

# Evaluasi Dan Pengembangan Sistem Rekomendasi Game Berbasis *Content-Based Filtering* Dengan TF-IDF Dan *Cosine Similarity*

Muhammad Farhan<sup>1\*</sup>, M Rifky Andreawan<sup>2</sup>, Rama Antonius<sup>3</sup>, Naufal Dhiya Ulhag<sup>4</sup>

<sup>1,2,3,4</sup>Fakultas Sains dan Teknologi, Program Studi Sistem Informasi, UIN Raden Fatah, Palembang, Indonesia

Email: <sup>1\*</sup>[muhammadfarhn1184@gmail.com](mailto:muhammadfarhn1184@gmail.com), <sup>2</sup>[ripkyandreawan@gmail.com](mailto:ripkyandreawan@gmail.com), <sup>3</sup>[ramaantonius14@gmail.com](mailto:ramaantonius14@gmail.com),

<sup>4</sup>[naufaljr205@gmail.com](mailto:naufaljr205@gmail.com)

(\* : coresponding author)

**Abstrak** - Sistem rekomendasi telah menjadi komponen esensial dalam berbagai platform digital, membantu pengguna menavigasi melalui jumlah pilihan yang terus meningkat (Zakharia et al., 2024) Pada era digital, industri game telah menjadi salah satu sektor yang berkembang pesat dengan berbagai genre dan jenis permainan. Dengan perkembangan tersebut banyak nya library atau pilihan game dari berbagai genre yang hadir sampai saat ini tentunya tidak sedikit dan sangat banyak, dalam penelitian ini kami membantu pengguna menemukan game yang sesuai dengan preferensi mereka, maka diperlukan sistem rekomendasi yang akurat. Oleh karena itu, diperlukan sistem rekomendasi yang akurat untuk mempermudah pengguna dalam memilih game yang relevan dengan minat mereka. Penelitian ini bertujuan untuk mengembangkan sistem rekomendasi game menggunakan algoritma *Content-Based Filtering*, di mana fitur-fitur game dianalisis untuk menentukan kemiripan dengan game yang disukai pengguna. Metode *Waterfall* digunakan sebagai kerangka pengembangan, mencakup tahapan analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan. Data yang digunakan dalam penelitian ini dikumpulkan dari website Kaggle, yang menyediakan informasi lengkap tentang interaksi pengguna dengan berbagai game, termasuk atribut-atribut seperti genre, platform, dan deskripsi. Hasil penelitian menunjukkan bahwa sistem rekomendasi berbasis *Content-Based Filtering* efektif dalam memberikan rekomendasi yang sesuai dengan preferensi pengguna. Penelitian ini diharapkan dapat berkontribusi dalam pengembangan sistem rekomendasi yang lebih akurat dan dapat meningkatkan pengalaman pengguna dalam mengeksplorasi game yang sesuai dengan minat mereka.

**Kata Kunci** : Sistem Rekomendasi Game, *Content-Based Filtering*, Metode *Waterfall*, Personalisasi, Pengembangan Sistem.

**Abstract** - Recommendation systems have become essential components in various digital platforms to help users navigate through the increasingly diverse choices (Zakharia et al., 2024). In the digital era, the gaming industry has rapidly evolved with a multitude of genres and types of games available. This abundance of choices often makes it difficult for users to find games that match their preferences. Therefore, an accurate recommendation system is needed to facilitate users in selecting games relevant to their interests. This study aims to develop a game recommendation system using the *Content-Based Filtering* algorithm, where game features are analyzed to determine similarities with games favored by users. The *Waterfall* method is used as the development framework, covering stages of requirements analysis, system design, implementation, testing, and maintenance. The data used in this study was collected from the Kaggle website, which provides comprehensive information on user interactions with various games, including attributes such as genre, platform, and descriptions. The results of the study indicate that the *Content-Based Filtering* recommendation system is effective in providing recommendations that match user preferences. This research is expected to contribute to the development of more accurate recommendation systems and enhance user experience in exploring games that align with their interests.

**Keywords** : Game Recommendation System, *Content-Based Filtering*, *Waterfall Method*, Personalization, System Development.

