

WEB APPLICATION DESIGN USING SINGLE PAGE APPLICATION TO INCREASE WEBSITE PERFORMANCE ON RUMANTARA

Ilham Farobi

Information System Study Program,
Industrial Engineering Faculty,
Telkom University
Bandung, Indonesia
ilhamfarobi@gmail.com

Soni Fajar Surya Gumilang

Information System Study Program,
Industrial Engineering Faculty,
Telkom University
Bandung, Indonesia
mustonie@telkomuniversity.ac.id

Muhammad Azani Hasibuan

Information System Study Program,
Industrial Engineering Faculty,
Telkom University
Bandung, Indonesia
muhammad.azani@gmail.com

Abstract— One of the travelers needs when going to some places is a place to stay for a while and put their things. More importantly, a place that fits their needs and budget. This matter makes most of the travelers difficult to find a place to stay because of most of them is hotels with unneeded amenities that makes its price expensive. Therefore, they rely more on information from people around their destination place. Which that makes they waste more time to ask people than going to vacation. This research aims to develop a web application which provides room rental information and rental management for room owners. This research focused on frontend technology using Single Page Application concept which will be used and see by the website's users. With this website, travelers can search for rooms rental information around their destination which match with their needs then order it to the room owner via our website. Thus they will able to save their time more on vacation because their rooms are ready when they arrive at their destination place. The development methodology that used by this research is iterative and incremental which consist of six iterations. The front-end of this web is developed using ReactJS as Javascript library, Redux as state management of React Components, and Webpack as Javascript bundler. The result of this research is Single Page Application website to help travelers find their place to stay and put their things also able help room owner to utilize their unused room then rent it from this website.

Keywords— *rooms rental; single page application; iterative incremental.*

I. INTRODUCTION

Indonesia is a rich country of many beautiful and historical places for tourism or traveler. Indonesia has more than 962 tourism spot that spread all over the country. According to the central bureau of statistics, for the year 2016, Indonesia has increased the number of tourist or traveler visit almost in each month compared to the previous year [1]. On the other hand, the growth of traveler visitation gives an opportunity to the locals to improve their economy. Locals can rent their houses or empty rooms to the traveler. The concept of renting empty rooms, travelers can stay in the

same roof with the room owner. This will give the traveler a new experience for staying with locals. In addition, this will also help the traveler to reduce their money cost because of they can measure their needs based on their budget. Rumantara is made to be the media for travelers and room owners, help them to make a transaction of room renting, payments, etc. Rumantara is a startup that runs on the room rentals field. The main purpose of Rumantara is to help travelers find the place to stay easily and efficiently also help to increase the economy around the tourist attractions.

To get the best result on Rumantara website development, its team' job divided into some divisions. Rumantara's division consists of front-end, back-end API, back-end Admin, and Mobile Development (Android). In this study, the researchers will focus on the development of front-end on Rumantara website. The development concept that will be used in this research is Single Page Application. This concept can improve the performance on the website by simply retrieving only the data API from the server then the rendering process will fully in client side. To support this concept, the researcher uses ReactJS as Javascript library.

II. LITERATURE REVIEW

A. Single Page Application

Single Page Application is usually used to describe an application that built for the web. Websites that use this concept also accessed via a web browser like other websites, but offer more dynamic interactions resembling native mobile and desktop apps [2]. The technique of this application is reduced amount of page refreshes. SPAs used AJAX to communicate with back-end servers without doing the full page refresh. Then data result from server processed and rendered mostly on the client-side.

SPA can provide the best of both worlds — the immediacy of a desktop application and the portability and accessibility of a website. The JavaScript SPA is available everywhere there is a web browser, works on desktop and mobile devices runs on multiple Operating Systems, and doesn't require any proprietary plugins. SPAs are easily

updated and distributed, usually without requiring any action from the user [3]. SPA can be visited in two modes [4]:

1) App launch

This occurs when the app is initially loaded by entering the URL in the browser or by clicking on an email link. “App launch” mode is typically slow, as the application (JS/CSS) needs to be downloaded and booted before doing the work to render the page.

2) Subsequent

This occurs when the app has already been loaded and the page is visited by clicking on a link within the app. “Subsequent” mode is typically fast because the application is already downloaded and booted, and we just need to fetch new data for the page and render it.

