MACHINE LEARNING FINAL PROJECT
Spotify Genre Classification

Kelas:
Pembelajaran Mesin (H)

Dibuat Oleh:

Adhira Riyanti Amanda	5025211102
Altriska Izzati K. H.	5025211187
Gracetriana Survinta S.	5025211199

Dosen:
Dini Adni Navastara, S.Kom., M.Sc.

DEPARTMENT OF INFORMATICS ENGINEERING
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
FACULTY OF INTELLIGENT ELECTRICAL AND INFORMATICS TECHNOLOGY

ACADEMIC YEAR 2023/2024

Table of Contents

1. PENDAHULUAN

1.1. Latar belakang

Musik merupakan salah satu entitas seni yang telah melekat di kehidupan masyarakat. Menurut Haryandri (2020), Musik adalah ilmu dalam menyusun nada menjadi urutan, kombinasi, dan hubungan temporal sehingga membentuk komposisi suara yang memiliki kesinambungan dan kesatuan. Dalam penyusunan nada tersebut, musik tentunya memerlukan elemen-elemen penyusun. Pramono (2021) menyatakan bahwa ada delapan unsur penciptaan dari sebuah musik, yaitu melodi, birama, tempo, tangga nada, harmoni, timbre, dan dinamika. Semua elemen tersebut saling bersatu padu sehingga dapat menghasilkan suatu komposisi musik.

Di era digital ini, musik dapat diakses dan didengarkan secara mudah dimana saja dan kapan saja. Tersedia berbagai *digital platform* yang menawarkan layanan untuk mendengarkan musik. Salah satu *digital platform* terbesar di dunia yang memiliki jutaan layanan *podcast,* video, dan musik adalah Spotify yang dapat pengguna akses melalui perangkat digital dimanapun. Spotify memiliki puluhan juta koleksi lagu dengan karakteristik yang berbeda-beda dan berasal dari berbagai artis mancanegara. Spotify juga menawarkan fitur untuk mendengarkan lagu berdasarkan genrenya. Genre musik dapat diartikan sebagai pengelompokkan musik sesuai dengan kemiripan karakteristik suatu musik dengan musik lainnya (Sarofi, Irhamah, Mukarromah, 2020). Menurut Lau dan Ajooda (2021), umumnya musik yang dikategorikan dalam genre yang sama memiliki kesamaan dalam karakteristik instrumentasi, konten harmonik, dan struktur ritmis. Dalam dataset yang penulis gunakan, terdapat 15 karakteristik musik untuk menentukan ke genre mana sebuah lagu itu jatuh. Lima belas karakteristik tersebut diantaranya *popularity, duration, explicit, dance ability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo,* dan *time signature.* Penulis bertujuan untuk memahami relasi antar kelima belas karakteristik musik tersebut dan melakukan prediksi genre dari sebuah lagu dengan menerapkan metode Supervised Learning khususnya metode SVM (Support Vector Machine), ANN (Artificial Neural Network), dan XGBoost (Extreme Gradient Boosting).

1.2. Rumusan Masalah

Berikut adalah rumusan masalah dalam menyusun final project Spotify Genre Classification ini:

1. Bagaimana hasil matrik evaluasi pada setiap metode Supervised Learning?
2. Kinerja model mana yang lebih optimal di antara kinerja model yang menggunakan normalisasi atau tanpa normalisasi?
3. Kinerja model mana yang lebih optimal di antara kinerja model yang menggunakan *cross validation* atau tanpa *cross validation*?

1.3. Batasan Masalah

Berikut adalah batasan masalah yang perlu diperhatikan dalam final project Spotify Genre Classification ini:

1. Dataset project ini diambil dari internet yang telah dikumpulkan dari Spotify Web API dan dibersihkan oleh *creator* sehingga penulis tidak melakukan validasi data

2. Fitur yang digunakan untuk klasifikasi genre dibatasi pada kelima belas karakteristik yang telah disebutkan sebelumnya sehingga fitur-fitur kompleks yang memerlukan ekstraksi lanjutan tidak termasuk.
3. Metode yang diuji dibatasi pada tiga algoritma utama yaitu SVM, ANN, XGBoost sehingga tidak akan mengevaluasi metode-metode lain di luar ketiga metode tersebut.
4. Genre yang akan diprediksi akan dikelompokkan menjadi 3 genre utama yaitu EDM, rock, dan Blue Notes.

