

# **Perancangan Sistem Rekomendasi Film menggunakan metode Content-based Filtering**

*Evan Salim<sup>1</sup>, Jeanny Pragantha<sup>2</sup>, Manatap Dolok Lauro<sup>3</sup>*

Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara  
Jl. Letjen S. Parman No. 1, Jakarta Barat 11440

<sup>1</sup>*Email: evan.535190012@stu.untar.ac.id*

<sup>2</sup>*Email: jeannyp@fti.untar.ac.id*

<sup>3</sup>*Email: manataps@fti.untar.ac.id*

## **ABSTRAK**

Film merupakan salah satu media hiburan visual yang digemari masyarakat. Perkembangan film terutama pada era modern ini mulai berkembang cepat dengan mulai munculnya berbagai judul yang tak terbatas. Hal ini yang menjadi alasan mengapa sulit bagi orang untuk mencari suatu judul film. Oleh karena itu, dikembangkannya berbagai macam ragam sistem salah satunya adalah sistem rekomendasi. Sistem rekomendasi merupakan salah satu cara mencari rekomendasi untuk membantu orang merekomendasikan suatu judul film secara mudah dan efisien. Sistem rekomendasi ini dirancang menggunakan metode Content-Based Filtering, hal ini bertujuan supaya film yang direkomendasikan dipastikan mirip dengan judul, genre atau sinopsis film yang diberikan oleh pengguna. Dalam makalah ini, sistem rekomendasi menggunakan *dataset* yang berasal dari The Movie Database. Sistem dibangun berdasarkan gabungan antara nama direktur, aktor, dan genre film yang mungkin disukai pengguna untuk ditonton.

**Kata Kunci:** Film; Sistem rekomendasi; Content-Based Filtering; The Movie Database

## **ABSTRACT**

*Movie is one of the visual entertainment media that is popular with the public. The development of movies, especially in this modern era, produced a lot of movie titles. Sometimes people difficult to find a movies title. Therefore, various kinds of systems have been developed, one of which is a recommendation system. The recommendation system is a way to help people to look a movie title. This system provide recommendations about movie title easily and efficiently. This recommendation system is designed using the Content-Based Filtering method, this will ensure that the recommended movies are similar to the title, genre or synopsis of the movies provided by the user. In this paper, the recommendation system uses a dataset derived from The Movie Database. The system is built based on a combination of names of directors, actors, and movie genres that users might like to watch.*

**Keywords:** Film; Recommendation system; Content-Based Filtering; The Movie Database

## **1. PENDAHULUAN**

### **1.1. Latar Belakang**

Sistem rekomendasi memberikan saran untuk item yang paling mungkin menarik bagi *User* tertentu. Sistem rekomendasi memiliki kemampuan untuk memprediksi konten atau item sesuai dengan interaksi *User* berdasarkan informasi yang diberikan oleh *User* sebelumnya. Item tersebut, terutama dalam kasus ini merekomendasikan judul *film* berdasarkan konten yang memiliki keterkaitan yang sesuai dengan panggilan yang direkomendasikan [1].

Dalam perancangan ini, sistem rekomendasi yang dirancang menggunakan metode Content-based Filtering yang memberikan rekomendasi berdasarkan kemiripan atribut dari item atau item yang disukai [2]. Sistem ini menentukan rekomendasi pada item pada film yaitu, genre pada film. Selain itu, data yang diperoleh akan diambil melalui website Kaggle dan profil pada film diambil melalui web The Movies Database (TMDb).

### **1.2. Rumusan Masalah**

Sistem rekomendasi yang di rancang merupakan sistem rekomendasi berbasis web dan dirancang menggunakan program Python dengan mengambil *dataset* film melalui Kaggle. Kaggle merupakan website komunitas yang memiliki kumpulan *dataset*. Kaggle di rancang untuk memudahkan *programmer* dalam merancangkan suatu aplikasi tanpa membuat format sendiri.

