

Implementasi Front End Dan Crud User Website Inventaris Aset IT di Stasiun KCIC

1st Akmal Rizki Zaki

Fakultas Ilmu Terapan

Universitas Telkom

Bandung, Indonesia

akmalzaki@student.telkomuniversity.ac.id

2nd Prajna Deshanta Ibnugraha

Fakultas Ilmu Terapan

Universitas Telkom

Bandung, Indonesia

prajna@tass.telkomuniversity.ac.id

3rd Setia Juli Irzal Ismail

Fakultas Ilmu Terapan

Universitas Telkom

Bandung, Indonesia

julismail@telkomuniversity.ac.id

Pengelolaan inventaris aset IT di Stasiun Kereta Cepat Indonesia China (KCIC) yang masih dilakukan secara manual menimbulkan kendala berupa lamanya waktu pencatatan, kesulitan dalam pencarian informasi, dan rendahnya akurasi data. Penelitian ini bertujuan membangun *website* "Inventory Management System" untuk mengatasi masalah tersebut dengan fokus utamanya mengimplementasikan antarmuka pada *website*. Sistem ini dibangun menggunakan HTML, CSS, *native* JavaScript, dan PHP, serta dilengkapi fitur utama berupa manajemen aset dan fungsionalitas CRUD (*Create, Read, Update, Delete*) untuk pengelolaan pengguna dengan hak akses berbasis peran. Pengujian fungsionalitas dilakukan dengan metode *black-box testing* dengan hasil menunjukkan seluruh fitur seperti validasi *login* dan pembatasan hak akses pengguna berjalan sesuai dengan yang dirancang. *Google Lighthouse* untuk mengukur kinerja web dengan menunjukkan hasil kinerja pada halaman-halaman utama menunjukkan hasil sangat baik, dengan skor performa mencapai 98-100, *Total Blocking Time* (TBT) 0ms, dan *Interaction to Next Paint* (INP) pada interaksi *form* sebesar 60ms. Hasil ini membuktikan bahwa sistem yang diimplementasikan tidak hanya fungsional, tetapi juga memiliki performa yang cepat dan responsif, sehingga dapat menjadi solusi untuk mendukung pengelolaan inventaris aset IT di stasiun KCIC.

Kata kunci: Manajemen Aset IT, *Website Inventaris*, Antarmuka, CRUD, *Google Lighthouse*

I. PENDAHULUAN

Inventory merupakan persediaan barang atau produk mentah, material, atau barang jadi yang disimpan untuk digunakan dalam kurun waktu tertentu[1][2][3]. Dalam hal ini perlu adanya kegiatan inventarisasi. Inventarisasi aset merupakan kegiatan untuk mengontrol data aset atau barang yang diadakan menggunakan anggaran belanja maupun distribusi aset[3][4].

Pengelolaan inventaris yang efisien dan terorganisir menjadi bagian yang penting dalam mendukung operasional[5][6] Stasiun Kereta Cepat Indonesia China. Dengan jumlah aset yang ada dan dengan mobilitas yang bervariasi, diperlukan sistem yang mampu mengelola inventaris secara terpusat[7]. Sistem informasi berbasis *website* menjadi suatu solusi untuk menunjang kebutuhan operasional ini.

Namun, berdasarkan observasi selama magang di Stasiun Kereta Cepat Indonesia China pengelolaan inventaris

IT khususnya di stasiun seperti ketersediaan jumlah gulungan tiket, komputer, dan aset stasiun lainnya masih dicatat secara manual. Hal ini tidak hanya memakan waktu, tetapi juga berpotensi menghambat dalam pencarian informasi ketersediaan aset dan melacak riwayat keluarnya aset[8].

Berdasarkan analisis permasalahan tersebut, penelitian ini mengajukan implementasi sebuah antarmuka *website* untuk menjawab pertanyaan: bagaimana mengimplementasikan antarmuka *website* "Inventory Management System" agar mudah digunakan oleh pengguna, serta bagaimana implementasi fitur CRUD *user* yang mampu mengelola hak akses pengguna. Adapun solusi dari pertanyaan masalah tersebut, yaitu mengimplementasi antarmuka *website* yang memiliki navigasi yang jelas dan responsif agar mudah digunakan dengan menggunakan *native* JavaScript, PHP, dan CSS, dan mengimplementasikan CRUD (*Create, Read, Update, Delete*) *user* untuk mengatur hak akses pada akun.

Sejalan dengan itu, tujuan dari penelitian ini adalah mengimplementasi antarmuka agar mudah digunakan oleh pengguna dan CRUD (*Create, Read, Update, Delete*) *user* untuk manajemen pengguna pada *website* "Inventory Management System". Tidak hanya itu, pengujian dilakukan dalam mengoptimalkan kinerja antarmuka *website* dalam aspek *load time*, responsivitas pada *form* input aset dan fungsionalitas hak akses yang diimplementasi agar sistem yang diimplementasi lebih optimal.

II. KAJIAN TEORI

2.1 Laragon

Laragon adalah sebuah *tool* pengembangan web yang menyajikan lingkungan server lokal yang mudah digunakan. Dengan Laragon, para *developer* dapat membangun server web, mengelola basis data, dan melakukan pengujian web secara *offline*[9].

2.2 Visual Studio Code

Visual Studio Code adalah aplikasi teks editor buatan Microsoft yang tersedia di beberapa sistem operasi seperti Windows, macOS, dan Linux. Teks editor ini mendukung Bahasa pemrograman seperti Javascript, Typescript, dan Node.js. Namun, Visual Studio Code dapat mendukung Bahasa pemrograman lain seperti C++, C#, Java,

Python, dan PHP melalui ekstensi yang disediakan oleh Visual Studio Code[10].

