

A Hybrid Approach to Music Recommendations Based on Audio Similarity Using Autoencoder and LightGBM

Winda Ardelia Aristawidya^{1*}, Majid rahardi^{2*}**

* Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Amikom Yogyakarta

windaardelia@students.amikom.ac.id¹, majid@amikom.ac.id²

Article Info

Article history:

Received 2025-07-28

Revised 2025-10-23

Accepted 2025-11-05

Keyword:

*Music Recommendation System,
Audio Feature,
Autoencoder,
PCA,
LightGBM.*

ABSTRACT

Music recommendation systems help users navigate large music collections by suggesting songs aligned with their preferences. However, conventional methods often overlook the depth of audio content, limiting personalization and accuracy. This study proposes a hybrid approach that uses PCA and Autoencoder to extract audio embeddings. These embeddings are processed using K-Nearest Neighbors to find similar tracks, followed by a reranking step with LightGBM based on predicted relevance. The system achieved strong results: 98% accuracy, 0.96 precision, 0.96 recall, and 0.96 F1-score for the Similar class, with 0.99 precision and recall for Not Similar. Cross-validation confirmed model robustness, with an average accuracy of 97.99%, precision of 0.9577, recall of 0.9624, and F1-score of 0.9600, all with low standard deviations. These outcomes show that combining deep audio features with machine learning ranking enhances recommendation quality. Future improvements may involve incorporating metadata and genre-based visualizations for more diverse and interpretable results.



This is an open access article under the [CC-BY-SA](#) license.

I. PENDAHULUAN

Sistem rekomendasi memainkan peran krusial dalam berbagai layanan digital, terutama dalam industri musik yang menawarkan jutaan lagu kepada pengguna. Seiring dengan meningkatnya jumlah data musik digital, dibutuhkan pendekatan cerdas yang mampu secara efisien menyaring dan menyarankan konten yang sesuai[1]. Teknologi seperti machine learning dan data mining menjadi landasan utama dalam membangun sistem rekomendasi, terutama dalam mengenali pola kemiripan antar lagu maupun preferensi pengguna[2]. Tantangan dalam industri musik digital salah satunya adalah cold-start muncul ketika sistem kesulitan memberikan rekomendasi yang relevan bagi pengguna baru karena minimnya riwayat interaksi atau preferensi yang dapat dijadikan dasar pembelajaran[3]. Meski demikian, masih terdapat tantangan dalam menangkap kedekatan musical secara akurat jika hanya bergantung pada metadata seperti nama artis atau genre[4]. Oleh sebab itu, pemanfaatan fitur audio dan pembentukan representasi vektor menjadi pendekatan yang semakin relevan dalam menghasilkan rekomendasi lagu yang lebih kontekstual dan mendalam[5].

Dalam pengembangan sistem rekomendasi musik, pendekatan yang berbasis fitur audio semakin banyak diterapkan karena dinilai mampu merepresentasikan karakteristik musical secara lebih objektif dibandingkan dengan penggunaan metadata konvensional[6]. Penelitian ini mengimplementasikan strategi hybrid yang menggabungkan metode unsupervised learning, seperti autoencoder dan Principal Component Analysis (PCA), untuk mengekstraksi representasi vektor embedding dari fitur-fitur audio[7]. Selanjutnya, model supervised LightGBM digunakan dalam tahap reranking untuk mengurutkan kandidat lagu berdasarkan estimasi relevansi terhadap lagu masukan[8]. Pendekatan ini dirancang guna mengatasi keterbatasan metode tradisional yang umumnya hanya bertumpu pada informasi genre atau artis. Dengan memanfaatkan vektor hasil ekstraksi dari fitur audio sistem mampu mengidentifikasi kemiripan antar lagu secara lebih mendalam[9]. Meskipun demikian, efektivitas integrasi antara teknik embedding dan reranking masih menjadi topik yang terus dikaji, sehingga penelitian ini diharapkan dapat memberikan kontribusi dalam

pengembangan sistem rekomendasi lagu yang lebih akurat dan kontekstual.

