



## Penerapan Teknologi *Single Page Application* (SPA) pada Sistem Surat Disposisi Online (Sudion) di Universitas Muhammadiyah Prof Dr Hamka

Muhammad Fathan Aulia, Arry Avorizano

Program Studi Teknik Informatika Universitas Muhammadiyah Prof. Dr. Hamka, Indonesia  
[muhammadfathan23@gmail.com](mailto:muhammadfathan23@gmail.com) , [avorizano@uhamka.ac.id](mailto:avorizano@uhamka.ac.id)

### ABSTRAK

Sistem surat disposisi *online* (SUDION) di Universitas Muhammadiyah Prof. DR. HAMKA membutuhkan penanganan untuk mengatasi masalah performa. Hasil wawancara yang sudah dilakukan menghasilkan beberapa catatan penting yang perlu diperhatikan pada sistem SUDION. Hasil kuesioner menunjukkan 53,8% pengguna mengeluhkan lama waktu yang dibutuhkan untuk memuat data atau berpindah halaman, yaitu mencapai lebih dari 10 detik. Dari data tersebut menghasilkan fokus penelitian dengan melakukan penerapan teknologi *single page application* (SPA) sebagai langkah optimasi pada masalah performa. Penelitian ini membahas analisis, sprint dan *deployment* penerapan teknologi SPA. Metode yang digunakan merupakan pendekatan dari prinsip-prinsip yang dimiliki oleh metode *Agile*, yaitu menggunakan kerangka kerja *Scrum*. Pengujian sistem dilakukan menggunakan metode *blackbox* testing dengan cara pengujian secara fungsional. Pada akhir penelitian Penelitian menghitung uji efektivitas menggunakan metode skala likert dan melakukan uji perbandingan antara sebelum dan sesudah penerapan teknologi SPA. Hasil uji efektivitas menunjukkan bahwa sistem SUDION memiliki tingkat efektivitas sebesar 91,15% dari 52 responden. Hasil uji perbandingan antara sebelum dan sesudah penerapan teknologi SPA menunjukkan masing-masing hasil dari setiap alat pengujian seperti *gtmetrix*, *pingdom*, *pagespeed insight* dan *web.dev*. Penelitian ini berhasil mengatasi masalah pada performa, sehingga dapat membantu penyelenggaraan pelayanan administrasi suratmenyurat dan disposisi di lingkungan kerja Universitas Muhammadiyah Prof. DR. HAMKA yang lebih baik.

**Kata kunci :** *single page application, sistem surat disposisi, scrum*

### ABSTRACT

The online disposition letter system (SUDION) at the University of Muhammadiyah Prof. DR. HAMKA requires further handling to overcome performance problems. The results of the interviews that have been carried out have produced several important notes regarding the SUDION system. The results of the questionnaire showed 53.8% of users complained about the length of time it took to load data or switch pages, which was more than 10 seconds. The focus of the research is to apply single-page application (SPA) technology as an optimization step on performance problems from these data. This study discusses the analysis, sprint, and deployment of SPA technology applications. The method used is an approach to the principles of the Agile method, namely using the Scrum framework. System testing is done using the Blackbox testing method by way of functional testing. At the end of the study, the authors calculated the effectiveness test using the Likert scale method and conducted a comparison test between before and after the application of SPA technology. The effectiveness test results show that the SUDION system has an effective level of 91.15% from 52 respondents. The comparison test results between before and after the application of SPA technology show the results of each test tool such as GT Metrix, Pingdom, page speed insight, and web. dev. This research has succeeded in overcoming problems in performance so that it can assist in the

*administration of correspondence and disposition services in the work environment of the University of Muhammadiyah Prof. DR. better HAMKA.*

**Keywords :** *single page application, disposition letter system, scrum*

## 1. PENDAHULUAN

Sistem surat disposisi online (SUDION) telah dioperasikan di Universitas Muhammadiyah Prof. DR. HAMKA, harapan kompleksitas pekerjaan pada setiap unit kerja dapat terbantu, sehingga menjadi lebih efektif dan efisien. Hasil uji coba menggunakan metode kuesioner mengenai penilaian waktu yang dibutuhkan oleh sistem untuk memuat data surat atau berpindah halaman pada sistem SUDION. Dari total 52 responden sebanyak 53,8% (28 orang) menyatakan lebih dari 10 detik, sebanyak 34,6% (18 orang) menyatakan 5 sampai 10 detik dan sebanyak 11,5% (6 orang) menyatakan 1 menit lebih, bahkan gagal. Berdasarkan data tersebut menghasilkan fokus pada penelitian ini, yaitu tentang bagaimana cara untuk melakukan optimasi pada masalah performa menggunakan teknologi *single page application* (SPA), sehingga performa bukan lagi kendala yang harus dikhawatirkan.

