

Sistem Pemberi Rekomendasi Artis Berdasarkan Jumlah Interaksi Menggunakan Metode *Collaborative Filtering*

1st Chara Maria Emmanuel Yuliarta

Fakultas Informatika

Universitas Telkom

Bandung, Indonesia

charajuliarta@student.telkomuniversity.ac.id

2nd Agung Toto Wibowo

Fakultas Informatika

Universitas Telkom

Bandung, Indonesia

agungtoto@telkomuniversity.ac.id

Abstrak - Sistem rekomendasi adalah suatu sistem penyaringan yang bertujuan untuk memprediksi preferensi yang diberikan oleh pengguna terhadap suatu elemen tertentu, pada penelitian ini terhadap sebuah artis. Penelitian ini memiliki tujuan untuk meningkatkan hasil performansi sistem rekomendasi artis menggunakan metode *collaborative filtering*. Metode *collaborative filtering* menggunakan informasi pengguna dan artis dalam membangun rekomendasi. Dataset yang digunakan mencakup jumlah pemutaran lagu oleh pengguna. Metode *collaborative filtering* diimplementasikan dengan melakukan perhitungan *similarity* antar pengguna dan antar artis. Perhitungan *similarity* yang digunakan, menggunakan *cosine similarity*. Setelah dilakukan perhitungan kesamaan, dilakukan perhitungan *weighted sum* dan menghasilkan prediksi. Evaluasi performansi dihitung menggunakan MAE (*Mean Absolute Error*), MSE (*Mean Squared Error*), dan RMSE (*Root Mean Squared Error*). Hasil evaluasi yang didapatkan pada penelitian ini adalah MAE 9,474, MSE 52.653,40 dan RMSE 229,208 pada perbandingan 70:30. Sedangkan pada perbandingan 75:25 menghasilkan MAE 9,902, MSE 45.914,85 dan RMSE 210,017. Pada perbandingan 80:20 hasil yang didapatkan adalah MAE 10,486, MSE 48.764,51 dan RMSE 217,416. Hasil tersebut menunjukkan bahwa, semakin besar rasio data train terhadap data test, nilai MAE, MSE dan RMSE cenderung meningkat.

Kata kunci: Sistem Rekomendasi, *Collaborative filtering*, *Cosine similarity*, MAE, MSE, RMSE.

I. PENDAHULUAN

A. Latar Belakang

Dalam era digital, industri musik telah berkembang dengan pesat dengan adanya platform musik seperti *Spotify*, *Apple Musik*, *Last.fm* dan *Deezer*. Dalam lingkungan yang kaya dengan pilihan artis, sistem rekomendasi artis dapat membantu pengguna mengelola dan menemukan artis-artis baru sesuai dengan preferensi mereka. Sistem rekomendasi dapat memanfaatkan data riwayat pendengaran pengguna untuk mengidentifikasi artis favorite mereka, kemudian menganalisis pola pendengaran dan preferensi pengguna lain yang memiliki kesukaan serupa. Dengan demikian, sistem rekomendasi dapat memberikan rekomendasi artis yang cocok dengan selera pengguna, serta menyarankan artis-artis baru yang mungkin belum diketahui oleh pengguna.

Sistem rekomendasi juga dapat mempertimbangkan faktor-faktor lain seperti genre musik, artis terkait, ulasan pengguna, popularitas lagu, dan rekomendasi berbasis konten seperti analisis fitur musik. Faktor tersebut bertujuan untuk memberikan pengalaman musik yang lebih personal dan memuaskan bagi pengguna. Dan dalam penelitian ini faktor yang dipertimbangkan adalah pengguna, artis dan jumlah pemutaran musik. Sistem rekomendasi dapat memberikan rekomendasi musik yang sesuai dengan selera pengguna saat mendengarkan musik tersebut. Entah musik tersebut memberikan ketenangan, kesenangan ataupun kesedihan, sistem rekomendasi dapat mengeluarkan rekomendasi tersebut. Karena itu, diperlukan sebuah sistem rekomendasi yang dapat membantu pengguna untuk

menemukan musik yang sesuai dengan kebutuhannya [1].

Rekomendasi musik merupakan aplikasi yang aktif dalam melakukan perkembangan dalam beberapa tahun terakhir ini. Sebagai contoh, Pandora menciptakan model bisnis lengkap berdasarkan gagasan dalam menciptakan stasiun musik yang personal. Mereka mengembangkan pendekatan yang menggabungkan teknik *Collaborative filtering* yang tradisional dengan pendekatan yang digunakan dan disebut dengan *Musik Genome Project* [2]. Selain itu, aplikasi *iTunes* dari *Apple* yang menggunakan informasi tentang perpustakaan musik pengguna untuk membuat campuran dan daftar putar yang dipersonalisasi. Belakangan ini, *Spotify* mulai terlibat dalam memberikan personalisasi rekomendasi musik dalam layanannya. Pendekatan mereka untuk rekomendasi ini sebagian besar didasarkan pada *Matrix factorization* [3]. Selanjutnya, *EchoNest* adalah startup yang terkenal karena menyediakan mesin rekomendasi musik yang menggerakkan banyak layanan berbeda sebelum masuk ke dalam kepemilikan *Spotify*. Mereka menggabungkan pendekatan berbeda termasuk *Collaborative filtering*, metadata dan analisis sinyal audio dari musik [4]. Rekomendasi musik memiliki beberapa aspek yang unik [5], seperti sifat *multi-level* dari artis, album, *track*, dan daftar lagu yang putar dan dapat direkomendasikan. Selain itu, lagu memiliki durasi yang pendek dan sering didengarkan berulang kali. Hal ini mengacu kepada pendekatan yang menarik dalam memanfaatkan data dan perilaku pengguna.

Collaborative filtering didasarkan pada prinsip bahwa orang yang memiliki pendapat serupa tentang suatu *item* di masa lalu, kemungkinan besar setuju tentang *item* lain di masa depan. Dalam domain musik, metode rekomendasi ini dapat didasarkan pada pengukuran evaluasi eksplisit, seperti peringkat yang diberikan untuk lagu dan artis tertentu, atau umpan balik implisit, melihat kebiasaan yang didengarkan pengguna, seperti total pemutaran lagu yang didengarkan. Mengingat pengguna semakin kurang aktif dan tidak ingin menghabiskan waktu secara eksplisit dalam mengevaluasi suatu *item*, metode kedua saat ini lebih banyak digunakan, meskipun hanya mempertimbangkan umpan balik positif (yaitu lagu yang didengarkan oleh pengguna). Pada dasarnya setiap sistem *collaborative filtering* memiliki matriks preferensi pengguna untuk *item*, dengan baris mewakili profil pengguna dan kolom mewakili *item*. Yang memiliki tujuan untuk menghitung kemiripan antara pengguna dan *item* serta memprediksi seberapa banyak seorang pengguna yang menikmati suatu *item*.

