

Implementasi *Content-Based Filtering* Menggunakan TF-IDF dan *Cosine Similarity* untuk Rekomendasi Buku Akademik Mahasiswa

Yunisa Salsabila Anggraeni¹, Septiana Dewi Saputri², Tania Azzahra³, Aloysius Gonzaga Verrel⁴, Vitri Tundjungsari⁵

^{1,2,3,4,5}Program Studi Teknik Informatika, Universitas Esa Unggul Harapan Indah

¹yunisasalsabila08@student.esaunggul.ac.id, ²septianadewisaputri21@student.esaunggul.ac.id, ³thaniapoetry119@student.esaunggul.ac.id,

⁴anggoroverrel@student.esaunggul.ac.id, ⁵vitri.tundjungsari@esaunggul.ac.id

Info Artikel

Riwayat Artikel:

Received 2025-12-30

Revised 2026-03-05

Accepted 2026-04-09

Corresponding Author:

Yunisa Salsabila Anggraeni

Email:

yunisasalsabila08@student.esaunggul.ac.id



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

Abstract – The abundance of digital book collections in academic libraries creates information overload, making it difficult for students to find relevant references. This study aims to design and evaluate an academic book recommendation system based on Content-Based Filtering (CBF) using TF-IDF for text feature weighting and Cosine Similarity for measuring inter-content similarity. The dataset consists of 10 simulated academic book entries from Kaggle, covering title, author, category, and description attributes across Computers, Technology & Engineering, and Mathematics domains. Methodological stages include text preprocessing, TF-IDF feature extraction, similarity matrix construction, and top-N recommendation selection. Evaluation was conducted through subjective satisfaction testing involving 10 student respondents and quantitative evaluation using Precision@K and Recall@K metrics. Satisfaction results showed recommendation relevance (87%), system speed (90%), ease of use (85%), and overall satisfaction (86%). Quantitative evaluation revealed limitations with Mean Precision@5 of 0.0921, Mean Recall@5 of 0.0772, Mean Precision@10 of 0.0815, and Mean Recall@10 of 0.1136, exhibiting a consistent precision-recall trade-off. The system is concluded to be functionally effective but semantically limited. Future development is recommended to integrate word embedding techniques such as Word2Vec or BERT alongside hybrid filtering to substantially improve system performance.

Keywords: Academic Book Recommendation; Content-Based Filtering; Cosine Similarity; Recommendation System; TF-IDF.

Abstrak – Kelembapan udara merupakan salah satu parameter yang berpengaruh terhadap lingkungan. Melimpahnya koleksi buku digital di perpustakaan akademik menimbulkan informasi overload yang menyulitkan mahasiswa menemukan referensi relevan. Penelitian ini bertujuan merancang dan mengevaluasi sistem rekomendasi buku akademik berbasis Content-Based Filtering (CBF) menggunakan TF-IDF sebagai pembobotan fitur teks dan Cosine Similarity sebagai pengukur kemiripan antarkonten. Dataset berjumlah 10 entri buku akademik simulasi dari Kaggle, mencakup atribut judul, penulis, kategori, dan deskripsi dari domain Computers, Technology & Engineering, serta Mathematics. Tahapan metode meliputi pra-pemrosesan teks, ekstraksi fitur TF-IDF, pembangunan similarity matrix, dan seleksi rekomendasi top-N. Evaluasi dilakukan melalui pengujian kepuasan subjektif terhadap 10 mahasiswa dan evaluasi kuantitatif menggunakan metrik Precision@K serta Recall@K. Hasil kepuasan menunjukkan skor relevansi rekomendasi (87%), kecepatan sistem (90%), kemudahan penggunaan (85%), dan kepuasan umum (86%). Evaluasi kuantitatif mengungkap keterbatasan dengan Mean Precision@5 sebesar 0,0921, Mean Recall@5 sebesar 0,0772, Mean Precision@10 sebesar 0,0815, dan Mean Recall@10 sebesar 0,1136, disertai pola trade-off konsisten antara presisi dan cakupan. Disimpulkan bahwa sistem efektif secara fungsional namun terbatas secara semantik. Pengembangan selanjutnya direkomendasikan mengintegrasikan word embedding seperti Word2Vec atau BERT serta hybrid filtering untuk meningkatkan performa sistem.

Kata Kunci: Content-Based Filtering, Cosine Similarity, Rekomendasi Buku Akademik, Sistem Rekomendasi, TF-IDF.

I. PENDAHULUAN

Perkembangan teknologi informasi yang berlangsung secara masif dalam satu dekade terakhir telah mengubah cara mahasiswa mengakses sumber daya akademik secara fundamental. Perpustakaan digital, repositori ilmiah daring, dan platform e-learning kini menyediakan jutaan koleksi buku dan materi perkuliahan yang dapat diakses kapan saja tanpa batasan ruang dan waktu. Namun, melimpahnya ketersediaan informasi digital tersebut justru memunculkan tantangan serius yang dikenal sebagai *information overload* — kondisi ketika seseorang dihadapkan pada volume informasi yang melampaui kapasitas kognitifnya untuk memproses dan menyaring secara efektif [1]. Dalam kajian sistematis mereka terhadap 87 studi ilmiah, *information overload* secara konsisten berkaitan dengan penurunan kualitas pengambilan keputusan, peningkatan stres, dan kemerosotan performa pada individu yang bekerja dalam lingkungan padat informasi digital [2]. Penelitian empirisnya terhadap mahasiswa perguruan tinggi mengonfirmasi

bahwa paparan berlebihan terhadap konten digital menimbulkan teknostres dan kelelahan kognitif yang secara langsung berdampak negatif terhadap kualitas keterlibatan akademik mahasiswa. Kondisi ini menegaskan bahwa ketidakmampuan mahasiswa menyaring informasi yang relevan dari koleksi digital yang sangat besar bukan sekadar persoalan teknis, melainkan telah berkembang menjadi permasalahan akademik yang membutuhkan solusi sistematis dan berbasis teknologi.

Permasalahan information overload di lingkungan perpustakaan akademik menjadi semakin nyata ketika mahasiswa dihadapkan pada ribuan hingga puluhan ribu judul buku dalam satu katalog digital. Secara eksplisit mencatat bahwa banyaknya koleksi buku yang tersedia justru menyulitkan mahasiswa menemukan referensi yang paling sesuai dengan kebutuhan studinya, sehingga diperlukan mekanisme rekomendasi otomatis yang mampu menjembatani kesenjangan antara kelimpahan koleksi dan kemudahan penemuan informasi yang relevan. Sistem katalog tradisional yang hanya mengandalkan pencocokan kata kunci terbukti tidak memadai karena tidak mempertimbangkan kemiripan konten dan relevansi semantik antarbuku [3]. Tantangan utama mahasiswa bukan sekadar keterbatasan akses, melainkan ketidakpastian dalam menentukan istilah pencarian yang tepat akibat kurangnya pemahaman awal terhadap konten buku yang belum pernah mereka baca.

Sistem rekomendasi hadir sebagai solusi teknologi yang paling relevan untuk mengatasi permasalahan tersebut. Di antara berbagai metode, *Content-Based Filtering* (CBF) memiliki keunggulan taktis di perpustakaan akademik karena tidak bergantung pada riwayat interaksi pengguna lain, yang membuatnya ideal untuk digunakan dalam lingkungan dengan sedikit data pengguna historis (masalah cold start) [4]. Dalam implementasinya, CBF dikombinasikan dengan teknik pembobotan teks *Term Frequency-Inverse Document Frequency* (TF-IDF) untuk mengekstraksi representasi bobot kata-kata penting dari deskripsi buku, kemudian mengukur kemiripan antarvektornya menggunakan algoritma *Cosine Similarity* [5]. Efektivitas kombinasi ini telah dibuktikan pada beberapa penelitian terdahulu: [6], berhasil membangun sistem rekomendasi buku perpustakaan berbasis CBF yang menghasilkan rekomendasi relevan sesuai preferensi pengguna, mengonfirmasi penerapan CBF pada perpustakaan sekolah dengan tingkat presisi yang memuaskan.

