

SKRIPSI

IMPLEMENTASI *ROLE-BASED ACCESS CONTROL* PADA SISTEM INFORMASI MANAJEMEN STOK OBAT BERBASIS WEBSITE

*Sebagai Salah Satu Syarat Untuk Menyelesaikan
Program Studi Sarjana Terapan Keamanan Sistem Informasi*



Oleh:

DEA AGUSTINA PURBA
6404221110

**JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI BENGKALIS
TAHUN 2026**

HALAMAN PENGESAHAN

SKRIPSI

IMPLEMENTASI *ROLE-BASED ACCESS CONTROL* PADA SISTEM INFORMASI MANAJEMEN STOK OBAT BERBASIS WEBSITE

Oleh:

DEA AGUSTINA PURBA
6404221110

Telah diujikan dan dinyatakan lulus ujian skripsi pada tanggal 23 Februari 2026
oleh tim penguji Program Studi Sarjana Terapan Keamanan Sistem Informasi

Pembimbing Utama

Bengkalis, 23 Februari 2026
Anggota Tim Penguji



Nurmi Hidayasari, ST., M.Kom
NIP. 199109012022032006



Kasmawi, S.Kom., M.Kom
NIP. 197706072014041001



Rezki Kurniati, S.Kom., M.Kom
NIP. 198306162018032001

Mengetahui,

Ketua Jurusan Teknik Informatika



Kasmawi, S.Kom., M.Kom
NIP. 197706072014041001

PERNYATAAN KEASLIAN SKRIPSI

Saya menyatakan dengan sesungguhnya bahwa Laporan Skripsi ini adalah asli hasil karya saya dan tidak terdapat karya yang pernah dilakukan untuk memperoleh gelar kesarjanaan di perguruan tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau dipublikasikan oleh orang lain, kecuali yang secara tertulis disebutkan sumbernya dalam naskah dan dalam daftar pustaka.

Bengkalis, 23 Februari 2026



Dea Agustina Purba
6404221110

IMPLEMENTASI *ROLE-BASED ACCESS CONTROL* PADA SISTEM INFORMASI MANAJEMEN STOK OBAT BERBASIS WEBSITE

Nama : Dea Agustina Purba

NIM : 6404221110

Dosen Pembimbing : Nurmi Hidayasari, ST., M.Kom

ABSTRAK

Role-Based Access Control (RBAC) merupakan mekanisme pengendalian akses berbasis peran yang membatasi hak pengguna sesuai kewenangannya dalam sistem informasi. Penerapan mekanisme ini menjadi penting pada sistem pengelolaan stok obat di apotek, khususnya di Apotek Pratama dr. Moris, yang hingga saat ini masih melakukan pencatatan stok secara manual. Kondisi tersebut menyebabkan sering terjadinya ketidaksesuaian data serta keterlambatan dalam pemantauan obat kedaluwarsa, kesulitan dalam penyusunan laporan, serta tidak adanya pembatasan hak akses yang jelas bagi setiap pengguna. Penelitian ini difokuskan pada pengembangan sebuah sistem informasi manajemen stok obat yang diimplementasikan dalam bentuk aplikasi web dengan menerapkan RBAC sebagai mekanisme utama dalam pengamanan dan otorisasi pengguna. Pengembangan sistem dilakukan menggunakan metode *Waterfall* yang mencakup analisis kebutuhan, perancangan, implementasi, serta pengujian. Sistem ini dikembangkan dengan framework Laravel versi 11, basis data MySQL, serta *package Laravel Spatie Permission* untuk pengelolaan *role* dan *permission*. Hasil penelitian menunjukkan bahwa penerapan RBAC mampu membatasi akses pengguna sesuai dengan perannya, meningkatkan keamanan data, serta mendukung akurasi pencatatan dan efisiensi pengelolaan stok obat dibandingkan dengan sistem manual sebelumnya.

Kata kunci: apotek, keamanan data, manajemen stok obat, RBAC, sistem informasi

IMPLEMENTATION OF ROLE-BASED ACCESS CONTROL IN A WEBSITE-BASED MEDICINE STOCK MANAGEMENT INFORMATION SYSTEM

Name : Dea Agustina Purba
Student ID : 6404221110
Supervisor : Nurmi Hidayasari, ST., M.Kom

ABSTRACT

Role- Based Access Control (RBAC) is an access control mechanism that restricts user privileges based on assigned roles within an information system. The implementation of this mechanism is essential for drug inventory management systems in pharmacies, particularly at Apotek Pratama dr. Moris, which previously relied on manual stock recording. This condition led to frequent data inconsistencies, delays in monitoring expired drugs, difficulties in report preparation, and the absence of clear access restrictions for each user. This study focuses on the development of a web-based drug inventory management information system that applies RBAC as the main mechanism for user security and authorization. The system was developed using the Waterfall method, which includes requirement analysis, system design, implementation, and testing. The system was built using the Laravel 11 framework, MySQL database, and the Laravel Spatie Permission package for managing roles and permissions. The results show that the implementation of RBAC effectively restricts user access according to their roles, enhances data security, and improves the accuracy and efficiency of drug inventory management compared to the previous manual system.

Keywords: *pharmacy, data security, drug inventory management, RBAC, information system*

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Tuhan Yang Maha Esa atas segala rahmat, dan anugrah-Nya sehingga penulis dapat menyelesaikan penyusunan skripsi yang berjudul “Implementasi *Role-Based Access Control* pada Sistem Informasi Manajemen Stok Obat Berbasis Website” ini dengan baik. Skripsi ini disusun sebagai salah satu syarat untuk menyelesaikan Program Studi Sarjana Terapan Keamanan Sistem Informasi, Jurusan Teknik Informatika, Politeknik Negeri Bengkalis.

Dalam proses penyusunan skripsi ini, penulis menyadari bahwa tanpa bantuan, bimbingan, dan dukungan dari berbagai pihak, penyusunan laporan ini tidak akan dapat terselesaikan dengan baik. Oleh karena itu, pada kesempatan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Johny Custer, S.T., M.T. selaku Direktur Politeknik Negeri Bengkalis.
2. Bapak Kasmawi, S.Kom., M.Kom. selaku Ketua Jurusan Teknik Informatika.
3. Ibu Nurmi Hidayasari, ST., M.Kom. selaku Ketua Program Studi Sarjana Terapan Keamanan Sistem Informasi sekaligus pembimbing utama yang telah memberikan arahan, bimbingan, motivasi, serta meluangkan waktu dalam proses penyusunan skripsi ini.
4. Bapak Kasmawi, S.Kom., M.Kom., dan Ibu Rezki Kurniati, S.Kom., M.Kom., selaku penguji yang telah memberikan kritik dan saran yang membangun demi penyempurnaan skripsi ini.
5. Seluruh dosen dan staf Jurusan Teknik Informatika Politeknik Negeri Bengkalis yang telah memberikan ilmu, pengalaman, serta pelayanan terbaik selama masa perkuliahan.
6. Pimpinan dan staf Apotek Pratama dr. Moris yang telah memberikan kesempatan, data, serta kerja sama sehingga penelitian dan implementasi sistem dapat dilakukan dengan baik.
7. Kedua orang tua tercinta, Bapak Benhur Purba & Mama Mariana Hutapea

8. serta seluruh keluarga besar secara khusus kepada Kakakku, Ida Lestari Purba, Abangku, Firman Syahpurta Purba dan Adik-adikku, Daniel Purba, Yehezkiel Purba dan Luga Fransisco Purba yang selalu mendoakan, memberikan dukungan moril maupun materiil, serta menjadi sumber semangat bagi penulis.
9. Sahabat-sahabat dan rekan seperjuangan di Program Studi Sarjana Terapan Keamanan Sistem Informasi yang selalu menemani, membantu, dan memberi dukungan selama proses perkuliahan hingga penyusunan skripsi.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna, baik dari segi isi maupun penyajian, karena keterbatasan pengetahuan dan pengalaman. Oleh sebab itu, penulis sangat mengharapkan kritik dan saran yang bersifat membangun demi perbaikan di masa mendatang.

Akhir kata, penulis berharap semoga skripsi ini dapat memberikan manfaat bagi pembaca, menjadi tambahan referensi bagi pengembangan sistem informasi terutama yang berkaitan dengan keamanan akses menggunakan Role-Based Access Control (RBAC), serta dapat memberikan kontribusi positif bagi Apotek Pratama dr. Moris dan pihak-pihak lain yang berkepentingan.

Bengkalis, 23 Februari 2026



Dea Agustina Purba

6404221110

DAFTAR ISI

	Halaman
HALAMAN PENGESAHAN.....	ii
HALAMAN PERNYATAAN KEASLIAN SKRIPSI.....	ii
ABSTRAK	iii
<i>ABSTRAK</i>.....	<i>iv</i>
KATA PENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR TABEL	ix
DAFTAR GAMBAR.....	x
DAFTAR LAMPIRAN	xii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	3
1.3 Batasan Masalah.....	4
1.4 Tujuan.....	5
1.5 Manfaat.....	6
BAB II TINJAUAN PUSTAKA.....	7
2.1 Kajian Terdahulu.....	7
2.2 Landasan Teori	12
2.1.1 Sistem Informasi	12
2.1.2 Manajemen Stok Obat	14
2.1.3 Website dan Laravel	15

2.1.4 Keamanan: <i>Role-Based Access Control</i> (RBAC).....	16
BAB III METODE PENELITIAN.....	21
3.1 Data dan Alat Penelitian.....	21
3.2 Prosedur Penelitian.....	23
3.3 Analisis Kebutuhan	23
3.4 Perancangan	25
BAB IV HASIL DAN PENGUJIAN.....	54
4.1 Hasil Implementasi.....	54
4.2 Pengujian.....	82
4.2.1 Deskripsi Pengujian	82
4.2.2 Prosedur Pengujian	82
4.2.3 Data Hasil Pengujian	83
4.3 Analisis Data	87
BAB V PENUTUP	90
5.1 Kesimpulan.....	90
5.2 Saran.....	91
DAFTAR PUSTAKA	92
LAMPIRAN.....	95

DAFTAR TABEL

	Halaman
Tabel 2.1 Review Berdasarkan Penelitian Terkait	10
Tabel 2.2 Penambahan Role dan Permission List Sistem	20
Tabel 3. 1 Tabel users	34
Tabel 3. 2 Tabel roles	34
Tabel 3. 3 Tabel permissions	34
Tabel 3. 4 Tabel role_has_permissions	35
Tabel 3. 5 Tabel obats	35
Tabel 3. 6 Tabel stok_obats	35
Tabel 3. 7 Tabel distributors	36
Tabel 3. 8 Tabel transaksis	36
Tabel 3. 9 Tabel notifikasi	37
Tabel 3. 10 activity_log	38
Tabel 4. 1 <i>Black Box Testing</i> – Autentikasi & RBAC	84
Tabel 4. 2 <i>Black Box Testing</i> – Modul Data Obat & Distributor	85
Tabel 4. 3 <i>Black Box Testing</i> – Transaksi Obat Masuk & Keluar	85
Tabel 4. 4 <i>Black Box Testing</i> – Notifikasi <i>Dashboard</i>	86
Tabel 4. 5 <i>Black Box Testing</i> – Laporan	86

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Model Waterfall	13
Gambar 2.2 Komponen RBAC	18
Gambar 2.3 Konsep Model RBAC	19
Gambar 2.4 Penambahan Trait pada Model <i>User</i>	19
Gambar 3. 1 Usecase.....	27
Gambar 3. 2 Alur Sistem Login	29
Gambar 3. 3 Alur Sistem Superadmin	30
Gambar 3. 4 Alur Sistem Admin	30
Gambar 3. 5 Alur Sistem Apoteker.....	32
Gambar 3. 6 Struktur Database	33
Gambar 3. 7 Halaman Login.....	39
Gambar 3. 8 Dashboard Superadmin	39
Gambar 3. 9 Dashboard Admin	40
Gambar 3. 10 Dashboard Apoteker.....	41
Gambar 3. 11 Dashboard Owner.....	42
Gambar 3. 12 Halaman data obat Admin.....	43
Gambar 3. 13 Halaman data obat Apoteker dan Owner	43
Gambar 3. 14 Halaman distributor Admin.....	44
Gambar 3. 15 Halaman Distributor Owner	44
Gambar 3. 16 Halaman transaksi obat masuk Admin.....	45
Gambar 3. 17 Halaman transaksi obat keluar Apoteker	45
Gambar 3. 18 Halaman laporan Admin dan Owner.....	46
Gambar 3. 19 Halaman laporan Apoteker.....	46
Gambar 3. 20 Halaman Notifikasi	47
Gambar 3. 21 Halaman Kelola User	48
Gambar 3. 22 Halaman Kelola Role	48

Gambar 3. 23 Hak Akses	49
Gambar 3. 24 Halaman Atur Hak Akses.....	50
Gambar 3. 25 Perancangan Hak Akses RBAC di Laravel Spatie	52
Gambar 4. 1 Halaman <i>Login</i>	67
Gambar 4. 2 Halaman <i>Dashboard</i> Superadmin	68
Gambar 4. 3 Halaman Kelola <i>User</i>	68
Gambar 4. 4 Halaman Kelola <i>Role</i>	69
Gambar 4. 5 Halaman Hak Akses	69
Gambar 4. 6 Halaman Edit Hak Akses	70
Gambar 4. 7 Halaman <i>Log</i> Aktifitas	71
Gambar 4. 8 Halaman <i>Dashboard</i> Admin	71
Gambar 4. 9 Halaman Data Obat	72
Gambar 4. 10 Halaman Distributor.....	73
Gambar 4. 11 Halaman Transaksi Obat	74
Gambar 4. 12 Halaman Laporan	75
Gambar 4. 13 Halaman Notifikasi	76
Gambar 4. 14 Halaman <i>Dashboard</i> Apoteker.....	77
Gambar 4. 15 Halaman Data Obat	77
Gambar 4. 16 Halaman Transaksi Obat	78
Gambar 4. 17 Halaman Laporan	79
Gambar 4. 18 Halaman <i>Dashboard</i> Owner.....	80
Gambar 4. 19 Halaman Distributor.....	81

DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Pencatatan Stok Obat Manual.....	95
Lampiran 2. Surat Pesanan.....	96
Lampiran 3. Daftar Distributor	97
Lampiran 4. Dokumentasi Wawancara	98
Lampiran 5. Dokumentasi Obat-obatan di Apotek	99

BAB I

PENDAHULUAN

1.1 Latar Belakang

Manajemen adalah serangkaian kegiatan yang meliputi perencanaan, pengorganisasian, pengarahan dan pengendalian penggunaan sumber daya yang ada, baik sumber daya manusia maupun sumber daya material, untuk mencapai tujuan organisasi secara efisien [1]. Dari sisi pengelolaan barang, fokusnya adalah pada pengelolaan arus barang masuk dan keluar untuk menjamin ketersediaan dan efisiensi biaya [2], sedangkan manajemen inventaris lebih fokus pada pemantauan jumlah stok, kondisi fisik barang, dan perputaran barang [3].

Pengelolaan stok obat di lingkungan Apotek merupakan aspek penting untuk menjamin ketersediaan obat sesuai kebutuhan dan tepat waktu. Namun, di fasilitas kesehatan, sistem pencatatan stok masih dilakukan secara manual menggunakan media seperti buku besar [4]. Pencatatan manual yang masih menggunakan buku besar seringkali memakan waktu lama, membebani tenaga kerja, dan berpotensi menurunkan kualitas pelayanan akibat kesalahan pencatatan [5]. Hal ini menyebabkan pengelolaan data menjadi tidak efisien, terutama ketika diperlukan pelacakan data yang cepat.

Situasi serupa juga terjadi di Apotek Pratama dr. Moris, dimana pencatatan persediaan obat masih dilakukan secara manual dengan menggunakan buku catatan berukuran besar. Akibatnya, data yang tercatat seringkali tidak sesuai dengan jumlah stok yang sebenarnya tersedia. Sehingga apoteker harus melakukan pemeriksaan fisik secara langsung, terlambat mengetahui obat kadaluarsa, dan kesulitan menyusun laporan persediaan dengan cepat dan akurat.

Aspek keamanan menjadi perhatian penting dalam sistem pengelolaan data. Sistem manual tidak memiliki kontrol akses yang maksimal. Setiap individu yang memiliki akses fisik terhadap dokumen dapat dengan bebas melihat, mengubah, atau bahkan menghapus informasi penting tanpa batasan, sehingga data berisiko disalahgunakan oleh pihak yang tidak berkepentingan. Untuk menjawab tantangan

ini, pendekatan *Role-Based Access Control* (RBAC) dinilai sangat relevan. Dalam penelitian Yoga Prasetya dkk. [6] menunjukkan bahwa RBAC dapat menyederhanakan proses otorisasi dan memberikan tingkat keamanan yang lebih baik dibandingkan model lain seperti DAC dan MAC.

RBAC merupakan model kontrol akses yang memberikan izin kepada pengguna berdasarkan peran (*role*) mereka dalam organisasi, sehingga hanya pengguna yang berwenang yang dapat mengakses data atau fitur tertentu. RBAC mampu meningkatkan keamanan sistem, menyederhanakan pengelolaan hak akses, dan mendukung prinsip "*least privilege*" dalam pengembangan sistem informasi. Penerapan *Role-Based Access Control* (RBAC) diperlukan untuk membatasi akses sesuai peran pengguna, sehingga keamanan data lebih terjamin dan pengelolaan stok obat menjadi lebih efisien, serta mendukung peningkatan kualitas pelayanan kefarmasian.

Penelitian Savira Yudith Parameswari dkk. [7], Merancang sistem monitoring stok barang berbasis web pada suatu perusahaan. Kimia Farma Diagnostika Malang menggunakan pendekatan Laravel dan Waterfall. Sistem yang dikembangkan mendukung penggunaan oleh berbagai peran dan terbukti mampu meningkatkan efisiensi pencatatan, dengan tingkat kepuasan pengguna mencapai 89,64% berdasarkan hasil pengujian.

Penelitian oleh Arista Rahmat Ramadhan dkk. [8] di Apotek Salam merancang sistem informasi berbasis web yang menerapkan metode First Expired First Out (FEFO) dalam pengelolaan stok obat. Sistem ini memberikan data langsung mengenai persediaan, harga, serta notifikasi mengenai masa kadaluarsa, sehingga pengelolaan stok menjadi lebih tepat dan terintegrasi, serta meningkatkan kualitas pelayanan.

Penelitian lain yang dilakukan oleh I Made Satrya Ramayu [9] di Apotek Nusa Farma memperlihatkan bahwa penerapan sistem informasi pengelolaan stok barang berbasis web menggunakan metode System Development Life Cycle (SDLC) mampu meningkatkan efisiensi operasional serta akurasi data dalam pelayanan kefarmasian. Sistem ini meliputi pencatatan data obat masuk, keluar, kadaluarsa, serta proses pelaporan dengan dukungan database MySQL.

Implementasi sistem tersebut berkontribusi pada peningkatan efisiensi kerja dan keandalan data dalam pelayanan kefarmasian.

Berdasarkan kondisi di atas, penelitian ini tidak hanya berfokus pada pembangunan sistem informasi manajemen stok obat berbasis website, tetapi juga mengimplementasikan model keamanan *Role-Based Access Control* (RBAC) secara menyeluruh. Implementasi RBAC dilakukan menggunakan Laravel Spatie yang mendukung pengelolaan *role* dan *permission* secara modular dan fleksibel. Sistem dirancang untuk membatasi hak akses berdasarkan tugas dan tanggung jawab pengguna, seperti Superadmin, Admin, dan Apoteker.

Pengembangan sistem ini menggunakan metode Waterfall agar proses dapat berjalan secara sistematis dan terdokumentasi dengan baik. Diharapkan hasil penelitian ini mampu meningkatkan efisiensi pengelolaan stok, menjamin keamanan data melalui pengaturan akses yang tepat, serta mendukung kualitas pelayanan kefarmasian di Apotek Pratama dr. Moris.

1.2 Perumusan Masalah

Pengelolaan stok obat di Apotek Pratama dr. Moris masih dilakukan secara manual, sehingga menghadapi berbagai kendala, baik dari aspek teknis maupun operasional. Tanpa sistem digital, proses pencatatan transaksi obat menjadi tidak konsisten dan rawan kesalahan. Kondisi ini menyulitkan proses pemantauan ketersediaan stok serta identifikasi obat yang mendekati masa kedaluwarsa. Tidak hanya itu saja, ketiadaan sistem yang mampu melakukan pemantauan stok secara *real-time* menghambat pengambilan keputusan yang cepat dan tepat terkait *restocking* maupun penarikan obat yang sudah kedaluwarsa. Situasi ini berdampak langsung pada pelayanan kepada pasien, terutama pada kondisi darurat.

Aspek keamanan data, sistem manual tidak memiliki mekanisme manajemen akses berbasis peran. Setiap individu yang memiliki akses fisik ke buku catatan dapat melihat atau mengubah informasi tanpa batasan, dan tanpa catatan aktivitas (*log*) yang dapat diaudit. Situasi tersebut juga berisiko menyebabkan penyalahgunaan data atau hilangnya informasi penting. Untuk itu, dibutuhkan sistem informasi berbasis web yang mampu mencatat transaksi obat secara digital,

memantau stok secara waktu nyata, serta menerapkan mekanisme *Role-Based Access Control* (RBAC) untuk menjamin keamanan data dan membatasi hak akses sesuai peran masing-masing pengguna pada sistem.

1.3 Batasan Masalah

Agar penelitian ini lebih terarah dan fokus terhadap tujuan yang ingin dicapai, maka ruang lingkup permasalahan dalam penelitian ini dibatasi sebagai berikut:

1. Objek penelitian difokuskan pada Apotek Pratama dr. Moris sebagai studi kasus utama yang masih menggunakan sistem pencatatan stok obat secara manual.
2. Sistem yang dikembangkan merupakan sistem informasi manajemen stok obat berbasis website yang dibangun menggunakan framework Laravel 11 dengan penerapan keamanan *Role-Based Access Control* (RBAC) menggunakan Laravel Spatie.
3. Peran pengguna (*role*) yang diterapkan dalam sistem ini hanya mencakup empat kategori utama, yaitu:
 - a. Superadmin: memiliki hak akses penuh terhadap sistem, termasuk pengelolaan pengguna dan pengaturan peran.
 - b. Admin: bertanggung jawab mengelola data obat, transaksi obat masuk, data distributor, laporan, dan menerima notifikasi.
 - c. Apoteker: bertugas mencatat transaksi obat keluar, melihat laporan stok, serta menerima notifikasi obat yang menipis atau mendekati kedaluwarsa.
 - d. Owner: mengawasi operasional apotek dan memantau kinerja staf. Hanya dapat melihat data obat, distributor, laporan dan statistik stok obat.
4. Ruang lingkup fungsional sistem dibatasi pada:
 - a. Pengelolaan data obat, distributor, dan transaksi obat (masuk dan keluar).
 - b. Pembuatan dan pencetakan laporan stok serta transaksi obat.
 - c. Notifikasi otomatis melalui dashboard terkait stok menipis atau obat

mendekati kedaluwarsa.

- d. Penerapan sistem keamanan berbasis peran (RBAC) untuk membatasi akses antar pengguna sesuai dengan tugas dan tanggung jawabnya.
5. Sistem tidak mencakup proses penjualan obat secara finansial (transaksi pembayaran), integrasi dengan sistem kasir, atau pengelolaan data pasien secara medis. Fokus penelitian ini hanya pada manajemen stok obat dan kontrol keamanan data.
6. Metode pengembangan sistem yang digunakan adalah metode Waterfall, dimulai dari tahap analisis kebutuhan, perancangan, implementasi, pengujian, hingga pemeliharaan.
7. Platform dan teknologi pendukung yang digunakan terbatas pada:
 - a. Bahasa pemrograman PHP dengan framework Laravel;
 - b. Database MySQL;
 - c. Server lokal XAMPP;
 - d. Browser modern seperti Google Chrome atau Mozilla Firefox.
8. Pengujian sistem dilakukan menggunakan metode Black Box Testing untuk memastikan fungsionalitas sistem berjalan sesuai kebutuhan tanpa membahas aspek performa server atau keamanan jaringan secara mendalam.

1.4 Tujuan

Tujuan penelitian ini adalah sebagai berikut:

1. Merancang dan membangun sistem informasi manajemen stok obat berbasis website untuk Apotek Pratama dr. Moris.
2. Mengimplementasikan keamanan *Role-Based Access Control* (RBAC) agar setiap pengguna hanya memiliki akses sesuai dengan tugas dan tanggung jawabnya.

1.5 Manfaat

Diharapkan penelitian ini dapat memberikan kontribusi sebagai berikut:

1. Memberikan solusi terhadap permasalahan pengelolaan stok obat yang sebelumnya dilakukan secara manual dengan menyediakan sistem informasi berbasis website yang terkomputerisasi dan terintegrasi.
2. Meningkatkan efisiensi dan akurasi dalam pencatatan data obat, transaksi obat masuk dan keluar, pemantauan stok, serta pengelolaan laporan stok obat secara sistematis dan *real-time*.
3. Mendukung pengendalian dan pengamanan data melalui penerapan mekanisme *Role-Based Access Control* (RBAC) sehingga hak akses pengguna dapat dibatasi sesuai dengan peran dan kewenangannya masing-masing.
4. Menjadi referensi dan bahan acuan bagi pengembangan sistem informasi di bidang kefarmasian maupun bidang lain yang membutuhkan penerapan manajemen stok dan keamanan akses berbasis peran.

BAB II

TINJAUAN PUSTAKA

2.1 Kajian Terdahulu

Beberapa penelitian terdahulu telah membahas permasalahan terkait pengelolaan stok obat dan pencatatan manual barang medis yang berpotensi menimbulkan berbagai kendala, seperti pencatatan yang tidak akurat, data yang hilang, dan keterlambatan pelaporan. Namun setiap penelitian memiliki pendekatan dan fokus pengembangan yang berbeda dari penelitian ini, khususnya pada aspek keamanan sistem dengan menggunakan *Role-Based Access Control* (RBAC) sebagai pembeda utama (*uniqueness*).