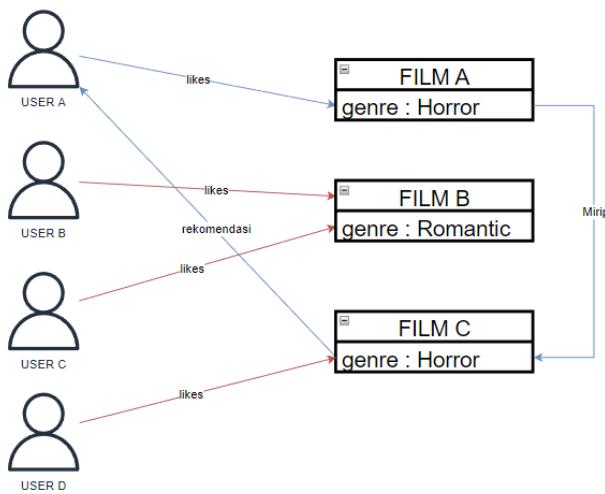
## **2. METODE PENELITIAN**

Ada beberapa tahapan yang dipakai dalam rancangan sistem rekomendasi film dapat disampaikan pada metode berikut ini.

### **2.1. Content-Based Filtering**

Content-Based Filtering adalah metode yang dapat menghasilkan rekomendasi yang bersifat independen kepada pengguna. Content-based Filtering merekomendasikan item yang mirip dengan item lainnya yang sesuai dengan peminatan pengguna. Sistem ini dapat merekomendasikan film berdasarkan perbandingan antara profil item dan profil *User* [3].

Profil *User* mengandung konten yang dapat ditemukan secara relevan dengan *User* dalam bentuk kata kunci (atau fitur). Profil *User* dapat dilihat sebagai sekumpulan kata kunci yang diterapkan (istilah, fitur) dan dikumpulkan oleh algoritma melalui item yang dianggap menarik oleh *User*. Dari rancangan yang diterapkan, ada beberapa komponen yang digunakan sebelum melakukan implementasi sistem rekomendasi film. Konsep Content-Based Filtering yang digunakan adalah rekomendasi film dapat dilihat pada **gambar 1** di bawah ini. Dianggap *user A* sebagai *user* utama, setelah *user* tersebut menyukai film Adengan genre horror, maka sistem akan menanggapi akan menanggapi dengan mencari film lain dengan genre yang sama, yaitu film C.



**Gambar 1.** Konsep cara kerja Content-based Filtering

Sistem ini berbasis web sehingga dapat diakses oleh banyak orang. Pembuatan sistem rekomendasi film menggunakan pemrograman Python. Dalam web ini, *user* dapat mencari judul film sesuai dengan keinginan-Nya. Setelah judul tersebut ditemukan, sistem akan menampilkan beberapa judul yang direkomendasikan.

## 2.2. Term Frequency–Inverse Document Frequency

Term Frequency–Inverse Document Frequency (TF-IDF) merupakan suatu metode untuk memberikan bobot hubungan suatu kata (term) terhadap dokumen. Metode ini merupakan tata penting cara untuk mengimplementasi Cosine-Similarity ke dalam sistem rekomendasi karena tingkat akurasi sangat tinggi dan efisien. TF-IDF dapat mengevaluasi beberapa kalimat yang dianggap penting dalam suatu kumpulan data yang besar. Metode ini menggabungkan dua konsep untuk perhitungan bobot, yaitu frekuensi kemunculan sebuah kata di dalam sebuah dokumen tertentu dan *inverse* frekuensi dokumen yang mengandung kata tersebut [4].

Rumus perhitungan TF-IDF:

$$W(t, d) = tf(t, d) * idf(t) \quad (1)$$

Keterangan :

d = data ke-d

t = teks ke-t dari kata kunci

W = bobot dokumen ke-d terhadap kata ke-t

tf = term frekuensi

idf= inverse document frequency

Setelah bobot (W) diketahui, maka dilakukan proses *sorting* di mana nilai W semakin besar, semakin besar tingkat kesamaan data tersebut terhadap kata yang dicari, demikian pula sebaliknya.

