

Evaluasi Fungsi Pencarian di React.js Untuk Pengambilan Data Dalam Sistem Wiki-Inf

Muhammad Zuhrul Ikfa¹, Ninik Agustin², Tri Aggoro³

*Informatics Study Program, Faculty of Mathematics and Computer Science, University of Nahdlatul Ulama Al Ghazali
Address Including Country Name*

¹ zuhrulikfa19@gmail.com

² ninik.agustin@unugha.id

³ trianggoro1103@gmail.com

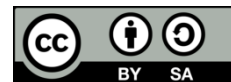
Dikirim pada 17-10-2024, Direvisi pada 28-10-2024, Diterima pada 10-11-2024

Abstract

This research discusses the importance of information centers in a higher education environment, especially at Nahdlatul Ulama Al Ghazali University Cilacap, which currently uses a WhatsApp-based chatbot. ReactJS is the main focus in developing the information center system, considering that ReactJS offers modularity, scalability, and high performance in web application front-end development. The main problem faced is the difficulty in finding information quickly due to the large volume of data. To overcome this problem, this research proposes the use of a ReactJS-based search filter feature to facilitate access and retrieval of relevant information for students. The use of ReactJS in front-end development, Express.js for back-end, and a database using MySQL for structured data management is the right choice for the development of Wiki-Inf because it can make Wiki-Inf an efficient and reliable information center. The research method uses the Agile Scrum Methodology approach, which allows system development to be carried out iteratively and responsive to change. The developed Wiki-Inf system shows high performance in article search with an average precision of 97.18% and recall of 100%, ensuring that relevant information is found effectively. This research proves that optimizing search features with ReactJS significantly improves search accuracy and speed, providing an optimal search experience for students. The Wiki-Inf system proved to be an accurate, relevant, and easily accessible information solution, making it an efficient tool in supporting information retrieval activities in the university environment.

Keywords: Search, Information Retrieval, Recall, Precision

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Muhammad Zuhrul Ikfa

Fakultas Matematika dan Ilmu Komputer, Universitas Nahdlatul Ulama Al Ghazali Cilacap, Jalan Kemerdekaan Barat No. 17, Gligir, Kesugihan Kidul, Kec. Kesugihan Kabupaten Cilacap, Jawa Tengah, 53274, Indonesia. Email: adri. zuhrulikfa19@gmail.com

I. INTRODUCTION

A. Latar Belakang

Pusat informasi merupakan tempat yang menyediakan informasi yang dibutuhkan oleh pengguna dalam berbagai bentuk, seperti chatbot dan website [1]. Pada perguruan tinggi, khususnya di Universitas Nahdlatul Ulama Al Ghazali Cilacap, mahasiswa masih mengandalkan pusat informasi berbasis chatbot melalui WhatsApp yang memiliki keterbatasan informasi [2]. Masalah utama yang dihadapi adalah sulitnya mencari informasi dengan cepat karena volume data yang besar. Untuk mengatasi hal tersebut, diperlukan sistem pusat informasi berbasis web yang dapat mempermudah akses dan pengambilan informasi yang relevan bagi mahasiswa. Sistem pusat informasi dirancang untuk mengambil data dari database yang telah diinputkan oleh administrator. Data yang diambil adalah kata kunci yang telah dimasukkan oleh pengguna

melalui fitur pencarian. Proses pengambilan data pada platform ini didasarkan pada prinsip-prinsip pencarian informasi.

