FunkSVD untuk Sistem Rekomendasi Film yang Efisien dan Optimal Menggunakan Dataset IMDb

1st David Dharmawan Saputra
Faculty of Informatics
Telkom University
Bandung, Indonesia
davidharmawan@student.telkomuniversity.ac.id

2nd Aji Gautama Putrada
Faculty of Informatics
Telkom University
Bandung, Indonesia
ajigps@telkomuniversity.ac.id

3rd Endro Ariyanto
Faculty of Informatics
Telkom University
Bandung, Indonesia
endroa@telkomuniversity.ac.id

***Abstrak* — Pada era saat ini, banyak sekali video konten kreator di luar sana yang disediakan oleh platform video seperti YouTube, yang memiliki jutaan video di dalamnya. Platform video tersebut memberikan videonya kepada para penggunanya dengan cara menggunakan sistem rekomendasi, dimana konten-konten video tersebut diberikan kepada para penggunanya berdasarkan preferensi/selera para penggunanya. Cara kerja sistem rekomendasi adalah dengan mengumpulkan data dari masa lalu yang sesuai dengan preferensi penggunanya, lalu akan dianalisis untuk mengidentifikasi pola dalam preferensi penggunanya. Namun, sistem rekomendasi tidak selalu memberikan hasil yang bagus dalam memberikan hasil rekomendasinya kepada target penggunanya. Selain itu untuk mendapatkan hasil yang optimal dibutuhkan waktu dalam menjalankan program prediksinya. Oleh karena itu, penelitian ini bertujuan untuk mencari metode mana yang memberikan hasil prediksi terbaik dan memiliki waktu paling singkat dalam menjalankan program prediksi pada sistem rekomendasi film. Pada penelitian ini digunakan metode FunkSVD dan user/item-based K-Nearest-Neighbor (KNN) untuk perbandingannya. Hasilnya menunjukkan, FunkSVD memberikan performa yang lebih baik, dengan nilai RMSE = 0,86, nilai MAE = 0,65, nilai Precision@k = 0,67, dan nilai NDGC@k = 0,81. Untuk waktu tercepat dalam menjalankan program prediksinya dipegang oleh FunkSVD, dengan waktu 0,10 detik dalam menjalankan programnya.**

***Kata kunci*— sistem rekomendasi, FunkSVD, collaborative filtering, film**

I. PENDAHULUAN

Pada era saat ini, berbagai platform video dan media sosial seperti YouTube, Twitch, Instagram, dan TikTok yang menyediakan beragam pilihan konten video pada platform tersebut. Jumlah konten video pada setiap platform tersebut tidak hanya ribuan konten video, melainkan mencapai jutaan. Namun, konten video yang ditampilkan oleh platform tersebut berbeda-beda kepada setiap pengguna. Perbedaan ini menimbulkan pertanyaan mengenai bagaimana platform tersebut menyajikan konten video kepada pengguna. Jika konten video diberikan secara acak, maka kemungkinan besar konten tersebut tidak relevan bagi pengguna. Namun, konten video yang disajikan dinilai relevan dengan preferensi pengguna.

Pada platform Netflix, sistem yang sekilas "acak" tersebut dapat membantu pengguna dalam memilih film mana yang akan ditonton. Menurut Gomez-Urbe dan Hunt [1], sekitar 80% dari total jam yang ditonton pengguna berasal dari video yang diberikan Netflix pada halaman utama, dan sekitar 20% berasal dari pengguna yang mencari sendiri film yang akan ditonton. Hal ini menunjukkan bahwa konten yang diberikan Netflix memiliki tingkat relevansi yang tinggi terhadap preferensi pengguna.

Hal tersebut menunjukkan bahwa penyajian film pada platform Netflix tidak dilakukan secara acak, melainkan dengan bantuan sistem rekomendasi yang dibangun oleh Netflix. Menurut Gomez-Urbe dan Hunt [1], manusia sangat buruk dalam memilih sesuatu dari banyaknya pilihan yang ditawarkan, cepat merasa kewalahan atau membuat pilihan yang buruk.

Menurut Falk [2], sistem rekomendasi adalah sistem yang akan menampilkan atau menawarkan suatu konten video, film, dan lain-lain ke user sesuai dengan yang disukai dari user tersebut. Cara kerja sistem rekomendasi adalah dengan mengambil data apa saja yang disukai oleh user, lalu data tersebut akan dianalisis olehnya, sehingga didapatkan pola kesukaan dari suatu user tersebut. Misalkan user A melihat suatu konten video A dan B, lalu user A menyukai video B tersebut maka sistem rekomendasi akan memberikan video lainnya yang mirip dengan video B ke user A.

