

# Pengoptimalan Sistem Pencarian *Repository* Berbasis *Website* dengan *Cosine Similarity*

(Studi Kasus : Universitas Nusantara PGRI Kediri)

**<sup>1\*</sup>Deni Kristanto, <sup>2</sup>Risky Aswi Ramadhani, <sup>3</sup>Ahmad Bagus Setiawan**

<sup>1,2,3</sup> Teknik Informatika, Universitas Nusantara PGRI Kediri

E-mail: [\\*1denikristanto90@gmail.com](mailto:*1denikristanto90@gmail.com), [2riskyaswiramadhani@gmail.com](mailto:2riskyaswiramadhani@gmail.com),

[3ahmadbagus@unpkediri.ac.id](mailto:3ahmadbagus@unpkediri.ac.id)

**Penulis Korespondens : Risky aswi ramadhani**

**Abstrak** — Penelitian ini bertujuan mengoptimalkan sistem pencarian dokumen pada repository Universitas Nusantara PGRI Kediri melalui penerapan algoritma *Cosine Similarity* dan pembobotan *Term Frequency–Inverse Document Frequency* (TF-IDF). Sistem dirancang berbasis web menggunakan *framework* React dan mencakup tahapan preprocessing teks, yaitu tokenisasi, *case folding*, *stopword removal*, dan *stemming*. Setiap *term* diberi bobot menggunakan TF-IDF, kemudian dihitung tingkat kemiripan antara kueri dan dokumen menggunakan *Cosine Similarity* untuk menghasilkan daftar dokumen yang relevan beserta tautan langsung. Hasil pengujian menunjukkan tingkat keberhasilan 100 persen dengan waktu respons 2 sampai 8 detik. Evaluasi *usability* yang melibatkan 10 responden mahasiswa menghasilkan skor rata-rata di atas 4 dari skala 5, menunjukkan bahwa sistem mudah digunakan dan berjalan secara optimal. Kendati demikian, keterbatasan dalam pemahaman konteks semantik masih ditemukan. Oleh karena itu, pengembangan lebih lanjut disarankan dengan integrasi pendekatan BERT guna meningkatkan akurasi dan relevansi pencarian.

**Kata Kunci** — *Cosine Similarity*, Sistem Pencarian Dokumen, React.js

**Abstract** — This study aims to optimize the document search system of the Universitas Nusantara PGRI Kediri repository by implementing the Cosine Similarity algorithm and Term Frequency–Inverse Document Frequency (TF-IDF) weighting. The web-based system is developed using the React framework and involves text preprocessing stages, including tokenization, case folding, stopword removal, and stemming. Each term is weighted using TF-IDF, and the similarity between query and documents is computed using Cosine Similarity to generate a list of relevant documents with direct links. Functional testing demonstrated a 100 percent success rate with a response time ranging from 2 to 8 seconds. Usability evaluation involving 10 student respondents yielded an average score above 4 on a 5-point scale, indicating the system is user-friendly and operates effectively. However, limitations in semantic understanding remain. Thus, future work is recommended to integrate BERT to enhance search accuracy and relevance.

**Keywords**— *Cosine similarity*, *Document Retrieval System*, *React.js*

This is an open access article under the CC BY-SA License.



## I. PENDAHULUAN

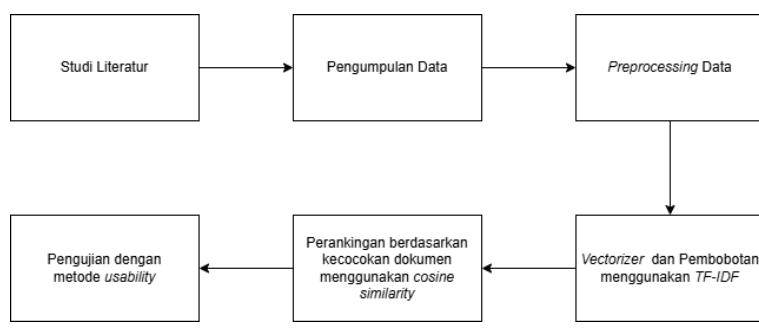
Perkembangan teknologi informasi mendorong institusi pendidikan tinggi, termasuk Universitas Nusantara PGRI Kediri, untuk menyediakan repository berbasis website sebagai sarana penyimpanan dan akses karya ilmiah. Meskipun keberadaan repository meningkatkan