Temu kembali informasi adalah bidang yang membahas struktur, analisis, organisasi, penyimpanan, pencarian, dan pengambilan informasi [3]. Model yang paling awal, Boolean Retrieval, menggunakan operator seperti AND, OR, dan NOT untuk mengambil dokumen yang sama persis dengan kueri, tetapi tidak memiliki fleksibilitas dan peringkat relevansi. Vector Space Model (VSM) memperbaiki hal ini dengan merepresentasikan dokumen dan kueri sebagai vektor dalam ruang multi-dimensi, yang memungkinkan pencocokan parsial dan pemeringkatan relevansi melalui kemiripan kosinus. Namun, VSM mengasumsikan independensi istilah, yang dapat membatasi. Pencarian informasi secara probabilistik, khususnya Binary Independence Model (BIM), mengurutkan dokumen berdasarkan probabilitas relevansinya, namun membutuhkan estimasi probabilitas yang akurat. Pengindeksan Semantik Laten (Latent Semantic Indexing/LSI) mengurangi dimensi data untuk mengungkapkan hubungan laten antara istilah dan dokumen, secara efektif menangani sinonimitas, namun secara komputasi sangat intensif. Pencarian informasi neural menggunakan teknik pembelajaran mendalam untuk memodelkan pola data yang kompleks, yang unggul dalam menangani sinonimitas dan polisemi, tetapi intensif secara komputasi. Mesin pencari modern menggabungkan teknik-teknik ini dengan perayapan web, pengindeksan, dan pembelajaran mesin untuk memberikan hasil yang cepat dan relevan, tetapi sering kali tetap tidak jelas karena algoritme yang dipatenkan. Selain itu, Pencarian Informasi Fuzzy menangani ketidakpastian dalam kueri dengan memungkinkan tingkat relevansi, sementara Pencarian Berbasis Kluster mengatur dokumen ke dalam kluster berbasis kemiripan untuk mengoptimalkan waktu pencarian dan relevansi. Penyaringan kolaboratif memprediksi relevansi berdasarkan preferensi pengguna, memberikan hasil yang dipersonalisasi tetapi kesulitan dengan pengguna baru, sementara pencarian berbasis konten berfokus pada konten dokumen, memberikan hasil yang efektif untuk pencarian spesifik tetapi berpotensi kehilangan konteks yang lebih luas. Teknik IR yang berbeda ini terus berkembang, sering kali dalam kombinasi, untuk mengatasi kompleksitas pencarian informasi modern dan memastikan bahwa pengguna menerima informasi yang paling relevan dari kumpulan data yang semakin besar [4]. Pada penelitian ini, information retrieval akan diterapkan dengan menggunakan filter pencarian pada ReactJS. ReactJS dipilih karena merupakan bahasa pemrograman yang populer, didukung oleh komunitas yang besar dan aktif, serta menyediakan banyak sumber daya, tutorial, dan library yang membantu dalam pengembangan ReactJS [5]. Filter pencarian merupakan fitur React.js yang memungkinkan penyaringan informasi berdasarkan parameter tertentu. Fitur ini memfasilitasi pengambilan data yang efisien dan meningkatkan aksesibilitas informasi. Fitur ini sering digunakan dalam e-commerce, sistem manajemen sumber daya manusia, platform video/blogging, dan banyak situs web lainnya.

Berdasarkan latar belakang tersebut, maka diperlukan sebuah pengembangan pusat informasi berbasis web yang dinamakan Wiki-inf. Sistem berbasis web ini dikembangkan dengan menggunakan ReactJS dan mengoptimalkan filter pencarian untuk memudahkan pengguna dalam mencari informasi. Fitur pencarian pada ReactJS diimplementasikan pada Wiki-Inf, dengan domain deployment wiki-if.unugha.ac.id. Tujuan dari penelitian ini adalah untuk mengevaluasi fungsi pencarian pada React.js yang diimplementasikan pada Wiki-Inf.

Parameter yang diukur adalah nilai recall dan precision dari sistem temu kembali informasi [6]. Dengan adanya optimasi filter pencarian pada ReactJS, diharapkan mahasiswa dapat mengakses informasi dengan lebih mudah dan efisien.

B. Permasalahan

Berdasarkan penjabaran latar belakang diatas, maka dapat dirumuskan permasalahan yaitu:

1. Bagaimana mengembangkan sistem berbasis web menggunakan ReactJS pada Wiki-Inf?
2. Bagaimana optimasi search filter pada Wiki-Inf menggunakan ReactJS? ?

