



Sistem Rekomendasi Musik Spotify Berbasis Pendekatan *Hybrid Alternating Least Square dan Content-Based Filtering*

Andy Hermawan^{1*}, Akbar Kanugraha², Indira Faisa Afgani³,
Khaerun Nisa' Tri Safaati⁴, Mutiara Ayu Alzahra Ramadhani⁵

¹Universitas Indraprasta PGRI Jakarta, Indonesia

²⁻⁵Purwadhika Digital Technology School, Indonesia

Email: andy.hermawan@unindra.ac.id¹, kanuakbar27@gmail.com², indiraafgani@gmail.com³,
khaerunnisa2903@gmail.com⁴, mutiararamadhani2002@gmail.com⁵

*Penulis Korespondensi: andy.hermawan@unindra.ac.id

Abstract. The exponential growth of digital music catalogs on streaming platforms such as Spotify has made personalized recommendation systems crucial for enhancing user experience. This study develops a hybrid music recommendation system that addresses both warm-user and cold-user scenarios by combining Alternating Least Squares (ALS) collaborative filtering with content-based filtering (CBF) augmented by a popularity component. The dataset consists of 8,549,544 user-track interactions and a master file of 1,204,025 tracks with ten audio features. After preprocessing, users were segmented into 14,880 warm users and 723 cold users based on a five-interaction threshold. The ALS model was trained on the user-item implicit feedback matrix and tuned through grid search over factors, alpha, and regularization. CBF was implemented using cosine similarity on normalized audio features, while popularity scores were applied for new users with insufficient history. Evaluation used Precision@10, Recall@10, and NDCG@10. The final ALS configuration achieved NDCG@10 of 0.1116, representing a 30% improvement over baseline, while the hybrid CBF improved NDCG@10 for cold users from 0.0070 to 0.0201. Findings indicate that adaptive routing among ALS, CBF, and popularity reliably handles different user states, providing a practical foundation for production-grade music recommendation systems.

Keywords: Cold Start; Collaborative Filtering; Content-Based; Hybrid Recommender; Music Streaming.

Abstrak. Pertumbuhan katalog musik digital yang sangat pesat pada platform *streaming* seperti Spotify menjadikan sistem rekomendasi yang dipersonalisasi sebagai komponen krusial untuk meningkatkan pengalaman pengguna. Penelitian ini membangun sistem rekomendasi musik berbasis pendekatan hibrida yang mampu menangani skenario *warm user* maupun *cold user* dengan memadukan *collaborative filtering* menggunakan *Alternating Least Squares* (ALS) dan *content-based filtering* (CBF) yang dilengkapi komponen popularitas. Dataset yang digunakan terdiri atas 8.549.544 interaksi pengguna-lagu serta *master file* berisi 1.204.025 lagu dengan sepuluh fitur audio. Setelah praproses, pengguna disegmentasi menjadi 14.880 *warm user* dan 723 *cold user* berdasarkan ambang lima interaksi. Model ALS dilatih pada matriks *implicit feedback* dan dioptimalkan melalui *grid search* terhadap parameter *factors*, *alpha*, dan *regularization*. CBF diimplementasikan menggunakan *cosine similarity* pada fitur audio yang dinormalisasi, sedangkan skor popularitas digunakan untuk pengguna baru yang belum memiliki riwayat interaksi. Evaluasi dilakukan menggunakan metrik Precision@10, Recall@10, dan NDCG@10. Konfigurasi akhir ALS menghasilkan NDCG@10 sebesar 0,1116 atau meningkat 30% dibandingkan *baseline*, sementara pendekatan hibrida CBF meningkatkan nilai NDCG@10 *cold user* dari 0,0070 menjadi 0,0201. Penelitian ini menunjukkan bahwa skema pengarah adaptif antara ALS, CBF, dan popularitas mampu menangani berbagai kondisi pengguna secara konsisten serta menjadi fondasi praktis bagi sistem rekomendasi musik skala produksi.

Kata kunci: Berbasis Konten; Layanan Streaming Musik; Masalah Awal (*Cold Start*); Penyaringan Kolaboratif; Sistem Rekomendasi Hibrid.

1. LATAR BELAKANG

Industri musik global telah mengalami transformasi signifikan sejak hadirnya layanan *streaming* digital. Spotify sebagai salah satu platform *streaming* musik terbesar mengelola katalog berisi jutaan lagu yang terus bertambah setiap harinya. Volume konten yang sangat besar ini memunculkan fenomena *information overload*, yaitu kondisi ketika pengguna

mengalami kesulitan menemukan lagu yang relevan secara manual akibat melimpahnya pilihan yang tersedia (Schedl, 2019). Dalam konteks tersebut, sistem rekomendasi (*recommender system*) memegang peranan penting sebagai penghubung antara pengguna dan konten yang sesuai dengan preferensi serta perilaku mereka (Ricci et al., 2022).

Secara umum, terdapat tiga pendekatan utama dalam pengembangan sistem rekomendasi, yaitu *collaborative filtering* (CF), *content-based filtering* (CBF), dan pendekatan *hybrid* (Aggarwal, 2016). CF memanfaatkan pola interaksi antar pengguna untuk mengidentifikasi kesamaan preferensi, sedangkan CBF menggunakan atribut atau karakteristik dari suatu item untuk merekomendasikan item yang serupa. Sementara itu, pendekatan *hybrid* menggabungkan kedua metode tersebut guna saling melengkapi kelemahan masing-masing (Çano & Morisio, 2017). Dalam domain musik, metode CF terbukti efektif untuk pengguna yang memiliki riwayat interaksi yang cukup. Namun, performanya cenderung menurun ketika dihadapkan pada pengguna baru maupun lagu baru yang belum memiliki data interaksi, kondisi yang dikenal sebagai *cold-start problem* (Magron & Févotte, 2022; Panda & Ray, 2022).

Salah satu tipe *collaborative filtering* yang banyak digunakan pada data *implicit feedback* adalah *Alternating Least Squares* (ALS), yang awalnya dikembangkan untuk menangani skenario tanpa adanya rating eksplisit (Rendle et al., 2020). Algoritma ini dirancang khusus untuk situasi ketika preferensi pengguna direpresentasikan secara tidak langsung, misalnya melalui jumlah pemutaran (*play count*) lagu, durasi mendengarkan, atau frekuensi interaksi lainnya. ALS memodelkan tingkat kepercayaan (*confidence*) pengguna terhadap suatu item sebagai fungsi dari intensitas konsumsi, kemudian melakukan dekomposisi matriks menjadi representasi vektor laten pengguna dan item untuk mempelajari pola preferensi tersembunyi (Bobadilla et al., 2020). Dengan kemampuan tersebut, ALS mampu menghasilkan rekomendasi yang relevan meskipun tidak tersedia data rating eksplisit dari pengguna. Oleh karena itu, pendekatan ini hingga kini masih banyak digunakan sebagai *baseline* utama dalam penelitian sistem rekomendasi berbasis *implicit feedback* (Rendle et al., 2022).