Tidak hanya pada platform film, melainkan pada platform e-commerce pun sistem rekomendasi sangat membantu untuk meningkatkan penjualan dari platform e-commerce tersebut. Pada awalnya, sistem rekomendasi dibangun pada tahun 1996 oleh suatu perusahaan yang menjual sistemnya kepada perusahaan Amazon, Best Buy dan JC Penney. Semenjak hal ini, penelitian akademis dan *industrial practical applications* menjadi pilar dari perkembangan di dunia Sistem Rekomendasi [3].

Namun, sistem rekomendasi tidak selalu mampu memberikan hasil yang benar-benar tepat sasaran kepada usernya, misalkan konten yang ditampilkan tidak sesuai dengan minat dari suatu user. Selain itu, proses evaluasi

terhadap kualitas rekomendasi tidak dapat dilakukan dalam waktu singkat, melainkan membutuhkan waktu untuk mendapatkan hasil pengujian yang benar-benar memberikan hasil yang optimal.

Oleh karena itu, penelitian ini difokuskan pada analisis dari model *collaborative filtering* dengan menggunakan dua metode populer dalam sistem rekomendasi, yaitu FunkSVD (*matrix factorization*) dan KNN (*neighborhood*).

II. KAJIAN TEORI

A. Penelitian Terdahulu

Pada penelitian [5], dibandingkan beberapa metode untuk mendapatkan metode mana yang unggul. Metode seperti SVD, CoClustering, NMF, SlopeOne, KNNBaseline, dan BaselineOnly akan dibandingkan satu sama lainnya. Didapatkan bahwa Metode SVD memberikan performa terbaik dengan RMSE = 0.863, MAE = 0.668, Precision = 0.827, Recall = 0.394, dan F1 = 0.483 dibandingkan dengan metode lainnya. Sebaliknya Metode CoClustering dan SlopeOne mendapatkan akurasi yang rendah serta waktu komputasi yang lebih lama. Penelitian ini tidak hanya menekankan pada akurasi prediksi rating, tetapi pada kualitas metrik seperti Precision dan F1 Score. Hal ini menjadi landasan penting untuk evaluasi model, tidak hanya terpaku pada akurasi prediksi rating tetapi juga pada kualitas metrik relevansi.

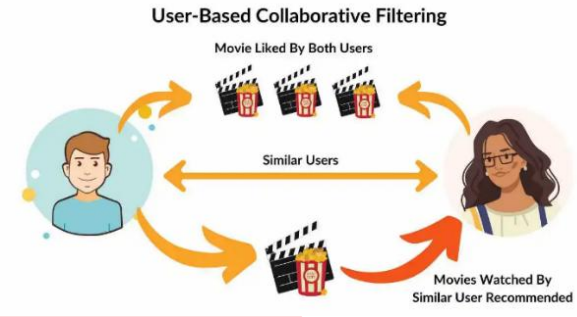
Pada penelitian [6], dibandingkan metode KNN, SVD, FunkSVD, dan SVD++ untuk mencari metode mana yang bagus. Didapatkan bahwa metode SVD++ mendapatkan nilai RMSE = 0.920 dan MAE = 0.721 yang dimana metode ini memberikan nilai terbaik dibandingkan dengan metode lainnya dan metode dengan performa terbawah adalah KNN dengan RMSE = 0.979 dan MAE = 0.773. Pada penelitian ini, walaupun KNN mudah diimplementasikan namun FunkSVD memiliki keunggulan pada prediksi yang lebih stabil dan akurat pada dataset yang banyak, hal ini menjadi landasan utama untuk penggunaan metode FunkSVD.

Pada penelitian [7], dibandingkan metode ContentBased (KNN), Collaborative-Based, SVD, SVD++, Hybrid Model untuk mencari model mana yang bagus. Hasilnya, didapatkan bahwa Hybrid model mendapatkan nilai paling bagus dari yang lainnya dengan nilai RMSE = 0.76, dan model SVD++ mendapatkan nilai paling terendah dari yang lainnya dengan nilai RMSE = 0.938. Penelitian ini menekankan pada akurasi prediksi rating menggunakan dekomposisi matrik. Hal ini menjadi landasan penting untuk evaluasi model yang andal dalam menghasilkan nilai error yang rendah.

Pada penelitian [8], dibandingkan metode KNN, SVD, SVD++, dan CoClustering untuk mencari metode mana yang bagus. Hasil dari perbandingan tersebut, KNN memiliki nilai error tertinggi dengan RMSE = 0.979 dan MAE = 0.773 dari metode lainnya sedangkan SVD++ mendapatkan error terendah dengan RMSE = 0.920 dan MAE = 0.721. Penelitian ini menjadi landasan teori untuk *collaborative filtering* menjadi pendekatan yang paling efektif untuk domain rekomendasi film.