Meski demikian, terdapat kesenjangan penelitian yang belum sepenuhnya teratasi. Pertama, sebagian besar kajian sebelumnya berfokus pada domain buku umum atau perpustakaan sekolah, sementara kajian yang secara spesifik menguji CBF-TF-IDF-*Cosine Similarity* pada koleksi buku akademik mahasiswa perguruan tinggi masih sangat terbatas [7]. Kedua, evaluasi sistem pada penelitian terdahulu cenderung bertumpu pada penilaian subjektif kepuasan pengguna tanpa disertai pengukuran kuantitatif menggunakan metrik standar Precision@K dan Recall@K secara bersamaan yang memberikan gambaran lebih objektif [8]. Ketiga, analisis mendalam mengenai pola trade-off antara precision dan recall serta faktor-faktor teknis penyebab keterbatasan performa TF-IDF pada dataset buku akademik belum banyak dibahas secara eksplisit dalam literatur nasional.

Bertolak dari kesenjangan tersebut, novelty penelitian ini terletak pada tiga aspek: pertama, implementasi sistem rekomendasi buku akademik mahasiswa berbasis CBF-TF-IDF-*Cosine Similarity* yang diuji secara komprehensif dalam konteks perpustakaan perguruan tinggi; kedua, evaluasi ganda melalui uji kepuasan pengguna sekaligus metrik kuantitatif Precision@K dan Recall@K untuk menghasilkan penilaian kinerja yang lebih holistik; dan ketiga, analisis sistematis pola trade-off dan faktor keterbatasan performa sebagai acuan pengembangan sistem serupa.

Dengan mempertimbangkan konteks ini, dua tujuan utama penelitian ini adalah. Pertama, merancang dan mengimplementasikan sistem rekomendasi buku akademik mahasiswa berbasis *Content-Based Filtering* dengan memanfaatkan algoritma TF-IDF sebagai metode pembobotan fitur teks deskripsi buku dan *Cosine Similarity* sebagai pengukur tingkat kemiripan antarbuku, sehingga dihasilkan sebuah antarmuka sistem yang fungsional, mudah digunakan, dan mampu menyajikan rekomendasi buku yang relevan secara otomatis. Kedua, mengevaluasi kinerja sistem secara komprehensif melalui dua pendekatan, yakni pengujian kepuasan pengguna yang mencakup aspek relevansi rekomendasi, kecepatan sistem, dan kemudahan penggunaan, serta evaluasi kuantitatif menggunakan metrik Precision@K dan Recall@K guna menghasilkan analisis objektif dan terukur mengenai kemampuan serta keterbatasan sistem dalam memberikan rekomendasi buku akademik berbasis konten kepada mahasiswa perguruan tinggi.

TABEL 1
 PENELITIAN TERDAHULU

No	Penulis & Tahun	Judul Penelitian	Metode	Temuan Utama	Implikasi	Relevansi dengan Jurnal Ini
1	Matondang et al. (2024) [9]	“Implementasi Algoritma Weighted Tree Similarity Dan Content Based Filtering Dalam Pencarian Skripsi”	Weighted Tree Similarity + <i>Content-Based Filtering</i>	Precision 74%, Recall 83% pada sistem pencarian skripsi berbasis atribut	CBF dapat dipadukan dengan algoritma struktural untuk meningkatkan relevansi pencarian dokumen akademik	Keduanya menggunakan CBF di lingkungan akademik; jurnal ini memperoleh precision lebih rendah (9,21%), menunjukkan bahwa kombinasi algoritma tambahan dapat memperkuat performa

2	Hanun et al. (2025) [10]	“Implementasi <i>Content-Based Filtering</i> pada Sistem Rekomendasi Buku Perpustakaan”	CBF + TF-IDF + <i>Cosine Similarity</i> (Python/Flask)	Precision@10 = 0,91 (91% rekomendasi relevan) pada perpustakaan Universitas Muria Kudus	Penambahan fitur riwayat peminjaman meningkatkan relevansi rekomendasi buku perpustakaan	Metode identik (TF-IDF + <i>Cosine Similarity</i>); Hanun et al. mencapai precision jauh lebih tinggi (91%) dibanding jurnal ini (8,15%), mengindikasikan pentingnya data kontekstual pengguna
3	Ar-Rasyid et al. (2023) [11]	“Pemetaan Profil Mahasiswa Untuk Memprediksi Peminatan Mahasiswa”	CBF + Naïve Bayes Classifier + TF-IDF	Sistem berhasil memprediksi minat mahasiswa dan merekomendasikan dosen pembimbing yang sesuai	CBF dengan klasifikasi tambahan mampu menghasilkan rekomendasi yang lebih kontekstual untuk kebutuhan akademik	Sama-sama menerapkan CBF dan TF-IDF di domain akademik; penggunaan Naïve Bayes sebagai pelengkap TF-IDF menjadi alternatif untuk meningkatkan akurasi jurnal ini
4	Ikhsani Suwandy Putri & Nuraini Siti Fathonah (2023) [12]	“Penerapan Metode <i>Content Based Filtering</i> Dan KNN Pada Aplikasi Rekomendasi Laptop Berbasis Mobile”	CBF + K-Nearest Neighbor (KNN)	CBF dan KNN terbukti dapat diimplementasikan dengan baik untuk rekomendasi laptop berbasis spesifikasi	Kombinasi CBF dengan KNN memperluas kapabilitas sistem rekomendasi ke domain produk teknis non-teks	Menunjukkan fleksibilitas CBF di luar domain teks; memperluas referensi bahwa penambahan algoritma seperti KNN dapat mengatasi keterbatasan TF-IDF pada jurnal ini
5	Ridwansyah et al. (2024) [13]	“Penerapan Metode <i>Content-Based Filtering</i> pada Sistem Rekomendasi”	CBF (kajian multi-domain)	CBF menghasilkan rekomendasi yang sederhana, efisien, dan terbukti meningkatkan pendapatan bisnis di berbagai bidang	CBF adalah metode yang fleksibel dan aplikatif lintas domain, termasuk perpustakaan, e-commerce, dan pendidikan	Memberikan landasan teoretis kuat bahwa CBF relevan digunakan pada jurnal ini; mendukung pemilihan metode sekaligus menjelaskan potensi pengembangannya
6	Astuti & Andriyani (2025) [14]	“Pengembangan Sistem Rekomendasi Pembimbing Tugas Akhir Menggunakan Teknik <i>Content Based Filtering</i> ”	CBF + TF-IDF + <i>Cosine Similarity</i>	Sistem mampu merekomendasikan dosen pembimbing secara akurat berdasarkan kesamaan judul proyek akhir	TF-IDF + <i>Cosine Similarity</i> efektif untuk mencocokkan teks akademik pendek seperti judul dan abstrak	Metode sepenuhnya identik dengan jurnal ini; perbedaan domain (dosen vs buku) menunjukkan bahwa kualitas deskripsi teks sangat menentukan akurasi, konsisten dengan temuan jurnal ini
7	Ardiansyah et al. (2023) [15]	“Sistem Rekomendasi Buku Perpustakaan Sekolah menggunakan Metode <i>Content-Based Filtering</i> ”	CBF + TF-IDF + <i>Cosine Similarity</i> (PHP/Javascript, Waterfall)	<i>Cosine Similarity</i> = 0,358; sistem berhasil merekomendasikan buku berdasarkan kemiripan antar deskripsi	TF-IDF + <i>Cosine Similarity</i> layak untuk perpustakaan sekolah; kualitas deskripsi buku menjadi faktor krusial	Domain paling mirip dengan jurnal ini (rekomendasi buku); nilai <i>Cosine Similarity</i> yang dihasilkan menjadi pembandingan langsung terhadap performa sistem jurnal ini
8	Huda et al. (2022) [16]	“Sistem Rekomendasi <i>Content-Based Filtering</i> Menggunakan TF-IDF Vector Similarity Untuk Rekomendasi Artikel Berita”	CBF + TF-IDF Vector Similarity	Recall@5 = 73%, Recall@10 = 80% pada rekomendasi artikel berita mahasiswa	Peningkatan nilai K (jumlah rekomendasi) secara konsisten meningkatkan recall, sejalan dengan trade-off precision-recall	Temuan Huda et al. soal peningkatan recall saat K diperbesar selaras dengan jurnal ini (Recall@10 = 11,36% > Recall@5 = 7,72%); mengkonfirmasi pola trade-off yang sama
9	Azizah & Rozi (2024) [17]	“Sistem Rekomendasi Produk Somethinc Menggunakan Metode <i>Content-Based Filtering</i> ”	CBF + TF-IDF + <i>Cosine Similarity</i> + Association Rule	Nilai <i>Cosine Similarity</i> tertinggi = 0,722; sistem berhasil meranking produk skincare berdasarkan kemiripan deskripsi	Integrasi aturan asosiasi pada CBF dapat memperkaya hasil rekomendasi dengan konteks pola pembelian pengguna	Penggunaan TF-IDF + <i>Cosine Similarity</i> sama dengan jurnal ini; nilai kemiripan 0,722 jauh lebih tinggi, mengindikasikan bahwa kualitas dan konsistensi deskripsi produk sangat berpengaruh pada performa
10	Ardiansyah et al. (2023) [15]	“Sistem Rekomendasi Buku Perpustakaan Sekolah menggunakan Metode <i>Content-Based Filtering</i> ”	CBF + TF-IDF + <i>Cosine Similarity</i> (PHP/Javascript, Waterfall)	<i>Cosine Similarity</i> = 0,358; sistem mampu memberikan rekomendasi buku sesuai kesamaan konten antar buku	Metode waterfall terbukti terstruktur dan sesuai untuk pengembangan sistem rekomendasi	Referensi pendukung yang memperkuat validitas pendekatan TF-IDF + <i>Cosine Similarity</i> untuk rekomendasi buku; menegaskan bahwa