Meskipun ALS memiliki performa yang baik untuk *warm user*, sistem rekomendasi yang efektif juga harus mampu melayani *cold user*, yaitu pengguna baru atau pengguna dengan jumlah interaksi yang masih sangat terbatas. Pada kondisi ini, pendekatan *collaborative filtering* menjadi kurang optimal karena minimnya data historis yang dapat digunakan untuk mempelajari preferensi pengguna. Beberapa penelitian menunjukkan bahwa kombinasi antara *collaborative filtering* dan *content-based filtering* mampu mengatasi keterbatasan tersebut.

Hal ini dimungkinkan karena fitur audio lagu, seperti *danceability*, *energy*, dan *valence*, dapat dimanfaatkan untuk menghitung tingkat kemiripan antar lagu tanpa bergantung pada riwayat interaksi pengguna yang panjang (Briand et al., 2021; Vall et al., 2019). Selain itu, pendekatan berbasis popularitas (*popularity-based recommendation*) juga terbukti menjadi strategi cadangan yang efektif ketika informasi mengenai preferensi pengguna hampir tidak tersedia (Ferrari Dacrema et al., 2019).

Beberapa penelitian sebelumnya di Indonesia telah membahas pengembangan sistem rekomendasi musik dengan berbagai pendekatan. Anggoro & Izzatillah (2022) mengimplementasikan *collaborative filtering* berbasis Android untuk memberikan rekomendasi musik, namun penelitian tersebut belum secara eksplisit menangani permasalahan *cold user*. Putra & Santika (2020) menerapkan *content-based filtering* menggunakan fitur audio lagu sebagai dasar rekomendasi, tetapi belum melakukan evaluasi kualitas peringkat menggunakan metrik seperti *Normalized Discounted Cumulative Gain* (NDCG). Sementara itu, Yuniardini & Widiyaningtyas (2024) melakukan perbandingan dengan metode *pearson correlation* dan *cosine similarity* pada sistem *collaborative filtering* musik, meskipun cakupan dataset yang digunakan masih relatif terbatas. Berbeda dengan penelitian-penelitian sebelumnya, penelitian ini mengusulkan pendekatan yang lebih adaptif dengan menggabungkan tiga strategi rekomendasi sekaligus, yaitu *Alternating Least Squares* (ALS), *Content-Based Filtering* (CBF), dan rekomendasi berbasis popularitas. Ketiga pendekatan tersebut diintegrasikan dalam satu kerangka sistem yang memetakan setiap pengguna ke strategi rekomendasi yang paling sesuai berdasarkan jumlah interaksi yang dimiliki pengguna.

Kebaruan penelitian ini terletak pada tiga aspek utama. Pertama, penelitian ini menerapkan ambang batas (*threshold*) sebanyak lima interaksi untuk mengklasifikasikan pengguna ke dalam kategori *warm user* dan *cold user*, sekaligus melakukan pemetaan otomatis ke model rekomendasi yang paling sesuai. Kedua, penelitian ini mengembangkan strategi *hybrid* antara *content-based filtering* dan rekomendasi berbasis popularitas dengan rasio adaptif yang ditentukan berdasarkan jumlah lagu yang pernah didengarkan oleh pengguna. Pendekatan ini memungkinkan sistem memberikan rekomendasi yang lebih fleksibel sesuai tingkat ketersediaan data interaksi pengguna. Ketiga, penelitian ini melakukan evaluasi secara komprehensif menggunakan tiga metrik pemeringkatan, yaitu *Precision@10*, *Recall@10*, dan *Normalized Discounted Cumulative Gain* (NDCG@10), yang diterapkan secara konsisten pada seluruh skenario pengujian. Dengan demikian, penelitian ini tidak hanya mengevaluasi akurasi rekomendasi, tetapi juga kualitas urutan rekomendasi yang dihasilkan sistem.

Penelitian ini bertujuan untuk: (1) membangun model rekomendasi musik Spotify berbasis *Alternating Least Squares* (ALS) untuk *warm user* serta menguji pengaruh *hyperparameter* melalui metode *grid search*; (2) mengembangkan model *content-based filtering* (CBF) yang dilengkapi dengan komponen popularitas untuk menangani *cold user*; dan (3) mengintegrasikan kedua model tersebut ke dalam satu fungsi pengarahan (*routing function*) yang mampu melayani *warm user*, *cold user*, maupun pengguna baru (*new user*). Hasil penelitian ini diharapkan dapat memberikan kontribusi praktis dalam pengembangan sistem rekomendasi musik yang adaptif dan siap diterapkan pada lingkungan produksi. Selain itu, penelitian ini juga diharapkan dapat memperkaya literatur ilmiah di bidang sistem rekomendasi, khususnya pada konteks penelitian di Indonesia.

2. KAJIAN TEORITIS

Sistem Rekomendasi

Sistem rekomendasi (*recommender system*) merupakan salah satu sub bidang *machine learning* yang berfokus pada prediksi preferensi pengguna terhadap suatu item (Ricci et al., 2022). Tujuan utama sistem ini adalah menyaring sejumlah besar pilihan menjadi daftar rekomendasi terurut yang paling relevan bagi setiap individu. Dengan demikian, pengguna dapat menemukan konten yang sesuai secara lebih efisien di tengah melimpahnya informasi yang tersedia. Dalam praktek industri, sistem rekomendasi telah banyak diterapkan pada berbagai platform digital seperti Netflix, Amazon, YouTube, dan Spotify untuk meningkatkan *engagement*, kepuasan, serta retensi pengguna (Gomez-Uribe & Hunt, 2016). Menurut Aggarwal (2016), sistem rekomendasi dapat diklasifikasikan berdasarkan jenis data dan algoritma yang digunakan ke dalam tiga kategori utama, yaitu *collaborative filtering*, *content-based filtering*, dan *hybrid recommender system*.

Collaborative Filtering dan Alternating Least Squares

Collaborative filtering (CF) bekerja berdasarkan asumsi bahwa pengguna yang memiliki pola preferensi serupa di masa lalu cenderung akan memiliki preferensi yang mirip di masa mendatang (Bobadilla et al., 2020). Pendekatan ini memanfaatkan data interaksi antar pengguna dan item untuk menghasilkan rekomendasi tanpa memerlukan informasi eksplisit mengenai karakteristik item. Salah satu metode CF yang paling banyak digunakan adalah *matrix factorization*, yaitu teknik yang mendekomposisi matriks interaksi pengguna-item menjadi dua matriks laten berdimensi rendah yang merepresentasikan preferensi pengguna dan karakteristik item secara tersembunyi (*latent factors*) (Rendle et al., 2020).

