

Fetching dan Display Data API Menggunakan Library AXIOS Pada ReactJS

Gat¹

¹Sistem Informasi
STMIK Pontianak
Pontianak, Indonesia
e-mail: gat@stmikpontianak.ac.id

Abstrak

Membangun aplikasi sering kali membutuhkan interaksi dengan API untuk mengirimkan dan mengambil data. Setiap framework memiliki cara dan teknik pengkodean yang berbeda dalam berinteraksi dengan API. Axios adalah salah satu sintaks yang sangat sederhana di reactJS untuk berinteraksi dengan API dan secara otomatis mengubah data JSON. Beranjak dari keunggulan dari axios, penelitian ini bermaksud untuk meningkatkan kinerja sistem pada saat melakukan permintaan dan respon data melalui REST APIs. Data API yang dipergunakan dalam penelitian ini adalah *tribunnews.com*. Perancangan sistem menggunakan pendekatan berorientasi objek karena aplikasi di reactJS dibangun di atas komponen. Berdasarkan hasil pembahasan, library axios dapat digunakan untuk melakukan permintaan dengan metode *axios.get()* melalui URLs dengan kinerja yang baik dari Fetch API. Axios secara drastis mengurangi jumlah boilerplate yang dibutuhkan untuk melakukan permintaan ke API dan inilah yang menyebabkan request dan respon teknik Axios memiliki kinerja yang lebih baik dari sisi waktu. Dalam melakukan penanganan terhadap kesalahan, axios menggunakan *error boundary*.

Kata kunci API, JSON, ReactJS, Komponen.

Abstract

Building applications often requires interaction with APIs to send and retrieve data. Each framework has a different way and coding technique of interacting with the API. Axios is one of the very simple syntaxes in reactJS for interacting with the API and automatically converting JSON data. Moving on from the advantages of axios, this research intends to improve system performance when requesting and responding to data through REST APIs. The API data used in this research is *tribunnews.com*. System design uses an object-oriented approach because applications in reactJS are built on components. Based on the results of the discussion, the axios library can be used to make requests with the *axios.get()* method via URLs with good performance from the Fetch API. Axios drastically reduces the amount of boilerplate required to make requests to the API and this is what causes Axios' request and response techniques to perform better in terms of time. In handling errors, Axios uses an *error boundary*.

Keywords: API, JSON, ReactJS, Component.

1. Pendahuluan

Membangun website dengan framework PHP seperti CodeIgniter dan Laravel sudah menjadi hal yang umum bagi kalangan web developer [1]. Masing-masing dari jenis framework tersebut menawarkan kelebihannya, baik itu dari sisi kemudahan penggunaan maupun dari sisi ketersediaan fiturnya. Framework CodeIgniter dan Laravel memang sudah terbukti menjadi pilihan bagi web developer dan bahkan memiliki popularitas yang teratas dari framework PHP lainnya [2]. Namun berbeda dengan ReactJS yang baru dikenal sebagai salah satu library untuk membangun web dari sisi frontend. Sejak diperkenalkan tahun 2013, ReactJS dikatakan sebagai library Java Script open source yang paling populer untuk pengembangan web [3]. Dengan segala kelebihan yang ditawarkan, seperti component reusability, virtual DOM, SEO friendly serta dukungan komunitas yang besar, telah membuat ReactJS menjadi pilihan utama dalam pengembangan aplikasi frontend [4]. React JS telah menjadi salah satu pustaka UI (User Interface) terpenting dalam Javascript yang dikelola oleh perusahaan ternama didunia seperti Facebook dan Instagram bersama dengan komunitas pengembang dari seluruh dunia [5].

Sebagai salah satu library yang berfungsi untuk merancang UI dari sisi frontend[6], React JS tidak mengakses secara langsung database server, melainkan melalui REST APIs [7]. Ini berbeda dengan framework PHP lainnya dimana untuk mengirim atau menerima data dari server langsung menggunakan

fetch. React JS menggunakan Axios yang merupakan sebuah library open source yang berfungsi untuk menangani HTTP request dan responses [8]. Library Axios bisa mengatasi berbagai masalah yang akan terjadi pada saat pengiriman data tanpa memerlukan perintah khusus untuk mengatasi pesan error tersebut. Mengingat banyak kelebihan yang ditawarkan ReactJS, maka penelitian ini merasa perlu melakukan pendalaman terhadap cara kerja React JS terutama penggunaan Axios dalam pengambilan dan menampilkan data melalui REST API [9]. Untuk mendalami cara kerja Axios di ReatJS, penelitian ini menggunakan data dari public API yaitu tribunnews.com yang dapat diakses melalui endpoint yang sudah disediakan. Penelitian ini diharapkan bisa memberikan pemahaman yang lebih mendalam bagi web developer terhadap penggunaan axios dalam mengambil dan menampilkan data dari API dengan menggunakan method GET.

