



Klasifikasi Genre Musik Menggunakan Machine Learning

Garda Zidane Dhamara¹, Sucipto^{2*}, Arie Nugroho³

Fakultas Teknik dan Ilmu Komputer, Program Studi Sistem Informasi, Universitas PGRI Kediri, Kota Kediri, Indonesia

Email: ¹zidmara08@email.com, ^{2*}sucipto@unpkediri.ac.id

Abstrak- Penelitian ini mengkaji implementasi klasifikasi genre musik menggunakan Machine Learning untuk mengembangkan aplikasi rekomendasi musik yang akurat dan efisien. Permasalahan utama yang ditangani adalah kesulitan dalam mengidentifikasi genre musik secara otomatis akibat kemiripan fitur antar genre dan beragamnya preferensi pengguna, yang menyebabkan rendahnya akurasi sistem rekomendasi tradisional. Metode yang digunakan melibatkan penerapan algoritma Machine Learning Naive Bayes, K-Nearest Neighbors (K-NN), Random Forest, dan Support Vector Machine (SVM) pada dataset GTZAN. Tujuan penelitian ini adalah membangun sistem klasifikasi genre musik otomatis sebagai dasar bagi sistem rekomendasi musik yang lebih cerdas. Hasil pengujian menunjukkan bahwa algoritma SVM memberikan performa terbaik dengan akurasi sebesar 83%, diikuti oleh Random Forest sebesar 73%, K-NN sebesar 71%, dan Naive Bayes sebesar 66%. Temuan ini mengindikasikan bahwa Machine Learning efektif dalam mengelompokkan genre musik dan mampu meningkatkan akurasi rekomendasi. Penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam pengembangan aplikasi musik berbasis kecerdasan buatan.

Kata Kunci: Klasifikasi; Genre Musik; Machine Learning; Rekomendasi Musik; Sistem Informasi

Abstract- This study examines the implementation of music genre classification using Machine Learning to develop an accurate and efficient music recommendation application. The main problem addressed is the difficulty of automatically identifying music genres due to feature similarities among genres and diverse user preferences, which reduce the accuracy of conventional recommendation systems. The method applied involves four Machine Learning algorithms: Naive Bayes, K-Nearest Neighbors (K-NN), Random Forest, and Support Vector Machine (SVM), using the GTZAN dataset. The objective of this research is to build a system capable of automatically classifying music genres and serving as the foundation for smarter music recommendations. The experimental results show that the SVM algorithm achieved the highest accuracy at 83%, followed by Random Forest at 73%, K-NN at 71%, and Naive Bayes at 66%. These findings indicate that Machine Learning is effective in music classification, contributing to improved recommendation accuracy. This research is expected to make a significant contribution to the development of intelligent, AI-based music applications.

Keywords: Classification; Music Genre; Machine Learning; Music Recommendation; Information Systems

1. PENDAHULUAN

Musik adalah media yang digunakan oleh manusia untuk mengekspresikan pesan, ekspresi, dan emosi pribadi mereka [1]. Seiring berkembangnya dunia musik di Indonesia, lahirlah berbagai genre musik baru baik dari dalam maupun luar negeri [2]. Genre Musik adalah pengklasifikasian beberapa gaya musik terhadap tipe tertentu. Mulai dari genre *Pop*, *Rock*, *Jazz*, *Dangdut*, dan masih banyak lagi. Oleh karena itu, mengembangkan model *machine learning* yang lebih baik yang secara otomatis mengklasifikasikan musik berdasarkan *genre* akan sangat bermanfaat bagi penyedia *streaming* musik *online*. Tujuan dibuatnya genre musik adalah untuk membedakan warna musik antara satu grup musik dengan grup yang lainnya. Pembagian musik berdasarkan genre memudahkan pengguna *Spotify* memilih dan menemukan lagu favoritnya di berbagai *genre* dan *playlist*. Permasalahan penelitian muncul akibat adanya perbedaan preferensi dan kecenderungan pengguna ketika mendengarkan sebuah lagu dan musik, sehingga pemahaman dan pengelompokan musik menjadi kompleks [3]. Hal tersebut dikarenakan beragamnya warna dan pola musik yang membuat kaum awam kesulitan dalam membedakan *genre* musik

Penelitian genre musik telah dilakukan oleh Carl Ray Wairata dan kawan-kawan pada tahun 2021 dengan judul "Pengklasifikasian *Genre* Musik Indonesia Menggunakan *Convolutional Neural Network*". Penelitian tersebut menggunakan algoritma *CNN* (*Convolutional Neural Network*) untuk mengklasifikasi *genre* musik. Hasil dari penelitian ini setelah melakukan percobaan pada tabel pertama sampai dengan yang kesepuluh dengan 35 *epoch* merupakan yang paling efektif karena mencapai akurasi validasi 100% dan akurasi tes 81,33% [4]. Penelitian lain juga dilakukan oleh Salma Navisa dan kawan-kawan pada tahun 2021 dengan judul "Komparasi Algoritma Klasifikasi *Genre* Musik pada *Spotify* Menggunakan *CRISP-DM*". Penelitian tersebut menggunakan algoritma *K-NN*, *Naive Bayes*, dan *Random Forest* untuk mengklasifikasikan *genre*. Hasil dari penelitian ini untuk algoritma *Naive Bayes* memiliki accuracy 58,91%. Untuk algoritma *K-NN* memiliki accuracy 33,30 %. Sedangkan algoritma *Random Forest* memiliki accuracy 52,81% [5]. Penelitian lain juga dilakukan oleh I Gede ismanmadi pada tahun 2023 dengan judul "Perbandingan Kinerja Algoritma *K-NN* dan *SVM* dalam Sistem Klasifikasi *Genre* Musik Gamelan Bali". Penelitian tersebut menggunakan algoritma *K-NN* dan *SVM* untuk mengklasifikasikan *genre* musik Gamelan Bali. Hasil dari penelitian tersebut membandingkan persentase akurasi 85,38% pada algoritma *K-NN* dengan pemilihan nilai tetangga $k=3$ berdasarkan nilai Euclidean Distance sebagai ukuran jarak tetangga. Sementara itu, sistem klasifikasi *SVM* mendapatkan akurasi sebesar 66,9% [6]. Penelitian lain juga dilakukan oleh Sally Lutfiani dan kawan-kawan pada tahun 2023 dengan judul "Perbandingan Metode



Extreme Gradient Boosting Dan Metode *Decision Tree* Untuk Klasifikasi *Genre* Musik”. Penelitian tersebut menggunakan metode *Extreme Gradient Boosting* dan *Decision Tree*. Hasil dari penelitian tersebut untuk algoritma *Decision Tree* memiliki akurasi 51%, sedangkan untuk algoritma *Extreme Gradient Boosting* memiliki akurasi 72%. Penelitian lain dilakukan oleh Yusti Vita Via dan kawan-kawan pada tahun 2022 dengan judul “Analisa Algoritma *Convolutional Neural Network (CNN)* Pada Klasifikasi *Genre* Musik Berdasar Durasi Waktu”. Penelitian tersebut menggunakan metode *CNN (Convolution Neural Network)* untuk mengklasifikasikan *genre* lagu berdasarkan durasi waktu. Hasil dari penelitian tersebut untuk *dataset* durasi 10 detik memiliki *accuracy* 0.81. Sedangkan untuk *dataset* durasi 30 detik memiliki *accuracy* 0.58 [7].

