

---

# **DEVELOPMENT OF A SEMANTIC-BASED COLLECTION SEARCH SYSTEM PROTOTYPE ON THE ONLINE PUBLIC ACCESS CATALOG (OPAC) WEBSITE OF THE STMIK WIDYA CIPTA DHARMA LIBRARY**

Sandy Febrian<sup>1)</sup>, Amelia Yusnita<sup>2)</sup>, dan Ita Arfyanti<sup>3)</sup>

<sup>1,2,3</sup>Teknik Informatika, STMIK Widya Cipta Dharma  
<sup>1,2,3</sup>Jl. M. Yamin No. 25, 75123

E-mail: 2143060@wicida.ac.id<sup>1)</sup>, amelia@ wicida.ac.id<sup>2)</sup>, ita@ wicida.ac.id<sup>3)</sup>

## **ABSTRACT**

The Online Public Access Catalog (OPAC) system on the website of the STMIK Widya Cipta Dharma Library still relies on literal keyword matching in the information retrieval process. This approach causes the system to be unable to effectively understand the context of user queries and to optimally handle spelling errors, which in turn reduces the effectiveness of information retrieval. Therefore, this study aims to develop a semantic-based OPAC search system prototype to improve the relevance of search results. The system was developed using the Prototyping method, which consists of the stages of initial communication; rapid planning and design; prototype development; deployment, delivery, and feedback; as well as prototype refinement. Semantic search was implemented by utilizing the Sentence-Bidirectional Encoder Representations from Transformers (S-BERT) model to transform search keywords into semantic vector representations. Facebook AI Similarity Search (FAISS) was employed as the vector search mechanism, while Cosine Similarity was used to measure the level of similarity between data. Spelling error handling was performed using RapidFuzz with the Levenshtein Distance algorithm and was complemented by search suggestion. In addition, the system provides a collection recommendation feature using a Content-Based Filtering approach based on users' search history. The testing results indicate that the system is able to present relevant search results even when the entered keywords do not exactly match the collection data or contain spelling errors. User evaluation was conducted using a five-point Likert scale questionnaire analyzed with the Mean analysis method. The evaluation results show an average score of 4.0 (good) for search result relevance, 4.4 (very good) for spelling error handling, 4.2 (good) for ease of use, 4.1 (good) for the usefulness of the recommendation feature, and 4.4 (very good) for overall system feasibility. Based on these results, the developed semantic-based OPAC system prototype is considered effective and well accepted by users.

**Keywords:** OPAC, Semantic Search, Information Retrieval System, Spelling Error Handling, Collection Recommendation

---

# **PENGEMBANGAN PROTOTIPE SISTEM PENCARIAN KOLEKSI BERBASIS SEMANTIK PADA WEBSITE ONLINE PUBLIC ACCESS CATALOG (OPAC) PERPUSTAKAAN STMIK WIDYA CIPTA DHARMA**

## **ABSTRAK**

Sistem Online Public Access Catalog (OPAC) pada website Perpustakaan STMIK Widya Cipta Dharma masih menggunakan pencocokan kata kunci secara *literal* dalam proses pencarian informasi. Pendekatan ini menyebabkan sistem belum mampu memahami konteks pencarian pengguna dan belum optimal dalam menangani kesalahan penulisan kata kunci, sehingga berdampak pada rendahnya efektivitas pencarian informasi. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan prototipe sistem pencarian OPAC berbasis semantik guna meningkatkan relevansi hasil pencarian. Pengembangan sistem dilakukan menggunakan metode Prototyping yang meliputi tahapan komunikasi awal; perancangan dan desain cepat; pembangunan prototipe; penyebaran, pengiriman, dan umpan balik; serta penyempurnaan prototipe. Pencarian semantik diterapkan dengan memanfaatkan model *Sentence-Bidirectional Encoder Representations from Transformers* (S-BERT) untuk mengubah kata kunci menjadi representasi vektor semantik, *Facebook AI Similarity Search* (FAISS) sebagai mekanisme pencarian vektor, serta algoritma Cosine Similarity untuk mengukur tingkat kemiripan antar data. Penanganan kesalahan penulisan kata kunci dilakukan menggunakan RapidFuzz dengan algoritma Levenshtein Distance, serta dilengkapi fitur saran pencarian. Sistem juga menyediakan fitur rekomendasi koleksi menggunakan pendekatan *Content-Based Filtering* berdasarkan riwayat pencarian pengguna. Hasil pengujian menunjukkan bahwa sistem mampu menampilkan hasil pencarian yang relevan meskipun kata kunci tidak sama persis dengan data koleksi atau mengandung kesalahan penulisan. Evaluasi pengguna menggunakan kuesioner skala Likert lima tingkat dengan metode *Mean analysis* menunjukkan bahwa relevansi hasil pencarian memperoleh nilai rata-rata 4,0 (baik), penanganan kesalahan

---

penulisan 4,4 (sangat baik), kemudahan penggunaan 4,2 (baik), manfaat fitur rekomendasi 4,1 (baik), serta kelayakan sistem secara keseluruhan 4,4 (sangat baik). Berdasarkan hasil tersebut, prototipe sistem OPAC berbasis semantik dinyatakan efektif dan dapat diterima dengan baik oleh pengguna.

**Kata Kunci:** OPAC, Pencarian Semantik, Sistem Temu Kembali Informasi, Penanganan Kesalahan Penulisan, Rekomendasi Koleksi

---

## 1. PENDAHULUAN

Perkembangan teknologi informasi saat ini telah mengubah cara manusia dalam memperoleh informasi, termasuk dalam pengelolaan dan penyebaran informasi di perpustakaan. Informasi yang sebelumnya hanya dapat diperoleh melalui pencarian secara manual kini dapat diakses secara daring seiring dengan diterapkannya proses digitalisasi pada sistem informasi perpustakaan. Digitalisasi tersebut mengubah koleksi perpustakaan ke dalam format elektronik sehingga lebih mudah diakses dan dikelola. Transformasi ini bertujuan untuk meningkatkan kemudahan dan kecepatan dalam memperoleh sumber informasi yang dibutuhkan.

Dalam mendukung kegiatan akademik di lingkungan STMIK Widya Cipta Dharma, perpustakaan menyediakan layanan sistem pencarian koleksi berbasis web melalui fitur Online Public Access Catalog (OPAC). Fitur ini digunakan untuk membantu pengguna dalam menemukan informasi terkait koleksi buku, karya ilmiah, skripsi, serta sumber daya lainnya berdasarkan kata kunci tertentu. Namun, sistem pencarian yang diterapkan pada layanan tersebut belum sepenuhnya mampu menyajikan informasi secara optimal. Hal ini disebabkan karena hasil pencarian masih memiliki keterbatasan dalam tingkat relevansi, di mana sistem hanya mengandalkan pencocokan kata kunci secara *literal*.

Fenomena yang terjadi menunjukkan bahwa pengguna sering mengalami kesulitan dalam menemukan referensi yang relevan karena sistem hanya menampilkan hasil pencarian berdasarkan kesesuaian kata kunci dengan *metadata* yang tersimpan. Apabila pengguna tidak mengetahui kata kunci yang tepat atau melakukan kesalahan dalam penulisan, maka hasil pencarian yang ditampilkan menjadi tidak sesuai dengan kebutuhan informasi. Kondisi tersebut dapat menghambat efisiensi pencarian informasi dan berdampak pada produktivitas mahasiswa. Hal ini menunjukkan perlunya pengembangan sistem pencarian yang mampu memahami konteks dan makna dari *input* pengguna, bukan sekadar melakukan pencocokan kata secara *literal*.

Permasalahan utama terletak pada penerapan teknologi *information retrieval* (IR) yang belum mampu memahami makna dan konteks kata kunci secara mendalam. Selain itu, kesalahan penulisan kata kunci oleh pengguna dapat menyebabkan hasil pencarian tidak muncul atau tidak relevan, sehingga menurunkan kualitas pengalaman pengguna. Di sisi lain, sistem yang ada belum menyediakan fitur rekomendasi koleksi berdasarkan riwayat pencarian pengguna, padahal informasi tersebut dapat dimanfaatkan untuk

mengidentifikasi kebutuhan pengguna dan menyarankan koleksi yang memiliki keterkaitan.

Oleh karena itu, diperlukan pengembangan sistem pencarian yang mampu memahami makna kata kunci secara lebih mendalam guna meningkatkan efektivitas pencarian informasi. Pendekatan berbasis kemiripan makna (*semantic search*) dengan menggunakan teknik *Natural Language Processing* (NLP) memungkinkan sistem untuk memproses dan memahami bahasa alami secara kontekstual. Model *Sentence-Bidirectional Encoder Representations from Transformers* (S-BERT) digunakan untuk mengubah kalimat atau frasa menjadi representasi vektor semantik (*semantic embedding*). Untuk mempercepat proses pencarian vektor yang memiliki tingkat kemiripan tinggi, digunakan pustaka *Facebook AI Similarity Search* (FAISS) sebagai metode indeks dan pencarian berbasis vektor, yang dikombinasikan dengan algoritma Cosine Similarity sebagai pengukur tingkat kemiripan secara konseptual.

Selain itu, dalam menangani kesalahan penulisan kata kunci, digunakan pustaka RapidFuzz dengan pendekatan algoritma Levenshtein Distance untuk mengukur tingkat kemiripan karakter antar string. Untuk mendukung proses pencarian sejak tahap awal pengetikan, sistem juga dilengkapi fitur saran pencarian (*suggestion/autocomplete*) yang membantu pengguna dalam menentukan kata kunci yang sesuai dalam mengurangi kesalahan *input* dan meningkatkan akurasi pencarian sebelum proses pencocokan dilakukan. Sementara itu, sistem rekomendasi dikembangkan menggunakan pendekatan *Content-Based Filtering* dengan memanfaatkan riwayat pencarian pengguna untuk memberikan rekomendasi koleksi yang relevan berdasarkan kesamaan kategori, di mana pengukuran kemiripan juga dilakukan menggunakan algoritma Cosine Similarity.

Pendekatan ini memungkinkan sistem pencarian memahami makna kata kunci secara lebih komprehensif, tidak hanya bergantung pada pencocokan kata secara *literal*, tetapi juga mampu menangani kesalahan penulisan serta memberikan rekomendasi berdasarkan riwayat pencarian pengguna. Tujuan dari penelitian ini adalah menghasilkan sistem pencarian yang cerdas, responsif, dan relevan dengan kebutuhan pengguna. Penelitian ini berfokus pada pengembangan sistem yang diharapkan dapat diterapkan pada website Perpustakaan STMIK Widya Cipta Dharma guna meningkatkan kualitas layanan pencarian informasi.

## 2. RUANG LINGKUP

### 2.1 Rumusan Masalah

Bagaimana mengoptimalkan sistem pencarian Online Public Access Catalog (OPAC) perpustakaan STMIK Widya Cipta Dharma melalui penerapan *Natural Language Processing* (NLP) untuk meningkatkan akurasi pencarian dan kualitas, serta pengalaman pengguna?

