

SISTEM REKOMENDASI FILM MENGGUNAKAN CONTENT-BASED FILTERING PADA FRAMEWORK GRADIO

¹Muhammad Irvan Arfirza, ²Dr. Dina Indarti, S.Si., M.Si., M.T.

^{1,2} Jurusan Informatika Fakultas Teknologi Industri Universitas Gunadarma

Jl. Margonda Raya No. 100, Depok 16424, Jawa Barat

¹firzzairvn@gmail.com, ²dina_indarti@staff.gunadarma.ac.id

Abstrak

Film merupakan media komunikasi visual bergerak yang mengandung elemen-elemen seperti gambar, suara, dan teks untuk menyampaikan cerita atau informasi kepada penonton. Kemudahan aksesibilitas terhadap film memungkinkan penonton memiliki berbagai preferensi tontonan. Namun, ini juga dapat menimbulkan kebingungan dalam memilih film yang diinginkan, sering kali memerlukan pencarian review atau konsultasi dengan teman. Tantangan lainnya adalah biasanya rekomendasi yang diberikan oleh beberapa media digital yang dapat berkolaborasi dengan pemilik atau penerbit film untuk tujuan promosi, mengurangi validitas rekomendasi tersebut. Untuk mengatasi masalah ini, penelitian ini mengembangkan sistem rekomendasi berdasarkan kemiripan film. Metode yang digunakan adalah content-based filtering dengan algoritma TF-IDF dan cosine similarity, yang mengandalkan konten dari film untuk merekomendasikan film serupa. Penelitian ini menggunakan dataset dari Grouplens dengan nama dataset yaitu MovieLens 25M. Dataset ini mencakup 25 juta rating untuk film dari tahun 1995 hingga 2019 yang menyediakan data rating, judul film, dan tag. Model rekomendasi ini mampu menyediakan rekomendasi 10 film dengan similaritas tertinggi berdasarkan konten film. Penelitian ini juga menguji model rekomendasi menggunakan framework Gradio untuk memvalidasi kinerja sistem rekomendasi yang dikembangkan. Hasil penelitian menunjukkan bahwa sistem rekomendasi film yang dikembangkan dapat berjalan dengan baik pada framework Gradio.

Kata Kunci : Film, Sistem Rekomendasi, Content-based Filtering, Gradio

Abstract

Film is a moving visual communication medium that contains elements such as images, sound, and text to convey stories or information to the audience. The ease of accessibility to movies allows viewers to have a variety of viewing preferences. However, this can also lead to confusion in selecting the desired movie, often requiring a search for reviews or consultation with friends. Another challenge is the bias of recommendations provided by some digital media that may collaborate with movie owners or publishers for promotional purposes, reducing the validity of such recommendations. To address these issues, this research develops a recommendation system based on movie similarity. The

method used is content-based filtering with TF-IDF and cosine similarity algorithms, which rely on the content of movies to recommend similar movies. This research uses a dataset from Grouplens with the dataset name MovieLens 25M. This dataset includes 25 million ratings for movies from 1995 to 2019 that provide rating data, movie titles, and tags. This recommendation model is able to provide recommendations for 10 movies with the highest similarity based on movie content. This research also tests the recommendation model using the Gradio framework to validate the performance of the developed recommendation system. The results show that the developed movie recommendation system can run well on the Gradio framework.

Keywords: *Movies, Recommendation System, Content-based Filtering, Gradio*

PENDAHULUAN

Industri perfilman telah mengalami banyak kemajuan sejak awal kemunculannya pada abad ke-19. Setiap tahunnya, ribuan film diproduksi dan dirilis di seluruh dunia, dengan tren dari film disetiap tahunnya berganti-ganti seperti *horror*, *superhero*, *comedy* dan lainnya dengan begitu penonton akan memiliki lebih banyak pilihan film. Dengan perkembangan teknologi yang ada penonton semakin memudahkan dalam mengakses suatu film contoh yang dahulunya untuk menonton sebuah film, penonton harus datang langsung ke bioskop namun sekarang penonton dapat mengakses film hanya menggunakan *smartphone* saja.