Penelitian ini memberikan rekomendasi berdasarkan jumlah pemutaran lagu yang diputar yang didengarkan oleh pengguna. Sehingga, diperlukan

algoritma untuk memproses tingkat popularitas musik tersebut menggunakan metode *collaborative filtering*. Sistem rekomendasi musik yang dibangun pada website *Last.fm*. Dengan pertimbangan, *Last.fm* dapat memudahkan pengguna dalam mendengarkan berbagai macam jenis artis. Jumlah pemutaran lagu kemudian dikonversi menjadi rating dengan skala satu sampai lima. Dimana jumlah pemutaran lagu yang minimum adalah 1 dan jumlah pemutaran lagu yang maksimum adalah 320.725. Dalam penelitian ini, yang menjadi fokus utama adalah mengimplementasikan model sistem rekomendasi artis menggunakan metode *Collaborative filtering* dan menggunakan dataset yang disediakan dengan memperhatikan popularitas artis dan menghasilkan performansi MAE, MSE, RMSE.

Dengan menggunakan metode *item-based collaborative filtering*, pengguna dapat dengan mudah menemukan artis dengan pengguna lain yang memiliki kesamaan preferensi. Algoritma *item-based collaborative filtering* mencari keterkaitan antara item berdasarkan jumlah pemutaran lagu untuk menghasilkan rekomendasi bagi pengguna terkait item tertentu. Agar menghasilkan rekomendasi, diperlukan sebuah perhitungan kesamaan untuk mengetahui hubungan antara *item* dengan *item* menurut jumlah pemutaran lagu yang ada.

Oleh karena itu, pernyataan masalah penelitian ini adalah bagaimana mengembangkan sistem rekomendasi artis yang efektif dan dapat mengatasi kendala perilaku pengguna yang kurang aktif, agar dapat memberikan solusi untuk meningkatkan kualitas rekomendasi artis berdasarkan jumlah interaksi dengan menggunakan metode *Collaborative Filtering*.

B. Topik dan Batasan

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, maka topik yang diangkat oleh penulis adalah tentang bagaimana menghasilkan rekomendasi artis berdasarkan metode yang dipilih oleh penulis yaitu *Collaborative filtering*. Algoritma yang digunakan adalah *Item-based collaborative filtering*. Dimana, algoritma ini mencari *item* lain yang memiliki pola preferensi yang mirip dengan *item* yang sedang dipertimbangkan. Batasan masalah pada topik ini menggunakan data dari *Information Heterogeneity and Fusion in Recommender systems (HetRec 2011)*. Berisi informasi mengenai lagu yang didengarkan dari 2000 pengguna dari sistem musik online *Last.fm*.

C. Tujuan

Mengimplementasikan algoritma *item-based* pada sistem rekomendasi artis menggunakan metode *collaborative filtering* dan dapat mengeluarkan hasil performansi akurasi berdasarkan nilai yang didapatkan dari evaluasi MAE, MSE dan RSME.

D. Organisasi Tulisan

Struktur penelitian pada jurnal ini setelah bagian pendahuluan, bagian yang dibahas selanjutnya adalah studi terkait pada bagian 2. Dilanjutkan bagian 3, menyajikan teori dan perancangan sistem yang dibangun. Kemudian bagian 4 menyajikan hasil yang didapatkan dan hasil analisis. Pada bagian akhir yaitu bagian 5 berisikan kesimpulan dari penelitian yang telah dilakukan.

II. STUDI TERKAIT

Penelitian ini dilakukan dari sejumlah referensi penelitian yang ada dengan mengaitkan metode dan topik yang dituju. Tujuan dari referensi ini adalah untuk memberikan referensi tentang keterbatasan metode yang digunakan dalam penelitian ini.

Berdasarkan penelitian [1] Ivander Yoshua, dll dengan judul “Pengimplementasian Sistem Rekomendasi Musik Dengan Metode *Collaborative filtering*”, 2021. Menunjukkan bahwa algoritma SVD++ menunjukkan kinerja optimal dibandingkan dengan metode kolaboratif filtering lainnya, karena nilai RSME dan MAE yang diperoleh oleh SVD++ adalah yang terendah dibandingkan dengan metode kolaboratif lainnya. Selain itu, algoritma KNN juga menunjukkan kinerja yang lebih unggul daripada metode faktorisasi matriks, karena tingkat presisi dan recall yang dihasilkan oleh KNN lebih tinggi dibandingkan dengan metode faktorisasi matriks.

Berdasarkan penelitian Arwin Halim, et al [6], dll dengan judul “Sistem Rekomendasi Film menggunakan *Bisecting K-Means* dan *Collaborative filtering*”, 2017. Tingkat kesalahan dalam sistem rekomendasi telah dihitung dengan menggunakan nilai rata-rata MAE yang dikombinasikan antara *Bisecting K-Means* dan kolaboratif *filtering* berbasis pengguna adalah 1.63, lebih rendah daripada nilai rata-rata MAE yang dikombinasikan antara *Bisecting K-Means* dan kolaboratif filtering berbasis item. Selain itu, distribusi nilai rating pada dataset juga memiliki pengaruh yang signifikan terhadap nilai MAE. Ini juga terlihat pada cluster 11 dan 17, di mana distribusi nilai rating tidak merata, yang menyebabkan peningkatan kesalahan pada sistem rekomendasi.

Berdasarkan penelitian [7] S. Badrul, et al yang berjudul “*Item-based collaborative filtering Recommendation Algorithms*”, 2011. Penelitian ini menyajikan dan mengevaluasi algoritma baru untuk sistem rekomendasi berbasis *Collaborative filtering*. Hasil yang didapatkan oleh penulis menunjukkan bahwa *item-based* memungkinkan algoritma berbasis *Collaborative filtering* untuk menghasilkan skala set data yang besar dan pada saat yang sama menghasilkan rekomendasi berkualitas tinggi.

Berdasarkan penelitian oleh [8] Chen, et al yang berjudul judul “HPCF: *Hybrid Musik Group Recommendation System based on Item Popularity and Collaborative filtering*” dilakukan pada tahun 2020, dengan judul yang memanfaatkan jumlah pemutaran lagu menggunakan metode berbasis popularitas dan SVD. Sistem yang dibangun merupakan sistem rekomendasi dengan dataset yang didapatkan dari *Last.fm* dan menggunakan pendekatan hibrida. [16] Penelitian ini dibagi menjadi 3 kelompok yaitu kelompok acak, kelompok dengan kesamaan yang tinggi, dan kelompok dengan kesamaan yang rendah. Dan, didapatkan hasil evaluasi yang menunjukkan bahwa HPCF memiliki hasil yang lebih unggul dibandingkan metode berbasis popularitas dan SVD.

Berdasarkan penelitian oleh [5] Oscar Celma dan Pedro Cano, yang berjudul “*From Hits to Niches? Or How Popular Artist Can Bias Musik Recommendation and Discovery*” yang dilakukan pada tahun 2008. Penelitian ini dilakukan dengan mengukur popularitas dalam jumlah pemutaran lagu terhadap *Long Tail* data untuk merangking semua artis. Dan, hasilnya menunjukkan bahwa *Collaborative Filtering* menghasilkan rekomendasi yang lebih baik dibandingkan *Content-based*.