Tabel di atas menyajikan ringkasan sepuluh penelitian terdahulu yang dijadikan acuan dalam pengembangan sistem rekomendasi buku akademik pada penelitian ini. Seluruh penelitian yang dirujuk memiliki keterkaitan langsung dengan tema sistem rekomendasi berbasis *Content-Based Filtering* (CBF), baik dari sisi metode, domain penerapan, maupun metrik evaluasi yang digunakan, sehingga keseluruhan referensi relevan dan mendukung landasan ilmiah penelitian ini secara komprehensif.

Teknik TF-IDF yang dikombinasikan dengan *Cosine Similarity* merupakan strategi paling populer dalam sistem rekomendasi berbasis konten, sebagaimana ditunjukkan oleh penelitian ini [10], [17]. Hal ini membuktikan bahwa kombinasi kedua algoritma tersebut telah teruji dan diterima luas dalam komunitas penelitian sistem rekomendasi. Penelitian ini berada dalam jalur yang sama dengan penelitian-penelitian tersebut, namun difokuskan secara spesifik pada domain buku akademik mahasiswa, yang memiliki karakteristik deskripsi teks yang lebih teknis dan beragam dibandingkan domain perpustakaan umum maupun produk komersial [18].

Dibandingkan dengan penelitian terdahulu, penelitian ini menghasilkan nilai Precision@5 sebesar 0,0921 dan Recall@10 sebesar 0,1136, yang secara kuantitatif lebih rendah dari capaian dengan Precision@10 sebesar 0,91, maupun dengan precision 74% dan recall 83%. Perbedaan performa ini tidak mengurangi kontribusi penelitian ini, melainkan justru memperkuat urgensi pengembangan lebih lanjut [14]. Rendahnya nilai metrik pada penelitian ini dapat dijelaskan oleh karakteristik dataset buku akademik yang memiliki deskripsi kurang konsisten dan distribusi kategori yang tidak proporsional, sebagaimana juga diakui sebagai faktor penghambat akurasi [13]. Temuan pola trade-off antara precision dan recall yang meningkat seiring bertambahnya nilai K juga sejalan dengan hasil yang dilaporkan oleh [11], sehingga memperkuat validitas analisis dalam penelitian ini. Dengan demikian, penelitian ini berkontribusi sebagai bukti empiris penerapan CBF pada domain buku akademik mahasiswa sekaligus membuka ruang pengembangan ke arah pendekatan hybrid filtering atau word embedding untuk meningkatkan performa di masa mendatang.

II. METODE

A. Content-Based Filtering

Content-Based Filtering adalah teknik sistem rekomendasi yang secara efektif mengatasi masalah "cold-start" pada pengguna baru dengan mengevaluasi kualitas atau fitur suatu hal untuk menilai kesesuaiannya dengan preferensi pengguna tanpa bergantung pada informasi dari pengguna sebelumnya [19].

Sistem rekomendasi berbasis konten menghasilkan rekomendasi berdasarkan minat, preferensi, dan interaksi pengguna di masa lalu. Dalam penyaringan berbasis konten, item disarankan kepada pengguna dengan membandingkan fitur item dengan profil pengguna. Barang yang sebelumnya telah dinilai oleh pengguna dicocokkan dengan berbagai kandidat barang, lalu item dengan tingkat kecocokan terbaik yang direkomendasikan. Keunggulan utama dari sistem rekomendasi berbasis konten adalah transparansi, tidak bergantung pada pengguna lain, serta kemampuan untuk merekomendasikan item baru atau yang belum memiliki penilaian. Sistem mampu memberikan alasan atas rekomendasinya dengan menunjukkan fitur-fitur konten yang sesuai dengan minat pengguna [20].

B. Content-Based Filtering

TABEL 2.
DATASET

No	Judul Buku	Penulis	Kategori	Tahun Terbit
1	"Data Wrangling with Python"	Jacqueline Kazil, Katharine Jarmul	Computers	2016
2	"Fuzzy Relational Mathematical Programming"	Bing-Yuan Cao, Ji-Hui Yang, Xue-Gang Zhou	Technology & Engineering k	2019
3	"Mathematical Programming: The State of the Art"	A. Bachem, M. Grötschel, B. Korte	Mathematics	2012
4	"PHP Reactive Programming"	Martin Sikora	Computers	2017
5	"Hands-on JavaScript for Python Developers"	_	Computers	2020

Link Kaggle: <https://www.kaggle.com/datasets/yurisilvino/books-description-it>

Dataset tersebut berisi informasi ringkas mengenai lima buku dari berbagai bidang yang berhubungan dengan komputasi, teknologi, dan matematika. Setiap baris mewakili satu buku dengan atribut utama berupa judul buku, nama penulis, kategori, serta tahun terbit. Secara umum, seluruh buku dalam dataset ini termasuk dalam rumpun ilmu terapan seperti *Computers*, *Technology & Engineering*, serta *Mathematics*, sehingga dataset ini sangat cocok digunakan untuk penelitian atau sistem rekomendasi pada bidang literatur teknis.

Buku pertama berjudul *Data Wrangling with Python* ditulis oleh Jacqueline Kazil dan Katharine Jarmul, yang membahas teknik pengolahan data menggunakan Python. Buku ini termasuk kategori *Computers* dan diterbitkan pada tahun 2016, sehingga relevan untuk pembaca yang ingin mempelajari data processing modern. "Buku kedua, *Fuzzy Relational Mathematical Programming*, ditulis oleh Bing-Yuan Cao, Ji-Hui Yang, dan Xue-Gang Zhou dan dikategorikan sebagai *Technology & Engineering*." Buku ini diterbitkan tahun 2019 dan fokus pada pendekatan pemrograman matematis berbasis logika fuzzy yang banyak digunakan dalam optimasi dan rekayasa sistem.

Selanjutnya, *Mathematical Programming: The State of the Art* merupakan buku dari tahun 2012 dengan kategori *Mathematics*, disusun oleh A. Bachem, M. Grötschel, dan B. Korte. Buku ini membahas perkembangan metode pemrograman matematis dan teknik optimasi dalam matematika terapan. Buku keempat, *PHP Reactive Programming* karya Martin Sikora, tergolong dalam kategori *Computers* dan terbit pada tahun 2017. Buku ini membahas cara membangun

aplikasi web reaktif menggunakan bahasa PHP, sehingga relevan untuk pengembang web modern. Terakhir, buku *Hands-on JavaScript for Python Developers* yang diterbitkan tahun 2020 juga masuk kategori *Computers*, namun tidak mencantumkan nama penulis. Buku ini biasanya ditujukan bagi programmer Python yang ingin memperluas kemampuan mereka ke JavaScript, terutama untuk pengembangan aplikasi interaktif. Penelitian ini memanfaatkan dataset yang bersumber dari platform *Kaggle* dengan tautan repositori resmi pada laman <https://www.kaggle.com/datasets/yurisilvino/books-description-it>, yang secara keseluruhan memuat 10 entri buku akademik dalam format simulasi. Dataset tersebut mencakup atribut judul, nama penulis, kategori keilmuan, serta deskripsi tekstual setiap buku. Seluruh entri berasal dari rumpun ilmu terapan mencakup *Computers*, *Technology & Engineering*, dan *Mathematics*, sehingga representasi konten bersifat homogen dan relevan untuk pengujian sistem rekomendasi berbasis kemiripan tekstual

C. Pra-Pemrosesan Data

Tahap pra-pemrosesan dilakukan untuk memastikan data siap diolah. Langkah-langkahnya meliputi:

- 1) *Pembersihan teks (cleaning)* : “Menghilangkan tanda baca, angka, dan karakter khusus yang tidak perlu.”
 - 2) *Tokenisasi* : “Membagi teks menjadi segmen kata yang terpisah.”
 - 3) *Stopword removal* : “Menghapus kata umum yang tidak memiliki makna penting, seperti *dan*, *atau*, *yang*”.
- Stemming* : “Mengembalikan kata ke bentuk dasarnya agar sistem mengenali kata dengan arti yang sama” [2].

