

Sistem Informasi Manajemen Pergudangan Berbasis Web Pada Kantor Wilayah BRI Jakarta 3

Muhamad Reizal Putra Hidayat¹, Hadi Zakaria^{2*}, Fahri Rachmad Husaeni³, Natalia Anggreani Br.Sormin⁴

^{1,2,3,4} Ilmu Komputer, Teknik Informatika, Universitas Pamulang, Tangerang Selatan, Indonesia

Email: ¹muhmadreizalputra@gmail.com, ^{2*}dosen00274@unpam.ac.id, ³sanemi100203@gmail.com,

⁴natalia.anggreani.sormin@gmail.com

(* : coressponding author)

Abstrak – Teknologi informasi mendorong implementasi sistem untuk meningkatkan efisiensi operasional, terutama pada manajemen pergudangan yang masih banyak menggunakan metode manual rentan human error dan pelacakan stok non-real-time. Penelitian ini mengusulkan pengembangan Sistem Informasi Manajemen Pergudangan Digital Berbasis Web untuk Kantor Wilayah BRI Jakarta 3. Sistem ini dirancang untuk mengatasi kendala tersebut, mendukung transparansi data, efisiensi waktu, dan pengambilan keputusan yang akurat melalui akses digital terpusat. Dikembangkan menggunakan metode Waterfall, aplikasi ini bertujuan mengurangi human error dan meningkatkan akurasi serta mobilitas manajemen gudang.

Kata Kunci: Sistem *Inventory*; Berbasis Web; *Waterfall*; Manajemen Stok; Sveltekit

Abstract – Rapid advancements in information technology necessitate system implementation to enhance operational efficiency, particularly in warehouse management, which often still relies on manual methods prone to human error and lacking real-time stock tracking. This study proposes the development of a Web-Based Digital Warehouse Management Information System for BRI Jakarta 3 Regional Office. Designed to overcome these limitations, the system supports data transparency, time efficiency, and accurate decision-making through centralized digital access. Developed using the Waterfall method, this application aims to reduce human error and improve warehouse management accuracy and mobility

Keywords: *Inventory System*; *Web Based*; *Waterfall*; *Stock Managemen*, *Sveltekit*

1. PENDAHULUAN

Perkembangan teknologi informasi yang sangat pesat telah membawa perubahan signifikan dalam berbagai aspek kehidupan, termasuk dalam dunia industri dan logistik. Hal ini mendorong implementasi sistem informasi untuk meningkatkan efisiensi dan akurasi operasional di berbagai sektor. Salah satu aspek penting yang memerlukan adaptasi teknologi adalah sistem pergudangan. Gudang berfungsi sebagai pusat penyimpanan barang sementara sebelum dilakukan distribusi lebih lanjut. Oleh karena itu, manajemen gudang yang baik dan efisien sangat berpengaruh terhadap kelancaran operasional suatu perusahaan, terutama dalam bidang perdagangan. Pada kenyataannya, banyak perusahaan masih menggunakan metode manual dalam proses pencatatan dan pengelolaan barang di gudang, seperti menggunakan buku catatan atau aplikasi spreadsheet sederhana. Metode ini memiliki banyak keterbatasan, seperti *human error*, risiko kesalahan pencatatan, keterlambatan dalam laporan, kehilangan data, dan kesulitan dalam melakukan pelacakan stok barang secara real-time.

Melihat permasalahan tersebut, dibutuhkan sebuah solusi berbasis teknologi yang dapat mengatasi kendala-kendala tersebut. Sistem manajemen pergudangan digital berbasis web merupakan salah satu solusi yang tepat. Sistem ini dirancang untuk membantu proses pengelolaan stok barang secara digital, terpusat, dan dapat diakses dari mana saja melalui jaringan internet. Penggunaan sistem berbasis web juga mendukung transparansi data, efisiensi waktu kerja, dan pengambilan keputusan yang lebih akurat karena informasi yang ditampilkan selalu mutakhir. Selain itu, implementasi sistem ini juga membantu dalam mengefisiensikan penggunaan kertas (*paperless*) dan mendukung praktik ramah lingkungan, sejalan dengan transformasi digital yang kini sedang berkembang di berbagai sektor industri.

