

# Klasifikasi Musik Berdasarkan Genre dengan Metode *Extreme Gradient Boosting*

Muhammad Luqman Aristio<sup>a1</sup>, I Ketut Gede Suhartana<sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana  
Kampus Unud Bukit Jimbaran, Bali, Indonesia

<sup>1</sup>aristiooooo@email.com

<sup>2</sup>ikg.suhartana@unud.ac.id

## Abstrak

Perkembangan musik diseluruh dunia saat ini menyebabkan peningkatan jumlah musik yang ada dibandingkan dengan era sebelumnya. Hal tersebut menimbulkan dampak baru yakni sulitnya dalam pengidentifikasian genre musik secara manual. Dengan memanfaatkan teknologi yang sudah berkembang saat ini, klasifikasi genre musik dapat dilakukan secara otomatis dengan memanfaatkan fitur-fitur yang dimilikinya. Penelitian ini akan melakukan klasifikasi genre musik dengan menggunakan metode *eXtreme Gradient boosting (XGBoost)* dengan *hyperparameter tuning Grid Search*. Terdapat 10 label dan 28 fitur yang akan digunakan dalam penelitian ini yakni tempo, beats, chroma, RMSE, spectral centroid, bandwidth, rolloff, zero crossing rate dan mfcc 1 sampai 20. Dari semua fitur tersebut, pada tahap pengujian dilakukan tahapan reduksi dimensi fitur dengan teknik *Principal Component Analysis (PCA)*. PCA akan menggabungkan beberapa fitur yang berkorelasi tinggi sehingga dapat meningkatkan kecepatan pengklasifikasian. Dari hasil evaluasi menghasilkan nilai validasi akurasi sebesar 67%. Dari hasil evaluasi tersebut menunjukkan bahwa metode XGBoost dapat melakukan klasifikasi musik berdasarkan genre walaupun memiliki akurasi cukup rendah.

**Kata kunci:** Klasifikasi Musik, *eXtreme Gradient Boosting*, Genre Musik, Grid Search

## 1. Pendahuluan

Musik adalah karya cipta berupa bunyi atau suara yang memiliki nada, irama dan keselarasan. Saat ini diseluruh dunia jumlah musik yang berkembang mungkin sudah tidak terhitung jumlahnya, namun setiap musik pasti akan memiliki kemiripan. Kemiripan tersebut biasanya berasal dari ritme, frekuensi dan latar belakang musik tersebut [1]. Kemiripan inilah yang disebut dengan *genre* musik. *Genre* musik merupakan hal yang penting untuk diketahui dikarenakan melalui *genre* inilah para pendengar atau penikmat musik dapat mencari jenis musik yang sesuai keinginan mereka dalam suatu basis data musik yang besar. Oleh karena itu dibutuhkan teknologi atau sistem yang dapat memberikan label *genre* secara otomatis sehingga dapat memudahkan pengguna dari segi waktu dan usaha.

Pada penelitian [2] sebelumnya, klasifikasi *genre* musik dilakukan dengan menggunakan metode *K-Nearest Neighbor* dengan 10 label dan 11 fitur. Pada penelitian tersebut akurasi rata-rata cukup rendah yakni sebesar 44,8% dikarenakan perbedaan akurasi antar label yang cukup besar. Perbedaan tersebut diakibatkan karena terdapat fitur yang tidak dapat merepresentasikan *genre* musiknya sehingga mengakibatkan model tidak dapat memprediksi dengan baik. Oleh karena itu pemilihan fitur yang akan digunakan harus lebih diperhatikan lagi untuk meningkatkan akurasi model. Kemudian pada penelitian [3], digunakan model *Hidden Markov* yang berbasis peluang dan probabilitas yang menghasilkan akurasi 80%. Akurasi yang diperoleh cukup tinggi dikarenakan model *Hidden Markov* merupakan algoritma yang sangat kompleks dan lama namun algoritma ini merupakan teknik dasar untuk *automatic speech recognition* dan *part-of-speech tagging*.

