Jurnal Ilmiah MEDIA SISFO
Vol. 19, No. 2, Oktober 2025

ISSN: 1978-8126
e-ISSN: 2527-7340

Optimasi Kecepatan dan Responsivitas Aplikasi Presensi Perkuliahan Berbasis Web melalui Arsitektur Single Page Application

Ali Sadikin[1], Abdul Rahim[2], Agus Siswanto[3],Muhammad Wardani[4], Ricky Fajar Adiputra[5]*

Sistem Informasi, Fakultas Ilmu Komputer, Universitas Dinamika Bangsa [1,4]
Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dinamika Bangsa [2,3]
Jalan Jendral Sudirman Thehok, Jambi, Indonesia
nikidasilva@gmail.com[1], abdulrahim@unama.ac.id[2], agussiswanto@unama.ac.id[3], ardan@unama.ac.id[4], rickyyla@gmail.com[5]

Submitted : **04/09/2025**; Reviewed : **17/09/2025**; Accepted : **29/10/2025**; Published : **31/10/2025**

Abstract

Web-based lecture attendance applications that use Multi-Page Application (MPA) architecture often face speed and responsiveness issues when the number of users increases. This study aims to optimize the attendance application at Universitas Dinamika Bangsa (UNAMA) by implementing Single Page Application (SPA) architecture to improve system access speed and responsiveness. Using the waterfall development method, the results of this study show that SPA implementation is significantly more efficient than Multi-page Application (MPA) architecture in several aspects, including the number of Requests (RC), Transferred File Size (TF), Resources (RS), and Request Duration (RD). For example, the "Home" page on MPA requires 51 requests with a total transfer of 741 kB and a duration of 945 ms, while on SPA it only requires 1 request with a transfer of 5.9 kB and a duration of 187 ms. This proves that the SPA architecture successfully reduces network load and speeds up response time, resulting in a more responsive user experience and efficient use of resources.

Keywords : web optimization; application; attendance; lecture; single page application

Abstrak− Aplikasi presensi perkuliahan berbasis web yang menggunakan arsitektur Multi-page Application (MPA) sering menghadapi kendala kecepatan dan responsivitas saat jumlah pengguna meningkat. Penelitian ini bertujuan untuk mengoptimalkan aplikasi presensi di Universitas Dinamika Bangsa (UNAMA) dengan menerapkan arsitektur Single Page Application (SPA) untuk meningkatkan kecepatan akses dan responsivitas sistem. Dengan menggunakan metode pengembangan waterfall , hasil penelitian menunjukkan bahwa implementasi SPA secara signifikan lebih efisien dibandingkan dengan arsitektur MPA dalam beberapa aspek, termasuk jumlah Request (RC), Transferred File Size (TF), Resources (RS), dan Request Duration (RD). Sebagai contoh, halaman "Beranda" pada MPA membutuhkan 51 request dengan total transfer 741 kB dan durasi 945 ms, sedangkan pada SPA hanya membutuhkan 1 request dengan transfer 5.9 kB dan durasi 187 ms. Hal ini membuktikan bahwa arsitektur SPA berhasil mengurangi beban jaringan dan mempercepat waktu respon, sehingga menghasilkan pengalaman pengguna yang lebih responsif dan efisien dalam penggunaan sumber daya.

Kata kunci : optimalisasi web; aplikasi; presensi; single page application

1. Pendahuluan

Dalam era digital, sistem informasi berbasis web semakin banyak digunakan untuk meningkatkan efisiensi dan efektivitas layanan di berbagai sektor, termasuk dunia pendidikan. Salah satu implementasi teknologi web yang umum digunakan dalam institusi akademik adalah aplikasi presensi perkuliahan berbasis web. Sistem Presensi memungkinkan mahasiswa melakukan presensi melalui sistem tanpa memerlukan pencatatan manual oleh dosen atau staf administrasi . Namun, dengan meningkatnya jumlah pengguna, aplikasi berbasis Multi-Page Application (MPA) sering mengalami kendala dalam hal kecepatan akses dan responsivitas akibat tingginya permintaan ke server setiap kali halaman dimuat ulang [1], [2].