### 2.2 Batasan Masalah

Permasalahan yang ada di dalam penelitian memiliki batasan-batasan tertentu untuk memastikan penelitian berfokus pada permasalahan dan memperjelas ruang lingkup pada penelitian, yaitu antara lain sebagai berikut.

1. Pengembangan sistem pencarian dibatasi pada fitur Online Public Access Catalog (OPAC) yang terdapat pada website Perpustakaan STMIK Widya Cipta Dharma, dengan tujuan untuk meningkatkan relevansi dan efektivitas hasil pencarian koleksi perpustakaan.
2. Ruang lingkup pengembangan model sistem pencarian difokuskan pada fitur-fitur yang berkaitan dengan penggunaan kata kunci dalam sistem pencarian, meliputi pemahaman semantik, penanganan kesalahan penulisan, serta penyediaan rekomendasi berdasarkan riwayat pencarian pengguna.
3. Implementasi dan pengujian sistem pencarian dilakukan pada sebuah website yang dirancang sebagai media simulasi dari sistem yang dikembangkan. Beberapa penyesuaian dilakukan pada fitur tertentu, seperti filter pencarian, saran pencarian, penanganan kesalahan pencarian, dan rekomendasi berdasarkan pencarian. Selain itu dilakukan juga penyesuaian

### 2.3 Tujuan Penelitian

Penelitian ini dilakukan untuk mengembangkan metode sistem pencarian agar dapat memahami maksud dari kata kunci dengan teknologi semantik pada website perpustakaan STMIK Widya Cipta Dharma, sehingga dapat menghasilkan topik yang relevan dengan kebutuhan.

### 2.4 Manfaat

Permasalahan yang ada di dalam penelitian memiliki batasan-batasan tertentu untuk memastikan penelitian berfokus pada permasalahan dan memperjelas ruang lingkup pada penelitian, yaitu antara lain sebagai berikut.

#### 1. Bagi Mahasiswa

Penelitian ini memberikan pengalaman praktis bagi mahasiswa untuk mengembangkan kemampuan teknis dalam membangun sistem pencarian yang berbasis pada teknologi *Natural Language Processing* (NLP). Selain itu, penelitian ini memberikan kesempatan untuk memahami konsep secara mendalam, termasuk dalam menerapkan analisis mengenai metode yang cocok digunakan

dalam mengatasi permasalahan yang ada. Melalui proses ini, mahasiswa juga dilatih untuk mengintegrasikan teori yang diperoleh selama perkuliahan dengan penerapan nyata dalam pengembangan sistem berbasis teknologi informasi.

#### 2. Bagi Perguruan Tinggi

Penelitian ini memberikan perguruan tinggi gambaran mengenai metode yang dapat dikembangkan pada sistem pencarian di dalam Online Public Access Catalog (OPAC) untuk pencarian yang lebih fleksibel dan efisien pada website perpusatkaan STMIK Widya Cipta Dharma. Hal ini akan berdampak pada proses kegiatan akademik kampus yang lebih baik dan maksimal. Sehingga, penelitian ini diharapkan dapat mendukung pengembangan sistem informasi, meningkatkan aksesibilitas sumber daya akademik.

## 3. BAHAN DAN METODE

### 3.1 Online Public Access Catalog (OPAC)

Online Public Access Catalog (OPAC) merupakan sistem katalog perpustakaan berbasis komputer yang digunakan untuk mengelola dan menelusuri koleksi perpustakaan secara digital. OPAC menyediakan informasi bibliografis serta lokasi penyimpanan koleksi, sehingga memudahkan pengguna dalam menemukan sumber informasi yang dibutuhkan (Elsadantia, 2023). Sistem OPAC yang terkomputerisasi mampu meningkatkan efisiensi pengelolaan data dan pelayanan perpustakaan dibandingkan sistem manual (Kusnandar & Anshary, 2022).

Dalam penelitian ini, OPAC digunakan sebagai objek pengembangan sistem pencarian koleksi. Fokus pengembangan diarahkan pada mekanisme pencarian informasi agar mampu menampilkan hasil yang lebih relevan berdasarkan kebutuhan pengguna, khususnya melalui penerapan pencarian berbasis kemiripan makna.

### 3.2 Sistem Pencarian Berbasis Kemiripan (Semantik)

Sistem pencarian berbasis kemiripan (semantik) merupakan metode pencarian informasi yang menekankan pada kesamaan makna antar data, bukan hanya kesesuaian kata kunci secara literal. Pendekatan ini memungkinkan sistem memahami hubungan semantik antar konsep dalam basis data sehingga hasil pencarian menjadi lebih fleksibel dan relevan (Al'Izza dkk., 2022).

Penerapan pencarian semantik memungkinkan sistem menampilkan hasil yang sesuai meskipun terdapat perbedaan istilah, struktur kalimat, atau variasi penulisan kata kunci (Umri dkk., 2025). Dalam penelitian ini, pencarian semantik digunakan sebagai pendekatan utama untuk meningkatkan kualitas temu kembali informasi pada sistem OPAC.

### 3.3 Sistem Rekomendasi

Sistem rekomendasi merupakan aplikasi yang dirancang untuk memberikan saran item yang relevan

---

kepada pengguna berdasarkan data dan pola tertentu. Sistem ini bertujuan membantu pengguna dalam menemukan informasi yang sesuai tanpa harus melakukan pencarian berulang (Murti dkk., 2019).

Pada penelitian ini, sistem rekomendasi dikembangkan untuk menyarankan koleksi perpustakaan berdasarkan riwayat pencarian pengguna. Rekomendasi diharapkan dapat meningkatkan efektivitas penelusuran informasi serta memperkaya pengalaman pengguna dalam menggunakan sistem OPAC.

### 3.4 Natural Language Processing (NLP)

*Natural Language Processing* (NLP) merupakan bidang kecerdasan buatan yang berfokus pada pemrosesan dan pemahaman bahasa alami manusia oleh komputer (Rantung, 2023). NLP memungkinkan sistem menganalisis teks secara linguistik dan semantik agar dapat diproses lebih lanjut.

Dalam penelitian ini, NLP digunakan untuk memproses metadata koleksi dan masukan pencarian pengguna melalui tahapan pra-pemrosesan teks, meliputi *lowercasing*, *tokenization*, *stopword removal*, dan *stemming*. Tahapan ini bertujuan menormalkan teks sehingga representasi semantik dapat dibentuk secara lebih konsisten.

### 3.5 Sentence-Bidirectional Encoder Representation from Transformers (S-BERT)

*Sentence-Bidirectional Encoder Representation from Transformers* (S-BERT) merupakan pengembangan dari model BERT yang dirancang untuk menghasilkan embedding kalimat secara efisien dan kontekstual. Model ini memungkinkan perbandingan semantik antar kalimat secara langsung tanpa perhitungan embedding ulang pada setiap pencarian (Agiharta dkk., 2024).

Dalam sistem yang dikembangkan, S-BERT digunakan untuk mengubah kata kunci pencarian pengguna dan metadata koleksi menjadi representasi vektor semantik. Representasi ini memungkinkan sistem memahami kesamaan makna antar teks meskipun terdapat perbedaan kata atau susunan kalimat (Samosir dkk., 2022).

### 3.6 Facebook AI Similarity Search (FAISS)

*Facebook AI Similarity Search* (FAISS) merupakan pustaka open-source yang digunakan untuk melakukan pengindeksan dan pencarian kemiripan vektor berdimensi tinggi secara efisien (Krisnawati dkk., 2024). FAISS dirancang untuk mendukung sistem pencarian berbasis embedding yang membutuhkan kecepatan dan skalabilitas tinggi.

Dalam penelitian ini, FAISS digunakan untuk mengindeks embedding hasil representasi S-BERT sehingga proses pencarian dokumen dengan tingkat kemiripan semantik tertinggi dapat dilakukan secara cepat dan efisien.

### 3.7 Cosine Similarity

Cosine Similarity merupakan metode pengukuran kemiripan yang menghitung nilai kosinus sudut antara dua vektor dalam ruang multidimensi. Nilai kemiripan yang mendekati 1 menunjukkan tingkat kesamaan semantik yang tinggi (Meilina dkk., 2021).

Metode ini digunakan untuk menghitung tingkat relevansi antara embedding kata kunci pencarian dan embedding metadata koleksi, serta sebagai dasar pemeringkatan hasil pencarian dan rekomendasi yang ditampilkan kepada pengguna.

### 3.8 RafidFuzz

RafidFuzz merupakan pustaka *fuzzy string matching* yang digunakan untuk menangani kesalahan penulisan kata kunci melalui pendekatan Levenshtein Distance. Metode ini mengukur tingkat kemiripan karakter antar string sehingga sistem mampu mendeteksi kesalahan ejaan secara otomatis (Mahesh dkk., 2025).

Dalam penelitian ini, RafidFuzz digunakan untuk meningkatkan toleransi sistem pencarian terhadap kesalahan pengetikan, sehingga sistem tetap dapat menampilkan hasil pencarian yang relevan meskipun input pengguna tidak tepat secara ejaan.

### 3.9 Content-Based Filtering

*Content-Based Filtering* merupakan pendekatan sistem rekomendasi yang menghasilkan saran item berdasarkan kesamaan karakteristik konten antar item. Pendekatan ini tidak bergantung pada data pengguna lain, melainkan pada riwayat interaksi pengguna sendiri (Humairo dkk., 2023).

Pada penelitian ini, *Content-Based Filtering* digunakan untuk menghasilkan rekomendasi koleksi perpustakaan berdasarkan kesamaan kategori dan konten dari riwayat pencarian pengguna. Tingkat kemiripan dihitung menggunakan Cosine Similarity untuk menentukan relevansi rekomendasi.

### 3.10 Prototyping

Metode Prototyping merupakan pendekatan pengembangan perangkat lunak yang menekankan pada pembuatan model awal sistem dan evaluasi berulang berdasarkan umpan balik pengguna (Pressman & Maxim, 2019). Pendekatan ini digunakan ketika kebutuhan pengguna belum terdefinisi secara rinci.

Dalam penelitian ini, metode Prototyping digunakan karena sistem pencarian berbasis semantik dan rekomendasi memerlukan penyesuaian bertahap agar sesuai dengan kebutuhan pengguna. Setiap iterasi dilakukan untuk menyempurnakan fungsi dan tampilan sistem.

#### 1. Komunikasi Awal (*Communication*)

Tahap pertama dalam proses Prototyping melibatkan komunikasi aktif antara pengembang dan pengguna untuk mengidentifikasi kebutuhan umum dan batasan sistem. Penekanan utama adalah memahami masalah inti

---

dan tujuan pengguna, bukan langsung mendalami aspek teknis.

## 2. Perencanaan dan Desain Cepat (*Quick Plan and Quick Modeling*)

Setelah kebutuhan awal diperoleh, dilakukan perencanaan ringan untuk menentukan ruang lingkup dari *prototype* yang akan dibangun. Kemudian dibuat desain awal (*quick design*) yang merepresentasikan elemen-elemen antarmuka yang akan terlihat oleh pengguna, seperti tata letak, aliran kerja, dan tampilan keluaran.

