

Sistem Rekomendasi Film Menggunakan Algoritma Natural Language Processing Berbasis Web

Ahmad Rifan Akmal Farizal¹⁾ Adi Sucipto²⁾, Teguh Tamrin³⁾

Teknik Informatika, Fakultas Sains dan Teknologi Universitas Islam Nahdlatul Ulama Jepara

Email : ¹putrakrajan3538@gmail.com, ²adisucipto@unisnu.ac.id, ³teguh@unisnu.ac.id

Abstract. Perkembangan platform *streaming* digital yang semakin pesat mendorong meningkatnya kebutuhan akan sistem rekomendasi yang mampu memberikan saran film secara lebih personal dan sesuai dengan preferensi pengguna. Pendekatan yang umum digunakan masih bergantung pada *metadata* seperti genre dan *rating* pengguna, sehingga rekomendasi yang dihasilkan cenderung bersifat umum dan belum sepenuhnya mencerminkan isi cerita film. Penelitian ini mengembangkan sistem rekomendasi film berbasis web dengan memanfaatkan teknik *Natural Language Processing* untuk menganalisis sinopsis film sebagai sumber informasi utama. Metode *Term Frequency–Inverse Document Frequency* digunakan untuk mengubah teks menjadi representasi numerik, sedangkan *cosine similarity* digunakan untuk mengukur tingkat kemiripan antar sinopsis film. Penelitian ini menggunakan pendekatan terapan dan eksperimental yang mencakup tahapan pengumpulan dataset, *preprocessing* teks (*case folding*, *tokenization*, *stopword removal*, dan *stemming*), pembentukan representasi vektor, perhitungan nilai *similarity*, serta evaluasi sistem. Hasil pengujian menunjukkan bahwa nilai *similarity* yang dihasilkan mampu menggambarkan tingkat kemiripan antara film yang dijadikan input dengan film yang direkomendasikan, di mana semakin tinggi nilai *similarity* maka semakin tinggi pula tingkat kesesuaian konten cerita antar film. Berdasarkan hal tersebut, penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem rekomendasi film berbasis web menggunakan teknik *Natural Language Processing* dengan metode TF-IDF dan *cosine similarity* guna menghasilkan rekomendasi film yang lebih relevan berdasarkan kemiripan sinopsis.

Keyword : *Sistem Rekomendasi, Natural Language Processing, TF-IDF, Cosine Similarity, Sistem Berbasis Web*

Abstract. The rapid development of digital streaming platforms has increased the demand for recommendation systems capable of providing more personalized movie suggestions based on user preferences. Most existing recommendation approaches still rely heavily on metadata such as genres and user ratings, resulting in recommendations that are relatively general and unable to fully represent the actual storyline content of movies. This study develops a web-based movie recommendation system by utilizing Natural Language Processing (NLP) techniques to analyze movie synopses as the primary source of information. The Term Frequency–Inverse Document Frequency (TF-IDF) method is employed to transform textual data into numerical vector representations, while cosine similarity is used to measure the similarity level between movie synopses. The research applies an experimental and applied approach consisting of several stages, including dataset collection, text preprocessing (*case folding*, *tokenization*, *stopword removal*, and *stemming*), vector representation construction, similarity calculation, and system evaluation. The experimental results indicate that the generated similarity scores are capable of accurately representing the relationship between the input movie and the recommended movies. Higher similarity values indicate a higher level of storyline relevance between movies. Therefore, this study aims to design and implement a web-based movie recommendation system using Natural Language Processing techniques with TF-IDF and cosine similarity methods to produce more relevant movie recommendations based on synopsis similarity.

Keywords: Recommendation System, Natural Language Processing, TF-IDF, Cosine Similarity, Web-Based System



PENDAHULUAN

Perkembangan teknologi digital telah mendorong berkembangnya berbagai layanan *streaming* yang menyediakan koleksi film dalam jumlah sangat besar. Melalui platform tersebut pengguna dapat mengakses berbagai jenis film dari berbagai kategori dan genre secara mudah. Namun, banyaknya pilihan film yang tersedia sering kali menyebabkan pengguna mengalami kesulitan dalam menentukan film yang sesuai dengan minat atau preferensi tontonan mereka. Situasi ini menimbulkan kebutuhan akan suatu mekanisme yang mampu membantu pengguna dalam menyeleksi konten yang paling relevan dari banyaknya pilihan yang tersedia [1].

Salah satu pendekatan yang banyak digunakan untuk mengatasi permasalahan tersebut adalah sistem rekomendasi. Sistem ini dirancang untuk memberikan saran atau rekomendasi item tertentu kepada pengguna berdasarkan karakteristik data atau pola preferensi yang dimiliki. Dalam berbagai aplikasi berbasis web, sistem rekomendasi berperan penting dalam meningkatkan pengalaman pengguna karena mampu menyajikan informasi yang lebih relevan dan sesuai dengan kebutuhan pengguna [2] [3].

Dalam konteks rekomendasi film, salah satu pendekatan yang banyak digunakan adalah *content-based filtering*. Metode ini bekerja dengan membandingkan karakteristik konten antar film untuk menemukan kesamaan tertentu. Salah satu sumber informasi yang dapat digunakan untuk menggambarkan karakteristik film adalah sinopsis. Sinopsis memuat ringkasan cerita yang mencerminkan unsur naratif seperti alur cerita, konflik, serta tema yang terdapat pada film tersebut [4][5]. Dengan menganalisis teks sinopsis, sistem dapat mengidentifikasi kemiripan konten antar film sehingga rekomendasi yang dihasilkan menjadi lebih relevan bagi pengguna [6][7].