2. Metode Penelitian

Metode penelitian ini menggunakan Object-Oriented Analysis and Design Methodology (OOADM). OOADM adalah pendekatan teknis untuk menganalisis dan merancang aplikasi, sistem, atau bisnis dengan menerapkan pemrograman berorientasi objek, serta menggunakan pemodelan visual di seluruh proses pengembangan perangkat lunak [10]. Analisis dan desain berorientasi objek adalah cara paling umum untuk memanfaatkan strategi lokasi objek untuk merencanakan kerangka kerja [11]. Antarmuka aplikasi Web lebih kompleks dari pada di sistem perangkat lunak tradisional, navigasi dan fungsionalitas harus terintegrasi dengan baik dan struktur navigasi harus dipisahkan dari model domain aplikasi, oleh karena itu OOADM dipilih karena fungsinya yang memungkinkan untuk menganalisis dan desain aplikasi web. Fokus pada model ini cenderung lebih ke arah perilaku sistem dan fitur utama yang didokumentasikan di sini adalah kelas dan objek [12]. Pemodelan berorientasi objek yang dipergunakan adalah *Unified Modeling Language (UML)* sebagai salah satu alat terbaik dalam pengembangan perangkat lunak dan mewakili interaksi antara pengguna dan sistem informasi yang disajikan [13], [14].

3. Hasil dan Pembahasan

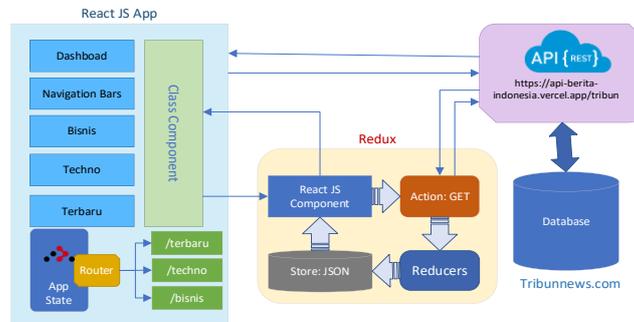
Membuat panggilan API merupakan bagian integral dari sebagian besar aplikasi saat ini dan Axios adalah klien HTTP populer yang tersedia sebagai pustaka JavaScript. Penelitian ini hanya menghasilkan aplikasi pada sisi front-end dengan menggunakan reactjs dan bootstrap. Aplikasi yang dihasilkan hanya menampilkan informasi berita dari tibunnews.com dimana berita ini diakses dari API (<https://api-berita-indonesia.vercel.app/tribun/>). Aplikasi ini tidak melakukan pengolahan data lebih dan hanya membuktikan bagaimana library axios mampu membuat permintaan GET dan menampilkan data API melalui reactjs. Metode get() membutuhkan dua parameter, pertama membutuhkan URI layanan endpoint dan kedua harus melewati objek yang berisi properti yang untuk dikirim ke server. Axios dapat membuat panggilan API dari aplikasi JavaScript terlepas dari apakah JavaScript berjalan di front-end seperti browser atau sisi server. Keuntungan Axios terletak pada promise-based, sehingga memungkinkan implementasi kode secara asinkron. Kode asinkron akan memungkinkan halaman memuat beberapa elemen secara bersamaan, bukan berurutan, sehingga mengurangi waktu pemuatan secara signifikan. Komponen pada ReactJS bekerja dengan Virtual DOM dan bersifat reuseable sehingga komponen yang sudah dibuat dapat dipergunakan kembali di aplikasi lain. Komponen pada Virtual DOM berfungsi untuk memberitahu React tentang apa saja yang harus ditampilkan (render) ke Real DOM (HTML). Berikut ini adalah daftar class components yang dipergunakan dalam penelitian ini.

