**Jurnal Qua Teknika, Vol. 16 No. 2 September 2026**
**ISSN 2088-2424 (cetak); ISSN 2527-3892 (elektronik)**
**Fakultas Teknik Universitas Islam Balitar, Blitar**
**Https://ejournal.unisbablitar.ac.id/index.php/qua;**
**Email: quateknika@gmail.com**

**Zunita Wulansari[1)], Mukh. Taofik Chulkamdi[2)], Candra Wibowo[3)]**
**PERANCANGAN SISTEM MANAJEMEN STOK PERSEDIAAN RUMAH SAKIT DENGAN NOTIFIKASI REAL-TIME BERBASIS TELEGRAM BOT API**
**Jurnal *Qua Teknika*, (2026), 16 (2): 91-105**

# PERANCANGAN SISTEM MANAJEMEN STOK PERSEDIAAN RUMAH SAKIT DENGAN NOTIFIKASI REAL-TIME BERBASIS TELEGRAM BOT API

**Zunita Wulansari[1)], Mukh Taofik Chulkamdi[2)], Candra Wibowo[3)]**
**[1,2,3]Fakultas Teknik Dan Informatika, Universitas Islam Balitar**
**Jl. Imam Bonjol No. 16, Kel. Sananwetan, Kec. Sananwetan, Kota Blitar**
**email: zunitawulansari@gmail.com[1], chulkamdi@gmail.com[2] candra294@gmail.com[3]**

## Abstrak

Pengelolaan persediaan logistik non-medis rumah sakit memerlukan pemantauan stok yang akurat dan tepat waktu untuk mencegah kekurangan persediaan yang dapat mengganggu pelayanan administratif. Penelitian ini bertujuan mengembangkan sistem manajemen stok persediaan rumah sakit dengan notifikasi secara real-time menggunakan Telegram Bot API di RSUD Mardi Waluyo Blitar. Sistem dikembangkan menggunakan metode Waterfall dengan teknologi React.js, Node.js, dan MySQL, sedangkan UML digunakan untuk perancangan sistem dan Black Box Testing untuk pengujian fungsional. Hasil penelitian menunjukkan bahwa sistem mampu mengelola data persediaan secara real-time dan mengirimkan peringatan dini secara otomatis ketika stok mencapai batas minimum. Pengujian terhadap 50 skenario menghasilkan tingkat keberhasilan keseluruhan sebesar 96%. Sistem yang dikembangkan mendukung pengelolaan persediaan rumah sakit yang lebih akurat, transparan, dan proaktif.

**Kata kunci**: Persediaan Rumah Sakit, Internet of Things, Telegram Bot, Sistem Peringatan Dini, Black Box Testing.

## PENDAHULUAN

Era digitalisasi telah membawa perubahan fundamental dalam tata kelola sektor publik, khususnya pada aspek pelayanan dan manajemen operasional. Transformasi ini mendorong instansi pemerintah untuk mengadopsi sistem berbasis teknologi guna meningkatkan kualitas layanan (Prasetyo, 2022). Dalam konteks ini, aspek manajemen logistik memegang peranan krusial dalam menjaga keberlangsungan operasional instansi pelayanan kesehatan. Implementasi sistem informasi berbasis digital terbukti mampu menekan tingkat kesalahan pencatatan manual serta mempercepat proses pelaporan data transaksi secara signifikan dibandingkan metode konvensional. Penelitian yang dilakukan oleh (Anam & Nurgiyatna, 2021) tentang sistem informasi inventaris di Klinik Pratama Permata Hati membuktikan bahwa penerapan sistem berbasis web mampu meningkatkan efisiensi pengelolaan barang dan menghasilkan tingkat kepuasan pengguna hingga 85% melalui pengujian black box.

Pada instansi rumah sakit, ruang lingkup logistik tidak terbatas pada obat-obatan, melainkan juga mencakup logistik non-medis seperti alat tulis kantor (ATK), bahan cetak, dan perlengkapan listrik. Berdasarkan data inventaris RSUD Mardi Waluyo, terdapat lebih dari 200 jenis barang non-medis yang harus dikelola, dengan tingkat perputaran (turnover) mencapai 30 – 40 transaksi perhari. Kompleksitas ini semakin memperkuat kebutuhan akan system terdigitalisasi. Seluruh komponen tersebut menjadi penunjang utama kelancaran aktivitas administratif. Namun, hasil observasi selama pelaksanaan penelitian menunjukkan bahwa proses pencatatan stok logistik umum masih dilakukan secara manual. Praktik semacam ini rentan menimbulkan ketidaksinkronan data administratif dengan kondisi fisik barang serta ketiadaan mekanisme peringatan dini (early warning system) saat stok menipis. Hal ini sejalan dengan penelitian (Hidayat & Setyanto, 2022) yang menyatakan bahwa pemantauan stok konvensional seringkali mengakibatkan keterlambatan pengadaan barang akibat lambatnya arus informasi.

Berdasarkan kebutuhan pengembangan sistem manajemen stok persediaan yang teritegrasi dengan notifikasi Telegram Bot API, penelitian ini difokuskan pada penyediaan data presisi antara kondisi gudang fisik dengan dashboard manajerial untuk menjamin efisiensi operasional. Di RSUD Mardi Waluyo, ketersediaan logistik non-medis memiliki signifikansi vital bagi kontinuitas operasional; kegagalan deteksi dini terhadap stok kritis berpotensi memicu diskontinuitas pelayanan yang berdampak langsung pada penurunan kecepatan layanan pasien. Secara spesifik, ketiadaan formulir fisik atau bahan cetak akibat keterlambatan pengadaan akan menyebabkan stagnasi di loket pendaftaran, sehingga pasien tidak dapat terlayani dengan cepat dan mengakibatkan penumpukan antrean yang tidak seharusnya terjadi. Selain itu, kekosongan stok logistik juga menghambat penyusunan pelaporan medis tepat waktu, yang pada akhirnya memperpanjang durasi birokrasi

administratif dan mengurangi efektivitas waktu penanganan pasien secara keseluruhan. Oleh karena itu, diperlukan sebuah instrumen yang tidak hanya berfungsi sebagai basis data statis, tetapi juga sebagai pemantau aktif melalui penerapan mekanisme notifikasi otomatis. Implementasi IoT memungkinkan sistem untuk melakukan pemantauan secara mandiri. Dalam konteks ini, otomatisasi informasi diimplementasikan pada layer software (virtual IoT) di mana data stok bertindak sebagai objek virtual yang terhubung ke internet dan mampu memicu aksi notifikasi secara otonom, sesuai dengan prinsip dasar konektivitas dan automasi dalam paradigma IoT.

Pengembangan sistem ini dilakukan menggunakan metode Waterfall, yang mencakup tahapan analisis kebutuhan, desain sistem menggunakan UML (Unified Modeling Language), pengkodean dengan stack teknologi React.js, Node.js, dan MySQL, hingga pengujian menggunakan Black Box Testing. Pemanfaatan Telegram Bot API dipilih sebagai media notifikasi karena keunggulannya dalam memberikan push notification yang instan dan memiliki latensi rendah dibandingkan media komunikasi lainnya (Pratama, 2022).

Meskipun beberapa penelitian terdahulu telah membahas mengenai sistem inventarisasi, namun sebagian besar masih terfokus pada pencatatan statis yang memerlukan interaksi aktif pengguna untuk memeriksa ketersediaan barang. Terdapat celah penelitian (research gap) dalam hal efektivitas penyampaian informasi stok kritis secara proaktif di lingkungan rumah sakit. Kebaruan (novelty) dalam perancangan ini terletak pada integrasi IoT yang mentransformasi sistem informasi pasif menjadi instrumen monitoring aktif. Dengan memanfaatkan Telegram Bot sebagai jembatan komunikasi antara basis data dan pengguna, sistem ini mampu memangkas latensi informasi yang menjadi kelemahan utama pada sistem konvensional, sehingga menjamin kesinambungan penyediaan logistik secara otomatis dan tepat waktu.