keterbukaan dan ketersediaan informasi, masih ditemukan kendala dalam proses pencarian dokumen yang relevan secara cepat dan tepat. Hal ini disebabkan oleh sistem pencarian yang masih bersifat konvensional dan belum mampu menyesuaikan dengan kebutuhan pengguna. Masalah tersebut menjadi penting untuk ditangani agar potensi repository sebagai sumber daya ilmiah dapat dimanfaatkan secara optimal. Pencarian yang tidak akurat dapat menghambat proses penelitian dan pembelajaran, serta menurunkan efektivitas pemanfaatan dokumen yang tersedia.

Penelitian ini bertujuan mengembangkan sistem pencarian yang optimal dengan menerapkan metode *Term Frequency–Inverse Document Frequency* (TF-IDF) dan *Cosine Similarity* untuk meningkatkan relevansi hasil. TF-IDF menghitung bobot kata berdasarkan frekuensi dalam satu dokumen (TF) dan kelangkaan dalam seluruh dokumen (IDF). Kata kunci pengguna dan judul penelitian diubah menjadi vektor numerik, lalu dihitung kemiripannya menggunakan Cosine Similarity. Hasil pencarian ditampilkan lengkap dengan tautan langsung ke dokumen, sehingga pencarian menjadi lebih efisien bagi civitas akademika.

Pada Penelitian terdahulu Hasil pengujian kelayakan sistem berdasarkan pengujian ahli sistem menunjukkan hasil rata-rata persentase kelayakan sebesar 88,3% dan masuk dalam kriteria “sangat layak” untuk digunakan. Berdasarkan hasil klasifikasi yang dilakukan oleh sistem, sebanyak 49 dokumen telah berhasil diklasifikasikan ke dalam kategori yang tepat, sedangkan 1 dokumen tidak dapat terkласifikasikan dengan tepat. Persentase rata-rata hasil ketepatan klasifikasi yang dilakukan oleh sistem adalah sebesar 98%, maka dapat disimpulkan bahwa sistem dapat mengklasifikasikan dokumen ke dalam kategori yang sesuai dengan tingkat akurasi yang sangat tinggi [1].

## II. METODE



Gambar 1. Metodologi penelitian

Pada Gambar 1 menunjukkan metodologi penelitian yang disusun untuk mencapai tujuan, meliputi studi literatur, pengumpulan data, preprocessing, implementasi TF-IDF, perangkingan dengan cosine similarity, dan pengujian usability. Data diperoleh melalui web scraping dari repository Universitas Nusantara PGRI Kediri, menghasilkan sekitar 14.000 judul penelitian beserta tautan repository. Data ini menjadi objek utama yang diolah mulai dari preprocessing hingga pengujian. Pada tahap pencarian, pengguna memasukkan kata kunci yang diproses menggunakan TF-IDF dan cosine similarity untuk menghasilkan daftar dokumen paling relevan.

### 2.1 Studi literatur

Tahap awal penelitian difokuskan pada studi literatur yang komprehensif. Tujuan dari studi ini adalah untuk mengumpulkan landasan teoretis yang kuat dan meninjau penelitian-penelitian terdahulu yang relevan dengan sistem pencarian informasi,

arsitektur repository berbasis website, metode pembobotan kata seperti TF-IDF, algoritma pengukuran kemiripan teks khususnya Cosine Similarity, serta prinsip-prinsip dan metode pengujian usability untuk aplikasi web. Hasil kajian literatur ini menjadi dasar fundamental dalam perancangan, implementasi, dan evaluasi sistem pencarian yang dioptimalkan.