Berdasarkan latar belakang tersebut, Kelompok Kerja Praktek Universitas Pamulang (UNPAM) yang beranggotakan 3 mahasiswa, mengajukan sebuah topik penelitian yang bertujuan untuk memberikan solusi terhadap permasalahan manajemen pergudangan di era digital saat ini.

Topik yang diusulkan berjudul “Sistem Informasi Manajemen Pergudangan Digital Berbasis Web Pada Kantor Wilayah Bri Jakarta 3”, yang dirancang untuk meningkatkan efisiensi, transparansi, dan akurasi dalam proses pengelolaan barang di gudang melalui pemanfaatan teknologi informasi berbasis web.

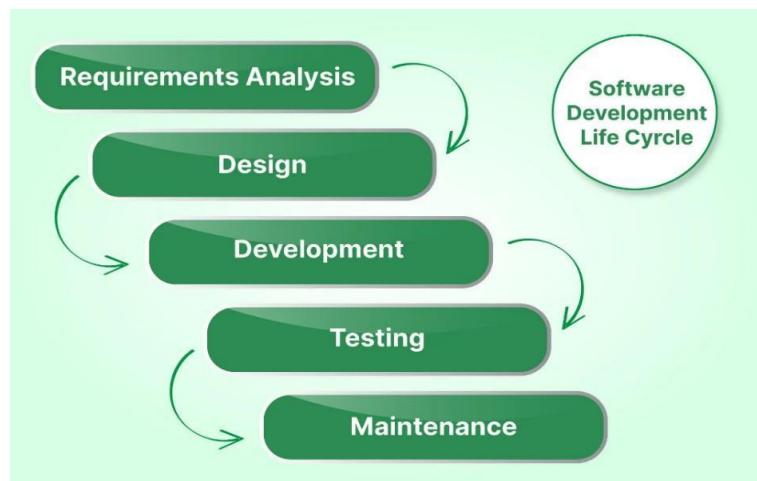
2. METODE PENELITIAN

2.1 Metode Pengumpulan Data

Data dikumpulkan melalui observasi serta wawancara dengan pegawai pada Kantor Wilayah BRI JAKARTA 4. Selain itu, dilakukan studi literatur terhadap pengembangan sistem manajemen gudang atau barang berbasis web.

2.2. Metode Pengembangan Sistem

Sistem dikembangkan menggunakan metode pengembangan sistem yang linier dan sekuensial, di mana setiap fase dalam siklus hidup pengembangan harus diselesaikan sepenuhnya sebelum fase berikutnya dimulai. Dalam konteks sistem inventory, metode Waterfall dapat diterapkan untuk membangun atau mengoptimalkan manajemen barang.



Gambar 1. Metode Waterfall

3. ANALISA DAN PEMBAHASAN

3.1 Kerangka Kerja Berdasarkan Waterfall

Skema kerja yang digunakan pada pembuatan aplikasi ini adalah metode waterfall, yang Dimana waterfall merupakan salah satu skema kerja yang banyak digunakan oleh pengembang software. Metode ini memiliki konsep linier dimana dalam penggerajannya dilakukan secara berurutan. Metode waterfall pada pembuatan aplikasi sebagai berikut:

a. *Requirements Analysis*(Analisa Kebutuhan sistem)

Menganalisa kebutuhan yang diperlukan sistem baik secara fungsional dan non fungsional. Hal yang diperlukan dalam membuat aplikasi manajemen barang dianalisis. Secara garis besar, alur sistem dimulai dari proses *login*, input barang masuk atau keluar, mengurangi atau menambahkan stok berdasarkan transaksi barang dan menyimpan di dalam database.

b. *Design*(Perancangan)

Tahap selanjutnya adalah perancangan. Perancangan sistem adalah tahapan Dimana alur sistem secara detail dibuat dan merancang desain masing-masing alur. Seperti bagaimana barang masuk atau keluar, data barang, data user.

c. *Development*(Implementasi)

Tahapan ini merubah semua tahapan sebelumnya menjadi sebuah kode. Pada aplikasi ini dijalankan pada Nodejs *runtime* dengan basis bahasa pemrograman Javascript dan menggunakan database Postgresql.

d. *Testing*(Pengujian)

Setelah tahapan Implementasi dilakukan, tahapan pengujian dilakukan. Tahapan ini pertama menguji masing-masing sub sistem pada aplikasi. Pengujian dilakukan agar mengecek apakah aplikasi berjalan sesuai dengan yang diinginkan. Jika pengujian masing- masing sub sistem berhasil, maka dilakukan pengujian keseluruhan sistem aplikasi.

e. *Maintanance*(Pemeliharaan)

Pemeliharaan dilakukan jika sistem rusak. Namun pemeliharaan juga dilakukan dalam jangka tertentu untuk memperbarui sistem aplikasi.

