Institut Riset dan Publikasi Indonesia (IRPI)
MALCOM: Indonesian Journal of Machine Learning and Computer Science
Journal Homepage: https://journal.irpi.or.id/index.php/malcom
Vol. 6 Iss. 3 July 2026, pp: 1642-1654
ISSN(P): 2797-2313 | ISSN(E): 2775-8575

Cold-Start Synopsis-Based Movie Recommendation: Comparing TF-IDF, Word2Vec, and FastText

Evaluasi Sistem Rekomendasi Film Berbasis Sinopsis Cold-Start: Perbandingan TF-IDF, Word2Vec, dan FastText

Khadijah Fahmi Hayati Holle[1*], Ela Ilmatul Hidayah[2], Renata Amalia Putri[3], Citra Ayu Putri Ningrum[4]

[1,2,3,4]Program Studi Teknik Informatika, Fakultas Sains dan Teknologi,
Universitas Islam Negeri Maulana Malik Ibrahim Malang, Indonesia

E-Mail: [1]khadijah.holle@uin-malang.ac.id, [2]elailma217@gmail.com,
[3]renataamaliap04@gmail.com, [4]citra18ayu@gmail.com

Received Jun 04th 2026; Revised Jul 8th 2026; Accepted Jul 20th 2026; Available Online Jul 28th 2026
Corresponding Author: Khadijah Fahmi Hayati Holle

Abstract

This study aims to determine which lightweight text representation is most effective and computationally efficient for cold-start movie recommendation when only synopsis content is available, and no user-interaction history exists. The study compares TF-IDF, Word2Vec, and FastText representations combined with cosine similarity on a TMDb movie dataset. From 10,001 records, 9,924 valid movies were used after removing empty synopses, duplicates, and non-informative tokens. Experiments were conducted across 20 cold-start query scenarios, comprising 10 long narrative queries and 10 short keyword queries. Because explicit user relevance labels were unavailable, relevance was measured using genre-proxy relevance. Evaluation used Precision@10, HitRate@10, AP@10, MRR@10, nDCG@10, scenario F1, and average retrieval time. The results show that TF-IDF + cosine similarity achieved the best overall performance with Precision@10 of 0.835, AP@10 of 0.884, MRR@10 of 0.942, nDCG@10 of 0.942, and scenario F1 of 0.902. The main contribution of this study is a controlled empirical comparison showing that lexical representation can outperform averaged word embeddings in synopsis-based cold-start recommendation. TF-IDF was more effective because short movie synopses and explicit thematic queries preserve discriminative keywords, whereas averaged Word2Vec and FastText vectors tend to smooth keyword-specific signals. These results indicate that TF-IDF remains a strong, interpretable, and efficient baseline for resource-constrained movie recommendation systems.

Keywords: Cold-Start, Cosine Similarity, Fasttext, Recommendation System, TF-IDF, Word2Vec

Abstrak

Penelitian ini bertujuan untuk menentukan representasi teks ringan yang paling efektif dan efisien untuk rekomendasi film cold-start ketika sistem hanya memiliki sinopsis dan belum memiliki riwayat interaksi pengguna. Penelitian membandingkan TF-IDF, Word2Vec, dan FastText yang dikombinasikan dengan cosine similarity pada dataset film IMDb. Dari 10.001 data film, sebanyak 9.924 data valid digunakan setelah penghapusan sinopsis kosong, duplikasi, dan token tidak informatif. Pengujian dilakukan menggunakan 20 skenario kueri cold-start yang terdiri atas sepuluh kueri panjang bernarasi dan sepuluh kueri pendek berbasis kata kunci. Karena dataset tidak menyediakan label relevansi eksplisit dari pengguna, relevansi diukur menggunakan genre-proxy relevance. Evaluasi menggunakan Precision@10, HitRate@10, AP@10, MRR@10, nDCG@10, F1 skenario, serta waktu retrieval rata-rata. Hasil menunjukkan bahwa TF-IDF + cosine similarity memperoleh kinerja terbaik dengan Precision@10 sebesar 0,835, AP@10 sebesar 0,884, MRR@10 sebesar 0,942, nDCG@10 sebesar 0,942, dan F1 skenario sebesar 0,902. Kontribusi utama penelitian ini adalah pembuktian empiris terkontrol bahwa representasi leksikal dapat mengungguli rata-rata word embedding dalam rekomendasi cold-start berbasis sinopsis. TF-IDF lebih unggul karena sinopsis film yang pendek dan kueri tematik yang eksplisit masih mempertahankan kata kunci diskriminatif, sedangkan rata-rata vektor Word2Vec dan FastText cenderung meratakan sinyal kata kunci yang spesifik. Hasil ini menunjukkan bahwa TF-IDF tetap menjadi baseline yang kuat, mudah diinterpretasikan, dan efisien untuk sistem rekomendasi film dengan sumber daya komputasi yang terbatas.

Kata Kunci: Cold-Start, Cosine Similarity, Fasttext, Sistem Rekomendasi, TF-IDF, Word2Vec

1. PENDAHULUAN

Pertumbuhan katalog film pada layanan hiburan digital membuat proses pencarian film semakin kompleks. Pengguna tidak hanya menghadapi jumlah pilihan yang besar, tetapi juga harus menafsirkan genre, popularitas, durasi, bahasa, dan deskripsi cerita yang belum tentu langsung sesuai dengan preferensi naratifnya. Permasalahan menjadi lebih jelas pada kondisi cold-start, yaitu ketika sistem belum memiliki riwayat klik, rating, atau tontonan pengguna baru. Dalam kondisi tersebut, sistem rekomendasi tidak dapat mengandalkan pola interaksi historis, sehingga perlu memanfaatkan informasi intrinsik item yang sudah tersedia sejak awal, seperti judul, genre, dan sinopsis.

Sistem rekomendasi dapat dikembangkan melalui beberapa pendekatan, antara lain collaborative filtering, content-based filtering, dan hybrid filtering. Collaborative filtering memanfaatkan pola interaksi pengguna, seperti rating, klik, atau riwayat tontonan. Pendekatan ini efektif ketika data interaksi pengguna tersedia dalam jumlah besar, sebagaimana terlihat pada penerapan sistem rekomendasi berskala besar pada platform video digital [1], [2], [3]. Namun, pendekatan ini memiliki kelemahan mendasar pada kondisi cold-start karena pengguna baru atau item baru belum memiliki riwayat interaksi yang cukup. Sebaliknya, content-based filtering dapat menghasilkan rekomendasi berdasarkan karakteristik item, seperti judul, genre, sinopsis, aktor, sutradara, kata kunci tematik, atau metadata lain yang melekat pada item [4], [5].

Dalam konteks rekomendasi film, sinopsis merupakan atribut penting karena memuat informasi naratif mengenai tema, konflik, tokoh, suasana, dan alur cerita. Genre memang dapat memberikan gambaran umum mengenai kategori film, tetapi genre sering kali terlalu luas. Dua film dengan genre yang sama belum tentu memiliki kedekatan naratif, sedangkan dua film dengan genre berbeda dapat memiliki kemiripan tema cerita. Oleh karena itu, pemanfaatan sinopsis sebagai dasar sistem rekomendasi menjadi penting untuk menangkap kedekatan isi film secara lebih kontekstual.

Pendekatan klasik dalam representasi teks banyak menggunakan Bag-of-Words dan Term Frequency-Inverse Document Frequency (TF-IDF). TF-IDF masih relevan karena sederhana, efisien, mudah diinterpretasikan, dan tidak memerlukan korpus pelatihan besar [6], [7]. Beberapa penelitian menunjukkan bahwa TF-IDF dan cosine similarity dapat menghasilkan rekomendasi berbasis konten secara cukup baik pada domain film maupun konten digital [4], [5], [8], [9]. Namun, representasi leksikal memiliki keterbatasan dalam menangkap relasi semantik antarkata apabila kata yang maknanya berkaitan tidak muncul secara eksplisit dalam dokumen yang sama.

