Penerapan TF-RF dan *Cosine Similarity* Untuk Rekomendasi Kursus Online

Frans Damai Zalukhu[1,*], Fajri Profesio Putra[2]

[1, 2] urusan Teknik Informatika, Rekayasa Perangkat Lunak, Politeknik Negeri Bengkalis, Kota Bengkalis, Indonesia

Email: [1,*]fransdamai2020@gmail.com, [2]fajri@polbeng.ac.id

[*)] Email Penulis Utama

Abstrak− Perkembangan platform *e-learning* telah meningkatkan jumlah kursus online yang tersedia, tetapi juga menimbulkan masalah *information overload* dan *choice paradox* sehingga pengguna kesulitan menemukan kursus yang sesuai dengan minat, kebutuhan, dan tingkat kemampuannya. Penelitian ini bertujuan mengembangkan sistem rekomendasi kursus online berbasis *Content-Based Filtering* menggunakan metode *Term Frequency-Relevance Frequency* (TF-RF) dan *Cosine Similarity*. Dataset yang digunakan terdiri atas 8.423 kursus bidang Teknologi Informasi yang diperoleh dari berbagai platform *e-learning*. Tahapan penelitian meliputi pembersihan data, *preprocessing* teks, pelabelan topik, pembobotan TF-RF, penyaringan berdasarkan topik dan tingkat kesulitan, serta perhitungan kemiripan menggunakan *Cosine Similarity*. Sistem menghasilkan 10 rekomendasi kursus yang paling relevan berdasarkan preferensi pengguna tanpa memerlukan data riwayat interaksi. Hasil Evaluasi menunjukkan nilai rata-rata Precision@10 sebesar 0,8750, Recall@10 sebesar 0,2556, dan *F1@10* sebesar 0,3331. Hasil tersebut menunjukkan bahwa kombinasi TF-RF dan *Cosine Similarity* mampu menghasilkan rekomendasi yang relevan serta membantu pengguna mengurangi beban pencarian informasi pada platform *e-learning*.

Kata Kunci: Sistem Rekomendasi, *Content-Based Filtering*, TF-RF, *Cosine Similarity*, Kursus Online

Abstract− *The growth of* e-learning *platforms has increased the number of available online courses, creating challenges such as information overload and choice paradox that make it difficult for users to identify courses matching their interests, learning needs, and skill levels. This study develops an online course recommendation system based on Content-Based Filtering using the Term Frequency-Relevance Frequency (TF-RF) weighting method and Cosine Similarity. The dataset consists of 8,423 Information Technology courses collected from various e-learning platforms. The research process includes data cleaning, text preprocessing, topic labeling, TF-RF weighting, filtering based on topic and difficulty level, and similarity calculation using Cosine Similarity. The proposed system generates the Top-10 most relevant course recommendations without requiring historical user interaction data. The results achieved an average Precision@10 of 0.8750, Recall@10 of 0.2556, and F1@10 of 0.3331. These findings indicate that the combination of TF-RF and Cosine Similarity is effective for providing relevant course recommendations while reducing users' information search effort on* e-learning *platforms.*

Keywords: Recommendation System, Content-Based Filtering, TF-RF, Cosine Similarity, Online Course

1. PENDAHULUAN

Perkembangan teknologi informasi digital telah membawa perubahan yang signifikan dalam dunia pendidikan. Salah satu dampaknya adalah meningkatnya penggunaan platform kursus *online* atau *e-Learning* sebagai sarana belajar yang mudah diakses oleh berbagai kalangan. Pemanfaatan *e-Learning* semakin berkembang sejak masa pandemi. Saat ini, berbagai platform kursus *online* menyediakan beragam materi pembelajaran dalam jumlah yang terus bertambah untuk memenuhi kebutuhan pengguna [1]. Namun, banyaknya pilihan kursus dan informasi yang tersedia sering kali membuat pengguna kesulitan menemukan materi yang paling sesuai dengan minat dan kebutuhannya. Banyaknya pilihan kursus yang tersedia di internet dapat menyebabkan masalah kelebihan informasi (*information overload*). Kondisi ini terjadi ketika pengguna dihadapkan pada terlalu banyak informasi sehingga sulit untuk memproses dan mengevaluasi seluruh pilihan yang ada. Selain itu, banyaknya alternatif yang tersedia juga dapat membuat proses pengambilan keputusan menjadi lebih sulit. Fenomena ini dikenal sebagai *Choice Paradox*, yaitu kondisi ketika semakin banyak pilihan yang tersedia, semakin besar pula kemungkinan pengguna mengalami kebingungan dalam menentukan pilihan yang sesuai [2].

Banyaknya pilihan yang tersedia dapat memberikan dampak negatif terhadap minat pengguna dalam mencari dan memilih kursus. Beberapa penelitian menunjukkan bahwa terlalu banyak alternatif dapat membuat pengguna lebih sulit mengambil keputusan, sehingga mereka cenderung menunda bahkan membatalkan pilihannya [3]. Oleh karena itu, diperlukan sistem rekomendasi yang dapat membantu pengguna menyaring informasi dan menemukan kursus yang relevan. Dengan adanya sistem rekomendasi, proses pencarian kursus

menjadi lebih mudah, waktu yang dibutuhkan untuk memilih kursus dapat berkurang, dan pengguna dapat memperoleh rekomendasi yang lebih sesuai dengan preferensi mereka. Untuk membantu mengatasi permasalahan banyaknya pilihan kursus pada platform pembelajaran online, penelitian ini mengembangkan sistem rekomendasi berbasis konten (*Content-Based Filtering*) yang memanfaatkan metode pembobotan teks *Term Frequency-Relevance Frequency* (TF-RF) serta pengukuran tingkat kemiripan menggunakan *Cosine Similarity*.

Pendekatan *Content-Based Filtering* ini merekomendasikan item berdasarkan karakteristik dan preferensi pengguna [4]. Kemudian TF-RF adalah sebuah metode pembobotan kata atau teks berbasis *supervised term weighting* yang digunakan dalam klasifikasi dan analisis teks. Metode ini menggabungkan frekuensi kemunculan kata di suatu dokumen (*Term Frequency*/TF) dengan tingkat relevansi kata tersebut di seluruh kategori dokumen (*Relevance Frequency*/RF) yang bertujuan untuk meningkatkan kinerja penentuan bobot kata dibandingkan dengan metode lain. Metode ini mempertimbangkan relevansi suatu dokumen dengan melihat frekuensi kemunculan istilah-istilah dalam kategori yang terkait [5]. *Cosine Similarity* adalah salah satu metrik yang digunakan untuk menghitung tingkat kesamaan (*similarity score*) antara dua objek yang direpresentasikan dalam bentuk vektor [6].

Beberapa penelitian terdahulu juga membahas penerapan *Content-Based Filtering* dan *Cosine Similarity* untuk Sistem Rekomendasi. Salah satu contohnya adalah penelitian oleh [7]. Penelitian ini mengembangkan sistem rekomendasi destinasi wisata berbasis *Content-Based Filtering* dengan pendekatan TF-IDF dan *Cosine Similarity*. Penelitian ini menunjukkan bahwa metode *Content-Based Filtering* efektif dalam memberikan rekomendasi wisata yang personal dan relevan. Kemudian penelitian oleh [8] yang membangun sistem rekomendasi film berbasis *Conten-Based Filtering* dengan pendekatan TF-IDF dan *Cosine Similarity*. Penelitian in menunjukkan bahwa *Cosine Similarity* secara efektif mengukur tingkat kesamaan antar film. Kemudian Penelitian oleh [9] mengembangkan sistem rekomendasi pencarian buku berbasis *Content-Based Filtering* dengan *TF-RF* dan *Cosine Similarity*, dan berhasil mencapai nilai *precision* rata-rata sebesar 86%. Penelitian ini menyimpulkan bahwa implementasi metode *Content-Based Filtering* dan TF-RF dapat meningkatkan performa sistem rekomendasi pencarian buku secara signifikan [9].