Berdasarkan penelitian oleh [9] Oleg Lesota, et al yang berjudul “*Analyzing Item Popularity Bias of Musik Recommender Systems: Are Different Genders Equally Affected?*” yang dilakukan pada tahun 2021. Penelitian ini menganalisis secara mendalam bias popularitas dalam sistem rekomendasi musik dengan menggunakan interaksi pengguna dengan jumlah pemutaran lagu lebih dari 1. Dan yang menjadi fokus utama terdapat pada efek popularitas terhadap rekomendasi yang diberikan pengguna. Penelitian ini menggunakan algoritma *collaborative filtering* umum yang berbeda, yaitu *Random Item* (algoritma dasar yang memberikan rekomendasi *item* secara acak untuk setiap pengguna), *Most Popular Items* (algoritma berbasis heuristic yang merekomendasikan kumpulan *item* paling populer secara keseluruhan kepada setiap pengguna), *Item K-Nearest Neighbors* (algoritma berbasis tetangga yang memberikan rekomendasi *item* berdasarkan kesamaan antar *item*), *Sparse Linear Method* (algoritma berbasis tetangga yang mempelajari langsung dari data dengan model regresi), *Alternating Least Squares* (sebuah pendekatan faktorisasi matriks yang mempelajari embedding pengguna dan *item* sehingga hasil perkalian titik dari keduanya mendekati matriks interaksi pengguna-*item* asli), *Matrix Factorization with Bayesian Personalized Ranking* (algoritma yang bertujuan untuk memberikan ranking pada *item* yang dikonsumsi oleh pengguna sesuai dengan preferensi mereka), *Variational Autoencoder* (memperkirakan

distribusi probabilitas atas semua *item* menggunakan arsitektur yang bervariasi autoencoder).

Berdasarkan penelitian oleh [10] Chigozirim Ajaegbu, yang berjudul “*An Optimized Item-Based Collaborative Filtering Algorithm*” pada tahun 2021, dari hasil eksperimen yang telah dilakukan menunjukkan bahwa algoritma *item-based collaborative filtering* yang ditingkatkan menambahkan nilai yang lebih pada algoritma dengan meningkatkan kemampuannya dalam mengatasi masalah ekstrim pada spasitas data dan *cold start* sehingga menghasilkan sistem rekomendasi yang lebih baik. Penelitian ini menggunakan perhitungan kesamaan dengan *cosine-based similarity*, *pearson correlation similarity*, dan *adjusted cosine similarity*.

Berdasarkan penelitian oleh [11] Abdalla, Hassan I., et al. “*Boosting the item-based collaborative filtering model with novel similarity measures.*” Pada tahun 2023, tujuan utama dari penelitian ini adalah menggunakan model berbasis *item* dan pengguna secara efisien untuk mengurangi dampak masalah sparsitas pada kinerja *Collaborative Filtering*. Untuk menghasilkan performansi *Collaborative Filtering* yang terbaik menggunakan model *item-based*, penelitian ini mengembangkan ukuran baru berdasarkan ideologi dari TF (*Term Frequency*) dan IDF (*Inverse Document Frequency*). Pengukuran ini dapat dilihat sebagai quasi-IDF.

A. *Recommender system*

Sistem rekomendasi adalah suatu sistem penyaringan yang bertujuan untuk memprediksi preferensi yang diberikan oleh pengguna terhadap suatu elemen tertentu, pada penelitian ini terhadap sebuah lagu. Dalam proses penyaringan, dibutuhkan algoritma yang memberikan rekomendasi tertentu dan menyarankan satu *item* atau sekelompok *item* kepada pengguna berdasarkan prediksi preferensi. Sistem rekomendasi adalah sistem atau teknik yang merekomendasikan atau menyarankan produk, layanan, atau entitas tertentu. Namun, sistem-sistem ini dapat diklasifikasikan ke dalam dua kategori berikut, berdasarkan pendekatan mereka dalam memberikan rekomendasi. [7]

Telah banyak pekerjaan yang dilakukan, baik di industri maupun di dunia akademis dalam mengembangkan pendekatan baru untuk sistem rekomendasi selama beberapa tahun terakhir. Minat dalam bidang ini masih tinggi karena merupakan area penelitian yang kaya masalah dan karena adanya berbagai aplikasi praktis yang membantu pengguna mengatasi kelebihan informasi dan memberikan rekomendasi, konten, dan layanan yang personal untuk mereka. Contoh aplikasi tersebut termasuk memberikan rekomendasi buku, CD, dan produk lainnya di *Amazon.com* [12], film oleh *MovieLens* [13], dan berita di *VERSIFI Technologies*

(sebelumnya *AdaptiveInfoe.com*) [14]. Selain itu, beberapa vendor telah menggabungkan kemampuan rekomendasi ke dalam server perdagangan mereka [15].

B. *Collaborative filtering*

Collaborative filtering merupakan kolaborasi antara *user* yang melakukan interaksi dengan *item* dan mengeluarkan hasil berupa eksplisit dan implisit. Atribut pada *user* yaitu seperti umur, hobi, usia, bahasa dan lain lain. Sedangkan atribut pada *item* yaitu seperti judul, tahun, genre dan lain lain. Pengguna yang memiliki preferensi yang serupa dalam interaksi mereka dengan *item* atau konten tertentu dalam memberikan rekomendasi kepada satu sama lain. Sistem *Collaborative filtering* bekerja dengan mengumpulkan pendapat pengguna tentang *item* di suatu kategori dan menggunakan kesamaan dalam dengan cara pengguna memberikan nilai untuk menentukan rekomendasi *item*. [16]

Collaborative filtering dibangun berdasarkan preferensi dan penilaian bersama pengguna terhadap lagu-lagu. Sistem ini memberikan asumsi jika pengguna 1 dan pengguna 2 memiliki preferensi yang serupa. *Item-item* serupa dapat direkomendasikan kepada mereka. Artinya jika pengguna 1 menyukai suatu *Item* tertentu, kemungkinan besar lagu tersebut juga disukai oleh pengguna 2 dan sebaliknya. *Collaborative filtering* umumnya dianggap lebih akurat, karena mengandalkan interaksi langsung pengguna dibandingkan dengan *Content-based*.

Riwayat pengguna sebelumnya, sangat diandalkan oleh algoritma dalam memberikan rekomendasi hal-hal yang disukai, selera, preferensi serupa bagi pengguna selanjutnya. Filter kolaboratif adalah salah satu model rekomendasi paling populer yang digunakan dalam industri dan telah meraih kesuksesan besar bagi perusahaan seperti *Amazon*. Dalam dunia musik, sistem rekomendasi merupakan bagian dari aplikasi streaming besar seperti *Spotify*, *Youtube Musik*, *Deezer*, *Tidal*, dan sejenisnya. Filter kolaboratif dapat secara umum dikelompokkan menjadi dua tipe. [7]

TABEL 1. Perbandingan Teknik Rekomendasi

Reccomendation Approach	Recommendation Technique	
	Heuristic-based	Model-based
<i>Collaborative</i>	<i>Nearest neighbour (cosine, correlation) Clustering Graph theory</i>	<i>Bayesian networks Clustering Artificial neural networks Linear regression Probabilistic models</i>