3.2 Analisa Kebutuhan Sistem

3.2.1 Kebutuhan Fungsional

Persyaratan fungsional memberikan gambaran tentang persyaratan sistem dan langkah-langkah yang harus dijalankan agar sistem dapat berfungsi dengan baik. Beberapa kebutuhan fungsional untuk rancang bangun sistem informasi manajemen barang antara lain sebagai berikut:

- a. Sistem dapat melakukan authentikasi(login)
- b. Sistem dapat membatasi user dan non-user
- c. Sistem dapat menambahkan barang beserta stok awal
- d. Sistem dapat menampilkan barang masuk dan keluar
- e. Sistem dapat menampilkan semua data barang(nama, stok)
- f. Sistem dapat membuat transaksi masuk dan keluar
- g. Sistem dapat menampilkan transaksi barang
- h. Sistem dapat melihat detail transaksi
- i. Sistem dapat menghapus transaksi
- j. Sistem dapat membuat user
- k. Sistem dapat mengubah user
- l. Sistem dapat menghapus user
- m. Sistem dapat melakukan logout

3.2.2 Kebutuhan non Fungsional

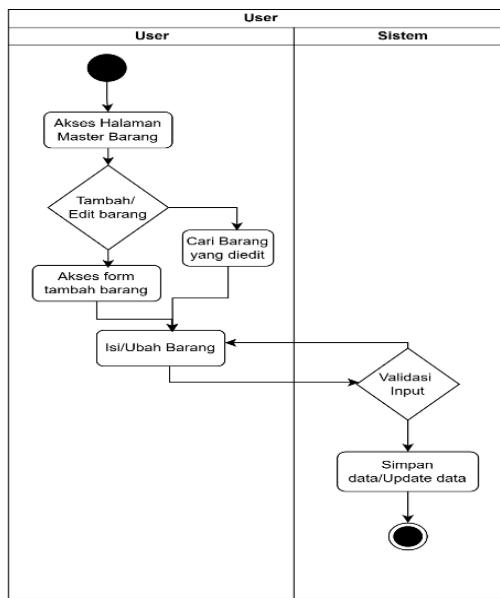
Kriteria yang digunakan untuk menilai kualitas suatu sistem, daripada perilaku atau fungsi spesifiknya. Berikut beberapa kebutuhan non fungsional :

- a. Login menggunakan username dan password
- b. Menggunakan session based
- c. Respon Time tidak lebih dari 2 detik
- d. Layar responsive
- e. UI simpel

