

# **Analisis Penerapan *Single Page Application (SPA)* dalam Meningkatkan *User Experience* pada Sebuah Website**

Sagung Putri Nariswari<sup>1</sup>, I Gusti Agung Gede Arya Kadyanan<sup>2</sup>

Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam  
Universitas Udayana  
Bukit Jimbaran, Badung, Bali, Indonesia

sagungputri1530@gmail.com  
gungde@unud.ac.id

## **Abstrak**

*Setiap website ataupun aplikasi tidak hanya harus memiliki tampilan yang menarik, dibalik itu perlu adanya kenyamanan saat user mengakses dan menggunakan aplikasi tersebut. Single Page Application (SPA) atau secara sederhana diartikan sebagai aplikasi satu halaman merupakan salah satu teknik dalam development website yang bekerja secara dinamis mengubah tampilan maupun berpindah halaman tanpa perlu memuat ulang keseluruhan halaman. Teknik ini dipercaya dapat membantu dalam menghemat waktu tunggu yang tentunya dapat berpengaruh terhadap peningkatan user experience sebuah website. Oleh karena itu, dilakukan penelitian dan uji coba pada sebuah aplikasi website yang menerapkan teknik tersebut. Berdasarkan hasil uji coba, didapatkan bahwa penggunaan Single Page Application (SPA) dapat menghemat penggunaan data yang cukup signifikan serta dengan kecepatan memuat (loading time) yang tergolong cepat sehingga sangat efisien dalam membantu meningkatkan user experience bagi aplikasi yang menggunakan data yang tidak begitu kompleks.*

**Kata Kunci :** *Single Page Application, SPA, Website, User Experience*

## **1. Pendahuluan**

Selain mengembangkan sebuah sistem atau aplikasi kita perlu memperhatikan tampilan aplikasi serta kenyamanan pengguna saat mengakses aplikasi yang kita kembangkan. Sebuah aplikasi baik website, mobile, desktop, dan lainnya perlu memperhatikan kenyamanan pengguna saat mengakses aplikasi yang dikembangkan. Jika ditinjau dari segi estetika, sebuah halaman website yang terlihat begitu menarik, baik dari desain hingga kemudahan pengguna (user-friendly), maka hal ini akan ikut berimbas pada saat halaman web tersebut akan ditampilkan.[3] Hal ini tentu menuntut para programmer agar dapat memaksimalkan dan mengoptimalkan aplikasi yang dikembangkan, salah satunya dengan membuat agar saat mengakses dan beralih ke berbagai halaman pengguna tidak membutuhkan waktu lama dan tentunya akan memberikan efisiensi dari segi waktu. Salah satu teknik dalam pengembangan website untuk melakukan optimalisasi ini adalah *Single Page Application (SPA)* yang kini cukup sering digunakan oleh beberapa perusahaan.

*Single Page Application* merupakan aplikasi yang bekerja di dalam browser yang tidak membutuhkan reload page saat digunakan. Dengan kata lain, pengguna atau user tidak akan berpindah halaman dengan melakukan request kepada server setiap kali terjadi interaksi pada aplikasi. Yang membedakan SPA dengan non-SPA adalah single page application hanya akan melakukan load terhadap satu halaman dari server kemudian mekanisme routing yang biasanya di-handle oleh server kini dibebankan pada client. Akibatnya, website yang menggunakan SPA memiliki performa yang lebih cepat tanpa harus load halaman secara terus menerus.[1]. SPA menyediakan cara yang lebih fleksibel dan elegan berkaitan dengan data. Menyegarkan (refresh) bagian tertentu atau bagian dari halaman tanpa memuat ulang seluruh halaman adalah tujuan utama layanan SPA, tetapi semua fleksibilitas ini membutuhkan user interface yang lebih interaktif dan hal ini menyebabkan *user experience* yang lebih baik.