1.4. Tujuan dan Manfaat

Tujuan :

1. Mendapatkan hasil evaluasi metrik untuk ketiga metode Supervised Learning yang digunakan (SVM, ANN, dan XGBoost).
2. Mengetahui kinerja model mana yang lebih optimal antara kinerja model menggunakan normalisasi atau kinerja model tanpa normalisasi.
3. Mengetahui kinerja model mana yang lebih optimal antara kinerja model menggunakan *cross validation* atau kinerja model tanpa *cross validation*.

Manfaat :

1. Memahami relasi antar fitur untuk mengklasifikasi genre musik.
2. Melakukan prediksi genre dari sebuah lagu.
3. Memahami data melalui visualisasi dan analisis.
4. Dapat dikembangkan lebih lanjut untuk melakukan sistem rekomendasi genre bagi pengguna.

2. STUDI LITERATUR

2.1. Studi Literatur Terkait Genre Classification

Ditelusuri dari sejarah, musik telah hadir di peradaban manusia sejak era pra-sejarah. Telah terjadi evolusi musik selama bertahun-tahun hingga saat ini terdapat puluhan juta lagu yang berasal dari seluruh penjuru dunia. Dengan banyaknya lagu yang tersebar, manusia melakukan pengelompokkan terhadap lagu-lagu tersebut. Menurut laman *startlemusic.com*, saat ini terdapat 1.300 genre musik yang ada di dunia, termasuk sub-genre dari genre-genre populer. Dalam final project ini, digunakan dataset yang berasal dari *huggingface* (https://huggingface.co/datasets/maharshipandya/spotify-tracks-dataset) yang memiliki 21 kolom dengan 1 fitur target dan 144.000 baris data. Dataset ini memiliki klasifikasi genre sebanyak 114 genre. Pada proses *EDA* dan *pre-processing,* data akan dibersihkan dan diseleksi sehingga akan memiliki 15 fitur dan diklasifikasi ke dalam 3 *primary* genre menurut laman MusicMap (https://musicmap.info) yaitu rock, EDM, dan Blue Notes.