TABEL 2. Kelebihan dan kekurangan metode *Collaborative filtering*

Sistem Rekomendasi <i>Collaborative filtering</i>	
Kelebihan	Kekurangan
<ul style="list-style-type: none"> • Hasil rekomendasi yang beragam • Fokus pada penggunaan informasi dari pengguna lain preferensi atau perilaku serupa. 	<ul style="list-style-type: none"> • Sistem tidak dapat memberikan rekomendasi jika belum ada penilaian pada objek yang direkomendasikan (tidak ada informasi preferensi untuk pengguna baru dan <i>item</i> baru) • <i>Collaborative filtering</i> menghasilkan data yang kurang akurat ketika penilaian pada satu data terlalu sedikit dan menjadi salah persepsi (mempengaruhi keakuratan algoritma)

C. Implisit

Implisit *Feedback* melibatkan masukan tidak langsung dari pengguna seperti pengguna yang memutar lagu secara terus menerus, maka sistem dapat mengeluarkan asumsi bahwa pengguna tersebut menyukai lagu tersebut. Pada Sistem Rekomendasi Lagu ini, hasil yang dikeluarkan merupakan implisit karena *Feedback* yang diberikan melalui perilaku pengguna tanpa membutuhkan langsung tanggapan dari pengguna. Pada kasus ini merupakan data “*Playcount*” yang merupakan jumlah pemutaran lagu oleh pengguna.

Pengguna memberikan umpan balik langsung mengenai preferensi mereka melalui tindakan, yaitu dengan berapa kali mereka memutar lagu tertentu. Data implisit seperti ini dapat memberikan informasi yang kuat untuk memahami preferensi pengguna karena mencerminkan tindakan yang dilakukan oleh pengguna. Pengoptimalan kesalahan prediksi pada implisit lebih melibatkan pengguna daripada mengoptimalkan kesalahan prediksi pada eksplisit. [17]

D. *Item-based collaborative filtering*

Jika sekelompok orang telah memberi peringkat yang serupa pada dua *item*, maka kedua *item* tersebut harus serupa. Oleh karena itu, jika seseorang menyukai satu *item* tertentu, kemungkinan besar

mereka juga tertarik pada *item* lainnya. Ini merupakan prinsip yang mendasari cara kerja filter berbasis *item*. Dimana, *item-based* melakukan proses perhitungan *similarity* antar *item* [7].

Item-based collaborative filtering dihitung menggunakan *sparse matrix*, sehingga untuk melakukan perhitungan *similarity*, perlu dikonversi terlebih dahulu menjadi *dense matrix*, sehingga hasilnya sama dan dapat dilakukan perhitungan *similarity*.

Kelebihan *Item-based collaborative filtering*: [7]

1. Skalabilitas

Metode ini menghasilkan model yang ringan, bahkan pada dataset yang besar. Model ini tetap efisien bahkan saat diterapkan pada dataset yang lebih besar.

2. Kecepatan Prediksi

Metode ini lebih cepat dibandingkan dengan metode berbasis pengguna karena dataset yang digunakan untuk prediksi yang lebih kecil, sehingga prosesnya lebih cepat.

3. Karakteristik Data

Cocok untuk data dengan perbedaan jumlah *item* yang besar antara satu pengguna dengan pengguna yang lainnya.

Kelemahan *Item-based collaborative filtering*:

1. *Inflexibility*

Ketika ada penambahan *item* baru, perlu dilakukan perubahan pada query, yang dapat menjadi proses yang rumit atau memakan waktu.

E. *Item Similarity*

Item similarity adalah konsep dalam *Collaborative filtering* yang mengukur sejauh mana dua *item* mirip atau memiliki kesamaan berdasarkan preferensi pengguna. Dalam sistem rekomendasi, *item similarity* digunakan untuk memprediksi preferensi atau rekomendasi pengguna berdasarkan kesamaan antara *item* yang disukai atau diinteraksi oleh pengguna [12].

Terdapat berbagai metode untuk mengukur *item similarity*, dan metode yang umum digunakan adalah:

1. *Cosine similarity*

Cosine similarity mengukur sudut kosinus antara dua vektor *item* dalam ruang fitur. Semakin kecil sudut kosinus, semakin mirip kedua *item* tersebut dianggap. Nilai *cosine similarity* berkisar antara -1 (*item* bertolak belakang), 0 (*item* tidak memiliki kesamaan), dan 1 (*item* identik).

Item similarity dapat digunakan untuk memperkirakan entri untuk pengguna U dan *item* I. Dengan cara menemukan M *item* yang paling mirip dengan I, untuk beberapa nilai M, dan mengambil rata-rata peringkat diantara M *item* dari peringkat yang telah diberikan oleh pengguna U. [18]

F. User Similarity

User similarity adalah konsep dalam collaborative filtering yang mengukur sejauh mana dua pengguna mirip atau memiliki kesamaan dalam preferensi mereka terhadap item. Dalam sistem rekomendasi, user similarity digunakan untuk memprediksi preferensi atau rekomendasi untuk seorang pengguna berdasarkan kesamaan preferensi dengan pengguna lain.

Dengan memahami dan mengukur sejauh mana dua pengguna memiliki preferensi yang serupa, sistem dapat menghasilkan rekomendasi yang lebih personal dan relevan, dapat menciptakan pengalaman pengguna yang lebih memuaskan serta meningkatkan efektivitas sistem rekomendasi secara keseluruhan.

G. Perhitungan Similarity (Cosine similarity)

Pada Cosine similarity, dua item dianggap sebagai dua vektor dalam ruang pengguna berdimensi m. Kesamaan antara keduanya diukur dengan menghitung cosinus dari sudut antara kedua vektor ini. Kesamaan antara item a dan b, yang dilambangkan dengan sim (a, b) [19].

Rumus Cosine similarity antara dua vektor a dan b adalah:

$$\cos(a, b) = \frac{a \cdot b}{\|a\| \cdot \|b\|} \tag{1}$$

Dimana:

1. a, b adalah vektor
2. a, b adalah hasil dari produk titik antara vektor A dan vektor B
3. ||a|| adalah panjang atau norma dari vektor A.
4. ||b|| adalah panjang atau norma dari vektor B.

TABEL 3.
Kelebihan dan kekurangan Cosine similarity

Cosine similarity	
Kelebihan	Kekurangan
<ul style="list-style-type: none"> • Memiliki tingkat akurasi yang tinggi • Mudah untuk dipahami. Nilai kesamaan cosinus berkisar dari -1 hingga 1. Dimana, 1 berarti vektor identik dan -1 berarti keduanya 	<ul style="list-style-type: none"> • Membutuhkan waktu yang lama dalam tahap perhitungan nilai kesamaan. • Peka terhadap keberadaan nilai jarang atau nol dalam vektor, yang dapat membuat sudut antar vektor sangat kecil dan menghasilkan skor kesamaan yang tinggi bahkan ketika

sama sekali berbeda.	vektornya berbeda.
----------------------	--------------------

H. Weighted Sum

Metode Weighted Sum merupakan pendekatan dalam sistem rekomendasi yang digunakan untuk menghitung prediksi peringkat suatu item oleh seorang pengguna. Pendekatan ini melibatkan perhitungan prediksi item tertentu pada seorang pengguna dengan cara menjumlahkan produk dari penilaian pengguna lain yang memiliki kesamaan preferensi terhadap suatu item tertentu. Bobot atau nilai penting diberikan pada penilaian pengguna lain yang memiliki kemiripan preferensi terhadap suatu item tertentu.

