

Development of Front-End Web Applications Utilizing Single Page Application Framework and React.js Library

Ricky Jonathan *

Informatics Engineering Study Program, Faculty of Information Technology, Universitas Kristen Satya Wacana, Salatiga City, Central Java Province, Indonesia.

Email: rickjo.coder@gmail.com

Suprihadi

Informatics Engineering Study Program, Faculty of Information Technology, Universitas Kristen Satya Wacana, Salatiga City, Central Java Province, Indonesia.

Email: suprihadi@uksw.edu

Received: 8 December 2023; Accepted: 15 December 2023; Published: 30 December 2023.

Abstract: In an era marked by rapid technological advancements, the impact of these developments is profoundly significant, particularly in the context of the ubiquity and indispensability of the internet. Motivated by this modern landscape, this study focuses on employing the Single Page Application (SPA) technique for the development of the Mbanu website. Central to this research is the utilization of the React.js library. The methodology encompasses several key phases: a comprehensive literature review, thorough analysis of the problem domain, design and prototyping, and the implementation of React.js for front-end development. The application of the SPA technique is observed to offer notable benefits for developers, including enhanced efficiency and time savings in the web development process. The findings presented in the Results and Discussion section demonstrate that the incorporation of React.js substantially contributes to the development of Mbanu's front-end architecture. Rigorous unit testing is performed on each component to ensure robustness and error minimization. The study concludes with suggestions for future research aimed at expanding the Mbanu web application's feature set to further improve user experience.

Keywords: React.Js; Single Page Application; Front-End.

1. Introduction

Indonesia, being the fourth most populous country in the world, faces the imperative of promoting equitable national development to ensure the welfare of its people. This is particularly crucial in the face of economic challenges, which often lead to social disparities between the affluent and the impoverished. Consequently, the government frequently intervenes through social assistance programs distributed via local social agencies. However, the vast number of beneficiaries and the emergence of organizations claiming humanitarian motives have led to new challenges, including the manipulation of aid funds for personal or organizational gain. This issue has caused public unrest regarding the trustworthiness of social organizations in handling aid funds. According to Transparency International Indonesia (TII) and Indonesia Corruption Watch (ICW), inaccuracies in data have resulted in some affected populations not receiving COVID-19 social assistance, highlighting the potential for misappropriation of funds, as exemplified by the corruption scandal involving former Social Minister Juliari Batubara (BBC Indonesia, 2021).

With the rapid advancement of technology, particularly in an era where instant solutions are sought and internet usage is ubiquitous, there has been a significant increase in online activities. A survey by the Indonesian Internet Service Provider Association (APJII) for the period 2019-II/2020 indicates that the number of internet users in Indonesia reached 196.7 million, a 8.9% increase from 2018, with the highest usage in regions like West Java and Jakarta [1]. This surge in internet usage indirectly necessitates the development of web technologies. Traditionally, web development has relied on HTML (Hypertext Markup Language) for structuring web pages and CSS (Cascading Style Sheets) for styling and layout. These technologies, combined, create comprehensive web pages accessible on various devices, including mobile and desktop [2].

In advancing technology, meticulous preparation is vital to ensure that the developed technology serves its intended purpose effectively. Systematic planning in development minimizes potential errors during the process. In developing the Mbantu Web, careful consideration was given to the selection of appropriate technologies to not only reduce errors but also facilitate ease in the initial development stages. This led to the decision to use the Single Page Application (SPA) technique for the Mbantu website. Mbantu, a startup under an educational company, aims to assist the public in accessing various forms of social assistance, including financial, logistical, educational, and more. The choice of SPA is driven by its benefits, such as providing an optimal user experience with minimal page load times. Studies show that 20% of users abandon a website if its loading time exceeds three seconds [3]. Unlike Multi-Page Applications (MPA), SPA does not require reloading the entire page for each user action, allowing for faster rendering and quicker page loads. In the realm of web development, numerous frameworks and libraries such as Angular, Svelte, Vue, and React offer varied options for developers. This research will employ React.js, chosen for its feature of reusable components, which facilitates easier development and maintenance of React applications [4].

