Indonesian Journal of Multidisciplinary on Social and Technology
Homepage: https://journal.ilmudata.co.id/index.php/ijmst
Vol. 4 No. 3 (2026) pp: 2756-2765
P-ISSN: 2986-6790, e-ISSN: 2986-6782

Implementasi Metode *Clustering* dengan Algoritma *K-Means* pada Sistem Rekomendasi Lagu berbasis Similaritas Fitur Audio

Naufal Ammanullah Arrasyid Yusuf[1], Ilham Saifudin[2*], Ari Eko Wardoyo[3]
[123]Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Jember
rasyidnaufal09@gmail.com, ilham.saifudin@unmuhjember.ac.id*, arieko@unmuhjember.ac.id

Abstract
Music recommender systems based on Collaborative Filtering face a fundamental limitation known as the cold-start problem when dealing with new songs or users with no prior interaction history. This study implements a Content-Based Filtering (CBF) approach using the K-Means Clustering algorithm based on audio feature extraction to address this limitation. The dataset consists of 3,200 audio files in MP3, WAV, AAC, and M4A formats, collected from local storage, covering six popular genres: Pop, Rock, Jazz, Electronic/EDM, Hip-Hop/Rap, and Classical. Audio features were extracted using the Librosa library, producing numerical representations, including Mel-Frequency Cepstral Coefficients (MFCCs), Spectral Centroid, Zero-Crossing Rate (ZCR), RMS Energy, Spectral Rolloff, Chroma STFT, Tempo, and Beat Count. Following Z-Score normalization to standardize feature scales, the K-Means Clustering algorithm was applied and evaluated using Within-Cluster Sum of Squares (WCSS) and the Calinski-Harabasz (CH) Index, yielding an optimal partition at k=6 that aligns with the six genre representations in the dataset, with a WCSS value of 95,616.62 and a CH Index of 85.92. The recommender system produces Top-N outputs using two distance metrics, namely Euclidean Distance and Cosine Similarity, evaluated through Overlap Coefficient, Jaccard Similarity, and Intra-List Similarity (ILS). Evaluation results demonstrate absolute convergence between both methods at Top-3 with a Jaccard Similarity of 1.0. At the same time, the average ILS score remains stable around 0.5, indicating an optimal balance between recommender relevance and diversity. The system is implemented as a Flask-based web application featuring Venn Diagram and Heatmap visualizations to support interactive evaluation for users.

Keywords: Music Recommender System, K-Means, Audio Feature Extraction, Intra-List Similarity, Jaccard Similarity.

Abstrak
Sistem rekomendasi lagu berbasis Collaborative Filtering menghadapi keterbatasan fundamental berupa cold-start problem ketika berhadapan dengan lagu atau pengguna baru yang belum memiliki riwayat interaksi. Penelitian ini mengimplementasikan pendekatan Content-Based Filtering (CBF) menggunakan algoritma K-Means Clustering berbasis ekstraksi fitur audio untuk mengatasi permasalahan tersebut. Dataset yang digunakan terdiri dari 3.200 file lagu dalam format mp3, wav, aac, dan m4a yang dikumpulkan dari penyimpanan lokal, mencakup enam genre populer, yaitu Pop, Rock, Jazz, Electronic/EDM, Hip-Hop/Rap, dan Classical. Fitur audio diekstraksi menggunakan library Librosa, menghasilkan representasi numerik berupa Mel-Frequency Cepstral Coefficients (MFCC), Spectral Centroid, Zero Crossing Rate (ZCR), RMS Energy, Spectral Rolloff, Chroma STFT, Tempo, dan Beat Count. Setelah normalisasi Z-Score untuk menyeragamkan skala fitur, algoritma K-Means Clustering diterapkan dan dievaluasi menggunakan Within-Cluster Sum of Squares (WCSS) dan Calinski-Harabasz (CH) Index, menghasilkan partisi optimal pada k=6 yang selaras dengan representasi enam genre dalam dataset, dengan nilai WCSS 95,616,62 dan CH Index 85,92. Sistem rekomendasi menghasilkan output Top-N menggunakan dua metrik jarak, yaitu Euclidean Distance dan Cosine Similarity, yang dievaluasi menggunakan Overlap Coefficient, Jaccard Similarity, dan Intra-List Similarity (ILS). Hasil evaluasi menunjukkan konvergensi mutlak antara kedua metode pada Top-3 dengan nilai Jaccard Similarity 1,0, sementara rata-rata skor ILS stabil pada rentang 0,5 yang mengindikasikan keseimbangan optimal antara relevansi dan diversitas rekomendasi. Sistem diimplementasikan dalam aplikasi web berbasis Flask dengan visualisasi Diagram Venn dan heatmap sebagai pendukung evaluasi interaktif.

Kata Kunci: Sistem Rekomendasi Lagu, K-Means, Ekstraksi Audio, Intra-List Similarity, Jaccard Similarity.

1. Latar Belakang

Lagu merupakan aspek yang sangat penting bagi manusia, dan cara kita berinteraksi dengannya juga sangat memengaruhi kehidupan kita secara keseluruhan. Oleh karena itu, mempelajari teori lagu dan cara implementasinya bertujuan untuk mengembangkan, mempelajari, dan menerapkan teori tersebut dalam kehidupan manusia (Yuniar et al., 2022). Perkembangan dan popularitas *platform streaming* musik seperti *Spotify, Youtube Music, Apple Music, Deezer, Soundcloud, Joox,* dan beberapa *platform* musik lainnya telah mengubah cara orang mengakses musik. Meskipun memberikan akses tak terbatas ke berbagai jenis atau genre musik, pengguna sering mengalami kesulitan menemukan lagu yang sesuai dengan preferensi mereka (Pake et al., 2023).

Platformisasi telah menggeser *curatorial power* dari *human gatekeepers* (*radio DJs, music journalists*) ke *algorithmic gatekeeping* yang *opaque*, di mana produser elektronik independen tidak memiliki akses atau pemahaman tentang mengapa musik mereka tidak direkomendasikan meskipun mendapat *engagement* tinggi dari *dedicated niche audience* (Prey, 2020). Dua tahun kemudian, permasalahan tersebut kembali diliput, menyatakan bahwa situasi ini menciptakan *structural advantage* untuk produser yang berafiliasi dengan industri *mainstream*, salah satunya seperti *ghost producers* untuk grup ternama dari media *K-Pop, J-Pop,* maupun *media genre* musik sejenisnya, dibandingkan dengan *major label Electronic Dance Music (EDM)*, yang mendapat *algorithmic boost factor* hingga 15-20x dibandingkan *underground counterparts* mereka karena *volume initial engagement* yang dimediasi oleh *fanbase* dan infrastruktur promosi dari *label* musik maupun *channel platform* resmi (Prey et al., 2022).

