

Analisis Performa dan Implementasi Component-Based Architecture pada Antarmuka Minilemon Learning System Menggunakan ISO/IEC 25010

Nur Fatiq^{1*}, Bayu Adhi Nugroho², Moch Yasin³, Anang Kunaefi⁴

^{1,2,3,4} Fakultas Sains dan Teknologi, Sistem Informasi, Universitas Islam Negeri Sunan Ampel, Surabaya, Indonesia

Email: ^{1*}fatiqnur@gmail.com, ²bayu.lecture@gmail.com, ³moch.yasin@uinsby.ac.id, ⁴akunaefi@uinsby.ac.id

(* Email Corresponding Author: fatiqnur@gmail.com)

Received: February 28, 2026 | Revision: March 8, 2026 | Accepted: June 6, 2026

Abstrak

Transformasi digital di sektor pendidikan menuntut ketersediaan *Learning Management System (LMS)* dengan antarmuka yang responsif dan berkinerja tinggi. Namun, pengembangan antarmuka skala besar seringkali dihadapkan pada masalah redundansi kode dan performa muat (*loading time*) yang lambat. Penelitian ini bertujuan untuk mengimplementasikan dan mengevaluasi kualitas purwarupa antarmuka *Minilemon Learning System* menggunakan pendekatan *Component-Based Architecture (CBA)* pada kerangka kerja Next.js. Evaluasi dilakukan pada lingkungan sisi klien (*client-side*) tanpa integrasi *backend* menggunakan standar kualitas perangkat lunak internasional ISO/IEC 25010. Tiga karakteristik utama yang diukur meliputi *Functional Suitability*, *Performance Efficiency*, dan *Usability* (Aksesibilitas). Hasil pengujian *Black Box* menunjukkan tingkat keberhasilan 100% pada validasi fungsionalitas formulir dan manajemen status lokal. Pengujian otomatis menggunakan Google Lighthouse mencatatkan skor performa rata-rata di atas 95, dengan *First Contentful Paint (FCP)* rata-rata 0,33 detik dan ketiadaan pergeseran tata letak (*Cumulative Layout Shift* 0,009). Selain itu, skor aksesibilitas mencapai metrik ideal (86-92) berkat kepatuhan pada rasio kontras warna dan web semantik. Kesimpulannya, arsitektur berbasis komponen menggunakan Next.js secara efektif mengeliminasi utang teknis dan menghasilkan antarmuka berkualitas yang siap diproduksi (*production-ready*) untuk tahapan integrasi sistem selanjutnya.

Kata Kunci: *Component-Based Architecture*, Evaluasi Antarmuka, ISO/IEC 25010, *Learning Management System*, Next.js

Abstract

Digital transformation in the education sector demands *Learning Management Systems (LMS)* with responsive and high-performance interfaces. However, large-scale interface development often faces issues with code redundancy and slow loading times. This study aims to implement and evaluate the quality of the *Minilemon Learning System* interface prototype using a *Component-Based Architecture (CBA)* approach on the Next.js framework. The evaluation was conducted in a client-side environment without backend integration, employing the international software quality standard ISO/IEC 25010. The three main characteristics measured included *Functional Suitability*, *Performance Efficiency*, and *Usability* (Accessibility). *Black Box* testing results demonstrated a 100% success rate in form functionality validation and local state management. Automated testing using Google Lighthouse recorded an average performance score above 95, with an average *First Contentful Paint (FCP)* of 0.33 seconds and near-zero layout shifts (*Cumulative Layout Shift* 0.009). Furthermore, accessibility scores reached ideal metrics (86-92) due to compliance with color contrast ratios and semantic web standards. In conclusion, component-based architecture using Next.js effectively eliminates technical debt and produces a high-quality, production-ready interface for subsequent system integration phases.

Keywords: *Component-Based Architecture*, Interface Evaluation, ISO/IEC 25010, *Learning Management System*, Next.js

1. PENDAHULUAN

Transformasi digital yang masif di sektor pendidikan pasca-pandemi telah mengubah lanskap pembelajaran secara fundamental [1]. Institusi pendidikan dan perusahaan pelatihan kini tidak hanya menuntut ketersediaan materi daring, tetapi juga pengalaman pengguna yang responsif dan menarik [2]. *Learning Management System (MLS)* telah menjadi tulang punggung infrastruktur pendidikan modern. Namun, seiring dengan meningkatnya kompleksitas fitur yang dibutuhkan membuat pengembangan antarmuka web juga menghadapi tantangan teknis yang signifikan terutama dalam waktu muat (*loading time*). Studi menunjukan bahwa keterlambatan *loading time* beberapa detik saja bisa menurunkan tingkat kepuasan pengguna [3]. Oleh karena itu, performa dan struktur kode yang efisien menjadi faktor yang penting dalam keberhasilan sebuah platform pendidikan digital.

Permasalahan utama yang sering muncul dalam pengembangan antarmuka *LMS* berskala besar adalah ketidakkonsistenan desain dan redundansi kode. Dalam metode pengembangan web tradisional, pengembang sering kali menulis ulang kode yang sama untuk elemen visual yang berulang, seperti tombol atau formulir *input*. Hal ini menyebabkan basis kode menjadi bengkak, sulit dipelihara dan rentan *bug* ketika terjadi perubahan desain. Selain itu, aplikasi web yang sangat bergantung pada *Client-Side Rendering (CSR)* murni sering kali mengalami penurunan performa atau latensi pemuatan awal (*initial load*), terutama pada perangkat dengan spesifikasi rendah karena beratnya beban eksekusi JavaScript [4]. PT. Minilemon Nusantara, sebagai perusahaan yang berfokus pada pendidikan karakter anak melalui nilai-nilai budaya, menginisiasi pengembangan *Minilemon Learning System (MLS)*. Tantangan teknis yang dihadapi dalam proyek ini adalah bagaimana menerjemahkan rancangan desain *High-Fidelity* yang kompleks dari Figma menjadi sebuah aplikasi web yang fungsional, memiliki performa rendering tinggi dan mudah dikembangkan lebih lanjut oleh tim di masa depan tanpa harus merombak ulang struktur dasar. Tanpa arsitektur *frontend* yang kokoh, risiko

terjadinya utang teknis (*technical debt*) akan sangat tinggi [5], yang pada akhirnya yang menghambat integrasi dengan sistem *backend* di kemudian hari.