Sejumlah penelitian terkini telah menerapkan berbagai metode state-of-the-art dalam sistem rekomendasi musik, seperti Factorization Machine (FM), Singular Value Decomposition (SVD), Content-Based Filtering (CBF), serta model hybrid yang mengombinasikan beberapa pendekatan untuk meningkatkan akurasi rekomendasi[10], integrasi beragam metode embedding seperti PCA dan Autoencoder dalam satu alur kerja[11], disertai penerapan LightGBM sebagai reranker berbasis relevansi vektor audio[8], masih jarang dijumpai dalam literatur. Selain itu, pendekatan yang menggabungkan pembelajaran tak terawasi dan terawasi dalam konteks sistem rekomendasi musik umumnya masih bersifat terpisah-pisah dan belum banyak dianalisis secara terpadu. Kondisi ini menunjukkan adanya kesenjangan pengetahuan yang penting, terutama terkait efektivitas penerapan strategi hybrid dalam menghasilkan rekomendasi lagu yang lebih presisi dan sesuai konteks.

Penelitian ini mengembangkan sebuah sistem rekomendasi musik berbasis pendekatan hybrid yang mengintegrasikan teknik Principal Component Analysis dan Autoencoder untuk menghasilkan representasi embedding dari fitur audio lagu. Embedding tersebut kemudian digunakan oleh algoritma K-Nearest Neighbors dalam tahap pencarian kandidat lagu yang memiliki kemiripan tinggi. Untuk meningkatkan relevansi hasil rekomendasi, algoritma LightGBM diterapkan dalam proses reranking, yaitu mengurutkan kembali kandidat lagu berdasarkan prediksi tingkat keterkaitan terhadap lagu input. Hasil evaluasi menunjukkan bahwa model ini mampu memberikan rekomendasi yang lebih kontekstual dibandingkan pendekatan tradisional yang mengandalkan metadata[12]. Kontribusi utama dari penelitian ini terletak pada integrasi menyeluruh antara teknik ekstraksi fitur dan pembelajaran terawasi dalam satu arsitektur sistem rekomendasi musik, yang masih jarang dijumpai dalam literatur sebelumnya.

II. METODE

Alur penelitian ini terstruktur dalam beberapa tahapan berurutan, yang divisualisasikan pada Gambar 1.

A. Data Collection

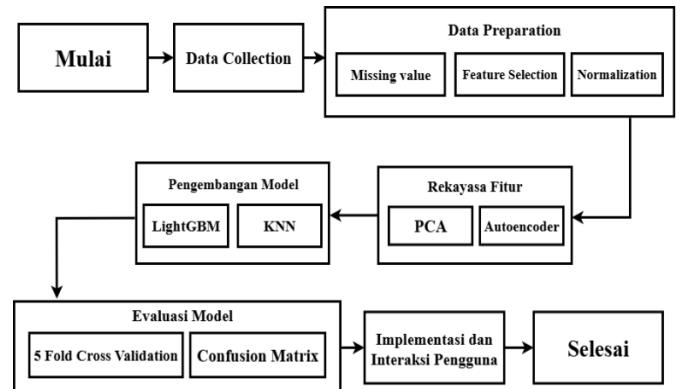
Penelitian ini menggunakan data sekunder berupa dataset lagu dari platform Spotify yang diunduh dari repositori publik. Dataset ini memuat berbagai fitur audio dan metadata lagu yang relevan untuk analisis karakteristik musik.

B. Data Preparation

Pada tahap ini, data dipersiapkan dan dikonfirmasi konsistensinya agar siap digunakan dalam proses pemodelan. Berikut adalah langkah-langkahnya:

1. Penanganan Missing Value

Pada tahap pra-pemrosesan data, langkah awal dilakukan dengan menangani missing value karena keberadaannya dapat menurunkan akurasi model[13]. Baris data dengan nilai kosong pada kolom track_genre dihapus karena track_genre dianggap sebagai fitur fundamental untuk analisis kemiripan dan preferensi lagu.



Gambar 3. Contoh gambar dengan resolusi yang dapat diterima

2. Seleksi Fitur

Seleksi fitur dilakukan untuk mengurangi kompleksitas data dan noise dengan menghapus kolom-kolom tidak relevan[14]. Kolom Unnamed: 0, track_id dan explicit dieliminasi karena tidak berkontribusi pada karakteristik musical atau preferensi pengguna.