Dengan semakin mudahnya penonton dalam mengakses sebuah film maka penonton akan memiliki semakin banyak preferensi tontonan yang mereka bisa pilih. Namun dengan begitu muncul satu permasalahan baru yaitu

bagaimana ketika pengguna suka terhadap suatu film dan ingin menonton film yang mirip dengan film yang penonton suka, maka dengan permasalahan tersebut sebuah sistem rekomendasi perlu untuk dikembangkan. Sistem rekomendasi berfungsi untuk membantu pengguna dalam mendapatkan suatu rekomendasi berdasarkan kemiripan dari *item* atau *content* dari suatu film.

Diketahui bahwa terkadang sebagai pengguna cukup merasa bingung tentang rekomendasi film yang diinginkan, lalu biasanya dilakukan pencarian *review* atau ditanyakan kepada teman untuk mendapatkan suatu rekomendasi. Dengan mencari *review*, menonton rekomendasi film di internet atau bertanya kepada teman, terkadang ditemukan masalah baru yaitu rekomendasi yang bias. Terkadang beberapa *media digital* yang

memberikan rekomendasi film sudah berkolaborasi dengan pemilik atau *publisher* dari film tersebut untuk melakukan promosi terhadap film tersebut, dengan begitu orang yang melakukan *review* film akan mencoba untuk merekomendasikan film tersebut yang artinya rekomendasi yang diberikan bisa dikatakan kurang valid.

Berdasarkan permasalahan tersebut dibutuhkan suatu sistem rekomendasi yang memberikan suatu rekomendasi kepada pengguna berdasarkan kemiripan dari film yang mereka inginkan. Model rekomendasi juga harus memberikan rekomendasi berdasarkan data yang dimiliki dari masing-masing film. Dengan membuat suatu model rekomendasi yang menganalisis data dari masing-masing *content*, maka model ini juga dapat dikembangkan ke berbagai macam kebutuhan seperti *e-commerce*, rekomendasi buku, rekomendasi kendaraan dan lain sebagainya dengan catatan data yang diolah harus menjelaskan *content* dari setiap *item*.

Model rekomendasi ini dibangun dengan menggunakan algoritma *Content-Based Filtering*. Menurut Bhatt (2014), algoritma *Content-Based Filtering system* menganalisis dokumen

atau preferensi yang diberikan oleh pengguna tertentu, dan mencoba membangun model berdasarkan data ini. Sistem ini memanfaatkan minat khusus pengguna dan berusaha mencocokkan profil pengguna dengan atribut yang dimiliki oleh berbagai objek konten yang akan direkomendasikan [4]. Contohnya, jika pengguna menyukai film bergenre komedi, maka algoritma ini akan merekomendasikan film lain dengan genre yang sama.

Penelitian ini dikembangkan berdasarkan referensi dari penelitian yang dilakukan sebelumnya dengan metode yang sama yaitu *Content-Based Filtering* (CBF) dengan judul “Sistem Rekomendasi Artikel Jurnal Machine Learning Menggunakan *Content Based Filtering*” [5] dan penelitian dengan judul “*Job Recommendation System Using Content and Collaborative Filtering Based Techniques*” [2]. Berdasarkan penelitian yang dilakukan Rianti et al., bahwa sistem rekomendasi yang dilakukan menghitung kemiripan dari dataset jurnal yang dimiliki, dengan hasil rata-rata presisi 98,33%. Untuk penelitian yang dilakukan oleh Dhemeliya & Desai membahas tentang sistem rekomendasi yang diterapkan

untuk rekomendasi pekerjaan dalam proses *e-recruitment* dengan hasil evaluasi *recall* sebesar 63.97%.