Penelitian ini berfokus pada penerapan metode berbasis konten pada sistem rekomendasi yang ditujukan. Dengan tujuan pendekatan *multimodal fusion* yang berfungsi sebagai pemecah masalah dan mengoptimalkan hasil rekomendasi [5], pengelompokan menggunakan K-Means menawarkan fleksibilitas yang lebih besar meskipun biasanya kurang akurat (Nurhalizah et al., 2024).

K-Means memberikan solusi yang dapat diperluas dan lebih efektif dalam mengelompokkan data dengan cara yang lebih baik, terutama untuk dataset berdimensi tinggi dan jumlah data yang besar. Algoritma ini lebih unggul ketika ukuran data semakin besar seiring dengan meningkatnya jumlah dimensi (Gul & Rehman, 2023). Misalnya, algoritma *clustering* mengelompokkan data berdasarkan tingkat kesamaan tanpa pengetahuan sebelumnya tentang kategori. Algoritma ini bertujuan untuk memaksimalkan kesamaan dalam klaster dan perbedaan antar klaster (Valkenborg et al., 2023). Beberapa library Python, seperti scikit-learn, memudahkan implementasi algoritma tak terawasi, seperti *K-Means* untuk *clustering*. Hal ini memfasilitasi penerapan praktis dalam klasifikasi dan eksplorasi data (Retnoningsih & Pramudita, 2020).

Sistem rekomendasi adalah sistem yang membantu pengguna mengatasi informasi yang meluap dengan memberikan rekomendasi yang spesifik, dan diharapkan rekomendasi tersebut dapat memenuhi keinginan serta kebutuhan pengguna (Visher Laja Jaja et al., 2020). Sistem rekomendasi musik merupakan aplikasi khusus dari information filtering system yang bertujuan membantu pengguna menemukan konten musik yang relevan di tengah kelimpahan pilihan yang tersedia pada platform streaming modern (Schedl et al., 2022). Dengan mengartikan sebagai sistem yang secara otomatis mengidentifikasi item musik yang berpotensi disukai pengguna berdasarkan berbagai sinyal, mulai dari preferensi eksplisit hingga karakteristik konten audio itu sendiri.Pendekatan sistem rekomendasi musik dikelompokkan ke dalam tiga kategori utama: Collaborative Filtering yang bergantung pada data interaksi antarpengguna, Content-Based Filtering yang menganalisis karakteristik intrinsik konten, dan Hybrid Approach yang mengombinasikan keduanya.

Berbagai pendekatan telah dikembangkan untuk mengoptimalkan sistem rekomendasi, salah satunya melalui pendekatan *Collaborative Filtering* (CF). Seperti pada implementasi metode SVD-GSetRank, sebuah pendekatan *collaborative filtering berbasis memori* yang mengintegrasikan *Matrix Factorization* (SVD) dengan teknik pemeringkatan *Gower's Set Rank*. Pendekatan tersebut terbukti secara komputasi mampu menyeimbangkan efisiensi dan meningkatkan akurasi pemeringkatan (*ranking*) secara signifikan pada *dataset* publik. Namun demikian, apabila pendekatan CF yang berorientasi pada matriks *rating* tersebut diimplementasikan secara langsung dalam spesifikasi sistem rekomendasi lagu, metode ini akan terbentur masalah fundamental, yaitu *Cold Start Problem*. Sistem rekomendasi yang murni bergantung pada riwayat interaksi atau *rating* pengguna tidak memiliki kapabilitas untuk merekomendasikan karya musisi *indie* atau lagu-lagu baru yang belum pernah diputar oleh pendengar sebelumnya. Ketergantungan pada *metadata* interaksi ini berpotensi menciptakan ketimpangan, di mana sistem hanya akan merotasi lagu-lagu populer (Widiyaningtyas et al., 2025).

Untuk menjembatani celah (*gap*) tersebut, pendekatan *Content-Based Filtering* (CBF) diusulkan sebagai arsitektur yang paling relevan untuk domain rekomendasi musik. Alih-alih bergantung pada matriks kolaboratif antarpengguna, CBF beroperasi dengan membedah representasi intrinsik *item*. Dalam konteks spesifik audio, sistem CBF tidak lagi membutuhkan *metadata* tekstual (seperti *genre* atau label rekaman), melainkan mengekstraksi fitur akustik tingkat rendah (*low-level features*) secara langsung dari sinyal audio mentah.

Penerapan *Content-Based Filtering* memungkinkan setiap *file* lagu diekstraksi menjadi representasi matematis, seperti *Mel-Frequency Cepstral Coefficients* (MFCC), *Spectral Centroid*, dan *Zero Crossing Rate*. Dengan mengevaluasi karakteristik gelombang suara, tingkat kebisingan, dan tempo secara objektif, algoritma klasterisasi dan perhitungan jarak (seperti *Euclidean* maupun *Cosine*) dapat menemukan ketetanggaan antarlagu dengan presisi yang sangat tinggi. Pendekatan otonom ini menjamin bahwa setiap lagu, terlepas dari popularitas atau ketiadaan *rating*, memiliki peluang yang setara dan adil untuk direkomendasikan semata-mata berdasarkan kecocokan tekstur instrumen dan suasana (*mood*) vokalnya.

Keunggulan utama CBF dibandingkan dengan collaborative filtering terletak pada kemampuannya mengatasi masalah *cold-start*. CBF dapat mendemonstrasikan bahwa sistem rekomendasi berbasis fitur musik intrinsik (tempo, key, mode, duration, loudness) yang diekstraksi menggunakan Librosa mampu mencapai akurasi 76,75% dalam memprediksi preferensi pengguna tanpa memerlukan riwayat interaksi apa pun (Srivastava, 2024). Selain

DOI: https://doi.org/10.69693/ijmst.v4i2.12785

itu, mengombinasikan CBF dengan Principal Component Analysis (PCA) untuk reduksi dimensi sebelum K-Means Clustering menghasilkan pengelompokan lagu yang konsisten dan interpretable berdasarkan karakteristik musikal intrinsik. (Xinyue Li, 2024). Efektivitas CBF menggunakan 11 fitur audio Spotify dengan K-Means Clustering menghasilkan silhouette score 0.68, menunjukkan pemisahan cluster yang cukup baik untuk keperluan rekomendasi berbasis similaritas konten (Oh et al., 2023).