Researcher uses this concept of website development because it can improve the performance of the website by only get needed data from the server and not to refresh the page. It can improve user satisfaction, websites able to minimize the load of resources that makes users wait. The website that can perform like desktop application with quite fast speed for client-side makes performance better than usual websites.

B. Iterative Incremental

Iterative and incremental is a combination of two design or method. Meaning of iterative and incremental is a system that grows incrementally over time, iteration by iteration [5]. The key of success from this iterative and incremental Software Development Life Cycle (SDLC) is strict validation from requirement and verification, also there will be product testing in every version or iterative. Like other SDLC, iterative and incremental have several phases.

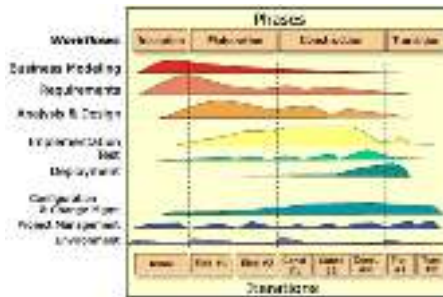


Fig. 1. Iterative Incremental Phase (Larman, 2005)

From Figure 1, Lamran explain the phases in iterative and incremental method, consist of:

1) Inception

Inception is not a requirement phase; rather, it is a kind of feasibility phase, where just enough investigation is done to support a decision to continue or stop.

2) Elaboration

Most of this phase is to define the architecture of the application, risk, identification of most requirements and scope, also define more realistically estimates.

3) Construction

This phase implement of requirements and others that already defined before. Give priority for remaining lower risks and easier to implement, then prepare for deployment.

4) Transition

After deployment of the application, this phase will be focused on testing to users (User Acceptance Test). Purpose of this phase is to analyze whether the result of product already met with user requirements

In the development of application in this research, researcher choose this methodology rather than other methodology because this methodology is matched with the small team of developers but can understand user requirements more perfectly also easily define the system functionality.

C. ReactJS

ReactJS is an Open Source Javascript Library which maintained by wold's topmost community such as Facebook, Instagram, and community of individual developers. Its aims to develop reusable UI components and can be represented as view 'V' in MVC model in web development. It focuses on efficiency and flexibility to create user interfaces [6]. The view of ReactJS is represented as a component which can be reusable. Encapsulated components manage their own states and merged to build more interactive User Interfaces. The strongest point of ReactJS is that it does not perform only on client side. ReactJS can be rendered on the server side. The server will render React Component that already defined then the server will give a response to the client as pure HTML.

III. IMPLEMENTATION

The web application of this research called Rumanlara. This implementation will focus on front-end development of the web application. Using the concept of Single Page Application that will improve the performance of the website and reduce usage of bandwidth.

Development of Rumanlara uses the iterative and incremental methodology in 6 iterations. The following will only explain the result of all iterations by phase.

A. Inception Phase

The result of this phase is the software requirement specification which defined from surveys and interview from target market/user. Table of software requirement will be shown in Table 1 below.

TABLE I. USER REQUIREMENTS

REQ-ID	Requirement Group	Requirement	Description	User
REQ-01.01	Manage Account	Login	Login process in web application.	Room hoster and travelers.
REQ-		Logout	Logout	Room