## 3. Pembangunan Prototipe (*Construction of Prototype*)

Berdasarkan desain cepat, *prototype* awal dibangun dengan alat bantu atau komponen yang memungkinkan pengembangan cepat, seperti *library front-end*, *framework*, atau pemanfaatan komponen yang sudah ada secara berulang. Tujuan utama pada tahap ini bukan pada kualitas teknis akhir, melainkan membangun fungsi dasar dan tampilan antarmuka secepat mungkin.

## 4. Penyebaran, Pengiriman, dan Umpaman Balik (*Deployment, Delivery, and Feedback*)

Prototipe yang telah selesai diuji coba langsung oleh pengguna. Mereka memberikan umpan balik terkait apakah sistem sesuai ekspektasi, bagaimana tampilan dan fungsinya, serta apa yang masih perlu ditambah atau diubah. Evaluasi ini menjadi dasar untuk iterasi berikutnya.

## 5. Penyempurnaan Prototipe (*Refinement*)

Berdasarkan masukan pengguna, prototipe diperbaiki melalui iterasi berulang hingga semua kebutuhan dan ekspektasi pengguna terpenuhi. Proses ini bisa dilakukan berkali-kali sampai sistem mencapai bentuk akhir yang diinginkan. dilakukan hingga kebutuhan pengguna benar-benar terpenuhi dan sistem mencapai bentuk akhir yang lebih stabil dan realistik.

### 3.11 White Box

*White Box Testing* merupakan metode pengujian perangkat lunak yang berfokus pada pemeriksaan struktur internal dan alur logika program. Pengujian ini dilakukan dengan memahami kode sumber untuk memastikan setiap proses berjalan sesuai rancangan (Ndaumanu, 2023).

Pada penelitian ini, *White Box Testing* digunakan untuk menguji logika pemrosesan pencarian semantik, penanganan kesalahan penulisan, serta sistem rekomendasi agar sesuai dengan algoritma yang dirancang.

### 3.12 Black Box

*Black Box Testing* merupakan metode pengujian fungsional yang menilai kesesuaian antara input dan output sistem tanpa memperhatikan struktur internal kode program. Pengujian ini dilakukan dari sudut pandang pengguna (Nurfauziah & Jamaliyah, 2022).

Pengujian *Black Box* pada penelitian ini dilakukan untuk memastikan seluruh fitur sistem OPAC, termasuk

pencarian, filter, dan rekomendasi, dapat berjalan dengan baik sesuai skenario penggunaan.

### 3.13 Mean Analysis

*Mean Analysis* merupakan metode analisis statistik deskriptif yang digunakan untuk menghitung nilai rata-rata dari data hasil evaluasi, khususnya kuesioner skala Likert. Nilai *mean* digunakan untuk menggambarkan tingkat persepsi dan penerimaan pengguna terhadap sistem (Nyoman Sri Ardiyanti dkk., 2022).

Dalam penelitian ini, *Mean Analysis* digunakan untuk menganalisis hasil kuesioner pengguna terhadap sistem pencarian berbasis semantik guna menilai tingkat kelayakan dan penerimaan sistem secara keseluruhan.

## 4. PEMBAHASAN

Pengembangan sistem dilakukan menggunakan metode Prototyping yang mencakup tahapan komunikasi awal, perancangan dan desain cepat, pembangunan prototipe, penyebaran dan pengujian prototipe, pengumpulan umpan balik pengguna, serta penyempurnaan prototipe.

Setiap tahapan dilakukan secara berurutan dan iteratif untuk memastikan sistem yang dikembangkan sesuai dengan kebutuhan pengguna dan mampu mengatasi permasalahan pada sistem OPAC yang berjalan.

### 4.1 Komunikasi Awal (*Communication*)

#### 4.1.1 Analisis Permasalahan

Berdasarkan hasil observasi, sistem pencarian OPAC masih bergantung pada pencocokan kata kunci secara *literal*, di mana hasil pencarian hanya ditampilkan jika kata kunci memiliki kesesuaian teks dengan data koleksi. Proses ini tidak mempertimbangkan variasi istilah, sinonim, maupun konteks makna, sehingga relevansi hasil pencarian menjadi terbatas dan pengguna harus menyesuaikan kata kunci secara berulang untuk memperoleh hasil yang sesuai.

Selain itu, sistem belum mampu merespons kesalahan penulisan kata kunci, baik berupa kesalahan ejaan maupun variasi penulisan istilah. Perbedaan teks pada kata kunci menyebabkan sistem tidak menampilkan hasil pencarian meskipun koleksi yang relevan tersedia dalam basis data. Hal ini menunjukkan bahwa sistem belum memiliki mekanisme penanganan kesalahan penulisan (*typo handling*) maupun fitur saran perbaikan kata kunci, sehingga proses pencarian menjadi tidak toleran terhadap variasi *input* pengguna.

Permasalahan lainnya muncul ketika pengguna memasukkan kata kunci yang tidak identik secara teks dengan data koleksi, tetapi memiliki keterkaitan makna. Dalam kondisi tersebut, sistem tidak mampu menampilkan hasil pencarian yang relevan karena belum memiliki kemampuan untuk memahami hubungan semantik antar kata. Akibatnya, konteks pencarian pengguna tidak dapat diinterpretasikan secara tepat meskipun koleksi yang berkaitan secara konsep tersedia.

Selain keterbatasan pada mekanisme pencarian, sistem OPAC juga belum menyediakan fitur pendukung berupa rekomendasi atau alternatif koleksi berdasarkan kemiripan konten dan riwayat pencarian pengguna. Ketiadaan fitur ini menyebabkan pengguna hanya bergantung pada hasil pencarian utama, sehingga mengurangi efektivitas penelusuran informasi serta berdampak pada pengalaman pengguna dalam menemukan koleksi yang sesuai dengan kebutuhan informasi.

Berdasarkan analisis permasalahan tersebut, dapat disimpulkan bahwa sistem pencarian OPAC yang berjalan masih memiliki keterbatasan dalam memahami variasi *input* pengguna, baik dari sisi kesalahan penulisan, perbedaan istilah, maupun keterkaitan makna antar kata. Selain itu, ketiadaan fitur rekomendasi koleksi turut membatasi efektivitas proses pencarian informasi. Oleh karena itu, diperlukan pengembangan sistem pencarian yang lebih adaptif, mampu memahami makna pencarian, toleran terhadap kesalahan penulisan, serta didukung oleh mekanisme rekomendasi untuk meningkatkan relevansi hasil pencarian dan kualitas pengalaman pengguna.

#### 4.1.2 Analisis Kebutuhan

Berdasarkan permasalahan yang ditemukan, dirumuskan kebutuhan sistem sebagai dasar pengembangan solusi untuk meningkatkan pengalaman pengguna dalam melakukan pencarian informasi. Perumusan ini bertujuan mengatasi keterbatasan pencarian informasi dan menyediakan informasi pendukung untuk meningkatkan efektivitas terhadap pengguna.

1. Kebutuhan fungsional pencarian informasi, yaitu kemampuan sistem dalam memahami makna kata kunci pencarian sehingga hasil yang ditampilkan tidak hanya bergantung pada kesesuaian teks secara *literal*, tetapi juga mempertimbangkan kesamaan konteks dan topik koleksi.
2. Kebutuhan fitur pendukung pencarian, berupa mekanisme penanganan kesalahan penulisan kata kunci (*typo handling*) agar sistem tetap dapat menampilkan hasil yang relevan meskipun pengguna melakukan kesalahan ejaan atau variasi penulisan istilah. Selain itu, diperlukan fitur saran pencarian (*suggestion/autocomplete*) untuk membantu pengguna dalam menentukan kata kunci yang lebih sesuai.
3. Kebutuhan sistem rekomendasi koleksi, yaitu kemampuan sistem dalam menyajikan alternatif koleksi yang relevan berdasarkan riwayat pencarian dan kemiripan konten koleksi. Sistem rekomendasi ini dirancang menggunakan pendekatan *content-based filtering* untuk mendukung proses penelusuran informasi yang lebih efektif.

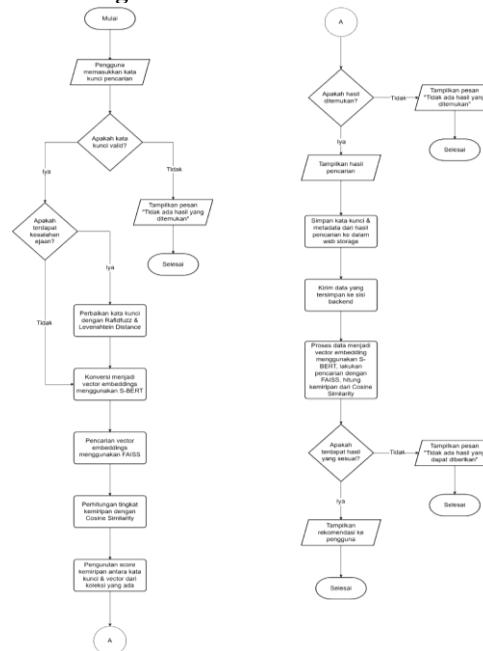
#### 4.2 Perancangan Cepat dan Desain Awal (*Planning and Quick Design*)

#### 4.2.1 Perumusan Struktur Fungsionalitas Sistem

Sistem menyediakan fungsi pencarian koleksi berbasis kemiripan makna, sehingga hasil pencarian tidak hanya bergantung pada pencocokan teks secara *literal*, tetapi juga mempertimbangkan konteks dan kesamaan topik koleksi. Untuk memastikan proses pencarian tetap berjalan meskipun terjadi ketidaktepatan *input*, sistem dilengkapi dengan fungsi penanganan kesalahan penulisan kata kunci, baik berupa kesalahan ejaan maupun variasi istilah. Selain itu, sistem menyediakan dukungan interaksi pengguna melalui penyajian hasil pencarian secara ringkas dan terstruktur serta akses detail koleksi yang memudahkan pengguna dalam menelusuri informasi. Sebagai fungsi pendukung, sistem juga menyediakan rekomendasi koleksi berdasarkan riwayat dan pola pencarian pengguna untuk membantu menemukan alternatif koleksi yang relevan.

Perumusan struktur fungsionalitas ini memastikan bahwa sistem pencarian OPAC memiliki fungsi utama yang saling terintegrasi, mencakup pencarian berbasis makna, toleransi terhadap kesalahan penulisan, dukungan interaksi pengguna, dan penyajian rekomendasi koleksi. Dengan struktur tersebut, sistem diharapkan mampu meningkatkan relevansi hasil pencarian serta efektivitas penelusuran informasi, sehingga pengalaman pengguna dalam menemukan koleksi menjadi lebih optimal.