Penelitian I Made Satria Ramayu yang berjudul “Perancangan Sistem Informasi Persediaan Obat Dengan Metode SDLC Di Apotek Nusa Farma Nusa Penida” menyajikan solusi berupa sistem berbasis web untuk mencatat stok obat masuk, keluar dan kadaluarsa dengan menggunakan database MySQL. Sistem ini bertujuan untuk meningkatkan efisiensi dan keakuratan pencatatan yang sebelumnya dilakukan secara manual. Secara abstrak disebutkan bahwa “perancangan sistem ini dapat membantu pengelolaan data obat secara efektif dan akurat di Apotek.” [9]. Namun penelitian ini belum membahas aspek keamanan akses seperti RBAC.

Savira Yudith Parameswari bersama rekan-rekannya dalam “Pembangunan Aplikasi Monitoring Stok Berbasis Website (Studi Kasus: PT. Kimia Farma Diagnostika Malang)” mengembangkan sistem dengan framework Laravel dan metode air terjun, yang berfungsi untuk memantau stok laboratorium klinik oleh beberapa peran seperti kepala outlet dan staf pengadaan. Sistem ini diuji dengan metode Black Box dan UAT menghasilkan nilai sebesar 89,64% [7]. Meskipun bersifat *multi-User*, namun sistem ini belum menerapkan manajemen akses pengguna berbasis peran (RBAC).

Ajirizky P. S. Ashshidiq bersama tim Puskesmas Klagenserut membangun sistem informasi inventaris obat berbasis web dengan menggunakan metode air

terjun. Sistem ini memudahkan pencatatan dan pelaporan data stok obat secara *real-time*. Peneliti menyatakan bahwa “aplikasi ini dapat membantu petugas memantau data stok secara rutin dan akurat.” [19]. Namun pengaturan hak aksesnya masih bersifat umum dan belum menggunakan pendekatan RBAC.

Penelitian Nadira Ilmitha dan Nunuk Novianti dalam “Sistem Pencatatan Stok Obat Menggunakan Aplikasi Komputerisasi di Puskesmas Banjar Baru Lampung” menggantikan sistem kartu stok manual dengan aplikasi Akurat versi 5. Sistem ini meningkatkan efisiensi pencatatan dan mengurangi kesalahan manusia dengan mendukung metode FIFO. Namun aspek keamanan dan pembagian hak akses pengguna belum dikelola secara sistematis dengan pendekatan RBAC [20].

Junior Panderaya Zaitun bersama rekan-rekannya di Apotek Sumur Bor merancang aplikasi web pengelolaan stok obat dengan metode air terjun. Penelitian ini menyoroti bahwa “sistem yang dibangun dapat mempercepat proses pencatatan stok dan meningkatkan akurasi dalam pelaporan obat masuk dan keluar” [13]. Namun sistem ini belum mengatur hak akses berdasarkan jabatan atau peran.

Arista Rahmat Ramadhan bersama tim dalam “Rancang Bangun Sistem Informasi Persediaan Obat Berbasis Website (Studi Kasus di Apotek Salam)” mengembangkan sistem informasi stok obat berbasis website dengan pendekatan FEFO (*First Expired First Out*) dan dilengkapi dengan fitur notifikasi masa kadaluarsa, transaksi penjualan dan pembelian, serta laporan stok. Aplikasi ini menggunakan teknologi PHP, Ajax dan MySQL. Meskipun memberikan akses pengguna tingkat dasar (admin dan kasir), sistem ini belum menerapkan model RBAC secara keseluruhan [8].

Penelitian yang menerapkan RBAC secara eksplisit dilakukan oleh Yoga Agung Prasetya dan Danny Manongga dalam “*Role-Based Access Control (RBAC) for Flask-Based Centralized Authorization Systems: Case Study of PT.* Penelitian ini menyimpulkan bahwa “penggunaan model RBAC dalam pemberian hak akses sangat cocok untuk suatu organisasi karena dapat meningkatkan efisiensi dan keamanan sistem informasi manajemen.” [6]. Sistem diuji menggunakan test case dan dinyatakan berhasil memenuhi kebutuhan otorisasi pengguna terpusat.

Krisna Ruliyanto bersama rekan-rekannya pada “Sistem Informasi

Manajemen Persediaan Obat Berbasis Web Menggunakan Metode Prototipe di Apotek” merancang prototipe sistem informasi stok obat berbasis website yang menggantikan pencatatan manual menggunakan Excel. Penelitian ini menghasilkan sistem dengan antarmuka yang berbeda untuk admin dan apoteker, serta menyediakan akses laporan dan pengaturan hak pengguna, namun belum mengimplementasikan RBAC secara eksplisit [4].

Yuricha dan Irwan Phan dalam penelitiannya “Penerapan Kontrol Akses Berbasis Peran dalam Sistem Manajemen Rantai Pasokan Berbasis Cloud” mengimplementasikan RBAC menggunakan kerangka Laravel dan perpustakaan Spatie. Sistem yang dibangun berhasil menjaga integritas otorisasi antar pemangku kepentingan dan mencegah akses yang tidak sah. Penelitian ini menunjukkan efektivitas RBAC dalam sistem berskala besar dan kompleks, serta menegaskan bahwa "RBAC adalah solusi untuk meningkatkan keamanan dan fleksibilitas sistem multi-pengguna berbasis cloud." [18].

M. Sahyudi dan Erliyan Redy Susanto dalam “Analisis Implementasi Sistem Keamanan Database Berbasis RBAC pada Aplikasi Enterprise Resource Planning” menekankan bahwa RBAC merupakan model yang relevan untuk menghadapi kompleksitas manajemen akses pada sistem modern. Penelitian ini menggunakan pendekatan Tinjauan Literatur Sistematis (SLR) dan menunjukkan bahwa integrasi RBAC dengan teknologi cerdas seperti pembelajaran mesin dan blockchain meningkatkan efisiensi manajemen akses dan mendukung kepatuhan terhadap peraturan data [16].

Dengan demikian, kesepuluh penelitian di atas memiliki kesamaan dalam mengatasi permasalahan pencatatan dan pengelolaan stok obat, namun belum sepenuhnya memadukan pengelolaan hak akses berbasis peran seperti yang dirancang dalam penelitian ini. Oleh karena itu, keunikan penelitian ini terletak pada integrasi penuh sistem keamanan menggunakan *Role-Based Access Control* (RBAC) untuk mengelola hak akses pengguna secara terstruktur, hal yang belum diterapkan secara menyeluruh pada penelitian-penelitian sebelumnya.

Tabel 2.1 Review Berdasarkan Penelitian Terkait

No	Judul Jurnal	Penulis dan Tahun	Masalah Penelitian	Metode Penelitian	Solusi yang Ditawarkan	Hasil Penelitian
1	Rancangan Sistem Informasi Persediaan Stok Obat	I Made Satrya Ramayu (2023)	Pengelolaan data apotek secara manual dan tidak efisien	SDLC, Black Box Testing	Sistem informasi stok obat berbasis web	Sistem dapat mencatat transaksi, memantau stok dan masa kedaluwarsa obat dengan lebih akurat dan efisien
2	Pengembangan Aplikasi Monitoring Stok Barang Berbasis Website	Savira Y. Parameswari et al. (2023)	Pencatatan stok manual di PT Kimia Farma Diagnostika Malang	Waterfall, Black Box, UAT	Aplikasi monitoring stok berbasis Laravel	Pengujian valid 100%, UAT 89,64% menunjukkan aplikasi layak digunakan
3	Pengembangan Sistem Informasi Inventory Obat Berbasis Web	Ajirizky P. S. A. et al. (2022)	Pencatatan dan distribusi obat manual	Waterfall, Black Box, UAT	Sistem stok obat berbasis web	Sistem sesuai kebutuhan pengguna, memudahkan pemantauan stok
4	Sistem Pencatatan Stok Obat Menggunakan Aplikasi Komputerisasi	Nadira Ilmitha, Nunuk Novianti (2022)	Sering terjadi kesalahan pencatatan manual	Observasi, Wawancara	Penerapan aplikasi Accurate v5	Mengurangi human error, pencatatan lebih cepat dan akurat
5	Perancangan Sistem	Junior P. Zaitun et	Manajemen stok manual	Waterfall	Aplikasi web pengelolaan	Meningkatkan efisiensi dan

	Aplikasi Web Pengelolaan Stok Obat	al. (2024)	menyebabkan kehabisan stok		stok	mengurangi kesalahan stok
6	<i>Design and Build a Website-Based Drug Supply Info System</i>	Arista R. Ramadhan et al. (2024)	Tidak akuratnya data stok dan kesulitan melacak obat kadaluarsa	FEFO, PHP-Ajax	Sistem informasi stok obat berbasis website	Mempercepat pemantauan kadaluarsa, efisiensi layanan apotek
7	RBAC untuk Sistem Otorisasi Terpusat Berbasis Flask	Yoga A. Prasetya, Danny Manongga (2024)	Kompleksitas hak akses sistem informasi	Studi literatur, Perancangan, Flask	Sistem otorisasi RBAC	RBAC cocok untuk organisasi dengan struktur kompleks
8	Sistem Informasi Manajemen Persediaan Obat	Krisna Ruliyanto et al. (2021)	Pencatatan manual dengan Excel, lambat dan tidak aman	Prototype	Prototype sistem berbasis web	Meningkatkan kecepatan dan akurasi pengelolaan stok
9	Penerapan RBAC dalam Sistem Supply Chain Berbasis Cloud	Yuricha, Irwan K. Phan (2023)	Potensi akses ilegal pada sistem SCM	Studi kasus, Laravel Spatie	Penerapan RBAC	Menjamin otorisasi peran dan keamanan data cloud
10	Analisis Implementasi Keamanan Basis Data Berbasis	M. Sahyudi, E. R. Susanto (2025)	Kompleksitas akses dan regulasi di ERP	Systematic Literature Review	Integrasi RBAC dengan AI, NLP, Blockchain	Efisiensi meningkat, keamanan data naik 37%, biaya turun 42%

	RBAC pada ERP					
11	Implementasi <i>Role-Based Access Control</i> (RBAC) pada Sistem Informasi Manajemen Stok Obat Berbasis Website	Dea Agustina Purba (2026)	Tidak adanya sistem informasi manajemen stok dan keamanan berbasis peran di Apotek Pratama dr. Moris	Waterfall, Laravel	Sistem informasi stok obat berbasis website menggunakan Laravel + RBAC	Sistem diharapkan meningkatkan efisiensi dan membatasi akses sesuai peran pengguna

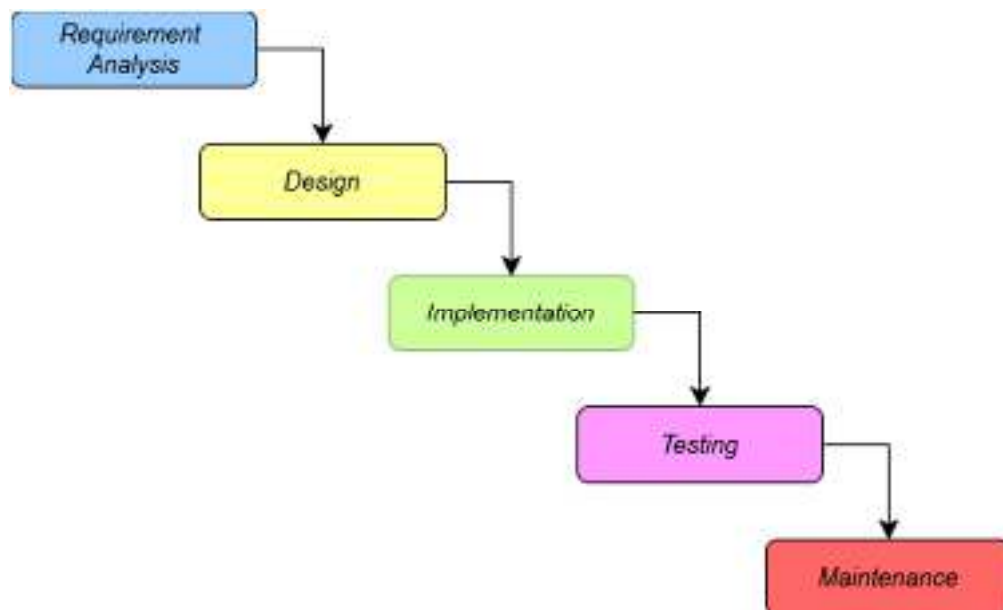
2.2 Landasan Teori

Bab ini membahas tentang teori-teori yang menjadi landasan dalam pengembangan sistem pengelolaan stok obat berbasis web dengan keamanan *Role-Based Access Control* (RBAC) di Apotek Dr. Pratama. Moris. Teori-teori yang akan dibahas antara lain perancangan sistem, pengelolaan stok obat, pengembangan website menggunakan Laravel, dan *Role-based access security* (RBAC).

2.1.1 Sistem Informasi

Sistem informasi adalah suatu proses yang mengubah kebutuhan pengguna menjadi rancangan teknis, yang mencakup pendeskripsian, perencanaan, dan penataan komponen-komponen sistem menjadi satu kesatuan yang fungsional [9], Proses ini bertujuan untuk membangun sistem baru atau menyempurnakan sistem yang sudah ada dengan menggunakan pendekatan rekayasa perangkat lunak terstruktur, salah satunya metode Waterfall.

Metode Waterfall merupakan model pengembangan perangkat lunak yang bersifat linear dan sekuensial. Setiap tahap dalam model ini harus diselesaikan secara tuntas sebelum melanjutkan ke tahap berikutnya [9]. Model ini sangat cocok digunakan ketika kebutuhan sistem telah dipahami dengan jelas sejak awal, karena memberikan struktur kerja yang sistematis dan terarah.



Gambar 2.1 Model Waterfall [9]

Tahapan dalam Model Waterfall adalah sebagai berikut[13] :

1. Analisis Kebutuhan (*Requirements Analysis*)
Proses ini melibatkan identifikasi kebutuhan fungsional dan non-fungsional yang harus dipenuhi oleh sistem. Tahap ini memerlukan komunikasi yang intens antara pengembang dan pemilik sistem.
2. Perancangan Sistem (*System Design*)
Tahapan ini meliputi desain antarmuka, struktur database, dan arsitektur sistem. Hasil akhir berupa rancangan teknis yang siap diimplementasikan.
3. Implementasi (*Implementation*)
Pada tahap ini, rancangan sistem diterjemahkan ke dalam kode program dengan menggunakan bahasa dan framework pemrograman tertentu.
4. Pengujian (*Testing*)
Proses ini bertujuan untuk memverifikasi apakah sistem berfungsi sesuai dengan persyaratan yang telah ditentukan, termasuk pengujian fungsionalitas dan keamanan.
5. Pemeliharaan (*Maintenance*)
Setelah sistem diimplementasikan, pemeliharaan rutin diperlukan untuk memperbaiki bug, beradaptasi dengan fitur baru, dan menjaga kinerja sistem.

tetap optimal.

Meskipun metode ini memberikan alur kerja yang jelas, namun kelemahannya adalah kurang fleksibel terhadap perubahan mendadak yang mungkin terjadi di tengah pengembangan. Selain itu, proses pengerjaannya bisa menjadi lambat karena satu tahap harus menunggu selesainya tahap sebelumnya [13].

Penerapan model Waterfall dalam pengembangan sistem manajemen stok obat bertujuan untuk memastikan bahwa proses pengembangan dapat dilakukan secara terstruktur dan terdokumentasi dengan baik terutama dalam memenuhi kebutuhan pengguna di Apotek Dr. Pratama. Moris.

2.1.2 Manajemen Stok Obat

Manajemen stok obat mencakup serangkaian kegiatan pengelolaan persediaan obat, mulai dari perencanaan, pengadaan, penyimpanan, pendistribusian, hingga pelaporan. Tujuan utamanya adalah menjamin ketersediaan obat dalam jumlah yang tepat, menjaga kualitas, dan memastikan waktu distribusi sesuai kebutuhan [4].

1. Fungsi Utama Manajemen Stok Obat [9]:

Fungsi utama dari manajemen stok obat adalah sebagai berikut:

- a. Perencanaan kebutuhan obat: Menentukan jenis dan jumlah obat yang dibutuhkan berdasarkan riwayat konsumsi dan prediksi kebutuhan di masa mendatang.
- b. Pengadaan: Proses pemesanan atau pembelian dari distributor resmi atau instansi kesehatan terkait.
- c. Penerimaan dan penyimpanan: Pencatatan dan pengorganisasian obat berdasarkan sistem tertentu, seperti FIFO (*First In First Out*) atau FEFO (*First Expired First Out*).
- d. Distribusi dan penggunaan: Distribusi obat kepada pasien atau unit pelayanan internal.
- e. Pemantauan dan evaluasi: Meliputi kegiatan pencatatan stok, pemantauan kadaluwarsa, serta pencatatan dan pelaporan sisa stok.

2. Permasalahan Umum dalam Sistem Manual

Menurut penelitian J.P. Zaitun [13] sistem pengelolaan yang masih berbasis manual sering kali mengalami kendala seperti keterlambatan pencatatan, data tidak konsisten, kesalahan dalam mendeteksi stok kosong, dan kesulitan dalam pencarian data. Proses manual juga memiliki risiko tinggi kehilangan data karena kesalahan manusia. Dalam manajemen stok obat, hal ini dapat menyebabkan kekeliruan pengadaan, stok kosong yang tidak terdeteksi, serta kesulitan melacak riwayat keluar-masuk obat. Kondisi ini menunjukkan pentingnya penerapan sistem digital untuk meningkatkan akurasi, efisiensi, dan keamanan data obat.

3. Sistem Otomatis dan Keunggulannya

Studi oleh Althaf Haaerun dan Mardhiyyah (2024) [14], menunjukkan bahwa penggunaan sistem pengelolaan stok berbasis web menawarkan berbagai keuntungan, antara lain:

- a. Pemantauan stok secara real-time, sehingga memudahkan pengambilan keputusan.
- b. Pemberitahuan otomatis mengenai stok rendah atau obat-obatan yang mendekati tanggal kedaluwarsa.
- c. Laporan stok yang dapat dihasilkan secara otomatis, lebih cepat dan akurat.
- d. Akses sistem dapat diberikan kepada banyak pengguna dengan tingkat otorisasi berbeda, sehingga meningkatkan keamanan dan efisiensi.

Dengan penerapan sistem digital, apoteker dan petugas terkait tidak perlu lagi melakukan pencatatan secara manual, namun dapat memanfaatkan data sistem yang sebenarnya untuk mengambil keputusan dengan cepat dan tepat.

2.1.3 Website dan Laravel

Website merupakan kumpulan halaman informasi yang terhubung melalui hyperlink dan dapat diakses melalui internet. Halaman-halaman tersebut membentuk struktur hierarki, dimulai dari halaman utama (*homepage*) hingga halaman anak [10]. Dokumen-dokumen yang membentuk sebuah situs web dikenal sebagai halaman web, dan pengguna dapat menelusuri antar halaman menggunakan browser seperti Google Chrome, Mozilla Firefox, dan lain-lain. Website berfungsi sebagai platform penyajian informasi secara luas, baik dalam skala lokal maupun global. Kemampuannya menyajikan konten yang dinamis dan

terintegrasi menjadikannya solusi ideal untuk pengembangan sistem informasi manajemen, termasuk di sektor farmasi [10].

Laravel adalah kerangka pengembangan aplikasi web berbasis PHP open source. Framework ini dirancang oleh Taylor Otwell untuk menyederhanakan proses pengembangan aplikasi dengan mengikuti pola arsitektur *Model-View-Controller* (MVC)[6]. Salah satu fitur utama Laravel adalah sistem *routing* yang berfungsi sebagai penghubung antara permintaan dari pengguna (*request*) dan logika program di sisi *controller*. Selain itu Laravel juga menyediakan berbagai fitur bawaan seperti otentikasi, validasi, dan manajemen database yang mempercepat proses pengembangan aplikasi. Framework ini memiliki keunggulan pada kesederhanaan sintaksis, kemudahan konfigurasi, dan dokumentasi yang lengkap. Dengan struktur modular dan dukungan komunitas yang luas, Laravel memungkinkan pengembang membangun aplikasi web yang aman, terstruktur, dan mudah dipelihara [8].

Dalam konteks penelitian ini, Laravel digunakan sebagai dasar pengembangan sistem manajemen stok obat karena kompatibel dengan implementasi *Role-Based Access Control* (RBAC) dan mendukung integrasi dengan perpustakaan pihak ketiga seperti Spatie untuk mengelola peran dan izin.

2.1.4 Keamanan: *Role-Based Access Control* (RBAC)

Keamanan sistem merupakan aspek yang sangat penting dalam pengembangan aplikasi, terutama untuk aplikasi yang menangani data sensitif seperti informasi obat dan transaksi medis. Salah satu jenis model kontrol akses yang sering diterapkan adalah *Role-Based Access Control* (RBAC).

1. Pengertian RBAC

Role-Based Access Control (RBAC) merupakan model manajemen hak akses yang memberikan izin kepada pengguna berdasarkan peran (*Roles*) yang dimilikinya dalam suatu organisasi. Prinsip utama RBAC adalah “hak istimewa terkecil”, yang berarti setiap pengguna hanya diberikan hak akses minimum yang diperlukan untuk melaksanakan implementasi [17]. Sebagai contoh:

- a. Admin: Memiliki akses penuh, termasuk manajemen pengguna dan data obat.
- b. Staf Farmasi: Hanya dapat menginput data stok masuk dan keluar.

- c. Owner: Hanya dapat melihat laporan dan statistik stok obat.

RBAC dikembangkan oleh Sandhu dkk. (1996) sebagai alternatif yang lebih efisien dan aman dibandingkan model seperti *Discretionary Access Control* (DAC) atau *Mandatory Access Control* (MAC), karena RBAC lebih mudah dalam mengelola hak akses dan lebih fleksibel dalam implementasinya [17]. Salah satu implementasi RBAC yang populer dalam pengembangan aplikasi berbasis Laravel adalah Laravel Spatie *Permission*, sebuah *package open-source* yang menyediakan pengelolaan hak akses berbasis peran dan izin secara terstruktur dan sederhana.

2. Tujuan dan Prinsip

Tujuan utama dari RBAC adalah untuk menyederhanakan manajemen hak akses dan memastikan bahwa pengguna hanya memiliki hak istimewa minimum (*principle of least privilege*) yang diperlukan untuk menjalankan tugasnya. Dengan demikian, risiko terhadap akses tidak sah atau penyalahgunaan data dapat diminimalkan. RBAC juga memiliki prinsip yang membantu, di antaranya ialah:

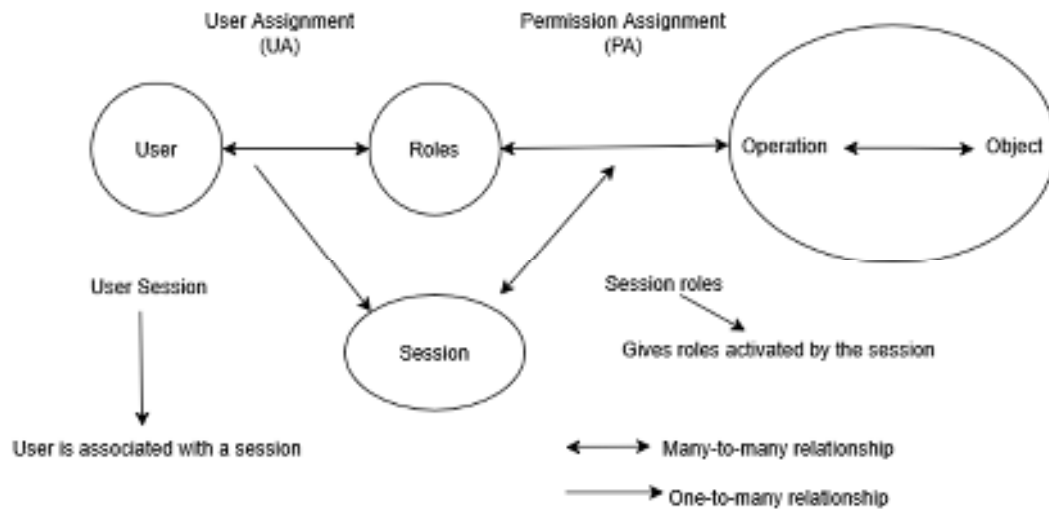
1. Meningkatkan keamanan sistem.
2. Mengurangi beban administrator dalam memberikan izin akses.
3. Menjamin kepatuhan terhadap kebijakan keamanan internal atau eksternal.

3. Komponen RBAC

Menurut NIST Model RBAC secara umum terdiri dari empat komponen utama yaitu sebagai berikut:

- a. *User*: Entitas yang mewakili individu atau sistem yang menggunakan aplikasi.
- b. *Role*: Posisi atau fungsi dalam sistem, seperti admin, apoteker, atau kasir.
- c. *Permission*: Hak akses terhadap fitur atau data tertentu pada sistem.
- d. *Session*: Interaksi aktif antara pengguna dan sistem saat menggunakan aplikasi.

Pengguna dikaitkan dengan peran tertentu, dan setiap peran memiliki hak akses spesifik yang telah ditentukan, seperti yang ditunjukkan pada gambar 2.2 [18].



Gambar 2.2 Komponen RBAC [17]

4. Keunggulan RBAC dibandingkan Model Lain

RBAC memiliki sejumlah keunggulan dibandingkan model kontrol akses lainnya seperti DAC dan MAC, antara lain:

- Mempermudah pengelolaan hak akses karena tidak perlu memberikan izin kepada setiap individu secara terpisah.
- Mengurangi kesalahan pemberian izin: karena akses dikontrol melalui peran yang sudah terstandar.
- Meningkatkan keamanan sistem dengan mencegah akses berlebihan (*over-privilege*).
- Mendukung kebijakan keamanan yang fleksibel dan dapat disesuaikan seiring pertumbuhan organisasi [18].