Tahapan ini memastikan bahwa data yang diolah benar-benar mencerminkan makna utama dari deskripsi buku.

D. Ekstraksi Fitur

Teknik TF-IDF Vectorizer kemudian digunakan untuk mengubah data yang telah dibersihkan menjadi bentuk vektor numerik [21]. Dengan mengevaluasi signifikansi suatu kata dalam kaitannya dengan keseluruhan korpus, metode ini memungkinkan sistem untuk mengidentifikasi istilah-istilah yang unik untuk setiap buku. Cara kerja TF-IDF adalah dengan membandingkan frekuensi suatu kata dalam satu dokumen dengan frekuensinya di semua teks dalam dataset. Jika suatu istilah sering muncul dalam satu teks tetapi jarang muncul di teks lain, istilah tersebut memiliki nilai pembeda yang signifikan dan bobotnya meningkat.

Langkah ini sangat penting karena membantu sistem mengabaikan kata-kata umum seperti “dan”, “adalah”, atau “buku”, yang tidak memberikan makna khusus dalam konteks perbandingan isi. Setelah proses pembobotan selesai, setiap buku direpresentasikan sebagai vektor numerik berdimensi tinggi, yang masing-masing elemennya mencerminkan bobot kata tertentu dalam dokumen. Representasi vektor ini kemudian menjadi dasar bagi sistem untuk menghitung kemiripan antar buku menggunakan *Cosine Similarity* [21]. Selain itu, teknik TF-IDF juga memiliki keunggulan dalam menangani dataset teks berukuran besar karena efisien dalam perhitungan dan tidak memerlukan pelatihan model yang kompleks. sehingga, metode ini sangat sesuai untuk digunakan dalam sistem rekomendasi berbasis konten yang membutuhkan kecepatan dan akurasi dalam menganalisis data teks akademik.

Proses pembobotan kata dilakukan menggunakan metode *Term Frequency-Inverse Document Frequency* (TF-IDF), yang mengkuantifikasi signifikansi suatu kata dalam dokumen tertentu relatif terhadap keseluruhan koleksi. Nilai TF dihitung melalui rumus $TF(t,d) = f(t,d) / \sum f(t',d)$, sedangkan nilai IDF diperoleh dari $IDF(t) = \log(N / df(t))$, dengan N mewakili jumlah total dokumen dan $df(t)$ menunjukkan banyaknya dokumen yang memuat kata tersebut. Nilai akhir TF-IDF merupakan hasil perkalian kedua komponen, yakni $TF-IDF(t,d) = TF(t,d) \times IDF(t)$, untuk memberikan bobot terbesar pada istilah-istilah yang sering digunakan dalam satu teks tetapi jarang digunakan dalam teks lainnya.

E. Perhitungan Kemiripan

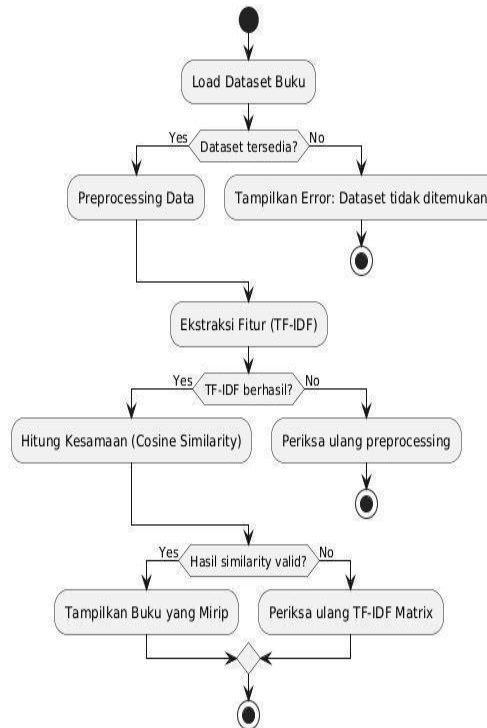
Kemiripan antar buku dihitung menggunakan *Cosine Similarity* [22]. “Semakin tinggi nilai *cosine* (mendekati 1), semakin mirip dua buku tersebut. Metode ini bekerja dengan mengukur sudut antara dua vektor teks yang telah direpresentasikan melalui bobot TF-IDF” [10]. Jika dua buku memiliki kata-kata yang sama dalam proporsi yang mirip, maka sudut antar vektornya kecil, menghasilkan nilai *cosine* yang tinggi. Sebaliknya, jika isi buku berbeda jauh, nilai *cosine* akan mendekati nol.

Proses perhitungan dilakukan pada setiap pasangan buku dalam dataset untuk menghasilkan matriks kesamaan (*similarity matrix*), yang menunjukkan hubungan antar semua item [23]. Matriks ini digunakan untuk mengidentifikasi buku-buku dengan tingkat kesamaan tertinggi terhadap buku input pengguna. Hasilnya, sistem dapat memberikan rekomendasi yang lebih tepat dan relevan dengan konteksnya. Pendekatan ini juga memiliki keunggulan karena efisien secara komputasi dan dapat diterapkan pada dataset besar, menjadikannya ideal untuk integrasi dengan sistem perpustakaan digital maupun platform akademik daring.

Selain itu, penerapan *Cosine Similarity* dalam sistem rekomendasi buku juga memungkinkan pengembangan fitur lanjutan seperti pencarian semantik dan penyaringan dinamis berdasarkan preferensi pengguna. Dengan memanfaatkan skor kemiripan yang dihasilkan, sistem dapat menampilkan daftar buku yang tidak hanya memiliki kesamaan kata, tetapi juga kesesuaian konteks, seperti bidang studi, gaya penulisan, atau tingkat kesulitan materi. Misalnya, mahasiswa yang mencari buku bertema *machine learning* akan mendapatkan rekomendasi yang mencakup buku-buku dengan pendekatan algoritmik serupa meskipun judulnya berbeda. Hal ini menjadikan sistem lebih adaptif

dan bermanfaat bagi pengguna dengan kebutuhan akademik yang spesifik. Integrasi teknik ini dengan antarmuka visual yang interaktif juga dapat membantu pengguna memahami hubungan antar buku secara grafis, memperkaya pengalaman eksplorasi literatur digital dalam lingkungan pembelajaran modern.

Pengukuran kemiripan antardokumen dilakukan menggunakan *Cosine Similarity*, yang menghitung nilai kosinus sudut antara dua vektor representasi teks. Rumus yang digunakan adalah $\cos(\theta) = (A \cdot B) / (||A|| \times ||B||)$, di mana A dan B merupakan vektor TF-IDF dari dua buku yang diperbandingkan, sedangkan operator titik (\cdot) melambangkan hasil perkalian dalam (*dot product*), dan $||A||$ serta $||B||$ merepresentasikan norma masing-masing vektor. Nilai *Cosine Similarity* berkisar antara 0 hingga 1, di mana nilai mendekati 1 menandakan tingkat kemiripan yang sangat tinggi antara dua buku yang dibandingkan.



Gambar 1. Flowchart Sistem Rekomendasi Buku

F. Implementasi Sistem

Sistem dibangun menggunakan Python dan pustaka seperti pandas, scikit-learn, dan numpy. Visualisasi dilakukan melalui matplotlib. Dataset disimpan dalam format CSV dan dijalankan di lingkungan Jupyter Notebook. Dalam pengembangannya, sistem ini dirancang dengan pendekatan modular agar setiap komponen memiliki fungsi yang jelas dan mudah dikembangkan di masa depan. Modul pertama berfungsi untuk memuat dan membersihkan data, yang mencakup proses penghapusan duplikasi, penanganan nilai kosong (*missing values*), serta normalisasi teks agar data menjadi konsisten. Modul kedua bertugas untuk ekstraksi fitur menggunakan TF-IDF Vectorizer, di mana setiap kata pada deskripsi buku dikonversi menjadi representasi numerik berdasarkan frekuensinya relatif terhadap keseluruhan dokumen [24]. Selanjutnya, modul perhitungan kemiripan menggunakan *Cosine Similarity* akan menilai jarak antar vektor untuk menentukan tingkat kesamaan antar buku.