Keterbatasan tersebut mendorong penggunaan representasi berbasis embedding. Word2Vec merepresentasikan kata dalam bentuk vektor berdimensi tetap berdasarkan konteks kemunculannya dalam korpus [10], sedangkan FastText memperluas Skip-Gram dengan informasi subword sehingga dapat menangani kata langka dan variasi bentuk kata secara lebih baik [11]. Dalam domain rekomendasi, embedding juga telah digunakan untuk merepresentasikan item dan menganalisis kedekatan semantik antarfilm [12], [13].

Perkembangan sistem rekomendasi modern menunjukkan peningkatan penggunaan deep learning dan transformer, seperti Neural Collaborative Filtering, session-based recommendation, SASRec, BERT4Rec, BERT, Sentence-BERT, dan berbagai model neural recommendation lain [14]-[20], [3]. Meskipun demikian, model-model tersebut membutuhkan sumber daya komputasi dan proses pelatihan yang lebih besar dibandingkan TF-IDF, Word2Vec, dan FastText. Oleh sebab itu, evaluasi model ringan tetap relevan, terutama untuk sistem rekomendasi yang membutuhkan retrieval cepat, mudah direplikasi, dan dapat diterapkan pada lingkungan dengan sumber daya terbatas.

Berdasarkan kajian penelitian terdahulu, masih terdapat beberapa celah yang perlu diperjelas. Pertama, sejumlah penelitian rekomendasi film berbasis konten masih berfokus pada penerapan satu metode, misalnya TF-IDF dan cosine similarity, sehingga belum memberikan perbandingan terkontrol dengan embedding berbasis konteks dan subword [6], [8], [9]. Kedua, penelitian berbasis embedding sering menekankan kemampuan semantik, tetapi belum selalu menjelaskan kondisi data yang membuat embedding tidak otomatis lebih baik daripada representasi leksikal. Ketiga, evaluasi cold-start berbasis kueri sinopsis masih perlu dikaitkan dengan karakteristik sinopsis film yang pendek, bentuk kueri pengguna, serta efisiensi retrieval. Dengan demikian, research gap penelitian ini terletak pada kebutuhan membandingkan representasi leksikal dan embedding ringan secara adil pada dataset, kueri, definisi relevansi, dan metrik Top-K yang sama.

Evaluasi sistem rekomendasi perlu dilakukan secara hati-hati karena hasil offline dapat dipengaruhi oleh desain skenario, pemilihan kandidat item, definisi relevansi, dan metrik yang digunakan [21], [22]. Selain akurasi, sistem rekomendasi juga perlu memperhatikan potensi bias kategori, bias popularitas, coverage, diversity, novelty, dan serendipity [23]. Dalam penelitian ini, relevansi didekati menggunakan genre-proxy karena dataset tidak menyediakan label relevansi eksplisit dari pengguna. Oleh karena itu, hasil penelitian diinterpretasikan sebagai evaluasi offline berbasis proxy, bukan sebagai pengukuran langsung kepuasan pengguna.

Berdasarkan permasalahan tersebut, penelitian ini mengevaluasi sistem rekomendasi film berbasis sinopsis pada skenario cold-start dengan membandingkan TF-IDF sebagai baseline leksikal, Word2Vec sebagai representasi semantik berbasis konteks, dan FastText sebagai representasi semantik berbasis subword. Kontribusi penelitian ini dirumuskan dalam empat aspek. Pertama, penelitian ini menyusun pipeline

rekomendasi film berbasis sinopsis yang dapat digunakan tanpa data interaksi pengguna. Kedua, penelitian ini membandingkan TF-IDF, Word2Vec, dan FastText secara empiris menggunakan dataset TMDb yang sama, skenario kueri yang sama, dan metrik evaluasi Top-K yang sama. Ketiga, penelitian ini menjelaskan alasan empiris mengapa TF-IDF dapat mengungguli embedding pada sinopsis pendek, yaitu karena kata tematik eksplisit masih menjadi sinyal pembeda yang kuat. Keempat, penelitian ini memberikan implikasi praktis bahwa model ringan dan interpretabel masih layak digunakan untuk prototipe sistem rekomendasi film, terutama ketika sumber daya komputasi terbatas dan data interaksi pengguna belum tersedia.

2. METODOLOGI

Sistem rekomendasi dikembangkan dengan memanfaatkan sinopsis film sebagai sumber utama representasi konten. Tahapan penelitian meliputi pengumpulan dataset, seleksi atribut, pembersihan data, eksplorasi data, prapemrosesan teks, pembentukan representasi teks menggunakan TF-IDF, Word2Vec, dan FastText, perhitungan kemiripan menggunakan cosine similarity, pemeringkatan rekomendasi Top-10, serta evaluasi hasil menggunakan metrik Top-K.

2.1. Dataset

Dataset yang digunakan berasal dari The Movie Database (TMDb) dan memuat 10.001 data film dengan 15 atribut. Atribut yang tersedia meliputi id film, judul, genre, tanggal rilis, bahasa asli, vote average, vote count, popularity, overview, budget, production companies, revenue, runtime, dan tagline. Penelitian ini berfokus pada tiga atribut utama, yaitu title, genres, dan overview. Atribut title digunakan untuk menampilkan hasil rekomendasi, genres digunakan sebagai dasar evaluasi genre-proxy relevance, sedangkan overview digunakan sebagai sinopsis yang menjadi sumber utama representasi teks.

Dataset diperiksa terlebih dahulu untuk memastikan bahwa sinopsis tersedia dan dapat diproses. Data dengan sinopsis kosong, data duplikat berdasarkan kombinasi id, title, dan overview, serta data yang tidak memiliki token informatif setelah prapemrosesan dihapus. Setelah proses pembersihan, sebanyak 9.924 film digunakan dalam eksperimen. Dengan demikian, 99,23% data masih dapat digunakan sebagai korpus pelatihan dan kandidat rekomendasi. Ringkasan jumlah data awal, data valid, data terhapus, serta statistik panjang sinopsis ditampilkan pada Tabel 1.

Tabel 1. Ringkasan dataset penelitian

Komponen	Nilai
Jumlah data awal	10.001 film
Jumlah atribut	15 atribut
Data valid setelah pembersihan	9.924 film
Data terhapus	77 film
Persentase data usable	99,23%
Rata-rata panjang sinopsis	25,43 token
Median panjang sinopsis	23 token
Panjang maksimum sinopsis	112 token

2.2. Prapemrosesan Teks

Prapemrosesan teks dilakukan untuk memastikan bahwa data sinopsis memiliki format yang konsisten sebelum digunakan dalam pembentukan representasi teks. Tahapan prapemrosesan meliputi konversi huruf kecil, tokenisasi alfabetis, penghapusan stopword bahasa Inggris, serta penyaringan token dengan panjang kurang dari atau sama dengan 2 karakter. Tokenisasi alfabetis digunakan untuk mengambil token berbasis huruf dan menghilangkan angka, simbol, serta tanda baca. Stopword removal digunakan untuk menghapus kata-kata umum yang kurang berkontribusi terhadap makna, seperti “the”, “and”, “of”, “in”, dan “to”.

Proses pemrosesan yang sama diterapkan pada sinopsis film dan kueri uji. Hal ini dilakukan agar representasi kueri dan representasi sinopsis berada dalam ruang teks yang konsisten. Hasil prapemrosesan kemudian digunakan sebagai masukan untuk tiga pendekatan representasi teks, yaitu TF-IDF, Word2Vec, dan FastText.