Oleh karena itu, untuk membuktikan beberapa keuntungan yang diberikan teknik *Single Page Application (SPA)* ini terutama pada sisi *user experience* maka pada penelitian ini akan dilakukan uji coba kecepatan waktu tunggu saat memuat halaman dan juga penggunaan data. Untuk melakukan pengujian ini digunakan sebuah aplikasi menggunakan Reactjs dan Node.js sebagai *library* Javascript yang akan digunakan, serta penggunaan *react-router-dom* untuk mengimplementasikan SPA pada website yang akan dibuat. Selain itu pada pembuatan website ini akan menggunakan data *fake API* dari *jsonplaceholder.com* sebagai alat yang digunakan sebagai parameter *request* untuk tahap pengujian nantinya.

## 2. Metode Penelitian

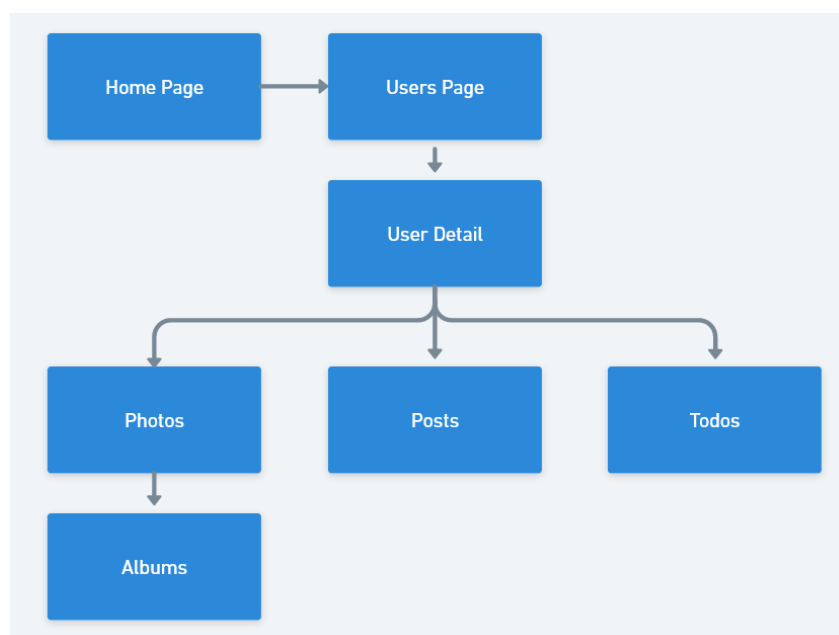
Penelitian dilakukan melalui beberapa tahapan penelitian yang terbagi menjadi 5 tahapan, yaitu : 1) Studi Literatur, 2) Perancangan dan Implementasi Desain, 3) Analisis dan Pengujian Sistem, 4) Penarikan Kesimpulan. Tahapan penelitian tersebut dapat dijelaskan sebagai berikut.

### 2.1 Studi Literatur

Pada tahap ini dilakukan studi terhadap beberapa jurnal dan artikel yang berkaitan dengan teknik pembuatan web aplikasi menggunakan metode *Single Page Application (SPA)*. Pada tahapan ini juga dilakukan untuk menemukan rumusan masalah dan menemukan solusi yang akan ditawarkan. Pertama dilakukan pemahaman mengenai konsep *Single Page Application (SPA)* pada sebuah website dan perbedaannya dengan metode lain. Lalu dilanjutkan dengan menentukan kebutuhan untuk membuat sebuah aplikasi untuk melakukan uji coba terhadap metode ini. Aplikasi yang akan dibuat yaitu berupa website dengan menggunakan *library* JavaScript yaitu ReactJS dan menggunakan API yang bersumber dari *jsonplaceholder.com* untuk ditampilkan dalam website.

### 2.2 Perancangan dan Implementasi

Pada tahap perancangan desain dalam penelitian ini, disiapkan beberapa *mockup* dari website yang akan dibuat sebagai model dari website yang akan dikerjakan. Alur setiap halaman dalam aplikasi yang akan dibuat dapat dilihat pada *sitemap* di bawah ini.



