

IMPLEMENTASI METODE *HYBRID FILTERING (CONTENT BASED AND COLLABORATIVE)* DALAM SISTEM REKOMENDASI FILM

Muhamad Septian, Hendrik Fery Herdiatmoko

Informatika, Universitas Katolik Musi Charitas

Jl. Bangau No.60, 9 Ilir, Kec. Ilir Timur III, Palembang, Sumatera Selatan, Indonesia

mseptian939@gmail.com

ABSTRAK

Lonjakan volume data pada *platform streaming* memicu fenomena kelebihan informasi (*information overload*), sehingga diperlukan sistem rekomendasi yang mampu menyajikan konten secara personal dan efisien. Namun, penerapan algoritma tunggal seperti *Content-Based Filtering* (CBF) maupun *Collaborative Filtering* (CF) seringkali dihadapkan pada batasan isolasi algoritma, seperti rentannya sistem terhadap ketidaksediaan interaksi pengguna baru (*cold-start*) dan kekosongan matriks (*data sparsity*). Oleh karena itu, penelitian ini bertujuan mengembangkan arsitektur sistem rekomendasi film dengan mengusulkan pendekatan *Weighted Hybrid Filtering* untuk mengatasi kelemahan tersebut. Metode penelitian mengintegrasikan algoritma *Content-Based Filtering* (berbasis TF-IDF + *cosine similarity*) dan *Collaborative Filtering* (berbasis *Truncated SVD* $k=50$). Model dibangun menggunakan data multi-sumber, yaitu metadata film dari TMDB dan data rating dari MovieLens, dengan skema pembobotan 40:60. Evaluasi dilakukan menggunakan *5-Fold Cross Validation* dengan metrik RMSE, *Hit Rate@10*, dan *Mean Average Precision* (MAP). Hasil eksperimen menunjukkan bahwa model hybrid yang diusulkan mampu mengungguli metode tunggal dengan nilai RMSE sebesar 0,731, *Hit Rate@10* sebesar 0,716, dan MAP sebesar 0,268. Hasil ini menunjukkan bahwa pendekatan hybrid efektif dalam meningkatkan akurasi serta kualitas rekomendasi. Dengan demikian, penelitian ini memberikan kontribusi dalam pengembangan sistem rekomendasi yang tidak hanya akurat secara prediktif, tetapi juga aplikatif untuk diimplementasikan dalam platform berbasis web.

Kata kunci : Sistem Rekomendasi, *Hybrid Filtering*, TF-IDF, *Singular Value Decomposition*, Django

1. PENDAHULUAN

Dalam era digital saat ini, sistem rekomendasi telah menjadi elemen krusial pada berbagai platform *streaming* dan hiburan. Lonjakan volume data film menuntut adanya sistem pencarian informasi yang terpersonalisasi dan efisien [1].

Sistem ini menganalisis preferensi dan riwayat interaksi pengguna untuk menyajikan saran film yang relevan, sehingga mengurangi kelebihan informasi (*information overload*) bagi pengguna [2].

Bagi penyedia layanan, keberadaan sistem rekomendasi yang akurat berkorelasi langsung dengan peningkatan kepuasan pelanggan, retensi pengguna, dan pada akhirnya berdampak positif pada keuntungan perusahaan [3].

Meskipun platform *streaming* menawarkan akses tak terbatas ke ribuan judul film, kemudahan ini sering kali memicu fenomena *paradox of choice*. Berdasarkan survei terhadap 72 responden penikmat film, sekitar 88,6% di antaranya justru mengalami kebingungan saat menentukan film yang hendak ditonton [4].

Hal ini mengindikasikan bahwa ketersediaan opsi yang melimpah tanpa panduan yang personal dapat menurunkan kualitas pengalaman menonton. Oleh karena itu, sistem rekomendasi tidak hanya berfungsi sebagai fitur pelengkap, melainkan sebagai instrumen strategis untuk meningkatkan durasi kunjungan (*screen time*) dan keterlibatan pengguna di dalam [5], [6]

Tanpa adanya sistem rekomendasi yang andal, potensi film-film menarik untuk ditemukan menjadi terlewatkan.

Untuk memecahkan masalah tersebut, berbagai pendekatan algoritma telah banyak diimplementasikan. Namun, permasalahan utama yang sering dihadapi adalah pendekatan konvensional seperti *Content-Based* yang bekerja dengan merekomendasikan item berdasarkan ekstraksi fitur, rentan terhadap masalah *over-specialization*. Di sisi lain, *Collaborative* memanfaatkan pola interaksi komunitas pengguna untuk membuat rekomendasi, tetapi memiliki kelemahan mendasar pada masalah *cold-start* (kesulitan memberikan rekomendasi pada pengguna/item baru) dan data *sparsity* (kekosongan matriks peringkat yang ekstrem) [6].

Dengan demikian, metode tunggal belum mampu menghasilkan rekomendasi yang optimal, baik dari sisi akurasi maupun keberagaman hasil. Sebagai rencana penyelesaian masalah untuk menutupi kelemahan masing-masing metode dasar tersebut, penelitian ini mengusulkan pendekatan *Hybrid Filtering* dengan teknik *weighted*. Integrasi dari kedua metode ini terbukti mampu menghasilkan rekomendasi yang jauh lebih relevan, beragam, dan akurat.

Metode *hybrid filtering* yang menggabungkan *Collaborative Filtering* (CF) dan *Content-Based Filtering* (CBF) secara konsisten terbukti lebih unggul dalam berbagai metrik evaluasi dibandingkan metode tunggal [7].

Pendekatan hibrida ini menjadi solusi ideal untuk saling menutupi kelemahan tiap metode, terutama untuk mengatasi *data sparsity* dan *cold-start problem* pada pengguna baru di model CF [6].

Walaupun penerapan algoritma SVD pada CF terbukti sangat efektif dalam mereduksi dimensi dan memprediksi rating secara akurat, integrasinya dengan CBF tetap sangat disarankan agar performa rekomendasi lebih maksimal [8].