2.3. Representasi Teks Menggunakan TF-IDF

TF-IDF digunakan sebagai baseline karena metode ini masih banyak digunakan dalam sistem rekomendasi berbasis teks. TF-IDF menghitung bobot suatu kata berdasarkan frekuensi kemunculannya dalam dokumen dan tingkat kelangkaannya dalam seluruh korpus. Konfigurasi TF-IDF yang digunakan adalah unigram, stopword bahasa Inggris sesuai daftar prapemrosesan, min_df = 1, max_df = 1,0, tanpa pembatasan jumlah fitur, dan normalisasi L2. Min_df = 1 dipilih karena sinopsis film relatif pendek sehingga kata tematik yang jarang muncul, seperti zombie, dragon, outlaw, atau haunted, masih perlu dipertahankan sebagai sinyal pembeda. Normalisasi L2 digunakan agar skor cosine similarity dapat dihitung secara konsisten pada vektor dokumen dan vektor kueri. Bobot TF-IDF dinyatakan pada Persamaan (1), sedangkan nilai inverse document frequency dinyatakan pada Persamaan (2).

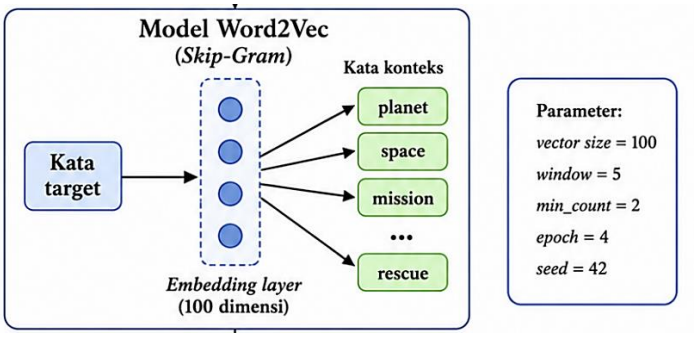
$$tf - idf_{t,d} = tf_{t,d} \times idf_t \quad (1)$$

$$idf_t = \log\left(\frac{N}{df_t}\right) \quad (2)$$

Pada Persamaan (1) dan Persamaan (2), $tf - idf_{t,d}$ menyatakan bobot TF-IDF term t pada dokumen d, $tf_{t,d}$ menyatakan frekuensi term, idf_t menyatakan inverse document frequency, N menyatakan jumlah dokumen, dan df_t menyatakan jumlah dokumen yang memuat term t. Dalam penelitian ini, kosakata TF-IDF yang terbentuk berjumlah 27.393 token.

2.4. Representasi Teks Menggunakan Word2Vec

Word2Vec digunakan untuk merepresentasikan kata dalam ruang vektor berdimensi tetap berdasarkan konteks kemunculannya dalam korpus. Penelitian ini menggunakan arsitektur Skip-Gram karena arsitektur tersebut sesuai untuk mempelajari representasi kata yang relatif jarang muncul pada korpus berukuran sedang. Konfigurasi Word2Vec yang digunakan adalah vector size 100, window size 5, min_count 2, epoch 4, dan seed 42. Vector size 100 dipilih untuk menjaga keseimbangan antara kapasitas representasi dan efisiensi komputasi. Window size 5 digunakan untuk menangkap konteks lokal dalam kalimat sinopsis, sedangkan min_count 2 digunakan untuk mengurangi token yang terlalu jarang dan berpotensi menambah noise. Epoch 4 digunakan karena eksperimen ini berfokus pada perbandingan terkendali antarrepresentasi teks, bukan optimasi hiperparameter yang agresif. Kosakata Word2Vec yang terbentuk berjumlah 15.969 token. Alur konseptual Word2Vec yang digunakan ditunjukkan pada Gambar 1.



Gambar 1. Arsitektur model Word2Vec

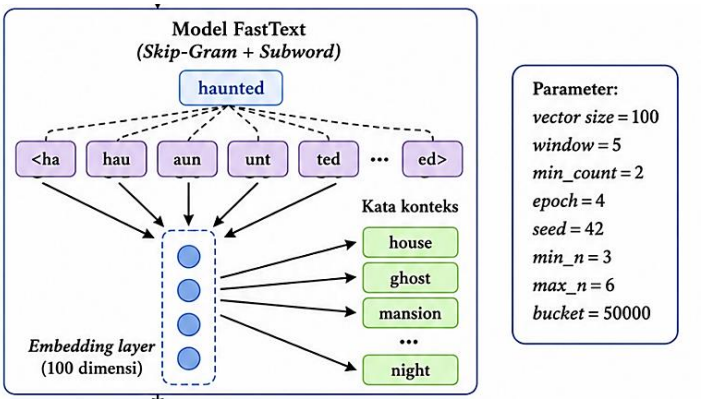
Setelah model Word2Vec dilatih, setiap sinopsis direpresentasikan sebagai satu vektor dokumen. Vektor dokumen diperoleh dengan menghitung rata-rata seluruh vektor kata yang tersedia dalam sinopsis. Representasi vektor sinopsis dinyatakan pada Persamaan (3). Pada Persamaan (3), $\vec{d}_i$ adalah vektor sinopsis ke-i, $\vec{w}_j$ adalah vektor kata ke-j, dan n adalah jumlah token dalam sinopsis yang dikenali oleh model Word2Vec atau FastText.

$$\vec{d}_i = \frac{1}{n}\sum_{j=1}^{n} \vec{w}_j \quad (3)$$

2.5. Representasi Teks Menggunakan FastText

FastText digunakan sebagai model pembanding lanjutan karena mampu memanfaatkan informasi subword. Berbeda dari Word2Vec yang merepresentasikan setiap kata sebagai unit utuh, FastText merepresentasikan kata sebagai gabungan n-gram karakter. Dengan demikian, FastText dapat menghasilkan representasi yang lebih baik untuk kata langka, variasi bentuk kata, dan kata dengan struktur morfologis yang serupa [13].

Konfigurasi FastText yang digunakan adalah arsitektur Skip-Gram dengan subword n-gram, vector size 100, window size 5, min_count 2, epoch 4, seed 42, min_n 3, max_n 6, dan bucket 50.000. Nilai min_n 3 dan max_n 6 dipilih agar model dapat mempelajari potongan karakter yang umum digunakan dalam representasi subword, sedangkan bucket 50.000 digunakan untuk menyediakan ruang hashing subword yang memadai tanpa membuat model terlalu berat. Kosakata eksplisit FastText yang terbentuk sama dengan Word2Vec, yaitu 15.969 token. Setelah model dilatih, setiap sinopsis direpresentasikan sebagai rata-rata vektor kata, sama seperti pendekatan Word2Vec. Struktur konseptual FastText ditunjukkan pada Gambar 2.



Gambar 2. Arsitektur model FastText

2.6. Perhitungan Cosine Similarity

Setelah sinopsis dan kueri pengguna direpresentasikan sebagai vektor, sistem menghitung kemiripan antara vektor kueri dan seluruh vektor sinopsis film menggunakan cosine similarity. Pada Persamaan (4), A adalah vektor kueri pengguna, B adalah vektor sinopsis film dalam dataset, A·B adalah perkalian titik antara kedua vektor, sedangkan ||A|| dan ||B|| adalah norma masing-masing vektor. Dalam sistem rekomendasi, skor kemiripan diurutkan dari nilai tertinggi ke nilai terendah untuk menghasilkan daftar rekomendasi Top-10.

$$s(A,B) = \frac{A \cdot B}{\| A \| \| B \|} \tag{4}$$

2.7. Skenario Evaluasi

Evaluasi dilakukan pada skenario cold-start. Dalam skenario ini, sistem tidak menggunakan data riwayat pengguna, rating pribadi, riwayat klik, atau riwayat menonton. Sistem hanya menerima kueri berupa kata kunci, frasa, atau deskripsi naratif dari pengguna, kemudian mencari film yang sinopsisnya paling mirip dengan kueri tersebut. Penelitian ini menggunakan 20 skenario kueri yang terdiri dari 10 kueri panjang dan 10 kueri pendek. Kueri panjang digunakan untuk menguji kemampuan model dalam memahami konteks naratif yang lebih kaya, sedangkan kueri pendek digunakan untuk menguji kemampuan model dalam menangani input yang ringkas dan berpotensi ambigu. Rincian kueri, tema, tipe kueri, dan genre ekspektasi ditampilkan pada Tabel 2.