RBAC sangat sesuai diterapkan di sistem manajemen stok obat karena mendukung pengelolaan akses berdasarkan peran seperti admin, apoteker, dan owner. Model ini terbukti meningkatkan keamanan dan efisiensi dibanding DAC yang lebih rentan, serta lebih praktis dibanding MAC yang terlalu kompleks untuk sistem non-militer. [18].

5. Konsep Model RBAC

Model RBAC secara konseptual menggambarkan hubungan antara pengguna (*User*), peran (*role*), dan izin akses (*Permission*). Setiap pengguna memperoleh hak akses melalui peran yang dimilikinya, dan peran tersebut menentukan fitur atau

operasi sistem yang dapat diakses. Ilustrasi sederhana konsep ini ditampilkan pada Gambar 2.3.



Gambar 2.3 Konsep Model RBAC [18]

Berdasarkan gambar tersebut, setiap *User* diberikan satu atau lebih *Role* (peran), dan setiap *Role* memiliki akses terhadap satu atau lebih *object* atau *Permission*, baik berupa fitur menu maupun operasi tertentu dalam sistem. Pola ini membentuk rantai akses yang terstruktur, fleksibel, dan efisien dalam pengelolaan hak akses pengguna, khususnya pada sistem multi-pengguna.

Sebagai penguatan konsep, ilustrasi model RBAC sederhana ini juga merujuk pada model yang digunakan dalam penelitian oleh Yuricha & Phan [18], yang menggunakan framework Laravel dan pustaka Laravel Spatie untuk pengelolaan peran dan izin. Laravel yang menerapkan arsitektur Model-View-Controller (MVC) menyediakan model *User* bawaan yang telah mendukung proses autentikasi. Implementasi RBAC dimulai dengan menambahkan trait *HasRoles* dari Spatie pada model *User*, sebagaimana terlihat pada Gambar 2.4.

```
use Illuminate\Foundation\Auth\User as Authenticatable;
use Spatie\Permission\Traits\HasRoles;

class User extends Authenticatable
{
    use HasRoles;

    // ...
}
```

Gambar 2.4 Penambahan Trait pada Model *User* [18]

Dalam sistem manajemen stok obat berbasis Laravel, setiap peran pengguna didefinisikan secara jelas menggunakan sintaks Laravel Spatie, dan setiap peran

diberikan izin akses sesuai dengan tugas dan tanggung jawabnya. Contoh pengaturan *Role* dan perizinan pada sistem yang terlihat pada tabel 2.2 [18]:

Tabel 2.2 Penambahan *Role* dan *Permission List* Sistem [18]

Nama <i>Role</i>	Pengaturan <i>Role</i>	Pengaturan <i>Permission</i>
ADMIN_PEMBELIAN	<i>Role::create</i> (['name' => 'ADMIN_PEMBELIAN']);	<i>Permission::create</i> (['name' => 'PEMBELIAN.SHOW']); (Tampilkan) <i>Permission::create</i> (['name' => 'PEMBELIAN.CREATE']); (Buat) dan seterusnya
ADMIN_PENJUALAN	<i>Role::create</i> (['name' => 'ADMIN_PENJUALAN']);	<i>Permission::create</i> (['name' => 'PENJUALAN.DELETE']); (Hapus) <i>Permission::create</i> (['name' => 'PENJUALAN.PRINT']); (Cetak) dan seterusnya
ADMIN_KEUANGAN	<i>Role::create</i> (['name' => 'ADMIN_KEUANGAN']);	<i>Permission::create</i> (['name' => 'PELUNASAN_HUTANG.PRINT']); (Cetak) dan seterusnya
ADMIN_GUDANG	<i>Role::create</i> (['name' => 'ADMIN_GUDANG']);	<i>Permission::create</i> (['name' => 'AKTIVITAS_LOGISTIK.SHOW']); (Tampilkan) <i>Permission::create</i> (['name' => 'KEUANGAN.PRINT']); (Cetak) dan seterusnya

Setelah semua peran dan izin berhasil ditentukan, langkah selanjutnya adalah mengaitkan pengguna dalam sistem dengan peran dan izin yang sesuai. Proses ini dilakukan melalui fungsi *applyRole()* untuk menetapkan peran kepada pengguna, dan *givePermissionTo()* untuk memberikan hak akses pada peran yang ditentukan. Dalam implementasinya, sistem akan memverifikasi hak akses ketika pengguna mencoba mengakses halaman tertentu dengan menggunakan perintah *hasPermissionTo('PERMISSION_NAME')*, untuk memastikan hanya pengguna yang berwenang yang dapat mengakses fitur tersebut [18].

Teori-teori yang diuraikan di atas menjadi landasan penting dalam membangun sistem manajemen stok obat di Apotek Dr. Pratama. Moris. Dengan pendekatan Waterfall, dukungan dari Laravel, dan implementasi RBAC, sistem yang dikembangkan diharapkan mampu mengatasi permasalahan pengelolaan stok yang ada secara fungsional, efisien dan aman.

BAB III

METODE PENELITIAN

3.1 Data dan Alat Penelitian

Bagian ini menjelaskan data apa yang dikumpulkan serta bahan dan alat yang digunakan selama proses perancangan dan pengembangan sistem informasi manajemen stok obat di Apotek Pratama dr. Moris.

3.1.1 Data Penelitian

Data penelitian yang digunakan dalam penyusunan dan pengembangan sistem ini diperoleh secara langsung dari Apotek Pratama dr. Moris sebagai objek penelitian. Data tersebut digunakan sebagai dasar analisis kebutuhan sistem serta perancangan fitur-fitur yang sesuai dengan kondisi operasional apotek. Adapun data penelitian yang dikumpulkan meliputi:

1. Data Obat

Data ini mencakup informasi nama obat, kode obat, kategori, satuan, stok awal, tanggal kedaluwarsa, dan harga obat. Data obat digunakan sebagai basis utama dalam pengelolaan stok obat pada sistem.

2. Data Stok Obat

Data stok berisi informasi jumlah persediaan obat yang tersedia, perubahan stok akibat transaksi obat masuk dan keluar, serta status stok (aman, menipis, atau habis).

3. Data Transaksi Obat

Data transaksi terdiri dari transaksi obat masuk dan transaksi obat keluar, yang meliputi tanggal transaksi, jumlah obat, jenis transaksi, dan pengguna yang melakukan transaksi.

4. Data Distributor

Data distributor mencakup nama distributor, alamat, dan informasi kontak yang digunakan dalam proses pengadaan obat.

5. Data Pengguna (*User*)

Data pengguna sistem meliputi informasi akun seperti nama pengguna,

email, kata sandi, serta peran (*role*) yang dimiliki, yaitu Superadmin, Admin, Apoteker, dan Owner.

6. Data Hak Akses (*Role* dan *Permission*)

Data ini digunakan untuk mengatur pembatasan akses terhadap fitur sistem berdasarkan peran pengguna dengan menerapkan mekanisme *Role-Based Access Control* (RBAC).

Data-data tersebut diperoleh melalui observasi langsung, wawancara dengan pihak apotek, serta dokumentasi pencatatan stok obat manual yang sebelumnya digunakan.

3.1.2 Bahan dan Alat Penelitian

Bahan dan alat penelitian digunakan untuk mendukung proses perancangan, pengembangan, serta pengujian sistem. Adapun bahan dan alat yang digunakan adalah sebagai berikut:

1. Perangkat Keras (*Hardware*)
 - a. Laptop atau komputer sebagai perangkat pengembangan sistem
 - b. Perangkat jaringan (WiFi) untuk koneksi server lokal
 - c. Perangkat pendukung seperti keyboard dan mouse
2. Perangkat Lunak (*Software*)
 - a. Sistem Operasi Windows
 - b. Visual Studio Code sebagai code editor
 - c. XAMPP sebagai web server lokal
 - d. PHP versi yang mendukung Laravel 11
 - e. Framework Laravel 11
 - f. Database MySQL
 - g. Laravel Spatie Permission untuk implementasi RBAC
 - h. Browser (Google Chrome / Mozilla Firefox) untuk pengujian sistem.
3. Bahan Pendukung
 - a. Dokumen pencatatan stok obat manual
 - b. Surat pesanan obat
 - c. Data distributor
 - d. Dokumentasi wawancara dan observasi

Bahan dan alat penelitian yang digunakan dalam penelitian ini disesuaikan dengan kebutuhan perancangan dan pengembangan sistem. Penggunaan perangkat keras, perangkat lunak, serta bahan pendukung yang tepat diharapkan dapat menunjang proses pengembangan sistem informasi manajemen stok obat agar berjalan secara efektif dan sesuai dengan tujuan penelitian.

3.2 Prosedur Penelitian

Penelitian ini menggunakan model pengembangan sistem *Waterfall*, yaitu model pengembangan perangkat lunak yang dilakukan secara bertahap dan berurutan. Setiap tahapan harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahapan berikutnya. Model ini dipilih karena memberikan alur kerja yang sistematis dan sesuai untuk pengembangan sistem informasi dengan kebutuhan yang telah terdefinisi dengan jelas seperti pada gambar 2.1.

3.3 Analisis Kebutuhan

Tahap analisis kebutuhan dilakukan melalui observasi langsung dan wawancara dengan pihak Apotek Pratama dr. Moris. Berdasarkan hasil analisis, diketahui bahwa proses pencatatan stok obat masih dilakukan secara manual menggunakan buku besar, sehingga menimbulkan beberapa permasalahan seperti kesulitan dalam memantau obat yang mendekati masa kedaluwarsa, keterlambatan dalam pembuatan laporan, serta belum adanya pengamanan akses data pengguna.

Berdasarkan permasalahan tersebut, kebutuhan sistem dibagi menjadi dua, yaitu kebutuhan fungsional dan kebutuhan non fungsional.

3.3.1 Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang berkaitan langsung dengan fungsi dan layanan yang harus disediakan oleh sistem agar dapat digunakan sesuai tujuan pengembangannya. Adapun kebutuhan fungsional sistem adalah sebagai berikut:

1. Sistem harus menyediakan fitur login dan logout untuk setiap pengguna. Setelah login, sistem harus melakukan pengecekan peran (*role*) pengguna dan memberikan akses sesuai dengan hak akses yang dimiliki berdasarkan mekanisme RBAC.

2. Sistem harus mampu mengelola data pengguna, meliputi penambahan, pengubahan, dan penghapusan akun pengguna, serta pengaturan peran (*role*) dan hak akses (*permission*) pada setiap pengguna.
3. Sistem harus mendukung pengelolaan *role* dan *permission* secara dinamis menggunakan konsep *Role-Based Access Control* (RBAC), sehingga setiap peran memiliki batasan akses terhadap fitur sistem sesuai dengan tugas dan tanggung jawabnya.
4. Sistem harus menyediakan fitur untuk menambah, mengubah, menghapus, dan menampilkan data obat, yang meliputi informasi nama obat, kode obat, kategori, satuan, stok, dan tanggal kedaluwarsa.
5. Sistem harus mampu mengelola data distributor sebagai pemasok obat, meliputi informasi nama distributor, alamat, kontak, dan status distributor.
6. Sistem harus menyediakan fitur pencatatan transaksi obat masuk dari distributor yang secara otomatis menambah jumlah stok obat dalam sistem dan fitur pencatatan transaksi obat keluar yang secara otomatis mengurangi jumlah stok obat berdasarkan jumlah obat yang dikeluarkan.
7. Sistem harus mampu menampilkan informasi stok obat secara real-time, termasuk status stok (aman, menipis, atau habis) dan informasi obat yang mendekati masa kedaluwarsa.
8. Sistem harus menyediakan notifikasi otomatis melalui dashboard terkait stok obat yang menipis atau obat yang mendekati tanggal kedaluwarsa.
9. Sistem harus mampu menghasilkan laporan stok obat, laporan transaksi obat masuk, dan laporan transaksi obat keluar yang dapat ditampilkan dan dicetak sesuai dengan hak akses pengguna.
10. Sistem harus mampu mencatat aktivitas pengguna dalam sistem, seperti login, penambahan data, perubahan data, dan penghapusan data, sebagai bentuk pengawasan dan audit aktivitas sistem.

3.3.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional merupakan kebutuhan yang berkaitan dengan kualitas, kinerja, dan karakteristik sistem dalam mendukung operasional secara keseluruhan. Adapun kebutuhan non fungsional sistem adalah sebagai berikut:

1. Sistem harus menerapkan mekanisme keamanan berbasis *Role-Based Access Control* (RBAC) untuk membatasi akses pengguna sesuai peran.
2. Sistem Sistem harus mampu memproses data dan menampilkan informasi stok, transaksi, dan laporan dengan waktu respon yang cepat dan stabil dalam lingkungan penggunaan normal.
3. Sistem harus memiliki antarmuka yang mudah dipahami, konsisten, dan ramah pengguna, sehingga dapat digunakan oleh pengguna dengan latar belakang teknis yang berbeda.
4. Sistem harus dapat diakses selama jam operasional apotek dan mampu berjalan dengan baik pada server lokal menggunakan web browser modern.
5. Sistem harus dapat dijalankan pada berbagai browser modern seperti Google Chrome dan Mozilla Firefox tanpa mengalami perbedaan fungsi yang signifikan.

3.4 Perancangan

Sistem Manajemen Stok Obat Berbasis Website ini dirancang dengan pendekatan keamanan *Role-Based Access Control* (RBAC) untuk memastikan bahwa setiap pengguna hanya dapat mengakses data dan fitur yang sesuai dengan peran (*Role*) mereka dalam sistem. Terdapat empat peran utama dalam sistem ini: SuperAdmin, Admin, Apoteker, dan Owner.

3.4.1 Perancangan Sistem (*System Design*)

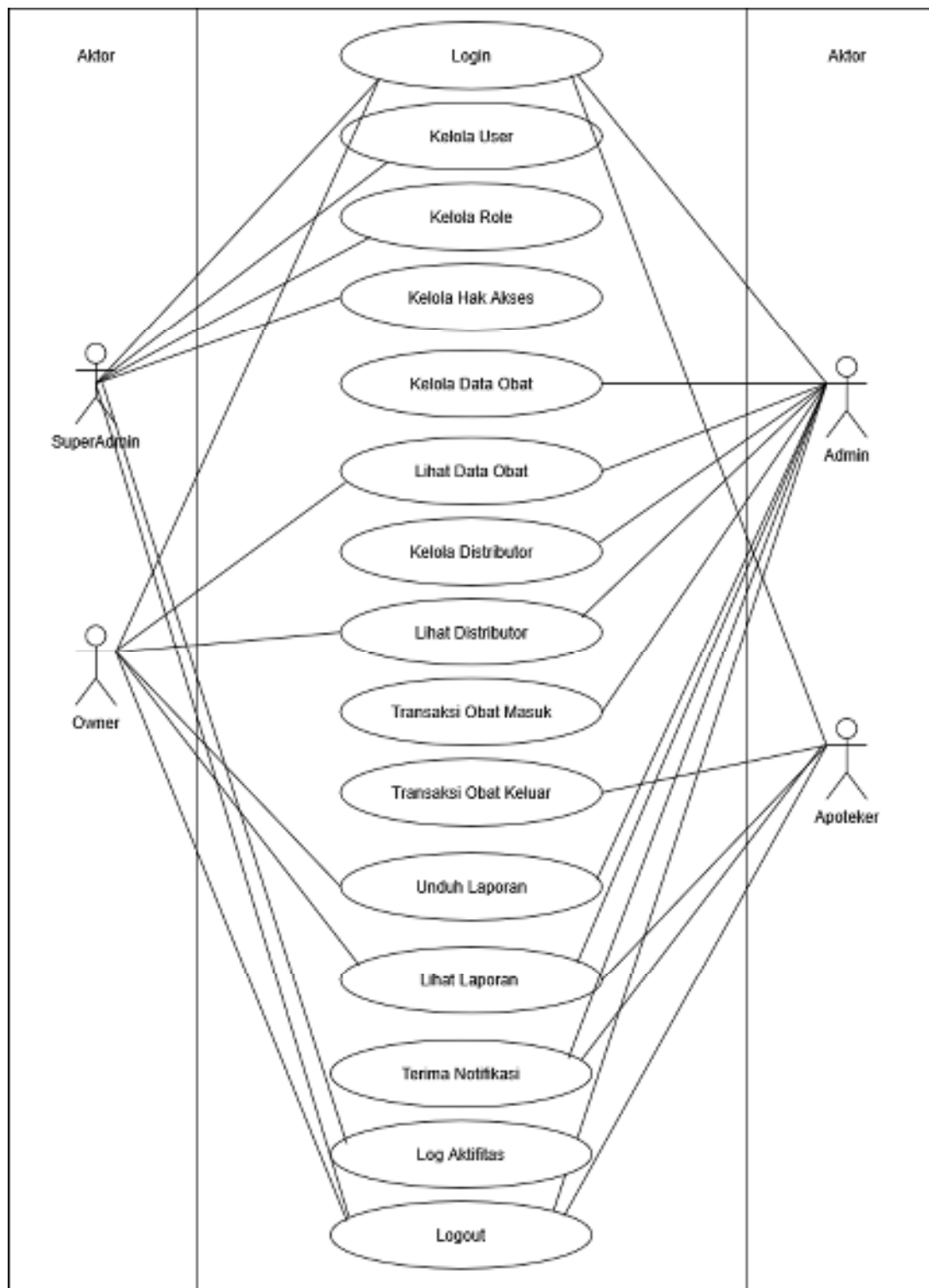
Tahap perancangan telah dilaksanakan dengan menyusun desain sistem berdasarkan kebutuhan yang telah dianalisis. Desain antarmuka pengguna dirancang untuk empat peran utama yaitu SuperAdmin, Admin, Apoteker dan Owner, masing-masing dengan akses fitur yang berbeda sesuai dengan hak dan tanggung jawabnya.

Struktur database dirancang dengan entitas seperti tabel obat, transaksi masuk, transaksi keluar, distributor, pengguna, dan *Role*, yang saling terhubung dan disesuaikan dengan proses bisnis yang ada di Apotek Pratama dr. Moris. Selain itu, telah dibuat diagram alur sistem dan use case diagram untuk menggambarkan interaksi antara pengguna dan sistem.

Teknologi yang dipilih adalah framework Laravel untuk pengembangan *backend* dan *frontend*, serta pustaka Laravel Spatie untuk implementasi *Role-Based Access Control (RBAC)*. Adapun komponen-komponen utama yang dirancang meliputi:

1. Diagram *Usecase*

Menggambarkan hubungan antara aktor (SuperAdmin, Admin, Apoteker, dan Owner) dengan *fungsi* sistem seperti *login*, manajemen *User*, manajemen obat, mengelola distributor, transaksi obat, dan pembuatan laporan.



Gambar 3. 1 Usecase

Gambar 3.1 diagram *usecase* menggambarkan interaksi antara tiga aktor utama, yaitu SuperAdmin, Admin, Apoteker dan Owner, dalam sistem manajemen

stok obat berbasis website. Diagram ini digunakan untuk mendeskripsikan fitur-fitur utama yang dapat diakses berdasarkan peran masing-masing pengguna sesuai dengan prinsip *Role-Based Access Control (RBAC)*.

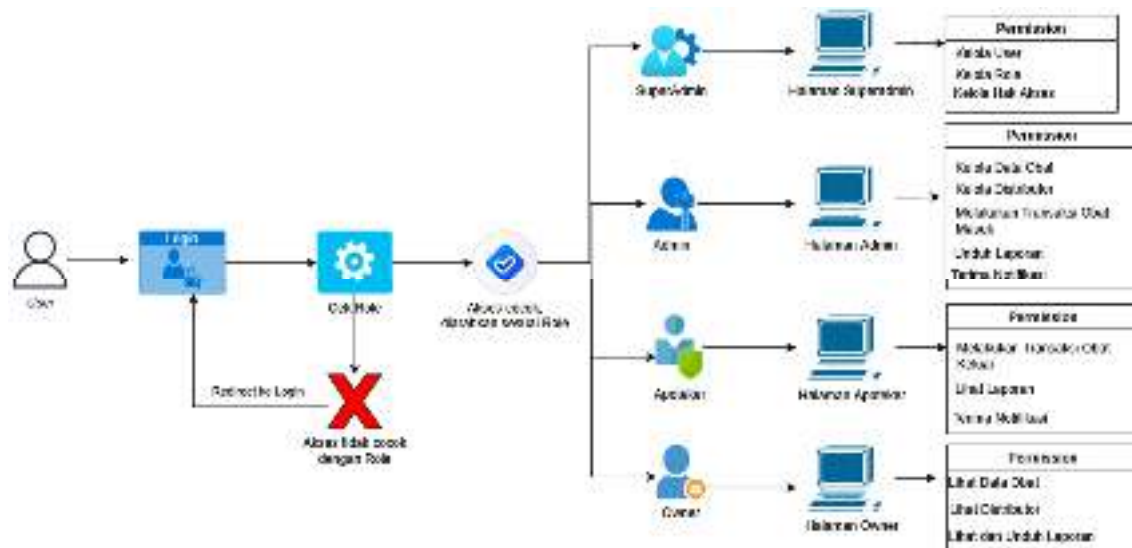
- a. SuperAdmin bertanggung jawab dalam mengelola pengguna sistem dengan konsep RBAC dan juga penambahan, pengubahan, dan penghapusan akun pengguna.
- b. Admin memiliki akses terhadap pengelolaan data obat dan distributor, serta melakukan pencatatan transaksi obat masuk. Admin juga dapat memantau stok dan melihat atau mengunduh laporan dalam format PDF. Admin juga akan menerima notifikasi stok obat menipis atau mendekati kedaluwarsa melalui dashboard.
- c. Apoteker terlibat langsung dalam aktivitas operasional seperti pencatatan transaksi obat keluar, pemantauan stok, serta menerima notifikasi jika terdapat stok menipis atau obat kedaluwarsa. Apoteker juga memiliki akses untuk melihat laporan tanpa akses mengunduh laporan.
- e. Owner sebagai pengawas operasional apotek dan memantau kinerja staf. Hanya dapat melihat data obat, distributor, laporan dan statistik stok obat.

Setiap aktor harus melakukan login terlebih dahulu sebelum dapat mengakses fitur sistem dan dapat melakukan logout setelah selesai menggunakan sistem. Fitur notifikasi otomatis juga disediakan untuk admin dan apoteker agar memperoleh informasi penting secara real-time.

2. Diagram Alur Sistem

Menunjukkan alur proses pengguna mulai dari login, pengecekan *Role*, manajemen pengguna hingga akses ke fitur sesuai peran. Termasuk juga alur proses notifikasi stok menipis dan kadaluarsa. Berikut Arsitektur Sistem:

- a. Alur Sistem Login



Gambar 3. 2 Alur Sistem Login

Gambar 3.2 menunjukkan alur sistem login, berikut penjelasannya: Pengguna melakukan login melalui halaman login sistem, setelah berhasil login, sistem akan mengecek *Role* pengguna berdasarkan data yang tersimpan di database. Jika *Role* pengguna valid, pengguna akan diarahkan ke masing-masing *Role*, seperti Superadmin, Admin, atau Apoteker. Sehingga pengguna dapat masuk ke halaman yang sesuai dengan hak aksesnya, dan dapat mengakses fitur-fitur yang sesuai dengan perannya.

1. *SuperAdmin Permission*: Mengelola *User* yaitu Admin dan Apoteker
2. *Admin Permission*: Mengelola data obat, memantau stok obat dan tanggal kedaluarsa, melakukan transaksi obat masuk, mengelola distributor, dan mencetak laporan stok obat dan laporan transaksi, dan menerima notifikasi.
3. *Apoteker Permission*: Melakukan transaksi obat keluar, melihat stok obat dan transaksi obat masuk dan keluar tanpa akses unduh dan Menerima notifikasi.
4. *Owner Permission*: hanya melihat data obat, distributor dan laporan.

Jika *Role* pengguna tidak sesuai atau tidak memiliki izin, sistem akan menolak akses dan melakukan redirect kembali ke halaman login.

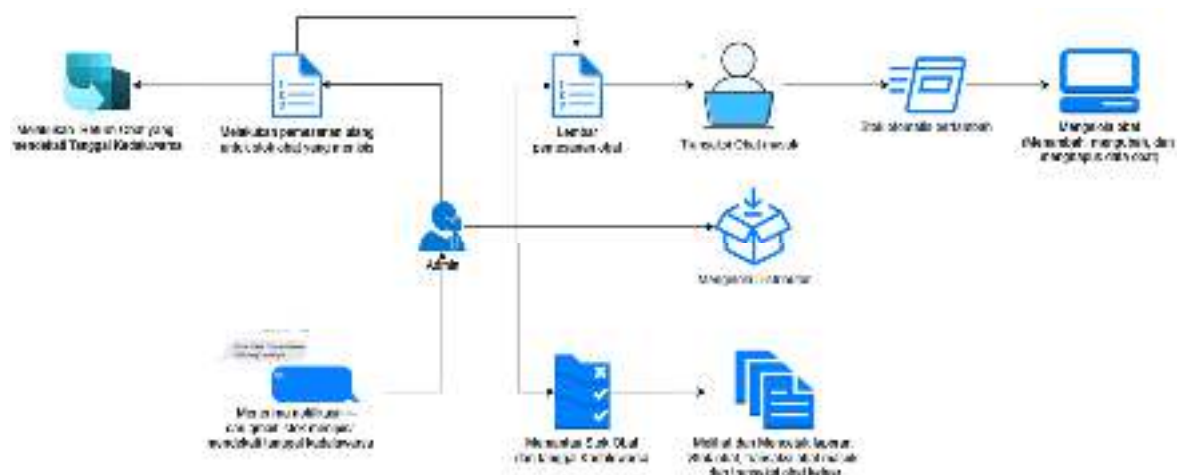
b. Alur Sistem Superadmin



Gambar 3. 3 Alur Sistem Superadmin

Gambar 3.3 menunjukkan alur sistem manajemen pengguna yang dilakukan oleh Superadmin. Superadmin bertanggung jawab dalam mengelola *User* (Admin dan Apoteker) serta melakukan penambahan, edit, atau hapus *User* sesuai kebutuhan Apotek, dan juga mengubah *Role* sesuai permintaan Apotek. Superadmin juga memantau *Log* aktifitas yang dilakukan oleh pengguna sistem.

c. Alur Sistem Admin



Gambar 3. 4 Alur Sistem Admin

Gambar 3.4 menunjukkan alur kerja sistem manajemen stok obat yang digunakan oleh Admin dalam mengelola persediaan obat di Apotek. Sistem ini dirancang untuk memudahkan proses pengelolaan stok, pemantauan kedaluwarsa,

serta pencatatan transaksi secara otomatis dan terstruktur.