Pada penelitian ini, penulis akan melakukan klasifikasi *genre* musik menggunakan metode *eXtreme Gradient boosting* dengan *hyperparameter tuning Grid Search* dengan fitur yang berbeda dan lebih

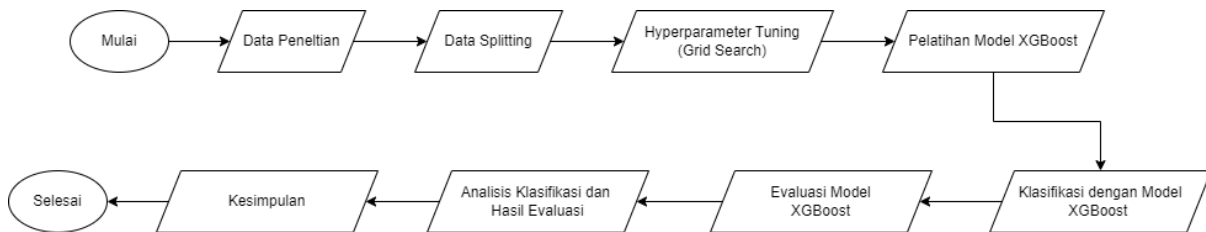
banyak dibandingkan penelitian sebelumnya. Metode *XGBoost* merupakan jenis algoritma yang dapat digunakan untuk klasifikasi maupun regresi yang cukup *powerful*. Pada dasarnya, algoritma ini sama dengan algoritma *gradient boost* hanya saja menggunakan beberapa proses tambahan.

## 2. Metodologi Penelitian

### 2.1 Data Penelitian

*Dataset* yang digunakan dalam penelitian ini merupakan data sekunder yang diperoleh dari GTZAN - *Music Genre Classification Data Set* berjumlah 1000 sampel dengan 10 label *genre* yakni *Blues, Classical, Country, Disco, Hiphop, Jazz, Metal, Pop, Reggae* dan *Rock*. Setiap label memiliki jumlah sampel yang sama yakni 100 sampel. File berformat *.csv* yang berisi hasil preprocessing berbagai fitur pada data musik GTZAN yang berdurasi 30 detik. Sehingga pada tahapan klasifikasi nanti tidak akan dilakukan preprocessing data lagi. Jumlah fitur awal *dataset* adalah 30 diantaranya yakni *filename, tempo, beats, chroma, RMSE, spectral centroid, bandwidth, rolloff, zero crossing rate, mfcc* yang berjumlah 20 dan label. Pada fitur *filename* tidak akan digunakan karena merupakan fitur yang berisi nama file yang tidak relevan. Sehingga tersisa 29 fitur saja, seluruh fitur selain label merupakan fitur numerik yang berupa angka negatif dan positif. *Dataset* akan dibagi menjadi 80% data *training* dan 20% data *testing*.

### 2.2 Desain Penelitian



**Gambar 1.** Flowchart Desain Klasifikasi

Gambar 1 merupakan flowchart desain penelitian. Pertama data penelitian akan dimasukkan kedalam model yang akan dikembangkan. Kemudian data tersebut akan dibagi menjadi 80% data *training* dan 20% data *testing*. Pada tahapan ini juga fitur dan label dipisahkan dari *dataset* dan kemudian masuk ke tahapan pengembangan model. Pertama akan dilakukan *hyperparameter tuning* dengan teknik *grid search*. Dengan teknik ini, model dapat menggunakan berbagai kombinasi *hyperparameter* dengan *range* yang sudah ditentukan dan kemudian memilih kombinasi terbaik dari *range* tersebut.

Setelah mengatur *hyperparameter*, selanjutnya masuk ke tahapan pelatihan dan klasifikasi model dengan *XGBoost Classifier*. Setelah pelatihan model selesai, akan ditampilkan parameter terbaik pada tahapan *grid search* kemudian hasil akurasi. Berikut yakni tahapan evaluasi model, pada evaluasi ini akan menggunakan matriks konfusi dengan melihat nilai presisi, *recall* dan *f1-score*.