3. Normalisasi

Normalisasi fitur adalah tahap penting untuk menyeragamkan skala fitur-fitur numerik. Fitur-fitur audio numerik seperti danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo memiliki rentang nilai yang bervariasi. Untuk menstandarkan skala dan menghindari bias pada algoritma yang sensitif terhadap magnitudo fitur, semua fitur ini dinormalisasi menggunakan StandardScaler. Transformasi ini memastikan setiap fitur memiliki rata-rata nol dan varians satu.

C. Rekayasa Fitur

Fitur yang telah diskalakan kemudian diproses lebih lanjut melalui dua pendekatan utama berikut:

1. Principal Component Analysis

Principal Component Analysis (PCA) digunakan untuk mereduksi 12 fitur audio menjadi 10 komponen utama. Langkah ini bertujuan untuk mengurangi redundansi dan noise dalam data dengan tetap mempertahankan varians informasi yang maksimal[7].

2. Autoencoder

Data 10 dimensi hasil PCA selanjutnya digunakan untuk melatih sebuah Autoencoder menggunakan library TensorFlow/Keras. Arsitektur deep learning ini dirancang untuk mengompresi encode data input menjadi sebuah

representasi laten embedding berdimensi 8, lalu merekonstruksinya decode kembali ke 10 dimensi. Dengan meminimalkan mean squared error rekonstruksi, proses pelatihan ini mengoptimalkan kemampuan encoder untuk menyarikan pola-pola paling esensial dari data. Pada akhirnya, vektor embedding 8 dimensi yang telah teroptimasi ini diekstrak dan digunakan sebagai fitur final untuk setiap lagu[15].

D. Pengembangan Model

Pada tahap ini, dikembangkan beberapa model yang berfungsi sebagai komponen utama dalam sistem rekomendasi.

1. K-Nearest Neighbors

Dua model K-Nearest Neighbors dirancang untuk tujuan berbeda. Model pertama bertugas mengenali lagu-lagu yang serupa berdasarkan embedding dari fitur audio[16], sedangkan model kedua menggunakan rata-rata embedding tiap genre guna menilai kedekatan antar genre. Kedua model menerapkan jarak Euclidean untuk mengukur kedekatan antar vektor, dengan rumus:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

2. Light Gradient Boosting Machine

Untuk memberikan rekomendasi yang lebih terpersonalisasi, dikembangkan sebuah model klasifikasi berbasis LightGBM[17]. Model ini bertugas memprediksi probabilitas kemiripan antara dua lagu. Dataset pelatihan disusun secara semi-otomatis dengan memanfaatkan model KNN sebagai sumber pembelajaran awal. Pasangan lagu yang diprediksi saling berdekatan oleh model KNN ditandai sebagai data positif label = 1, sedangkan pasangan lagu yang dipilih secara acak dianggap tidak mirip dan diberi label negatif label = 0.

E. Evaluasi Model

1. Confusion Matrix

Evaluasi model dilakukan menggunakan confusion matrix untuk mengukur performa klasifikasi pada data uji. Confusion matrix menampilkan jumlah prediksi yang benar dan salah, yang dikelompokkan berdasarkan kelas aktual dan kelas yang diprediksi, sebagaimana ditunjukkan pada Tabel 1.

TABEL 1
EVALUASI CONFUSION MATRIX

Kategori Aktual	Positive	Negative
Prediksi Positif	TP	FP
Prediksi Negatif	FN	TN

2. 5-Fold Cross-Validation

Selain evaluasi pada data uji, model juga divalidasi menggunakan teknik 5-Fold Cross-Validation untuk mengukur konsistensi performa. Dalam metode ini, dataset dibagi menjadi lima bagian. Model dilatih dan diuji sebanyak lima kali, dengan setiap bagian bergantian menjadi data uji. Hasil evaluasi dari tiap fold dirata-rata untuk mendapatkan performa keseluruhan yang lebih stabil dan tidak bergantung pada satu subset data saja. Teknik ini penting untuk menghindari overfitting dan memberikan gambaran lebih umum tentang kemampuan model dalam melakukan prediksi pada data yang belum pernah dilihat sebelumnya.