1. Monitoring dan Notifikasi

Sistem secara otomatis melakukan pemantauan stok obat dan tanggal kedaluwarsa, jika terdapat obat yang mendekati tanggal kedaluwarsa atau stok menipis, sistem akan mengirimkan notifikasi ke Admin melalui dashboard.

2. Tindakan Lanjutan oleh Admin

Berdasarkan notifikasi tersebut, beberapa tindak lanjut yang akan dilakukan oleh Admin sebagai berikut:

- a. Melakukan pemesanan ulang untuk obat yang stoknya menipis.
- b. Melakukan *return* (pengembalian) obat yang mendekati tanggal kedaluwarsa kepada distributor.

3. Pengelolaan Pemesanan dan Transaksi

Setelah melakukan pemesanan ulang, surat pesanan obat tersebut akan diberikan kepada admin untuk dikelola seperti berikut:

- a. Admin membuat lembar pemesanan obat kepada distributor.
- b. Setelah obat diterima, Admin melakukan transaksi obat masuk di sistem.
- c. Sistem akan secara otomatis menambahkan stok obat yang masuk ke dalam database.

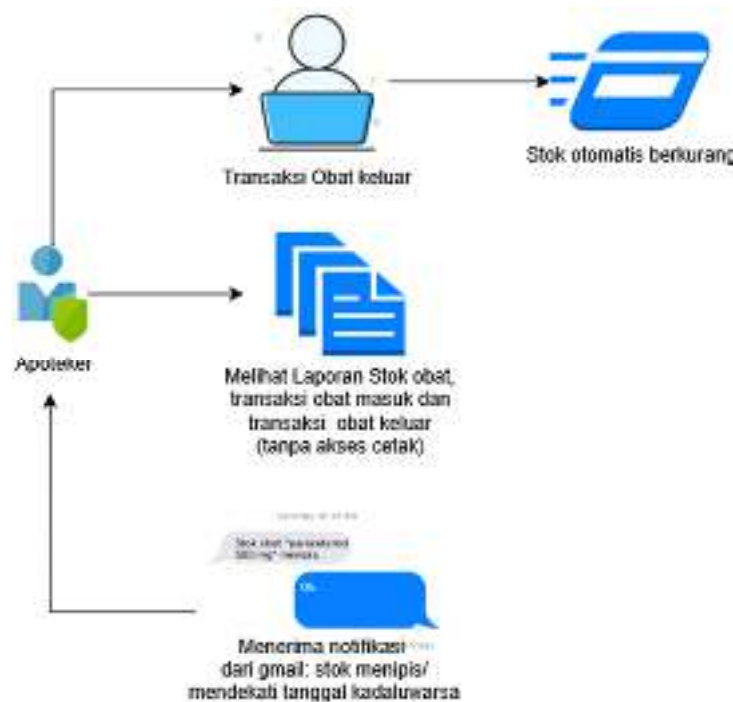
4. Manajemen Data

Admin juga dapat mengelola data obat, seperti menambah, mengubah, atau menghapus informasi obat. Selain itu, Admin memiliki akses untuk mengelola data distributor, sebagai mitra penyedia obat.

5. Pelaporan

Sistem menyediakan fitur untuk melihat dan mencetak laporan, yaitu laporan stok obat terkini, laporan transaksi obat masuk, dan laporan transaksi obat keluar.

d. Alur Sistem Apoteker



Gambar 3. 5 Alur Sistem Apoteker

Gambar 3.5 menunjukkan alur aktivitas Apoteker dalam sistem manajemen stok obat. Alur ini menggambarkan peran apoteker dalam proses pengeluaran obat dan pemantauan ketersediaan stok serta tanggal kedaluwarsa.

1. Transaksi Obat Keluar

Apoteker bertanggung jawab melakukan proses transaksi obat keluar, yaitu pencatatan obat yang diberikan kepada pasien atau unit layanan. Setiap kali terjadi transaksi, sistem secara otomatis akan mengurangi jumlah stok dari obat terkait di dalam database.

2. Pemantauan Data Transaksi

Apoteker memiliki akses untuk melihat data stok obat, serta rekam jejak transaksi obat masuk dan keluar. Namun, akses ini hanya bersifat baca (*read-only*), sehingga apoteker tidak memiliki hak untuk mencetak laporan atau melakukan perubahan data.

3. Notifikasi Otomatis

Sistem akan secara otomatis mengirimkan notifikasi melalui dashboard kepada apoteker jika stok suatu obat menipis, dan terdapat obat yang mendekati tanggal kedaluwarsa. Notifikasi ini memungkinkan apoteker untuk segera

Tabel 3. 1 Tabel *users*

Field	Tipe Data	Deskripsi
user_id	int(11)	Primary key, ID unik pengguna
name	varchar(50)	Nama lengkap pengguna
email	varchar(50)	Email unik pengguna
email_verified_at	timestamp	Waktu ketika email pengguna diverifikasi
password	varchar(255)	Password pengguna yang sudah di-hash
role_id	int(11)	ID role pengguna (relasi ke tabel roles)
created_at	timestamp	Tanggal dibuat
updated_at	timestamp	Tanggal diperbarui

b. Tabel *roles*

Menyimpan daftar peran (*Roles*) dalam sistem. Contohnya: superadmin, admin, dan apoteker. Setiap peran mengatur hak akses pengguna terhadap fitur-fitur dalam sistem.

Tabel 3. 2 Tabel *roles*

Field	Tipe Data	Deskripsi
role_id	int(11)	Primary key untuk data role
name	varchar(50)	Nama role (misal: admin)
guard_name	varchar(50)	Jenis guard Laravel (web/api)
created_at	timestamp	Tanggal dibuat
updated_at	timestamp	Tanggal diperbarui

c. Tabel *permissions*

Berisi daftar izin akses (*Permissions*) yang menggambarkan tindakan yang diperbolehkan, seperti obat.create, laporan.view, pengguna.manage, dan sebagainya.

Tabel 3. 3 Tabel *permissions*

Field	Tipe Data	Deskripsi
permission_id	int(11)	Primary key untuk data permission
name	varchar(50)	Nama permission (misal: lihat_data)
guard_name	varchar(50)	Jenis guard Laravel (web/api)

created_at	timestamp	Tanggal dibuat
updated_at	timestamp	Tanggal diperbarui

d. Tabel *role_has_permissions*

Tabel pivot yang menghubungkan *Roles* dan *Permissions*. Fungsinya untuk menentukan *Role* mana memiliki izin akses ke fitur tertentu.

Tabel 3. 4 Tabel *role_has_permissions*

Field	Tipe Data	Deskripsi
role_id	int(11)	FK ke tabel roles
permission_id	int(11)	FK ke tabel permissions

e. Tabel obats

Menyimpan informasi semua jenis obat yang dikelola di sistem. Data yang disimpan mencakup kode_obat, nama_obat, satuan, dan kategori. Obat memiliki relasi ke tabel stok_obats, transaksis, dan notifikasi.

Tabel 3. 5 Tabel obats

Field	Tipe Data	Deskripsi
kode_obat	varchar(30)	Primary key, kode unik obat
nama_obat	varchar(50)	Nama lengkap dari obat
satuan	varchar(50)	Satuan obat, contoh: strip, botol
kategori	varchar(50)	Kategori obat, contoh: antibiotik
created_at	timestamp	Tanggal dibuat
updated_at	timestamp	Tanggal diperbarui

f. Tabel stok_obats

Mencatat jumlah stok dan tanggal kedaluwarsa dari masing-masing obat. Berguna untuk mengetahui ketersediaan obat serta manajemen stok berdasarkan kedaluwarsa.

Tabel 3. 6 Tabel stok_obats

Field	Tipe Data	Deskripsi
stok_obat_id	int(11)	Primary key, ID unik stok
kode_obat	varchar(30)	Foreign key ke obats.kode_obat
tanggal_expired	data	Tanggal kadaluarsa

jumlah	int(11)	Jumlah stok tersedia
created_at	timestamp	Tanggal dibuat
updated_at	timestamp	Tanggal diperbarui

g. Tabel distributors

Menyimpan data distributor yang menyuplai obat ke apotek. Informasi meliputi nama perusahaan, alamat, kontak, dan penanggung jawab. Distributor ini menjadi referensi dalam setiap transaksi masuk.

Tabel 3. 7 Tabel distributors

Field	Tipe Data	Deskripsi
distributor_id	int(11)	Primary key, ID unik distributor
nama_perusahaan	varchar(50)	Nama perusahaan distributor
alamat	text	Alamat lengkap distributor
telepon	varchar(20)	Nomor telepon distributor
email	varchar(50)	Email distributor
penanggung_jawab	varchar(50)	Nama penanggung jawab
status_aktif	boolean	Status aktif/tidak aktif
created_at	timestamp	Tanggal dibuat
updated_at	timestamp	Tanggal diperbarui

h. Tabel transaksi

Mencatat seluruh aktivitas transaksi obat, baik masuk (penerimaan dari distributor) maupun keluar (pengeluaran kepada pasien). Transaksi masuk akan menambah stok, sedangkan transaksi keluar akan mengurangnya.

Tabel 3. 8 Tabel transaksi

Field	Tipe Data	Deskripsi
transaksi_id	int(11)	Primary key, ID unik transaksi
kode_obat	varchar(30)	Foreign key ke obats.kode_obat
distributor_id	int(11)	Foreign key ke distributors.distributor_id
tipe	enum('masuk','keluar')	Jenis transaksi obat (masuk atau keluar)

nama_obat	varchar(50)	Nama obat yang ditransaksikan
tanggal	date	Tanggal transaksi dilakukan
jumlah	int(11)	Jumlah obat yang ditransaksikan
satuan	varchar(50)	Satuan obat
kategori	varchar(50)	Kategori obat
expired_date	date	Tanggal kadaluarsa obat
created_at	timestamp	Tanggal dibuat
updated_at	timestamp	Tanggal diperbarui

i. Tabel notifikasi

Digunakan untuk mengirim peringatan kepada admin atau apoteker terkait stok obat yang menipis atau obat yang akan kadaluarsa. Jenis notifikasi ditentukan oleh tipe (misalnya: stok_menipis, kadaluarsa. Tabel ini juga menyimpan informasi jumlah, pesan, dan status_baca.

Tabel 3. 9 Tabel notifikasi

Field	Tipe Data	Deskripsi
notifikasi_id	int(11)	Primary key, ID unik notifikasi
user_id	int(11)	Foreign key ke users.user_id
kode_obat	varchar(30)	Foreign key ke obats.kode_obat
tipe	enum('stok_habis', 'kadaluarsa')	Jenis notifikasi
pesan	text	Isi pesan notifikasi
jumlah	int(11)	Jumlah obat terkait
tanggal_expired	date	Tanggal kadaluarsa obat
dikirim_via	enum('email', 'dashboard')	Media notifikasi dikirim
status_baca	boolean	Status apakah notifikasi sudah dibaca
created_at	timestamp	Tanggal dibuat
updated_at	timestamp	Tanggal diperbarui

j. Tabel *activity_log*

Digunakan untuk Menyimpan catatan aktivitas pengguna di sistem.

Tabel 3. 10 *activity_log*

Field	Tipe Data	Deskripsi
id	int(11)	Primary key, ID aktivitas
user_id	int(11)	ID pengguna yang melakukan aktivitas
action	varchar(100)	Jenis aksi (create, update, delete, login, logout, dll)
module	varchar(100)	Modul yang diakses (stok_obat, transaksi, dll)
description	text	Detail aktivitas
ip_address	varchar(45)	Alamat IP pengguna (IPv4/IPv6)
user_agent	text	Informasi browser/perangkat pengguna
created_at	timestamp	Waktu aktivitas terjadi

4. Desain Antarmuka (UI/UX)

Menyusun tampilan halaman login, dashboard masing-masing peran, halaman data obat, halaman distributor, halaman transaksi obat masuk dan keluar, halaman laporan, halaman manajemen *User*, dan halaman RBAC.

a. Halaman Login

Halaman Login merupakan halaman bagi *User* untuk masuk kedalam sistem. Halaman login diisi dengan email dan *Password* sesuai dengan yang diberikan oleh superadmin.



Gambar 3. 7 Halaman Login

b. Dashboard Superadmin

Dashboard Superadmin merupakan halaman setelah superadmin melakukan proses otentikasi login berhasil berdasarkan peran. Halaman ini terdiri dari submenu yaitu manajemen *User*, dan *Log* aktifitas.



Gambar 3. 8 Dashboard Superadmin

c. Dashboard Admin

Dashboard Admin merupakan halaman setelah admin melakukan proses otentikasi login berhasil berdasarkan peran. Halaman ini terdiri dari submenu yaitu

data obat, transaksi obat, distributor, dan laporan.



Gambar 3. 9 Dashboard Admin

d. Dashboard Apoteker

Dashboard Apoteker merupakan halaman setelah apoteker melakukan proses otentikasi login berhasil berdasarkan peran. Halaman ini terdiri dari submenu yaitu data obat, transaksi obat, distributor, dan laporan.



Gambar 3. 10 Dashboard Apoteker

e. Dashboard Owner

Halaman ini hanya sebagai pengawas operasional apotek dan memantau kinerja staf. Hanya dapat melihat data obat, distributor, laporan dan statistik stok obat.



Gambar 3. 11 Dashboard Owner

f. Halaman data obat

Pada halaman ini admin dapat mengubah, menghapus dan menambah obat baru pada data obat seperti nama obat, satuan, kategori, stok, dan tanggal kedaluarsa. Sementara Apoteker hanya dapat melihat data obat.

NO	KODE OBAT	NAMA OBAT	SATUAN	KATEGORI	STOK	EXPIRED	KETERANGAN	AYO
1.1	00001	Amoxicillin	Botol	Antibiotik	5	22-09-2025	Stok menipis	<button>Ubah</button> <button>Hapus</button> <button>Tambah</button>
1.2	00002	Amoxicillin	Botol	Antibiotik	7	20-07-2025	Stok menipis & Sudah kadaluarsa	<button>Ubah</button> <button>Hapus</button> <button>Tambah</button>
2.1	00001	Parasetamol	Tablet	Analgesik	50	17-06-2026	Aman	<button>Ubah</button> <button>Hapus</button> <button>Tambah</button>
2.2	00001	Parasetamol	Tablet	Analgesik	2	21-07-2027	Stok menipis	<button>Ubah</button> <button>Hapus</button> <button>Tambah</button>
2.3	00001	Parasetamol	Tablet	Analgesik	2	17-11-2027	Stok menipis	<button>Ubah</button> <button>Hapus</button> <button>Tambah</button>

Gambar 3. 12 Halaman data obat Admin

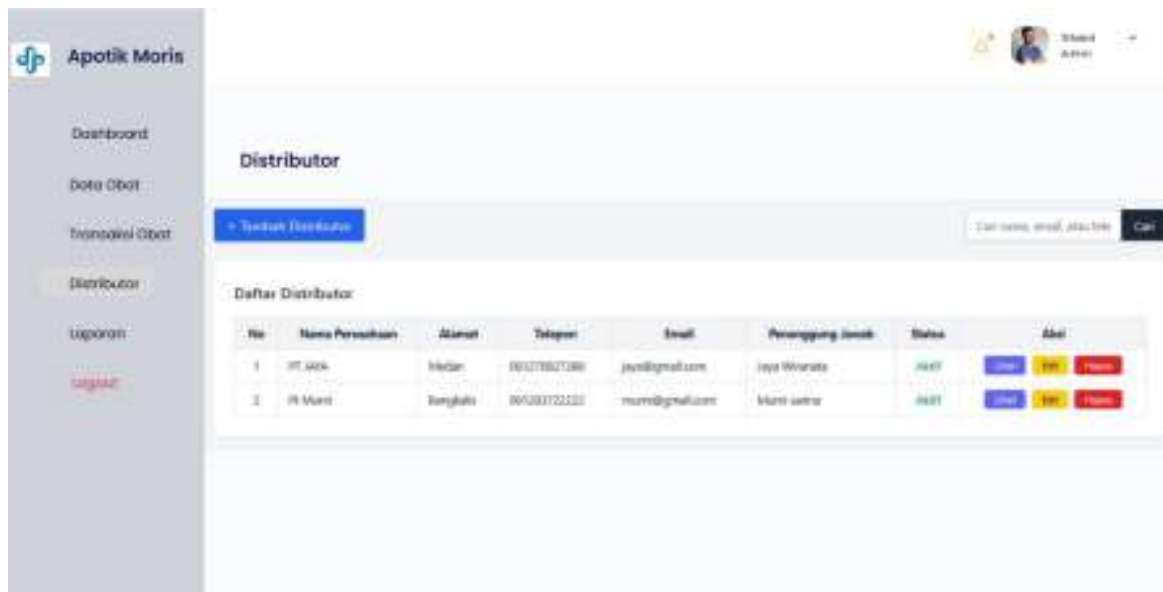
NO	KODE OBAT	NAMA OBAT	SATUAN	KATEGORI	STOK	EXPIRED	KETERANGAN	AYO
1.1	00001	Amoxicillin	Botol	Antibiotik	5	22-09-2025	Stok menipis	<button>Ubah</button>
1.2	00002	Amoxicillin	Botol	Antibiotik	7	20-07-2025	Stok menipis & Sudah kadaluarsa	<button>Ubah</button>
2.1	00001	Parasetamol	Tablet	Analgesik	50	17-06-2026	Aman	<button>Ubah</button>
2.2	00001	Parasetamol	Tablet	Analgesik	2	21-07-2027	Stok menipis	<button>Ubah</button>
2.3	00001	Parasetamol	Tablet	Analgesik	2	17-11-2027	Stok menipis	<button>Ubah</button>

Gambar 3. 13 Halaman data obat Apoteker dan Owner

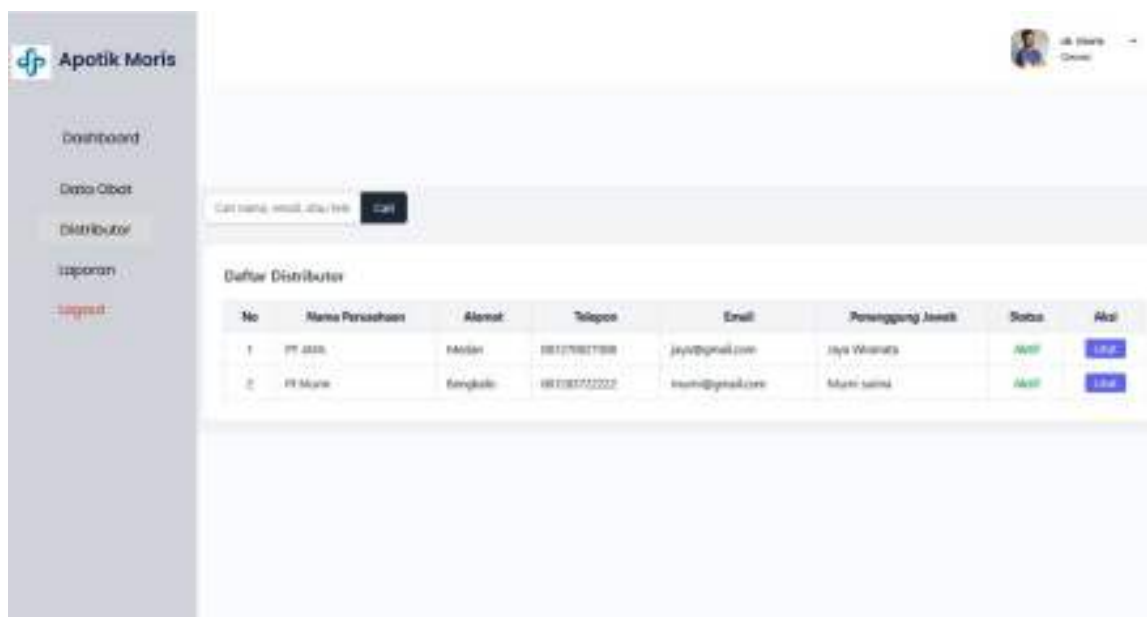
g. Halaman distributor

Halaman ini hanya dapat diakses oleh admin dan owner. Admin dapat mengubah, menghapus dan menambah distributor seperti nama distributor, email,

nomor telepon, dan alamat sedangkan owner hanya dapat melihat.



Gambar 3. 14 Halaman distributor Admin



Gambar 3. 15 Halaman Distributor Owner

h. Halaman transaksi obat masuk dan keluar

Pada halaman ini admin dapat melakukan transaksi obat masuk dari distributor dengan menambahkan nama obat, tanggal, jumlah, satuan, kategori, dan distributor. Sementara apoteker dapat melakukan transaksi obat keluar untuk pasien dengan memasukkan nama obat, tanggal, jumlah, kategori, dan tanggal

kedaluarsa. Yang dimana nantinya akan masuk ke dalam laporan.

TRANSAKSI OBAT

Daftar Transaksi Obat Masuk

No	Nama Obat	Dokter	Tanggal	Jumlah	Status	Kategori	ID	Aksi
1	Parasetamol	Dr. Jhon	06-07-2023	50	Tersedia	Analgetik	17-06-2023	Detail Hapus
2	Amoxicillin	Dr. Murni	06-07-2023	10	Tersedia	Antibiotik	22-06-2023	Detail Hapus
3	Parasetamol	Dr. Jhon	06-07-2023	20	Tersedia	Analgetik	17-06-2023	Detail Hapus
4	Parasetamol	Dr. Jhon	06-07-2023	2	Tersedia	Analgetik	07-07-2027	Detail Hapus
5	Amoxicillin	Dr. Jhon	06-07-2023	10	Tersedia	Antibiotik	16-07-2023	Detail Hapus

Daftar Transaksi Obat Keluar

No	Nama Obat	Tanggal	Jumlah	Status	Kategori	ID	Aksi
1	Amoxicillin	06-07-2023	2	Tersedia	Antibiotik	22-06-2023	Detail
2	Parasetamol	06-07-2023	2	Tersedia	Analgetik	17-06-2023	Detail
3	Amoxicillin	06-07-2023	1	Tersedia	Antibiotik	22-06-2023	Detail
4	Parasetamol	06-07-2023	2	Tersedia	Analgetik	17-06-2023	Detail
5	Amoxicillin	06-07-2023	1	Tersedia	Antibiotik	22-06-2023	Detail
6	Parasetamol	06-07-2023	2	Tersedia	Analgetik	17-06-2023	Detail
7	Parasetamol	06-07-2023	2	Tersedia	Analgetik	17-06-2023	Detail

Gambar 3. 16 Halaman transaksi obat masuk Admin

TRANSAKSI OBAT

Daftar Transaksi Obat Masuk

No	Nama Obat	Dokter	Tanggal	Jumlah	Status	Kategori	ID	Aksi
1	Parasetamol	Dr. Jhon	06-07-2023	50	Tersedia	Analgetik	17-06-2023	Detail Hapus
2	Amoxicillin	Dr. Murni	06-07-2023	10	Tersedia	Antibiotik	22-06-2023	Detail Hapus
3	Parasetamol	Dr. Jhon	06-07-2023	20	Tersedia	Analgetik	17-06-2023	Detail Hapus
4	Parasetamol	Dr. Jhon	06-07-2023	2	Tersedia	Analgetik	07-07-2027	Detail Hapus
5	Amoxicillin	Dr. Jhon	06-07-2023	10	Tersedia	Antibiotik	16-07-2023	Detail Hapus

Daftar Transaksi Obat Keluar

No	Nama Obat	Tanggal	Jumlah	Status	Kategori	ID	Aksi
1	Amoxicillin	06-07-2023	2	Tersedia	Antibiotik	22-06-2023	Detail Hapus
2	Parasetamol	06-07-2023	2	Tersedia	Analgetik	17-06-2023	Detail Hapus
3	Amoxicillin	06-07-2023	1	Tersedia	Antibiotik	22-06-2023	Detail Hapus
4	Parasetamol	06-07-2023	2	Tersedia	Analgetik	17-06-2023	Detail Hapus
5	Amoxicillin	06-07-2023	1	Tersedia	Antibiotik	22-06-2023	Detail Hapus
6	Parasetamol	06-07-2023	2	Tersedia	Analgetik	17-06-2023	Detail Hapus
7	Parasetamol	06-07-2023	2	Tersedia	Analgetik	17-06-2023	Detail Hapus

Gambar 3. 17 Halaman transaksi obat keluar Apoteker

i. Halaman laporan

Halaman laporan ini terdiri dari laporan stok obat, laporan transaksi obat masuk, dan laporan transaksi obat keluar. Pada halaman laporan ini apoteker hanya dapat melihat apa saja yang masuk pada hari itu dan obat apa saja yang keluar pada hari itu beserta data-data terkait dengan obat masuk dan obat yang keluar, sementara admin dapat melihat dan mencetak seluruh laporan.