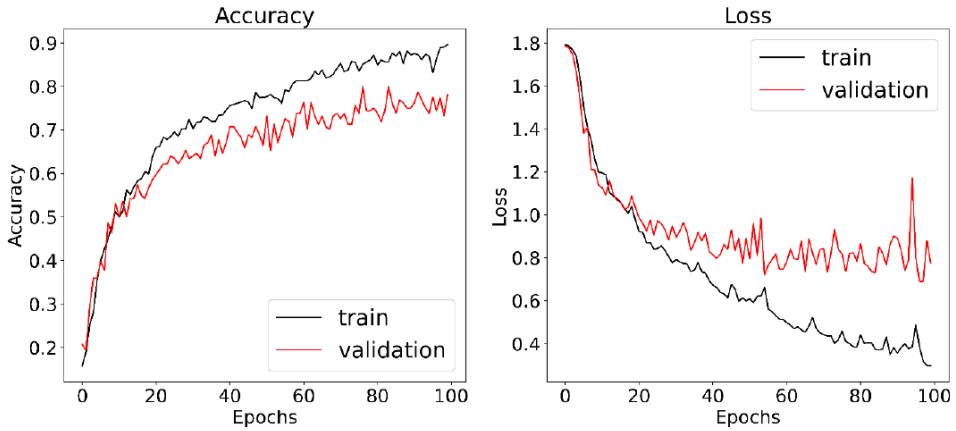
Melihat ke penelitian yang telah ada yaitu 1) Penelitian Setiadi, dkk. (2020). Penelitian tersebut menggunakan Spotify dataset yang diunduh dari laman www.crowdai.org yang memiliki 228.159 data dengan 18 fitur yang akan mengklasifikasikan ke dalam 26 genre musik. Namun, peneliti mengerucutkan 26 genre tersebut menjadi 5 kelompok genre yaitu *pop, electric, rap, opera,* dan *folk* dari 6.000 baris data, 2) Penelitian Hendri, D., dkk. (2024) juga memanfaatkan Spotify dataset dari https://kaggle.com dengan total data sebanyak 32.833 baris dan pengelompokkan ke dalam 6 genre utama yaitu *pop, EDM, rock, rap, R&B*, dan *latin*. Sama seperti kedua penelitian sebelumnya, 3) Tesis dari Muren (2019) mengambil Spotify dataset dengan 6.847 baris data dan 6 klasifikasi genre musik (*rock, pop, R&B, electro, hip hop,* dan *jazz).* Sedangkan, 4) Penelitian oleh Vodithala, S., dkk. (2023) melakukan analisis untuk memberikan perbandingan antara model *classifier* dan memberikan rekomendasi model mana yang paling optimal dengan *musical* dataset. Berbeda dengan keempat penelitian lain, 5) Penelitian dari Akter, Sultana, Noori, dan Hasan (2023) mengumpulkan data dan membuat datasetnya sendiri yang terdiri dari lagu Bangladesh yang menggunakan YouTube sebagai sumbernya. Dataset lagu Bangladesh tersebut terdiri dari 1.820 lagu dengan 6 genre yaitu *Adunik*, *rock, hip hop, Nazrul, Rabindra,* dan *folk.*

Kelima penelitian tersebut menggunakan metode yang berbeda-beda untuk melakukan klasifikasi genre. Penelitian Setiadi menggunakan metode SVM (*Support Vector Machine)* dengan RBF kernel dan melakukan eksperimen atas 4 macam kombinasi fitur. Di dalam evaluasi akurasi ini terdapat 2 hyperparameters yaitu C *value* dan Gamma *value* yang mana kombinasi C Value dan Gamma tersebut merupakan kombinasi yang menghasilkan nilai akurasi yang optimal saat sebelumnya dilakukan Grid Search. Kelebihan dari penelitian ini adalah mengetahui fitur mana yang harus diseleksi untuk mendapatkan hasil optimal sedangkan kekurangan dari penelitian ini adalah tidak ada perbandingan dengan metode lain karena metode yang digunakan hanya satu jenis. Berikut adalah nilai akurasi yang diperoleh dari eksperimen 4 kombinasi fitur pada Penelitian Setiadi.

Penelitian Hendri menggunakan model *classifier* Decision Tree, Naive Bayes, dan KNN. Nilai akurasi tertinggi diraih oleh model Decision Tree dengan akurasi sebesar 23% kemudian K-Nearest Neighbours sebesar 19% dan Naive Bayes 17%. Kelebihan dari penelitian ini yaitu penelitian ini menawarkan 3 metode yang dapat menjadi perbandingan. Namun, kekurangannya sendiri adalah nilai akurasi model yang cukup kecil. Sedangkan, penelitian Mauren menggunakan model *classifier* XGBoost dengan nilai akurasi model sebesar 73.43%. Nilai akurasi tersebut merupakan nilai yang

cukup besar tetapi penulis merasa dataset yang digunakan masih kurang dari berbagai aspek sehingga tidak bagus jika diaplikasikan di kehidupan nyata.