Arsitektur Single Page Application (SPA) dapat digunakan sebagai solusi untuk meningkatkan efisiensi dan pengalaman pengguna dalam sistem berbasis web [3]. SPA memungkinkan seluruh pemrosesan dilakukan di sisi klien setelah halaman pertama dimuat, sehingga mengurangi jumlah permintaan ke server dan mempercepat waktu respon aplikasi [4], [5]. Laravel Livewire, sebagai salah satu teknologi berbasis PHP, menawarkan pendekatan modern dalam pengembangan SPA [6], [7] dengan

ISSN: 1978-8126
e-ISSN: 2527-7340

mengintegrasikan komunikasi antara frontend dan backend baik secara realtime ataupun non realtime. Dengan menggunakan Laravel Livewire, sistem dapat merespons interaksi pengguna dengan lebih cepat tanpa perlu melakukan pemuatan ulang halaman, sehingga meningkatkan pengalaman pengguna dan mengurangi beban server.

Beberapa penelitian sebelumnya menunjukkan bahwa penerapan SPA dapat meningkatkan performa sistem berbasis web secara signifikan. Penelitian yang dilakukan oleh Purnama et al, telah menerapkan sistem informasi berbasis Single Page Application (SPA) dalam pengelolaan Zakat, Infaq, dan Sadaqah (ZIS) di DKM Masjid Riyadhul Jannah, Kota Bandung. Sistem ini dikembangkan untuk mengatasi permasalahan yang muncul akibat pengelolaan ZIS secara manual, seperti rendahnya efektivitas pengawasan keuangan, ketidakefisienan dalam administrasi, serta risiko kehilangan data penting. Dengan menggunakan SPA yang terintegrasi dengan RESTful API, sistem ini mampu meningkatkan transparansi dan akuntabilitas dalam pencatatan serta distribusi dana ZIS. Keunggulan utama dari SPA adalah kemampuannya dalam memberikan pengalaman pengguna yang lebih interaktif dan responsif dibandingkan aplikasi web tradisional, serta memastikan proses pertukaran data yang efisien dan terstandarisasi. Diharapkan, implementasi sistem ini dapat memperkuat peran sosial dan spiritual DKM, meningkatkan keterlibatan jamaah, serta mendukung pengelolaan ZIS yang lebih profesional dan terpercaya [8].

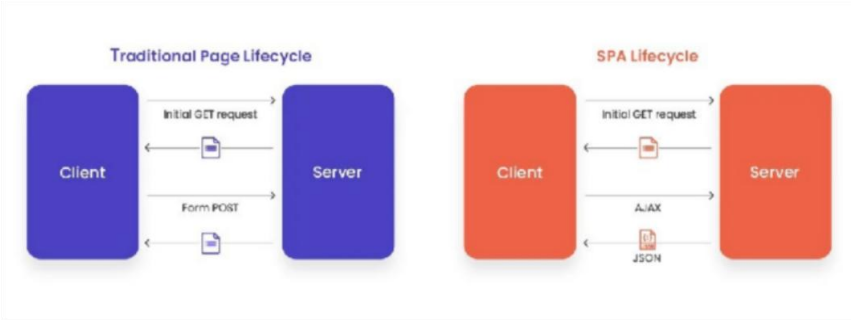
Penelitian yang dilakukan oleh Atifah et al, telah menerapkan Single Page Application (SPA) pada pengembangan aplikasi e-learning Nusabot, yang berfokus pada bidang Internet of Things (IoT). Dalam sistem sebelumnya, interaksi antara admin dan instruktur masih dilakukan secara manual, terutama dalam pengelolaan kelas dan kurikulum, yang menyebabkan ketidakefisienan dan seringnya terjadi pemuatan ulang halaman saat terjadi perubahan data. Untuk mengatasi masalah ini, penelitian ini mengembangkan sistem menggunakan SPA dengan React JS sebagai front-end, Laravel sebagai back-end dan API services, serta Inertia JS sebagai penghubung antara Laravel dan React JS. Metode pengembangan yang digunakan adalah model Prototype, yang menekankan keterlibatan pengguna dalam proses pengembangan. Implementasi SPA dalam sistem ini berhasil meningkatkan efisiensi dan efektivitas dalam pengelolaan kelas, memungkinkan instruktur untuk menyusun dan mengajukan kurikulum secara langsung dalam satu halaman tanpa perlu memuat ulang, serta memberikan pengalaman pengguna yang lebih interaktif dan responsif [9].

Penelitian ini bertujuan untuk menerapkan arsitektur SPA menggunakan Laravel Livewire pada aplikasi presensi kuliah guna meningkatkan kecepatan akses dan responsivitas sistem. Implementasi ini akan diuji dengan skenario penggunaan, termasuk analisis performa berdasarkan metrik seperti response time, server request load, dan time to interactive. Dengan adanya solusi ini, diharapkan sistem presensi kuliah dapat berjalan lebih efisien, memberikan pengalaman pengguna yang lebih baik, serta mampu menangani jumlah pengguna yang lebih besar tanpa mengalami degradasi performa.

