

# Sistem Rekomendasi Film Berdasarkan Genre Menggunakan Metode Content-Based Filtering dengan Algoritma Cosine Similarity

<sup>1</sup>Trisna Wahyu Intan Permadani, <sup>2</sup>Dika Adi Prasetya, <sup>3</sup>Erna Daniati

<sup>1-3</sup>. Teknik Sistem Informasi, Universitas Nusantara PGRI Kediri

E-mail:<sup>1</sup>. [trisnawahyuintan@gmail.com](mailto:trisnawahyuintan@gmail.com), <sup>2</sup>. [dikaadi040@gmail.com](mailto:dikaadi040@gmail.com),

<sup>3</sup>[ernadaniati@unpkediri.ac.id](mailto:ernadaniati@unpkediri.ac.id)

*Penulis Korespondens : Erna Daniati*

**Abstrak**— Perkembangan teknologi informasi telah memberikan kemudahan dalam mengakses hiburan digital dalam memilih film melalui layanan streaming seperti Netflix dan Disney+. Namun, banyaknya pilihan justru menyulitkan pengguna dalam menentukan film yang sesuai dengan preferensi mereka. Penelitian ini bertujuan untuk membangun sistem rekomendasi film berbasis konten (Content-Based Filtering) yang dapat memberikan saran film yang relevan berdasarkan kemiripan konten, khususnya dari sisi genre. Pendekatan yang digunakan melibatkan metode pembobotan TF-IDF untuk merepresentasikan genre film dalam bentuk vektor, kemudian menghitung kemiripan antar film menggunakan cosine similarity. Sistem bekerja dengan mencocokkan film yang disukai pengguna terhadap seluruh data, lalu merekomendasikan film lain yang memiliki tingkat kemiripan tertinggi. Hasil implementasi menunjukkan bahwa sistem ini mampu memberikan rekomendasi film yang relevan dan masuk akal. Contohnya, ketika pengguna memilih "Toy Story (1995)", sistem berhasil merekomendasikan film-film sekuel dan bertema serupa.

**Kata Kunci**— content-based filtering, cosine similarity, genre film, sistem rekomendasi, TF-IDF.

**Abstract**— The rapid development of information technology has significantly changed how people access digital entertainment, especially through streaming platforms such as Netflix and Disney+. However, the growing number of available films often makes it difficult for users to choose content that matches their preferences. This study aims to develop a content-based film recommendation system that suggests relevant movies by analyzing content similarity, particularly in terms of genre. The approach utilizes TF-IDF weighting to represent film genres as vectorized textual features, followed by cosine similarity to calculate the level of similarity between movies. The system matches a user's preferred film against the entire dataset and then recommends other films with the highest similarity scores. The implementation results demonstrate that the system is capable of generating relevant and logical recommendations. For example, when the user selects "Toy Story (1995)," the system successfully recommends related sequels and thematically similar films.

**Keywords:** recommendation system, content-based filtering, TF-IDF, cosine similarity, film genre.

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



## I. PENDAHULUAN

Perkembangan teknologi informasi yang pesat telah membawa perubahan signifikan dalam berbagai aspek kehidupan, termasuk dalam cara masyarakat mengakses hiburan. Seperti yang dikemukakan oleh Setiawan (2021), "perkembangan teknologi informasi telah memberikan berbagai kemudahan dalam kehidupan masyarakat modern, salah satunya dalam menikmati hiburan digital"[1]. Salah satu bentuk hiburan yang paling digemari Sejak era pandemi berlangsung sekitar hampir satu setengah tahun ini, menjadi salah satu alternatif pilihan yang sangat diminati oleh masyarakat Indonesia untuk menikmati hiburan baik melalui layanan streaming seperti *Netflix*, *Disney+*, maupun platform digital lainnya[2][3][4]. Seiring dengan bertambahnya jumlah film, baik dari dalam negeri maupun luar negeri, banyak pengguna yang mengalami kesulitan dalam menentukan film mana yang sesuai dengan selera mereka.