C. Tujuan

Berdasarkan topik diatas maka tujuan dari penelitian ini adalah untuk :

1. Mengetahui cara mengembangkan sistem berbasis web pada Wiki-Inf sebagai pusat informasi.
2. Mengetahui hasil implementasi search filter ReactJS pada Wiki-Inf melalui pengukuran kecepatan kinerja pencarian.Reality.

II. RESULTS AND DISCUSSION

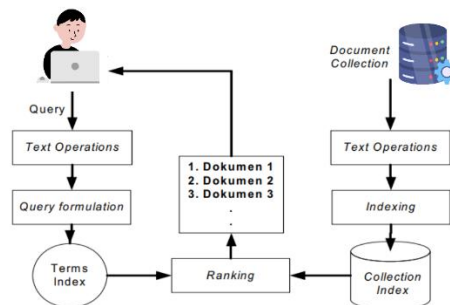
A. Tahap Penelitian

Penelitian ini bertujuan untuk mengatasi permasalahan dalam mengakses informasi pada pusat informasi Wiki-Inf dengan mengoptimalkan fitur pencarian menggunakan ReactJS. Identifikasi masalah menunjukkan bahwa tanpa adanya filter pencarian, pengguna harus mencari data secara manual di dalam sistem informasi yang dapat menyulitkan dan memakan waktu [7][8]. Oleh karena itu, implementasi filter pencarian yang dioptimalkan diharapkan dapat memudahkan pengguna untuk menemukan informasi yang relevan dengan cepat dan efisien [9].

Langkah pertama dalam penelitian ini adalah melakukan studi literatur untuk mengumpulkan ide mengenai optimasi filter pencarian, teknologi ReactJS, dan teori-teori yang relevan. Selanjutnya, dilakukan analisis kebutuhan untuk menentukan kebutuhan fungsional dan non-fungsional dari sistem yang akan dikembangkan. Tahap perancangan meliputi perancangan arsitektur sistem, antarmuka pengguna, struktur basis data, dan alur kerja sistem, yang kesemuanya berfungsi sebagai panduan pengembangan sistem berbasis web Wiki-Inf.

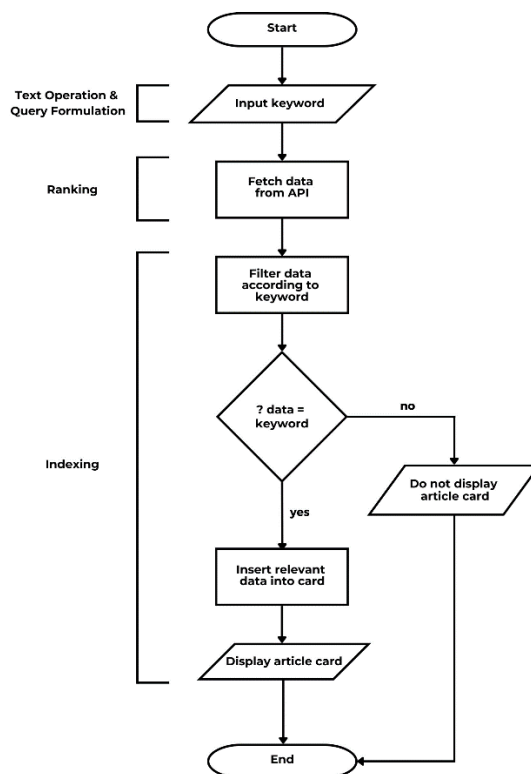
Pengembangan sistem menggunakan metodologi Agile dengan pendekatan Scrum, yang memungkinkan pengembangan dan peningkatan fitur secara berulang dan berkelanjutan [10]. Sistem dibangun menggunakan ReactJS untuk front-end, Express.js untuk back-end, dan MySQL untuk manajemen data terstruktur [11]. Pengujian sistem dilakukan secara menyeluruh dengan menggunakan metode blackbox testing [12]. Hasil pengujian diharapkan dapat mengidentifikasi masalah atau bug yang mungkin terjadi sehingga dapat dilakukan perbaikan.

