

IMPLEMENTASI REACT SERVER COMPONENT DAN SERVER ACTION UNTUK MENINGKATKAN PERFORMA APLIKASI WEB

(IMPLEMENTATION OF REACT SERVER COMPONENTS AND SERVER ACTIONS TO IMPROVE WEB APPLICATION PERFORMANCE)

Yudi Mulyanto¹⁾, Eka Haryanti²⁾, dan Lazarus³⁾

^{1, 2, 3)} Informatika, Rekayasa Sistem, Universitas Teknologi Sumbawa, Sumbawa, Indonesia
e-mail: yudi.mulyanto@uts.ac.id¹⁾, eka.haryanti@uts.ac.id²⁾, lazminutes@gmail.com³⁾

ABSTRAK

Penelitian ini mengeksplorasi penggunaan *React Server Components (RSC)* dan *Server Action* untuk meningkatkan performa aplikasi web, dengan fokus pada sistem penghitungan suara pemilihan legislatif di Kabupaten Sumbawa, Nusa Tenggara Barat untuk kebutuhan internal partai. Melihat tantangan geografis dan infrastruktur di Sumbawa, penelitian ini bertujuan untuk mengatasi keterbatasan sinyal dan distribusi infrastruktur telekomunikasi melalui penggunaan *RSC* dan *Server Action*. *RSC* memungkinkan server untuk merender komponen sambil mempertahankan interaktivitas di sisi klien, mengurangi beban dan mempercepat waktu muat. Kelebihannya termasuk menghindari permintaan jaringan berantai dan meminimalisir Javascript di sisi klien, yang sangat bermanfaat di daerah dengan konektivitas internet yang terbatas. *Server Action* mengeliminasi lapisan API tradisional, mengurangi latensi dan meningkatkan responsivitas aplikasi. Karena itu, tujuan dari penelitian ini adalah untuk mendesain dan menerapkan *RSC* dan *Server Action* di aplikasi web penghitungan suara, yang dibuat menggunakan metode pengembangan *extreme programming*. Pengujian menggunakan metode *black box* melibatkan auditor dan ahli IT menghasilkan kesimpulan yang berhasil dari berbagai jenis pengujian. Implementasi ini diharapkan tidak hanya meningkatkan performa aplikasi secara keseluruhan tetapi juga memastikan aksesibilitas aplikasi yang lancar di daerah dengan keterbatasan infrastruktur internet.

Kata Kunci: *react server component, server action.*

ABSTRACT

This research explores the use of *React Server Components (RSC)* and *Server Action* to improve web application performance, with a focus on the legislative election vote counting system in Sumbawa Regency, West Nusa Tenggara for internal party needs. Looking at the geo-graphic and infrastructure challenges in Sumbawa, this research aims to overcome signal limitations and telecommunication infrastructure distribution through the use of *RSC* and *Server Action*. *RSC* allows the server to render components while maintaining interactivity on the client side, reducing load and speeding up load times. Its advantages include avoiding chain network requests and minimizing Javascript on the client side, which is especially beneficial in areas with limited internet connectivity. *Server Action* eliminates the traditional API layer, reducing latency and improving application responsiveness. Therefore, the objective of this research is to design and implement *RSC* and *Server Action* in a vote counting web application, which is built using the *extreme programming* method of development. Testing using the *black box* method involving auditors and IT experts resulted in successful conclusions from various types of tests. This implementation is expected to not only improve overall application performance but also ensure smooth application accessibility in areas with limited internet infrastructure.

Keywords: *react server component, server action.*

I. PENDAHULUAN

Kabupaten Sumbawa, yang terletak di Provinsi Nusa Tenggara Barat, Indonesia, merupakan salah satu wilayah dengan keragaman geografis yang tinggi, termasuk dataran tinggi, gunung, hingga wilayah pesisir yang luas. Kondisi geografis ini menghadirkan tantangan yaitu ketidaksamaan distribusi sinyal dan infrastruktur telekomunikasi, dimana beberapa area memiliki

akses yang baik sedangkan area lainnya menghadirkan kesulitan signifikan dalam komunikasi dan transfer data. Memahami kondisi geografis Sumbawa yang luas dan tantangan distribusi sinyal yang tidak merata ini merupakan langkah penting dalam merancang aplikasi web dalam hal ini aplikasi penghitungan suara pemilihan umum legislatif untuk kebutuhan internal Partai Keadilan Sejahtera.

React, yang diperkenalkan oleh Facebook pada tahun 2013 merupakan salah satu *pustaka* Javascript yang populer [1]. *React* membawa paradigma baru dalam pembangunan antarmuka pengguna dengan konsep komponen. Komponen merupakan blok-blok *User Interface* (UI) yang digunakan untuk kemudian dikonstruksi menjadi satu halaman website. Komponen ini dapat berupa kumpulan kode yang mewakili elemen UI seperti tombol, form, atau bahkan layout yang lebih kompleks ini memungkinkan pengembang melakukan *reusable* kode.