Meskipun demikian, banyak sistem rekomendasi film masih bergantung pada atribut sederhana seperti genre atau penilaian pengguna. Pendekatan tersebut memang dapat memberikan gambaran umum mengenai kategori film, tetapi belum sepenuhnya mampu merepresentasikan isi cerita yang sebenarnya. Preferensi pengguna terhadap film tidak hanya dipengaruhi oleh kategori genre, tetapi juga oleh unsur cerita yang terkandung dalam film tersebut, seperti tema, konflik, maupun suasana emosional yang digambarkan dalam sinopsis [8][9].

Dalam analisis teks, salah satu metode yang umum digunakan untuk merepresentasikan dokumen adalah *Term Frequency–Inverse Document Frequency* (TF-IDF) [10][11]. Metode ini digunakan untuk mengubah teks menjadi representasi numerik dengan memberikan bobot pada setiap kata berdasarkan frekuensi kemunculannya dalam dokumen serta distribusinya pada keseluruhan dokumen dalam korpus teks [12]. Setelah dokumen direpresentasikan dalam bentuk vektor, tingkat kemiripan antar dokumen dapat dihitung menggunakan metode *cosine similarity*, yaitu teknik yang mengukur kesamaan antara dua vektor dalam ruang multidimensi [13][14].

Selain itu, pemanfaatan teknik *Natural Language Processing* (NLP) memungkinkan sistem untuk melakukan pengolahan teks secara lebih terstruktur melalui beberapa tahap seperti *tokenization*, *stopword removal*, dan *stemming*. Tahapan tersebut bertujuan untuk membersihkan teks dari unsur-unsur yang tidak relevan sehingga informasi penting dalam sinopsis dapat dianalisis dengan lebih efektif [15]. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sistem rekomendasi film berbasis web dengan memanfaatkan teknik *Natural Language Processing* menggunakan metode TF-IDF dan *cosine similarity* sehingga sistem dapat menghasilkan rekomendasi film yang lebih relevan berdasarkan kemiripan sinopsis.

METODE PENELITIAN

Penelitian ini menggunakan pendekatan eksperimental untuk mengembangkan sistem rekomendasi film berbasis web dengan memanfaatkan teknik *Natural Language Processing* (NLP). Proses penelitian dilakukan melalui beberapa tahapan utama yang meliputi pengumpulan dataset film, pengolahan teks sinopsis, pembentukan representasi dokumen menggunakan metode TF-IDF, perhitungan tingkat kemiripan dokumen dengan metode *cosine similarity*, implementasi sistem berbasis web, serta evaluasi performa sistem menggunakan metrik *Precision@5*.



1. Data Penelitian

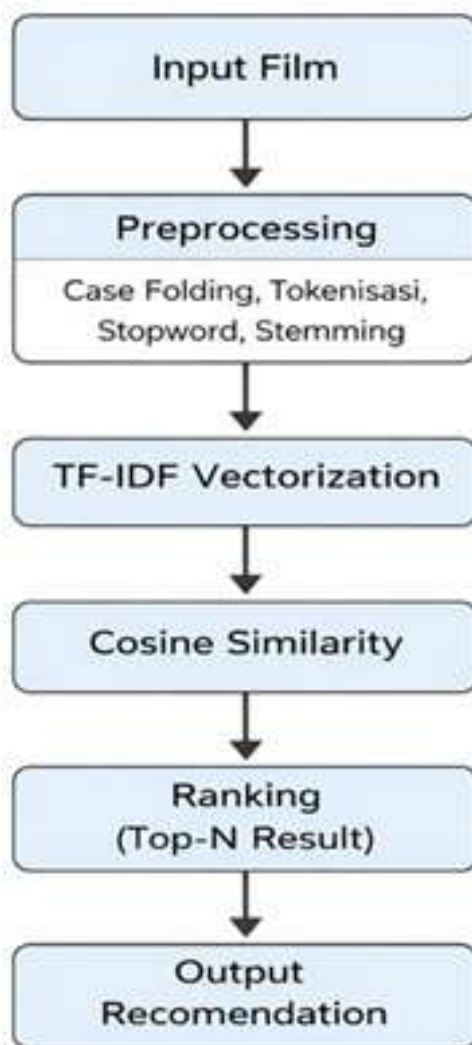
Data yang digunakan dalam penelitian ini berupa dataset film yang memiliki beberapa atribut utama, yaitu judul film, sinopsis (*overview*), genre, dan tahun rilis. Informasi sinopsis dipilih sebagai sumber data utama karena memuat ringkasan cerita yang dapat digunakan untuk menggambarkan karakteristik suatu film.

Dataset film diperoleh dari platform *The Movie Database* (TMDb) dengan memanfaatkan layanan *Application Programming Interface* (API). Proses pengambilan data dilakukan menggunakan bahasa pemrograman Python sehingga pengumpulan data dapat dilakukan secara otomatis. Melalui proses tersebut diperoleh sebanyak 1000 data film yang selanjutnya disimpan dalam format CSV untuk mempermudah proses pengolahan pada tahap berikutnya.

2. Tahapan Sistem

Pengembangan sistem rekomendasi film dalam penelitian ini terdiri dari beberapa tahapan utama, yaitu proses *preprocessing* teks, pembentukan representasi dokumen menggunakan metode TF-IDF, perhitungan kemiripan antar film menggunakan metode *cosine similarity*, serta proses perankingan untuk menghasilkan daftar rekomendasi film [16][17].

