulih
 Di
 Tempat

Dalam rangka pelaksanaan skripsi mahasiswa Program Studi Informatika Fakultas Ilmu Komputer Universitas Prabumulih, maka kami harapkan bantuan dari Bpk/Ibu kiranya mengizinkan mahasiswa:

Nama : Belaazahwa
 Nim : 2022230029
 Program Studi : Informatika
 Judul : Implementasi Metode RAD dalam perancangan Aplikasi *Android* Manajemen Ketersediaan Tempat Tidur Pasien

Untuk melaksanakan penelitian dan pengambilan data ditempat Bpk/Ibu Pimpin guna menyelesaikan skripsi. Atas bantuan dan kerjasamanya kami ucapkan terima kasih.

Ketua Program Studi,
 Informatika,

Iwan Setiawan, S.Kom., M.Kom.
 NIDN.0209129101

Lampiran 2 Surat Balasan Izin Penelitian



PEMERINTAH KOTA PRABUMULIH
RUMAH SAKIT UMUM DAERAH
KOTA PRABUMULIH

Jln. Lingkar Kel. Gunung Ibul Prabumulih Timur 31113
Telp : (0713) 3300400, Fax : (0713) 3300402 / (0713) 3300404
Website : www.rsudprabumulih.co.id e-mail : office@rsudprabumulih.co.id

Prabumulih, 12 November 2025

Nomor : 800/ 7440 /RSUD-PBM/XI/2025
Lampiran : 1 (satu) berkas
Perihal : Izin Penelitian dan Pengambilan Data

Yth. Ketua Program Studi Informatika
Universitas Prabumulih
di
Tempat

Menindaklanjuti surat Ketua Program Studi Informatika Universitas Prabumulih Nomor : 036/FASILKOM/UNPRA/Inf-S1/XI/2025 Tanggal 07 November 2025, Perihal Permohonan Penelitian dan Pengambilan Data untuk Skripsi Mahasiswa Program Studi Informatika Fakultas Ilmu Komputer Universitas Prabumulih, atas nama:

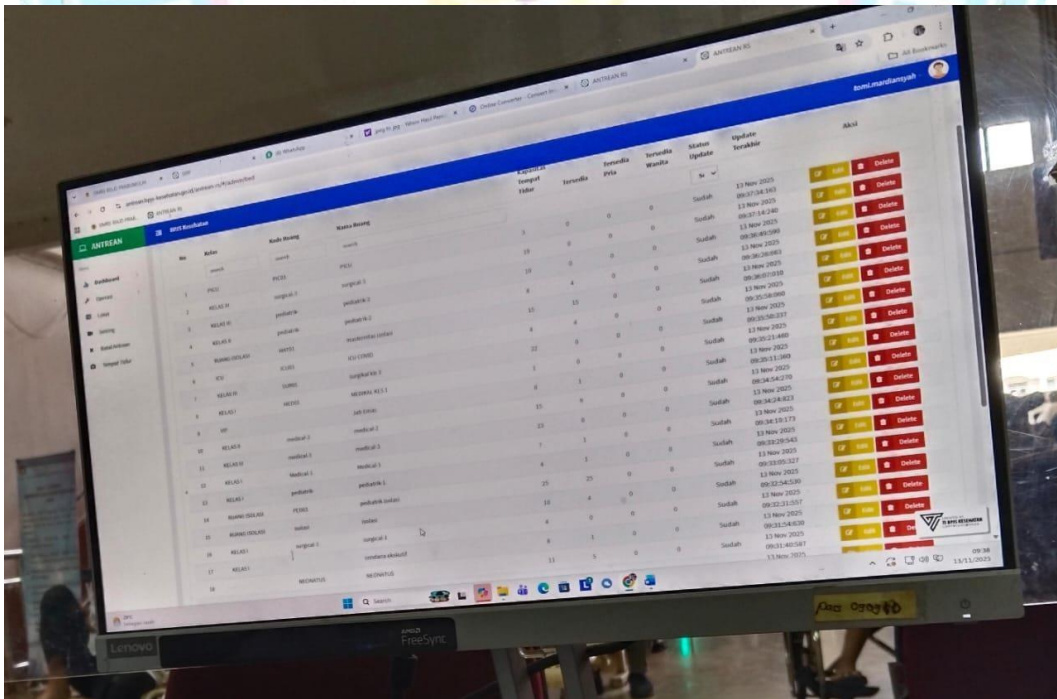
Nama Mahasiswa	: Belaazahwa
NIM	: 2022230029
Program Studi	: Informatika
Judul	: Implementasi Metode RAD dalam perancangan Aplikasi <i>Android</i> Manajemen Ketersediaan Tempat Tidur Pasien

Untuk itu Rumah Sakit Umum Daerah Kota Prabumulih **Memberikan Izin** Mahasiswa tersebut untuk melakukan Pengambilan Data di RSUD Kota Prabumulih dengan catatan **Tidak Mengganggu Pelayanan** di RSUD Kota Prabumulih selama kegiatan tersebut berlangsung.