Data yang dikumpulkan selama tahap uji coba dan penggunaan nyata digunakan untuk menganalisis kinerja sistem. Perhitungan recall dan precision bertujuan untuk mengetahui efektivitas sistem temu kembali informasi melalui sistem Wiki-Inf.



Gambar 1 Bagian utama dari Pencarian Informasi

Temu Kembali Informasi terdiri dari empat bagian utama: Operasi Teks, yang memilih kata-kata penting dari dokumen atau kueri; Formulasi Kueri, yang memberikan bobot pada kata-kata untuk menunjukkan tingkat kepentingannya; Pemeringkatan, yang mengurutkan dokumen berdasarkan relevansi dengan kueri; dan Pengindeksan, tahap awal di mana indeks kata dari dokumen dikompilasi sebelum pencarian dilakukan. Keempat tahap ini bekerja sama untuk menghasilkan hasil pencarian yang relevan dan terurut. Recall dan precision adalah dua metrik utama yang digunakan untuk mengevaluasi kinerja sistem pencarian informasi. Information Retrieval yang terdiri dari beberapa bagian ini akan diterapkan pada pencarian Wiki-Inf. Penerapan bagian-bagian IR pada pencarian Wiki-Inf ditunjukkan pada gambar 2.



Gambar 2 Penerapan bagian IR pada pencarian Wiki-Inf

Fitur pencarian dimulai dengan input kata kunci oleh pengguna melalui kolom pencarian pada halaman utama Wiki-Inf, kemudian sistem akan mengambil data artikel dari API yang dibangun dengan menggunakan database MySQL, lalu data yang diambil akan disaring dengan mencocokkan kata kunci dan isi dari data artikel tersebut sehingga data yang akan ditampilkan hanya data yang relevan dengan kata kunci dari pengguna. Sebelum data artikel ditampilkan, data tersebut akan ditampilkan dengan menggunakan kartu untuk memudahkan user melihat sekilas informasi mengenai artikel dan dapat menuju ke halaman artikel dengan memilih kartu tersebut. Proses Text Operation dan Query Formulation terjadi ketika pengguna memasukkan kata kunci ke dalam sistem Wiki-Inf. Kemudian data yang ada di dalam database akan melalui proses indexing untuk menyediakan database indeks koleksi dokumen dan proses ranking berjalan untuk menampilkan artikel yang relevan dengan kata kunci yang dimasukkan pengguna. Recall mengukur sejauh mana sistem mampu menemukan semua dokumen yang relevan dengan kata kunci yang dimasukkan, sedangkan precision mengukur seberapa relevan dokumen yang ditemukan oleh sistem dengan kata kunci yang dimasukkan.

Setelah pengembangan dan pengujian, sistem diluncurkan ke lingkungan produksi. Umpan balik dari pengguna akhir digunakan untuk melakukan penyesuaian dan perbaikan pada sistem. Analisis data dilakukan untuk mengevaluasi kinerja sistem, termasuk pengukuran recall dan presisi untuk menilai efektivitas fitur pencarian. Kesimpulan dibuat berdasarkan hasil analisis ini, termasuk waktu yang dibutuhkan dan efektivitas fitur pencarian pada Wiki-Inf, sehingga dapat memberikan gambaran yang jelas mengenai keberhasilan implementasi filter pencarian ReactJS dalam membantu pengguna menemukan informasi yang mereka butuhkan.

B. Hasil Penelitian

Pada penelitian ini, pengembangan sistem Wiki-Inf berhasil dilakukan melalui beberapa tahapan yang terstruktur dan sistematis. Hasil dari pengembangan ini adalah terciptanya sebuah platform pencarian artikel berbasis web yang dinamakan Wiki-Inf (Wikipedia Informatika). Tampilan aplikasi disajikan pada gambar berikut dan dapat diakses melalui domain wiki-if.ungha.ac.id.