Karena dataset tidak memiliki label relevansi eksplisit dari pengguna, penelitian ini menggunakan genre-proxy relevance. Sebuah film dinilai relevan apabila daftar genre film tersebut memiliki irisan dengan genre yang diharapkan oleh kueri. Pendekatan ini bukan pengganti sempurna untuk penilaian pengguna, tetapi dapat digunakan sebagai evaluasi awal yang sistematis pada dataset tanpa label relevansi. Penggunaan evaluasi offline pada sistem rekomendasi perlu dilakukan dengan hati-hati karena hasilnya dapat dipengaruhi oleh desain skenario, target item, dan definisi relevansi [21], [22].

Tabel 2. Kueri evaluasi

Kode	Tipe	Tema	Query	Genre Ekspektasi
L01	Panjang	Romance / first sight love	A woman falls in love at first sight with a stranger she meets at a train station. A brief glance and a small smile are enough to stir her heart, leaving behind a curiosity she cannot forget.	Drama, Romance
L02	Panjang	Magical world / mystical creatures	A young hero enters a magical world filled with fairies, elves, enchanted forests, ancient spells, and mystical creatures while searching for a lost kingdom.	Adventure, Animation, Family, Fantasy
L03	Panjang	Supportive family	A loving and supportive family stands by each other through every joy, loss, and challenge as they learn the meaning of home, forgiveness, and togetherness.	Comedy, Drama, Family, Romance
L04	Panjang	Space mission / distant planet	A team of astronauts travels across deep space to explore a distant planet, confront dangerous unknown forces, and complete a mission that may decide the future of humanity.	Action, Adventure, Science Fiction
L05	Panjang	Detective / unsolved murder	A brilliant detective investigates an unsolved murder in a city full of secrets, false identities, hidden evidence, and suspects who all have something to hide.	Crime, Mystery, Thriller

Kode	Tipe	Tema	Query	Genre Ekspektasi
L06	Panjang	Zombie outbreak / survival	A group of survivors struggles to stay alive after a terrifying zombie outbreak spreads through the city, forcing them to fight, escape, and distrust one another.	Action, Horror, Thriller
L07	Panjang	Musician / fame and love	A talented young musician rises to fame while struggling with ambition, heartbreak, friendship, and the difficult choice between personal love and a public career.	Drama, Music, Romance
L08	Panjang	Soldier / war trauma	A soldier returns home after a devastating war and tries to rebuild his life while haunted by painful memories, lost comrades, and the cost of survival.	Drama, History, War
L09	Panjang	Western / outlaw gang	A lonely cowboy protects a small frontier town from a ruthless outlaw gang, facing gunfights, betrayal, and a final showdown under the desert sun.	Action, Adventure, Western
L10	Panjang	Animated animal adventure	A brave young animal leaves home on a funny and emotional adventure with new friends, learning courage, loyalty, and the importance of family along the way.	Adventure, Animation, Comedy, Family
S01	Pendek	Cold-blooded killer	A cold-blooded killer	Action, Crime, Horror, Mystery, Thriller
S02	Pendek	Haunted house	A haunted house	Horror, Mystery, Thriller
S03	Pendek	Space mission	Space mission	Action, Adventure, Science Fiction
S04	Pendek	High school romance	High school romance	Comedy, Drama, Romance
S05	Pendek	Bank robbery	Bank robbery	Action, Crime, Thriller
S06	Pendek	Dragon quest	Dragon quest	Adventure, Animation, Family, Fantasy
S07	Pendek	Time travel	Time travel	Adventure, Fantasy, Science Fiction
S08	Pendek	Secret agent	Secret agent	Action, Adventure, Thriller
S09	Pendek	Lost child	Lost child	Drama, Family, Mystery
S10	Pendek	World war	World war	Action, Drama, History, War

2.8. Metrik Evaluasi

Metrik evaluasi yang digunakan adalah Precision@10, HitRate@10, AP@10, MRR@10, nDCG@10, F1 skenario, dan waktu retrieval rata-rata. Metrik-metrik ini dipilih karena sistem rekomendasi bertujuan menghasilkan daftar item teratas, bukan melakukan klasifikasi biner terhadap seluruh film. Evaluasi Top-K lebih sesuai untuk menilai apakah sistem mampu menempatkan item relevan pada posisi atas daftar rekomendasi.

Precision@K mengukur proporsi item relevan dalam Top-K rekomendasi yang dinyatakan pada Persamaan (5). HitRate@K menunjukkan apakah terdapat minimal satu item relevan dalam Top-K yang dinyatakan pada Persamaan (6). Average Precision@K menghitung rerata precision pada posisi ketika item relevan ditemukan, persamaannya dinyatakan pada Persamaan (7). Mean Reciprocal Rank@K mengukur posisi item relevan pertama dalam daftar rekomendasi, persamaannya dinyatakan pada Persamaan (8). Discounted Cumulative Gain@K digunakan untuk menghitung kualitas ranking berdasarkan posisi item relevan, persamaannya dinyatakan pada Persamaan (9). Normalized Discounted Cumulative Gain@K diperoleh dengan membagi DCG@K dengan nilai ideal DCG@K sebagaimana dinyatakan pada Persamaan (10). F1 skenario digunakan untuk melihat keseimbangan antara Precision@10 dan HitRate@10 yang dinyatakan pada Persamaan (11).

$$P_K = \frac{|R_K \cap G|}{K} \tag{5}$$

$$H_K = \begin{cases} 1, & |R_K \cap G| > 0 \\ 0, & |R_K \cap G| = 0 \end{cases} \tag{6}$$

$$A_K = \frac{1}{\sum_{i=1}^{K} r_i} \sum_{i=1}^{K} P_i \, r_i \tag{7}$$

$$M_K = \frac{1}{q_1} \tag{8}$$

$$D_K = \sum_{i=1}^{K} \frac{r_i}{\log_2(i+1)} \tag{9}$$

$$N_K = \frac{D_K}{I_K} \tag{10}$$

$$F_s = 2 \times \frac{P_{10} \times H_{10}}{P_{10} + H_{10}} \tag{11}$$

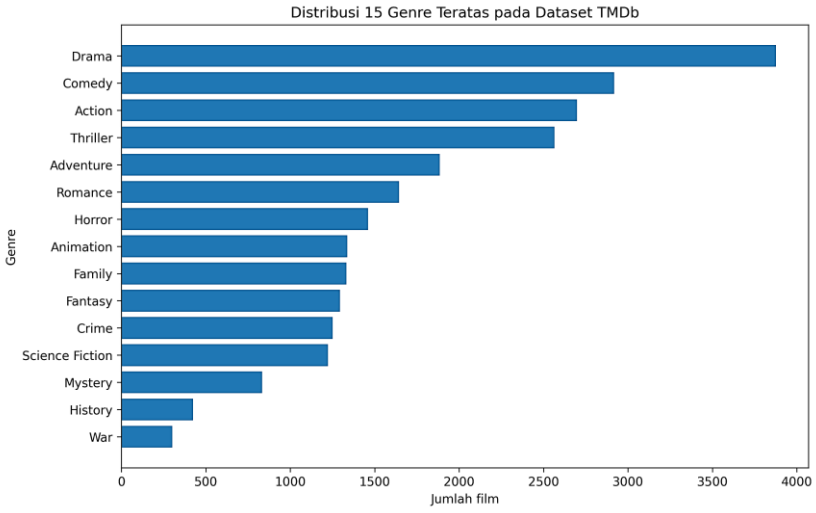
3. HASIL DAN DISKUSI

3.1. Eksplorasi Dataset

Hasil eksplorasi menunjukkan bahwa dataset TMDb memiliki keragaman genre yang cukup besar. Genre Drama menjadi genre paling dominan dengan 3.876 kemunculan, diikuti Comedy sebanyak 2.918, Action sebanyak 2.699, Thriller sebanyak 2.565, dan Adventure sebanyak 1.885. Genre Romance, Horror, Animation, Family, Fantasy, Crime, Science Fiction, dan Mystery juga muncul dalam jumlah cukup besar. Ringkasan lima belas genre teratas disajikan pada Tabel 3, sedangkan pola distribusinya divisualisasikan pada Gambar 3.