Penelitian keempat dari Vodithala, menggunakan 5 model *classifier* yaitu Random Forest, Extra Trees, LightGBM, XGBoost, dan CatBoost. Berikut merupakan nilai akurasi model secara berurutan dari model Random Forest, Extra Trees, LightGBM, XGBoost, dan CatBoost: 0.83, 0.83, 0.84, 0.84, dan 0.82. Kelebihan dari penelitian ini adalah menggunakan banyak metode sehingga bisa melakukan perbandingan mana metode yang paling optimal, sedangkan kekurangannya adalah kurang penjelasan detail tentang proses seleksi fitur. Penelitian terakhir menggunakan ANN sebagai metode klasifikasinya memiliki akurasi sebesar 89.43% pada *training set* dan 78.49% pada *validation set*. Penulis telah membandingkan model ANN dengan model machine learning lain dan hasilnya model ANN memiliki nilai akurasi paling besar dibandingkan dengan model lain. Berikut adalah grafik nilai akurasi dan *loss* dari model pada Penelitian Akter, Sultana, Noori, dan Hasan.



Gambar 1. Grafik *accuracy* dan *loss* untuk *training set* dan *validation set*

2.2. Studi Literatur Terkait Exploratory Data Analysis (EDA)

2.2.1.

2.2.2. Exploratory Data Analysis (EDA)

Menurut Sagala dan Aryatama (2022), Exploratory Data Analysis adalah pendekatan yang digunakan untuk menganalisis dan merangkum karakteristik utama dari data menggunakan grafik atau visualisasi data yang lain agar memudahkan pemahaman tentang dataset yang digunakan. Tujuan utama dari EDA adalah untuk memeriksa data untuk menemukan pola, mendeteksi anomali, menguji hipotesis, dan memeriksa asumsi menggunakan ringkasan statistik dan representasi grafik. Tahap ini juga dilakukan untuk memahami dataset yang sudah bersih dengan lebih dalam sebelum dilakukan feature engineering. Beberapa teknik yang sering dilakukan pada EDA antara lain *data collection, data cleaning, data visualization,* dan sebagainya.

2.2.2. Data Cleaning

Data Cleaning adalah salah satu langkah awal dan paling krusial dalam menganalisis data. Proses ini akan mengidentifikasi dan membersihkan data dari data yang rusak, data yang memiliki duplikat, dan data lain yang menyebabkan error. Menurut Lee (2021), *Data Cleaning* akan memastikan konsistensi dari training data untuk mempertahankan performa dari model karena inkonsistensi dan error pada training data dapat mengurangi kemampuan algoritma dalam mendeteksi pattern. Beberapa teknik yang digunakan saat data cleaning antara lain drop data yang

mengandung NA, drop data yang mengandung duplikat, drop kolom atau fitur yang tidak relevan, dan lainnya.

2.2.3. Data Visualization

Data Visualization atau visualisasi data merupakan proses untuk merepresentasikan data dalam bentuk visual yang dapat dipahami oleh pembacanya. Proses ini akan menggambarkan grafik untuk menunjukkan data, dapat berupa scatterplot, histogram, maupun bentuk visualisasi lain. Tampilan data umumnya bersifat deskriptif, berfokus pada data mentah dan ringkasan sederhana (Unwin, 2020).

2.2.4. Label Encoding

Menurut Herdian, Kamila, Budidarma (2024), *Label encoding* merupakan metode mengubah variabel kategorikal menjadi representasi numerikal agar dapat dipahami oleh algoritma model yang akan digunakan. Misalnya, dimiliki tiga kategori genre “Pop”, “Rock”, dan “Jazz”. Ketiga kategori tersebut akan diberi label numerik menjadi Pop : 1, Rock : 2, dan Jazz : 3.

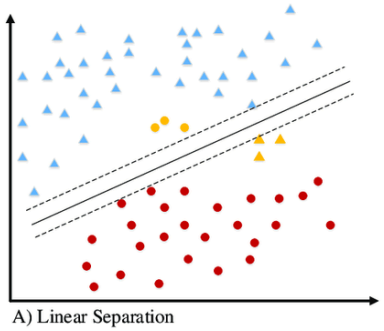
2.3. Studi Literatur Terkait Model *Classifier*

2.3.1. Support Vector Machine (SVM)

Classifier ini bekerja dengan memisahkan beberapa kelas ke dalam training set dengan surface yang memaksimalkan margin antara mereka. SVM sendiri dibagi menjadi dua tipe utama, yaitu Linear SVM dan Non-Linear SVM.