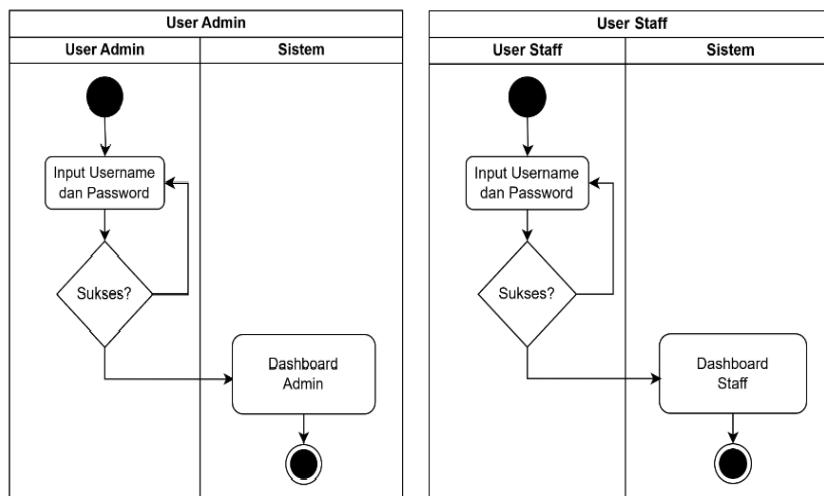
3.3 Perancangan Sistem

3.4.1 Activity Diagram

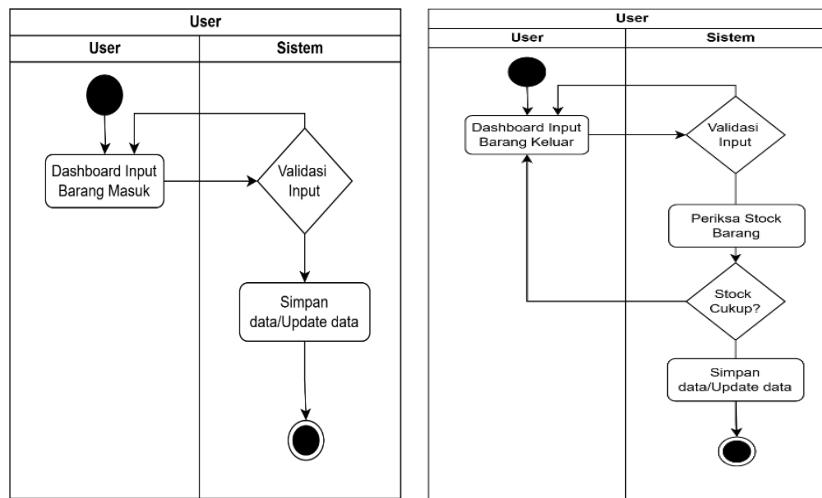
Activity Diagram dibuat untuk mendapatkan Gambaran atau ilustrasi bagaimana sistem berjalan atau *workflow*. Diagram yang dibuat dibawah ini menjelaskan bagaimana proses Ketika menambah barang, melakukan transaksi barang keluar dan masuk serta login untuk mengakses dashboard.



Gambar 2. *Activity Diagram* melakukan tambah barang



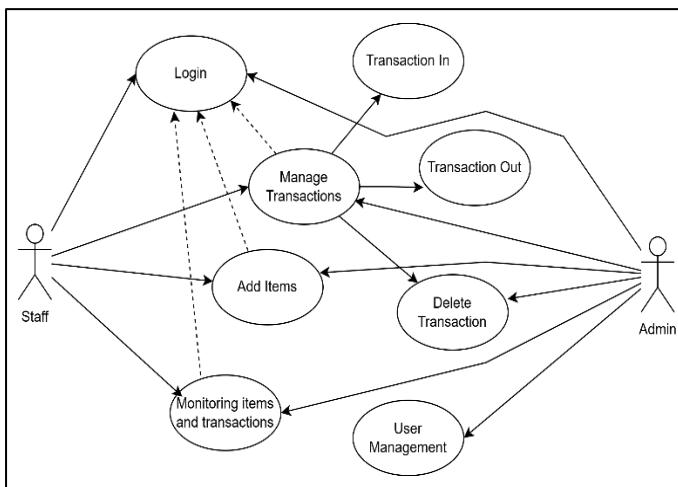
Gambar 3. *Activity Diagram* login admin dan staff



Gambar 4. Activity Diagram melakukan transaksi barang masuk dan keluar

3.4.2 Use Case Diagram

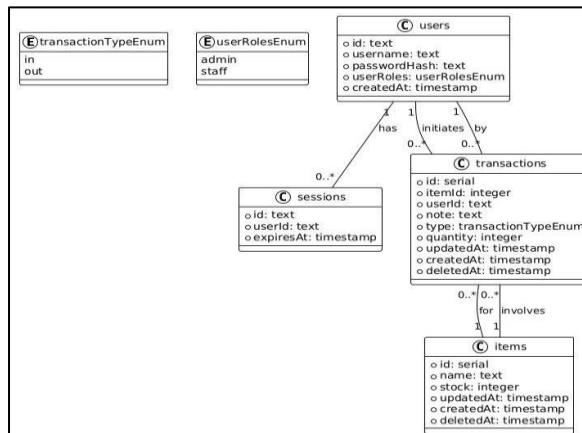
Use Case Diagram adalah salah satu jenis diagram dalam *UML (Unified Modeling Language)* yang digunakan untuk menggambarkan interaksi antara pengguna (aktor) dan sistem yang akan dibangun. Diagram ini menunjukkan fungsi-fungsi utama yang disediakan sistem dan siapa saja yang dapat mengaksesnya. Seperti dibawah ini ada 2 aktor yaitu, staff dan admin. Staff bisa mengakses hampir semua fungsi yang ada kecuali bagian *user managenet* dan harus login. Sedangkan. Admin bisa mengakses semua fitur.