Penelitian oleh [9] menggunakan teknik *stemming* pada tahap *preprocessing*, namun disarankan untuk mempertimbangkannya kembali karena dapat mengubah kata menjadi bentuk dasar dan mengurangi representasi informasi pada judul. Berdasarkan hal tersebut, penelitian ini tidak menerapkan *stemming*. Keputusan ini menyesuaikan karakteristik dataset kursus Teknologi Informasi yang banyak memuat istilah teknis, bahasa pemrograman, *framework*, dan teknologi (misalnya *Machine Learning*, *Flutter*, *Unreal Engine*). Penerapan *stemming* berpotensi mengubah atau menghilangkan istilah tersebut sehingga memengaruhi pembobotan TF-RF dan perhitungan *Cosine Similarity*. Oleh karena itu, bentuk asli kata dipertahankan agar informasi penting pada metadata kursus tetap terjaga dalam proses ekstraksi fitur.

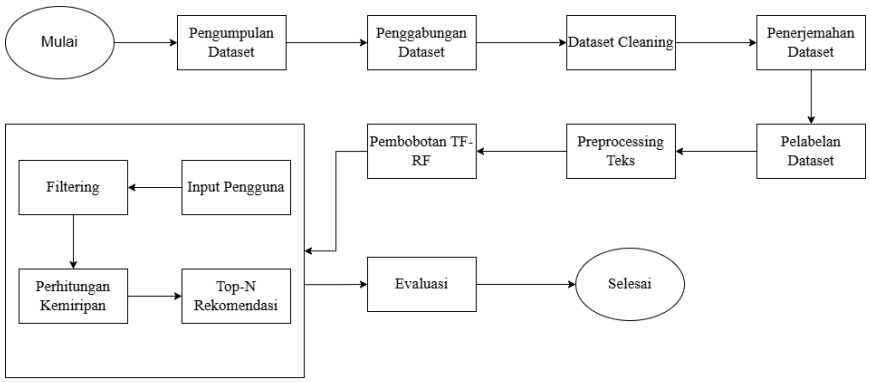
Berbeda dengan penelitian [9] yang hanya menggunakan metadata judul, penelitian ini mengombinasikan judul dan deskripsi kursus untuk membangun representasi dokumen. Penggunaan metadata berupa judul dan deskripsi menghasilkan representasi dokumen yang lebih lengkap karena informasi mengenai materi pembelajaran, teknologi yang digunakan, serta kompetensi yang diperoleh tidak hanya terdapat pada judul, tetapi juga dijelaskan pada deskripsi kursus. Oleh karena itu, proses pembobotan TF-RF dapat memanfaatkan informasi tekstual yang lebih lengkap dibandingkan jika hanya menggunakan metadata judul.

Penelitian ini menerapkan tahap penyaringan (*filtering*) berdasarkan topik dan tingkat kesulitan kursus sebelum proses pembobotan menggunakan TF-RF dan perhitungan *Cosine Similarity*. Tahap penyaringan ini dilakukan untuk mengurangi jumlah kandidat kursus yang diproses sehingga sistem dapat lebih fokus pada kursus yang sesuai dengan preferensi pengguna. Dataset kursus dalam penelitian ini telah dikelompokkan ke dalam beberapa kategori, seperti *Programming Fundamentals*, *Web Development*, dan *Data Science & AI*. Informasi kategori tersebut menjadi salah satu alasan penggunaan metode TF-RF dalam penelitian ini, karena metode tersebut memanfaatkan informasi label kategori atau topik dalam proses pembobotan kata. Berbeda dengan TF-RF, metode TF-IDF umumnya bekerja tanpa menggunakan informasi kategori dokumen (*unsupervised*) dan hanya mempertimbangkan frekuensi kemunculan kata dalam dokumen. Kondisi ini menyebabkan kata yang sering muncul pada banyak dokumen memiliki nilai IDF yang lebih rendah, sehingga bobot kata tersebut dapat berkurang meskipun kata tersebut masih berkaitan dengan suatu topik tertentu. Sementara itu, TF-RF mempertimbangkan perbandingan kemunculan kata pada suatu kategori dengan kategori lainnya (*supervised*), sehingga informasi kategori dapat digunakan dalam menentukan bobot fitur [10]. Pemanfaatan label kategori pada TF-RF memungkinkan proses pembobotan mempertimbangkan distribusi kata antar kategori dokumen, sehingga fitur yang diperoleh tidak hanya berdasarkan frekuensi kemunculan kata, tetapi juga berdasarkan relevansinya terhadap kategori tertentu.

Berdasarkan kajian penelitian terdahulu, sebagian besar sistem rekomendasi berbasis *Content-Based Filtering* masih menggunakan TF-IDF sebagai metode pembobotan teks. Penelitian yang menerapkan TF-RF pada sistem rekomendasi, khususnya untuk rekomendasi kursus online yang memanfaatkan cakupan informasi yang lebih luas serta penyaringan berdasarkan topik dan tingkat kesulitan, masih relatif terbatas. Oleh karena itu, penelitian ini mengeksplorasi penerapan TF-RF pada karakteristik dataset tersebut.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan membangun sistem rekomendasi kursus online berbasis *Content-Based Filtering* dengan menerapkan metode pembobotan TF-RF dan *Cosine Similarity*. Sistem yang dikembangkan diharapkan mampu memberikan rekomendasi kursus yang relevan sesuai dengan preferensi pengguna.

2. METODE PENELITIAN



Gambar 1. Tahapan Penelitian

2.1 Pengumpulan Dataset

Pengumpulan data dilakukan dengan menggunakan teknik *Web Scraping*. *Web scraping* merupakan teknologi yang memungkinkan pengambilan data terstruktur dari dokumen berbasis teks, seperti HTML [11]. *Scraping* dilakukan menggunakan Python dengan memanfaatkan *library BeautifulSoup* untuk mengekstraksi elemen HTML, dan *Selenium* dipakai untuk mengambil data dari halaman dinamis. Dataset yang diperoleh mencakup 8.423 kursus IT dari berbagai platform e-learning. Data kursus platform Udemy didapat dari situs *Kaggle*, sementara data kursus dari platform lain dikumpulkan melalui *web scraping*.

Tabel 1. Dataset Penelitian

No	Data	Jumlah
1	Udemy	4241
2	Coursera	2609
3	Udacity	452
4	Dicoding	86
5	BuildWithAngga	493
6	Codepolitan	119
7	Brainmatics	220
8	Multimatics	132
9	Inixindo	71
Jumlah		8423

2.2 Penggabungan Dataset

Dataset dalam penelitian ini berasal dari beberapa sumber yang berbeda. Untuk memastikan konsistensi dan kemudahan pengolahan, seluruh dataset terlebih dahulu digabungkan menjadi satu dataset terpadu. Proses penggabungan dilakukan dengan menyelaraskan struktur kolom dan format data sehingga seluruh kursus dapat dianalisis sebagai satu kesatuan data penelitian.