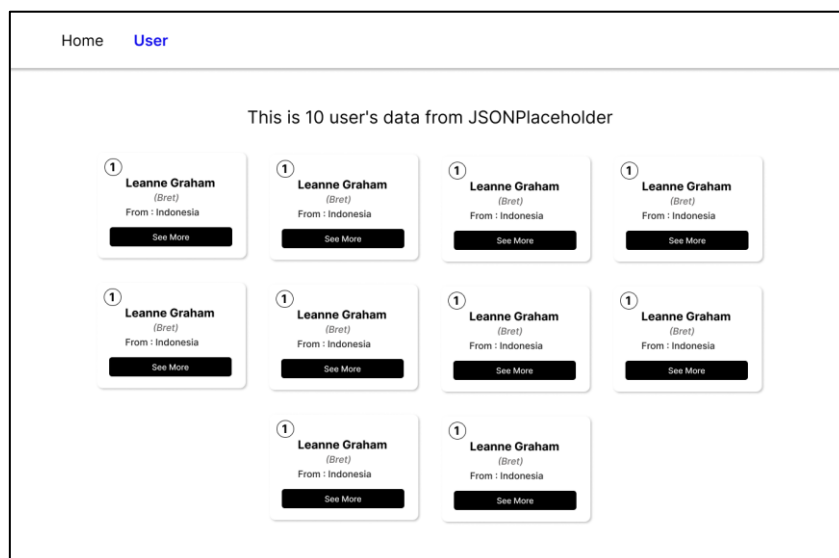
Gambar 1. Sitemap Website

Adapun beberapa tampilan *mockup* tersebut sebagai berikut.



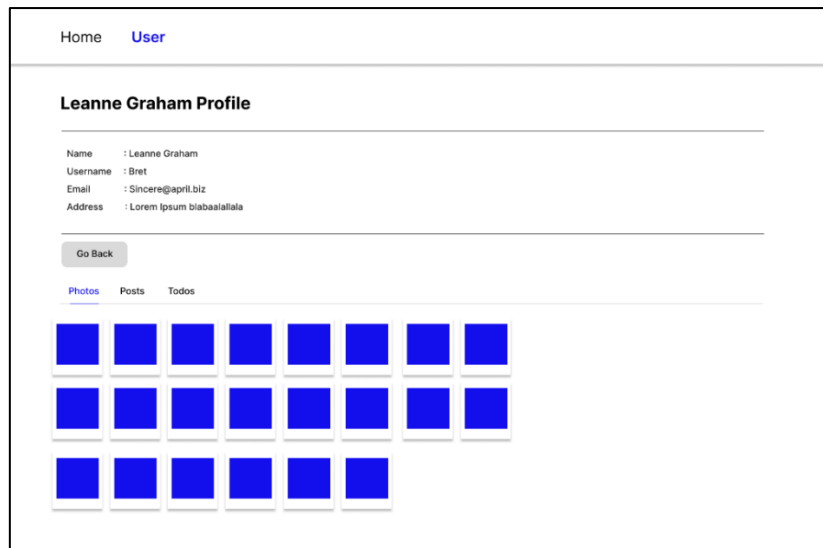
**Gambar 2.** Home Page

Gambar 2 merupakan *mockup* untuk halaman utama atau *home page* yang berisi sedikit penjelasan program dan sebuah tombol CTA (*Call-to-action*) yang jika diklik akan diarahkan ke menu users yang menampilkan data user-user yang didapat dari API.