2. Research Method

The research methodology for the development of the Mbantu Web Application using Single Page Application (SPA) and React.js Library involves several key stages:

- 1) Literature Review
The initial phase involves an extensive literature review to gather supporting references from journals and websites related to Single Page Applications and React.js. This stage serves as a learning process for the researcher to comprehend the concepts of SPA, which aids in problem analysis, system design, and implementation.
- 2) Problem Analysis
This stage focuses on identifying and analyzing the emerging problems and determining the necessary requirements for the application development process. Information gathered during this phase includes system specifications, desired features, and user needs. This data forms the foundation for designing the user interface and backend system requirements.
- 3) Design and Prototyping
In this phase, formulated solutions are translated into a design, followed by the development of a prototype. The process begins with the initial design of the prototype, creation of the prototype, evaluation, and iterative refinements based on feedback and analysis of requirements.
- 4) Implementation
This stage involves the development of the web application from the prototype using tools like the Visual Studio Code editor and employing React.js as the technology for web application development.
- 5) Conclusion and Recommendations
The final stage encompasses drawing conclusions and presenting findings obtained from the implementation process. This phase also includes conducting tests on the developed application and providing recommendations based on the outcomes and insights gained during the research.

3. Result and Discussion

3.1 Results

3.1.1 Development Methodology

The application was developed using the design thinking method, which involved collaboration with team members and focusing on user perspectives. This approach aimed to develop a product tailored to user needs, ultimately leading to satisfactory results.

3.1.2 Workflow

The first step involved mapping the application's workflow. This phase was crucial for understanding the user's interaction process with the application.

3.1.3 Design and Prototype

The application's design was crafted using Figma. This stage facilitated the transition of the design into code. A prototype was created based on the application flow and tested with users to gather feedback for further refinement.

3.1.4 Development Technology

The application was developed using the React-Redux package for state management. Redux was chosen to avoid props drilling, allowing global state management through a centralized store, thus simplifying access and modification of state data.

3.2 Implementation

3.2.1 Development Method

The first stage involved implementing the design thinking methodology. This approach was expected to yield a targeted and high-quality solution, as it enabled the development team to focus on user experience and generate a multitude of ideas.

1) Empathize

The initial phase involved understanding the problems faced by users, formulating their needs, and gaining a deeper insight into the issues through empathy, observation, and engagement with the target audience.

2) Define

This phase involved analyzing information gathered during the Empathize stage. It focused on identifying the core problems and developing a comprehensive understanding of these issues by breaking down assumptions into pain points.

3) Ideate

This was the ideation stage, where ideas generated during the Define phase were collected for problem-solving. Prioritization of ideas and potential designs were also determined at this stage.

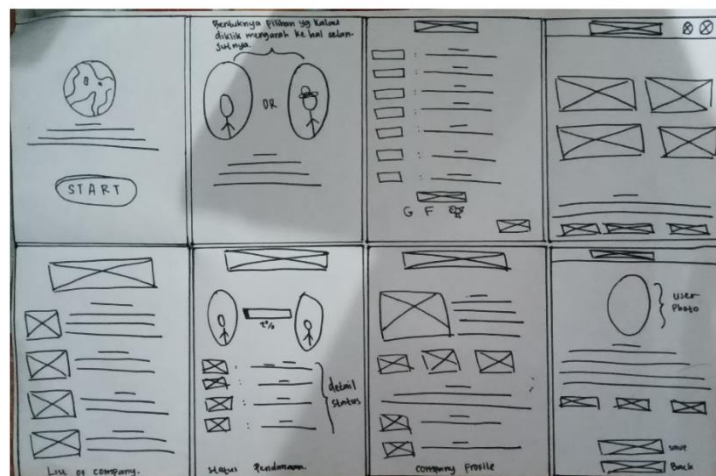


Figure 1. Crazy 8's

4) Prototyping

The subsequent stage was prototyping. Here, the devised solutions were shaped into a tangible prototype, which was then tested with users to gather feedback and make necessary revisions.