LAPORAN

Laporan Stok Obat

No	Kode Obat	Nama Obat	Satuan	Kategori	Stok	Expired	Keterangan
1	08001	Parasetamol	Tablet	Antipiretik	100	28-08-2025	Adapt
2	08002	Vitamin C	Botol	Antibiotik	8	18-08-2025	Obat pengganti
3	08003	Amoxicillin	Botol	Antibiotik	100	02-08-2025	Memiliki kadaluarsa

Rincian Transaksi Obat Masuk

No	Nama Obat	Distributor	Tanggal	Jumlah	Satuan	Kategori	Stok
1	Parasetamol	P1 08001	27-08-2023	50	Tablet	Antipiretik	28-08-2025
2	Amoxicillin	P1 08002	27-08-2023	50	Botol	Antibiotik	02-07-2024

Rincian Transaksi Obat Keluar

No	Nama Obat	Tanggal	Jumlah	Satuan	Kategori	Stok
1	Parasetamol	27-08-2023	10	Tablet	Antipiretik	28-08-2025
2	Parasetamol	28-08-2023	20	Tablet	Antipiretik	28-08-2025

Gambar 3. 18 Halaman laporan Admin dan Owner

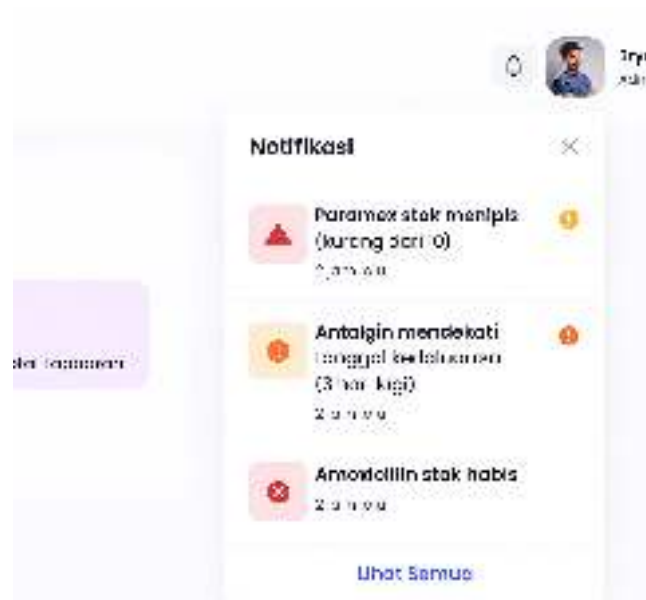
Laporan Stok Obat

No	Kode Obat	Nama Obat	Satuan	Kategori	Stok	Expired	Keterangan
1.1	08076	Combamtrin 125 mg	Tablet	Antipiretik	20	17-12-2025	Sudah kedaluwarsa
2.1	08011	Omeprazole 20 mg	Kapsul	Gastrointestinal	60	12-12-2025	Sudah kedaluwarsa
3.1	08075	Parasetamol	Tablet	Respiratori	25	26-12-2025	Sudah kedaluwarsa
4.1	08099	Plester Kain	Roll	Luka	25	11-12-2025	Sudah kedaluwarsa
5.1	08140	Salicylic Acid 2%	Botol	Kulit	5	10-12-2025	Stok minim & Sudah kedaluwarsa

Gambar 3. 19 Halaman laporan Apoteker

j. Halaman Notifikasi

Halaman ini merupakan tampilan notifikasi untuk admin dan apoteker, yang menampilkan informasi penting terkait stok obat menipis dan obat yang mendekati tanggal kedaluwarsa.



Gambar 3. 20 Halaman Notifikasi

k. Halaman Kelola *User*

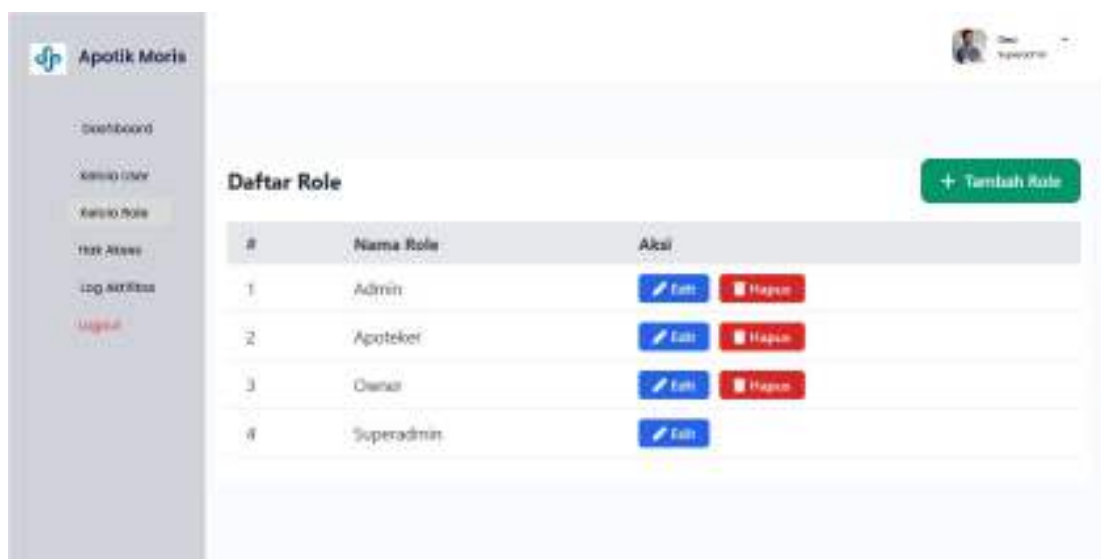
Halaman ini merupakan halaman Role untuk superadmin, dimana superadmin dapat mengelola User dengan mengubah, menghapus dan menambah *User* seperti nama, email, *Password*, *Role* dan status.



Gambar 3. 21 Halaman Kelola User

1. Halaman Kelola Role

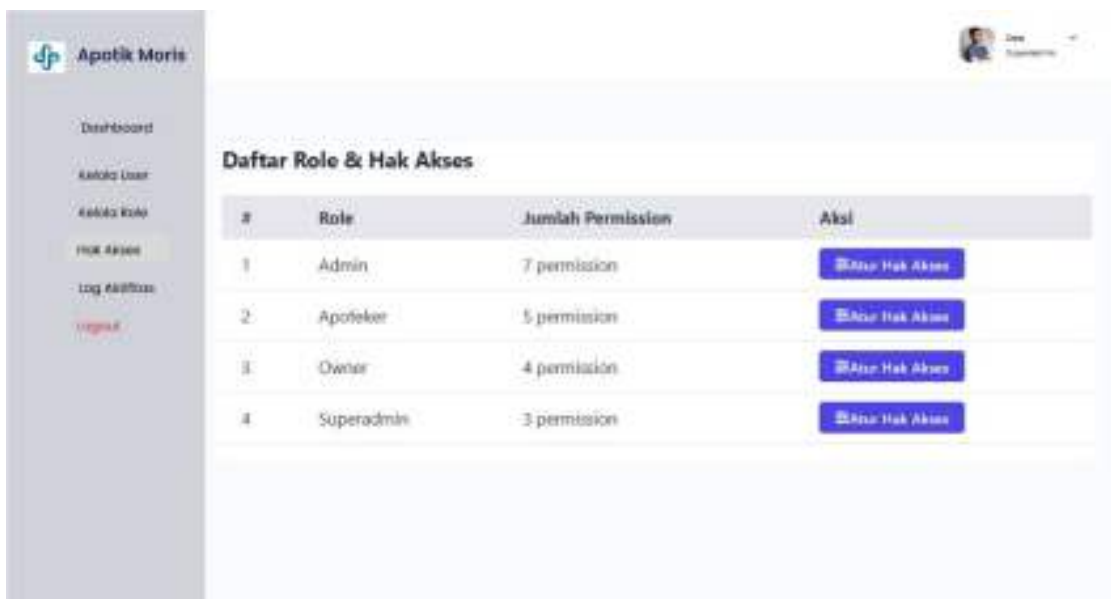
Halaman ini merupakan halaman Kelola Role untuk Superadmin, dimana Superadmin dapat mengatur daftar peran dalam sistem dengan menambah Role baru, mengubah nama Role, serta menghapus Role yang sudah tidak digunakan. Fitur ini memastikan setiap pengguna memiliki hak akses sesuai perannya masing-masing.



Gambar 3. 22 Halaman Kelola Role

m. Halaman Hak Akses

Halaman ini merupakan halaman Hak Akses untuk Superadmin, dimana Superadmin dapat melihat jumlah permission pada setiap Role dan mengatur hak akses masing-masing Role melalui tombol 'Atur Hak Akses'. Fitur ini memastikan setiap Role hanya dapat menjalankan fungsi sesuai kewenangan yang telah ditetapkan.”



#	Role	Jumlah Permission	Aksi
1	Admin	7 permission	Atur Hak Akses
2	Apoteker	5 permission	Atur Hak Akses
3	Owner	4 permission	Atur Hak Akses
4	Superadmin	3 permission	Atur Hak Akses

Gambar 3. 23 Hak Akses

n. Halaman Atur Hak Akses

Halaman ini merupakan halaman pengaturan permission untuk setiap Role, dimana Superadmin dapat menentukan izin apa saja yang boleh diakses oleh Role tersebut. Pada halaman ini, Superadmin dapat mencentang atau menghapus centang pada daftar permission, memilih semua izin sekaligus, menghapus semua izin, serta menyimpan perubahan. Fitur ini memastikan setiap Role hanya memperoleh akses sesuai kebutuhan dan tanggung jawabnya.



Gambar 3. 24 Halaman Atur Hak Akses

5. Analisis Role

Dalam penerapan RBAC pada sistem informasi manajemen stok obat, terdapat tiga role utama:

1. Superadmin: Role ini ditambahkan sebagai pengelola teknis yang mengatur akun pengguna, role, dan permission. Keberadaannya penting untuk mencegah admin maupun apoteker memiliki kontrol penuh, sehingga keamanan sistem tetap terjaga. Superadmin tidak terlibat dalam pengelolaan obat maupun transaksi, melainkan berfokus pada pengawasan dan pengaturan hak akses.
2. Admin: Admin bertanggung jawab terhadap data obat, distributor, transaksi masuk, dan laporan. Peran ini diperlukan agar pencatatan stok lebih akurat serta laporan dapat disusun dengan baik. Admin juga menerima notifikasi stok menipis, tetapi aksesnya dibatasi hanya pada fungsi pengelolaan data tanpa kewenangan mengelola akun pengguna atau transaksi keluar.
3. Apoteker: Apoteker berfokus pada pelayanan pasien dengan mencatat transaksi obat keluar serta memantau stok harian. Perannya dipisahkan dari admin agar aksesnya terbatas sesuai kebutuhan operasional. Apoteker dapat melihat data dan laporan serta menerima notifikasi stok, tetapi tidak dapat

mengubah data obat, mencetak laporan, atau mengatur akun pengguna.

4. Owner: berperan sebagai pengawas operasional yang memastikan seluruh proses pencatatan obat berjalan dengan benar. Role ini tidak melakukan input data, tetapi berfokus pada evaluasi, monitoring, dan memantau kinerja staf.

Dengan pembagian ini, Superadmin berfungsi sebagai pengawas teknis, Admin sebagai pengelola data, dan Apoteker sebagai pelaksana operasional. Pemisahan peran ini mendukung keamanan, efisiensi, dan kepatuhan terhadap prinsip RBAC.

6. Perancangan Hak Akses Berbasis *Role* (RBAC)

Sistem dirancang untuk menerapkan kontrol akses berbasis peran (RBAC) menggunakan Laravel Spatie. Terdapat empat peran utama: Superadmin, Admin, Apoteker, dan Owner. Masing-masing peran hanya dapat mengakses halaman dan fitur tertentu sesuai tanggung jawabnya. Berikut rencana hak akses untuk masing-masing peran:

```

class RolePermissionSeeder extends Seeder
{
    /** @var array */
    public function run(): void
    {
        // 1. Definisikan semua permission yang digunakan sistem
        $permissions = [
            'kelola pengguna',
            'kelola obat',
            'lihat obat',
            'kelola distributor',
            'lihat distributor',
            'transaksi obat masuk',
            'transaksi obat keluar',
            'lihat laporan',
            'cetak laporan',
        ];

        foreach ($permissions as $perm) {
            Permission::firstOrCreate(attributes: [
                'name' => $perm,
                'guard_name' => 'web',
            ]);
        }

        // 2. Role Superadmin + mendapat semua permission
        $superadmin = Role::firstOrCreate(attributes: [
            'name' => 'superadmin',
            'guard_name' => 'web',
        ]);
        $superadmin->syncPermissions(Permission::all());
    }
}

```

Gambar 3. 25 Perancangan Hak Akses RBAC di Laravel Spatie [12].

Perancangan RBAC di atas digunakan untuk menginisialisasi *role* dan *permission* awal. Selanjutnya, Superadmin dapat mengubah relasi *role-permission* melalui antarmuka Kelola *Role* dan Hak Akses, sehingga mekanisme RBAC yang diterapkan bersifat dinamis.

Dengan rancangan tersebut, setiap *role* hanya mengakses fitur yang relevan dengan tugasnya. Jika di kemudian hari terjadi perubahan kebutuhan (misalnya penambahan *role* baru atau penyesuaian hak akses), Superadmin cukup mengubah relasi *role-permission* melalui modul RBAC pada antarmuka web, tanpa perlu memodifikasi kode sumber. Hal ini menunjukkan bahwa mekanisme RBAC yang digunakan pada sistem bersifat dinamis dan fleksibel.

3.4.2 Implementasi (*Implementation*)

Tahap implementasi dilakukan dengan membangun sistem berdasarkan desain dan struktur database yang telah dirancang pada tahap sebelumnya. Proses implementasi ini memanfaatkan framework Laravel 11, database MySQL, serta package Laravel Spatie *Permission* untuk sistem keamanan *Role-Based Access Control* (RBAC). Pada tahap ini, seluruh rancangan antarmuka, proses bisnis, dan hak akses diimplementasikan ke dalam bentuk kode program.

3.4.3 Pengujian (*Testing*)

Setelah implementasi selesai, tahap pengujian akan dilakukan untuk memastikan seluruh fungsi sistem berjalan sesuai kebutuhan. Pengujian akan menggunakan metode *Black Box Testing* untuk menguji input dan output sistem, serta memastikan bahwa setiap peran pengguna hanya dapat mengakses fitur yang sesuai. Uji coba juga akan dilakukan bersama pengguna akhir (admin, apoteker dan Owner) sebagai bagian dari validasi sistem.

3.4.4 Pemeliharaan (*Maintenance*)

Tahap pemeliharaan akan dilakukan setelah sistem diterapkan di Apotek Pratama dr. Moris. Pemeliharaan mencakup perbaikan *bug*, pembaruan fitur sesuai kebutuhan, serta pemantauan kinerja sistem secara berkala. Selain itu, pelatihan dan panduan penggunaan juga akan diberikan kepada pengguna agar dapat mengoperasikan sistem secara mandiri dan optimal.

BAB IV

HASIL DAN PENGUJIAN

4.1 Hasil Implementasi

Tahap implementasi merupakan realisasi dari rancangan sistem pada Bab III menjadi sebuah aplikasi yang dapat dijalankan dan digunakan oleh pengguna. Implementasi mencakup konfigurasi lingkungan pengembangan, pembuatan basis data, penerapan RBAC menggunakan Laravel Spatie, serta pembangunan modul-modul fungsional seperti data obat, transaksi, distributor, laporan, notifikasi, dan *log* aktivitas.

4.1.1 Lingkungan Implementasi

Implementasi sistem informasi manajemen stok obat pada Apotek Moris dilakukan pada spesifikasi lingkungan sebagai berikut:

1. Perangkat keras
 - a. Laptop dengan prosesor minimal Intel Core i5 atau setara
 - b. RAM minimal 8 GB
 - c. Ruang penyimpanan yang cukup untuk web server dan basis data
2. Perangkat lunak
 - a. Sistem operasi: Windows 10/11
 - b. Web server: XAMPP (Apache, PHP, MySQL)
 - c. Bahasa pemrograman: PHP 8.x
 - d. Framework: Laravel 11
 - e. Manajemen paket: Composer
 - f. Basis data: MySQL
 - g. Browser: Google Chrome / Mozilla Firefox
 - h. Text editor/IDE: VS Code

Konfigurasi awal dilakukan dengan menginstal XAMPP, mengaktifkan modul Apache dan MySQL, kemudian membuat database baru untuk sistem manajemen stok obat. Setelah itu, proyek Laravel 11 diinisialisasi menggunakan Composer dan dihubungkan dengan database melalui file `.env` pada proyek Laravel.

4.1.2 Implementasi Basis Data

Struktur basis data yang telah dirancang pada Bab III diimplementasikan menggunakan fitur *migration* pada Laravel. Tabel-tabel utama seperti *users*, *roles*, *permissions*, *role_has_permissions*, *obats*, *stok_obats*, *distributors*, *transaksis*, *notifikasi*, dan *activity_log* dibuat sesuai rancangan sebelumnya.

Beberapa langkah utama pada tahap ini adalah:

1. Membuat file *migration* untuk masing-masing tabel beserta relasi kunci asing (*foreign key*).
2. Menjalankan perintah php artisan *migrate* untuk membentuk struktur tabel di MySQL.
3. Menambahkan data awal (*seeding*) untuk:
 - a. *Role*: superadmin, admin, apoteker, dan owner
 - b. *Permission* sesuai kebutuhan fitur yaitu: kelola obat, lihat obat, transaksi obat masuk, transaksi obat keluar, lihat distributor, kelola distributor, lihat laporan, cetak laporan, dan sebagainya.

Dengan demikian, struktur basis data yang telah dirancang secara konseptual pada Bab III benar-benar terwujud dalam bentuk tabel yang siap digunakan oleh aplikasi.

4.1.3 Implementasi Role-Based Access Control (RBAC)

Penerapan RBAC dilakukan menggunakan package Laravel Spatie *Permission*. Implementasi dilakukan melalui beberapa tahap berikut:

1. Menambahkan *Trait HasRoles* pada Model *User*

Model *User* diberikan *trait HasRoles* sehingga setiap *user* dapat diberi *role* dan *permission* sesuai kebutuhan.

```
// app/Models/User.php
<?php
namespace App\Models;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Illuminate\Notifications\Notifiable;
use Spatie\Permission\Traits\HasRoles;
```

```

class User extends Authenticatable
{
    use Notifiable, HasRoles;
    /**
     * Guard yang dipakai Spatie Permission.
     * Pastikan sama dengan guard yang dipakai untuk
    login (biasanya 'web').
     */
    protected $guard_name = 'web';
    /**
     * Kolom yang boleh diisi mass assignment.
     */
    protected $fillable = [
        'name',
        'email',
        'password',
    ];
    // ...
}

```

2. Inisialisasi *Role* dan *Permission*

Role dan permission diinisialisasi melalui seeder seperti sebagai berikut:

- a. *Role* superadmin memiliki akses penuh terhadap pengelolaan *user*, *role*, dan hak akses.
- b. *Role* admin memiliki hak untuk mengelola data obat, distributor, transaksi obat masuk, laporan, dan menerima notifikasi.
- c. *Role* apoteker memiliki hak mengelola transaksi obat keluar serta melihat stok dan laporan tanpa hak mengubah atau mencetak.
- d. *Role* owner memiliki hak membaca data obat, distributor, dan laporan, serta mencetak laporan sebagai bahan evaluasi.

Berikut kodingannya:

```
// database/seeder/RolePermissionSeeder.php
use Spatie\Permission\Models\Role;
use Spatie\Permission\Models\Permission;

class RolePermissionSeeder extends Seeder
{
    public function run(): void
    {
        // daftar permission
        $permissions = [
            'kelola pengguna',      // Superadmin
            'kelola obat',         // Admin
            'lihat obat',          // Admin & Apoteker
            'kelola distributor',    // Admin
            'lihat distributor',     // Admin & Apoteker
            'transaksi obat masuk', // Admin
            'transaksi obat keluar', // Apoteker
            'lihat laporan',        // Admin & Apoteker
            & Owner
            'cetak laporan',        // Admin & Owner
        ];
        // membuat permission jika belum ada
        foreach ($permissions as $perm) {
            Permission::firstOrCreate([
                'name' => $perm,
                'guard_name' => 'web',
            ]);
        }
    }
}
```

```

// role default
$superadmin = Role::firstOrCreate(['name' =>
'superadmin', 'guard_name' => 'web']);
$admin      = Role::firstOrCreate(['name' =>
'admin', 'guard_name' => 'web']);
$apoteker   = Role::firstOrCreate(['name' =>
'apoteker', 'guard_name' => 'web']);

// mapping default:
// superadmin mendapat semua permission
$superadmin->syncPermissions($permissions);
// admin: fokus ke pengelolaan data & transaksi
masuk
$admin->syncPermissions([
    'kelola obat',
    'lihat obat',
    'kelola distributor',
    'lihat distributor',
    'transaksi obat masuk',
    'lihat laporan',
    'cetak laporan',
]);
// apoteker: fokus ke transaksi keluar dan
monitoring
$apoteker->syncPermissions([
    'lihat obat',
    'lihat distributor',
    'transaksi obat keluar',
    'lihat laporan',
]);

```



```
}  
}
```

3. Pengaturan *Role* dan *Permission* pada *User*

Penugasan *role* dan *permission* ke setiap *user* diatur melalui Superadmin\UserController. Ketika superadmin menambah *user* baru, sistem akan otomatis memberikan *role* dan (*opsional*) *permission* tambahan berdasarkan input form. Berikut potongan kode programnya:

```
// app/Http/Controllers/Superadmin/UserController.php  
  
public function store(Request $request)  
{  
    $data = $request->validate([  
        'name' => ['required', 'string',  
'max:255'],  
        'email' => ['required', 'email', 'max:255',  
'unique:users,email'],  
        'password' => ['required', 'string',  
'min:6'],  
        'role' => ['required', 'string',  
Rule::exists('roles', 'name')],  
        'permissions' => ['nullable', 'array'],  
        'permissions.*' => ['string'],  
    ]);  
    $user = User::create([  
        'name' => $data['name'],  
        'email' => $data['email'],  
        'password' => bcrypt($data['password']),  
    ]);  
}
```

```

// berikan role utama ke user
$user->assignRole($data['role']);
// jika ada permission tambahan, disimpan juga
if (!empty($data['permissions'])) {
    $user->syncPermissions($data['permissions']);
}
// log aktivitas pembuatan user
log_activity(
    'Membuat user baru: ' . $user->name .
    ' (' . $user->email . ') dengan role: ' .
    $data['role']
);
return redirect()
    ->route('superadmin.user.index')
    ->with('success', 'User baru berhasil dibuat.');
```

Pada saat dilakukan proses edit user, role dan permission yang dimiliki user juga diperbarui menggunakan `syncRoles()` dan `syncPermissions()`. Dengan cara ini, setiap perubahan hak akses terekam secara konsisten dan tercatat dalam log aktivitas.

Selain itu, controller ini juga menyiapkan kelompok permission yang ditampilkan di form “Kelola User”, misalnya kelompok Pengguna, Obat, Distributor, Transaksi, Laporan, dan RBAC, sehingga superadmin dapat memberi hak akses secara lebih terstruktur:

```

// cuplikan pengelompokan permission di UserController
$permissionGroups = [
    'Pengguna' => [
        'kelola pengguna',
    ],
    'Obat' => [
```

```

        'kelola obat',
        'lihat obat',
    ],
    'Distributor' => [
        'kelola distributor',
        'lihat distributor',
    ],
    'Transaksi' => [
        'lihat transaksi',
        'transaksi obat masuk',
        'transaksi obat keluar',
    ],
    'Laporan' => [
        'lihat laporan',
        'cetak laporan',
    ],
    'RBAC' => [
        'kelola role',
        'kelola hak akses',
    ],
];

```

4. Pengaturan *Middleware* dan *Guard*

Middleware auth dan *role* digunakan untuk membatasi akses ke route tertentu seperti sebagai berikut:

- a. Route dengan prefix */superadmin* hanya dapat diakses oleh *user* dengan *role* superadmin.
- b. Route untuk pengelolaan obat hanya dapat diakses oleh *user* dengan *role* admin yang memiliki *permission* kelola obat.
- c. Route laporan untuk owner dan admin memiliki *permission* lihat

laporan atau cetak laporan.

Kode program untuk prefix Superadmin:

```
// routes/web.php
// SUPERADMIN AREA (khusus role superadmin)
Route::prefix('superadmin')
    ->middleware(['auth', 'role:superadmin'])
    ->name('superadmin.')
    ->group(function () {

        // Dashboard khusus superadmin
        Route::get('/dashboard',
[SuperAdminController::class, 'index'])
            ->name('dashboard');

        // Kelola User (permission "kelola pengguna")
        Route::resource('user', UserController::class)
            ->except(['show'])
            ->middleware('permission:kelola pengguna');

        // route lain: kelola role, hak akses, log
        aktivitas, dst.
    });
```

Kode program untuk prefix Admin:

```
// ADMIN MODULE (akses diatur oleh permission, bukan role)
Route::prefix('admin')
    ->middleware(['auth'])
    ->name('admin.')
    ->group(function () {
```

```

        Route::get('/dashboard',
[AdminController::class, 'index'])
            ->name('dashboard');

        // Modul Obat (hanya yang punya permission "kelola
obat")
        Route::resource('obat', ObatController::class)
            ->middleware('permission:kelola obat');

        // route lain untuk distributor, transaksi masuk,
laporan, dst.
    });

```

Kode program untuk prefix Apoteker:

```

// APOTEKER MODULE (akses by permission)
Route::prefix('apoteker')
    ->middleware(['auth'])
    ->name('apoteker.')
    ->group(function () {

        Route::get('/dashboard',
[ApotekerController::class, 'index'])
            ->name('dashboard');

        // Obat: apoteker hanya lihat
        Route::get('/obat',      [ObatController::class,
'index'])
            ->name('obat.index')
            ->middleware('permission:lihat obat');
    });

```

```
        // route lain untuk transaksi obat keluar,
        laporan, dst.
    });
```

Kode program untuk prefix Owner:

```
// OWNER MODULE (akses by permission)
Route::prefix('owner')
    ->middleware(['auth'])
    ->name('owner.')
    ->group(function () {

        Route::get('/dashboard',
[DashboardController::class, 'index'])
            ->name('dashboard');

        // Obat: owner lihat
        Route::get('/obat',      [ObatController::class,
'index'])
            ->name('obat.index')
            ->middleware('permission:lihat obat');

        // route lain: distributor, laporan, dst., dengan
        permission "lihat laporan" / "cetak laporan"
    });
```

5. Implementasi Antarmuka Pengelolaan RBAC

Di sisi antarmuka, dibangun beberapa halaman:

- a. Halaman Kelola *User* untuk menambah, mengubah, dan menghapus pengguna serta mengatur *role*.
- b. Halaman Kelola *Role* untuk mengatur daftar peran dalam sistem.