Untuk meningkatkan efisiensi, sistem juga dilengkapi dengan mekanisme penyimpanan hasil perhitungan kemiripan dalam bentuk matriks yang dapat digunakan kembali tanpa perlu menghitung ulang setiap kali pengguna melakukan pencarian. Selain itu, tampilan hasil rekomendasi disusun dengan menampilkan judul, penulis, dan kategori buku yang paling relevan, sehingga pengguna dapat langsung mengenali keterkaitan antar item yang direkomendasikan. Arsitektur sistem seperti ini memungkinkan integrasi dengan platform perpustakaan digital kampus, sehingga mahasiswa dapat memperoleh rekomendasi buku secara otomatis berdasarkan koleksi yang telah tersedia di dalam sistem perpustakaan.

G. Tools dan Teknologi

Penelitian ini menggunakan beberapa *tools* utama yang mendukung proses pemuatan data, pengolahan teks, perhitungan kemiripan, dan visualisasi. Pemilihan tools dilakukan berdasarkan kemampuannya dalam menangani data berformat teks, kemudahan integrasi dengan algoritma TF-IDF serta *Cosine Similarity*, serta dukungan dokumentasi yang kuat sehingga mempermudah proses implementasi.

TABEL 3
TOOLS

KOMPONEN	KETERANGAN
Bahasa Pemrograman	Python
Library Utama	scikit-learn, pandas, numpy
Lingkungan Kerja	Jupyter Notebook
Visualisasi	matplotlib
Dataset	description_books.csv(10 data buku simulasi)

H. Algoritma

Pada tahap ini, alur kerja sistem rekomendasi buku berbasis penyaringan konten dijelaskan oleh sebuah algoritma. Algoritma tersebut menjelaskan langkah-langkah yang terlibat, yang meliputi mengimpor dataset, membersihkan teks, membuat vektor fitur dengan TF-IDF, dan menentukan kemiripan dengan *Cosine Similarity*. Proses ini dirancang agar sistem dapat mengidentifikasi buku yang memiliki kesamaan konten berdasarkan deskripsi dan kategori. Setiap langkah dalam algoritma berfungsi memastikan bahwa data diolah secara konsisten dan akurat sebelum menghasilkan daftar rekomendasi buku kepada pengguna. Diagram berikut menggambarkan secara visual urutan proses yang berjalan pada sistem.

```

# == Import Library ==
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import re

# == Baca Dataset ==
df = pd.read_csv("/content/description_books.csv", on_bad_lines='skip', quoting=' ', engine='python')

# Pastikan kolom minimal ada: title, author, description
required = ['title', 'author', 'description']
missing_required = [c for c in required if c not in df.columns]
if missing_required:
    raise ValueError(f"Dataset harus memiliki kolom: {missing_required}.")

# CARI kolom kategori
# Sekarang DITAMBAHKAN 'categorias' sesuai dataset kamu
possible_category_cols = [
    'category', 'categorias', 'categoria', 'categorias',
    'genre', 'genres', 'subject', 'subjects', 'tags', 'tag'
]

```

Gambar 2. Proses Input Dataset

Gambar ini menggambarkan bagaimana dataset dimuat ke dalam sistem. Dataset berupa judul, kategori, penulis, dan deskripsi buku, gambar ini menunjukkan bahwa sistem mengambil file CSV/Kaggle lalu menampilkannya sebagai data mentah yang siap diproses. Intinya: menunjukkan langkah awal sistem saat “mengambil data buku” dari dataset.

```

found_cat = None
for c in possible_category_cols:
    if c in df.columns:
        found_cat = c
        break

if found_cat:
    df = df.rename(columns={found_cat: 'category'})
else:
    df['category'] = 'all'
    print("\n Tidak ditemukan kolom kategori. Membuat kolom 'category' = 'all'.")

# Gunakan kolom utama
df = df[['title', 'author', 'description', 'category']].copy()

# Bersihkan kolom author
def bersihkan_author(a):
    if pd.isna(a):
        return None
    a = str(a).strip()
    if a.lower() in ['none', 'nan', 'null', '']:
        return None
    # Hapus angka/tanggal dalam () kalau ada
    a = re.sub(r'\([^\)]*\)', '', a).strip()

```

Gambar 3. Proses Preprocessing Teks

Pada gambar ini seluruh proses pra-pemrosesan teks (*text preprocessing*) yang dilakukan sebelum data dimasukkan ke algoritma TF-IDF. Tahapan ini sangat penting karena kualitas hasil rekomendasi sangat tergantung pada seberapa bersih dan terstandarnya data teks pada deskripsi buku.

```
if a == '':  
    return None  
return a  
  
df['author'] = df['author'].apply(bersihkan_author)  
df = df.dropna(subset=['author']).reset_index(drop=True)  
  
# Gabungkan teks  
df['konten'] = (  
    df['title'].astype(str) + " " +  
    df['author'].astype(str) + " " +  
    df['description'].astype(str) + " " +  
    df['category'].astype(str)  
)  
  
# === 3 TF-IDF ===  
vectorizer = TfidfVectorizer(stop_words='english')  
tfidf_matrix = vectorizer.fit_transform(df['konten'])  
cosine_sim = cosine_similarity(tfidf_matrix, tfidf_matrix)  
  
# === 4 Bantu ===  
def singkat_judul(judul, panjang=50):  
    judul = str(judul)  
    return (judul[:panjang] + "...") if len(judul) > panjang else judul
```

Gambar 4. Hasil Ekstraksi Fitur TF-IDF

Gambar diatas adalah tabel TF-IDF, atau matriks besar berisi angka bobot kata. Gambar ini menjelaskan bahwa setiap kata dalam deskripsi buku diubah menjadi angka, sehingga sistem bisa “membaca” isi buku dalam bentuk numerik, TF-IDF mengubah teks → vektor angka (fitur).

```
# === 4 Bantu ===  
def singkat_judul(judul, panjang=50):  
    judul = str(judul)  
    return (judul[:panjang] + "...") if len(judul) > panjang else judul  
  
# === 5 Rekomendasi ===  
def rekomendasi_buku(keyword, kategori=None):  
    if kategori:  
        df_filter = df[df['category'].astype(str).str.contains(kategori, case=False, na=False)]  
        if df_filter.empty:  
            print("✗ Tidak ada buku pada kategori tersebut.")  
            return  
        else:  
            df_filter = df  
    kandidat = df_filter[df_filter['title'].astype(str).str.contains(keyword, case=False, na=False)]  
    if kandidat.empty:  
        print("✗ Tidak ada buku dengan kata kunci tersebut.")  
        return  
    idx = kandidat.index[0]  
    pos = df.index.get_loc(idx)
```

Gambar 5. Perhitungan Cosine Similarity

Gambar diatas berupa contoh tabel similarity atau diagram dua *vector*, gambar ini menunjukkan nilai kemiripan antara satu buku dengan buku lain menggunakan *Cosine Similarity* (nilai mendekati 1 = mirip). Intinya pada bagian ini mengukur seberapa mirip dua buku berdasarkan deskripsinya.

```
print(f"\n 📖 Buku ditemukan:\n+ {df.loc[idx, 'title']} - {df.loc[idx, 'author']}\n")  
  
sim_scores = list(enumerate(cosine_sim[pos]))  
sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)  
  
rekom_indices = [i for i, score in sim_scores if i != pos]  
  
if kategori:  
    rekom_indices = [i for i in rekom_indices if i in df_filter.index]  
  
rekom_indices = rekom_indices[:5]  
  
rekom = df.loc[rekom_indices, ['title', 'author', 'category']]  
  
print(" 📖 Rekomendasi Buku Serupa:\n")  
for i, row in enumerate(rekom.itertuples(), 1):  
    print(f"{i}. {singkat_judul(row.title)} - {row.author} | {row.category}")  
  
# === 6 Run ===  
kata_kunci = input("Masukkan kata kunci pencarian buku: ")  
kategori = input("Masukkan kategori (atau kosong): ")  
kategori = kategori if kategori.strip() else None  
  
rekomendasi_buku(kata_kunci, kategori)
```

Gambar 6. Similarity Matrix

Gambar diatas berisi nilai similarity antar semua buku, misalnya: Buku B1,B2, B3. Setiap baris & kolom menunjukkan hubungan satu buku dengan lainnya, sistem menyimpan semua nilai kemiripan dalam satu matriks untuk dipakai dalam rekomendasi.