Tahap pertama adalah melakukan *preprocessing* terhadap teks sinopsis untuk membersihkan data sebelum dilakukan analisis. Setelah proses tersebut selesai, teks sinopsis diubah menjadi representasi vektor numerik menggunakan metode TF-IDF. Selanjutnya, tingkat kemiripan antar film dihitung menggunakan metode *cosine similarity* [18]. Nilai kemiripan yang dihasilkan kemudian diurutkan sehingga sistem dapat menampilkan daftar film dengan tingkat kemiripan tertinggi berdasarkan parameter Top-N.



Gambar 1. Alur Sistem Rekomendasi Film

3. Preprocessing Teks

Tahap *preprocessing* merupakan proses awal dalam pengolahan teks yang bertujuan untuk menyiapkan data sebelum dianalisis lebih lanjut. Pada penelitian ini proses *preprocessing* dilakukan pada atribut sinopsis film untuk menghilangkan elemen teks yang tidak relevan.

Beberapa tahapan yang dilakukan dalam proses ini antara lain:

- Cleaning text, yaitu proses menghapus karakter yang tidak diperlukan seperti tanda baca dan simbol tertentu.
- Tokenization, yaitu memecah kalimat menjadi unit kata sehingga teks lebih mudah dianalisis.
- Case folding, yaitu mengubah seluruh huruf menjadi huruf kecil agar tidak terjadi perbedaan representasi kata akibat penggunaan huruf kapital.
- Stopword removal, yaitu menghilangkan kata-kata umum yang tidak memberikan informasi penting dalam analisis teks.

Melalui proses tersebut, teks sinopsis yang sebelumnya berupa kalimat panjang akan diubah menjadi kumpulan kata yang lebih sederhana sehingga dapat digunakan pada tahap representasi dokumen.

4. Representasi Dokumen dengan TF-IDF

Setelah tahap *preprocessing* selesai dilakukan, teks sinopsis kemudian direpresentasikan dalam bentuk numerik menggunakan metode *Term Frequency–Inverse Document Frequency* (TF-IDF). Metode TF-IDF digunakan untuk menghitung bobot suatu kata dalam dokumen dengan mempertimbangkan frekuensi kemunculannya pada dokumen tertentu serta distribusinya pada seluruh dokumen dalam dataset.

Melalui proses pembobotan TF-IDF, setiap dokumen sinopsis akan direpresentasikan dalam bentuk vektor numerik. Representasi tersebut kemudian digunakan sebagai dasar untuk menghitung tingkat kemiripan antar dokumen pada tahap berikutnya.

5. Perhitungan Kemiripan dengan Cosine Similarity

Setelah dokumen direpresentasikan dalam bentuk vektor, tahap selanjutnya adalah menghitung tingkat kemiripan antar dokumen menggunakan metode *cosine similarity*. *Cosine similarity* digunakan untuk menghitung tingkat kemiripan antar dokumen dengan memanfaatkan representasi vektor dalam ruang multidimensi.

Nilai *cosine similarity* berada pada rentang 0 hingga 1. Nilai yang mendekati 1 menunjukkan bahwa dua dokumen memiliki tingkat kemiripan yang tinggi, sedangkan nilai yang mendekati 0 menunjukkan bahwa kedua dokumen memiliki tingkat kemiripan yang rendah.

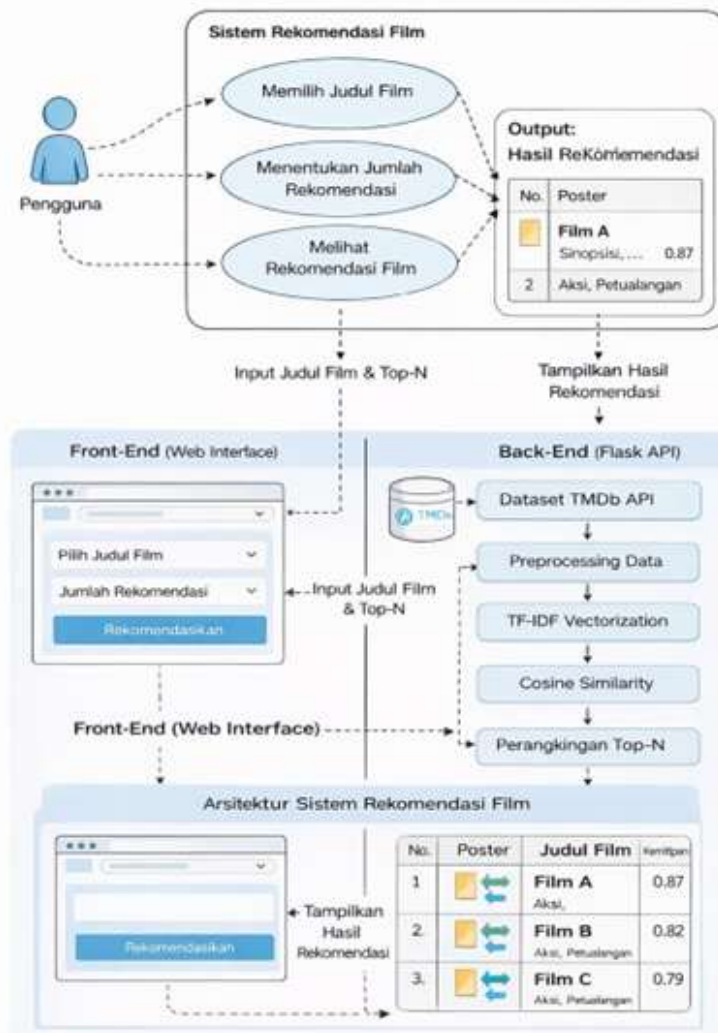
Dalam penelitian ini, metode *cosine similarity* digunakan untuk menghitung kemiripan antara sinopsis film yang dipilih oleh pengguna dengan seluruh sinopsis film yang terdapat dalam dataset. Hasil perhitungan tersebut kemudian diurutkan sehingga sistem dapat menampilkan daftar film dengan tingkat kemiripan tertinggi.