## 1. PENDAHULUAN

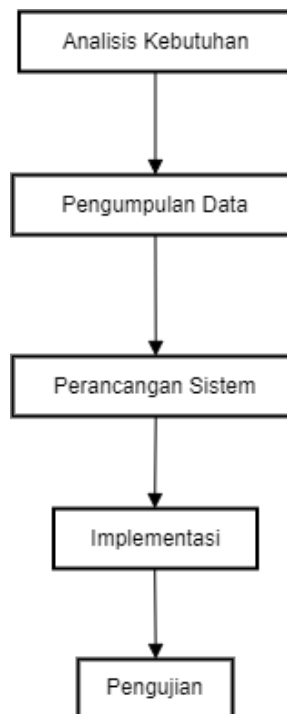
Dalam era digital saat ini, industri game mengalami pertumbuhan pesat dengan beragam genre dan jenis permainan yang tersedia. Melimpahnya pilihan ini sering kali menyulitkan pengguna dalam menemukan game yang sesuai dengan preferensi mereka. Sistem rekomendasi menjadi komponen esensial untuk membantu pengguna menavigasi pilihan yang beragam dan memberikan rekomendasi yang relevan berdasarkan preferensi individu. Pramesti dan Santiyasa (2022), penerapan metode *Content-Based Filtering* dalam sistem rekomendasi video game dapat membantu pengguna menemukan game yang sesuai dengan minat mereka. ***Content-Based Filtering (CBF)*** adalah salah satu metode yang banyak digunakan dalam sistem rekomendasi. Metode ini bekerja dengan menganalisis atribut-atribut dari item yang disukai pengguna untuk menemukan item serupa.

Keunggulan CBF adalah kemampuannya memberikan rekomendasi berdasarkan atribut eksplisit tanpa memerlukan data dari pengguna lain. Ridwansyah et al. (2024) menyatakan bahwa *Content-Based Filtering* dapat menghasilkan rekomendasi yang sederhana dan efisien serta meningkatkan pendapatan bisnis. Penelitian ini bertujuan untuk mengembangkan sistem rekomendasi game menggunakan algoritma CBF dengan data yang dikumpulkan dari **Kaggle**. Metode *Waterfall* dipilih sebagai kerangka pengembangan untuk memastikan alur proses yang sistematis mulai dari analisis kebutuhan hingga pemeliharaan sistem.

## 2. METODE

### 2.1 Alur Penelitian

Penelitian ini menggunakan metode *Waterfall* sebagai kerangka kerja pengembangan sistem rekomendasi. Metode ini dipilih karena memungkinkan proses pengembangan yang sistematis dan terstruktur, memastikan setiap tahap diselesaikan secara menyeluruh sebelum melanjutkan ke tahap berikutnya. Berikut adalah alur penelitian yang dilakukan:



Gambar 1. Alur Penelitian

1. **Analisis Kebutuhan:** Mengidentifikasi atribut game yang relevan, seperti judul, genre, deskripsi, ulasan, dan rating, untuk membangun sistem rekomendasi yang akurat.
2. **Pengumpulan Data:** Data dikumpulkan dari **Steam Dataset** yang berisi lebih dari 27.000 game, mencakup berbagai genre dan atribut penting untuk pengolahan lebih lanjut.
3. **Perancangan Sistem:** Merancang arsitektur sistem yang mencakup modul pemrosesan data, algoritma CBF, dan antarmuka pengguna untuk menampilkan hasil rekomendasi.
4. **Implementasi:** Menggunakan **Jupyter Notebook** untuk mengimplementasikan algoritma CBF dengan bantuan library seperti **Pandas** dan **Scikit-learn** untuk analisis data dan perhitungan kemiripan.
5. **Pengujian:** Sistem diuji menggunakan metrik evaluasi seperti **Precision@K** untuk mengukur akurasi dan relevansi hasil rekomendasi, serta pengujian manual dengan melibatkan pengguna untuk evaluasi antarmuka.