Tabel 1. Daftar class components

No.	Nama Class Component	Keterangan
1	class Bisnis extends Component {}	Mengelola state bisnis
2	class Terbaru extends Component {}	Mengelola state terbaru
3	class Techno extends Component {}	Mengelola state techno
4	class Menu extends Component {}	Mengelola state menu
5	class Dashboard extends Component {}	Mengelola state dashboard

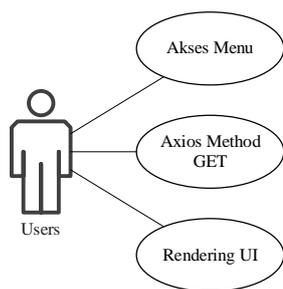
Class Component pada tabel 1 di atas memiliki metode render yang dipergunakan untuk menampilkan UI (User Interface). Render komponen bisnis dipergunakan untuk menampilkan informasi bisnis, render komponen terbaru dipergunakan untuk menampilkan informasi terbaru, render komponen techno dipergunakan untuk menampilkan informasi techno, render komponen menu dipergunakan untuk menampilkan informasi menu dan render komponen dashboard dipergunakan untuk menampilkan informasi dashboard. Selain class component yang diperlukan dalam menghasilkan aplikasi, reactJS membutuhkan library seperti React Router DOM dan axios. Router diperlukan menentukan komponen mana yang akan dijalankan. Dalam penelitian ini adalah 4 route yaitu home, terbaru, techno dan bisnis. Arsitektur API yang dipergunakan adalah REST (Representational State Transfer) dengan bentuk datanya

adalah JSON (JavaScript Object Notation). Dari sejumlah data JSON yang terdapat pada API, penelitian ini hanya mengambil data berita dari tribunnews.com. Data yang diambil hanya untuk kategori berita terbaru, berita bisnis dan berita teknologi. Dalam arsitektur REST, terdapat beberapa metode yang didukung oleh http, namun dalam penelitian ini hanya menggunakan metode GET Requests dengan Axios. Penggunaan method GET untuk menampilkan semua informasi dari API sesuai dengan endpoint masing-masing. Rancangan arsitektur aplikasi dalam penelitian ini melibatkan banyak komponen terutama pada sisi komponen reactJS dan hubungannya dengan API. Berikut ini adalah Gambar 1 arsitektur sistem yang dikembangkan:

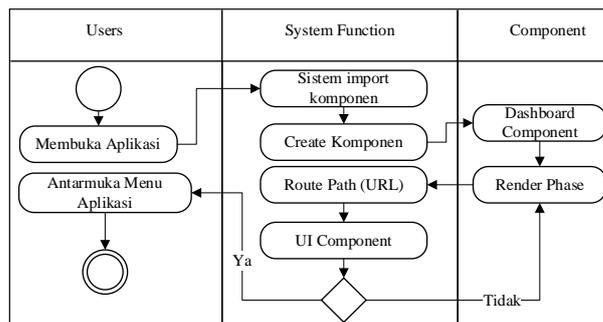


Gambar 1 Arsitektur Sistem

Berdasarkan rancangan arsitektur sistem pada Gambar 1 di atas, dapat dijelaskan bahwa class component pada reactJS akan melakukan aksi ketika router mengarahkan kepada salah satu dari 3 endpoint. Ketika ada pemanggilan salah satu komponen, maka axios akan melakukan method GET untuk meminta data dari API. Dari sisi client akan melakukan HTTP request ke REST server melalui URLs dan kemudian rest server akan merespon dengan mengirim kembali HTTP yang dimintal oleh client. Penggunaan use case diagram dianggap penting untuk mengilustrasikan fungsi-fungsi utama dari sistem dan pengguna yang akan berinteraksi dengan sistem tersebut. Dalam pengembangan aplikasi fetching dan display data API, use case diagram terdiri dari satu actor dan tiga use case. Seperti yang tergambar pada gambar 2 (a) di bawah, seorang actor ketika berinteraksi dengan sistem secara sederhana melakukan 3 kegiatan utama yaitu membuka menu aplikasi, melakukan GET data dan rendering data melalui user interface. Penggunaan activity diagram berfungsi untuk menggambarkan aktivitas-aktivitas utama dan hubungan-hubungan antar aktivitas dalam sebuah proses. Ada beberapa hal yang harus dipersiapkan dalam sebelum merancang activity diagram yaitu pahami proses bisnis, lakukan identifikasi aktivitas-aktivitas, identifikasi control flows dan nodes, identifikasi object flows dan nodes, serta merancang dan menggambar diagram. Dalam penelitian ini ada 2 activity diagram yaitu activity akses menu, activity axios method get dan rendering UI.