Kondisi ini mendorong kebutuhan akan sistem yang dapat membantu pengguna dalam menemukan film yang sesuai dengan preferensi mereka. Sistem rekomendasi di gunakan sebagai solusi untuk mencari rekomendasi produk atau layanan berdasar untuk preferensi pengguna ada. Menurut Sari et al. (2020), *"sistem rekomendasi merupakan teknik yang digunakan dalam perkembangan teknologi untuk memberikan kemudahan dalam menentukan pilihan berdasarkan rekomendasi yang diberikan"* [5][6]. Sistem ini memiliki kemampuan untuk memprediksi konten atau item berdasarkan interaksi pengguna sebelumnya, seperti rating, riwayat tontonan, serta informasi personal lainnya yang berkaitan dengan preferensi pengguna tersebut

Studi ini ditujukan untuk mendukung pengembangan sistem rekomendasi film dengan metode Content-Based Filtering[7][8][9]. Metode ini bekerja dengan menganalisis karakteristik atau fitur dari film, seperti genre, aktor, atau sutradara, kemudian mencocokkannya dengan preferensi pengguna berdasarkan histori atau data yang telah diberikan sebelumnya. Sebagaimana dijelaskan oleh Fajriansyah et al. (2021), *Algoritme penyaringan berbasis konten diterapkan dengan mencari kesamaan bobot dari istilah pada bag of words hasil pra-pemrosesan sinopsis film dan judul film* [10][11]. Pada penelitian ini, data yang digunakan untuk membangun sistem rekomendasi difokuskan pada genre film yang sering ditonton dan rating yang diberikan oleh pengguna. Dalam implementasian Content-Based Filtering menggunakan TF-IDF untuk melakukan pembobotan nilai atribut[12].

Pembobotan dilakukan menggunakan metode TF-IDF yang telah dinormalisasi. Kemudian hasil pembobotan akan melalui tahap cosine similarity untuk mencari kemiripan berdasarkan bobot dan diakhiri dengan filtering berdasarkan genre. Hasil pengujian yang dilakukan pada penelitian dengan melibatkan tiga partisipan dengan total jumlah film sebanyak 4000 judul film didapatkan nilai akurasi menggunakan mean average precision @K (MAP@K) sebesar 0.823254 untuk jenis kueri single kueri dan 0.7500556 untuk jenis kueri multiple seeds kueri. Dari hasil tersebut, terlihat bahwa kueri tunggal menghasilkan rekomendasi yang lebih baik dibandingkan dengan kueri multi-seed [13], [14]

## II. METODE

Sistem rekomendasi merupakan program atau sistem penyaringan informasi yang menjadi solusi dalam masalah kelebihan informasi dengan cara menyaring sebagian informasi penting dari banyaknya informasi yang ada, dan bersifat dengan kecenderungan, ketertarikan, atau tindakan pengguna terhadap suatu produk.. Seperti diungkapkan oleh Fajriansyah et al. (2021), “*sistem rekomendasi dapat membantu pengguna dalam memilih item sesuai dengan preferensi mereka melalui proses penyaringan informasi yang relevan*”[10]

Dalam penelitian ini, Peneliti menerapkan algoritma Content-Based Filtering, yang merupakan metode yang cukup umum dalam area Information Retrieval. Area riset seputar Content-based Filtering berfokus pada rekomendasi konten dalam membangun sistem rekomendasi film[15][16]. Fokus utama dari metode ini adalah menganalisis karakteristik film, khususnya **genre**, sebagai dasar perhitungan kemiripan antar film. Sistem bekerja dengan melihat genre dari film-film yang telah disukai atau ditonton oleh pengguna, kemudian mencari film lain yang memiliki genre serupa[17].

Untuk meningkatkan akurasi hasil rekomendasi, dalam penelitian ini kami menggabungkan metode **Term Frequency-Inverse Document Frequency (TF-IDF)** dalam representasi data genre film. “TF-IDF berfungsi untuk memberikan bobot pada kata-kata dalam dokumen berdasarkan frekuensi kemunculannya dalam satu dokumen dibandingkan dengan seluruh koleksi dokumen yang ada[18][19][20].