Teknologi SPA diterapkan pada *browser* dan tidak memerlukan *reload* pada halaman saat digunakan. Artinya pengguna tidak melakukan berpindah halaman, membuat permintaan ke *server* setiap kali terjadi interaksi dalam aplikasi. Perbedaan antara aplikasi berbasis SPA dan nonSPA adalah aplikasi SPA memuat satu halaman dari *server* dan mekanisme perutean yang biasanya berjalan di *server* berjalan di klien. Dengan teknologi SPA, beban pada *server* menjadi lebih ringan karena beban kerja server tidak seberat dengan teknologi Non-SPA, dan jarang sekali *server* otomatis *down* sehingga dapat menangani lebih banyak pelanggan (Wibowo & Wiguna, 2019).

Teknologi SPA yang akan digunakan adalah turbolinks, yaitu dapat bekerja dengan sangat baik pada aplikasi yang digunakan. Pada hasil akhir penelitian akan melihat perbandingan data antara sebelum-sesudah penerapan teknologi tersebut dan hasil penilaian end-user melalui perhitungan skala likert melalui kuesioner (*feedback*) selama menggunakan SUDION, sehingga menjadi kesimpulan berhasil-tidaknya penelitian ini. Dengan dilakukannya penelitian diharapkan untuk kedepannya sistem SUDION dapat memperlancar dan tidak menghambat administrasi surat-menyurat dan disposisi di lingkungan kerja Universitas Muhammadiyah Prof. DR. HAMKA.

## 2. LANDASAN TEORI

### 2.1. UNIFIED MODELLING LANGUAGE (UML)

*Unified Modelling Language* dan UML mendukung proses pembuatan model abstrak suatu sistem dalam bentuk diagram. Dalam metode pemodelan sistem, setiap model memberikan tampilan sistem yang berbeda, (Setiyani & Setiawan, 2021).

#### 1. Use Case Diagram

Use case diagram menjelaskan mengenai sistem yang akan dibangun berupa gambaran interaksi antar masing-

masing fungsi pada sistem serta interaksi pengguna dengan sistem. *Use case* adalah serangkaian tindakan terkait, baik otomatis maupun manual, dirancang untuk menyelesaikan tugas bisnis. Hubungan menggambarkan hubungan antara dua simbol jika digunakan (Setiyani & Setiawan, 2021).

#### 2. Activity Diagram

*Activity Diagram* untuk memberikan gambaran umum tentang proses bisnis, langkah-langkah dalam kasus penggunaan, dan metode objek (Setiyani & Setiawan, 2021).

#### 3. Class Diagram

*Class diagram* digunakan untuk mendefinisikan struktur dari sebuah objek yang disebut kelas yang membentuk sistem dan juga hubungan antara kelas-kelas (Setiyani & Setiawan, 2021).

#### 4. Deployment Diagram

*Deployment diagram* memberikan deskripsi umum tentang arsitektur fisik perangkat lunak, perangkat keras dan sistem artefak. *Deployment diagram* dapat dianggap sebagai akhir ruang lingkup kasus penggunaan, yang mewakili bentuk fisik sistem, alih alih gambar konseptual interaksi antara pengguna dan perangkat dan sistem (Siregar & Melani, 2019).

### 2.2. SINGLE PAGE APPLICATION (SPA)

*Single Page Application* adalah teknologi yang berfungsi di *browser* dan tidak perlu memuat ulang halaman saat digunakan. Saat menggunakan teknologi SPA, pekerjaan server tidak lagi seberat penggunaan teknologi Non-SPA, sehingga beban server menjadi lebih ringan, server dapat menangani lebih banyak pelanggan, dan jarang memulai *server* secara otomatis (Wibowo & Wiguna, 2019).