Sebagai solusi atas permasalahan tersebut, penelitian ini mengusulkan penerapan *Component-Based Architecture (CBA)* menggunakan *framework* Next.js. Pendekatan berbasis komponen memungkinkan pengembang untuk memecah antarmuka pengguna menjadi bagian-bagian kecil yang independen dan digunakan kembali (*reusable*), seperti atom, molekul dan organisme dalam konsep desain atomik [6]. Next.js dipilih karena kemampuan dalam melakukan *Server-Side Rendering (SSR)* dan *Static Site Generation (SSG)*, yang secara teoritis mampu meningkatkan kecepatan *First Contentful Paint (FCP)* dan *Largest Contentful Paint (LCP)* secara signifikan dibandingkan aplikasi React konvensional [7]. Dengan pendekatan ini, diharapkan antarmuka *Minilemon Learning System* tidak hanya konsistensi secara visual, tetapi juga efisiensi dalam menggunakan sumber daya peramban (*browser resource*).

Sejumlah penelitian terdahulu telah membahas pengembangan *LMS* dan teknologi web modern. Penelitian oleh [8] menyoroti efektivitas *e-Learning* menggunakan *LMS* seperti Moodle dan Google Classroom, namun fokus kajiannya terbatas pada aspek fundamental pendidikan dan penerimaan pengguna, tanpa membahas aspek rekayasa perangkat lunak di balik antarmuka tersebut. Selanjutnya, penelitian [9] melakukan pengembangan desain *UI/UX* untuk *LMS* dengan pendekatan *User Centered Design*, meskipun menghasilkan rancangan yang baik, lingkupnya berhenti pada tahap purwarupa desain dan tidak sampai pada tahap implementasi kode serta pengukuran performa teknis.

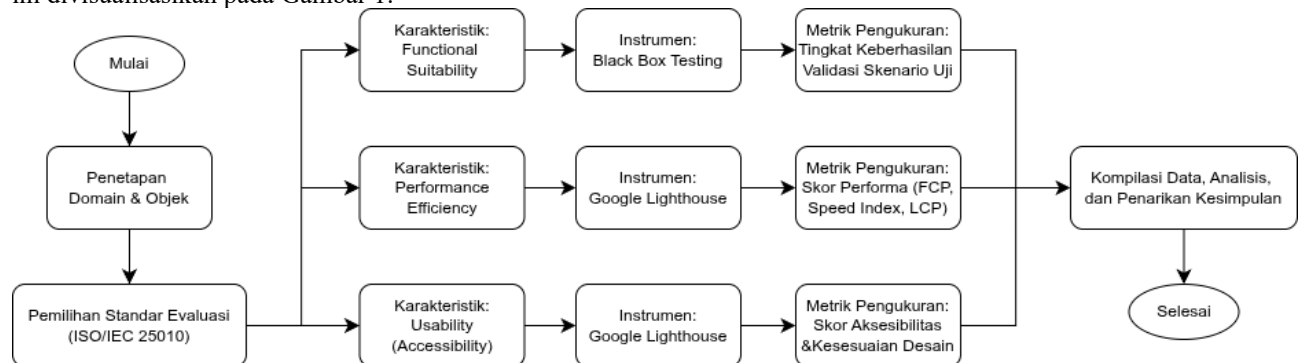
Di sisi teknis, [10] telah meneliti pengembangan *frontend* menggunakan Next.js pada sistem layanan Tekos. penelitian tersebut berhasil membuktikan keunggulan Next.js, namun studi kasus yang diangkat bukanlah *LMS* yang memiliki karakteristik interaksi pengguna yang spesifik dan kompleks. Penelitian oleh [11] membahas implementasi *LMS* untuk pengembangan guru, namun menggunakan arsitektur monolitik yang kurang fleksibel dibandingkan arsitektur modern berbasis komponen.

Berdasarkan tinjauan terhadap penelitian di atas, ditemukan adanya *gap analysis* yaitu kurangnya penelitian yang secara spesifik mengintegrasikan implementasi *Component-Based Architecture* pada domain *MLS* pendidikan karakter dengan performa teknis yang mendalam sebelum tahap integrasi *backend*. Kebanyakan penelitian berfokus pada hasil akhir sistem secara utuh atau hanya pada desain visual semata. Penelitian ini hadir untuk mengisi celah tersebut dengan fokus pada validasi kualitas rekayasa *frontend*.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk mengimplementasikan dan menganalisis performa antarmuka *Minilemon Learning System*. Fokus utama penelitian adalah penerapan *Component-Based Architecture* menggunakan Next.js berdasarkan desain *High-Fidelity* yang telah dirancang, serta melakukan pengukuran performa teknis untuk memvalidasi kualitas rekayasa antarmuka sebelum tahap integrasi *backend* dilakukan.

2. METODOLOGI PENELITIAN

Penelitian ini merupakan jenis penelitian evaluatif terapan yang berfokus pada tahapan *Quality Assurance (QA)* perangkat lunak. Objek utama dalam penelitian ini adalah purwarupa antarmuka web dari *Minilemon Learning System* yang dikembangkan menggunakan kerangka kerja Next.js. Mengingat batasan penelitian yang dilakukan sebelum tahap integrasi dengan *backend*, metodologi tidak mencakup siklus pengembangan perangkat lunak secara utuh, melainkan difokuskan secara eksklusif pada evaluasi teknis sisi klien (*client-side evaluation*). Untuk memastikan bahwa evaluasi dilakukan secara objektif, terukur, dan diakui secara akademis, penelitian ini menggunakan kerangka kerja kualitas perangkat lunak internasional, yaitu ISO/IEC 25010 [12]. Secara sistematis, alur pelaksanaan pengujian dalam penelitian ini divisualisasikan pada Gambar 1.



