Meningkatkan User Experience Aplikasi GIS Melalui Penerapan Quality Assurance

Megah Juliardi Sondara Wicaksana[1], Mohammad Nurkamal Fauzan[2]

D4 Teknik Informatika, Sekolah Vokasi Universitas Logistik dan Bisnis Internasional[1]
email: 1214066@std.ulbi.ac.id
D4 Teknik Informatika, Sekolah Vokasi Universitas Logistik dan Bisnis Internasional[2]
email: m.nurkamal.f@ulbi.ac.id

Abstrak

Aplikasi web bangunan bersejarah di Singapura menghadapi tantangan dalam kompatibilitas dan performa lintas platform. Untuk mengatasi hal ini, penelitian ini menggunakan pendekatan mass testing dan Lighthouse guna mengevaluasi performa aplikasi di berbagai sistem operasi dan browser. Mass testing dilakukan untuk mengidentifikasi kendala kompatibilitas dan stabilitas aplikasi saat diakses oleh banyak pengguna secara bersamaan. Lighthouse digunakan untuk menganalisis aspek performa, aksesibilitas, praktik terbaik, dan SEO. Hasil pengujian menunjukkan bahwa 60% skenario mengalami kendala, terutama pada Chrome (Android) dan Edge (Windows), sementara Safari di iOS menunjukkan kompatibilitas terbaik. Skor performa Lighthouse lebih rendah pada Chrome dan Edge dibandingkan Firefox. Rekomendasi perbaikan meliputi optimasi kode, penyesuaian rasio aspek gambar, serta pengujian lintas platform guna meningkatkan kualitas dan pengalaman pengguna aplikasi.
Kata Kunci: Aplikasi web, *Quality Assurance*, Performa, Pengujian, Kompatibilitas

Abstract

Historic building web applications in Singapore face challenges in cross-platform compatibility and performance. To address this, this research utilizes mass testing and Lighthouse approaches to evaluate application performance across different operating systems and browsers. Mass testing was conducted to identify compatibility and stability issues when accessed by multiple users simultaneously. Lighthouse was used to analyze aspects of performance, accessibility, best practices, and SEO. The test results showed that 60% of the scenarios had issues, mainly on Chrome (Android) and Edge (Windows), while Safari on iOS showed the best compatibility. Lighthouse performance scores were lower on Chrome and Edge than Firefox. Recommended improvements include code optimization, image aspect ratio adjustment, and cross-platform testing to improve the quality and user experience of the app.

Keywords: *Web application, Quality, Performance, Testing, Compatibility*

1. PENDAHULUAN

Dalam industri perangkat lunak, kualitas produk menentukan keberhasilan. Studi menunjukkan bahwa perangkat lunak berkualitas meningkatkan kepuasan pelanggan dan daya saing. Perangkat lunak yang baik juga mengurangi biaya pemeliharaan dengan meminimalisir kesalahan. Software Quality Assurance (SQA) berperan penting dalam memastikan standar kualitas terpenuhi sepanjang siklus pengembangan perangkat lunak.

Salah satu komponen penting SQA adalah inspeksi, yaitu evaluasi artefak perangkat lunak untuk mengidentifikasi cacat sejak dini. Inspeksi berbeda dari pengujian karena berfokus pada pemeriksaan sistematis artefak oleh tim yang terdiri dari berbagai ahli. Pendekatan ini meningkatkan deteksi cacat serta transfer pengetahuan dalam tim.

Meski mampu mendeteksi 50-90% masalah persyaratan, inspeksi sering kurang dimanfaatkan akibat tantangan otomatisasi. Namun, pendekatan sistematis dapat meningkatkan keandalan perangkat lunak. Selain inspeksi, model pengujian hybrid yang menggabungkan manual dan otomatisasi dapat meningkatkan efektivitas pengujian.
Penerapan standar internasional seperti ISO/IEC 25010 membantu memastikan kualitas perangkat lunak dalam aspek karakteristik *Functional Suitability*, *Performance Efficiency*, *Compatibility*, *Interaction Capability*, *Reliability*, *Security*, *Maintainability*, dan *Flexibility*. Penelitian ini mengevaluasi performa aplikasi GIS bangunan bersejarah di Singapura melalui pengujian Lighthouse dan mass testing untuk memperoleh gambaran menyeluruh.

Penelitian ini berfokus pada penilaian performa aplikasi bangunan bersejarah di Singapura. Aplikasi ini berbasis Geographic Information System (GIS) dan ditampilkan dalam bentuk peta. Penilaian performa dilakukan melalui berbagai metode pengujian untuk mendapatkan gambaran komprehensif tentang kinerja aplikasi. Metode-metode tersebut meliputi:

1. Pengujian lighthouse yang digunakan untuk mengukur kualitas halaman web, termasuk performa, aksesibilitas, praktik terbaik, dan SEO.
2. Pengujian masal (*Mass Test*) yang dilakukan untuk mengevaluasi performa aplikasi dalam menangani sejumlah besar data atau pengguna secara bersamaan.

Pengujian masal (*mass test*) dilakukan dengan mensimulasikan akses bersamaan oleh banyak pengguna untuk mengidentifikasi potensi masalah performa seperti *bottleneck*, waktu respons yang lambat, dan kegagalan sistem. Data yang dikumpulkan selama pengujian masal akan dianalisis untuk mengukur skalabilitas dan stabilitas aplikasi dalam menangani lonjakan trafik. Hasil pengujian ini diharapkan dapat memberikan informasi berharga bagi tim pengembang untuk mengoptimalkan performa aplikasi dan memastikan pengalaman pengguna yang positif, bahkan ketika diakses oleh banyak pengguna secara bersamaan.