Penentuan jumlah cluster k yang optimal merupakan salah satu tantangan utama dalam implementasi K-Means. Dalam penelitian ini, pemilihan k=6 tidak hanya didasarkan pada metrik evaluasi statistik semata, melainkan juga memiliki justifikasi musikal yang kuat berdasarkan representasi 6 genre populer dalam dataset, yaitu Pop, Rock, Jazz, Electronic/EDM, Hip-Hop/Rap, dan Classical. Pendekatan ini sejalan dengan pendekatan berbasis genre yang mengorganisir lagu berdasarkan kemiripan akustik, yang mengidentifikasi bahwa genre-genre tersebut memiliki profil audio yang khas dan dapat dibedakan secara komputasional (Chen et al., 2024).

Dalam sistem rekomendasi berbasis konten, perhitungan jarak atau kemiripan antarvektor fitur audio merupakan komponen inti yang menentukan kualitas rekomendasi. Penelitian ini mengimplementasikan dua metrik jarak yang memiliki karakteristik berbeda namun komplementer.

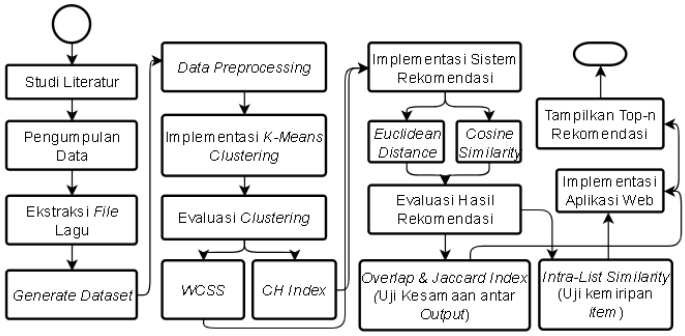
Euclidean Distance mengukur jarak geometris secara langsung antara dua vektor dalam feature space, dengan menghitung perbedaan absolut pada setiap dimensi fitur. Euclidean Distance sensitif terhadap perbedaan magnitude pada setiap dimensi, sehingga efektif untuk mengidentifikasi lagu yang secara absolut paling mirip dalam semua aspek fitur audio secara bersamaan. Kelemahan utamanya adalah susceptibilitas terhadap high-dimensional feature spaces yang memerlukan normalisasi yang cermat untuk mencegah dominasi fitur berskala besar (Mukhopadhyay et al., 2024).

Cosine Similarity mengukur kemiripan arah antara dua vektor dalam feature space dengan menghitung cosine dari sudut antara keduanya, dengan mengabaikan magnitude dan berfokus pada proporsi atau pola distribusi fitur. Cosine Similarity sangat efektif untuk music recommendation karena kemampuannya menangkap pola relatif dari fitur audio — misalnya, dua lagu yang sama-sama memiliki emphasis kuat pada certain frequency ranges akan dianggap similar oleh Cosine meskipun absolute magnitude-nya berbeda (Bevec et al., 2024). Efektivitas Cosine Similarity dalam content-based music filtering mencapai precision 82%.

Penggunaan kedua metrik secara paralel dalam penelitian ini memungkinkan analisis komparatif yang komprehensif tentang bagaimana perbedaan perspektif spasial (Euclidean) dan directional (Cosine) dalam mengukur similarity menghasilkan rekomendasi yang berbeda, yang dievaluasi melalui Overlap Coefficient, Jaccard Similarity, dan Intra-List Similarity.

2. Metode Penelitian

Penelitian ini menggunakan pendekatan sistematis yang terdiri dari beberapa tahapan utama sebagaimana ditunjukkan pada **Gambar 2.1**. Tahap pertama dimulai dengan identifikasi masalah dan studi literatur untuk membangun landasan teoretis. Selanjutnya dilakukan pengumpulan data lagu yang kemudian diekstraksi fitur-fitur audionya menggunakan Librosa untuk menghasilkan dataset representatif sebagai bahan utama dalam pengembangan sistem.



Gambar 2.1 Flowchart Penelitian

Tahap kedua merupakan fase *data preprocessing*, di mana *dataset* melalui proses normalisasi untuk menyeragamkan skala fitur. Setelah *preprocessing*,*K-Means Clustering* diimplementasikan untuk mengelompokkan lagu berdasarkan kemiripan fitur *audio* secara *unsupervised*. Kualitas *clustering* kemudian dievaluasi menggunakan metrik *Calinski-Harabasz Index* dan *Within-Cluster Sum of Squares* untuk memastikan pemisahan *cluster* yang optimal.

Tahapan selanjutnya yaitu menampilkan hasil *output Top-n* pada sistem rekomendasi dengan dua jenis *output*, yaitu *Euclidean Distance* dan *Cosine Similarity*, disertai pengujian kebenaran setiap output yang ditampilkan melalui uji validitas dengan metrik evaluasi seperti *Overlap* dan indeks *Jaccard* untuk menguji kesamaan antara

DOI: https://doi.org/10.69693/ijmst.v4i2.12785

kedua *output* rekomendasi yang digunakan, serta pengujian rerata kemiripan setiap peringkat *Top-n* rekomendasi dengan *Intra-List Similarity*.

Tahap akhir adalah implementasi sistem dalam bentuk aplikasi web interaktif. Pengguna dapat memilih atau mengunggah lagu sebagai *input query*, kemudian sistem akan mengekstraksi fitur *audio* dari lagu tersebut, mencocokkannya dengan *cluster* yang sesuai, dan menampilkan *top-n* rekomendasi lagu teratas yang memiliki karakteristik *audio* serupa berdasarkan perhitungan *euclidean distance* dan *cosine similarity* dalam *cluster* yang sama melalui beberapa metrik evaluasi yang digunakan seperti *intra-list similarity* untuk menghitung jarak antar kemiripan metrik peringkat dari atas hingga bawah, *overlap* dan indeks *Jaccard similarity* untuk menghitung kesamaan antar kedua *output* dari *euclidean* maupun dari *cosine*. Implementasi ini memberikan pengalaman penemuan lagu yang personal dan efisien tanpa bergantung pada data historis interaksi pengguna.