## 2.3. Cosine-Similarity

Cosine-similarity merupakan metode yang dapat digunakan untuk melihat kesamaan konten antar data[5]. Penggunaan Content-based Filtering dibutuhkan sebelum melakukan implementasi karena

Cosine-similarity mempunyai konsep normalisasi panjang vektor dengan membandingkan dua dokumen antara dua vektor A dan B.

Cosine-similarity digunakan sebagai sistem membandingkan antar *text to text* untuk mengetahui apakah antara dokumen serupa. Kesamaan *cosinus* digunakan dalam ruang positif. Sehingga persamaan *cosinus*-nya sebanding tingkat kemiripan dokumen menggunakan konsep derajat kosinus di mana nilai memiliki batas antara nilai 0 dan 1. Jika hasil *cosinus similarity* adalah 0, dikatakan tidak serupa. Jika hasil dari kesamaan *cosinus* adalah 1, maka dokumen dianggap serupa [6].

Rumus perhitungan *Cosine-similarity*:

$$\text{similarity}(x, y) = \frac{\sum x_i \cdot y_i}{\sqrt{\sum x_i^2} \cdot \sqrt{\sum y_i^2}} \quad (2)$$

Keterangan :

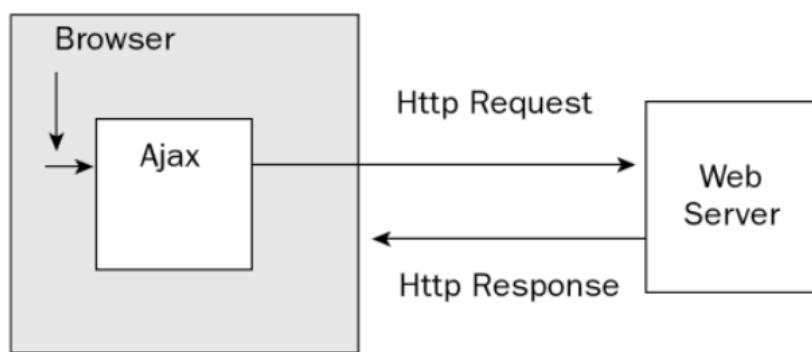
$\sum x_i \cdot y_i$  = vektor (dot) produk dari vektor x dan y

$\sqrt{\sum x_i^2} \cdot \sqrt{\sum y_i^2}$  = panjang produk antara vektor x dan y

## 2.4. Asynchronous JavaScript and XML

Asynchronous JavaScript and XML atau disebut AJAX merupakan sistem pendukung yang digunakan untuk memudahkan perancangan aplikasi web yang pada umumnya digunakan untuk mengambil data dari server. AJAX pada umumnya dibuat dengan Javascript yang berfungsi untuk memberikan tampilan kepada *user* dan berkomunikasi dengan server secara langsung.

Dengan menggunakan AJAX, interaksi antara *client* dan server dapat berlangsung terus menurut secara tidak langsung (*asynchronous*). Sehingga untuk mengirimkan dan menerima data dari server tidak merepotkan *user experience*. AJAX dapat meng-update tampilan browser *client* beberapa waktu sekali sesuai dengan informasi yang didapatkan dari server, dan *client* tidak perlu pindah ke halaman lain ketika melakukan *submit* ke server [7].



**Gambar 1.** Model aplikasi AJAX

Sumber Gambar: C. Ullman and L. Dykes, *Beginning Ajax*. John Wiley & Sons, 2007.

### 3. HASIL DAN PEMBAHASAN

#### 3.1. Preprocesing dataset

Data yang dipakai dalam rancangan ini diambil melalui web Kaggle. Ada 2 *dataset* yang akan dipakai, tiap data tersebut memiliki lebih dari 5000 data film yang mengandung judul, tahun terbit, sinopsis, nama aktor, nama direktur, dan genre. Data pertama merupakan data film dapat dilihat pada **gambar 2** dan yang kedua merupakan data kredit orang-orang yang terlibat dalam suatu film pada **gambar 3**.

adult	belongs_to_collection	budget	genres	homepage	id	...
False	[{"id": 10194, "name": "Toy Story Collection", "poster_path": "/7G9915LfUQzvqg..."}]	30000000	[{"id": 16, "name": "Animation"}, {"id": 35, "name": "Comedy"}, {"id": 10751, "name": "Family"}]	http://toystory.disney.com/toy-story	862	...