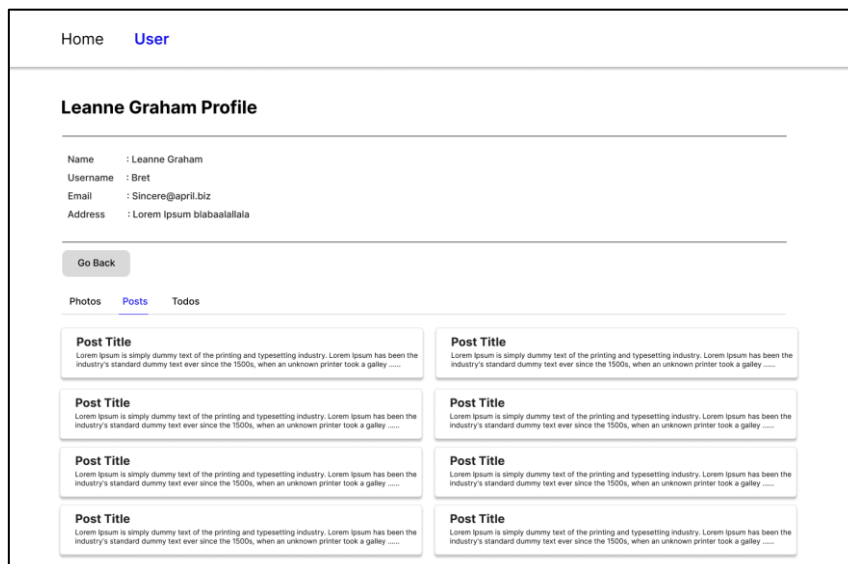


**Gambar 3.** Users Page

Gambar 3 merupakan tampilan untuk halaman users yang akan menampilkan daftar users beserta beberapa informasi singkat dari masing-masing user. Selain itu pada tiap user terdapat tombol (*button*) "see more" yang digunakan jika pengguna ingin melihat data yang lebih detail dari user yang dipilih.



Gambar 4. Users Detail Page



Gambar 5. Users Detail Page

Gambar 4 dan 5 adalah tampilan user detail yang merupakan lanjutan dari halaman sebelumnya, jika pengguna menekan tombol “see more” pada bagian bawah dari idenditas setiap user maka akan dialihkan ke halaman di atas, yaitu halaman detail dari masing-masing user. Selain itu, di halaman ini terdapat beberapa navigasi lain yaitu photos, posts dan todos. Jika *user* ingin melihat daftar foto dari *profile* yang dipilih maka ditampilkan data beberapa foto, begitu juga dengan menu lainnya.

Selanjutnya tahap implementasi, pada tahap implementasi digunakan beberapa *component* dalam ReactJS yang nantinya akan digunakan sebagai halaman yang akan diakses dari halaman utama secara dinamis menggunakan konsep react-router-dom. Adapun beberapa *component* yang akan digunakan antara lain, Navbar, Homepage, Userpage, User Detail Page (identitas, photos, post, dan todos). Pada component User Detail Page terdapat *nested routes* atau halaman bercabang, yang mana pada halaman ini akan menampilkan data photos, posts, dan todos dari user yang memiliki halaman tersebut berdasarkan fitur yang dipilih seperti *mockup* di atas. Lalu untuk memanggil setiap halaman menggunakan beberapa modul yang disediakan oleh react-router-dom, antara lain “Link”, “Routes”, “Route”, “Outlet”, dan “BrowserRouter”.

## 2.3 Analisis dan Pengujian Sistem

Pada tahap ini akan dilakukan pengujian untuk menguji tingkat kecepatan memuat (*loading time*) dan juga besarnya penggunaan data yang digunakan saat pertama memuat dan berpindah halaman dalam website yang dikembangkan dengan metode *Single Page Application (SPA)* menggunakan ReactJS dan react-router-dom. Hasil pengujian yang didapatkan akan berupa data waktu dalam sekon (s) dan besar data dalam *kilobytes/megabytes (kb/mb)* dari setiap halaman yang ada pada website. Pada tahap ini juga akan dilakukan perbandingan hasil pengujian terhadap website yang dibuat dengan website tradisional atau dengan teknik *Multipage Application (MPA)*.