"TF-IDF menghitung seberapa penting suatu kata dalam dokumen dengan mempertimbangkan frekuensi kemunculan term (TF) dan kelangkaan term di seluruh koleksi dokumen (IDF). Komponen IDF berfungsi menurunkan nilai kata-kata umum seperti 'film' atau 'movie' yang kurang informatif, sementara meningkatkan bobot kata spesifik seperti 'action' atau 'romantic' yang merepresentasikan tema spesifik[21], [22], [23].

$$TF(t, d) = \frac{\text{jumlah kemunculan kata } t \text{ di dokumen } d}{\text{jumlah total kata di dokumen } d}$$

$$IDF(t) = \log\left(\frac{N}{df(t)}\right)$$

$N$  = jumlah total dokumen (film)

$df(t)$  = jumlah dokumen yang mengandung kata

Elanjutnya, Pembobotan dilakukan menggunakan metode TF-IDF yang telah dinormalisasi. Kemudian hasil pembobotan akan melalui tahap cosine similarity untuk mencari kemiripan berdasarkan bobot dan diakhiri dengan filtering berdasarkan genre[24][25]. Berdasarkan hasil perhitungan ini, sistem kemudian merekomendasikan film dengan nilai kemiripan tertinggi terhadap film-film yang telah disukai oleh pengguna[26][27]. Dengan pendekatan ini, kami berharap sistem mampu memberikan rekomendasi film yang lebih relevan sesuai dengan preferensi genre masing-masing pengguna.

Sebagai contoh, jika sebuah film memiliki genre **action** dan **comedy**, maka vektor genre untuk film tersebut bisa terlihat seperti ini: [0.45, 0.60, 0, 0, ...], di mana setiap angka merupakan bobot TF-IDF yang mewakili genre tertentu dalam urutan tetap.

Setelah representasi genre film selesai, kami menghitung **cosine similarity** antara dua film. Cosine similarity dihitung dengan rumus:

$$\text{Cosin similarity} = \frac{A \cdot B}{\|A\| \|B\|}$$

*A* dan *B* adalah vector genre dari dua film yang dibandingkan

*A.B* adalah hasil perkalian scalar antara dua vector

$\|A\|$  dan  $\|B\|$  adalah panjang (norm) dari vector vector tersebut



**Gambar 1.** alur sistem rekomendasi film

Pada gambar 1. ini kami membahas bagaimana alur sistem rekomendasi film yang kami bangun bekerja, dimulai dari proses awal hingga akhirnya menghasilkan daftar rekomendasi yang ditampilkan ke pengguna. Dalam pengembangan sistem rekomendasi, flowchart digunakan sebagai alat bantu visual untuk memetakan alur proses mulai dari pengumpulan data, pemrosesan, hingga penyajian hasil rekomendasi kepada pengguna. Setiap tahapan dalam flowchart menjelaskan bagaimana data diolah dan bagaimana keputusan diambil untuk menghasilkan rekomendasi yang relevan[28]. Untuk menjelaskan alur kerja sistem ini secara lebih jelas, kami menggunakan sebuah flowchart sebagai gambaran umum dari langkah-langkah yang dilakukan.

### III. HASIL DAN PEMBAHASAN

movieId	title	genres	rating
0	1 Toy Story (1995)	Adventure Animation Children Comedy Fantasy	4.0
215	2 Jumanji (1995)	Adventure Children Fantasy	4.0
325	3 Grumpier Old Man (1995)	Comedy Romance	4.0
377	4 Waiting to Exhale (1995)	Comedy Drama Romance	3.0
384	5 Father of the Bride Part II (1995)	Comedy	5.0
433	6 Heat (1995)	Action Crime Thriller	4.0
535	7 Sabrina (1995)	Comedy Romance	4.0
589	8 Tom and Huck (1995)	Adventure Children	3.0
597	9 Sudden Death (1995)	Action	4.0
613	10 GoldenEye (1995)	Action Adventure Thriller	3.0

Gambar 2. data film

Tabel Genre yang Tersedia:	
0	Action
1	Adventure
2	Animation
3	Children
4	Comedy
5	Crime
6	Documentary
7	Drama
8	Fantasy
9	Film-Noir
10	Horror
11	IMAX
12	Musical
13	Mystery
14	Romance
15	Sci-Fi
16	Thriller
17	War
18	Western

Gambar 3. data genre yang tersedia

```
# === 2. Bersihkan Data ===
df_movies = df_movies.dropna(subset=['genres'])
df_movies = df_movies[df_movies['genres'] != '(no genres listed)'].copy()
df_movies = df_movies.drop_duplicates(subset=['title']).copy()
df_movies = df_movies.head(10000).copy()
```