Gambar 1. Alur Penelitian

Berdasarkan Gambar 1, proses penelitian dan pengujian kualitas antarmuka dilakukan dengan enam tahapan utama yang terstruktur sebagai berikut:

- Penetapan Domain dan Objek:** Tahap inisiasi dimana objek penelitian ditetapkan secara spesifik, yaitu antarmuka *Minilemon Learning System* yang dibangun menggunakan Next.js. Mengingat sistem belum terintegrasi dengan *backend*, lingkungan pengujian dikondisikan menggunakan mock data (data simulasi statis).

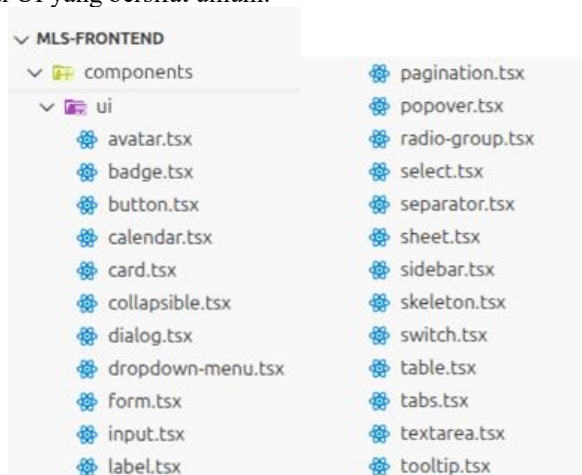
- b. **Pemilihan Standar Evaluasi:** Penelitian ini menggunakan kerangka kerja evaluasi berdasarkan standar internasional ISO/IEC 25010 [12], yang telah terbukti keandalannya dalam mengukur kualitas berbagai sistem informasi dan situs web [13]. Model yang digunakan difokuskan pada *Product Quality Model* untuk mengevaluasi properti perangkat lunak secara statis maupun dinamis pada sisi klien.
- c. **Pemilihan Karakteristik:** Dari delapan karakteristik utama ISO/IEC 25010, tiga karakteristik diekstraksi karena paling relevan dengan batasan pengujian *frontend*, yaitu *Functional Suitability* (kesesuaian fungsi), *Performance Efficiency* (kecepatan dan efisiensi), dan *Usability* (khususnya pada aspek aksesibilitas) [12].
- d. **Penentuan Instrumen Pengujian:** Setiap karakteristik dipetakan dengan instrumen yang tepat. Black Box Testing digunakan untuk menguji fungsionalitas secara manual, sementara Google Lighthouse digunakan untuk mengukur performa dan aksesibilitas secara objektif, sejalan dengan praktik evaluasi performa digital web modern yang komprehensif [14].
- e. **Pengukuran Metrik:** Tahap eksekusi di mana data ditarik berdasarkan instrumen. Pada pengujian fungsional, metrik yang dinilai adalah tingkat keberhasilan dari skenario uji coba. Pada pengujian performa dan aksesibilitas, metrik yang diambil adalah skor kuantitatif (skala 0-100) yang dihasilkan oleh Lighthouse, mencakup parameter *Core Web Vitals* seperti *First Contentful Paint (FCP)* dan *Speed Index*.
- f. **Kompilasi dan Analisis Data:** Seluruh data hasil pengujian dikumpulkan untuk dianalisis. Analisis ini akan menjawab hipotesis apakah arsitektur komponen Next.js yang diimplementasikan telah memenuhi standar kelayakan perangkat lunak sebelum memasuki tahap pengembangan selanjutnya.

Secara keseluruhan, metodologi ini dirancang untuk menghasilkan data pengujian yang empiris dan independen dari ketiadaan sistem backend. Melalui kombinasi pengujian fungsional manual dan pengujian performa otomatis, hasil yang ditarik pada akhir proses evaluasi akan memberikan gambaran komprehensif mengenai kesiapan teknis dan kualitas rekayasa antarmuka produk.

3. HASIL DAN PEMBAHASAN

3.1 Implementasi Antarmuka Berbasis Komponen

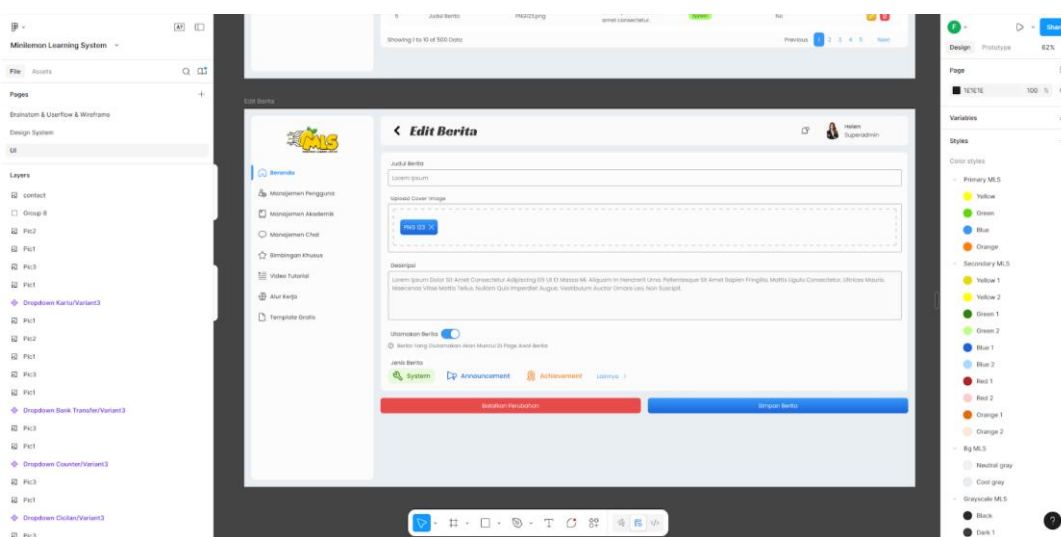
Tahap implementasi antarmuka *Minilemon Learning System* diawali dengan pembentukan fondasi sistem desain menggunakan pendekatan *Component-Based Architecture*. Sesuai dengan prinsip *Atomic Design*, elemen-elemen dasar penyusun antarmuka (seperti atom dan molekul) dibangun secara spesifik dan diisolasi agar dapat digunakan kembali (*reusable*). Mengingat sifat proyek yang merupakan *repository* privat perusahaan, pembahasan arsitektur dibatasi pada direktori komponen fundamental UI yang bersifat umum.