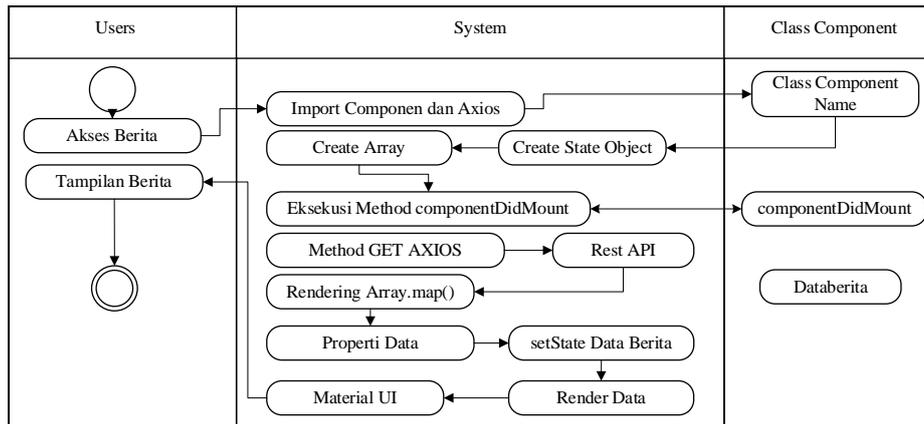


Gambar 2 (a) Use Case Diagram Sistem



Gambar 2 (b) Activity Akses Menu

Pada gambar 2 (b) di atas dapat dijelaskan bahwa untuk menjalankan aplikasi, seorang pengguna harus melalui antar menu. Ketika pengguna mengakses menu, maka sistem akan melakukan import terhadap komponen yang diperlukan seperti komponen react-router-dom, react, dan komponen file (menu.jsd, home.js, terbaru.js, bisnis.js dan techno.js). Setelah kebutuhan import komponen, selanjutnya sistem membuat class component bernama dashboard. Pada bagian class component, dilakukan fase render untuk menghasilkan user interface menu aplikasi. Pada fase render juga ada penentuan rute sesuai dengan path yang sudah ditentukan. Jika pada salah satu path dipilih, maka sistem akan mengakses komponen dan jika tidak maka sistem melakukan render ulang.



Gambar 3 Activity Diagram Axios Method GET dan Rendering UI

Pada gambar 3 di atas dapat dijelaskan bahwa ketika pengguna mengakses salah satu menu berita, maka sistem akan melakukan import komponen yaitu komponen react dan axios. Selanjutnya sistem membuat class component. Sistem melanjutkan proses dengan membuat state yang berfungsi untuk menyimpan data pada React dan akan di render atau muat ulang ketika data mengalami perubahan. State yang dibuat bertipe array dan hal ini disesuaikan dengan data JSON yang tersusun dalam format multiplearray. Selanjutnya sistem mengeksekusi method componentDidMount yang merupakan fase siklus hidup React ketika pertama kali DOM di render dan pada fase ini dapat melakukan operasi sebelum elemen UI/JSX di muat. Selanjutnya menjalankan method GET untuk mendapatkan data dari API dan data yang didapat sebagai respon dari server disimpan pada properti data yang dibuat oleh axios. Setelah properti terbentuk, maka setState dijalankan untuk merencanakan suatu pembaruan ke suatu state objek komponen. Ketika state berubah, komponen merespons dengan me-render ulang. Selanjutnya dilakukan proses me-render elemen yang akan mempersiapkan apa yang mau ditampilkan pada layar melalui material user interface. Tahap terakhir adalah hasil render ditampilkan pada layar yang dapat dilihat dan dibaca oleh pengguna. Selanjutnya adalah gambar activity rendering UI.