2.3 Pembersihan Dataset (Data Cleaning)

Dataset Cleaning dilakukan untuk memastikan kualitas dataset kursus yang digunakan dalam penelitian. Pembersihan data (*data cleaning*) adalah proses mengidentifikasi dan memperbaiki data yang rusak, duplikat, tidak lengkap, tidak akurat, atau mengandung *noise* dalam suatu dataset, dengan tujuan meningkatkan kualitas data tersebut [11]. Tahap ini mencakup penghapusan duplikat, penanganan data kosong atau *missing values*, dan penghapusan atribut yang tidak relevan. Hasil akhirnya, dataset hanya menyisakan atribut penting yaoitu judul, deskripsi, topik/kategori, dan tingkat kesulitan.

2.4 Penerjemahan Dataset

Dataset yang digunakan dalam penelitian ini berasal dari berbagai sumber dengan penggunaan bahasa yang tidak seragam. Sebagian besar data tersedia dalam bahasa Inggris, namun masih terdapat sejumlah metadata yang menggunakan bahasa Indonesia. Untuk menjaga konsistensi representasi teks, penelitian ini menerapkan tahap standarisasi bahasa dengan menyeragamkan seluruh metadata kursus ke dalam bahasa Inggris.

2.5 Pelabelan Dataset

Pelabelan dataset dilakukan untuk mengelompokkan kursus ke dalam topik atau kategori sesuai judul dan deskripsi, sehingga kategori yang terbentuk merepresentasikan bidang pembelajaran dan memudahkan penyaringan sesuai preferensi pengguna. Dalam penelitian ini, kursus dikelompokkan ke topik teknologi informasi seperti *Programming*, *Data Science*, *Cyber Security*, *Web Development*, *Mobile Development*, dan lainnya. Hasil pelabelan digunakan sebagai salah satu atribut dalam proses rekomendasi untuk memastikan bahwa kursus yang direkomendasikan memiliki kesesuaian topik dengan minat pengguna.

2.6 Tahap Preprocessing Text

Tahap *Preprocessing* dilakukan untuk mempersiapkan data teks agar lebih terstruktur dan siap digunakan pada proses pembobotan TF-RF. Tahap ini bertujuan untuk mengurangi *noise* serta meningkatkan kualitas representasi teks yang terdapat pada judul dan deskripsi kursus. Proses *preprocessing* meliputi *Case Folding* untuk mengubah seluruh huruf menjadi huruf kecil, *Tokenization* untuk memisahkan teks menjadi kumpulan kata, *Stopword Removal* untuk menghapus kata-kata umum yang tidak memiliki kontribusi signifikan terhadap makna dokumen, kemudian pembersihan tanda baca, simbol khusus, angka yang tidak relevan [12]. Implementasi dilakukan dengan Python menggunakan *library* re untuk pembersihan teks, pandas untuk pengolahan dataset, serta NLTK untuk tokenisasi dan penghapusan *stopwords*.

2.7 Pembobotan TF-RF

Pembobotan TF-RF dilakukan untuk mengubah data teks hasil *preprocessing* menjadi representasi numerik yang dapat digunakan dalam proses perhitungan kemiripan. Metode TF-RF (*Term Frequency–Relevance Frequency*) merupakan kombinasi antara *Term Frequency* (TF) dan *Relevance Frequency* (RF) untuk mengevaluasi relevansi dokumen dengan mempertimbangkan seberapa sering kata muncul dalam kategori yang bersangkutan [9] [13]. Tahap pembobotan dilakukan dengan Python menggunakan *scikit-learn* melalui *CountVectorizer* untuk membentuk matriks TF. Bobot RF dihitung dengan *NumPy*, lalu dibentuk menjadi matriks diagonal menggunakan *SciPy*. Matriks TF kemudian dikalikan dengan matriks RF sehingga menghasilkan matriks TF-RF, yang digunakan sebagai representasi dokumen dalam perhitungan kemiripan dengan *Cosine Similarity*.

TF-RF dihitung dengan langkah berikut:

1. *Relevance Frequency* (RF):

$$\mathrm{RF(t)} = \log\left(2 + \frac{b}{max(1, c)}\right) \quad (1)$$

2. *Term Frequency – Relevance Frequency* (TF-RF):

$$TF\text{-}RF(t,d) = TF(d,t) \times RF \quad (2)$$

Keterangan:

TF = frekuensi kemunculan istilah t pada dokumen (d)

RF = relevansi istilah t dalam seluruh koleksi pada dokumen

t = istilah atau kata kunci yang sedang dihitung bobotnya

d = dokumen yang sedang dievaluasi

b = jumlah dokumen yang mengandung istilah atau kata kunci (*term*)

c = jumlah dokumen yang tidak mengandung istilah atau kata kunci (*term*)

2.9 Proses Rekomendasi

2.9.1 Input Pengguna

Pada tahap ini, pengguna memasukkan preferensi berupa kata kunci, topik, dan tingkat kesulitan sebagai dasar rekomendasi. Kata kunci menggambarkan materi yang ingin dipelajari, sedangkan topik dan tingkat kesulitan membatasi ruang pencarian sesuai kebutuhan. Input pengguna diproses dengan tahapan *preprocessing* yang sama seperti data kursus agar representasi teks konsisten. Representasi tersebut kemudian digunakan dalam pembobotan dan perhitungan kemiripan untuk menghasilkan rekomendasi yang relevan.

2.9.2 Proses Filtering

Setelah pengguna memasukkan preferensi, sistem melakukan *filtering* berdasarkan topik dan tingkat kesulitan untuk membatasi kandidat kursus. Hanya kursus yang sesuai kriteria dipertimbangkan, lalu sistem menghitung *Cosine Similarity* antara preferensi pengguna dan kandidat terpilih guna menentukan urutan rekomendasi berdasarkan tingkat kemiripan.

2.9.3 Perhitungan Kemiripan

Pada tahap ini perhitungan kemiripan atau menghitung tingkat kesesuaian antara preferensi pengguna dan data kursus menggunakan metode *Cosine Similarity*. *Cosine Similarity* merupakan metode untuk menghitung tingkat kemiripan antara dua vektor dengan cara membagi hasil perkalian titik (*dot product)* kedua vektor dengan perkalian panjang (magnitudo) masing-masing vektor. Secara umum, jika nilai *cosine similarity* antara dua item adalah 1, maka kedua item tersebut dianggap sangat mirip. Metode ini digunakan untuk mengukur kesamaan antara dua objek dalam berbagai jenis data [14]. Implementasi dilakukan menggunakan bahasa pemrograman Python dengan memanfaatkan fungsi *cosine_similarity* dari pustaka *scikit-learn*.

$$\text{Cosine Similarity}\,(A, B) = \frac{A \,.\, B}{\|A\| \times \|B\|} \tag{3}$$

2.9.4 Top-N Rekomendasi

Setelah nilai *Cosine Similarity* dihitung, sistem mengurutkan kandidat kursus dari yang paling miri hingga paling rendah, lalu menampilkan 10 kursus teratas (Top-10) sebagai rekomendasi. Setiap rekomendasi memuat informasi kursus seperti judul, topik, tingkat kesulitan, platform, dan tautan.

2.10 Evaluasi

Dalam sistem rekomendasi, evaluasi performa sangatlah krusial untuk mengukur seberapa efektif sistem tersebut dalam menyajikan rekomendasi yang relevan bagi pengguna. Metrik evaluasi digunakan untuk mengukur akurasi dari rekomendasi yang dihasilkan. Untuk penelitian ini, metrik yang digunakan adalah *Precision@K*, *Recall@K*, dan *F1@K*.