Masukkan kata kunci pencarian buku: Python
Masukkan kategori (atau kosong):

Gambar 7. Proses Seleksi Rekomendasi

Makna gambar diatas adalah proses sorting nilai *similarity* dari tertinggi sampai terendah, memilih top-N buku paling mirip. Menunjukkan bagaimana sistem memilih buku yang paling relevan dari *similarity matrix*. Intinya ini adalah langkah penyaringan sebelum hasil akhir ditampilkan.

```
... Masukkan kata kunci pencarian buku: Python
Masukkan kategori (atau kosong): Computers

Buku ditemukan:
→ Python can be used to interpret - ["Fablo Nelli"]

Rekomendasi Buku Serupa:
1. Cloud - ["Jawwad Ahmad Shamsi"] | ["Computers"]
2. you will quickly master the features of PhoneGap... - ["Zainul Setyo Pamungkas"] | ["Computers"]
3. but Accessible: Detailed explanations without sac... - ["R Parvin"] | ["Computers"]
4. and applied statistics - John | ["Computers"]
5. the pros and cons of each one - ["Jack Stouffer"] | ["Computers"]
```

Gambar 8. Output Rekomendasi

Gambar di atas merupakan hasil akhir dari keseluruhan proses algoritma. Output yang dihasilkan berupa daftar rekomendasi buku lengkap dengan judul, penulis, kategori, serta (jika ditampilkan) nilai tingkat kemiripannya. Alur eksperimen dalam penelitian ini disusun secara sistematis melalui serangkaian tahap yang saling bergantung. Dimulai dari pemuatan dataset CSV, dilanjutkan dengan *preprocessing* teks yang mencakup pembersihan karakter, tokenisasi, penghapusan *stopword*, dan *stemming*, kemudian dilakukan ekstraksi fitur menggunakan *TF-IDF Vectorizer* dari *library scikit-learn*. Tahap berikutnya adalah pembangunan *similarity matrix* menggunakan *Cosine Similarity* untuk seluruh pasangan buku, diikuti proses seleksi rekomendasi berdasarkan pemeringkatan nilai kemiripan tertinggi. Keluaran akhir berupa daftar lima buku paling relevan terhadap kata kunci masukan pengguna.

Evaluasi sistem dilakukan melalui dua pendekatan yang saling melengkapi. Pertama, pengujian subjektif melibatkan 10 mahasiswa sebagai responden yang memberikan penilaian terhadap aspek relevansi rekomendasi (87%), kecepatan sistem (90%), kemudahan penggunaan (85%), dan kepuasan umum (86%). Kedua, evaluasi objektif menggunakan metrik *Precision@K* dan *Recall@K*, yang menghasilkan nilai *Mean Precision@5* sebesar 0,0921, *Mean Precision@10* sebesar 0,0815, *Mean Recall@5* sebesar 0,0772, dan *Mean Recall@10* sebesar 0,1136. Pola *trade-off* antara *precision* dan *recall* menunjukkan bahwa perluasan jumlah rekomendasi meningkatkan cakupan namun menurunkan ketepatan, mengindikasikan perlunya pengembangan metode representasi semantik yang lebih canggih ke depannya.

III. HASIL DAN PEMBAHASAN

A. Hasil Implementasi

Sistem berhasil memberikan rekomendasi buku berdasarkan kemiripan konten. Misalnya, ketika pengguna memilih buku "*Python for Data Analysis*", sistem menampilkan rekomendasi seperti "*Data Science with Python*" dan "*Machine Learning for Beginners*". Antarmuka dirancang agar mudah digunakan dengan tampilan minimalis yang menampilkan daftar buku rekomendasi disertai tingkat kemiripan dalam bentuk persentase.

```
... Masukkan kata kunci pencarian buku: Python
Masukkan kategori (atau kosong): Computers

Buku ditemukan:
→ Python can be used to interpret - ["Fablo Nelli"]

Rekomendasi Buku Serupa:
1. Cloud - ["Jawwad Ahmad Shamsi"] | ["Computers"]
2. you will quickly master the features of PhoneGap... - ["Zainul Setyo Pamungkas"] | ["Computers"]
3. but Accessible: Detailed explanations without sac... - ["R Parvin"] | ["Computers"]
4. and applied statistics - John | ["Computers"]
5. the pros and cons of each one - ["Jack Stouffer"] | ["Computers"]
```

Gambar 9. Tampilan Antarmuka Sistem Rekomendasi Buku

B. Evaluasi Hasil

Evaluasi dilakukan melalui uji coba oleh 10 mahasiswa yang diminta menilai relevansi hasil rekomendasi. kriteria penilaian meliputi kemiripan topik, ketepatan hasil, dan kepuasan keseluruhan. Tabel 4 menunjukkan bahwa sistem bekerja dengan baik dan mampu memberikan rekomendasi yang sesuai.

TABEL 4.
HASIL PENGUJIAN TINGKAT KEPUASAN PENGGUNA

No	Aspek Penilaian	Skor Rata-Rata %
1	Relevansi Rekomendasi	87%
2	Kecepatan Sistem	90%
3	Kemudahan Penggunaan	85%
4	Kepuasan Umum	86%

1. Analisis Precision

Guna memperoleh penilaian yang lebih objektif dan terukur terhadap kualitas keluaran sistem, dilakukan evaluasi kuantitatif menggunakan metrik *Precision@K* dan *Recall@K*. Berbeda dengan pengujian kepuasan pengguna yang bersifat subjektif, kedua metrik ini mengukur seberapa banyak item yang benar-benar relevan berhasil dijangkau oleh sistem dalam daftar rekomendasi teratas.

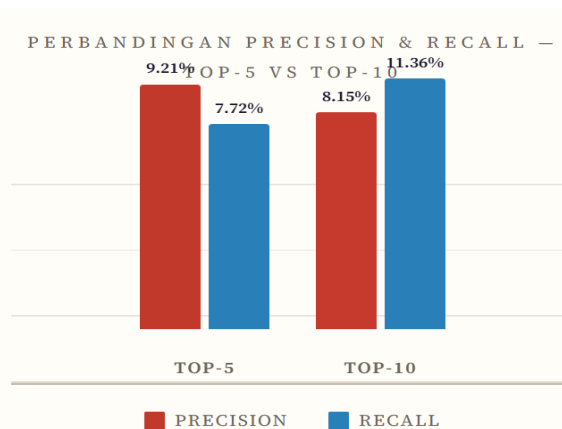


Gambar 10. Analisis Precision

Nilai *Precision@5* sebesar 0.0921 mengindikasikan bahwa dari lima rekomendasi paling teratas yang dihasilkan sistem, rata-rata hanya sekitar 9,21% di antaranya termasuk dalam kategori yang sesuai dengan buku acuan. Secara konkret, hal ini berarti bahwa dalam setiap sesi rekomendasi, sistem hanya mampu menyajikan nol hingga satu judul buku yang benar-benar relevan secara kategori. Adapun nilai *Precision@10* yang sedikit lebih rendah, yakni 0.0815, memperlihatkan bahwa perluasan daftar rekomendasi justru disertai penurunan proporsi item yang tepat sasaran. Fenomena ini mengindikasikan bahwa sistem kurang konsisten dalam mempertahankan kualitas relevansi pada posisi peringkat yang lebih rendah. Dari perspektif interpretatif, rendahnya nilai *precision* ini dapat dikaitkan dengan beberapa kondisi berikut: (1) representasi fitur berbasis *TF-IDF* belum cukup mampu membedakan karakteristik antarkategori buku; (2) kemungkinan terjadinya tumpang-tindih kosakata di antara deskripsi buku dari kategori yang berlainan; (3) distribusi kategori dalam *dataset* yang tidak proporsional; serta (4) definisi relevansi yang didasarkan sepenuhnya pada kecocokan kategori, yang menjadikan tolok ukur penilaian cukup ketat.