REQ-ID	Requirement Group	Requirement	Description	User
01.02	Manage Profil		process after login in web application.	hoster and travelers.
REQ-01.03		<i>Register</i>	Registration process in web application.	Room hoster and travelers.
REQ-02.01		<i>Add / Change ID Card</i>	Process to add or change ID Card Information.	Room hoster and travelers.
REQ.02.02		<i>Change Profile</i>	Process edit detail profile.	Room hoster and travelers.
REQ.02.03		<i>Change Password</i>	Process change password security.	Room hoster and travelers.
REQ.02.04		<i>Add / Change Bank Account</i>	Process add or change user registered bank account.	Room hoster and travelers.
REQ.03.01	Manage Room	<i>Add Room</i>	Process register room that will be rent.	Room Hoster
REQ.03.02		<i>Edit Room</i>	Process edit detail Room	Room Hoster
REQ.03.03		<i>Delete Room</i>	Process to delete registered Room	Room Hoster
REQ.04.01	Order Room	<i>Order</i>	Process to order / book Room.	Travelers
REQ.04.02		<i>Confirm Order</i>	Process Confirmation of order	Room Hoster
REQ.05.01	Topup	<i>Add balance nominal</i>	Process add topup balance in web application	Room Hoster and Traveler
REQ.05.02		<i>Upload proof of transfer (Confirmation)</i>	Process topup confirmation from proof of bank transfer	Room hoster and traveler
REQ.05.03		<i>Withdraw Balance</i>	Process withdraw balance that will be transferred to registered bank account	Room Hoster and Traveler
REQ.06.01	Feedback	<i>Add feedback</i>	Process add feedback to room	Traveler
REQ.06.02		<i>Remove feedback</i>	Process to delete submitted feedback.	Traveler

From Table 1 above, researcher defines the user requirements from interview and surveys into 6 requirements. Every requirement has its own sub-requirements.

B. Elaboration Phase

In this phase, the researcher defines the diagrams and architecture of Rumantara's web application from Inception Phase.

1) Use Case Diagram

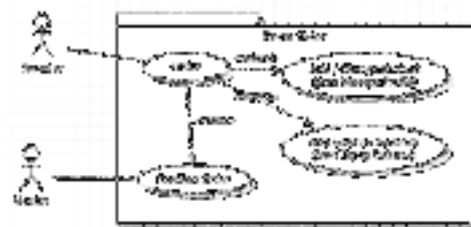


Fig. 2. Use Case Add Order

Figure 2 explain about use case diagram of core business of Rumantara. Add Order functional is process that need 2 actors, traveler will doing the order and hoster will confirm the order.

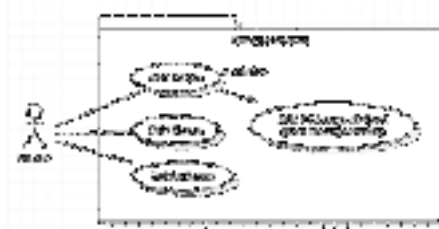


Fig. 3. Manage Room Use Case

Figure 3 above explain about use case of Manage Rooms requirements, which consist of add room, edit room and delete room.

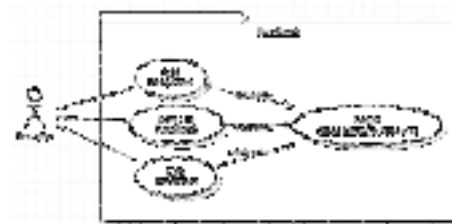


Fig. 4. Feedback Use Case

Figure 4 above shows the feedback use case which consist of add feedback, remove feedback and edit feedback. All off the functionality need to login.

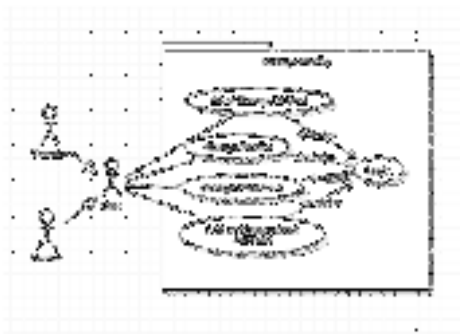


Fig. 5. Manage Profile

Figure 5 above explain about manage profile use case that consist of several functionality. The actors that consist of traveler and hoster will have the same functionality.

2) Activity Diagram

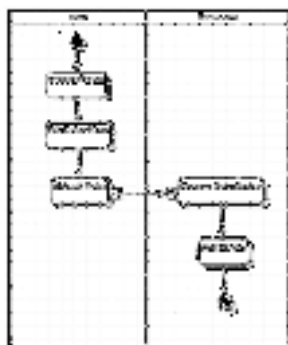


Fig. 6. Activity Diagram of Add Order

Figure 6 above explain about activity diagram from add order use case. The activity diagram will involve 2 swimlanes, that are user and Rumantara system.



Fig. 7. Activity Diagram of Add Balance

Figure 7 above explain about activity diagram of add balance. Add balance functionality is going to be used by all actors. It is the core business of transaction.