2.10.1 Precision@K

Precision@K menunjukkan proporsi item yang diprediksi dengan benar terhadap jumlah total prediksi (K) [15].

$$Precision = \frac{\text{TP@K}}{\text{TP@K} + \text{FP@K}} \tag{4}$$

2.10.2 Recall@K

Recall@K merepresentasikan proporsi item yang berhasil diprediksi dengan benar terhadap jumlah item *ground-truth* [15].

$$Recall = \frac{\text{TP@K}}{\text{TP@K} + \text{FN@K}} \tag{5}$$

2.10.3 F1@K

F1@K menggunakan rata-rata harmonik dari *Precision@K* dan *Recall@K* [15].

$$\text{F1} = \frac{2 \times Precision@K \times Recall@K}{Precision@K + Recall@K} \tag{6}$$

3. HASIL DAN PEMBAHASAN

3.1 Hasil Rekomendasi

Sistem menghasilkan rekomendasi kursus yang diurutkan berdasarkan nilai kemiripan tertinggi. Sistem menampilkan 10 kursus teratas yang dianggap paling relevan dengan preferensi pengguna.

Tabel 2. Hasil Rekomendasi

Judul Kursus	Topic	Level	Similarity
React Three Fiber	Web Development	Beginner	0.5933

React Styled Components Course (V5)	Web Development	All Levels	0.3977
React - testing	Web Development	Beginner	0.3642
React, React Redux and Redux Saga - Master React State/Hooks	Web Development	All Levels	0.3427
MobX In Depth With React(Hooks+TypeScript)	Web Development	All Levels	0.3063
ReactJs - The Complete ReactJs Course For Beginners	Web Development	Beginner	0.2815
Complete DApp - Solidity & React - Blockchain Development	Web Development	All Levels	0.2705
Hello React - React Training for JavaScript Beginners	Web Development	All Levels	0.2680
Learn Complete Front-End Web Development Course	Web Development	All Levels	0.2638
React Js With Laravel Build Complete PWA Ecommerce Project	Web Development	All Levels	0.2613

Berdasarkan implementasi perhitungan yang telah dilakukan, Tabel 2 menampilkan hasil akhir (*output*) dari mesin rekomendasi. Pada skenario ini, disimulasikan sebuah pencarian di mana pengguna memasukkan parameter Topik: Web Development, Level: Beginner, dan Kata Kunci: "React".

Dari hasil rekomendasi yang ditampilkan, dapat diambil beberapa kesimpulan mengenai kinerja algoritma sistem.

1. Sistem mampu menghitung tingkat kemiripan antara preferensi pengguna dan konten kursus, kemudian mengurutkan hasil rekomendasi dari nilai *Cosine Similarity* tertinggi hingga terendah. Kursus *"React Three Fiber"* berada pada peringkat pertama dengan nilai kemiripan tertinggi sebesar 0,593, sehingga dapat dianggap sebagai kursus yang paling sesuai dengan kebutuhan pengguna.
2. Sistem berhasil menerapkan penyaringan berdasarkan topik dan tingkat kesulitan sebelum proses perhitungan kemiripan dilakukan. Hal ini terlihat dari seluruh rekomendasi yang dihasilkan yang hanya berasal dari topik "Web Development" serta sesuai dengan tingkat kesulitan "Beginner" dan "All Levels".
3. Hasil rekomendasi menunjukkan tingkat kesesuaian yang tinggi terhadap kata kunci yang dimasukkan pengguna. Semua kursus yang direkomendasikan mengandung kata kunci *React* atau teknologi yang masih berada dalam ekosistem yang sama, seperti *React Redux*, *React Native*, dan *ReactJS* meskipun nilai *Cosine Similarity* berada pada rentang 0,255–0,593.

Secara keseluruhan, hasil pengujian menunjukkan bahwa metode yang diterapkan pada penelitian ini dapat digunakan untuk membangun sistem rekomendasi kursus berbasis *Content-Based Filtering* yang menghasilkan rekomendasi berdasarkan topik, tingkat kesulitan, dan kata kunci yang dimasukkan oleh pengguna.

3.2 Evaluasi Sistem

Tahap evaluasi dilakukan untuk memastikan bahwa sistem rekomendasi yang dikembangkan dapat berjalan dengan baik serta memiliki performa yang terukur. Evaluasi dilakukan dengan mengukur performa model dengan metrik *Precision@K, Recall@K* dan *F1@K*.

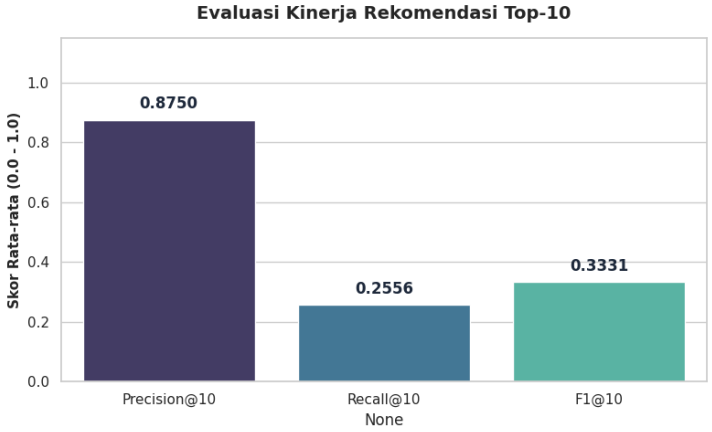
Tabel 3. Hasil Evaluasi Model Rekomendasi

No	Keyword	Total Relevan	Top-K	Hasil Relevan	Precision@10	Recall@10	F1@10
1	Python	435	10	10	1.0000	0.0230	0.0449
2	Java	254	10	10	1.0000	0.0394	0.0758
3	Docker	53	10	10	1.0000	0.1887	0.3175
4	Selenium	58	10	10	1.0000	0.1724	0.2941
5	Javascript	30	10	10	1.0000	0.3333	0.5000
6	Flutter	35	10	10	1.0000	0.2857	0.4444
7	Wireframe	7	10	7	0.7000	1.0000	0.8235
8	PHP	13	10	10	1.0000	0.7692	0.8696
9	SQL	147	10	10	1.0000	0.0680	0.1274
10	HTML	306	10	10	1.0000	0.0327	0.0633
11	Machine Learning	296	10	9	0.9000	0.0304	0.0588

12	Unreal Engine	54	10	10	1.0000	0.1852	0.3125
13	Ethical Hacking	20	10	10	1.0000	0.5000	0.6667
14	Web Design	75	10	9	0.9000	0.1200	0.2118
15	Data Science	135	10	10	1.0000	0.0741	0.1379
16	Mobile App	32	10	6	0.6000	0.1875	0.2857
17	Cloud Computing	14	10	7	0.7000	0.5000	0.5833
18	Software Testing	42	10	9	0.9000	0.2143	0.3462
19	Game Design	29	10	6	0.6000	0.2069	0.3077
20	SQL Database	11	10	2	0.2000	0.1818	0.1905

3.2.1 Analisis Hasil Metrik Evaluasi

Analisis dilakukan untuk memahami performa sistem rekomendasi, mengidentifikasi faktor yang memengaruhi hasil evaluasi, serta menjelaskan kelebihan dan keterbatasan sistem yang dikembangkan. Berdasarkan hasil evaluasi, sistem rekomendasi memperoleh nilai rata-rata *Precision@10* sebesar 0,8750, *Recall@10* sebesar 0,2556, dan *F1@10* sebesar 0,3331.