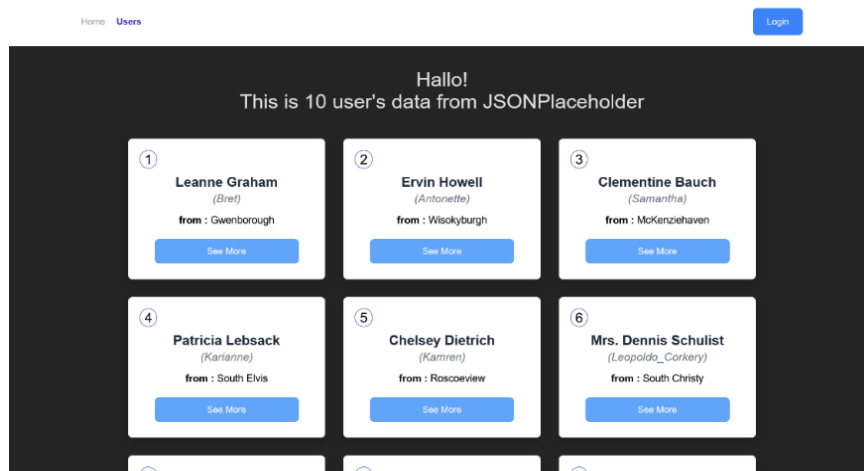
## 2.4 Penarikan Kesimpulan

Tahap terakhir ini akan berisi kesimpulan akhir berdasarkan hasil pengujian dan analisis yang telah dilakukan, yaitu berdasarkan hasil pengujian dan juga perbandingan website yang tidak menggunakan teknik *Single Page Application (SPA)* ini.

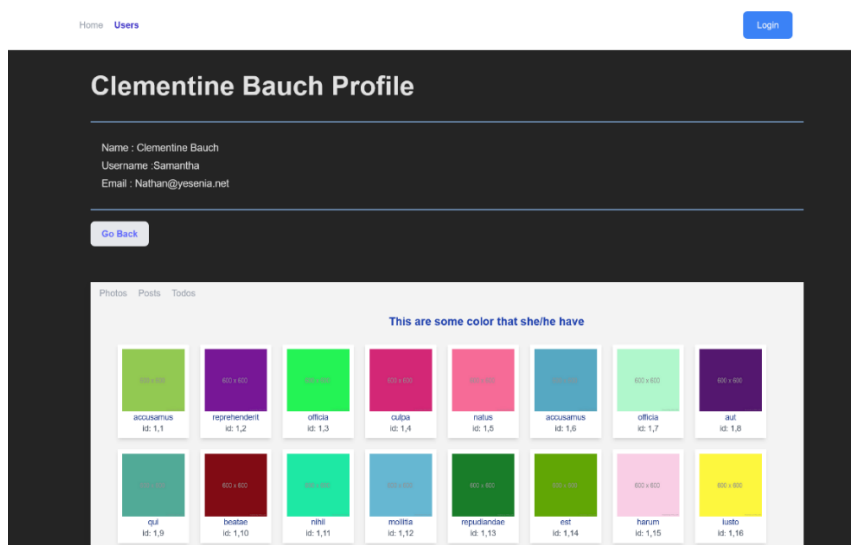
## 3. Hasil dan Pembahasan

### 3.1. Implementasi *UI Design* menjadi Website

Berikut beberapa gambar tampilan website yang sudah dikerjakan menggunakan teknik *Single Page Application (SPA)* menggunakan ReactJS.

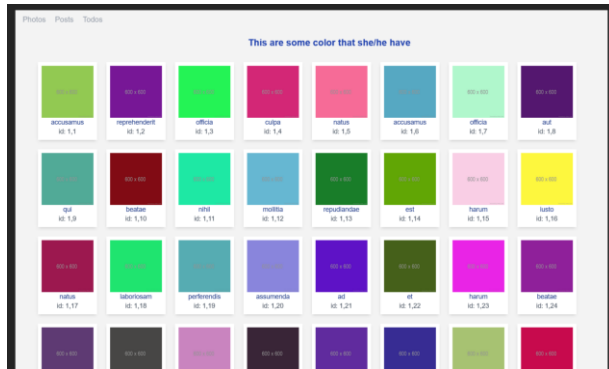


Gambar 6. Halaman Daftar Users

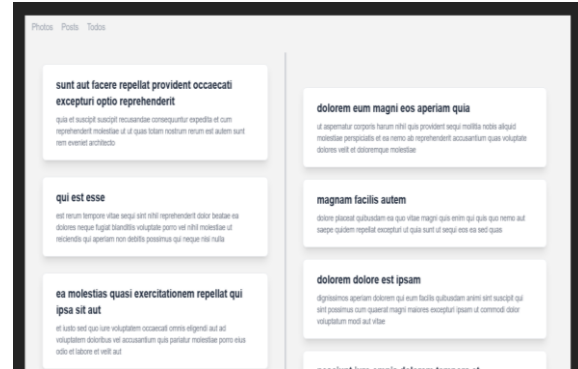


Gambar 7. Halaman Detail Users

Berikut navigasi dari beberapa halaman media yang terdapat pada halaman detail user (post, photos, todos).