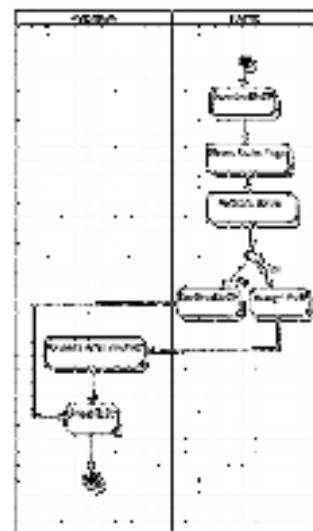


Fig. 8. Activity Diagram of Confirm Order

Figure 8 is explain about use case of confirm order which is from the core business of Rumantara. Confirm order doing by Hoster actor.

3) System Architecture

Architecture of system application is important, wether for client-side or developer-side. Rumantara build an architecture that able to be scalable architecture with minimum cost and able to make deployment easier.

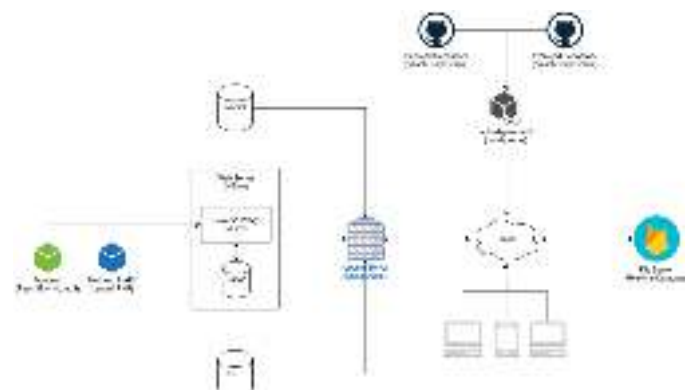


Fig. 9. Rumantara's Architecture

Figure 9 above is system architecture which is used by Rumantara's website. This architecture use some tools to support for client and developers, bellow is the explanation of used tools:

a) MySQL

The most important things of integrated application are centralized database server for all client. Rumantara uses MySQL to support production or development version of the database.

b) NGinx

Rumantara uses Nginx as web server of PHP application which is used by back-end API (Application Programming Interface) and Admin. Rumantara also uses Nginx as the

reverse proxy and buffer to support performance and deployment.

c) *Firebase*

To easily maintain files that uploaded to Rumantara website, Rumantara uses Firebase as files server. Rumantara will upload files to Firebase then save the file URL in Mysql database. Firebase also used as a real-time database, that will be used by chat functionality in Rumantara.

d) *Github*

Developers use this tool to collaborate each other. The project in Github is called repository which is consist of several branches. Branch in this research represented as developer branch and as deployment branch. Deployment branch will be used to deploy the source code to the server.

e) *Buddy.works*

Auto deployment of application is the main functionality that makes developer more comfortable to develop. Buddy.works will integrated with Github especially is deployment branch. Buddy.works do auto deployment in every push of branch deployment. Means that if there are new source codes in deployment branch Buddy.works will upload it to server and able to run some commands as it set by developers.

C. Construction Phase

The construction phase is the development phase of requirements that already defined before. In this research is going to show only the result of this phase. The interface of website application is the result of front-end development. Bellow are some of the interfaces of Rumantara's web application.



Fig. 10. Home Page Interface

Figure 10 above is the interface of Rumantara's home page. Which is display the favorite rooms and favorite destination. In this page will show the search bar for instance search so user can search their destination based on their place, check-in date and check-out date. The bottom of this page will show the most visited destination in form of city and the culture image of those city.



Fig. 11. Detail Room's Page

Figure 11 above is the interface of detail room which is going to show all room's attributes. In this page also display order panel so user can order the room from those panel.



Fig. 12. User's room listing

Figure 12 above is the interface of user's room listing. This page is shown the list of room that user own with the action button edit and delete.



Fig. 13. Add Room's Interface

Figure 13 above is the interface of Add Room. This page is the place for room hoster to register their room. In this page shows the forms that need to be fill by the room owners if they are going to register their room. The forms divided into several sections basic information, room photos, location and room amenities.