Melalui kombinasi pengujian *Lighthouse* dan *mass test*, penelitian ini berupaya untuk mengumpulkan data performa yang komprehensif dan memberikan rekomendasi yang terukur kepada tim pengembang. Tujuan akhir dari penelitian ini adalah untuk meningkatkan performa dan pengalaman pengguna (User Experience) pada aplikasi yang sedang dikembangkan.

1.1 *Research question*

Sebelum melakukan penyusunan jurnal ini, penulis memunculkan beberapa pertanyaan yang memungkinkan penulisan kesimpulan yang ringkas dan efisien. Berikut adalah ringkasan dari pertanyaan yang dimunculkan oleh penulis.

1. RQ1: Apa yang mempengaruhi *behavior* dari aplikasi web ketika berada di *os* dan *browser* yang berbeda?
2. RQ2: Apa yang bisa diperbaharui dari kode aplikasi agar dapat meningkatkan performa dari aplikasi web berdasarkan hasil dari test *lighthouse*?

2. METODE PENELITIAN

Untuk menilai performa aplikasi bangunan bersejarah di Singapura ini, penelitian ini akan menggunakan berbagai metode pengujian. Pertama, pengujian *Lighthouse* akan dilakukan. *Lighthouse* adalah alat analisis open-source yang terintegrasi dengan *Google Chrome* dan digunakan untuk mengukur kualitas halaman *web*, mencakup aspek performa, aksesibilitas, praktik terbaik, dan *SEO* dengan skor 0-100. **(Darlis et al., 2016)** *Lighthouse* akan memberikan laporan lengkap yang menunjukkan area perbaikan dan memberikan saran untuk meningkatkan kualitas halaman *web* tersebut. Dalam penelitian ini, *Lighthouse* memberikan evaluasi komprehensif terhadap kinerja halaman web melalui metrik seperti *Performance Score*, *Speed Index*, dan *First Meaningful Paint.***(Hericko et al., 2021)**

Terakhir, *Mass testing* dilakukan untuk mengevaluasi performa aplikasi dalam menangani banyak pengguna secara bersamaan. Tujuan utama adalah mengidentifikasi bottleneck, waktu respons lambat, dan kegagalan sistem. Data dari pengujian ini dianalisis untuk mengukur skalabilitas dan stabilitas aplikasi saat mengalami lonjakan trafik.

Dengan menggabungkan kedua metode pengujian ini, yaitu *Lighthouse* dan pengujian masal, penelitian ini diharapkan dapat memberikan evaluasi performa yang komprehensif pada aplikasi bangunan bersejarah di Singapura.

3. HASIL DAN PEMBAHASAN

Pada bagian ini, disajikan data yang dikumpulkan pada penelitian yang telah dilakukan dengan menggunakan metode yang dipilih. Temuan dan Jawaban dari *Research Question* yang telah ditetapkan akan dibahas pada bagian Diskusi Pertanyaan Penelitian. Pada pelaksanaan *mass testing* digunakan beberapa perangkat dengan spesifikasi yang berbeda dengan total kasus pengujian sebanyak 25 kasus pengujian. Untuk pengujian dengan *Lighthouse* dilaksanakan sebanyak 3 kasus pengujian

3.1 Hasil dari *Mass Testing*

Dalam proses pengujian dan peninjauan ini, penulis melakukan penyebaran kuisioner dan pengujian internal. Dari kegiatan penyebaran kuisioner dan pengujian internal didapatkan 25 hasil pengujian, hal ini dapat dilihat pada tabel 1.

Tabel 1. Data Hasil Pengujian Masal (Mass Test)

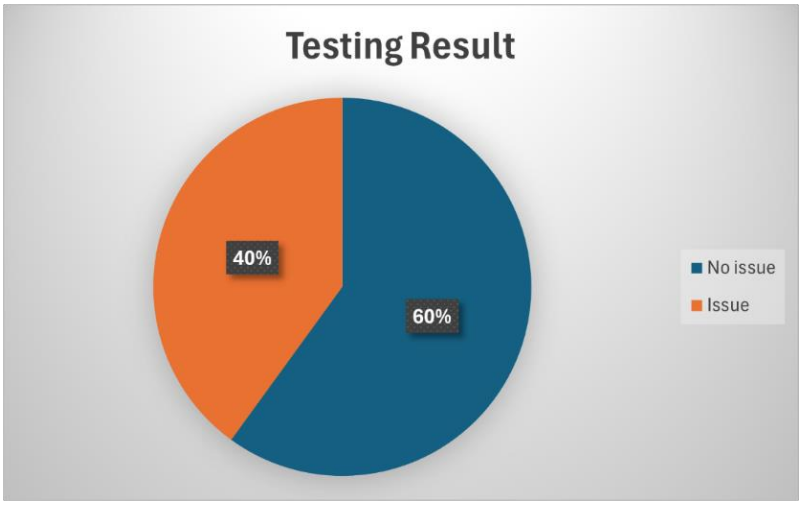
Perangkat yang digunakan	Sistem operasi perangkat	Browser yang digunakan	Apakah aplikasi mengalami error?	Sebutkan error apabila ditemukan
infinix inbook x1 pro	windows 11	Chrome	Ya	Pada saat zoom peta, terjadi sedikit lag.
Thinkpad E14	Windows 11	Chrome	Tidak	
Thinkpad E14	Windows 11	Edge	Tidak	
Thinkpad E14	Windows 11	Firefox	Tidak	
Thinkpad E14	Windows 11	Opera	Tidak	
Ideapad 300	Windows 10	Edge	Tidak	
Lenovo Thinkpad	Windows 10	Chrome	Tidak	
Ideapad 300	Windows 10	Chrome	Tidak	
Ideapad 300	Windows 10	Firefox	Tidak	
MacBook Air M1	MacOs Sonoma 14.5	Chrome	Tidak	