Pada penelitian [9], dibandingkan metode ContentBased, KNN item, KNN user, Alternative Least Squares, Funk SVD, Implicit Matrix Factorization, Hierarchical Poisson Factorization, Popular, Bias. Didapatkan bahwa metode KNN Item mendapatkan RMSE dan MAE yang terendah dari metode lainnya, dengan RMSE = 0.896 dan MAE = 0.703, sedangkan yang paling tertinggi adalah metode Bias, dengan RMSE = 0.934 dan MAE = 0.738. Penelitian ini menekankan pada evaluasi berbasis *Top-N recommendation* seperti Precision@k dan nDGC, hal ini menjadi landasan untuk menilai kualitas hasil rekomendasi akhir.

B. *Collaborative Filtering*

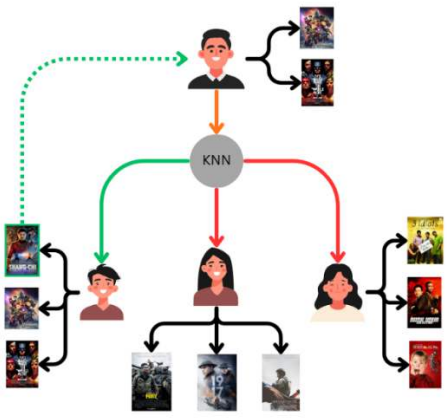


GAMBAR 1
(B)

Collaborative filtering merekomendasikan item berdasarkan kesamaan antar pengguna dan/atau item, istilah *collaborative* merujuk antar pengguna untuk menyaring data dan menghasilkan rekomendasi. Misalkan, jika ada pengguna a dan b memiliki kesamaan penilaian ke item di masa lalu, maka riwayat preferensi mereka akan mempengaruhi hasil dari rekomendasi mereka masing-masing [10], seperti pada gambar 1 (B).

C. *K-Nearest Neigbor*

Secara garis besar, KNN bertujuan untuk mengklasifikasikan objek tidak berlabel dengan cara mengelompokkan ke dalam kelas berlabel yang paling mirip. Contohnya pada klasifikasi buah-buahan, sayuran, dan biji-bijiannya yang dapat dibedakan berdasarkan kekenyalan dan kemanisannya. Secara umum, buah akan lebih manis daripada sayuran, dan biji-bijian tidak mempunyai kerenyahan ataupun kemanisan. Dalam contoh ini, akan menentukan kategori dari ubi jalar yang dimana ubi jalar memiliki suara terbanyak ke sayuran. Maka dari itu, ubi jalar diklasifikasikan ke dalam kelas sayuran [14].



GAMBAR 2
(C)

Pada dasarnya, *Memory-Based* memiliki kemiripan dengan algoritma *K-Nearest Neighbor* (KNN). Pendekatan dari *Memory-Based* yaitu dihitungnya seluruh data interaksi pengguna-item untuk mendapatkan kemiripan antar entitas. Hal ini dapat diimplementasikan pada algoritma KNN, KNN akan menemukan K pengguna yang paling mirip dengan pengguna target berdasarkan preferensi mereka di masa lalu, lalu merekomendasikan item yang juga disukai oleh pengguna yang mirip [11], seperti pada gambar 2 (C).

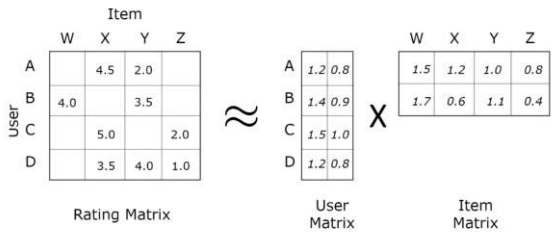
Prediksi dari KNN untuk peringkat pengguna A pada item B direpresentasikan rumus seperti berikut:

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in U_{ui}^K} sim(u,v)(r_{vi} - \bar{r}_v)}{\sum_{v \in U_{ui}^K} |sim(u,v)|} \quad (1)$$

Yang dimana U_{ui}^K mewakili himpunan *K-nearest neighbors* dari pengguna u yang memberikan rating ke item i, r_{vi}, yang dimana item i, r_{vi} adalah peringkat aktual dari yang diberikan *neighbor* v untuk item i, $\bar{r}_u$ dan $\bar{r}_v$ adalah rata-rata rating dari setiap pengguna yang dihitung berdasarkan riwayat rating mereka. *Similarity* sim(u, v) antar pengguna dihitung menggunakan *distance metric*, seperti *cosine similarity* atau *pearson correlation coefficient* [11].