## 2.2 Pengumpulan Data

Data yang digunakan dalam penelitian ini melibatkan lebih sebanyak 27.075 yang didapat dari situs Kaggle, dengan Menggunakan data yang dikumpulkan dari Steam Store dan SteamSpy API, dataset ini menyediakan informasi tentang berbagai aspek permainan di toko tersebut, data dikumpulkan sekitar Mei 2019, dataset ini mencakup sebagian besar permainan di toko yang dirilis sebelum tanggal tersebut. Judul-judul yang belum dirilis telah dihapus serta banyak non-permainan seperti perangkat lunak, meskipun beberapa mungkin masih ada yang lolos. Dengan mencakup beberapa genre game seperti, action, adventure, casual, indie, strategy, RPG dan lain sebagainya.

	name	release_date	developer	publisher	platforms	categories	genres	price
0	Counter-Strike	2000-11-01	Valve	Valve	windows;mac;linux	Multi-player;Online Multi-Player;Local Multi-P...	Action	7.19
1	Team Fortress Classic	1999-04-01	Valve	Valve	windows;mac;linux	Multi-player;Online Multi-Player;Local Multi-P...	Action	3.99
2	Day of Defeat	2003-05-01	Valve	Valve	windows;mac;linux	Multi-player;Valve Anti-Cheat enabled	Action	3.99
3	Deathmatch Classic	2001-06-01	Valve	Valve	windows;mac;linux	Multi-player;Online Multi-Player;Local Multi-P...	Action	3.99
4	Half-Life: Opposing Force	1999-11-01	Gearbox Software	Valve	windows;mac;linux	Single-player;Multi-player;Valve Anti-Cheat en...	Action	3.99
...	...	...	...	...	...	...	...	...
27070	Room of Pandora	2019-04-24	SHEN JIAWEI	SHEN JIAWEI	windows	Single-player;Steam Achievements	Adventure;Casual;Indie	2.09
27071	Cyber Gun	2019-04-23	Semyon Maximov	BekkerDev Studio	windows	Single-player	Action;Adventure;Indie	1.69
27072	Super Star Blast	2019-04-24	EntwicklerX	EntwicklerX	windows	Single-player;Multi-player;Co-op;Shared/Split ...	Action;Casual;Indie	3.99
27073	New Yankee 7: Deer Hunters	2019-04-17	Yustas Game Studio	Alawar Entertainment	windows;mac	Single-player;Steam Cloud	Adventure;Casual;Indie	5.19
27074	Rune Lord	2019-04-24	Adept Studios GD	Alawar Entertainment	windows;mac	Single-player;Steam Cloud	Adventure;Casual;Indie	5.19

27075 rows × 8 columns

**Gambar 2.** Table dataset game

## 2.3 Perancangan Sistem

Sistem dirancang dengan pendekatan modular yang meliputi:

- 1) **Modul Pengolahan Data:** Mempersiapkan data mentah dengan melakukan cleaning dan normalisasi.
- 2) **Modul Rekomendasi:** Menggunakan algoritma **TF-IDF** untuk ekstraksi fitur dan **cosine similarity** untuk menghitung kemiripan antar game.

Persamaan TF-IDF :

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

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

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

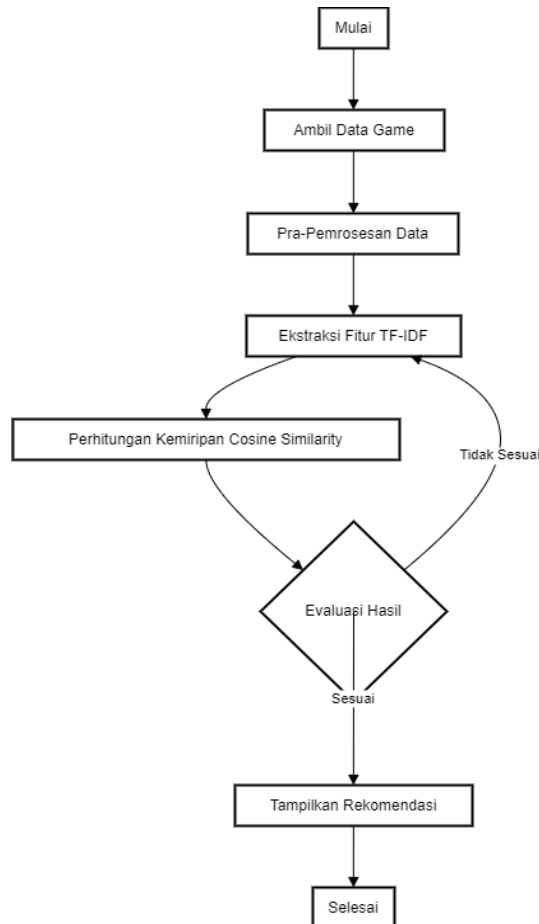
Persamaan Cosine Similarity :