Keandalan arsitektur hibrida ini telah dibuktikan secara spesifik oleh penelitian Mukti dan Mardhiyah [9] pada domain film animasi, di mana sistem mereka sukses menunjukkan performa solid dengan pencapaian presisi 71%, MAE 0,9, dan RMSE 1,1985. Meski terbukti akurat dan relevan, sistem hibrida pada penelitian tersebut masih dibangun menggunakan *Flask*. Hal ini membuka peluang untuk pengembangan lebih lanjut menggunakan *framework Python* lainnya, seperti *Django*, guna meningkatkan kapabilitas sistem.

Untuk mengisi celah tersebut, tujuan utama dari penelitian ini adalah untuk merancang bangun sistem rekomendasi film berbasis *Hybrid Filtering* yang tidak hanya tangguh secara algoritma, tetapi juga diterapkan penuh ke dalam aplikasi web dinamis menggunakan *framework Django*. Pembaruan utama dalam penelitian ini terletak pada dua aspek: (1) Integrasi data multi-sumber dengan pemodelan hibrida terbobot (*weighted*), di mana sistem menggabungkan kelengkapan metadata dari API TMDb dan data rating dari MovieLens untuk diproses melalui perpaduan *Model-Based Collaborative Filtering* (menggunakan Truncated SVD) dan *Content-Based Filtering* (menggunakan TF-IDF serta *Cosine Similarity*), dan (2) Transisi dari evaluasi teoretis ke implementasi praktis, yang membawa hasil riset dari sekadar perhitungan metrik statistik menjadi sebuah platform web interaktif yang siap digunakan. Melalui arsitektur sistem ini, luaran penelitian diharapkan tidak sebatas menghasilkan metrik evaluasi yang rendah *error*, tetapi murni menghadirkan solusi praktis dan *real-time* bagi pengguna dalam memecahkan masalah pencarian film secara personal.

2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Penelitian mengenai sistem rekomendasi berbasis *Collaborative Filtering* (CF) sering kali dihadapkan pada kendala *data sparsity* dan *cold-start problem*, yang terbukti mengakibatkan penurunan performa secara drastis pada pengguna baru [6].

Meskipun penerapan algoritma *Singular Value Decomposition* (SVD) efektif dalam mereduksi dimensi dan memprediksi *rating* secara akurat, Sitorus [8] merekomendasikan perlunya integrasi metode ini dengan *Content-Based Filtering* (CBF). Pendekatan *Hybrid Filtering* tersebut dinilai sebagai solusi ideal untuk menutupi kelemahan inheren dari masing-masing metode, yakni mengatasi *over-specialization* pada CBF sekaligus memecahkan masalah *cold-start* pada CF. Keandalan arsitektur hibrida ini telah

dikonfirmasi oleh Mukti dan Mardhiyah [9] yang sukses mencapai tingkat presisi sebesar 71% pada sistem rekomendasi yang mereka kembangkan. Meskipun terbukti akurat, sistem rekomendasi tersebut masih diimplementasikan berbasis web menggunakan *framework Flask*. Keterbatasan ini membuka peluang pengembangan lebih lanjut untuk membangun sistem rekomendasi menggunakan arsitektur *framework Python* lainnya yang lebih komprehensif, seperti *Django*.

2.2. Hybrid Filtering

Hybrid filtering adalah metode yang menggabungkan beberapa model yang Salah satu pendekatannya adalah *weighted hybrid*, yaitu menggabungkan skor dari *content-based* dan *collaborative filtering* berdasarkan bobot tertentu [10]. Yang dapat dihitung dengan rumus:

$$hybrid = (\alpha \times CBF) + (\beta \times CF) \quad (1)$$

Dimana:

α dan β = Bobot yang diberikan dari setiap metode

CBF = Bobot hasil dari metode *Content-Based*

CF = Bobot hasil dari metode *Collaborative*

Karena rentang skor prediksi CF (1-5) dan jarak CBF (0-1) berbeda, maka dilakukan tahap normalisasi menggunakan Min-Max Scaling untuk mendistribusikan kedua skor tersebut ke dalam rentang rasio (0-1) yang seragam. Dimana dapat di hitung dengan rumus:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (2)$$

Dimana:

X = Skor aktual

X_{max} = Skor maksimum

X_{min} = Skor Minimum

Content-based memilih *item* untuk pengguna berdasarkan kesamaan antara konten *item* dan preferensi pengguna [11].

Banyak algoritma digunakan dalam *content-based*, salah satunya adalah TF-IDF, yang menghitung pentingnya kata-kata dalam deskripsi atau sinopsis film pada, yang dapat dihitung dengan rumus:

$$tf_i = \frac{n_{i,u}}{|u|} \quad (3)$$

$$idf_i = \log \left(\frac{|D|}{df_i} \right) \quad (4)$$

$$tfidf_i = tf_i \times idf_i \quad (5)$$

Dimana:

$n_{i,u}$ = jumlah kemunculan kata dalam suatu film

$|u|$ = panjang kata dalam suatu film

$|D|$ = total film yang ada

df_i = jumlah film yang ada kata i

Dan *cosine similarity*, yang mengukur kemiripan antara film berdasarkan vektor fitur, yang dapat dihitung dengan rumus:

$$\text{Cos} = \frac{x \cdot y}{||x|| \times ||y||} \quad (6)$$

Dimana:

x = vektor *item* aktif dari perhitungan TF-IDF

y = vektor *item* acuan dari perhitungan TF-IDF

$|x|$ = panjang satuan vektor x

$|y|$ = panjang satuan vektor y

Collaborative memanfaatkan preferensi pengguna, salah satunya melalui pendekatan *model-based* dengan *Truncated SVD*. Metode ini mereduksi dimensi dengan mengambil k faktor laten terbesar, sehingga efektif mengatasi data sparsity, mengurangi noise, dan lebih efisien secara komputasi [11], [12].