D. FunkSVD

Berdasarkan pada paper [12], Funk SVD dibuat oleh Simon Funk dalam kompetisi yang diselenggarakan oleh Netflix yaitu, Netflix Prize Competition pada tahun 2006. Metode ini tidak didasarkan pada *heuristic similarity* tetapi menerapkan proses *explicit learning*, Funk SVD juga dapat disebut sebagai *latent factor model* (LFM). Berikut adalah cara kerja dari Funk SVD:



GAMBAR 3
(D)

Berdasarkan contoh gambar di atas, *rating matrix* $R \in \mathbb{R}^{4\times4}$ memiliki 4 pengguna dan 4 item, untuk mewakili *rating matrix* ini digunakan perkalian hasil dari *user matrix* $P^T \in \mathbb{R}^{4\times2}$ dengan *item matrix* $Q \in \mathbb{R}^{2\times4}$. Untuk nilai yang sudah ada $R_{AY} = 2.0$ hampir di representasi oleh $P_A^T \times Q_Y = 1.2 * 1.0 + 0.8 * 1.1 = 2.08$, dan setelah mendapatkan kedua nilai tersebut selanjutnya dapat bisa melakukan prediksi nilai hilang pada *rating matrix* dari hasil perkalian P dan Q. Untuk memprediksikan nilai hilang pada R_{AZ} dapat diperoleh dari hasil perkalian $P_A^T \times Q_Z = 1.2 * 0.8 + 0.8 * 0.4 = 1.28$. Rumus dari Funk SVD adalah sebagai berikut:

$$\begin{aligned} \hat{R} &= P^T Q \\ \hat{R}(u,i) &= \hat{r}_{ui} \\ \hat{r}_{ui} &= \sum_f p_{uf}^T q_{if} \end{aligned} \quad (2)$$

Di mana $\hat{r}_{ui}$ merupakan nilai rating prediksi yang didapatkan dari perhitungan menggunakan matriks faktor pengguna (P) dan matriks faktor item (Q). Dengan begitu, matriks rating dari $\hat{R}$ dapat mengandung banyak nilai yang kosong, dan untuk mempresentasikannya digunakan perkalian antar P dan Q. Melalui perkalian kedua matriks tersebut, nilai-nilai yang kosong dapat diprediksi. Melalui rumus (2), dapat ditunjukkan bagaimana cara proses dari dekomposisi dan prediksi bekerja.

III. METODE

A. Dataset

Pada penelitian ini, digunakan dataset "IMDB movies dataset to use as recommender system" yang diambil dari website Kaggle. Dataset tersebut memiliki 2 file yaitu movies.csv dan ratings.csv, pada penelitian ini digunakan hanya file rating.csv. Untuk mengetahui karakteristik dataset rating.csv, berikut adalah tabel statistik dari dataset tersebut:

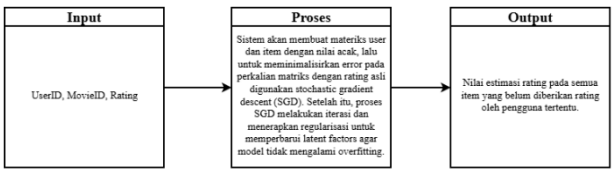
TABEL 1
(A)

Atribut	Nilai Statistik
Total data	100836
Total pengguna unik	610
Total film unik	9724
Rating minimal	0.5
Rating maksimal	5
Rata-rata rating	3.5
Sparsity	98.30%
Density	1.70%

Sebelum melakukan proses pelatihan model, tahapan preprocessing pada dataset dilakukan. Hal ini berguna untuk memastikan kualitas pada datasetnya. Pada tahapan preprocessing dilakukannya penghapusan nilai yang kosong dan penghapusan nilai yang duplikasi. Dataset akan dibagi menjadi 2 bagian untuk dijadikan sebagai data latih dan data uji, data latih memiliki sebesar 80% dan data uji sebesar 20% dari dataset.

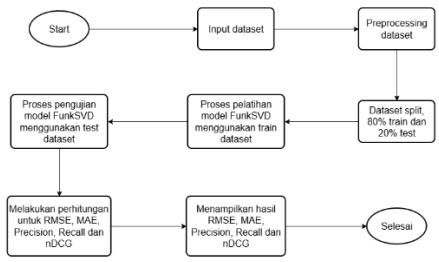
B. FunkSVD

Cara kerja dari FunkSVD adalah dengan merepresentasikan pengguna dan item ke dalam ruang laten, sehingga didapatkan pola-pola tersembunyi dari preferensi pengguna yang dapat dianalisis menggunakan nilai rating yang tersedia. Berikut adalah diagram sistem dari FunkSVD:



GAMBAR 4
(B)

Untuk mendapatkan gambaran yang jelas mengenai tahapan implementasi sistem rekomendasi dengan model FunkSVD dari mulai input dataset sampai evaluasi model, dibuatkan flowchart untuk menggambarkan alur sistem FunkSVD. Berikut adalah flowchart dari FunkSVD:



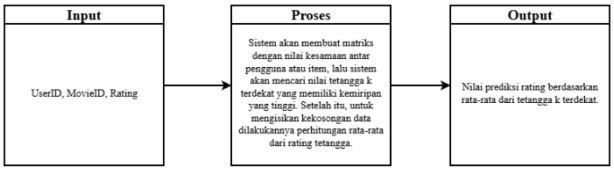
GAMBAR 5
(B)

Pada gambar 5 (B), proses dimulai dari input dataset lalu dilakukannya tahapan preprocessing dataset. Pada tahapan ini, dilakukannya penghapusan data yang memiliki data kosong dan data duplikat. Setelah preprocessing, dataset akan dipecah menjadi dua data yaitu, data latih sebesar 80% dan data uji sebesar 20% dari total dataset.

Selanjutnya, proses pelatihan model FunkSVD menggunakan data latih dijalankan dengan tujuan untuk mempelajari faktor laten pengguna dan film. Setelah model terbentuk, dijalankan proses pengujian model FunkSVD menggunakan data uji dengan tujuan untuk mendapatkan nilai estimasi rating. Setelah mendapatkan hasil estimasi rating, dilakukan perhitungan untuk menghitung hasil evaluasi terhadap metode FunkSVD, lalu ditampilkan hasil evaluasi tersebut.

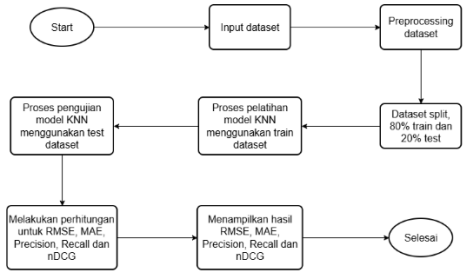
C. KNN

Cara kerja dari KNN adalah dengan menghitung kemiripan antar pengguna atau item untuk menemukan tetangga terdekat, sehingga didapatkan hasil rekomendasi berdasarkan preferensi pengguna atau karakteristik item. Berikut adalah diagram sistem dari KNN:



GAMBAR 6
(C)

Untuk mendapatkan gambaran yang jelas mengenai tahapan implementasi sistem rekomendasi dengan model KNN dari mulai input dataset sampai evaluasi model, dibuatkan flowchart untuk menggambarkan alur sistem KNN. Berikut adalah flowchart dari KNN:



GAMBAR 7
(C)

Pada gambar 7 (C), proses dimulai dari input dataset lalu dilakukannya tahapan preprocessing dataset. Pada tahapan ini, dilakukannya penghapusan data yang memiliki data kosong dan data duplikat. Setelah preprocessing, dataset akan dipecah menjadi dua data yaitu, data latih sebesar 80% dan data uji sebesar 20% dari total dataset.

Selanjutnya, proses pelatihan model KNN menggunakan data latih dijalankan dengan tujuan untuk menghitung bobot kemiripan antar entitas. Setelah model terbentuk, dijalankan proses pengujian model KNN menggunakan data uji dengan tujuan untuk mendapatkan nilai prediksi rating. Setelah mendapatkan hasil prediksi rating, dilakukan perhitungan untuk menghitung hasil evaluasi terhadap metode KNN, lalu ditampilkan hasil evaluasi tersebut.

Pada penelitian ini metode KNN dipecah menjadi 3 metode: KNN Basic, KNN mean, dan KNN Baseline. Perbedaannya adalah KNN Basic hanya menghitung prediksi berdasarkan weighted average dari tetangga terdekat tanpa menghitung bias pengguna ataupun item, KNN Mean mengatasi bias pengguna dengan meancentering, dan KNN Baseline mengatasi bias menggunakan global mean, user bias, dan item bias sehingga metode dapat memberikan akurasi yang lebih baik. Pada penelitian ini juga metode KNN menggunakan prediksi dari user-based dan item-based.

D. Skenario Pengujian

Untuk mendapatkan nilai performa dari metode FunkSVD dan KNN, dibutuhkannya penilaian matriks evaluasi untuk mengukur hasil performa antar metode. Penilaian matriks evaluasi pada penelitian ini menggunakan 2 jenis, yaitu *error-based evaluation metrics* dan *ranking-based evaluation metrics*.