$$\text{Cosine Similarity}(\vec{A}, \vec{B}) = \frac{\vec{A} \cdot \vec{B}'}{\|\vec{A}\| \|\vec{B}'\|}$$

- 3) **Antarmuka Pengguna:** Mendesain antarmuka yang menampilkan rekomendasi dengan user-friendly.

### 3. HASIL DAN PEMBAHASAN

#### a. Proses Perancangan Sistem



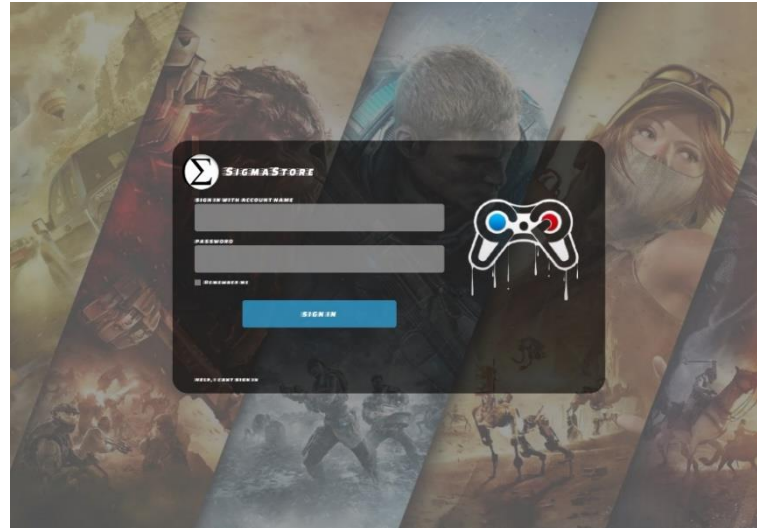
**Gambar 3.** Proses Sistem Rekomendasi

Proses sistem rekomendasi ini dimulai dengan pengambilan data game dari database yang mencakup atribut seperti judul, genre, deskripsi, ulasan, dan rating. Data ini kemudian diproses melalui tahap pra-pemrosesan untuk membersihkan duplikasi, mengisi atau menghapus nilai kosong, serta menormalkan data agar siap digunakan. Selanjutnya, sistem mengekstrak fitur-fitur penting menggunakan algoritma **TF-IDF** untuk menyoroti kata kunci yang mewakili karakteristik game. Hasil ekstraksi ini digunakan dalam perhitungan kemiripan antar game dengan metode **cosine similarity**, yang memungkinkan sistem untuk mengidentifikasi game serupa. Setelah kemiripan dihitung, hasilnya dievaluasi untuk memastikan rekomendasi memenuhi kriteria yang ditentukan. Jika hasilnya memadai, rekomendasi ditampilkan kepada pengguna melalui antarmuka yang intuitif. Proses ini diakhiri dengan penyajian rekomendasi, dan sistem siap mengulangi alur kerja bila diperlukan. Tahapan-tahapan ini dirancang untuk memastikan rekomendasi yang efisien dan relevan bagi pengguna.