Weighted average dengan menggunakan cosine similarity untuk menentukan rating dapat dilakukan dengan melakukan langkah pertama yaitu menghitung cosine similarity, mendapatkan pengguna dengan kesamaan tertinggi, lalu menghitung bobot, dimana bobot tersebut adalah cosine similarity + 1. Hal ini dilakukan agar memastikan bahwa bobot minimum adalah 1. Setelah itu menghitung prediksi rating dan yang terakhir mengeluarkan hasil performansi.

Metode Weighted Sum adalah metode untuk menghitung prediksi item i pada user u dengan menjumlahkan produk dari penilaian pengguna lain berupa rating pada item yang mirip dengan item i [19].

Rumus Weighted Sum untuk prediksi dalam Collaborative filtering dapat dinyatakan sebagai berikut:

$$P(u, i) = \frac{\sum_{all\ similar\ items, N} (S_{i,N} * R_{u,N})}{\sum_{all\ similar\ items, N} (S_{i,N})} \tag{2}$$

Dimana:

1. P_{u, i} adalah prediksi rating item i pada user u
2. S_{i, N} adalah similarity pada item i dan item N
3. R_{u, N} adalah rating item N pada user u

MAE, MSE, dan RMSE

Tingkat kesalahan model prediksi dapat dievaluasi dengan menerapkan beberapa matriks akurasi dalam pembelajaran mesin dan statistik. Konsep dasar evaluasi akurasi adalah membandingkan target asli dengan target yang diprediksi, dan menerapkan matriks seperti MAE, MSE dan RMSE untuk menjelaskan kesalahan dan kemampuan prediktif model.

Rumus MAE dihitung dengan menjumlahkan selisih absolut antara nilai prediksi dan nilai sebenarnya, kemudian membagi jumlah tersebut dengan jumlah sample dalam data.

$$MAE = \frac{\sum_{t=1}^n |A_t - F_t|}{n} \tag{3}$$

1. n adalah jumlah sample dalam data
2. A_t merupakan nilai sebenarnya
3. F_t merupakan nilai prediksi

Rumus MSE menghitung rata-rata dari kuadrat selisih antara nilai prediksi dan nilai sebenarnya.

$$MSE = \frac{\sum_{t=1}^n (A_t - F_t)^2}{n} \tag{4}$$

1. n adalah jumlah *sample* dalam data
2. A_t merupakan nilai sebenarnya
3. F_t merupakan nilai prediksi

Rumus RMSE adalah akar kuadrat dari MSE. Ini memberikan indikasi tentang besarnya kesalahan rata-rata dalam satuan yang sama dengan variable terikat.

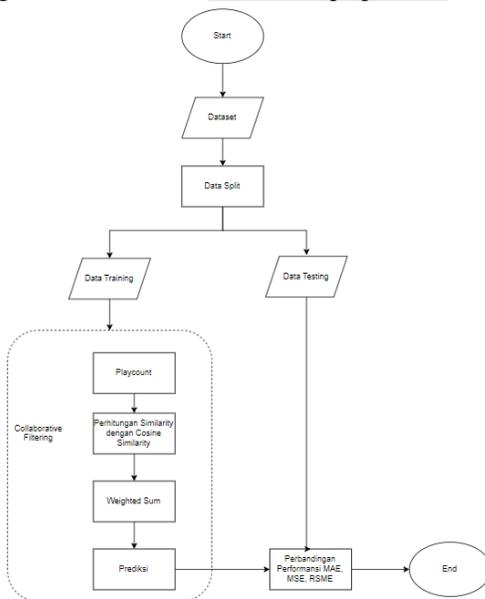
$$RMSE = \sqrt{\frac{\sum_{t=1}^n (A_t - F_t)^2}{n}} \tag{5}$$

1. n adalah jumlah *sample* dalam data
2. A_t merupakan nilai sebenarnya
3. F_t merupakan nilai prediksi

III. SISTEM YANG DIBANGUN

Dalam penelitian ini, yang menjadi utama yaitu menghasilkan perbandingan perfomansi MAE, MSE dan RMSE dengan menggunakan metode *Collaborative filtering*. *Collaborative filtering* merupakan sebuah proses filtering *item* menggunakan informasi dari pengguna lain. Proses ini melibatkan perhitungan kesamaan (*similarity*) antar pengguna dan antar *item*, seperti artis, untuk menghasilkan rekomendasi.

Rancangan sistem yang dibangun untuk penelitian ini dapat dilihat pada Gambar 1. Sistem ini mencakup langkah-langkah seperti memasukkan dataset, dilakukan pembagian dataset menjadi *data training* dan *data testing*, perhitungan *similarity* dengan *cosine similarity*, menghasilkan prediksi dengan perhitungan *weighted sum*, dan hasil akhir berupa perfomansi.



GAMBAR 1 Rancangan Sistem Rekomendasi

A. Dataset

Data lagu yang ada, diperoleh dari *2nd International Workshop* yang diadakan oleh *Information Heterogeneity and Fusion in Recommender systems (HetRec 2011)* <http://ir.ii.uam.es/hetrec2011> pada *5th ACM Conference on Recommender systems (RecSys 2011)* <http://recsys.acm.org/2011> dan berisi informasi mengenai lagu yang didengarkan dari 2000 pengguna dari sistem musik *online Last.fm*. Terdapat 1892 pengguna, 17.632 artis dan 92.834 relasi antar pengguna dan artis yang menunjukkan interaksi dalam bentuk jumlah pemutaran lagu oleh pengguna.

Dataset ini menjadi referensi utama dalam mengembangkan Sistem Rekomendasi Lagu Berdasarkan Popularitas dengan Menggunakan Metode *Collaborative filtering*. Dataset ini dipilih dengan mempertimbangkan variable data yang terdapat didalamnya. Dengan dataset ini, diharapkan hasil penelitian dapat menghasilkan akurasi dan yang perfomansi yang baik.

TABEL 4.

Representasi dataset *Playcount* artis-pengguna

<i>userID</i>	<i>artistID</i>	<i>Playcount</i>
2	51	13883
2	52	11690
:	:	:
1225	99	62
1225	153	16

B. Data Split

Data yang diperoleh dari kumpulan dataset, selanjutnya dibagi menjadi data pelatihan (*training*) dan data pengujian (*testing*). Proses pembagian ini sangat penting untuk menguji dan memvalidasi kinerja model yang dikembangkan. Dalam penelitian ini, perbandingan data yang digunakan adalah 70:30, 75:25 dan 80:20.

Pemilihan perbandingan ini memiliki dampak signifikan pada evaluasi dan akurasi model yang dihasilkan, Dengan 3 variasi perbandingan tersebut, penelitian ini bertujuan untuk mengeksplorasi pengaruh proposi data pelatihan dan data pengujian terhadap perfomansi sistem rekomendasi lagu. Dengan melakukan pemisahan data yang berbeda, diharapkan hasil penelitian dapat menyajikan rekomendasi lagu yang akurat dan sesuai dengan preferensi pengguna.