#### 4.2.2 Perancangan Alur Proses Pencarian



Gambar 1. Diagram Alur Proses Pencarian Sistem

Figure 1. System Search Process Flowchart

Alur kerja sistem pencarian OPAC dimulai saat pengguna memasukkan kata kunci melalui antarmuka sistem. Sistem melakukan validasi awal terhadap input, di mana input kosong tetap menampilkan seluruh koleksi, sedangkan input berupa karakter acak yang tidak memiliki kemiripan dengan metadata koleksi akan langsung menghasilkan pesan bahwa data tidak

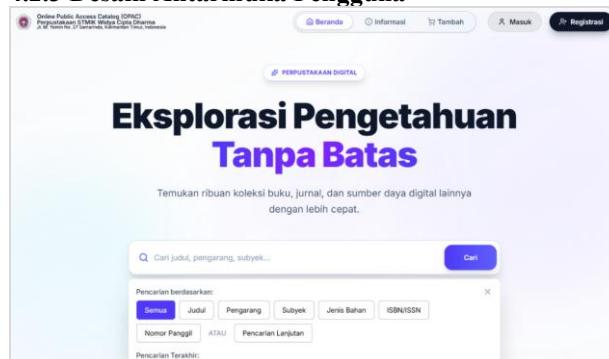
ditemukan. Kata kunci dinyatakan valid apabila memiliki kemiripan awal dengan metadata sehingga dapat diproses lebih lanjut untuk menjaga efisiensi pemrosesan dan menghindari pencarian yang tidak relevan.

Jika valid, sistem melanjutkan ke tahap pemeriksaan kesalahan penulisan dengan membandingkan kata kunci terhadap metadata koleksi sesuai jenis filter menggunakan RapidFuzz dengan algoritma Levenshtein Distance. Perbaikan kata kunci hanya dilakukan apabila nilai kemiripan berada di atas threshold koreksi typo, misalnya  $\geq 70\%$ , seperti kata "algoritmap" yang diperbaiki menjadi "algoritma". Kata kunci yang tidak memenuhi ambang batas koreksi tetap diproses sebagai input asli untuk mencegah kesalahan interpretasi.

Selanjutnya, kata kunci dikonversi menjadi vektor embedding menggunakan model S-BERT multilingual paraphrase-multilingual-MiniLM-L12-v2, kemudian dilakukan pencarian kemiripan menggunakan FAISS. Tingkat kemiripan dihitung menggunakan Cosine Similarity dan hasil diurutkan berdasarkan skor tertinggi.

Hasil pencarian divalidasi menggunakan threshold relevansi, misalnya  $\geq 0,30$ . Apabila tidak ditemukan hasil yang memenuhi kriteria, sistem menampilkan pesan bahwa data tidak ditemukan. Jika hasil tersedia, sistem menampilkan koleksi yang relevan kepada pengguna serta menyimpan kata kunci mentah dan metadata pencarian ke dalam Web Storage. Data tersebut kemudian dikirim ke backend untuk diproses kembali menggunakan S-BERT, FAISS, dan Cosine Similarity dengan threshold rekomendasi, misalnya  $\geq 0,60$ , guna menghasilkan rekomendasi koleksi berbasis content-based filtering. Jika rekomendasi tidak tersedia, sistem menampilkan pesan informasi dan proses pencarian dinyatakan selesai.

#### 4.2.3 Desain Antarmuka Pengguna

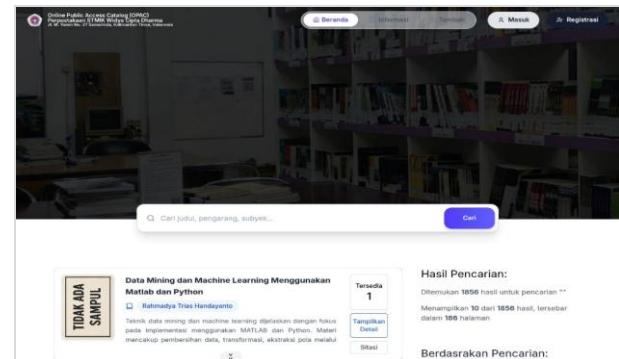


**Gambar 2. Tampilan Web Awal Dan Bar Pencarian**

*Figure 2. Home Page and Search Bar*

Tampilan kolom pencarian dirancang sebagai komponen utama antarmuka yang menjadi titik awal interaksi pengguna dengan sistem pencarian OPAC. Kolom ini memungkinkan pengguna memasukkan kata kunci pencarian secara langsung melalui tampilan yang sederhana dan mudah diakses. Pada tahap perancangan, disertakan pula fitur filter pencarian untuk mempersempit hasil berdasarkan kriteria tertentu serta

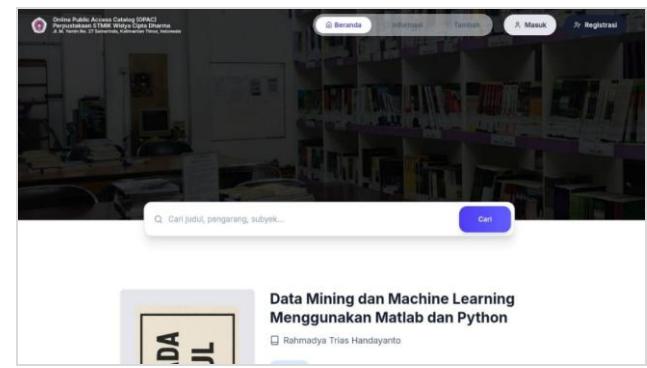
daftar pencarian terakhir yang memudahkan pengguna mengakses kembali kata kunci yang pernah digunakan.



**Gambar 3. Tampilan Koleksi Dari Hasil Pencarian**

*Figure 3. Display Of Collection From Search Results*

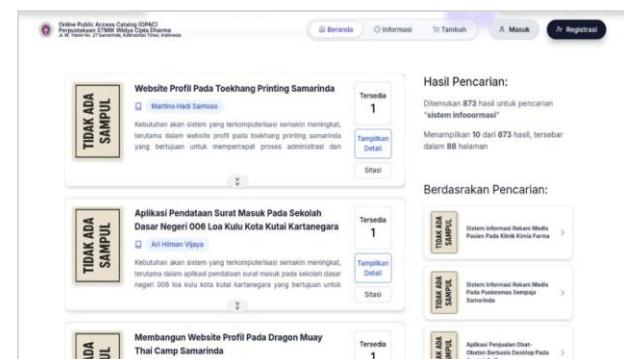
Tampilan daftar hasil pencarian dirancang untuk menampilkan koleksi secara terstruktur dengan informasi ringkas, seperti judul dan atribut pendukung, sehingga memudahkan pengguna membandingkan dan memilih koleksi yang relevan serta meningkatkan relevansi hasil pencarian dibandingkan sistem sebelumnya yang hanya mengandalkan pencocokan kata kunci *literal*.



**Gambar 4. Tampilan Detail Koleksi Yang Dipilih**

*Figure 4. Selected Collection Details Display*

Tampilan detail koleksi dirancang untuk menyajikan informasi koleksi secara lengkap dan terstruktur, sehingga pengguna dapat memperoleh pemahaman yang lebih menyeluruh dan menilai kesesuaian koleksi dengan kebutuhan informasinya secara lebih akurat.

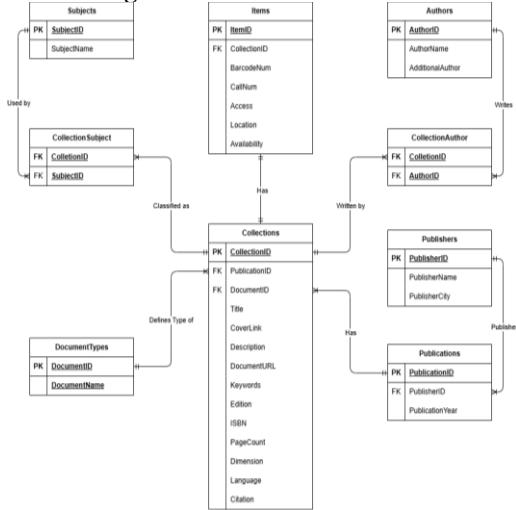


**Gambar 5. Tampilan Rekomendasi**

*Figure 5. Recommendations Display*

Tampilan rekomendasi koleksi dirancang untuk menyajikan alternatif koleksi yang relevan berdasarkan kemiripan konten dan riwayat pencarian pengguna, sehingga pengguna dapat melakukan eksplorasi lanjutan tanpa memasukkan kata kunci baru dan melengkapi proses pencarian yang sebelumnya belum didukung fitur rekomendasi.

#### 4.2.4 Perancangan Database



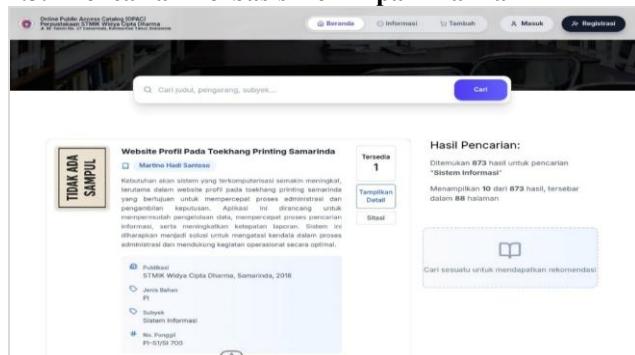
Gambar 6. Entity Relational Database wicida\_opac

Figure 6. Entity Relational Database wicida\_opac

Perancangan basis data sistem OPAC menghasilkan struktur relasional yang terintegrasi, dengan entitas Collections sebagai pusat yang terhubung dengan entitas pendukung seperti DocumentTypes, Publications, Publishers, dan Items, sehingga pengelolaan data koleksi dapat dilakukan secara konsisten dan terstruktur tanpa duplikasi. Selain relasi *one-to-many*, diterapkan juga relasi *many-to-many* antara Collections dengan Authors dan Subjects melalui tabel penghubung, yang memungkinkan satu koleksi memiliki lebih dari satu penulis dan subjek. Struktur ini memberikan fleksibilitas dalam pengelolaan *metadata* serta mendukung proses pencarian, pengelompokan, dan rekomendasi koleksi secara lebih efisien.

### 4.3 Pembangunan Prototipe (*Construction Prototype*)

#### 4.3.1 Pencarian Berbasis Kemiripan Makna

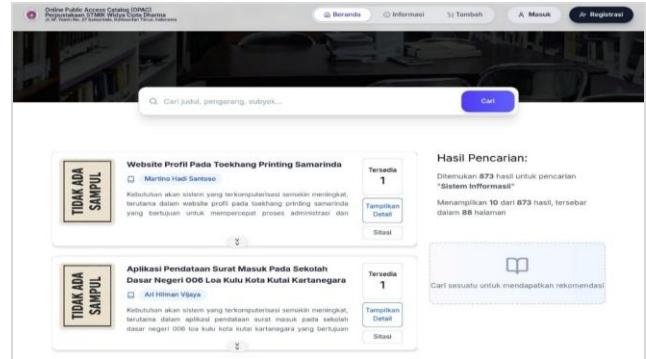


Gambar 7. Pencarian Dengan Kata Kunci Benar

Figure 7. Searching With The Right Keywords

Setelah kata kunci dimasukkan, sistem melakukan validasi dan mengubahnya menjadi representasi vektor untuk menangkap makna pencarian, kemudian menghitung tingkat kemiripan dengan koleksi menggunakan metode cosine similarity guna mengurutkan hasil berdasarkan relevansi tertinggi. Hasil pencarian menunjukkan bahwa sistem mampu menampilkan koleksi yang relevan meskipun kata kunci tidak identik secara tekstual, sehingga pencarian menjadi lebih luas dan efektif dalam membantu pengguna menemukan informasi relevan.