## 2.2 Pengumpulan data

Setelah pemahaman konseptual diperoleh, tahapan selanjutnya adalah pengumpulan data yang akan digunakan sebagai korpus dalam sistem pencarian. Data utama dalam penelitian ini adalah dokumen-dokumen ilmiah seperti skripsi, tesis, dan publikasi penelitian lainnya yang tersimpan dalam repository Universitas Nusantara PGRI Kediri (UNP Kediri), yang diakses melalui laman <https://repository.unpkediri.ac.id/>. Proses pengumpulan data dilakukan dengan teknik *web scraping* secara langsung dari laman repository tersebut. Informasi yang diambil meliputi metadata dokumen (seperti judul, abstrak, penulis, tahun terbit) serta tautan ke file dokumen. Seluruh data hasil *scraping* ini kemudian dikumpulkan dan distrukturkan ke dalam format JSON untuk pengolahan lebih lanjut. Data ini kemudian akan menjadi dasar bagi sistem untuk melakukan pencarian dan perangkingan.

## 2.3 Preprocessing Data

Data dokumen dalam format JSON diperoleh melalui web scraping dari laman repository Universitas Nusantara PGRI Kediri, berisi metadata seperti judul, penulis, tahun terbit, dan tautan ke dokumen lengkap. *Preprocessing* difokuskan pada bagian judul penelitian untuk membersihkan dan mentransformasi teks mentah menjadi format terstruktur yang siap digunakan dalam analisis pencarian dan klasifikasi. Tahap ini penting sebagai langkah awal pengolahan teks. Pada tahap ini, teknik preprocessing yang digunakan meliputi *case folding* (mengubah seluruh huruf menjadi huruf kecil), *tokenizing* (memecah kalimat menjadi kata-kata), *stopword removal* (menghilangkan kata-kata yang dianggap tidak penting dalam Bahasa Indonesia), dan *stemming* (mengubah kata ke bentuk dasarnya menggunakan library Sastrawi untuk Bahasa Indonesia). Proses ini bertujuan agar teks judul dari setiap penelitian menjadi lebih bersih, terstruktur, dan siap untuk tahap analisis berikutnya, seperti pembobotan atau klasifikasi[2][3][4].

## 2.4 Vectorizer dan Pembobotan menggunakan TF-IDF

Setelah data teks bersih diperoleh, langkah berikutnya adalah mengubah representasi teks menjadi format numerik (vektor) menggunakan teknik vectorizer dan melakukan pembobotan setiap term (kata) dalam dokumen dengan metode TF-IDF (*Term Frequency-Inverse Document Frequency*). TF mengukur frekuensi kemunculan sebuah term dalam dokumen, sedangkan IDF mengukur tingkat keunikan atau kepentingan term tersebut dalam keseluruhan koleksi dokumen. Dengan metode TF-IDF, setiap dokumen direpresentasikan sebagai vektor numerik di mana setiap dimensi merepresentasikan bobot dari term tertentu. Vektor inilah yang menjadi input utama dalam perhitungan kemiripan antar dokumen[5][6]. Berikut adalah rumus dari TF-IDF:

$$TF - IDF(t, d, D) = TF(t, d) \times IDF(t, D) \quad (1)$$

## 2.5 Contoh Perhitungan Pembobotan TF-IDF

Pada tahap ini disajikan contoh bagaimana sistem pencarian berbasis teks menghitung bobot kemunculan kata (term) menggunakan metode TF-IDF (*Term*

*Frequency–Inverse Document Frequency*). TF-IDF digunakan untuk mengukur seberapa penting suatu kata dalam sebuah dokumen relatif terhadap keseluruhan koleksi dokumen.

Contoh ini menggunakan tiga dokumen berupa judul penelitian:

- 1) D1: Implementasi YOLO untuk Deteksi Objek pada CCTV Berbasis Raspberry Pi
- 2) D2: Sistem Pendukung Keputusan Pemilihan Beasiswa Menggunakan Metode SAW
- 3) D3: Klasifikasi Citra Daun Menggunakan *Convolutional Neural Network* (CNN)

Sementara itu, digunakan sebuah *query* pencarian:

"deteksi objek menggunakan cnn"

*Query* ini akan dibandingkan dengan masing-masing dokumen untuk melihat seberapa besar tingkat kemiripannya, berdasarkan bobot kata yang muncul di antara keduanya.