C. Data Train

Data training adalah kumpulan data yang memiliki atribut label atau kelas, yang digunakan oleh sistem untuk mempelajari karakteristik data sehingga dapat mengidentifikasi pola atau model data[20]. Proses ini bertujuan agar mesin mampu menghasilkan model yang dapat memberikan respon yang tepat

terhadap dataset baru yang diberikan. *Data training* menjadi pondasi utama dalam mengembangkan algoritma dan model yang efektif.

Melalui *data training*, mesin belajar untuk mengeksplorasi dan memahami sejumlah informasi yang terkandung dalam atribut label atau kelas. Dengan mengumpulkan pola-pola yang relevan dari data tersebut, model dapat dihasilkan sehingga mampu memberikan prediksi yang akurat. Karena itu, peran *data training* sangat penting dalam memastikan kualitas dan kemampuan model untuk menanggapi variasi data yang diberikan.

D. Data Test

Data testing merupakan kumpulan data yang juga dilengkapi dengan label atau kelas yang digunakan untuk menguji keakuratan pola atau model dalam mengklasifikasikan data pengujian [20]. Data tersebut dihasilkan dari *data training*, dan setiap data dalam kelompok ini juga dilengkapi dengan atribut label yang berfungsi untuk mengukur sejauh mana model mampu mengklasifikasikan data secara akurat. Proses ini penting untuk menilai ketepatan model dan memastikan bahwa model tersebut dapat memberikan prediksi yang tepat.

Selanjutnya, *data testing* memungkinkan pengguna untuk mengukur sejauh mana model dapat memproses informasi yang diperoleh dari *data training*. Pengujian ini juga memberikan gambaran tentang seberapa baik model mampu mengolah data. Oleh karena itu, kualitas *data testing* sangat menentukan keberhasilan suatu model dan terhadap hasil performanya.

E. Playcount

Sistem rekomendasi berbasis *Collaborative Filtering* biasanya menggunakan matriks pengguna-*item* yang terdiri dari *feedback* dari setiap pengguna untuk setiap *item*. *Feedback* dapat berupa eksplisit (misalnya, penilaian) atau implisit (misalnya, jumlah interaksi) [21]. Eksplisit *feedback* membutuhkan pengguna untuk memberikan masukan tambahan, sementara *implicit feedback* tidak perlu.

Pada penelitian ini, menggunakan jumlah interaksi yaitu *Playcount* digunakan langsung untuk menghitung kemiripan antar pengguna atau *item* tanpa perlu dikonversi menjadi rating. Dalam konteks *collaborative filtering*, data *playcount* dapat berfungsi sebagai umpan balik implisit, dimana interaksi (*playcount*) antara pengguna dan *item* digunakan untuk menghitung skor kemiripan. Dimana, nilai maksimum *playcount* adalah 320.725 dan nilai minimum *playcount* adalah 1.

F. Perhitungan Similarity (Cosine similarity)

Dalam *cosine similarity*, dua *item* dianggap sebagai dua vektor dalam ruang pengguna berdimensi *m*. Tingkat kesamaan antara keduanya diukur dengan menghitung kosinus dari sudut antara kedua vektor tersebut. Kesamaan antara *item* a dan b, yang dilambangkan dengan sim (a, b) [19].

Cosine similarity diperoleh dengan membagi hasil *dot product* dari dua vektor dengan hasil kali panjang vektor tersebut. Semakin tinggi nilai sim (a, b), semakin tinggi tingkat kesamaan antara *item* a dan b dalam konteks pengguna atau fitur yang dimiliki oleh sistem. *Cosine similarity* menjadi kunci dalam algoritma *Collaborative filtering*, dimana kemiripan antara preferensi pengguna diukur untuk membuat rekomendasi berbasis pengguna atau *item*.

TABEL 5. Representasi hasil perhitungan *similarity user* antar *user*

userID	2	3	...	1224	1225
2	1.00000	0.00000	...	0.00000	0.219327
3	0.00000	1.00000	...	0.00000	0.00000
:	:	:	...	:	:
1224	0.00000	0.00000	...	1.00000	0.00000
1225	0.219327	0.786995	...	0.00000	1.00000

TABEL 6. Representasi hasil perhitungan *similarity artis* antar artis

artisID	1	2	..	18.744	18.745
1	1.00000	0.00000	..	0.00000	0.00000
2	0.00000	1.00000	..	0.00000	0.00000
:	:	:	..	:	:
18.744	0.00000	0.00000	..	1.00000	0.00000
18.745	0.00000	0.00000	..	0.00000	1.00000

G. Weighted Sum

Metode *weighted sum* adalah metode untuk menghitung prediksi *item* i pada *user* u dengan menjumlahkan rating-rating yang diberikan oleh *user* pada *item* yang mirip dengan *item* I [7]. Dalam proses perhitungan rekomendasi menggunakan *Weighted average* dengan *Cosine Similarity*, langkah awal yang dilakukan adalah menghitung *cosine similarity* antara pengguna dan pengguna, juga artis antar artis dalam dataset. Proses ini bertujuan untuk menentukan sejauh mana kemiripan preferensi antara pengguna-pengguna dan artis artis tersebut.

Selanjutnya, dilakukan perhitungan bobot dan memastikan bahwa nilai bobot memiliki nilai minimum sebesar 1. Proses perhitungan bobot memiliki tujuan untuk meningkatkan kepuasan pengguna dengan memberikan rekomendasi yang lebih sesuai dengan preferensi mereka berdasarkan

pengguna dan artis yang memiliki kesamaan yang tinggi. Setelah mendapatkan bobot, langkah berikutnya adalah menghitung prediksi.

H. Prediksi

Prediksi dalam *Collaborative filtering* bertujuan untuk memberikan prediksi atau menilai sejauh mungkin preferensi atau tingkat ketertarikan pengguna terhadap *item* yang belum dievaluasi oleh pengguna tersebut. Dengan kata lain, *Collaborative filtering* bertujuan untuk memberikan rekomendasi yang sesuai kepada pengguna berdasarkan informasi dari pengguna lain yang memiliki preferensi serupa.

Pada laporan ini, yang menjadi *item* merupakan artis. Sehingga, tujuannya adalah menghasilkan prediksi berdasarkan artis kepada pengguna berdasarkan preferensi saat mendengarkan lagu dari pengguna lain yang memiliki kesamaan dalam selera lagu. Sumber data yang digunakan untuk melakukan prediksi adalah jumlah pemutaran lagu, dilihat berapa kali jumlah pemutaran lagu yang dilakukan oleh pengguna tersebut. Semakin banyak lagu tersebut diputar, semakin direkomendasikan artis tersebut kepada pengguna lain.

I. Perbandingan Performansi MAE, MSE, dan RSME

Setelah didapatkan nilai prediksi, dilakukan perhitungan performansi antara MAE, MSE, dan RSME.

Sebagai perbandingan, dilakukan analisis antara *data training* dan *data testing* yang akan dipisahkan dalam tiga perbandingan, yaitu 70:30, 75:25 dan 80:20. Pada setiap perbandingan, telah dilakukan uji coba sebanyak 5 kali. Selanjutnya, diamati perbandingan mana yang menghasilkan kesalahan yang lebih besar diantara ketiga perbandingan tersebut dan perbandingan mana yang menghasilkan performansi yang akurat.