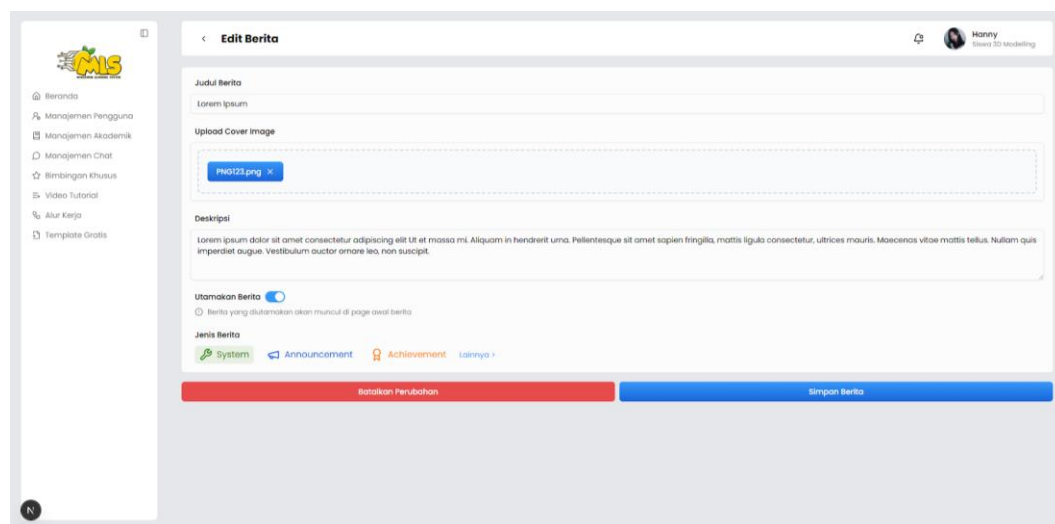
Gambar 2. Struktur Direktori Arsitektur Komponen Fundamental pada Next.js

Berdasarkan Gambar 2, dapat dilihat implementasi pemisahan elemen antarmuka ke dalam direktori terpusat (*components/ui*). Setiap elemen, mulai dari *button*, *dialog*, *form*, *input*, *table*, hingga *tabs*, dienkapsulasi ke dalam berkas modul *.tsx* yang independen. Meskipun komponen-komponen ini berfungsi sebagai elemen dasar, di dalam masing-masing berkas tersebut terdapat logika penataan gaya (*styling*) telah dimodifikasi menggunakan utilitas Tailwind CSS untuk menyuntikkan token warna khusus milik identitas visual Minilemon. Melalui pendekatan modular ini, pengembang cukup memanggil komponen-komponen tersebut di berbagai halaman tanpa perlu menulis ulang struktur HTML secara redundan, yang secara signifikan meningkatkan skalabilitas dan pemeliharaan kode (*maintainability*).

Setelah fondasi komponen terbangun, langkah selanjutnya adalah merakit komponen-komponen tersebut menjadi sebuah halaman utuh (*organisme*) berdasarkan rancangan visual yang telah disetujui. Gambar 3 menunjukkan desain High-Fidelity dari halaman "Edit Berita" yang dirancang menggunakan perangkat lunak Figma.



Gambar 3. Desain High-Fidelity Halaman Edit Berita pada Figma



Gambar 4. Hasil Implementasi Halaman Edit Berita Menggunakan Next.js

Seperti yang terlihat pada Gambar 4, seluruh struktur visual—mulai dari tata letak grid, form input khusus, ukuran tipografi, hingga varian tombol bersyarat—berhasil diterjemahkan dengan presisi yang tinggi. Halaman ini dibangun sepenuhnya dengan merakit ulang komponen-komponen reusable yang telah didefinisikan pada Gambar 2. Implementasi antarmuka ini membuktikan bahwa arsitektur berbasis komponen tidak hanya menjaga konsistensi desain, tetapi juga memastikan kesiapan antarmuka untuk dievaluasi lebih lanjut kinerjanya menggunakan instrumen pengujian.

3.2 Evaluasi *Functional Suitability*

Karakteristik *Functional Suitability* dievaluasi untuk memastikan bahwa antarmuka yang dibangun menyediakan fungsi yang tepat dan merespons interaksi pengguna sesuai dengan rancangan spesifikasi kebutuhan. Pengujian dilakukan menggunakan metode *Black Box Testing* dengan teknik *Equivalence Partitioning* [15], di mana antarmuka diuji secara fungsional tanpa melihat struktur internal kodenya.

Mengingat pengujian dilakukan pada lingkungan *frontend* mandiri (tanpa integrasi backend), skenario uji difokuskan pada tiga aspek utama: (1) validasi formulir sisi klien, (2) interaktivitas komponen dinamis, dan (3) kelancaran navigasi routing antar halaman. Hasil pengujian dirangkum pada Tabel 2.