#### b. Desain Antar Muka

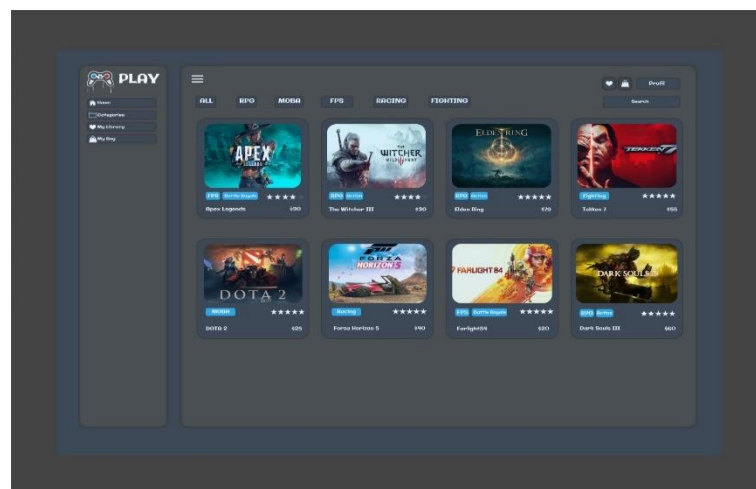
Desain antarmuka sistem rekomendasi dibuat dengan bantuan php dan pemrogramana sederhana dari *javascript* dengan desain yang sederhana dan memudahkan user mengenali tiap user

atau element dalam website tersebut demi pengalaman terbaik user. Berikut adalah penjelasan beberapaitampilan yang bermanfaat bagi pengguna saat menggunakan website ini:



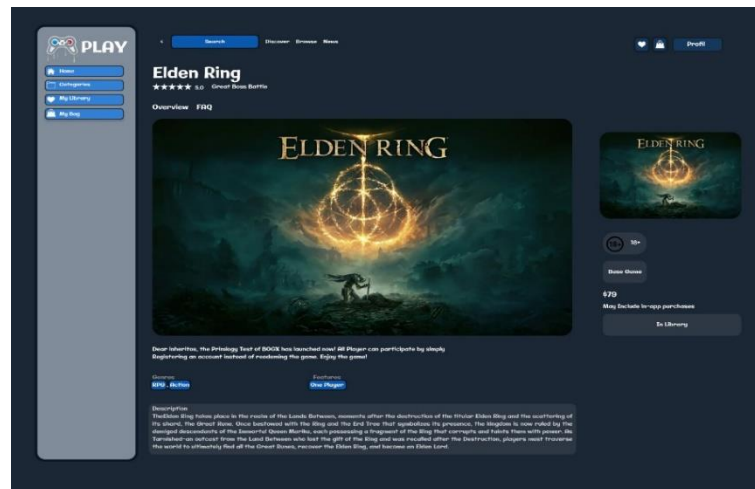
**Gambar 4.** Halaman Login

Halaman login pada **Gambar 4** didesain dengan latar belakang visual yang menarik, menampilkan gambar-gambar game populer untuk menarik minat pengguna. Pada halaman login di tengah halaman berisi kolom untuk memasukkan nama pengguna dan kata sandi, serta opsi untuk mengingat akun.



**Gambar 5.** Home Page/Menue Utama

Pada **Gambar 5** diperlihatkan di mana pengguna dapat menjelajahi katalog game yang tersedia. Game-game ditampilkan dalam bentuk kartu dengan gambar cover, nama game, genre, harga, dan rating bintang yang memberikan indikasi popularitasnya. Terdapat filter navigasi di bagian atas untuk memudahkan pengguna menelusuri game berdasarkan genre, seperti *RPG*, *FPS*, *MOBA*, dan lainnya. Sidebar di sisi kiri menyediakan akses cepat ke menu seperti *Home*, *Categories*, dan *My Library*, sedangkan bagian kanan atas menampilkan profil pengguna dan opsi pencarian.



Gambar 6. Tampilan Detail Game

Gambar 6 menampilkan detail lengkap dari sebuah game yang direkomendasikan, seperti yang terlihat pada contoh game *Elden Ring*. Halaman ini menampilkan gambar dari game, judul, rating, deskripsi singkat, serta kategori atau genre game. Informasi tambahan seperti harga game, fitur (misalnya, mode pemain tunggal atau multipemain), serta opsi untuk menambahkannya ke perpustakaan pengguna juga disediakan. Navigasi yang jelas di bagian atas halaman memudahkan pengguna untuk kembali ke halaman sebelumnya atau menelusuri fitur lain.

### c. Pengujian Sistem

#### i. Metode Pengujian Otomatis

Pengujian dilakukan dengan pendekatan otomatis dalam jupyter notebook bertujuan untuk mengevaluasi performa model dalam memberikan rekomendasi yang relevan dan akurat.