Bertolak dari permasalahan dan landasan teori tersebut, penelitian ini bertujuan mengembangkan "Perancangan Sistem Manajemen Stok Persediaan Rumah Sakit dengan Notifikasi Real-Time Berbasis Telegram Bot API". Implementasi sistem ini diharapkan mampu memperkuat akurasi inventaris, mempercepat pengambilan keputusan manajerial, dan menjamin keberlanjutan operasional instansi secara terintegrasi, transparan, dan akuntabel.

## METODE PENELITIAN

### 1. Sistem Manajemen Inventaris Rumah Sakit

Manajemen inventaris merupakan serangkaian kebijakan pengawasan untuk menentukan tingkat stok yang harus dijaga. Dalam konteks rumah sakit, pengelolaan barang non-medis menuntut akurasi tinggi untuk menghindari defisit logistik. Menurut (Savitri, Hanafi, & Setyo, 2021), digitalisasi manajemen stok menggunakan aplikasi berbasis web terbukti mampu meminimalisir kesalahan pencatatan manual dan mempercepat proses pelaporan data transaksi secara signifikan dibandingkan metode konvensional.

### 2. Internet of Things (IoT) dalam Pemantauan Stok

*Internet of Things* (IoT) adalah paradigma teknologi yang memungkinkan perangkat fisik saling berkomunikasi melalui jaringan internet. Pada sistem manajemen gudang, IoT berperan dalam otomasi pengumpulan data. (Hidayat & Setyanto, 2022) menjelaskan bahwa integrasi perangkat keras berbasis IoT dengan sistem basis data memungkinkan pemantauan stok secara *real-time*, di mana data mutasi barang dapat langsung terakumulasi ke dalam server tanpa intervensi manusia yang berlebihan.

### 3. Framework Node.js dan Express.js

Node.js menyediakan lingkungan *runtime* JavaScript yang bersifat *asynchronous* dan *event-driven*, yang sangat efisien untuk aplikasi intensif data. Express.js digunakan sebagai kerangka kerja untuk membangun API. Berdasarkan penelitian (Nugroho, Sutrisno, & Saputra, 2023), penggunaan Node.js dan Express.js dalam pengembangan sistem informasi memberikan performa *throughput* yang lebih baik dalam menangani permintaan data yang besar secara simultan dibandingkan dengan PHP tradisional.

### 4. JSON Web Token (JWT) sebagai Standar Keamanan

Keamanan data pada sistem informasi rumah sakit bersifat krusial. JWT adalah standar untuk mendefinisikan cara pengiriman informasi secara aman antara klien dan server. Menurut (Kurniawan & Rahardjo, 2021), JWT memungkinkan mekanisme autentikasi yang *stateless*, sehingga server tidak perlu menyimpan sesi *login*, yang pada akhirnya meningkatkan skalabilitas sistem saat diakses oleh banyak admin secara bersamaan.

**5. Layanan Notifikasi Telegram Bot API**

*Telegram Bot API* merupakan antarmuka pemrograman aplikasi berbasis HTTP yang memungkinkan pengembang untuk mengintegrasikan program eksternal dengan platform *Telegram* secara otomatis. Dalam sistem peringatan dini (*Early Warning System*), Telegram sering digunakan karena keandalan *push notification*-nya. (Pratama, 2022) dalam studinya menyebutkan bahwa penggunaan *Telegram Bot* sebagai media notifikasi stok rendah lebih efektif karena memiliki latensi yang rendah dan aksesibilitas yang mudah di berbagai perangkat seluler dibandingkan pengiriman melalui email.

Dalam arsitektur sistem manajemen stok ini, Telegram Bot API berfungsi sebagai *gateway* komunikasi antara *server* basis data dengan pengguna akhir (*end-user*).

Secara teknis, fungsi-fungsi spesifik yang diimplementasikan dalam sistem ini meliputi:

1. Metode *SendMessage* : Fungsi utama yang digunakan untuk mengirimkan pesan teks secara otomatis dari *server* ke *chat ID* admin ketika sistem mendeteksi bahwa kuantitas stok barang telah mencapai atau berada di bawah ambang batas minimum (*minimum stock level*) yang telah ditentukan.
2. Mekanisme *Push Notification* : Berbeda dengan metode penarikan data manual (*pulling*), fungsi ini memungkinkan sistem untuk memberikan peringatan secara proaktif dengan latensi rendah, sehingga menjamin responsivitas dalam pengambilan keputusan manajerial terkait pengadaan barang.
3. Integrasi *Token-Based Authentication* : Penggunaan token unik yang diberikan oleh *BotFather* berfungsi sebagai instrumen keamanan untuk memastikan bahwa instruksi pengiriman notifikasi hanya dapat dilakukan oleh *server* resmi yang terdaftar, menjaga integritas alur informasi di lingkungan RSUD Mardi Waluyo.

Implementasi fungsi-fungsi tersebut mentransformasi sistem informasi inventaris konvensional menjadi sebuah instrumen monitoring aktif yang bekerja secara otonom, sesuai dengan prinsip automasi pada paradigma *Internet of Things* di lapis perangkat lunak (*software layer*).

**6. Basis Data Relasional MySQL**

*MySQL* adalah sistem manajemen basis data relasional (RDBMS) yang menggunakan *SQL* sebagai bahasa komunikasi. (Munawar, 2021) menyatakan bahwa *MySQL* tetap menjadi pilihan utama dalam sistem manajemen stok karena kemampuannya dalam menjaga integritas data melalui skema relasi antar tabel (ERD), yang mencegah terjadinya redundansi data transaksi barang.

**7. Framework React.js**

*React.js* merupakan pustaka *JavaScript* yang dikembangkan oleh Facebook untuk membangun antarmuka pengguna (*user interface*) yang dinamis dan interaktif. Menurut (Fadli & Ramdani, 2022), keunggulan utama React.js terletak pada konsep *Virtual DOM* dan *Component-Based Architecture*, yang memungkinkan aplikasi memperbarui bagian tampilan tertentu tanpa harus memuat ulang keseluruhan halaman. Hal ini sangat krusial pada sistem manajemen stok yang memerlukan pembaruan data secara cepat saat terjadi mutasi barang dari sensor IoT.

**8. Tailwind CSS untuk Antarmuka Responsif**

*Tailwind CSS* adalah kerangka kerja CSS berbasis *utility-first* yang memungkinkan pengembang membangun desain antarmuka secara cepat langsung di dalam *file* HTML/JSX. Berdasarkan penelitian (Sanjaya, 2023), penggunaan *Tailwind CSS* pada aplikasi berbasis web meningkatkan efisiensi waktu pengembangan dan menghasilkan beban *file* CSS yang lebih ringan dibandingkan framework konvensional seperti *Bootstrap*. Hal ini mendukung performa *dashboard* inventaris rumah sakit agar tetap responsif saat diakses melalui perangkat seluler oleh petugas gudang.

**9. Unified Modeling Language (UML)**

*Unified Modeling Language* (UML) merupakan bahasa standar yang digunakan untuk memvisualisasikan, menspesifikasikan, dan mendokumentasikan artefak dari sistem perangkat lunak. Dalam perancangan sistem ini, *diagram* UML yang digunakan meliputi *Use Case Diagram* untuk memodelkan interaksi aktor, serta *Class Diagram* untuk menggambarkan struktur data dan hubungan antar entitas dalam sistem manajemen stok. Menurut (Sukamto & Shalahuddin, 2022), pemodelan menggunakan UML sangat krusial untuk memastikan alur kerja sistem terdefinisi dengan baik sebelum tahap pengkodean dimulai.