2.3.2. Linear SVM

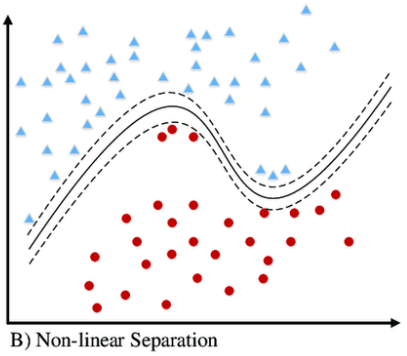
Tipe ini berguna dalam kasus di mana data perlu dipisahkan secara linier, yang berarti bahwa dataset dapat dibagi menjadi dua kelas yang dipisahkan oleh satu garis lurus.



Gambar 2. Linear SVM

2.3.3. Non-Linear SVM

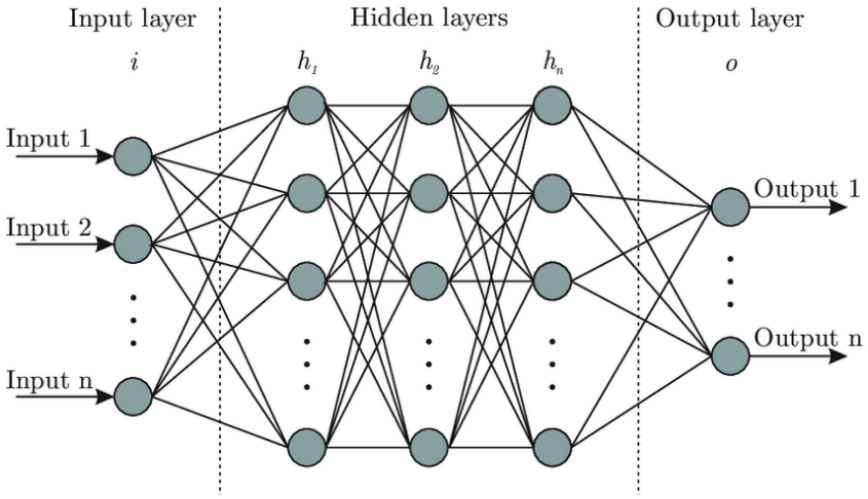
Tipe ini berguna dalam kasus di mana data perlu dipisahkan secara non-linear, yang berarti bahwa dataset tidak dapat dibagi menjadi kelas-kelas menggunakan garis lurus sehingga hyperplane akan berbentuk non-linear.



Gambar 3. Non-Linear SVM

2.3.4. Artificial Neural Network (ANN)

Sebuah classifier yang bekerja dengan cara mensimulasikan fungsi otak manusia. ANN sendiri terdiri dari sejumlah unit proses yang disebut neuron yang diatur dalam lapisan dan terhubung satu sama lain dengan weight tertentu. Output dihasilkan berdasarkan komputasi matematika di input dan hidden layernya. Algoritma dari neural network menyesuaikan weight dengan meminimalisir error di prediksi output dari ANN dan target output.



Gambar 4. Artificial Neural Network (ANN)

2.3.5. XGBoost (Extreme Gradient Boosting)

Classifier ini menggunakan framework *gradient boosting* dan merupakan sebuah classifier yang berbasis *Decision Tree* yang menggunakan gradient boosting. Dengan teknik boosting, model tree yang terbuat di atas model tree yang lain. Tree baru yang terbuat membantu mengoreksi error yang dihasilkan oleh tree sebelumnya

2.4. Studi Literatur Terkait Metrik Evaluasi

Untuk melihat performa dari model yang dipilih, maka dilakukan pengujian dengan menggunakan precision, recall, F1-score, dan support sebagai metrik evaluasi.