Tabel 3. Lima belas genre teratas pada dataset

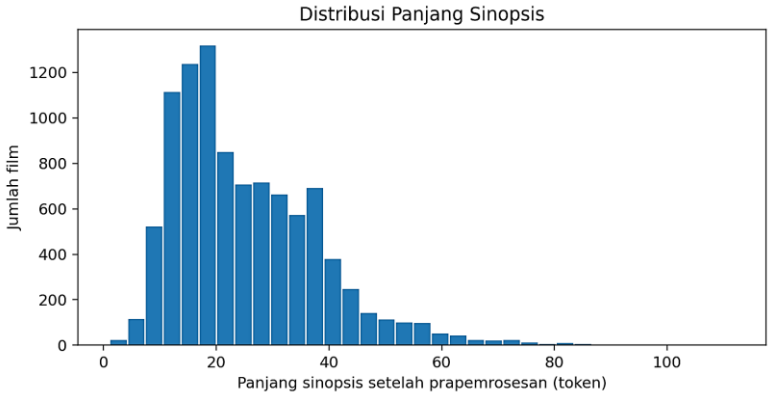
Peringkat	Genre	Jumlah
1	Drama	3.876
2	Comedy	2.918
3	Action	2.699
4	Thriller	2.565
5	Adventure	1.885
6	Romance	1.644
7	Horror	1.461
8	Animation	1.338
9	Family	1.333
10	Fantasy	1.295
11	Crime	1.251
12	Science Fiction	1.223
13	Mystery	834
14	History	424
15	War	302



Gambar 3. Distribusi 15 genre teratas pada dataset TMDb

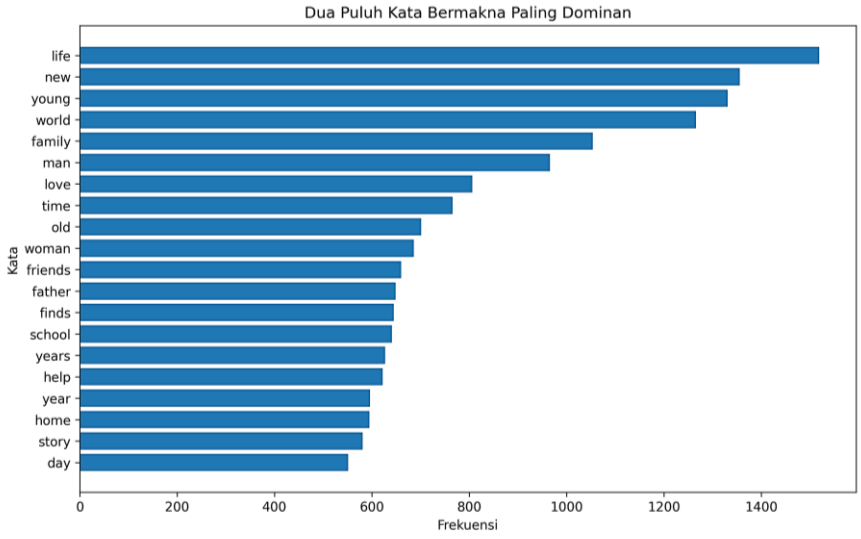
Distribusi ini menunjukkan bahwa dataset tidak berpusat pada satu kategori film saja, tetapi tetap memiliki ketidakseimbangan. Genre Drama, Comedy, Action, dan Thriller jauh lebih dominan dibandingkan genre History atau War. Ketidakseimbangan ini berpotensi memengaruhi hasil rekomendasi karena model berbasis konten dapat lebih sering menemukan kemiripan pada genre yang memiliki jumlah film lebih banyak. Oleh sebab itu, hasil rekomendasi perlu dianalisis tidak hanya dari skor kemiripan, tetapi juga dari relevansi dan kualitas ranking.

Panjang sinopsis setelah prapemrosesan menunjukkan rata-rata 25,43 token dan median 23 token. Kuartil pertama berada pada 15 token, sedangkan kuartil ketiga berada pada 33 token. Hal ini menunjukkan bahwa sebagian besar sinopsis cukup ringkas. Sinopsis yang ringkas dapat membuat representasi dokumen lebih sederhana, tetapi juga dapat membatasi konteks semantik yang dipelajari oleh model. Distribusi panjang sinopsis setelah prapemrosesan ditunjukkan pada Gambar 4.



Gambar 4. Distribusi panjang sinopsis setelah prapemrosesan

Analisis kata dominan menunjukkan bahwa kata “life”, “new”, “young”, “world”, “family”, “man”, “love”, “time”, “old”, dan “woman” merupakan kata yang sering muncul. Kata-kata tersebut merepresentasikan pola naratif umum dalam film, seperti kehidupan tokoh, dunia baru, konflik keluarga, relasi cinta, perjalanan waktu, dan karakter utama. Hal ini menunjukkan bahwa sinopsis film memuat sinyal semantik yang dapat digunakan untuk membangun sistem rekomendasi berbasis konten. Dua puluh kata bermakna paling dominan divisualisasikan pada Gambar 5.



Gambar 5. Dua puluh kata bermakna paling dominan pada sinopsis film

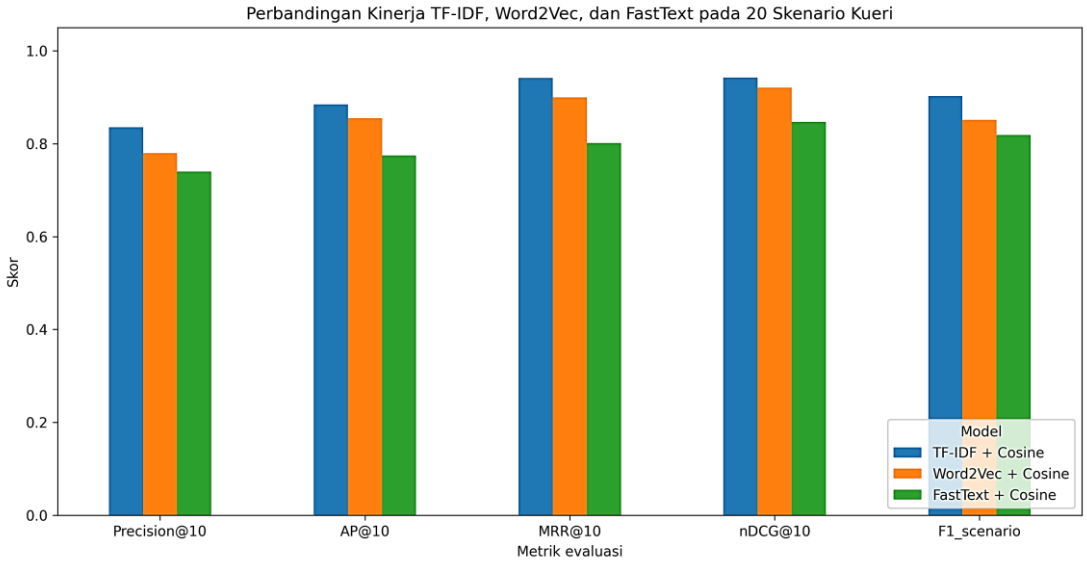
3.2. Perbandingan Model

Eksperimen dilakukan dengan membandingkan tiga model, yaitu TF-IDF + Cosine Similarity, Word2Vec + Cosine Similarity, dan FastText + Cosine Similarity. Seluruh model menggunakan dataset, prapemrosesan, kueri evaluasi, definisi relevansi, dan nilai K yang sama, yaitu Top-10. Tabel 4 menunjukkan bahwa TF-IDF + Cosine Similarity memperoleh performa terbaik secara keseluruhan. Model ini memperoleh Precision@10 sebesar 0,835, yang berarti rata-rata 8,35 dari 10 film yang direkomendasikan relevan

berdasarkan genre ekspektasi kueri. TF-IDF juga memperoleh AP@10 sebesar 0,884, MRR@10 sebesar 0,942, nDCG@10 sebesar 0,942, dan F1 skenario sebesar 0,902. Nilai ini menunjukkan bahwa pendekatan leksikal mampu menghasilkan daftar rekomendasi yang relevan dan memiliki kualitas pemeringkatan yang baik. Perbandingan visual metrik evaluasi ketiga model ditunjukkan pada Gambar 6.