ISSN: 0216-2539 (Print)
E-ISSN: 2656-4157 (Online)

MacBook Air M1	MacOs Sonoma 14.4	Safari	Tidak	
iPhone 15 Pro	IOS 17.6.1	Safari	Tidak	
iPhone 15 Pro	IOS 17.6.1	Chrome	Tidak	
iPad Gen 9	IOS 17.5.2	Safari	Tidak	
iPad Gen 9	IOS 17.5.1	Chrome	Tidak	
iPhone 11	IOS 17.4.1	Chrome	Tidak	
iPhone 11	IOS 17.4.1	Safari	Tidak	
Redmi 6A	Android 9	Chrome	Ya	Tombol kontrol keluar dari bagian layar dan tidak dapat di *scroll*
Oppo A3	Android 14	Chrome	Ya	Video intro terpotong
Redmi 10	Android 13	Chrome	Ya	Video intro terpotong
Redmi 10	Android 13	Edge	Ya	Video intro terpotong
Redmi 10	Android 13	Firefox	Ya	Video intro terpotong
Redmi 10	Android 13	Opera	Ya	Video intro terpotong
Samsung A51	Android 13	Chrome	Ya	Teks deskripsi terpotong pada kedua sisi sehingga tidak dapat dibaca
Nubia neo 2 5g	Android 13	Chrome	Ya	Video intro terpotong

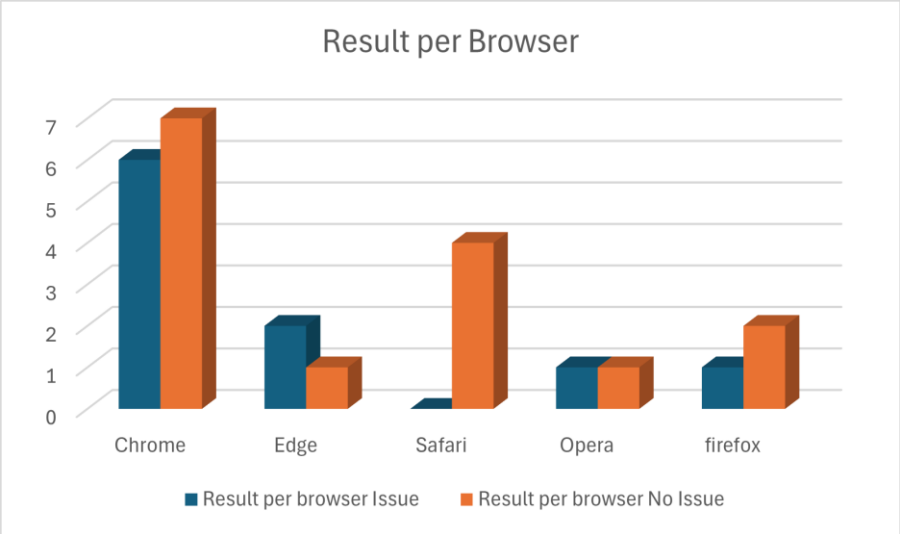
Berdasarkan data pada table 1, teridentifikasi pada diagram berikut bahwa 60% pengujian mengalami Kendala, sedangkan 40% pengujian berjalan tanpa masalah. Hasil ini menunjukkan bahwa mayoritas skenario pengujian menghadapi kendala kompatibilitas atau fungsionalitas. Distribusi ini perlu menjadi fokus perbaikan, terutama pada kombinasi sistem operasi (OS), browser, dan perangkat yang spesifik yang akan dijelaskan pada pembahasan berdasarkan browser tertentu pada siistem operasi tertentu dibawah.

ISSN: 0216-2539 (Print)
E-ISSN: 2656-4157 (Online)



Gambar 1. Distribusi Hasil Pengujian Secara Keseluruhan

Dengan pengolahan data lanjutan yang ditampilkan pada gambar 9 dibawah ini menunjukkan hasil pengujian pada setiap peramban (*browser)* tanpa memperhatikan jenis perangkat dan sistem operasi yang digunakan untuk menggambarkan compatibilitas aplikasi yang diuji. Berdasarkan data yang dihasilkan, Chrome menghasilkan 54% pengujian tanpa kendala dan 46% pengujian dengan kendala sehingga dinyatakan bahwa aplikasi yang diuji tidak sepenuhnya kompatibel dengan peramban chrome, pada peramban edge dihasilkan 34% pengujian dengan kendala dan 66% pengujian dengan kendala sehingga dinyatakan bahwa aplikasi yang diuji tidak kompatibel dengan peramban edge. Pada peramban safari dihasilkan 100% pengujian tanpa kendala sehingga dinyatakan bahwa safari adalah peramban yang kompatibel dengan aplikasi yang diuji. Lalu pada peramban opera dihasilkan 50% pengujian dengan kendala dan 50% pengujian tanpa kendala sehingga menyatakan bahwa opera kurang kompatibel dengan aplikasi yang dikembangkan, serta pada peramban firefox dinyatakan 66% hasil pengujian tanpa kendala dam 34% hasil pengujian dengan kendala. Sehingga dapat diambil kesimpulan bahwa safari adalah peramban yang paling kompatibel lalu disusul oleh firefox, chrome dan opera, diposisi terakhir atau peramban yang paling tidak kompatibel diikuti oleh edge.