3.1. Rancangan Sistem

Pembentukan komponen dipergunakan untuk membagi user interface menjadi bagian-bagian yang independen dan dapat digunakan kembali. Rancangan antarmuka sistem ini sederhana yaitu hanya menampilkan menu dan informasi dari setiap menu yang diakses. Rancangan lebih fokus pada bagaimana cara kerja dari library axios pada method GET. Berikut ini adalah potongan script untuk mengakses class component lifecycle reactJS.

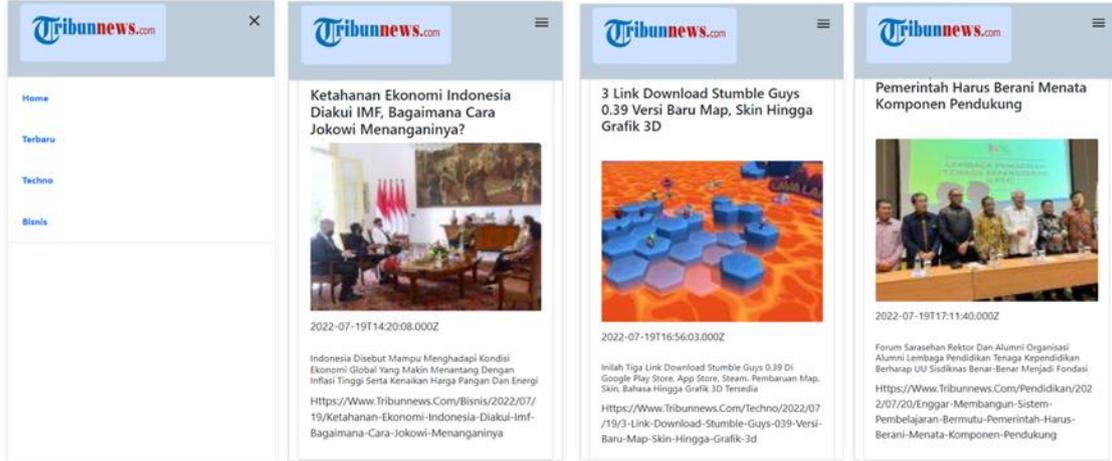
```

componentDidMount() {
  const controller = new AbortController()
  const abortSignal = controller.signal
  axios.get('https://api-berita-indonesia.vercel.app/tribun/bisnis', {signal: abortSignal})
    .then(response =>
      response.data.data.posts.map(
        newsdata => ({
          description: `${newsdata.description}`,
          title: `${newsdata.title}`,
          thumbnail: `${newsdata.thumbnail}`,
          pubDate: `${newsdata.pubDate}`,
          link: `${newsdata.link}`
        })
      )
    )
}

```

Seperti yang terlihat dalam potongan script ini di atas, terdapat fungsi untuk Cancelling Initiated Requests yaitu objek pengontrol yang menggunakan konstruktor AbortController(), kemudian menyimpan referensi ke objek AbortSignal yang terkait menggunakan properti sinyal dari AbortController. Saat permintaan axios dimulai, proses meneruskan AbortSignal sebagai opsi di dalam objek opsi permintaan {signal: abortSignal}. Ini mengaitkan sinyal dan pengontrol dengan permintaan axios dan memungkinkan untuk membatalkan permintaan dengan memanggil metode abort() pada pengontrol. Perintah axios.get(url) melakukan permintaan GET ke API dan akan mengembalikan objek yang berisi data. Method then() dipergunakan untuk memproses hasil ketika promise terpenuhi. Dalam objek respon berisi data yang diinginkan yaitu tepatnya pada array posts. Data ini berisi description, title, thumbnail, pubDate dan link.

Penangan kesalahan harus menambahkan fungsi catch() di akhir satu atau lebih fungsi then(). Selanjutnya rendering pada reactJS diperlukan untuk mengirimkan data ke dalam komponen React dengan cara menambahkan atribut ke dalam elemen. Pengisian atribut ke dalam elemen dilakukan dengan menggunakan pemisah {} dan nilai dari atribut diakses menggunakan properti post pada objek. Dari pembuatan komponen berupa file dan komponen berupa class, maka didapatkan sebuah rancangan antarmuka sistem. Berikut ini adalah Gambar 4 rancangan antarmuka aplikasi.