Tabel 1. Hasil Pengujian Black Box pada Antarmuka Minilemon Learning System

No	Halaman	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Autentikasi Siswa & Mentor	Menekan tombol otentikasi saat kolom email atau password dibiarkan kosong atau format salah.	Formulir ditahan (<i>prevent default</i>) dan memunculkan indikator visual wajib diisi pada antarmuka.	Validasi formulir berjalan instan melalui manajemen status lokal.	Valid

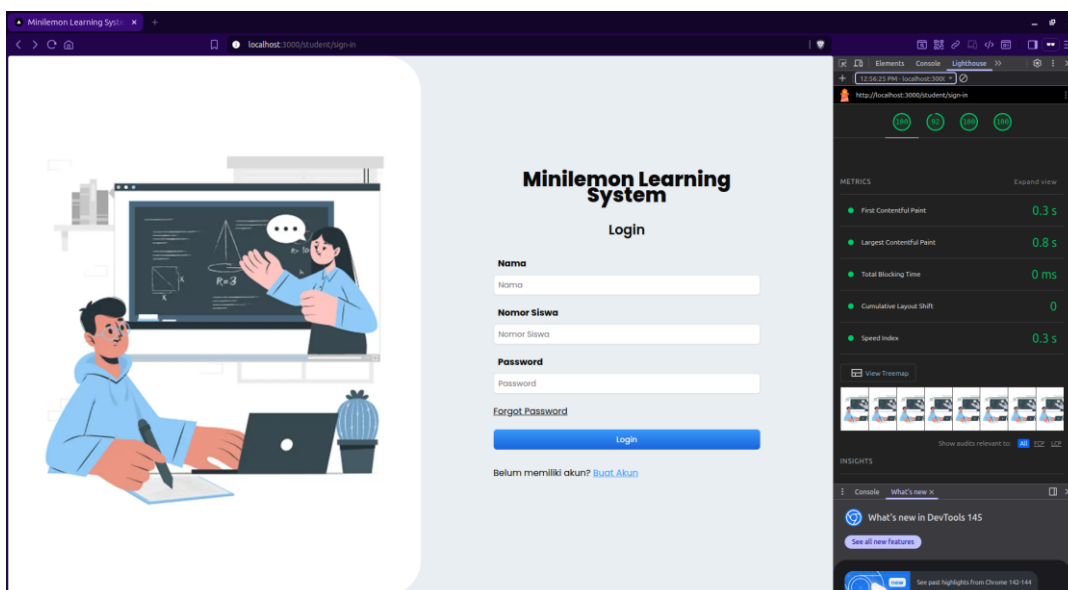
2	Beranda & Dasbor Admin	Memuat elemen Card pada halaman Dasbor dan mengubah ukuran layar ke resolusi perangkat mobile (< 768px).	Grid komponen Card menyesuaikan diri (dari baris horizontal menjadi tumpukan vertikal) tanpa elemen yang terpotong.	Antarmuka bersifat responsif sesuai dengan konfigurasi utility class Tailwind CSS.	Valid
3	CRUD Template Gratis	Memilih opsi penyaringan data (filter) dan menekan tombol untuk membuka Modal CRUD Hashtag.	Opsi filter merespons pilihan pengguna dan komponen Modal muncul menutupi layar dengan efek transisi <i>overlay</i> .	Interaksi Dropdown dan Modal beroperasi lancar tanpa <i>full-page reload</i> .	Valid
4	CRUD Template Gratis	Mengunggah berkas/gambar melalui komponen Custom File Input.	Komponen memperbarui tampilannya secara reaktif untuk menunjukkan nama file yang dipilih atau pratinjau (<i>preview</i>).	Pratinjau berkas berhasil ditampilkan sebelum proses pengiriman data.	Valid
5	CRUD Berita	Membuka Modal CRUD Jenis Berita, kemudian menekan pemicu untuk membuka Modal Status di atasnya.	Modal Status (Dialog) dirender dengan urutan indeks-Z (<i>Z-index</i>) yang benar tanpa merusak status komponen di bawahnya.	Manajemen state antarmuka berhasil mengisolasi status masing-masing Modal.	Valid

Berdasarkan hasil pengujian yang disajikan pada Tabel 1, pengujian *Functional Suitability* menunjukkan tingkat keberhasilan sebesar 100% dari total lima skenario yang dijalankan. Seluruh interaksi kompleks pengguna berhasil ditangani secara mandiri dan presisi melalui manajemen status lokal (*local state management*). Selain itu, responsivitas *layout* dapat beradaptasi dengan baik pada resolusi layar perangkat mobile berkat konfigurasi utility class Tailwind CSS yang terstruktur.