**Gambar 2.** contoh sampel dataset film

...	imdb_id	original_language	original_title	overview	popularity	poster_path	...
...	tt0114709	en	Toy Story	Led by Woody, Andy's toys live happily in his ...	21,94694	/rhIRbceoE9IR4veEXuwCC2wARTG.jpg	...

**Gambar 3.** (lanjutan)

...	production_companies	production_countries	release_date	revenue	runtime	spoken_languages	...
...	[{"name": "Pixar Animation Studios", "id": 3}]	[{"iso_3166_1": "US", "name": "United States of America"}]	30/10/1995	3,74E+08	81	[{"iso_639_1": "en", "name": "English"}]	...

**Gambar 4.** (lanjutan)

...	status	tagline	title	video	vote_average
...	Released		Toy Story	False	7,7

**Gambar 5.** (lanjutan)

cast	crew	id
[{"cast_id": 14, "character": "Woody (voice)", "credit_id": "52fe4284c3.."}]	[{"credit_id": "52fe4284c3a36847f8024f49", "department": "Directing", "id": 862} ...]	862

**Gambar 6.** contoh sampel dataset kredit

Ada beberapa tahap untuk membersihkan data diantaranya.

### 1. Reduksi data

Tahap Reduksi data mengurangi volume data yang berlebihan tetapi tetap mempertahankan kualitas dari hasil analisis data. Oleh karena itu, perlu adanya teknik data *reduction* dengan tujuan untuk meningkatkan efisiensi penyimpanan serta mengurangi biaya penyimpanan dan analisis data.

release_date
1995-10-30

→

release_date
1995

**Gambar 7.** contoh reduksi data

### 2. Integrasi data

Tahap Integrasi data menggabungkan beberapa *database* dan *file* menjadi 1 sehingga menghasilkan sumber data yang besar. Pada tahap ini, dilakukan penggabungan dengan *dataset* pertama pada **gambar 2** dengan kedua pada **gambar 3** berdasarkan ‘id’.

genres	id	title	year	cast	crew	id
[{"id": 16, "name": "Animation"}, {"id": 35, "name": "Comedy"}]	862	Toy Story	1995.0	[{"cast_id": 14, "character": "Woody (voice)", "credit_id": "52fe4284c3.."}]	[{"credit_id": "52fe4284c3a36847f8024f49", "department": "Directing", "id": 862, "name": "..."}]	

**Gambar 8.** Tahap integrasi dataset

genres	id	title	year	cast	crew
[{"id": 16, "name": "Animation"}, {"id": 35, "name": "Comedy"}]	862	Toy Story	1995.0	[{"cast_id": 14, "character": "Woody (voice)", "credit_id": "52fe4284c3.."}]	[{"credit_id": "52fe4284c3a36847f8024f49", "department": "Directing", "id": 862, "name": "..."}]

**Gambar 9.** dataset setelah integrasi

### 3. Pembersihan data

Pembersihan Data mengubah *missing value*, menormalkan data yang bermasalah, mengidentifikasi dan menghilangkan data yang tidak konsisten dan data yang berulang.