Oleh karena itu, pada penelitian ini dikembangkan suatu sistem rekomendasi film menggunakan *Content-Based Filtering*. Metode

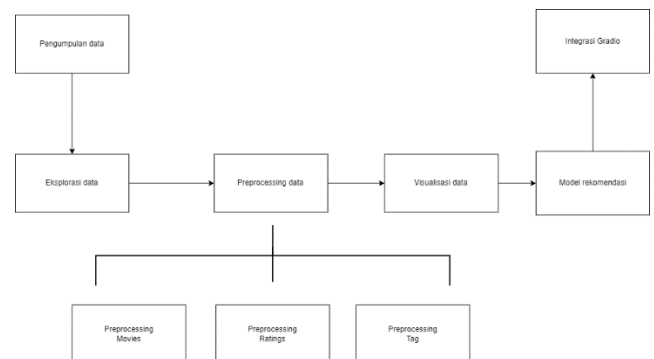
METODE PENELITIAN

Analisis Masalah

Pada tahap ini, dilakukan sebuah analisis tentang bagaimana suatu masalah dapat diselesaikan dengan menggunakan sistem content-based filtering diinginkan, dengan begitu tujuan dari penulisan ini akan disesuaikan oleh metode yang digunakan.

Untuk membuat suatu model yang menyelesaikan masalah yang sudah dijelaskan sebelumnya, beberapa tahapan dalam pembuatan model akan dilakukan mulai dari menyiapkan data sampai model berhasil diintegrasikan ke dalam framework Gradio. Semua tahapan itu bisa dilihat pada Gambar 1.

Content-Based Filtering menggunakan algoritma TF-IDF dan *cosine similarity* untuk menghitung similaritas antar film. Sistem rekomendasi yang dikembangkan selanjutnya diintegrasikan menggunakan *framework Gradio*.



Gambar 1. Alur Pemodelan Sistem Rekomendasi

Seperti terlihat pada gambar, langkah pertama adalah mengumpulkan data yang sesuai dengan kebutuhan pemodelan. Pada penulisan ini, digunakan dataset Movielens 25M. Setelah data terkumpul, perlu dilakukan eksplorasi untuk mengetahui jumlah data, parameter, dan kolom yang ada.

Langkah berikutnya adalah preprocessing data yang berbeda-beda tergantung kebutuhan tiap dataset. Tahap ini penting untuk memastikan model yang dibuat memberikan hasil yang tepat. Setelah preprocessing, data perlu divisualisasikan agar lebih mudah

dibaca. Visualisasi yang baik penting untuk menghindari kesalahan interpretasi data.

Terakhir, model rekomendasi dibuat menggunakan metode *content-based* dengan menghitung kemiripan menggunakan *cosine similarity*. Model ini kemudian diintegrasikan ke dalam framework Gradio.

Pengumpulan Data

Data berperan penting dalam pembuatan model latih yang bisa diibaratkan bahwa data merupakan bahan bakar untuk membuat model yang baik, dengan data memiliki suatu data yang berkualitas maka informasi yang dipaparkan akan berkualitas pula.

Sumber data yang digunakan untuk membuat model rekomendasi berasal dari dataset yang disediakan oleh *website* Grouplens dengan nama *dataset MovieLens 25M Data*, yang terdiri dari *dataset genome-score.csv*, *genome-tags.csv*, *links.csv*, *movies.csv*, *ratings.csv*, dan *tags.csv*. Namun, pada penulisan ilmiah ini, hanya *movies.csv*, *ratings.csv*, dan *tags.csv* yang digunakan dan akan disatukan menjadi satu *dataset*

isi dari *dataset* tersebut bisa dilihat pada Tabel 1.

Tabel 1. Struktur Dataset

Dataset	Variable	Description
Movies	movieId	Identitas dari setiap film
	title	Judul dari setiap film
	genres	Genre yang ada dari setiap film
Ratings	userId	Identitas pengguna yang melakukan <i>review</i> film
	movieId	Identitas dari tiap film
	rating	Nilai dari setiap film menurut pengguna
	timestamp	Lamanya waktu tayang dari setiap film
Tag	userId	Identitas pengguna yang memberikan 'tag' terhadap tiap film
	movieId	Identitas dari tiap film
	Tag	Kesan yang diberikan pengguna terhadap film
	Timestamp	Lamanya waktu tayang dari setiap film