Arsitektur *Multi-Page Application* (MPA) merupakan pendekatan konvensional dalam pengembangan aplikasi web. Model ini didasarkan pada siklus permintaan-respons (*request-response*) yang sederhana antara klien (peramban) dan server. Setiap kali pengguna melakukan tindakan yang memerlukan data atau tampilan baru—seperti mengklik tautan navigasi atau mengirimkan data melalui formulir—peramban akan mengirimkan permintaan HTTP penuh ke server. Server kemudian memproses permintaan tersebut, merender (*render*) halaman HTML yang baru secara keseluruhan di sisi server (*server-side rendering*), dan mengirimkannya kembali ke peramban untuk ditampilkan [10], [11].

Bismoputro et al [12] menggambarkan *Single Page Application* (SPA) sebagai salah satu jenis aplikasi daring yang memuat semua sumber daya yang dibutuhkan (*HTML, CSS, JavaScript*, dll.) saat pengguna pertama kali mengunjungi halaman tersebut. Saat pengguna berinteraksi dengan aplikasi, konten diperbarui secara dinamis tanpa memerlukan pemuatan ulang halaman. Menurut Wibowo dan Wiguna [13], *single page application* merupakan teknologi yang beroperasi di dalam browser tanpa memerlukan pemuatan ulang halaman.

ISSN: 1978-8126
e-ISSN: 2527-7340

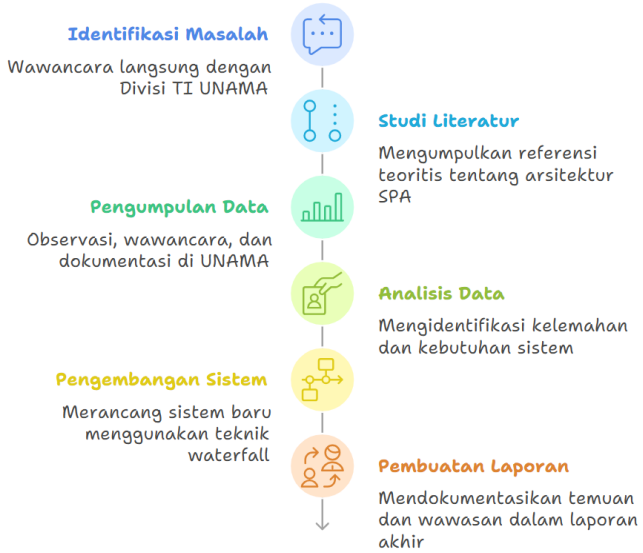


Gambar 1. *SPA vs MPA* [14]

Pada *Multi-page Application*, setiap kali user mengklik link atau mengirimkan form, browser akan melakukan *full page reload* untuk menampilkan halaman baru, yang melibatkan *request* dan *response* dari server. Berbeda dengan *Single Page Application Lifecycle* yang hanya memuat halaman awal satu kali, setelah itu aplikasi berjalan di *client-side*, dan perubahan yang terjadi di halaman dilakukan melalui AJAX, yang mengirimkan request kecil untuk mendapatkan data yang diperlukan dari server. Data yang diterima dari server kemudian digunakan untuk memperbarui tampilan tanpa harus memuat seluruh halaman. Hal ini membuat SPA lebih responsif dan interaktif.

2. Metodologi

Langkah-langkah yang diselesaikan selama investigasi dikenal sebagai teknik kerangka kerja penelitian. Tujuan dari kerangka kerja penelitian adalah untuk memungkinkan tercapainya hasil penelitian, memastikan penyelesaian studi tepat waktu, dan memungkinkan studi berjalan sesuai rencana. Gambar 1 menampilkan kerangka kerja penelitian yang digunakan:



Gambar 2 *Metodologi Penelitian*

Pembahasan setiap langkah penelitian dapat dikarakterisasikan sebagai berikut, berdasarkan kerangka penelitian yang digambarkan pada Gambar 2:

1. Identifikasi Masalah
   Penulis mencatat dan mendokumentasikan berbagai masalah terkait kehadiran di perkuliahan di UNAMA selama fase ini. Untuk mengidentifikasi tantangan dalam memunculkan konsep dan solusi yang sesuai terkait kehadiran di perkuliahan, penulis melakukan wawancara langsung dengan Divisi TI UNAMA. Fase ini sangat penting karena akan sulit melakukan penelitian pada langkah berikutnya jika masalah tersebut tidak teridentifikasi.
2. Studi Literatur
   Pada tahap ini, penulis melakukan sejumlah tugas yang berkaitan dengan proses pengumpulan data pustaka, membaca dan mencatat, serta mengawasi tahap penelitian. Penulis mencari