D. Transition Phase

This section will explain about transition phase which is doing some test to Rumanatra web application. This research is trying to implement one of several testing techniques, that is black box testing. Black box testing also known as

behavioral testing, a software testing method which the internal structure / design / implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional [7].

This research which is focuses on how we measure the performance of Single Page Application, will do some test of performance on Rumantara website. The test will measure amount of time needed to access several pages. The pages will describe in the table below.

TABLE II. PAGE TEST SCENARIO

Page Code	Page Name	Description
P.1	Home	The first page that will be loaded
P.2	Login	The second page that will be loaded.
P.3	User Dashboard	The third page that will be loaded.
P.4	User Reservation	The fourth page that will be loaded.
P.5	Detail Room	The Fifth page that will be loaded.

Table 2 above explain about the flow of page that will be accessed during tests. These pages not only accessed but also we try to fill all available form like login which is already hardcoded, so it's only need to be submitted, no need to type the forms value. The test will be executed twice so we can know the difference between first time access and already accessed, because most of browsers able to save the website documents like javascript, css and html files to their cache or memory.

TABLE III. TEST RESULT OF FIRST TIME BROWSER LOADS

No	Page ID	Load Time	Description
1	P.1	4.3s	First time browser loads, its need to download the resource files and then save it to browser cache.
2	P.2	0.2s	Quite fast because the website only getting the data, rendering process on client side.
3	P.3	0.3s	Quite fast because the website only getting the data, rendering process on client side.
4	P.4	0.4s	Quite fast because the website only getting the data, rendering process on client side.
5	P.5	0.5s	Little bit slow because after getting the data, website need to download images of room.

Table 3 above is explained about the result of performance test by look at how long the browser need to load the website for the first time.

TABLE IV. TEST RESULT OF SECOND TIME BROWSER LOADS

No	Page ID	Load Time	Description
1	P.1	1.2s	The second times browser loads the website, it is quite fast because browser already have the cached resources files.
2	P.2	0.2s	Quite fast because the website only getting the data, rendering process on client side.

3	P.3	0.4s	Quite fast because the website only getting the data, rendering process on client side.
4	P.4	0.3s	Quite fast because the website only getting the data, rendering process on client side.
5	P.5	0.4s	Little bit slow because after getting the data, website need to download images of room.

Table 4 above explained about result of how long the browser need to load the website for the second time. We can summary that using Single Page Application concept can improve performance of the website. Single Page Application make website feel more like desktop application because of its speed.

IV. CONCLUSION

Conclusion from this research are as follows:

1. By using web application for room rentals can help travelers to find the best place to stay that fits with budget and needs, also help the economy of people who rent their empty room to Rumantara.
2. This development of the application, collaborate with other developer's divisions to create a whole finished application.
3. The good system architecture is not only scalable and saving cost, it also to support the developers and client side.
4. Single Page Application is a concept that makes performance better which only load data from the server then render it only from the client-side without refresh the page.

REFERENCES

- [1] BPS, "Lapbul," agustus 2016. [Online]. Available: <https://www.bps.go.id/Subjek/view/id/16#subjekViewTab3>
- [2] . Codeschool. (n.d.). *Single Page Applications - Codeschool*. Retrieved from <https://www.codeschool.com/beginners-guide-to-web-development/single-page-applications>
- [3] Mikowski, M. S., & Powell, J. C. (2013). Single page web applications. *B and W*.
- [4] LinkedIn, "LinkedIn Blog," LinkedIn, 2017. [Online]. Available: <https://engineering.linkedin.com/blog/2017/02/measuring-and-optimizing-performance-of-single-page-applications>. [Accessed 2017].
- [5] Larman, C. (2012). Applying UML and Patterns: An Introduction to Object Oriented Analysis and Design and Iterative Development. India: Pearson Education.
- [6] Brainvire. (n.d.). *Why React.JS is Preferable for Web Application Development?* Retrieved from https://www.brainvire.com/react-js-preferable-web-application-development/?utm_content=bufferf7ed1&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer
- [7] Software Testing Fundamentals. (n.d.). *Blackbox Testing*. Retrieved from <http://softwaretestingfundamentals.com/black-box-testing/>