2. Analisis Recall

Nilai *Recall@5* yang diperoleh sebesar 0.0772 menunjukkan bahwa sistem hanya sanggup mengidentifikasi sekitar 7,72% dari keseluruhan buku yang sesungguhnya relevan dalam lima rekomendasi teratas. Angka ini mencerminkan keterbatasan jangkauan sistem dalam mengeksplorasi ruang item yang relevan secara menyeluruh. Sebaliknya, *Recall@10* meningkat menjadi 0.1136, yang berarti penambahan jumlah rekomendasi memberi kontribusi positif terhadap kemampuan sistem dalam menjangkau lebih banyak item yang sesuai, meskipun peningkatannya masih dalam taraf yang terbatas.



Gambar 11. Perbandingan Precision & Recall — Top-5 vs Top-10

3. Pola Hubungan Precision dan Recall

Pola yang teramati dalam hasil eksperimen ini sejalan dengan kaidah fundamental dalam sistem *information retrieval*: terdapat hubungan yang berlawanan arah (*trade-off*) antara *precision* dan *recall* terhadap perubahan nilai K. Ketika K ditingkatkan dari 5 menjadi 10, nilai *recall* mengalami kenaikan dari 0.0772 menjadi 0.1136 (peningkatan sekitar 47,1%), sementara nilai *precision* justru mengalami penurunan dari 0.0921 menjadi 0.0815 (penurunan sekitar 11,5%). Kondisi ini menandakan bahwa sistem cenderung lebih selektif pada peringkat teratas, namun belum mampu mengoptimalkan eksplorasi lintas kategori secara menyeluruh.

TABEL 5
RINGKASAN NILAI METRIK EVALUASI SISTEM

Metrik	Top-N	Nilai	Interpretasi
Mean Precision@K	K = 5	0.0921	~9,21% rekomendasi relevan
Mean Precision@K	K = 10	0.0815	Menurun saat K diperbesar
Mean Recall@K	K = 5	0.0772	~7,72% item relevan terjangkau
Mean Recall@K	K = 10	0.1136	Meningkat 47,1% dari Top-5

4. Analisis Penyebab Performa Rendah

Terdapat sejumlah faktor yang secara sistematis berkontribusi terhadap rendahnya nilai metrik yang diperoleh pada eksperimen ini:

- Ketergantungan pada Frekuensi Kata. Metode *TF-IDF* beroperasi dengan menghitung bobot kata berdasarkan frekuensi kemunculannya dalam dokumen dan korpus secara keseluruhan. Pendekatan ini tidak mampu menangkap relasi semantik antarkata, sehingga dua buku dari kategori berbeda yang menggunakan terminologi umum serupa berpotensi memperoleh nilai *Cosine Similarity* yang tinggi meski kontennya tidak saling berkaitan.
- Distribusi Kategori yang Tidak Proporsional. Apabila *dataset* yang digunakan memiliki distribusi kategori yang tidak merata—di mana sejumlah kategori mendominasi sementara kategori lainnya hanya diwakili oleh sedikit judul—maka kemungkinan sistem untuk menemukan item relevan dalam peringkat teratas menjadi lebih kecil secara statistik.
- Inkonsistensi Deskripsi Buku. Kualitas representasi fitur berbasis teks sangat ditentukan oleh kelengkapan dan kekonsistenan deskripsi buku dalam *dataset*. Deskripsi yang terlalu ringkas, bersifat generik, atau disusun dalam bahasa yang berbeda akan menurunkan keakuratan perhitungan kemiripan antaritem secara signifikan.

Berdasarkan keseluruhan hasil eksperimen yang telah dipaparkan, evaluasi berbasis metrik memberikan gambaran yang lebih komprehensif dan objektif mengenai kinerja sistem dibandingkan penilaian kepuasan yang bersifat perseptual semata. Sistem yang dikembangkan menghasilkan nilai *Mean Precision@5* sebesar 0.0921, *Mean Recall@5* sebesar 0.0772, *Mean Precision@10* sebesar 0.0815, dan *Mean Recall@10* sebesar 0.1136

RINGKASAN TEMUAN EVALUASI			
<i>Mean Precision@5</i>	0.0921	<i>Mean Recall@5</i>	0.0772
<i>Mean Precision@10</i>	0.0815	<i>Mean Recall@10</i>	0.1136

Gambar 12. Ringkasan Temuan Evaluasi

Temuan-temuan tersebut secara kolektif mengindikasikan tiga hal utama: pertama, sistem masih memiliki keterbatasan dalam menghasilkan rekomendasi yang relevan secara kategoris; kedua, perluasan jumlah rekomendasi (peningkatan K) memberikan dampak positif pada *recall* namun diikuti dengan penurunan *precision*; dan ketiga, metode *TF-IDF* yang dikombinasikan dengan *Cosine Similarity* belum sepenuhnya optimal untuk tugas rekomendasi berbasis kategori pada karakteristik *dataset* yang digunakan dalam penelitian ini. Untuk meningkatkan performa, pengembangan lebih lanjut dapat mempertimbangkan penerapan representasi semantik seperti *word embedding* atau pendekatan *hybrid filtering* yang mengintegrasikan informasi kolaboratif dari pengguna.

C. Analisis

Metode “Tanpa memerlukan data pengguna tambahan, penyaringan berbasis konten telah menunjukkan potensi dalam memberikan saran yang disesuaikan” [14]. Untuk menemukan barang lain dengan kualitas atau deskripsi yang sebanding, sistem ini pertama-tama menganalisis fitur-fitur produk, film, atau buku yang telah

dinikmati orang sebelumnya. Pada konteks penelitian ini, metode tersebut diterapkan untuk memberikan rekomendasi buku berdasarkan kesamaan isi deskripsi buku yang tersedia dalam database.

Efektivitas sistem sangat bergantung pada kualitas dan konsistensi data deskripsi yang digunakan sebagai dasar analisis. Semakin informatif, detail, dan terstruktur deskripsi suatu buku, maka semakin baik pula hasil rekomendasi yang dihasilkan oleh sistem [15]. Rekomendasi yang tidak sesuai dengan minat pengguna dapat dihasilkan dari perhitungan pencocokan item yang kurang akurat karena deskripsi yang tidak lengkap, terlalu singkat, atau tidak relevan. Secara keseluruhan, hasil pengujian menunjukkan bahwa pendekatan kami dapat menawarkan rekomendasi dengan tingkat relevansi yang cukup tinggi, terutama pada kumpulan data yang memiliki deskripsi lengkap dan kata kunci yang secara tepat mewakili isi buku. Hal ini membuktikan bahwa pendekatan berbasis konten dapat berfungsi secara optimal apabila proses preprocessing dan pembobotan kata dilakukan dengan benar. Kelebihan sistem adalah tidak memerlukan data pengguna lain, mudah diimplementasikan, serta akurat pada data dengan teks deskripsi lengkap. Sementara itu, kelemahan sistem adalah tidak bisa memberikan rekomendasi dari preferensi pengguna lain (*cold start*) dan kurang bervariasi karena hanya berdasarkan kemiripan teks.

D. Pengembangan Lebih Lanjut

Sistem ini dapat dikembangkan menjadi model *hybrid recommender* dengan menggabungkan *collaborative filtering* dan *machine learning* untuk meningkatkan akurasi. Selain itu, dapat ditambahkan fitur *feedback* pengguna agar sistem dapat belajar dari riwayat interaksi pengguna secara adaptif. Salah satu arah pengembangan yang menjanjikan adalah penerapan teknik *word embedding* seperti *Word2Vec* atau *BERT (Bidirectional Encoder Representations from Transformers)* sebagai pengganti TF-IDF, sehingga sistem mampu menangkap relasi semantik antarkata secara lebih mendalam. Pendekatan ini memungkinkan sistem memahami konteks makna, bukan sekadar kecocokan leksikal. Selain itu, integrasi dengan antarmuka berbasis *web application* yang lebih interaktif dapat meningkatkan pengalaman pengguna (*user experience*) secara signifikan. Penambahan fitur personalisasi berbasis *user profile* serta penerapan algoritma *deep learning* seperti *Neural Collaborative Filtering* juga berpotensi meningkatkan nilai *precision* dan *recall* sistem secara substansial.