Tabel 4. Perbandingan kinerja tiga model rekomendasi pada 20 skenario kueri

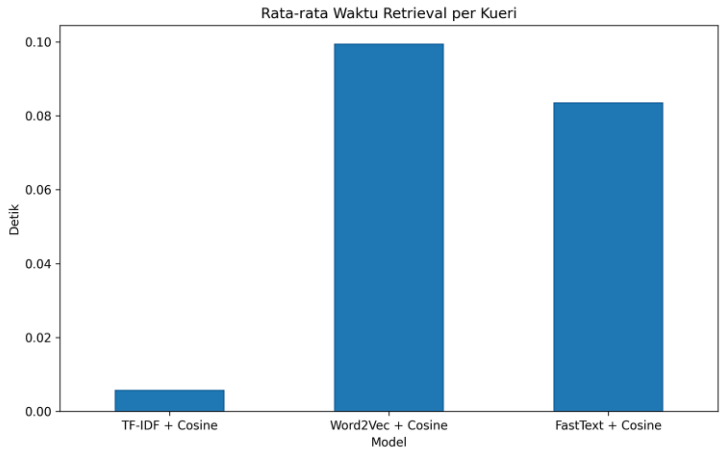
Model	P@10	HitRate	AP@10	MRR@10	nDCG@10	F1	Time (s)
TF-IDF + Cosine	0.835	1.000	0.884	0.942	0.942	0.902	0.0058
Word2Vec + Cosine	0.780	1.000	0.855	0.900	0.920	0.851	0.0995
FastText + Cosine	0.740	0.950	0.775	0.802	0.847	0.819	0.0836



Gambar 6. Perbandingan metrik evaluasi TF-IDF, Word2Vec, dan FastText

Word2Vec + Cosine Similarity berada pada posisi kedua dengan Precision@10 sebesar 0,780 dan F1 skenario sebesar 0,851. Model ini tetap menghasilkan HitRate@10 sebesar 1,000, yang berarti seluruh skenario kueri masih memiliki minimal satu item relevan dalam Top-10 rekomendasi. FastText + Cosine Similarity memperoleh Precision@10 sebesar 0,740 dan HitRate@10 sebesar 0,950. Nilai ini menunjukkan bahwa FastText tetap mampu menghasilkan rekomendasi relevan pada sebagian besar skenario, tetapi secara rata-rata belum mengungguli TF-IDF dan Word2Vec dalam eksperimen ini.

Dari sisi waktu retrieval, TF-IDF menjadi model tercepat dengan rata-rata 0,0058 detik per kueri. FastText membutuhkan rata-rata 0,0836 detik, sedangkan Word2Vec membutuhkan rata-rata 0,0995 detik. Perbedaan ini menunjukkan bahwa TF-IDF tidak hanya unggul pada metrik relevansi, tetapi juga lebih efisien dalam proses retrieval pada eksperimen ini. Perbandingan waktu retrieval rata-rata per model ditampilkan pada Gambar 7.



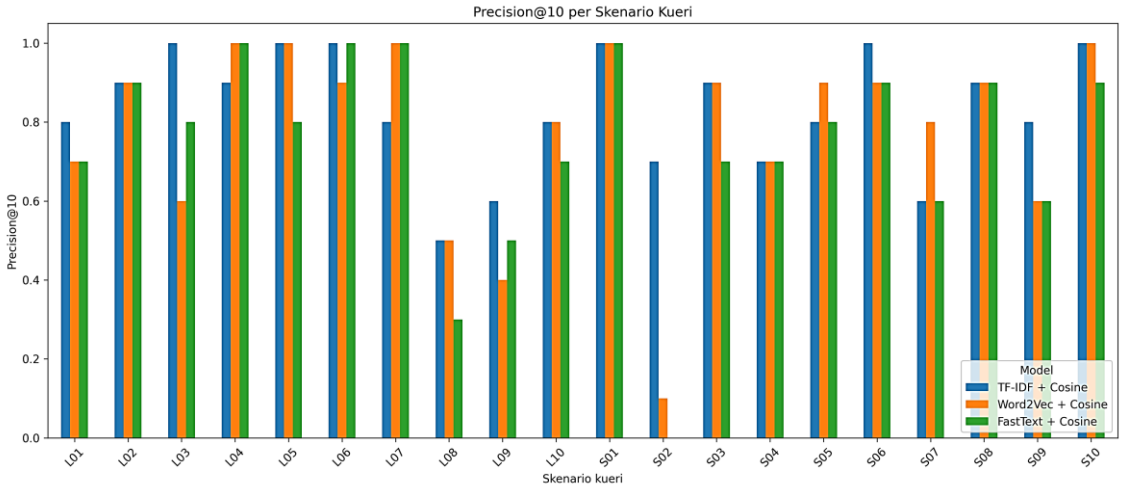
Gambar 7. Waktu retrieval rata-rata per model

3.3. Precision@10 per Skenario Kueri

Analisis per skenario kueri dilakukan untuk mengetahui bagaimana setiap model merespons tema yang berbeda. Hasil Precision@10 untuk 20 skenario kueri ditampilkan pada Tabel 5. Visualisasi perbandingan Precision@10 antarmodel untuk setiap skenario disajikan pada Gambar 8.

Tabel 5. Precision@10 per skenario kueri

Kode	Tema	TF-IDF	Word2Vec	FastText
L01	Romance / first sight love	0.80	0.70	0.70
L02	Magical world / mystical creatures	0.90	0.90	0.90
L03	Supportive family	1.00	0.60	0.80
L04	Space mission / distant planet	0.90	1.00	1.00
L05	Detective / unsolved murder	1.00	1.00	0.80
L06	Zombie outbreak / survival	1.00	0.90	1.00
L07	Musician / fame and love	0.80	1.00	1.00
L08	Soldier / war trauma	0.50	0.50	0.30
L09	Western / outlaw gang	0.60	0.40	0.50
L10	Animated animal adventure	0.80	0.80	0.70
S01	Cold-blooded killer	1.00	1.00	1.00
S02	Haunted house	0.70	0.10	0.00
S03	Space mission	0.90	0.90	0.70
S04	High school romance	0.70	0.70	0.70
S05	Bank robbery	0.80	0.90	0.80
S06	Dragon quest	1.00	0.90	0.90
S07	Time travel	0.60	0.80	0.60
S08	Secret agent	0.90	0.90	0.90
S09	Lost child	0.80	0.60	0.60
S10	World war	1.00	1.00	0.90



Gambar 8. Precision@10 per skenario kueri untuk tiga model

TF-IDF unggul atau setara pada sebagian besar skenario kueri, terutama ketika kueri memuat kata kunci eksplisit seperti zombie, dragon, killer, agent, dan war. Word2Vec menunjukkan kinerja baik pada beberapa skenario yang kaya secara semantik, seperti space mission, detective murder, musician, time travel, dan world war. FastText mencapai skor tinggi pada beberapa skenario, tetapi mengalami penurunan tajam pada kueri yang sangat pendek dan ambigu, seperti haunted house. Hal ini menunjukkan bahwa informasi subword saja belum cukup ketika kueri terlalu pendek dan representasi dokumen hanya dibentuk melalui rata-rata vektor kata.