Gambar 5. Use Case Diagram

3.4.3 Class Diagram

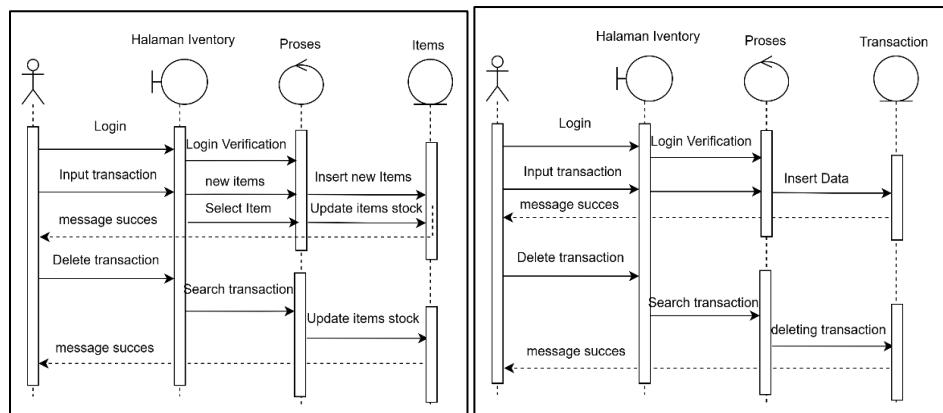
Class Diagram digunakan untuk memodelkan struktur statis dari sistem. Diagram ini menggambarkan kelas-kelas dalam sistem, beserta atribut (data), metode (fungsi), dan relasi antar *class*. *Class* yang dibuat ada sebanyak 4 kelas utama yaitu *users*, *session*, *transaction* dan *items*.



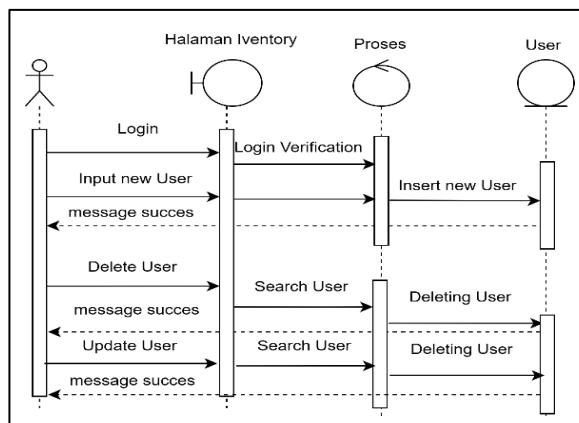
Gambar 6. Class Diagram

3.4.4 Sequence Diagram

Digunakan untuk memodelkan urutan interaksi antar objek dalam suatu skenario atau proses dalam sistem, berdasarkan urutan waktu. Sequence diagram sangat membantu untuk menunjukkan bagaimana objek (atau aktor dan sistem) berkomunikasi melalui pesan-pesan yang dikirimkan dari waktu ke waktu. Pada diagram yaitu **Gambar 7** menjelaskan bagaimana komunikasi aktor dengan *Items* dan *transaction*. Dari mulai login, input transaksi dan barang hingga menghapus. Pada **Gambar 8**, menjelaskan bagaimana aktor yaitu admin membuat akun untuk user lain



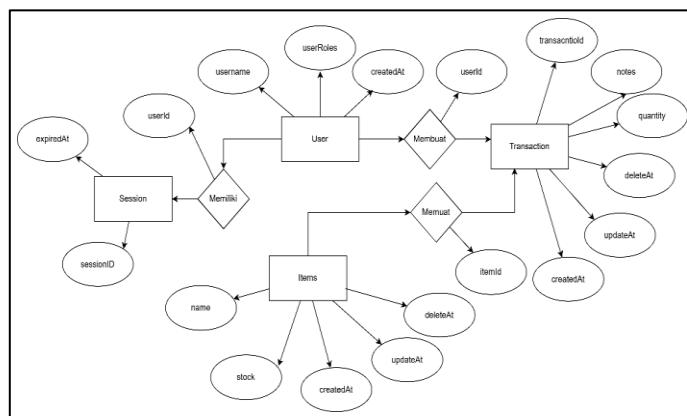
Gambar 7. Sequence Diagram *Items*(barang) dan *transaction*(transaksi)



Gambar 8. Sequence Diagram User

3.4.5 Entity Relationship Diagram

ERD menjelaskan bagaimana entitas seperti *user*, *items*, *transaction* saling terhubung dan membentuk sebuah relasi antar entitas atau tabel. Pada gambar 9 user terhubung dengan 2 entitas yaitu *session* dan *transaction*. Sedangkan *items* hanya terhubung ke *transaction*. Dan pada tiap entitas memiliki atribut masing-masing. Atribut yang merupakan *Primary key* menjadi penghubung antar entitas.