Dalam pengembangan aplikasi web modern, terdapat dua pendekatan dominan yang seringkali digunakan yaitu *client-side rendering* dan *server-side rendering*. *Client-side rendering* memanfaatkan kemampuan browser untuk menghasilkan antarmuka pengguna secara dinamis melalui Javascript, menciptakan pengalaman yang interaktif dan responsif dengan keunggulan waktu muat awal yang cepat, meskipun dihadapkan pada loading state hingga seluruh Javascript terunduh. Namun, pendekatan ini memiliki kelemahan seperti waktu loading yang lebih lama untuk aplikasi skala besar dan dependensi pada kapabilitas eksekusi Javascript di sisi klien [2].

Di sisi lain, *server-side rendering* menjalankan Javascript di *server*, mengirimkan halaman web yang telah dirender sebagai string HTML ke browser, memberikan keuntungan dalam *Search Engine Optimization* (SEO) dengan konten yang langsung tersedia saat halaman dimuat. Meskipun begitu, teknik ini juga memiliki kelemahan di mana halaman akan tampak *blank* sementara menunggu *server* menyelesaikan proses rendering [3].

Inovasi terbaru dari *React* adalah *React Server Components* (RSC), yang memungkinkan server untuk melakukan rendering komponen. Ini meminimalkan beban pada perangkat pengguna dan mempercepat waktu respons aplikasi. *Server Action* diperkenalkan bersamaan dengan RSC sebagai solusi untuk meningkatkan efisiensi aplikasi web [4]. Dengan mengeliminasi kebutuhan lapisan *Application Programming Interface* (API) tradisional yang sering menjadi *bottleneck*, serta memungkinkan komunikasi langsung antara client dan server dan mengurangi latensi.

Dengan menggabungkan kelebihan *React Server Component* dan *Server Action*, dengan menggunakan framework Next.js, penelitian ini mengimplementasikan sistem penghitungan suara yang andal untuk kebutuhan internal Partai

Keadilan Sejahtera di Kabupaten Sumbawa. Pendekatan ini memastikan ketersediaan dan aksesibilitas web di daerah dengan sinyal internet yang lemah.

II. STUDI PUSTAKA

Sebagai perbandingan dan sumber acuan pada penelitian ini, maka diperlukan acuan dari penelitian yang telah dilakukan sebelumnya. Madsen, Lhoták, & Tip menguraikan semantik dari *React* sebagai framework dalam pembuatan aplikasi web, menawarkan pemahaman yang lebih sistematis dan mendalam [5].

Teknik optimisasi kinerja untuk aplikasi *ReactJS* dijelaskan oleh Javeed [2], menyoroti pentingnya algoritma pencarian yang efisien dalam produksi. Pushkarev & Yakubailik memaparkan implementasi *client* dan *server* dari aplikasi web menggunakan *React* dan *Redux* [6], yang mengungkap potensi pengembangan lebih lanjut. Mlynarski & Nurzynska melakukan perbandingan efisiensi berbagai *pustaka* Javascript, termasuk *React*, memberikan pandangan yang luas tentang kinerja aplikasi web [7].

Singh mengeksplorasi penggunaan *React js* dalam pengembangan web, dengan menekankan fitur *reusability* dan *hot reloading* yang membuat pengembangan menjadi lebih efisien [8].

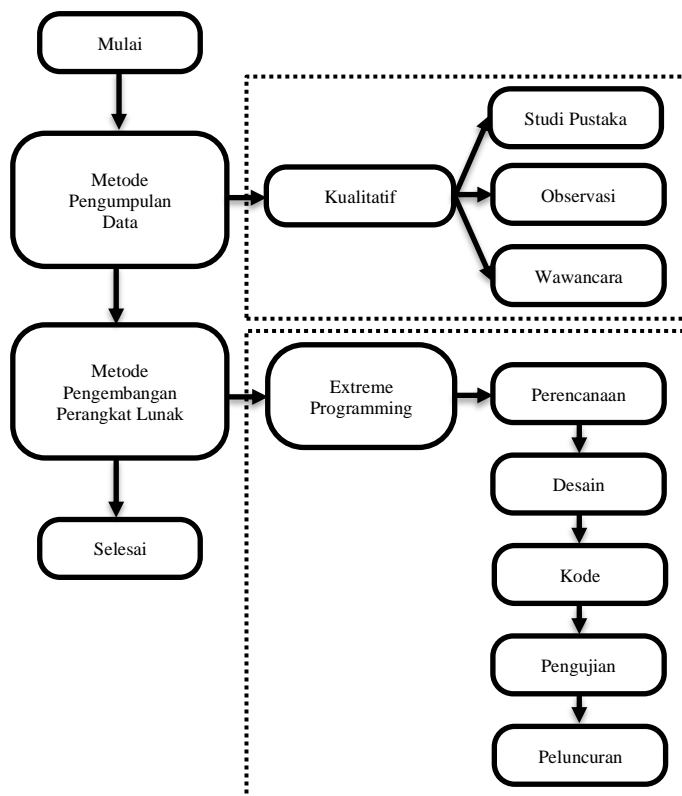
Dengan mempertimbangkan beberapa referensi di atas, maka kekuatan dari penelitian ini yaitu implementasi *React Server Component* dan *Server Action* yang inovatif dan efisien dalam meningkatkan kinerja aplikasi web. Optimisasi ini didasarkan pada pengembangan yang adaptif, yang memanfaatkan teknologi terkini untuk mencapai hasil yang optimal.