2.3 PHP

Bahasa skrip server-side bernama PHP (*Hypertext Preprocessor*) dirancang khusus untuk pengembangan web. Selain digunakan untuk itu, PHP juga bisa dimanfaatkan sebagai bahasa pemrograman umum. PHP bersifat *open source* dan dapat digunakan secara gratis. Lisensi yang digunakan adalah PHP *License*, yang memiliki perbedaan sedikit dibandingkan dengan GNU *General Public License* (GPL) yang umumnya digunakan pada proyek-proyek open source[11].

2.4 Hypertext Markup Language

Bahasa yang digunakan untuk membangun struktur dan menyusun konten pada halaman web disebut *Hypertext Markup Language* (HTML). Hampir semua situs web pada umumnya dibuat dengan dasar HTML. Melalui HTML, pengembang bisa menentukan berbagai elemen yang menyusun sebuah halaman, mulai dari teks, gambar, video, tautan, hingga komponen lain yang tampil pada antarmuka web[11].

2.5 Cascading Style Sheet

Cascading Style Sheet (CSS) adalah bahasa *stylesheet* yang digunakan untuk memodifikasi tampilan dokumen HTML atau XML di berbagai media, seperti layar, kertas, atau audio. CSS juga dapat mengatur *style font*, jarak, warna, bentuk, dan lain-lain. Dengan adanya CSS mempermudah pengembang untuk mengatur dan memilih tampilan web yang dikembangkan[12].

2.6 JavaScript

JavaScript adalah bahasa pemrograman web yang termasuk *client-side programming language*, yaitu jenis bahasa pemrograman yang berjalan di sisi klien. Klien di sini merujuk pada aplikasi peramban web. Penulisan JavaScript dapat disisipkan langsung dalam dokumen HTML atau dibuat sebagai file terpisah yang dihubungkan dengan dokumen HTML. Bahasa ini memiliki fitur-fitur yang memungkinkan pengaturan interaksi antara halaman web dengan penggunaanya[13].

2.7 Google Lighthouse

Google Lighthouse adalah sebuah alat audit otomatis berbasis open source yang digunakan untuk meningkatkan kualitas sebuah situs web. Alat ini memberikan cara yang jelas bagi para pengembang untuk mengevaluasi performa website. *Google Lighthouse* berjalan secara lokal di peramban *Chrome* dan hasil dari analisa *Google Lighthouse* juga memberikan cara untuk meningkatkan performa website[14].

2.8 Blackbox Testing

Dalam metode *blackbox testing*, pengujian dilakukan dengan berfokus pada spesifikasi fungsional perangkat lunak, di mana tester menentukan kumpulan kondisi input kemudian menguji spesifikasi fungsional program tersebut[15].

pada pengelolaan inventaris aset IT di Stasiun KCIC yang masih dilakukan secara manual. Metode ini mencakup beberapa tahapan, dimulai dari perancangan website, implementasi *website*, hingga pengujian kinerja website menggunakan tool *Google Lighthouse*.

3.1 Gambaran Sistem Saat Ini

Sistem pengelolaan aset IT stasiun KCIC masih dilakukan secara manual. Proses ini melibatkan:

1. Petugas memberitahu lewat pesan grup ketika ada barang yang masuk dan pencatatan jumlah dilakukan secara manual.
2. Jumlah barang yang keluar tidak tercatat.
3. Petugas menghitung manual terhadap ketersediaan barang.

3.2 Pengembangan Sistem

Sistem yang diimplementasi akan terdiri dari beberapa fitur seperti tampilan *dashboard* yang menampilkan data jumlah aset, tampilan menu pelaporan yang di dalamnya terdapat pilihan tombol opsi format unduhan berupa excel atau pdf, tampilan menu aset yang didalamnya terdapat tampilan menu-menu seperti penambahan aset, melihat data aset, pengambilan aset, melihat data aset yang diambil yang telah di kategorikan setiap bulan, melihat riwayat aktivitas seperti penambahan ataupun pengurangan aset, menu *user management* yang berfungsi untuk menambahkan pengguna baru, menghapus pengguna, dan mengedit pengguna. Fitur CRUD (*Create, Read, Update, Delete*) user yang diimplementasikan untuk mengatur hak akses pengguna.

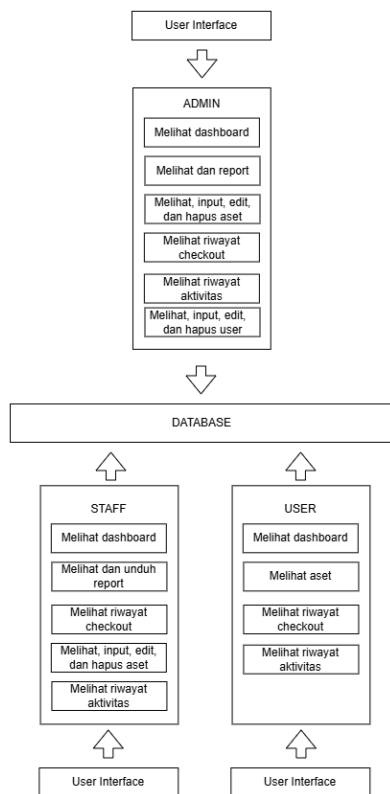
3.3 Tahapan Pengembangan Sistem

Gambaran umum website

Secara umum CRUD (*Create, Read, Update, Delete*) user terkait hak akses pengguna pada *website* "Inventory Management System" akan diimplementasikan seperti gambar berikut:

III. METODE

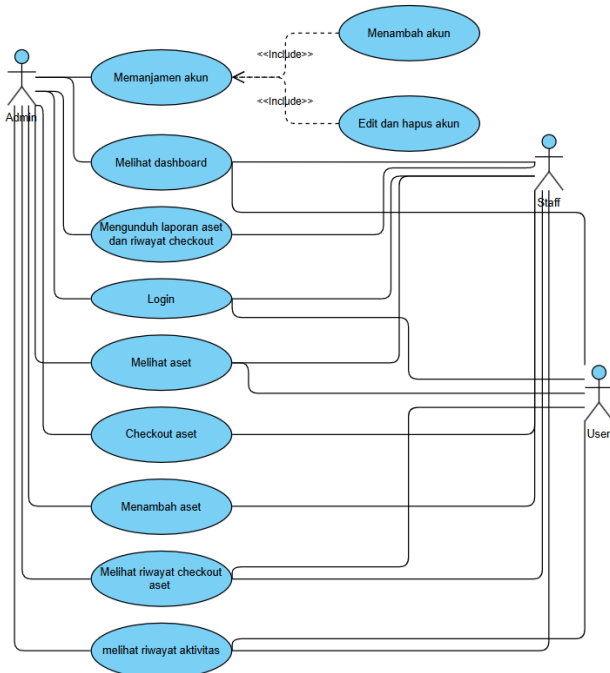
Penelitian ini menguraikan metode pengembangan sistem yang dirancang untuk mengatasi ketidakefisienan



GAMBAR 1
Perancangan Perancangan Website

2. Use case diagram

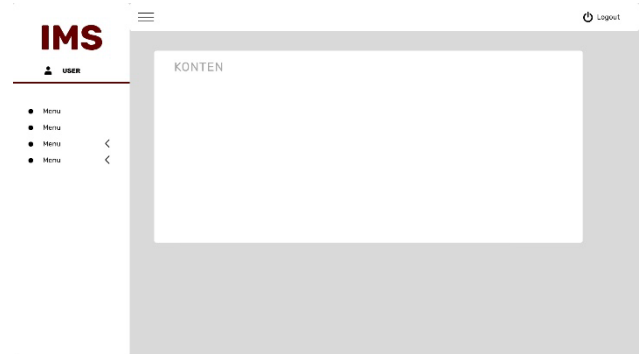
Use case diagram merupakan diagram yang menjelaskan hubungan interaksi antara aktor dengan sistem. Pada *use case diagram* sistem ini menggambarkan hubungan interaksi antara *role* akun yang pada website. *Role* akun yang terdapat pada *website* ini ada tiga, yaitu admin, staff, user.



GAMBAR 2
Usecase Diagram

3. Tampilan website

Tampilan *website* ini diimplementasikan dari figma dan tanpa menggunakan *framework*. Didalam desain tampilan *website* terdapat beberapa halaman utama *website*:



GAMBAR 3
Tampilan Layout Website

Pada *layout website* tersebut terdapat *sidebar* yang berisi menu-menu pada *website*. Pada bagian tengah berisi konten dari menu-menu tersebut.

3.4 Metode Pengujian

Untuk memastikan sistem yang diimplementasi memenuhi kriteria kualitas akan dilakukan beberapa skenario pengujian. Pengujian ini untuk mengukur performa, fungsionalitas hak akses, dan manajemen pengguna *website*. Serangkaian pengujian tersebut dilakukan dengan tiga skenario pengujian sebagai berikut:

1. Pengujian Load Time

Pengujian ini untuk mengukur seberapa lama halaman dimuat yang dipengaruhi ukuran gambar dan elemen yang diimplementasikan pada antarmuka *website*. Pengujian ini menggunakan *tool Google Lighthouse* dengan mode *navigation*. Pada mode navigasi ini terdapat metrik pengukuran yang di uji, yaitu waktu muat halaman (*Largest Contentful Paint & First Contentful Paint*), dan responsivitas awal (*Total Blocking Time*).

2. Pengujian Responsivitas pada Form Tambah Aset

Pengujian ini mengukur responsivitas saat pengguna melakukan input data pada form tambah aset seperti mengetik dan upload gambar. Pengujian ini menggunakan *tool Google Lighthouse* dengan mode *timespan*. Pada mode *timespan* ini terdapat metrik pengukuran yang diuji, yaitu responsivitas awal ketika halaman sepenuhnya sudah dimuat (*Total Blocking Time*), dan responsivitas halaman (*Interaction to Next Paint*).

3. Pengujian Fungsionalitas

Pengujian ini dilakukan untuk memastikan fitur CRUD (*Create, Read, Update, Delete*) user dan navigasi antarmuka *website* dapat berfungsi sebagai mana mestinya. Metode pengujian ini menggunakan metode *blackbox testing*. Dalam pengetesan ini dilakukan berinteraksi terhadap antarmuka *website* tanpa melihat struktur kode. Pengujian ini mencakup validasi *login* dan interaksi terhadap menu serta navigasi antarmuka *website*.

IV. HASIL DAN PEMBAHASAN

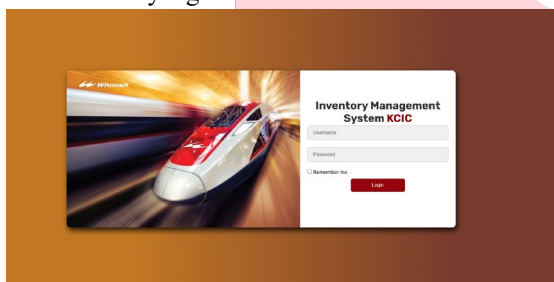
Bagian ini menjelaskan mengenai hasil implementasi sistem yang bertujuan menjawab permasalahan yang saat ini dihadapi, khususnya dalam pengelolaan aset IT di stasiun KCIC.