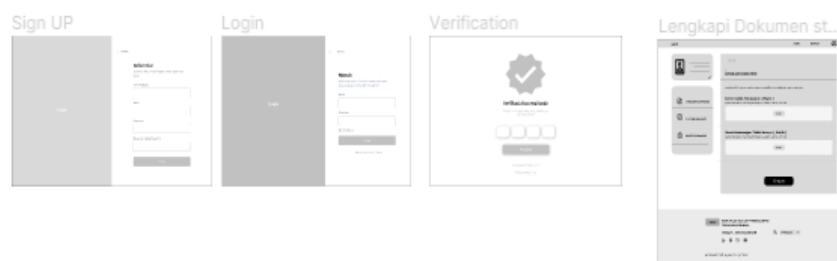


Figure 2. Prototyping

3.2.2 Workflow

The workflow developed for the application is illustrated in the next figure, detailing each step from user login to completion of their primary task. As for the workflow, the user enters the home page, the user views or seeks help, the user selects available assistance, the user views the requirements, the user makes a submission, but if the user is a new user, he will be redirected to the registration page, but if the user has already registered, the user will be redirected to the registration page. can continue, the User completes the document, the User waits for the status of receiving assistance, and the User receives assistance.

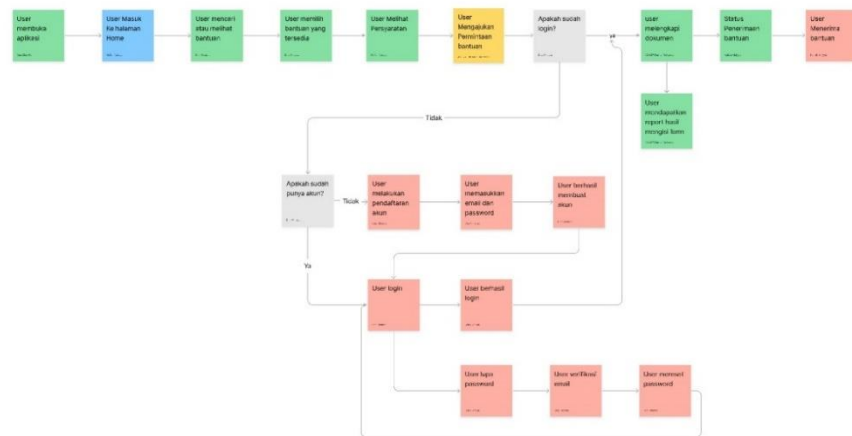


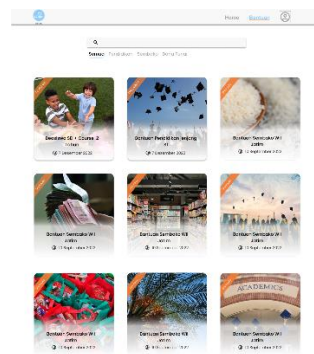
Figure 3. Workflow

3.2.3 Design & Prototype

The design aimed for simplicity and user-friendliness. The creation of the design and prototype was intended to streamline the development process, providing developers with a clear vision of the application's design and operational flow.