III. METODE PENELITIAN

Tahapan penelitian mengikuti *metode extreme programming* sebagai metode pengembangan. Metode ini dipilih untuk mengatur langkah-langkah dalam mengimplementasikan *react server component* dan *server action* pada aplikasi penghitungan suara pemilihan umum legislatif berbasis web dan memastikan prosesnya terorganisir serta sistematis. Penelitian ini mengadopsi pendekatan kualitatif yang dimana memfokuskan pada pemahaman mendalam tentang masalah yang diteliti.

Alur Tahapan Penelitian dapat dilihat pada Gambar 1, yaitu menggunakan model *extreme programming* lalu model ini disesuaikan dengan

tahapan penelitian yang menerapkan metode penelitian kualitatif.



Gambar 1. Alur Tahapan Penelitian

Gambar diatas merupakan alur tahapan penelitian yang dapat dijabarkan sebagai berikut:

A. Metode Pengumpulan Data

Tahapan pertama yaitu metode pengumpulan data menggunakan kualitatif. Ini karena metode ini mengacu pada data dan teori yang telah digunakan sebelumnya, serta hasil analisis menggunakan pendekatan deskriptif.

Studi Pustaka: melibatkan analisis literatur tentang teknologi web modern, manajemen basis data, serta desain antarmuka pengguna, untuk memastikan bahwa aplikasi yang dikembangkan berlandaskan pada pengetahuan yang solid dan terkini.

Observasi: langsung terhadap proses pemilihan legislatif juga menjadi bagian penting dari penelitian ini. Melalui observasi, peneliti dapat memahami dinamika pemilihan legislatif di Sumbawa.

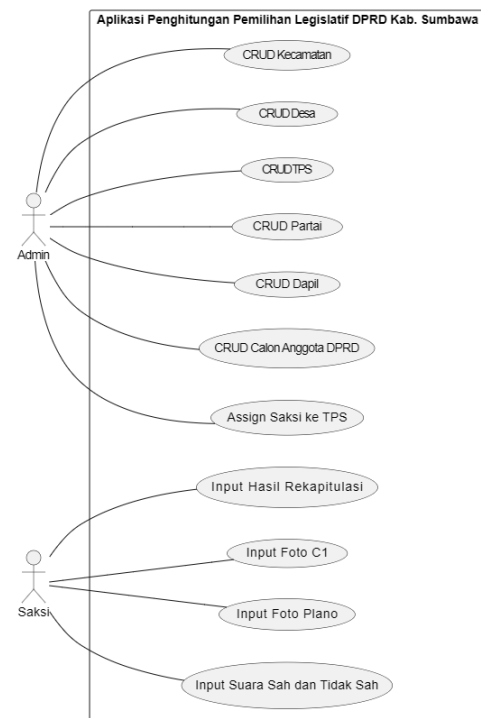
Wawancara: dengan Ketua Dewan Perwakilan Daerah dari Partai Keadilan Sejahtera. Wawancara ini bertujuan untuk mendapatkan *insight* langsung dari stakeholder mengenai kebutuhan, tantangan, dan harapan mereka terhadap sistem penghitungan suara.

B. Metode Pengembangan Perangkat Lunak

Untuk pengembangan perangkat lunak penelitian ini mengadopsi metode *Extreme Programming* untuk membangun aplikasi penghitungan suara pemilihan legislatif di Sumbawa. *Extreme Programming* dipilih karena keunggulannya dalam menangani perubahan sesuai kebutuhan [9].

1. Perencanaan: Fase ini melibatkan pengumpulan kebutuhan data melalui wawancara, studi pustaka dan observasi.
2. Desain: Desain sistem dilakukan dengan menggunakan *Unified Modeling Language* (UML) untuk menggambarkan secara visual arsitektur sistem, komponen, dan interaksi antar komponen [10].