Berbagai algoritma pembelajaran mesin digunakan untuk mengklasifikasikan *genre* musik. Tujuannya adalah untuk mengukur seberapa tinggi akurasi klasifikasi yang diperoleh. Beberapa penelitian telah menggunakan metode yang berbeda untuk melakukan klasifikasi *genre* musik seperti *K-NN (K-Nearest Neighbors)*, *CNN (Convolutional Neural Network)*, *SVM (Support Vector Machine)*, *Naive Bayes*, *Multi Layer Perseption*. Pada beberapa penelitian tersebut menggunakan dataset GTZAN.

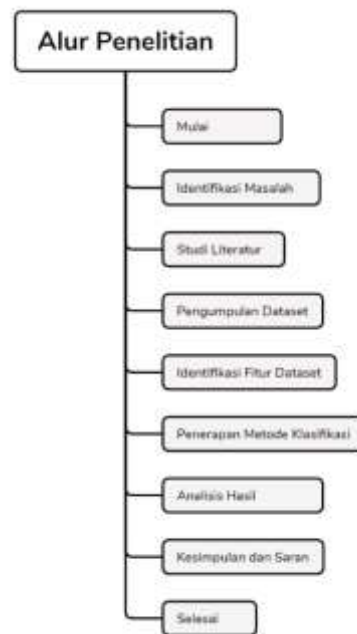
Diantara metode tersebut memiliki performa model yang berbeda. Maka dilakukanlah penelitian ini untuk mengetahui hasil performa klasifikasi *genre* musik menggunakan *dataset GTZAN* dengan empat metode algoritma. Pada penelitian ini menggunakan metode *Naive Bayes*, *K-NN*, *Random Forest*, dan *SVM* untuk membandingkan performa masing-masing model.

Dengan demikian, tujuan penelitian ini untuk mengetahui perbedaan *genre* musik yang tepat dengan melakukan klasifikasi. Oleh karena itu, penelitian ini penting untuk dilaksanakan agar permasalahan tersebut bisa terselesaikan.

2. METODOLOGI PENELITIAN

2.1 Alur Penelitian

Alur penelitian dilaksanakan sesuai dengan langkah-langkah yang dirancang agar penelitian bisa berjalan dengan baik. Berikut untuk diagram alur penelitian :



Gambar 1. Diagram Alur Penelitian

Alur penelitian pada gambar 3.1 mengacu pada penelitian “PERBANDINGAN METODE *EXTREME GRADIENT BOOSTING* DAN METODE *DECISION TREE* UNTUK KLASIFIKASI *GENRE* MUSIK” [8]. Untuk penjelasan detail setiap tahapan dimulai dari :

- a Identifikasi Masalah : Pada tahap ini, identifikasi masalah atau fenomena yang ingin diteliti. Pada tahap identifikasi ini terdapat penjelasan mengenai masalah yang sedang dihadapi
- b Studi Literatur : Peneliti melakukan kajian pustaka untuk menggali lebih dalam mengenai permasalahan yang telah diidentifikasi. Tujuannya adalah untuk memahami teori-teori yang relevan, penelitian sebelumnya, dan kerangka kerja yang akan digunakan.



- c Pengumpulan *Dataset* : Setelah memiliki pemahaman yang baik tentang masalah pada penelitian, peneliti mengumpulkan data yang diperlukan dan melakukan klasifikasi untuk menjawab pertanyaan penelitian. Data yang diperoleh pada penelitian ini diperoleh dari situs Kaggle.com. Pada penelitian ini *dataset GTZAN* berupa data kuantitatif dan data numerik.
- d Identifikasi Fitur *Dataset* : Data yang telah dikumpulkan kemudian dianalisis untuk mengidentifikasi fitur-fitur penting yang relevan dengan pertanyaan penelitian. Fitur-fitur ini akan menjadi dasar untuk tahap selanjutnya.
- e Penerapan Metode Klasifikasi : Pada tahap ini, peneliti akan memilih dan menerapkan metode klasifikasi yang sesuai. Metode yang digunakan pada penelitian ini adalah algoritma machine learning *Naive Bayes*, *K-NN*, *Random Forest*, dan *SVM*.
- f Analisis Hasil : Setelah menerapkan metode klasifikasi, hasil analisis akan diperoleh. Hasil ini kemudian akan diinterpretasikan untuk menjawab pertanyaan penelitian. Tujuan analisis hasil adalah untuk mengevaluasi performa model, seperti *accuracy*, *precision*, dan *recall*.
- g Kesimpulan dan Saran : Peneliti menyimpulkan hasil penelitian dan memberikan saran untuk penelitian selanjutnya atau implikasi praktis dari hasil penelitian.

2.2 Klasifikasi

Klasifikasi adalah pengelompokan objek ke dalam kelas-kelas tertentu yang biasanya disebut dengan kelas (class) berdasarkan pengelompokannya[9]. Proses klasifikasi umumnya diawali dengan membangun sebuah model menggunakan data pelatihan (*training data*) yang telah diberi label kelas. Data pelatihan ini digunakan untuk mengidentifikasi pola atau hubungan antara fitur-fitur pada data dengan kelas yang sesuai. Setelah model selesai dibangun, model tersebut dapat digunakan untuk mengklasifikasikan data baru yang belum diketahui kelasnya. (Rinanda et al., 2022). Klasifikasi bertujuan untuk mempelajari struktur dari data pelatihan agar dapat melakukan prediksi terhadap data uji dengan tingkat akurasi yang tinggi.

2.3 Machine Learning

Machine learning adalah bidang ilmu komputer yang berhubungan dengan metode merancang algoritma yang dapat mempelajari atau beradaptasi dengan pola data tanpa diprogram secara eksplisit [10]. *Machine learning* berfokus pada pengembangan algoritma dan teknik yang memungkinkan komputer untuk belajar dari data dan membuat keputusan. *Machine learning* bekerja ketika tersedia data sebagai input untuk dilakukan analisa terhadap kumpulan data besar (*big data*) guna menemukan pola tertentu. Melalui pendekatan ini, sistem dapat mengevaluasi dan mengidentifikasi pola tersembunyi dari kumpulan data yang sangat besar (*big data*). Proses ini melibatkan berbagai tahapan seperti praproses data, pelatihan model, validasi, hingga pengujian model. Dengan menemukan pola-pola dari data, mesin dapat melakukan prediksi atau pengambilan keputusan secara otomatis terhadap data baru yang sebelumnya belum pernah dikenali.