Tabel 1 Nilai TF-IDF pada Dokumen dan *Query* untuk Beberapa Term Penting

Term	D1 - YOLO	D2 - SPK	D3 - CNN	<i>Query</i>
deteksi	0.26	0.00	0.00	. 0.52
objek	0.26	0.00	0.00	0.52
menggunakan	0.00	0.23	0.24	0.42
cnn	0.00	0.00	0.30	0.52
yolo	0.33	0.00	0.00	0.00
saw	0.00	0.37	0.00	0.00
citra	0.00	0.00	0.38	0.00

Dari tabel terlihat bahwa setiap term memiliki bobot TF-IDF yang berbeda di tiap dokumen, tergantung pada frekuensi kemunculan dan relevansinya terhadap seluruh koleksi. Semakin tinggi bobot suatu term, semakin besar kontribusinya dalam merepresentasikan isi dokumen. Misalnya, term "deteksi" dan "objek" memiliki bobot tinggi di Dokumen D1 karena relevan dengan topik YOLO, sedangkan "cnn" lebih menonjol di Dokumen D3. Nilai-nilai TF-IDF inilah yang digunakan dalam perhitungan *Cosine Similarity* untuk mengukur kemiripan antara query dan dokumen. Hanya term yang muncul di keduanya yang memengaruhi hasil pencarian.

## 2.6 Perangkingan Berdasarkan Kecocokan Dokumen Menggunakan *Cosine Similarity*

Sistem pencarian dalam penelitian ini berfokus pada mekanisme perangkingan dokumen berdasarkan tingkat kecocokan dengan kueri pengguna. Untuk mengukur kemiripan antara vektor kueri dan vektor dokumen, yang keduanya telah melalui tahap preprocessing dan pembobotan TF-IDF, digunakan algoritma *Cosine Similarity*. Metode ini menghitung nilai kosinus dari sudut antara dua vektor, dengan nilai antara 0 hingga 1, di mana nilai yang mendekati 1 menandakan kemiripan yang tinggi. Nilai Cosine Similarity tersebut digunakan untuk mengurutkan dokumen sehingga dokumen paling relevan tampil di posisi teratas hasil pencarian[7]. Berikut adalah rumus dari *Cosine Similarity*:

$$\text{Similarity } (A, B) = \frac{A \cdot B}{\|A\| \cdot \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \cdot \sqrt{\sum_{i=1}^n (B_i)^2}} \quad (2)$$

## 2.7 Penjelasan Perhitungan *Cosine Similarity*

Untuk mengukur seberapa mirip query dengan setiap dokumen, digunakan metode Cosine Similarity. Metode ini menghitung sudut antara dua vektor (dokumen dan query) yang dibentuk dari bobot TF-IDF setiap *term*. Semakin kecil sudutnya (semakin mendekati 1 nilainya), semakin mirip kedua teks tersebut. Proses ini tidak menghitung berapa kali term muncul secara langsung, tetapi mempertimbangkan bobot *term* di seluruh dokumen (TF-IDF), sehingga menghasilkan perbandingan yang lebih adil.

Tabel 2 Hasil Kemiripan antara Query dan Dokumen

Term	Cosine Similarity ke Query
D1 - YOLO	0.271
D3 - CNN	0.257
D2 - SPK	0.099

Dari hasil di atas, dapat dilihat bahwa dokumen D1 memiliki nilai kemiripan tertinggi terhadap query karena mengandung kata-kata yang relevan seperti “deteksi” dan “objek”. Meskipun D3 juga relevan (mengandung cnn dan menggunakan), kombinasi *term* pada D1 lebih mendekati struktur *query* secara keseluruhan.

#### 2.8 Pengujian dengan Metode *Usability*

Pengujian *usability* dilakukan melalui observasi langsung terhadap interaksi pengguna saat menggunakan sistem pencarian berbasis web yang didukung algoritma *Cosine Similarity*. Evaluasi difokuskan pada efektivitas dan efisiensi, seperti waktu penyelesaian tugas dan jumlah langkah pencarian. Pengujian melibatkan kuisioner dan uji coba langsung untuk memperoleh gambaran objektif tentang perilaku dan performa pengguna.. Pendekatan observasi langsung ini didukung oleh penelitian-penelitian sebelumnya yang menegaskan bahwa metode ini memungkinkan peneliti menangkap perilaku nyata pengguna, kesalahan, dan pola navigasi secara real-time yang seringkali tidak terungkap melalui survei atau kuesioner, sehingga memberikan data empiris yang objektif dan minim bias untuk mengidentifikasi masalah *usability* serta meningkatkan efektivitas dan efisiensi sistem[8][9].