Gambar 8. Menu Photos



Gambar 9. Menu Posts



Gambar 10. Menu Todos

### 3.2. Pengujian Aplikasi

Berikut adalah data hasil pengujian setelah program dijalankan yang mencatat penggunaan data dan juga kecepatan memuat saat dilakukan perpindahan ke beberapa halaman.

#### a. Hasil Penggunaan Data

Penerapan Single Page Application pada sisi client tidak akan memuat ulang halaman dari awal, tetapi akan memperbarui bagian-bagian tertentu sesuai permintaan pengguna atau client. Hal ini akan meminimalkan penggunaan bandwidth dan meningkatkan kecepatan akses. [4]. Berdasarkan penjabaran tersebut, berikut ini merupakan hasil penggunaan data saat webiste pertama kali berpindah halaman. Terdapat 6 halaman yang digunakan untuk pengujian, yaitu home, users, user detail, posts, photos, dan todos.

Tabel 1. Hasil Pengujian Penggunaan Data

Nama Halaman	Jumlah Penggunaan Data (kb)
<b>Halaman Utama</b>	
Home Page	26,27 KB
Users Page	20,12 KB
<b>Halaman Detail</b>	
User Detail (Main)	38,38 KB
Photos	27,95 KB
Posts	27,96 KB
Todos	28,01 KB

Berdasarkan data yang didapatkan pada tabel 1, rata-rata ukuran data pada halaman user detail saat berpindah dari page photos, posts, dan todos memiliki nilai yang hampir sama karena response data yang diterima hampir mirip yaitu berupa data dan sedikit gambar. Saat perpindahan dari home page dan users page juga tergolong sedikit. Sedangkan saat jumlah data total yang

diterima pada halaman user detail cukup besar dibanding yang lain karena di dalam halaman tersebut menampung data dari *nested routes*.

**b. Load Time Testing**

Pengujian ini digunakan untuk mengetahui waktu tunggu (*loading time*) dari aplikasi saat memuat data setiap berpindah halaman. Berikut ini adalah hasil pengujiannya.

**Tabel 2.** Hasil Pengujian *Loading Time*

Nama Halaman	<i>Loading Time</i>
<b>Halaman Utama</b>	
Home Page	1,19 s
Users Page	1,46 s
<b>Halaman Detail</b>	
User Detail (Main)	1,75 s
Photos	2,16 s
Posts	3,47 s
Todos	3,66 s

Berdasarkan tabel di atas, hasil loading time saat pertama kali memuat keenam halaman tersebut, tidak memiliki rentang yang cukup jauh. Dengan besaran *loading time* tersebut, proses loading komponen masih relatif ringan dan tidak memakan waktu yang lama untuk menampilkan data pada setiap halaman, di mana bertambah sekian detik saja bahkan tidak mencapai lebih dari 1 detik.

**c. Perbandingan *loading time* dan penggunaan data dengan website jenis lain**

Perbandingan ini dilakukan terhadap website yang telah dibuat dengan website atlassian.com yang merupakan salah satu website yang dalam pengembangannya menggunakan teknik atau metode *multi-page application* atau website tradisional, dimana perlu melakukan reload ulang data pada server setiap melakukan perpindahan halaman. Untuk uji coba dan perbandingan ini akan digunakan beberapa halaman dari website atlassian.com yang memiliki alur navigasi, jumlah data yang dimuat dalam page, dan sistem yang mirip dengan website yang sudah dibuat. Adapun data yang dihasilkan dapat dilihat pada dua tabel berikut.