2.4 K-Nearest Neighbor

K-Nearest Neighbor (K-NN) merupakan salah satu algoritma klasifikasi yang cukup populer karena kesederhanaan dan efektivitasnya dalam mengelompokkan data berdasarkan kemiripan. *K-NN* merupakan algoritma untuk pengklasifikasian objek berdasarkan data latih yang memiliki jarak terdekat. Cara kerja algoritma dari *K-NN* yaitu menentukan beberapa nilai *K* (jumlah tetangga terdekat) yang berbeda pada data latih dan data uji. [11] Nilai *k* yang optimal untuk algoritma ini bergantung pada nilai sebuah data. Umumnya, semakin tinggi nilai *k* maka semakin kecil dampak *error* dan *noise* pada proses klasifikasi. [12] Proses klasifikasi dilakukan dengan mencari titik *c* yang paling dekat dengan titik *c* yang baru (*nearest neighbor*). Teknik pencarian jarak tetangga terdekat biasanya dilakukan menggunakan rumus *Euclidean Distance*. *Euclidean Distance* adalah rumus untuk menentukan jarak antara dua titik dalam dua dimensi. *Euclidean Distance* paling umum digunakan untuk melakukan perhitungan data numerik. [13] Berikut ini adalah rumus *Euclidean Distance* yang digunakan dalam algoritma *K-NN*. [14]

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

2.5 Naive Bayes

Naive Bayes merupakan salah satu algoritma klasifikasi yang didasarkan pada teori probabilitas *Bayes*, yang terkenal karena kesederhanaan dan efisiensinya dalam menangani data berukuran besar. *Naive Bayes* adalah metode pengklasifikasian probabilistik sederhana yang menghitung sekumpulan probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari kumpulan data yang dipilih. [15] Algoritma ini mengasumsikan bahwa setiap fitur atau atribut bersifat independen satu sama lain, meskipun dalam kenyataannya hal tersebut jarang terjadi—oleh karena itu disebut "*naive*" atau naif. Namun, asumsi kesederhanaan ini justru membuat *Naive Bayes* sangat cepat dan efektif dalam banyak aplikasi klasifikasi, terutama dalam teks *mining*, *spam detection*, dan analisis sentimen.





Algoritma *Naive Bayes* merupakan metode klasifikasi yang menggunakan teknik probabilistik dan statistik untuk membagi suatu permasalahan ke dalam kelas-kelas label. [16] *Naive Bayes* menggunakan metode probabilitas dan statistik untuk menghasilkan data yang terklasifikasi dan memprediksi masa datang berdasarkan masa lalu. Di bawah ini adalah persamaan teorema bayes [15] :

$$P(H|X) = \frac{P(X|H).P(H)}{P(X)} \quad (2)$$

Di mana :

X : Data dengan class yang belum diketahui

H : Hipotesis data merupakan suatu class spesifik

P(H|X) : Probabilitas hipotesis H berdasar kondisi X (posteriori probabilitas)

P(H) : Probabilitas hipotesis H (prior probabilitas)

P(X|H) : Probabilitas X berdasarkan kondisi pada hipotesis H

P(X) : Probabilitas X

2.6 Random Forest

Random Forest adalah suatu algoritma pembelajaran mesin dengan menghasilkan banyak pohon keputusan. *Random Forest* merupakan kombinasi dari metode *Bagging* dan *Random Sub Spaces* yang digunakan untuk klasifikasi dan regresi. Ada 3 aspek penting dalam penggunaan metode algoritma ini diantaranya: (1) membangun pohon prediksi dengan melakukan *bootstrap sampling*, (2) memprediksi dengan predictor acak pada masing-masing pohon keputusan, dan (3) selanjutnya metode ini dapat melakukan prediksi dengan menggabungkan hasil dari setiap pohon keputusan melalui *majority vote* untuk menentukan klasifikasi atau regresi. [17]

Membentuk pohon keputusan pada metode *Random Forest* sama dengan proses pada *Classification and Regression Tree (CART)*, hanya saja pada *Random Forest* tidak dilakukan *pruning* (pemangkasan). Indeks Gini digunakan untuk memilih fitur di setiap simpul internal dari pohon keputusan. Nilai Indeks Gini dapat dihitung sebagai berikut :

$$Gini(S_i) = 1 - \sum_{i=0}^{c-1} p_i^2 \quad (3)$$

dengan p_i merupakan frekuensi relative kelas C_i di dalam set.

C_i merupakan kelas untuk $i = 1, \dots, c-1$, dan c adalah jumlah kelas yang telah ditentukan. Kualitas *split* pada fitur k ke dalam *subset* S_i merupakan jumlah sampel milik kelas C_i , kemudian dihitung sebagai jumlah pertimbangan indikasi Gini dari *subset* yang dihasilkan. Data dapat dihitung dengan rumus sebagai berikut :

$$Gini_{split} = \sum_{i=0}^{k-1} \left(\frac{n_i}{n}\right) Gini S_i \quad (4)$$

dimana n_i merupakan jumlah sampel dalam *subset* S_i setelah di *split* dan n merupakan jumlah sampel di *node* yang diberikan.

Misalkan $\{h(x, \Theta_k), k = 1, \dots\}$ dimana $\{\Theta_k\}$ merupakan *vector random* yang *independent identically distributed (iid)* dan tiap pohon memilih kelas yang paling banyak dari rata rata (*majority vote*). Untuk *Random Forest*, batas atas dapat diturunkan untuk kesalahan generalisasi dalam hal dua *parameter* yang mengukur seberapa kuat pengklasifikasian individu dan ketergantungan diantara keduanya.

Fungsi margin untuk RF adalah :

$$mr(X, Y) = P_{\Theta}(h(X, \Theta) = Y) - \max_{j \neq y} P_{\Theta}(h(X, \Theta) = j) \quad (5)$$

dan kekuatan himpunan pengklasifikasi $\{h(X, \Theta)\}$ adalah

$$s = E_{X,Y} mr(X, Y) \quad (6)$$

Dengan asumsi $s \geq 0$, ketidaksamaan *Chebychev* serta penurunan variansi mr dari fungsi margin untuk metode RF, akan didapatkan persamaan batas atas kesalahan generalisasi sebagai berikut :

$$PE \leq \frac{\bar{p}(1 - s_2)}{s_2} \quad (7)$$





Dimana $\bar{\rho}$ adalah nilai rata-rata korelasi, yaitu :

$$\bar{\rho} = \frac{E_{\theta, \theta'}(\rho(\theta, \theta')sd(\theta)sd(\theta'))}{E_{\theta, \theta'}sd(\theta)sd(\theta')} \quad (8)$$

2.7 Support Vector Machine

Algoritma *Support Vector Machine (SVM)* adalah algoritma berbasis diskriminasi yang bertujuan untuk menemukan batas pemisahan optimal yang disebut *hyperplane* untuk membedakan kelas dari satu sama lain. Sampel yang paling dekat dengan *hyperplanes* ini disebut vektor dukungan, dan perbedaan tersebut dinyatakan sebagai jumlah bobot dari *subset* sampel yang membatasi kerumitan masalah. [18] Untuk memisahkan dua klasifikasi melalui *hyperplane* yang optimal menggunakan persamaan (1) dan (2) berikut :

$$f(x) = w \cdot x + b \quad (9)$$

Atau

$$f(x) = \sum_{i=1}^n a_i y_i K(x, x_i) + b \quad (10)$$

Harus berada diantara dua kelas sampel *hyperplane* pemisah yang optimal

$$y_i(wx_i + b) \geq 1 (i = 1, 2, \dots, n) \quad (11)$$

Dimana :

w : beban vektor (garis tegak lurus)

x : titik data masukan SVM

b : bias

a_i : nilai bobot setiap titik data

$K(x, x_i)$: fungsi karnel

2.8 MFCC (Mel-Frequency Cepstral Coefficients)

MFCC (Mel-Frequency Cepstral Coefficients) adalah representasi dari spektrum daya (*power spectrum*) suatu sinyal suara berdasarkan skala *mel*. *MFCC* mempunyai kemampuan seperti pada pendengaran manusia, jadi *MFCC* dapat mengadaptasi pendengaran manusia dalam bentuk digital. Selain itu penggunaannya adalah untuk mengekstrak data sinyal suara. Keunggulan penggunaan *MFCC* adalah *MFCC* mampu menangkap karakteristik suara yang sangat penting bagi pengenalan suara atau dapat menangkap informasi yang penting dalam sinyal suara, menghasilkan data seminimal mungkin tanpa menghilangkan informasi penting yang ada di dalam sinyal dan mempunyai kemampuan seperti organ pendengaran suara dalam mengenali sinyal suara.