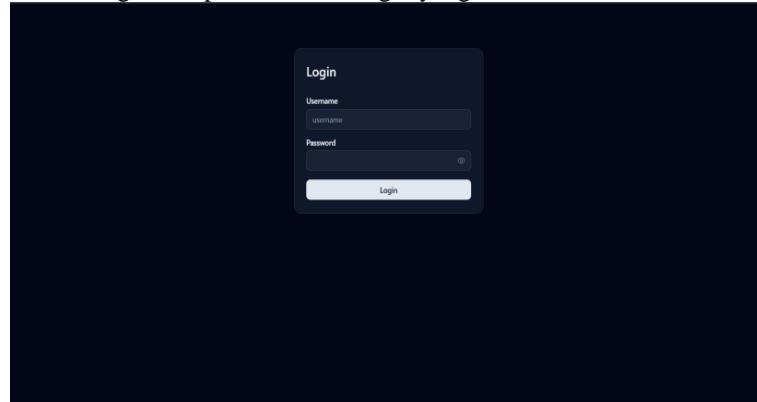
Gambar 9. ERD

3.4 Implementasi Sistem

Pada implementasi *Framework* yang digunakan adalah SvelteKit yang menggunakan teknik SSR (Server Side Rendering) juga. Server-Side Rendering (SSR) adalah pendekatan di mana server menghasilkan konten HTML penuh sebelum dikirim ke browser, menawarkan waktu muat awal yang lebih cepat, kinerja SEO yang lebih baik, dan kompatibilitas lebih luas dengan perangkat dan browser lama, meskipun dapat meningkatkan beban server dan membatasi interaktivitas. Sebaliknya, Client-Side Rendering (CSR) melakukan rendering di browser klien menggunakan JavaScript, memungkinkan interaktivitas yang ditingkatkan, pengalaman pengguna yang lebih mulus setelah muat awal, dan dukungan yang lebih baik untuk antarmuka pengguna yang kompleks, namun bisa mengakibatkan waktu muat awal yang lebih lambat, tantangan SEO, dan masalah aksesibilitas.

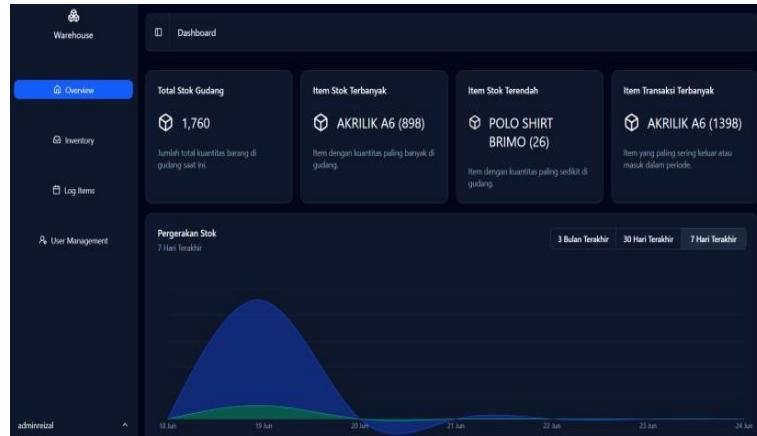
Untuk bagian database menggunakan Postgresql. Lalu authentikasi yang digunakan adalah metode *session based*(bersesi) karena mudah diimplementasi dan sistem belum membutuhkan sistem yang lebih lanjut.

Untuk masuk kedalam aplikasi diperlukan login untuk mendapatkan akses kedalam aplikasi yang dibuat. Berikut gambar pada halaman login yang memerlukan *username* dan *password*.



Gambar 10. Halaman Login

Halaman yang pertama kali muncul setelah login yaitu *Overview* yang berisi ringkasan tentang barang-barang yang tersimpan di aplikasi ini. Halaman ini juga menampilkan visualiasi barang yang masuk dan keluar.