Gambar 4. membersihkan data

Pada pembahasan untuk rekomendasi film data yang dipakai dalam rancangan ini diambil melalui web Kaggle. Ada 1 dataset yang akan di pakai dan memiliki lebih dari 100855 data film yang mengandung movieId, title, tahun, genres, rating. Data pertama merupakan data film dapat dilihat pada **gambar 2.** dan pada **gambar 3.** merupakan data genre yang tersedia pada dataset rekomendasi\_film serta pada **gambar 4.** adalah bagian dari proses pembersihan data (data cleaning) untuk dataset film yang disimpan dalam DataFrame, tujuan nya agar tidak ada nilai kosong pada kolom genres, tidak ada entri genre yang bernilai (no genres listed), tidak ada duplikat berdasarkan judul film, hanya mengambil data sebanyak 10.000 data

```
# === 3. Gabungkan judul dan genre ===
# Format genre jadi token terpisah
df_movies['genre_string'] = df_movies['genres'].str.replace("|", " ")

# Gabungkan judul dan genre
df_movies['judul_dan_genre'] = df_movies['title'] + " " + df_movies['genre_string']
```

Gambar 5. Gabungkan judul dan genre

Pada **gambar 5.** data genre untuk tiap film ditulis dalam format terpisah, menggunakan simbol “|” sebagai pemisah antar genre. Contohnya, genre seperti “Action|Adventure|Fantasy” perlu kami ubah menjadi format yang lebih ramah terhadap proses pengolahan teks, yakni “Action Adventure Fantasy”. Perubahan ini penting karena metode seperti TF-IDF bekerja berdasarkan token atau kata yang dipisahkan oleh spasi, sehingga setiap genre bisa dibaca sebagai kata yang berdiri sendiri. Setelah format genre disesuaikan, langkah berikutnya adalah menggabungkannya dengan judul film. Jadi, untuk setiap entri film, kami menyatukan nama judul dan genre yang sudah diproses ke dalam satu kolom baru. Misalnya, untuk film Toy Story (1995) yang memiliki genre Adventure, Animation, Children, Comedy, dan Fantasy, maka akan digabung menjadi sebuah teks seperti “Toy Story (1995) Adventure Animation Children Comedy Fantasy”.

```
# === 4. TF-IDF dan Cosine Similarity ===
vectorizer = TfidfVectorizer()
tfidf_matrix = vectorizer.fit_transform(df_movies['judul_dan_genre'])

cosine_sim = cosine_similarity(tfidf_matrix, tfidf_matrix)
```

**Gambar 6.** ubah TF – IDF & hitung cosine

pada **gambar 6.** memper lihatkan Proses ini dimulai dengan inisialisasi objek *TfidfVectorizer()* dari pustaka sklearn. Objek ini digunakan untuk mengubah kumpulan teks pada kolom *judul\_dan\_genre* menjadi matriks vektor TF-IDF. Setiap baris dalam matriks tersebut merepresentasikan satu film, sedangkan setiap kolom merepresentasikan sebuah kata unik yang muncul dalam kumpulan teks. Nilai pada setiap sel menunjukkan bobot TF-IDF dari kata tersebut dalam film tertentu. Setelah mendapatkan matriks TF-IDF, langkah berikutnya adalah menghitung tingkat kemiripan antar film menggunakan fungsi *cosine\_similarity()*. Cosine similarity adalah metode yang digunakan untuk mengukur seberapa mirip dua vektor dalam ruang berdimensi tinggi, dengan cara menghitung nilai cosinus dari sudut antara dua vektor tersebut. Nilai hasil perhitungannya berada di antara 0 dan 1, di mana angka 1 berarti kedua film sangat mirip (atau identik secara fitur), sedangkan angka mendekati 0 menunjukkan kemiripan yang rendah.

```
# === 5. Fungsi Rekomendasi Ambil Top-N Film Terdekat===
judul_film = df_movies['title'].tolist()

def rekomendasi_film(judul_input, cosine_sim_matrix, judul_film, top_n=5):
    mapping_judul = {judul.strip().lower(): idx for idx, judul in enumerate(judul_film)}
    judul_input = judul_input.strip().lower()

    if judul_input not in mapping_judul:
        return f"Film '{judul_input}' tidak ditemukan dalam data."

    idx_input = mapping_judul[judul_input]
    sim_scores = list(enumerate(cosine_sim_matrix[idx_input]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    sim_scores = [score for score in sim_scores if score[0] != idx_input]
    top_similar = sim_scores[:top_n]
    return [(judul_film[i], skor) for i, skor in top_similar]
```