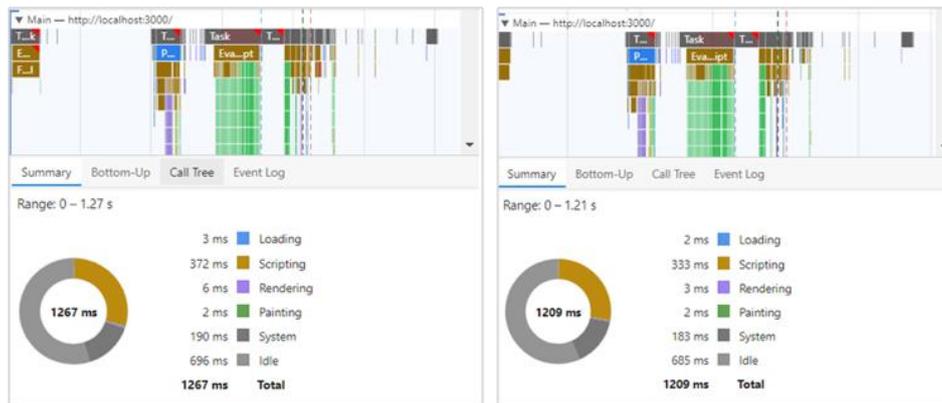
Gambar 4 (a) Menu

Gambar 4 (b) Bisnis

Gambar 4 (c) Techno

Gambar 4 (d) Terbaru

Pada empat gambar di atas, dapat dijelaskan bahwa gambar 4 (a) dihasilkan dari rancangan komponen untuk mengakses component dalam file. Gambar 4 (b) dihasilkan oleh method get.axios pada endpoint /api/bisnis. Gambar 4 (c) dihasilkan oleh method get.axios pada endpoint /api/techno dan gambar 4 (d) dihasilkan oleh method get.axios pada endpoint /api/terbaru. Penggunaan library axios dalam melakukan fetching dan display data pada API terbukti mampu mendapatkan data sesuai dengan endpoint yang terdapat pada API tersebut. Teknik request http get dengan axios juga menunjukkan kinerja yang baik melalui pencapaian waktu respon yang lebih cepat bila dibandingkan dengan teknik Fetch. Axios secara drastis mengurangi jumlah boilerplate yang dibutuhkan untuk permintaan API dan inilah alasan mengapa harus menggunakan axios. Berikut ini adalah pengujian performance requests http get dengan metode Fetch Javascript dan Axios.



Gambar 6 (a) Pengujian Teknik Fetch()

Gambar 6 (b) Pengujian Teknik Axios()

Dari pengujian seperti yang ditunjukkan pada Gambar 6 di atas dapat dijelaskan bahwa total waktu yang diperlukan teknik Fetch() untuk melakukan request http GET ke API lebih lama dibandingkan dengan teknik Axios (). Perbedaan waktu juga bisa dilihat dari sisi loading time dan rendering time. Cara kerja axios cukup sederhana yaitu cukup dengan membuat state sebagai tempat untuk menyimpan data array, melakukan GET melalui url, menyimpan hasil response dalam properti data dan me-render atribut kedalam elemen html untuk menghasilkan user interface. Axios melakukan proses request secara asinkron, sehingga membuat laman web tanpa reload. Teknik request secara asinkron pada request HTTP API dan kesederhanaan dalam sintaks seperti axios secara otomatis mengubah respons server ke objek JSON telah membuat axios semakin banyak dipergunakan oleh web developer

4. Kesimpulan

Penelitian ini menggunakan axios GET untuk membuat permintaan ke API publik. Secara detail sudah dibahas bagaimana axios berinteraksi dengan aplikasi, menjelaskan struktur permintaan dan tanggapan Axios, cara membuat permintaan ke API dan penanganan kesalahan. Membuat permintaan dasar di Axios itu mudah karena satu-satunya opsi yang diperlukan adalah url. Axios memudahkan pengiriman dan permintaan HTTP asinkron ke endpoints REST. Axios mengirim data secara asynchronous dan menggunakan properti data sehingga memungkinkan proses permintaan bisa lebih cepat. Hal ini dikarenakan teknik asynchronous tidak perlu direload pada saat mendapatkan update dan inilah yang menyebabkan permintaan menggunakan Axios lebih bagus dan cepat dari pada teknik Fetch lainnya. Dari pengujian telah membuktikan bahwa request http GET ke API lebih cepat dibandingkan dengan teknik Fetch. Aplikasi yang dihasilkan pada penelitian ini sangat sederhana dikarenakan tidak fokus pada pengembangan aplikasi yang siap dipergunakan, namun hanya membuat aplikasi yang menyediakan user interface untuk melihat hasil dari permintaan data API menggunakan library axios melalui metode GET. Untuk kedepannya perlu menambah method PUT, POST dan DELETE untuk memperluas lagi pemahaman terhadap penggunaan axios dalam berinteraksi dengan sistem sehingga axios benar-benar bisa direkomendasikan sebagai cara yang paling tepat, cepat, handal dan memiliki kinerja yang lebih baik dalam melakukan permintaan data ke API.