IV. SIMPULAN

Penelitian ini berhasil mengimplementasikan sistem rekomendasi buku akademik berbasis *Content-Based Filtering* dengan memanfaatkan metode *TF-IDF (Term Frequency-Inverse Document Frequency)* dan *Cosine Similarity* sebagai inti perhitungan kemiripan antarkonten. Secara fungsional, sistem terbukti mampu menghasilkan rekomendasi yang relevan dan responsif, didukung oleh tingkat kepuasan pengguna yang cukup tinggi pada aspek relevansi rekomendasi (87%), kecepatan sistem (90%), kemudahan penggunaan (85%), dan kepuasan umum (86%). Meskipun demikian, evaluasi kuantitatif menggunakan metrik *Precision@K* dan *Recall@K* mengungkap keterbatasan yang lebih mendasar: nilai *Mean Precision@5* sebesar 0,0921 dan *Mean Recall@5* sebesar 0,0772 mengindikasikan bahwa sistem belum optimal dalam menghasilkan rekomendasi yang tepat sasaran secara kategoris. Fenomena *trade-off* antara *precision* dan *recall* juga teramati secara konsisten, di mana peningkatan nilai *K* memperluas jangkauan *recall* namun menurunkan ketepatan *precision*. Keterbatasan ini secara sistematis bersumber dari tiga faktor utama: ketidakmampuan *TF-IDF* dalam menangkap relasi semantik antarkata, distribusi kategori *dataset* yang tidak proporsional, serta inkonsistensi kualitas deskripsi buku. Penelitian ini dibatasi pada pendekatan *lexical matching* tanpa mempertimbangkan konteks makna dan data interaksi pengguna, sehingga sistem rentan terhadap permasalahan *cold start*. Ke depan, penggantian *TF-IDF* dengan teknik *word embedding* seperti *Word2Vec* atau *BERT*, serta integrasi *hybrid filtering* berbasis *Neural Collaborative Filtering*, direkomendasikan untuk meningkatkan performa sistem secara substansial.

UCAPAN TERIMAKASIH

Kami berterima kasih kepada pembimbing yang telah membantu perencanaan dan pengembangan sistem ini dengan memberikan saran, arahan, dan masukan yang berharga. Selain itu, terima kasih juga kepada para mahasiswa yang telah membantu menguji dan mengevaluasi sistem ini, memastikan sistem berfungsi sesuai rencana. Selain itu, penulis ingin mengucapkan terima kasih kepada semua dosen dan staf akademik yang telah membantu penelitian dengan menyediakan fasilitas dan dukungan ilmiah. Terakhir, penulis ingin menyampaikan rasa terima kasih kepada semua orang yang telah berkontribusi dalam penyelesaian sistem rekomendasi buku ini; tidak semua orang dapat disebutkan secara spesifik. Beliau percaya bahwa kemajuan penelitian dan penggunaan teknologi di sektor terkait akan mendapatkan manfaat dari upaya ini.

DAFTAR PUSTAKA

- [1] Nuiptian and J. Chuaykhun, "Book Recommendation System based on Course Descriptions using Book Recommendation System based on Course Descriptions using *Cosine Similarity*," no. December 2023, 2026, doi: 10.1145/3639233.3639335.
- [2] Z. A. Hamisu and N. Mukhtar, "An Author-Centric Scientific Paper Recommender System to Improve *Content-Based Filtering* Approach," vol. 10, no. 1, pp. 40–49, 2024.
- [3] X. Tian, "*Content-Based Filtering* for Improving Movie Recommender System," no. Dai 2023, 2024, doi: 10.2991/978-94-6463-370-2.
- [4] W. Tafesse, M. P. Aguilar, S. Sayed, and U. Tariq, "Digital overload, coping mechanisms, and student engagement: An empirical investigation based on the SOR framework," *Sage Open*, vol. 14, no. 1, p. 21582440241236090, 2024.
- [5] L. Rosidah and P. Dellia, "Library book recommendation system using *Content-Based Filtering*," *Internet Things Artif. Intell. J.*, vol. 4, no. 1, pp. 42–65, 2024.
- [6] M. Arnold, M. Goldschmitt, and T. Rigotti, "Dealing with information overload: a comprehensive review," *Front. Psychol.*, vol. 14, p. 1122200, 2023.
- [7] Y. Sari, A. R. Baskara, P. B. Prakoso, and N. Royani, "Perbandingan Metode Pembobotan TF-RF dan TF-IDF dengan Dikombinasikan dengan Weighted Tree Similarity untuk Sistem Rekomendasi Buku," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 9, no. 6, 2022.
- [8] S. Rahmadhani, L. Hakim, G. H. Wibowo, S. P. Kristanto, and E. M. Rini, "Sistem Rekomendasi Penelusuran Buku Berbasis *Content-Based Filtering* dengan Pembobotan TF-RF," *J. Inform. Polinema*, vol. 10, no. 4, pp. 491–500, 2024.
- [9] N. Matondang, Y. V. Via, and F. A. Akbar, "Implementasi Algoritma Weighted Tree Similarity Dan Content Based Filtering Dalam Pencarian Skripsi," vol. 12, no. 3, 2024.
- [10] R. Hanun *et al.*, "Implementasi *Content-Based Filtering* Pada Sistem Rekomendasi Buku Perpustakaan," vol. 9, no. 2, pp. 3243–3250, 2025.
- [11] H. A. Rasyid, S. F. Pane, M. Yusril, and H. Setyawan, "Pemetaan Profil Mahasiswa Untuk Memprediksi Peminatan Mahasiswa Data Mahasiswa Setiap Angkatan," vol. 16, no. 1, pp. 1–10, 2023.
- [12] N. Ikhsani, S. Putri, R. Nuraini, and S. Fathonah, "Penerapan Metode Content Based Filtering Dan Knn Pada Aplikasi Rekomendasi Laptop Berbasis Mobile Systematic Literature Review," vol. 7, no. 2, pp. 1229–1236, 2023.
- [13] T. Ridwansyah, B. Subartini, and S. Sylviani, "Penerapan Metode *Content-Based Filtering* pada Sistem Rekomendasi," vol. 4, no. 2, 2024.
- [14] F. Dwi and W. Andriyani, "Pengembangan Sistem Rekomendasi Pembimbing Tugas Akhir Menggunakan Teknik Content Based Filtering," no. 2, pp. 474–483, 2025, doi: 10.26798/jiko.v9i2.1599.
- [15] R. Ardiansyah *et al.*, "Sistem Rekomendasi Buku Perpustakaan Sekolah menggunakan Metode *Content-Based Filtering*," vol. 4, no. 2, pp. 510–517, 2023.
- [16] A. A. Huda, R. Fajarudin, and A. Hadinegoro, "Sistem Rekomendasi *Content-Based Filtering* Menggunakan TF-IDF Vector Similarity Untuk Rekomendasi Artikel Berita," vol. 4, no. 3, pp. 1679–1686, 2022, doi: 10.47065/bits.v4i3.2511.
- [17] N. Azizah, "Sistem Rekomendasi Produk Something Menggunakan Metode *Content-Based Filtering*," vol. 6, no. 3, pp. 461–468, 2024.
- [18] D. R. Pratama, G. S. Nugraha, and R. Dwiyanaputra, "Sistem Rekomendasi Topik Penelitian Menggunakan Metode *Content-Based Filtering* (Studi Kasus Program Studi Teknik Informatika)," vol. x, no. 36, pp. 1–9, 2019.
- [19] U. Javed, K. Shaukat, I. A. Hameed, F. Iqbal, T. M. Alam, and S. Luo, "A Review of Content-Based and Context-Based Recommendation Systems," pp. 274–306, 2021.
- [20] D. Septiani, I. Isabela, and H. A. Ochobi, "Analisis term frequency inverse document frequency (tf-idf) dalam temu kembali informasi pada dokumen teks," vol. 25, pp. 81–88, 2020.
- [21] D. Ridhwanullah, Y. K. Kumarahadi, and B. D. Raharja, "*Content-Based Filtering* pada Sistem Rekomendasi Buku Informatika," *J. Ilm. SINUS*, vol. 22, no. 2, p. 57, 2024.
- [22] A. D. Safitri, V. Atina, A. Farida, and A. D. Safitri, "Sistem rekomendasi buku menggunakan metode *Content-Based Filtering* The book recommended system employs the method of *Content-Based Filtering*," vol. 5, pp. 218–227, 2024, doi: 10.37373/infotech.v5i2.1302.
- [23] F. Wayesa, M. Leranso, G. Asefa, and A. Kediri, "Pattern - based hybrid book recommendation system using semantic relationships," *Sci. Rep.*, pp. 1–12, 2023, doi: 10.1038/s41598-023-30987-0.
- [24] H. Abdollahpouri *et al.*, "Beyond Personalization: Research Directions in Multistakeholder Recommendation," pp. 1–64, 2019