Tahapan analisis desain sistem dimulai dengan mengidentifikasi aktor dan kebutuhan fungsional melalui *Use Case Diagram* yang memetakan interaksi antara pengguna dengan sistem dan berbagai skenario penggunaan aplikasi [11]. *Use Case Diagram* dapat dilihat pada Gambar 2:



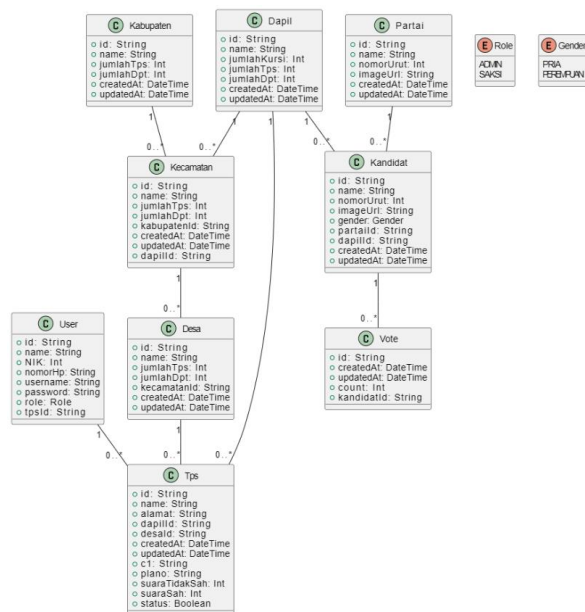
Gambar 2. Use Case Diagram

Aplikasi penghitungan suara pemilihan umum legislatif memiliki dua aktor. Pertama admin, yang memiliki hak akses tingkat tinggi, use case meliputi operasi *Create, Read, Update, Delete* (CRUD) pada entitas penting seperti kecamatan, desa, Tempat Pemungutan Suara (TPS), partai,

dapil, dan calon anggota Dewan Perwakilan Daerah (DPRD). Hal ini memungkinkan admin untuk memelihara dan *update* informasi yang relevan dengan dinamika pemilihan. Selain itu, admin bertugas menugaskan saksi ke TPS tertentu, dimana mereka akan bertanggung jawab dalam mengawasi jalannya pemilihan dan menginput data rekapitulasi suara.

Kedua saksi, sebagai pengguna dengan tanggung jawab khusus di TPS, memiliki use case yang berkaitan dengan penginputan data. Mereka bertanggung jawab dalam memasukkan data rekapitulasi perolehan suara, mengunggah foto C1 dan plano yang merupakan dokumen vital dalam pemilihan, serta menginput jumlah suara sah dan tidak sah.

Perancangan desain sistem juga melibatkan pembuatan Class Diagram yang mendefinisikan struktur kelas dan hubungan antar kelas dalam ap-likasi. Diagram ini memperlihatkan bagaimana data disimpan, diproses, dan dikelola dalam sistem. Dengan Class Diagram, pengembang dapat memahami struktur dan desain sistem yang akan dibangun lebih mendalam. Class Diagram dapat dilihat pada Gambar 3.



Gambar 3. Class Diagram

Setiap kelas merepresentasikan tabel dalam database dengan kolomnya dijadikan sebagai atribut. Role dan Gender adalah enum yang merepresentasikan nilai tetap. Relasi direpresentasikan dengan garis dan notasi "1" -- "0..*"

3. Pengkodean: Fase menggunakan Visual Studio Code sebagai text editor untuk implementasi aplikasi.
4. Pengujian: Pengujian menggunakan metode *Black Box Testing*, di mana berfokus pada output yang dihasilkan aplikasi tanpa harus memperhatikan bagaimana proses internal dari kode tersebut [12]. Hal ini memungkinkan untuk menemukan bug atau masalah dari perspektif pengguna, memastikan aplikasi berjalan sesuai dengan kebutuhan dan ekspektasi pengguna.
5. Peluncuran: Setelah aplikasi diuji dan memenuhi kriteria, maka akan dilakukan peluncuran.

IV. HASIL DAN PEMBAHASAN

A. Implementasi

Proses implementasi merupakan tahapan pengkodean untuk membangun aplikasi. Gambar 4 dan 6 adalah kode untuk melakukan pengambilan data. Daripada menggunakan cara pada Gambar 4, yaitu menunggu *browser* melakukan *initial load* kemudian melakukan *request* ke *server* melalui *Application Programming Interface* (API) seperti pada Gambar 5 lalu *server* berkomunikasi ke *database*, maka pada aplikasi ini menggunakan cara seperti pada Gambar 6, dimana component bisa langsung berkomunikasi ke *database* karna *request* dilakukan pada sisi *server*.

```

const [dapil, setDapil] = useState<any>();
useEffect(() => {
  const fetchData = async () => {
    const response = await fetch("/api/dapil");
    const data = await response.json();
    setDapil(data);
  };
  fetchData();
}, []);
  
```

Gambar 4. Fetch Data ke Route /api/dapil

Gambar 4 merupakan kode fetch data dengan client-side menggunakan *useEffect* dan *fetch* ke API /api/dapil.