### 2.3 Klasifikasi *eXtreme Gradient boosting*

Pada penelitian ini, klasifikasi *genre* musik akan dilakukan menggunakan metode *XGBoost* dengan *hyperparameter tuning Grid Search*. *XGBoost* merupakan algoritma *machine learning* yang dapat digunakan untuk kasus klasifikasi maupun regresi [4]. *XGBoost* adalah algoritma yang ditingkatkan berdasarkan *gradient boosting decision tree* dan dapat membangun *boosted trees* secara efisien dan beroperasi secara paralel [5]. Pada dasarnya, algoritma ini sama dengan algoritma *gradient boost* hanya saja menggunakan beberapa proses tambahan sehingga lebih *powerful*.

Dalam algoritma ini, pohon keputusan dibuat dalam bentuk sekuensial. Bobot memainkan peran penting dalam *XGBoost*. Bobot diberikan ke semua variabel independen yang kemudian dimasukkan ke dalam pohon keputusan yang memprediksi hasil. Bobot variabel yang diprediksi salah oleh pohon dinaikkan dan variabel-variabel ini kemudian diumpungkan ke pohon keputusan kedua. Skor prediksi masing-masing pohon keputusan kemudian dijumlahkan. Secara matematis, dapat dijabarkan pada persamaan (1).

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), f_k \in F \quad (1)$$

Kemudian terdapat fungsi objektif dari *XGBoost* yakni fungsi *loss* dan regularisasi pada iterasi  $t$  yang harus diminimalkan pada persamaan (2).

$$L^{(t)} = \sum_{i=1}^n l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) \quad (2)$$

Pada algoritma machine learning, biasanya terdapat beberapa nilai parameter yang diperkirakan dapat meningkatkan kinerja model yang disebut hyperparameter. Hyperparameter digunakan untuk meningkatkan hasil kinerja algoritma, yang mana cukup mempengaruhi berbagai uji model. Hyperparameter dilakukan secara fisik atau dengan menguji sekelompok hyperparameter pada batas yang telah ditentukan. Pencarian hyperparameter dilakukan secara manual atau dengan menguji kumpulan hyperparameter pada parameter yang ditentukan sebelumnya. Pada *XGBoost*, terdapat 7 parameter yang digunakan yang dapat dilihat pada Tabel 1.

**Table 1.** Parameter *XGBoost*

Parameter	Rentang Nilai
max_depth	Tingkat kedalaman pohon
n_estimators	Banyaknya pohon yang digunakan untuk proses klasifikasi
learning_rate	Membantu mempersingkat langkah dalam pembaruan model
random_state	menghasilkan angka acak yang diambil dari berbagai distribusi probabilitas

Selanjutnya yakni menentukan nilai pada parameter tersebut, terdapat proses *tuning hyperparameter* dengan teknik *Grid Search* dengan nilai seperti pada Tabel 2.

**Table 2.** *Hyperparameter Tuning*

Parameter	Rentang Nilai
max_depth	2 – 10, interval 1
n_estimators	60 – 220, interval 40
learning_rate	0,1
random_state	0

Keuntungan dari menggunakan *Grid Search* adalah nilai *hyperparameter* dapat dibuatkan *range* beserta interval kemudian setiap nilai akan dilakukan kombinasi dan hanya *hyperparameter* terbaik yang akan digunakan pada model akhir nanti.

## 2.4 *Principal Component Analysis (PCA)*

Pada pengujian penelitian ini, akan dilakukan teknik reduksi dimensi yakni *Principal Component Analysis (PCA)*. Teknik ini bertujuan untuk mereduksi dimensi, mengekstraksi fitur, dan mentransformasi data dari “n-dimensional space” ke dalam sistem berkoordinat baru dengan dimensi  $m$ , di mana  $m$  lebih kecil dari  $n$ . *PCA* bekerja menggunakan metode aljabar linier. Ia mengasumsikan bahwa sekumpulan data pada arah dengan varians terbesar merupakan yang paling penting (utama). *PCA* umumnya digunakan ketika variabel dalam data memiliki korelasi yang tinggi. Korelasi tinggi ini menunjukkan data yang berulang atau redundan.