4.1 Hasil Akhir (Luaran)

Antarmuka berhasil diimplementasikan dan fitur CRUD *user* untuk memberikan hak akses terhadap tiap jenis akun telah berhasil diimplementasikan.

1. Halaman Login

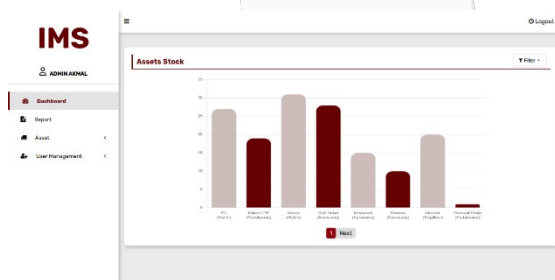
Halaman *login* diimplementasikan dengan antarmuka yang minimalis untuk memberikan kemudahan pengguna. Implementasi antarmuka pada halaman ini mencakup *form* input untuk validasi data yang valid dan tidak valid.



GAMBAR 4
Halaman Login

2. Dashboard Website

Halaman *dashboard* menampilkan ringkasan data inventaris. Implementasi pada halaman ini adalah menampilkan grafik sebagai informasi visual. Grafik ini menampilkan informasi, yaitu data jumlah total aset yang terdaftar di dalam sistem yang disertai filter. Implementasi tampilan ini bertujuan untuk mempermudah dalam melacak ketersediaan aset.



GAMBAR 5
Halaman Dashboard

3. Manajemen aset

Implementasi antarmuka pada bagian ini terdapat berbagai menu untuk melakukan penginputan aset, melihat jumlah dan jenis aset, mengambil aset, dan menampilkan riwayat aset yang diambil. Menu *add asset* berisi *form* yang digunakan untuk menambahkan aset. Pada menu "View Assets" menampilkan jumlah dan jenis aset dalam bentuk tabel dan terdapat *pagination* untuk menampilkan jumlah data yang ditampilkan disetiap halamannya. Pada halaman ini juga dilengkapi dengan elemen

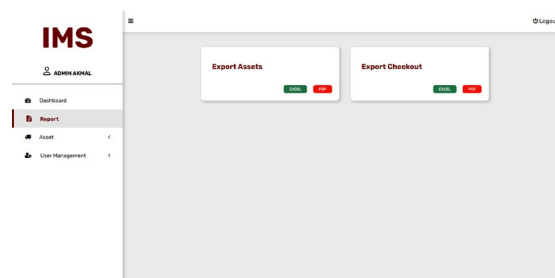
edit yang berfungsi untuk memodifikasi data aset. Terakhir, halaman *checkout* dan riwayat *checkout* aset. Halaman ini menampilkan aset yang keluar langsung tercatat dan ditampilkan pada sistem.

ID	ASSET NAME	IMAGE	LOCATION	ASSET TYPE	ASSET INFO	STOCK	CREATED BY	CREATED AT	UPDATED AT	EDIT	DELETE
1	PC		Lab	new moving	PC Office	27	Admin Admin	Oct 25, 2024	Jul 11, 2025		
2	Kasut LTR		Madurejo	new moving	Jumlah - besar	10	Admin Admin	Oct 25, 2024	Jul 11, 2025		
3	Mouse		Lab	new moving	Mouse office	31	Admin Admin	Oct 25, 2024	Jul 11, 2025		
4	Roll Tiket		Kecoaan	Roll moving	Roll Tiket TMS dan UAT	28	Admin Admin	Nov 12, 2024	Jul 05, 2025		
5	Keyboard		Kecoaan	new moving	23123	15	Admin Admin	Jan 10, 2025	Jul 05, 2025		

GAMBAR 6
Menu dan Halaman Manajemen

4. Menu Report

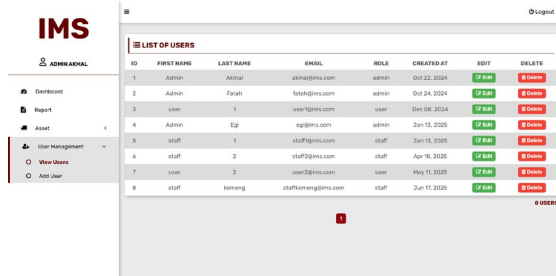
Menu *report* diimplementasikan untuk pengguna dengan hak akses Admin dan *Staff*, dalam mengelola aset inventaris. Melalui halaman ini, pengguna dapat mengunduh laporan yang mencakup data keseluruhan aset yang ada serta riwayat aset yang keluar (*checkout*). Didalam menu ini terdapat opsi format unduhan excel dan PDF.



GAMBAR 7
Halaman Report

5. Manajemen User

Menu manajemen pengguna hanya dapat diakses oleh admin. Fitur ini berfungsi untuk mengontrol akses pengguna pada *website*. Tampilan *view user* disajikan tabel yang menampilkan daftar seluruh pengguna yang terdaftar. Pada tampilan ini juga terdapat tombol edit dan *delete* untuk memodifikasi pengguna dalam sistem. Pada menu *add user* berfungsi untuk menambahkan pengguna baru kedalam *website*. Di dalam *form* tersebut terdapat opsi memilih *role* (admin, *staff*, *user*). Pilihan *role* ini memberikan hak akses pengguna. Untuk memodifikasi data pengguna yang terdaftar dapat tersedia tombol edit yang dapat untuk melakukan perubahan data nama, email, dan peran.



GAMBAR 8

Menu dan Halaman Manajemen Pengguna

4.2 Pengujian Luaran

Pengujian luaran dilakukan untuk memastikan bahwa antarmuka yang diimplementasi mampu memenuhi tujuan utama.