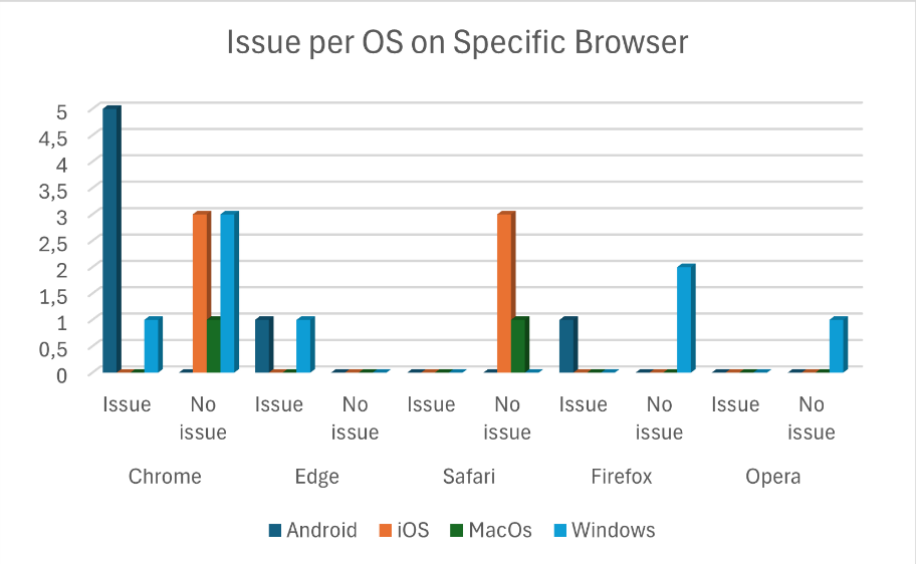
Gambar 2. Distribusi Issue per Browser

Berdasarkan diagram "Issue per OS on Specific Browser" yang terdapat pada gambar 10 dibawah, dapat diamati bahwa setiap peramban (browser) menunjukkan proporsi kendala (issue) dan tanpa kendala (no issue) yang berbeda-beda pada masing-masing sistem operasi (Android, iOS, MacOS, dan Windows). Secara umum, temuan ini sejalan dengan hasil pengolahan data lanjutan yang telah dijelaskan pada referensi, yaitu bahwa Safari memiliki tingkat kompatibilitas tertinggi, sedangkan Edge memiliki tingkat kendala paling banyak dibandingkan peramban lain.

Jika ditinjau secara lebih terperinci, Safari hampir tidak mengalami kendala pada semua sistem operasi. Bar "No issue" untuk Safari jauh lebih tinggi daripada bar "Issue", menandakan bahwa pengujian di Android, iOS, MacOS, maupun Windows sebagian besar berjalan lancar tanpa masalah. Hasil ini mendukung kesimpulan pada referensi yang menyatakan Safari 100% kompatibel. Di sisi lain, Firefox juga menunjukkan kinerja cukup baik dengan proporsi "No issue" yang lebih besar dibandingkan "Issue" pada seluruh sistem operasi, sehingga dapat dikatakan Firefox menempati urutan kedua dalam hal kompatibilitas.

Untuk Chrome, terlihat bahwa masih terdapat selisih yang cukup signifikan antara "Issue" dan "No issue" di beberapa sistem operasi, khususnya iOS. Meskipun Chrome menunjukkan hasil yang lumayan (terdapat porsi "No issue" yang tidak kecil), nilai "Issue" yang tinggi pada iOS menandakan bahwa aplikasi belum sepenuhnya kompatibel di lingkungan Chrome, sebagaimana diuraikan pula dalam referensi (masih terdapat sekitar 46% pengujian dengan kendala). Opera memperlihatkan hasil yang cenderung seimbang antara "Issue" dan "No issue" di semua sistem operasi, menandakan bahwa kompatibilitas Opera terhadap aplikasi yang diuji masih perlu ditingkatkan. Terakhir, Edge menunjukkan bar "Issue" yang lebih tinggi daripada "No issue" di hampir semua sistem operasi, sehingga menegaskan bahwa peramban ini adalah yang paling tidak kompatibel dibandingkan yang lain.

Dengan demikian, rangkuman keseluruhan dari diagram dan referensi yang diberikan menunjukkan urutan kompatibilitas tertinggi hingga terendah sebagai berikut: Safari, Firefox, Chrome, Opera, dan terakhir Edge.



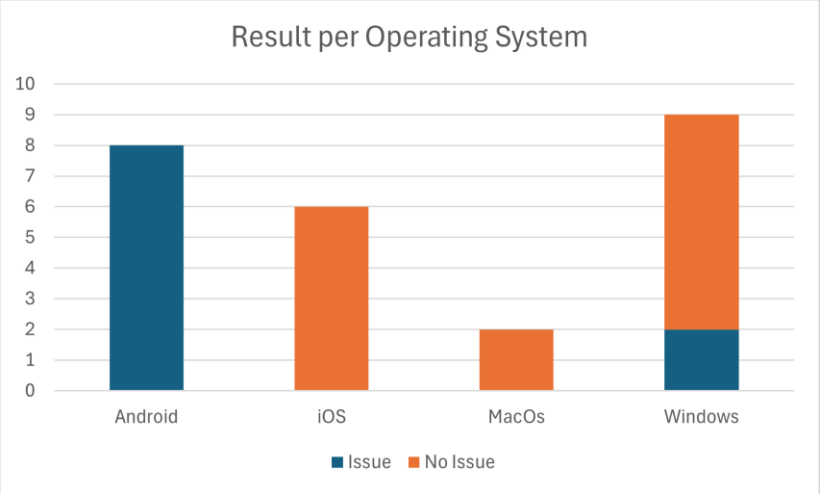
Gambar 3. Issue per Kombinasi Browser dan OS