## 2.5 Evaluasi

Evaluasi dilakukan dengan matriks konfusi dengan menghitung akurasi, *recall*, presisi dan *f1-score*. Matriks konfusi dapat digunakan untuk mengevaluasi kualitas classifier [6]. Pada confusion matrix dua

kelas, matriks menunjukkan *true positives*, *true negatives*, *false positives*, dan *false negatives*. Matriks konfusi untuk 2 kelas ditunjukkan pada Tabel 2.

**Table 3.** Matriks Konfusi

Kelas Sebenarnya	Prediksi Kelas	
	Positif	Negatif
Positif	TP	FN
Negatif	FP	TN

Keterangan :

TP = *True Positive* (total prediksi benar dari data positif)  
 FN = *False Negative* (total prediksi salah dari data positif)  
 TN = *True Negative* (total prediksi benar dari data negatif)  
 FP = *False Positive* (total prediksi salah dari data negatif)

Berikut persamaan untuk menghitung akurasi, presisi, *recall* dan *f1-score* [7] :

$$Akurasi = \frac{TP+FP}{TP+FP+TN+FN} \quad (3)$$

$$Presisi = \frac{TP}{TP+FP} \quad (4)$$

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

$$F1 - Score = \frac{2 \times recall \times presisi}{recall+presisi} \quad (6)$$

### 3. Hasil dan Diskusi

Pada penelitian ini dilakukan penerapan metode *Extreme Gradient Boosting* pada pengklasifikasian musik berdasarkan *genre*. Dataset diperoleh dari GTZAN - *Music Genre Classification Data Set* berjumlah 1000 sampel dengan 10 label *genre*. Dalam pengujian kali ini terdapat 2 jenis percobaan yang dilakukan yakni dengan metode XGBoost yang dilakukan reduksi dimensi dengan PCA dan XGBoost tanpa dilakukan reduksi dimensi. Selain itu juga, kedua pengujian tersebut akan dilakukan dengan menggunakan hyperparameter tuning dan tanpa hyperparameter tuning. Kemudian digunakan juga *K-Fold Cross Validation* untuk proses validasi dan pelatihan, dengan  $k = 5$

#### 3.1. Pengujian Model XGBoost tanpa PCA

Pada pengujian ini, dilakukan pelatihan model dengan metode XGBoost tanpa menggunakan PCA untuk mereduksi dimensi dari fitur yang ada pada dataset. Oleh karena itu seluruh fitur pada dataset digunakan secara langsung dalam pelatihan. Dataset dibagi menjadi 80% data training dan 20% data testing untuk memvalidasi hasil prediksi. Pengujian ini akan dicoba menggunakan *hyperparameter tuning* dan tanpa *hyperparameter tuning* untuk melihat hasil terbaik.

**Table 4.** Hasil Evaluasi Model dengan *Hyperparameter Tuning Grid Search*

Label	Presisi	Recall	F1-Score
<b>blues</b>	57%	72%	63%
<b>classical</b>	91%	91%	91%
<b>country</b>	56%	48%	51%
<b>disco</b>	47%	58%	52%
<b>hiphop</b>	57%	54%	55%

<b>jazz</b>	71%	56%	63%
<b>metal</b>	89%	83%	86%
<b>pop</b>	76%	81%	79%
<b>reggae</b>	68%	65%	67%
<b>rock</b>	43%	47%	45%
<b>Rata-rata</b>	<b>65,5%</b>	<b>65,5%</b>	<b>65,2%</b>

Bisa dilihat pada Tabel 4. Untuk hasil evaluasi dari model dengan hyperparameter tuning memperoleh akurasi sebesar 67% dengan nilai *presisi*, *recall* dan *F1-Score* 65%. Adapun parameter terbaik dari hasil tuning tersebut bisa dilihat pada Tabel 5.