### 2.3. WEBPACK

*Webpack* adalah alat proses manufaktur yang memeriksa semua modul tempat aplikasi bergantung, dan kemudian dengan cerdas mengintegrasikannya ke dalam sebuah paket, sehingga penyertaan modul-modul ini tidak terlalu monoton. *Webpack* memerlukan titik masuk aplikasi untuk mengetahui di mana menemukan *import*, dan memerlukan deklarasi modul. *Webpack* juga perlu mengetahui transformasi apa yang akan dilakukan dalam aplikasi. Hal terakhir yang dibutuhkan adalah lokasi untuk meletakkan paket yang baru dimodifikasi (Johansson, 2021).

### 2.4. SCRUM

*Agile software development* adalah metode untuk mengembangkan berbagai prinsip perangkat lunak di mana persyaratan dan solusi dilakukan melalui upaya kolaboratif tim dan pelanggan fungsional. Ini mendukung perencanaan

adaptif, pengembangan evolusi, pengiriman awal dan peningkatan berkelanjutan, dan mendorong tanggapan yang cepat dan fleksibel (Indah Puspitorini, 2021).

Scrum adalah metodologi yang memberikan lebih banyak orientasi kepada tim untuk melakukan apa yang ingin dilakukan, dan selalu siap menghadapi semua perubahan yang terjadi. Scrum adalah metode tangkas yang digunakan untuk mengembangkan produk dan layanan yang lebih inovatif. *Scrum* menekankan penggunaan skalabilitas, dan pentingnya menggunakan *Scrum* pada komputer yang diatur secara terpisah, dan kemudian dijelaskan oleh setiap manajer dalam setiap proses pada peralatan (Indah Puspitorini, 2021).

2.5. BLACK BOX TESTING

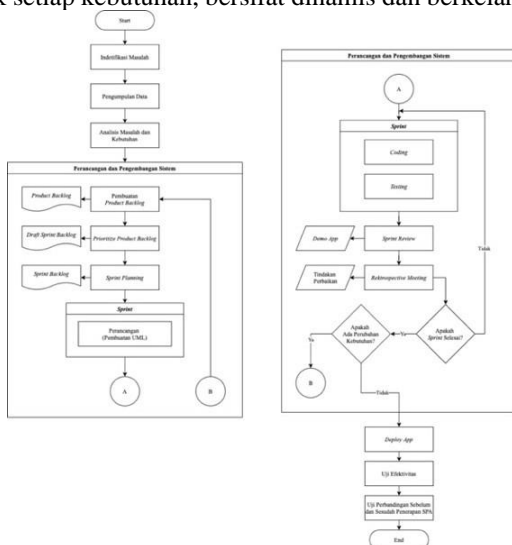
*Blackbox Testing* adalah metode untuk menguji perangkat lunak yang telah dibangun, baik tes dalam unit kecil dan hasil terintegrasi untuk menguji perangkat lunak fungsional. Tes perangkat lunak dalam hal spesifikasi fungsional tanpa menguji kode desain dan kode untuk mengetahui apakah fungsinya, masukkan perangkat lunak sesuai dengan spesifikasi yang diperlukan (M Syarif et al., 2021).

2.6. SKALA LIKERT

Skala Likert digunakan untuk mengukur sudut pandang, sikap atau pendapat individu atau kelompok tentang peristiwa sosial atau fenomena. Ada dua jenis pertanyaan skala Likert, yaitu cara positif pertanyaan untuk mengukur skala positif dan formulir pertanyaan negatif untuk mengukur skala negatif. Pertanyaan positif menerima skor 5, 4, 3, 2 dan 1; Pertanyaan negatif diberikan skor 1, 2, 3, 4 dan 5 (Pranatawijaya et al., 2019).

3. METODE PENELITIAN

Metodologi yang digunakan adalah pendekatan prinsip-prinsip metode *Agile* yaitu menggunakan kerangka kerja *Scrum*. Penelitian menilai metode ini tepat untuk Penelitian gunakan karena memiliki fleksibilitas yang tinggi untuk setiap kebutuhan, bersifat dinamis dan berkelanjutan.



Gambar 1. Flowchart Metode Penelitian

Metode perancangan yang digunakan adalah menggunakan kerangka kerja Scrum yang merupakan turunan dari metode *Agile*. Pada *Scrum*, terdapat beberapa rancangan yang harus dibuat terlebih dahulu sebelum masuk ke dalam pengembangan sistem.