Berdasarkan diagram "Result per Operating System" yang terlampir, terlihat bahwa proporsi kendala (issue) dan tanpa kendala (no issue) pada setiap sistem operasi (Android, iOS, MacOS, dan Windows)

beragam, namun pola tersebut sejalan dengan penjelasan pada referensi yang menyebutkan bahwa Safari menunjukkan tingkat kompatibilitas tertinggi, sementara Edge cenderung memiliki kendala paling banyak. Pada Android, jumlah kendala terlihat cukup signifikan, terutama saat dikaitkan dengan beberapa peramban seperti Chrome dan Edge, sedangkan iOS meskipun masih menghadapi kendala (terutama di Chrome), secara keseluruhan menunjukkan kinerja yang lebih baik dengan proporsi "No issue" lebih tinggi—terutama pada Safari yang hampir tidak mengalami masalah. MacOS juga memperlihatkan kondisi serupa, di mana Safari kembali unggul dibandingkan peramban lain. Sementara itu, Windows sebagai sistem operasi yang paling umum digunakan tetap memunculkan jumlah "Issue" yang cukup tinggi, khususnya ketika diuji dengan Edge. Temuan ini mendukung kesimpulan pada referensi bahwa Safari adalah peramban yang paling kompatibel di berbagai sistem operasi, disusul Firefox, lalu Chrome dan Opera, serta Edge di posisi terakhir.

Pada perangkat Windows, hal ini mungkin disebabkan oleh keragaman versi OS (misalnya Windows 10 dan Windows 11) atau konfigurasi driver grafis. Pada perangkat Android, fragmentasi spesifikasi perangkat seperti resolusi layar dan versi OS menjadi faktor utama. Perangkat MacOS dan iOS menunjukkan hasil lebih stabil, didukung ekosistem tertutup yang minim variasi spesifikasi.

.

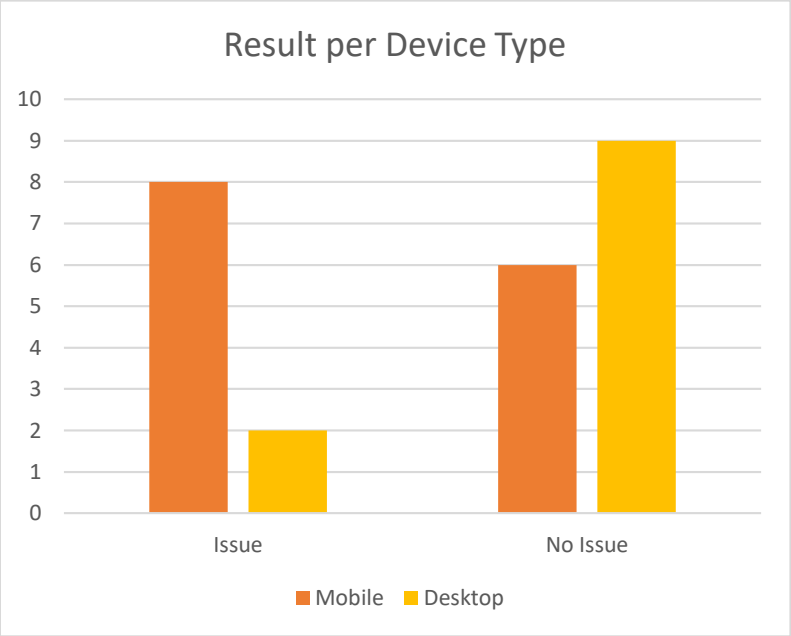


Gambar 4. Distribusi Issue per Sistem Operasi

Berdasarkan diagram "Result per Device Type" yang terlampir, dapat diamati bahwa jumlah "Issue" (kendala) pada perangkat Mobile lebih tinggi dibandingkan Desktop, sementara proporsi "No Issue" (tanpa kendala) justru lebih besar di Desktop. Hasil ini sejalan dengan temuan pada referensi sebelumnya yang menekankan pentingnya pengujian lintas peramban (browser) dan lintas sistem operasi. Perangkat Mobile, yang umumnya menggunakan Android (dengan Chrome, Opera, Edge) atau iOS (dengan Safari atau Chrome), cenderung menunjukkan kendala lebih banyak—terutama pada kombinasi browser dan OS tertentu seperti Chrome di iOS atau Edge di Android.

Di sisi lain, perangkat Desktop, khususnya yang menggunakan Safari di MacOS atau Firefox di Windows, memperlihatkan tingkat kompatibilitas yang lebih baik, sehingga menghasilkan proporsi "No Issue" yang lebih tinggi. Temuan ini juga mendukung kesimpulan pada referensi bahwa Safari (yang banyak digunakan di ekosistem Apple) adalah peramban paling kompatibel, disusul oleh Firefox, lalu Chrome dan Opera, dan terakhir Edge. Dengan demikian, perbedaan proporsi kendala antara Mobile dan Desktop pada diagram ini menegaskan perlunya pengujian menyeluruh pada berbagai kombinasi sistem operasi dan peramban agar aplikasi dapat berjalan optimal di semua jenis perangkat.