#### 4.3.2 Kesalahan Penulisan Kata Kunci

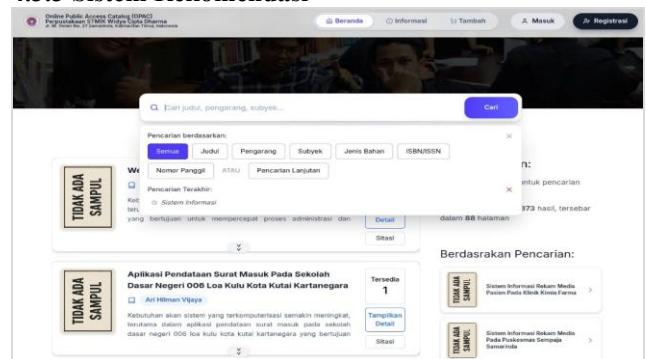


Gambar 8. Pencarian Dengan Kata Kunci Salah

Figure 8. Search With The Wrong Keywords

Prototipe sistem OPAC dilengkapi dengan mekanisme *typo handling* untuk menangani kesalahan penulisan kata kunci pada proses pencarian. Ketika pengguna memasukkan kata kunci yang tepat, sistem dapat menampilkan hasil pencarian secara normal sesuai dengan data koleksi yang tersedia. Mekanisme ini memastikan bahwa pencarian dapat berjalan dengan optimal ketika *input* pengguna sesuai secara teknis.

#### 4.3.3 Sistem Rekomendasi

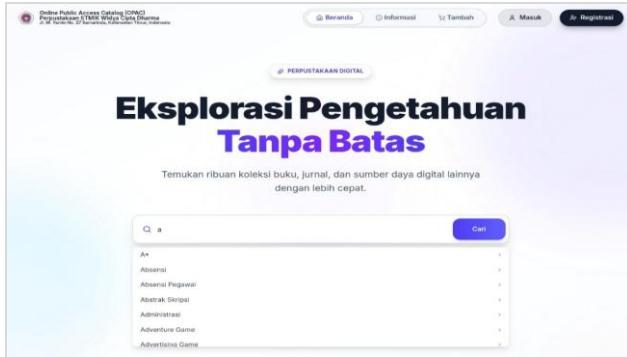


Gambar 9. Rekomendasi Berdasarkan Pencarian

Figure 9. Recommendations Based on Search

Sistem rekomendasi pada prototipe OPAC menggunakan pendekatan *content-based filtering* dengan memanfaatkan riwayat pencarian pengguna untuk menampilkan koleksi yang relevan sebagai rekomendasi “Berdasarkan Pencarian”, sehingga membantu pengguna menemukan alternatif koleksi secara lebih efisien dan meningkatkan efektivitas penelusuran informasi.

#### 4.3.4 Saran Pencarian



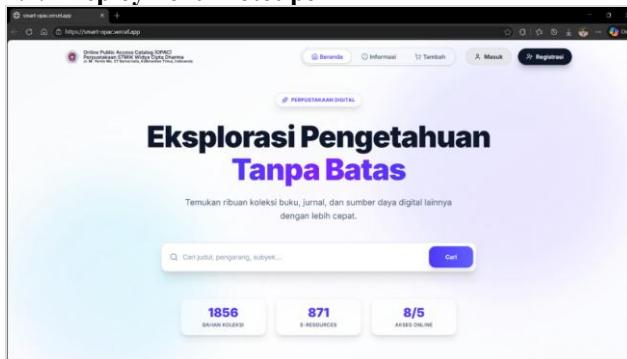
Gambar 10. Saran Pencarian Otomatis

Figure 10. Auto-Search Suggestions

Sistem dilengkapi dengan fitur *suggestion (autocomplete)* yang menampilkan saran kata kunci secara real-time saat pengguna mengetik pada kolom pencarian berdasarkan data koleksi di dalam basis data. Fitur ini membantu pengguna menentukan kata kunci yang tepat, mengurangi kesalahan penulisan, dan mempercepat proses pencarian informasi.

#### 4.4 Pengujian dan Umpaman Balik (*Deployment, Delivery, and Feedback*)

##### 4.4.1 Deployment Prototipe



Gambar 11. Penempatan web

Figure 11. Deployment web

##### 4.4.2 Pengujian White Box

###### 1. Unit Test Semantic Search

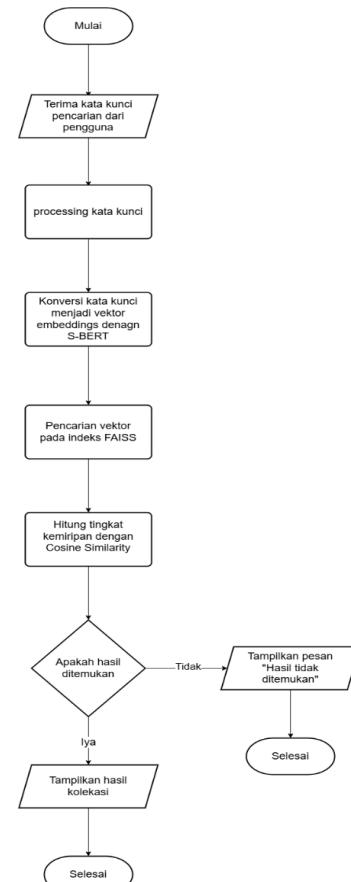
```
# tests/test_semantic_search.py
import unittest
import requests
from unittest.mock import patch, MagicMock
from app.services.ml.search.hybrid import hybrid_search
from app.services.ml.loader import get_ml_loader

class TestSemanticSearch(unittest.TestCase):
    @patch('app.services.ml.loader.get_ml_loader')
    def test_hybrid_search_returns_results(self, mock_loader):
        mock_model = MagicMock()
        mock_model.encode.return_value = np.random.rand(1, 384)
        mock_index = MagicMock()
        mock_index.search.return_value = (np.array([[0.1, 0.2]]), np.array([[0, 1]]))
        mock_index.get_index.return_value = mock_index
        mock_loader.return_value.get_index.return_value = mock_index
        mock_loader.return_value.get_metadata.return_value = [{"id": 2, "title": "Test1"}, {"id": 2, "title": "Another"}]
        results = hybrid_search("machine learning", top_k=2)
        self.assertEqual(len(results), 2)
        self.assertDictEqual(results[0], {"id": 2, "score": 0.1})

    if __name__ == '__main__':
        unittest.main()
```

Gambar 12. Pencarian Semantik Uji Unit

Figure 12. Unit Test Semantic Search



Gambar 13. Alur Semantic Search

Figure 13. Semantic Search Flow

Tabel 1. Pengujian Semantic Search

Table 1. Semantic Search Testing

Jalur Pengujian	Kondisi Uji	Hasil yang Diharapkan	Hasil Pengujian	Keterangan
1-2-3-4-5-6-7-8-10	hasil pencarian tersedia	hasil koleksi ditampilkan	sesuai	valid
1-2-3-4-5-6-7-9-10	hasil pencarian kosong	pesan tidak ditemukan	sesuai	valid

#### 2. Unit Test Typo Handling

```
# tests/test_typo_correction.py
import unittest
from app.services.typo_corrector import TypeCorrector

class TestTypoCorrection(unittest.TestCase):
    def test_blended_similarity_exact_match(self):
        self.assertAlmostEqual(typecorrector.blended_similarity("komputer", "komputer"), 1.0)

    def test_blended_similarity_type(self):
        candidates = ["komputer", "komputer programming", "laptop gaming"]
        top_k = 2
        top_kw = 1
        self.assertEqual(typecorrector.blended_similarity("komputer", "komputer"), 1.0)
        self.assertGreaterEqual(score, 0.9) # Harus cukup tinggi

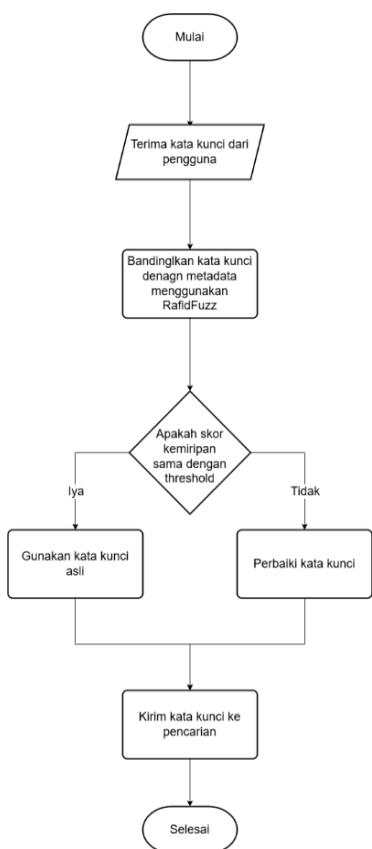
    def test_find_best_match(self):
        candidates = ["komputer", "laptop", "mouse"]
        top_k = 2
        top_kw = 1
        self.assertEqual(typecorrector.find_best_match("laptop", candidates), "laptop")
        self.assertEqual(typecorrector.find_best_match("mouse", candidates), "mouse")

    def test_find_top_n_matches(self):
        # Cukup banyak untuk lebih dekat agar skor di atas threshold
        candidates = ["komputer", "komputer programming", "laptop gaming"]
        top_k = 2
        top_kw = 1
        self.assertEqual(typecorrector.find_top_n_matches("komputer", candidates, threshold=0.0), 2) # Turunkan threshold
        self.assertEqual(typecorrector.find_top_n_matches("laptop", candidates, threshold=0.0), 1) # Perbaiki threshold
        self.assertEqual(typecorrector.find_top_n_matches("mouse", candidates, threshold=0.0), 1)

    if __name__ == '__main__':
        unittest.main(verbosity=2)
```

Gambar 14. Pengujian unit penanganan kesalahan ketik

Figure 14. Unit test typo handling



**Gambar 15. Alur Typo Handling**  
*Figure 15. Typo Handling Flow*

**Table 2. Pengujian Typo Handling**  
*Table 2. Typo Handling Testing*

Jalur Pengujian	Kondisi Uji	Hasil yang Diharapkan	Hasil Pengujian	Keterangan
1-2-3-4-5-6-8-9	kata kunci benar	kata kunci tidak diubah	sesuai	valid
1-2-3-4-5-7-8-9	kata kunci typo	kata kunci diperbaiki	sesuai	valid