(a) Homepage



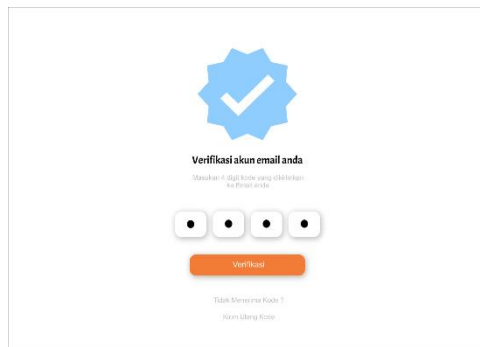
(b) Assistance Page



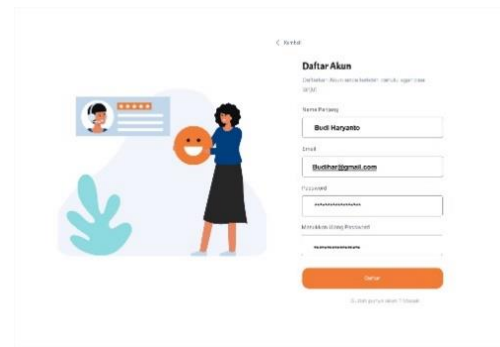
(c) Assistance Detail Page



(d) Complete Documents Page



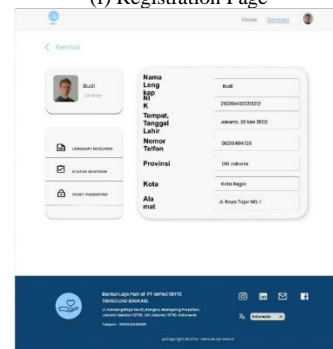
(e) Verification Page



(f) Registration Page



(g) Login Page



(h) Profile

Gambar 4. Design & Prototype

The homepage features a clean and intuitive design. At the top, there's a navigation bar for easy access to different sections. The main area of the page showcases the core services of the Mbantu application, with visually appealing graphics and succinct descriptions. A search bar is prominently placed, allowing users to quickly find the services they need (Figure 4.a Homepage). The Assistance Page lists available support options in a well-organized format. Each assistance type is represented with an icon and a brief description. Users can click on any of the assistance types to learn more about them. This page is designed to provide a quick overview of all the support services offered by Mhelp (Figure 4.b Assistance Page). On the Assistance Detail Page, users find comprehensive information about a specific type of assistance. This includes eligibility criteria, application procedures, and deadlines. The page layout is designed for clarity, ensuring that users can easily understand and follow the necessary steps to apply for assistance (Figure 4.c Assistance Detail Page). The Complete Documents Page guides users through the process of uploading necessary documentation. It features a simple upload interface and provides clear instructions on the types and formats of documentation required. The page is designed to minimize user confusion and streamline the document submission process (Figure 4.d Complete Documents Page). The Verification Page is designed to ensure user identity and security. It may include steps for email or phone verification, with clear instructions and a straightforward interface. This page is crucial for maintaining the integrity and security of the application and its users (Figure 4.e Verification Page). The Registration Page is user-friendly and invites new users to create an account. It includes fields for essential information like name, email, and password, along with privacy policy and terms of service agreements. The design emphasizes ease of use, encouraging more users to join the platform (Figure 4.f Registration Page). The Login Page is straightforward, with fields for username and password. There's also a feature for password recovery and a link to the registration page for new users. The page design balances simplicity and security, providing a seamless login experience (Figure 4.g Login Page). The Profile Page allows users to view and edit their personal information. It includes sections for contact details, preferences, and historical data of assistance received or applied for. The layout is clean and user-centric, enabling easy navigation and updates to personal information (Figure 4.h Profile).

3.3 Development Technology

The application of Redux in the Mbantu Website development began with the creation of a store, which aimed to consolidate all application states, avoiding props drilling and maintaining a global state. The process involved combining multiple reducers into one store using the Combine Reducer function, and the store was exported for use where needed. The reducer's role in state changes according to given actions and the actions themselves in altering the state within the reducer are also elaborated.

```

import { createStore, combineReducers } from "redux";
import bantuanReducer from "../BantuanReducers";
import thunk from "redux-thunk";
import { applyMiddleware } from 'redux'

const allReducers = combineReducers({
  bantuan: bantuanReducer,
})

const store = createStore(allReducers, applyMiddleware(thunk))

export default store

```

Figure 5. Store.js

This store aims to unify all the states of the application. So having a store can avoid props drilling because of its global nature. In Figure 5 you can see that allReducer contains a reduction named help. To be able to use the reduction, you have to call the reduction by importing it first, then the Combine Reducer itself is a function that is used to combine several reductions into one store. So, when there are many reducers you have to use the Combine Reducer to place them in one store. The next step is to pass AllReducer into the store accompanied by Thunk as middleware. Then we export the store itself so that it can be called where needed.

```

import { FETCH_START, SUCCESS_GET_DATA } from "../Action/BantuanAction";

const initialState = {
  bantuan: [],
  isLoading: false,
  err: null
}

const bantuanReducer = (state = initialState, action) => {
  switch (action.type){
    case FETCH_START:
      return{
        ...state,
        isLoading: true
      }
    case SUCCESS_GET_DATA:
      return{
        ...state,
        bantuan: action.payload,
        isLoading : false
      }
    default:
      return state;
  }
};

export default bantuanReducer;