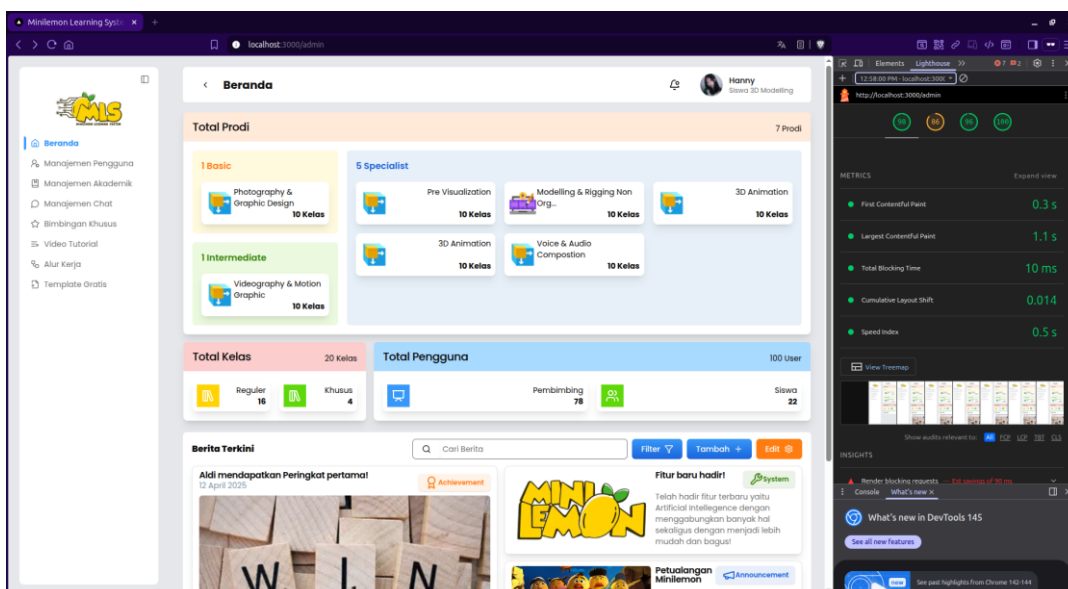
Keberhasilan validasi sisi klien dalam menahan input yang tidak valid juga membuktikan bahwa lapisan antarmuka ini mampu meminimalisasi beban pemrosesan (*request payload*) yang tidak perlu ke server. Sintesis dari hasil pengujian ini menegaskan bahwa antarmuka *Minilemon Learning System* telah memenuhi kelayakan standar kesesuaian fungsional tingkat purwarupa, terhindar dari cacat fungsional visual, dan secara teknis sepenuhnya siap untuk dilanjutkan ke tahap integrasi melalui *API (Application Programming Interface)* dengan sistem *backend*.

3.3 Evaluasi *Performance Efficiency*

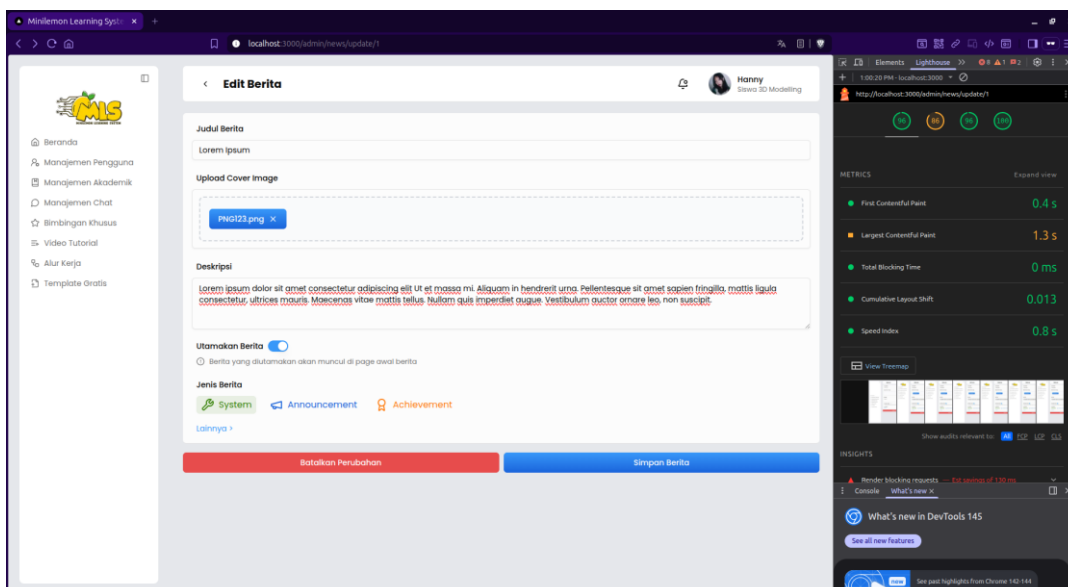
Karakteristik *Performance Efficiency* dievaluasi untuk mengukur sejauh mana antarmuka sistem meminimalkan penggunaan sumber daya peramban (browser) dan memberikan waktu respons yang cepat kepada pengguna. Mengingat arsitektur yang diuji murni berada di sisi klien (*client-side*), matrik evaluasi difokuskan pada kecepatan rendering visual dan efisiensi pemuatan aset *frontend*. Pengujian ini dilakukan secara otomatis menggunakan instrumen Google Lighthouse pada lingkungan production build Next.js.



Gambar 5. Hasil Audit Metrik Performance Halaman Login (Siswa) Menggunakan Google Lighthouse



Gambar 6. Hasil Audit Metrik Performance Halaman Dashboard (Admin) Menggunakan Google Lighthouse



Gambar 7. Hasil Audit Metrik Performance Halaman Edit Berita Menggunakan Google Lighthouse

Gambar 5 hingga Gambar 7 menunjukkan ringkasan skor performa sampel antarmuka *Minilemon Learning System*. Untuk membedah kualitas efisiensi tersebut secara lebih teknis, data empiris yang dihasilkan oleh Lighthouse diuraikan ke dalam parameter *Core Web Vitals* pada Tabel 3.

Tabel 2. Rincian Metrik Performance Efficiency Berdasarkan Sampel Halaman

No	Metrik Pengukuran	Standar Ideal	Halaman Login (Siswa)	Halaman Dashboard (Admin)	Halaman Edit Berita	Rata-Rata	Kesimpulan
1	First Contentful Paint (FCP)	< 1,8 s	0,3 s	0,3 s	0,4 s	0,33 s	Sangat Baik
2	Largest Contentful Paint (LCP)	< 2,5 s	0,8 s	1,1 s	1,3 s	1,06 s	Sangat Baik
3	Speed Index	< 3,4 s	0,3 s	0,5 s	0,8 s	0,53 s	Sangat Baik
4	Total Blocking Time (TBT)	< 200 ms	0 ms	10 ms	0 ms	3,33 ms	Sangat Baik
5	Cumulative Layout Shift (CLS)	< 0.1	0	0,014	0,013	0,009	Sangat Baik

Berdasarkan data kuantitatif pada Tabel 2, antarmuka *Minilemon Learning System* terbukti memiliki efisiensi kinerja yang optimal. Nilai *First Contentful Paint (FCP)* dan *Largest Contentful Paint (LCP)* rata-rata berada jauh di bawah ambang batas standar ideal. Hal ini menunjukkan bahwa pengguna tidak mengalami waktu tunggu (latensi) yang berarti saat memuat elemen visual utama. Selain itu, metrik *Total Blocking Time (TBT)* yang rendah (rata-rata 3,33 ms) membuktikan bahwa interaksi pengguna tidak terhambat oleh proses JavaScript di latar belakang. Angka *Cumulative Layout Shift (CLS)* yang mendekati nol juga mengkonfirmasi stabilitas visual antarmuka yang tinggi; tidak terjadi pergeseran komponen tak terduga yang sering kali mengganggu User Experience (UX) selama proses rendering.