6. Implementasi Sistem

Sistem rekomendasi film pada penelitian ini diimplementasikan dalam bentuk aplikasi berbasis web menggunakan bahasa pemrograman Python dengan *framework Flask* sebagai pengelola sisi backend. Antarmuka pengguna dikembangkan menggunakan teknologi *HTML*, *CSS*, dan *JavaScript* sehingga pengguna dapat berinteraksi dengan sistem melalui halaman web.

Melalui antarmuka tersebut, pengguna dapat memilih judul film sebagai input serta menentukan jumlah rekomendasi film yang ingin ditampilkan. Sistem kemudian akan memproses sinopsis film menggunakan metode TF-IDF dan *cosine similarity* untuk menghasilkan daftar rekomendasi film yang memiliki tingkat kemiripan cerita paling tinggi.





Gambar 2. Arsitektur Sistem Rekomendasi Film

7. Evaluasi Sistem

Evaluasi sistem dilakukan untuk mengetahui kemampuan sistem dalam menghasilkan rekomendasi film yang relevan. Pada penelitian ini, proses evaluasi dilakukan menggunakan metrik *Precision@5*, yaitu ukuran yang digunakan untuk menghitung proporsi rekomendasi yang relevan dari lima film teratas yang dihasilkan oleh sistem.

Sebuah film rekomendasi dikategorikan sebagai relevan apabila memiliki minimal satu genre yang sama dengan film yang dijadikan sebagai input oleh pengguna. Melalui pendekatan tersebut, tingkat ketepatan sistem dalam memberikan rekomendasi film dapat dianalisis secara kuantitatif.

HASIL DAN PEMBAHASAN

Sistem rekomendasi digunakan dalam berbagai aplikasi digital untuk membantu pengguna memperoleh item yang sesuai dengan kebutuhan atau preferensi mereka dengan mempertimbangkan karakteristik data maupun preferensi pengguna. Dalam konteks film, sistem rekomendasi dapat memanfaatkan informasi teks seperti sinopsis untuk mengidentifikasi kesamaan konten antar film sehingga pengguna dapat memperoleh rekomendasi film yang memiliki cerita serupa.

Pada penelitian ini, pendekatan yang digunakan adalah *content-based recommendation* yang memanfaatkan analisis teks pada sinopsis film. Dataset film diperoleh dari platform *The Movie Database* (TMDb) melalui layanan *Application Programming Interface* (API). Dataset

yang digunakan terdiri dari 1000 data film yang mencakup beberapa atribut utama seperti *title*, *overview*, *genres*, dan *release_year*. Atribut *overview* atau sinopsis film dipilih sebagai sumber informasi utama karena mampu merepresentasikan isi cerita yang dapat digunakan untuk mengukur tingkat kemiripan antar film.

```

import requests
import pandas as pd
import time

API_KEY = "67424ce23aefd2fd9cd487539faf4ce5"
BASE_URL = "https://api.themoviedb.org/3"

target = 1000
movies = []
pages = (target // 20) + (1 if target % 20 != 0 else 0) # 1000 -> 50 halaman

for page in range(1, pages + 1):
    url = f"{BASE_URL}/movie/popular?api_key={API_KEY}&language=en-US&page={page}"
    r = requests.get(url)
    data = r.json()

    # cek jika ada error / limit
    if "results" not in data:
        print("Gagal di page:", page, "Response:", data)
        break

    for m in data["results"]:
        movies.append({
            "tmdb_id": m.get("id"),
            "title": m.get("title"),
            "overview": m.get("overview"),
            "release_date": m.get("release_date"),
            "vote_average": m.get("vote_average"),
            "vote_count": m.get("vote_count"),
            "original_language": m.get("original_language")
        })

    time.sleep(0.2) # jeda kecil agar aman

df = pd.DataFrame(movies)

# Bersihkan data yang sinopsisnya kosong
df = df[df["overview"].notnull()]
df = df[df["overview"].str.strip() != ""]

# Jika jumlah > 1000 karena filtering tidak banyak, potong aman
df = df.head(target)

df.to_csv("movies_raw_1000.csv", index=False)

print("Selesai. Jumlah film tersimpan:", len(df))
print("Nama file: movies_raw_1000.csv")

*** Selesai. Jumlah film tersimpan: 950
    Nama file: movies_raw_1000.csv

from google.colab import files
files.download("movies_raw_1000.csv")
    
```

Gambar 3. Proses pengambilan data film menggunakan TMDb API

Dataset yang telah diperoleh kemudian diproses melalui tahap *preprocessing* pada teks sinopsis film sebelum dilakukan analisis lebih lanjut. Tahapan ini bertujuan untuk membersihkan data teks dari elemen-elemen yang tidak relevan sehingga teks dapat diproses dengan lebih efektif. Proses preprocessing yang dilakukan meliputi *cleaning text*, *tokenization*, *case folding*, *stopword removal*, dan normalisasi teks. Melalui proses tersebut, sinopsis film yang awalnya berbentuk kalimat panjang diubah menjadi kumpulan kata yang lebih sederhana sehingga lebih mudah dianalisis oleh sistem. Hasil proses *preprocessing* sinopsis film ditunjukkan pada Tabel 1.