2.4.1. Precision

Proporsi dari *true positive* dari semua prediksi positif yang dibuat dari model. Ini dikalkulasikan sebagai rasio dari *true positive* dengan *true positive* ditambah *false positive.*

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

2.4.2. Recall

Proporsi dari *true positive* dari semua instansi positif yang ada di dataset. Ini dikalkulasikan sebagai rasio dari *true positive* dengan *true positive* ditambah *false negative.*

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Recall mengindikasikan ada berapa banyak data yang benar-benar positif untuk diidentifikasi dengan benar.

2.4.3. F1-score

Rata-rata dari *presisi* dan *recall*. Ini menghasilkan metrik yang mengkombinasikan *presisi* dan *recall*, menyeimbangkan nilai dari kedua metrik.

$$F1\ score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

F1-score mempertimbangkan kedua nilai *false positive* dan *false negative* sehingga cocok untuk digunakan di dataset yang imbalance.

3. METODOLOGI

3.1. Dataset

Dataset diambil dari laman *huggingface* yang dapat diakses pada tautan https://huggingface.co/datasets/maharshipandya/spotify-tracks-dataset. Dataset memiliki ukuran 20.1 MB. Setiap baris dari dataset ini menyimpan data lagu dan fitur audio terkait dari 114 genre musik yang berbeda. Dataset ini memiliki total baris sebanyak 144.000 dan total kolom sebanyak 21 kolom.

3.1.1. Spesifikasi Data

a) Deskripsi Data

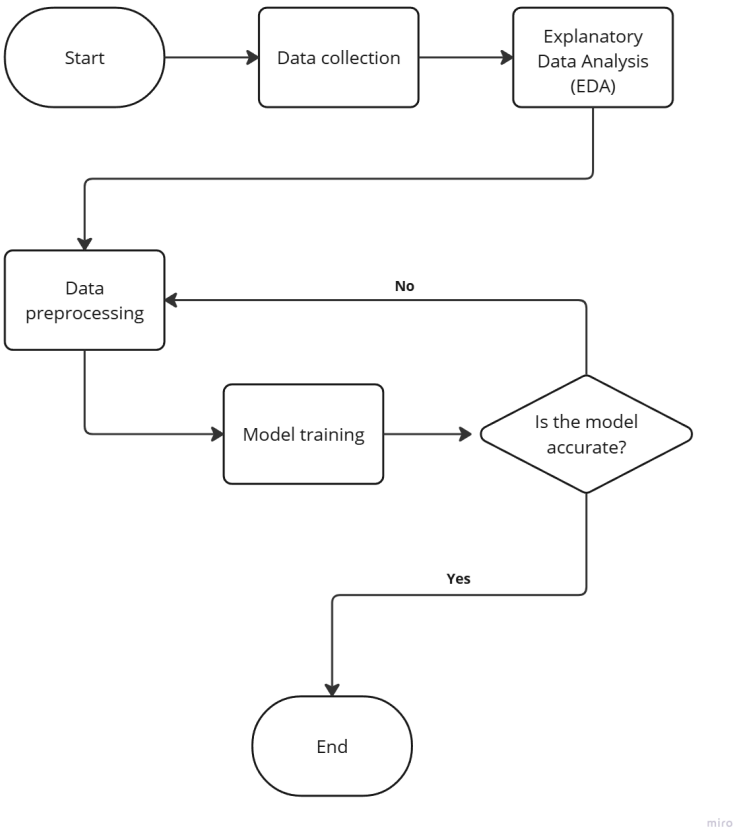
1. **Unnamed**: 0 : Unique identifier untuk sebuah row
2. **track_id**: ID Lagu
3. **artists**: Nama-nama artis yang membawakan lagu tersebut. Jika ada lebih dari satu artis, mereka dipisahkan dengan tanda titik koma (;)
4. **album_name**: Nama album di mana lagu tersebut muncul
5. track_name: Judul lagu
6. **popularity**: Nilai Popularitas dari sebuah lagu dengan rentang nilai 0 hingga 100. Popularitas ini dihitung oleh algoritma dan sebagian besar didasarkan pada jumlah total pemutaran lagu dan seberapa baru pemutaran tersebut. Secara umum, lagu-lagu yang sedang sering diputar sekarang akan memiliki popularitas lebih tinggi daripada lagu-lagu yang sering diputar di masa lalu. Lagu duplikat (misalnya, lagu yang sama dari single dan album) dinilai secara independen. Popularitas artis dan album diturunkan secara matematis dari popularitas lagu.
7. **duration_ms**: Durasi lagu dalam millisecond
8. **explicit**: Apakah lagu memiliki lirik eksplisit atau tidak (true = ya, memiliki lirik eksplisit; false = tidak memiliki lirik eksplisit atau tidak diketahui).
9. **danceability**: Fitur ini mendeskripsikan apakah sebuah lagu cocok untuk digunakan untuk dibuat menari atau tidak berdasarkan kombinasi dari elemen musik, termasuk dari tempo, stabilitas ritme, dan kecepatan ketukan. Nilai 0.0 menunjukkan bahwa lagu tidak dapat dibuat untuk menari, sedangkan semakin