- c. Halaman Hak Akses dan Atur Hak Akses untuk menghubungkan *role* dengan *permission* secara dinamis.
- d. Halaman *Log Aktivitas* untuk menampilkan catatan aktivitas pengguna dalam sistem sebagai bagian dari fungsi audit.

Selain halaman khusus tersebut, RBAC juga diterapkan di menu navigasi dan dashboard menggunakan *directive Blade @role*, *@can*, dan *@canany*. Contohnya, bagian ringkasan RBAC di dashboard hanya muncul jika *user* memiliki *permission* kelola pengguna:

```

{{-- resources/views/dashboard.blade.php --}}
{{-- RINGKASAN USER & HAK AKSES (KHUSUS SUPERADMIN) --}}
}}
@can('kelola pengguna')
    <div class="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-6">
        {{-- Total User --}}
        <a href="{{ route('superadmin.user.index') }}" class="...">
            <div class="flex items-center justify-between mb-3">
                <h3 class="text-lg font-bold break-words">Total User</h3>
                <i class="fa-solid fa-users text-3xl opacity-90"></i>
            </div>
            {{-- dst. --}}
        </a>
    </div>
@endcan

```

Pada layout utama, menu RBAC juga dibatasi menggunakan kombinasi *@role('superadmin')* dan *@can(...)*:

```

{{-- resources/views/components/app-layout.blade.php --}}
{{-- MENU SUPERADMIN (khusus pengelolaan RBAC) --}}
@role('superadmin')
    @can('kelola pengguna')

```

```

<a href="{{ route('superadmin.user.index') }}" class="...">
    <i class="fa-solid fa-users w-5 mr-3"></i>
    Kelola User
</a>

@endcan

@can('kelola role')
    <a href="{{ route('superadmin.role.index') }}" class="...">
        <i class="fa-solid fa-key w-5 mr-3"></i>
        Kelola Role
    </a>
@endcan

@can('kelola hak akses')
    <a href="{{ route('superadmin.hak-akses.index') }}" class="...">
        <i class="fa-solid fa-lock w-5 mr-3"></i>
        Hak Akses
    </a>
@endcan
@endrole

```

Dengan penerapan RBAC ini, sistem memastikan bahwa setiap pengguna hanya dapat mengakses modul dan fungsi yang sesuai dengan tugas dan tanggung jawabnya, sehingga memperkecil risiko akses tidak sah sekaligus menyederhanakan tampilan antarmuka..

4.1.4 Implementasi Modul Fungsional

Berdasarkan perancangan pada Bab III, modul-modul fungsional utama yang diimplementasikan meliputi:

1. Superadmin
 - a. Halaman *Login*

Halaman ini adalah halaman awal yang harus diakses oleh setiap pengguna

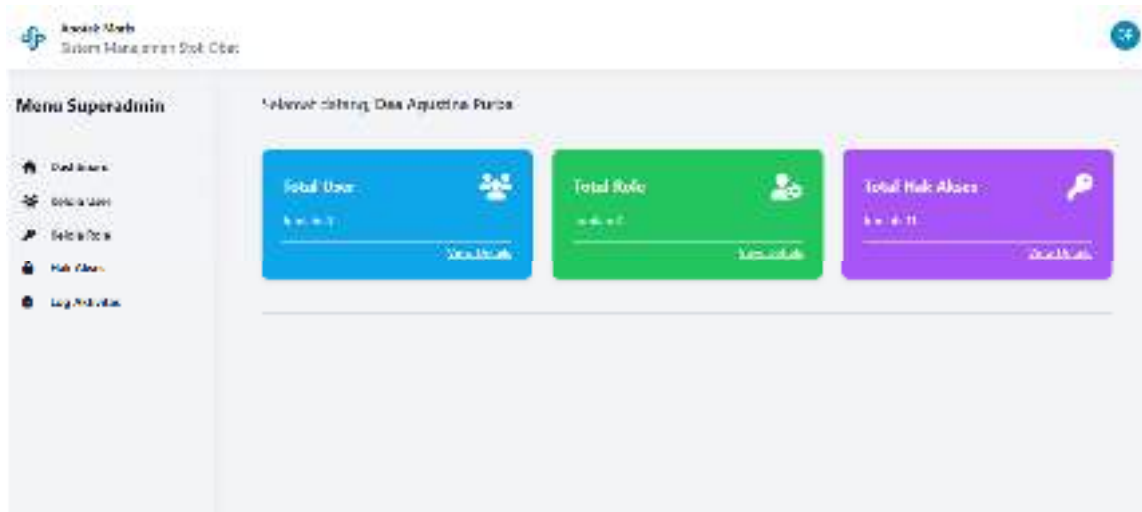
sebelum masuk ke dalam sistem. Pada halaman ini terdapat form *login* yang berisi inputan email dan *password* sebagai syarat autentikasi. Setelah pengguna memasukkan email dan *password* yang benar, sistem akan mengarahkan pengguna ke dashboard sesuai dengan peran yang dimilikinya. Jika *user* yang *login* tidak sesuai dengan *role* yang ada pada sistem maka akan muncul pesan “*These credentials do not match our records.*” Berikut tampilan halaman *login*:



Gambar 4. 1 Halaman *Login*

b. Halaman *Dashboard* Superadmin

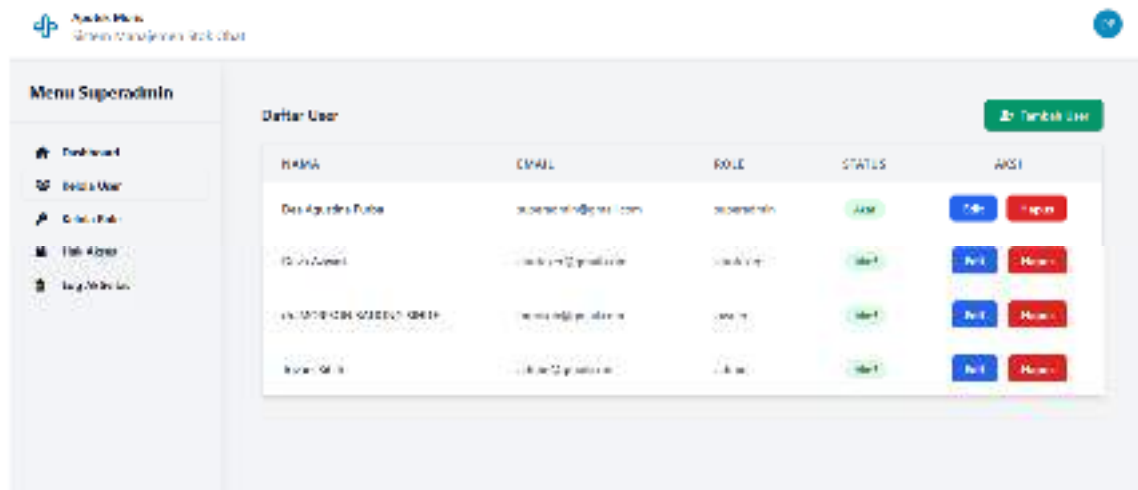
Halaman ini adalah halaman utama yang muncul ketika Superadmin berhasil *login* ke dalam sistem. Pada halaman *dashboard* ini ditampilkan informasi ringkas mengenai jumlah *user* yang terdaftar, jumlah *role* yang tersedia, serta jumlah hak akses yang ada dalam sistem. Halaman ini membantu Superadmin memantau keseluruhan konfigurasi akses sistem. Berikut tampilan *dashboard* Superadmin:



Gambar 4. 2 Halaman *Dashboard* Superadmin

c. Halaman *Kelola User*

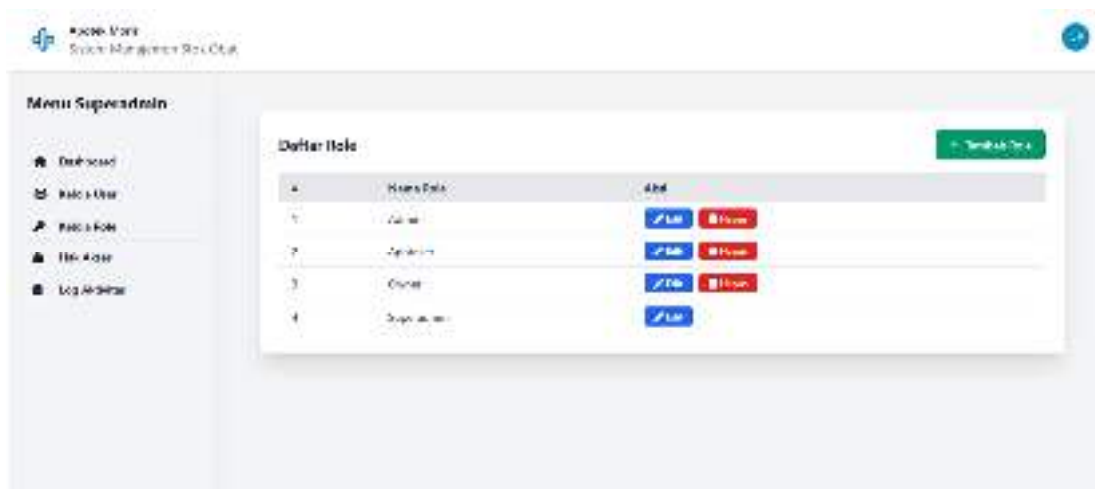
Halaman ini digunakan Superadmin untuk mengelola seluruh *user* yang menggunakan aplikasi. Halaman ini memungkinkan Superadmin mengatur siapa saja yang dapat menggunakan sistem. Berikut tampilan halaman *kelola user*:



Gambar 4. 3 Halaman *Kelola User*

d. Halaman *Kelola Role*

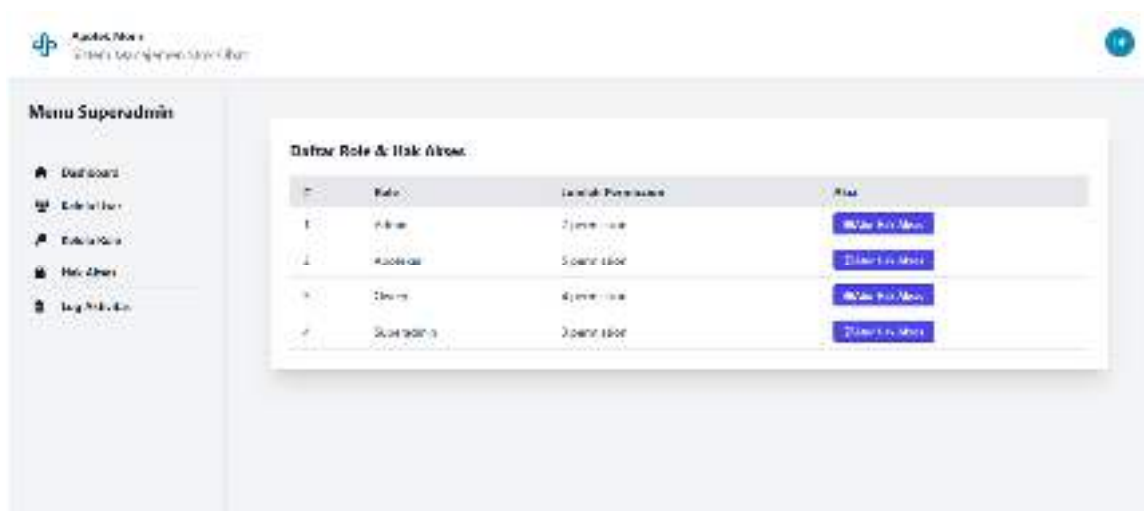
Halaman ini adalah halaman yang berfungsi untuk mengelola daftar *role* dalam sistem. Melalui halaman ini Superadmin dapat mengelola struktur peran yang digunakan dalam mekanisme RBAC. Berikut tampilan halaman *kelola role*:



Gambar 4. 4 Halaman Kelola *Role*

e. Halaman Hak Akses

Halaman ini menampilkan daftar *role* beserta jumlah *permission* yang dimilikinya. Superadmin dapat masuk ke halaman pengaturan hak akses untuk mengelola *permission* yang dimiliki oleh setiap *role*. Halaman ini memudahkan Superadmin dalam melihat gambaran umum pembagian hak akses dalam sistem. Berikut tampilan halaman daftar *role* dan hak akses:

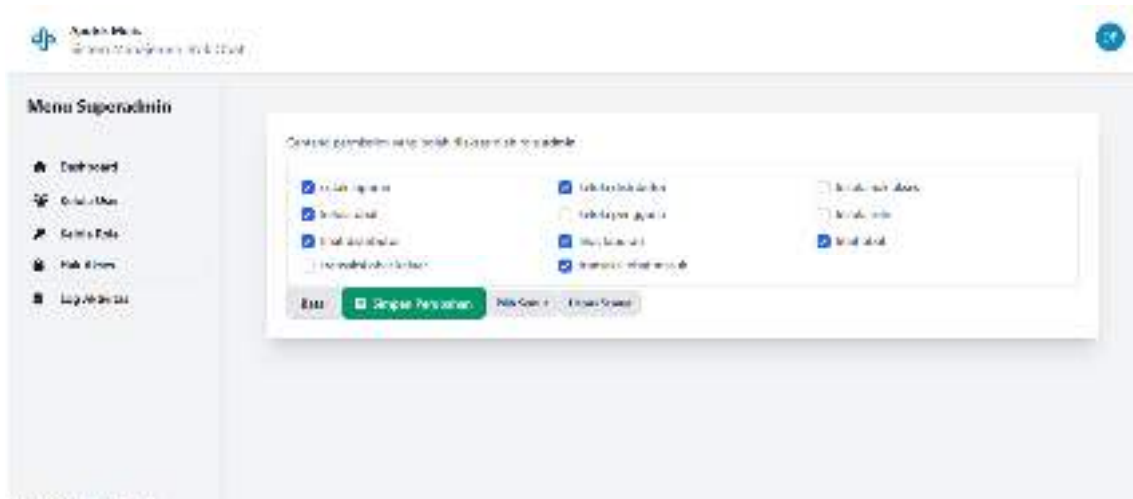


Gambar 4. 5 Halaman Hak Akses

f. Halaman Edit Hak Akses

Halaman ini digunakan Superadmin untuk menentukan hak akses yang dimiliki oleh suatu *role*. Pada halaman ini ditampilkan seluruh *permission* dalam bentuk *checkbox*. Superadmin dapat memberi atau menghapus izin dengan mencentang atau menghapus centang pada *permission* yang tersedia.

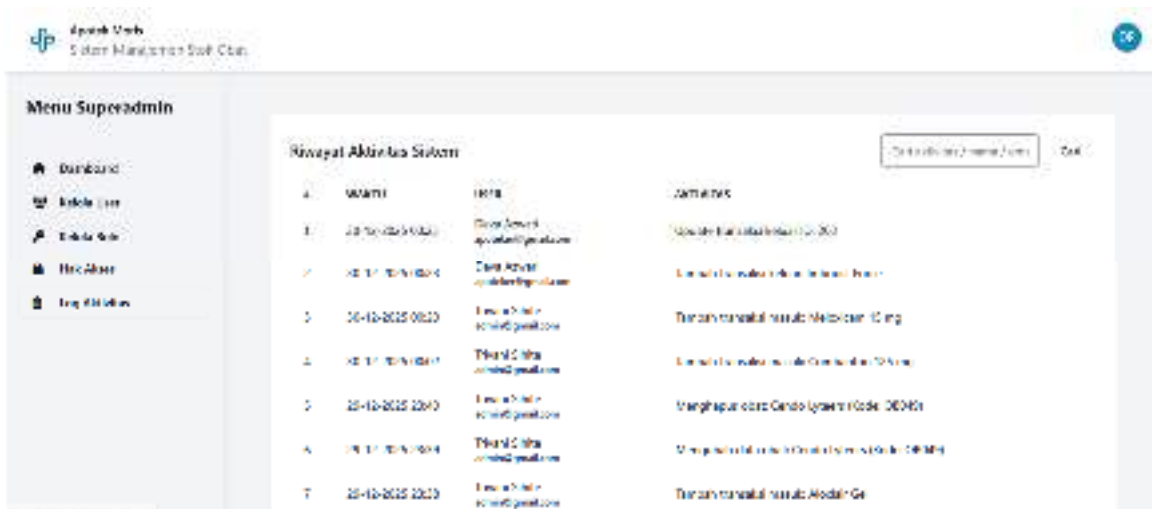
Halaman ini merupakan implementasi utama dari fitur RBAC dinamis yang memungkinkan pengaturan akses dilakukan dengan fleksibel. Berikut tampilan halaman pengaturan hak akses:



Gambar 4. 6 Halaman Edit Hak Akses

g. Halaman *Log* Aktivitas

Halaman ini menampilkan riwayat aktivitas sistem yang dilakukan oleh para pengguna. Informasi yang ditampilkan meliputi waktu aktivitas, nama pengguna, dan jenis aktivitas yang dilakukan seperti perubahan data, transaksi, atau pengaturan hak akses. Halaman ini dilengkapi kolom pencarian untuk mempermudah Superadmin menemukan aktivitas tertentu. Halaman ini sangat berguna untuk kebutuhan audit dan pelacakan aktivitas dalam sistem. Berikut tampilan halaman *log* aktivitas:

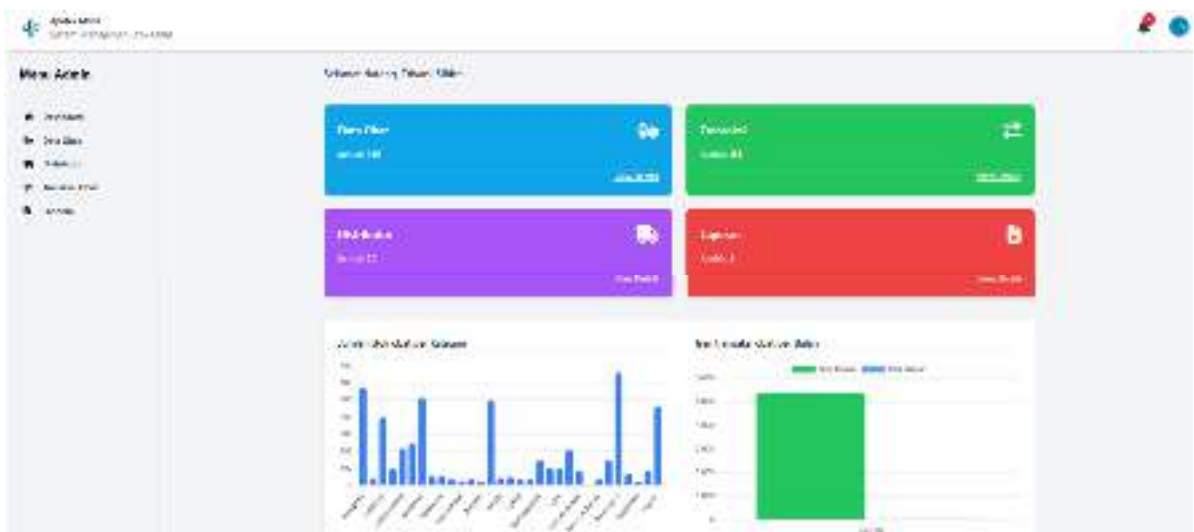


Gambar 4. 7 Halaman Log Aktivitas

2. Admin

a. Halaman *Dashboard* Admin

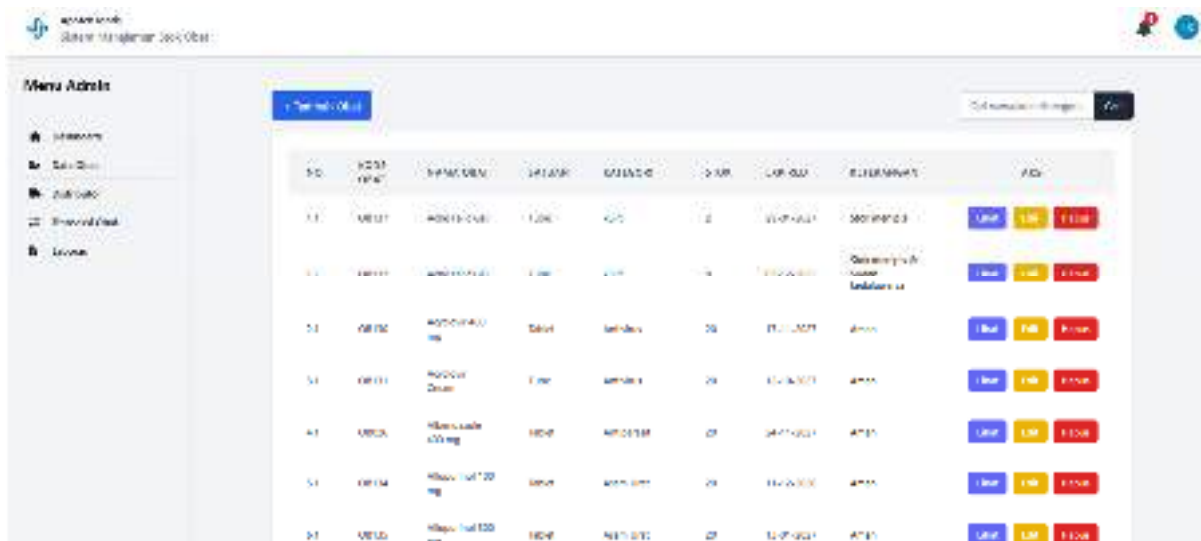
Halaman ini adalah halaman awal yang akan muncul ketika pengguna dengan *role* Admin berhasil *login* ke dalam sistem. Pada halaman *dashboard* ini ditampilkan empat kartu informasi utama, yaitu Data Obat, Transaksi, Distributor, dan Laporan. Di bagian bawah dashboard juga terdapat grafik jumlah stok obat per kategori dan grafik tren transaksi obat per bulan yang membantu Admin memantau kondisi stok dan pergerakan obat dari waktu ke waktu. Melalui halaman ini Admin dapat memperoleh ringkasan kondisi stok dan aktivitas transaksi secara cepat. Berikut tampilan *dashboard* Admin:



Gambar 4. 8 Halaman *Dashboard* Admin

b. Halaman Data Obat

Halaman ini digunakan Admin untuk mengelola seluruh data obat yang tersedia di apotek. Pada halaman ini hanya admin yang dapat mengelola data obat seperti melihat, mengedit, dan menghapus. Di bagian atas tabel terdapat tombol Tambah Obat untuk menambahkan data obat baru, serta kolom pencarian untuk mencari obat berdasarkan nama, kode, atau kategori. Berikut tampilan halaman data obat untuk Admin:



ID	KODE	NAMA OBAT	KATEGORI	STATUS	DOKUMEN	PILIHAN
1.1	00101	ASPIRIN 100mg	1.2.1	KURS	1.2.1.1	1.2.1.1.1
1.2	00102	ASPIRIN 200mg	1.2.2	KURS	1.2.2.1	1.2.2.1.1
1.3	00103	ASPIRIN 400mg	1.2.3	KURS	1.2.3.1	1.2.3.1.1
1.4	00104	ASPIRIN 600mg	1.2.4	KURS	1.2.4.1	1.2.4.1.1
1.5	00105	ASPIRIN 800mg	1.2.5	KURS	1.2.5.1	1.2.5.1.1
1.6	00106	ASPIRIN 1000mg	1.2.6	KURS	1.2.6.1	1.2.6.1.1
1.7	00107	ASPIRIN 1200mg	1.2.7	KURS	1.2.7.1	1.2.7.1.1

Gambar 4. 9 Halaman Data Obat

c. Halaman Distributor

Halaman ini digunakan Admin untuk mengelola data distributor yang memasok obat ke apotek. Pada halaman ini hanya admin yang dapat mengelola distributor seperti melihat, mengedit, dan menghapus. Di bagian atas tabel terdapat tombol Tambah distributor untuk menambahkan data distributor baru, serta kolom pencarian untuk mencari distributor berdasarkan nama, email, atau nomor telepon. Halaman ini membantu Admin menjaga data pemasok obat agar selalu terbaru. Berikut tampilan halaman distributor:

No	Nama Perusahaan	Alamat	Telepon	Email	Username	Password	Status	Aksi
1	PT. Sinar Mas	Jl. Sinar Mas No. 123	021-1234567	sinar.mas@pt.sinar-mas.co.id	admin	12345678	aktif	[Edit] [Hapus] [Detail]
2	PT. Sinar Mas	Jl. Sinar Mas No. 123	021-1234567	sinar.mas@pt.sinar-mas.co.id	admin	12345678	aktif	[Edit] [Hapus] [Detail]
3	PT. Sinar Mas	Jl. Sinar Mas No. 123	021-1234567	sinar.mas@pt.sinar-mas.co.id	admin	12345678	aktif	[Edit] [Hapus] [Detail]
4	PT. Sinar Mas	Jl. Sinar Mas No. 123	021-1234567	sinar.mas@pt.sinar-mas.co.id	admin	12345678	aktif	[Edit] [Hapus] [Detail]
5	PT. Sinar Mas	Jl. Sinar Mas No. 123	021-1234567	sinar.mas@pt.sinar-mas.co.id	admin	12345678	aktif	[Edit] [Hapus] [Detail]
6	PT. Sinar Mas	Jl. Sinar Mas No. 123	021-1234567	sinar.mas@pt.sinar-mas.co.id	admin	12345678	aktif	[Edit] [Hapus] [Detail]