Gambar 3 Mencari artikel di bilah pencarian Wiki-Inf

Cara kerja sistem ini dimulai ketika pengguna mengetikkan kata kunci pada kolom pencarian yang tersedia di halaman utama. Setelah kata kunci dimasukkan, filter pencarian akan bekerja untuk mencari dan menyaring artikel yang sesuai dari database. Hasil pencarian kemudian ditampilkan dalam bentuk kartu-kartu artikel pada halaman utama di bawah kolom pencarian. Setiap kartu berisi judul artikel dan ringkasan singkat yang relevan dengan kata kunci yang dimasukkan, sehingga pengguna dapat dengan mudah memilih dan membaca artikel yang mereka butuhkan.

Cara kerja sistem ini dimulai ketika pengguna mengetikkan kata kunci pada kolom pencarian yang tersedia di halaman utama. Setelah kata kunci dimasukkan, filter pencarian akan bekerja untuk mencari dan menyaring artikel yang sesuai dari database. Hasil pencarian kemudian ditampilkan dalam bentuk kartu-kartu artikel pada halaman utama di bawah kolom pencarian. Setiap kartu berisi judul artikel dan ringkasan singkat yang relevan dengan kata kunci yang dimasukkan, sehingga pengguna dapat dengan mudah memilih dan membaca artikel yang mereka butuhkan.

Administrator telah memasukkan 72 artikel berbahasa Indonesia ke dalam wiki. Artikel-artikel ini berisi informasi yang relevan dengan mahasiswa, termasuk rincian tentang beasiswa, tesis akademik, program studi, dan nama-nama institusi akademik. Platform ini menyediakan fitur untuk mencari, membaca, dan mengelola artikel yang relevan dengan ruang lingkup pendidikan tinggi. Salah satu hasil yang signifikan adalah implementasi antarmuka pengguna menggunakan ReactJS, yang memungkinkan tampilan yang dinamis dan responsif. Dalam pengujian fungsional, semua fitur bekerja dengan baik, menampilkan artikel yang sesuai dengan kata kunci yang dimasukkan oleh pengguna, dan memungkinkan admin untuk mengelola artikel melalui dashboard yang efisien.

Wiki-Inf menggunakan filter pencarian pada library ReactJS dengan pertimbangan implementasi yang sederhana dan performa yang optimal. Untuk mengukur keberhasilan filter pencarian yang dioptimalkan, diperlukan parameter evaluasi yang jelas. Dalam konteks ini, recall dan precision merupakan dua metrik utama yang digunakan untuk menilai efektivitas sistem pencarian [6][7]. Dengan fokus pada dua parameter tersebut, batasan masalah dalam penelitian ini adalah memastikan bahwa filter pencarian di Wiki-Inf tidak hanya cepat dan responsif, tetapi juga akurat dalam menyajikan informasi yang benar-benar dibutuhkan oleh pengguna.

Recall adalah ukuran proporsi jumlah dokumen yang berhasil ditemukan kembali oleh sistem temu kembali informasi dalam satu kali pencarian. Konsep ini berkaitan erat dengan kemampuan sistem untuk mengambil informasi yang sebelumnya telah disimpan[9].

Tabel 1 *Keyword* Sebagai Tes Pencarian Informasi

No	Keyword	Rendering Time
1	Mahasiswa	3.2 ms
2	Cara	3.8 ms
3	Daftar	4.1 ms
4	Dosen	3.8 ms
5	Langkah	3 ms
6	Login	4.2 ms
7	Skripsi	3.4 ms
8	Bagaimana	3.8 ms
9	Akun	4.3 ms
10	Kelas	3.1 ms
11	Dan	3.1 ms
12	Yang	3.5 ms
13	Untuk	3.5 ms