Dengan representasi laten tersebut, sistem dapat memprediksi tingkat ketertarikan pengguna terhadap item yang belum pernah diinteraksikan sebelumnya. Pada data *implicit feedback* seperti *play count*, jumlah pemutaran lagu, atau frekuensi konsumsi konten, pendekatan *Alternating Least Squares* (ALS) banyak digunakan karena mampu menangani data tanpa rating eksplisit. ALS memodelkan tingkat kepercayaan (*confidence*) pengguna terhadap suatu item sebagai fungsi linear dari jumlah interaksi yang terjadi (He et al., 2016). Semakin sering suatu lagu diputarkan oleh pengguna, maka semakin tinggi tingkat kepercayaan sistem bahwa lagu tersebut disukai oleh pengguna.

Pada ALS untuk *implicit feedback*, fungsi tujuan yang diminimumkan berupa jumlah kuadrat *error* berbobot antara skor prediksi dan label preferensi biner, yang kemudian ditambahkan dengan komponen regularisasi pada vektor laten pengguna dan item untuk mencegah *overfitting* (Rendle et al., 2022). Optimisasi dilakukan secara bergantian (*alternating optimization*), yaitu dengan mengestimasi vektor pengguna ketika vektor item dipertahankan tetap, lalu mengestimasi vektor item ketika vektor pengguna dipertahankan tetap. Proses ini diulangi secara iteratif hingga model mencapai kondisi konvergen. Beberapa *hyperparameter* utama yang mempengaruhi performa ALS meliputi jumlah faktor laten (*factors*), koefisien kepercayaan (*alpha*), dan koefisien regularisasi (*regularization*). Jumlah faktor laten menentukan kompleksitas representasi preferensi pengguna dan karakteristik item, sedangkan parameter *alpha* mengontrol tingkat kepercayaan terhadap data interaksi implisit. Di sisi lain, parameter regularisasi berfungsi mengurangi risiko *overfitting* pada model.

ALS menjadi salah satu *baseline* yang paling populer dalam penelitian sistem rekomendasi karena memiliki efisiensi komputasi yang tinggi, khususnya pada data berskala besar, sekaligus mampu memberikan performa akurasi yang kompetitif dibandingkan metode lainnya (He et al., 2016).

Dekomposisi Nilai Singular (*Singular Value Decomposition/SVD*)

Singular Value Decomposition (SVD) merupakan teknik dekomposisi matriks klasik dalam aljabar linear yang menguraikan suatu matriks menjadi tiga komponen utama, yaitu matriks ortogonal kiri, matriks diagonal yang berisi nilai *singular*, dan matriks ortogonal kanan. Dalam konteks sistem rekomendasi, SVD digunakan untuk mengaproksimasi matriks interaksi pengguna–item dengan mempertahankan sejumlah K nilai *singular* terbesar sehingga diperoleh representasi laten yang lebih ringkas dan efisien (Rendle et al., 2020). Melalui pendekatan tersebut, SVD mampu menangkap pola tersembunyi (*latent patterns*) antara pengguna dan item, sehingga dapat digunakan untuk memprediksi preferensi pengguna terhadap item yang belum pernah diinteraksikan sebelumnya.

Berbeda dengan ALS, SVD tidak secara eksplisit memodelkan tingkat kepercayaan (*confidence*) pada data *implicit feedback*. Oleh karena itu, performa SVD cenderung kurang optimal ketika diterapkan pada data seperti *play count* tanpa adanya modifikasi tambahan atau penyesuaian khusus terhadap karakteristik data implisit (He et al., 2016).

Content-Based Filtering pada Domain Musik

Content-Based Filtering (CBF) merupakan pendekatan sistem rekomendasi yang memanfaatkan atribut atau karakteristik item untuk menghitung tingkat kemiripan, kemudian merekomendasikan item yang paling mirip dengan item yang sebelumnya disukai pengguna (Pérez-Almaguer et al., 2021). Pada domain musik, atribut yang umum digunakan berupa *audio features* yang diperoleh melalui Spotify Web API, seperti *danceability*, *energy*, *valence*, *acousticness*, *instrumentalness*, *speechiness*, *liveness*, *loudness*, *tempo*, dan *popularity* (Vall et al., 2019). Dalam pendekatan ini, profil pengguna dibangun berdasarkan rata-rata tertimbang dari fitur audio lagu-lagu yang pernah didengarkan. Selanjutnya, profil tersebut dibandingkan dengan profil setiap kandidat lagu menggunakan metode *cosine similarity* untuk menghasilkan skor relevansi rekomendasi.

Keunggulan utama CBF adalah kemampuannya untuk bekerja tanpa bergantung pada data interaksi dari pengguna lain, sehingga relatif lebih tahan terhadap permasalahan *item cold-start* (Schedl, 2019). Namun demikian, pendekatan ini cenderung menghasilkan rekomendasi yang homogen karena hanya berfokus pada kemiripan fitur antar item. Akibatnya, CBF kurang mampu menangkap keragaman preferensi dan pola eksplorasi pengguna sebagaimana yang dapat dilakukan *collaborative filtering* oleh Magron & Févotte (2022). Selain itu, pada pengguna yang baru memiliki sedikit interaksi, khususnya hanya satu lagu yang pernah didengarkan, CBF murni seringkali kurang andal karena profil pengguna yang terbentuk masih sangat dipengaruhi oleh satu titik data tunggal.

Pendekatan Hybrid dan Cold-Start Problem

Cold-start problem mengacu pada kondisi ketika sistem rekomendasi kesulitan memberikan rekomendasi yang relevan akibat minimnya data yang tersedia, baik untuk pengguna baru (*user cold-start*) maupun item baru (*item cold-start*) (Panda & Ray, 2022). Pada platform *streaming* musik seperti Spotify, permasalahan ini menjadi sangat umum karena jumlah pengguna baru terus bertambah setiap harinya. Minimnya riwayat interaksi menyebabkan metode *collaborative filtering* sulit membangun profil preferensi pengguna secara akurat (Çano & Morisio, 2017).

Pendekatan *hybrid recommender system* sebagai metode yang mengkombinasikan beberapa teknik rekomendasi untuk saling menutupi kelemahan masing-masing. Kombinasi tersebut dapat dilakukan melalui berbagai strategi, seperti *weighted hybrid*, *switching hybrid*, maupun *mixed hybrid*. Pada pendekatan *switching*, sistem secara dinamis memilih metode rekomendasi yang paling sesuai berdasarkan kondisi data pengguna.