2.1 Dataset

Penelitian ini membutuhkan sejumlah kumpulan file lagu yang tersimpan dalam penyimpanan lokal atau perangkat personal, yang berjumlah sekitar 3.200 file dan memiliki ekstensi mp3, wav, aac, m4a, dan sejenisnya. Setelah itu, dikumpulkan menjadi satu folder dan diekstrak menggunakan program Python dengan library khusus, Librosa. Librosa menyediakan berbagai fungsi umum yang sering digunakan dalam analisis audio dan musik, seperti ekstraksi fitur, representasi spektrogram, pemisahan sumber harmonik dan perkusi, serta pemrosesan sinyal audio pada domain waktu dan domain frekuensi.Setelah dilakukan ekstraksi audio, hasil tersebut digenerasi menjadi file tabel dengan format ekstensi comma separated values (CSV), yang mana merupakan siap dipakai untuk pra-pemrosesan data pada machine learning

Dataset yang dikumpulkan mencakup enam genre populer sebagai representasi keragaman karakteristik musikal, yaitu Pop, Rock, Jazz, Electronic/EDM, Hip-Hop/Rap, dan Classical, dengan distribusi antargenre yang relatif seimbang. Pemilihan keenam genre ini memiliki kesesuaian langsung dengan justifikasi pemilihan k=6 pada algoritma K-Means Clustering, di mana setiap cluster diharapkan merepresentasikan satu kelompok genre dengan karakteristik audio yang khas dan dapat dibedakan secara komputasional.

2.2 Ekstraksi Lagu

Sebelum melakukan tahapan analisis dan penelitian, hal pertama yang perlu dilakukan adalah mengumpulkan terlebih dahulu *file* atau kumpulan koleksi lagu yang tersimpan di penyimpanan lokal atau media awan, kemudian mengumpulkan semuanya ke dalam satu *folder* untuk mengekstraksi fitur-fiturnya. Fungsi ini bertujuan untuk mengekstraksi dan menganalisis fitur elemen instrumental pada setiap lagu. Sebagai rincian alur tahapan pengumpulan data, akan diilustrasikan dalam bentuk *flowchart* pada **Gambar 2.2** berikut.

Gambar 2.2 Proses Ekstraksi Lagu

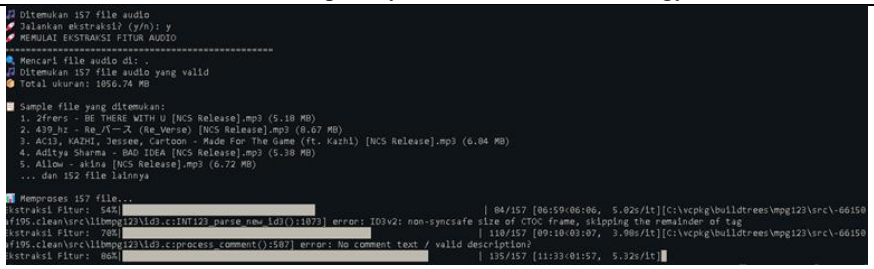
Tahapan paling awal sebelum melakukan analisis adalah menghimpun koleksi *file* lagu yang dapat diambil dari penyimpanan lokal, koleksi *album original,* maupun media penyimpanan awan (*cloud storage*). Kemudian, seluruh *file* lagu yang telah terkumpul dikonsolidasikan dan disimpan dalam satu folder khusus. Langkah ini dilakukan untuk memudahkan *program* melacak dan membaca seluruh populasi data secara otomatis.Setelah *folder/direktori* siap, program ekstraksi Python dijalankan. Eksekusi skrip ini dilakukan melalui antarmuka baris perintah seperti *Terminal* pada *macOS/Linux* atau *PowerShell pada Windows*.

Sistem mulai membaca *file* di dalam *folder* dan mengekstraksi fitur elemen instrumental dari setiap lagu secara berurutan (satu per satu). Proses komputasi ini akan mengurai sinyal audio mentah menjadi representasi fitur numerik

Setelah seluruh populasi lagu berhasil diekstraksi, program akan mengemas keluaran data numerik tersebut ke dalam format *Comma Separated Values* (*.csv*). Format ini dipilih karena ringan dan *universal* di dunia *Machine Learning*.

File *dataset .csv* yang telah terbentuk menandakan bahwa proses ekstraksi telah selesai. Data matriks numerikal tersebut kini siap untuk diumpankan ke tahapan pra-pemrosesan lanjutan dan pemodelan *machine learning*, khususnya untuk membangun arsitektur sistem rekomendasi. Sebagai contoh ilustrasi pada **Gambar 2.3** berikut.

DOI: https://doi.org/10.69693/ijmst.v4i2.12785



Gambar 2.3 Tampilan Terminal Proses Ekstraksi lagu dengan Librosa

2.3 Normalisasi Fitur

Setelah ekstraksi *file* lagu dan dibuat menjadi *dataset* sebagai bahan utama selama proses, langkah berikutnya adalah melakukan pengecekan terhadap setiap atribut. Dimulai dengan *memuat dataset menggunakan beberapa fungsi yang digunakan, seperti memeriksa missing values,* mengonversi tipe data pada tabel*,* hingga menampilkan sampel data. Tujuan dilakukannya cleaning pada data pre-processing antara lain untuk mempermudah proses selama normalisasi dan tahapan *clustering*. Sebagai contoh, pada tabel tempo yang sebelumnya dianggap sebagai tipe data string, dibutuhkan konversi ke tipe data float agar prosesnya lebih mudah. Lalu dilanjutkan pembuatan label dari tabel *filename* menjadi dua tabel baru (*artist, song_name*) agar mempermudah *output top-n* rekomendasi lagu, baik di workspace Jupyter maupun dalam implementasi aplikasi *web*.

Tahapan normalisasi fitur dilakukan untuk menyederhanakan jumlah numerik yang dinilai terlalu ilmiah atau terlalu teknis sehingga menyulitkan pengolahan dalam pembelajaran mesin. Hasil dari normalisasi ditampilkan dalam **Gambar 2.4.**

tempo	rms_mean	spectral_centroid_mean	zcr_mean	mfcc_1_mean
99.384	0.299883	0.487746	-0.437179	0.244647
135.999	5.142063	0.387069	-0.219665	0.032761
112.347	-1.349039	0.093427	-0.406105	0.289998
161.499	-0.708860	0.185714	-0.392122	-0.068416
83.354	0.086567	-1.113022	0.164091	-0.478588

Gambar 2.4 Hasil Normalisasi dengan Z-Score

Algoritma *K-Means Clustering* mengelompokkan data berdasarkan metrik jarak antartitik *data*. Karena algoritma ini sangat sensitif terhadap skala, variansi skala yang besar antar fitur *audio* dapat menyebabkan *bias* pada *model*. Pada data mentah hasil ekstraksi fitur dari *Librosa*, terdapat ketimpangan rentang nilai yang sangat ekstrem. Fitur spasial seperti *spectral_rolloff_mean* memiliki skala ribuan (berkisar antara 2.000 hingga 4.700), sedangkan fitur tekstur seperti *zcr_mean* memiliki skala pecahan desimal yang sangat kecil (di bawah 0,1). Jika proses *clustering* dilakukan secara langsung tanpa penyesuaian, fitur dengan skala ribuan akan sepenuhnya mendominasi perhitungan jarak, sehingga bobot fitur penting lainnya akan terabaikan oleh model.