TABEL 7. Hasil Perbandingan

Perbandingan	Data Train	Data Test	Uji Ke-	MAE	MSE	RMSE
70:30	38.479	16.492	1	9,514	52.577,457	229,297
			2	9,545	53.403,922	231,092
			3	9,411	58.502,253	241,872
			4	9,475	54.935,068	234,382
			5	9,426	43.848,322	209,399
Rata - rata				9,474	52.653,40	229,208
75:25	41.228	13.743	1	9,907	40.171,129	200,427
			2	9,911	58.557,738	241,987
			3	9,984	20.235,203	142,250
			4	9,939	39.520,206	198,796
			5	9,771	71.089,972	266,627
Rata - rata				9,902	45.914,85	210,017

80:20	43.976	10.995	1	10,504	39.294,439	198,228
			2	10,583	59.250,658	243,414
			3	10,401	40.719,616	201,791
			4	10,576	76.871,448	277,257
			5	10,368	27.686,388	166,392
Rata-rata				10,486	48.764,51	217,416

IV. EVALUASI

Penelitian ini menggunakan dataset yang terdiri dari 54.971 data artis. Dataset ini mencakup *userID*, *artistID*, dan *Playcount*. Evaluasi yang dilakukan untuk mengukur akurasi performansi terhadap *Playcount* (jumlah pemutaran lagu) dan membandingkan performansi model pada *data training* dan *data testing* dalam 3 perbandingan dan telah didapatkan hasil rata-rata dari 5 uji coba yang telah dilakukan.

A. Hasil Pengujian

Tabel 3 menampilkan pembagian perbandingan ke dalam tiga bagian, yaitu 70:30, 75:25, dan 80:20. Penelitian ini dilakukan dengan melakukan lima percobaan uji coba untuk mendapatkan rata-rata dari hasil pengujian. Tujuan pengujian ini adalah untuk mengevaluasi akurasi performansi nilai *Mean Absolute Error*, *Mean Squared Error*, dan *Root Mean Squared Error* dan membandingkan manakah perbandingan yang menghasilkan performansi terbaik.

Jika dilihat pada rata-rata MAE dan RMSE, yang menghasilkan performansi paling baik adalah perbandingan 70:30 karena menghasilkan nilai 9,469 dan merupakan nilai dengan hasil terkecil diantara perbandingan yang lain. Sementara pada rata-rata MSE yang menghasilkan performansi paling baik adalah perbandingan 75:25 yang menghasilkan nilai 47.439,171.

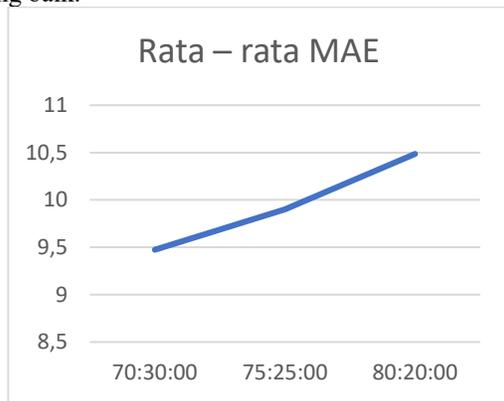
TABEL 8. Hasil Perbandingan

Perbandingan	Data Train	Data Test	Rata - rata MAE	Rata - rata MSE	Rata - rata RMSE
70:30	38.479	16.492	9,474	52.653,40	229,208
75:25	41.228	13.743	9,902	45.914,85	210,017
80:20	43.976	10.995	10,486	48.764,51	217,416

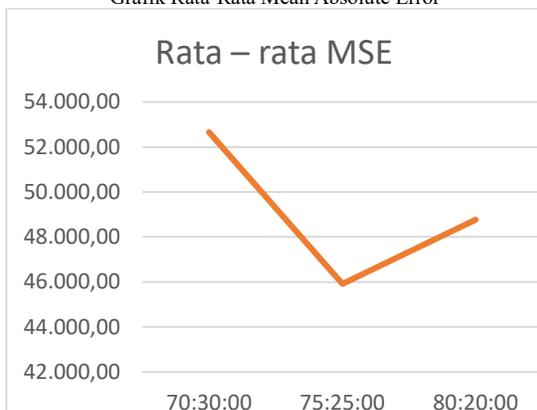
B. Analisis Hasil Pengujian

Berdasarkan hasil pengujian yang telah diperoleh dari ketiga perbandingan, didapatkan bahwa dari tiga perbandingan dengan 5 kali uji coba yang telah dilakukan, perbandingan yang menghasilkan hasil terbaik adalah perbandingan dengan nilai MAE, MSE, dan RMSE terendah. Dalam penelitian ini, perbandingan 75:25 menunjukkan nilai MAE, MSE, dan RMSE yang lebih rendah daripada perbandingan 70:30 dan 80:20, menunjukkan kinerja yang lebih baik dalam memprediksi preferensi pengguna. Sementara,

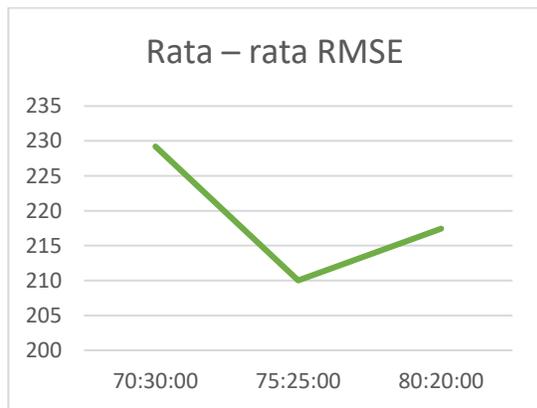
perbandingan 80:20 menghasilkan perbandingan yang kurang baik.



GAMBAR 2
Grafik Rata-Rata Mean Absolute Error



GAMBAR 3
Grafik Rata-Rata Mean Squared Error



GAMBAR 4
Grafik Rata-Rata Root Mean Squared Error

Berdasarkan hasil dari grafik yang ditampilkan, dapat diberikan kesimpulan bahwa semakin besar rasio data train terhadap data test, nilai MAE, MSE, dan RMSE cenderung mengalami peningkatan. Hal ini menunjukkan bahwa kinerja model cenderung memburuk saat data yang digunakan untuk melatih model sedikit. Sehingga, perlu diperhatikan dalam

menggunakan rasio data yang seimbang antara data train dan data test.

V. KESIMPULAN

Penelitian ini bertujuan untuk menghasilkan perbandingan performansi berdasarkan *artist* dengan menggunakan metode *Collaborative filtering* berdasarkan Jumlah Interaksi. Berdasarkan hasil pengujian yang telah dilakukan terhadap tiga perbandingan dengan pembagian data yang berbeda, didapatkan kesimpulan sebagai berikut:

1. Banyaknya jumlah pemutaran lagu yang didengarkan oleh pengguna dapat mempengaruhi hasil rekomendasi. Semakin banyak lagu yang didengarkan oleh pengguna, maka dapat menghasilkan rekomendasi yang baik.
2. Dengan menggunakan *playcount*, semakin tinggi rasio data training yang digunakan, semakin berpengaruh dengan MAE, MSE dan RMSE yang dihasilkan. Semakin kecil nilai MAE, MSE dan RMSE yang dihasilkan, menunjukkan bahwa hasil prediksi oleh algoritma terkait menghasilkan kinerja yang optimal.
3. Kinerja model cenderung kurang baik saat semakin berkurangnya data yang digunakan dalam melatih model. Oleh karena itu, perlu dilakukan pertimbangan saat melakukan pembagian data yang lebih seimbang antara data train dan data test agar model dapat memberikan hasil yang lebih baik.