Untuk menangani skenario *cold user*, kombinasi antara *content-based filtering* dan pendekatan berbasis popularitas (*popularity baseline*) terbukti efektif karena popularitas dapat berfungsi sebagai rekomendasi aman ketika belum tersedia sinyal personal yang kuat dari pengguna (Ferrari Dacrema et al., 2019). Menurut Briand et al. (2021) menunjukkan bahwa strategi *semi-personalized recommendation* berbasis pengelompokan pengguna baru (*user clustering*) mampu meningkatkan akurasi rekomendasi pada platform musik Deezer. Pendekatan serupa diterapkan dalam penelitian ini melalui pengalokasian proporsi rekomendasi antara CBF dan popularitas berdasarkan jumlah interaksi yang dimiliki pengguna. Dengan demikian, sistem dapat secara adaptif menyesuaikan strategi rekomendasi sesuai tingkat ketersediaan data preferensi pengguna.

Matrik Evaluasi Sistem Rekomendasi

Evaluasi sistem rekomendasi umumnya dilakukan menggunakan metrik yang sesuai untuk tugas pemeringkatan *top-K*, seperti *Precision@K*, *Recall@K*, dan *Normalized Discounted Cumulative Gain (NDCG@K)* (Cañamares & Castells, 2020). *Precision@K* digunakan untuk mengukur proporsi item relevan di antara K rekomendasi teratas yang diberikan sistem, sedangkan *Recall@K* mengukur seberapa besar proporsi seluruh item relevan yang berhasil ditemukan dalam K rekomendasi teratas tersebut. Sementara itu, *NDCG@K* memperhitungkan posisi item relevan dalam daftar rekomendasi. Item relevan yang ditempatkan pada peringkat lebih atas akan memberikan kontribusi nilai yang lebih besar dibandingkan item relevan pada posisi lebih bawah melalui mekanisme diskon logaritmik (Krichene & Rendle, 2020).

Penggunaan ketiga metrik tersebut secara bersamaan memberikan evaluasi yang lebih komprehensif terhadap performa sistem rekomendasi. *Precision@K* berfokus pada tingkat keakuratan rekomendasi, *Recall@K* menekankan kemampuan sistem dalam mencakup item relevan, sedangkan *NDCG@K* mengevaluasi kualitas urutan atau peringkat rekomendasi yang dihasilkan. Pemilihan nilai $K=10$ dalam penelitian ini didasarkan pada praktik yang umum digunakan di industri, karena daftar rekomendasi *top-10* dianggap merepresentasikan cara pengguna mengonsumsi rekomendasi pada antarmuka aplikasi musik secara nyata (Briand et al., 2021).

3. METODE PENELITIAN

Rancangan Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan tahapan penelitian sebagai berikut: (1) pengumpulan dan pemahaman data, (2) praproses data, (3) segmentasi pengguna menjadi *warm* dan *cold*, (4) pemilihan model terbaik antara ALS dan SVD pada *warm user*, (5) *tuning* hyperparameter metode ALS melalui *grid search*, (6) pemodelan CBF dan *hybrid CBF-popularity* pada *cold user*, (7) evaluasi menggunakan *Precision@10*, *Recall@10*, dan *NDCG@10*, dan (8) integrasi seluruh model ke dalam satu fungsi pengarah adaptif (*dispatcher*). Implementasi dilakukan menggunakan bahasa pemrograman Python dengan pustaka utama *implicit* untuk ALS, *scipy* untuk SVD, *scikit-learn* untuk pemrosesan fitur, dan *pandas* untuk manipulasi data.

Sumber dan Karakteristik Data

Data yang digunakan terdiri dari dua berkas utama. Berkas pertama adalah *user_data.csv* yang berisi 8.549.544 baris interaksi dengan empat kolom: *user_id*, *artistname*, *trackname*, dan *playlistname*. Berkas kedua adalah *master_track_features.csv* yang merupakan basis data lagu Spotify dengan kolom *id* (Spotify track ID), *name* (judul lagu), *artist_main* (nama artis), daftar *genre*, dan sepuluh fitur audio numerik (termasuk *popularity*). Tidak terdapat rating eksplisit, sehingga sistem rekomendasi dibangun berbasis *implicit feedback* dengan *play count* sebagai sinyal preferensi.

Tabel 1 merangkum statistik dasar kedua berkas setelah dimuat. Volume data yang besar mendorong penggunaan strategi pemrosesan berbasis *chunk* pada tahap penggabungan dataset untuk menjaga stabilitas proses. Pada tahap pemodelan, matriks interaksi direpresentasikan dalam format *Compressed Sparse Row* (CSR) untuk menghemat memori, mengingat sebagian besar entri matriks bernilai nol.

Tabel 1. Statistik Dasar Dataset Penelitian.

Berkas	Jumlah Baris	Atribut Utama
<i>user_data.csv</i>	8.549.544	<i>user_id</i> , <i>artist</i> , <i>track</i> , <i>playlist</i>
<i>master_track_features.csv</i>	1.204.025	<i>id</i> , <i>audio features</i> , <i>genre</i>

Praproses Data

Praproses dilakukan dalam beberapa langkah berurutan. Pertama, kolom *name* dan *artist_main* pada berkas master dinormalisasi dengan operasi *lowercase*, penghapusan spasi awal atau akhir, dan *parsing* kolom *genre* dari format *string* ke *list* Python untuk mengekstraksi *primary genre*. Kedua, kolom *artistname* dan *trackname* pada berkas *user* juga dinormalisasi

dengan logika serupa. Ketiga, *play count* untuk setiap kombinasi *user-artist-track* dihitung menggunakan agregasi *groupby* sebagai sinyal *implicit feedback*.

Tahap penggabungan dataset dilakukan dengan *exact key matching* berbasis normalisasi nama artis, di mana kedua berkas distandarisasi melalui penghapusan tanda baca, karakter khusus, dan kata umum seperti *'feat'* serta *'ft'* sebelum dicocokkan. Karena ukuran dataset sangat besar, proses dijalankan dalam *chunk* berukuran 500.000 baris untuk menjaga stabilitas proses. Setelah penggabungan, duplikat *user-track* digabungkan dengan menjumlahkan *play count*, kemudian baris yang tidak memiliki padanan di master dieliminasi. Hasil akhirnya adalah dataset gabungan yang mempertahankan informasi interaksi sekaligus seluruh fitur audio lagu yang relevan.

Segmentasi Pengguna

Penelitian ini menerapkan ambang lima interaksi untuk memisahkan pengguna ke dalam dua segmen. Pengguna dengan minimal lima interaksi diklasifikasikan sebagai *warm user* dan diarahkan ke model CF, sedangkan pengguna dengan kurang dari lima interaksi disebut *cold user* dan diarahkan ke model berbasis konten (CBF). Pemilihan ambang ini didasarkan pada pertimbangan bahwa lima titik data dianggap minimum yang masih memungkinkan pola laten terbentuk pada *matrix factorization*. Hasil segmentasi menghasilkan 14.880 *warm user* dan 723 *cold user*.