**Tabel 5.** Nilai Parameter Terbaik

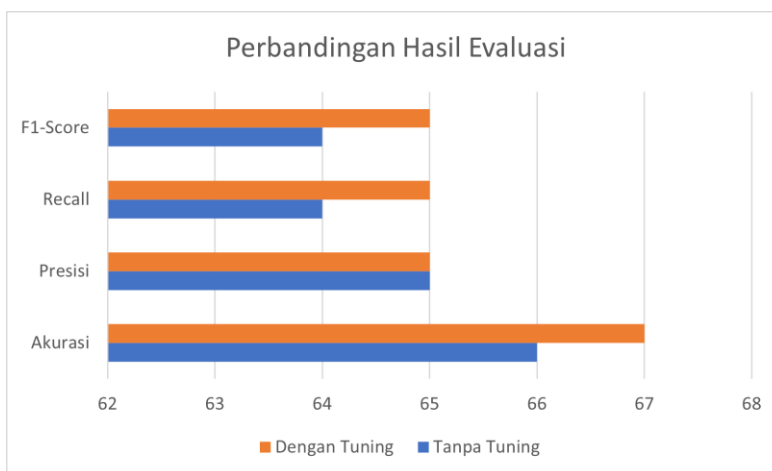
<b>Learning Rate</b>	<b>Max Depth</b>	<b>N_estimators</b>	<b>N_jobs</b>	<b>Random_state</b>
0.1	4	180	-1	0

Berikutnya akan dilakukan pengujian model *XGBoost* tanpa dilakukan *hyperparameter tuning* dengan *grid search*. Hasilnya bisa dilihat pada Tabel 6.

**Table 6.** Hasil Evaluasi Model tanpa *Hyperparameter Tuning Grid Search*

<b>Label</b>	<b>Presisi</b>	<b>Recall</b>	<b>F1-Score</b>
<b>blues</b>	67%	67%	67%
<b>classical</b>	96%	92%	94%
<b>country</b>	50%	53%	52%
<b>disco</b>	64%	62%	63%
<b>hiphop</b>	54%	68%	60%
<b>jazz</b>	56%	88%	68%
<b>metal</b>	93%	67%	78%
<b>pop</b>	71%	65%	68%
<b>reggae</b>	61%	50%	55%
<b>rock</b>	38%	36%	37%
<b>Rata-rata</b>	<b>65%</b>	<b>64,8%</b>	<b>64,2%</b>

Dari hasil tersebut diperoleh akurasi keseluruhan sebesar 66% dengan nilai *presisi* 65%, *recall* 64,8% dan *F1-Score* 64,2%. Akurasi yang dihasilkan jika dibandingkan dengan yang menggunakan hyperparemeter tuning tidak jauh berbeda. Hal ini menunjukkan bahwa perubahan parameter dalam model tersebut tidak memberikan dampak yang signifikan pada hasil. Berikut perbandingan hasil evaluasi kedua pengujian yang ditunjukkan pada Gambar 3.



**Gambar 2.** Perbandingan Hasil

### 3.2 Pengujian Model XGBoost dengan PCA

Pada pengujian ini dilakukan teknik reduksi dimensi fitur dengan memanfaatkan teknik PCA. Teknik PCA akan menggabungkan beberapa nilai fitur jika memiliki korelasi yang tinggi. Oleh karena itu untuk melihat korelasi digunakan heatmap fitur. Hasil heatmap ditunjukkan pada Gambar 3.



**Gambar 3.** Heatmap Korelasi Fitur

Pada gambar 2, bisa dilihat terdapat beberapa fitur yang memiliki korelasi sangat tinggi mencapai 0.9. Fitur-fitur tersebut diantaranya *tempo* dengan *beats* mencapai 0.97 kemudian *spectral centroid*, *spectral bandwidth*, *rolloff*, dan *mfcc2*. Dari hasil tersebut kita dapat melakukan reduksi dimensi dengan PCA untuk menggabungkan fitur-fitur tersebut menjadi 1 fitur yang relevan. Sehingga jumlah fitur yang digunakan pada pengujian ini berjumlah 25. Setelah reduksi dimensi tersebut, berikutnya dilakukan pelatihan model berdasarkan 25 fitur tersebut. Berikutnya akan dilakukan pengujian dengan menggunakan *hyperparameter tuning* terlebih dahulu. Hasilnya ditunjukkan pada Tabel 7.