Error-Based Evaluation Metrics digunakan untuk mengukur seberapa dekat nilai rating yang telah diprediksikan oleh FunkSVD dan KNN dengan nilai rating asli yang pengguna berikan. Pada pendekatan ini dibagi menjadi dua bagian, yaitu Root Mean Squared Error (RMSE) dan Mean Absolute Error (MAE). RMSE digunakan untuk mengukur selisih antara nilai prediksi model dan nilai aktual yang diberikan oleh pengguna, hal ini berguna untuk melihat seberapa besar penyimpangan nilai model prediksi berikan. MAE digunakan untuk menghitung rata-rata kesalahan dalam prediksi model, hal ini berguna untuk melihat akurasi dari prediksi model secara keseluruhan.

Ranking-Based Evaluation Metrics digunakan untuk menilai seberapa relevan hasil rekomendasi dari FunkSVD dan KNN dalam memberikan item ke pengguna. Sistem akan mengevaluasikan 10 item teratas dengan menetapkan ambang atasnya adalah ≥ 4, artinya item akan dianggap relevan jika memiliki nilai rating ≥ 4. Pada pendekatan ini dibagi menjadi tiga bagian, yaitu *Precision@10*, *Recall@10*, dan *nDCG@10*. *Precision@10* digunakan untuk menilai seberapa relevan item yang diberikan oleh model prediksi diantara 10 item teratas. *Recall@10* digunakan untuk menilai kemampuan sistem dalam menemukan kembali item yang relevan dari seluruh kumpulan item relevan lainnya ke dalam 10 rekomendasi utama. *nDCG@10* digunakan untuk menilai evaluasi kualitas urutan dari 10 rekomendasi yang diberikan, berdasarkan kesesuaian peringkat item yang ideal.

Tujuan dari pengujian ini adalah untuk melihat metode mana yang dapat memberikan hasil rekomendasi dengan tingkat kesalahan yang kecil, relevan bagi pengguna, dan

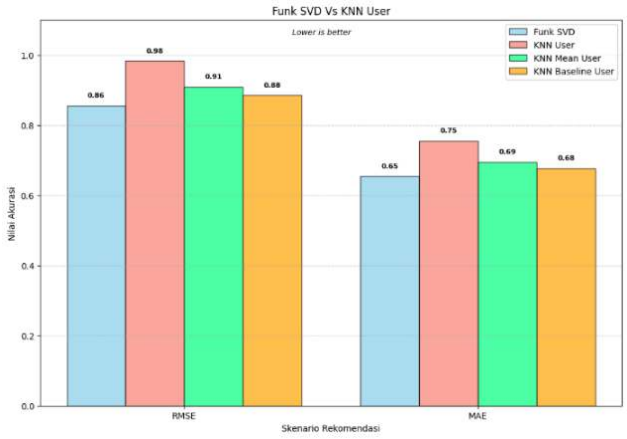
cepat dalam menjalankan program prediksinya. Oleh karena itu, output yang diharapkan tidak hanya memfokuskan pada akurasi model, tetapi juga pada seberapa relevan hasil rekomendasi item yang diberikan dan kecepatan dalam menjalankan program prediksinya.

IV. HASIL DAN PEMBAHASAN

Berikut adalah hasil pengujian dari perbandingan Funk SVD dengan KNN item serta KNN user:

A. Perbandingan FunkSVD dan KNN User-Based

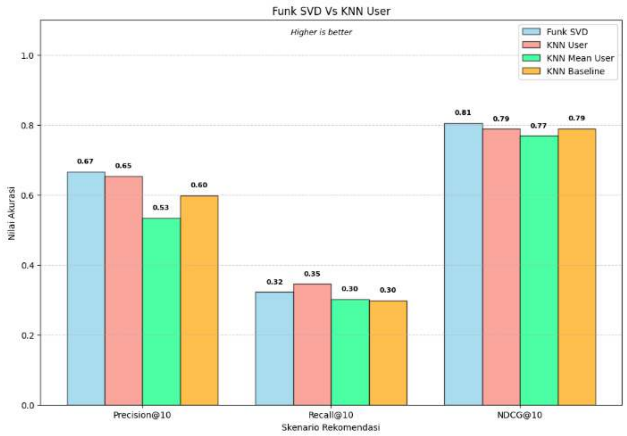
Berikut adalah perbandingan antar metode menggunakan *error-based evaluation metrics*:



GAMBAR 8
(A)

Pada gambar di atas, didapatkan bahwa nilai RMSE dan MAE yang paling bagus adalah metode Funk SVD, dengan RMSE = 0.86 dan MAE = 0.65. Metode yang paling bawah adalah metode KNN User, dengan RMSE = 0.98 dan MAE = 0.75.