4. HASIL DAN PEMBAHASAN

Hasil penelitian yang telah dilakukan akan diuraikan dan dibahas, yaitu sebagai berikut:

A. Perancangan Product Backlog

Membuat rancangan *product backlog* berdasarkan kebutuhan yang didapat dari tahap sebelumnya.

Tabel 1. Rancangan Product

No	Fitur Backlog	Keterangan
1	Pembuatan <i>use case diagram</i>	Dapat mengetahui proses interaksi pada sistem SUDION
2	Pembuatan <i>activity diagram</i>	Dapat mengetahui proses aktivitas pada sistem SUDION
3	Pembuatan <i>class diagram</i>	Dapat mengetahui alur data input-output pada system dan mengetahui bagaimana objek antar kelas terhubung
4	Pembuatan <i>deployment diagram</i>	Dapat seolah-olah memahami bentuk fisik pada sistem SUDION
5	Penerapan <i>single page application (SPA)</i>	a. Pengguna dapat bernavigasi tanpa <i>reload page</i> b. Pengguna dapat memuat surat, waktu tidak lebih dari 10 detik c. Pengguna dapat mengakses paginasi, tanpa <i>reload page</i>
6	Penerapan <i>webpack</i>	Pengguna hanya memerlukan <i>resource</i> yang lenih minimal

B. Prioritize Product Backlog

Membuat urutan skala prioritas dari perancangan *product backlog* yang telah dibuat sebelumnya

Tabel 2. Draft Spint Backlog

No	Product Backlog	Priority	Iterasi
1	Pembuatan <i>use case diagram</i>	Medium priority	3
2	Pembuatan <i>activity diagram</i>	Medium priority	3
3	Pembuatan <i>class diagram</i>	Medium priority	3
4	Pembuatan <i>deployment diagram</i>	Medium priority	3
5	Penerapan <i>single page application (SPA)</i>	Medium priority	4
6	Penerapan <i>webpack</i>	High priority	4

C. Sprint Planning

Setelah melakukan sprint planning dibuatlah tabel *sprint backlog* untuk melakukan sprint.

Tabel 3. Perancangan Sprint Iterasi Ketiga

No	Fitur Backlog	Task	Est. (Jam)
<b>PERANCANGAN</b>			
1		<i>Use case diagram</i>	5
2	Pembuatan UML	<i>Activity diagram</i>	3
3		<i>Class diagram</i>	3
4		<i>Deployment diagram</i>	3

Tabel 4. Perancangan Sprint Iterasi Keempat

No	Fitur Backlog	Task	Est. (Jam)
<b>PERANCANGAN</b>			
5	Penerapan <i>single page application</i>	Penerapan tubolink	4
6	Penerapan <i>webpack</i>	Penerapan Laravel-mix <i>webpack</i>	4

TESTING

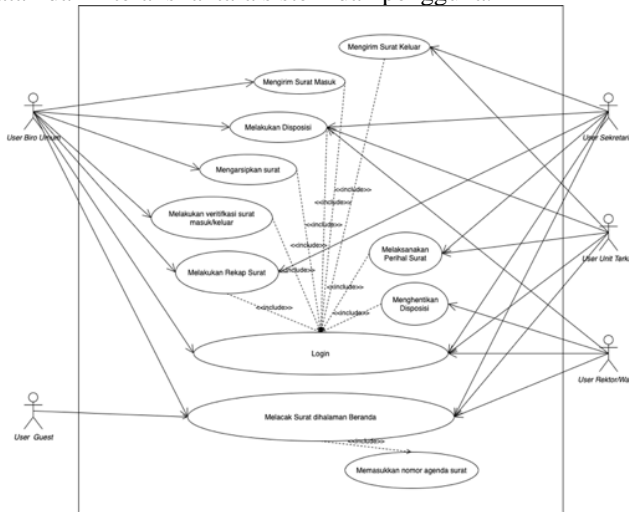
7	Blackbox testing	Pengujian fungsional sistem	8
8	Lighthouse	Pengujian performance, accessibility, etc	8

**D. Sprint**

Proses ini dilakukan untuk mencapai tujuan dari setiap melakukan *sprint*. Output akhiran dari seluruh proses *sprint* ini adalah peluncuran produk.