F. Pengembangan Sistem dan Antarmuka Pengguna

Sistem ini merupakan implementasi rekomendasi lagu berbasis konten yang dijalankan melalui antarmuka teks interaktif. Pengguna dapat memasukkan judul lagu secara langsung atau memilih dari hasil pencarian dalam dataset untuk dijadikan referensi. Setelah lagu dipilih, sistem akan mengumpulkan kandidat rekomendasi berdasarkan kemiripan genre, popularitas, dan embedding lagu. Proses pencarian dimulai dengan lagu dari genre yang sama menggunakan algoritma K-nearest neighbors, dilanjutkan dengan lagu dari genre terdekat serta lagu-lagu dengan tingkat popularitas tinggi. Selanjutnya, kandidat lagu yang telah diperoleh direranking menggunakan model LightGBM yang mengevaluasi kesamaan embedding antara lagu input dan kandidat. Nilai akhir dari tiap lagu ditentukan berdasarkan gabungan skor prediksi dari LightGBM dan popularitas, untuk menghasilkan rekomendasi akhir yang relevan.

III. HASIL DAN PEMBAHASAN

Pada bab ini, disajikan berbagai hasil yang diperoleh dari masing-masing tahapan yang telah diuraikan sebelumnya dalam metode penelitian. Uraian berikut akan menjelaskan temuan-temuan yang muncul berdasarkan proses yang telah dilaksanakan.

A. Data Collection

Data yang digunakan dalam penelitian ini berasal dari dataset publik Spotify yang berisi informasi audio dan metadata lagu. Dataset ini dipilih karena lengkap dan sesuai dengan kebutuhan analisis rekomendasi musik berbasis fitur audio. Beberapa fitur yang tersedia antara lain danceability, energy, valence, tempo, dan loudness, serta informasi lagu seperti judul, artis, dan popularitas. Data ini nantinya digunakan untuk menganalisis kemiripan antar lagu. Tabel 2 menunjukkan cuplikan isi dataset untuk memberi gambaran awal struktur dan jenis data yang dianalisis.

TABEL 2
ISI DATASET FITUR AUDIO

Nama Kolom	Nilai
track_id	0snQkGI5qnAmohLE7jTsTn
artist	System Of A Down
album_name	Toxicity

track_name	Toxicity
popularity	81
duration_ms	218933
explicit	FALSE
danceability	0.413
energy	0.873
key	8
loudness	-4.151
mode	1
spechiness	0.0536
acouticness	0.000264
instrumentalness	0.00249
liveness	0.204
valance	0.48
tempo	116.68
time_signature	3
track_genre	alternative

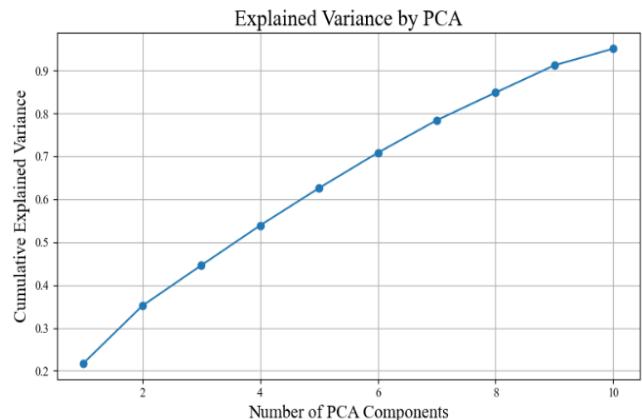
B. Data Preparation

Dataset yang digunakan berasal dari koleksi lagu dari berbagai genre sebanyak 8.000 entri. Setelah proses pembersihan data dilakukan, lagu-lagu tanpa label genre dihapus sehingga tersisa sebanyak 7.894 lagu. Lalu untuk tahapan seleksi fitur kolom seperti Unnamed: 0, track_id dan explicit di hapus karena tidak mendukung karena tidak memberikan informasi prediktif mengenai karakteristik audio. Setelah dilakukan normalisasi menggunakan StandardScaler, semua fitur numerik memiliki skala yang seragam dengan rata-rata mendekati 0 dan standar deviasi mendekati 1. Hasil ini memastikan bahwa tidak ada fitur yang lebih dominan dari yang lain, sehingga proses perhitungan jarak antar lagu dapat dilakukan secara adil dan seimbang.