#### 2.9 Teknologi yang digunakan

##### 1) React.js

Antarmuka pengguna (*User Interface/UI*) sistem ini dikembangkan menggunakan React.js, sebuah pustaka JavaScript yang deklaratif dan efisien untuk membangun UI berbasis komponen yang kompleks. React.js memungkinkan pengembang membuat antarmuka pengguna yang dinamis dan interaktif dengan memperbarui hanya bagian DOM yang berubah menggunakan algoritma diffing cerdas, sehingga mengoptimalkan performa aplikasi dan pengalaman pengguna. Pendekatan berbasis komponen juga memudahkan pengelolaan kode dan meningkatkan modularitas aplikasi[10].

##### 2) API Django

Sistem backend menggunakan API berbasis Django REST Framework untuk menjembatani komunikasi antara frontend React.js dan database. Django dipilih karena keamanannya, struktur yang solid, dan kemampuannya membangun API RESTful secara efisien. API ini menangani permintaan data, menjalankan proses

pencarian dengan TF-IDF dan Cosine Similarity, serta mengirimkan hasil ke pengguna. Autentikasi dilakukan menggunakan JSON Web Token (JWT) untuk menjaga keamanan data[11]. API ini bertanggung jawab untuk menangani permintaan data, melakukan proses pencarian dan pembobotan menggunakan TF-IDF serta Cosine Similarity, dan mengirimkan hasilnya kembali ke antarmuka pengguna.

3) VPS

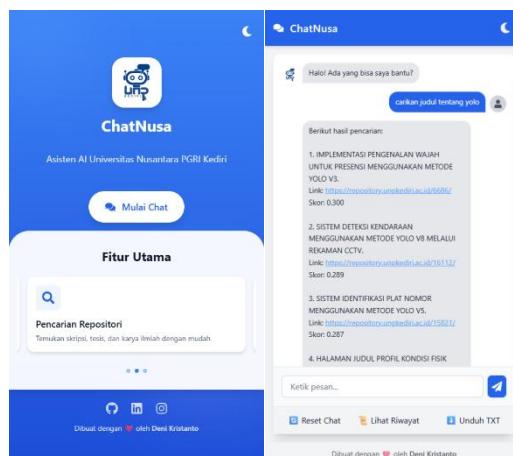
Penempatan backend, termasuk API Django, dilakukan pada layanan *Virtual Private Server* (VPS) yang menyediakan lingkungan server terisolasi dengan sumber daya khusus. VPS memungkinkan kontrol penuh terhadap konfigurasi, instalasi perangkat lunak, dan pengelolaan keamanan, sehingga mendukung operasi backend yang stabil dan andal. VPS juga memungkinkan pengelolaan sumber daya yang fleksibel dan skalabilitas yang diperlukan untuk menjaga performa aplikasi[12].

4) Vercel

Penempatan aplikasi *frontend* yang dikembangkan menggunakan React.js dilakukan melalui platform Vercel. Vercel merupakan layanan *cloud computing* yang mendukung penyebaran otomatis, integrasi berkelanjutan (*continuous integration*), serta kompatibel dengan berbagai kerangka kerja JavaScript modern. Platform ini memungkinkan distribusi antarmuka pengguna secara cepat dan efisien melalui infrastruktur global. Vercel menyediakan continuous deployment dengan zero configuration melalui integrasi Git, serta menggunakan jaringan edge global untuk memastikan performa tinggi dan latensi rendah di seluruh dunia[13].

### III. HASIL DAN PEMBAHASAN

Penelitian ini berhasil mengimplementasikan fitur pencarian yang dioptimalkan untuk repository Universitas Nusantara PGRI Kediri melalui antarmuka percakapan berbasis web. Proses pencarian menggunakan metode *TF-IDF* dan *Cosine Similarity* untuk menghitung relevansi antara kueri dan dokumen. Pada Gambar 2 menampilkan antarmuka sistem, mulai dari *landing page* hingga halaman percakapan interaktif tempat pengguna melakukan pencarian dan menerima hasil.