Tabel 1. Hasil preprocessing sinopsis film

No	Sinopsis (overview_id)	Hasil preprocessing (overview_clean)
1	Seorang pria yang tinggal di pengasingan di sebuah pulau terpencil bertemu seorang gadis muda.	pria tinggal asing pulau pencis selamat gadis
2	Ketika kehidupannya yang tenang di sebuah pulau tiba-tiba berubah setelah kejadian misterius.	hidup tenang pulau pencis hancur kembali
3	Terisolasi di sebuah rumah pedesaan yang mewah, seorang seniman menghadapi masa lalunya.	isolasi rumah desa megah seniman sendiri
4	Saat pembunuh Ghostface kembali muncul di kota kecil Sidney.	bunuh ghostface muncul kota sepi sidney
5	James menerima surat misterius dari seseorang yang pernah dicintainya.	james terima surat misterius cinta hilang

Teks sinopsis yang telah melalui tahap preprocessing selanjutnya direpresentasikan dalam bentuk numerik menggunakan metode *Term Frequency–Inverse Document Frequency* (TF-IDF). Metode ini digunakan untuk memberikan bobot pada setiap kata berdasarkan frekuensi



kemunculan kata tersebut dalam dokumen serta distribusinya pada keseluruhan dokumen dalam dataset. Dengan proses pembobotan ini, kata-kata yang memiliki peran lebih penting dalam suatu dokumen akan memperoleh nilai bobot yang lebih tinggi. Dapat dirumuskan sebagai berikut

$$TFIDF(t, d) = TF(t, d) \times IDF(t) \tag{1}$$

dengan:

$$IDF(t) = \log \left(\frac{N}{df(t)} \right) \tag{2}$$

di mana:

1. t = kata.
2. d = dokumen.
3. TF(t,d) = frekuensi yang muncul pada dokumen kata.
4. df(t) = total dokumen yang mengandung kata t.
5. N = total dokumen dalam dataset.

Setelah dokumen direpresentasikan dalam bentuk vektor numerik menggunakan TF-IDF, tahap selanjutnya adalah menghitung tingkat kemiripan antar dokumen menggunakan metode *cosine similarity*. Cosine similarity merupakan teknik yang digunakan untuk menghitung tingkat kemiripan antar dokumen dengan memanfaatkan representasi vektor dari masing-masing dokumen. Semakin kecil sudut yang terbentuk antara dua vektor, maka semakin tinggi tingkat kemiripan antar dokumen tersebut.

Pada penelitian ini, nilai *cosine similarity* digunakan untuk menghitung kemiripan antara film yang dipilih oleh pengguna dengan seluruh film yang terdapat dalam dataset. Hasil perhitungan tersebut kemudian diurutkan sehingga sistem kemudian menampilkan daftar film dengan nilai kemiripan tertinggi sebagai rekomendasi dengan menggunakan rumus berikut.

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

di mana:

1. A = vektor dokumen pertama.
2. B = vektor dokumen kedua.
3. A·B = perkalian antara dua vector.
4. ||A|| dan ||B|| = panjang vector.

Sebagai ilustrasi, digunakan dua dokumen sinopsis film yang direpresentasikan dalam vektor berbentuk *TF-IDF* sebagai berikut:

$$A = [0.12, 0.08, 0.10, 0.05, 0.03]$$

$$B = [0.04, 0.02, 0.07, 0.01, 0.02]$$

Berdasarkan kedua vektor tersebut, langkah pertama adalah menghitung perkalian titik hasil antara vektor A dan B. Perhitungan dilakukan dengan mengalikan setiap elemen vektor A dengan elemen vektor B pada posisi yang sama, kemudian seluruh hasil perkalian dijumlahkan.

$$A \cdot B = (0.12 \times 0.04) + (0.08 \times 0.02) + (0.10 \times 0.07)$$

$$+ (0.05 \times 0.01) + (0.03 \times 0.02)$$

$$A \cdot B = 0.0048 + 0.0016 + 0.0070 + 0.0005 + 0.0006$$

$$A \cdot B = 0.0145$$

Setelah nilai titik diperoleh, langkah berikutnya adalah menghitung panjang masing-masing vektor. Panjang vektor dihitung dengan menjumlahkan kuadrat dari setiap elemen vektor, kemudian diambil nilai akarnya.