Data Preparation

Tahap ini bertujuan untuk mempersiapkan data melalui manipulasi seperti preprocessing data. Tahap ini sangat penting untuk menghasilkan data berkualitas yang akan digunakan dalam pembuatan model. Tahapan preprocessing data sangat penting dalam pembuatan model karena pada tahapan ini semua data akan diolah dengan berbagai cara untuk menghasilkan dataset yang sesuai untuk pemodelan. Proses ini memastikan bahwa data tersebut layak, baik, dan sesuai dengan

kebutuhan pemodelan yang akan dilakukan, serta menentukan apakah model yang dibuat akan menghasilkan output yang sesuai atau tidak untuk tahapan langkah *preprocessing* dapat dilihat pada Gambar 2. Pada Gambar 2 berbagai tahapan dilakukan seperti untuk menghapus kolom *timestamp*, menghitung rerata rating, melakukan normalisasi data pada *dataset ratings* dan lain sebagainya semua tahapan itu perlu dilakukan untuk menghasilkan data yang sesuai untuk membuat model yang baik dalam memberikan rekomendasi.



Gambar 2. Tahapan *Pre-processing*

Preprocessing Dataset Ratings

Pada tahapan ini beberapa langkah *preprocessing* yang harus dilakukan seperti melakukan *drop* pada kolom *timestamp* karena pada kolom tersebut tidak digunakan. *Preprocessing* selanjutnya adalah untuk menghitung rerata rating pada *dataset* tersebut, rating digunakan untuk perhitungan dari masing-masing film yang tersedia. Pada *dataset* awal setiap film memiliki

beberapa rating maka dari itu rating harus dijadikan satu untuk menghasilkan setiap satu film hanya memiliki satu rating saja dan yang terakhir karena perhitungan dari hasil rerata rating memiliki format yang cukup sulit untuk dibaca maka diperlukan normalisasi data dengan melakukan pemformatan hanya 2 angka dibelakan koma untuk mempermudah dalam pembacaan data.

Preprocessing Dataset Tag

Beberapa langkah yang harus dilakukan pada tahap *preprocessing* ini mencakup penghapusan kolom *userid* dan *timestamp*, yang dianggap tidak relevan untuk analisis lebih lanjut, serta penggabungan beberapa kolom *tag* menjadi satu kolom yang lebih komprehensif. Setelah itu, dilakukan normalisasi data pada kolom *tag* dengan tujuan untuk membersihkan data dari simbol-simbol yang tidak perlu, seperti tanda baca atau karakter khusus, yang dapat mengganggu analisis. Selain itu, semua teks dalam kolom *tag* diubah menjadi huruf kecil untuk memastikan konsistensi dan mempermudah proses analisis lebih lanjut.

Preprocessing Dataset Movies

Pada dataset movies, beberapa langkah preprocessing yang akan dilakukan meliputi ekstraksi tahun dari kolom judul film untuk memisahkan tahun rilis dari judul utama, serta penghapusan informasi tahun dari judul film itu sendiri agar lebih bersih dan sesuai dengan format analisis yang diinginkan.

Selain itu, akan diterapkan berbagai teknik pembersihan data pada kolom genre untuk memastikan kategori film terdefinisi dengan baik dan tidak ada data yang tidak relevan atau redundan. Langkah terakhir dalam preprocessing ini adalah melakukan penyaringan data, yaitu memilih film-film yang dirilis antara tahun 1995 hingga 2019, sehingga dataset yang dihasilkan lebih fokus dan relevan untuk analisis yang akan dilakukan.

Pembuatan model

Pada tahapan pembuatan model disini akan mengolah data yang dimiliki dengan metode *content-based filtering*, model akan mengukur seberapa seringnya suatu kalimat muncul menggunakan TF-IDF dan akan dihitung

kemiripan dari masing-masing array film menggunakan *cosine similarity* dan yang terakhir model akan dievaluasi. Content-Based Filtering (CBF) memberikan rekomendasi berdasarkan preferensi pengguna dengan kategori yang diberikan [3].

Dengan menggunakan sistem rekomendasi berbasis konten, sistem ini dapat menyediakan kebebasan kepada pengguna untuk menilai item secara eksklusif sesuai dengan preferensi dari masing-masing pengguna. Penilaian ini nantinya akan digunakan untuk membangun profil pengguna yang menjadi dasar untuk rekomendasi yang lebih personal.