Gambar 5. API Melakukan Komunikasi ke Database

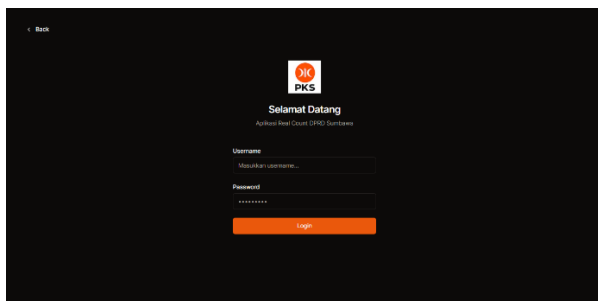
Cara biasanya adalah *Application Programming Interface* (API) yang akan melakukan komunikasi ke *database*. Dengan *react server component* pengembang tidak perlu bekerja dua kali dengan membuat API untuk bagian *server* komunikasi dengan database, karena komponen ini dieksekusi di server maka pengembang bisa komunikasi langsung ke database.



Gambar 6. API melakukan komunikasi ke database

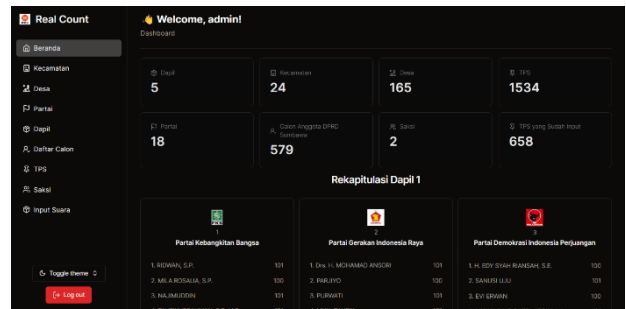
Adapun berikut implementasi aplikasi penghitungan suara pemilihan legislatif Sumbawa untuk kebutuhan internal Partai Keadilan Sejahtera:

1) Tampilan untuk Role Admin



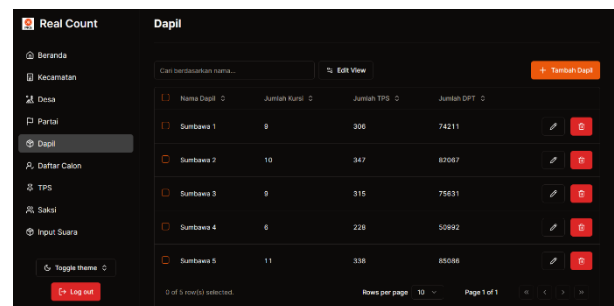
Gambar 7. Tampilan Halaman Login

Sebelum masuk ke halaman home pengguna harus login dengan memasukkan username dan password yang benar.



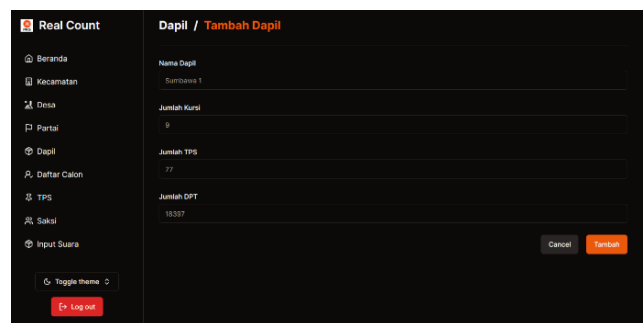
Gambar 8. Tampilan Halaman Home

Setelah login pengguna akan dihadapkan dengan halaman home yang berisi rangkuman data seperti jumlah partai, calon anggota DPRD, saksi, dan hasil perolehan suara para calon anggota DPRD.



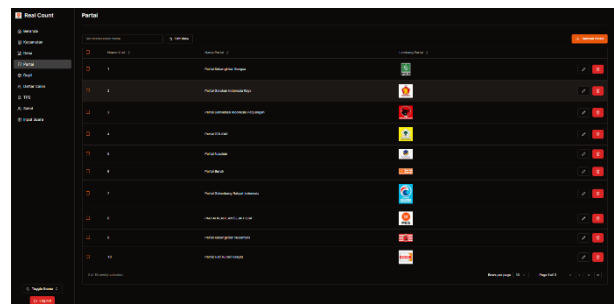
Gambar 9. Tampilan Halaman Dapil

Berisi daftar dapil yang ada dan detail jumlah kebutuhan kursi, TPS dan DPT.



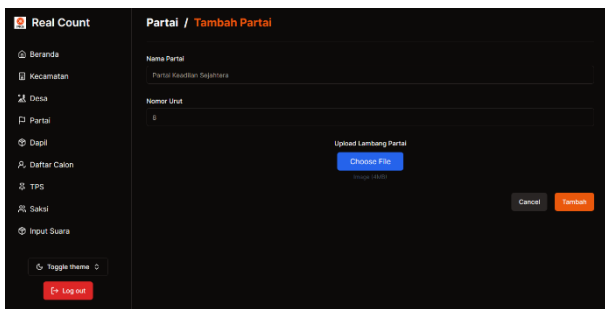
Gambar 10. Tampilan Halaman *Create Dapil*

Admin dapat membuat entitas dapil dengan menginput semua data yang dibutuhkan.