1. Pengujian Load Time

Pengujian dilakukan menggunakan tool *Google Lighthouse* menggunakan mode *navigation*, pengujian ini mengukur performa loading pada setiap halaman menu utama. Parameter yang terdapat pada *Lighthouse* seperti waktu muat halaman (LCP, FCP), responsivitas awal (TBT), dan stabilitas visual (CLS)[16].

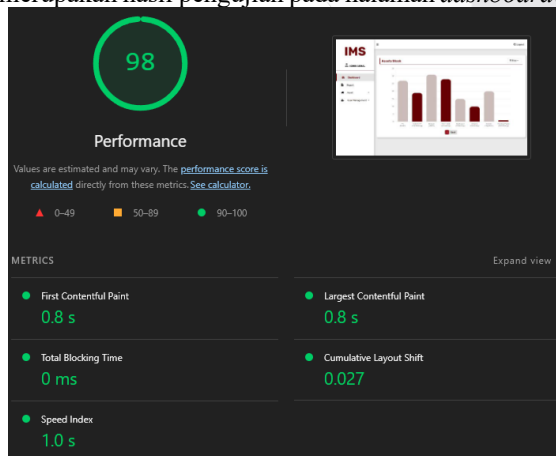
Parameter nilai pengujian *Google Lighthouse* terdapat pada tabel berikut:

TABEL 1

Parameter dan Pengkategorian Nilai pada Google Lighthouse

Parameter	Kategori	Waktu Rendering
First Contentful Paint	Cepat (Hijau)	0s – 0.9s
	Sedang (Orange)	0.9s – 1.6s
	Lambat (Merah)	> 1.6s
Largest Contentful Paint	Cepat (Hijau)	0s – 1.2s
	Sedang (Orange)	1.2s – 2.4s
	Lambat (Merah)	> 2.4s
Speed Index	Cepat (Hijau)	0s – 1.3s
	Sedang (Orange)	1.3s – 2.3s
	Lambat (Merah)	> 2.3s
Total Blocking Time	Cepat (Hijau)	0ms – 150ms
	Sedang (Orange)	150ms – 350ms
	Lambat (Merah)	> 350ms
Cumulative Layout Shift	Cepat (Hijau)	0 - 0.1 %
	Sedang (Orange)	0.2 - 0.25 %
	Lambat (Merah)	> 0.25 %
Score Performance	Baik (Hijau)	90 – 100
	Sedang (Orange)	50 – 89
	Buruk (Merah)	0 – 49

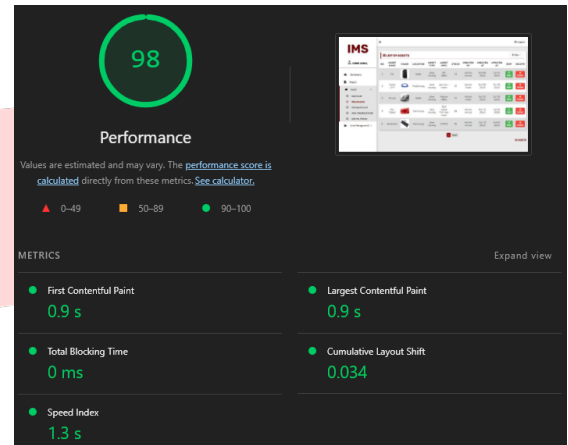
Berikut merupakan hasil pengujian pada halaman *dashboard*:



GAMBAR 9

Hasil Pengujian Halaman dashboard

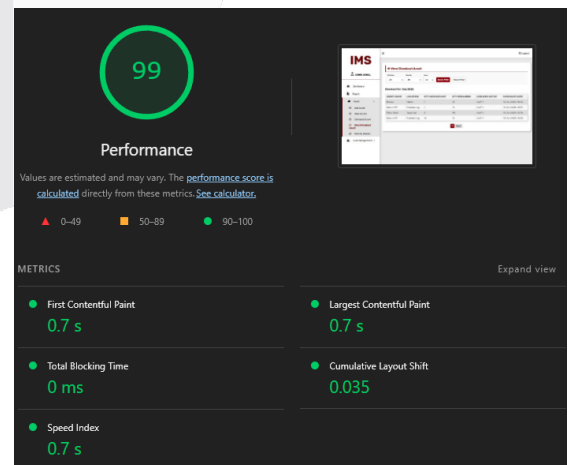
Pengujian pada halaman *dashboard* menggunakan mode *navigation* menunjukkan skor performa 98. Pada *Total Blocking Time* (TBT) dihasilkan angka sebesar 0 ms. Waktu menampilkan konten pertama muncul (FCP) tercatat 0,8s. Waktu *loading* konten terbesar (LCP) tercatat selama 0,8 detik. Stabilitas visual halaman ini juga masuk dalam kategori cepat (hijau) dengan skor CLS 0,027 dan *speed index* mencatatkan waktu 1 detik. Pada pengujian ini menunjukkan bahwa halaman *dashboard* memiliki performa yang baik dalam aspek *load time*.



GAMBAR 10

Hasil Pengujian Halaman View Assets

Pada gambar 1.10 merupakan pengujian pada halaman "View Asset" menunjukkan skor performa 98. Pengujian ini menggunakan mode *navigation*. Pada pengujian halaman ini mencatatkan waktu *loading* konten pertamanya (FCP) dalam waktu 0,9s, waktu *loading* konten terbesar (LCP) dalam waktu 0,9 detik, dan mencatatkan TBT sebesar 0 ms. Hasil ini membuktikan bahwa halaman ini memiliki performa yang baik dalam aspek *load time*. Dengan skor CLS 0,034, tata letak tabel stabil saat *loading* serta *speed index* 1,3 detik.