**a. Pembuatan Use case diagram**

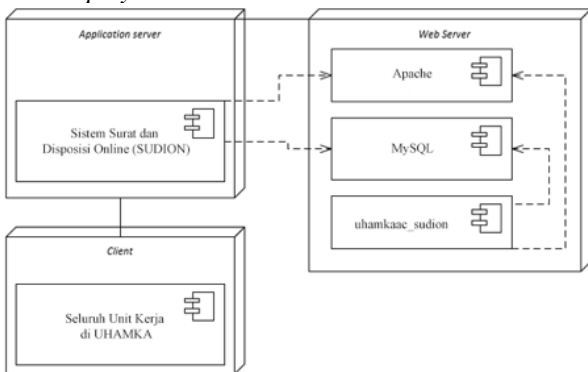
Use Case Diagram dirancang untuk memahami kegiatan dan interaksi antara sistem dan pengguna.



Gambar 2. Use case Diagram Sistem SUDION

**b. Pembuatan Deployment Diagram**

Deployment Diagram adalah diagram yang merepresentasikan arsitektur di dunia nyata, dimana sistem akan di *deploy*.



Gambar 3. Deployment Diagram Sistem SUDION

**c. Penerapan Teknologi SPA**

**1) Instalasi menggunakan NPM**

Tambahkan command pada cli sebagai berikut:

```
npm install -save turbolinks
```

**2) Menjalankan turbolinks**

Tambahkan kode pada berkas *asset javascript* sebagai berikut:

```
Var Turbolinks = require(“trubolinks”)
Turbolinks.start()
```

**d. Penerapan Webpack**

**1) Instalasi webpack**

Tambahkan command pada CLI, sebagai berikut:

```
npm install Laravel-mix -save-dev
```

**2) Konfigurasi webpack.mix.js**

Konfigurasi yang dilakukan adalah mendaftarkan beberapa berkas aset untuk dilakukan penggabungan dan *minified* berkas aset, sebagi berikut:

```
const mix = require('laravel-mix');
mix.js('resources/js/app.js', 'public/js');
mix.styles([
  'public/css/app.css',
  'public/css/style.css',
  'public/progress-tracker/progress-tracker.css',
  'public/css/all.css'
], 'public/css/all.css');
mix.scripts([
  'public/js/jquery.js',
  'public/js/scripts.js',
  'public/js/manifest.js',
  'public/js/vendor.js',
  'public/js/sweetalert.min.js',
  // post.js
  'public/js/post/dispositions/up.js',
  'public/js/post/dispositions/archive.js',
  'public/js/post/dispositions/change-archive.js',
  'public/js/post/dispositions/stop.js',
  'public/js/post/dispositions/finish.js',
  'public/js/post/settings/accounts/delete.js',
  'public/js/post/settings/letter-code/delete.js',
  'public/js/post/settings/divisions/delete.js',
  'public/js/post/settings/units/delete.js',
  'public/js/post/settings/ececlons/delete.js',
  'public/js/all.js'
]);
```

Gambar 4. Konfigurasi Webpack.mix.js

**3) Menjalankan webpack**

Setelah melakukan konfigurasi pada file *webpack.mix.js* langkah terakhir, yaitu Penelitian menjalankan command sebagai berikut.

```
→ sudion_spa git:(master) x npm run prod
> laravel-sudion-client@1.0.0 prod
> npm run production
```

Gambar 5. Command untuk Melakukan Compile Webpack

Setelah proses compile selesai, maka akan terlihat seperti berikut.

```
DONE Compiled successfully in 1635ms
Asset      Size  Chunks  [emitted]  Chunk Names
/css/all.css  325 KIB  [emitted]
/js/all.js   722 KIB  [emitted]
/js/app.js   656 KIB  0 [emitted] [big] /js/app
→ sudion_spa git:(master) x
```

Gambar 6. Keterangan Compile Webpack Selesai

**e. Testing**

Melakukan tahap pengujian dengan menggunakan metode blackbox testing, sehingga diperoleh skenario dan hasil pengujian.