Gambar 2. Halaman *Landing page* dan halaman chat

### 3.1 Pengujian Menggunakan Metode *Usability Testing*.

Sebelum sistem diuji coba di lapangan, dilakukan evaluasi awal menggunakan metode *usability testing* untuk mengetahui sejauh mana sistem mudah digunakan oleh pengguna. Pengujian ini bertujuan untuk menilai keberhasilan pengguna dalam menyelesaikan tugas-tugas tertentu, waktu penyelesaian tugas, serta mencatat observasi terhadap interaksi yang terjadi selama proses pengujian. Pengujian dilakukan terhadap sepuluh tugas utama yang mencerminkan alur penggunaan sistem, mulai dari membuka halaman utama hingga mengakses fitur riwayat pencarian. Berikut Setiap tugas diamati berdasarkan tabel dibawah ini:

*Tabel 3. Testing Usability*

No	Tugas Pengguna	Berhasil?	Waktu Selesai (detik)	Hasil Observasi / Catatan
1	Membuka halaman utama	Ya	3	Tampilan bersih, elemen langsung terlihat dan mudah dipahami.
2	Mengklik tombol “Mulai Chat”	Ya	4	Tombol terlihat jelas, desain menarik dan mudah digunakan.
3	Mengetik dan mengirim pertanyaan	Ya	8	Kolom input responsif, tidak ada <i>lag</i> saat mengetik.
4	Menunggu respon dengan efek animasi “mengetik”	Ya	2	Efek animasi muncul tepat waktu, menambah kesan interaktif.
5	Melihat hasil pencarian	Ya	6	Hasil tersusun rapi, skor relevansi tampil dengan jelas.
6	Mengakses dokumen dari link hasil pencarian	Ya	3	Link aktif dan langsung membuka halaman dokumen yang tepat.
7	Membuka fitur “Lihat Riwayat”	Ya	4	Riwayat tampil dalam modal popup, navigasi lancar.
8	Mengaktifkan dark mode	Ya	2	Mode langsung berubah, tidak ada gangguan tampilan.
9	Mengunduh riwayat chat	Ya	5	File berhasil terunduh dengan isi yang sesuai.
10	Mengakses melalui perangkat mobile	Ya	-	Tampilan responsif, tetap rapi dan fungsional di layar kecil

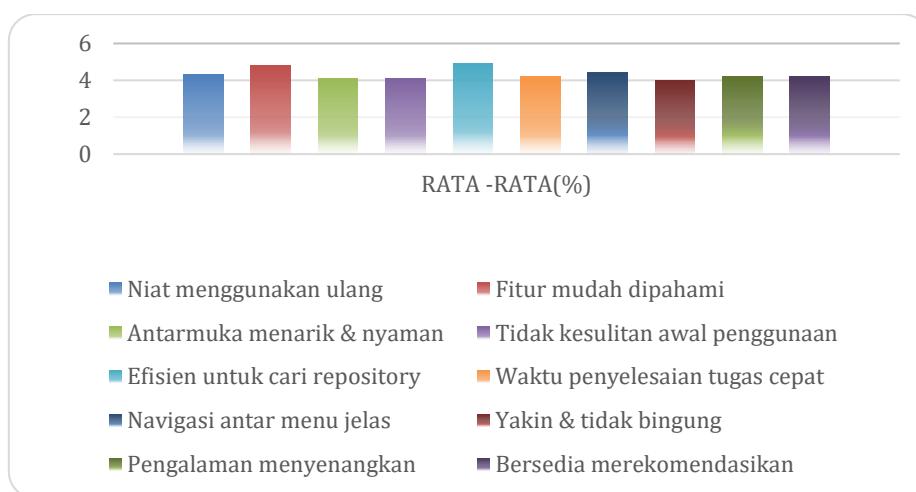
### 3.2 Hasil Kuisioner Pengguna

Setelah pelaksanaan uji coba langsung, responden diminta untuk mengisi kuesioner melalui Google Form guna memberikan penilaian terhadap kemudahan penggunaan, tampilan, dan fungsionalitas sistem. Kuesioner ini diberikan kepada 10 mahasiswa dan satu orang dosen yang jabat sebagai Kepala Perpustakaan Universitas Nusantara PGRI Kediri.