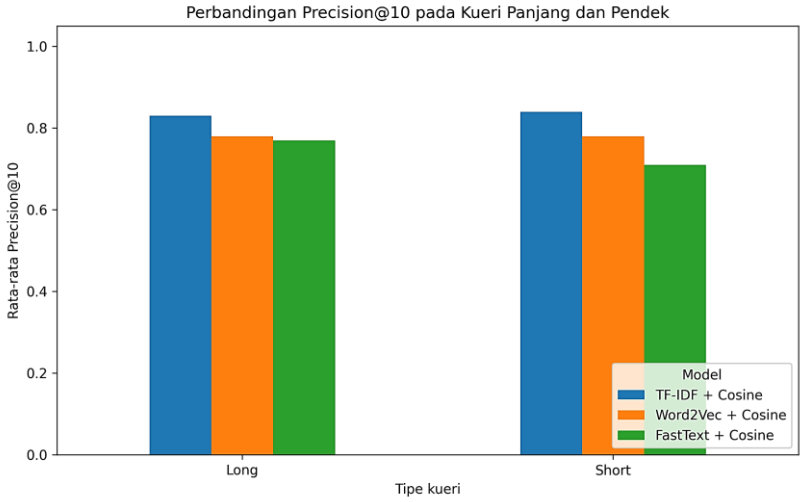
3.4. Kinerja Berdasarkan Tipe Kueri

Untuk memahami pengaruh panjang kueri, hasil evaluasi dikelompokkan menjadi dua tipe, yaitu kueri panjang dan kueri pendek. Ringkasan hasil ditampilkan pada Tabel 6. Perbandingan performa berdasarkan tipe kueri menunjukkan bahwa TF-IDF memperoleh Precision@10 terbaik baik pada kueri panjang maupun kueri pendek. Pada kueri panjang, TF-IDF memperoleh Precision@10 sebesar 0,830, sedangkan Word2Vec dan FastText masing-masing memperoleh 0,780 dan 0,770. Pada kueri pendek, TF-IDF memperoleh 0,840, sedangkan Word2Vec dan FastText memperoleh 0,780 dan 0,710. Hasil ini menunjukkan bahwa panjang kueri bukan satu-satunya faktor yang menentukan keunggulan model. Keberadaan kata tematik eksplisit dan karakter

sinopsis yang relatif pendek membuat pencocokan leksikal menjadi sangat efektif. Visualisasi perbandingan Precision@10 berdasarkan tipe kueri ditunjukkan pada Gambar 9.

Tabel 6. Perbandingan performa berdasarkan tipe kueri

Model	Tipe	P@10	AP@10	MRR@10	nDCG@10	F1
TF-IDF + Cosine	Panjang	0.830	0.862	0.883	0.924	0.898
Word2Vec + Cosine	Panjang	0.780	0.817	0.850	0.894	0.860
FastText + Cosine	Panjang	0.770	0.800	0.753	0.867	0.850
TF-IDF + Cosine	Pendek	0.840	0.906	1.000	0.960	0.907
Word2Vec + Cosine	Pendek	0.780	0.892	0.950	0.947	0.843
FastText + Cosine	Pendek	0.710	0.750	0.850	0.827	0.788



Gambar 9. Perbandingan Precision@10 pada kueri panjang dan kueri pendek

3.5. Pembahasan

Hasil penelitian menunjukkan bahwa TF-IDF + Cosine Similarity menjadi model terbaik secara keseluruhan dalam eksperimen 20 skenario kueri. Hasil ini sejalan dengan penelitian rekomendasi berbasis konten yang menunjukkan bahwa TF-IDF dan cosine similarity masih efektif untuk mencocokkan deskripsi item berbasis teks [6], [8], [9]. Namun, penelitian ini menambahkan kontribusi melalui perbandingan langsung dengan Word2Vec dan FastText pada kondisi cold-start berbasis sinopsis. Dalam dataset ini, sinopsis memiliki rata-rata panjang 25,43 token, sehingga representasi leksikal masih mampu menangkap kata-kata kunci utama yang membedakan tema film.

Keunggulan TF-IDF terlihat konsisten pada kueri panjang dan kueri pendek. Pada kueri panjang, TF-IDF mampu mempertahankan Precision@10 sebesar 0,830. Pada kueri pendek, TF-IDF memperoleh Precision@10 sebesar 0,840. Penyebab utamanya adalah banyak skenario kueri memuat kata tematik eksplisit, seperti killer, dragon, war, space, zombie, detective, haunted, dan agent. Kata-kata tersebut memiliki daya diskriminasi tinggi dalam korpus sinopsis film. TF-IDF memberi bobot lebih besar pada kata yang informatif dan relatif jarang, sehingga film dengan sinopsis yang memuat kata tematik serupa dapat ditempatkan pada peringkat atas.

Word2Vec berada pada posisi kedua dan tetap menunjukkan performa yang baik pada beberapa skenario. Word2Vec unggul atau setara dengan model terbaik pada kueri seperti space mission, detective murder, musician, time travel, dan world war. Hal ini menunjukkan bahwa representasi berbasis konteks kata tetap berguna ketika kueri mengandung nuansa semantik yang dapat ditangkap melalui kedekatan embedding. Namun, performa Word2Vec menurun pada beberapa kueri yang sangat bergantung pada kata kunci spesifik, seperti haunted house. Pada skenario tersebut, representasi dokumen berbasis rata-rata embedding dari sedikit kata cenderung menghaluskan sinyal kata kunci, sehingga film yang tidak memiliki kedekatan genre dapat memperoleh skor kemiripan yang relatif tinggi.

FastText memperoleh performa lebih rendah dibandingkan TF-IDF dan Word2Vec pada eksperimen 20 kueri. Meskipun FastText dirancang untuk memanfaatkan informasi subword, keunggulan tersebut belum terlihat dominan pada dataset ini. Salah satu kemungkinan penyebabnya adalah kueri dan sinopsis menggunakan kosakata bahasa Inggris yang relatif umum, sehingga manfaat subword tidak sebesar pada kasus yang memiliki banyak kata langka, kesalahan ejaan, variasi morfologis kompleks, atau out-of-vocabulary words. Selain itu, representasi dokumen yang digunakan dalam penelitian ini masih berupa rata-rata vektor

kata. Teknik rata-rata vektor sederhana belum memanfaatkan urutan kata, struktur frasa, dan relasi antarkalimat, sehingga informasi naratif yang lebih halus dapat hilang.

Hasil ini memperjelas bahwa klaim "embedding semantik selalu lebih baik daripada TF-IDF" tidak tepat untuk konteks penelitian ini. Efektivitas model sangat dipengaruhi oleh karakteristik dataset, panjang sinopsis, panjang kueri, cara membentuk vektor dokumen, serta cara relevansi didefinisikan. Pada dataset sinopsis pendek dengan kueri tematik eksplisit, TF-IDF dapat menjadi baseline yang sangat kuat. Sebaliknya, Word2Vec dan FastText tetap relevan sebagai pembanding semantik, tetapi membutuhkan strategi agregasi yang lebih kuat, pelatihan dengan korpus yang lebih besar, atau representasi tingkat kalimat agar keunggulan semantiknya lebih terlihat.

Implikasi praktis dari hasil penelitian adalah bahwa sistem rekomendasi film berbasis sinopsis tidak selalu harus menggunakan model neural yang berat pada tahap awal pengembangan. Untuk katalog berukuran sedang, skenario cold-start, dan kebutuhan retrieval cepat, TF-IDF + cosine similarity dapat digunakan sebagai baseline produksi awal karena prosesnya cepat, mudah dijelaskan kepada pengembang, dan tidak membutuhkan GPU. Model embedding atau transformer tetap dapat dipertimbangkan pada tahap berikutnya apabila sistem membutuhkan pemahaman semantik yang lebih mendalam, koreksi variasi bahasa, atau personalisasi berbasis interaksi pengguna.