tinggi nilainya maka semakin besar kemungkinan bahwa suatu lagu dapat dibuat menari.

10. **energy**: Mengukur nilai energi dari 0.0 hingga 1.0 dan merepresentasikan intensitas dari sebuah lagu. Lagu dengan energi yang tinggi terdengar cepat, lantang, dan berisik. Seperti death metal yang memiliki energi tinggi, sedangkan Lagu Bach bernilai rendah.
11. **key**: Sebuah kunci dari sebuah lagu. Diambil dari notasi standar dari Pitch Class dengan Integers map, contohnya 0 = C, 1 = C♯/D♭, 2 = D, dan seterusnya. Jika tidak ada kunci yang terdeteksi, maka valuenya akan menjadi -1.
12. **loudness**: Fitur ini merupakan kebisingan yang diukur berdasarkan desibelnya (dB).
13. **mode**: Mode menandakan modalitas dari sebuah lagu (mayor atau minor) diambil dari melodi yang disampaikan. Mayor direpresentasikan sebagai 1 dan minor sebagai 0.
14. **speechiness**: Fitur ini berisi nilai tentang apakah sebuah lagu penuh dengan percakapan atau tidak. Semakin sebuah lagu penuh dengan percakapan (seperti *talk show*, *audiobook*, puisi), semakin dekat nilainya dengan 1.0. Nilai di atas 0.66 mendeskripsikan bahwa sebuah lagu berisi sepenuhnya dari percakapan. Nilai antara 0.33 hingga 0.66 menandakan bahwa sebuah lagu mengandung percakapan dan musik, entah itu berupa beberapa lapisan dan bagian atau tidak, seperti musik rap. Nilai di bawah 0.33 memiliki kemungkinan besar bahwa lagu berisi musik sepenuhnya.
15. **acousticness**: Sebuah pengukuran dari 0.0 hingga 1.0 apakah sebuah lagu adalah akustik atau tidak. 1.0 menandakan bahwa kemungkinan bahwa suatu lagu akustik adalah tinggi.
16. **instrumentalness**: Memprediksi apakah sebuah lagu tidak mengandung vokal. Dalam konteks ini, “ooh” dan “aah” dianggap sebagai instrumental. Sedangkan rap dan lagu yang banyak mengandung kata-kata yang terucapkan dianggap sebagai “vokal”. Semakin dekat nilai *instrumentalness* kepada 1.0, semakin besar kemungkinan bahwa lagu merupakan sebuah lagu yang tidak mengandung vokal.
17. **liveness**: Mendeteksi suatu presensi dari suatu audien di dalam sebuah lagu. Semakin tinggi liveness, nilai di atas 0.8, menandakan kemungkinan bahwa lagu dibawakan secara langsung tinggi.
18. **valence**: Sebuah perhitungan dari 0.0 hingga 1.0 yang mendeskripsikan energi positif yang tersampaikan dari sebuah lagu. Lagu dengan valence tinggi terdengar lebih positif (seperti bahagia dan ceria), sedangkan lagu dengan valence rendah terdengar lebih negatif (seperti sedih, depresi, dan marah).
19. **tempo**: Perkiraan keseluruhan tempo dari sebuah lagu yang berupa *beats per minute* (BPM) atau ketukan per menit. Dalam konteks musik, tempo adalah kecepatan dari sebuah lagu.
20. **time_signature**: Sebuah perkiraan tanda notasi yang menentukan bagaimana irama atau ritme dalam sebuah komposisi musik diorganisir dan dihitung dengan cara berapa banyak ketukan dalam satu bar. Rentang tanda notasi tersebar dari ¾ hingga 7/4.
21. **track_genre**: Genre dari sebuah lagu