C. Rekayasa Fitur

1. Principal Component Analysis

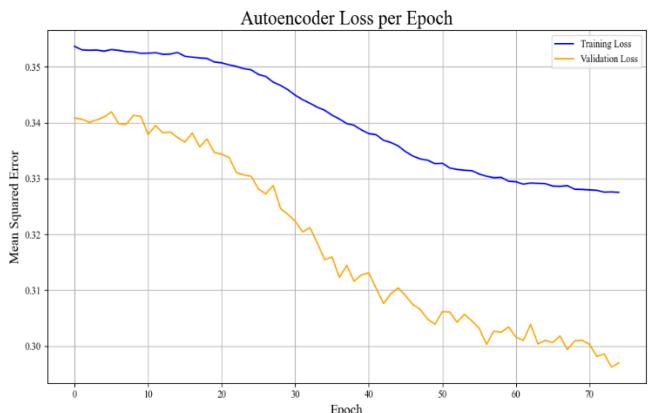
Langkah awal dalam proses ekstraksi fitur adalah dengan mereduksi dimensi data menggunakan algoritma Principal Component Analysis. Berdasarkan Gambar 1, terlihat bahwa sekitar 90% variansi dalam data dapat dijelaskan dengan menggunakan 9 hingga 10 komponen utama. Oleh karena itu, pemilihan jumlah komponen PCA ditetapkan sebanyak 10 untuk tahap embedding selanjutnya, guna menjaga keseimbangan antara efisiensi dan kualitas informasi.



Gambar 1. Grafik Cumulative Explained Variance dari PCA

2. Autoencoder

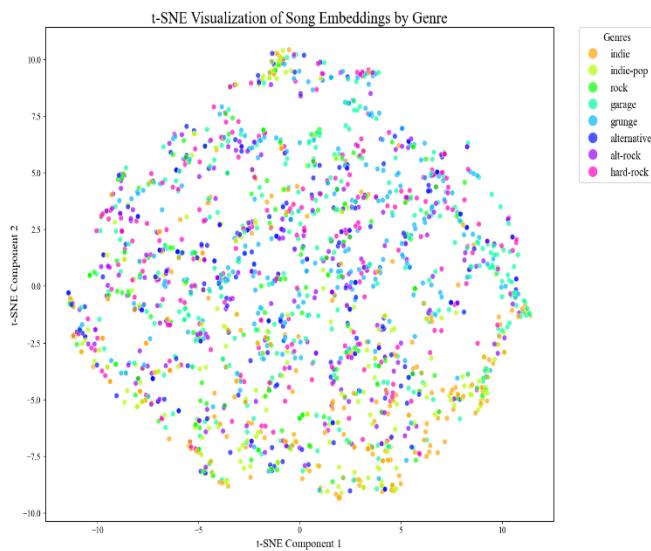
Setelah data direduksi menggunakan PCA, proses selanjutnya adalah melakukan embedding menggunakan autoencoder. Dapat diamati bahwa nilai training loss secara konsisten menurun seiring bertambahnya epoch, demikian pula dengan validation loss dengan nilai akhir MSE rekonstruksi sebesar 0.34475 yang menunjukkan penurunan signifikan dan stabil hingga akhir pelatihan. Hal ini menunjukkan bahwa model berhasil mempelajari representasi fitur tanpa mengalami overfitting. Loss pada data validasi yang lebih rendah dibanding data pelatihan juga menunjukkan generalisasi model yang baik. Untuk meningkatkan kinerja model, dilakukan parameter tuning pada arsitektur Autoencoder. Hasil pengujian menunjukkan konfigurasi terbaik dengan batch size 16, epochs 50, hidden dimension 10, dan latent dimension 8, menghasilkan nilai Mean Squared Error (MSE) terendah sebesar 0.047583. Nilai ini menunjukkan bahwa model mampu mempelajari representasi fitur audio secara efisien dengan tingkat kesalahan rekonstruksi yang rendah.



Gambar 2. Visualisasi Autoencoder Loss per Epoch

Visualisasi t-SNE dari embedding menunjukkan kluster yang terbentuk berdasarkan genre, menunjukkan bahwa embedding yang dihasilkan oleh autoencoder secara efektif

menangkap dan mewakili karakteristik khas dari setiap genre musik.