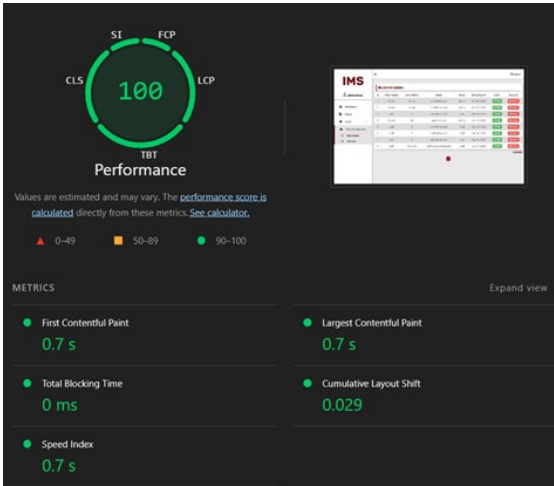


GAMBAR 11

Hasil Pengujian Halaman View Checkout Assets

Pada gambar 1.11 merupakan pengujian pada halaman "View Checkout Asset" menggunakan mode

navigation menunjukkan skor performa 99. Mencatatkan *loading* konten pertamanya (FCP) dalam waktu 0,7s, diikuti dengan hasil LCP dan TBT tercatat dalam waktu 0,7 detik dan 0 ms. Skor CLS pada halaman ini adalah 0,035 serta hasil dari *speed index* tercatat 0,7 detik. Hasil pengujian ini menunjukkan bahwa halaman ini memiliki performa yang baik dalam aspek *load time*.

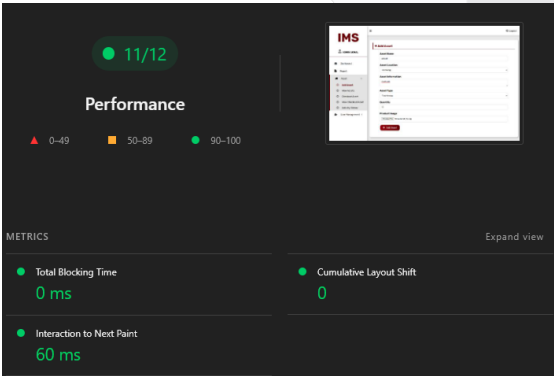


GAMBAR 12
Hasil Pengujian Halaman View Users

Pada gambar 1.12 merupakan pengujian pada halaman "View Users" menggunakan mode *navigation*. Hasil pengujian ini meraih Skor Performa 100. Mencatatkan *loading* konten pertamanya (FCP) dalam waktu 0,7s, diikuti dengan hasil LCP dan TBT tercatat dalam waktu 0,7 detik, TBT 0 ms, dan CLS 0.0029, serta mencatatkan hasil *speed index* 0,7 detik. Hasil pengujian menunjukkan bahwa performa halaman ini baik dalam aspek *load time*.

2. Pengujian Resposivitas

Pengujian ini menggunakan *tool Lighthouse* dengan mode *Timespan*, pengujian ini mengukur kelancaran interaksi yang tidak menyebabkan *refresh* halaman, seperti saat mengisi form tambah aset. Parameter yang diukur adalah TBT, INP, dan CLS selama interaksi berlangsung[16]. Berikut merupakan hasil pengujian pada *halaman add asset*:



GAMBAR 13
Hasil Pengujian Form Add Asset

Berdasarkan pengujian dengan mode *Timespan* pada interaksi pengisian *form* pada menu *add asset* seperti mengisi

form dan *upload* gambar dengan menggunakan *Lighthouse*. Pengujian ini menghasilkan metrik *Total Blocking Time* (TBT), yang mengukur total waktu *delay* yang dialami pengguna sebelum halaman dapat merespons kembali[18], yaitu 0 ms. *Interaction to Next Paint* (INP) menunjukkan hasil 60ms yang berarti responsivitas *form* tersebut dalam mengisi *form* dan *upload* gambar masuk kedalam kategori cepat[19][20][21]. *Cumulative Layout Shift* (CLS) menghasilkan skor 0, yang berarti tidak ada pergeseran tampilan yang mengganggu saat pengguna mengisi *form* dan *upload* gambar[22]. Hasil pengujian ini membuktikan bahwa pada aspek interaksi saat mengisi *form* dan *upload* gambar memiliki performa yang baik.

3. Pengujian Fungsionalitas

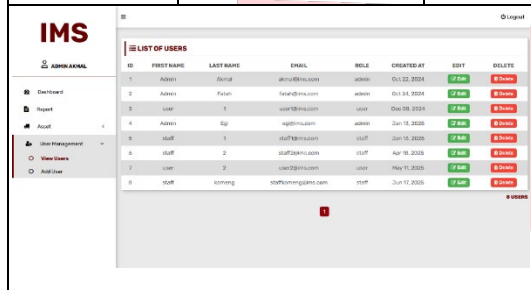
Metode pengujian fungsionalitas ini dilakukan dengan metode *blackbox testing*. Pengujian fungsionalitas ini membuktikan kesesuaian fungsi *website*. Pengujian ini dilakukan dengan menguji fungsi validasi *form login* dan hak akses pengguna. Hasil pengujian fungsionalitas diuraikan dalam tabel berikut:

TABEL 2
Pengujian Login

Pengujian	Hasil yang diharapkan	Hasil
Menguji validasi <i>form login</i> dengan memasukkan data yang invalid ketika <i>login</i>	<i>Website</i> memvalidasi bahwa <i>username</i> atau <i>password</i> yang di input salah yang muncul pada bagian bawah tombol login dengan pesan: "Error: Please make sure that <i>username</i> and <i>password</i> are correct".	Sesuai