### 3. Unit Test Rekomendasi Koleksi

```

def get_recommendations(self, query, top_k=5):
    # Jika query kosong, hentikan proses
    if not query:
        return []

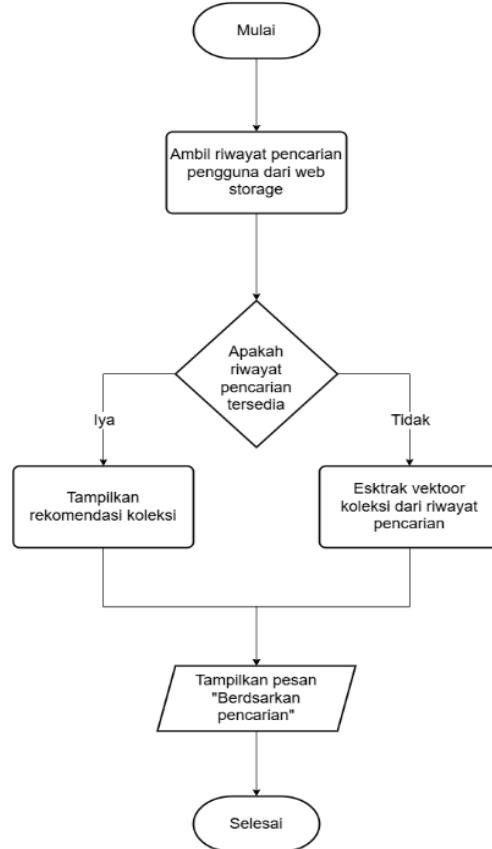
    # Pencarian gabungan semantic dan fuzzy
    results = hybrid_search(query, top_k)

    recommendations = []
    for item in results:
        # Validasi ID koleksi
        if 'collection_id' in item:
            recommendations.append({
                'collection_id': item['collection_id'],
                'score': item.get('relevance_score', item.get('score', 0))
            })

    # Mengurutkan hasil berdasarkan skor tertinggi
    recommendations.sort(key=lambda x: x['score'], reverse=True)

```

**Gambar 16. Rekomendasi Koleksi Uji Unit**  
*Figure 15. Unit Test Collection recommendation*



**Gambar 17. Alur Rekomendasi Koleksi**  
*Figure 16. Collection Recommendation Flow*

**Tabel 3. Pengujian Rekomendasi Koleksi**  
*Table 3. Collection Recommendation Testing*

Jalur Pengujian	Kondisi Uji	Hasil yang Diharapkan	Hasil Pengujian	Keterangan
1-2-3-4-7	riwayat tersedia	rekomendasi ditampilkan	sesuai	valid
1-2-3-5-6-7	riwayat tidak tersedia	pesan informasi ditampilkan	sesuai	valid

### 4. Unit Test Suggestion

```

import unittest
from unittest.mock import patch, Mock
from typing import List, int

class AutocompleteSuggestionService:
    """Service untuk memberikan suggestion/autocomplete"""
    def __init__(self, db: Mock, vocab_cache: Mock):
        self.db = db
        self.vocab_cache = vocab_cache

    def get_subjects(self, prefix: str, limit: int = 5) -> List[str]:
        """Dapatkan daftar subjek berdasarkan prefix"""
        if not prefix or len(prefix) < 2:
            return []

        subjects = self.vocab_cache.load_subjects(prefix)
        return subjects[:limit]

    def suggest(self, prefix: str):
        """Test suggestion untuk subject field"""
        def setup(self):
            self.subjects = Mock()
            self.subjects.load.return_value = [
                'Python', 'Java', 'C', 'JavaScript', 'React'
            ]
            self.db.get_subjects.return_value = self.subjects
            self.vocab_cache.load_subjects.return_value = self.subjects

        def test_suggest(self):
            self.suggest(prefix)
            self.subjects.assert_called_with(prefix)
            self.db.get_subjects.assert_called_once()
            self.vocab_cache.load_subjects.assert_called_once()

        setup(self)
        test_suggest(self)

    class TestSubjectSuggestion(unittest.TestCase):
        """Test suggestion untuk subject field"""
        def test_suggest(self):
            self.subjects = Mock()
            self.subjects.load.return_value = [
                'Python', 'Java', 'C', 'JavaScript', 'React'
            ]
            self.db.get_subjects.return_value = self.subjects
            self.vocab_cache.load_subjects.return_value = self.subjects

            self.suggest('Py')
            self.subjects.assert_called_with('Py')
            self.db.get_subjects.assert_called_once()
            self.vocab_cache.load_subjects.assert_called_once()

            self.assertEqual(self.subjects.load.call_args[0][0], 'Py')

        def test_suggest_case_insensitive(self):
            self.subjects = Mock()
            self.subjects.load.return_value = [
                'Python', 'Java', 'C', 'JavaScript', 'React'
            ]
            self.db.get_subjects.return_value = self.subjects
            self.vocab_cache.load_subjects.return_value = self.subjects

            self.suggest('py')
            self.subjects.assert_called_with('py')
            self.db.get_subjects.assert_called_once()
            self.vocab_cache.load_subjects.assert_called_once()

            self.assertEqual(self.subjects.load.call_args[0][0], 'Py')

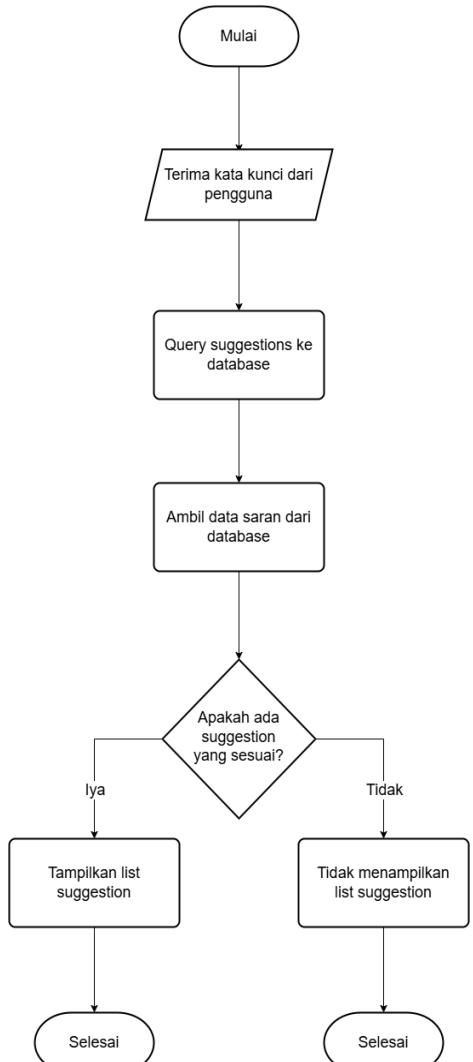
        def test_suggest_no_results(self):
            self.subjects = Mock()
            self.subjects.load.return_value = []
            self.db.get_subjects.return_value = self.subjects
            self.vocab_cache.load_subjects.return_value = self.subjects

            self.suggest('xyz')
            self.subjects.assert_called_with('xyz')
            self.db.get_subjects.assert_called_once()
            self.vocab_cache.load_subjects.assert_called_once()

            self.assertEqual(self.subjects.load.call_args[0][0], 'xyz')

```

**Gambar 18. Saran Uji Unit**  
*Figure 17. Unit Test Suggestion*



**Gambar 19. Alur Suggestion**  
Figure 18. Suggestion Flow

**Tabel 4. Pengujian Suggestion**  
Table 4. Suggestion Testing

Jalur Pengujian	Kondisi Uji	Hasil yang Diharapkan	Hasil Pengujian	Keterangan
1-2-3-4-5-6-8	suggestion tersedia	list suggestion tampil	sesuai	valid
1-2-3-4-5-7-8	suggestion kosong	suggestion tidak tampil	sesuai	valid

#### 4.4.3 Pengujian Black Box

Pengujian *black box* dilakukan untuk mengevaluasi fungsionalitas sistem OPAC dari sudut pandang pengguna tanpa memperhatikan struktur internal sistem. Pengujian ini mencakup berbagai skenario penggunaan yang merepresentasikan interaksi nyata pengguna dengan sistem, meliputi pencarian menggunakan kata kunci valid, pencarian dengan kesalahan penulisan kata kunci, pemanfaatan fitur *suggestion*, penggunaan filter pencarian, navigasi hasil pencarian, serta penampilan detail dan rekomendasi koleksi.

Pengujian *Black Box* dilakukan untuk memastikan bahwa seluruh fungsi utama pada sistem OPAC dapat berjalan sesuai dengan kebutuhan fungsional yang telah ditetapkan. Pengujian ini berfokus pada kesesuaian antara *input* yang diberikan oleh pengguna dan *output* yang dihasilkan oleh sistem, tanpa memperhatikan struktur internal atau kode program yang digunakan.

Pengujian pencarian koleksi dilakukan dengan memberikan kata kunci yang valid dan sesuai dengan data yang tersimpan di dalam sistem. Hasil pengujian menunjukkan bahwa sistem mampu menampilkan daftar koleksi yang relevan sesuai dengan kata kunci yang dimasukkan. Selain itu, pengujian juga dilakukan menggunakan kata kunci yang tidak identik secara teks namun memiliki kesamaan makna. Pada skenario ini, sistem tetap mampu menampilkan koleksi yang relevan secara semantik, sehingga menunjukkan bahwa mekanisme pencarian berbasis kemiripan makna telah berjalan dengan baik.

Selanjutnya, pengujian fitur *typo handling* dilakukan dengan memberikan kata kunci yang mengandung kesalahan penulisan. Hasil pengujian menunjukkan bahwa sistem tetap dapat menampilkan hasil pencarian yang relevan meskipun terdapat kesalahan ejaan pada kata kunci, yang menandakan bahwa mekanisme toleransi terhadap kesalahan pengetikan telah berfungsi dengan baik.

Pengujian juga dilakukan pada fitur *suggestion (autocomplete)* dengan memberikan sebagian karakter awal dari kata kunci. Sistem mampu menampilkan daftar saran kata kunci yang sesuai, sehingga memudahkan pengguna dalam melakukan pencarian. Selain itu, fitur filter pencarian diuji dengan menerapkan filter berdasarkan judul, pengarang, dan subjek. Hasil pengujian menunjukkan bahwa sistem dapat menampilkan hasil pencarian sesuai dengan filter yang dipilih oleh pengguna.

Pada aspek navigasi hasil pencarian, pengujian dilakukan dengan berpindah antar halaman hasil pencarian. Sistem mampu menampilkan halaman hasil selanjutnya sesuai dengan perintah pengguna. Pengujian pada fitur detail koleksi juga menunjukkan bahwa sistem dapat menampilkan informasi detail koleksi dengan benar ketika pengguna memilih salah satu hasil pencarian.