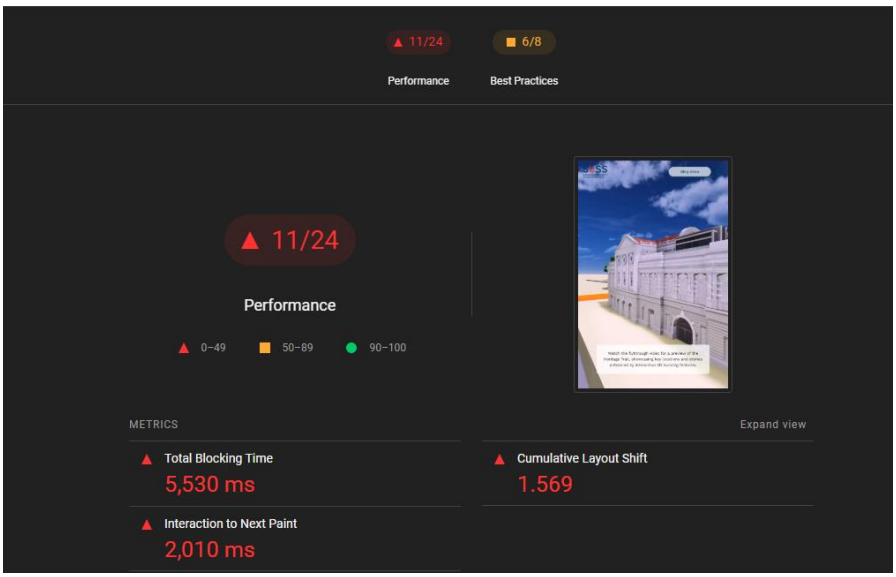
ISSN: 0216-2539 (Print)
E-ISSN: 2656-4157 (Online)



Gambar 5. Distribusi Issue per Device Type

3.2 Hasil dari *Lighthouse*

Dalam proses pengujian dan peninjauan ini, penulis melakukan juga melakukan pengujian dengan menggunakan *Lighthouse* pada browser Edge, Chrome dan Firefox yang akan digunakan untuk meninjau karakteristik aplikasi sesuai dengan ISO/IEC pada bagian *performance efficiency*. Hasil pengujian menunjukkan bahwa situs web dari aplikasi yang diuji memiliki skor performa yang rendah di ketiga browser tersebut, yang menunjukkan adanya beberapa area yang perlu ditingkatkan.

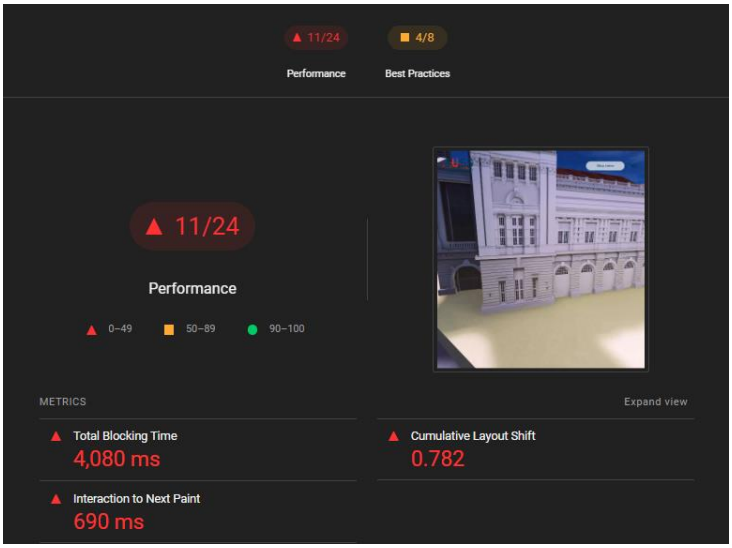


Gambar 6. Hasil Pengujian Lighthouse pada Chrome

Situs web dari aplikasi yang diuji memiliki skor performa 11 dari 100 di Chrome. Metrik utama yang diukur adalah Total Blocking Time (TBT), Interaction to Next Paint (INP), dan Cumulative Layout Shift (CLS). Situs web memiliki TBT 5,530 ms, INP 2.010 ms, dan CLS 1.569. TBT yang tinggi

ISSN: 0216-2539 (Print)
E-ISSN: 2656-4157 (Online)