Gambar 3. Visualisasi embedding lagu dengan (t-SNE) berdasarkan genre

D. Pengembangan Model

1. K-Nearest Neighbors

Dua model KNN dikembangkan untuk dua fungsi berbeda. Model pertama digunakan untuk mencari lagu-lagu yang mirip secara audio berdasarkan embedding dari fitur lagu. Sistem menghitung jarak antar embedding, lalu memilih beberapa lagu terdekat sebagai rekomendasi. Model kedua menggunakan rata-rata embedding dari setiap genre. Dengan cara ini, sistem bisa menilai seberapa dekat satu genre dengan genre lain berdasarkan ciri-ciri audio yang terkandung dalam lagu-lagunya. Kedua model ini menjadi dasar sistem dalam memahami kemiripan lagu dan genre secara numerik.

2. Light Gradient Boosting Machine

Model LightGBM dikembangkan sebagai klasifikasi untuk memprediksi apakah dua lagu mirip atau tidak. Dataset pelatihan dibuat dengan cara mengambil pasangan lagu yang mirip menurut KNN sebagai data positif, dan pasangan lagu acak sebagai data negatif. Model ini belajar dari pola embedding antar lagu untuk menghasilkan prediksi kemiripan. Dengan begitu, sistem bisa memberi rekomendasi yang tidak hanya berdasarkan jarak, tapi juga berdasarkan pola yang dipelajari dari data sebelumnya. Selanjutnya, dilakukan hyperparameter tuning pada model LightGBM untuk memperoleh performa optimal. Hasil Grid Search menunjukkan konfigurasi terbaik dengan learning rate 0.05, max depth 10, min child samples 20, n estimators 100, dan num leaves 15, menghasilkan nilai F1-Score sebesar 0.9653. Nilai tersebut mencerminkan keseimbangan yang baik antara precision dan recall, sehingga model mampu mengklasifikasikan pasangan lagu mirip secara akurat dan konsisten.

E. Evaluasi Model

Sebagai tolok ukur awal, model PCA dan KNN dievaluasi dan dibandingkan dengan Autoencoder dan KNN. PCA dan KNN memiliki Average Cosine Similarity 0.9810, sedikit lebih rendah dari Autoencoder 0.9955, menunjukkan kemiripan lagu yang kurang presisi. Namun, diversity lebih tinggi 0.0299 dibandingkan 0.0058 dan coverage sedikit lebih baik 9.59% dibandingkan 9.23. Sementara itu, model utama menghasilkan average cosine similarity 0.9936 dengan diversity 0.0059 dan coverage 32.2%, menandakan rekomendasi sangat relevan dan konsisten dalam memberikan lagu dengan karakter yang serupa, sesuai dengan tujuan sistem yang berfokus pada kemiripan antar lagu. Model menghasilkan rekomendasi yang sangat relevan dengan tingkat kesamaan tinggi antar lagu. Autoencoder memerlukan waktu pelatihan lebih lama dibanding PCA, namun hal ini wajar karena proses pelatihan jaringan saraf. Efisiensi komputasi bukan fokus utama penelitian ini.

1. Confusion Matrix

Evaluasi awal dilakukan menggunakan confusion matrix seperti ditunjukkan pada Gambar 4.

		Confusion Matrix - LightGBM (Test Data)	
		Not Similar	Similar
True Label	Not Similar	2964	36
	Similar	40	960
		Not Similar	Similar
		Predicted Label	

Gambar 4. Confusion Matrix model LightGBM berdasarkan Data Test

Model LightGBM berhasil mencapai akurasi sebesar 98% dalam mengklasifikasikan pasangan lagu mirip dan tidak mirip. Berdasarkan confusion matrix, model mampu mengenali 2964 data tidak mirip dan 960 data mirip dengan benar, serta menghasilkan 36 false positive dan 40 false negative. Kinerja model tergolong stabil, dengan nilai precision, recall, dan F1-score untuk kelas "Not Similar" sebesar 0.99, dan untuk kelas "Similar" masing-masing sebesar 0.96. Hasil ini menunjukkan bahwa model cukup presisi dan seimbang dalam membedakan kedua kelas.

2. 5-fold Cross-Validation

Model dievaluasi menggunakan teknik 5-fold cross-validation, dan hasilnya menunjukkan performa yang

konsisten dengan akurasi tinggi. Nilai precision, recall, dan F1-score yang diperoleh berada pada kisaran sangat baik, menandakan bahwa model mampu mengenali pola kemiripan lagu secara efektif dan seimbang antara prediksi positif dan negatif.