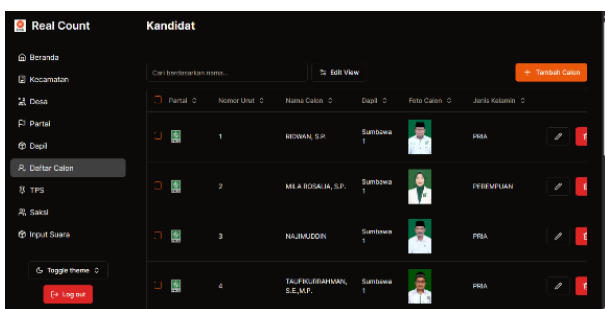
Gambar 11. Tampilan Halaman Partai

Memuat daftar partai yang ada dan detail nama partai, nomor urut dan lambang partai.



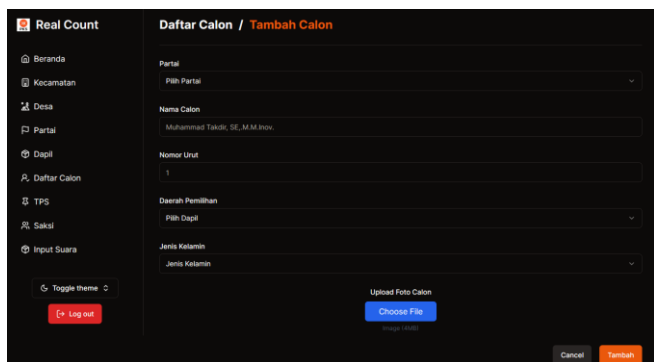
Gambar 12. Tampilan *Create Partai*

Admin dapat membuat entitas partai dengan menginput semua data yang dibutuhkan.



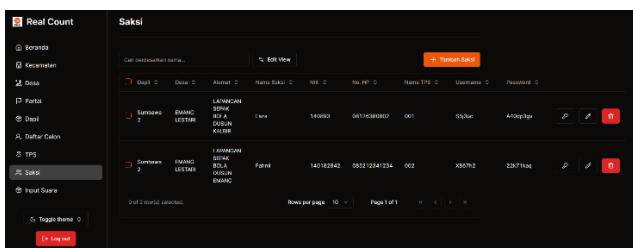
Gambar 13. Tampilan Daftar Calon

Berisi daftar calon DPRD yang ada dan detail nama calon, nomor urut, dapil, gender dan pas foto.



Gambar 14. Tampilan *Create Calon DPRD*

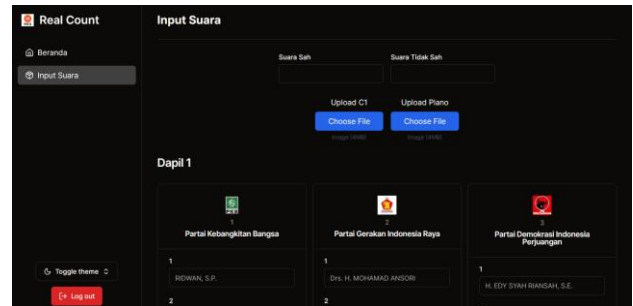
Admin dapat membuat entitas calon anggota DPRD dengan menginput semua data yang dibutuhkan.



Gambar 15. Tampilan *Crate Saksi*

Berisi daftar partai yang ada dan detail nama partai, nomor urut dan lambing partai.

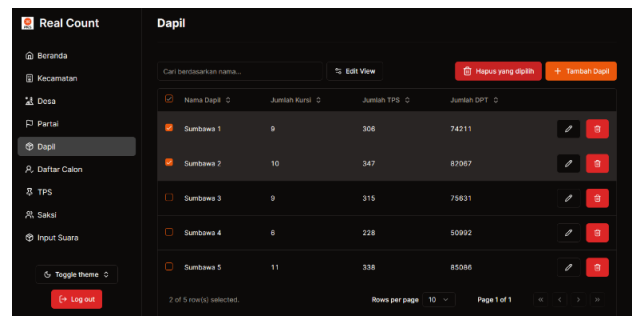
2) Tampilan untuk Role Saksi



Gambar 16. Tampilan Input Suara

Halaman saksi untuk menginput suara berdasarkan TPS yang ditugaskan admin kepada saksi

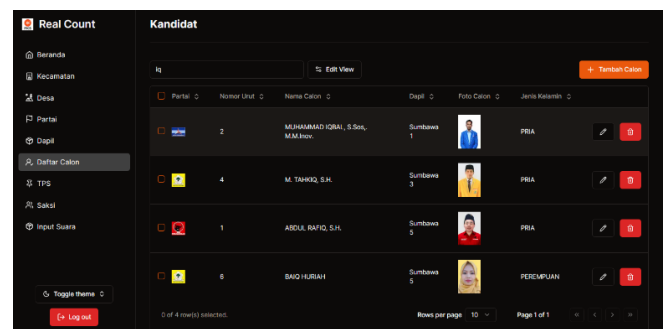
3) Tampilan fitur bulk delete



Gambar 17. Tampilan Fitur *Bulk Delete*

Dengan menceklis pengguna dapat memanfaatkan fitur *bulk delete*.