Terakhir, pengujian dilakukan pada fitur rekomendasi koleksi dengan memanfaatkan riwayat pencarian pengguna. Hasil pengujian menunjukkan bahwa sistem mampu menampilkan rekomendasi koleksi yang relevan berdasarkan riwayat pencarian yang tersedia.

Berdasarkan keseluruhan hasil pengujian *Black Box* yang telah dilakukan, dapat disimpulkan bahwa seluruh fitur utama sistem OPAC berjalan sesuai dengan skenario pengujian dan menghasilkan *output* yang sesuai dengan yang diharapkan. Dengan demikian, sistem telah memenuhi kebutuhan fungsional yang ditetapkan pada tahap analisis kebutuhan dan dinyatakan layak untuk digunakan.

#### 4.4.4 Uji Produk dan Kuesioner Pengguna

Uji produk dilakukan untuk mengetahui tanggapan pengguna terhadap prototipe sistem OPAC yang telah dikembangkan. Pengujian dilakukan dengan melibatkan pengguna yang mencoba langsung sistem berdasarkan skenario penggunaan yang telah ditentukan. Setelah melakukan uji coba, pengguna diminta mengisi kuesioner untuk memberikan penilaian terhadap kemudahan penggunaan, relevansi hasil pencarian, serta manfaat fitur rekomendasi koleksi.

Kuesioner disusun menggunakan skala Likert lima tingkat dengan rentang nilai 1 sampai 5, di mana nilai 1 menunjukkan "Sangat Tidak Setuju" dan nilai 5 menunjukkan "Sangat Setuju". Kuesioner mencakup beberapa aspek penilaian, yaitu relevansi hasil pencarian, penanganan kesalahan penulisan kata kunci, kemudahan penggunaan sistem, serta manfaat sistem rekomendasi.

**Tabel 5. Kategori Penilaian Mean Analysis**

*Table 5. Mean Analysis Assessment Category*

Rata-rata	Kategori
4,21 – 5,00	Sangat Baik
3,41 – 4,20	Baik
2,61 – 3,40	Cukup
1,81 – 2,60	Kurang
1,00 – 1,80	Sangat Kurang

Pengambilan kesimpulan dari hasil kuesioner dilakukan menggunakan pendekatan kuantitatif deskriptif dengan metode analisis skor rata-rata. Data kuesioner yang diperoleh dari responden dikonversi ke dalam bentuk numerik menggunakan skala Likert lima tingkat, kemudian dihitung nilai rata-ratanya untuk setiap aspek penilaian. Hasil perhitungan tersebut digunakan sebagai dasar dalam menentukan tingkat penerimaan dan evaluasi terhadap prototipe sistem OPAC yang dikembangkan. Pendekatan ini memungkinkan penilaian dilakukan secara objektif berdasarkan persepsi pengguna terhadap fitur dan kinerja sistem.

**Tabel 6. Rekapitulasi Hasil Kuesioner Pengguna**

*Table 6. Summary of User Questionnaire Results*

Aspek Penilaian	Rata-rata	Kategori
Relevansi hasil pencarian	4,0	Baik
Penanganan kesalahan penulisan	4,4	Sangat Baik
Kemudahan penggunaan sistem	4,2	Baik
Manfaat sistem rekomendasi	4,1	Baik
Kelayakan sistem secara keseluruhan	4,4	Sangat Baik

Berdasarkan hasil rekapitulasi kuesioner, seluruh aspek penilaian memperoleh nilai rata-rata pada kategori "Baik" hingga "Sangat Baik". Aspek penanganan kesalahan penulisan kata kunci dan kelayakan sistem memperoleh nilai tertinggi, yang menunjukkan bahwa sistem mampu mengatasi permasalahan utama pada

sistem OPAC sebelumnya. Selain itu, hasil penilaian terhadap relevansi hasil pencarian dan sistem rekomendasi menunjukkan bahwa fitur pencarian berbasis kemiripan makna memberikan manfaat nyata bagi pengguna dalam menemukan koleksi yang sesuai.

**Tabel 7. Hasil Kuesioner Dari Responden**

*Table 7. Questionnaire Results From Respondents*

Pertanyaan	R <sub>1</sub>	R <sub>2</sub>	R <sub>3</sub>	R <sub>4</sub>	R <sub>5</sub>	R <sub>6</sub>	R <sub>7</sub>	R <sub>8</sub>	R <sub>9</sub>	R <sub>10</sub>
P1	4	4	5	4	4	4	3	5	3	4
P2	4	5	5	5	2	4	3	5	3	4
P3	4	4	5	5	4	4	3	3	4	3
P4	4	5	5	4	5	5	3	4	4	4
P5	4	4	5	4	5	5	3	4	5	5
P6	4	5	5	4	5	4	3	4	5	3
P7	4	5	5	4	5	5	3	4	3	4
P8	4	5	5	4	5	5	3	3	4	4
P9	4	4	5	4	5	5	3	5	5	3
P10	4	5	5	4	5	4	3	4	4	4
P11	4	5	5	4	5	3	3	4	4	3
P12	4	5	5	4	5	4	3	4	4	3
P13	4	5	5	4	4	4	3	3	4	5
P14	4	4	5	5	5	5	3	3	5	5
P15	4	5	5	5	5	5	3	3	5	4
Rata-rata	4,00	4,67	5,00	3,33	5,53	4,40	3,00	3,87	4,13	3,87

Hasil uji produk dan kuesioner pengguna menunjukkan bahwa prototipe sistem OPAC yang dikembangkan dapat diterima dengan baik oleh pengguna. Sistem dinilai mampu meningkatkan efektivitas pencarian koleksi, menangani kesalahan *input* pengguna, serta menyediakan rekomendasi koleksi yang relevan.

#### 4.5 Penyempurnaan Prototipe (*Refinement*)

Setelah prototipe sistem dievaluasi melalui pengujian fungsional dan kuesioner pengguna pada tahap sebelumnya. Penyempurnaan dilakukan untuk menindaklanjuti temuan evaluasi yang menunjukkan bahwa meskipun sistem telah berjalan dengan baik, masih terdapat beberapa aspek yang dapat ditingkatkan agar sistem lebih optimal dan sesuai dengan kebutuhan pengguna.

##### 4.5.1 Identifikasi Hasil Evaluasi

Berdasarkan hasil pengujian *white box*, seluruh modul utama sistem, meliputi mekanisme pencarian berbasis kemiripan makna, penanganan kesalahan penulisan kata kunci, penyimpanan riwayat pencarian, dan sistem rekomendasi, telah berjalan sesuai dengan alur logika yang dirancang tanpa ditemukan kesalahan fungsional. Hal ini menunjukkan bahwa implementasi sistem telah memenuhi rancangan teknis yang ditetapkan. Hasil pengujian *black box* menunjukkan

---

bahwa seluruh fitur sistem berfungsi dengan baik dan dapat digunakan oleh pengguna sesuai dengan skenario pengujian, termasuk pencarian, penggunaan filter, navigasi hasil pencarian, serta penampilan rekomendasi koleksi. Seluruh skenario pengujian berhasil dijalankan tanpa kendala yang menghambat penggunaan sistem.

Selain itu, hasil kuesioner pengguna menunjukkan bahwa sistem memperoleh penilaian pada kategori Baik hingga Sangat Baik pada seluruh aspek penilaian. Penilaian tertinggi diperoleh pada aspek penanganan kesalahan penulisan kata kunci dan kelayakan sistem secara keseluruhan. Hal ini menunjukkan bahwa sistem telah mampu menjawab permasalahan utama yang terdapat pada sistem OPAC sebelumnya.

#### 4.5.2 Proses Penyempurnaan Sistem

Meskipun sistem telah berjalan dengan baik dan memenuhi kebutuhan pengguna, dilakukan beberapa penyesuaian minor sebagai bentuk penyempurnaan akhir. Penyesuaian ini tidak mengubah konsep maupun mekanisme utama sistem, melainkan bertujuan untuk meningkatkan kenyamanan dan konsistensi penggunaan. Penyempurnaan minor yang dilakukan meliputi penyesuaian tampilan hasil pencarian dan rekomendasi agar informasi lebih ringkas dan mudah dibaca, serta penyelarasan tata letak antarmuka untuk meningkatkan kejelasan visual. Penyesuaian ini dilakukan berdasarkan saran pengguna tanpa mempengaruhi kinerja inti sistem.

#### 4.5.3 Kondisi Prototipe Setelah Penyempurnaan

Setelah dilakukan penyempurnaan minor, prototipe sistem OPAC berada pada kondisi stabil dan siap digunakan. Sistem mampu menyajikan hasil pencarian yang relevan, menangani kesalahan penulisan kata kunci, serta memberikan rekomendasi koleksi yang sesuai dengan kebutuhan pengguna. Dengan demikian, prototipe sistem OPAC yang dikembangkan telah memenuhi tujuan penelitian dan dapat dijadikan sebagai dasar untuk implementasi lanjutan atau pengembangan sistem OPAC pada tahap berikutnya.

### 5. KESIMPULAN

Berdasarkan hasil perancangan, pembangunan, dan pengujian sistem pencarian OPAC yang dikembangkan, dapat disimpulkan beberapa hal sebagai berikut.

1. Sistem pencarian berbasis kemiripan makna (*semantic search*) yang diterapkan mampu meningkatkan relevansi dan fleksibilitas pencarian koleksi perpustakaan, karena proses pencarian tidak lagi bergantung pada pencocokan kata kunci secara *literal*, melainkan mempertimbangkan kesamaan makna dan konteks pencarian pengguna.
2. Penerapan model S-BERT dalam pembentukan representasi vektor kata kunci dan koleksi terbukti efektif dalam menampilkan hasil pencarian yang sesuai dengan kebutuhan informasi pengguna, khususnya pada koleksi akademik yang memiliki variasi istilah dan topik.
3. Mekanisme penanganan kesalahan penulisan kata kunci (*typo handling*) menggunakan RapidFuzz

dengan algoritma Levenshtein Distance mampu mengurangi kegagalan pencarian akibat kesalahan ejaan, sehingga sistem tetap dapat menampilkan hasil pencarian yang relevan meskipun terjadi ketidaksetepatan *input* dari pengguna.

4. Fitur saran pencarian (*suggestion/autocomplete*) yang bersumber dari data koleksi membantu pengguna dalam menentukan dan menyelesaikan penulisan kata kunci, sehingga proses pencarian menjadi lebih cepat, efisien, dan terarah.
5. Sistem rekomendasi koleksi berbasis *content-based filtering* yang memanfaatkan riwayat pencarian pengguna memberikan nilai tambah dalam proses penelusuran informasi dengan menyajikan koleksi yang relevan secara kontekstual tanpa memerlukan data dari pengguna lain.
6. Arsitektur sistem berbasis RESTful API memungkinkan pemisahan yang jelas antara antarmuka pengguna dan pemrosesan data, sehingga sistem lebih terstruktur, responsif, serta memiliki potensi untuk dikembangkan dan diintegrasikan lebih lanjut pada sistem OPAC di masa mendatang.