Gambar 2. Grafik Rata-Rata Metrik

a. Analisis *Precision@10*

Nilai rata-rata *Precision@10* sebesar 0,8750 menunjukkan bahwa sebagian besar kursus yang muncul pada sepuluh rekomendasi teratas memenuhi kriteria relevansi yang digunakan pada penelitian ini. Hasil tersebut menunjukkan bahwa kombinasi pembobotan TF-RF dan perhitungan *Cosine Similarity* mampu mengurutkan dokumen yang memiliki karakteristik teks paling mirip dengan kueri pengguna ke posisi teratas. Pembobotan TF-RF menghasilkan representasi fitur berdasarkan distribusi istilah pada setiap kategori, sedangkan *Cosine Similarity* digunakan untuk mengukur tingkat kemiripan antara vektor kueri dan vektor dokumen sehingga proses pemeringkatan dapat dilakukan berdasarkan nilai kemiripan tersebut.

b. Analisis *Recall@10*

Nilai rata-rata *Recall@10* sebesar 0,2556 menunjukkan bahwa sistem hanya berhasil merekomendasikan sebagian dari seluruh dokumen relevan yang terdapat pada *ground truth*. Nilai tersebut dipengaruhi oleh mekanisme evaluasi Top-10, karena sistem hanya menampilkan sepuluh dokumen dengan nilai kemiripan tertinggi. Akibatnya, dokumen relevan yang berada di luar sepuluh peringkat teratas tidak ikut direkomendasikan dan dihitung sebagai *False Negative* pada proses evaluasi.

c. Pengaruh Jumlah *Ground Truth* terhadap *Recall*

Nilai Recall@10 juga dipengaruhi oleh jumlah dokumen relevan pada setiap skenario pengujian. Kata kunci seperti “Wireframe”, “PHP”, dan “Cloud Computing” memiliki jumlah *ground truth* yang relatif sedikit sehingga proporsi dokumen relevan yang berhasil direkomendasikan menjadi lebih besar. Sebaliknya, kata kunci seperti “Python”, “HTML”, dan “Machine Learning” memiliki jumlah *ground truth* yang jauh lebih banyak. Meskipun sistem tetap mampu menampilkan sepuluh rekomendasi yang relevan, proporsi dokumen relevan yang berhasil direkomendasikan menjadi lebih kecil karena jumlah hasil yang ditampilkan dibatasi pada sepuluh dokumen.

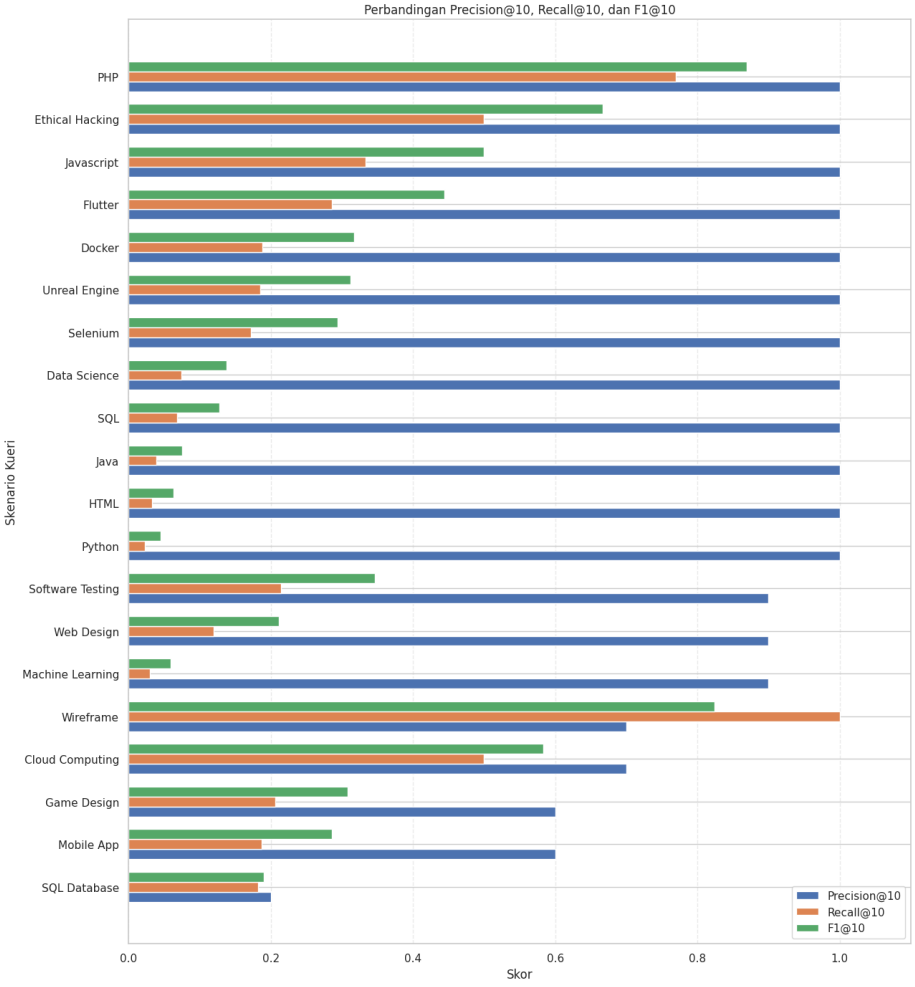
d. Pengaruh Distribusi Dataset Terhadap Evaluasi

Perbedaan jumlah kursus pada setiap topik juga memengaruhi hasil evaluasi. Dataset pada penelitian ini memiliki distribusi topik yang tidak seimbang sehingga jumlah dokumen relevan pada setiap skenario pengujian berbeda-beda. Kondisi tersebut terutama memengaruhi nilai *Recall@10*, karena semakin banyak dokumen yang termasuk dalam *ground truth*, semakin kecil proporsi dokumen yang dapat direkomendasikan ketika sistem hanya menghasilkan rekomendasi Top-10. Dengan demikian, variasi nilai *recall* pada setiap skenario tidak hanya dipengaruhi oleh metode yang digunakan, tetapi juga oleh karakteristik distribusi dataset.

e. Analisis *F1@10*

Nilai rata-rata *F1@10* sebesar 0,3331 merupakan hasil keseimbangan antara *Precision@10* dan *Recall@10*. Meskipun nilai *precision* relatif tinggi, nilai *recall* yang lebih rendah menyebabkan nilai F1@10 menjadi lebuh rendah. Hal ini menunjukkan bahwa sistem lebih efektif dalam menjaga relevansi rekomendasi yang ditampilkan dibandingkan dalam mencakup seluruh dokumen yang dianggap relevan pada *ground truth*.