Daftar Pustaka

- [1] P. R. Chavan & S. Pawar, "Comparison Study Between Performance of Laravel and Other PHP Frameworks," *International Journal of Research in Engineering, Science and Management*, vol. 4, no.10, pp. 27-29, 2021
- [2] M. Laaziri, K. Benmoussa, S. Khouli & M. L. Kerkeb, "A Comparative study of PHP frameworks performance," *Procedia Manufacturing*, Tirgu Mures, Romania, vol.32, 2019, pp. 864-871.
- [3] V. Vijayakumar, "Protrack: A Student-Teacher Project Management Tool," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, vol. 12, no. 10, pp. 3173-3186, 2021. DOI: <https://doi.org/10.17762/turcomat.v12i10.4972>.
- [4] G. Baskaran, K. Saundariya, D. Prabakaran & R. Senthikumar, "A Web Application Based Administration Panel For Handyman Services," *In 2022 IEEE Delhi Section Conference (DELCON)*, New Delhi, 2022, pp. 1-5. IEEE.
- [5] P. S. Maratkar & P. Adkar, "React JS – An Emerging Frontend Javascript Library," *Conic Research And Engineering Journals (IRE)*, vol. 4, no. 12, pp. 99-102, 2021.
- [6] P. Jena, H. Daroch, P. Malik, P. Saini & R. Nagpal, "International Journal of Research Publication and Reviews," *Journal homepage: www.ijrpr.com ISSN, 2582, 7421*, vol. 3, no. 5, pp. 809-824, 2022.
- [7] S. Siddabasappa, S. S. Halikar, R. Dodda, N. Hiremath & S. Bansal, "Smart Museum Tour using Linked Open Data," *In Proceedings of the Symposium on Open Data and Knowledge for a Post-Pandemic Era ODAK22*, UK, 2022, pp. 1-8.
- [8] P. Rawat & A. N. Mahajan, "ReactJS: A Modern Web Development Framework," *International Journal of Innovative Science and Research Technology*, vol. 5, no. 11, pp. 698-702, 2020.
- [9] D. B. Duldulao & R. J. L. Cabagnot, "Writing Local States, Sending HTTP Requests, and ApexCharts," *In Practical Enterprise React*. Berkeley, CA: Apress, 2021, pp. 91-119.
- [10] A. Szewc, V. Karovič & P. Veselý, "The documentation in the project of software creation," *In Developments in Information & Knowledge Management for Business Applications*. Springer, 2022, pp. 361-441.
- [11] J. O. Jatto, U. A. Okengwu, "Case-Based Reasoning Recommender System for Diagnosing Breast Cancer Disease: Confusable Disease Using Genetic Algorithm," *International Journal of Engineering and Science (IJES)*, vol. 11, no. 4, pp. 1-9, 2022. DOI:10.9790/1813-1104010109
- [12] P. O. Uzoma, B. C. Amanze, A. O. Agbakwuru & V. C. Agbasonu, "Development of a Visual Semantic Web Ontology Based Learning Management System," *International Journal of Engineering Applied Sciences and Technology*, vol. 6, no. 10, pp. 226-238, 2022.
- [13] E. Semenova, V. Tynchenko, S. Chashchina, V. Suetin & A. Stashkevich, "Using UML to Describe the Development of Software Products Using an Object Approach," *In 2022 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, Toronto, Canada, 2022, pp. 1-4. IEEE.
- [14] S. Parusheva & D. Pencheva, "Modeling a Business Intelligent System for Managing Orders to Supplier in the Retail Chain with Unified Model Language," *In Digital Transformation Technology*, Singapore, 2022, pp. 375-393.