2.4 Implementasi K-Means Clustering

Setelah dilakukan *cleaning* dan normalisasi, proses selanjutnya adalah pembentukan klaster berdasarkan fitur audio yang merepresentasikan beberapa elemen instrumen pada setiap lagu. Tabel tersebut antara lain memiliki filename, artist, song_name, tempo, dan *n_beats*. Tahapan selanjutnya yaitu mencoba menampilkan metrik evaluasi dari hasil tahapan *clustering*. Metrik evaluasi yang digunakan yaitu Indeks *Calinski* dan *WCSS* sebagai pembanding untuk metrik kualitas hasil *clustering*.

2.5 Implementasi Sistem Rekomendasi

Sistem rekomendasi umumnya menggunakan pemilihan basis konten karena memanfaatkan beberapa atribut dari karakteristik fitur audio. Namun, ada dua pendekatan yang digunakan, dan salah satunya menggunakan basis hasil pembentukan klaster sebagai patokan genre atau hasil rekomendasi yang dihasilkan dari pemodelan keempat algoritma yang digunakan.

Adapun 3 alur utama dalam sistem ini di antaranya: pengguna memilih daftar lagu yang tersedia, kemudian sistem memunculkan letak klaster dan daftar Top-n yang tersedia, sehingga dapat menampilkan 2 output sistem rekomendasi dengan hasil metrik evaluasi sebagai penentu nilai kebenaran. Adapun dua metrik output yang digunakan antara lain *Euclidean Distance* , yang mengukur jarak langsung antara dua vektor fitur. Dan *Cosine Similarity,* yang mengukur kemiripan arah vektor dan mengabaikan magnitude

DOI: https://doi.org/10.69693/ijmst.v4i2.12785

2.6 Metrik Evaluasi

Untuk melakukan perhitungan manual yang komprehensif dan mudah diikuti tanpa menghasilkan ratusan baris perhitungan, digunakan pengambilan *top-n* sebagai sampel representatif yang terdapat dalam *dataset.* Dilakukan pengujian dengan evaluasi kesamaan melalui overlap (irisan antara dua output) dan indeks Jaccard sebagai metrik kecocokan dan kesamaan antar-Euclidean dan Cosine, serta pengujian jarak antarperingkat pada setiap output *top-n* menggunakan *Intra-List Similarity*. *Intra-List Similarity* (*ILS*) mengukur diversitas atau homogenitas dalam daftar rekomendasi itu sendiri. Semakin tinggi nilai *ILS* (mendekati 1), semakin homogen (mirip satu sama lain) lagu-lagu dalam daftar tersebut. Semakin rendah atau negatif, semakin beragam (*diverse*) hasil tersebut

2.6.1 Overlap Coefficient

Overlap Coefficient mengukur proporsi irisan antara dua set output rekomendasi terhadap set yang lebih kecil, sehingga memberikan indikasi tentang seberapa banyak lagu yang sama muncul dalam kedua output (Euclidean dan Cosine) (Tan et al., 2022), sebagaimana dirumuskan dalam Persamaan **1**.

$$O(A,B) = \frac{|A \cap B|}{\min(|A|,|B|)} \qquad (1)$$

2.6.2 Jaccard Similarity

Jaccard Similarity mengukur similarity antara dua set sebagai rasio ukuran intersection terhadap ukuran union, sehingga memberikan representasi yang lebih balanced tentang kesamaan antara dua output dibandingkan dengan Overlap Coefficient. Jaccard Index sangat sesuai untuk membandingkan rekomendasi karena sensitif terhadap perbedaan antara kedua set, bukan hanya pada interseksinya (Senapati et al., 2026). Untuk rumus penggunaan dipaparkan dalam Persamaan **2**.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \quad (2)$$

2.6.3 Intra-List Similarity

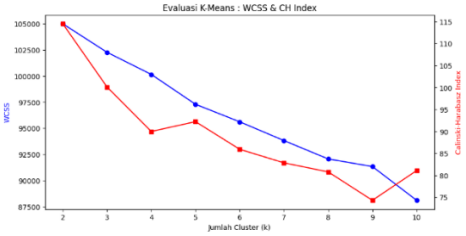
Intra-List Similarity (ILS) mengukur average pairwise similarity antara semua lagu dalam daftar rekomendasi, sehingga memberikan indikasi tentang homogenitas atau diversitas output. ILS merupakan metrik penting untuk mengevaluasi keseimbangan antara relevance dan diversity. Nilai ILS yang tinggi (mendekati 1) mengindikasikan list yang homogen dan berisiko menciptakan filter bubble, sementara nilai yang rendah (mendekati 0) mengindikasikan diversity yang tinggi, namun berisiko kehilangan relevansi. Nilai optimal ILS berkisar pada 0.5 yang merepresentasikan sweet spot antara relevance dan diversity dalam ekosistem sistem rekomendasi (Bianchi, 2025). Untuk penggunaan dirumuskan dalam Persamaan **3**.

$$ILS(P) = \frac{\sum_{i \in P} \sum_{j \in P, j \neq i} s(i,j)}{|P|(|P|-1)} \quad (3)$$

3. Hasil dan Diskusi

3.1 Clustering dengan K-Means

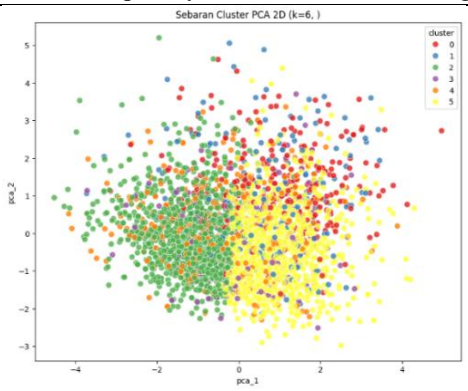
Hasil pembentukan *cluster* akan dijelaskan pada **Gambar 3.1** berikut**.**



Gambar 3.1 Chart Persebaran K-Means Clustering

Kemudian dilanjutkan dengan persebaran indeks per klaster pada **Gambar 3.2.**

DOI: https://doi.org/10.69693/ijmst.v4i2.12785



Gambar 3.2 Scatter-Plot persebaran cluster

Sebagai validasi tambahan atas metode *Elbow* yang terkadang bersifat subjektif, digunakan Indeks *Calinski* (juga dikenal sebagai *Variance Ratio Criterion*). Indeks ini menghitung rasio antara dispersi antar-*cluster* (*between-cluster dispersion*) dan dispersi *intra-cluster* (*within-cluster dispersion*). Semakin tinggi nilai *Calinski*, semakin baik performa *clustering*, yang mengindikasikan bahwa beberapa *cluster* yang terbentuk berdekatan secara internal, namun terpisah secara tegas satu sama lain.