Tingginya metrik efisiensi ini dapat diatribusikan secara langsung pada penerapan arsitektur *frontend* yang tepat. Kerangka kerja Next.js memfasilitasi optimasi bawaan seperti pemuatan pra-render (*pre-rendering*) dan pemecahan kode otomatis (*automatic code splitting*), sehingga peramban hanya memuat pustaka JavaScript yang relevan untuk halaman yang sedang aktif. Dipadukan dengan pendekatan *Component-Based Architecture*, antarmuka terhindar dari pembengkakan elemen HTML. Hasil pengujian ini memvalidasi bahwa rekayasa antarmuka *Minilemon Learning System* telah sangat siap untuk diimplementasikan ke tahap produksi (*production-ready*) setelah integrasi *backend* dengan jaminan performa antarmuka yang tinggi.

3.4 Evaluasi Usability

Evaluasi kualitas antarmuka yang ketiga didasarkan pada karakteristik *Usability* berdasarkan standar ISO/IEC 25010, yang difokuskan secara spesifik pada sub-karakteristik Aksesibilitas (*Accessibility*). Karakteristik ini mengukur sejauh mana antarmuka sistem dapat dikenali, dioperasikan, dan dipahami oleh pengguna dengan berbagai tingkat kemampuan, termasuk pemenuhan standar bagi pengguna yang bergantung pada teknologi bantu (*assistive technologies*) seperti pembaca layar (*screen reader*). Data diekstraksi secara serentak bersamaan dengan pengujian performa menggunakan instrumen Google Lighthouse.

Berdasarkan hasil audit yang telah tertera pada Gambar 5 hingga Gambar 7 sebelumnya, metrik Aksesibilitas menunjukkan skor yang sangat memadai. Halaman Login (Siswa) meraih skor 92/100, sementara Halaman Dasbor (Admin) dan Halaman Edit Berita masing-masing meraih skor 86/100. Tingginya skor ini mengindikasikan bahwa implementasi *styling* menggunakan Tailwind CSS tidak sekadar mengejar presisi estetika dari Figma, melainkan juga mematuhi kaidah web semantik. Indikator keberhasilan aksesibilitas pada kode *frontend* ini meliputi rasio kontras warna yang telah memenuhi batas minimum standar *Web Content Accessibility Guidelines (WCAG)* [16], penggunaan *tag* HTML terstruktur pada komponen form dan card, serta penyematan pelabelan atribut ARIA pada elemen interaktif sehingga fungsi antarmuka dapat dideskripsikan dengan baik secara sistem.

3.5 Analisis dan Pembahasan Keseluruhan

Merujuk pada serangkaian evaluasi yang telah dilakukan, dapat dianalisis bahwa adopsi *Component-Based Architecture (CBA)* menggunakan kerangka kerja Next.js merupakan solusi teknis yang efektif untuk mengatasi permasalahan redundansi kode dan inefisiensi performa pada pengembangan *Learning Management System (LMS)* berskala besar.

Hasil pengujian *Functional Suitability* membuktikan bahwa pendekatan arsitektur komponen mampu menangani manajemen *local state* dengan tingkat keberhasilan 100%, memvalidasi input secara ketat di sisi klien, dan menampilkan interaksi komponen multi-layer secara presisi tanpa memerlukan integrasi *backend*. Secara bersamaan, hasil pengujian *Performance Efficiency* dan *Usability* menggunakan metrik standar peramban menunjukkan skor *Core Web Vitals* yang mendekati sempurna (rata-rata kecepatan muat *FCP* di bawah 0,5 detik). Hal ini mematahkan kelemahan metode *Client-Side Rendering (CSR)* murni yang sering kali lambat, karena Next.js berhasil mendistribusikan beban rendering melalui mekanisme *code splitting*. Secara keseluruhan, rekayasa antarmuka *Minilemon Learning System* terbukti bebas dari *technical debt*, mudah dipelihara, dan sepenuhnya siap untuk memasuki fase integrasi *Application Programming Interface (API)*.

4. KESIMPULAN

Penelitian ini telah berhasil mengimplementasikan dan mengevaluasi purwarupa antarmuka *Minilemon Learning System* menggunakan pendekatan *Component-Based Architecture (CBA)* berbasis Next.js. Berdasarkan pengujian kualitas perangkat lunak standar ISO/IEC 25010, hasil evaluasi pada aspek *Functional Suitability* membuktikan bahwa antarmuka mampu merespons interaksi pengguna dengan tingkat keberhasilan 100%. Manajemen *local state* terbukti secara mandiri mampu menangani validasi sisi klien dan interaksi komponen multi-lapis tanpa bergantung pada integrasi *backend*. Selanjutnya, pada aspek *Performance Efficiency*, implementasi Next.js terbukti mampu menggunakan sumber daya peramban secara optimal. Hal ini ditunjukkan oleh hasil pengukuran parameter *Core Web Vitals* yang mencatatkan kecepatan rendering visual instan, dengan rata-rata waktu *First Contentful Paint (FCP)* sebesar 0,33 detik dan *Total Blocking Time (TBT)* hanya 3,33 milidetik. Selain itu, pada aspek *Usability* yang difokuskan pada aksesibilitas, antarmuka berhasil mencapai skor rata-rata di atas 86/100, yang memvalidasi bahwa desain arsitektur yang dibangun tidak hanya estetis, tetapi juga inklusif dan mematuhi kaidah *Web Content Accessibility Guidelines (WCAG)*.