Term Frequency – Inverse Document Frequency merupakan metode statistik yang digunakan dalam pencarian informasi dan pemrosesan bahasa alami untuk mengukur pentingnya suatu kata terhadap dokumen. *Term frequency* menyatakan jumlah kata/term/token yang muncul pada sebuah dokumen [1]. Untuk menghitung *Term frequency* menggunakan persamaan (1) berikut:

$$TF(t,d)=\frac{\text{Jumlah kemunculan term } t \text{ dalam dokumen } d}{\text{Jumlah total term dalam dokumen } d} \quad (1)$$

Untuk perhitungan IDF sendiri menggunakan persamaan (2) dan untuk

perhitungan TF-IDF menggunakan persamaan (3) berikut:

$$IDF(t,D)=\log\left(\frac{N}{1+\text{Jumlah dokumen yang mengandung } t}\right) \quad (2)$$

$$TF-IDF(t,d,D) = TF(t,d) \times IDF(t,D) \quad (3)$$

Cosine similarity merupakan salah satu metode yang sering diterapkan untuk proses temu kembali informasi dan klastering dimana akan diukur derajat kesamaan jarak antara dua dokumen [1]. Nilai cosine similarity berkisar antara 0 dan 1, dengan 0 menunjukkan ketidaksamaan dan 1 menunjukkan kesamaan. Secara matematik, rumus dari cosine similarity antara dua vector A dan B terdapat pada persamaan (4) berikut:

$$\text{Cosine Similarity}(A,B) = \frac{A.B}{||A|| \times ||B||} \quad (4)$$

Di mana:

- A . B adalah hasil dari *dot product* dari vektor A dan B.
- $||A||$ adalah magnitudo (norma) dari vector A, yang dihitung sebagai $\sqrt{a_1^2 + a_2^2 + \dots + a_n^2}$
- $||B||$ adalah magnitudo (norma) dari vektor B, yang dihitung sebagai $\sqrt{b_1^2 + b_2^2 + \dots + b_n^2}$

Evaluasi Model

Untuk melihat bagaimana model memberikan rekomendasi yang aktual maka model harus di lakukan evaluasi. Evaluasi akan dihitung berdasarkan perbandingan antar dua film berdasarkan perhitungan cosine similarity. Hasil dari perhitungan akan berbentuk persentase kemiripan dari perbandingan tersebut dan hasil tersebut akan dicocokkan dengan rekomendasi yang diberikan.

Maka dari data tersebut harus divalidasi apakah rekomendasi yang diberikan benar-benar memiliki kemiripan paling tinggi. Maka model harus dievaluasi dengan cara membandingkan sample dari list kemiripan yang ada pada Gambar 3.

```
# Test the recommendation function with user input
recommendation_input = input('Masukkan judul film yang kamu inginkan: ')
recommendations_system = get_recommendations(recommendation_input)
print(recommendations_system)
```

Masukkan judul film yang kamu inginkan: cars 2
Toy Story 2 (Rating: 3.81)
Bug's Life, A (Rating: 3.57)
Cars (Rating: 3.33)
Monsters, Inc. (Rating: 3.85)
Toy Story (Rating: 3.89)
Monsters University (Rating: 3.5)
Finding Dory (Rating: 3.62)
Finding Nemo (Rating: 3.83)
Ratatouille (Rating: 3.81)
Toy Story 3 (Rating: 3.86)

Gambar 3. Rekomendasi Film

Pada Gambar 4 berikut diketahui bahwa hasil perhitungan kemiripan dengan menggunakan cosine similarity antara film Cars 2 dengan film Toy Story 2 sebesar 59.20%.


```

sim_per = sim_score * 100

return f'Film '{title1}' dan '{title2}' memiliki tingkat kemiripan sebesar (sim_per:.2f)%'

# Test dengan input manual
title1 = input('Masukkan judul film pertama: ')
title2 = input('Masukkan judul film kedua: ')
similarity_result = hitung_kemiripan(title1, title2)
print(similarity_result)

Masukkan judul film pertama: cars 2
Masukkan judul film kedua: toy story 2
Film 'cars 2' dan 'toy story 2' memiliki tingkat kemiripan sebesar 59.28%