3. HASIL DAN PEMBAHASAN

3.1 Deskripsi Data

Dataset GTZAN dikumpulkan oleh George Tzanetakis dan Perry Cook pada tahun 2002. *File audio* dikumpulkan antara tahun 2000 dan 2001 dari berbagai sumber, termasuk *CD* pribadi, siaran radio, dan rekaman mikrofon. Keberagaman ini bertujuan untuk merepresentasikan kondisi perekaman yang berbeda-beda. *Dataset* ini awalnya digunakan dalam makalah penting mereka, "*Musical genre classification of audio signals*," yang diterbitkan di *IEEE Transactions on Audio and Speech Processing*.

Dataset GTZAN memiliki karakteristik utama sebagai berikut:

- a Ukuran dan Durasi : Terdiri dari 1000 trek audio, yang masing-masing berdurasi tepat 30 detik.
- b Genre : *Dataset* ini dibagi menjadi 10 *genre* musik yang berbeda: *blues*, klasik, *country*, *disco*, *hip-hop*, *jazz*, *metal*, *pop*, *reggae*, dan *rock*.
- c Jumlah Sampel per *Genre* : Setiap dari 10 *genre* berisi 100 trek audio, menjadikannya *dataset* yang seimbang dalam hal distribusi kelas.





d *Format Audio* : Semua file audio dalam format .wav, dengan sampling rate 22050 Hz, resolusi 16-bit, dan bersifat mono-channel.

Tabel 1. Isi Dataset GTZAN

Folder ke-	Genre	Jumlah File	Format
1	Blues	100	.wav
2	Classical	100	.wav
3	Country	100	.wav
4	Disco	100	.wav
5	Hip Hop	100	.wav
6	Jazz	100	.wav
7	Metal	100	.wav
8	Pop	100	.wav
9	Reggae	100	.wav
10	Rock	100	.wav

Namun, karena *genre* yang dipakai pada penelitian ini hanya 7, maka total data dari *dataset GTZAN* yang dipakai hanya 700 file musik. Tujuh *genre* tersebut adalah *Classical, Jazz, Pop, Metal, Reggae, Blues dan Rock*. Fitur yang diekstraksi adalah *Mel-Frequency Cepstral Coefficients (MFCC)* melalui *coding python*. Nilai koefisien dari *Mel-Frequency Cepstral Coefficients (MFCC)* yang digunakan adalah 30. Pembagian *dataset* untuk seluruh metode klasifikasi adalah 20% dari seluruh data akan dialokasikan untuk data pengujian. Sedangkan sisanya 80% akan digunakan sebagai data pelatihan

3.2 Temuan Hasil Penelitian

Pada penelitian ini menggunakan metode *Naive Bayes, K-Nearest Neighbors (K-NN), Random Forest, dan Support Vector Machine (SVM)*. Berikut untuk metrik evaluasi kinerja dari masing-masing metode klasifikasi yang digunakan.

Tabel 2. Metrik Evaluasi Metode *Naive Bayes*

Kategori <i>Genre</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Support</i>
<i>Blues</i>	0.46	0.40	0.43	15
<i>Classical</i>	0.75	0.82	0.78	22
<i>Jazz</i>	0.59	0.54	0.57	24
<i>Metal</i>	0.84	0.80	0.82	20
<i>Pop</i>	0.76	0.88	0.81	25
<i>Reggae</i>	0.40	0.46	0.43	13
<i>Rock</i>	0.61	0.52	0.56	21
<i>Accuracy</i>			0.66	140
<i>Macro Avg.</i>	0.63	0.63	0.63	140
<i>Weighted Avg.</i>	0.65	0.66	0.65	140

Tabel 2. menampilkan hasil evaluasi performa model klasifikasi menggunakan metode *Naive Bayes* terhadap data musik yang terdiri dari tujuh *genre: blues, classical, jazz, metal, pop, reggae, dan rock*. Evaluasi dilakukan dengan menggunakan tiga metrik utama yaitu *precision, recall, dan f1-score*, serta jumlah data (*support*) pada masing-masing kelas. Secara umum, model memperoleh akurasi sebesar 0.6571 atau sekitar 66%. Dari hasil tersebut, terlihat bahwa performa model bervariasi untuk setiap *genre*.

Genre yang memperoleh performa tertinggi adalah "*metal*" dan "*pop*", dengan *f1-score* masing-masing sebesar 0.84 dan 0.81. Hal ini menunjukkan bahwa model cukup baik dalam mengenali dan mengklasifikasikan lagu-lagu pada dua *genre* tersebut. *Genre "classical"* juga menunjukkan performa yang cukup baik dengan *f1-score* sebesar 0.78. Di sisi lain, *genre* seperti "*blues*", "*reggae*", dan "*rock*" memiliki nilai *f1-score* yang relatif rendah, yaitu 0.43, 0.42, dan 0.56, yang menunjukkan bahwa model mengalami kesulitan dalam mengklasifikasikan *genre-genre* tersebut secara akurat.

Secara keseluruhan, nilai rata-rata dari *precision, recall, dan f1-score* pada evaluasi makro masing-masing adalah 0.63, 0.63, dan 0.63, sedangkan pada evaluasi berbobot (*weighted average*) masing-masing sebesar 0.65, 0.66, dan 0.65. Nilai ini menunjukkan bahwa meskipun model cukup baik dalam mengenali *genre* tertentu, performa secara keseluruhan masih bisa ditingkatkan terutama untuk *genre-genre* dengan jumlah data yang lebih sedikit atau memiliki kemiripan fitur dengan





genre lain. Evaluasi ini penting untuk memberikan gambaran tentang kekuatan dan kelemahan model dalam menangani *data multi-kelas* seperti klasifikasi *genre* musik.

Tabel 3. Metrik Evaluasi Metode *K-Nearest Neighbors (KNN)*

Kategori <i>Genre</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Support</i>
<i>Blues</i>	0.53	0.67	0.59	15
<i>Classical</i>	0.91	0.91	0.91	22
<i>Jazz</i>	0.68	0.71	0.69	24
<i>Metal</i>	0.84	0.80	0.82	20
<i>Pop</i>	0.81	0.84	0.82	25
<i>Reggae</i>	0.64	0.54	0.58	13
<i>Rock</i>	0.50	0.43	0.46	21
<i>Accuracy</i>			0.71	140
<i>Macro Avg.</i>	0.70	0.70	0.70	140
<i>Weighted Avg.</i>	0.71	0.71	0.71	140

Tabel 3. menyajikan hasil evaluasi model klasifikasi *genre* musik menggunakan metode *K-Nearest Neighbors (K-NN)*. Evaluasi dilakukan dengan menggunakan metrik *precision*, *recall*, dan *f1-score* untuk masing-masing dari tujuh *genre* musik, yaitu *blues*, *classical*, *jazz*, *metal*, *pop*, *reggae*, dan *rock*. Model ini menunjukkan akurasi keseluruhan sebesar 0.71 atau 71%, yang berarti performa model secara umum lebih baik dibandingkan metode *Naive Bayes* yang hanya mencapai akurasi sekitar 66%.