Gambar 11. Halaman beranda *Dashboard*

Halaman *Inventory* berisi list dari item yang disimpan. Dan juga memuat *Form* untuk input transaksi masuk dan keluar.

The screenshot shows the 'Inventory Management' page. The left sidebar includes Overview, Inventory (selected), Log Items, and User Management. The main area has a title 'Inventory Management' with 'Banyak Masuk' and 'Banyak keluar' buttons. Below is a table titled 'this inventory' with columns: No, ID, Name Item, Stock, and Terakhir Diperbaiki. The table lists items like SIGN DOOR, KAOS BRIMO (BABUNG), POLOSHIRT BRIM, POLO SHIRT BRIMO, and AKRILIK A6. At the bottom of the table is a note: 'Daftar 10 Item dengan Stok Terbanyak'.

Gambar 12. Halaman transaksi barang keluar dan masuk

The screenshot shows the 'Inventory Management' page with the 'Banyak Masuk' button selected. A modal dialog titled 'Catat Barang Masuk' is open. It has fields for 'Pilih Item Baru?' (checkbox), 'Pilih Barang yang Ada' (dropdown menu with options 'Pilih barang...', 'Pilih barang...'), 'Jumlah' (input field with value '0'), and 'Catatan (Opcional)' (text area). A note at the bottom says 'Ceklist Stok awal'. At the bottom right of the modal is a 'Simpan Transaksi' button.

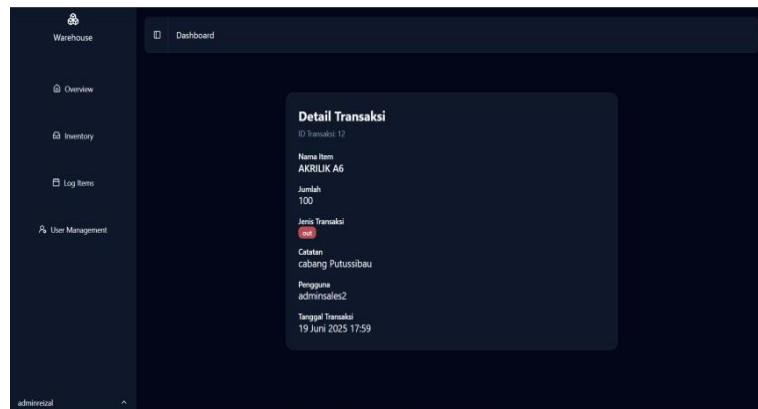
Gambar 12. Form *Input* transaksi

Halaman log-items atau Riwayat, memuat semua catatan transaksi masuk dan keluar serta siapa yang menginput, kapan terjadi dan barang apa yang dilakukan transaksi. Kita juga bisa

menghapus transaksi dan juga melihat detail dari transaksi.

ID	Nama Item	Jumlah	Jenis	Catatan	User	Tanggal	Hapus	Detail
12	AKRILIK A6	100	ada	Ada	adminseal2	19 Juni 2025 17:59	<button>Hapus</button>	<button>Detail</button>
11	AKRILIK A6	50	ada	Tidak Ada	funding1	19 Juni 2025 15:48	<button>Hapus</button>	<button>Detail</button>
10	AKRILIK A6	100	ada	Tidak Ada	funding1	19 Juni 2025 15:47	<button>Hapus</button>	<button>Detail</button>
9	POLO SHIRT BRIMO	10	ada	Tidak Ada	funding1	19 Juni 2025 15:46	<button>Hapus</button>	<button>Detail</button>
8	POLOSHIRT BRIM	36	ada	Tidak Ada	funding1	19 Juni 2025 15:45	<button>Hapus</button>	<button>Detail</button>
6	POLO SHIRT BRIMO	36	ada	Tidak Ada	funding1	19 Juni 2025 11:17	<button>Hapus</button>	<button>Detail</button>
5	KAOS BRIMO (TABUNG)	50	ada	Tidak Ada	funding1	19 Juni 2025 11:16	<button>Hapus</button>	<button>Detail</button>
4	AKRILIK A6	800	ada	Tidak Ada	funding1	19 Juni 2025 08:13	<button>Hapus</button>	<button>Detail</button>
3	SIGN DOOR	750	ada	Tidak Ada	funding1	19 Juni 2025 08:13	<button>Hapus</button>	<button>Detail</button>
1	AKRILIK A6	348	ada	Tidak Ada	funding1	19 Juni 2025 08:11	<button>Hapus</button>	<button>Detail</button>