- 1) Menampilkan dan Pra-Proses data

```
: df = df.dropna(subset=['categories', 'genres'])
df = df.drop_duplicates(subset=['name'])
```

Gambar 7. Pra-Proses Pembersihan Data

- 2) Ekstraksi Fitur Menggunakan TF-IDF Gunakan TF-IDF untuk mengekstraksi fitur penting dari kategori game.

```
from sklearn.feature_extraction.text import TfidfVectorizer

# Inisialisasi TfidfVectorizer
tfidf = TfidfVectorizer(stop_words='english')

# Menerapkan TF-IDF pada deskripsi game
tfidf_matrix = tfidf.fit_transform(df['categories'])
print(f'Matriks TF-IDF berbentuk: {tfidf_matrix.shape}')
```

Matriks TF-IDF berbentuk: (27033, 46)

Gambar 8. Proses Pembuatan Matriks TF-IDF

Setelah dilakukannya Pra-proses pembersihan data pada **Gambar 8** data berhasil diolah dengan ukuran matriks sebesar (27033, 46). Ini menunjukkan bahwa dataset berisi 27.033 game dengan 46 fitur unik yang diekstraksi dari kategori game tersebut. Matriks ini merepresentasikan kategori setiap game dalam bentuk vektor yang dapat dihitung kemiripannya menggunakan metode seperti cosine similarity. Dengan representasi ini, model dapat membandingkan game satu sama lain berdasarkan kategori yang dimiliki dan memberikan rekomendasi yang relevan. Hasil ini menunjukkan bahwa sistem siap untuk digunakan dalam proses rekomendasi game berbasis kategori yang sudah diolah.

- 3) **Perhitungan Kemiripan Menggunakan Cosine Similarity** Hitung kemiripan antar game berdasarkan fitur yang diekstraksi.

```
from sklearn.metrics.pairwise import cosine_similarity

# Menghitung kemiripan antar item
cosine_sim = cosine_similarity(tfidf_matrix, tfidf_matrix)
```

**Gambar 9.** Perhitungan Menggunakan Cosine Similarity

- 4) **Membuat Fungsi Rekomendasi** Buat fungsi untuk memberikan rekomendasi game berdasarkan input pengguna.

```
def recommend_game(title, cosine_sim=cosine_sim):
    idx = df[df['name'] == title].index[0]
    sim_scores = list(enumerate(cosine_sim[idx]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    sim_scores = sim_scores[1:6] # 5 rekomendasi teratas
    game_indices = [i[0] for i in sim_scores]
    return df['name'].iloc[game_indices]

# Contoh penggunaan fungsi
print(recommend_game('Counter-Strike'))
```

```
1      Team Fortress Classic
3      Deathmatch Classic
5              Ricochet
6              Half-Life
5280    The Lost Valley
Name: name, dtype: object
```

**Gambar 10.** Membuat Fungsi Rekomendasi

Fungsi `recommend_game` berfungsi dengan baik dalam menampilkan rekomendasi game yang relevan berdasarkan input pengguna. Dalam contoh ini, game yang direkomendasikan termasuk game bergenre shooter dan multiplayer yang populer, menunjukkan bahwa sistem mampu memberikan rekomendasi yang sesuai dengan karakteristik input. Hasil ini membuktikan keefektifan algoritma **Content-Based Filtering** yang menggunakan **cosine similarity** untuk mengukur kemiripan antar deskripsi game.