Berdasarkan hasil komputasi model yang telah dilakukan pada variasi $k = 2$ hingga $k = 10$, diputuskan bahwa jumlah *cluster* yang paling optimal untuk *dataset* ini adalah $k = 6$. Model mencatatkan hasil evaluasi dengan nilai WCSS (Inertia): 95,616,62 . dan Nilai CH Index: 85,92

3.2 Hasil Evaluasi

Evaluasi ini bertujuan untuk mengukur tingkat konsistensi peringkat, kesamaan keluaran antara kedua metode jarak (diukur melalui *Overlap Coefficient* dan *Jaccard Similarity*), serta mengukur keberagaman daftar rekomendasi yang diukur melalui *Intra-List Similarity / ILS*. Hasil komputasi dari antarmuka aplikasi berbasis *Flask* direpresentasikan pada **Tabel 3.1** berikut.

Tabel 3.1 Tabel Hasil Evaluasi Rekomendasi

Cluster	Top-N	Overlap	Jaccard	ILS-E	ILS-C
0	Top 3	1,0	1,0	0,38	0,38
1	**Top 3**	**1,0**	**1,0**	**0,73**	**0,73**
2	Top 5	1,0	1,0	0,4	0,4
3	Top 3	1,0	1,0	0,67	0,67
4	Top 3	1,0	1,0	0,63	0,63
5	Top 10	0,9	0,82	0,47	0,47

Kemudian, pendukung visualisasi melalui grafik batang akan diilustrasikan dalam **Gambar 3.3** berikut.

Gambar 3.3 Grafik Bar Hasil Evaluasi

Pada keenam klaster yang diuji, tren kesepakatan antara metode *Euclidean* dan *Cosine* menunjukkan pola yang konsisten. Pada rentang *Top-3* hingga *Top-5*, nilai *Jaccard* dan *Overlap* berfluktuasi secara dinamis bergantung pada karakteristik lagu *target*, menunjukkan bahwa kedua metode tersebut mengekstrak fitur ketetanggaan terdekat dengan cara pandang spasial yang berbeda. Namun, seiring dengan pelebaran batas pencarian ke arah *Top-20*, indeks kesamaan antara kedua metode pada seluruh klaster kembali mengalami konvergensi (peningkatan irisan himpunan). Hal ini membuktikan bahwa untuk pencarian dalam skala yang lebih luas, *Euclidean* dan *Cosine* pada akhirnya menjaring kolam lagu yang ekuivalen dalam ruang klaster tersebut.

Skor *ILS* pada rentang 0,5 merepresentasikan titik keseimbangan ideal (*sweet spot*) dalam ekosistem rekomendasi. Makna analitis dari angka ini menjabarkan dua kondisi krusial, antara lain sistem terhindar dari fenomena Filter

DOI: https://doi.org/10.69693/ijmst.v4i2.12785

Bubble (kejenuhan daftar rekomendasi). Jika nilai *ILS* terlalu tinggi (mendekati 1), daftar rekomendasi akan berisi lagu-lagu yang terlampau identik (monoton), yang berpotensi menurunkan kepuasan eksplorasi pengguna.Selain itu, sistem terhindar dari *noise acak (random noise)*. Jika nilai *ILS* terlalu rendah (mendekati 0), daftar rekomendasi dianggap gagal menangkap preferensi target dan menghasilkan keluaran yang tidak memiliki benang merah.

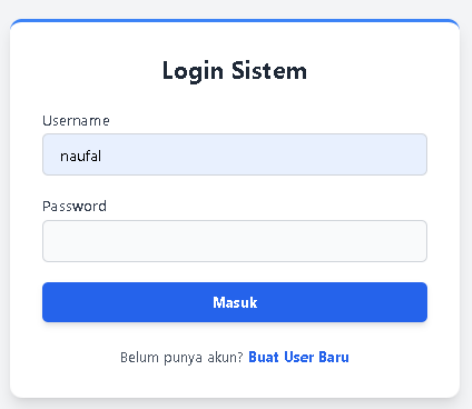
Nilai konsisten di sekitar 0,5 membuktikan bahwa algoritma *K-Means* berhasil membentuk ruang *cluster* yang memuat keberagaman tekstur instrumen yang sehat (healthy diversity), sembari tetap mempertahankan relevansi fitur utama dari lagu target. *Cluster* 1 mencatatkan rentang skor *Intra-List Similarity* (ILS) tertinggi di antara seluruh *cluster* lainnya, yakni **0,73**. Lebih jauh, nilai ILS sebesar 0,73 ini dicapai secara identik dan presisi, baik melalui metode ekstraksi *Euclidean Distance* maupun *Cosine Similarity*.

Fakta bahwa *Euclidean* dan *Cosine* menghasilkan skor *ILS* 0,73 pada *Top-3* membuktikan adanya konvergensi mutlak antaralgoritma. Pada jarak pencarian yang sangat sempit (*Top-3*), apabila sebuah *cluster* sangat padat, maka pencarian berdasarkan magnitudo selisih jarak absolut (*Euclidean*) dan pencarian berdasarkan sudut orientasi fitur (*Cosine*) akan secara absolut menjaring 3 kandidat lagu yang sama persis. Tidak ada divergensi atau perbedaan pemeringkatan di antara kedua metode ini karena 3 lagu teratas tersebut secara harfiah merupakan "kembaran" dari lagu target dalam ruang fitur.

Mencuatnya nilai *ILS* tertinggi pada rentang *Top-3* merupakan perilaku yang logis dalam sistem rekomendasi *Content-Based Filtering*. Pada *Top-3*, sistem dipaksa untuk hanya menampilkan lagu-lagu yang jarak fitur akustiknya paling minimum (paling identik) dengan input. Akibatnya, keberagaman (*diversity*) harus dikorbankan demi mengejar tingkat relevansi (*precision*) yang mutlak. Nilai *ILS* ini secara teoretis akan berangsur turun (lebih bervariasi) apabila batas rekomendasi untuk target *Cluster* 1 tersebut diperlebar ke arah *Top-10* atau *Top-20*.