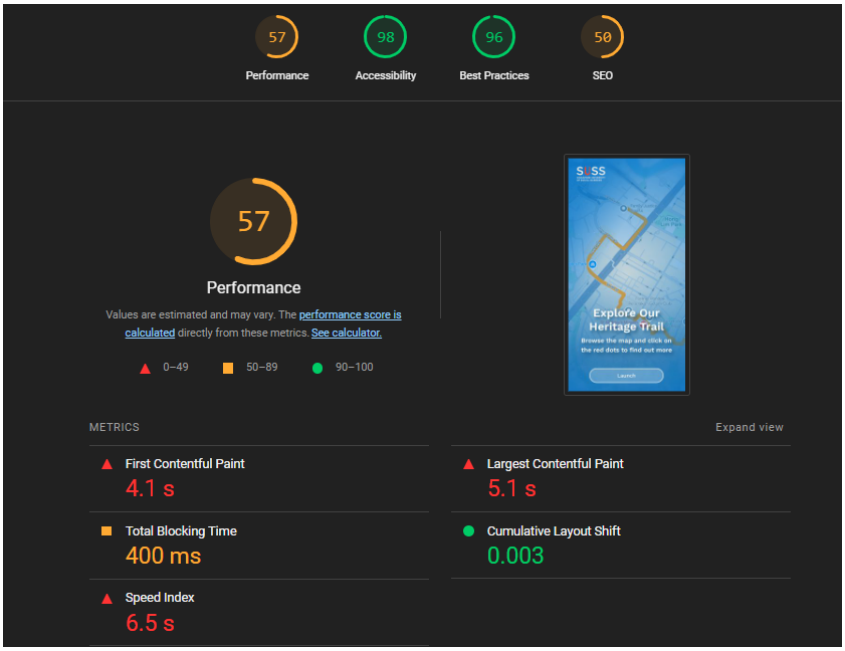
menunjukkan bahwa *main thread* diblokir untuk waktu yang signifikan, yang dapat menyebabkan keterlambatan dalam pemuatan dan interaksi pengguna. INP mengukur latensi respons terhadap interaksi pengguna, dan skor yang tinggi menunjukkan bahwa situs web mungkin terasa lambat untuk merespons tindakan pengguna. CLS mengukur stabilitas visual, dan skor yang tinggi menunjukkan bahwa elemen-elemen pada halaman mungkin bergeser secara tidak terduga, yang dapat mengganggu pengalaman pengguna.
Situs web dari aplikasi yang diuji menerima skor 6 dari 8 untuk praktik terbaik dan memiliki beberapa masalah pengalaman pengguna, seperti menampilkan gambar dengan rasio aspek yang salah dan resolusi rendah.



Gambar 7. Hasil Pengujian Lighthouse pada Edge

Situs web dari aplikasi yang diuji memiliki skor performa 11 dari 24 di Edge. Metrik utama yang diukur adalah Total Blocking Time (TBT), Interaction to Next Paint (INP), dan Cumulative Layout Shift (CLS). Situs web memiliki TBT 4,080 ms, INP 690 ms, dan CLS 0.782. mengukur latensi respons terhadap interaksi pengguna, dan skor yang tinggi menunjukkan bahwa situs web mungkin terasa lambat untuk merespons tindakan pengguna. **CLS** yang cukup tinggi menunjukkan bahwa stabilitas visual perlu ditingkatkan. Situs web menerima skor 4 dari 8 untuk praktik terbaik dan memiliki beberapa masalah pengalaman pengguna, seperti menampilkan gambar dengan rasio aspek yang salah dan resolusi rendah.

ISSN: 0216-2539 (Print)
E-ISSN: 2656-4157 (Online)



Gambar 8. Hasil Pengujian Lighthouse pada Firefox

Situs web memiliki skor performa 57 dari 100 di Firefox. Metrik utama yang diukur adalah First Contentful Paint (FCP), Total Blocking Time (TBT), Speed Index, Largest Contentful Paint (LCP), dan Cumulative Layout Shift (CLS). Situs web memiliki FCP 4,1 detik, TBT 400 ms, Speed Index 6,5 detik, LCP 5,1 detik, dan CLS 0,003. FCP dan LCP yang tinggi menunjukkan bahwa pengguna harus menunggu beberapa saat sebelum konten utama halaman ditampilkan. TBT yang rendah menunjukkan kinerja *main thread* yang relatif baik. CLS yang sangat rendah menunjukkan stabilitas visual yang baik. Situs web menerima skor 96 dari 100 untuk praktik terbaik dan dinyatakan cukup baik untuk digunakan.

Meskipun Lighthouse adalah alat yang konsisten, perbedaan *browser engine* dan cara mereka merender situs web dapat menyebabkan variasi dalam skor metrik. Perbedaan cara *rendering engine* Blink (yang digunakan oleh Chrome dan Edge) dan Gecko (yang digunakan oleh Firefox) dalam merender halaman web dapat menyebabkan variasi dalam skor metrik Lighthouse. Blink menggunakan arsitektur multi-proses yang memungkinkan isolasi yang lebih baik, sedangkan Gecko secara tradisional menggunakan arsitektur *single-process*, meskipun versi terbaru telah mengadopsi beberapa elemen multi-proses.

3.3 Jawaban atas *Research Question*

3.3.1 Apa yang mempengaruhi *behavior* dari aplikasi web ketika berada di *os* dan *browser* yang berbeda?

Berdasarkan penelitian yang dilakukan oleh penulis, didapatkan bahwa *behavior* aplikasi web saat diakses di berbagai *OS* dan *browser* ternyata dipengaruhi oleh sejumlah faktor. Kombinasi antara browser dan sistem operasi memegang peran penting, contohnya Chrome dan Edge di Android menunjukkan banyak masalah, kemungkinan karena ukuran layar yang berbeda dari apa yang digunakan sebagai perangkat target penggunaan aplikasi. Selain itu, fragmentasi pada sistem operasi, seperti variasi versi dan konfigurasi di Windows dan Android, juga menjadi penyebab munculnya *issue*. Keragaman driver grafis di Windows dan perbedaan resolusi serta versi OS di Android turut berkontribusi pada masalah ini. Integrasi antara browser bawaan dan sistem operasi juga mempengaruhi stabilitas, seperti Safari di iOS yang minim *issue* karena integrasinya yang erat. Terakhir, perbedaan ukuran layar antara perangkat *mobile* dan desktop mempengaruhi responsivitas UI, sehingga *issue* seperti elemen terpotong sering terjadi di perangkat *mobile*.