Pada *Truncated SVD*, matriks rating didekonstruksi menjadi estimasi 3 bagian matriks yang difilter sebanyak k dimensi, seperti pada rumus sebagai berikut:

$$\hat{R} = U_k \Sigma_k V_k^T \quad (7)$$

Dimana:

$\hat{R}$ = matriks estimasi probabilitas *rating*

U_k = matriks singular pengguna (*user*) berukuran $m \times k$ (faktor laten)

Σ_k = matriks diagonal *singular value* berisi bobot sebesar $k \times k$

V_k^T = matriks transpose singular *item* berukuran $k \times n$ (faktor laten)

k = jumlah dimensi laten spesifik yang dipertahankan (*truncated*)

2.3. Evaluasi

Evaluasi sistem dilakukan untuk mengukur kinerja dan akurasi rekomendasi film menggunakan metode *k-Fold Cross Validation*. Dataset dibagi menjadi k bagian, lalu dilakukan k iterasi pelatihan dan validasi secara bergantian sehingga hasil lebih objektif dan minim bias [6].

Pengujian mencakup pengukuran *error* prediksi menggunakan RMSE untuk menilai tingkat kesalahan prediksi rating sistem [10].

Karena skor *Content-Based* berada pada rentang (0–1), maka skor *Collaborative* diubah ke skala (1–5) menggunakan normalisasi *Min-Max* agar perbandingan dengan rating aktual menjadi setara. Rumus normalisasi *Min-Max*:

$$X_{scaled} = X \times (max - min) + min \quad (8)$$

Dimana:

X = Skor aktual

Setelah penskalaan disamakan, akurasi *error* secara keseluruhan dapat dihitung dengan rumus:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (predicted_i - actual_i)^2} \quad (9)$$

Dimana:

n = jumlah data

$predicted_i$ = nilai prediksi

$actual_i$ = nilai aktual

Serta evaluasi dengan *Hit Rate@10* dan *Mean Average Precision@10* untuk menilai kualitas ranking. Yang dapat dihitung dengan rumus:

$$HR = \frac{|U_{hit}^k|}{|U_{all}|} \quad (10)$$

Dimana:

$|U_{hit}^k|$ = Jumlah pengguna dengan minimal 1 hit

$|U_{all}|$ = Total jumlah Pengguna

$$MAP@N = \frac{1}{|U|} \sum_{u \in U} (AP@N)_u \quad (11)$$

Dimana:

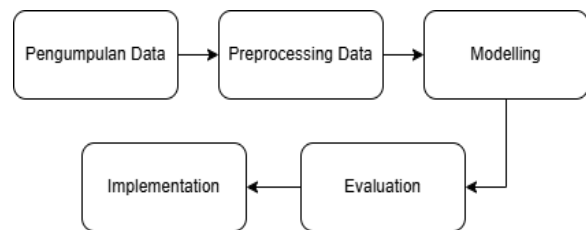
N = Jumlah item yang direkomendasikan

U = Kumpulan pengguna

3. METODE PENELITIAN

Dalam penelitian ini, sistem rekomendasi dibangun dengan pendekatan *Hybrid Filtering* dengan menggabungkan Term Frequency-Inverse Document Frequency dan Cosine Similarity (*Content-Based*), dan SVD (*Collaborative*). Teknik *Hybrid* yang dipilih adalah dengan menggunakan teknik *Weighted* sebagaimana ditemukan dalam penelitian sebelumnya [10].

Tahapan proses yang ditempuh dalam penelitian ini sebagai berikut:



Gambar 1. Tahapan Penelitian

3.1. Pengumpulan Data

Pada tahap awal penelitian ini, pengumpulan data dilakukan melalui dua jenis sumber yang berbeda. Sumber pertama merupakan data item film yang ditarik menggunakan antarmuka TMDB API, mencakup atribut ID, judul, tanggal rilis, poster, sinopsis (*overview*), dan genre, dengan total keseluruhan data terkumpul mencapai 9.619 film. Sementara itu, sumber kedua adalah data interaksi pengguna yang berasal dari *dataset* MovieLens hasil publikasi grup riset GroupLens. Data ini berisi rekam jejak historis yang memuat atribut User ID, Movie ID, serta nilai *rating* dengan skala 1 hingga 5. Setelah melalui tahapan pencocokan (*mapping*) antara data interaksi pengguna dengan metadata dari TMDB, penelitian ini secara keseluruhan berhasil mengekstraksi sebanyak 100.513 data *rating* yang berasal dari 611 pengguna.

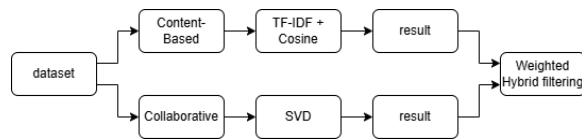
3.2. Preprocessing Data

Tahapan Tahap pra-pemrosesan data dilakukan untuk mengintegrasikan dan menyaring kualitas *dataset* melalui *index mapping*. Proses ini mencocokkan movieId (MovieLens) dengan tmdbId (TMDB) menggunakan links.csv, sekaligus membuang data *rating* yang tidak memiliki padanan valid. Pembersihan data kemudian dilanjutkan dengan normalisasi teks untuk *content-based* [13] dan penanganan *missing value* pada matriks *rating* [14].

Untuk efisiensi memori komputasi, *missing value* tersebut direpresentasikan sebagai nilai 0 menggunakan struktur *Compressed Sparse Row* (CSR) *Matrix* sebelum tahap faktorisasi.