**Gambar 7.** Top N Film Terdekat

Film yang mirip dengan 'Toy Story (1995)':			
	Judul Film	Skor	Kemiripan
1	Toy Story 2 (1999)	0.880486	
2	Toy Story 3 (2010)	0.821171	
3	Toy, The (1982)	0.525882	
4	Toy Soldiers (1991)	0.400321	
5	NeverEnding Story, The (1984)	0.381297	

**Gambar 8.** Uji Rekomendasi Film

Setelah sistem menghitung nilai *Cosine similarity* antar seluruh film berdasarkan representasi teks yang telah dibentuk sebelumnya, langkah selanjutnya adalah Pengujian dilakukan untuk mengklasifikasikan pada tahap ini[29]. Pengguna akan masukan data film yang akan diklasifikasikan[30]. Input dari pengguna kemudian dicocokkan dengan daftar judul film dalam dataset yang telah diproses—dengan format huruf kecil untuk memastikan pencocokan berjalan konsisten. Jika judul yang dimasukkan tidak ditemukan, sistem secara otomatis memberikan notifikasi bahwa film tidak tersedia. Namun jika ditemukan, sistem akan mengambil indeks film tersebut dari matriks similarity dan menghitung tingkat kemiripannya terhadap seluruh film lain. Pada **gambar 7.** nilai-nilai similarity tersebut diurutkan dari yang paling tinggi, lalu dipilih sejumlah *Top-N* film dengan skor tertinggi, tidak termasuk film itu sendiri. Hasil dari proses ini ditampilkan kepada pengguna dalam bentuk daftar rekomendasi film yang paling mirip berdasarkan konten. Pada **gambar 8.** Sebagai contoh, ketika pengguna memasukkan judul "Toy Story (1995)", sistem menghasilkan rekomendasi film seperti *Toy Story 2 (1999)* dan *Toy Story 3 (2010)*, yang memiliki skor kemiripan sangat tinggi. Hal ini menunjukkan bahwa sistem mampu mengenali pola kemiripan konten dengan baik berdasarkan fitur-fitur yang telah diekstraksi, dan memberikan hasil rekomendasi yang masuk akal dan relevan dengan preferensi pengguna.

#### IV. KESIMPULAN

Tujuan dari penelitian ini adalah untuk membangun menggunakan algoritma *Content-Based Filtering* berbasis konten menggunakan TF-IDF dan pendekatan *Cosine Similarity*, menggunakan algoritma *Content-Based Filtering* berbasis konten yang berfokus pada kesamaan genre film. Sistem yang dirancang berhasil memberikan rekomendasi film yang relevan berdasarkan pengaturan pengguna untuk genre tertentu. Menggunakan presentasi teks dalam TF-IDF memungkinkan sistem untuk memahami konten genre secara lebih rinci, tetapi perhitungan kesamaan kosinus secara efektif mengukur tingkat kesamaan antara film. Hasil tes menunjukkan bahwa sistem dapat merekomendasikan film yang sangat relevan, seperti yang ditunjukkan dalam kasus "Toy Story (1995)." Ini menciptakan sekuel dan film dengan topik serupa

Saran untuk pengembangan selanjutnya adalah menggabungkan metode Content-Based Filtering dengan teknik lain seperti Collaborative Filtering atau Hybrid Filtering, agar sistem rekomendasi dapat mempertimbangkan lebih banyak faktor, termasuk perilaku pengguna lain, bukan hanya genre. Selain itu, akan lebih baik jika sistem diperluas dengan mempertimbangkan elemen konten lain seperti sinopsis, sutradara, dan aktor, agar hasil rekomendasi menjadi lebih personal dan akurat.