3.3.2 Apa yang bisa diperbaharui dari kode aplikasi agar dapat meningkatkan performa dari aplikasi web berdasarkan hasil dari test *lighthouse*?

Berdasarkan hasil pengujian, untuk meningkatkan performa aplikasi web, pengembang harus berfokus pada optimasi Total Blocking Time (TBT) dengan mengurangi jumlah dan durasi tugas di *main thread*, serta optimasi Interaction to Next Paint (INP) dengan meningkatkan responsivitas aplikasi. Cumulative Layout Shift (CLS) juga perlu dioptimalkan dengan memastikan ukuran gambar ditentukan secara eksplisit dan menghindari penambahan konten di atas konten yang sudah ada. Selain itu, perlu dilakukan optimasi First Contentful Paint (FCP) dan Largest Contentful Paint (LCP) dengan memprioritaskan pemuatan sumber daya kritis dan optimasi gambar. Perbaikan juga harus mencakup penggunaan rasio aspek dan resolusi gambar yang tepat, serta memperhatikan perbedaan browser dalam menangani kode dan melakukan pengujian di berbagai browser.

4. KESIMPULAN DAN SARAN

Penelitian ini menunjukkan bahwa kualitas dan performa aplikasi GIS bergantung pada kombinasi OS, browser, dan perangkat yang digunakan. Meskipun aplikasi berjalan baik di beberapa skenario, mayoritas pengujian menunjukkan masalah kompatibilitas, terutama pada Chrome di Android dan Edge di Windows. Fragmentasi OS dan browser menjadi tantangan utama.

Hasil Lighthouse menyoroti perlunya optimasi kode untuk meningkatkan performa. Peningkatan harus difokuskan pada TBT, INP, CLS, FCP, dan LCP. Oleh karena itu, pengembang perlu melakukan pengujian menyeluruh di berbagai platform, memperhatikan fragmentasi sistem, serta mengoptimalkan kode agar kompatibilitas dan performa aplikasi dapat ditingkatkan..

5. REFERENSI

Agyei, M. Y., Ajoke, O. A., & Asa, M. A. (2023, August 30). Hybrid Software Testing Model to Improve Software Quality Assurance. *Global Journal of Engineering and Technology Advances, 17*, 104-116. doi:https://doi.org/10.30574/gjeta.2023.17.1.0206

Aisah, N. (2015). Rancangan bangunan sistem perangkat lunak seleksi pegawai tetap dengan metode Weighted Product di PT Denki Engineering. (Skripsi T Informatika 2015).

Akinyede, J. O., Ponnle, A. A., Olebu, C., Akinluyi, F., Thompson, A., Dahunsi, O. A., . . . Oyinloye, M. (2023). Development of a Software System for Realtime Management of Crime Reports in Southwestern Nigeria: The Administrative Approach. *American Journal of Science Engineering and Technology, 8*(1), 23-32. doi:10.11648/j.ajset.20230801.13

Aniwange, A., Nyishar, P., Afolabi, B., & Ejidokun, A. (2021, August). A Hybrid Software Test Automation for Educational Portals. *INTERNATIONAL JOURNAL OF INNOVATIONS IN ENGINEERING RESEARCH AND TECHNOLOGY, VII*(8), 209-216. doi:10.17605/OSF.IO/4ADP8

Atoum, I., Baklizi, M. K., Alsmadi, I., Otoom, A. A., Alhersh, T., Ababneh, J., . . . Alshahrani, S. M. (2021, October). Challenges of Software Requirements Quality Assurance and Validation: A Systematic Literature Review. *IEEE Access, 9*, 137613 - 137634. doi: 10.1109/ACCESS.2021.3117989

Badran, O. N., & Al-Haddad, S. I. (2018). The Impact of Software User Experience on Customer Satisfaction.

Bozbay, Z., & Başlar, E. K. (2020). IMPACT OF BRAND TRUST ON BRAND LOYALTY: MEDIATING ROLE OF BRAND AFFECT.

Darlis, A. R., Darlis, D., Cahyadi, W. A., & Chung, Y.-H. (2016). Underwater Visible Light Communication using Underwater Visible Light. Busan.

Heričko, T., Šumak, B., & Brdnik, S. (2021). Towards Representative Web Performance Measurments with Google Lighthouse. *Student Computer Science Research Conference.* Maribor.

Ikhwan, A. N. (2020). Jurnal Software. doi:10.31219/osf.io/n9gts

Khan, S. M. (2023). *Software Measurement Metrics in Software Quality Assurance.* Dipetik October 18, 2024, dari https://www.researchgate.net/publication/371905205_Software_Measurement_in_Software_Quality_Assurance

Komalasari, D., & Ulfa, M. (2020). Pengujian Usability Heuristic Terhadap Perangkat Lunak Pembelajaran Matematika. doi:10.30812/matrik.v19i2.687

Pargaonkar, S. (2021, March). The Crucial Role of Inspection in Software Quality Assurance. *Journal of Science & Technology*, *II*(1), 70-77. Dipetik January 16, 2025, dari https://thesciencebrigade.com/jst/article/view/42

Silitonga, M. K., & Rosyida, S. (2015). Animasi Interaktif Sebagai Media Sosialisasi Infonesia Tsunami Early Warning System (INATEWS).

Vukašinović, M. (2023). SOFTWARE QUALITY ASSURANCE. *SOFTWARE QUALITY ASSURANCE* (hal. 1-16). Montenegro: ResearchGate. Dipetik October 18, 2024, dari https://www.researchgate.net/publication/374755430_SOFTWARE_QUALITY_ASSURANCE