Dari hasil klasifikasi, *genre* dengan performa tertinggi adalah "*classical*", "*metal*", dan "*pop*" yang masing-masing memiliki nilai *f1-score* sebesar 0.91, 0.82, dan 0.81. Hal ini menunjukkan bahwa model *K-NN* mampu mengklasifikasikan lagu-lagu dalam tiga *genre* tersebut dengan cukup akurat dan konsisten. Sebaliknya, performa model terhadap *genre* "*rock*" dan "*reggae*" masih rendah, dengan *f1-score* masing-masing hanya 0.46 dan 0.58. *Genre* "*blues*" juga menunjukkan performa yang sedang, dengan *f1-score* sebesar 0.59.

Secara keseluruhan, nilai rata-rata makro (*macro average*) untuk *precision*, *recall*, dan *f1-score* masing-masing adalah 0.70, sedangkan nilai rata-rata berbobot (*weighted average*) sedikit lebih tinggi yaitu 0.71. Ini menunjukkan bahwa *K-NN* tidak hanya mampu mengklasifikasikan *genre* dengan *data* besar secara efektif, tetapi juga relatif seimbang dalam menangani *genre* dengan jumlah *data* yang lebih sedikit. Dibandingkan dengan metode *Naive Bayes*, model *K-NN* menunjukkan peningkatan kinerja di hampir semua *genre*, yang membuatnya menjadi alternatif yang lebih andal dalam konteks klasifikasi *genre* musik berdasarkan *data* yang digunakan.

Tabel 4. Metrik Evaluasi Metode *Random Forest*

Kategori <i>Genre</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Support</i>
<i>Blues</i>	0.73	0.73	0.73	15
<i>Classical</i>	0.91	0.91	0.91	22
<i>Jazz</i>	0.71	0.83	0.83	24
<i>Metal</i>	0.94	0.85	0.85	20
<i>Pop</i>	0.86	0.76	0.76	25
<i>Reggae</i>	0.37	0.54	0.54	13
<i>Rock</i>	0.50	0.38	0.38	21
<i>Accuracy</i>			0.73	140
<i>Macro Avg.</i>	0.72	0.72	0.71	140
<i>Weighted Avg.</i>	0.74	0.73	0.73	140

Tabel 4. menampilkan hasil evaluasi performa model klasifikasi *genre* musik menggunakan metode *Random Forest*. Evaluasi ini mencakup tiga metrik utama, yaitu *precision*, *recall*, dan *f1-score*, serta jumlah data (*support*) untuk masing-masing dari tujuh *genre* musik: *blues*, *classical*, *jazz*, *metal*, *pop*, *reggae*, dan *rock*. Secara keseluruhan, model *Random Forest* berhasil mencapai akurasi sebesar 0.73 atau 73%, yang merupakan nilai tertinggi dibandingkan dua metode sebelumnya, yaitu *Naive Bayes* dan *K-NN*.

Genre dengan performa paling menonjol pada metode ini adalah "*metal*", "*pop*", dan "*classical*", masing-masing dengan *f1-score* sebesar 0.89, 0.87, dan 0.91. Ini menunjukkan bahwa model sangat baik dalam mengidentifikasi lagu-lagu dari ketiga *genre* tersebut. *Genre* "*jazz*" juga menunjukkan performa yang baik dengan *f1-score* 0.77. Sebaliknya, *genre* "*rock*" dan "*reggae*" memperoleh *f1-score* yang lebih rendah, masing-masing sebesar 0.43 dan 0.44, menunjukkan bahwa model masih kesulitan dalam mengklasifikasikan dua *genre* ini secara akurat.



Secara rata-rata, nilai *macro average* untuk *precision*, *recall*, dan *f1-score* masing-masing adalah 0.72, sedangkan *weighted average* sedikit lebih tinggi yaitu *precision* 0.74, *recall* 0.73, dan *f1-score* 0.73. Hasil ini mencerminkan bahwa metode *Random Forest* mampu memberikan performa yang seimbang pada berbagai kelas, baik yang memiliki jumlah data besar maupun kecil. Dibandingkan dengan metode *Naive Bayes* dan *K-NN*, *Random Forest* memberikan performa yang lebih stabil dan akurat secara keseluruhan dalam klasifikasi *genre* musik berdasarkan dataset yang digunakan.

Tabel 5. Metrik Evaluasi Metode *Support Vector Machine (SVM)*

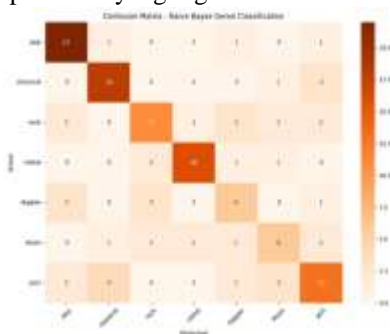
Kategori <i>Genre</i>	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>	<i>Support</i>
<i>Blues</i>	0.82	0.40	0.43	15
<i>Classical</i>	0.95	0.90	0.86	20
<i>Jazz</i>	0.75	0.95	0.95	20
<i>Metal</i>	0.95	0.90	0.82	20
<i>Pop</i>	0.88	0.75	0.92	20
<i>Reggae</i>	0.75	0.75	0.81	20
<i>Rock</i>	0.72	0.65	0.75	20
<i>Accuracy</i>			0.83	140
<i>Macro Avg.</i>	0.83	0.83	0.83	140
<i>Weighted Avg.</i>	0.83	0.83	0.83	140

Tabel 5. menunjukkan hasil evaluasi performa model klasifikasi *genre* musik menggunakan metode *Support Vector Machine (SVM)*. Evaluasi ini menggunakan tiga metrik utama yaitu *precision*, *recall*, dan *f1-score* untuk masing-masing dari tujuh *genre* musik: *blues*, *classical*, *jazz*, *metal*, *pop*, *reggae*, dan *rock*. Secara keseluruhan, model *SVM* mencapai akurasi tertinggi sebesar 0.83, dibandingkan metode-metode lain yang telah dibahas sebelumnya. Ini menunjukkan bahwa *SVM* mampu melakukan klasifikasi *genre* musik dengan tingkat keakuratan yang sangat baik.

Hampir semua *genre* musik memperoleh nilai *f1-score* yang tinggi, dengan *genre* “*metal*” mencatat performa tertinggi, yaitu *f1-score* sebesar 0.92, disusul oleh “*classical*” dengan 0.90 dan “*blues*” dengan 0.86. *Genre* lainnya seperti “*pop*”, “*jazz*”, dan “*reggae*” juga memiliki performa yang sangat baik, dengan nilai *f1-score* di atas 0.80. Satu-satunya *genre* dengan performa sedikit lebih rendah adalah “*rock*”, yang memiliki *f1-score* sebesar 0.68, meskipun nilai ini masih terbilang cukup baik untuk standar klasifikasi *multi-kelas*.