	genres	id	title	year	overview	cast	crew
0	[{"id": 16, "name": "Animation"}, {"id": 35, "name": "Comedy"}, {"id": ...]	862	Toy Story	1995.0	Led by Woody, Andy's toys live happily in his room until Andy's birthday...	[{"cast_id": 14, "character": "Woody (voice)", "credit_id": "52fe4284c3a36847f8024f49", ...]	[{"credit_id": "52fe4284c3a36847f8024f49", "department": "Directing", ...}
1	[{"id": 12, "name": "Adventure"}, {"id": 14, "name": "Fantasy"}, {"id": ...]	8844	Jumanji	1995.0	When siblings Judy and Peter discover an enchanted board game that opens...	[{"cast_id": 1, "character": "Alan Parrish", "credit_id": "52fe44bfc3a36847f80a7cd1", ...]	[{"credit_id": "52fe44bfc3a36847f80a7cd1", "department": "Production", ...}
2	[{"id": 10749, "name": "Romance"}, {"id": 35, "name": "Comedy"}]	15602	Grumpier Old Men	1995.0	A family wedding reignites the ancient feud between next-door neighbors...	[{"cast_id": 2, "character": "Max Goldman", "credit_id": "52fe466a9251416c75077a89", ...]	[{"credit_id": "52fe466a9251416c75077a89", "department": "Directing", ...}
3	[{"id": 35, "name": "Comedy"}, {"id": 18, "name": "Drama"}, {"id": 1074...	31357	Waiting to Exhale	1995.0	Cheated on, mistreated and stepped on, the women are holding their breath...	[{"cast_id": 1, "character": "Savannah Vannah Jackson", "credit_id": ...}]	[{"credit_id": "52fe44779251416c91011ac8", "department": "Directing", ...}
4	[{"id": 35, "name": "Comedy"}]	11862	Father of the Bride Part II	1995.0	Just when George Banks has recovered from his daughter's wedding, he re...	[{"cast_id": 1, "character": "George Banks", "credit_id": "52fe44959251416c75039ed7", ...}]	[{"credit_id": "52fe44959251416c75039ed7", "department": "Sound", "gen...

Gambar 10. dataset sebelum proses *cleaning*

Proses *cleaning* dilakukan menggunakan Jupyter Notebook supaya dalam tahap pen-deploy-an aplikasi dalam sistem dapat memberikan nilai kemiripan yang optimal, serta mengurangi beban memori pada preprocessing sistem rekomendasi.

	director_name	actor_1_name	actor_2_name	actor_3_name	genres	movie_title	overview	comb
0	John Lasseter	Tom Hanks	Tim Allen	Don Rickles	Animation Comedy Family	toy story	Led by Woody, Andy's toys live happily in his room until Andy's birthday...	Tom Hanks Tim Allen Don Rickles John Lasseter Animation Comedy Family
1	Joe Johnston	Robin Williams	Jonathan Hyde	Kirsten Dunst	Adventure Fantasy Family	jumanji	When siblings Judy and Peter discover an enchanted board game that opens...	Robin Williams Jonathan Hyde Kirsten Dunst Joe Johnston Adventure Fanta...
2	Howard Deutch	Walter Matthau	Jack Lemmon	Ann-Margret	Romance Comedy	grumpier old men	A family wedding reignites the ancient feud between next-door neighbors...	Walter Matthau Jack Lemmon Ann-Margret Howard Deutch Romance Comedy
3	Forest Whitaker	Whitney Houston	Angela Bassett	Loretta Devine	Comedy Drama Romance	waiting to exhale	Cheated on, mistreated and stepped on, the women are holding their breath...	Whitney Houston Angela Bassett Loretta Devine Forest Whitaker Comedy Dr...
4	Charles Shyer	Steve Martin	Diane Keaton	Martin Short	Comedy	father of the bride part ii	Just when George Banks has recovered from his daughter's wedding, he re...	Steve Martin Diane Keaton Martin Short Charles Shyer Comedy

Gambar 11. dataset setelah proses cleaning

Data yang telah dibersihkan, dapat dilanjutkan pada tahap preproccesing

### 3.2. Proses Content-based Filtering

Contoh perhitungan manual pada sistem rekomendasi Content-based Filtering.

Tabel 2. Contoh tabel queri

output	title	actor_1	actor_2	actor_3	director	genres
Queri	Iron Man	Robert Downey Jr.	Jeff Bridges	Jon Favreau	Jon Favreau	Action Adventure Sci-Fi
1	Iron Man 2	Robert Downey Jr.	Scarlett Johansson	Jon Favreau	Jon Favreau	Action Adventure Sci-Fi
2	Iron Man 3	Robert Downey Jr.	Jon Favreau	Don Cheadle	Shane Black	Action Adventure Sci-Fi

3	Made	Jon Favreau	Faizon Love	Jonathan Silverman	Jon Favreau	Comedy Crime Drama Thriller
---	------	-------------	-------------	--------------------	-------------	--------------------------------------

Dianggap *Queri* (Q) merupakan *input* judul film yang dicari oleh *user*, dan *D1* hingga *D3* merupakan judul film yang akan direkomendasikan dan total dokumen dalam dataset (N) memiliki 5000 data. Untuk kasus ini, data yang direkomendasikan dan digabungkan berdasarkan nama direktur dan genre seluruh film. Tahap pertama dalam perhitungan manual sebelum mencari nilai *similarity* tiap dokumen dapat disampaikan sebagai berikut.