3.3 Implementasi Web

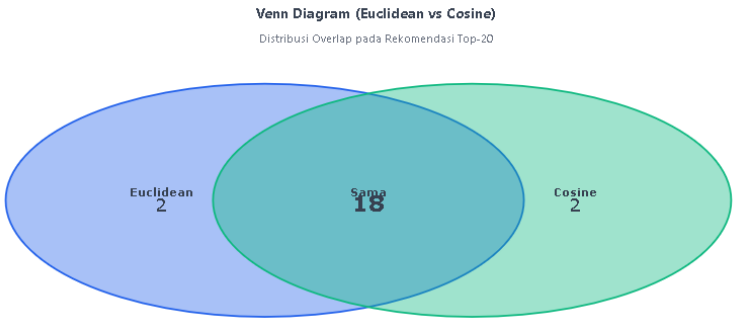
Sebagai hasil implementasi untuk aplikasi berbasis *web*, seperti pada contoh dari implementasi halaman login pada **Gambar 3.4**



Gambar 3.4 Halaman Login

Sistem rekomendasi telah berhasil diintegrasikan ke dalam antarmuka aplikasi *web* dinamis menggunakan *framework Flask*. Antarmuka ini dirancang secara analitis dan *user-friendly*, dengan keunggulan utama, antara lain, pengguna dapat melakukan pencarian lagu input secara *real-time* berbasis teks melalui implementasi skrip *Tom Select.js* yang mempercepat proses *query* ke *dataset*.

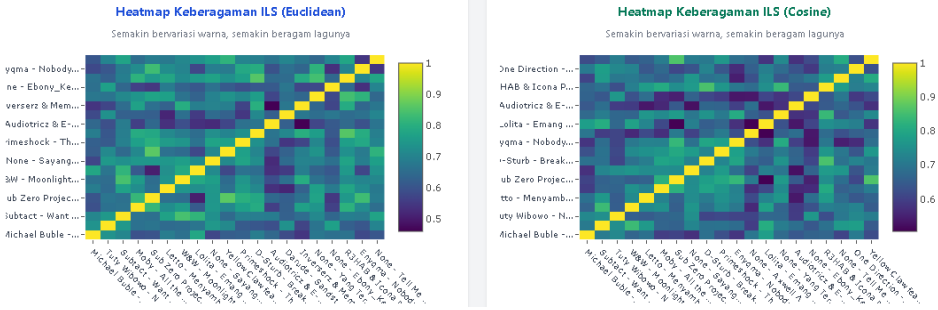
Sebagai pendukung pada metrik evaluasi pada penggunaan *Top-N* yang digunakan, sistem rekomendasi juga menampilkan beberapa visualisasi data dengan implementasi Diagram *Venn* untuk menampilkan irisan dari metrik *Overlap* dan indeks *Jaccard*, seperti contoh pada **Gambar 3.5**



Gambar 3.5 Diagram Venn

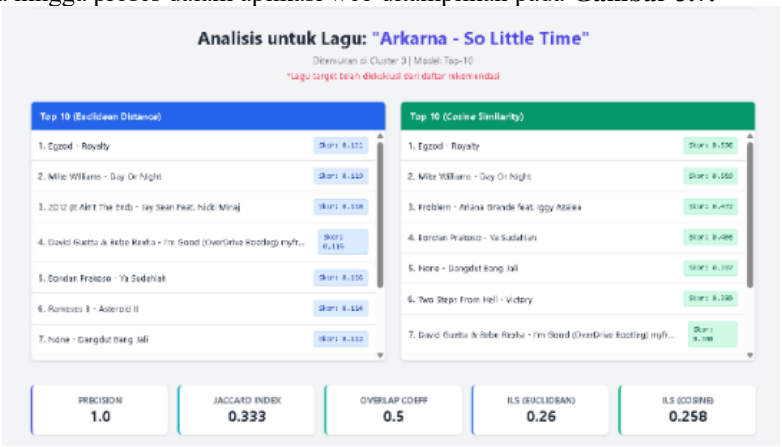
DOI: https://doi.org/10.69693/ijmst.v4i2.12785

Kemudian dilanjut dengan implementasi visualisasi *Heatmap* untuk menampilkan sebaran diversitas pada metrik *Euclidean* maupun *Cosine* dengan metode *ILS*, seperti contoh penggunaannya pada **Gambar 3.6**



Gambar 3.6 Visualisasi Heatmap untuk uji Intra-List Similarity

Aplikasi tidak hanya menampilkan tabel pemeringkatan, tetapi juga membuktikan efektivitas metode secara *visual*. Implementasi Diagram *Venn* dengan *Plotly* secara intuitif menggambarkan irisan (*overlap*) antara *Euclidean* dan *Cosine*, sementara *Heatmap* memberikan representasi visual yang jelas terhadap keberagaman (*ILS*) tanpa mengharuskan pengguna membaca matriks angka secara manual.
Tampilan antarmuka hingga proses dalam aplikasi web ditampilkan pada **Gambar 3.7.**



Gambar 3.7 Tampilan lengkap antarmuka web

Aplikasi dapat secara cerdas membuang (*drop*) lagu target dari daftar keluaran, sehingga mencegah anomali ketika sistem merekomendasikan lagu yang sama persis dengan yang dicari pengguna.

4. Kesimpulan

Sistem berhasil beroperasi secara mandiri menggunakan fitur sinyal akustik tingkat rendah (low-level features) yang diekstraksi langsung dari file audio mentah menggunakan Librosa. Proses ekstraksi MFCC, Spectral Centroid, dan ZCR yang dilanjutkan dengan normalisasi Z-Score terbukti mampu menyeimbangkan skala dimensi data tanpa menghilangkan pola sebaran aslinya. Pendekatan ekstraksi secara offline ini berhasil mengeliminasi ketergantungan sistem pada dataset sekunder atau metadata manual, sekaligus memecahkan masalah cold-start untuk lagu-lagu baru.
Evaluasi *model K-Means Clustering* menggunakan kombinasi *Within-Cluster Sum of Squares (WCSS)* dan *Calinski-Harabasz (CH) Index* berhasil menemukan partisi dataset yang paling optimal pada nilai k = 6 dengan skor *WCSS* **95.616,62** dan *CH Index* **85,92**. Pengelompokan ini terbukti sukses sebagai *filter* ruang pencarian tingkat pertama yang mereduksi kompleksitas waktu secara drastis, sehingga beban pemrosesan pada *server* aplikasi *Flask* tetap ringan dan waktu respons berjalan nyaris instan (kurang dari 1 detik).
Perbandingan metode *Euclidean Distance* dan *Cosine Similarity* pada batas *Top-N* menunjukkan konvergensi yang dinamis. Berdasarkan metrik *Overlap Coefficient* dan *Jaccard Similarity*, kedua metode memiliki **tingkat kesepakatan mutlak** pada pencarian terdekat (*Top-3*), berfluktuasi pada pencarian menengah, dan kembali menjaring himpunan yang ekuivalen pada pencarian luas (*Top-20*). Uji validasi silang pada seluruh perwakilan cluster menunjukkan bahwa rata-rata skor *Intra-List Similarity (ILS)* untuk kedua algoritma pencarian stabil di rentang **0,5**, kecuali pada anomali kepadatan *Cluster* 1. Angka ini menunjukkan bahwa algoritma berhasil menemukan sweet spot (titik keseimbangan) yang menyajikan rekomendasi tekstur instrumen yang relevan sekaligus mempertahankan keberagaman genre yang sehat agar pengguna terhindar dari *filter bubble*.