Tabel 5. Skenario dan Hasil Pengujian

No	Skenario Uji	Output	Validasi
1	Mengakses halaman beranda		Berhasil
2	Mengakses halaman login		Berhasil
3	Mengakses halaman dashboard		Berhasil
4	Mengakses halaman transaksi surat keluar dan surat masuk	Halaman berpindah secara cepat,	Berhasil
5	Mengakses halaman disposisi surat keluar dan surat masuk	tanpa reload page	Berhasil
6	Mengakses halaman arsip surat sementara dan tetap		Berhasil

7	Mengakses halaman pengaturan	Berhasil
---	------------------------------	----------

**E. Uji Efektifitas**

Kuesioner memiliki 52 responden, akan dihitung dengan skala likert dan berikut adalah instrumennya.

Tabel 6. Instrumen Uji Efektivitas

No	Partanyaan	STS	TS	RG	S	SS
1	Dari sejak awal rilis SUDION, apakah pembaruan yang dilakukan sudah tepat?	0	0	4	19	29
2	Apakah pembaruan tersebut membuat SUDION sekarang lebih baik dari yang sebelumnya?	0	0	2	19	31
3	Secara "performa" berapa skor untuk SUDION versi sekarang dibandingkan dengan versi yang sebelumnya?	0	0	3	17	32
4	Tingkat kepuasan Anda secara "keseluruhan" terhadap SUDION	0	0	1	17	34

Tabel 7. Hasil Perhitungan Skala Likert

No	Skor					N-Max	Jumlah Skor					Total	Pres. (%)
	1	2	3	4	5		1	2	3	4	5		
	1	0	0	4	19		29	0	0	12	76		
2	0	0	2	19	31	0	0	6	76	155	237	91,15	
3	0	0	3	17	32	0	0	9	68	160	237	91,15	
4	0	0	1	17	32	0	0	3	68	170	241	92,69	
Rata-rata												91,15	

Berdasarkan hasil uji efektivitas yang telah diisi oleh 52 responden menunjukkan bahwa hasil perhitungan skala likert tersebut mempunyai tingkat efektivitas sebesar 91,15%.

**F. Uji Perbandingan**

Supaya mudah untuk membedakan antara kedua nya kita sebut versi terbaru dengan sebutan SUDION (surat disposisi *online*) dan versi awal dengan sebutan SUSI (surat disposisi).

1) Uji perbandingan menggunakan GTmetrix

Tabel 8. Compare Reports SUDION dan SUSI

Nama Indikator	SUDION	SUSI	Efisiensi
GTmetrix grade	F	F	-10%
Performance score	34%	31%	-3%
Structure score	67%	40%	-27%
Largest contentful paint	6.4 s	6.4 s	+80 ms
Total blocking time	0 ms	7 ms	-7 ms
Cumulative layout shift	0	0.001	-0.01

2) Uji perbandingan menggunakan Pingdom

Tabel 9. Pingdom Hasil Uji Perbandingan SUDION dan SUSI

Nama Indikator	SUDION	SUSI
Performance grade	A(92)	C(73)
Page size	1.2 MB	3.9 MB
Load time	4.83 s	7.07 s
Requests	11	24
Compress components with gzip	C(78)	F(0)
Add expires headers	E(56)	F(0)
Use cookie-free domains	B(85)	F(25)
Avoid empty src or href	A(100)	A(100)
Put javascript at bottom	A(100)	A(100)
Reduce the number of DOM elements	A(100)	A(100)
Make favicon small and cacheable	A(100)	-
CONTENT SIZE BY CONTENT TYPE		
Script	11.26% (134.9 KB)	68% (2.7 MB)
Image	5.70% (68.3 KB)	11.05% (433.1 KB)
CSS	35.67% (427.4 KB)	10.68% (418.6 KB)
Front	12.91% (154.7 KB)	5.83% (228.6 KB)
HTML	28.77% (344.7 KB)	3.44% (134.9 KB)

3) Uji perbandingan menggunakan PageSpeed Insight

Tabel 10. PageSpeed Insight Hasil Uji Perbandingan SUDION dan SUSI

Nama Indikator	SUDION	SUSI
Skor	82	63
First contentful paint	1.4 s	1.4 s
Time to interactive	1.4 s	4.0 s
Total blocking time	0 ms	50 ms
Largest contentful paint	1.4 s	3.8 s
Cumulative layout shift	0	0.007

4) Uji perbandingan menggunakan Web.Dev

Tabel 11. Web.Dev Hasil Uji Perbandingan SUDION dan SUSI

Nama Indikator	SUDION	SUSI
Performance	52	27
SEO	69	54
First Contentful paint	6.8 s	5.8 s
Time to interactive	69 ms	20.8 s

**5. KESIMPULAN**

Penelitian ini telah berhasil menganalisis, menyelesaikan sprint dan melakukan *deployment* pada pembaruan sistem SUDION. Pembaruan dengan melakukan penerapan teknologi SPA ini mengatasi masalah performa dan penggunaan *resource* pengguna, sehingga dapat membantu penyelenggaraan pelayanan administrasi surat-menyurat dan disposisi yang lebih baik di lingkungan kerja Universitas Muhammadiyah Prof. DR. HAMKA.