3.3. Modelling

Pada tahap ini, model akan dibangun dengan membuat model *hybrid filtering* dengan teknik *weighted* dari gabungan *content-based* dan *collaborative* sebagai berikut:



Gambar 2. Model *Hybrid filtering*

Pemodelan sistem rekomendasi ini menggunakan pendekatan *Parallel Hybrid Design* yang memproses data melalui dua jalur algoritma secara bersamaan. Jalur pertama mengeksekusi C CBF menggunakan metode TF-IDF dan *Cosine Similarity* untuk memprediksi kemiripan fitur teks film. Secara paralel, jalur kedua menerapkan CF berbasis algoritma SVD pada matriks *rating* guna mereduksi dimensi data *sparse* dan mengekstrak preferensi pengguna yang tersembunyi. Pada tahap akhir, hasil prediksi dari kedua model digabungkan melalui *Weighted Hybrid Filtering* dengan kalibrasi bobot asimetris (40% CBF dan 60% CF) untuk menghasilkan daftar rekomendasi yang komprehensif dan mampu menutupi kelemahan masing-masing metode tunggal.

3.4. Evaluation

Evaluasi sistem rekomendasi ini menggunakan *k-Fold Cross Validation* guna menghindari bias acak pemilihan data [6]. Tingkat kesalahan prediksi dievaluasi dengan *Root Mean Square Error* (RMSE) [10], dengan menerapkan *Min-Max scaling* terlebih dahulu untuk menyetarakan metrik keluaran model *Collaborative* ke rentang (0, 1) milik model *Content-Based*. Selanjutnya, kualitas daftar rekomendasi diukur menggunakan *Hit Rate@10* untuk melihat peluang munculnya minimal satu film relevan pada 10 rekomendasi teratas [6], serta *MAP@10* untuk mengevaluasi kualitas pemeringkatan (*ranking optimization*) berdasarkan penempatan posisi prioritas film yang relevan.

3.5. Implementation

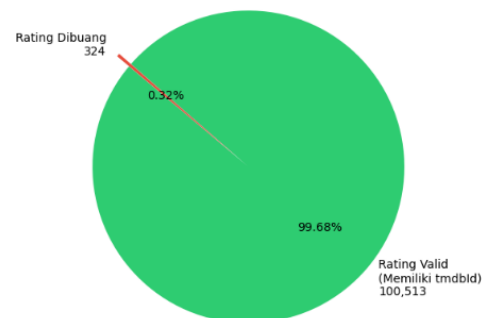
Tahap akhir penelitian ini mengintegrasikan model *hybrid filtering* ke dalam aplikasi web *real-time* menggunakan *framework* Django berbasis Python dengan arsitektur *Model-View-Template* (MVT) [15]. Lapis basis data dikelola menggunakan *SQLite3* untuk

menyimpan interaksi *rating* dan metadata film dari TMDB API. Pada mesin rekomendasi, kalkulasi komputasi *Content-Based* dieksekusi menggunakan pustaka *Scikit-Learn*, sementara prediksi *Collaborative* diselesaikan melalui kompresi matriks *Compressed Sparse Row* (CSR) dan dekomposisi *Truncated SVD* dari *SciPy*. Seluruh pemrosesan *backend* tersebut kemudian dirender secara dinamis menjadi antarmuka visual katalog film menggunakan *Django Template Language* (DTL), HTML, dan CSS.

4. HASIL DAN PEMBAHASAN

Pada bab ini disajikan hasil dari proses implementasi. Pembahasan dimulai dari tahap *preprocessing* data. Proses pengumpulan data awal menghasilkan 100.837 rekam *rating* dari 610 pengguna (MovieLens) dan 9.619 film. Berdasarkan hasil pemetaan indeks (*index mapping*) menggunakan parameter *links.csv*, sistem mendeteksi adanya data *rating* pada film yang tidak memiliki ID ekuivalen di database TMDB lokal, sehingga data tersebut dibuang demi menjaga konsistensi. Seperti yang ditunjukkan pada gambar berikut:

Proporsi Data *Rating* Setelah Pemetaan (Mapping) tmdbid



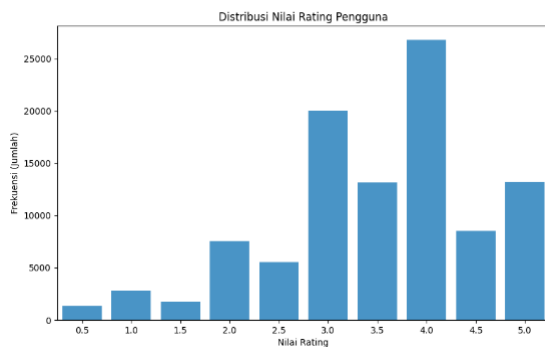
Gambar 3. Proporsi Data *Rating* Setelah Pemetaan

Setelah proses pembersihan (*cleaning*) selesai, sistem berhasil mempertahankan sebanyak 100.513 *rating* yang *valid*. Untuk kebutuhan pembobotan TF-IDF pada algoritma *Content-Based*, atribut teks berupa sinopsis (*overview*) dan genre setiap film dinormalisasi. Sistem telah mengeksekusi tahapan *case folding* (mengubah ke huruf kecil), *punctuation removal* (penghapusan tanda baca), dan *stopword removal* (penghapusan kata hubung yang tidak bermakna). Sampel perubahan teks fitur sebelum dan sesudah pra-pemrosesan dapat dilihat pada tabel berikut:

Tabel 1. Sampel Tranformasi Teks

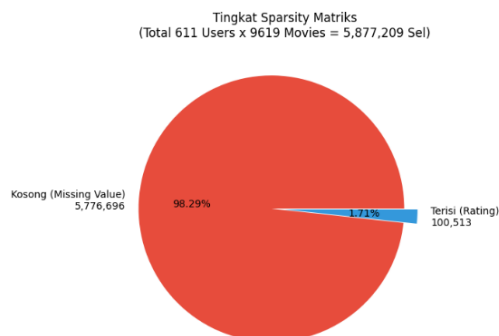
Judul	Genre	Overview	Hasil Preprocessing
Toy Story	Adventure, Animation, Comedy, Family	Led by Woody, Andy's toys live happily in his room until Andy's birthday brings Buzz Lightyear onto the scene. Afraid of losing his place in Andy's heart, Woody plots against Buzz. But when circumstances separate Buzz and Woody from their owner, the duo eventually learns to put aside their differences.	toy story adventure animation comedy family led woody andys toys live happily room until birthday brings buzz lightyear onto scene afraid losing place heart plots against circumstances separate owner duo eventually learns put aside differences

Untuk memahami profil ketertarikan dari 611 pengguna, sistem memvisualisasikan sebaran rating (dari skala 1.0 hingga 5.0) yang tersimpan dalam basis data akhir. Gambar (4) mengilustrasikan frekuensi pemberian nilai rating oleh pengguna. Berdasarkan grafik tersebut, terlihat bahwa mayoritas pengguna cenderung memberikan rating 4.0 disusul oleh rating 3.0 dan 5.0. Visualisasi ini merepresentasikan tingkat kecenderungan positif (bias positif) pengguna terhadap film yang pernah mereka tonton.