Pemodelan Collaborative Filtering

Encoding dan Matriks Sparse

Setiap *user_id* dan *id* lagu pertama-tama dikonversi menjadi indeks *integer* terurut menggunakan *astype("category")* dan *cat.codes*. Konversi ini diperlukan karena pustaka *implicit* bekerja pada indeks *integer*, bukan *string*. Ukuran matriks *user-item* yang terbentuk adalah 14.880×15.502 , dimana sebagian besar kombinasi *user-lagu* tidak memiliki interaksi, sehingga matriks direpresentasikan dalam format *Compressed Sparse Row* (CSR) untuk menghemat memori.

Train-Test Split

Pemisahan *train-test split* dilakukan per pengguna dengan rasio 80:20. Untuk setiap pengguna, interaksi diurutkan berdasarkan *play count* secara menurun, kemudian 20% interaksi dengan *play count* terkecil dijadikan data uji. Logika ini dipilih karena lagu yang lebih jarang didengar cenderung lebih sulit diprediksi sehingga menghasilkan evaluasi yang lebih realistis dan tidak terlalu mudah. Hasil *split* akhir menghasilkan 6.843.981 baris *training* (80%) dan 1.703.611 baris *testing* (20%).

Pelatihan ALS dan SVD

Model ALS diimplementasikan menggunakan pustaka *implicit* dengan fungsi *Alternating Least Squares*. Konfigurasi awal yang menjadi *baseline* adalah *factors*=64, *regularization*=0.1, *alpha*=40, dan *iterations*=20. Model SVD diimplementasikan menggunakan fungsi *svds* dari *scipy.sparse.linalg* dengan *k*=64 sebagai jumlah faktor laten. Untuk SVD, *play count* terlebih dahulu ditransformasi *log1p* agar nilai *play count* yang ekstrim tidak mendominasi dekomposisi matriks. Setelah pelatihan, ALS menghasilkan *user factors* dan *item factors*, sedangkan SVD menghasilkan matriks U, sigma, dan Vt, yang keduanya dapat digunakan untuk menghitung skor prediksi pengguna terhadap setiap lagu. Berdasarkan hasil evaluasi, nilai *Precision@K*, *Recall@K*, dan *Normalized Discounted Cumulative Gain* (NDCG@K) pada metode ALS lebih baik dibanding model SVD.

Hyperparameter Tuning

Tuning ALS dilakukan dalam dua tahap. Pertama, *grid search* dijalankan dengan kombinasi *factors* $\in \{32, 64, 128, 256\}$, *alpha* $\in \{10, 40, 80\}$, dan *regularization* $\in \{0.01, 0.05, 0.1\}$, sementara *iterations* dipertahankan tetap pada nilai 20 untuk seluruh kombinasi. Total kombinasi yang diuji adalah 36 set parameter, masing-masing dievaluasi dengan NDCG@10 sebagai metrik utama karena lebih sensitif terhadap posisi item relevan dibandingkan *Precision* dan *Recall* yang tidak mempertimbangkan urutan. Kedua, setelah konfigurasi terbaik dari *grid search* ditemukan, dilakukan *tuning iterations* dengan nilai $\in \{20, 30, 50, 100\}$ untuk menentukan jumlah iterasi optimal pada parameter tersebut.

Pemodelan Content-Based Filtering untuk Cold User

Pembentukan Profil Item

Sepuluh fitur audio (*valence*, *energy*, *danceability*, *acousticness*, *instrumentalness*, *speechiness*, *liveness*, *loudness*, *tempo*, dan *popularity*) dinormalisasi ke rentang 0-1 menggunakan *MinMaxScaler*. Hasil normalisasi membentuk matriks profil item berukuran *jumlah_lagu_unik* \times 10 fitur audio yang digunakan sebagai basis perhitungan kemiripan antar lagu.

Pembentukan Profil Pengguna

Profil pengguna pada CBF dibangun melalui rata-rata tertimbang dari profil seluruh lagu yang pernah didengar, dengan *play count* digunakan sebagai bobot. Rumusnya dapat dituliskan sebagai $user_profile = \frac{\sum(profile_i \times play_count_i)}{\sum(play_count_i)}$, dimana *i* adalah indeks lagu yang pernah didengar pengguna. Pendekatan ini memberikan bobot lebih besar

pada lagu yang lebih sering didengar, sehingga profil mencerminkan preferensi dominan pengguna.

Perhitungan Rekomendasi

Cosine similarity dihitung antara profil pengguna dengan profil seluruh lagu kandidat (lagu yang belum pernah didengar). Sepuluh lagu dengan *similarity* tertinggi kemudian dikembalikan sebagai rekomendasi. Untuk variasi *hybrid CBF-popularity*, alokasi disesuaikan: pengguna dengan nol interaksi menerima *pure popularity*, pengguna dengan satu interaksi menerima 50% CBF dan 50% *popularity*, sedangkan pengguna dengan dua atau lebih interaksi menerima *pure CBF*. Skor popularitas dihitung berdasarkan jumlah pengguna unik yang pernah mendengar suatu lagu, kemudian dinormalisasi ke rentang 0-1.

Fungsi Pengarahan Adaptif

Setelah seluruh model dilatih, dibangun fungsi tunggal bernama *get_recommendation* yang secara otomatis pemilihan model dilakukan berdasarkan hasil segmentasi pengguna sebelumnya, yaitu membership pengguna pada kelompok *warm user* atau *cold user* yang dibentuk menggunakan ambang jumlah interaksi.

Logika pengarahan adalah sebagai berikut: jika *user_id* tidak dikenal sama sekali maka sistem mengembalikan *top-N* berdasarkan popularitas; jika dikenal sebagai *cold user* maka sistem menjalankan *hybrid CBF-popularity*; jika dikenal sebagai *warm user* maka sistem menjalankan ALS final. Fungsi pengarahan ini memastikan bahwa setiap pengguna selalu mendapatkan rekomendasi terbaik sesuai dengan ketersediaan data interaksinya.

Matrik Evaluasi

Tiga metrik utama digunakan untuk menilai kinerja sistem yaitu *Precision@10*, *Recall@10*, dan *NDCG@10*. *Precision@10* dihitung sebagai proporsi item relevan dalam sepuluh rekomendasi teratas dibagi sepuluh. *Recall@10* dihitung sebagai proporsi item relevan dalam sepuluh rekomendasi teratas dibagi jumlah item relevan dalam data uji pengguna tersebut. *NDCG@10* dihitung dengan menjumlahkan diskon logaritmik atas posisi item relevan, kemudian dinormalisasi terhadap ideal DCG.