#### DAFTAR PUSTAKA

- [1] R. A. Sari, S. Fajar Isnaini, and E. Seniwati, "Sistem Rekomendasi Film Menggunakan Content Based Filtering," 2025. [Online]. Available: <https://subset.id/index.php/IJCSR>
- [2] M. Eko Prasetyo, "KAJIAN KOMPOSISI VISUAL PADA FILM SERIAL NETFLIX DRAMA FIKSI ILMIAH BERJUDUL THE 100 KARYA JASON ROTHENBERG STUDY OF VISUAL COMPOSITION ON THE FILM SERIAL NETFLIX SCIENTIFIC DRAMA TITLE THE 100 BY JASON ROTHENBERG," vol. 4, pp. 45–64, 2021, [Online]. Available: <http://journal.ubm.ac.id/index.php/titik-imaji/>
- [3] Y. Anjani, M. Diandra Wicaksana, A. Kuswanti, U. R. Pembangunan Nasional Veteran Jakarta Jl Fatmawati, P. Labu, and J. Selatan, "PENGGUNAAN APLIKASI STREAMING NETFLIX PADA GENERASI Z," 2023.
- [4] M. I. Djamzuri and A. Putra Mulyana, "Fenomena Netflix Platform Premium Video Streaming Membangun Kesadaran Cyber Etik Dalam Perspektif Ilmu Komunikasi," *Jurnal Ilmu Sosial dan Pendidikan (JISIP)*, vol. 6, no. 1, pp. 2598–9944, 2022, doi: 10.36312/jisip.v6i1.2804/http.
- [5] R. A. Sari, S. Fajar Isnaini, and E. Seniwati, "Sistem Rekomendasi Film Menggunakan Content Based Filtering," 2025. [Online]. Available: <https://subset.id/index.php/IJCSR>
- [6] "SISTEM REKOMENDASI PEMILIHAN PRODUK SKINCARE DENGAN PENDEKATAN CONTENT-BASED FILTERING."
- [7] D. Theo Santoso, V. Atina, and D. Hartanti, "Prototipe Sistem Rekomendasi Film Indonesia Menggunakan Pendekatan Content Based Filtering dan Metode Vector Space Model," *Infotek: Jurnal Informatika dan Teknologi*, vol. 7, no. 2, pp. 444–455, Jul. 2024, doi: 10.29408/jit.v7i2.26083.
- [8] H. H. Arfisko, "Sistem Rekomendasi Film Menggunakan Metode Hybrid Collaborative Filtering Dan Content-Based Filtering."
- [9] E. Salim, J. Pragantha, and M. D. Lauro, "Perancangan Sistem Rekomendasi Film menggunakan metode Content-based Filtering."