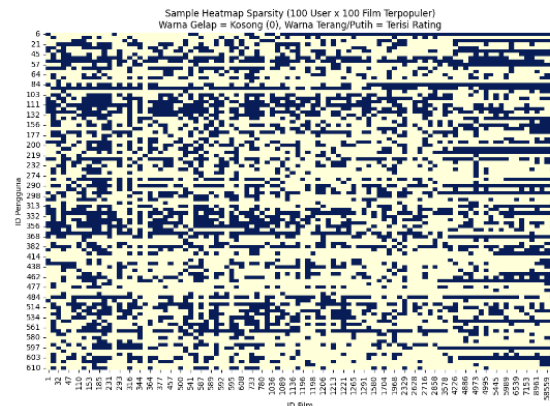
Gambar 4. Distribusi Rating

Berdasarkan data yang telah dibersihkan, matriks interaksi pengguna-film (*User-Item Matrix*) berhasil dibentuk dengan dimensi 611 baris pengguna $\times$ 9619 kolom film. Dari sel matriks yang tersedia, hanya 100.513 sel yang memiliki rating, sedangkan sisa sel bernilai nol (*missing value*) sebanyak 5.776.696 yang tidak memiliki rating. Berdasarkan Gambar *pie chart* dibawah, matriks ini memiliki tingkat kekosongan (*sparsity*) yang sangat tinggi, yaitu mencapai 98.29%..



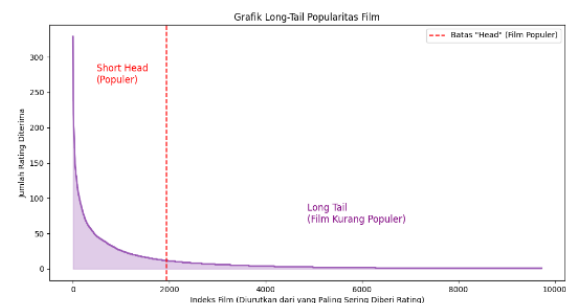
Gambar 5. Pie Chart Missing Value

Bentuk kekosongan ini dapat diamati melalui sampel *Heatmap* acak untuk kelompok matriks 100 film $\times$ 100 user, di mana sebagian besar wilayah berarsir warna gelap menandakan nilai 0/belum diisi. Tingkat ekstrim atas rasio kekosongan inilah yang membuktikan perlunya implementasi algoritma pemfaktoran matriks, yakni *SVD Truncated*. Sistem menggunakan *Compressed Sparse Row (CSR) Matrix* agar titik kosong berkapasitas 98.29% tersebut tidak menumpuk dalam ruang RAM di server Django.



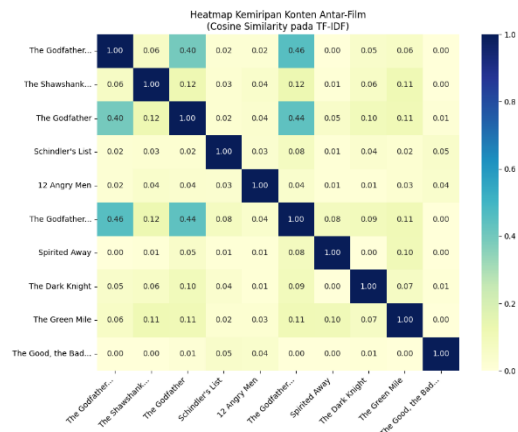
Gambar 6. Heatmap Missing Value

Visualisasi fenomena Long-Tail terhadap popularitas film yang ada dalam dataset pada sumbu X merepresentasikan indeks film, sementara sumbu Y menunjukkan jumlah rating yang tertumpuk pada indeks tersebut. Terlihat secara signifikan bahwa mayoritas interaksi rating pengguna hanya dihabiskan pada kelompok seperempat kecil film di sebelah kiri kurva (*Short Head*), sisa ribuan katalog film lainnya jatuh drastis ke dasar horizontal (area *Long Tail*). Oleh karena model CF sangat terpengaruh oleh jumlah rating masa lalu, maka film-film di area *Long-Tail* kerap kesulitan mencapai perhitungan kedekatan (*cold-start*). Tantangan kurva *long-tail* inilah yang akan dilampaui melalui integrasi pembobotan kombinasi dengan ekstraksi konten CBF.



Gambar 7. Long Tail Plot

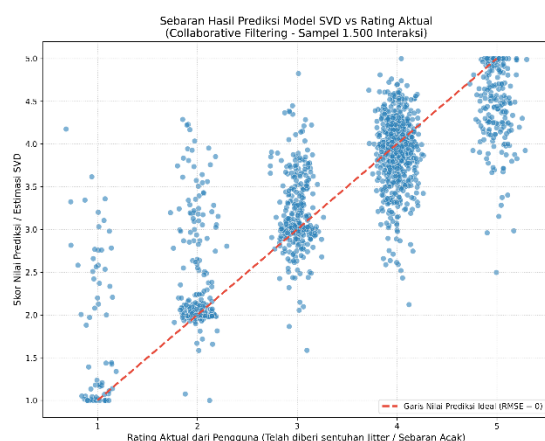
Sebelum sistem menerapkan pembobotan penggabungan (*Hybrid Filtering*), dilakukan iterasi pemodelan terhadap dua algoritma dasar (CBF dan CF) secara tunggal untuk mengevaluasi kapabilitas awal sistem. Algoritma *Content-Based* memproses data menggunakan kombinasi *Term Frequency-Inverse Document Frequency* (TF-IDF) untuk mengekstraksi matriks bobot kosakata sinopsis dan label genre film. Dari matriks turunan tersebut, tingkat kedekatan kemiripan (*relevansi*) antar satu film dengan yang lain dikalkulasi menggunakan persamaan matematika *Cosine Similarity*.