TABEL 3
EVALUASI 5-FOLD CROSS-VALIDATION

Metrix Evaluasi	Value
Accuracy	0.97
Precision	0.95
Recall	0.96
F1 Score	0.96

F. Implementasi dan Interaksi Pengguna

Pada fase pengujian sistem rekomendasi musik, dilakukan simulasi dengan menggunakan judul lagu sebagai kueri masukan. Salah satu kasus uji melibatkan pengguna yang memasukkan ‘Come As You Are’ oleh Nirvana sebagai lagu referensi. Berdasarkan masukan ini, sistem menghasilkan sepuluh lagu rekomendasi, diurutkan berdasarkan skor akhir. Skor ini diperoleh dari kombinasi kemiripan vektor embedding dan popularitas lagu yang diprediksi menggunakan model LightGBM.

```
Enter the title of the song you want: Come As You Are

--- RECOMMENDATION RESULTS ---
   track_name      artists  track_genre  final_score
3     Sweater Weather The Neighbourhood alt-rock    0.974454
7002   Sweater Weather The Neighbourhood      rock    0.974454
1003   Sweater Weather The Neighbourhood alternative  0.974454
3001     In the End Linkin Park        grunge   0.953891
1006     In the End Linkin Park alternative  0.953891
4000   Mary On A Cross       Ghost    hard-rock  0.953327
7051   Mary On A Cross       Ghost      rock   0.953327
3009   Come As You Are      Nirvana        grunge  0.936999
4003 Sweet Child O' Mine Guns N' Roses hard-rock  0.923376
300     Cry Baby The Neighbourhood alt-rock    0.911618
-----
Would you like to get another recommendation? (y/n): n
```

Gambar 5. Hasil dari pencarian sistem rekomendasi

Proses rekomendasi dimulai dengan memilih lagu kandidat melalui algoritma K-Nearest Neighbors berdasarkan genre yang sama, genre terkait, dan tingkat popularitas, diikuti dengan tahap penyortiran ulang menggunakan model prediktif. Sebagian besar lagu yang direkomendasikan termasuk dalam genre serupa seperti grunge, rock alternatif, dan rock keras, mencerminkan kemampuan sistem untuk mengidentifikasi dan menafsirkan kesamaan musik antara lagu berdasarkan fitur audio dan representasi embedding yang dipelajari.

IV. KESIMPULAN

Sistem rekomendasi musik yang dikembangkan dengan pendekatan hibrida berhasil mencapai performa tinggi, dengan akurasi 98% dan F1-score 0,98 pada set uji. Validasi silang menunjukkan konsistensi model dengan akurasi rata-rata 97,99% dan skor F1 sebesar 0,96. Hasil ini membuktikan bahwa kombinasi PCA, autoencoder, KNN, dan LightGBM mampu secara efektif mengenali kemiripan lagu berdasarkan fitur audio. Model utama menunjukkan average cosine similarity 0.9936 dengan diversity 0.0059 dan coverage 32.2%, menandakan rekomendasi yang sangat relevan dan konsisten dalam menampilkan lagu dengan karakter serupa. Secara kualitatif, kemunculan lagu yang berulang meskipun berbeda genre menunjukkan bahwa model lebih menekankan kesamaan karakter audio, sesuai dengan tujuan sistem untuk merekomendasikan lagu berdasarkan kemiripan suara.