*Tabel 4. Rata-rata Hasil Kuisioner Evaluasi Usability*

No	Pernyataan	Nilai Rata-rata (%)
1	Saya berpikir bahwa saya akan sering menggunakan Chat Nusa.	4,3
2	Saya merasa fitur-fitur dalam Chat Nusa mudah dipahami.	4,8

3	Saya merasa antarmuka Chat Nusa menarik dan nyaman digunakan.	4,1
4	Saya tidak merasa kesulitan saat menggunakan aplikasi ini untuk pertama kali.	4,1
5	Saya merasa Chat Nusa dapat membantu saya dalam mencari repository secara efisien.	4,9
6	Waktu yang dibutuhkan untuk menyelesaikan tugas di aplikasi ini terasa cepat.	4,2
7	Navigasi antar menu atau halaman di Chat Nusa terasa jelas dan mudah	4,4
8	Saya merasa yakin dan tidak bingung saat menggunakan aplikasi ini.	4
9	Saya merasa pengalaman menggunakan Chat Nusa menyenangkan.	4,2
10	Saya bersedia merekomendasikan Chat Nusa kepada orang lain.	4,2



Gambar 3. Diagram Batang Rata-rata Penilaian Responden

Hasil evaluasi menunjukkan bahwa aplikasi mendapatkan respons positif dari pengguna. Aspek yang paling menonjol adalah kemudahan memahami fitur dengan nilai rata-rata tertinggi sebesar 4,8. Secara umum, aplikasi dinilai mudah digunakan, efisien, dan layak untuk digunakan kembali maupun direkomendasikan. Beberapa aspek seperti antarmuka dan keyakinan dalam penggunaan masih dapat ditingkatkan untuk pengalaman yang lebih optimal.

### 3.3 Dokumentasi Hasil Uji Coba Sistem

Kegiatan uji coba sistem dilaksanakan di ruang Perpustakaan. Mahasiswa dan dosen diminta untuk mencoba aplikasi sesuai skenario tugas yang disediakan, dengan pendampingan langsung dari peneliti. Dokumentasi visual dilakukan untuk merekam proses penggunaan sistem secara langsung.

Berikut beberapa dokumentasi foto selama proses uji coba:



Gambar 4. Pengujian aplikasi bersama dosen kepala perpustakaan.



Gambar 5. Pengujian aplikasi bersama mahasiswa

#### IV. KESIMPULAN

Penelitian ini berhasil mengembangkan aplikasi pencarian pada repository berbasis web Universitas Nusantara PGRI Kediri dengan menerapkan metode TF-IDF untuk pembobotan *term* dan *Cosine Similarity* sebagai metrik kemiripan kueri-dokumen. Aplikasi ini mampu menampilkan hasil berupa judul dokumen, skor relevansi, dan tautan langsung ke repository, sehingga meningkatkan efisiensi akses informasi ilmiah.

Pengujian *usability* melibatkan 10 responden mahasiswa dan 1 dosen selaku Kepala Perpustakaan, dengan hasil menunjukkan tingkat keberhasilan pencarian 100% dan waktu respons rata-rata 2 hingga 8 detik per kueri. Rata-rata penilaian terhadap fitur dan antarmuka berada di atas 4 dari skala 5, menandakan bahwa fitur seperti mode gelap, riwayat pencarian, dan tampilan responsif telah mendukung pengalaman pengguna secara optimal. Namun, pendekatan berbasis TF-IDF dan *Cosine Similarity* masih terbatas dalam memahami konteks semantik. Untuk pengembangan selanjutnya, disarankan integrasi model *transformer* seperti BERT (*Bidirectional Encoder Representations from Transformers*) guna meningkatkan akurasi dan relevansi hasil pencarian.

#### UCAPAN TERIMAKASIH

Penulis juga ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada pihak Khususnya Universitas Nusantara PGRI Kediri atas ketersediaan data dokumen dari repository. Data tersebut telah menjadi landasan utama dan sangat berharga bagi pelaksanaan penelitian serta pengembangan sistem ini.