**Table 7.** Hasil Evaluasi Model dengan *Hyperparameter Tuning Grid Search*

Label	Presisi	Recall	F1-Score
blues	60%	41%	49%
classical	100%	82%	90%
country	46%	61%	52%
disco	62%	42%	50%
hiphop	58%	70%	64%

<b>jazz</b>	50%	67%	57%
<b>metal</b>	83%	93%	88%
<b>pop</b>	75%	94%	83%
<b>reggae</b>	53%	48%	50%
<b>rock</b>	33%	30%	32%
<b>Rata-rata</b>	<b>62%</b>	<b>62,8%</b>	<b>61,5%</b>

Bisa dilihat pada Tabel 7. Untuk hasil evaluasi dari model dengan hyperparameter tuning memperoleh akurasi sebesar 62% dengan nilai *presisi* 62%, *recall* 62% dan *F1-Score* 61%. Adapun parameter terbaik dari hasil tuning tersebut bisa dilihat pada Tabel 8.

**Tabel 8.** Nilai Parameter Terbaik

<b>Learning Rate</b>	<b>Max Depth</b>	<b>N_estimators</b>	<b>N_jobs</b>	<b>Random_state</b>
0.1	2	180	-1	0

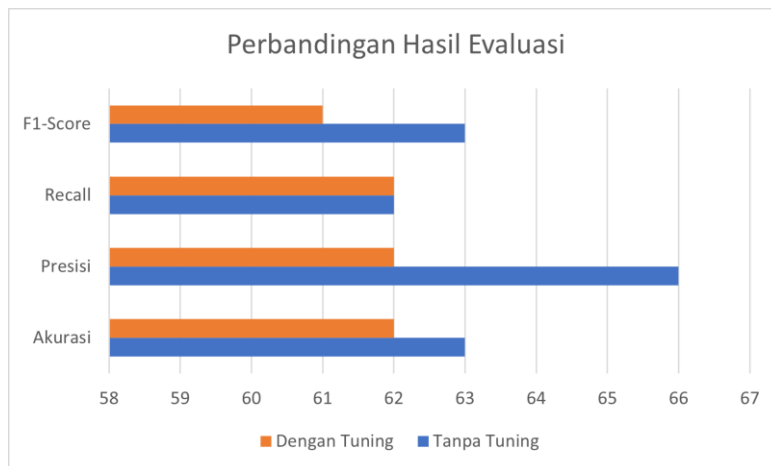
Berikutnya akan dilakukan pengujian model *XGBoost* dengan PCA tanpa *hyperparameter tuning* dengan *grid search*. Hasilnya bisa dilihat pada Tabel 9.

**Table 9.** Hasil Evaluasi Model tanpa *Hyperparameter Tuning Grid Search*

<b>Label</b>	<b>Presisi</b>	<b>Recall</b>	<b>F1-Score</b>
<b>blues</b>	75%	60%	67%
<b>classical</b>	100%	74%	85%
<b>country</b>	45%	74%	56%
<b>disco</b>	62%	62%	62%
<b>hiphop</b>	71%	52%	60%
<b>jazz</b>	69%	69%	69%
<b>metal</b>	87%	83%	85%
<b>pop</b>	77%	85%	81%
<b>reggae</b>	56%	43%	49%
<b>rock</b>	18%	27%	22%
<b>Rata-rata</b>	<b>66%</b>	<b>62,9%</b>	<b>63,6%</b>

Dari hasil tersebut diperoleh akurasi keseluruhan sebesar 63% dengan nilai *presisi* 66%, *recall* 62,9% dan *F1-Score* 63,6%. Akurasi yang dihasilkan jika dibandingkan dengan yang menggunakan

hyperparameter tuning tidak jauh berbeda juga. Berikut perbandingan hasil evaluasi kedua pengujian yang ditunjukkan pada Gambar 4.