Penelitian berhasil melakukan pengujian efektivitas dan menunjukkan bahwa sistem SUDION ini mempunyai tingkat efektivitas sebesar 91,15%. Penelitian berhasil melakukan pengujian perbandingan menggunakan alat pengujian gtmatrix dan menunjukkan hasil tingkat efisiensi memperoleh selisih 10% dengan perbedaan gtmatrix grade 44% berbanding 34%. Kemudian menggunakan alat pengujian pingdom dan menunjukkan hasil tingkat efisiensi memperoleh selisih 19 poin dengan perbedaan grade A (92 poin) berbanding grade C (73 poin). Kemudian menggunakan pagespeed insight dan menunjukkan hasil tingkat efisiensi memperoleh selisih 19 poin dengan perbedaan 82 poin berbanding 63 poin. Kemudian menggunakan web.dev dan menunjukkan hasil tingkat

efisiensi memperoleh selisih 25 poin dengan perbedaan 52 poin berbanding 27 poin.

## REFERENCES

- Indah Puspitorini. (2021). Penerapan Agile Methode Berpola Scrum Dalam Sistem Informasi Pengajuan Cuti Kerja Berbasis Web. In *akrabjuara.com* (Vol. 6).<http://akrabjuara.com/index.php/akrabjuara/article/view/1424>
- Johansson, J. (2021). *Create React App vs NextJS A comparison of two ReactJS based web application frameworks*. <https://www.diva-portal.org/smash/record.jsf?pid=diva2:1573881>
- Pranatawijaya, V. H., Widiatry, W., Priskila, R., & Putra, P. B. A. A. (2019). Penerapan Skala Likert dan Skala Dikotomi Pada Kuesioner Online. *Jurnal Sains Dan Informatika*, 5(2), 128–137. <https://doi.org/10.34128/jsi.v5i2.185>
- Setiyani, L., & Setiawan, B. (2021). Analisis Dan Design Manajemen Control Produksi Menggunakan Business Process Improvement Dan Unified Modelling Language (STUDI KASUS: PT. MULTISTRADA). *Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi Dan Komunikasi*, 16(1), 27–37. <https://doi.org/10.35969/interkom.v16i1.94>
- Siregar, H. F., & Melani, M. (2019). Perancangan Aplikasi Komik Hadist Berbasis Multimedia. *Jurnal Teknologi Informasi*, 2(2), 113. <https://doi.org/10.36294/jurti.v2i2.425>
- Syarif, M, Informatika, E. P.-J. (Jurnal T., & 2021, undefined. (2021). ANALISIS METODE PENGUJIAN PERANGKAT LUNAK BLACKBOX TESTING DAN PEMODELAN DIAGRAM UML PADA APLIKASI VETERINARY SERVICES YANG. *Jurnal.Kaputama.Ac.Id*, 5(2). <http://jurnal.kaputama.ac.id/index.php/JTIK/article/view/622>
- Syarif, Muhamad, & Nugraha, W. (2020). PEMODELAN DIAGRAM UML SISTEM PEMBAYARAN TUNAI PADA TRANSAKSI E-COMMERCE. *Jurnal Teknik Informatika Kaputama (JTIC)*, 4(1). <https://jurnal.kaputama.ac.id/index.php/JTIK/article/view/240>
- Wibowo, A. T., & Wiguna, A. S. (2019). Pemanfaatan Teknologi Single Page Application (SPA) dalam Pembuatan Aplikasi Feedback Dosen dari Mahasiswa Sebagai Bentuk Pengawasan Lembaga Terhadap Kinerja Dosen di Bidang Pengajaran. *SMARTICS Journal*, 5(1), 34–43. <https://doi.org/10.21067/smartics.v5i1.3327>