DOI: https://doi.org/10.69693/ijmst.v4i2.12785

5. Referensi

Bevec, M., Tkalčič, M., & Pesek, M. (2024). Hybrid music recommendation with graph neural networks. *User Modeling and User-Adapted Interaction*, *34*(5), 1891–1928. https://doi.org/10.1007/s11257-024-09410-4

Bianchi, E. (2025). Beyond Relevance: An Adaptive Exploration-Based Framework for Personalized Recommendations. *Faculty of Engineering, Free University of Bozen-Bolzano, NOI Techpark - via Bruno Buozzi, 1, Bozen-Bolzano, 39100, Italy.* http://arxiv.org/abs/2503.19525

Chen, P., Dong, L., & Liu, Y. (2024). Design of music recommendation system based on EDA and K-means cluster analysis. *Applied and Computational Engineering*, *50*(1), 317–323. https://doi.org/10.54254/2755-2721/50/20241695

Gul, M., & Rehman, M. A. (2023). Big data: an optimized approach for cluster initialization. *Journal of Big Data*, *10*(1). https://doi.org/10.1186/s40537-023-00798-1

Mukhopadhyay, S., Kumar, A., Parashar, D., & Singh, M. (2024). Enhanced Music Recommendation Systems: A Comparative Study of Content-Based Filtering and K-Means Clustering Approaches. *Revue d'Intelligence Artificielle*, *38*(1), 365–376. https://doi.org/10.18280/ria.380138

Nurhalizah, R. S., Ardianto, R., & Purwono, P. (2024). Analisis Supervised dan Unsupervised Learning pada Machine Learning: Systematic Literature Review. *Jurnal Ilmu Komputer Dan Informatika*, *4*(1), 61–72. https://doi.org/10.54082/jiki.168

Oh, Y., Yun, S., Hyun, D., Kim, S., & Park, C. (2023). MUSE: Music Recommender System with Shuffle Play Recommendation Enhancement. *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM '23*, 1928–1938. https://doi.org/10.1145/3583780.3614976

Pake, G. V. R., Mawardi, V. C., & Sutrisno, T. (2023). PENERAPAN METODE CONTENT BASED FILTERING DALAM IMPLEMENTASI SISTEM REKOMENDASI MUSIK. *Jurnal Serina Sains, Teknik Dan Kedokteran*, *1*(2), 455–462. https://doi.org/10.24912/jsstk.v1i2.31037

Prey, R. (2020). Locating Power in Platformization: Music Streaming Playlists and Curatorial Power. *Social Media and Society*, *6*(2). https://doi.org/10.1177/2056305120933291

Prey, R., Esteve Del Valle, M., & Zwerwer, L. (2022). Platform pop: disentangling Spotify's intermediary role in the music industry. *Information Communication and Society*, *25*(1), 74–92. https://doi.org/10.1080/1369118X.2020.1761859

Retnoningsih, E., & Pramudita, R. (2020). Mengenal Machine Learning Dengan Teknik Supervised dan Unsupervised Learning Menggunakan Python. *BINA INSANI ICT JOURNAL*, *7*(2), 156–165. https://www.python.org/

Schedl, M., Zamani, H., Chen, C. W., Deldjoo, Y., & Elahi, M. (2022). Current challenges and visions in music recommender systems research. *International Journal of Multimedia Information Retrieval*, *7*(2), 95–116. https://doi.org/10.1007/s13735-018-0154-2

Senapati, A., Kujur, N., & Yelleti, V. (2026). Dynamic Graph with Similarity-Aware Attention Graph Neural Network for Recommender Systems. *Department of Computer Science and Engineering, SRM University Andhra Pradesh, Mangalagiri, Andhra Pradesh 522502, India*. http://arxiv.org/abs/2605.05238

Srivastava, A. (2024). Feature-Based User-Centric Music Recommendation. *Journal of Student Research Volume 13 Issue 4*. www.JSR.org/hs

Tan, P. N., Karpatne, A., & Steinbach. (2022). *Introduction to Data Mining (2nd ed.)*. Pearson Education.

Valkenborg, D., Rousseau, A. J., Geubbelmans, M., & Burzykowski, T. (2023). Unsupervised learning. In *American Journal of Orthodontics and Dentofacial Orthopedics* (Vol. 163, Number 6, pp. 877–882). Elsevier Inc. https://doi.org/10.1016/j.ajodo.2023.04.001

Visher Laja Jaja, Y., Susanto, B., Ricky Sasongko, L., & Kunci, K. (2020). Penerapan Metode Item-Based Collaborative Filtering Untuk Sistem Rekomendasi Data MovieLens. *D'Cartesian: Jurnal Matematika Dan Aplikasi, Vol. 9, No. 2 (September 2020): 78-83*. https://ejournal.unsrat.ac.id/index.php/decartesian

Widiyaningtyas, T., Saifudin, I., Zaeni, I. A. E., Maulana, M. Z. N., & Caesarendra, W. (2025). Memory-based collaborative filtering based on matrix factorization and Gower's set rank. *Journal of King Saud University - Computer and Information Sciences*, *37*(9). https://doi.org/10.1007/s44443-025-00261-6

Xinyue Li. (2024). Analysis of machine learning-based music recommendation system using Spotify datasets. *Applied and Computational Engineering*, *77*(1), 49–55. https://doi.org/10.54254/2755-2721/77/20240650

Yuniar, P., Sitoena, J. K., Matius, D. M., Obed, G. B., Tinggi, S., & Jaffray, F. (2022). Sejarah Musik sebagai Dasar Pengetahuan dalam Pembelajaran Teori Musik. In *Jurnal Musik dan Pendidikan Musik* | (Vol. 3, Number 2).

DOI: https://doi.org/10.69693/ijmst.v4i2.12785