Gambar 4. 10 Halaman Distributor

d. Halaman Transaksi Obat

Halaman ini menampilkan riwayat transaksi obat masuk dan obat keluar yang tercatat di sistem. Melalui halaman ini Admin hanya dapat melakukan transaksi obat masuk sesuai permission yang dia punya pada sistem. Berikut tampilan halaman transaksi obat untuk Admin:

No	Nama Obat	Distributor	Tanggal	Jumlah	Satuan	Harga	Aksi
1	Aspirin 100 mg	PT. Sinar Mas	01-12-2023	50	Tablet	1000	[Edit] [Hapus] [Detail]
2	Aspirin 100 mg	PT. Sinar Mas	02-12-2023	50	Tablet	1000	[Edit] [Hapus] [Detail]
3	Aspirin 100 mg	PT. Sinar Mas	03-12-2023	50	Tablet	1000	[Edit] [Hapus] [Detail]
4	Aspirin 100 mg	PT. Sinar Mas	04-12-2023	50	Tablet	1000	[Edit] [Hapus] [Detail]
5	Aspirin 100 mg	PT. Sinar Mas	05-12-2023	50	Tablet	1000	[Edit] [Hapus] [Detail]
6	Aspirin 100 mg	PT. Sinar Mas	06-12-2023	50	Tablet	1000	[Edit] [Hapus] [Detail]
7	Aspirin 100 mg	PT. Sinar Mas	07-12-2023	50	Tablet	1000	[Edit] [Hapus] [Detail]
8	Aspirin 100 mg	PT. Sinar Mas	08-12-2023	50	Tablet	1000	[Edit] [Hapus] [Detail]
9	Aspirin 100 mg	PT. Sinar Mas	09-12-2023	50	Tablet	1000	[Edit] [Hapus] [Detail]
10	Aspirin 100 mg	PT. Sinar Mas	10-12-2023	50	Tablet	1000	[Edit] [Hapus] [Detail]
11	Aspirin 100 mg	PT. Sinar Mas	11-12-2023	50	Tablet	1000	[Edit] [Hapus] [Detail]

10	Indikator 1	REKAPITULASI PENYALAMAN OBAT/STOK OBAT	02 12 2018	02	Tanggal	Revisi	01 12 2018	01	Cetak PDF	Cetak PDF
11	Indikator 2	REKAPITULASI PENYALAMAN OBAT/STOK OBAT	02 12 2018	02	Tanggal	Revisi	01 12 2018	01	Cetak PDF	Cetak PDF
12	Indikator 3	REKAPITULASI PENYALAMAN OBAT/STOK OBAT	02 12 2018	02	Tanggal	Revisi	01 12 2018	01	Cetak PDF	Cetak PDF

Klik pada Kolom

1 2 3 4 5 6 7 8 9 10 11 12

Daftar Transaksi Obat Masuk

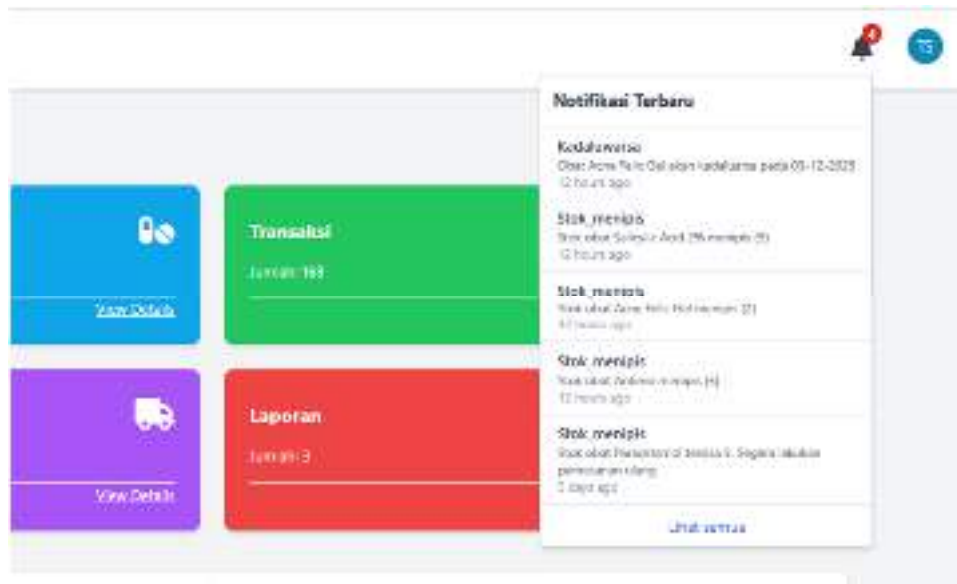
No	Nama Obat	Tempat	Jumlah	Saldo	Waktu	Uraian	Uraian	Aksi
1	Aspirin 100mg	02 12 2018	10	Tidak	100mg	02 12 2018	02 12 2018	Cetak PDF
2	Aspirin 100mg	02 12 2018	10	Tidak	100mg	02 12 2018	02 12 2018	Cetak PDF
3	Aspirin 100mg	02 12 2018	10	Tidak	100mg	02 12 2018	02 12 2018	Cetak PDF
4	Aspirin 100mg	02 12 2018	10	Tidak	100mg	02 12 2018	02 12 2018	Cetak PDF
5	Aspirin 100mg	02 12 2018	10	Tidak	100mg	02 12 2018	02 12 2018	Cetak PDF
6	Aspirin 100mg	02 12 2018	10	Tidak	100mg	02 12 2018	02 12 2018	Cetak PDF
7	Aspirin 100mg	02 12 2018	10	Tidak	100mg	02 12 2018	02 12 2018	Cetak PDF
8	Aspirin 100mg	02 12 2018	10	Tidak	100mg	02 12 2018	02 12 2018	Cetak PDF
9	Aspirin 100mg	02 12 2018	10	Tidak	100mg	02 12 2018	02 12 2018	Cetak PDF
10	Aspirin 100mg	02 12 2018	10	Tidak	100mg	02 12 2018	02 12 2018	Cetak PDF

Gambar 4. 11 Halaman Transaksi Obat

e. Halaman Laporan

Halaman ini digunakan Admin untuk melihat rekap laporan stok obat serta laporan transaksi obat masuk dan keluar. Pada sisi kanan atas tiap bagian laporan disediakan tombol Cetak PDF sehingga Admin dapat mencetak laporan dalam bentuk dokumen. Halaman ini membantu Admin menyusun laporan berkala mengenai persediaan obat di apotek. Berikut tampilan halaman laporan Admin:

kedaluwarsa obat yang dicek oleh sistem secara berkala. Melalui fitur ini Admin dapat segera mengetahui obat yang stoknya hampir habis atau mendekati kedaluwarsa tanpa harus membuka halaman data obat satu per satu. Berikut tampilan notifikasi *dashboard* Admin:

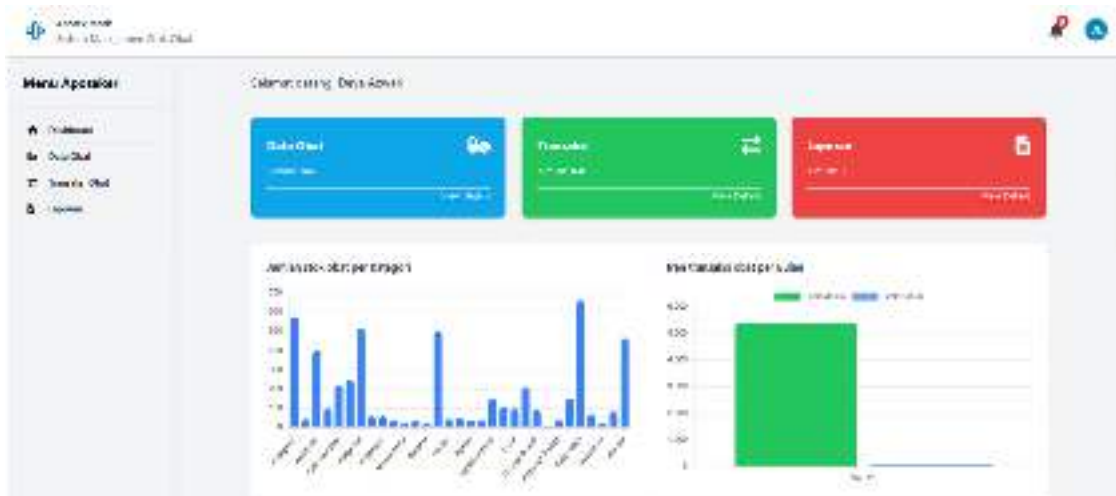


Gambar 4. 13 Halaman Notifikasi

3. Apoteker

a. Halaman *Dashboard* Apoteker

Halaman ini adalah halaman awal yang akan muncul ketika pengguna dengan *role* Apoteker berhasil *login* ke sistem. Pada halaman *dashboard* ini ditampilkan tiga kartu informasi utama, yaitu Data Obat, Transaksi, dan Laporan. Di bagian bawah terdapat grafik jumlah stok obat per kategori dan grafik tren transaksi obat per bulan. Meskipun apoteker hanya boleh melakukan transaksi obat keluar, ia tetap dapat melihat ringkasan seluruh transaksi dan stok sebagai bahan *monitoring* pelayanan. Berikut tampilan *dashboard* Apoteker:



Gambar 4. 14 Halaman *Dashboard* Apoteker

b. Halaman Data Obat

Halaman ini digunakan apoteker untuk melihat daftar obat yang tersedia di apotek. Berbeda dengan Admin, pada *role* Apoteker hanya tersedia tombol Lihat di kolom aksi tanpa tombol tambah, edit, dan hapus, sehingga apoteker hanya dapat membaca data obat tanpa mengubahnya. Hal ini sesuai dengan konsep RBAC bahwa pengelolaan data obat hanya dilakukan oleh Admin. Berikut tampilan halaman data obat untuk Apoteker:

The screenshot shows the 'Data Obat' page for an Apoteker. The sidebar menu is the same as in the dashboard. The main content area has a search bar with the text 'Cari obat berdasarkan nama obat'. Below the search bar is a table listing medicines. The table has columns: ID, KODE OBAT, NAMA OBAT, SATUAN, KATEGORI, STOK, POTENSI, SUPPLAER, and AKSI. The 'AKSI' column contains a 'Lihat' button for each row.

ID	KODE OBAT	NAMA OBAT	SATUAN	KATEGORI	STOK	POTENSI	SUPPLAER	AKSI
1	00102	Aspirin 400 mg	Tablet	Cardiovas	15	10-11-2017	Apotek	Lihat
2	00101	Aspirin 100 mg	Tablet	Cardiovas	20	12-10-2017	Apotek	Lihat
3	00101	Aspirin 100 mg	Tablet	Cardiovas	10	21-11-2016	Apotek	Lihat
4	00102	Aspirin 400 mg	Tablet	Cardiovas	20	24-11-2017	Apotek	Lihat
5	00104	Allopurinol 300 mg	Tablet	Cardiovas	20	10-11-2016	Apotek	Lihat

Gambar 4. 15 Halaman Data Obat

c. Halaman Transaksi Obat

Halaman ini digunakan apoteker untuk melihat riwayat transaksi obat serta melakukan transaksi obat keluar. Pada bagian atas halaman tersedia tombol

Transaksi Obat Keluar yang digunakan apoteker untuk mencatat pengeluaran obat saat pelayanan pasien. Tombol transaksi obat masuk tetap muncul namun ketika diklik muncul pesan “Hanya Admin yang dapat menambah transaksi masuk” karena hak akses tersebut hanya dimiliki Admin. Pengaturan ini sesuai dengan RBAC di mana apoteker hanya diberi *permission* transaksi obat keluar dan hak baca terhadap transaksi lainnya. Berikut tampilan halaman transaksi obat untuk Apoteker:

Menu Apoteker

Daftar Transaksi Obat Masuk

No	Nama Obat	Dosis/Kandungan	Tanggal	Jumlah	Satuan	Kategori	ID	Aksi
1	Parasetamol 500mg	500mg	01-11-2024	10	Tablet	Anda	01-11-2024	Edit
2	Amoxicillin 500mg	500mg	01-11-2024	10	Kapsul	Anda	01-11-2024	Edit
3	Zatam 100mg	100mg	01-11-2024	10	Kapsul	Anda	01-11-2024	Edit
4	Cefixime 500mg	500mg	01-11-2024	10	Kapsul	Anda	01-11-2024	Edit
5	Clarithromycin 500mg	500mg	01-11-2024	10	Tablet	Anda	01-11-2024	Edit
6	Clarithromycin 500mg	500mg	01-11-2024	10	Tablet	Anda	01-11-2024	Edit
7	Amoxicillin 500mg	500mg	01-11-2024	10	Tablet	Anda	01-11-2024	Edit
8	Amoxicillin 500mg	500mg	01-11-2024	10	Tablet	Anda	01-11-2024	Edit
9	Amoxicillin 500mg	500mg	01-11-2024	10	Tablet	Anda	01-11-2024	Edit

Daftar Transaksi Obat Keluar

No	Nama Obat	Tanggal	Jumlah	Satuan	Kategori	ID	Aksi
1	Parasetamol 500mg	01-11-2024	10	Tablet	Anda	01-11-2024	Edit Hapus
2	Amoxicillin 500mg	01-11-2024	10	Tablet	Anda	01-11-2024	Edit Hapus
3	Amoxicillin 500mg	01-11-2024	10	Tablet	Anda	01-11-2024	Edit Hapus
4	Amoxicillin 500mg	01-11-2024	10	Tablet	Anda	01-11-2024	Edit Hapus
5	Amoxicillin 500mg	01-11-2024	10	Tablet	Anda	01-11-2024	Edit Hapus
6	Amoxicillin 500mg	01-11-2024	10	Tablet	Anda	01-11-2024	Edit Hapus
7	Amoxicillin 500mg	01-11-2024	10	Tablet	Anda	01-11-2024	Edit Hapus
8	Amoxicillin 500mg	01-11-2024	10	Tablet	Anda	01-11-2024	Edit Hapus
9	Amoxicillin 500mg	01-11-2024	10	Tablet	Anda	01-11-2024	Edit Hapus
10	Amoxicillin 500mg	01-11-2024	10	Tablet	Anda	01-11-2024	Edit Hapus
11	Amoxicillin 500mg	01-11-2024	10	Tablet	Anda	01-11-2024	Edit Hapus
12	Amoxicillin 500mg	01-11-2024	10	Tablet	Anda	01-11-2024	Edit Hapus

Gambar 4. 16 Halaman Transaksi Obat

d. Halaman Laporan

Halaman ini menampilkan laporan stok obat dan riwayat transaksi obat yang dapat dibaca oleh apoteker. Berbeda dengan Admin, pada tampilan apoteker tidak disediakan tombol cetak laporan, karena *permission* cetak laporan hanya dimiliki Admin dan Owner. Dengan demikian apoteker dapat memantau kondisi stok dan transaksi sebagai bahan pelayanan, namun tidak dapat melakukan pencetakan laporan resmi. Berikut tampilan halaman laporan untuk Apoteker:

- ☐ Distributed
- ☐ Data Object
- ☐ Transacted Object
- ☐ Location

First Release of the paper for a second Read PDF with a watermark, please go to www.pdfbooks.org

No	Endo-Start	Monoclonal	Reagenz	Colony	Batch	Region	Feststellungen
1	0809	OncoScreen 124 mg	Titris	Genopurrol	18	11-12-2004	Schach, positiv, positiv
2	0809	OncoScreen 120 mg	Titris	Genopurrol	18	12-12-2005	Schach, positiv, positiv
3	0809	Reagenz	100 mg	Genopurrol	20	20-12-2005	Schach, positiv, positiv
4	0809	Reagenz	100 mg	Genopurrol	20	11-12-2005	Schach, positiv, positiv
5	0810	Reagenz, 100 mg	Reagenz	Genopurrol	5	20-12-2005	Reagenz positiv, positiv, positiv, positiv

[illegible]

2829 10 1 10 15 04 161 hours

C	1	1	2	4	2	6	7	5	9	18	11	2
---	---	---	---	---	---	---	---	---	---	----	----	---

No	Nome Obek	Tanggal	Jumlah	Kabupaten	Kategori	RS
1	Amfoud Tere	20-12-2025	30	Tuban	Alumina	20-12-2025
2	Amfoud Tere	09-12-2025	5	Tuban	Gardolordinal	27-12-2025
3	Amfoud Tere	01-12-2025	10	Tuban	Alumina	20-12-2025
4	Amfoud Tere	20-12-2025	15	Tuban	Alumina	20-12-2025
5	Amfoud Tere	20-12-2025	5	Tuban	Alumina	20-12-2025
6	Amfoud Tere	20-12-2025	2	Tuban	Alumina	20-12-2025
7	Amfoud Tere	20-12-2025	2	Tuban	Alumina	20-12-2025
8	Amfoud Tere	20-12-2025	2	Tuban	Alumina	20-12-2025
9	Amfoud Tere	20-12-2025	2	Tuban	Alumina	20-12-2025
10	Amfoud Tere	20-12-2025	1	Tuban	Alumina	20-12-2025
11	Amfoud Tere	20-12-2025	1	Tuban	Alumina	20-12-2025
12	Amfoud Tere	20-12-2025	15	Tuban	Alumina	20-12-2025

Gambar 4. 17 Halaman Laporan

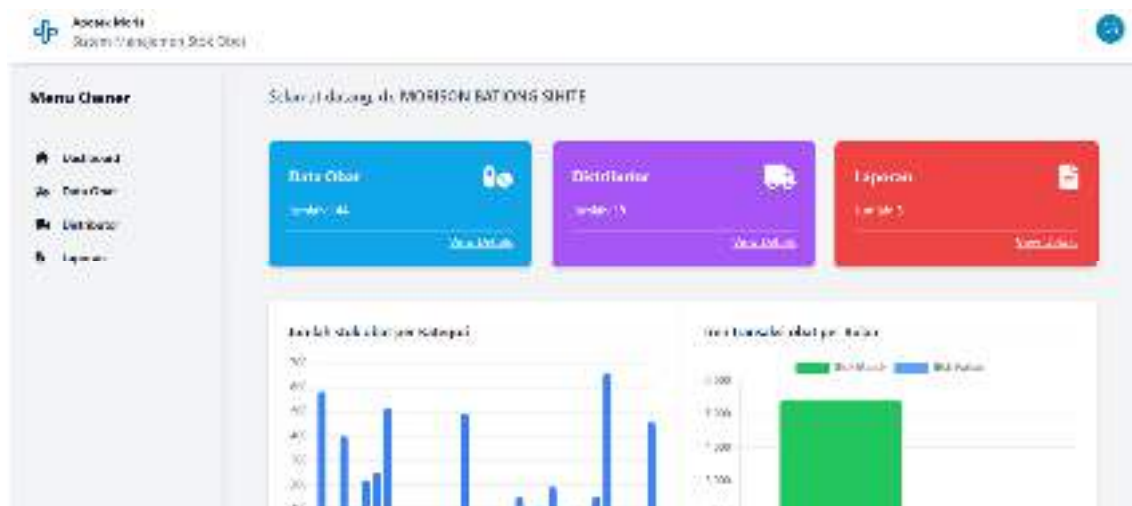
e. Halaman Notifikasi

Halaman ini menampilkan pesan notifikasi terbaru seperti “Stok Menipis” dan “Kedaluwarsa” lengkap dengan keterangan obat dan waktu terjadinya. Notifikasi ini dihasilkan secara otomatis berdasarkan data stok dan tanggal kedaluwarsa obat yang dicek oleh sistem secara berkala. Sama seperti admin, apoteker segera mengetahui obat yang stoknya hampir habis atau mendekati kedaluwarsa tanpa harus membuka halaman data obat satu per satu. Untuk tampilan halaman data obat bisa dilihat pada Gambar 4.13.

4. Owner

a. Halaman Dashboard Owner

Halaman ini adalah halaman awal yang akan muncul ketika pengguna dengan *role* Owner berhasil *login* ke sistem. Pada halaman dashboard ini ditampilkan tiga kartu informasi utama, yaitu Data Obat, Distributor, dan Laporan. Di bagian bawah terdapat grafik jumlah stok obat per kategori dan grafik tren transaksi obat per bulan yang membantu Owner memantau kondisi stok dan pergerakan obat secara menyeluruh. Owner tidak melakukan input data, tetapi hanya mengawasi data dan laporan yang dihasilkan oleh Admin dan Apoteker. Berikut tampilan *dashboard* Owner:



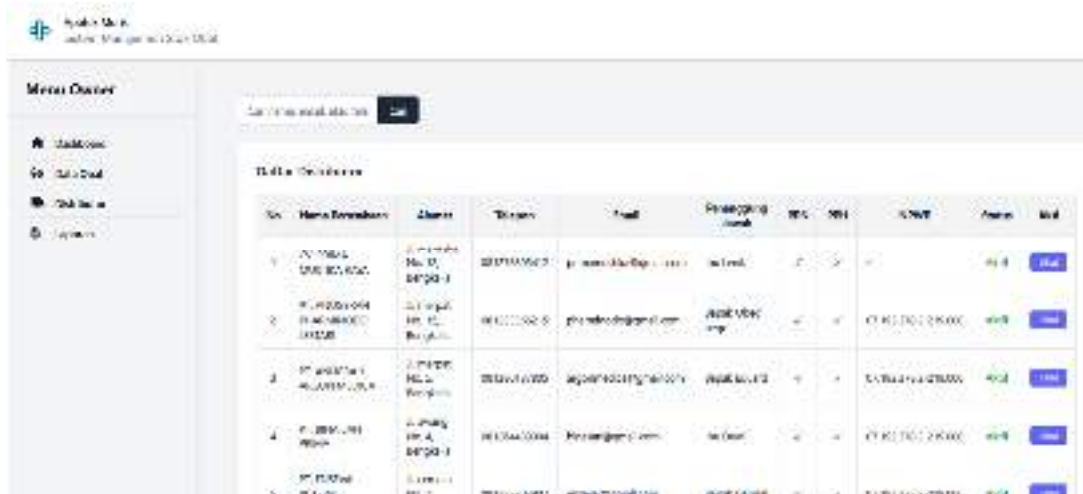
Gambar 4. 18 Halaman Dashboard Owner

b. Halaman Data Obat

Halaman ini digunakan owner untuk melihat daftar obat yang tersedia di apotek. Pada kolom aksi hanya tersedia tombol Lihat tanpa tombol tambah, edit, maupun hapus, sehingga owner hanya dapat memantau data obat tanpa melakukan perubahan. Pengaturan ini sesuai dengan konsep RBAC bahwa Owner berperan sebagai pengawas, bukan penginput data. Untuk tampilan halaman data obat bisa dilihat pada Gambar 4.15.

c. Halaman Distributor

Halaman ini adalah halaman yang digunakan Owner untuk melihat daftar distributor yang bekerja sama dengan apotek. Pada kolom aksi hanya tersedia tombol Lihat, tanpa fitur tambah, edit, ataupun hapus. Hal ini menggambarkan bahwa Owner hanya berperan memantau relasi distributor tanpa mengelola data secara langsung, sejalan dengan hak akses lihat distributor pada mekanisme RBAC. Berikut tampilan halaman distributor untuk Owner:



No	Nama Distributor	Alamat	Telepon	Email	Penanggung Jawab	NPK	NIB	NIBP	Alamat	Aksi
1	PT. KEMILAU Jl. Raya No. 100 Bogor	Jl. Raya No. 100 Bogor	021-1234567	pt.kemilau@gmail.com	Andi Setiawan	123456789	123456789	123456789	Jl. Raya No. 100 Bogor	Lihat
2	PT. KEMILAU Jl. Raya No. 100 Bogor	Jl. Raya No. 100 Bogor	021-1234567	pt.kemilau@gmail.com	Andi Setiawan	123456789	123456789	123456789	Jl. Raya No. 100 Bogor	Lihat
3	PT. KEMILAU Jl. Raya No. 100 Bogor	Jl. Raya No. 100 Bogor	021-1234567	pt.kemilau@gmail.com	Andi Setiawan	123456789	123456789	123456789	Jl. Raya No. 100 Bogor	Lihat
4	PT. KEMILAU Jl. Raya No. 100 Bogor	Jl. Raya No. 100 Bogor	021-1234567	pt.kemilau@gmail.com	Andi Setiawan	123456789	123456789	123456789	Jl. Raya No. 100 Bogor	Lihat
5	PT. KEMILAU Jl. Raya No. 100 Bogor	Jl. Raya No. 100 Bogor	021-1234567	pt.kemilau@gmail.com	Andi Setiawan	123456789	123456789	123456789	Jl. Raya No. 100 Bogor	Lihat

Gambar 4. 19 Halaman Distributor

d. Halaman Laporan

Halaman ini digunakan Owner untuk melihat dan mencetak laporan stok obat serta laporan transaksi obat di apotek. Hak akses ini sesuai dengan *permission* lihat laporan dan cetak laporan yang diberikan khusus pada *role* Owner. Untuk tampilan halaman laporan bisa dilihat pada Gambar 4.12.