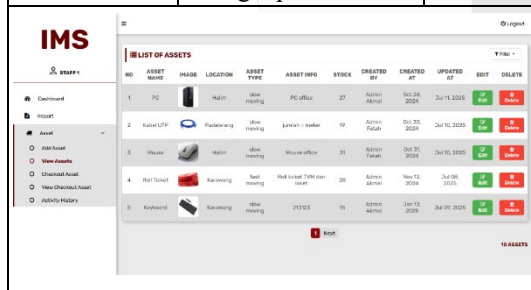
TABEL 3
Login sebagai Admin

Pengujian	Hasil yang diharapkan	Hasil
Menguji tampilan menu pada pengguna yang memiliki hak akses admin	Tampilan menu pada akun admin terdapat menu <i>dashboard</i> , <i>report</i> , <i>asset</i> , dan <i>user management</i>	Sesuai



TABEL 4
Login sebagai Staff

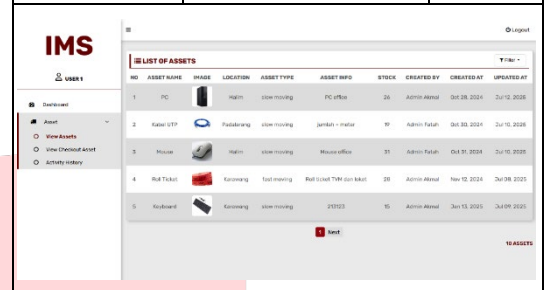
Pengujian	Hasil yang diharapkan	Hasil
Menguji tampilan menu pada pengguna yang memiliki hak akses staff	Tampilan menu pada akun <i>staff</i> terdapat menu <i>dashboard</i> , <i>report</i> , dan <i>asset</i> yang dapat memodifikasi, menambahkan dan menghapus.	Sesuai



TABEL 5
Login sebagai User

Pengujian	Hasil yang diharapkan	Hasil
Menguji tampilan menu pada pengguna	Tampilan menu pada akun <i>user</i> hanya terdapat menu <i>dashboard</i> ,	Sesuai

yang memiliki hak akses user dan mengunjungi menu View Assets	dan pada menu <i>asset</i> hanya terdapat <i>view asset</i> , <i>view checkout asset</i> , dan <i>activity history</i> . Pada halaman <i>view asset</i> tidak bisa melakukan edit maupun hapus aset.	
---	--	--



Semua pengujian yang dilakukan membuktikan bahwa implementasi antarmuka mencatatkan hasil yang baik. Semua pengujian fungsionalitas berhasil memenuhi kriteria dan tanpa kendala. Hal ini membuktikan bahwa *website* telah memenuhi kriteria kualitas sistem.

4.3 Analisa Hasil Pengujian

Hasil pengujian menunjukkan bahwa implementasi antarmuka serta fitur CRUD *user* pada *website* "Inventory Management System" telah berhasil memenuhi seluruh kriteria kualitas, baik dari aspek fungsionalitas maupun kinerja teknis. Antarmuka sistem terbukti memiliki performa yang optimal melalui pengujian kinerja dengan *Google Lighthouse*, yang menghasilkan skor 98–100 serta *Total Blocking Time* (TBT) sebesar 0 ms, menunjukkan waktu muat yang cepat dan tingkat responsivitas yang tinggi. Dari sisi fungsionalitas, seluruh fitur termasuk validasi *login* dan sistem hak akses berbasis peran divalidasi melalui pengujian *black-box* dan terbukti berjalan sesuai rancangan tanpa kesalahan. Gabungan antara fungsionalitas yang teruji dan performa yang terukur ini membuktikan bahwa *website* yang dikembangkan adalah solusi yang layak untuk digunakan.

4.4 Keterbatasan Sistem

Walaupun tujuan fungsional dan kinerja dari sistem yang dikembangkan telah tercapai, masih ditemukan beberapa keterbatasan. Pertama, desain antarmuka pengguna saat ini belum responsif, sehingga pengalaman dan tampilan pengguna menjadi kurang optimal saat diakses melalui perangkat dengan layar kecil seperti ponsel atau tablet. Kedua, fungsionalitas pemindaian *barcode* belum terintegrasi ke dalam sistem, sehingga proses pelacakan dan identifikasi aset masih dilakukan secara manual dan belum mencapai efisiensi maksimal. Kedua hal tersebut merupakan aspek penting yang dapat ditingkatkan lebih lanjut di masa mendatang.

V. KESIMPULAN

Antarmuka dan fungsionalitas CRUD (Create, Read, Update, Delete) *user* pada *website* "Inventory Management System" berhasil diimplementasikan melalui penelitian ini. Sistem yang dikembangkan dilengkapi dengan menu utama

seperti *Dashboard*, Manajemen Aset, Laporan yang dapat diunduh, serta Manajemen Pengguna dengan hak akses berbasis peran, yang semuanya dirancang untuk mendukung proses kerja yang lebih teratur dan efisien. Untuk memastikan kesesuaiannya dengan rancangan, telah dilakukan serangkaian pengujian kinerja dan fungsionalitas yang menunjukkan hasil sesuai ekspektasi. Dari aspek fungsional, hasil pengujian membuktikan bahwa seluruh tombol navigasi, fitur CRUD untuk manajemen pengguna, serta sistem hak akses berbasis peran berfungsi secara optimal. Sementara itu, pengujian performa menggunakan *Lighthouse* pada navigasi menu utama memperlihatkan kinerja yang baik, dengan skor performa pada seluruh halaman mencapai 98 hingga 100 serta Total Blocking Time (TBT) sebesar 0 ms. Selain itu, pengujian pada interaksi pengisian formulir juga menunjukkan hasil positif dengan TBT 0 ms dan INP 30 ms, yang mengindikasikan bahwa interaksi di dalam halaman berjalan sangat lancar.