```

Figure 6. Reducer

An overview of the Reducer can be seen in Figure 16 where the Reducer itself contains how the status changes according to the action given. In Figure 6, the fetch start case itself provides an action that changes the isLoading state from the initial state of false to true, then there is also the Successful Get Data case which contains an action that changes the isLoading state back to false and fills the help array with the payload or data. submitted by action Success Get Data.


```

import axios from 'axios';

export const GET_DATA = "GET_DATA";
export const FETCH_START = "FETCH_START";
export const SUCCESS_GET_DATA = "SUCCESS_GET_DATA";

function fetchStart(){
  return{
    type: FETCH_START,
  };
}

function successGetData(data){
  return{
    type: SUCCESS_GET_DATA,
    payload: data,
  };
}

export const getData = () => {
  return async (dispatch) =>{
    dispatch(fetchStart());

    const result = await axios.get(
      "https://6364e197b289ec4758f09b.mockapi.io/bantuan"
    );
    dispatch(successGetData(result.data));
  };
};

```

Figure 7. Action

An overview of the action can be seen in Figure 7. In the SuccessGetData function, the SuccessGetData property as the type of action to be performed will be returned with a "payload:data" object which contains information obtained from the api to change the state in the reducer. The getData() function has a function to fetch data from the API using dispatch to carry out the fetchstart action by bringing in information obtained from the API.

3.4 Discussion

The implementation of the Mbanu web application, as elucidated in the previous sections, provides valuable insights into the process of developing a user-centric and efficient digital platform. This discussion delves into the key aspects of the application's development, the challenges encountered, and the strategies adopted to address these challenges, culminating in the successful deployment of the application. At the heart of Mbanu development was a user-centric design philosophy. The emphasis on simplicity and intuitiveness in the design and prototype phases paid significant dividends in terms of user engagement and satisfaction. By streamlining the user journey from the homepage to specific functionalities like assistance details and document completion, the application effectively reduced user friction and enhanced the overall user experience. The application of the design thinking methodology fostered a collaborative environment. The involvement of various stakeholders, including developers, users, and subject matter experts, in the empathize and define stages, ensured that the application was not only technically sound but also aligned with the real needs of its target audience. This collaborative approach was instrumental in identifying and addressing the core requirements of the users. One of the more significant challenges faced during the development phase was state management in the application, particularly given the dynamic nature of user interactions. The decision to implement Redux as a state management solution proved to be effective in managing the global state of the application, thereby ensuring a seamless and consistent user experience. This choice underscored the importance of selecting the right technological tools to address specific challenges in web development. Another critical aspect of the discussion revolves around performance optimization. The Single Page Application (SPA) model adopted for Mbanu significantly reduced loading times, enhancing the application's responsiveness. This approach, coupled with the efficient use of React.js, ensured that the application was not only fast but also scalable. The iterative design and development process, characterized by continual prototyping and user testing, allowed for the rapid integration of user feedback. This process was vital in refining the application's features and interface, ensuring that the final product resonated well with the end-users. Looking forward, the development team acknowledges the potential for further enhancements in the application. The possibility of integrating advanced features such as AI-driven recommendations and more personalized user experiences are areas of interest. Additionally, continuous user feedback will remain a cornerstone of future updates, ensuring that Mbanu evolves in line with the changing needs of its users. The development and implementation of the Mbanu web application exemplifies the effective combination of user-centric design, collaborative development, and technical proficiency. The lessons learned and the successes achieved provide a valuable blueprint for future web application projects, particularly those aimed at delivering social assistance in an efficient and accessible manner.