4) Tampilan fitur pencarian



Gambar 18. Tampilan Fitur Pencarian dengan Kata Kunci "iq"

B. Pengujian

Metode *black-box testing*. Pengujian ini untuk menjamin bahwa sistem berjalan dengan baik secara fungsional tanpa kesalahan. Hal ini mencakup pengecekan kesalahan fungsi sehingga sangat berguna untuk mengidentifikasi kesalahan

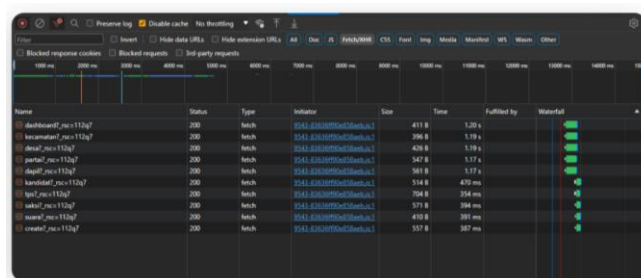
dalam perangkat lunak. Hasil pengujian *black box* sebagaimana disajikan pada Tabel 1.

Tabel 1. Hasil Pengujian Black-box.

Aktivitas dan Fungsional	Hasil	Kesimpulan
Reload halaman di aplikasi web	Network Tab berjalan paralel dan cepat	Berhasil
Simulasi jaringan 3G	Aplikasi web lebih cepat dimuat karena hasil yang dikirim dari server dilakukan paralel bukan waterfall	Berhasil
Menon-aktifkan javascript pada browser	Aplikasi web tetap berhasil melakukan <i>submission</i> pada pengisian form di aplikasi	Berhasil
Memasukkan data login yang valid (Login)	Menampilkan halaman Dashboard	Berhasil
Memasukkan data login yang tidak valid (Login)	Menampilkan pesan error	Berhasil
Memasukkan input yang sesuai (Halaman Create: Dapil, Kecamatan, Desa, TPS, Partai, Calon DPRD atau Saksi)	Data Dapil, Kecamatan, Desa, TPS, Partai, Calon DPRD atau Saksi Bertambah	Berhasil
Memasukkan input hanya sebagian atau yang tidak sesuai (Halaman Create: Dapil, Kecamatan, Desa, TPS, Partai, Calon DPRD atau Saksi)	Menampilkan pesan error	Berhasil
Memasukkan input yang sesuai (Halaman Edit: Dapil, Kecamatan, Desa, TPS, Partai, Calon DPRD atau Saksi)	Data Dapil, Kecamatan, Desa, TPS, Partai, Calon DPRD atau Saksi Berubah	Berhasil
Memasukkan input hanya sebagian atau yang tidak sesuai (Halaman Edit: Dapil, Kecamatan, Desa, TPS, Partai, Calon DPRD atau Saksi)	Menampilkan pesan error	Berhasil
Memasukkan input yang sesuai (Halaman Input Suara untuk Saksi TPS)	Data Suara Bertambah atau Berubah	Berhasil
Memasukkan input hanya sebagian atau yang tidak sesuai (Halaman Input Suara untuk Saksi TPS)	Menampilkan pesan error	Berhasil
Menghapus data yang dipilih (Dapil, Kecamatan, Desa, Tps, Partai, Kandidat, Saksi)	Data terhapus	Berhasil
Input pencarian pada tabel (Dapil, Kecamatan, Desa, Tps, Partai, Kandidat, Saksi)	Melakukan pencarian	Berhasil

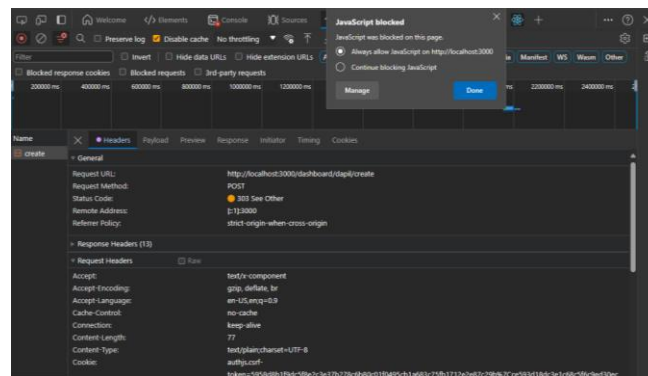
Selanjutnya pengujian dampak dari diimplementasikannya *React Server Component*, aplikasi dapat memuat komponen secara lebih

efisien. Ini terlihat dari struktur *network waterfall* yang berubah menjadi paralel, di mana setiap komponen memuat data yang diperlukan secara bersamaan daripada secara berurutan. Perubahan ini memberikan peningkatan signifikan dalam kecepatan pemuatan halaman, yang sangat kritis dalam aplikasi penghitungan suara yang memerlukan responsivitas tinggi. Hasil ini menunjukkan bahwa penggunaan *React Server Component* dapat menjadi solusi yang efektif untuk meningkatkan performa aplikasi web.