REFERENSI

- [1] S. Aji and D. Pratmanto, "SISTEM INFORMASI INVENTORY BARANG MENGGUNAKAN METODE WATERFALL," 2021. [Online]. Available: <http://ejournal.bsi.ac.id/ejurnal/index.php/ijse>
- [2] L. Nurlaela, A. Dharmalau, D. Nong, and T. Parida, "RANCANGAN SISTEM INFORMASI INVENTORY BARANG BERBASIS WEB STUDI KASUS PADA CV. LIMOPLAST," vol. 2, no. 5, 2020.
- [3] M. Usnaini, V. Yasin, and A. Z. Sianipar, "Perancangan sistem informasi inventarisasi aset berbasis web menggunakan metode waterfall," *Jurnal Manajemen Informatika Jayakarta*, vol. 1, no. 1, p. 36, Feb. 2021, doi: 10.52362/jmijayakarta.v1i1.415.
- [4] T. A. Kinawara, N. R. Hidayati, and F. Nugrahanti, "Rancang Bangun Aplikasi Inventaris Berbasis Website pada Kelurahan Bantengan," 2019.
- [5] J. S. Pasaribu, "PERANCANGAN SISTEM INFORMASI BERBASIS WEB PENGELOLAAN INVENTARIS ASET KANTOR DI PT. MPM FINANCE BANDUNG," 2021.
- [6] J. S. Pasaribu, "Development of a Web Based Inventory Information System," *International Journal of Engineering, Science and Information Technology*, vol. 1, no. 2, pp. 24–31, Mar. 2021, doi: 10.52088/ijesty.v1i2.51.
- [7] A. Ferry Qadafi and A. D. Wahyudi, "SISTEM INFORMASI INVENTORY GUDANG DALAM KETERSEDIAAN STOK BARANG MENGGUNAKAN METODE BUFFER STOK," *Jurnal Informatika dan Rekayasa Perangkat Lunak (JATIKA)*, vol. 1, no. 2, pp. 174–182, 2020, [Online]. Available: <http://jim.teknokrat.ac.id/index.php/informatika>
- [8] N. L. De and E. Wahyudiari, "SISTEM INFORMASI INVENTORY BERBASIS WEB PADA CV BALI BATIK," 2019.
- [9] H. Kholid Kz, M. Syahfiar, S. Jana, Muh. Fikri Abhyasa, and A. Firmansyah, "Pembuatan Website BUMDes Berbasis Opensid di Desa Wano untuk Meningkatkan Pengetahuan Masyarakat," 2025.
- [10] J. Sains *et al.*, "APLIKASI BUKU TAMU MENGGUNAKAN FITUR KAMERA DAN AJAX BERBASIS WEBSITE PADA KANTOR DISPORA KOTA MEDAN," Dec. 2022.
- [11] R. Noviana, "PEMBUATAN APLIKASI PENJUALAN BERBASIS WEB MONJA STORE MENGGUNAKAN PHP DAN MYSQL," *JTS*, vol. 1, no. 2, Jun. 2022.
- [12] F. Ahmad, "CASCADING STYLE SHEET," 2021.
- [13] I. P. Sari, A. Jannah, A. M. Meuraxa, A. Syahfitri, and R. Omar, "Perancangan Sistem Informasi Penginputan Database Mahasiswa Berbasis Web," *Hello World Jurnal Ilmu Komputer*, vol. 1, no. 2, pp. 106–110, Jul. 2022, doi: 10.56211/helloworld.v1i2.57.
- [14] A. Adli, T. Wihayanti, and D. Witasryah, "Top 5 E-Commerce Performance Analysis Using Google Lighthouse Matrix," 2023, pp. 239–248. doi: 10.2991/978-94-6463-340-5_21.
- [15] J. Shadiq, A. Safei, R. Wahyudin Ratu Loly, C. sitasi, L. Rwr, and P. Aplikasi Peminjaman Kendaraan Operasional Kantor Menggunakan BlackBox Testing, "INFORMATION MANAGEMENT FOR EDUCATORS AND PROFESSIONALS Pengujian Aplikasi Peminjaman Kendaraan Operasional Kantor Menggunakan BlackBox Testing," *Information Management for Educators and Professionals*, vol. 5, no. 2, pp. 97–110, 2021.
- [16] B. Kenny, "Lighthouse user flows," <https://web.dev/articles/lighthouse-user-flows>. Accessed: Jul. 03, 2025. [Online]. Available: <https://web.dev/articles/lighthouse-user-flows>
- [17] GOOGLE and Chrome For Developers, "METRIK DAN PERFORMA PADA GOOGLE LIGHTHOUSE," <https://developer.chrome.com/docs/lighthouse/performance/performance-scoring>. Accessed: Jun. 18, 2025. [Online]. Available: <https://developer.chrome.com/docs/lighthouse/performance/performance-scoring>
- [18] R. Susanti and N. Cahyono, "ANALISIS DAN PERBANDINGAN PERFORMA WEBSITE PENERIMAAN MAHASISWA BARU PERGURUAN TINGGI SWASTA YOGYAKARTA," 2024.
- [19] Z. Đuric, "Comparative Analysis of Server-Side Rendering: React vs. Next.js," Mar. 2025.
- [20] E. Kroon Celander and A. Möllestål, "A Comparative Analysis of Next.js, SvelteKit, and Astro for E-commerce Web Development," 2024.
- [21] G. Bolén, "What Influences Rendering Performance in React Applications?," 2025.
- [22] S. Fakhrunnisa and R. Kurniawan, "Perbandingan Kinerja Framework Front End Dalam Pengembangan Platform Talent Pool," *Technologia : Jurnal Ilmiah*, vol. 15, no. 4, p. 966, Oct. 2024, doi: 10.31602/tji.v15i4.16750.