Pengguna yang tidak memiliki item relevan pada data uji tidak disertakan dalam perhitungan metrik untuk menghindari pembagian nol dan bias evaluasi. Rumus dasarnya adalah $DCG@K = \sum rel_i / \log_2(i + 1)$ untuk $i = 1$ sampai K , sedangkan $NDCG@K = DCG@K / IDC@K$ (Krichene & Rendle, 2020).

4. HASIL DAN PEMBAHASAN

Hasil Praproses dan Segmentasi

Pemrosesan *dataset* gabungan menghasilkan *dataset* dengan tingkat kelengkapan fitur audio sebesar 100% (tidak ada missing value). Distribusi *play count* menunjukkan pola *long-tail* yang ekstrem: 6.706.020 baris memiliki *play count* = 1, 1.347.920 baris memiliki *play count* = 2, dan jumlah baris terus menurun secara eksponensial pada nilai *play count* yang lebih tinggi. Pola ini menegaskan karakteristik *implicit feedback* yang *sparse* dan didominasi interaksi tunggal. Setelah segmentasi, terdapat 14.880 *warm user* (95,4%) dan 723 *cold user* (4,6%) yang masuk ke dalam evaluasi.

Hasil Perbandingan antara *Baseline* ALS dan SVD

Model ALS *baseline* dengan parameter awal menghasilkan NDCG@10 sebesar 0,0859, Precision@10 sebesar 0,0759, dan Recall@10 sebesar 0,0410 pada 14.880 *warm user*. Sebagai pembanding, model SVD dengan $k = 64$ menghasilkan NDCG@10 sebesar 0,0715, Precision@10 sebesar 0,0638, dan Recall@10 sebesar 0,0254. Tabel 2 merangkum perbandingan kedua model.

Tabel 2. Perbandingan Kinerja ALS Baseline dan SVD.

Metrik	ALS	SVD	Pemenang
Precision@10	0,0759	0,0638	ALS
Recall@10	0,0410	0,0254	ALS
NDCG@10	0,0859	0,0715	ALS

ALS mengungguli SVD pada seluruh metrik. Keunggulan ini dapat dijelaskan oleh perbedaan fundamental dalam cara kedua model memperlakukan data *implicit feedback*. ALS secara eksplisit memodelkan tingkat kepercayaan terhadap preferensi positif melalui parameter *alpha* sesuai dengan He et al. (2016), sehingga *play count* yang tinggi diberi bobot keyakinan lebih besar. Sebaliknya, SVD menganggap seluruh entri matriks setara dan tidak membedakan antara preferensi yang kuat dan lemah seperti yang dikatakan Rendle et al. (2020), sehingga sinyal yang lebih halus pada data *play count* tidak tertangkap dengan baik. Hal ini sesuai dengan Rendle et al. (2022) yang menempatkan ALS sebagai *baseline* yang sangat kompetitif pada banyak *benchmark* sistem rekomendasi.

Hasil *Grid Search* ALS

Grid search atas 36 kombinasi hyperparameter menghasilkan konfigurasi terbaik $factors = 32$, $alpha = 10$, $regularization = 0,1$, dan $iterations = 20$ dengan NDCG@10 sebesar 0,1116. Tabel 3 menyajikan sepuluh kombinasi parameter teratas berdasarkan NDCG@10.

Tabel 3. Sepuluh Kombinasi Hyperparameter ALS Terbaik.

<i>Factors</i>	<i>Alpha</i>	<i>Reg.</i>	<i>Precision</i>	<i>Recall</i>	<i>NDCG</i>
32	10	0,10	0,0986	0,0446	0,1116
32	10	0,05	0,0984	0,0447	0,1115
32	10	0,01	0,0983	0,0445	0,1108
64	10	0,10	0,0922	0,0464	0,1052
64	10	0,05	0,0923	0,0463	0,1051
64	10	0,01	0,0908	0,0461	0,1038
128	10	0,10	0,0845	0,0462	0,0978
128	10	0,05	0,0844	0,0462	0,0974
128	10	0,01	0,0835	0,0458	0,0965
32	40	0,05	0,0847	0,0416	0,0946

Tiga pola menarik dapat diamati dari hasil *grid search*. Pertama, $\alpha = 10$ secara konsisten menghasilkan NDCG yang lebih tinggi dibandingkan $\alpha = 40$ atau 80. Hal ini menunjukkan bahwa pada *dataset* ini, sinyal *play count* tidak perlu diberi bobot kepercayaan yang terlalu agresif karena distribusinya sudah didominasi nilai-nilai kecil; α terlalu besar justru membuat model menjadi terlalu yakin pada interaksi tunggal dan kehilangan kemampuan generalisasi. Kedua, $factors = 32$ memberikan keseimbangan optimal antara kapasitas model dan risiko *overfitting*; $factors$ yang lebih besar (64, 128, 256) cenderung menurunkan NDCG, mengindikasikan bahwa kompleksitas model yang berlebihan justru menyerap *noise* dari data *sparse*. Ketiga, sensitivitas terhadap *regularization* relatif kecil pada rentang yang diuji, dengan $reg = 0,1$ memberikan hasil yang sedikit lebih baik.

Total peningkatan NDCG@10 dari baseline ke konfigurasi terbaik adalah +30%, dari 0,0859 menjadi 0,1116. Angka ini menunjukkan bahwa pemilihan *hyperparameter* yang tepat memiliki dampak yang substansial pada kinerja ALS. Tabel 4 merangkum perbandingan *baseline*, hasil *grid search*, dan konfigurasi final.

Tabel 4. Perjalanan Tuning ALS dari Baseline ke Final.

Tahap	<i>Precision</i>	<i>Recall</i>	<i>NDCG</i>
<i>Baseline (pre-tuning)</i>	0,0759	0,0410	0,0859
<i>Best grid search</i>	0,0986	0,0446	0,1116
<i>Final (f=32, $\alpha=10$, i=20)</i>	0,0986	0,0446	0,1116

Contoh Rekomendasi untuk *Warm User*

Untuk menggambarkan kualitas rekomendasi secara kualitatif, sistem dijalankan pada pengguna sampel dengan 98 lagu interaksi. Hasil top-10 rekomendasi ALS final ditampilkan pada Tabel 5.

Lagu-lagu yang direkomendasikan didominasi oleh *genre pop-rock* dan *indie* yang serupa dengan pola interaksi pengguna tersebut, dengan skor *confidence* tertinggi pada lagu "No Good in Goodbye" dari The Script. Daftar tersebut menunjukkan bahwa ALS *final* mampu menangkap pola laten preferensi pengguna dan menghasilkan rekomendasi yang variatif sekaligus tematik.

Tabel 5. Contoh Top-10 Rekomendasi ALS Final untuk *Warm User*.