#### DAFTAR PUSTAKA

- [1] R. T. Wahyuni, D. Prastyanto, and E. Supraptono, "Penerapan Algoritma Cosine

- Similarity dan Pembobotan TF-IDF pada Sistem Klasifikasi Dokumen Skripsi,” *J. Tek. Elektro Univ. Negeri Semarang*, vol. 9, no. 1, pp. 18–23, 2017, [Online]. Available: <https://journal.unnes.ac.id/nju/index.php/jte/article/download/10955/6659>
- [2] S. Khairunnisa, A. Adiwijaya, and S. Al Faraby, “Pengaruh Text Preprocessing terhadap Analisis Sentimen Komentar Masyarakat pada Media Sosial Twitter (Studi Kasus Pandemi COVID-19),” *J. Media Inform. Budidarma*, vol. 5, no. 2, p. 406, 2021, doi: 10.30865/mib.v5i2.2835.
- [3] D. Rifaldi, Abdul Fadlil, and Herman, “Teknik Preprocessing Pada Text Mining Menggunakan Data Tweet ‘Mental Health,’” *Decod. J. Pendidik. Teknol. Inf.*, vol. 3, no. 2, pp. 161–171, 2023, doi: 10.51454/decode.v3i2.131.
- [4] M. U. Albab, Y. Karuniawati P, and M. N. Fawaiq, “Optimization of the Stemming Technique on Text preprocessing President 3 Periods Topic,” *J. Transform.*, vol. 20, no. 2, pp. 1–10, 2023, [Online]. Available: <https://journals.usm.ac.id/index.php/transformatika/#page1>
- [5] A. Yodi Prayoga, A. Id Hadiana, and F. Rakhmat Umbara, “Deteksi Hoax pada Berita Online Bahasa Inggris Menggunakan Bernoulli Naïve Bayes dengan Ekstraksi Fitur Tf-Idf,” *J. Syntax Admiration*, vol. 2, no. 10, pp. 1808–1823, 2021, doi: 10.46799/jsa.v2i10.327.
- [6] T. I. Alfawas and A. Rahim, “Penerapan Fitur Ekstraksi TF-IDF untuk Analisis Sentimen Ulasan Game Bus Simulator Indonesia dengan Algoritma Naive Bayes,” vol. 4, pp. 3177–3193, 2024.
- [7] E. Poerbaningtyas and R. Maulidi, “Penerapan Metode Cosine Similarity Pada Sistem Informasi Retrieval Pencarian Terjemahan Ayat-Ayat Suci Al-Qur’An,” *J-Intech*, vol. 11, no. 2, pp. 325–335, 2023, doi: 10.32664/j-intech.v11i2.1020.
- [8] A. I. Anshori, H. Aryadita, and H. M. Az-zahra, “Evaluasi Usability Pada Sistem Monitoring Pengadaan Menggunakan Metode Usability Testing ( Studi Kasus PT Pembangkitan Jawa-Bali ),” *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 3, no. 1, pp. 119–128, 2019.
- [9] M. Farhat, B. A. Nugroho, and A. Yusuf, “EVALUASI USER INTERFACE DAN USER EXPERIENCE WEBSITE JUMPSTART COFFEE MENGGUNAKAN METODE USABILITY TESTING DAN SYSTEM USABILITY SCALE,” vol. 8, no. 6, pp. 12313–12320, 2024.
- [10] N. Raina, “Pengembangan React Js Pada Frontend Website Pengaduan Dan Pelayanan Publik Menggunakan Metode Scrum ( Studi Kasus : Dprd Jawa Barat ),” vol. 11, no. 4, pp. 4292–4300, 2024.
- [11] F. Wijaya, A. Jacobus, and A. Sambul, “Implementasi Web Service Pada Sistem Informasi Perpustakaan Perguruan Tinggi,” *J. Tek. Elektro dan Komput.*, vol. 16, no. 4, pp. 1–8, 2021, [Online]. Available: <https://ejournal.unsrat.ac.id/v3/index.php/informatika/article/view/34226>
- [12] A. Hidayat and D. Prabowo, “Implementation of Virtual Private Server (VPS) Using Digital Ocean Cloud Server on BMT. Mentari East Lampung,” *Jtksi*, vol. 03, no. 03, pp. 116–121, 2020.
- [13] Vercel, “How to Deploy a React Site with Vercel.” Accessed: Jun. 06, 2025. [Online]. Available: <https://vercel.com/guides/deploying-react-with-vercel>