ISSN: 1978-8126
e-ISSN: 2527-7340

referensi teoritis yang relevan dengan kasus atau isu yang diidentifikasi di perpustakaan dan internet, khususnya terkait dengan Optimasi Aplikasi Kehadiran Kuliah Menggunakan *Single Page Application Architecture.*

3. Pengumpulan Data
   Prosedur penelitian untuk mengumpulkan data merupakan salah satu unsur terpenting dalam penelitian. Kesalahan dalam prosedur pengumpulan data akan mempersulit analisis. Selain itu, jika pengumpulan data tidak dilakukan dengan benar, hasil dan simpulan akan rusak. Untuk itu, penulis melakukan kegiatan pengumpulan data, dimulai dari observasi, wawancara, dan dokumentasi, yang akan diuraikan sebagai berikut:
   a. Pengamatan (*Observation*)
      Metode pengumpulan data dilaksanakan dengan cara mengamati secara langsung dilakukan di Universitas Dinamika Bangsa mengenai hal-hal yang bersangkutan dengan presensi perkuliahan.
   b. Wawancara (*Interview*)
      Wawancara adalah teknik pengumpulan data yang dilakukan dengan cara bertanya secara langsung kepada Divisi IT UNAMA data yang diperoleh berupa data primer, yaitu data utama yang dibutuhkan dalam pengembangan sistem.
   c. Dokumentasi (*Documentation*)
      Dokumentasi dilakukan untuk menyediakan berbagai macam dokumen, salah satu caranya adalah dengan menggunakan bukti yang akurat dari Universitas Dinamika Bangsa dengan cara mencatat, memfotocopy berkas yang berhubungan dengan presensi perkuliahan. Dengan kata lain dokumentasi secara umum adalah suatu kegiatan untuk melakukan pencarian, penyelidikan, pengumpulan, dan penyediaan dokumen.
4. Analisis Data
   Dengan menganalisis data yang dikumpulkan untuk sepenuhnya memahami sistem saat ini, menentukan kekuatan dan kelemahannya, memastikan kebutuhan dan persyaratan sistem baru, dan merancang sistem baru yang memenuhi kebutuhan dan persyaratan ini, analisis ini berupaya mengidentifikasi kekurangan sistem serta cara untuk meningkatkan kinerja dan efisiensi sistem.
5. Pengembangan Sistem
   Dengan mempertimbangkan cara memaksimalkan sistem kehadiran kuliah dari tahap perencanaan, sketsa, dan penggambaran menjadi unit yang komprehensif dan fungsional, penulis memiliki gambaran yang jelas tentang apa yang harus dilakukan setelah tahap analisis data selesai. Teknik *waterfall*, yang populer dalam bisnis perangkat lunak dan sering digunakan oleh analis sistem secara umum karena prosedurnya yang berurutan dari analisis hingga pemeliharaan, akan digunakan untuk merancang tahap ini.
6. Pembuatan Laporan Hasil Penelitian
   Penulis membuat laporan penelitian pada tahap pelaporan, yang berfungsi sebagai dokumentasi yang berguna untuk penelitian, dengan menyusun temuan-temuan investigasi menjadi laporan penelitian, yang mencakup semuanya mulai dari identifikasi masalah hingga tahap pembuatan sistem.

3. Hasil dan Pembahasan

Sistem yang sedang berjalan atau mekanisme untuk presensi pada SIAKAD Universitas Dinamika Bangsa (UNAMA) memanfaatkan teknologi untuk memudahkan pencatatan kehadiran mahasiswa secara online. Mahasiswa dapat melakukan presensi dengan memasukkan kode unik atau memindai *QR code* yang telah diberikan oleh dosen pada setiap pertemuan perkuliahan. Proses ini memungkinkan dosen untuk mengelola dan memonitor kehadiran mahasiswa secara langsung, tanpa perlu mencatat secara manual. Selain itu, sistem ini juga menyediakan fitur bagi mahasiswa untuk mengajukan izin apabila tidak dapat hadir dalam perkuliahan, dengan mengisi formulir pengajuan izin yang tersedia di platform SIAKAD. Dengan adanya mekanisme ini, pencatatan kehadiran menjadi lebih efisien, akurat, dan transparan, serta mendukung kelancaran administrasi akademik di UNAMA. Namun Sistem ini masih menggunakan pendekatan tradisional dalam siklus hidup halaman (page lifecycle). Platform yang ada saat ini masih mengandalkan model *Multi-page application*, di mana setiap kali pengguna melakukan interaksi, seperti memindai *QR code* untuk presensi atau mengajukan izin, sistem akan memuat ulang seluruh halaman untuk menampilkan data yang baru. Hal ini menyebabkan sistem lama mengirim ulang seluruh data dan berdampak pada performa server serta kecepatan akses *web*.