4.2 Pengujian

4.2.1 Deskripsi Pengujian

Pengujian dilakukan setelah seluruh fitur sistem selesai dikembangkan dengan menggunakan metode Black Box Testing, yaitu pengujian yang berfokus pada input dan output tanpa melihat kode program. Skenario pengujian dilakukan dengan mencoba berbagai kombinasi input yang valid maupun tidak valid pada setiap fitur, seperti pengelolaan data obat, transaksi obat masuk dan keluar, pengelolaan distributor, laporan, notifikasi, serta hak akses pengguna. Pengujian ini bertujuan untuk memastikan bahwa setiap fungsi berjalan sesuai dengan kebutuhan sistem dan untuk mengetahui bagian mana yang masih perlu diperbaiki agar pengalaman pengguna menjadi lebih baik.

4.2.2 Prosedur Pengujian

Prosedur pengujian dilakukan untuk memastikan bahwa Sistem Informasi Manajemen Stok Obat berbasis website yang telah dikembangkan dapat berfungsi dengan baik serta memenuhi kebutuhan pengguna sesuai dengan perancangan sistem. Pengujian juga bertujuan untuk memverifikasi bahwa penerapan Role-Based Access Control (RBAC) mampu membatasi hak akses pengguna berdasarkan peran masing-masing.

Metode pengujian yang digunakan pada penelitian ini adalah *Black Box Testing*, yaitu metode pengujian yang berfokus pada pengujian fungsionalitas sistem dengan cara memberikan input dan mengamati output yang dihasilkan, tanpa memperhatikan struktur kode program secara internal. Adapun prosedur pengujian sistem dilakukan melalui tahapan sebagai berikut:

1. Persiapan Pengujian

Pada tahap ini dilakukan persiapan lingkungan pengujian, meliputi pengaturan server lokal menggunakan XAMPP, konfigurasi database MySQL, serta memastikan aplikasi sistem informasi manajemen stok obat dapat dijalankan dengan baik melalui browser.

2. Penentuan Skenario Pengujian

Skenario pengujian disusun berdasarkan kebutuhan fungsional sistem, meliputi:

- a. Pengujian autentikasi pengguna (*login* dan *logout*)
- b. Pengujian penerapan *Role-Based Access Control* (RBAC)
- c. Pengujian pengelolaan data obat dan distributor
- d. Pengujian transaksi obat masuk dan keluar
- e. Pengujian notifikasi stok dan kedaluwarsa
- f. Pengujian pembuatan dan tampilan laporan

3. Pelaksanaan Pengujian

Pengujian dilakukan dengan menjalankan setiap fitur sistem sesuai dengan skenario yang telah ditentukan. Setiap pengujian dilakukan menggunakan akun pengguna dengan peran yang berbeda, yaitu Superadmin, Admin, Apoteker, dan Owner, untuk memastikan pembatasan hak akses berjalan dengan benar.

4. Pencatatan Hasil Pengujian

Hasil dari setiap pengujian dicatat berdasarkan kesesuaian antara *output* yang dihasilkan sistem dengan hasil yang diharapkan. Jika fungsi berjalan sesuai dengan kebutuhan, maka pengujian dinyatakan berhasil.

5. Evaluasi Hasil Pengujian

Tahap evaluasi dilakukan untuk menilai apakah sistem telah memenuhi seluruh kebutuhan fungsional dan keamanan. Apabila ditemukan kesalahan atau ketidaksesuaian, maka dilakukan perbaikan sebelum sistem dinyatakan layak digunakan.

Berdasarkan prosedur pengujian yang telah dilakukan, sistem informasi manajemen stok obat berbasis website dengan penerapan *Role-Based Access Control* (RBAC) dapat berfungsi dengan baik dan sesuai dengan perancangan yang telah ditetapkan.

4.2.3 Data Hasil Pengujian

Pengujian ini melibatkan pengguna akhir sesuai role pada sistem, yaitu Admin, Apoteker, dan Owner, di mana masing-masing menjalankan uji berdasarkan tugas dan permission yang telah ditetapkan untuk memverifikasi bahwa RBAC berjalan benar. Admin menguji fitur yang menjadi tanggung jawabnya seperti kelola data obat dan distributor, pencatatan transaksi obat masuk, menerima notifikasi stok, serta melihat dan mencetak laporan. Apoteker menguji

fitur operasional pelayanan seperti transaksi obat keluar, melihat data obat/distributor dan laporan, serta menerima notifikasi, namun dipastikan tidak memiliki akses untuk mengubah data maupun mencetak laporan. Owner menguji fitur monitoring/pengawasan dengan akses baca terhadap data obat, distributor, laporan, dan statistik pada dashboard, termasuk memastikan Owner dapat mencetak laporan sesuai permission yang diberikan, tanpa melakukan input atau perubahan data. Dengan demikian, pengujian tidak hanya memastikan fungsi berjalan sesuai kebutuhan, tetapi juga mengecek pembatasan akses tiap role sudah sesuai prinsip least privilege dan alur kerja apotek.

Tabel 4. 1 Black Box Testing – Autentikasi & RBAC

Skenario Uji	Hasil yang Diharapkan	Hasil yang Didapatkan	Hasil
Login dengan email dan password benar	Berhasil login dan diarahkan ke <i>dashboard</i> sesuai <i>role</i> pengguna	Sistem berhasil memverifikasi data <i>login</i> dan mengarahkan pengguna ke <i>dashboard</i> sesuai <i>role</i>	Berhasil
Superadmin mengakses <i>dashboard</i>	<i>Dashboard</i> superadmin tampil dengan ringkasan <i>user</i> , <i>role</i> , dan <i>permission</i>	Sistem menampilkan <i>dashboard</i> superadmin sesuai rancangan	Berhasil
Superadmin mengakses kelola <i>user</i>	Halaman kelola <i>user</i> terbuka dan data <i>user</i> tampil	Halaman kelola <i>user</i> dapat diakses dan data <i>user</i> ditampilkan	Berhasil
Superadmin mengakses kelola <i>role</i>	Halaman kelola <i>role</i> tampil	Sistem menampilkan daftar <i>role</i>	Berhasil
Superadmin mengakses hak akses	Daftar <i>role</i> dan jumlah <i>permission</i> tampil	<i>Role</i> dan <i>permission</i> ditampilkan sesuai konfigurasi	Berhasil
Superadmin mengedit Hak Akses	<i>Checkbox permission</i> dapat diatur	Perubahan <i>permission</i> tersimpan dan diperbarui	Berhasil
Superadmin melihat log aktivitas	Riwayat aktivitas tampil	Data aktivitas pengguna berhasil ditampilkan	Berhasil

Tabel 4. 2 Black Box Testing – Modul Admin

Skenario Uji	Hasil yang Diharapkan	Hasil yang Didapatkan	Hasil
Login sebagai admin	Dashboard admin tampil	Sistem berhasil menampilkan <i>dashboard</i> admin dengan ringkasan data	Berhasil
Admin mengakses data obat	Halaman data obat tampil	Halaman data obat berhasil ditampilkan dan dapat dikelola	Berhasil
Admin mengakses distributor	Halaman distributor tampil	Halaman data distributor berhasil ditampilkan dan dapat dikelola	Berhasil
Admin mengakses transaksi obat	Halaman transaksi obat tampil	Halaman transaksi obat masuk berhasil ditampilkan dan aktif sedangkan transaksi obat keluar dibatasi	Berhasil
Admin mengakses laporan	Halaman laporan tampil	Laporan stok dan transaksi berhasil ditampilkan dan dapat dicetak PDF	Berhasil
Admin melihat notifikasi	Ikon lonceng menampilkan <i>badge</i> jumlah notifikasi belum dibaca, dan daftar notifikasi muncul di <i>dropdown</i>	Sistem menampilkan ikon lonceng notifikasi dengan <i>badge</i> jumlah notifikasi yang belum dibaca. Daftar notifikasi dapat dilihat melalui <i>dropdown</i> notifikasi pada <i>dashboard</i>	Berhasil

Tabel 4. 3 Black Box Testing – Modul Apoteker

Skenario Uji	Hasil yang Diharapkan	Hasil yang Didapatkan	Hasil
Login sebagai apoteker	Dashboard apoteker tampil	Dashboard apoteker berhasil ditampilkan	Berhasil
Apoteker mengakses data obat	Data obat tampil tanpa tombol tambah/edit/hapus	Data obat tampil dalam mode lihat saja	Berhasil
Apoteker mengakses Transaksi Obat	Halaman transaksi tampil dengan izin transaksi keluar saja	Tombol transaksi keluar aktif, transaksi masuk dibatasi	Berhasil

Apoteker mengakses Laporan	Halaman laporan tampil	Laporan berhasil ditampilkan sesuai hak akses	Berhasil
Apoteker melihat Notifikasi	Notifikasi tampil	Notifikasi stok dan kedaluwarsa berhasil muncul	Berhasil

Tabel 4. 4 Black Box Testing – Modul Owner

Skenario Uji	Hasil yang Diharapkan	Hasil yang Didapatkan	Hasil
Login sebagai owner	Dashboard owner tampil	Dashboard owner berhasil ditampilkan dengan grafik dan ringkasan	Berhasil
Owner mengakses data obat	Halaman data obat tampil dalam mode lihat saja	Halaman data obat berhasil ditampilkan tanpa tombol edit/hapus	Berhasil
Owner mengakses distributor	Halaman distributor tampil dalam mode lihat saja	Halaman data distributor berhasil ditampilkan tanpa fitur pengelolaan	Berhasil
Owner mengakses laporan	Halaman laporan terbuka dan data laporan dapat dilihat	Halaman laporan terbuka dan data laporan dapat dilihat	Berhasil

Tabel 4. 5 Black Box Testing – Fitur Fungsional Tambahan

Skenario Uji	Hasil yang Diharapkan	Hasil yang Didapatkan	Hasil
Sistem mendeteksi stok menipis	Notifikasi muncul di dashboard	Notifikasi stok menipis berhasil muncul sesuai batas minimum	Berhasil
Sistem mendeteksi obat mendekati kedaluwarsa	Notifikasi muncul	Notifikasi kedaluwarsa berhasil ditampilkan	Berhasil
Admin mencetak laporan	File laporan dapat dicetak	Laporan berhasil diunduh/dicetak dalam format PDF	Berhasil

Hasil pengujian menunjukkan bahwa seluruh skenario uji yang direncanakan telah berjalan dengan baik. Sistem mampu memberikan respon yang sesuai, baik

untuk input yang valid maupun tidak valid, serta berhasil membatasi hak akses berdasarkan *role* yang dimiliki pengguna.

4.3 Analisis Data

Berdasarkan hasil pengujian menggunakan metode *Black Box Testing*, sistem Manajemen Stok Obat menunjukkan bahwa seluruh fitur utama sudah berjalan sesuai dengan fungsionalitas yang diharapkan. Pengujian dilakukan pada beberapa skenario, mulai dari *login*, pengelolaan data obat, transaksi obat masuk dan keluar, pengelolaan distributor, laporan, notifikasi, hingga kontrol akses berbasis peran (RBAC). Sistem mampu memberikan *respons* yang tepat, baik ketika menerima input yang valid maupun menampilkan pesan peringatan pada input yang tidak sesuai.

Pada modul RBAC, setiap *role* hanya dapat mengakses fitur yang sesuai dengan *permission* yang diberikan. Akses yang tidak relevan berhasil dibatasi oleh sistem sehingga kontrol otorisasi berjalan dengan baik. Modul transaksi juga telah divalidasi dengan benar, di mana penambahan dan pengurangan stok berjalan sesuai logika, termasuk penolakan transaksi ketika jumlah yang dimasukkan melebihi stok tersedia. Notifikasi stok menipis dan obat mendekati kedaluwarsa juga berhasil muncul pada *dashboard* sesuai ketentuan *role*.

Secara keseluruhan, hasil pengujian menunjukkan bahwa sistem telah berfungsi dengan baik dan memenuhi kebutuhan operasional apotek, meskipun tetap memungkinkan adanya pengembangan lanjutan untuk meningkatkan kenyamanan penggunaan.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem yang telah dilakukan pada penelitian ini, maka terdapat beberapa kesimpulan yang disusun berdasarkan tujuan penelitian sebagai berikut:

1. Penelitian ini berhasil merancang dan membangun Sistem Informasi Manajemen Stok Obat berbasis website pada Apotek Pratama dr. Moris menggunakan *framework Laravel*. Sistem ini mampu mendukung proses pengelolaan obat secara terkomputerisasi, meliputi pencatatan data obat, transaksi obat masuk dan keluar, pengelolaan data distributor, penyediaan laporan stok dan transaksi, serta fitur notifikasi pada dashboard terkait stok obat menipis dan obat yang mendekati kedaluwarsa. Keberadaan sistem ini membantu meningkatkan efektivitas dan ketepatan dalam pengelolaan persediaan obat di apotek.
2. Sistem yang dikembangkan telah mengimplementasikan mekanisme keamanan *Role-Based Access Control* (RBAC) sesuai dengan perancangan. Setiap pengguna, yaitu Superadmin, Admin, Apoteker, dan Owner, dapat mengakses fitur yang sesuai dengan hak akses dan tanggung jawabnya masing-masing. Berdasarkan hasil pengujian menggunakan metode *Black Box Testing*, seluruh fungsi utama sistem, termasuk validasi input transaksi dan pembatasan akses berdasarkan role, telah berjalan dengan baik dan sesuai dengan kebutuhan pengguna, sehingga mampu meningkatkan keamanan serta keandalan sistem.

5.2 Saran

Meskipun sistem manajemen stok obat berbasis website ini telah berjalan dengan baik dan mampu mendukung proses operasional di Apotek Pratama dr. Moris, masih terdapat beberapa saran pengembangan agar sistem menjadi lebih lengkap dan optimal. Sistem ini dapat diperluas dengan menambahkan modul *return* obat dan pemusnahan obat kedaluwarsa, sehingga seluruh siklus pengelolaan obat dapat terdokumentasi secara menyeluruh mulai dari obat masuk, penggunaan, hingga proses pengembalian atau pemusnahan. Selain itu, diperlukan pula fitur pencadangan data otomatis untuk menjaga keamanan data apabila terjadi gangguan atau kerusakan pada perangkat. Dengan adanya pengembangan ini, sistem diharapkan mampu memberikan dukungan yang lebih baik dalam menjaga akurasi, keamanan, dan keberlanjutan pengelolaan stok obat di apotek.

DAFTAR PUSTAKA

- [1] N. Nursam, “*Manajemen Kinerja. Journal of Islamic Education Management*,” 2017
- [2] R. Sains Dan Teknologi Kelautan, M. Atika Puteri, and M. Putri Zabina dan Ezron Triputra, “Telaah Sistem Manajemen Pergudangan Dalam Berbagai Metode Inventory,” 2023.
- [3] L. H. S. dan Z. N. Suriyani, “Analisis Perencanaan Dan Pengawasan Persediaan Barang Dagang Pada CV. SUMBER INDORAYA,” *Jurnal Bisnis Net*, vol. III, no. 1, pp. 62–68, Jan.–Jun. 2020, Universitas Dharmawangsa. ISSN: 2621-3982 (Cetak); 2722-3574 (Online)., 2020.
- [4] K. Ruliyanto, S. Andryana, and A. Gunaryati, “Sistem Informasi Manajemen Persediaan Obat Berbasis Web Menggunakan Metode Prototype Pada Apotek.” *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, vol. 5, no. 3, pp. 284–290, Apr. 2021.
- [5] N. Novita, “Manajemen Proyek Sistem Informasi Pengelolaan Data Apotek Berbasis Database,” 2022. [Online]. Available: <http://ojs.fikom-methodist.net/index.php/methosisfo>
- [6] Y. A. Prasetia and D. Manongga, “Role-Based Access Control (RBAC) untuk Sistem Otorisasi Terpusat Berbasis Flask: Studi Kasus PT. XYZ,” vol. 9, no. 4, pp. 1768–1778, 2024, doi: 10.29100/jipi.v4i1.5403.
- [7] S. Y. Parameswari, I. Arwani, W. Hayuhardhika, and N. Putra, “Pengembangan Aplikasi Monitoring Stok Barang berbasis Website (Studi Kasus: PT. Kimia Farma Diagnostika Malang),” 2023. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [8] A. R. Ramadhan, S. Hidayat, A. Ikhsan, and B. Wulandari, “Design and Build A Website-Based Drug Supply Information System (Case Study at Apotek Salam),” *Journal of Universal Studies*, vol. 4, no. 12, pp. 12018–12039, 2024, [Online]. Available: <http://eduvest.greenvest.co.id>

- [9] I. M. S. Ramayu, “Rancangan Sistem Informasi Persediaan Stok Obat Dengan Metode System Development Life Cycle Di Apotek Nusa Farma Nusa Penida,” *Jurnal Indonesia : Manajemen Informatika dan Komunikasi*, vol. 4, no. 1, pp. 110–120, Jan. 2023, doi: 10.35870/jimik.v4i1.130.
- [10] N. Hasyim, N. A. Hidayah, and S. W. Latisuro, “Rancang Bangun Sistem Informasi Koperasi Berbasis Web Pada Koperasi Warga Baru MTS N 17 JAKARTA,” *Jurnal Sistem Informasi*, vol. 7, no. 2, pp. 1–10, 2014.
- [11] R. W. P. Pamungkas, A. N. Azizah, and B. S. Zebua, “Analisis penerapan metode scrum untuk meningkatkan efektivitas dalam pembuatan aplikasi melalui literature review,” *Jurnal Pendidikan Informatika dan Sains*, vol. 11, no. 2, pp. 156–164, Dec. 2022, doi: 10.31571/saintek.v11i2.4650.
- [12] S. I. Karepouwan, V. R. Palilingan, and O. Kembuan, “Perancangan Dan Implementasi Manajemen Stok Obat di Apotek RSUD Berbasis Web.” *JOINTER – Journal of Informatics Engineering*, vol. 2, no. 1, pp. 1–8, Jun. 2021.
- [13] R. H. dan E. V. J. P. Zaitun, “Perancangan Sistem Aplikasi Web Pengelolaan Stok Obat-Obatan Di Apotek Sumur Bor,” no. Bianglala Informatika: Jurnal Komputer dan Informatika, vol. 12, number 1, pp. 15–20, 2024, Universitas Bina Sarana Informatika Jakarta. ISSN: 2338-9761..
- [14] H. Althaf Haaerun and R. Mardhiyyah, “Perancangan Manajemen Aplikasi Inventory Obat Berbasis Android,” *Jurnal Informatika Teknologi dan Sains*, 2024.
- [15] M. R. M. dan D. H. G. Andira, “Sistem Informasi Desa Berbasis Web Untuk Meningkatkan Produk UMKM dan Layanan Masyarakat di Desa Bapangsari Kabupaten Purworejo,” *Jurnal Teknik Informatika*, vol. 15, no. 2, pp. 71–76, Apr. 2023, Universitas Logistik dan Bisnis Internasional..
- [16] M. Sahyudi and E. R. Susanto, “Analisis Implementasi Sistem Keamanan Basis Data Berbasis Role-Based Access Control (RBAC) pada Aplikasi Enterprise Resource Planning”, doi: 10.54259/satesi.v5i1.3997.

- [17] S. Osborn, R. Sandhu, Q. Munawer, ; R Sandhu, and Q. Munawer, "Configuring Role-Based Access Control to Enforce Mandatory and Discretionary Access Control Policies," *ACM Transactions on Information and System Security*, vol. 3, no. 2, pp. 85–106, May 2000.
- [18] Y. Yuricha and I. K. Phan, "Penerapan Role Based Access Control dalam Sistem Supply Chain Management Berbasis Cloud," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 3, no. 2, pp. 339–348, Nov. 2023, doi: 10.57152/malcom.v3i2.1259.
- [19] A. Priagus, S. Ashshidiq, S. A. Wicaksono, and D. Pramono, "Pengembangan Sistem Informasi Inventory Obat berbasis Web pada Puskesmas Klagenserut," 2022. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [20] N. Ilmitha and N. Novianti, "Sistem Pencatatan Stok Obat Menggunakan Aplikasi Berbasis Komputerisasi Pada Puskesmas Banjar Baru Lampung," *Universitas Bina Insani; Jl. Raya Siliwangi*, vol. 7, no. 1, p. 82400924, 2020.

LAMPIRAN

Lampiran 1. Pencatatan Stok Obat Manual

Samcofenac 50	100 = 2
Renadinac 50	100 = 1 + 1
Diiclophenac 50	100 = 1 + 1
Selesmol forte	100 = 1
Fasidol forte	100 = 1
Noragesic forte	100 = 2
Mirasic forte	100 = 1
Simvastatin 10	100 = 2
Plucadex	100 = 2 + 1 + 1
Alpara	150 = 1
Calortusin	100 = 1 + 1
Brochifar plus	100 = 1
Samcoanal forte	100 = 1
Pluanza	100 = 1
Etamoxul forte	100 = 1
Graxine	100 = 2 + 1
Anaron	100 = 1 + 1
Mirnatum forte	100 = 4
Berlimox	100 = 2
Alpara	150 = 1
Brodamox	100 = 1 + 1
Hufanoxil	100 = 2
Licogenta sip	7
Molexpla	150 = 1 + 1
Ambeven	100 = 2
Coric	100 = 2
Infabrim forte	100 = 1 + 1
Fasidol	100 = 1
SynalTen crm	10 + 10
Gentalex crm	12
Hrstrigo	50 = 2
Hufagesic	100 = 1 + 1
Ramagesic	100 = 1 + 1 + 1

No.	Nama Obat	Kemasan	Jumlah	Keterangan
1	Amalg Drops 10.5 ml	Box	2	
2	L-Rib	Box	1	
3	Pedematol Formula	Box	2	
4	Sargobiot vit K Tonik 150 ml	Botol	3	
5	Sargobiot vit K Tonik 40 ml	Botol	3	
6	Transpulmin K 10 Gr	Tube	5	
7	Transpulmin K 20 Gr	Tube	5	
8	Vicks VapoRub 10 Gr	POT	6	
9	Vicks VapoRub 25 Gr	POT	2	
10	Vicks VapoRub 60 Gr	POT	2	
11	Zambuk 25 Gr	Pes	6	
12	Zambuk 9 Gr	Pes	6	

PENEMIMA PESANAN
 Bengkulu, 17 Desember 2024
 APOTEKER PENANGGUNG JAWAB
 KLINIK
 (DEVA AZWARI, S.Farm, Apt)

NO	DISTRIBUTOR	PPN	PPH	NO. NPWP	KET
1	PT. MILLENNIUM PHARMACON INTERNATIONAL	✓	✓	07.193.378.2-219.000	✓
2	PT. GREAT DELI FARMA	-	-		*
3	PT. GLOBAL MITRA PEKANBARU	✓	✓	07.193.378.2-219.000	✓
4	PT. BINA SAN PRIMA	✓	✓	07.193.378.2-219.000	✓
5	PT. ANUGRAH ARGON MEDICA	✓	✓	07.193.378.2-219.000	✓
6	PT. SAPTA SARITAMA	✓	✓	07.193.378.2-219.000	✓
7	PT. ENSVAL PUTEWA MEGATRADING	✓	✓	07.193.378.2-219.000	✓
8	PT. INDOFARMA GLOBAL MEDICA	✓	✓	07.193.378.2-219.000	✓
9	PT. UNITED DICO OTAS	✓	✓	07.193.378.2-219.000	✓
10	CV. PATRA ANDAJAS MAKMUR	-	-		*
11	PT. MIFSA SINASURSES	✓	✓	07.193.378.2-219.000	✓
12	CV. PRIMA MUSTIKA RAYA	-	-		*
13	PT. MURNI UTAMA PHARMA	✓	✓	07.193.378.2-219.000	✓
14	PT. KUNALA MELUR	✓	✓	07.193.378.2-219.000	✓
15	PT. ANUGRAH PHARMINDO LESTARI	✓	✓	07.193.378.2-219.000	✓
16	PT. PUTRA SARI MAKMUR	-	-		*
17	PT. TEMPO	-	-		*
18	PT. UTAMA	-	-		*
19	PT. SAWAH BESAT FARMA	✓	✓	07.193.378.2-219.000	✓
20	PT. KESAYORAN PHARMA	✓	✓	07.193.378.2-219.000	✓
21	PT. NAREXI LESTARI	✓	✓	07.193.378.2-219.000	✓
22	PT. PIGU ANDAJAN FARMA	✓	✓	07.193.378.2-219.000	✓
23	PT. PAKIT PADANG GLOBAL	✓	✓	07.193.378.2-219.000	✓
24	PT. MANULAKA PUTRA MANDIRI	-	-		*
25	PT. TRIAPTA JAYA	✓	✓	07.193.378.2-219.000	✓
26	PT. MITRO DRUG INDONESIA	✓	✓	07.193.378.2-219.000	✓
27	PT. KIMA FARMA	✓	✓	07.193.378.2-219.000	✓
28	PT. BUKIT BARISAN	-	-		*
29	PT. MUTHA JAYA MEDICA	-	-		*
30	PT. INTI AGUNG	-	-		*
31	PT. MITRA MEDICA RAYA	-	-		*
32	PT. INTEGRASTA	-	-		*
33	PT. TALANG SUGUK	-	-		*
34	PT. OSMERONA	✓	✓	07.193.378.2-219.000	✓
35	PT. POKTA VALENT	✓	✓	07.193.378.2-219.000	✓
36	PT. ANTA MITRA SIMPANG	✓	✓	07.193.378.2-219.000	✓
37	SWARNA	✓	✓	07.193.378.2-219.000	✓
38	NEM KEM	-	-		*
39	PT. SAUKIA PRIMA	-	-		*
40	PT. SIKHA IMPRESIUS PERKASA	✓	✓	07.193.378.2-219.000	✓
41	PT. HOJAPAN RAYA MANEKO	✓	✓	07.193.378.2-219.000	✓
42	PT. SARILAS HASTAMA JAYA	✓	✓	07.193.378.2-219.000	✓
43	PT. Sinar Medica	-	-		*

Lampiran 4. Dokumentasi Wawancara



Lampiran 5. Dokumentasi Obat-obatan di Apotek