### 6. SARAN

Meskipun sistem yang dikembangkan telah berjalan dengan baik dan memenuhi kebutuhan pengguna, terdapat beberapa saran yang dapat dijadikan acuan untuk pengembangan selanjutnya, antara lain sebagai berikut.

1. Model S-BERT yang digunakan dalam penelitian ini masih merupakan model umum. Pada pengembangan selanjutnya, disarankan untuk melakukan *fine-tuning* menggunakan dataset koleksi perpustakaan agar representasi makna terhadap istilah akademik dan domain-spesifik menjadi lebih akurat.
2. Mekanisme penanganan kesalahan penulisan kata kunci yang saat ini menggunakan Levenshtein Distance melalui RapidFuzz dapat dikembangkan lebih lanjut dengan mengombinasikan pendekatan lain, seperti metode fonetik atau koreksi berbasis konteks, guna meningkatkan ketepatan pemahaman maksud pencarian pengguna.
3. Sistem rekomendasi yang saat ini menggunakan pendekatan *content-based filtering* dapat dikembangkan menjadi sistem rekomendasi hibrida (*hybrid recommendation system*) apabila di masa mendatang tersedia data interaksi pengguna yang lebih beragam.
4. Pengujian sistem pada penelitian ini masih dilakukan pada skala data yang terbatas. Oleh karena itu, disarankan untuk melakukan pengujian pada skala data yang lebih besar serta menerapkan optimasi performa, seperti pengaturan indeks FAISS dan manajemen memori, agar sistem tetap responsif pada beban data yang lebih tinggi.
5. Jumlah responden pada pengujian pengguna masih terbatas. Pada penelitian selanjutnya, disarankan untuk melibatkan lebih banyak pengguna agar hasil

evaluasi sistem menjadi lebih komprehensif dan representatif.

## 7. REFERENSI

- Agiharta, K. F., Suteja, B. R., & Ayub, M. (2024). Penerapan sentence-BERT untuk similaritas kompetensi pekerjaan dan mata kuliah. *Jurnal Teknik Informatika dan Sistem Informasi*, 10, 449–467.  
<https://doi.org/10.28932/jutisi.vXiX.X>
- Al'Izza, M. A., Jazuli, A., & Nurkamid, M. (2022). Implementasi teknologi semantic web untuk pencarian koleksi perpustakaan Universitas Muria Kudus. *Jurnal Dialektika Informatika (Detika)*, 2(2), 56–62.  
<https://doi.org/10.24176/detika.v2i2.7884>
- Amin, M. I. F. R., Amarharizqi, M. R., Sofi, F. A., & Sari, A. P. (2023). Sistem rekomendasi musik berdasarkan preferensi pengguna menggunakan metode natural language processing (NLP). *Prosiding Seminar Nasional Informatika Bela Negara (SANTIKA)*, 3, 129–133.  
<https://santika.upnjatim.ac.id/submissions/index.php/santika/article/view/213>
- Daniel, I., Ginting, Y. T. A., Tolle, H., & Brata, K. C. (2022). Pengembangan sistem manajemen sampah TPS3R Dinas Lingkungan Hidup Kota Batu berbasis web responsive menggunakan Tailwind dan ReactJS. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 6(12).  
<http://j-ptiik.ub.ac.id>
- Elsadantia, B. A. (2023). Perkembangan dan peran OPAC pada aplikasi CIP (Cerah Informasi Pustaka) untuk temu kembali informasi di perpustakaan Universitas Tridinanti Palembang. *Jurnal Multidisipliner Bharasumba*, 2(4), 296–315.  
<https://doi.org/10.62668/bharasumba.v2i04.809>
- Hendrayana, N., Wibowo, J. S., & Juang, J. T. L. (2024). Sistem rekomendasi pencarian buku perpustakaan dengan algoritma content-based filtering. *Jurnal Ilmiah Elektronika dan Komputer*, 17(1), 271–278.  
<https://doi.org/10.51903/elkom.v17i1.1927>
- Hidayat, A., & Prabowo, D. (2020). Implementation of virtual private server (VPS) using Digital Ocean cloud server on BMT Mentari East Lampung. *Jurnal Teknologi Komputer dan Sistem Informasi*, 3, 116–121.
- Humairo, A., Herdiani, A., & Puspitasari, S. Y. (2023). Pembangunan recommender system menggunakan content-based filtering pada aplikasi service desk. *LOGIC: Jurnal Penelitian Informatika*, 1(1), 20.  
<https://doi.org/10.25124/logic.v1i1.6427>
- Jaisy, M. A., Prasasti, A. L., & Novianty, A. (2025). Pengembangan backend untuk efisiensi pengelolaan dan penyebaran informasi di himpunan mahasiswa teknik komputer. *Jurnal Nasional Sains dan Teknik*, 2(2).  
<https://doi.org/10.25124/jnst.v2i2.8745>
- Kaniya, I. A., Paramitha, P., Wiharta, I. M. D., & Suyadnya, I. M. A. (2022). Perancangan dan implementasi RESTful API pada sistem informasi manajemen dosen Universitas Udayana. *Jurnal Teknologi Informasi*, 9(3).
- Krisnawati, L. D., Mahastama, A. W., Haw, S.-C., Ng, K.-W., & Naveen, P. (2024). Indonesian–English textual similarity detection using Universal Sentence Encoder and Facebook AI Similarity Search (FAISS). *CommIT Journal*, 18(2).  
<https://journal.binus.ac.id/index.php/commit/article/view/11274>
- Kusnadar, K., & Anshary, M. R. (2022). Rancang bangun aplikasi untuk menghitung indikator kepuasan pengunjung berbasis Java di Dinas Perpustakaan dan Kearsipan Kabupaten Berau. *Sebatik*, 26(2), 798–806.  
<https://doi.org/10.46984/sebatik.v26i2.1442>
- Mahesh, S., Shrivarshan, N. K., & Anbucchelvan, A. P. (2025). OCR and NLP based personalized allergen notifying system. *International Journal of Scientific Research in Engineering and Management*, 9(2), 1–9.  
<https://doi.org/10.5504/IJSREM41634>
- Meilina, L., Kumara, I. N. S., & Setiawan, I. N. (2021). Literature review klasifikasi data menggunakan metode cosine similarity dan artificial neural network. *Majalah Ilmiah Teknologi Elektro*, 20(2), 307.  
<https://doi.org/10.24843/mite.2021.v20i02.p15>
- Murti, H., Lestariningsih, E., & Sugiyamta. (2019). Perancangan sistem rekomendasi buku pada katalog perpustakaan menggunakan pendekatan content-based filtering dan algoritma FP-growth. Dalam *Proceeding SINTAK 2019* (hal. 532–536).  
<https://www.unisbank.ac.id/ojs/index.php/sintak/article/view/7643>
- Murtopo, A. A., Haryadi, D., & Fadilah, N. (2022). Penerapan model information retrieval untuk pencarian konten pada perpustakaan digital. *JUPTI*, 1(3), 63–70.
- Nasution, N., & Iswari, L. (2021). Penerapan ReactJS pada pengembangan frontend. *AUTOMATA*.  
<https://journal.uii.ac.id/AUTOMATA/article/view/19532>
- Ndaumanu, R. I. (2023). Pengujian sistem informasi perpustakaan berbasis website dengan basis path testing. *JUSTEK*, 6(1), 123–134.  
<https://doi.org/10.31764/justek.vXiY.ZZZ>
- Nurfauziah, H., & Jamaliyah, I. (2023). Perbandingan metode testing antara blackbox dengan whitebox pada sebuah sistem informasi. *Jurnal Visualika*, 8(2), 105–113.  
<https://jurnas.saintekmu.ac.id/index.php/visualika/article/view/24>

- 
- Nursobah, N., & Pahrudin, P. (2022). Penerapan algoritma pencarian Knuth–Morris–Pratt (KMP) dalam sistem informasi perpustakaan. *Sebatik*, 26(2), 798–806.  
<https://jurnal.wicida.ac.id/index.php/sebatik/article/view/451>
- Pressman, R. S., & Maxim, B. R. (2019). *Software engineering: A practitioner's approach*. McGraw-Hill Education.
- Qaulan, M. A., Wahyuni, & Adytia, P. (2025). Penerapan chatbot berbasis AI untuk pelayanan perpustakaan di STMIK Widya Cipta Dharma. *TEMATIK*, 12(1), 23–30.
- Rahman, S., Sembiring, A., Siregar, D., Khair, H., Prahmana, G., Puspadiini, R., & Zen, M. (2023). *Python: Dasar dan pemrograman berorientasi objek*. Tahta Media Group.
- Rantung, V. P. (2023). *Teknik-teknik pemrosesan bahasa alami (NLP)*. Lakeisha.
- Samosir, F. V. P., Toba, H., & Ayub, M. (2022). BESKlus: BERT extractive summarization with K-means clustering in scientific paper. *Jurnal Teknik Informatika dan Sistem Informasi*, 8(1).  
<https://doi.org/10.28932/jutisi.v8i1.4474>
- Santiago, J. (2025). Ciptaan disebarluaskan di bawah lisensi Creative Commons Atribusi 4.0 Internasional. *Journal of Information System, Applied, Management, Accounting and Research*, 9(3), 1219–1226.  
<https://doi.org/10.52362/jisamar.v9i3.1986>
- Sulistiyowati, I., Setyawan, D., & Arzaqi, N. Z. (2024). Sistem rekomendasi pemilihan layanan data dan internet segmen BGES dengan metode knowledge base berbasis web pada PT. Telkom Surakarta. *Journal of Information System Management (JOISM)*, 6(1).
- Umri, C., Pulungan, A. F., Halimatusyaddiah, A., Sinaga, I. E., Tarigan, L. A., Aini, N., & Mazaya, S. M. (2025). EduSearch: Web pencarian cerdas berbasis semantik untuk mencari data seluruh sekolah formal di Kota Medan. *Jurnal Minfo Polgan*, 13(2), 2699–2713.  
<https://doi.org/10.33395/jmp.v13i2.14597>

#### UCAPAN TERIMA KASIH

Alhamdulillah, terima kasih kepada Allah SWT atas rahmat dan karunia-Nya sehingga penelitian ini dapat diselesaikan. Ucapan terima kasih juga disampaikan kepada kedua orang tua atas doa, dukungan, dan motivasi yang senantiasa diberikan. Selanjutnya, penulis menyampaikan terima kasih kepada Ibu Amelia Yusnita, S.Kom., M.Kom. selaku dosen pembimbing utama dan Ibu Ita Arfyanti, S.Kom., M.M. selaku dosen pembimbing pendamping atas bimbingan dan arahan yang diberikan selama penelitian ini. Penulis juga mengucapkan terima kasih kepada teman-teman atas dukungan dan semangat yang diberikan selama proses penyusunan penelitian ini. Penulis menyadari bahwa naskah ini masih memiliki kekurangan dan kesalahan, serta diharapkan penelitian ini dapat memberikan manfaat.