Gambar 8. Heatmap Kemiripan Konten

Pada gambar 8 memvisualisasikan matriks persilangan tingkat kemiripan konten (0.0 hingga 1.0) untuk sampel sepuluh judul film acak di dalam sistem. Semakin pekat warna sel (mendekati 1.0), maka semakin identik karakteristik kedua film tersebut. Melalui pendekatan CBF, sistem dapat merekomendasikan daftar judul dengan aman tanpa dipengaruhi oleh tren popularitas maupun histori preferensi orang lain (menangkal masalah kurva *long-tail* dan *cold-start* item). Sayangnya, karena CBF bekerja secara statis berbasis deskripsi kata, metode ini rentan pada komplikasi *over-specialization* di mana rekomendasi yang keluar menjadi monoton karena kurangnya elemen eksploratif di luar minat awal pengguna.

Sebagai penyeimbang untuk CBF, iterasi kedua pemodelan diarahkan pada teknik CF. Guna mengatasi permasalahan minimnya titik data historis, dengan pendekatan *Truncated SVD*. Matriks antarmuka pengguna-dan-item difaktorisasi ke dalam 50 komponen sub-laten spasial (*Rank k=50*).

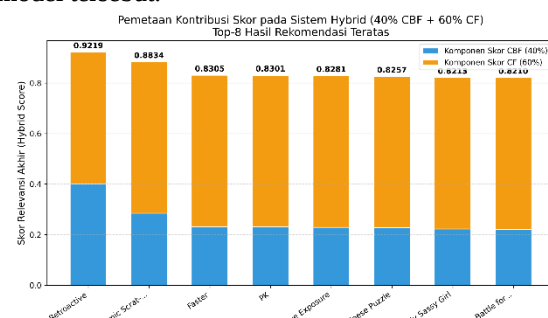


Gambar 9. Scatter Plot SVD

Pada gambar 9 diatas, menunjukkan sebaran akurasi prediksi model *Collaborative Filtering* (SVD) terhadap 1.500 sampel pengujian secara acak. Titik-titik distribusi (biru) merepresentasikan skor tebakan model SVD (Sumbu-Y) yang dibandingkan langsung

dengan nilai rating aktual di database (Sumbu-X). Berdasarkan *scatter plot* tersebut, kumpulan titik tebakan terdistribusi secara linier mengikuti kemiringan garis putus-putus merah (garis prediksi ideal di mana $error = 0$). Hal ini membuktikan algoritma faktorisasi matriks SVD bekerja secara akurat dalam memahami pola preferensi pengguna, model mampu memberikan estimasi skor yang tinggi (4-5) pada film yang memang disukai oleh pengguna, serta skor yang rendah pada film yang tidak disukai, tanpa mengalami *overfitting*.

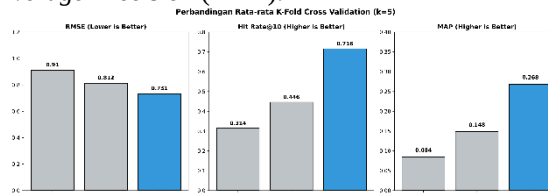
Sebelum dikalikan dengan bobot, skor mentah dari CF dinormalisasi menggunakan metode *Min-Max Scaling* ke dalam rentang 0 hingga 1 agar setara. Jika salah satu model gagal menghasilkan prediksi (misalnya item baru tanpa *rating* pada CF), sistem menanganinya dengan memberikan nilai rata-rata dari model tersebut.



Gambar 10. Bar Chart Hybrid

Pada gambar 10 diatas memvisualisasikan hasil komputasi *hybrid* pada 8 film rekomendasi teratas. Pengujian ini menggunakan data *User* yang paling aktif dengan rekam jejak 2.698 *rating*. Diagram batang tumpuk diatas secara detail memperlihatkan keunggulan proporsi dominan 60% dari pola *Collaborative* (warna oranye) dikombinasikan dengan dukungan 40% *Content-Based* (warna biru), yang kemudian diakumulasi sebagai skor pijakan sistem untuk merangking hasil prediksi kepada pengguna.

Keandalan sistem dievaluasi menggunakan metode *k-Fold Cross Validation* ($k=5$) guna menguji ketahanan model pada sampel data yang bervariasi. Evaluasi ini meninjau kinerja arsitektur tunggal (CBF dan CF murni) dengan model *Hybrid* (40% CBF / 60% CF) menggunakan tiga metrik pengujian: *Root Mean Squared Error* (RMSE), *Hit Rate @10*, dan *Mean Average Precision* (MAP).

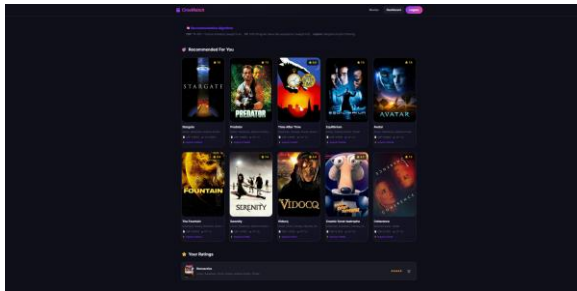


Gambar 11. Perbandingan Performa Tiap Metode

Perbandingan performa pada gambar 11 diatas membuktikan bahwa integrasi *Hybrid* sanggup menangani celah kelemahan algoritma konvensional

dengan sangat baik. Model *Hybrid* mencetak nilai kesalahan terendah dengan rata-rata RMSE sebesar 0.731, lebih akurat ketimbang performa murni model CBF (0.910) dan CF (0.812). Lebih lanjut, dominasi bobot CF sebesar 60% yang disisipi kepekaan 40% CBF sukses mendongkrak utilitas *ranking* ke angka signifikan, yakni tercatatnya akurasi penemuan item relevan sebesar 71.6% dan kualitas urutan persisi daftar usulan pada angka 0.268 pada MAP, jauh mengatasi kelemahan pada skenario isolasi algoritma murni.