No	Keyword	Rendering Time
14	Di	4.7 ms
15	Unugha	4 ms
16	Kuliah	3.8 ms
17	Beasiswa	3.9 ms
18	Panduan	3.5 ms
19	Wisuda	2.3 ms
20	Tahapan	2.8 ms

Pengujian recall dan precision pada sistem Wiki-Inf dilakukan dengan menggunakan 20 kata kunci yang relevan dengan konteks akademik di Universitas Nahdlatul Ulama Al Ghazali (UNUGHA) Cilacap. Kata kunci seperti 'Mahasiswa', 'Skripsi', 'Login', 'Dosen', dan 'Beasiswa' dipilih untuk mewakili topik-topik yang sering dicari oleh pengguna di lingkungan universitas. Pengujian juga mencakup pengukuran waktu rendering, yaitu waktu yang dibutuhkan sistem untuk menampilkan hasil pencarian setelah kata kunci dimasukkan. Hasil pengujian menunjukkan bahwa sistem Wiki-Inf memiliki waktu rendering yang stabil dan konsisten, dengan rata-rata berkisar antara 2,3 hingga 4,7 milidetik. Sebagai contoh, kata kunci 'Wisuda' memiliki waktu rendering tercepat yaitu 2,3 ms, sedangkan 'Di' membutuhkan waktu 4,7 ms, yang merupakan waktu rendering tertinggi dalam pengujian ini. Perbedaan waktu ini bisa jadi disebabkan oleh jumlah dan kompleksitas artikel yang terkait dengan kata kunci tersebut. Secara keseluruhan, sistem Wiki-Inf bekerja dengan baik dalam menampilkan hasil pencarian dengan cepat dan efisien, mempertahankan waktu render yang stabil meskipun diuji dengan berbagai kata kunci yang berbeda.

Matriks Recall dan Precision ditunjukkan pada Tabel 2.

Tabel 2 Matriks Recall Dan Precision

	Relevan	Tidak Relevan	Total
Ditemukan	a (<i>hits</i>)	b (<i>noise</i>)	a + b
Tidak Ditemukan	c (<i>missed</i>)	d (<i>rejected</i>)	c + d
Total	a + c	b + d	a + b + c + d

Berdasarkan tabel II, rumus recall dan precision menjadi :

$$\text{Recall} = [a / (a+c)] \times 100 \quad (1)$$

$$\text{Precision} = [a / (a+b)] \times 100 \quad (2)$$

Hasil pengukuran Wiki-Inf berdasarkan matriks Recall dan Precision ditunjukkan pada Tabel III.

Tabel 3 Hasil Pengujian Recall Dan Precision

No	(a)	(b)	(a+b)	(c)	(a+c)	Recall $[a / (a+c)] \times 100$	Precision $[a / (a+b)] \times 100$
1	35	0	35	0	35	100	100
2	16	10	26	0	16	100	61,54
3	25	0	25	0	25	100	100
4	5	0	5	0	5	100	100
5	8	0	8	0	8	100	100
6	7	0	7	0	7	100	100
7	10	0	10	0	10	100	100
8	17	1	18	0	17	100	94.44
9	4	0	4	0	4	100	100
10	3	0	3	0	3	100	100
11	47	4	51	0	47	100	92.16
12	48	0	48	0	48	100	100
13	34	0	34	0	34	100	100
14	64	3	67	0	64	100	95.52
15	25	0	25	0	25	100	100
16	25	0	25	0	25	100	100
17	5	0	5	0	5	100	100
18	4	0	4	0	4	100	100

No	(a)	(b)	(a+b)	(c)	(a+c)	Recall $[a / (a+c)] \times 100$	Precision $[a / (a+b)] \times 100$
19	1	0	1	0	1	100	100
20	4	0	4	0	4	100	100
Rata-rata						100	97.18