3.2.2 Analisis Performa Berdasarkan Skenario Kueri



Gambar 3. Grafik Perbandingan Metrik per Skenario Kueri

Gambar 3 menunjukkan distribusi performa sistem rekomendasi pada 20 skenario kueri yang diurutkan berdasarkan nilai *Precision@10*. Berdasarkan visualisasi tersebut, terlihat bahwa performa sistem tidak selalu sama pada setiap kueri. Perbedaan performa dipengaruhi oleh beberapa faktor, terutama jumlah kursus yang tersedia pada suatu topik dan tingkat kejelasan kata kunci yang digunakan pengguna.

a. *Precision@10* Tinggi pada Topik dengan Jumlah Data Besar

Pada beberapa kueri seperti ‘Javascript’, ‘HTML’, ‘Java’, dan ‘Python’, sistem berhasil memperoleh nilai *Precision@10* sebesar 1,0. Hasil ini menunjukkan bahwa seluruh kursus yang direkomendasikan pada posisi Top-10 sesuai dengan topik yang dicari pengguna. Namun, meskipun nilai *precision* sangat

tinggi, nilai *Recall@10* pada kueri-kueri tersebut relatif rendah. Kondisi ini terjadi karena topik-topik tersebut memiliki jumlah kursus yang sangat banyak di dalam dataset. Sistem dapat dengan mudah menemukan 10 kursus yang relevan, tetapi masih terdapat banyak kursus relevan lainnya yang tidak ditampilkan karena sistem hanya memberikan rekomendasi sebanyak 10 kursus (*Top-10 recommendation*). Akibatnya, *precision* tetap tinggi, sedangkan *recall* menjadi rendah karena hanya sebagian kecil dari seluruh kursus relevan yang berhasil direkomendasikan.

b. Performa Optimal pada Topik yang Lebih Spesifik

Performa terbaik sistem terlihat pada kueri seperti ‘PHP’ dan ‘Ethical Hacking’. Pada kedua kueri tersebut, sistem tidak hanya memperoleh nilai *Precision@10* yang tinggi, tetapi juga menghasilkan nilai *Recall* dan *F1-Score* yang lebih baik dibandingkan sebagian besar kueri lainnya. Hal ini menunjukkan bahwa metode TF-RF bekerja relatif lebih baik ketika digunakan pada topik yang lebih spesifik dan memiliki kata kunci yang jelas. Jumlah kursus yang relevan pada topik tersebut relatif lebih sedikit dibandingkan topik populer seperti “Python” atau “Java”. Oleh karena itu, rekomendasi yang diberikan mampu mencakup proporsi yang lebih besar dari keseluruhan kursus relevan yang tersedia, sehingga menghasilkan keseimbangan yang lebih baik antara *precision* dan *recall*.

c. Analisis Performa Rendah pada Kueri "SQL Database"

```
MENCARI REKOMENDASI...
   Topik    : Database
   Level    : Intermediate
   Keyword : SQL Database
------------------------------
```

	Title	Topic	Level	Similarity
2314	The Comprehensive SQL Course	Database	All Levels	0.428298
3369	A Beginners Guide To SQL. Master SQL Quickly	Database	All Levels	0.368271
1683	SQL Course For Beginners: Learn SQL Using MySQ...	Database	All Levels	0.359186
4193	The Complete MySQL Bootcamp: Go from Beginner ...	Database	All Levels	0.339991
1563	Learn DBT from Scratch	Database	Intermediate	0.333985
1901	Advanced SQL for Data Engineering	Database	Intermediate	0.331180
842	Entity Framework in Depth: The Complete Guide	Database	Intermediate	0.329011
1422	The Complete SQL Course 2024 - Learn by Doing ...	Database	All Levels	0.324774
2095	Complete MySQL Database Administration Course	Database	All Levels	0.320694
3919	The Ultimate SQL Bootcamp : Go From Zero to Hero	Database	All Levels	0.316928

Gambar 4. Contoh Hasil Rekomendasi Skenario Kueri SQL Database

Pada skenario kueri “SQL Database”, sistem memperoleh nilai *Precision@10* sebesar 0,20 karena hanya dua dari sepuluh rekomendasi yang memenuhi kriteria *ground truth*. *Ground truth* pada penelitian ini didefinisikan secara ketat, yaitu kursus harus berada pada topik “Database”, memenuhi tingkat kesulitan “Intermediate” atau “All Levels”, serta mengandung kata kunci yang sesuai dengan kueri. Sementara itu, metode TF-RF dan *Cosine Similarity* merepresentasikan kueri sebagai kumpulan istilah (*bag-of-words*), sehingga kata “SQL” dan “Database” diproses sebagai fitur yang terpisah. Akibatnya, beberapa kursus yang berkaitan dengan salah satu istilah, seperti kursus mengenai MySQL, DBT, atau *Entity Framework*, masih memperoleh nilai kemiripan yang cukup tinggi meskipun tidak sepenuhnya sesuai dengan definisi relevansi yang digunakan pada evaluasi. Kondisi tersebut menunjukkan bahwa pada kueri yang terdiri atas lebih dari satu istilah, beberapa dokumen masih memperoleh nilai kemiripan yang tinggi karena memiliki istilah yang sama dengan kueri, meskipun tidak memenuhi seluruh kriteria relevansi yang digunakan dalam *ground truth*., sehingga nilai *Precision@10* menjadi lebih rendah.

3.3 Kelebihan dan Keterbatasan Sistem

3.3.1 Kelebihan Sistem

Sistem rekomendasi yang dikembangkan memiliki beberapa keunggulan dalam membantu proses pencarian dan pemilihan kursus yang relevan.

a. Pembobotan Mempertimbangkan Informasi Topik atau Kategori

Metode TF-RF memanfaatkan informasi kategori topik dalam proses pembobotan sehingga bobot setiap istilah ditentukan berdasarkan distribusi kemunculannya pada kategori dokumen. Dengan pendekatan

tersebut, proses representasi fitur tidak hanya mempertimbangkan frekuensi kemunculan istilah, tetapi juga keterkaitannya dengan topik tertentu.

b. Perhitungan Kemiripan Berdasarkan Representasi Teks

Cosine Similarity digunakan untuk menghitung tingkat kemiripan antara kueri pengguna dan dokumen kursus berdasarkan representasi vektor hasil pembobotan TF-RF. Perhitungan ini membandingkan kesamaan distribusi istilah pada dokumen sehingga rekomendasi diurutkan berdasarkan nilai kemiripan yang diperoleh.

c. Tidak Bergantung pada Riwayat Interaksi Pengguna

Sistem menggunakan pendekatan *Content-Based Filtering* sehingga rekomendasi dihasilkan berdasarkan karakteristik konten kursus dan preferensi yang dimasukkan pengguna. Dengan demikian, sistem dapat memberikan rekomendasi tanpa memerlukan data historis, seperti rating atau riwayat penggunaan.

d. Penyaringan Berdasarkan Preferensi Pengguna

Sistem menerapkan penyaringan berdasarkan topik atau kategori dan tingkat kesulitan atau *level* sebelum proses perhitungan kemiripan. Tahap ini membatasi kandidat kursus yang diproses sehingga perhitungan hanya dilakukan pada dokumen yang sesuai dengan preferensi pengguna.

3.3.2 Keterbatasan Sistem

Meskipun mampu menghasilkan rekomendasi yang relevan, sistem yang dikembangkan masih memiliki beberapa keterbatasan yang ditemukan selama proses evaluasi.

a. Nilai *Recall* yang Relatif Rendah pada Pendekatan *Top-10 Recommendation*

Sistem hanya menampilkan sepuluh rekomendasi dengan nilai kemiripan tertinggi. Akibatnya, sebagian dokumen yang memenuhi kriteria ground truth tidak ditampilkan sehingga nilai *Recall@10* menjadi lebih rendah, terutama pada topik yang memiliki jumlah dokumen relevan cukup banyak.

b. Penggunaan Satu Label Topik atau Kategori untuk Setiap Kursus

Pada penelitian ini, setiap kursus hanya dikelompokkan ke dalam satu kategori topik atau kategori. Padahal, beberapa kursus dapat mencakup lebih dari satu bidang pembelajaran. Kondisi ini berpotensi menyebabkan beberapa kursus yang sebenarnya relevan tidak masuk ke dalam proses rekomendasi karena berada pada kategori yang berbeda.

c. Distribusi Data Antar Topik atau Kategori Tidak Seimbang

Jumlah kursus pada setiap kategori topik tidak merata sehingga jumlah dokumen relevan pada setiap skenario pengujian berbeda. Kondisi ini memengaruhi nilai metrik evaluasi, khususnya *Recall@10*, karena proporsi dokumen yang berhasil direkomendasikan berbeda pada setiap topik.