No	Track	Artist	Score
1	<i>No Good in Goodbye</i>	The Script	0,7986
2	<i>Immortals</i>	Fall Out Boy	0,7674
3	<i>Amsterdam</i>	Imagine Dragons	0,7562
4	<i>Let Her Go</i>	Passenger	0,7504
5	<i>Lay Me Down</i>	Sam Smith	0,7478
6	<i>Of The Night</i>	Bastille	0,7264
7	<i>Butterfly Fly Away</i>	Miley Cyrus	0,7171
8	<i>The Only Exception</i>	Paramore	0,7165
9	<i>Glory And Gore</i>	Lorde	0,7049
10	<i>Say Something</i>	A Great Big World	0,6982

Hasil CBF dan Hybrid CBF–Popularity pada *Cold User*

Evaluasi pada 723 *cold user* dilakukan dengan dua skenario. Skenario pertama menggunakan CBF murni, sedangkan skenario kedua menggunakan hybrid CBF–popularity. Hasil komparasi disajikan pada Tabel 6.

Tabel 6. Perbandingan CBF Murni dan Hybrid CBF–Popularity pada *Cold User*.

Metrik	CBF Murni	Hybrid	delta
<i>Precision@10</i>	0,0017	0,0043	+0,0026
<i>Recall@10</i>	0,0166	0,0429	+0,0263
<i>NDCG@10</i>	0,0070	0,0201	+0,0131

Hybrid CBF–popularity menghasilkan peningkatan signifikan pada seluruh metrik: *Precision@10* meningkat sekitar 2,5 kali lipat, *Recall@10* meningkat sekitar 2,6 kali lipat, dan *NDCG@10* meningkat sekitar 2,9 kali lipat dibandingkan CBF murni. Peningkatan ini menegaskan bahwa pada kondisi data yang sangat terbatas, *popularity* berperan sebagai *prior* yang efektif karena lagu-lagu populer memang lebih mungkin disukai oleh pengguna acak (Ferrari Dacrema et al., 2019). CBF murni sendiri tetap diperlukan agar rekomendasi tidak sepenuhnya kehilangan aspek personalisasi.

Walaupun demikian, nilai *absolut* metrik pada *cold user* masih tergolong rendah dibandingkan *warm user*. Hal ini dikarenakan *cold user* pada dataset hanya memiliki rata-rata satu hingga dua interaksi unik, sehingga sinyal preferensi sangat tipis dan presisi tinggi sulit dicapai. Menurut Briand et al. (2021) di aplikasi Deezer menunjukkan bahwa kondisi *cold user* pada platform musik memang sulit dipecahkan dengan baik tanpa data tambahan berupa informasi demografi atau konteks penggunaan.

Validasi Fungsi Pengarahan Adaptif

Untuk menguji *robustness* fungsi pengarahan, dilakukan tiga skenario validasi yang mewakili tiga-tipe pengguna yang berbeda. Skenario pertama menggunakan *warm user* dengan 98 interaksi; sistem secara otomatis memilih ALS dan menghasilkan rekomendasi seperti pada Tabel 5. Skenario kedua menggunakan *cold user* dengan satu interaksi (lagu Gang Starr); sistem mengarahkan ke *hybrid CBF–popularity* dan menghasilkan kombinasi lagu *hip-hop* dari komponen CBF dan lagu populer dari komponen *popularity*. Skenario ketiga menggunakan *user_id* yang tidak ada di dataset; sistem mengembalikan top-10 lagu paling populer seperti "Atlas" oleh Coldplay dan "Around The World" oleh Daft Punk.

Ketiga skenario membuktikan bahwa fungsi pengarahan adaptif mampu menentukan strategi yang sesuai tanpa intervensi manual. Hal ini sesuai dengan Çano & Morisio (2017) bahwa dengan prinsip *hybrid recommender* khususnya pola *switching*, dimana sistem memilih satu metode rekomendasi berdasarkan kondisi konteks. Implementasi seperti ini sangat relevan untuk lingkungan produksi karena memberikan fleksibilitas untuk menangani heterogenitas basis pengguna.

Implikasi Teoritis dan Praktis

Berdasarkan hasil penelitian ini, memperkuat bukti bahwa pemilihan model ALS lebih cocok untuk *implicit feedback* seperti *play count* dibanding SVD, dan didukung pemilihan *alpha* pada nilai yang relatif rendah. Penelitian ini juga menambah bukti empiris bahwa kombinasi CBF dengan *popularity* dapat mengurangi degradasi kinerja pada *cold user*, melengkapi penelitian Magron & Févotte (2022) yang membahas *neural content-aware collaborative filtering* dan peran fitur audio pada *cold-start*. Menurut Steck (2019) model linear sederhana seringkali kompetitif dengan pendekatan yang jauh lebih kompleks pada banyak benchmark sistem rekomendasi.

Secara praktis, arsitektur tiga-jalur (ALS untuk *warm*, *hybrid CBF–popularity* untuk *cold*, dan *popularity* murni untuk pengguna baru) memberikan blueprint yang dapat diadopsi oleh platform musik berbahasa Indonesia atau platform lokal yang ingin mengembangkan sistem rekomendasi sendiri.

Ukuran model artefak yang relatif kecil, ALS sekitar 7,4 MB, artifacts pendukung 2,0 MB, dan train matrix 4,2 MB, memungkinkan deployment pada infrastruktur yang ringan, misalnya melalui aplikasi Streamlit atau API *service* kecil. Pendekatan ini juga lebih efisien dibandingkan model deep learning kompleks yang dibahas Zhang et al., (2020), sehingga lebih sesuai untuk konteks dengan keterbatasan komputasi.

Keterbatasan

Penelitian ini memiliki keterbatasan yang perlu diperhatikan. Pertama, dataset yang digunakan tidak mengandung informasi demografi atau konteks penggunaan, sehingga rekomendasi sepenuhnya bergantung pada riwayat interaksi dan fitur audio. Kedua, evaluasi menggunakan strategi *offline split* yang mengasumsikan data uji bersifat statis; pada lingkungan nyata, perilaku pengguna dinamis dan dapat berubah dari waktu ke waktu. Ketiga, jumlah *cold user* yang relatif kecil (723 orang) membatasi presisi estimasi metrik untuk segmen tersebut.

5. KESIMPULAN DAN SARAN

Penelitian ini berhasil membangun sistem rekomendasi musik Spotify yang adaptif dengan menggabungkan *Alternating Least Squares* (ALS) untuk *warm user*, *hybrid content-based filtering* dan *popularity* untuk *cold user*, serta *popularity* murni untuk pengguna baru. Pada data warm user, ALS dengan konfigurasi *factors* = 32, *alpha* = 10, *regularization* = 0,1, dan *iterations* = 20 menghasilkan NDCG@10 sebesar 0,1116 atau meningkat 30% dibanding *baseline*, sekaligus mengungguli SVD yang hanya mencapai NDCG@10 sebesar 0,0715. Pada data *cold user*, *hybrid CBF-popularity* meningkatkan NDCG@10 dari 0,0070 menjadi 0,0201 atau sekitar 2,9 kali lipat dibandingkan CBF murni. Validasi tiga skenario pengguna menegaskan bahwa fungsi pengarah adaptif mampu memilih strategi yang tepat tanpa intervensi manual, menjadikan sistem siap diintegrasikan ke lingkungan produksi.