Gambar 13. Halaman Riwayat transaksi



Gambar 14. Halaman detail transaksi

Halaman *User Management* adalah halaman Dimana hanya admin yang bisa masuk dan di halaman ini admin bisa menghapus, edit dan menambahkan user untuk login kedalam aplikasi ini

Username	Password Baru (optional)	Edit	Hapus
adminseal			

Gambar 15. Halaman *Management User*

4. KESIMPULAN

Berdasarkan perancangan sistem informasi management pergudangan yang telah dibuat pada Kantor Wilayah BRI JAKARTA 3 dengan metode *Waterfall*. Penulis mengambil kesimpulan

bahwa penelitian ini telah menghasilkan output sebuah aplikasi berbasis web yang menggunakan Sveltekit *framework* dari javascript. Yang mana aplikasi ini akan menggantikan metode manual untuk input barang masuk dan keluar pada gudang yang berada di Kantor Wilayah BRI JAKARTA 3.

Aplikasi ini juga berguna untuk meningkatkan efisiensi dan mobilitas karena bisa diakses dimana saja dan kapan saja asal terhubung dengan internet. Aplikasi ini juga mengurangi *human error* yang disebabkan input manual. Dan sistem pada aplikasi ini update secara *realtime* sehingga memudahkan untuk *sharing*, melacak aktivitas untuk audit sehingga akan meningkatkan nilai transparansi.

REFERENCES

- Alif Fikri, M., Rahma Saputri, D., Hoerudin, D., & Saifudin, A. (2024). *BIIKMA : Buletin Ilmiah Ilmu Komputer dan Multimedia Penerapan Model Waterfall Untuk Meningkatkan Kecepatan Dan Fleksibilitas Pengembangan Sistem Inventaris* (Vol. 2, Issue 1). <https://jurnalmahasiswa.com/index.php/biikma>
- Firmansyah, R., Putra Pratama, Mh., & Ronaldo, R. (n.d.). *Perancangan Sistem Inventory Barang Berbasis Website Pada PT.ArForceAsia Dengan Menggunakan Metode Waterfall*. <https://jurnalmahasiswa.com/index.php/biikma>
- Herianto, A. D., Widya Kayohana, K., Ode, L., & Wahid, A. (2023). Pengembangan Sistem Informasi Manajemen Inventory Barang pada Distro ARJ88 Dengan metode pengembangan sistem Waterfall Development of Goods Inventory Management Information System on ARJ88 Distro Using the Waterfall system development method. In *JoMI: Journal of Millennial Informatics* (Vol. 1, Issue 1).
- Jourji Zaidan, A., & Fatrianto Suyatno, D. (2025). Journal of Emerging Information System and Business Intelligence Rendering Performance Analysis of Astro JS, Next JS, Nuxt JS, and SvelteKit Frameworks Using Google Lighthouse, PageSpeed Insight, and JMeter. *Journal of Emerging Information System and Business Intelligence*, 6(1), 1–13.
- Jurnal, S., Manajemen Pergudangan Pada Mitra Tiga Perkasa Indonesia Berbasis Web, A. P., Willianto, J., & Hardian, W. (2021). *JURNAL SISTEM INFORMASI DAN TEKNOLOGI (S I N T E K)*. 1(1). <https://sintek.stmikku.ac.id/index.php/SINTEK>
- Khalimah, S. (2020). Rancang Bangun Sistem Informasi Pergudangan Berbasis Website. *Jurnal Ilmu Komputer Dan Desain Komunikasi Visual*, 5(1).
- Rachma, N., & Muhlas, I. (2022). Comparison Of Waterfall And Prototyping Models In Research And Development (R&D) Methods For Android-Based Learning Application Design. *Jurnal Inovatif : Inovasi Teknologi Informasi Dan Informatika*, 5(1), 36–39. <https://doi.org/10.32832/inovatif>
- Verma, D., & Aland, P. (n.d.). A Comparative Review of Server Rendering and Client Side Rendering in Web Development. In *International Journal of Scientific Research & Engineering Trends* (Vol. 10, Issue 2). <https://benevity.com/why-benevity>