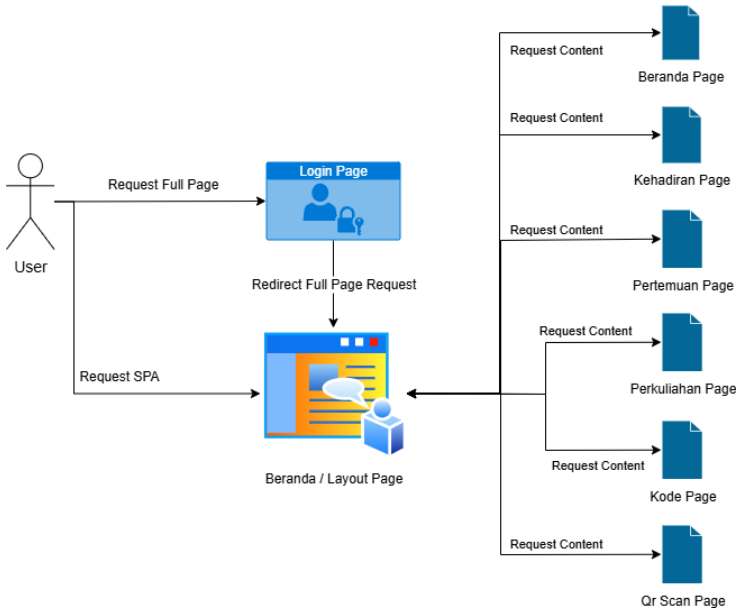
ISSN: 1978-8126
e-ISSN: 2527-7340

Tabel 1. *Penggunaan Multi-page Application*

Modul	*Request*	*Transferred*	*Resources*	*Request Duration*
Beranda	51	741 kB	5.0 MB	**945 ms**
Presensi Kehadiran	50	748 kB	5.0 MB	**687 ms**
Presensi Pertemuan	50	748 kB	5.0 MB	**602 ms**
Presensi Perkuliahan	53	747 kB	5.0 MB	**793 ms**
Presensi Kode	3	7.8 kB	5.0 MB	**750 ms**
Presensi QR Kode	20	735 kB	1.7 MB	**790 ms**
Pengajuan Izin	**4**	**8.3 kB**	**kB**	**760 ms**

3.1 Perancangan Arsitektur *Single Page Application*

Pada tahap ini, penulis melakukan perancangan arsitektur single page application.



Gambar 3 *Arsitektur Single Page Application*

Gambar 3 menampilkan arsitektur alur kerja yang diimplementasikan pada prototipe aplikasi presensi berbasis Single Page Application (SPA). Diagram pada gambar 4 membedakan secara fundamental antara paradigma permintaan-respons sinkron pada aplikasi multi-halaman konvensional dengan model interaksi asinkron yang menjadi landasan arsitektur SPA. Alur kerja ini dapat dianalisis melalui dua fase operasional utama: fase inisiasi sesi dan fase interaksi dinamis.

3.1.1 Fase Inisiasi Sesi: Pemuatan Aset Inti Aplikasi

Inisiasi sesi pengguna dengan aplikasi dapat terjadi melalui dua skenario yang berbeda, tergantung pada status autentikasi pengguna. Skenario pertama, dilambangkan dengan alur Request Full Page, merepresentasikan mekanisme pada Multi-Page Application (MPA). Pada alur ini, akses awal pengguna memicu permintaan halaman penuh ke server, yang direspons dengan dokumen HTML untuk halaman login. Pasca-autentikasi yang berhasil, server kembali mengirimkan dokumen HTML kedua untuk halaman utama melalui proses redirect. Mekanisme ini secara inheren melibatkan latensi yang disebabkan oleh dua siklus permintaan-respons halaman penuh. Sebaliknya, alur Request SPA merepresentasikan paradigma yang diadopsi dalam penelitian ini. Pada alur ini, server hanya mengirimkan satu dokumen HTML tunggal, yaitu Beranda / Layout Page, saat sesi pertama kali dimulai. Dokumen ini berfungsi sebagai cangkang (shell) aplikasi yang memuat seluruh aset inti, termasuk kerangka CSS dan pustaka JavaScript (React.js). Pemuatan tunggal ini secara signifikan mengurangi overhead pada pemuatan awal dan menjadi satu-satunya transfer dokumen HTML lengkap selama siklus hidup sesi pengguna.