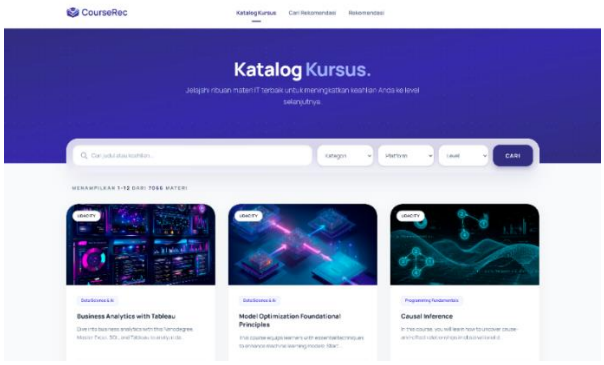
d. Representasi Berbasis *Bag-of-Words*

TF-RF dan *Cosine Similarity* menggunakan representasi berbasis *bag-of-words* yang memperlakukan setiap istilah sebagai fitur yang berdiri sendiri. Pada kueri yang terdiri atas lebih dari satu istilah, pendekatan ini masih dapat menghasilkan dokumen dengan nilai kemiripan tinggi meskipun tidak memenuhi seluruh kriteria relevansi yang digunakan pada *ground truth*, seperti yang terlihat pada skenario kueri "SQL Database".

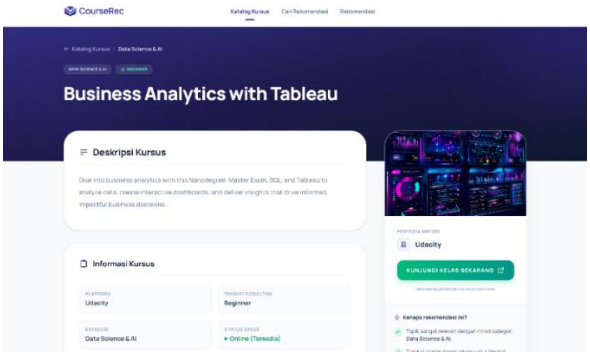
3.4 Implementasi Pada Website

Sistem pada penelitian ini diimplementasikan dalam bentuk aplikasi web menggunakan framework Laravel sebagai antarmuka pengguna (frontend dan backend aplikasi) serta framework Flask untuk menyediakan layanan API yang menangani proses pembobotan TF-RF dan perhitungan *Cosine Similarity*.

a. Halaman Katalog dan Halaman Detail Kursus
Halaman katalog kursus menampilkan seluruh daftar kursus IT dalam sistem, dilengkapi fitur penyaringan berdasarkan topik dan platform untuk memudahkan pencarian. Setiap kursus ditampilkan dalam kartu informasi berisi judul, tingkat kesulitan, deskripsi singkat, dan platform penyedia. Pada halaman detail, tersedia tombol tautan yang mengarahkan pengguna langsung ke platform kursus.



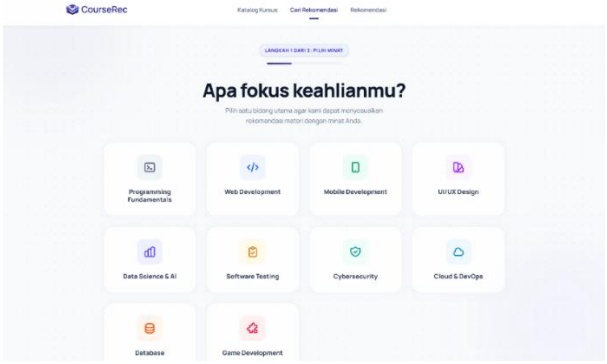
Gambar 5. Halaman Katalog Kursus



Gambar 6. Halaman Detail Kursus

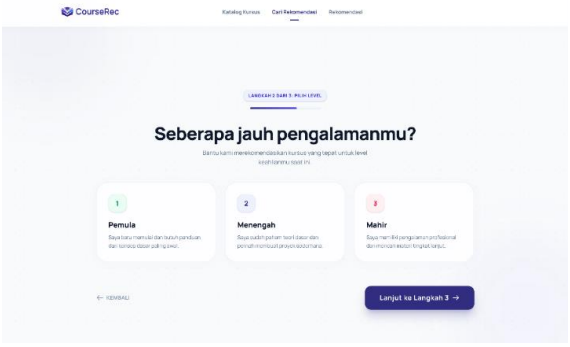
b. Halaman *Onboarding* (Pengisian Preferensi)
Data preferensi yang diinputkan oleh pengguna pada halaman ini akan digunakan sebagai bobot awal dalam pemrosesan rekomendasi berbasis *Content-Based Filtering*. Halaman ini dibagi menjadi tiga bagian/langkah input utama yaitu:

1. Halaman Pengisian Bidang Minat (Topik Utama)
   Pada tahap ini, pengguna memilih salah satu dari 10 topik utama yang ada pada sistem (seperti *Web Development*, *Data Science*, atau *Cybersecurity*). Parameter ini digunakan oleh sistem untuk melakukan penyaringan awal (filtering) guna membatasi ruang pencarian kursus hanya pada kategori yang relevan.



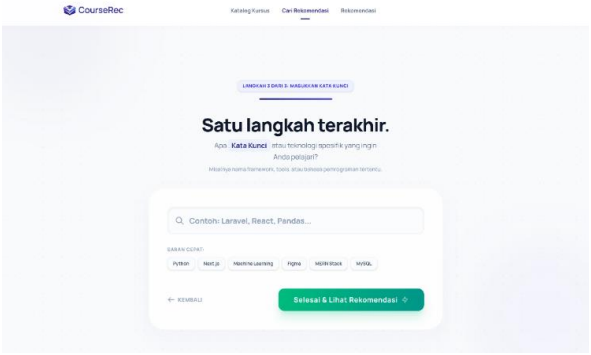
Gambar 7. Halaman Onboarding (Topik Utama)

2. Halaman Pengisian Level Kemampuan
   Pengguna menentukan tingkat pemahaman atau level kemampuan mereka terhadap topik yang dipilih.



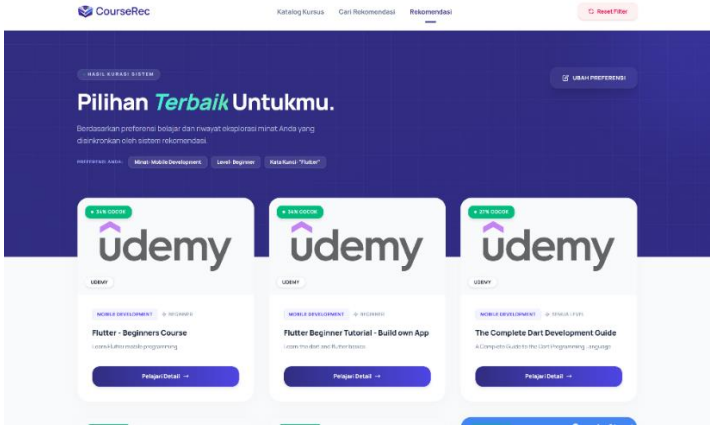
Gambar 8. Halaman Onboarding (Level Kemampuan)

3. Halaman Pengisian Kata Kunci Khusus
   Pengguna memasukkan kata kunci sesuai dengan materi yang ingin dipelajari, seperti "*React*", "*Docker*", atau "*Machine Learning*". Kata kunci tersebut kemudian diproses melalui tahap *preprocessing* teks dan dibandingkan dengan konten kursus menggunakan metode *Cosine Similarity* untuk menemukan kursus yang paling relevan.