Rata-rata kinerja model juga mencerminkan performa yang stabil dan merata di seluruh kelas. Nilai *macro average* untuk *precision*, *recall*, dan *f1-score* adalah 0.83, yang menunjukkan bahwa model memberikan perhatian yang seimbang terhadap semua *genre* tanpa terlalu *bias* terhadap kelas yang dominan. Nilai *weighted average* juga sama, yaitu 0.83 untuk ketiga metrik, menegaskan bahwa performa model tetap baik meskipun mempertimbangkan jumlah *data* yang berbeda di setiap kelas. Secara keseluruhan, metode *SVM* menunjukkan hasil terbaik dalam klasifikasi *genre* musik dibandingkan dengan metode *Naive Bayes*, *K-NN*, dan *Random Forest*.

Berikut untuk *Confusion Matrix* dari setiap metode yang digunakan :

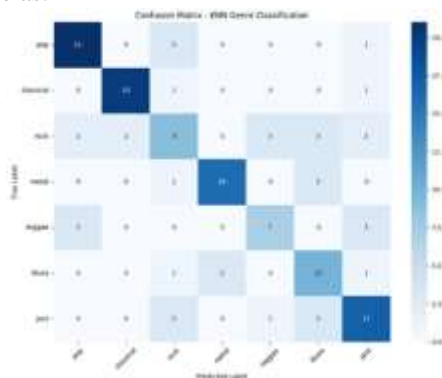


Gambar 2. Confusion Matrix Metode Naive Bayes

Gambar 2. menampilkan *confusion matrix* dari hasil klasifikasi *genre* musik menggunakan metode *Naive Bayes*. Tabel ini menggambarkan distribusi prediksi model terhadap data sebenarnya (*actual*) untuk masing-masing kelas *genre* musik, yaitu *pop*, *classical*, *rock*, *metal*, *reggae*, *blues*, dan *jazz*. Warna pada *heatmap* merepresentasikan intensitas jumlah prediksi, di mana warna yang lebih gelap menunjukkan jumlah prediksi yang lebih tinggi, memberikan visualisasi yang jelas terkait kekuatan dan kelemahan model dalam mengenali *genre* tertentu.

Dari *confusion matrix* ini, dapat dilihat bahwa *genre pop* diklasifikasikan dengan cukup baik, dengan 22 dari total 25 *data* diklasifikasikan secara benar. *Genre metal* juga menunjukkan performa cukup kuat dengan 16 prediksi yang tepat. Di sisi lain, *genre* seperti *rock* dan *reggae* memperlihatkan banyak kesalahan klasifikasi, terutama *rock* yang hanya memiliki 3 prediksi benar dari total 21 *data*, dan sisanya tersebar ke *genre* lain seperti *metal* dan *jazz*. Ini menunjukkan bahwa model kesulitan dalam membedakan *genre rock* dengan *genre* lain, khususnya *metal*.

Secara keseluruhan, *confusion matrix* ini mengindikasikan bahwa meskipun metode *Naive Bayes* mampu mengenali beberapa *genre* dengan baik, masih banyak terjadi kebingungan antar *genre* yang mirip, yang berdampak pada turunnya akurasi dan *f1-score* untuk beberapa kelas.

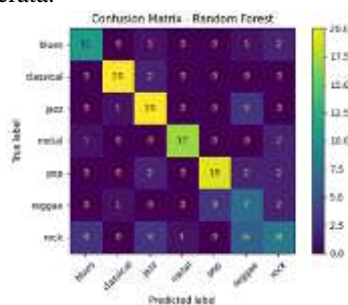


Gambar 3. Confusion Matrix Metode K-Nearest Neighbors

Gambar 3. menunjukkan *confusion matrix* hasil klasifikasi *genre* musik menggunakan metode *K-Nearest Neighbors (K-NN)*. Matriks ini memperlihatkan distribusi prediksi model terhadap *data* aktual dari masing-masing *genre* musik, termasuk *pop*, *classical*, *rock*, *metal*, *reggae*, *blues*, dan *jazz*. Warna biru yang lebih gelap menunjukkan jumlah prediksi yang lebih tinggi dan benar, sedangkan warna yang lebih terang menunjukkan nilai prediksi yang rendah atau kesalahan klasifikasi. Dari visualisasi ini, terlihat bahwa sebagian besar prediksi berada pada diagonal matriks, yang mengindikasikan bahwa model *K-NN* cukup akurat dalam mengklasifikasikan beberapa *genre*.

Beberapa *genre* diklasifikasikan dengan baik, seperti *genre pop*, dengan 21 dari 25 *data* berhasil diklasifikasikan dengan benar, dan *genre classical* dengan 20 dari 22 *data* juga diklasifikasikan secara tepat. *Genre jazz* menunjukkan performa yang sangat baik dengan 17 prediksi benar dari 20 *data*, hanya sedikit salah diklasifikasikan sebagai *blues*. *Genre metal* juga menunjukkan hasil memuaskan dengan 16 prediksi benar, walaupun masih ada beberapa prediksi yang meleset ke *genre pop* dan *rock*. Ini menandakan bahwa *K-NN* cukup efektif untuk *genre-genre* yang memiliki pola ciri khas yang konsisten.

Genre rock hanya berhasil diklasifikasikan dengan benar sebanyak 9 dari 21 *data*, dengan cukup banyak kesalahan prediksi ke *genre metal* dan *jazz*. *Genre reggae* juga hanya memiliki 7 prediksi benar dari 13 *data*, dengan sisa *data* tersebar ke *genre blues* dan *jazz*. Meskipun demikian, dibandingkan dengan metode *Naive Bayes*, *confusion matrix* ini menunjukkan bahwa metode *K-NN* mampu mengurangi jumlah kesalahan klasifikasi untuk sebagian besar *genre* dan memberikan hasil klasifikasi yang lebih merata.

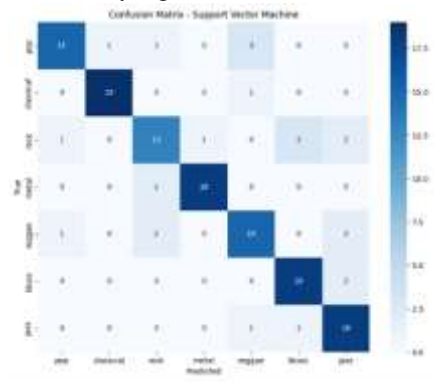


Gambar 4. Confusion Matrix Metode Random Forest

Gambar 4. menyajikan *confusion matrix* dari hasil klasifikasi *genre* musik menggunakan metode *Random Forest*. Matriks ini menggambarkan performa model dalam mengklasifikasikan tujuh *genre* musik berdasarkan jumlah prediksi yang benar (nilai diagonal) dan salah (nilai di luar diagonal). Warna pada matriks mencerminkan jumlah *data*, dengan warna kuning terang mewakili jumlah tertinggi dan ungu tua untuk jumlah yang lebih rendah. Secara umum, tampak bahwa *Random Forest* mampu memprediksi sebagian besar *genre* dengan tingkat akurasi yang baik.