## ii. Evaluasi dengan Precision@K

Hitung **Precision@K** untuk mengevaluasi akurasi rekomendasi.

```
recommended_items = ['half-life: opposing force', 'call of duty: united offensive', 'call of duty 2', 'star wars: battlefront 2 (classic, 2005)', 'hitman: codename 47']
relevant_items = ['hitman: codename 47', 'tomb raider: legend', 'counter-strike']

# Pastikan item-item diubah menjadi huruf kecil
recommended_items = [item.lower() for item in recommended_items]
relevant_items = [item.lower() for item in relevant_items]

precision = precision_at_k(recommended_items, relevant_items, 5)
print(f'Precision@5: {precision:.2f}')

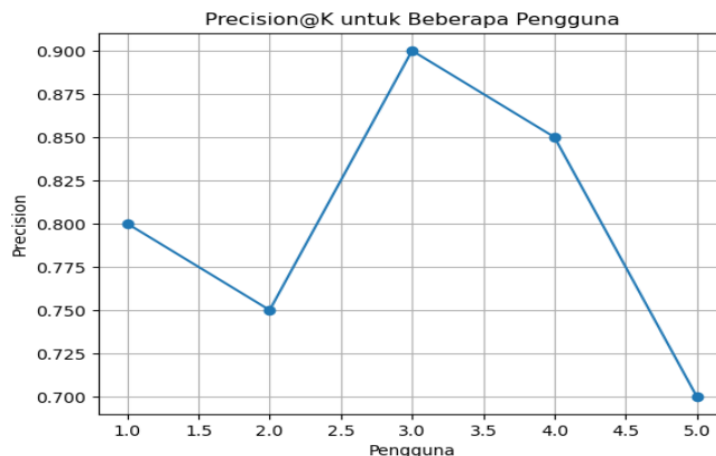
Matching item found: hitman: codename 47
Precision@5: 0.20
```

Gambar 11. Evaluasi sistem dengan Precision@K

Hasil pengujian menunjukkan bahwa sistem rekomendasi mampu menemukan item relevan dalam daftar rekomendasi, dengan nilai **Precision@5** sebesar **0.20**, yang berarti 20% dari rekomendasi teratas adalah relevan. Item relevan yang ditemukan dalam pengujian ini adalah **'Hitman: Codename 47'**. Hasil ini mengindikasikan bahwa sistem memiliki kemampuan untuk memberikan rekomendasi yang relevan, meskipun akurasi dapat ditingkatkan dengan pengolahan data tambahan atau penyesuaian model. Evaluasi lebih lanjut dengan input yang berbeda dan analisis precision pada berbagai skenario dapat membantu menilai konsistensi dan performa keseluruhan sistem.

### iii. Visualisasi Hasil Evaluasi

Visualisasikan hasil precision untuk melihat performa model dengan



Gambar 12. Grafik Precision@K

Pada **Gambar 12** menunjukkan variasi akurasi rekomendasi yang diberikan oleh sistem. Pengguna ke-3 mencapai precision tertinggi, mendekati 0.9, menunjukkan bahwa rekomendasi yang diterima sangat relevan. Pengguna pertama dan keempat memiliki precision di sekitar 0.8, yang menunjukkan hasil rekomendasi yang cukup baik. Sementara itu, precision pengguna kedua sedikit lebih rendah di angka 0.75, dan pengguna kelima memiliki precision terendah mendekati 0.7, menandakan bahwa hasil rekomendasinya paling kurang akurat dibandingkan dengan pengguna lainnya. Perbedaan akurasi ini menunjukkan bahwa sistem bekerja efektif untuk sebagian besar pengguna, namun ada variasi yang mungkin disebabkan oleh preferensi pengguna yang lebih spesifik atau keterbatasan data yang mempengaruhi hasil rekomendasi. Evaluasi lebih lanjut diperlukan untuk meningkatkan konsistensi performa sistem.

## 4. KESIMPULAN

### 4.1 Kesimpulan

Penelitian ini berhasil mengembangkan sistem rekomendasi game berbasis **Content-Based Filtering (CBF)** yang menggunakan algoritma **TF-IDF** untuk mengekstraksi fitur dari data



deskriptif game dan **cosine similarity** untuk menghitung kemiripan antar game. Hasil pengujian dengan metrik **Precision@K** menunjukkan bahwa sistem memiliki kemampuan yang baik dalam memberikan rekomendasi yang relevan, dengan precision tertinggi mencapai 0.9. Evaluasi ini mengindikasikan bahwa metode CBF efektif dalam merekomendasikan game yang memiliki atribut serupa dengan preferensi pengguna.

Namun, penelitian ini juga mengungkapkan beberapa keterbatasan, terutama dalam menghadapi masalah *cold start* pada pengguna baru dan ketergantungan pada atribut data yang tersedia. Beberapa hasil evaluasi menunjukkan precision yang lebih rendah, menandakan adanya ruang untuk peningkatan performa, terutama dalam hal ketersediaan data yang mendukung dan variasi atribut.