```

Gambar 4. Evaluasi 1

Pada Gambar 5 berikut akan membandingkan antara film Cars 2 dengan film Bug's Life, A memiliki kemiripan sebesar 58.87% dan dari 10 rekomendasi paling bawah dicoba untuk membandingkan film Cars 2 dengan film Toy Story 3 memiliki kemiripan sebesar 46.93% pada Gambar 6. Maka dari data tersebut bisa diketahui bahwa model rekomendasi mampu untuk menghitung kemiripan antar film dengan akurat yang dampaknya dapat mengurangi kesalahan dalam pemberian rekomendasi.

```

# Test dengan input manual
title1 = input('Masukkan judul film pertama: ')
title2 = input('Masukkan judul film kedua: ')
similarity_result = hitung_kemiripan(title1, title2)
print(similarity_result)

Masukkan judul film pertama: cars 2
Masukkan judul film kedua: Bug's Life, A
Film 'cars 2' dan 'Bug's Life, A' memiliki tingkat kemiripan sebesar 58.87%

```

Gambar 5. Evaluasi 2

```

# Test dengan input manual
title1 = input('Masukkan judul film pertama: ')
title2 = input('Masukkan judul film kedua: ')
similarity_result = hitung_kemiripan(title1, title2)
print(similarity_result)

Masukkan judul film pertama: cars 2
Masukkan judul film kedua: Toy Story 3
Film 'cars 2' dan 'Toy Story 3' memiliki tingkat kemiripan sebesar 46.93%

```

Gambar 6. Evaluasi 3

HASIL DAN PEMBAHASAN

Dataset

Dataset yang digunakan terdiri dari 25 Juta *dataset ratings* dan terdiri dari 1 juta tag yang diterapkan pada 62 ribu film yang sudah di berikan rating oleh 162 ribu orang. Setelah dilakukan seluruh rangkaian *preprocessing* data yang tersedia adalah sebanyak 23.681 baris data yang dikelompokkan dengan 6 kolom, banyaknya dari masing-masing data dapat dilihat pada Tabel 2.

Tabel 2. Banyak Dataset

Dataset	Quantity	Column
Movies	62.423	3
Ratings	25.000.095	4
Tag	1.093.360	4
Merge	23.681	6

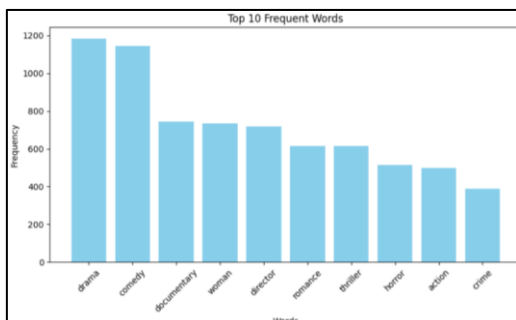
Visualisasi Data

Data perlu untuk dilakukan visualisasi yang tujuannya untuk mempermudah dalam pembacaan data. Beberapa teknik yang bisa digunakan salah satunya ada *wordcloud* untuk melihat kata-kata yang frekuensinya sering muncul maka kata tersebut akan semakin besar gambarnya, atau teknik dasar lainnya yaitu menggunakan

diagram batang seperti pada Gambar 8 yang merupakan diagram yang menunjukkan 10 kalimat yang paling sering muncul.



Gambar 7. Wordcloud



Gambar 8. Frekuensi Banyak Kalimat

Integrasi Model

Setelah model selesai dikembangkan, langkah selanjutnya adalah melakukan integrasi model ke dalam berbagai aplikasi. Pada tahap ini, integrasi dilakukan menggunakan framework Gradio.

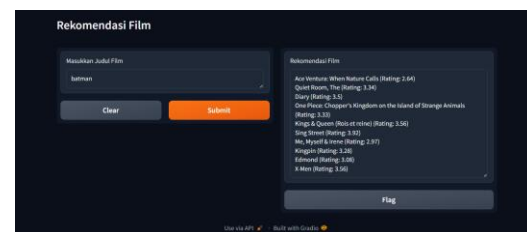
Selain untuk menguji model, penggunaan framework Gradio

bertujuan memudahkan pengguna dalam berinteraksi dengan sistem yang telah dibangun, berkat kemampuannya menyediakan antarmuka yang intuitif dan interaktif. Untuk integrasinya dapat dilihat pada Gambar 9 dan hasil dari integrasinya ada pada Gambar 10 berikut:

```
#interface gradio
iface = gr.Interface(
    fn=recommendation,
    inputs=gr.Textbox(label="Masukkan Judul Film"),
    outputs=gr.Textbox(label="Rekomendasi Film"),
    title = 'Rekomendasi Film'
)

# Jalankan antarmuka
iface.launch()
```