ISSN: 1978-8126
e-ISSN: 2527-7340

3.1.2 Fase Interaksi Dinamis: Permintaan Konten Asinkron

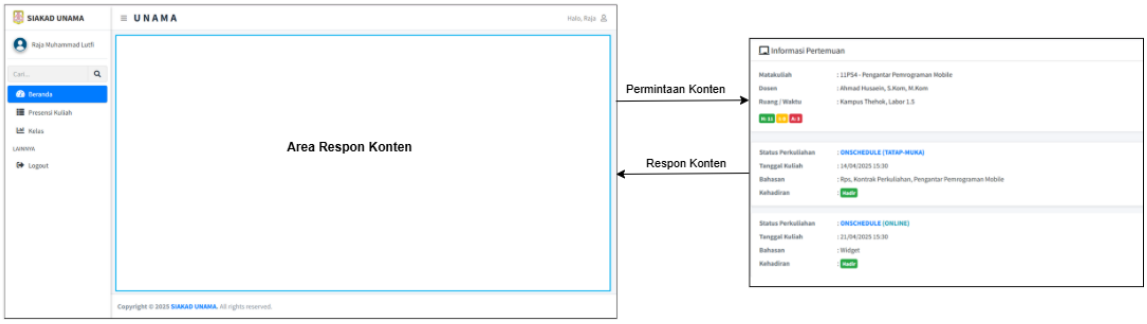
Setelah fase inisiasi selesai dan cangkang aplikasi telah di-render di peramban klien, seluruh interaksi pengguna selanjutnya ditangani secara asinkron. Sebagaimana diilustrasikan oleh alur Request Content, navigasi pengguna ke berbagai modul aplikasi—seperti Kehadiran Page, Pertemuan Page, atau Perkuliahan Page—tidak lagi memicu permintaan halaman baru ke server. Sebagai gantinya, interaksi tersebut ditangani oleh router di sisi klien (client-side router) yang kemudian menginisiasi permintaan data asinkron (melalui Fetch API atau AJAX) ke endpoint API yang relevan di server. Server tidak lagi merespons dengan HTML, melainkan dengan data murni dalam format JavaScript Object Notation (JSON). Pustaka JavaScript di sisi klien kemudian bertanggung jawab untuk memanipulasi Document Object Model (DOM) secara dinamis untuk me-render tampilan (view) baru dengan data yang diterima. Dengan demikian, halaman-halaman seperti Kehadiran Page dan lainnya bukanlah dokumen terpisah, melainkan komponen antarmuka yang di-render secara dinamis di dalam cangkang aplikasi yang sama.

3.1.3 Implikasi Arsitektural terhadap Kinerja

Arsitektur yang diilustrasikan pada Gambar 2 memiliki implikasi langsung terhadap optimasi kinerja. Dengan meminimalkan transfer data hanya pada informasi yang esensial dan mengeliminasi overhead dari pemuatan ulang halaman, latensi jaringan dapat ditekan secara signifikan. Hal ini tidak hanya mengurangi beban pada server, tetapi juga menghasilkan transisi antar muka yang lebih cepat dan responsif, sehingga secara positif meningkatkan user experience (UX) secara keseluruhan.

3.2 Implementasi

Pada tahap ini, penulis mengimplementasikan arsitektur SPA dan melakukan evaluasi. Implementasi arsitektur SPA dapat dilihat pada gambar berikut:



Gambar 4 *Implementasi SPA*

Gambar 4 menunjukkan cara kerja arsitektur Single Page Application (SPA) pada aplikasi presensi. Struktur utama di sebelah kiri adalah cangkang aplikasi yang berisi elemen tetap seperti header dan sidebar navigasi. Cangkang ini hanya dimuat satu kali saat aplikasi pertama kali dibuka. Di dalam cangkang, terdapat "Area Respon Konten" yang bersifat dinamis. Ketika pengguna mengklik menu, aplikasi tidak memuat ulang seluruh halaman. Sebaliknya, aplikasi hanya mengirim permintaan konten ke server. Server kemudian mengirimkan kembali respons berupa data (bukan halaman HTML), yang kemudian oleh aplikasi digunakan untuk memperbarui atau mengganti isi dari "Area Respon Konten" saja. Proses ini membuat transisi antar halaman terasa sangat cepat dan responsif, karena hanya bagian konten yang relevan yang diperbarui.