**Tabel 3.** Perbandingan berdasarkan penggunaan data

SPA website		Atlassian.com (MPA)	
Nama Halaman	Jumlah Penggunaan Data (kb)	Nama Halaman	Jumlah Penggunaan Data (kb)
<b>Halaman Utama</b>		<b>Halaman Utama</b>	
Home Page	26,27 KB	Home Page	114,04 KB
Users Page	20,12 KB	Jira Software (home)	222,72 KB
<b>Halaman Detail</b>		<b>Halaman Detail</b>	
User Detail (Main)	38,38 KB	Jira Features	608,83 KB
Photos	27,95 KB	Scrum	121,71 KB
Posts	27,96 KB	Kanban	121,71 KB
Todos	28,01 KB		

**Tabel 4.** Perbandingan berdasarkan *loading time*

SPA website		Atlassian.com (MPA)	
Nama Halaman	Jumlah Penggunaan Data (kb)	Nama Halaman	<i>Loading Time</i>
<b>Halaman Utama</b>		<b>Halaman Utama</b>	
Home Page	1,19 s	Home Page	9,05 s
Users Page	1,46 s	Jira Software (home)	12,28 s
<b>Halaman Detail</b>		<b>Halaman Detail</b>	

User Detail (Main)	1,75 s	Jira Features	13,07 s
Photos	2,16 s	Scrum	15,6 s
Posts	3,47 s	Kanban	10,23 s
Todos	3,66 s		

Dari data pada kedua tabel di atas, terlihat bahwa website yang menggunakan *Single Page Application* memiliki kecepatan dan juga penggunaan transfer daya saat memuat halaman yang cukup signifikan dibandingkan website pada umumnya. Pada website dengan SPA jumlah penggunaan data dan waktu tunggu yang dibutuhkan cenderung memiliki selisih yang tidak cukup jauh dengan halaman lainnya saat melakukan navigasi jika dibandingkan dengan website bertipe *multipage* yang lebih membutuhkan lebih banyak waktu untuk memuat ulang setiap kali melakukan perpindahan halaman.

#### 4. Kesimpulan

Berdasarkan data yang didapatkan pada proses pengujian website dapat disimpulkan bahwa pengimplementasian teknik *Single Page Application (SPA)* cukup mampu digunakan untuk meningkatkan kualitas user experience suatu aplikasi, khususnya pada sisi penggunaan data dan juga efisiensi waktu. Hal tersebut dapat dilihat dari hasil pengujian pada tabel pertama untuk aplikasi sederhana yang mendapatkan hasil di kisaran 26 – 28 kb yang termasuk cukup ringan, dimana jika pada website biasa dapat berada di atas 50 KB. Selain itu, berdasarkan *loading time* proses memuat dan perpindahan dari tiap halaman tergolong cukup cepat, hanya saja semakin banyak data dan semakin banyak urutan halaman dinamis yang dilewati *loading time* sedikit bertambah. Namun, berdasarkan data yang dihasilkan metode *Single Page Application (SPA)* ini masih sangat efisien digunakan untuk meningkatkan *user experience* bagi aplikasi yang menggunakan data yang tidak begitu kompleks. Jadi, dapat dikatakan teknik SPA ini tergolong cepat jika diandalkan untuk website yang memerlukan waktu *loading* cepat serta menggunakan interaksi yang dinamis.

#### References

- [1] J. Gunawan, "Teknologi Single Page Application (SPA)", 2022. [Online]. Available: <https://socs.binus.ac.id/2018/12/06/teknologi-single-page-application-spa/>. [01 Oktober 2022]
- [2] Lestari, P., & Belluano, L. PENGEMBANGAN SINGLE PAGE APPLICATION PADA SISTEM INFORMASI AKADEMIK. *ILKOM Jurnal Ilmiah*, 10. 2018
- [3] M. F. Santoso, "Teknik Single Application (SPA) Web Dengan Menggunakan ReactJS dan Bootstrap", *Jurnal Khatulistiwa Informatika*, vol. 8, no. 2, 107-114, 2021.
- [4] J., Wikardiyana, A. Made Widiartha and L. A. Rahning Putri, "Perancangan dan Implementasi Sistem Manajemen Proyek Perangkat Lunak Menggunakan Teknologi Single Page Application", *Jurnal Elektronik Ilmu Komputer Udayana*, vol. 8, no. 4, 2020.



*This page is intentionally left blank.*