Gambar 9. Integrasi Model ke Gradio



Gambar 10. Hasil Integrasi Model

PENUTUP

Dalam penelitian ini, telah berhasil dikembangkan sebuah sistem rekomendasi film berbasis konten yang mampu mempertimbangkan karakteristik unik dari setiap film untuk memberikan rekomendasi yang lebih tepat sasaran.

Model ini telah diuji dengan baik menggunakan framework Gradio, yang memungkinkan visualisasi hasil

rekomendasi secara interaktif dan menunjukkan efektivitas model dalam berbagai situasi. Dengan keberhasilan pengembangan ini, model rekomendasi yang dihasilkan memiliki potensi besar untuk diintegrasikan ke dalam berbagai platform, seperti situs web atau layanan streaming film, sehingga dapat menyediakan rekomendasi yang lebih personal dan relevan bagi pengguna.

Penggunaan model ini diharapkan tidak hanya meningkatkan pengalaman pengguna dalam menemukan film sesuai preferensi mereka, tetapi juga menjadi landasan bagi pengembangan sistem rekomendasi lebih lanjut di masa depan. Adapun saran untuk pengembangan penelitian di masa mendatang antara lain:

1. Mengimplementasikan model ini pada berbagai platform digital, termasuk situs web dan aplikasi mobile, untuk meningkatkan kualitas pengalaman pengguna melalui rekomendasi yang lebih relevan dan personal. Integrasi ini diharapkan dapat mengurangi bias dalam rekomendasi serta memberikan hasil yang lebih sesuai dengan preferensi individual setiap pengguna.

2. Melakukan pengembangan lanjutan terhadap sistem rekomendasi dengan mengeksplorasi metode-metode alternatif, seperti collaborative filtering dan hybrid filtering, yang berpotensi menghasilkan rekomendasi lebih akurat. Selain itu, integrasi dengan teknologi AI, seperti chatbot, dapat memungkinkan personalisasi yang lebih mendalam melalui rekomendasi proaktif berdasarkan konteks percakapan.

DAFTAR PUSTAKA

- [1] Alimah, N. L., Putra, P. A., & Sigit, A. 2019. "Rekomendasi Film Berdasarkan Sinopsis Menggunakan Metode Word2Vec". Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer Vol.3. Universitas Brawijaya. <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/5623> [Diakses pada 02 Mei 2024],
- [2] Juhi Dhameliya & Nikita Desai. 2019. "Job Recommendation System using Content and Collaborative Filtering based

- Techniques”. International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-9. DOI:10.35940/ijscce.C3266.099319 [Diakses pada 16 Juni 2024].
- [3] Pramarta, A., & Baizal, A. 2022. “Hybrid Recommender System Using Singular Value Decomposition and Support Vector Machine in Bali Tourism.” JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika), vol. 7, no. 2, pp. 408–418. <https://doi.org/10.29100/jipi.v7i2.2770> [Diakses pada 16 Juni 2024].
- [4] Reddy, S., Nalluri, S., Kuniseti, S., Ashok, S., & Venkatesh, B. 2019. “Content-Based Movie Recommendation System Using Genre Correlation.” In Smart Intelligent Computing and Applications, Smart Innovation, Systems and Technologies (pp. 419-427). Springer. https://doi.org/10.1007/978-981-13-1927-3_42 [Diakses pada 24 April 2024].
- [5] Rianti, A., Majid, N. W. A., & Fauzi, A. 2024. “MACHINE LEARNING JOURNAL ARTICLE RECOMMENDATION SYSTEM USING CONTENT BASED FILTERING.” JUTI: Jurnal Ilmiah Teknologi Informasi, Volume 22. <http://juti.if.its.ac.id/index.php/juti/article/view/1193/500> [Diakses pada 16 Juni 2024].