3.3 Hasil dan Evaluasi

Pada tahap ini penulis mengukur kinerja dari penerapan SPA dan membandingkan dengan arsitektur sebelumnya yang bisa dilihat di tabel berikut:

ISSN: 1978-8126
e-ISSN: 2527-7340

Tabel 2 *Perbandingan SPA dan MPA*

Modul	*Multiple Page Application*				*Single Page Application*			
	RC	TF	RS	RD	**RC**	**TF**	**RS**	**RD**
Beranda	51	741 kB	5.0 MB	945 ms	**1**	**5.9 kB**	**15.3 kB**	**187 ms**
Presensi Kehadiran	50	748 kB	5.0 MB	687 ms	**1**	**6.8 kB**	**25.8 kB**	**196 ms**
Kelas	-	-	-	-	**1**	**6.3 kB**	**18.2 kB**	**199 ms**
Presensi Pertemuan	50	748 kB	5.0 MB	602 ms	**1**	**8.9 kB**	**67.4 kB**	**187 ms**
Presensi Perkuliahan	53	747 kB	5.0 MB	793 ms	**1**	**6.7 kB**	**21.1 kB**	**168 ms**
Presensi Kode	4	7.8 kB	7.2 kB	750 ms	**1**	**2.3 kB**	**3.5 kB**	**197 ms**
Presensi QR Kode	20	735 kB	1.7 MB	790 ms	**2**	**5.2 kB**	**387 kB**	**189 ms**
Pengajuan Izin	4	8.3 kB	7.8 kB	760 ms	**1**	**2.5 kB**	**4.0 kB**	**149 ms**

Berdasarkan tabel 2, Single Page Application (SPA) lebih efisien dibandingkan dengan Multi-page Application (MPA) dalam beberapa aspek, termasuk jumlah Request (RC), Transferred File Size (TF), Resources (RS), dan Request Duration (RD). Pada MPA, jumlah RC dan ukuran file yang ditransfer TF lebih besar. Misalnya, pada halaman "Beranda", MPA membutuhkan 51 request dengan total ukuran file yang ditransfer sebesar 741 kB, dan memuat sekitar 5.0 MB dari resources. Hal ini mengindikasikan bahwa untuk setiap halaman, banyak elemen yang dimuat ulang, meningkatkan ukuran file yang perlu diunduh serta waktu yang dibutuhkan untuk memuat seluruh halaman. Durasi request (RD) pada MPA juga lebih lama, seperti pada halaman "Beranda" yang memerlukan 945 ms.

Sebaliknya, pada Single Page Application (SPA), jumlah request (RC) jauh lebih sedikit, dengan halaman "Beranda" hanya memerlukan 1 request. Ukuran file yang ditransfer (TF) juga lebih kecil, hanya sekitar 5.9 kB, serta jumlah resources (RS) yang lebih sedikit, yaitu 15.3 kB, dibandingkan dengan MPA. Durasi request (RD) di SPA juga lebih cepat, hanya 187 ms pada halaman "Beranda". Ini menunjukkan bahwa SPA hanya memuat bagian-bagian yang diperlukan secara dinamis, tanpa harus memuat ulang seluruh halaman, mengurangi beban jaringan dan mempercepat waktu respon, yang menghasilkan pengalaman pengguna yang lebih responsif dan efisien dalam penggunaan sumber daya.

4. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan dari pengguna *Single Page Application*, penulis menyatakan bahwa penggunaan SPA lebih efisien dibandingkan dengan MPA dalam beberapa aspek, termasuk jumlah *Request* (RC), *Transferred File Size* (TF), Resources (RS), dan Request Duration (RD). Pada MPA, jumlah *request* (RC) dan ukuran file yang ditransfer (TF) lebih besar. Misalnya, pada halaman "Beranda", MPA membutuhkan 51 request dengan total ukuran file yang ditransfer sebesar 741 kB, dan memuat sekitar 5.0 MB dari resources. Hal ini mengindikasikan bahwa untuk setiap halaman, banyak elemen yang dimuat ulang, meningkatkan ukuran file yang perlu diunduh serta waktu yang dibutuhkan untuk memuat seluruh halaman. Durasi *request* (RD) pada MPA juga lebih lama, seperti pada halaman "Beranda" yang memerlukan 945 ms. Sebaliknya, pada *Single Page Application* (SPA), jumlah *request* (RC) jauh lebih sedikit, dengan halaman "Beranda" hanya memerlukan 1 *request*. Ukuran file yang ditransfer (TF) juga lebih kecil, hanya sekitar 5.9 kB, serta jumlah *resources* (RS) yang lebih sedikit, yaitu 15.3 kB, dibandingkan dengan MPA. Durasi *request* (RD) di SPA juga lebih cepat, hanya 187 ms pada halaman "Beranda". Ini menunjukkan bahwa SPA hanya memuat bagian-bagian yang diperlukan secara dinamis, tanpa harus memuat ulang seluruh halaman, mengurangi beban jaringan dan mempercepat waktu respon, yang menghasilkan pengalaman pengguna yang lebih responsif dan efisien dalam penggunaan sumber daya.