### 10. Black Box Testing

*Black Box Testing* adalah metode pengujian perangkat lunak yang berfokus pada sisi fungsionalitas aplikasi tanpa harus mengetahui struktur kode internal. Pengujian ini bertujuan untuk memastikan bahwa setiap fitur, seperti *input* data stok, pembaruan status, dan pengiriman notifikasi Telegram, telah menghasilkan *output* yang sesuai dengan kebutuhan operasional yang telah ditetapkan. Berdasarkan penelitian (Cholifah, Yulianingsih, & Sagita, 2022), metode *Black Box Testing* dengan teknik *Equivalence Partitioning* efektif dalam mendeteksi kesalahan fungsionalitas antarmuka tanpa harus menguji alur logika program secara internal.

## HASIL DAN PEMBAHASAN

Pelaksanaan penelitian di RSUD Mardi Waluyo Kota Blitar telah menghasilkan sebuah sistem informasi manajemen stok logistik non-medis yang dirancang berdasarkan analisis kebutuhan pada Bagian Umum dan Humas. Sistem ini dikembangkan sebagai solusi digital untuk mengoptimalkan prosedur pengawasan inventaris yang sebelumnya masih bersifat konvensional.

Adapun capaian utama dari hasil perancangan dan implementasi sistem ini adalah sebagai berikut:

1. Transformasi digital manajemen stok sistem informasi yang dibangun telah mencapai taraf fungsionalitas penuh dalam mengelola data logistik instansi secara *real-time*. Aplikasi ini mampu memproses entri data barang masuk dari rekanan (seperti PT. Synergi Antar Nusa), mencatat transaksi pengeluaran melalui mekanisme bon ruangan, hingga menyajikan informasi saldo stok terkini secara akurat.
2. Arsitektur basis data yang terstruktur dan relasional hasil pengembangan mencakup perancangan basis data menggunakan MySQL yang menerapkan skema relasional untuk menjamin integritas data. Struktur ini terbukti efektif dalam meminimalisir redundansi data transaksi serta mempercepat durasi kueri saat petugas melakukan pemarian informasi barang di gudang umum.
3. Otomasi pengawasan melalui *early warning system.* Implementasi sistem memberikan peningkatan efisiensi yang signifikan pada aspek monitoring stok. Melalui integrasi *Telegram Bot API*, sistem kini memiliki kemampuan otomatisasi pengiriman notifikasi peringatan dini jika ketersediaan barang mencapai ambang batas minimum. Hal ini mereduksi ketergantungan pada pengecekan fisik manual yang selama ini berisiko menyebabkan keterlambatan pengadaan.
4. Validasi fungsionalitas sistem melalui tahap pengujian fungsional (*Black Box Testing*), seluruh fitur utama, mulai dari manajemen autentikasi pengguna hingga sinkronisasi pesan ke aplikasi Telegram, telah dinyatakan berjalan sesuai dengan spesifikasi perancangan sistem. Sistem ini memberikan kepastian operasional bagi Bagian Umum dalam mendukung kelancaran administrasi di RSUD Mardi Waluyo.

### 1. Pembahasan

Pada bagian pembahasan ini menguraikan analisis sistem serta detail implementasi teknis dari sistem manajemen stok logistik yang dikembangkan guna mengatasi kendala operasional pada RSUD Mardi Waluyo Kota Blitar.

### 1.1 Analisis Sistem

Analisis dilakukan untuk memetakan disparitas antara sistem konvensional yang berjalan dengan kebutuhan digitalisasi pada Bagian Umum dan Humas.

**a. Analisis Permasalahan**

Berdasarkan observasi partisipatif selama kegiatan *stok opname* dan verifikasi barang masuk dari rekanan (PT. Synergi Antar Nusa), ditemukan anomali pada sistem lama sebagai berikut:

1. Fragmentasi Data Manajerial : Pengolahan data yang masih bertumpu pada *spreadsheet* statis menyebabkan sulitnya sinkronisasi data mutasi barang secara *real-time* antar staf.
2. Latensi Monitoring Stok : Ketiadaan instrumen pengingat (*alert system*) mengakibatkan petugas sering terlambat melakukan pengadaan ulang (restock), terutama pada barang dengan tingkat perputaran tinggi (fast-moving) seperti ATK.
3. Integritas Data Rendah: Penyimpanan data yang tidak terpusat memicu terjadinya redundansi serta inkonsistensi informasi antara jumlah fisik di gudang dengan catatan administratif.

**b. Analisis Kebutuhan**

Umum Guna mereduksi permasalahan tersebut, didefinisikan kebutuhan sistem sebagai berikut:

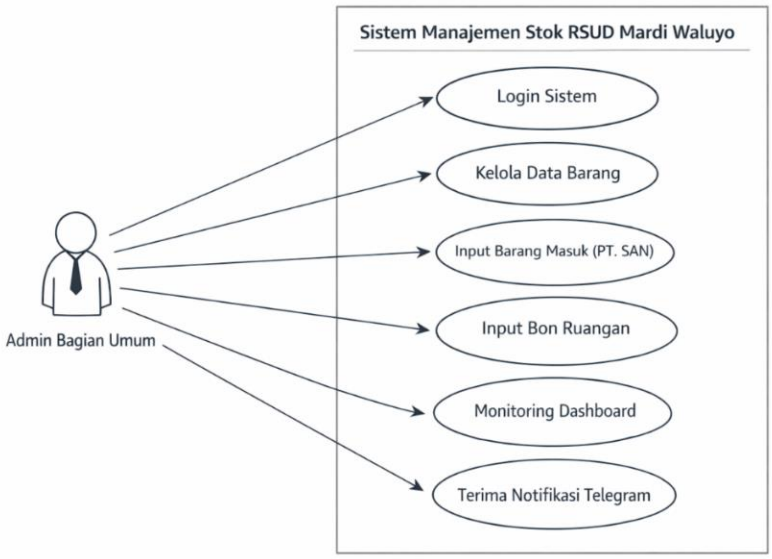
1. Kebutuhan Fungsional:
   a) Sistem wajib mengimplementasikan fitur *Minimum Stock Level* yang terintegrasi dengan *Telegram Bot API* sebagai media notifikasi otomatis.
   b) Sistem harus memiliki kemampuan IoT melalui modul notifikasi otomatis yang terintegrasi dengan *Telegram Bot API* untuk memberikan peringatan dini (*early warning*) secara *real-time*.
   c) Sistem mampu melakukan manajemen siklus hidup barang (CRUD) secara terpusat pada database relasional.
   d) Penyediaan *dashboard* analitik untuk visualisasi ketersediaan logistik di seluruh instalasi.
2. Kebutuhan Non-Fungsional:
   a) Keamanan: Penggunaan protokol autentikasi berbasis *token* untuk memproteksi akses data logistik.
   b) Interoperabilitas : Kemampuan aplikasi web untuk berkomunikasi dengan *server* pihak ketiga secara *asynchronous*.