Panjang vektor A dihitung sebagai berikut:

$$\|A\| = \sqrt{(0.12)^2 + (0.08)^2 + (0.10)^2 + (0.05)^2 + (0.03)^2}$$

$$\|A\| = \sqrt{0.0144 + 0.0064 + 0.0100 + 0.0025 + 0.0009}$$

$$\|A\| = \sqrt{0.0342}$$

$$\|A\| = 0.1849$$



Selanjutnya, panjang vektor B dihitung sebagai berikut:

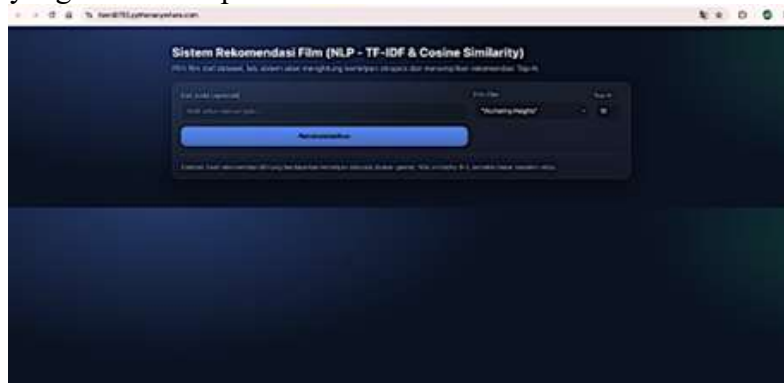
$$\begin{aligned} \|B\| &= \sqrt{(0.04)^2 + (0.02)^2 + (0.07)^2 + (0.01)^2 + (0.02)^2} \\ \|B\| &= \sqrt{0.0016 + 0.0004 + 0.0049 + 0.0001 + 0.0004} \\ \|B\| &= \sqrt{0.0074} \\ \|B\| &= 0.0860 \end{aligned}$$

Setelah diperoleh nilai *dot product*, panjang vektor A, dan panjang vektor B, langkah terakhir adalah menghitung nilai *cosine similarity* dengan mensubstitusikan seluruh nilai tersebut ke dalam rumus.

$$\begin{aligned} \text{Similarity}(A, B) &= \frac{0.0145}{0.1849 \times 0.0860} \\ \text{Similarity}(A, B) &= \frac{0.0145}{0.0159} \\ \text{Similarity}(A, B) &= 0.1591 \end{aligned}$$

Hasil perhitungan manual tersebut konsisten dengan hasil perhitungan yang dihasilkan oleh sistem, di mana salah satu film rekomendasi memperoleh nilai *similarity* sebesar 0.1591. Konsistensi tersebut menunjukkan bahwa implementasi metode *cosine similarity* pada sistem telah berjalan sesuai dengan perhitungan matematis yang digunakan.

Sistem rekomendasi film kemudian diimplementasikan dalam bentuk aplikasi berbasis web menggunakan bahasa pemrograman *Python* dengan *framework Flask* sebagai pengelola sisi backend. Antarmuka pengguna dikembangkan menggunakan teknologi *HTML*, *CSS*, dan *JavaScript* sehingga pengguna dapat memilih judul film sebagai input serta menentukan jumlah rekomendasi film yang akan ditampilkan oleh sistem.



Gambar 4. Tampilan halaman utama sistem rekomendasi film

Melalui antarmuka tersebut, sistem akan memproses sinopsis film menggunakan metode *TF-IDF* dan *cosine similarity* untuk menghitung tingkat kemiripan antar film. Hasil perhitungan kemudian diurutkan berdasarkan nilai *similarity* tertinggi sehingga diperoleh daftar film yang memiliki kemiripan cerita paling relevan dengan film yang dipilih oleh pengguna.



Gambar 5. Tampilan hasil rekomendasi film pada sistem berbasis web

Selain ditampilkan melalui antarmuka sistem, hasil rekomendasi juga dapat disajikan dalam bentuk tabel untuk menunjukkan nilai kemiripan antar film. Penyajian dalam bentuk tabel bertujuan untuk memberikan gambaran yang lebih jelas mengenai tingkat kemiripan antara film input dengan film yang direkomendasikan oleh sistem. Tabel 2 dibawah ini adalah hasil rekomendasi film berdasarkan nilai similarity

Tabel 2. Hasil rekomendasi film berdasarkan nilai similarity

No	Judul Film	Genre	Tahun	Nilai Similarity
1	Grizzly Night	Horror Thriller Drama	2026	0.1591
2	Iron Lung	Horror Science Fiction Mystery	2026	0.1199
3	Strawberry Estates	Horror	2001	0.1177
4	Nightmare Asylum	Horror	1992	0.0902
5	Raiders of the Lost Ark	Adventure Action	1991	0.0881

Nilai similarity yang ditampilkan menunjukkan tingkat kesamaan antara sinopsis film input dengan film yang direkomendasikan oleh sistem. Semakin besar nilai similarity yang dihasilkan, maka semakin tinggi tingkat kesesuaian konten cerita antara kedua film tersebut. Dengan demikian, film dengan nilai similarity tertinggi akan ditempatkan pada urutan teratas dalam daftar rekomendasi karena dianggap memiliki cerita yang paling mendekati film input.

Evaluasi sistem dilakukan untuk menilai kemampuan sistem rekomendasi film yang dikembangkan dalam menghasilkan rekomendasi yang sesuai dengan film yang dipilih oleh pengguna. Proses evaluasi ini bertujuan untuk mengetahui tingkat kualitas rekomendasi yang dihasilkan oleh sistem berdasarkan metode yang diterapkan.