Meskipun hasil penelitian menunjukkan performa yang cukup baik, terdapat beberapa keterbatasan. Pertama, evaluasi menggunakan genre-proxy relevance karena dataset tidak memiliki label relevansi eksplisit dari pengguna. Genre hanya merepresentasikan relevansi secara umum dan belum tentu mencerminkan kecocokan naratif secara mendalam. Kedua, penelitian ini belum menggunakan data interaksi pengguna seperti rating, klik, atau riwayat tontonan. Oleh karena itu, hasil penelitian lebih tepat diposisikan sebagai evaluasi content-based cold-start, bukan evaluasi personalisasi penuh. Ketiga, model Word2Vec dan FastText menggunakan rata-rata vektor kata, sehingga belum mempertimbangkan urutan kata dan struktur kalimat. Keempat, eksperimen ini belum menggunakan pengujian multi-seed atau uji signifikansi statistik. Kelima, penelitian ini belum mengevaluasi aspek coverage, diversity, novelty, serendipity, dan bias popularitas, padahal aspek tersebut penting dalam evaluasi sistem rekomendasi modern [23], [24].

4. KESIMPULAN

Penelitian ini mengevaluasi sistem rekomendasi film berbasis sinopsis pada skenario cold-start dengan membandingkan TF-IDF, Word2Vec, dan FastText yang dikombinasikan dengan cosine similarity. Hasil utama menunjukkan bahwa TF-IDF menjadi pendekatan paling stabil dan efektif pada dataset sinopsis film yang relatif pendek, baik untuk kueri panjang bernarasi maupun kueri pendek berbasis kata kunci.

Keunggulan TF-IDF tidak hanya terlihat dari metrik Top-K, tetapi juga dari efisiensi retrieval. Secara substantif, hasil ini menunjukkan bahwa kata kunci tematik eksplisit dalam sinopsis masih menjadi sinyal kuat untuk rekomendasi film cold-start. Word2Vec dan FastText tetap bermanfaat sebagai pembanding semantik, tetapi pada eksperimen ini rata-rata embedding belum mampu mengungguli representasi leksikal karena sinopsis pendek dan proses agregasi vektor cenderung meratakan informasi kata kunci spesifik.

Kontribusi ilmiah penelitian ini adalah memberikan bukti empiris bahwa model ringan dan interpretabel masih kompetitif untuk rekomendasi cold-start berbasis sinopsis ketika dibandingkan secara terkontrol dengan embedding ringan. Implikasi praktisnya, TF-IDF + cosine similarity layak digunakan sebagai baseline atau prototipe awal sistem rekomendasi film, terutama pada lingkungan dengan sumber daya komputasi terbatas, kebutuhan retrieval cepat, dan belum tersedianya data interaksi pengguna. Penelitian selanjutnya disarankan melibatkan anotasi relevansi dari pengguna atau pakar, melakukan pengujian multi-seed dan uji signifikansi statistik, menambahkan model kalimat seperti SBERT atau transformer-based embedding, serta mengevaluasi coverage, diversity, novelty, serendipity, dan bias popularitas.

REFERENSI

[1] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for YouTube recommendations," in Proc. 10th ACM Conf. Recommender Systems, 2016, pp. 191–198, doi: 10.1145/2959100.2959190.

[2] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," ACM Comput. Surv., vol. 52, no. 1, pp. 1–38, 2019, doi: 10.1145/3285029.

[3] O. A. S. Ibrahim, E. M. G. Younis, E. A. Mohamed, and A. Abraham, "Revisiting recommender systems: an investigative survey," Neural Comput. Appl., vol. 37, pp. 2145–2173, 2025, doi: 10.1007/s00521-024-10828-5.

[4] P. Khadka and P. Lamichhane, "Content-based recommendation engine for video streaming platform," arXiv, 2023, doi: 10.48550/arXiv.2308.08406.

[5] L. Wu, X. He, X. Wang, K. Zhang, and M. Wang, "A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation," IEEE Trans. Knowl. Data Eng., vol. 35, no. 5, pp. 4425–4445, 2023, doi: 10.1109/TKDE.2022.3145690.

[6] A. H. J. Permana and A. T. Wibowo, "Movie recommendation system based on synopsis using content-based filtering with TF-IDF and cosine similarity," Int. J. Inf. Commun. Technol., vol. 9, no. 2, pp. 1–14, 2023, doi: 10.21108/ijoict.v9i2.747.
[7] M. Chibb, P. Vashisht, A. Katti, and A. Narang, "Enhancing movie recommendations: A content based approach using TF-IDF weighted Word2Vec and cosine similarity," in 2024 4th Int. Conf. Technological Advancements in Computational Sciences (ICTACS), 2024, pp. 1449–1454, doi: 10.1109/ICTACS62700.2024.10841087.
[8] A. Febrian and E. D. Permana, "Sistem rekomendasi film menggunakan metode content based filtering dengan algoritma TF-IDF," Al-Aqlu: J. Mat. Tek. Sains, vol. 4, no. 1, pp. 19–25, 2026, doi: 10.59896/aqlu.v4i1.494.
[9] E. Z. Irabela and Norhikmah, "Implementation of content-based cosine similarity algorithm with TF-IDF and SBERT for movie recommendation," J. Sistem Cerdas, vol. 9, no. 1, pp. 114–122, 2026, doi: 10.37396/jsc.v9i1.633.
[10] O. Barkan and N. Koenigstein, "Item2Vec: Neural item embedding for collaborative filtering," in 2016 IEEE 26th Int. Workshop Machine Learning for Signal Processing, 2016, pp. 1–6, doi: 10.1109/MLSP.2016.7738876.
[11] C. González-Santos, M. A. Vega-Rodríguez, J. M. López-Muñoz, I. Martínez-Sarriegui, and C. J. Pérez, "Automatic assignment of microgenres to movies using a word embedding-based approach," Multimedia Tools Appl., vol. 83, pp. 48719–48735, 2023, doi: 10.1007/s11042-023-17442-y.
[12] A. Fellah, A. Zahaf, and A. Elçi, "Semantic similarity measure using a combination of Word2Vec and WordNet models," Indones. J. Electr. Eng. Informatics, vol. 12, no. 2, pp. 455–464, 2024, doi: 10.52549/ijeei.v12i2.5114.
[13] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," Trans. Assoc. Comput. Linguistics, vol. 5, pp. 135–146, 2017, doi: 10.1162/tacl_a_00051.
[14] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S. Chua, "Neural collaborative filtering," in Proc. 26th Int. Conf. World Wide Web, 2017, pp. 173–182, doi: 10.1145/3038912.3052569.
[15] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," in Int. Conf. Learning Representations, 2016.
[16] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in 2018 IEEE Int. Conf. Data Mining, 2018, pp. 197–206, doi: 10.1109/ICDM.2018.00035.
[17] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer," in Proc. 28th ACM Int. Conf. Inf. Knowl. Manage., 2019, pp. 1441–1450, doi: 10.1145/3357384.3357895.
[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.
[19] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in Proc. EMNLP-IJCNLP, 2019, pp. 3982–3992, doi: 10.18653/v1/D19-1410.
[20] S. Rendle, W. Krichene, L. Zhang, and J. Anderson, "Neural collaborative filtering vs. matrix factorization revisited," in Proc. 14th ACM Conf. Recommender Systems, 2020, pp. 240–248, doi: 10.1145/3383313.3412488.
[21] R. Cañamares and P. Castells, "On target item sampling in offline recommender system evaluation," in Proc. 14th ACM Conf. Recommender Systems, 2020, pp. 259–268, doi: 10.1145/3383313.3412259.
[22] E. Zangerle and C. Bauer, "Evaluating recommender systems: Survey and framework," ACM Comput. Surv., vol. 55, no. 8, pp. 1–38, 2022, doi: 10.1145/3556536.
[23] A. Klimashevskaia, D. Jannach, M. Elahi, and C. Trattner, "A survey on popularity bias in recommender systems," User Model. User-Adapted Interact., 2024, doi: 10.1007/s11257-024-09406-0.
[24] J. Chen, H. Dong, X. Wang, F. Feng, M. Wang, and X. He, "Bias and debias in recommender system: a survey and future directions," ACM Trans. Inf. Syst., vol. 41, no. 3, pp. 1–39, 2023, doi: 10.1145/3564284.