Saran yang dapat diambil sebagai pertimbangan untuk penelitian selanjutnya adalah mencoba menggunakan metode rekomendasi lain dan atau menggunakan pendekatan *Collaborative Filtering* tipe lain dan bervariasi, dan meningkatkan jumlah perbandingan data set yang digunakan dan melakukan lebih banyak uji coba.

REFERENSI

- [1] Yoshua, I., & Bunyamin, H., "Pengimplementasian Sistem Rekomendasi Musik Dengan Metode Collaborative Filtering," *Jurnal STRATEGI-Jurnal Maranatha*, vol. 3.1, pp. 1-16, 2021.
- [2] D. J. Levitin, "This is your brain on music: The science of a human obsession," *Penguin*, 2006.
- [3] E. Bernhardsson, "Music recommendations at spotify," *Online https://de. slideshare.net/erikbern/collaborative-filtering-at-spotify-16182818*, 2013.
- [4] P. B. Lamere, "I've got 10 million songs in my pocket: now what?," *Proceedings of the sixth ACM conference on Recommender systems*, no.

- <https://doi.org/10.1145/2365952.2365994>, p. Pages 207–208, 2012.
- [5] O. Celma, "Music recommendation." Music recommendation and discovery: The long tail, long fail, and long play in the digital music space," *Berlin, Heidelberg: Springer Berlin Heidelberg*, pp. 43-85, 2010.
- [6] Arwin Halim, Hernawati Gohzali, Dita Maria Panjaitan, Ilham Maulana, "Sistem Rekomendasi Film menggunakan Bisecting K-Means dan Collaborative Filtering," 2017.
- [7] Sarwar, B., Karypis, G., Konstan, J., & Riedl, J., "Item-based collaborative filtering recommendation algorithms.," *In Proceedings of the 10th international conference on World Wide Web*, no. <https://doi.org/10.1145/371920.372071>, pp. 285-295, 2001.
- [8] Chen, Shih-Han, Sok-Ian Sou, and Hsun-Ping Hsieh, "HPCF: hybrid music group recommendation system based on item popularity and collaborative filtering," *2020 International Computer Symposium (ICS), IEEE*, no. doi: 10.1109/ICS51289.2020.00019., pp. 43-49, 2020.
- [9] Lesota, Oleg, et al, "Analyzing item popularity bias of music recommender systems: are different genders equally affected?," *Proceedings of the 15th ACM Conference on Recommender Systems*, no. <https://doi.org/10.1145/3460231.3478843>, pp. 601-606, 2021.
- [10] C. Ajaegbu, "An optimized item-based collaborative filtering algorithm," *Journal of Ambient Intelligence and Humanized Computing*, no. <https://doi.org/10.1007/s12652-020-02876-1>, 2021.
- [11] Abdalla, Hassan I., et al, "Boosting the item-based collaborative filtering model with novel similarity measures," *International Journal of Computational Intelligence Systems* 16.1, no. <https://doi.org/10.1007/s44196-023-00299-2>, 2023.
- [12] Linden, Greg, Brent Smith, and Jeremy York, "Amazon. com recommendations: Item-to-item collaborative filtering," *IEEE Internet computing* 7.1, no. doi: 10.1109/MIC.2003.1167344., pp. 76-80, 2003.
- [13] Miller, Bradley N., et al, "Movielens unplugged: experiences with an occasionally connected recommender system," *Proceedings of the 8th international conference on Intelligent user interfaces*, no. https://doi.org/10.1007/978-1-4471-3754-2_16, pp. 263-266, 2003.
- [14] Billsus, Daniel, et al, "Adaptive interfaces for ubiquitous web access," *Communications of the ACM* 45.5, pp. 34-38, 2002.
- [15] Peddy, Clayton C., and Derek Armentrout, "Building solutions with microsoft commerce server 2002," *Microsoft Press*, no. <https://doi.org/10.1109/TKDE.2005.99>, 2003.
- [16] Melville, Prem, and Vikas Sindhwani, "Recommender System," *IBM TJ, Watson Research centre, Yorktown Heights, NY 10598*, pp. 1-18, 2010.
- [17] Zhao, Qian, et al., "Explicit or implicit feedback? Engagement or satisfaction? A field experiment on machine-learning-based recommender systems," *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, no. <https://doi.org/10.1145/3167132.3167275>, pp. 1331-1340, 2018.
- [18] Aggarwal, Charu C., and Charu C. Aggarwal, "An introduction to recommender systems." *Recommender systems: The textbook*, pp. 1-28, 2016.
- [19] Sunitha, M., T. Adilakshmi, and Mir Zahed Ali, "Enhancing Item-Based Collaborative Filtering for Music Recommendation System," *Smart Computing Techniques and Applications: Proceedings of the Fourth International Conference on Smart Computing and Informatics*, vol. 2, no. https://doi.org/10.1007/978-981-16-1502-3_28, pp. 269-283, 2021.
- [20] R. Banik, "Hands-on recommendation systems with Python: start building powerful and personalized, recommendation engines with Python," *Packt Publishing Ltd*, 2018.
- [21] Kelly, D., & Teevan, J, "Implicit feedback for inferring user preference: a bibliography," *In Acm Sigir Forum*, vol. 37, no. <https://doi.org/10.1145/959258.959260>, pp. 18-23, 2003.
- [22] Sunitha, M., and T. Adilakshmi, "Music recommendation system with user-based and item-based collaborative filtering technique.," *Networking Communication and Data Knowledge Engineering*, vol. 1, no. https://doi.org/10.1007/978-981-10-4585-1_22, pp. 267-278, 2018.
- [23] Musu, Wilem, Abdul Ibrahim, and Heriadi Heriadi, "Pengaruh Komposisi Data Training dan Testing terhadap Akurasi Algoritma C4. 5.,"

- SISITI: Seminar Ilmiah Sistem Informasi dan Teknologi Informasi*, vol. 10, pp. 186-195, 2021.
- [24] Adomavicius, Gediminas, and Alexander Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE transactions on knowledge and data engineering* 17.6, vol. 17, no. doi: 10.1109/TKDE.2005.99., pp. 734-749, 2005.
- [25] Bertin-Mahieux, Thierry, et al, "The million song dataset," *WWW '12 Companion: Proceedings of the 21st International Conference on World Wide Web*, no. <https://doi.org/10.1145/2187980.2188222>, pp. 591-596, 2011.
- [26] Celma, Òscar, and Pedro Cano, "From hits to niches? or how popular artists can bias music recommendation and discovery," *Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition*, no. <https://doi.org/10.1145/1722149.1722154>, pp. 1-8, 2008.