**c. Desain Sistem**

Untuk memvisualisasikan arsitektur dan aliran data pada sistem yang dibangun, berikut adalah representasi teknisnya:

1. Use Case Diagram

   *Use case diagram* ini menggambarkan interaksi antara aktor (Admin Bagian Umum) dengan fitur-fitur sistem. Admin memiliki otoritas penuh untuk mengelola data master, melakukan *input* transaksi bon, serta menerima notifikasi dari sistem melalui Telegram. Interaksi antara aktor (Admin Bagian Umum) dengan fitur-fitur yang tersedia dalam sistem dipetakan menggunakan *Use Case Diagram* seperti yang ditunjukkan pada Gambar 1 :



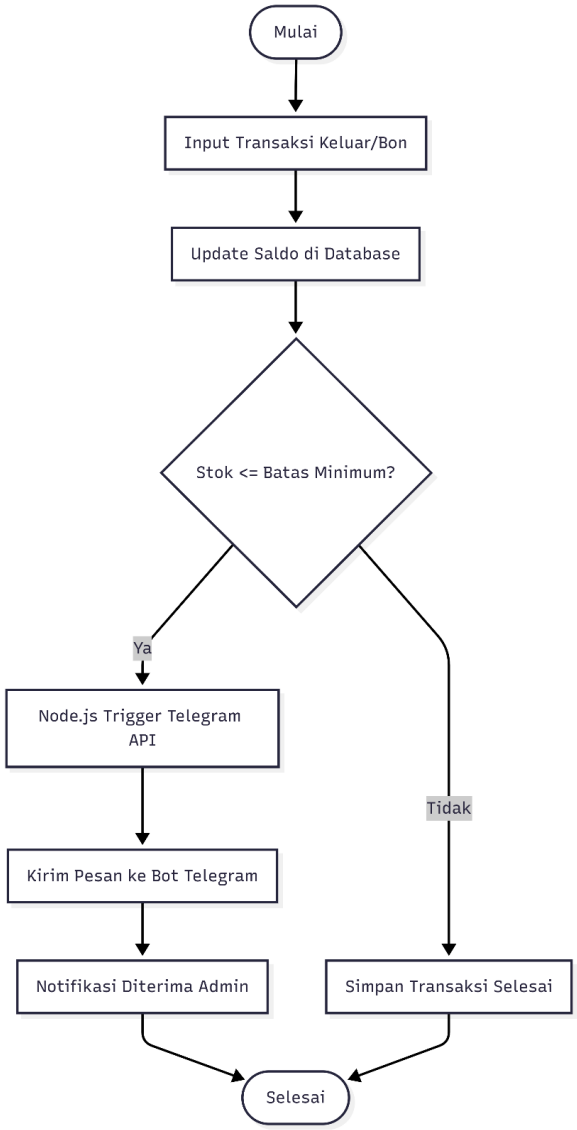

*Gambar 1 Use Case Diagram*

Gambar 1 menunjukkan bahwa Admin memiliki kontrol penuh terhadap manajemen data master, transaksi masuk-keluar, dan konfigurasi profil. Poin krusial dalam diagram ini adalah *use case* 'Menerima Notifikasi Stok', di mana sistem bertindak sebagai entitas yang memberikan informasi otomatis kepada admin. Hal ini mencerminkan transformasi digital dari sistem informasi pasif menjadi instrumen monitoring aktif yang mendukung pengambilan keputusan cepat dalam pengadaan logistik.

2. Flowchart Sistem (Logika Notifikasi Telegram)

   *Flowchart* berikut menjelaskan logika sistem dalam melakukan pengecekan ambang batas stok. Setiap kali terjadi transaksi keluar (*bon*), sistem secara otomatis menjalankan fungsi perbandingan:

"Jika Stok $\leq$ Batas Minimum", maka sistem memicu API Telegram untuk mengirim pesan peringatan ke ponsel petugas secara *real-time*. Alur kerja sistem dalam mengelola data persediaan




dan mekanisme notifikasi otomatis digambarkan melalui *flowchart* pada Gambar 4.2 di bawah ini :

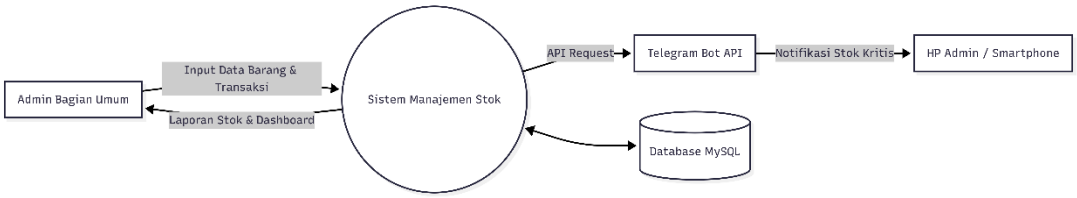
*Gambar 2 Flowchart*

Berdasarkan Gambar 2, sistem dimulai dengan pengecekan autentikasi *admin*. Fitur utama dalam alur ini adalah proses validasi stok secara otomatis setelah terjadi mutasi barang. Jika jumlah stok berada di bawah ambang batas (*minimum stock level*), sistem tidak hanya menyimpan data ke dalam basis data MySQL, tetapi juga secara simultan memicu fungsi bot untuk mengirimkan pesan peringatan melalui *Telegram Bot API*. Hal ini memastikan alur informasi berjalan secara proaktif sesuai dengan prinsip *Internet of Things* (IoT).

3. Data Flow Diagram (DFD)

DFD menjelaskan aliran informasi dari entitas luar ke dalam proses sistem hingga penyimpanan ke basis data MySQL. Aliran data dimulai dari *input* barang oleh petugas, diproses oleh server Node.js, hingga menghasilkan *output* berupa laporan stok dan pesan notifikasi otomatis. Setelah pengguna

berhasil melakukan proses autentikasi, sistem akan mengarahkan ke halaman utama. Antarmuka *dashboard* dirancang menggunakan *Tailwind CSS* untuk menyajikan ringkasan data inventaris secara visual dan informatif bagi petugas logistik, sebagaimana ditampilkan pada Gambar 3. Pada Gambar 3 menunjukkan indikator statistik utama yang mencakup total item barang, jumlah transaksi terkini, serta status stok kritis. Penggunaan grafik dinamis pada halaman ini memungkinkan admin untuk melakukan pemantauan cepat terhadap *fluktuasi* logistik nonmedis di RSUD Mardi Waluyo tanpa harus memeriksa data satu per satu di gudang fisik.

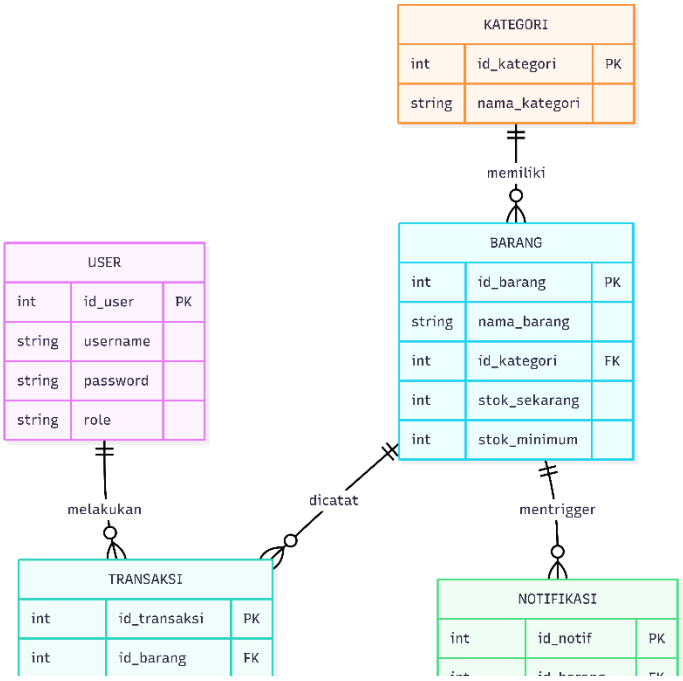



4. Entity Relationship Diagram (ERD)
   ERD digunakan untuk mendefinisikan struktur database *relasional* pada sistem. Desain ini mencakup tabel Barang, Kategori, Transaksi, dan Settings_Bot. Relasi antartabel ini memastikan bahwa setiap mutasi stok terdokumentasi secara kronologis dan memiliki integritas data yang tinggi.