Gambar 9. Halaman Onboarding (Kata Kunci Khusus)

c. Halaman Hasil Rekomendasi
   Halaman hasil rekomendasi berfungsi untuk menampilkan daftar kursus yang direkomendasikan oleh sistem berdasarkan preferensi pengguna. Sistem menyajikan 10 kursus dengan nilai kemiripan tertinggi yang diperoleh dari perhitungan *Cosine Similarity*. Informasi yang ditampilkan meliputi judul kursus, topik, tingkat kesulitan, platform penyedia, dan nilai kemiripan sebagai indikator tingkat kesuaian kursus terhadap kebutuhan pengguna.



Gambar 10. Halaman Hasil Rekomendasi

4. KESIMPULAN

Penerapan *Content-Based Filtering* menggunakan kombinasi TF-RF dan *Cosine Similarity* telah berhasil diimplementasikan untuk menghasilkan rekomendasi kursus berdasarkan preferensi pengguna berupa topik, tingkat kesulitan, dan kata kunci. Hasil evaluasi memperoleh nilai rata-rata *Precision@10* sebesar 0,8750,

Recall@10 sebesar 0,2556, dan *F1@10* sebesar 0,3331. Nilai *Precision@10* menunjukkan bahwa sebagian besar rekomendasi yang ditampilkan memenuhi kriteria relevansi yang digunakan pada penelitian ini, sedangkan nilai *Recall@10* yang lebih rendah dipengaruhi oleh pembatasan jumlah rekomendasi Top-10, variasi jumlah dokumen relevan pada setiap skenario pengujian, serta keterbatasan representasi leksikal TF-RF dan *Cosine Similarity* yang mengukur kemiripan berdasarkan kesamaan istilah. Keterbatasan tersebut terlihat terutama pada kueri yang terdiri atas lebih dari satu istilah, seperti pada kueri "SQL Database", di mana beberapa rekomendasi memiliki kemiripan kata tetapi belum sepenuhnya memenuhi kriteria relevansi pada *ground truth*. Oleh karena itu, penelitian selanjutnya dapat mengembangkan sistem dengan menggunakan metode representasi teks berbasis semantik, seperti Word2Vec, FastText, atau BERT, sehingga proses pencocokan tidak hanya mempertimbangkan kesamaan istilah, tetapi juga hubungan makna antar kata.

DAFTAR PUSTAKA

[1] K. K. Jena, K. S. Bhoi, K. T. Malik, S. K. Sahoo, S. Bhatia, and F. Amsaad, "E-Learning Course Recommender System Using Collaborative Filtering Models," *Electronics*, vol. 12, no. 157, 2023, doi: https://doi.org/10.3390/electronics12010157.

[2] L. R. Meza and G. D. Urso, "Navigating the Paradox of Choice in the Digital Age : A Scoping Review on the Potential Role of Recommender Systems," *World Futures*, vol. 81, no. 2, pp. 108–137, 2025, doi: 10.1080/02604027.2024.2424734.

[3] X. Long *et al.*, "The Choice Overload Effect in Online Recommender Systems," *Manuf. Serv. Oper. Manag.*, vol. 27, no. 1, pp. 249–268, 2025, doi: https://doi.org/10.1287/msom.2022.0659.

[4] S. T. Phalle and S. Bhushan, "Content Based Filtering And Collaborative Filtering: A Comparative Study," *J. Adv. Zool.*, vol. 45, no. 4, pp. 96–100, 2024, doi: 10.53555/jaz.v45is4.4158.

[5] K. P. Harmandini and K. M. L, "Analysis of TF-IDF and TF-RF Feature Extraction on Product Review Sentiment," *Sinkron*, vol. 8, no. 2, pp. 929–937, 2024, doi: 10.33395/sinkron.v8i2.13376.

[6] A. Apriani, H. Zakiyudin, and K. Marzuki, "Penerapan Algoritma Cosine Similarity dan Pembobotan TF-IDF System Penerimaan Mahasiswa Baru pada Kampus Swasta," *J. Bumigora Inf. Technol.*, vol. 3, no. 1, pp. 19–27, 2021, doi: 10.30812/bite.v3i1.1110.

[7] S. Oyadila, D. Abdullah, and A. Razi, "Implementasi Content-Based Filtering Dengan Tf-Idf Dan Cosine Similarity Untuk Sistem Rekomendasi Destinasi Wisata Di Aceh Tengah," *Rabit J. Teknol. dan Sist. Inf. Univrab*, vol. 10, no. 2, pp. 1329–1339, 2025, doi: 10.36341/rabit.v10i2.6532.

[8] T. Wahyu Intan Permadani, D. Adi Prasetya, E. Daniati, and P. Korespondens, "Sistem Rekomendasi Film Berdasarkan Genre Menggunakan Metode Content-Based Filtering dengan Algoritma Cosine Similarity," *Pros. SEMNAS INOTEK (Seminar Nas. Inov. Teknol.*, vol. 9, no. 3, pp. 1835–1843, 2025, [Online]. Available: https://proceeding.unpkediri.ac.id/index.php/inotek/article/view/7189

[9] S. Rahmadhani, L. Hakim, G. Hendra Wibowo, S. Purnama Kristanto, and E. Mistiko Rini, "SISTEM REKOMENDASI PENELUSURAN BUKU BERBASIS CONTENT-BASED FILTERING DENGAN PEMBOBOTAN TF-RF," *JIP (Jurnal Inform. Polinema)*, vol. 10, no. 4, pp. 491–500, 2024.

[10] C. Li, W. Li, Z. Tang, S. Li, and H. Xiang, "An improved term weighting method based on relevance frequency for text classification," *Soft Comput.*, vol. 5, 2022, doi: 10.1007/s00500-022-07597-5.

[11] M. A. Khder, "Web scraping or web crawling: State of art, techniques, approaches and application," *Int. J. Adv. Soft Comput. its Appl.*, vol. 13, no. 3, pp. 144–168, 2021, doi: 10.15849/ijasca.211128.11.

[12] M. Siino, I. Tinnirello, and M. La Cascia, "Is text preprocessing still worth the time ? A comparative survey on the influence of popular preprocessing methods on Transformers and traditional classifiers," *Inf. Syst.*, vol. 121, no. March 2023, p. 102342, 2024, doi: 10.1016/j.is.2023.102342.

[13] I. K. W. Dananjaya, I. G. Ayu, and A. Diatri, "Perbandingan Metode Pembobotan TF-RF Dan TF-ABS Pada Kategorisasi Berita Di BDI Denpasar," *SINTECH*, vol. 6, no. 1, pp. 16–25, 2023, doi: https://doi.org/10.31598.

[14] S. C. Mana and T. Sasipraba, "Research on cosine similarity and pearson correlation based recommendation models," *J. Phys. Conf. Ser.*, vol. 1770, no. 1, 2021, doi: 10.1088/1742-6596/1770/1/012014.

[15] Y. Zhou, G. F. Ma, X. Wen, X. H. Yang, and Y. C. Zhang, "Sequential recommender systems: A methodological taxonomy and research frontiers," *Comput. Sci. Rev.*, vol. 59, no. August 2025, p. 100818, 2026, doi: 10.1016/j.cosrev.2025.100818.