- [10] M. Fajriansyah, P. P. Adikara, and A. W. Widodo, "Sistem Rekomendasi Film Menggunakan Content Based Filtering," 2021. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [11] S. K. Dirjen *et al.*, "Terakreditasi SINTA Peringkat 4 Sistem Rekomendasi Produk Pena Eksklusif Menggunakan Metode Content-Based Filtering dan TF-IDF," 2018.
- [12] R. Ardiansyah, M. Ari Bianto, and B. D. Saputra, "Sistem Rekomendasi Buku Perpustakaan Sekolah menggunakan Metode Content-Based Filtering," *Jurnal CoSciTech (Computer Science and Information Technology)*, vol. 4, no. 2, pp. 510–518, Oct. 2023, doi: 10.37859/coscitech.v4i2.5131.
- [13] M. Fajriansyah, P. P. Adikara, and A. W. Widodo, "Sistem Rekomendasi Film Menggunakan Content Based Filtering," 2021. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [14] A. H. Azmi, I. Naufal, R. Mahardika, A. Shaktika, A. Prasetya, and S. Puspita, "Sistem Rekomendasi Film Berbasis Konten Menggunakan Teknik Cosine Similarity dan TF-IDF," *Journal of Multidisciplinary Inquiry in Science Technology and Educational Research*, vol. 1, no. 4, 2024, doi: 10.32672/mister.v1i4.2206.
- [15] A. Rochmad Wahono, B. Aji Saputra, and F. Fadlu Rahman, "Sistem Rekomendasi Film Menggunakan Metode Content-Based Filtering dan Algoritma K-Nearest Neighbors (KNN)," pp. 18–2024.
- [16] A. A. Huda, R. Fajarudin, and A. Hadinegoro, "Sistem Rekomendasi Content-based Filtering Menggunakan TF-IDF Vector Similarity Untuk Rekomendasi Artikel Berita," *Building of Informatics, Technology and Science (BITS)*, vol. 4, no. 3, Dec. 2022, doi: 10.47065/bits.v4i3.2511.
- [17] T. Ridwansyah, B. Subartini, and S. Sylviani, "Penerapan Metode Content-Based Filtering pada Sistem Rekomendasi," *Mathematical Sciences and Applications Journal*, vol. 4, no. 2, pp. 70–77, Apr. 2024, doi: 10.22437/msa.v4i2.32136.
- [18] D. A. Putri, D. Pramesti, D. I., and W. Santiyasa, "Penerapan Metode Content-Based Filtering dalam Sistem Rekomendasi Video Game," 2022.
- [19] S. Rosetya Wardhana and R. Kembang Hapsari, "Sistem Rekomendasi Film dengan Menggunakan Pendekatan Collaborative Filtering Berdasarkan Class," *Prosiding Seminar Implementasi Teknologi Informasi dan Komunikasi*, vol. 2, no. 1, 2023, doi: 10.31284/p.semtik.2023-1.4153.
- [20] R. Al Rasyid, D. Handayani, and U. Ningsih, "Penerapan Algoritma TF-IDF dan Cosine Similarity untuk Query Pencarian Pada Dataset Destinasi Wisata," *Jurnal Teknologi Informasi dan Komunikasi*, vol. 8, no. 1, p. 2024, 2024, doi: 10.35870/jti.
- [21] D. Remawati, H. Wijayanto, Y. Retno, W. Utami, and B. D. Raharja, "Pengelompokan Film Trending di Youtube Menggunakan TF-IDF dan K-Means Clustering", [Online]. Available: <https://ojs.trigunadharma.ac.id/index.php/jsi>
- [22] D. Septiani and I. Isabela, "SINTESIA: Jurnal Sistem dan Teknologi Informasi Indonesia ANALISIS TERM FREQUENCY INVERSE DOCUMENT FREQUENCY (TF-IDF) DALAM TEMU KEMBALI INFORMASI PADA DOKUMEN TEKS".
- [23] J. E. Br Sinulingga and H. C. K. Sitorus, "Analisis Sentimen Opini Masyarakat terhadap Film Horor Indonesia Menggunakan Metode SVM dan TF-IDF," *Jurnal Manajemen Informatika (JAMIKA)*, vol. 14, no. 1, pp. 42–53, Feb. 2024, doi: 10.34010/jamika.v14i1.11946.
- [24] M. Yusuf and A. Cherid, "Implementasi Algoritma Cosine Similarity Dan Metode TF-IDF Berbasis PHP Untuk Menghasilkan Rekomendasi Seminar."
- [25] A. Zakharia *et al.*, "Sistem Rekomendasi Film Indonesia Menggunakan Metode Content-Based Filtering", [Online]. Available: <https://journal.mediapublikasi.id/index.php/logic>
- [26] R. Harjo Utomo, G. Susrama, M. Diyasa, U. Pembangunan, N. " Veteran, and J. Timur, "MOVIEMU : SISTEM REKOMENDASI FILM MENGGUNAKAN ALGORITMA COSINE SIMILARITY," vol. 4, no. 2, pp. 22–32, 2024.
- [27] M. Robbani, R. Dias Ramadhani, A. Elok Amalia, P. Studi Informatika, and P. Studi Rekayasa Perangkat Lunak, "Analisa Algoritma Cosine Similarity dengan Pearson Correlation pada Metode Item-based Collaborative Filtering dengan Menggunakan Dataset MovieLens," 2018.

- [28]J. Aisyah and L. Cahyani, “Sistem Rekomendasi Program Studi Menggunakan Metode Hybrid Recommendation (Studi Kasus: MAN Sumenep),” *Jurnal Eksplora Informatika*, vol. 12, no. 1, pp. 59–72, Jan. 2024, doi: 10.30864/eksplora.v12i1.992.
- [29]E. Daniati, “KLASIFIKASI JENIS BIMBINGAN DAN KONSELING SISWA SMKN 1 KEDIRI MENGGUNAKAN NAIVE BAYES CLASSIFIER DAN NEAREST NEIGHBOR,” *NoE*, vol. 1.
- [30]E. Daniati, “Klasifikasi Kelompok Penjaminan Mutu Pada Karyawan Perusahaan XYZ Dengan KNN dan J48.”