**Gambar 4.** Perbandingan Hasil

#### 4. Kesimpulan

Dari hasil penelitian ini dapat disimpulkan bahwa klasifikasi *genre* musik menggunakan metode *XGBoost* dengan *hyperparameter tuning Grid Search* tanpa menerapkan *PCA* memperoleh akurasi tertinggi sebesar 67%. Selain itu juga diperoleh nilai parameter yang digunakan pada model terbaik berdasarkan *Grid Search* yang ditunjukkan pada Tabel 10.

**Table 10.** Nilai *Hyperparameter* Model Terbaik

Learning Rate	Max Depth	N_estimators	N_jobs	Random_state
0.1	4	180	-1	0

Hasil ini diperoleh dengan menggunakan seluruh fitur pada 10 label *genre* yang ada yakni *tempo*, *beats*, *chroma*, *RMSE*, *spectral centroid*, *bandwidth*, *rolloff*, *zero crossing rate* dan *mfcc* 1 sampai 20 tanpa melakukan reduksi dimensi. Dari model tersebut diperoleh beberapa nilai dari hasil evaluasi yakni nilai akurasi 67%, presisi 65,5%, *recall* 65,5% dan *F1-Score* 65,2%. Nilai evaluasi masih tergolong rendah, hal tersebut dikarenakan terdapat fitur-fitur yang tidak dapat merepresentasikan *genre* musik dengan baik seperti nilai *MFCC* yang berjumlah 20.

Pada penelitian selanjutnya diharapkan model yang dikembangkan menggunakan metode klasifikasi lainnya yang lebih sesuai seperti dengan memanfaatkan jaringan syaraf tiruan atau *neural network*. Selain itu juga sangat penting untuk melakukan reduksi dimensi fitur khususnya fitur *MFCC* dengan teknik lain seperti *Linear Discriminant Analysis* (*LDA*) yang lebih sesuai sehingga fitur yang digunakan dapat menghasilkan model yang lebih baik lagi.

#### Referensi

- [1] D. Lionel, R. Adipranata dan E. Setyati, "Klasifikasi Genre Musik Menggunakan Metode Deep Learning Convolutional Neural Network dan Mel-Spektrogram" *Jurnal Infra*, vol. 7, no. 1, 2019.
- [2] G. Ayu Vida Mastrika Giri, "Klasifikasi Musik Berdasarkan Genre dengan Metode K-Nearest Neighbor" *Jurnal Ilmu Komputer*, vol. XI, no. 2, p. 103-108, 2017.
- [3] R. Rahmawati, R. Magdalena dan I. N. A. Ramatryana, "Perbandingan dan Analisis K- Nearest Neighbor dan Linear Discriminant Analysis Untuk Klasifikasi Genre Musik" *eProceedings of Engineering*, vol. 3, no. 2, p. 1831-1837, 2016.



- [4] S. Ramaneswaran, K. Srinivasan dan P. M. Durai Raj Ramatryana, "Hybrid Inception v3 XGBoost Model for Acute Lymphoblastic Leukemia Classification" *Hindawi*, p. 1-10, 2021
- [5] T. Chen dan C. Guestrin, XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD &#x27;16. New York, NY, USA: ACM, pp. 785–794.
- [6] M. Audina *et al*, "Klasifikasi Berita Hoaks Covid-19 Menggunakan Kombinasi Metode K-Nearest Neighbor dan Information Gain" *Jurnal Elektronik Ilmu Komputer Udayana*, vol. 10, no. 4, p. 319-328, 2022.
- [7] S. Narkhede, "Toward Data Science", 9 Mei 2018. [Online]. Available: <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>. [2 Oktober 2022]

Halaman ini sengaja dibiarkan kosong