Berikut adalah perbandingan antar metode menggunakan *ranking-based evaluation metrics*:



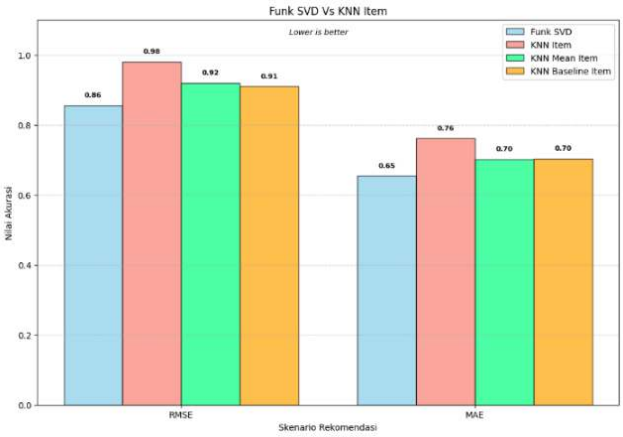
GAMBAR 9
(A)

Pada gambar di atas, di dapatkan bahwa nilai *Precision@10* dan *nDCG@10* yang paling bagus terdapat pada metode Funk SVD, dengan nilai *Precision@10* = 0.67 dan *nDCG@10* = 0.81 dan untuk *Recall@10* didapatkan yang paling bagus pada metode KNN User, dengan nilai *Recall@10* = 0.35.

Jadi secara keseluruhan, hasil pada gambar-gambar tersebut membuktikan bahwa Funk SVD lebih unggul dibandingkan dengan User Based Menggunakan KNN dalam merekomendasikan film, baik dalam aspek akurasi prediksi rating maupun kualitas dalam relevansi rekomendasi walaupun adanya kelemahan pada aspek *Recall@10*.

B. Perbandingan FunkSVD dan KNN User-Based

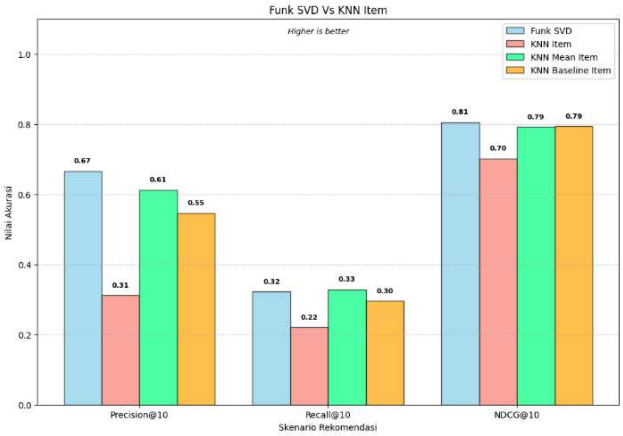
Berikut adalah perbandingan antar metode menggunakan *error-based evaluation metrics*:



GAMBAR 10
(B)

Pada gambar di atas, didapatkan bahwa nilai RMSE dan MAE yang paling bagus adalah pada metode Funk SVD, dengan RMSE = 0.86 dan MAE = 0.65 dan nilai yang paling bawah adalah pada metode KNN Item, dengan RMSE = 0.98 dan MAE = 0.76.

Berikut adalah perbandingan antar metode menggunakan *ranking-based evaluation metrics*:



GAMBAR 11
(B)

Pada gambar di atas, didapatkan bahwa nilai dari *Precision@10* dan *nDCG@10* yang paling bagus ada di metode Funk SVD, dengan nilai *Precision@10* = 0.67 dan *nDCG@10* = 0.81, dan nilai *Recall@10* paling bagus ada pada metode KNN Mean Item dengan nilai *Recall@10* = 0.33.

Jadi secara keseluruhan, hasil pada gambar-gambar tersebut membuktikan bahwa Funk SVD lebih unggul dibandingkan dengan Item Based Menggunakan KNN dalam merekomendasikan film, baik dalam aspek akurasi prediksi rating maupun kualitas dalam relevansi rekomendasi

walaupun adanya aspek kelemahan yang berbeda sedikit dengan metode KNN pada nilai *Recall@10*.

C. Waktu Eksekusi Antar Metode

Berikut adalah hasil perbandingan lamanya eksekusi waktu pada setiap metode:

TABLE 2
(C)

Metode	Eksekusi waktu ke-					Rata-rata
	1	2	3	4	5	
FunkSVD	0.09	0.07	0.06	0.07	0.06	**0.07**
KNN baseline user	0.6	0.81	0.86	0.91	0.67	0.77
KNN Mean user	0.58	0.59	0.59	0.6	0.61	0.59
KNN user	0.58	0.55	0.76	0.81	0.54	0.64
KNN baseline item	3	3.08	3.05	3	3.53	3.13
KNN Mean item	2.95	3.01	3.08	3.02	3.24	3.06
KNN item	2.73	2.9	2.94	3.01	3.29	2.97

Pada table di atas, metode yang paling cepat adalah metode Funk SVD, dengan rata-rata waktu eksekusinya adalah 0.07 detik sedangkan yang paling lama adalah metode KNN Baseline item dengan rata-rata waktu eksekusinya adalah 3.13 detik.