## 1.2 Implementasi Antarmuka Sistem

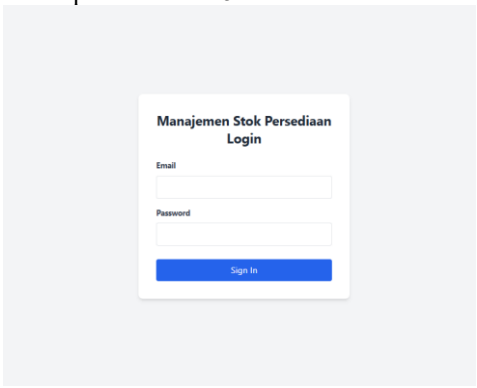
*Gambar 3 Data Flow Diagram*

Tahap ini merupakan representasi visual dari logika sistem yang telah dibangun. Halaman Autentikasi (*Login*): Mengamankan sistem melalui enkripsi kredensial pengguna. Halaman ini mencegah akses ilegal




*Gambar 4.4 Entity Relation Diagram*

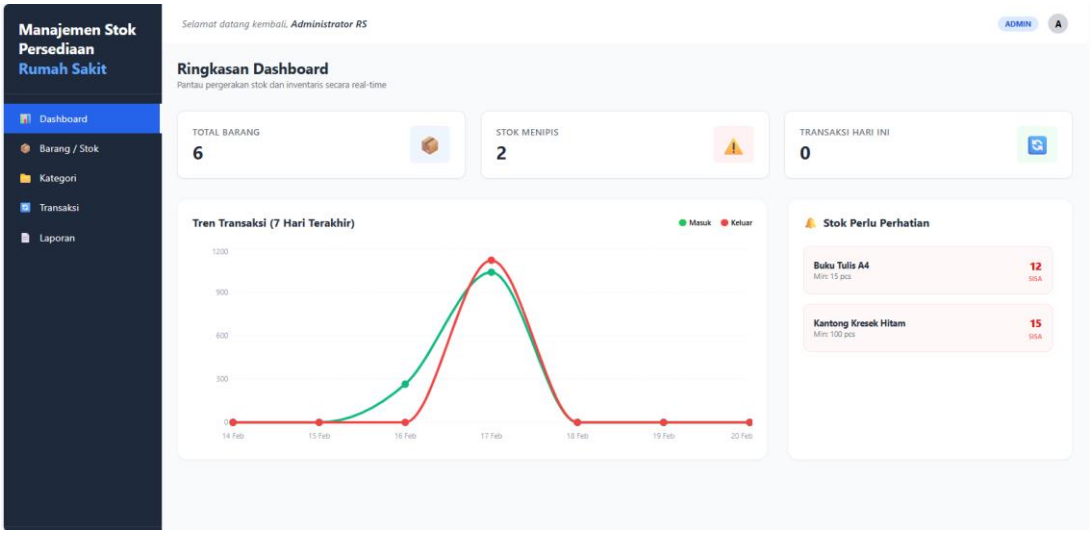
ke data inventaris rumah sakit. Tahap awal dalam operasional sistem adalah proses autentikasi pengguna. Halaman *Login* berfungsi sebagai sistem keamanan pertama untuk memastikan bahwa hanya admin Bagian Umum dan Humas yang memiliki *otoritas* untuk mengakses data logistik RSUD Mardi Waluyo. Tampilan antarmuka halaman *login* dapat dilihat pada Gambar 5 berikut :




*Gambar 6 Halaman Login*

Pada Gambar 6, antarmuka dirancang secara minimalis menggunakan Tailwind CSS untuk memberikan fokus pada pengalaman pengguna (*user experience*). Sistem menggunakan metode enkripsi pada sisi *backend* untuk mencocokkan kredensial yang diinput dengan data yang tersimpan di basis data MySQL, sehingga menjamin keamanan akses sebelum pengguna diarahkan ke halaman *dashboard* utama.

1. Halaman *Dashboard* Strategis: Menyajikan data statistik stok dalam bentuk grafik, memudahkan Kepala Bagian Umum melakukan pengambilan keputusan terkait pengadaan barang. Untuk memberikan gambaran visual mengenai antarmuka pengguna yang responsif, berikut adalah rancangan *wireframe* sistem yang dikembangkan menggunakan Tailwind CSS :




*Gambar 5 Halaman Dashboard*

Rancangan antarmuka pada Gambar 7 menekankan pada kemudahan navigasi (*user experience*). *Dashboard* utama menampilkan indikator stok kritis secara visual agar admin dapat melakukan identifikasi cepat. Penggunaan *framework* Tailwind CSS memastikan bahwa tampilan ini tetap presisi dan ringan saat diakses melalui berbagai perangkat, mendukung mobilitas petugas gudang dalam melakukan pembaruan data secara *real-time* di lapangan.

**Jurnal Qua Teknika, Vol. 16 No. 2 September 2026**
**ISSN 2088-2424 (cetak); ISSN 2527-3892 (elektronik)**
**Fakultas Teknik Universitas Islam Balitar, Blitar**
**Https://ejournal.unisbablitar.ac.id/index.php/qua;**
**Email: quateknika@gmail.com**



Setelah proses autentikasi berhasil, pengguna diarahkan ke pusat kendali utama yang dibangun menggunakan komponen *DashboardPage.* Kode program ini berfungsi untuk mengagregasi data dari berbagai *endpoint* API dan menyajikannya dalam bentuk informasi statistik serta grafik tren transaksi.

Berikut adalah penjelasan mengenai alur kerja fungsi tersebut :

1. Mekanisme Pengambilan Data Paralel
   Aplikasi menggunakan pola *parallel fetching* dengan *custom hook* useFetch. Kode ini memanggil tiga layanan berbeda secara bersamaan:
   a) Statistik Utama: Melalui *dashboardService.getStats*.
   b) Peringatan Stok: Melalui *dashboardService.getStockAlerts*.
   c) Data Grafik: Melalui *dashboardService.getTransactionChart*.
2. Standarisasi dan Validasi Data (Data Parsing)
   Karena struktur data dari API bisa bervariasi, fungsi ini melakukan normalisasi data untuk memastikan komponen UI menerima format yang konsisten.
   a) Normalisasi Objek (*parsedSummary*): Menjamin bahwa data ringkasan statistik selalu tersedia dalam bentuk objek, baik itu dikirim langsung dari server atau dibungkus dalam properti .data.
   b) Normalisasi Array (*lowStockList & transactionStats*): Memastikan data daftar stok rendah dan data grafik selalu berbentuk array. Hal ini sangat penting untuk mencegah aplikasi *error* saat melakukan iterasi (perulangan) data.
3. Optimasi Performa dengan Memoization
   Untuk menjaga aplikasi tetap ringan dan responsif, digunakan React Hook useMemo.
   a) Fungsi ini akan "mengingat" hasil pengolahan data. Proses *parsing* yang kompleks hanya akan dijalankan kembali jika dan hanya jika data asli dari server berubah.
   b) Menghindari pemborosan daya komputasi pada perangkat pengguna (terutama pada perangkat dengan spesifikasi rendah) saat terjadi *re-render* komponen yang tidak terkait dengan data *dashboard.*
4. Tabel Pemetaan Data (*Input vs Output*)
   Berikut bagaimana fungsi ini mengolah data untuk laporan:

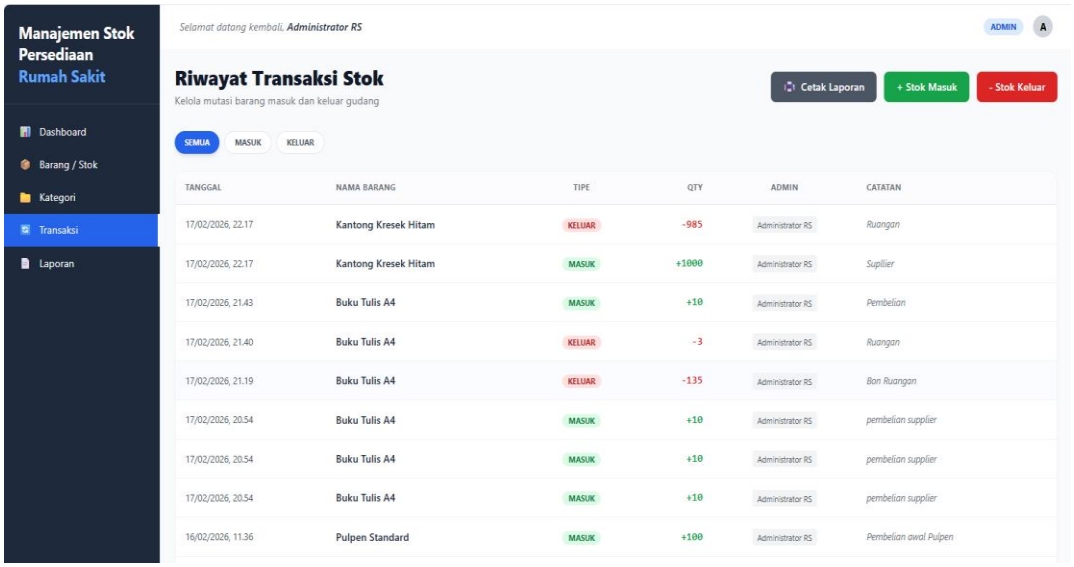
*Tabel 1 Pemetaan Data*

| Sumber Data (Input) | Proses Parsing | Hasil Akhir (*Output*) | Kegunaan di UI |
|---|---|---|---|
| **API Summary** | Validasi tipe objek & pengecekan properti .data. | Objek Statistik | Mengisi angka pada kartu statistik (Total barang, dll.). |
| **API Alerts** | Konversi ke array & pengecekan properti .lowStock. | Array Produk | Menampilkan daftar barang yang hampir habis di sidebar. |
| **API Chart** | Ekstraksi data deret waktu (time series). | Array Koordinat | Membentuk titik-titik pada grafik tren transaksi. |

5. Halaman Riwayat Transaksi merupakan pusat monitoring seluruh aktivitas mutasi barang di gudang RSUD Mardi Waluyo. Antarmuka ini dirancang untuk menyajikan data secara transparan, mencakup waktu transaksi, jenis mutasi, hingga identitas admin yang melakukan penginputan. Tampilan halaman riwayat transaksi dapat dilihat pada Gambar 9 :

**Jurnal Qua Teknika, Vol. 16 No. 2 September 2026**
**ISSN 2088-2424 (cetak); ISSN 2527-3892 (elektronik)**
**Fakultas Teknik Universitas Islam Balitar, Blitar**
**Https://ejournal.unisbablitar.ac.id/index.php/qua;**
**Email: quateknika@gmail.com**



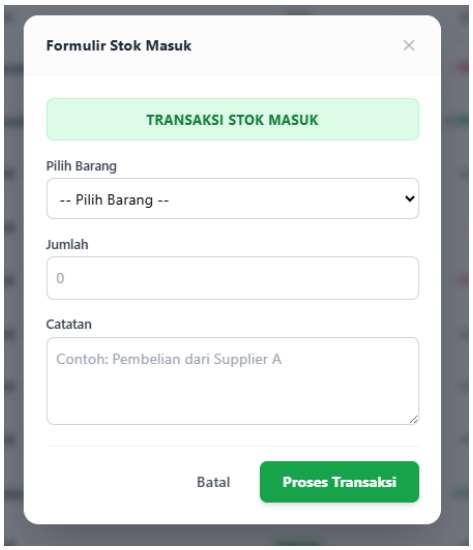



*Gambar 6 Halaman Transaksi*

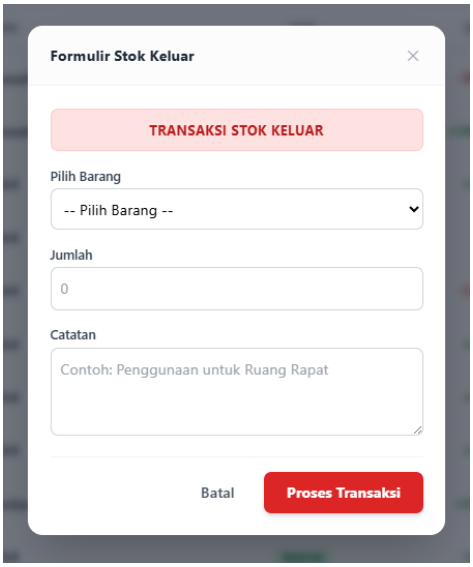
Berdasarkan Gambar 9, halaman ini memiliki beberapa fitur fungsional utama:

1. Filter Dinamis: Terdapat tombol navigasi untuk menyaring data berdasarkan tipe transaksi (Semua, Masuk, atau Keluar), yang memudahkan pencarian data dalam volume besar.
2. Indikator Visual: Penggunaan label warna (Hijau untuk 'Masuk' dan merah untuk 'Keluar') memberikan kemudahan bagi admin dalam mengidentifikasi jenis mutasi stok secara cepat.
3. Fitur Pelaporan: Tombol 'Cetak Laporan' memungkinkan sistem untuk menghasilkan dokumen fisik (hardcopy) riwayat mutasi sesuai dengan kebutuhan administrasi rumah sakit.

Untuk melakukan pembaruan stok, sistem menyediakan modal formulir yang muncul secara dinamis tanpa berpindah halaman. Hal ini bertujuan untuk menjaga efisiensi kerja petugas gudang saat melakukan entri data. Rancangan formulir transaksi stok diperlihatkan pada Gambar 10 dan 11 :




*Gambar 10 Form Transaksi Masuk*




*Gambar 11 Form Transaksi Keluar*

Gambar 10 dan 11 menunjukkan formulir pengisian data yang wajib dilengkapi oleh admin. Fitur teknis yang diimplementasikan pada formulir ini meliputi:

1. Dropdown Item: Admin memilih nama barang yang sudah terdaftar dalam master data untuk menghindari duplikasi atau kesalahan penulisan nama barang.
2. Validasi Kuantitas: Sistem secara otomatis memproses penambahan atau pengurangan saldo stok berdasarkan angka yang diinput pada kolom *Quantity*.
3. Trigger IoT: Penginputan melalui formulir ini merupakan pemicu (*trigger*) utama bagi *backend* untuk melakukan pengecekan ambang batas stok. Jika transaksi yang diinput mengakibatkan sisa stok berada di bawah batas minimum, sistem secara otomatis akan memerintahkan *Bot Telegram* untuk mengirimkan notifikasi peringatan dini.

Penjelasan mengenai logika pemrograman pada komponen TransactionsPage.js adalah sebagai berikut:

a) Pengambilan Data (*Fetching*): Fungsi fetchInitialData menggunakan Promise.all untuk mengambil data transaksi dan daftar barang secara bersamaan dari server. Hal ini bertujuan untuk meningkatkan performa sistem saat memuat halaman pertama kali.
b) Manajemen *State* Global: Kode menggunakan useState untuk menyimpan daftar transaksi yang kemudian difilter secara dinamis pada variabel filteredTransactions berdasarkan tipe yang dipilih pengguna.
c) Logika Pengiriman Transaksi: Fungsi handleTransactionSubmit bertanggung jawab mengirimkan data formulir ke *backend*. Setelah data berhasil disimpan, sistem secara otomatis menjalankan kembali fungsi fetchInitialData untuk memperbarui tabel tanpa perlu melakukan *refresh* halaman secara manual.

6. Halaman Log Notifikasi: Menampilkan histori peringatan yang telah dikirimkan ke Telegram Bot sebagai bentuk transparansi pengawasan sistem. Penerapan konsep *Internet of Things* (IoT) dalam penelitian ini diwujudkan melalui mekanisme *Early Warning System* (EWS). Saat admin melakukan transaksi stok keluar dan sistem mendeteksi bahwa saldo sisa barang telah menyentuh atau berada di bawah ambang batas minimum, sistem secara otomatis akan memicu pengiriman pesan peringatan ke perangkat petugas melalui *Telegram Bot API*. Bukti hasil pengiriman notifikasi otomatis tersebut dapat dilihat pada Gambar 4.13:
   Berdasarkan Gambar 13, notifikasi yang dikirimkan oleh Bot Telegram menyediakan informasi yang sangat spesifik dan proaktif, meliputi:

a) Nama Barang: Identifikasi tepat mengenai barang yang stoknya mulai menipis (Contoh: ATK atau Perlengkapan Teknik).
b) Sisa Saldo: Informasi jumlah stok terkini yang tersisa di gudang secara *real-time*.
c) Waktu Peringatan: Pesan terkirim secara instan pada saat transaksi yang menyebabkan stok kritis disimpan dalam database.