Pada penelitian ini, proses evaluasi menggunakan metrik *Precision@5*, yaitu ukuran yang digunakan untuk menghitung proporsi rekomendasi yang relevan dari lima hasil rekomendasi teratas yang diberikan oleh sistem. Pemilihan metrik tersebut didasarkan pada karakteristik sistem rekomendasi yang umumnya hanya menampilkan beberapa rekomendasi utama kepada pengguna. Secara matematis, nilai *Precision@5* dapat dihitung menggunakan persamaan berikut:

$$Precision@K = \frac{\text{Jumlah rekomendasi relevan}}{\text{Jumlah rekomendasi yang ditampilkan}} \tag{4}$$

Pada penelitian ini digunakan nilai $K = 5$, sehingga metrik yang digunakan adalah *Precision@5*. Sebuah film rekomendasi dikategorikan sebagai relevan apabila memiliki minimal satu genre yang sama dengan film yang dijadikan sebagai *input* oleh pengguna. Sebaliknya, jika tidak terdapat kesamaan genre antara film rekomendasi dan film *input*, maka rekomendasi tersebut dikategorikan sebagai tidak relevan.

Sebagai pengujian menggunakan film *Train Dreams* sebagai film *input*. Sistem kemudian menghasilkan lima film rekomendasi teratas berdasarkan nilai *similarity*. Hasil pengujian pada Tabel 3.

Tabel 3. Perhitungan Precision@5

No.	Judul Film Rekomendasi	Genre	Relevan
1	Scream 2	Horror Mystery	✓
2	Even If This Love Disappears Tonight	Romance Drama	✗
3	Train Dreams	Drama	✓
4	Anyone But You	Romance Comedy	✗
5	Forrest Gump	Comedy Drama Romance	✓

Berdasarkan hasil pengujian pada Tabel 3, terdapat 3 film yang memiliki kesamaan genre dengan film input, yaitu *Scream 2*, *Train Dreams*, dan *Forrest Gump*. Dengan demikian nilai *Precision@5* dapat dihitung sebagai berikut:

$$Precision@5 = \frac{3}{5} = 0.6 \tag{5}$$



Nilai tersebut menunjukkan bahwa 60% rekomendasi yang dihasilkan oleh sistem termasuk dalam kategori relevan terhadap film input.

Selain pengujian tersebut, dilakukan pula pengujian menggunakan beberapa film input lainnya untuk mengetahui konsistensi performa sistem rekomendasi. Hasil evaluasi sistem ditunjukkan pada Tabel 4.

Tabel 4. Hasil evaluasi Precision@5

No	Film Input	Precision@5
1	A Legend	0.62
2	Train Dreams	0.60
3	Forrest Gump	0.64

Berdasarkan hasil evaluasi yang dilakukan, diperoleh nilai *Precision@5* sebesar 0.62. Nilai tersebut menunjukkan bahwa sebagian besar film yang direkomendasikan oleh sistem memiliki kesesuaian dengan film input berdasarkan kriteria relevansi yang digunakan.

Hasil penelitian ini menunjukkan bahwa metode TF-IDF dan *cosine similarity* mampu mengidentifikasi kemiripan antar film berdasarkan analisis teks sinopsis secara efektif. Pendekatan ini memberikan alternatif solusi dalam pengembangan sistem rekomendasi berbasis konten yang tidak hanya bergantung pada atribut sederhana seperti genre atau rating pengguna, tetapi juga mempertimbangkan kesamaan cerita yang terkandung dalam sinopsis film.

KESIMPULAN

Penelitian ini berhasil merancang dan mengembangkan sebuah sistem rekomendasi film berbasis web yang memanfaatkan teknik Natural Language Processing (NLP) untuk menganalisis teks sinopsis dalam mengidentifikasi kemiripan antar film. Proses pengolahan data dilakukan melalui tahapan preprocessing teks, pembentukan representasi dokumen menggunakan metode TF-IDF, serta perhitungan kemiripan dokumen menggunakan metode cosine similarity. Sistem yang dikembangkan diimplementasikan dalam bentuk aplikasi berbasis web sehingga pengguna dapat memperoleh rekomendasi film berdasarkan kemiripan konten cerita dari film yang dipilih. Hasil pengujian menunjukkan bahwa sistem mampu menghasilkan nilai similarity yang konsisten dengan perhitungan matematis yang digunakan.

Evaluasi sistem menggunakan metrik Precision@5 menghasilkan nilai sebesar 0.62, yang menunjukkan bahwa sebagian besar film yang direkomendasikan memiliki kesesuaian dengan film input berdasarkan kesamaan genre. Hasil tersebut menunjukkan bahwa pendekatan berbasis analisis teks menggunakan metode TF-IDF dan cosine similarity dapat digunakan untuk menghasilkan rekomendasi film yang cukup relevan berdasarkan kemiripan sinopsis.

DAFTAR PUSTAKA

- [1] Agarwal, A., Mishra, D. S., & Kolekar, S. V. (2022). Knowledge-based recommendation system using semantic web rules based on Learning styles for MOOCs. *Cogent Engineering*,9(1).<https://doi.org/10.1080/23311916.2021.2022568>
- [2] Aizawa, A. (2021). An information-theoretic perspective of tf-idf measures q. Retrieved from www.elsevier.com/locate/infoproman
- [3] Ayyiyah, N. K., Kusumaningrum, R., & Rismiyati, R. (2023). Film Recommender System Menggunakan Metode Neural Collaborative Filtering. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 10(3), 699–708.<https://doi.org/10.25126/jtiik.2023106616>
- [4] Azri Saputra, J. M., Huizen, L. M., & Arianto, D. B. (2024). Sistem Rekomendasi Film pada Platform Streaming Menggunakan Metode Content-Based Filtering. *Jurnal Transformatika*,22(1),1021,<https://doi.org/10.26623/transformatika.v22i1.7041>