4. Related Work

Mbanu web application development leverages a wealth of existing literature and research in the field of web development and specific technologies such as React.js and Redux. Understanding State Management with Redux, Geary (2016) article "Introducing Redux" published in IBM DEVELOPER provides a basic understanding of Redux as a state management tool. These resources play an important role in informing the decision to deploy Redux in Mhelp applications to manage application state in a predictable manner [5]. Geary explanation of Redux's core principles and their application in complex web applications provided valuable insight into our development process. Ham (2019) discussion of "What

is React.js?” from the School of Computer Science at Binus University serves as an important resource in understanding the capabilities and advantages of React.js for building user interfaces [6]. Iswari (2021) study on the Application of React JS in FrontEnd Development of Ubaform Startup Applications is relevant in highlighting the practical implementation and challenges in a context like Mbantu, thereby offering practical insights into the real-world application of React.js in startup development [7]. Patel (2021) article on “What is a Website?” from Geeks For Geeks is a fundamental resource for understanding the basics and ins and outs of web applications [8]. This resource provides a broad overview of web application architecture and its evolution, laying the foundation for the conceptual framework behind the Mbantu application. Research by Solovei *et al.* (2018) on the differences between single-page applications and traditional web applications in mechatronic robot laboratory applications, helps in understanding the advantages and disadvantages of the SPA architecture, which is crucial in deciding to adopt the SPA for Help approach [9].

5. Conclusion

Based on the results obtained at the Results and discussion stage, the implementation of React JS helps develop front-end web applications, web applications help using the Single Page Application (SPA) method can offer convenience for developers and save time in the application development process because React JS uses components that can be reused and can be produced separately into several components which can help developers find out where errors occur, so the debugging process becomes faster and faster. Apart from that, using React Redux can help developers avoid props drilling so that developers don't have to Writing code on each page is pretty much all you need to do is call the store which contains all the state of each component that requires it.

References

- [1] Jarot Bayu, D. 2020. *Jumlah Pengguna Internet di Indonesia Capai 196,7 Juta*. Katadata. Available at: <https://databoks.katadata.co.id/datapublish/2020/11/11/jumlah-pengguna-internetdi-indonesia-capai-1967-juta>
- [2] Santoso, M.F., 2021. Teknik Single Page Application (Spa) Layout Web Dengan Menggunakan React Js Dan Bootstrap. *Jurnal Khatulistiwa Informatika*, 9(2), pp.107–114. DOI: <https://doi.org/10.31294/jki.v9i2.11357>.
- [3] D, A. 2022. *Single-page App vs Multi-page App: Choosing a Way to deliver the best user flow*. CLEVEROAD. Available at: <https://www.cleveroad.com/blog/single-page-app-vs-multi-page-application-what-to-choose/>
- [4] Sharma, I. 2022. *BENEFITS OF REACT JS FOR YOUR FRONT-END DEVELOPMENT PROCESS*. TatvaSoft. Available at: <https://www.tatvasoft.com/outsourcing/2022/02/benefits-of-react-js.html>
- [5] Geary, D. 2016. *Introducing Redux*. IBM DEVELOPER. Available at: <https://developer.ibm.com/tutorials/wa-manage-state-with-redux-p1-david-geary/>
- [6] Ham, Hanry. 2019. *Apa itu React.Js?* School of Computer Science Binus University. Available at: <https://socs.binus.ac.id/2019/12/30/apaitu-react-js/>
- [7] Iswari, L., 2021. Penerapan React JS Pada Pengembangan FrontEnd Aplikasi Startup Ubaform. *AUTOMATA*, 2(2).
- [8] Patel, nitish kumar. 2021. *What Is a Website ?* Geeks For Geeks. Available at: <https://www.geeksforgeeks.org/what-is-a-website/>.
- [9] Solovei, V., Olshevskaya, O. and Bortsova, Y., 2018. The difference between developing single page application and traditional web application based on mechatronics robot laboratory onaft application. *Автоматизация технологических и бизнес-процессов*, (10, Iss. 1), pp.4-8.