Pengujian kinerja menunjukkan bahwa sistem Wiki-Inf memiliki waktu pencarian yang cepat dan konsisten. Dari pengujian dengan 20 kata kunci yang berbeda, waktu pencarian rata-rata sekitar 3 milidetik, yang menunjukkan efisiensi implementasi filter pencarian. Selain itu, hasil pengujian recall dan precision menunjukkan nilai rata-rata recall sebesar 100% dan precision sebesar 97,18%. Hal ini berarti bahwa sistem ini sangat efektif dalam menemukan semua artikel yang relevan dan sebagian besar hasil pencarian sangat relevan dengan query yang diberikan. Meskipun ada sekitar 2,82% artikel yang tidak relevan, secara keseluruhan, kinerja filter pencarian pada Wiki-Inf sangat memuaskan.

C. Pembahasan

Pengembangan Wiki-Inf berdasarkan ReactJS menunjukkan hasil yang memuaskan dalam hal antarmuka pengguna dan kinerja sistem. Penggunaan ReactJS memungkinkan antarmuka yang responsif dan dinamis, serta integrasi filter pencarian yang efisien. Fitur-fitur utama seperti bilah pencarian, halaman artikel, dan dasbor admin berfungsi dengan baik dan memberikan pengalaman pengguna yang optimal. Integrasi dengan back-end menggunakan Node.js dan Express memastikan komunikasi yang lancar antara antarmuka pengguna dan basis data, sehingga semua operasi CRUD berjalan dengan efisien.

Hasil uji recall dan precision menunjukkan bahwa filter pencarian pada Wiki-Inf sangat efektif [6]. Nilai recall yang sempurna (100%) mengindikasikan bahwa sistem mampu menemukan semua artikel yang relevan untuk setiap kata kunci yang dimasukkan. Nilai precision yang tinggi (97.18%) menunjukkan bahwa mayoritas hasil pencarian relevan dengan query yang diberikan, meskipun masih terdapat beberapa artikel yang tidak relevan [9]. Performa ini menunjukkan bahwa optimasi filter pencarian yang diterapkan efektif dalam memberikan hasil pencarian yang akurat dan relevan. Namun, untuk mencapai kesempurnaan, perlu diperhitungkan ketidakakuratan filter atau algoritma pencarian yang masih memungkinkan munculnya artikel yang tidak relevan. Evaluasi dan perbaikan yang berkesinambungan terhadap algoritma pencarian dan kualitas data di Wiki-Inf dapat semakin meningkatkan kinerja sistem ini.

III. KESIMPULAN

Penelitian ini berhasil mengembangkan sistem berbasis web pusat informasi Wiki-Inf menggunakan ReactJS, yang dilengkapi dengan fitur filter pencarian yang efektif. Proses pengembangan sistem mengikuti tahapan-tahapan dalam Software Development Life Cycle (SDLC) dengan metodologi Scrum, yang meliputi analisis kebutuhan, perancangan sistem, pengembangan front-end dan back-end, implementasi basis data, serta pengujian dan optimasi sistem.

Front-end dikembangkan menggunakan ReactJS, yang memberikan modularitas, skalabilitas, dan kinerja tinggi pada aplikasi. Bagian belakang dikembangkan dengan Express.js, memastikan efisiensi dalam pengelolaan server dan API. Basis data menggunakan MySQL, yang menyediakan manajemen data yang andal dan terstruktur. Integrasi teknologi-teknologi ini berkontribusi pada sistem Wiki-Inf yang efektif yang mendukung pencarian informasi dan memenuhi kebutuhan pengguna secara optimal.

Sistem Wiki-Inf yang dioptimalkan dengan fitur filter pencarian di ReactJS menunjukkan performa yang tinggi dalam pencarian artikel. Dengan presisi rata-rata 97,18%, sistem secara konsisten menyajikan informasi yang relevan sesuai dengan kata kunci pencarian pengguna. Nilai recall yang mencapai rata-rata 100% memastikan bahwa sebagian besar informasi yang relevan ditemukan, yang menunjukkan keefektifan dalam pencarian informasi. Optimalisasi filter di ReactJS meningkatkan akurasi dan kecepatan pencarian, memberikan pengalaman pencarian yang optimal. Penelitian ini membuktikan bahwa sistem Wiki-Inf berhasil memberikan solusi informasi yang akurat, relevan, dan mudah diakses, menjadikannya sistem yang efisien dan dioptimalkan dalam lingkungan informasi pendidikan tinggi dan akademik.