Daftar Pustaka

[1] D. Hamdani, A. P. W. Wibowo, and H. Heryono, "Perancangan Sistem Presensi Online dengan QR Code Menggunakan Metode Prototyping," *J. Teknol. Dan Inf.*, vol. 14, no. 1, pp. 62–73, Mar. 2024, doi: 10.34010/jati.v14i1.11844.

[2] B. P. Ilham and F. Supriadi, "Pembatasan Jarak Presensi Dalam Sistem Presensi Daring Menggunakan Formula Haversine," vol. 14, 2024.

[3] A. S. Oktriwina, "Memahami Single Page Application, Keunggulan dan Kelemahannya," *Glints*, 2024, [Online]. Available: https://glints.com/id/lowongan/single-page-application/

ISSN: 1978-8126
e-ISSN: 2527-7340

[4] P. L. Lokapitasari Belluano, "Pengembangan Single Page Application Pada Sistem Informasi Akademik," *Ilk. J. Ilm.*, vol. 10, no. 1, pp. 38–43, Apr. 2018, doi: 10.33096/ilkom.v10i1.204.38-43.
[5] B. Sisephaputra *et al.*, *Buku Ajar Pemrograman Web*. PT. Sonpedia Publishing Indonesia, 2025.
[6] I. K. Natania, "Rancang Bangun Aplikasi Donor Darah Darurat Donora Berbasis Website Menggunakan Livewire Laravel," *JATISI J. Tek. Inform. Dan Sist. Inf.*, vol. 11, no. 3, 2024.
[7] F. P. E. Putra, R. W. Efendi, A. B. Tamam, and W. A. Pramadi, "Tren dan Praktik Terbaik dalam Pengembangan Web Berbasis API: Kajian Literatur terhadap Framework Laravel dan React," *Infomatek*, vol. 27, no. 1, pp. 165–178, 2025.
[8] A. Purnama, A. A. Rahman, E. Fauzi, B. A. Prasetyo, A. Nuryana, and H. Robani, "Optimasi Pengelolaan Zakat, Infaq, Dan Sadaqah (ZIS) Melalui Sistem Informasi Berbasis Single Page Application (SPA) Di DKM Masjid Riyadhul Jannah Ciwastra, Kota Bandung," *JAPI J. Akses Pengabdi. Indones.*, vol. 9, no. 1, pp. 9–18, Apr. 2024, doi: 10.33366/japi.v9i1.5775.
[9] S. N. Atifah, A. Primajaya, and D. Yusup, "Penerapan Single Page Application Pada Pengembangan Aplikasi E-Learning Nusabot," vol. 12, no. 1, 2024.
[10] J. Neagu, "Multi Page Application (MPA): A Good Business Fit," *Medium*, 2024, [Online]. Available: https://medium.com/@julianneagu/multi-page-application-mpa-a-good-business-fit-36029c7be9f0
[11] Pamungkas, M. S. B., Akbar, M. A., & Az-Zahra, H. M. (2023). Analisis Perbandingan Layout Form Single-Page dan Multi-Page dalam Pengisian Form Data Penduduk. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 7(1), 102–107.
[12] I. Bismoputro, F. Al Huda, and A. H. Brata, "Pengembangan Single Page Application Berbasis Reactjs Untuk Usaha Percetakan Online (Studi Kasus: Global Grafika)," *J. Pengemb. Teknol. Inf. Dan Ilmu Komput.*, vol. 8, no. 7, 2024.
[13] A. T. Wibowo, A. S. Wiguna, and others, "Pemanfaatan teknologi Single Page Application (SPA) dalam pembuatan aplikasi feedback dosen dari mahasiswa sebagai bentuk pengawasan lembaga terhadap kinerja dosen di bidang pengajaran," *SMARTICS J.*, vol. 5, no. 1, pp. 34–43, 2019.
[14] Lvivity, "Single-Page Application vs Multi-Page Application: Pros, Cons, and Which is Better?" Lvivity LLC, Dallas, TX, USA, 2025. [Online]. Available: https://lvivity.com/single-page-app-vs-multi-page-app