## 1. Perhitungan frekuensi data

**Tabel 3.** Contoh tabel dokumen film dan penjumlahan frekuensi data (df)

Term (t)	D				Df
	Q	D1	D2	D3	
Jon	1	1	0	1	3
Favreau	1	1	0	1	3
Shane	0	0	1	0	1
Black	0	0	1	0	1
Action	1	1	1	1	4
Adventure	1	1	1	0	3
Sci-Fi	1	1	1	0	3
Comedy	0	0	0	1	1
Crime	0	0	0	1	1
Drama	0	0	0	1	1
Thriller	0	0	0	1	1
dt =	5	5	5	7	

Dari tabel di atas, dapat dipersingkat bahwa D1 berjudul "Iron Man 2", D2 berjudul "Iron Man 3", D3 berjudul "Made", dan *Queri* (Q) berjudul "Iron Man". Selanjutnya, sistem mengolah untuk membuat vektor frekuensi seluruh dokumen.

## 2. Perhitungan TF-IDF dan Cosine Similarity

**Tabel 4.** Contoh tabel pencarian Tf- Idf

tf = D/dt				idf = $\log(N/df)$	W = tf*idf			
Q	D1	D2	D3		Q	D1	D2	D3
0,2	0,2	0	0,14	3,22184875	0,64437	0,64437	0	0,460264
0,2	0,2	0	0,14	3,22184875	0,64437	0,64437	0	0,460264
0	0	0,2	0	3,698970004	0	0	0,739794	0
0	0	0,2	0	3,698970004	0	0	0,739794	0

0,2	0,2	0,2	0,14	3,096910013	0,619382	0,619382	0,619382	0,442416
0,2	0,2	0,2	0	3,22184875	0,64437	0,64437	0,64437	0
0,2	0,2	0,2	0	3,22184875	0,64437	0,64437	0,64437	0
0	0	0	0,14	3,698970004	0	0	0	0,528424
0	0	0	0,14	3,698970004	0	0	0	0,528424
0	0	0	0,14	3,698970004	0	0	0	0,528424
0	0	0	0,14	3,698970004	0	0	0	0,528424

Dengan menggunakan data pada **tabel 2**, setelah perhitungan TF-IDF terolah, langkah terakhir dalam memberikan rekomendasi ini mencari kemiripan teks pada dokumen *Queri* menggunakan Cosine-similarity. Contoh pencarian nilai *similarity* pada *D3* dapat disampaikan pada perhitungan di bawah ini:

$$sim(x, y) = \frac{\sum x_i \cdot y_i}{\sqrt{\sum x_i^2} \cdot \sqrt{\sum y_i^2}}$$

$$sim(D3, Q) = \frac{\sum D3_i \cdot Q_i}{\sqrt{\sum D3_i^2} \sqrt{\sum Q_i^2}}$$

$$sim(D3, Q) = \frac{0,86718}{\sqrt{2,04448} \sqrt{1,73635}}$$

$$sim(D3, Q) = \frac{0,86718}{1,42985 \cdot 1,31771}$$

$$sim(D3, Q) = 0,46026$$

### 3.3. Hasil Similarity

**Tabel 5.** Contoh output hasil rekomendasi pencarian "Iron Man"

output	title	director	genres	similarity
Queri	Iron Man	<u>Jon Favreau</u>	<u>Action</u> <u>Adventure</u> <u>Sci-Fi</u>	main
1	Iron Man 2	<u>Jon Favreau</u>	<u>Action</u> <u>Adventure</u> <u>Sci-Fi</u>	1.0
2	Iron Man 3	Shane Black	<u>Action</u> <u>Adventure</u> <u>Sci-Fi</u>	0.56
3	Made	<u>Jon Favreau</u>	Comedy Crime Drama Thriller	0.46