Demikianlah, atas perhatian dan kerjasamanya diucapkan terima kasih..


drg Sriwidiastuti
Pembina Tk. I / IV b
NIP. 197407272009032001

Lampiran 3 Dokumentasi Wawancara



Lampiran 4 Draft Wawancara

Draft Pertanyaan Wawancara

Nama Narasumber : Dian Anugrahini, SKM

Jabatan : Staf Administrasi RSUD Prabumulih

Tempat Wawancara : Tempat Administrasi RSUD

Prabumulih Tanggal Wawancara : 13 November 2025

Nama Pewawancara ; Belaazahwa

No	Pertanyaan	Jawaban Responden
1	Bagaimana proses saat ini dalam mengelola data ketersediaan tempat tidur pasien di ruang rawat inap ?	Proses pengelolaan data tempat tidur pasien saat ini sudah memiliki sistem yang terintegrasi dengan BPJS namun komunikasi yang dilakukan untuk memastikan informasi tentang ketersediaan tempat tidur di ruangan rawat inap masih dilakukan dengan menggunakan telepon ke antar bagian.
	Berapa jumlah ruangan berdasarkan kelas disini?	Baiklah, disini terdiri dari beberapa kelas perawatan ada kelas 1 yang berjumlah 5 Kelas 2 berjumlah 3 Kelas 3 berjumlah 4 VIP berjumlah 1 PICU berjumlah 2 Ruang Isolasi berjumlah 3 ICU berjumlah 2 Ruang Bersalin berjumlah 1 NICU berjumlah 1 Neonatus berjumlah 1

2	Masalah atau kendala apa yang sering terjadi dalam proses pengecekan data ketersediaan tempat tidur pasien?	Kendala yang sering terjadi adalah karena komunikasi nya masih menggunakan telepon sehingga keterlambatan Dalam pnyampaian informasi yang bisa memakan waktu lama karena petugas harus melakukan pengecekan keruangan untuk memastikan ketersediaan tempat tidur. Terkadang juga tidak ada petugas yang <i>standby</i> di ruangan perawat sehingga tidak ada yg menjawab telpon.
3	Kapan biasanya masalah tersebut paling sering muncul dan bagian mana yang paling berdampak?.	Masalah biasanya muncul disaat jumlah pasien yang semakin padat. Dampaknya bisa terlihat pada bagian pelayanan karena update tentang ketersediaan tempat tidur tidak dapat dilakukan secara cepat.
4	Menurut ibu, apa penyebab utama dari masalah- masalah yang terjadi?.	Penyebab utamanya karena kurangnya bridging dan masih menggunakan telepon antar ruang untuk mengetahui kondisi yang sebenarnya.
5	Bagaimana dampaknya terhadap pelayanan pasien?.	Dampak nya terhadap pasien karena masih menggunakan telepon jadi jika ada sedikit keterlambatan pasien harus menunggu lebih lama untuk mengetahui kondisi kamar yang tersedia.
6	Apakah jika dibuatkan sebuah sistem aplikasi android bisa membantu pelayanan rumah sakit?.	Ya, pasti bisa membantu karena jika ada aplikasi android data ketersediaan tempat tidur langsung

		terlihat, penempatan pasien jadi jauh lebih cepat dan mengurangi komunikasi menggunakan telepon.
7	Bolehkah saya meminta ketersediaan ibu untuk memberikan informasi tentang data ruangan rawat inap seperti Kelas layanan dan jumlah tempat tidur di ruang rawat inap yang tersedia disini untuk keperluan pembuatan Aplikasi tersebut?	Ya, tentu saja boleh, kami bersedia membantu dan memberikan informasi yang di butuhkan terkait penelitian ini.
8	sebagai pertimbangan saya dalam perancangan aplikasi nya, bisakah ibu memberi saran untuk saya pembuatan aplikasi ini nanti, harapan ibu fitur apa yang paling dibutuhkan untuk membantu pelayanan informasi ketersediaan tempat tidur?.	Kalau menurut saya Fitur yang kami butuhkan untuk saat ini yaitu dimana aplikasi nya nanti bisa membantu kami dalam update ketersediaan langsung terlihat, dan notifikasi otomatis jika adanya perubahan
9	jika rumah sakit telah memiliki aplikasi untuk melihat ketersediaan tempat tidur, menurut ibu aplikasi ini boleh digunakan siapa saja?	Selain administrasi dan perawat sebaiknya calon pasien juga mendapatkan akses untuk menggunakan aplikasi agar mereka bisa melihat nya mellaui aplikasi tanpa harus datang dan bertanya langsung ke pihak rumah sakit.
10	Bagaimana melatih pengguna agar terbiasa dan cepat menggunakan aplikasi ini?.	Menurut saya calon pengguna bisa dilatih dengan pelatihan singkat tentang bagaimana aplikasi bekerja dan juga menggunakan bantuan tim IT agar calon pengguna cepat terbiasa dan aplikasi bisa digunakan dengan benar.



Dian Anugrahini, SKM