Genre classical, jazz, dan pop menunjukkan performa klasifikasi yang sangat tinggi, masing-masing dengan 20 prediksi benar. *Genre metal* juga berhasil diklasifikasikan dengan baik, dengan 17 *data* yang diprediksi benar dari total 20. Selain itu, *genre blues* menunjukkan hasil cukup baik dengan 11 prediksi benar, meskipun masih terdapat beberapa kesalahan ke *genre reggae dan jazz*. Ini menunjukkan bahwa metode *Random Forest* cukup mampu mengenali pola karakteristik dari *genre-gener* yang memiliki ciri akustik yang khas. Namun, *genre rock dan reggae* masih menghadirkan tantangan bagi model.

"*Rock*" cenderung keliru diprediksi sebagai "*blues*", "*jazz*", dan "*reggae*". *Genre rock* hanya berhasil diklasifikasikan benar sebanyak 8 kali, sementara *reggae* memiliki 7 prediksi yang tepat dari total 13 *data*, dan sisanya tersebar ke *genre pop, blues, dan jazz*. Kesalahan prediksi ini mengindikasikan adanya kemiripan fitur antara beberapa *genre*, yang menyebabkan model sulit membedakan satu sama lain. Meski begitu, dibandingkan dengan metode *Naive Bayes* dan *K-NN*, *Random Forest* memberikan hasil klasifikasi yang lebih akurat dan stabil untuk sebagian besar *genre*.



Gambar 5. *Confusion Matrix* Metode *Support Vector Machine*

Gambar 5. menunjukkan *confusion matrix* dari hasil klasifikasi *genre* musik menggunakan metode *Support Vector Machine (SVM)*. Matriks ini memberikan gambaran mengenai seberapa baik model *SVM* dalam mengenali dan membedakan tujuh *genre* musik, yakni *pop, classical, rock, metal, reggae, blues, dan jazz*. Warna pada matriks mewakili jumlah prediksi, dengan warna biru gelap menunjukkan jumlah yang lebih tinggi, mengindikasikan prediksi yang lebih akurat. Sebagian besar nilai prediksi berada pada diagonal matriks, yang menandakan bahwa model *SVM* memiliki performa yang sangat baik dalam mengenali *genre* musik dengan benar.

Secara rinci, *genre blues dan metal* menunjukkan performa klasifikasi yang sangat baik, masing-masing dengan 18 *data* yang berhasil diklasifikasikan secara benar dari total 20. *Genre reggae* juga menunjukkan hasil sempurna dengan 15 prediksi benar dari 20 *data*, tanpa adanya prediksi yang salah ke *genre* lain. Sementara itu, *genre classical* memiliki 19 prediksi benar dan hanya satu kesalahan klasifikasi. *Genre jazz* juga diklasifikasikan dengan sangat baik, yakni 18 *data* benar dari 20, dengan sedikit kesalahan ke *genre blues dan reggae*. Hasil ini mencerminkan kemampuan model *SVM* dalam mempelajari karakteristik *genre* secara mendalam.

Meskipun performanya secara umum sangat baik, masih terdapat beberapa kesalahan klasifikasi. Pola kesalahan pada metode *Support Vector Machine (SVM)* yaitu :

1. Terdapat sedikit kebingungan antara "*pop*" dengan "*classical*" dan "*rock*".
2. Model cukup baik dalam memprediksi "*classical*", "*metal*", "*reggae*", "*blues*", dan "*jazz*" karena sebagian besar prediksi yang benar terkonsentrasi di diagonal utama untuk *genre-gener* ini.
3. "*Rock*" terlihat memiliki beberapa kesalahan klasifikasi dengan "*pop*", "*metal*", "*reggae*", dan "*jazz*".

Misalnya, *genre pop* memiliki 15 prediksi benar, tetapi 3 *data* salah diklasifikasikan sebagai *jazz* dan 2 sebagai *rock*. Begitu juga dengan *genre rock*, yang diklasifikasikan dengan benar sebanyak 13 kali dari total 20 *data*, namun masih terdapat kesalahan klasifikasi ke *genre jazz dan reggae*. Walaupun demikian, dibandingkan dengan metode lainnya, *confusion matrix* ini mengindikasikan bahwa *SVM* merupakan metode yang paling andal dan konsisten dalam mengenali *genre* musik secara akurat pada eksperiment ini.

Berdasarkan perbandingan kinerja klasifikasi yang dilakukan, berbagai algoritma menunjukkan tingkat akurasi yang bervariasi. *Naive Bayes* tercatat memiliki akurasi terendah, yaitu 0.65 (65%). Kinerja ini mengindikasikan bahwa



algoritma tersebut kurang optimal dalam tugas klasifikasi yang diuji. Di atas *Naive Bayes*, *K-Nearest Neighbors (K-NN)* menunjukkan peningkatan performa dengan akurasi 0.71 (71%), menjadikannya pilihan yang lebih baik dibandingkan *Naive Bayes*.

Melanjutkan perbandingan, *Random Forest* berhasil melampaui kedua algoritma sebelumnya dengan akurasi 0.73 (73%). Ini menunjukkan bahwa *Random Forest* menawarkan kinerja yang lebih superior dibandingkan *Naive Bayes* dan *K-NN*. Namun, *Support Vector Machine (SVM)* muncul sebagai algoritma dengan kinerja terbaik di antara semua yang diuji, mencapai akurasi tertinggi sebesar 0.83 (83%). Hasil ini menempatkan *SVM* sebagai metode paling efektif dalam konteks klasifikasi yang dibahas.

Keunggulan *Support Vector Machine (SVM)* dapat dijelaskan oleh kemampuannya yang luar biasa dalam menangani ruang dimensi tinggi dan menemukan batas keputusan yang optimal. Hal ini sangat relevan dalam klasifikasi *genre musik*, di mana fitur-fitur *audio* cenderung kompleks dan beragam. Dengan demikian, *SVM* mampu mengidentifikasi pemisah yang jelas antar kelas, bahkan dalam data yang rumit.

Sementara itu, *Random Forest* juga menunjukkan kinerja yang baik berkat mekanisme bawaannya yang efektif dalam mengurangi *overfitting* dan kemampuannya menangani interaksi fitur non-linear. Meskipun demikian, performanya mungkin belum sebaik *SVM* dalam menemukan batas keputusan yang sangat kompleks. Di sisi lain, kinerja *K-Nearest Neighbors (K-NN)* sangat bergantung pada pemilihan nilai *K* dan metrik jarak yang tepat. Tantangan utama dalam klasifikasi *genre musik* dengan *K-NN* terletak pada penentuan "tetangga terdekat" yang paling relevan dalam ruang fitur musik yang kompleks.

Adapun *Naive Bayes* menunjukkan kinerja yang buruk karena asumsi independensi antar fitur yang menjadi dasarnya. Asumsi ini seringkali tidak terpenuhi dalam *data audio*, di mana fitur-fitur seperti spektral, temporal, dan frekuensi cenderung saling berkorelasi kuat. Korelasi ini melanggar asumsi inti *Naive Bayes*, sehingga membatasi kemampuannya untuk melakukan klasifikasi secara akurat pada *data audio*.