Seluruh arsitektur pemodelan *Hybrid Filtering* yang telah dievaluasi kemudian diintegrasikan ke dalam lingkungan produksi berbasis web interaktif. Sistem dibangun menggunakan arsitektur *Model-View-Template* (MVT) pada *framework Django (Python)*, dengan *SQLite* sebagai basis data manajemen riwayat operasional dan penataan antarmuka (*user interface*) yang dikembangkan berlandaskan HTML murni beserta modifikasi kustom CSS.



Gambar 12. Halaman Antarmuka Dashboard Pengguna

Aplikasi web CineMatch dirancang dengan mengedepankan transparansi algoritma. Pada bagian atas dashboard, sistem secara jelas menampilkan informasi parameter bobot yang digunakan (TF-IDF 0.4 dan SVD 0.6). Pada segmen *Recommended For You*, setiap kartu film menyajikan rincian perolehan komponen skor prediksinya (skor CBF, skor CF, dan nilai akhir *Hybrid*). Sementara itu, segmen *Your Ratings* di bagian bawah menampilkan riwayat pemeringkatan oleh pengguna secara *real-time*, yang bertindak sebagai masukan utama untuk memperbarui matriks *Collaborative Filtering* (SVD) di setiap sesi. Selain itu, sistem ini telah dilengkapi penanganan untuk pengguna baru tanpa riwayat rating (*Cold-Start Problem*). Karena model SVD tidak dapat bekerja tanpa data interaksi, sistem menerapkan *Fallback Strategy*. Model *hybrid* akan ditangguhkan sementara dan dialihkan menggunakan model CBF murni yang merekomendasikan film-film terpopuler. Begitu pengguna awam tersebut memberikan minimal satu interaksi rating, komputasi *Hybrid Weighted* (40/60) akan langsung mengambil alih untuk menghasilkan rekomendasi *real-time* yang terpersonalisasi.

Secara keseluruhan, hasil eksperimentasi arsitektur sistem rekomendasi hibrida berbasis *Weighted Hybrid Filtering* ini mengonfirmasi

sekaligus memperluas literatur terkait efektivitas penyatuan logika CBF dan CF. Temuan utama dari uji coba *k-Fold Cross Validation* secara empiris menunjukkan bahwa pembobotan modifikatif sebesar 40% CBF dan 60% CF sanggup menekan margin kesalahan prediksi dengan nilai metrik *Root Mean Squared Error* (RMSE) terendah di rentang 0.731, serta memacu efikasi penyortiran daftar rekomendasi pada *Hit Rate@10* dan *Mean Average Precision* (MAP) secara kualitatif di atas kemampuan algoritma tunggal.

Hasil evaluasi yang memvalidasi keunggulan model *Hybrid* atas model *Single* (CBF maupun CF secara terisolasi) ini sangat sejalan dengan penemuan riset terdahulu oleh Rafael dan Hasrullah [7] mengenai superioritas filtrasi gabungan pada ekologi rekam jejak pengguna yang heterogen. Meski demikian, penelitian ini mengajukan modifikasi spesifik melalui kalibrasi asimetris. Porsi dominan 60% untuk model SVD (*Collaborative*) ditujukan untuk mengakomodasi *information overload* pada film populer, sebagaimana isu yang disoroti oleh [2] dan [1] dengan mempertahankan perilaku interaksi komunal. Di saat bersamaan, sisipan persentase CBF (40%) berfungsi layaknya tuas penstabil (*stabilizer*) ketika sistem terkena isu *cold-start*. Lebih lanjut, keberhasilan integrasi arsitektur permesinan algoritma ini ke dalam antarmuka aplikasi *real-time* sangat didukung oleh keandalan *framework* Django. Penggunaan Django yang memfasilitasi arsitektur *Model-Template-View* secara terstruktur dan terukur di lingkungan produksi, turut mendukung dan mengonfirmasi pernyataan teori fundamental terkait arsitektur web yang dikemukakan oleh [15].

Meskipun hasil komputasi *hybrid* membanggakan, analisis mendalam mengungkap celah operasional pada sistem saat diaplikasikan murni di level *backend* web Django. Pembaruan rekomendasi secara dinamis masih bergantung pada kalkulasi ulang matriks laten SVD secara langsung tiap kali pengguna memberikan interaksi (*rating*). Mengingat dimensi matriks interaksi ulasan yang memiliki tingkat *sparsity* hingga 98,29%, pemrosesan perhitungan SVD akan semakin memakan memori dan komputasi server seiring dengan bertambahnya pengguna. Hal ini berpotensi memengaruhi *overhead* atau latensi pemuatan halaman Dashboard apabila sistem diakses secara bersamaan (konkurensi tinggi).

5. KESIMPULAN DAN SARAN

Penelitian ini membuktikan efektivitas arsitektur *Hybrid Weighted Filtering* sebagai solusi atas kendala utama sistem rekomendasi tunggal pada platform distribusi film digital. Integrasi algoritma *Content-Based* (berbasis TF-IDF dan *Cosine Similarity*) dan *Collaborative* (berbasis *Truncated SVD*) melalui kalibrasi bobot asimetris 40:60 secara empiris terbukti menghasilkan probabilitas prediksi yang superior dengan menekan tingkat kesalahan (*error*) dan meningkatkan metrik presisi pemeringkatan.