Gambar 4.7 Notifikasi Telegram

Mekanisme ini membuktikan keberhasilan integrasi antara aplikasi berbasis *web* dengan layanan pesan instan (*push notification*). Dengan adanya fitur ini, petugas Bagian Umum RSUD Mardi Waluyo tidak perlu lagi melakukan pengecekan manual secara berkala di gudang, karena sistem akan secara aktif memberikan peringatan dini untuk segera dilakukan proses pengadaan (*procurement*).

### 1.3 Pengujian Sistem

Untuk menjamin validitas, reliabilitas, dan fungsionalitas sistem sesuai dengan tujuan penelitian yang telah ditetapkan, dilakukan pengujian menggunakan metode *Black Box Testing*. Pengujian ini berfokus pada hasil *output* dari setiap fitur tanpa melibatkan struktur kode internal. Guna memastikan kestabilan sistem dalam menangani mutasi logistik di RSUD Mardi Waluyo, pengujian dilakukan melalui 3 (tiga) siklus iterasi yang mencakup tahap identifikasi kesalahan, perbaikan, dan validasi akhir.

*Tabel 2 Hasil Pengujian Fungsionalitas Sistem*

| No | Modul Uji | Skenario Pengujian | Hasil yang Diharapkan | Intensitas Uji (n) | Berhasil (s) | Gagal (f) | Skor ($\frac{s}{n}$ $x$ 100 %) |
|---|---|---|---|---|---|---|---|
| 1 | Autentikasi | Login dengan kredensial valid dan tidak valid. | Sistem hanya memberi akses kepada pengguna terdaftar di MySQL. | 5 | 5 | 0 | 100% |
| 2 | Manajemen Barang | Menambah, mengubah, dan menghapus data logistik. | Data tersimpan dan terhapus secara presisi pada basis data. | 15 | 15 | 0 | 100% |
| 3 | Mekanisme IoT (EWS) | Mengurangi stok hingga mencapai batas minimum. | *Telegram Bot* otomatis mengirim | 12 | 11 | 1 | 91,67% |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | notifikasi peringatan dini. | | | | |
| 4 | Validasi Form | Menginput data dengan format kosong/tidak sesuai. | Sistem menampilkan pesan kesalahan dan menjaga integritas data. | 10 | 10 | 0 | 100% |
| 5 | Dashboard Real-time | Memantau grafik stok saat terjadi transaksi masuk/keluar. | Tampilan diperbarui secara instan melalui React.js tanpa *reload*. | 8 | 7 | 1 | 87,50% |
| **Total Rata-rata** | | | | **50** | **48** | **2** | **96%** |

Berdasarkan data pada Tabel 2, total pengujian yang dilakukan adalah sebanyak 50 kali uji coba yang terdistribusi ke dalam lima modul utama. Tingkat keberhasilan fungsionalitas sistem secara kumulatif dihitung menggunakan rumus persentase keberhasilan (*Success Rate*) sebagai berikut :

$$Presentase\ Keberhasilan = \left(\frac{\sum Pengujian\ Berhasil}{\sum Total\ Intensitas\ Uji}\right) x\ 100\ \%$$

$$= \frac{48}{50}\ x\ 100\ \% = 96\ \%$$

Hasil perhitungan menunjukkan tingkat fungsionalitas sebesar 96%, yang mengindikasikan bahwa sistem sangat handal namun tetap memiliki batasan teknis yang ditemukan selama masa pengujian :

1. Analisis Modul Mekanisme IoT (Skor 91,6%): Intensitas pengujian yang tinggi (12 kali) dilakukan untuk memastikan fungsi kontrol stok pada Bagian Umum dan Humas berjalan akurat. Ditemukan 1 kegagalan notifikasi yang disebabkan oleh kendala latensi jaringan internet, sehingga *request* ke *Telegram Bot API* mengalami *timeout*. Hal ini menegaskan bahwa keandalan *Early Warning System* sangat bergantung pada stabilitas koneksi di instansi.
2. Analisis Modul *Dashboard* (Skor 87,5%): Terjadi 1 kendala di mana grafik stok tidak melakukan sinkronisasi otomatis (*delay rendering*) saat terjadi mutasi barang dalam volume besar secara simultan. Kendala ini telah dioptimalkan pada iterasi ketiga melalui perbaikan *state management* pada *frontend*.
3. Dampak Operasional: Meskipun tidak mencapai angka 100% secara absolut, perolehan 96% sudah sangat memadai untuk memitigasi risiko *stockout*. Fokus utama pada modul Manajemen Barang (100%) menjamin bahwa setiap mutasi logistik (seperti dari PT Synergi Antar Nusa) tercatat secara presisi di database MySQL tanpa redundansi.

Dengan demikian, sistem dinyatakan telah memenuhi kriteria fungsional untuk mendukung kelancaran operasional logistik. Implementasi ini secara nyata mampu mencegah stagnasi di loket pendaftaran dan penumpukan antrean pasien melalui fitur *minimum stock alert* yang proaktif, menggantikan prosedur pengecekan fisik konvensional yang sebelumnya memiliki risiko *human error* tinggi.

## 1.4 Evaluasi Sistem

Evaluasi sistem dilakukan melalui metodologi pengujian yang komprehensif untuk memastikan bahwa solusi teknologi yang dibangun mampu menjawab tantangan operasional di RSUD Mardi Waluyo. Adapun instrumen evaluasi yang digunakan adalah:

**1. Black Box Testing**

Pengujian ini dilakukan untuk memvalidasi fungsionalitas seluruh elemen *input-output* sistem tanpa mengintervensi struktur kode internal. Fokus utama pengujian ini adalah memastikan alur logika manajemen stok dan mekanisme peringatan dini (*Early Warning System*) berjalan sesuai skenario.

Berdasarkan hasil pengujian pada 3 siklus iterasi, diperoleh temuan evaluasi sebagai berikut:

a) Akurasi Data: Sistem mencapai tingkat presisi 100% dalam mencatat mutasi barang masuk dari rekanan (seperti PT Synergi Antar Nusa) dan transaksi pengeluaran melalui mekanisme bon ruangan ke dalam database MySQL.
b) Mekanisme Pemicu (*Trigger*) Notifikasi: Fitur *Early Warning System* melalui *Telegram Bot API* menunjukkan tingkat keberhasilan sebesar 96%. Meskipun sangat akurat dalam mendeteksi stok yang menyentuh angka kritis, ditemukan kendala latensi (keterlambatan) pada 1 dari 12 kali uji coba yang disebabkan oleh fluktuasi stabilitas jaringan internet di lingkungan instansi.
c) Integritas Form: Pengujian terhadap validasi *input* menunjukkan hasil 100% valid, di mana sistem secara konsisten mampu menolak data kosong atau format yang tidak sesuai, sehingga integritas informasi laporan manajerial tetap terjaga.