Gambar 19. Network Tab Menunjukkan Berjalan Paralel

Gambar 20 membuktikan bahwa *Server Action* tetap bisa melakukan submit walaupun javascript di browser di non-aktifkan, jadi apabila jaringan sedang lemah pengguna tetap bisa menginput dan melakukan *submission*.



Gambar 20. Melakukan Submission pada Halaman Create Dapil

V. KESIMPULAN

Implementasi RSC dan *Server Action* pada aplikasi web, memberikan perbaikan signifikan dalam efisiensi pemuatan komponen. Dengan struktur *network waterfall* yang berubah menjadi paralel, komponen dapat memuat data secara bersamaan dan meningkatkan responsivitas aplikasi. *Server Action* memungkinkan pengiriman data bahkan ketika Javascript di browser non-aktif, sehingga mempertahankan fungsionalitas meskipun dalam kondisi jaringan yang lemah.

Penelitian ini mengadopsi metode *Extreme Programming* dan *framework* Next.js untuk pengembangan, menjamin fleksibilitas dan struktur kode yang kokoh. Pendekatan UML untuk desain sistem. Dan melalui pengujian black-box aplikasi terbukti beroperasi sesuai spesifikasi, menangani beragam skenario dengan efisien, dan siap diluncurkan.

DAFTAR PUSTAKA

- [1] M. Thakkar, "Introducing React.js," pp. 41–91, 2020, doi: 10.1007/978-1-4842-5869-9_2.
- [2] A. Javeed, "Performance Optimization Techniques for ReactJS," *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pp. 1–5, 2019, doi: 10.1109/ICECCT.2019.8869134.
- [3] T. F. Iskandar, M. Lubis, T. F. Kusumasari, and A. R. Lubis, "Comparison between client-side and server-side rendering in the web development," *IOP Conf Ser Mater Sci Eng*, vol. 801, 2020, doi: 10.1088/1757-899X/801/1/012136.
- [4] "Introducing Zero-Bundle-Size React Server Components – React." Accessed: Jan. 02, 2024. [Online]. Available: <https://react.dev/blog/2020/12/21/data-fetching-with-react-server-components>
- [5] M. Madsen, O. Lhoták, and F. Tip, "A Semantics for the Essence of React," pp. 12:1–12:26, 2020, doi: 10.4230/LIPIcs.ECOOP.2020.12.
- [6] A. A. Pushkarev and O. Yakubailik, "A web application for visualization, analysis, and processing of agricultural monitoring spatial-temporal data," *Spatial Data Processing for Monitoring of Natural and Anthropogenic Processes 2021*, 2021, doi: 10.25743/sdm.2021.20.11.028.
- [7] A. Mlynarski and K. Nurzynska, "Comparative Analysis of JavaScript and Its Extensions for Web Application Optimization," pp. 539–550, 2017, doi: 10.1007/978-3-319-58274-0_42.
- [8] J. Singh, H. Sharma, R. Mishra, S. Hazra, and N. Sukhija, "Movie Review Sentimental Analysis Based on Human Frame of Reference," *Advances in Intelligent Systems and Computing*, 2021, doi: 10.1007/978-981-33-4367-2_27.
- [9] K. Niayu Illahi, "IMPLEMENTASI METODE EXTREME PROGRAMMING PADA SISTEM INFORMASI REPOSITORI SKRIPSI DI PERPUSTAKAAN UNIVERSITAS PRABUMULIH (IMPLEMENTATION OF THE EXTREME PROGRAMMING METHOD IN THESIS REPOSITORY INFORMATION SYSTEMS AT THE PRABUMULIH UNIVERSITY LIBRARY)."
- [10] M. R. Setyautami, D. Adianto, and A. Azurat, "Modeling multi software product lines using UML," *Proceedings of the 22nd International Systems and Software Product Line Conference - Volume 1*, 2018, doi: 10.1145/3233027.3236400.
- [11] K. Qi and B. Boehm, "Detailed Use Case Points (DUCPs): A Size Metric Automatically Countable from Sequence and Class Diagrams," *2018*

- IEEE/ACM 10th International Workshop on Modelling in Software Engineering (MiSE)*, pp. 17–24, 2018, doi: 10.1145/3193954.3193955.
- [12] H. Hendri, J. W. H. Manurung, R. A. Ferian, W. F. Hanaatmoko, and Y. Yulianti, "Pengujian Black Box pada Aplikasi Sistem Informasi Pengelolaan Masjid Menggunakan Teknik Equivalence Partitions," vol. 3, pp. 107–113, 2020, doi: 10.32493/jtsi.v3i2.4694.