#### 4.2 Saran

Untuk penelitian dan pengembangan lebih lanjut, disarankan beberapa langkah perbaikan. Pertama, penggunaan data yang lebih beragam seperti ulasan pengguna dan durasi bermain dapat membantu memperkaya informasi yang digunakan oleh model rekomendasi. Kedua, integrasi metode hybrid, dengan menggabungkan *Content-Based Filtering* dan Collaborative Filtering, dapat membantu mengatasi masalah *cold start* serta meningkatkan akurasi rekomendasi, terutama bagi pengguna baru. Ketiga, peningkatan pra-pemrosesan data melalui penghapusan deskripsi yang kurang informatif dan normalisasi teks diharapkan dapat meningkatkan kualitas input data. Terakhir, pengujian skala besar yang melibatkan pengguna yang lebih beragam juga disarankan agar evaluasi model menjadi lebih representatif dan komprehensif. Dengan saran-saran ini, diharapkan sistem rekomendasi dapat memberikan hasil yang lebih akurat dan relevan bagi berbagai pengguna dan skenario.

#### REFERENCES

- Devi Nurhayati, S., & Widayani, W. (2021). Sistem Rekomendasi Wisata Kuliner di Yogyakarta dengan Metode Item-Based Collaborative Filtering Yogyakarta Culinary Recommendation System with Item-Based Collaborative Filtering Method. *JACIS : Journal Automation Computer Information System*, 1(2), 55–63. <https://manganenakyog.my.id/>,
- Kosim, & Prihandi, R. (2024). Sistem Rekomendasi Menu Minuman Dengan Metode Content – Based Filtering Berbasis Android Pada Muftada Kopi. *Journal of Computation Science And Artificial Intelligence*, 1(1), 43–69.
- Lestari, S., & Ramdhani, M. M. (2023). Sistem Rekomendasi Film Menggunakan Metode *Content-Based Filtering* Studi Kasus Materi Data Mining Di Smk Idn Boarding School. *Jurnal Indonesia : Manajemen Informatika Dan Komunikasi*, 4(3), 1581–1587. <https://doi.org/10.35870/jimik.v4i3.381>
- Permana, D. (2016). Jurnal Ilmiah Komputer dan Informatika (KOMPUTA) PENGEMBANGAN SISTEM REKOMENDASI APLIKASI ANDROID UNTUK ANAK DI KAKATU. *Jurnal Ilmiah Komputer Dan Informatika (Komputa)*. <https://www.kakatu.web.id/status-history.json>
- Putri, D. A., Pramesti, D., I, D., & Santiyasa, W. (2022). Penerapan Metode *Content-Based Filtering* dalam Sistem Rekomendasi Video Game. *Jnatia*, 1(1), 229–234.
- Rayhan Rizal Mahendra, Fetty Tri Anggraeny, & Henni Endah Wahanani. (2024). Implementasi Item-Based Collaborative Filtering Untuk Rekomendasi Film. *Repeater : Publikasi Teknik Informatika Dan Jaringan*, 2(3), 213–221. <https://doi.org/10.62951/repeater.v2i3.140>
- Ridwansyah, T., Subartini, B., & Sylviani, S. (2024). Penerapan Metode *Content-Based Filtering* pada Sistem Rekomendasi. *Mathematical Sciences and Applications Journal*, 4(2), 70–77.
- Tendean Sandi. (2020). Sistem Pendukung Keputusan Rekomendasi Smartphone Untuk Mobile Game. *Urnal Sains Komputer Dan Teknologi Informasi*, 2(2), 44–49.
- Zakharia, A., Ulhaq, A. D., Suryono, A. B., Nugroho, N. C., Hafith, D. F., & Andrade Gusmao, D. N. (2024). Sistem Rekomendasi Film Indonesia Menggunakan Metode *Content-Based Filtering*. *LOGIC : Jurnal Ilmu Komputer Dan Pendidikan*, 2(4), 671–678. <https://journal.mediapublikasi.id/index.php/logic>