DAFTAR PUSTAKA

- [1] L. W. Longgupa, N. Nurfatimah, K. Kasmawati, F. Nadia, and K. Ramadhan, "Inisiasi Pembentukan Pusat Informasi Konseling Kesehatan Reproduksi Remaja," *JMM (Jurnal Masy. Mandiri)*, vol. 5, no. 6, pp. 3612–3621, 2021.
- [2] E. Yuniar and H. Purnomo, "Implementasi Chatbot ALITTA Asisten Virtual dari BALITTAS sebagai Pusat Informasi Di BALITTAS," 2019.
- [3] M. Wali and L. Ahmad, "Perancangan Access Open Journal System (AOJS) dengan menggunakan Framework Codeigniter dan ReactJs," *J. JTik (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 2, no. 1, p. 48, 2018.
- [4] N. Ahmad, A. A. Prasetyo, and A. Masruri, "Penerapan Information Retrieval Pada Search Engine," *J. Inov. Has. Penelit. dan Pengemb.*, vol. 1, no. no.1, pp. 15–23, 2021.
- [5] P. Studi, S. Informasi, H. Jihadi, and A. Fikhi Syarabil, "Perbandingan React Js Dan Vue Js Dalam Pengembangan Aplikasi Web Interaktif: Sebuah Studi Komparatif Penulis 1)," *IBI Kosgoro*, vol. 4, no. 2, pp. 70–79, 1957.
- [6] M. Martin and L. Nilawati, "Recall dan Precision Pada Sistem Temu Kembali Informasi Online Public Access Catalogue (OPAC) di Perpustakaan," *Paradig. - J. Komput. dan Inform.*, vol. 21, no. 1, pp. 77–84, 2019.
- [7] B. A. Elsadantia, "Perkembangan Dan Peran OPAC Pada Aplikasi CIP (Cerah Informasi Pustaka) Untuk Temu Kembali Informasi Di Perpustakaan Universitas Tridinantia Palembang," *J. Multidisipliner Bharasumba*, vol. 2, no. 04 Oktober, pp. 296–315, 2023.
- [8] T. T. Prabowo, "Efektivitas Sistem Temu Kembali Informasi Perpustakaan Digital Institut Seni Indonesia (ISI) Yogyakarta dalam Tinjauan Recall dan Precision," *Media Pustak.*, vol. 28, no. 1, pp. 37–48, 2021.
- [9] V. Amrizal, "Penerapan Metode Term Frequency Inverse Document Frequency (Tf-Idf) Dan Cosine Similarity Pada Sistem Temu Kembali Informasi Untuk Mengetahui Syarah Hadits Berbasis Web (Studi Kasus: Hadits Shahih Bukhari-Muslim)," *J. Tek. Inform.*, vol. 11, no. 2, pp. 149–164, 2018.
- [10] A. Anoesyirwan, H. Madiistriyatno, and S. Mutmainnah, "Peningkatan Kualitas Manajemen Publikasi Ilmiah Menggunakan Metode Agile," *ADI Bisnis Digit. Interdisiplin J.*, vol. 1, no. 2 Desember, pp. 31–39, 2020.
- [11] L. D. Andrianto and D. F. Suyatno, "Analisis Performa Load Testing Antara Mysql Dan Nosql Mongoddb Pada RestAPI Nodejs Menggunakan Postman," *J. Emerg. Inf. Syst. Bus. Intell.*, vol. 5, no. 1, pp. 18–26, 2024.
- [12] P. Ekene Eze, "Debug React apps with React Developer Tools," 2023