DAFTAR PUSTAKA

- [1] A. I. Putra and R. R. Santika, “Implementasi Machine Learning dalam Penentuan Rekomendasi Musik dengan Metode Content-Based Filtering,” *Eduomatic J. Pendidik. Inform.*, vol. 4, no. 1, pp. 121–130, 2020, doi: 10.29408/edumatic.v4i1.2162.
- [2] H. Siefkes, L. C. Oliveira, R. Koppel, and W. Hogan, “Machine Learning – Based Critical,” pp. 1–13, 2024, doi: 10.1109/CONFLUENCE47617.2020.9058196.
- [3] S. Pencarian *et al.*, “Lagu Untuk Pengalaman Mendengarkan Yang Lebih Personal Menggunakan Content-Based Filtering,” vol. 8, no. 2, pp. 169–174, 2025.
- [4] A. T. R. Dani, V. Ratnasari, L. Ni'matuzzahroh, I. C. AVIANTHOLIB, R. NOVIDIANTO, and N. Y. ADRIANINGSIH, “Analisis Klasifikasi Artist Music Menggunakan Model Regresi Logistik Biner Dan Analisis Diskriminan,” *Jambura J. Probab. Stat.*, vol. 3, no. 1, pp. 1–10, 2022, doi: 10.34312/jjps.v3i1.13708.
- [5] Y. Jiang and F. H. F. Leung, “Vector-Based Feature Representations for Speech Signals: From Supervector to Latent Vector,” *IEEE Trans. Multimed.*, vol. 23, no. c, pp. 2641–2655, 2021, doi: 10.1109/TMM.2020.3014559.
- [6] K. R. Putra and M. A. Rachman, “Perbandingan Metode Content-based , Collaborative dan Hybrid Filtering pada Sistem Rekomendasi Lagu,” vol. 9, no. 2, pp. 179–193, 2024, doi: <https://doi.org/10.26760/mindjournal.v1i1.49>.
- [7] X. Wang, Z. Wang, Y. Zhang, X. Jiang, and Z. Cai, “Latent representation learning based autoencoder for unsupervised feature selection in hyperspectral imagery,” *Multimed. Tools Appl.*, vol. 81, no. 9, pp. 12061–12075, 2022, doi: 10.1007/s11042-020-10474-8.
- [8] T. Sugiura, Y. Yamagishi, and Y. Kishimoto, “Leveraging LightGBM Ranker for Efficient Large-Scale News Recommendation Systems,” *ACM Int. Conf. Proceeding Ser.*, pp. 27–31, 2024, doi: 10.1145/3687151.3687156.
- [9] Karthik V, Savita Chaudhary, and Radhika A D, “Feature Extraction in Music information retrieval using Machine Learning Algorithms,” *Int. J. Data Informatics Intell. Comput.*, vol. 1, no. 1, pp. 1–10, 2024, doi: 10.59461/ijdiic.v1i1.11.
- [10] M. U. Hassan, N. Zafar, H. Ali, I. Yaqoob, S. A. A. Alaliyat, and I. A. Hameed, “Collaborative Filtering Based Hybrid Music Recommendation System,” *Lect. Notes Networks Syst.*, vol. 350, pp. 239–249, 2022, doi: 10.1007/978-981-16-7618-5_21.
- [11] M. Casella, P. Dolce, M. Ponticorvo, and D. Marocco, “Autoencoders as an alternative approach to Principal Component Analysis for dimensionality reduction. An application on simulated data from psychometric models,” *CEUR Workshop Proc.*, vol. 3100, pp. 0–2, 2021.

- [12] A. Y. Timur and A. N. Rohman, "Indonesia Metode Content-Based Filtering Dan Cosine Similarity," vol. 13, no. 1, pp. 1415–1423, 2025.
- [13] A. E. Karrar, "The Effect of Using Data Pre-Processing by Imputations in Handling Missing Values," *Indones. J. Electr. Eng. Informatics*, vol. 10, no. 2, pp. 375–384, 2022, doi: 10.52549/ijeei.v10i2.3730.
- [14] L. Wang, S. Jiang, and S. Jiang, "A feature selection method via analysis of relevance, redundancy, and interaction," *Expert Syst. Appl.*, vol. 183, no. August 2019, p. 115365, 2021, doi: 10.1016/j.eswa.2021.115365.
- [15] M. Alkhayrat, M. Aljnidi, and K. Aljoumaa, "A comparative dimensionality reduction study in telecom customer segmentation using deep learning and PCA," *J. Big Data*, vol. 7, no. 1, 2020, doi: 10.1186/s40537-020-0286-0.
- [16] T. Pratiwi, A. Sunyoto, and D. Ariatmanto, "Music Genre Classification Using K-Nearest Neighbor and Mel-Frequency Cepstral Coefficients," *Sinkron*, vol. 8, no. 2, pp. 861–867, 2024, doi: 10.33395/sinkron.v8i2.12912.
- [17] X. Cheng, K. Liu, X. Hu, T. Liu, C. Che, and C. Zhu, "Comparative Analysis of Machine Learning Models for Music Recommendation," *Theor. Nat. Sci.*, vol. 53, no. 1, pp. 249–254, 2024, doi: 10.54254/2753-8818/53/20240233.