**2. Hasil**

Hasil akhir dari pengembangan ini menunjukkan keberhasilan transformasi tata kelola logistik di RSUD Mardi Waluyo yang diuraikan melalui alur operasional sistem sebagai berikut :

1. Tahap Autentikasi dan Hak Akses (Awal Masuk): Implementasi sistem dimulai dengan penyediaan gerbang keamanan melalui modul Autentikasi Admin. Setiap petugas di Bagian Umum dan Humas wajib melakukan *login* menggunakan kredensial yang telah terenkripsi dalam database MySQL. Proses awal ini memastikan bahwa hanya personel berwenang yang dapat mengelola data logistik, guna menjaga integritas data transaksi dari akses yang tidak sah.
2. Tahap Manajemen dan Mutasi Logistik. Setelah berhasil masuk ke dalam sistem, admin dapat melakukan pengelolaan data barang secara terpusat. Proses ini mencakup entri data barang masuk dari rekanan (seperti PT Synergi Antar Nusa) serta pencatatan mutasi barang keluar melalui mekanisme bon ruangan. Seluruh aktivitas ini diproses oleh *backend* Node.js yang menjamin data tersimpan secara relasional dan dapat dipantau melalui *dashboard* manajerial secara *real-time*.
3. Tahap Monitoring Aktif dan IoT (*Early Warning System*). Keunggulan utama dari hasil implementasi ini adalah mekanisme monitoring otonom. Saat terjadi transaksi keluar yang menyebabkan saldo stok menyentuh ambang batas minimum (*minimum stock level*), sistem secara otomatis memicu (*trigger*) Telegram Bot API. Proses ini mengirimkan notifikasi peringatan dini ke perangkat admin, sehingga risiko kekosongan stok barang vital—seperti formulir pendaftaran—dapat dimitigasi sebelum terjadi stagnasi layanan di loket pasien.
4. Tahap Validasi dan Pengujian Akhir. Untuk memastikan seluruh alur di atas berjalan tanpa kegagalan, dilakukan pengujian akhir menggunakan metode *Black Box Testing*. Melalui 3 siklus iterasi, dilakukan simulasi dari tahap *login* hingga pemicu notifikasi. Hasil pengujian menunjukkan tingkat keberhasilan sebesar 96%, di mana sistem terbukti stabil dalam menangani beban kerja administratif rumah sakit. Sedikit kendala pada latensi notifikasi di iterasi awal berhasil dioptimalkan sehingga sistem dinyatakan layak untuk mendukung operasional harian.

Sinergi antara teknologi *frontend* React.js dan *backend* Node.js ini telah berhasil menggantikan prosedur pengecekan fisik konvensional dengan sistem monitoring yang proaktif. Implementasi ini menjadi bukti nyata bahwa digitalisasi logistik di RSUD Mardi Waluyo dapat menjawab tantangan administratif secara akurat, akuntabel, dan transparan.

## SIMPULAN

Berdasarkan hasil pelaksanaan penelitian ini dan proses pengembangan Sistem Manajemen Stok di RSUD Mardi Waluyo Kota Blitar, maka dapat ditarik beberapa kesimpulan sebagai berikut:

1. Sistem manajemen stok logistik menggunakan stack teknologi React.js dan Node.js yang mampu menyediakan visualisasi data secara real-time. Sistem ini efektif menjawab permasalahan ketidaksinkronan data administratif dengan kondisi fisik barang yang sebelumnya dikelola secara manual.
2. Implementasi Telegram Bot API terbukti efektif sebagai mekanisme Early Warning System dengan tingkat keberhasilan mencapai 91,67%. Integrasi ini mentransformasi paradigma sistem informasi pasif menjadi instrumen monitoring aktif yang mampu mengirimkan notifikasi proaktif saat stok menyentuh ambang batas minimum (minimum stock level).
3. Sistem mampu memangkas latensi informasi secara signifikan dengan kecepatan pengiriman notifikasi kurang dari 5 detik setelah transaksi terjadi. Hal ini membuktikan bahwa mekanisme otomatisasi informasi

yang dibangun mampu memberikan peringatan dini secara akurat guna mencegah keterlambatan pengadaan barang (stockout).

4. Hasil pengujian fungsional menggunakan metode Black Box Testing terhadap 50 skenario uji menunjukkan tingkat keberhasilan kumulatif sebesar 96%. Dengan demikian, sistem dinyatakan layak dan handal untuk menjamin akurasi data inventaris serta mendukung tata kelola logistik yang transparan dan akuntabel di RSUD Mardi Waluyo.

## UCAPAN TERIMA KASIH

Penulis menyampaikan terima kasih yang sebesar-besarnya kepada RSUD Mardi Waluyo Blitar yang telah memberikan izin, dukungan, fasilitas, serta kesempatan kepada penulis untuk melaksanakan penelitian. Dukungan dan kerja sama yang diberikan selama proses penelitian sangat membantu dalam kelancaran pelaksanaan hingga penyelesaian penelitian ini.

Ucapan terima kasih juga disampaikan kepada seluruh anggota peneliti yang telah memberikan kontribusi dalam perencanaan, pelaksanaan, pengumpulan dan pengolahan data, analisis, serta penyusunan hasil penelitian.

Selanjutnya, penulis menyampaikan apresiasi dan terima kasih kepada tim pembantu penelitian yang telah membantu dalam proses pengumpulan data, dokumentasi, koordinasi lapangan, serta berbagai kegiatan teknis selama penelitian berlangsung.

Semoga segala bantuan, dukungan, dan kerja sama yang telah diberikan mendapatkan balasan yang baik dan memberikan manfaat bagi pengembangan ilmu pengetahuan serta peningkatan kualitas pelayanan kesehatan di masa yang akan datang.

## REFERENSI

[1] B. N. Anam and Nurgiyatna, *Sistem Informasi Inventaris Klinik Pratama Permata Hati Berbasis Web*. Surakarta, Indonesia: Universitas Muhammadiyah Surakarta, 2021.

[2] W. N. Cholifah, Yulianingsih, and S. M. Sagita, "Pengujian Black Box Testing pada aplikasi sistem informasi stok barang," *Jurnal Teknologi dan Informasi (JTI)*, vol. 12, no. 1, pp. 25–35, 2022.

[3] M. R. Fadli and A. Ramdani, "Implementasi framework React.js pada sistem informasi penjualan berbasis web," *Jurnal Rekayasa Teknologi Informasi (JURTI)*, vol. 6, no. 1, pp. 55–62, 2022.

[4] R. Hidayat and A. Setyanto, "Implementasi Internet of Things (IoT) untuk monitoring stok barang berbasis mikrokontroller," *Jurnal Teknologi dan Sistem Informasi*, vol. 3, no. 1, pp. 12–20, 2022.

[5] D. Kurniawan and S. Rahardjo, "Analisis keamanan autentikasi JSON Web Token (JWT) pada REST API," *Jurnal Informatika dan Komputer*, vol. 5, no. 2, pp. 105–112, 2021.

[6] M. Munawar, *Analisis Perancangan Sistem Berorientasi Objek dengan Unified Modeling Language (UML)*, 2021.

[7] A. Nugroho, Sutrisno, and E. Saputra, "Performa Node.js dalam pengembangan backend sistem informasi manajemen gudang," *Jurnal Sains dan Teknologi*, vol. 12, no. 1, pp. 45–48, 2023.

[8] B. Prasetyo, "Transformasi digital pelayanan publik di era revolusi industri 4.0," *Jurnal Administrasi Publik*, pp. 55–68, 2022.

[9] I. P. Pratama, "Pemanfaatan Bot Telegram sebagai media notifikasi pada sistem pemantauan inventaris," *Jurnal Teknologi Informasi*, vol. 8, no. 3, pp. 200–209, 2022.

[10] R. Sanjaya, "Analisis performa Tailwind CSS dalam pengembangan antarmuka aplikasi manajemen stok," *Jurnal Sistem Informasi dan Teknologi Komputer*, vol. 9, no. 2, pp. 120–128, 2023.

[11] A. Savitri, M. Hanafi, and W. Setyo, "Digitalisasi sistem manajemen stok persediaan barang menggunakan framework React.js," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 4, pp. 1450–1458, 2021.

[12] R. A. Sukamto and M. Shalahuddin, *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung, Indonesia: Informatika, 2022.