- [5] Dai, H. J., Su, C. H., Lee, Y. Q., Zhang, Y. C., Wang, C. K., Kuo, C. J., & Wu, C. S. (2021). Deep Learning-Based Natural Language Processing for Screening Psychiatric Patients. *Frontiers in Psychiatry*, 11. <https://doi.org/10.3389/fpsy.2020.533949>
- [6] Harahap, R. M., & Rachman, A. N. (2025). SISTEM REKOMEDASI FILM BERDASARKAN KEMIRIPAN DESKRIPSI CERITA MENGGUNAKAN METODE CONTENT BASED FILTERING. *Jurnal Informatika Dan Teknik Elektro Terapan*, 13(3). <https://doi.org/10.23960/jitet.v13i3.6577>
- [7] Harjo Utomo, R., Susrama, G., Diyasa, M., Pembangunan, U., Veteran, N. ", & Timur, J. (2024). MOVIE MU : SISTEM REKOMENDASI FILM MENGGUNAKAN ALGORITMA COSINE SIMILARITY. 4(2), 22–32.
- [8] Holis, R. M., Utomo, P. E. P., & Hutabarat, B. F. (2025). Semantic FAQ Chatbot Using SBERT (Sentence-BERT) and Cosine Similarity for Academic Services. *Brilliance: Research of Artificial Intelligence*, 5(2), 915–922. <https://doi.org/10.47709/brilliance.v5i2.7027>
- [9] Javed, U., Shaukat, K., Hameed, I. A., Iqbal, F., Alam, T. M., & Luo, S. (2021). A Review of Content-Based and Context-Based Recommendation Systems. *International Journal of Emerging Technologies in Learning*, 16(3), 274306. <https://doi.org/10.3991/ijet.v16i03.18851>
- [10] Juni Permana, A. H. J. P., & Agung Toto Wibowo. (2023). Movie Recommendation System Based on Synopsis Using Content-Based Filtering with TF-IDF and Cosine Similarity. *International Journal on Information and Communication Technology(IJoICT)*, 9(2), 114. <https://doi.org/10.21108/ijoiect.v9i2.747>
- [11] Konstan, J. A. (2004). Introduction to recommender systems: Algorithms and evaluation. In *ACM Transactions on Information Systems* (Vol. 22, Number 1, pp. 1–4). <https://doi.org/10.1145/963770.963771>
- [12] Li, Jiacheng, Wang, M., Li, Jin, Fu, J., Shen, X., Shang, J., & McAuley, J. (2023). Text Is All You Need: Learning Language Representations for Sequential Recommendation. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 12581267. <https://doi.org/10.1145/3580305.3599519>
- [13] Lops, P., Jannach, D., Musto, C., Bogers, T., & Koolen, M. (2019). Trends in content-based recommendation: Preface to the special issue on Recommender systems based on rich item descriptions. In *User Modeling and User-Adapted Interaction* (Vol. 29, Number 2, pp. 239–249). Springer Netherlands. <https://doi.org/10.1007/s11257-019-09231-w>
- [14] Mulwa, C., Lawless, S., Sharp, M., & Wade, V. (2011). A web-based framework for user-centred evaluation of end-user experience in adaptive and personalized e-Learning systems. *Proceedings - 2011 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Workshops, WI-IAT 2011*, 3, 351–356. <https://doi.org/10.1109/WI-IAT.2011.203>
- [15] Perdana Sari, Y. (2024). Sistem Rekomendasi Drama dan Film Berbasis Website Dengan Metode Content Based Filtering. *OKTAL : Jurnal Ilmu Komputer Dan Science*, 3(10).
- [16] Salsabil, A., Setiawan, E B, Kurniawan, I, Salsabil, A. A., Setiawan, Erwin Budi, & Kurniawan, Isman. (2019). Kinetik: Game Technology, Information System. *Computer Network, Computing, Electronics, and Control Journal*, 4(3), 193–202. <https://kinetik.umm.ac.id/index.php/kinetik/article/view/1940> <https://kinetik.umm.ac.id/index.php/kinetik/article/view/1940>
- [17] Srivastava, R. P. (n.d.). A New Measure of Similarity in Textual Analysis: Vector Similarity Metric versus Cosine Similarity Metric A New Measure of Similarity in Textual Analysis: Vector Similarity Metric versus Cosine Similarity Metric A New Measure of Similarity in Textual Analysis: Vector Similarity Metric versus Cosine Similarity Metric.
- [18] Theo Santoso, D., Atina, V., & Hartanti, D. (2024). Prototipe Sistem Rekomendasi Film Indonesia Menggunakan Pendekatan Content Based Filtering dan Metode Vector Space Model. *Infotek: Jurnal Informatika Dan Teknologi*, 7(2), 444–455. <https://doi.org/10.29408/jit.v7i2.26083>



- [19] Velamentosa, D., Zuliarso, E., & Raya Tri Lomba Juang, J. (2025). SISTEM REKOMENDASI FILM MENGGUNAKAN METODE CONTENT-BASED FILTERING. In Jurnal Mahasiswa Teknik Informatika) (Vol. 9, Number 2).
- [20] S. Syam and M. Rabidin, “Metode Simple Wahyu Intan Permadani, T., Adi Prasetya, D., Daniati, E., & Korespondens, P. (2025). Sistem Rekomendasi Film Berdasarkan Genre Menggunakan Metode Content-Based Filtering dengan Algoritma Cosine Similarity. In INOTEK (Vol. 9).