KESIMPULAN

Penelitian ini telah berhasil melakukan klasifikasi *genre musik* menggunakan beberapa algoritma *Machine Learning*, yaitu *Naive Bayes*, *K-Nearest Neighbors (K-NN)*, *Random Forest*, dan *Support Vector Machine (SVM)*. Dari perbandingan kinerja yang dilakukan, ditemukan bahwa *Support Vector Machine (SVM)* adalah algoritma yang paling unggul dengan akurasi tertinggi sebesar 0.83 (83%). Nilai *presisi*, *recall*, dan *F1-score* rata-rata untuk *SVM* juga mencapai 0.83, menunjukkan keseimbangan yang baik antara kemampuan identifikasi positif yang benar dan cakupan identifikasi kasus positif secara keseluruhan.

Beberapa *genre musik* seperti "*Classical*," "*Metal*," "*Blues*," dan "*Jazz*" relatif mudah diklasifikasikan dengan tingkat akurasi yang tinggi. Namun, penelitian ini juga mengidentifikasi bahwa *genre "Pop"* dan "*Rock*" adalah *genre* yang paling sulit untuk diklasifikasikan, seringkali saling bertukar dalam prediksi. Kesulitan ini kemungkinan besar disebabkan oleh kemiripan fitur *audio* yang inheren antara kedua *genre* tersebut, termasuk penggunaan instrumen, struktur lagu, dan *tempo*, serta batasan yang kabur antara *subgenre*.

REFERENCES

- [1] D. Sukietra, "Aplikasi Rekomendasi Musik Berdasarkan Klasifikasi Genre Menggunakan Convolutional Neural Networks," *KALBISIANA J. Sains, Bisnis dan Teknol.*, vol. 10, no. 1, pp. 91–98, 2024, doi: 10.53008/kalbisia.v10i1.3527.
- [2] R. Indriati and S. B. Persada, "Information retrieval pencarian informasi jenis Musik," *Pros. Semin. Nas. Inov. Teknol.*, pp. 305–310, 2018, [Online]. Available: <https://proceeding.unpkediri.ac.id/index.php/inotek/article/view/500>
- [3] D. Nuriska, B. Irawan, A. Bahtiar, and A. Rinaldi Dikananda, "Klasterisasi Data Lagu Terpopuler Spotify 2023 Berdasarkan Suasana Hati Menggunakan Algoritma K-Means," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 7, no. 6, pp. 3843–3850, 2024, doi: 10.36040/jati.v7i6.8232.
- [4] C. R. Wairata, E. R. Swedia, and M. Cahyanti, "Pengklasifikasian Genre Musik Indonesia Menggunakan Convolutional Neural Network," *Sebatik*, vol. 25, no. 1, pp. 255–261, 2021, doi: 10.46984/sebatik.v25i1.1286.
- [5] S. Navisa, Luqman Hakim, and Aulia Nabilah, "Komparasi Algoritma Klasifikasi Genre Musik pada Spotify Menggunakan CRISP-DM," *J. Sist. Cerdas*, vol. 4, no. 2, pp. 114–125, 2021, doi: 10.37396/jsc.v4i2.162.
- [6] I. G. Harsemadi, "Perbandingan Kinerja Algoritma K-NN dan SVM dalam Sistem Klasifikasi Genre Musik Gamelan Bali," *INFORMATICS Educ. Prof. J. Informatics*, vol. 8, no. 1, p. 1, 2023, doi: 10.51211/itbi.v8i1.2417.
- [7] Y. V. Via, I. Y. Purbasari, and A. P. Pratama, "Analisa Algoritma Convolution Neural Network (Cnn) Pada Klasifikasi Genre Musik Berdasar Durasi Waktu," *Scan J. Teknol. Inf. dan Komun.*, vol. 17, no. 1, pp. 35–41, 2022, doi: 10.33005/scan.v17i1.3251.
- [8] S. LUTFIANI, T. H. Saragih, F. Abadi, M. R. Faisal, and D. Kartini, "Perbandingan Metode Extreme Gradient Boosting Dan Metode Decision Tree Untuk Klasifikasi Genre Musik," *J. Inform. Polinema*, vol. 9, no. 4, pp. 373–382, 2023, doi: 10.33795/jip.v9i4.1319.
- [9] H. F. Putro, R. T. Vulandari, and W. L. Y. Saptomo, "Penerapan Metode Naive Bayes Untuk Klasifikasi Pelanggan," *J.*





- Tekno. Inf. dan Komun.*, vol. 8, no. 2, 2020, doi: 10.30646/tikomsin.v8i2.500.
- [10] R. Kurniawan, P. B. Wintoro, Y. Mulyani, and M. Komarudin, "Implementasi Arsitektur Xception Pada Model Machine Learning Klasifikasi Sampah Anorganik," *J. Inform. dan Tek. Elektro Terap.*, vol. 11, no. 2, pp. 233–236, 2023, doi: 10.23960/jitet.v11i2.3034.
- [11] R. W. Putri, A. Ristyawan, and M. N. Muzaki, "Comparison Performance of K-NN and NBC Algorithm for Classification of Heart Disease," *JTECS J. Sist. Telekomun. Elektron. Sist. Kontrol Power Sist. dan Komput.*, vol. 2, no. 2, p. 143, 2022, doi: 10.32503/jtecs.v2i2.2708.
- [12] A. Tangkelayuk, "The Klasifikasi Kualitas Air Menggunakan Metode KNN, Naïve Bayes, dan Decision Tree," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 9, no. 2, pp. 1109–1119, 2022, doi: 10.35957/jatisi.v9i2.2048.
- [13] H. A. N. S. Habibi, A. Nugroho, and R. Firliana, "Perbandingan Algoritma Naïve Bayes Classifier Dan K-Nearest Neighbors Untuk Analisis Sentimen Covid-19 Di Twitter," *J. Ilm. Inform.*, vol. 11, no. 01, pp. 54–62, 2023, doi: 10.33884/jif.v11i01.7069.
- [14] Isman, Andani Ahmad, and Abdul Latief, "Perbandingan Metode KNN Dan LBPH Pada Klasifikasi Daun Herbal," *J. RESTI (Rekayasa Sist. dan Tekno. Informasi)*, vol. 5, no. 3, pp. 557–564, 2021, doi: 10.29207/resti.v5i3.3006.
- [15] G. P. Kawani, "Implementasi Naive Bayes," *J. Informatics, Inf. Syst. Softw. Eng. Appl.*, vol. 1, no. 2, pp. 73–81, 2019, doi: 10.20895/inista.v1i2.73.
- [16] I. Nurjanah, J. Karaman, I. Widaningrum, D. Mustikasari, and S. Sucipto, "Penggunaan Algoritma Naïve Bayes Untuk Menentukan Pemberian Kredit Pada Koperasi Desa," *Explorer (Hayward)*, vol. 3, no. 2, pp. 77–87, 2023.
- [17] M. M. Mutoffar and A. Fadillah, "Klasifikasi Kualitas Air Sumur Menggunakan Algoritma Random Forest," *Naratif J. Nas. Riset, Apl. dan Tek. Inform.*, vol. 4, no. 2, pp. 138–146, 2022, doi: 10.53580/naratif.v4i2.160.
- [18] W. Apriliah, I. Kurniawan, M. Baydhowi, and T. Haryati, "Prediksi Kemungkinan Diabetes pada Tahap Awal Menggunakan Algoritma Klasifikasi Random Forest," *Sistemasi*, vol. 10, no. 1, p. 163, 2021, doi: 10.32520/stmsi.v10i1.1129.