V. KESIMPULAN

Berdasarkan penelitian ini, didapatkan bahwa FunkSVD mampu memberikan hasil prediksi rating secara akurat dibandingkan KNN, dengan RMSE = 0.86 dan MAE = 0.65. Tidak hanya itu, melainkan juga pada kualitas peringkat rekomendasi yang diberikan oleh FunkSVD sangat relevan kepada pengguna, dengan Precision@10 = 0.67 dan nDCG@10 = 0.81, dan juga FunkSVD mampu memberikan hasil prediksinya dalam rata-rata waktu 0.07 detik. Artinya, FunkSVD mampu memberikan hasil prediksi rating yang akan diberikan ke pengguna secara akurat serta FunkSVD mampu memberikan peringkat 10 film teratasnya sangat relevan bagi pengguna, dan FunkSVD jauh lebih efisien dalam menjalankan program prediksinya.

REFERENSI

[1] C. A. Gomez-Uribe and N. Hunt, "The netflix recommender system: Algorithms, business value, and innovation," *ACM Transactions on Management Information Systems*, vol. 6, no. 4, Dec. 2015, doi: 10.1145/2843948.

[2] K. Falk, *Practical Recommender Systems*. Shelter Island, NY: Manning Publications, 2018.

[3] A. Brasoveanu, M. Moodie, and R. Agrawal, "Textual evidence for the perfunctoriness of independent medical reviews," in *CEUR Workshop Proceedings*, CEUR-WS, 2020, pp. 1–9.

[4] H. Ko, S. Lee, Y. Park, and A. Choi, "A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields," *Electronics (Switzerland)*, vol. 11, no. 1. MDPI, Jan. 01, 2022. doi: 10.3390/electronics11010141.

[5] B. Walek and O. Sládek, "Comparison of Selected Algorithms in Movie Recommender System," *Applied Sciences (Switzerland)*, vol. 15, no. 17, Sep. 2025, doi: 10.3390/app15179518.

[6] A. Chaudhari, A. Kale, and P. Lad, "Solution To Challenges In Recommendation Systems," *International Journal for Scientific Research & Technology (IJSART)*, vol. 8, no. 11, pp. 45–50, Nov. 2022, [Online]. Available: www.ijsart.com.

[7] S. Salmani and S. Kulkarni, "Hybrid Movie Recommendation System Using Machine Learning," in *Proceedings - International Conference on Communication, Information and Computing Technology, ICCICT 2021*, Institute of Electrical and Electronics Engineers Inc., 2021. doi: 10.1109/ICCICT50803.2021.9510058.

[8] T. Anwar and V. Uma, "Comparative study of recommender system approaches and movie recommendation using collaborative filtering," *International Journal of System Assurance Engineering and Management*, vol. 12, no. 3, pp. 426–436, Jun. 2021, doi: 10.1007/s13198-021-01087-x.

[9] A. Paullier and R. Sotelo, "A recommender systems' algorithm evaluation using the lenskit library and movielens databases," in *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting, BMSB*, IEEE Computer Society, Oct. 2020. doi: 10.1109/BMSB49480.2020.9379914.

[10] M. Jalili, S. Ahmadian, M. Izadi, P. Moradi, and M. Salehi, "Evaluating Collaborative Filtering Recommender Algorithms: A Survey," *IEEE Access*, vol. 6, pp. 74003–74024, 2018, doi: 10.1109/ACCESS.2018.2883742.

[11] L. V. Nguyen, Q. T. Vo, and T. H. Nguyen, "Adaptive KNN-Based Extended Collaborative Filtering Recommendation Services," *Big Data and Cognitive Computing*, vol. 7, no. 2, Jun. 2023, doi: 10.3390/bdcc7020106.

[12] Y. Zhang, "An Introduction to Matrix factorization and Factorization Machines in Recommendation System, and Beyond," *arXiv preprint arXiv:2203.11026*, vol. arXiv e-prints, no. arXiv:2203.11026, pp. 1–19, Mar. 2022, [Online]. Available: http://arxiv.org/abs/2203.11026.

[13] A. Jadon and A. Patil, "A Comprehensive Survey of Evaluation Techniques for Recommendation Systems," *arXiv preprint*, vol. arXiv:2312.16015v2, no. No. 2312.16015v2, pp. 1–25, Jan. 2024, [Online]. Available: http://arxiv.org/abs/2312.16015.

[14] Z. Zhang, "Introduction to machine learning: K-nearest neighbors," *Annals of Translational Medicine*, vol. 4, no. 11, Jun. 2016, doi: 10.21037/atm.2016.03.37.