Pada tabel di atas merupakan hasil rekomendasi yang terdekat dari *Queri* "Iron Man" menggunakan Cosine Similarty. Dapat dilihat bahwa film "Iron Man 2" memiliki kemiripan 100%. Dibandingkan

dengan film yang identik “Iron Man 3” hanya memiliki kemiripan sebesar 56%, hal ini disebabkan karena ketidakmiripan dua *term* nama direktur dari kedua film. Sehingga, perbedaan antara kedua film sangat signifikan.

#### 4. KESIMPULAN DAN SARAN

Berdasarkan hasil rekomendasi yang dijalankan, dapat disampaikan dengan kesimpulan sebagai berikut:

1. Sistem dapat menghasilkan nilai akurasi yang sangat akurat.
2. Dengan menggunakan metode Content-Based Filtering, sangat mudah terhadap pengguna dalam mencari judul film yang mirip. walaupun film yang direkomendasikan sudah lampau atau baru dipublikasikan.
3. Penggunaan *dataset* yang diperhitungkan dapat diperbanyak variasi. Seperti penambahan tahun terbit, bahasa, dan *reviews* supaya nilai kemiripan antar film menjadi maksimal.

Saran dalam perancangan sistem rekomendasi ini, dapat disimpulkan bahwa berdasarkan hasil pengujian secara manual, bahwa sistem yang dapat memberikan rekomendasi rancangan ini dapat direalisasikan menjadi sistem aplikasi.

#### REFERENSI

- [1] F. Ricci, L. Rokach, and B. Shapira, “Recommender Systems: Techniques, Applications, and Challenges,” *Recommender Systems Handbook*, pp. 1–35, Nov. 2021, doi: 10.1007/978-1-0716-2197-4\_1.
- [2] D. Daniel, B. Mulyawan, and T. Sutrisno, “PEMBUATAN APLIKASI E-COMMERCE BERBASIS WEB DENGAN FITUR REKOMENDASI MENGGUNAKAN METODE CONTENT-BASED FILTERING,” *Jurnal Ilmu Komputer dan Sistem Informasi*, vol. 10, no. 1, Mar. 2022, doi: 10.24912/jiksi.v10i1.17837.
- [3] R. H. Mondi, A. Wijayanto, and W. Winarno, “RECOMMENDATION SYSTEM WITH CONTENT-BASED FILTERING METHOD FOR CULINARY TOURISM IN MANGAN APPLICATION,” *ITSMART: Jurnal Teknologi dan Informasi*, vol. 8, no. 2, pp. 65–72, Jul. 2020, doi: 10.20961/itsmart.v8i2.35008.
- [4] M. Nurjannah, H. Hamdani, and I. F. Astuti, “PENERAPAN ALGORITMA TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF) UNTUK TEXT MINING,” *Informatika Mulawarman : Jurnal Ilmiah Ilmu Komputer*, vol. 8, no. 3, pp. 110–113, Jun. 2016, doi: 10.30872/jim.v8i3.113.
- [5] M. A. Budiman and Gst. A. V. Mastrika Giri, “Song Recommendations Based on Artists with Cosine Similarity Algorithms and K-Nearest Neighbor,” *JELIKU (Jurnal Elektronik Ilmu Komputer Udayana)*, vol. 8, no. 4, p. 367, Feb. 2020, doi: 10.24843/jlk.2020.v08.i04.p01.
- [6] I. Prisman, D. Prehanto, D. Dermawan, A. Herlingga, and S. Wibawa, “Nazief & Adriani Stemming Algorithm With Cosine Similarity Method For Integrated Telegram Chatbots With Service,” *IOP Conference Series: Materials Science and Engineering*, vol. 1125, no. 1, p. 012039, May 2021, doi: 10.1088/1757-899x/1125/1/012039.
- [7] C. Ullman and L. Dykes, *Beginning Ajax*. John Wiley & Sons, 2007.