Penerapan *Fallback Strategy* pada sistem juga tervalidasi berhasil mengatasi anomali *cold-start* secara fungsional di lingkungan aplikasi web nyata berbasis Django. Temuan ini berkontribusi dalam mengatasi kendala komputasi pada *dataset* dengan *sparsity* ekstrem, sehingga sistem mampu memberikan rekomendasi secara mandiri dan *real-time*. Ke depannya, disarankan agar operasional komputasi pelatihan matriks laten ditransisikan dari paradigma *synchronous* ke *asynchronous* menggunakan teknologi *message-broker*. Selain itu, penelitian mendatang diharapkan dapat mengganti rasio bobot statis dengan model *Hybrid Adaptive Weighting* berbasis *Machine/Deep Learning* untuk membangun sistem personalisasi yang dinamis sesuai dengan tingkat interaksi masing-masing pengguna.

DAFTAR PUSTAKA

- [1] M. A. Fikry, S. R. Wardhana, and R. K. Hapsari, "Sistem Rekomendasi Film dengan Menggunakan Sentiment Analysis dan Collaborative Filtering," *J. Ris. Inov. Bid. Inform. dan Pendidik. Inform.*, vol. 5, no. 2, 2024, doi: 10.31284/j.kernel.2024.v5i2.7315.
- [2] M. Fajriansyah, P. P. Adikara, and A. W. Widodo, "Sistem Rekomendasi Film Menggunakan ContentBasedFiltering," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 6, pp. 2188–2199, 2021, [Online]. Available: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/9163/4159>
- [3] A. Rochmad Wahono, B. Aji Saputra, and F. Fadlu Rahman, "Sistem Rekomendasi Film Menggunakan Metode Content-Based Filtering dan Algoritma K-Nearest Neighbors (KNN)," in *Prosiding Seminar Nasional Teknologi Informasi dan Bisnis (SENATIB)*, Surakarta, Indonesia: SENATIB, 2024, pp. 1–6. doi: 10.47701/senatib.v4i1.3994.
- [4] J. Lemantara, "Sistem Pendukung Keputusan Penentuan Film Berdasarkan Minat dengan Metode Weighted Product," *J. Ris. Sist. Inf. Dan Tek. Inform.*, vol. 8, no. 2, pp. 587–600, 2023, [Online]. Available: <https://tunasbangsa.ac.id/ejurnal/index.php/jurasisistemPendukungKeputusanPenentuanFilmdenganWP>
- [5] R. A. Sari, S. F. Isnaini, and E. Seniwati, "Sistem Rekomendasi Film Menggunakan Content Based Filtering," *Indones. J. Comput. Sci. Res.*, vol. 4, no. 1, pp. 19–27, 2025, doi: 10.37905.
- [6] H. H. Arfisko and A. T. Wibowo, "Sistem Rekomendasi Film Menggunakan Metode Hybrid Collaborative Filtering dan Content-Based Filtering," *e-Proceeding Eng.*, vol. 9, no. 3, pp. 2149–2159, 2022.
- [7] R. V. Pratama and Hasrullah, "Pengembangan Sistem Rekomendasi Buku Untuk Meningkatkan Minat Baca Dengan Pendekatan Hybrid Filtering," *J. Inov. Glob.*, vol. 3, no. 1, pp. 2182–2191, 2025, [Online]. Available: <https://jig.rivierapublishing.id/index.php/rv/article/view/255/438>
- [8] M. R. Sitorus, "Implementasi Model-Based Collaborative Filtering Pada Sistem Rekomendasi Film Menggunakan Algoritma SVD (Singular Value Decomposition)," Skripsi, Teknik Informatika, Universitas Medan Area, Medan, Indonesia, 2023.
- [9] K. T. Mukti and I. Mardhiyah, "Sistem Rekomendasi Pembelian Lisensi Film Menggunakan Pendekatan Hybrid Filtering (Studi Kasus: Film Animasi Jepang)," *JURSISTEKNI (Jurnal Sist. Inf. dan Teknol. Informasi)*, vol. 4, no. 3, pp. 127–139, 2022.
- [10] N. Valentino and E. B. Setiawan, "Movie recommender system on twitter using weighted hybrid filtering and GRU," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 9, no. 2, pp. 159–172, 2024, [Online]. Available: <https://kinetik.umm.ac.id/index.php/kinetik/article/view/1941>
- [11] O. D. Rossiiev, N. N. Shapovalova, O. H. Rybalchenko, and A. M. Striuk, "A comprehensive survey on reinforcement learning-based recommender systems: State-of-the-art, challenges, and future perspectives," *CEUR Workshop Proc.*, vol. 3917, no. 62, pp. 428–440, 2025, [Online]. Available: <https://ceur-ws.org/Vol-3917/paper62.pdf>
- [12] C. C. Aggarwal, *Recommender Systems*, 1st ed. Yorktown Heights, New York: Springer International Publishing, 2016. doi: 10.1007/978-3-319-29659-3.
- [13] R. C. Rivaldi and T. D. Wismarini, "Analisis Sentimen Pada Ulasan Produk Dengan Metode Natural Language Processing (NLP) (Studi Kasus Zalika Store 88 Shopee)," *J. Ilm. Elektron. DAN Komput.*, vol. 17, no. 1, pp. 120–128, 2024, [Online]. Available: <https://doi.org/10.51903/elkom.v17i1.1680>
- [14] A. P. Ariani, D. A. Handayani, P. M. Setyowati, and D. K. Nugroho, "Evaluasi Komparatif Hybrid Filtering dan Model-Based SVD pada Sistem Rekomendasi Film Menggunakan Dataset MovieLens," *J. Pustaka Data*, vol. 6, no. 1, pp. 22–31, 2026, [Online]. Available: <https://doi.org/10.55382/jurnalpustakadata.v6i1.1723%0A>
- [15] W. S. Vincent, *Django for Beginners*, 5th ed. leanpub.com, 2025. [Online]. Available: <https://leanpub.com/djangoforbeginners>