Berdasarkan penelitian tersebut, beberapa saran dapat diberikan untuk penelitian selanjutnya. Pertama, integrasi sinyal kontekstual seperti waktu putar, perangkat, dan lokasi geografis dapat meningkatkan relevansi rekomendasi pada *cold user*. Kedua, pendekatan *deep learning* seperti *Neural Collaborative Filtering* atau model berbasis transformer dapat dibandingkan dengan ALS untuk menilai potensi peningkatan akurasi. Ketiga, evaluasi *online* melalui *A/B testing* diperlukan untuk memvalidasi apakah peningkatan metrik *offline* benar-benar berdampak pada perilaku pengguna nyata. Keempat, eksplorasi strategi *cold-start* lanjutan seperti meta-learning atau side information ringan (*genre* favorit, *mood* preferensi) dapat menjadi arah pengembangan yang menjanjikan.

DAFTAR REFERENSI

- Aggarwal, C. C. (2016). *Recommender Systems*. Springer International Publishing. <https://doi.org/10.1007/978-3-319-29659-3>
- Anggoro, M. V., & Izzatillah, M. (2022). Sistem Rekomendasi Musik dengan Metode Collaborative Filtering Berbasis Android. *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)*, 7(1), 1. <https://doi.org/10.30998/string.v7i1.10300>
- Bobadilla, J., Alonso, S., & Hernando, A. (2020). Deep Learning Architecture for Collaborative Filtering Recommender Systems. *Applied Sciences*, 10(7), 2441. <https://doi.org/10.3390/app10072441>
- Briand, L., Salha-Galvan, G., Bendada, W., Morlon, M., & Tran, V.-A. (2021). A Semi-Personalized System for User Cold Start Recommendation on Music Streaming Apps. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2601–2609. <https://doi.org/10.1145/3447548.3467110>
- Cañamares, R., & Castells, P. (2020). On Target Item Sampling in Offline Recommender System Evaluation. *Fourteenth ACM Conference on Recommender Systems*, 259–268. <https://doi.org/10.1145/3383313.3412259>
- Çano, E., & Morisio, M. (2017). Hybrid recommender systems: A systematic literature review. *Intelligent Data Analysis*, 21(6), 1487–1524. <https://doi.org/10.3233/IDA-163209>
- Ferrari Dacrema, M., Cremonesi, P., & Jannach, D. (2019). Are we really making much progress? A worrying analysis of recent neural recommendation approaches. *Proceedings of the 13th ACM Conference on Recommender Systems*, 101–109. <https://doi.org/10.1145/3298689.3347058>
- Gomez-Uribe, C. A., & Hunt, N. (2016). The Netflix Recommender System: Algorithms, Business Value, and Innovation. *ACM Transactions on Management Information Systems*, 6(4), 1–19. <https://doi.org/10.1145/2843948>
- He, X., Zhang, H., Kan, M.-Y., & Chua, T.-S. (2016). Fast Matrix Factorization for Online Recommendation with Implicit Feedback. *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 549–558. <https://doi.org/10.1145/2911451.2911489>
- Krichene, W., & Rendle, S. (2020). On Sampled Metrics for Item Recommendation. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1748–1757. <https://doi.org/10.1145/3394486.3403226>
- Magron, P., & Févotte, C. (2022). Neural content-aware collaborative filtering for cold-start music recommendation. *Data Mining and Knowledge Discovery*, 36(5), 1971–2005. <https://doi.org/10.1007/s10618-022-00859-8>
- Panda, D. K., & Ray, S. (2022). Approaches and algorithms to mitigate cold start problems in recommender systems: A systematic literature review. *Journal of Intelligent Information Systems*, 59(2), 341–366. <https://doi.org/10.1007/s10844-022-00698-5>
- Pérez-Almaguer, Y., Yera, R., Alzahrani, A. A., & Martínez, L. (2021). Content-based group recommender systems: A general taxonomy and further improvements. *Expert Systems with Applications*, 184, 115444. <https://doi.org/10.1016/j.eswa.2021.115444>
- Putra, A. I., & Santika, R. R. (2020). Implementasi Machine Learning dalam Penentuan Rekomendasi Musik dengan Metode Content-Based Filtering. *Edumatic: Jurnal Pendidikan Informatika*, 4(1), 121–130. <https://doi.org/10.29408/edumatic.v4i1.2162>

- Rendle, S., Krichene, W., Zhang, L., & Anderson, J. (2020). Neural Collaborative Filtering vs. Matrix Factorization Revisited. *Fourteenth ACM Conference on Recommender Systems*, 240–248. <https://doi.org/10.1145/3383313.3412488>
- Rendle, S., Krichene, W., Zhang, L., & Koren, Y. (2022). Revisiting the Performance of iALS on Item Recommendation Benchmarks. *Proceedings of the 16th ACM Conference on Recommender Systems*, 427–435. <https://doi.org/10.1145/3523227.3548486>
- Ricci, F., Rokach, L., & Shapira, B. (Eds.). (2022). *Recommender Systems Handbook*. Springer US. <https://doi.org/10.1007/978-1-0716-2197-4>
- Schedl, M. (2019). Deep Learning in Music Recommendation Systems. *Frontiers in Applied Mathematics and Statistics*, 5, 44. <https://doi.org/10.3389/fams.2019.00044>
- Steck, H. (2019). Embarrassingly Shallow Autoencoders for Sparse Data. *The World Wide Web Conference*, 3251–3257. <https://doi.org/10.1145/3308558.3313710>
- Vall, A., Dorfer, M., Eghbal-zadeh, H., Schedl, M., Burjorjee, K., & Widmer, G. (2019). Feature-combination hybrid recommender systems for automated music playlist continuation. *User Modeling and User-Adapted Interaction*, 29(2), 527–572. <https://doi.org/10.1007/s11257-018-9215-8>
- Yuniardini, F., & Widiyaningtyas, T. (2024). Analisis Perbandingan Pearson Correlation dan Cosine Similarity pada Rekomendasi Musik berbasis Collaborative Filtering. *Edumatic: Jurnal Pendidikan Informatika*, 8(2), 555–564. <https://doi.org/10.29408/edumatic.v8i2.27781>
- Zhang, S., Yao, L., Sun, A., & Tay, Y. (2020). Deep Learning Based Recommender System: A Survey and New Perspectives. *ACM Computing Surveys*, 52(1), 1–38. <https://doi.org/10.1145/3285029>