Secara keseluruhan, pemecahan elemen antarmuka menjadi komponen yang reusable terbukti efektif dalam mengeliminasi redundansi kode dan mencegah pembengkakan pemuatan dokumen (*DOM bloat*). Produk frontend yang dihasilkan dari penelitian ini mudah dipelihara, terhindar dari *technical debt*, dan berada dalam kondisi siap untuk dilanjutkan ke fase integrasi *API (Application Programming Interface)* pada pengembangan sistem di masa mendatang.

REFERENCES

- [1] W. Leal Filho *et al.*, "Digital transformation and sustainable development in higher education in a post-pandemic world," *Int. J. Sustain. Dev. World Ecol.*, vol. 31, no. 1, pp. 108–123, 2024, doi: 10.1080/13504509.2023.2237933.
- [2] I. G. N. WIRAGUNAWAN, "Pemanfaatan Learning Management System (Lms) Dalam Pengelolaan Pembelajaran Daring Pada Satuan Pendidikan," *EDUTECH J. Inov. Pendidik. Berbantuan Teknol.*, vol. 2, no. 1, pp. 82–89, 2022.
- [3] Z. Hafizh, W. Hayuhardhika Nugraha Putra, and H. Muslimah Azzahra, "Perancangan User Experience Sistem Konversi Mata Kuliah Dan Pengakuan SKS Pada Kegiatan Mbkm Menggunakan Human-Centered Design," *J. Sist. Informasi, Teknol. Informasi, dan Edukasi Sist. Inf.*, vol. 5, no. 2, pp. 75–88, 2024, doi: 10.25126/justsi.v5i2.420.
- [4] S. Gangishetti and V. Jain, "Comparative Analysis of Client-Side vs . Server-Side Rendering for Large-Scale Content Platforms," vol. 7, no. 1, pp. 74–80, 2026.
- [5] D. Pina, C. Seaman, and A. Goldman, "Technical Debt Prioritization: A Developer's Perspective," *Proc. - Int. Conf. Tech. Debt 2022, TechDebt 2022*, no. May 2022, pp. 46–55, 2024, doi: 10.1145/3524843.3528096.
- [6] Y. Yang, "Web Front-End Application Performance Improvement Method Based on Component-Based Architecture," *Int. J. Eng. Adv.*, vol. 2, no. 2, p. 24, 2025, [Online]. Available: <https://doi.org/10.71222/sg9eef87>
- [7] A. Nugroho, K. Prihandani, and R. Mayasari, "Rancang Bangun Sistem Pembelian E-Ticket Berbasis Website Dengan Konsep Server-Side Rendering Menggunakan Framework Next Js Pada Wisata Telaga Kusuma Jumantono," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 8, no. 4, pp. 5771–5777, 2024, doi: 10.36040/jati.v8i4.9960.
- [8] M. Martha Auliasari and A. Dian Pratama, "Efektivitas e-Learning pada Pendidikan Tinggi Dengan Menggunakan Learning Management System (Moodle dan Google Classroom)," *J. Inov. Akad. | 43 J. Inov. Akad.*, vol. 2, no. 1, pp. 43–53, 2024.
- [9] A. R. Novianto and S. Rani, "Pengembangan Desain UI/UX Aplikasi Learning Management System dengan Pendekatan User Centered Design," *J. Sains, Nalar, dan Apl. Teknol. Inf.*, vol. 2, no. 1, 2022, doi:

- 10.20885/snati.v2i1.16.
- [10] M. A. Ramadhan, M. K. Sabariah, and S. Y. P. Fakultas, "Pengembangan Frontend Sistem Informasi Layanan Tekos Berbasis Web Menggunakan Framework NextJS," *e-Proceeding Eng.*, vol. 11, no. 4, pp. 5138–5143, 2024.
 - [11] S. Aji, Y. Mulyadi, and I. Kustiawan, "Implementation of Learning Management System (LMS) in the Self-Development of Productive Teachers of Vocational Schools," *Proc. 4th Int. Conf. Innov. Eng. Vocat. Educ. (ICIEVE 2021)*, vol. 651, no. Icieve 2021, pp. 14–18, 2022, doi: 10.2991/assehr.k.220305.004.
 - [12] ISO/IEC 25010, "ISO/IEC Systems and Software Engineering — Requirements and Evaluation," *Int. Organ. Stand. IEC*, vol. 2011, 2011.
 - [13] Nursahiba and Depandi Enda, "Pengukuran Kualitas Perangkat Lunak Dengan Standar ISO/IEC 25010 Pada Website Jurusan Teknik Informatika Politeknik Negeri Bengkalis," *JEKIN - J. Tek. Inform.*, vol. 5, no. 1, pp. 13–23, 2025, doi: 10.58794/jekin.v5i1.839.
 - [14] V. Dubey and S. Verma, "Evaluating The Digital Performance and Accessibility of IIT Library Websites Using Google Lighthouse," *Int. J. Comput. Sci. Eng.*, vol. 13, no. 3, pp. 16–23, 2025, doi: 10.26438/ijcse/v13i3.1623.
 - [15] D. R. Dayan, D. F. Wati, and M. G. Anggalana, "Implementasi Pengujian Black Box Dengan Teknik Equivalence," vol. 10, no. 1, pp. 326–332, 2026.
 - [16] K. R. Putra, S. Umaroh, and N. Rafina, "Web Accessibility Evaluation Using WCAG 2.0 and User-Centered Design Method," *E3S Web Conf.*, vol. 484, 2024, doi: 10.1051/e3sconf/202448402002.