3.2. Desain Sistem

Sub bab ini menjelaskan alur kerja proses klasifikasi yang dilakukan untuk mendapat prediksi genre dari lagu. Berikut merupakan sebuah *flowchart* yang menggambarkan alur kerja proses:



Gambar 5. Diagram Alir

3.2.1. Data Collection

Pertama-tama dilakukan install dan import library yang dibutuhkan untuk proses kerja. Setelah library ter-setup, dilakukan ekstraksi dataset dari *huggingface* dan melihat isi dari dataframe secara singkat, sebatas melihat informasi, data type fitur, dan overview dari dataframe.

a) Informasi Data

```
RangeIndex: 114000 entries, 0 to 113999
Data columns (total 21 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   Unnamed: 0        114000 non-null  int64
 1   track_id          114000 non-null  object
 2   artists           113999 non-null  object
 3   album_name        113999 non-null  object
 4   track_name        113999 non-null  object
 5   popularity        114000 non-null  int64
 6   duration_ms       114000 non-null  int64
 7   explicit          114000 non-null  bool
 8   danceability      114000 non-null  float64
 9   energy            114000 non-null  float64
 10  key               114000 non-null  int64
 11  loudness          114000 non-null  float64
 12  mode              114000 non-null  int64
 13  speechiness       114000 non-null  float64
 14  acousticness      114000 non-null  float64
 15  instrumentalness  114000 non-null  float64
 16  liveness          114000 non-null  float64
 17  valence           114000 non-null  float64
 18  tempo             114000 non-null  float64
 19  time_signature    114000 non-null  int64
 20  track_genre       114000 non-null  object
dtypes: bool(1), float64(9), int64(6), object(5)
```

Gambar 6. Informasi Dataset

b) Jumlah nilai unik dari dataset

```
df.nunique()
```

```
Unnamed: 0          114000
track_id             89741
artists              31437
album_name           46589
track_name           73608
popularity             101
duration_ms          50697
explicit                 2
danceability          1174
energy                2083
key                     12
loudness             19480
mode                     2
speechiness           1489
acousticness          5061
instrumentalness      5346
liveness              1722
valence               1790
tempo                45653
time_signature           5
track_genre            114
dtype: int64
```

Gambar 7. Nilai Unik Kolom

c) Baris yang memiliki duplikat di kolom ‘track_id’

```
df.duplicated('track_id').sum()

24259
```

Gambar 8. Nilai Duplikat

d) Contoh data teratas yang didapatkan dengan df.head()

Unnamed: 0	track_id	artists	album_name	track_name	popularity	duration_ms	explicit	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	time_signature	track_genre
0	5SuOikwiRyPMVoIQDJUgSV	Gen Hoshino	Comedy	Comedy	73	230666	FALSE	0.676	0.461	1	-6.746	0	0.143	0.0322	0.000001	0.358	0.715	87.917	4	acoustic
1	4qPNDBW1i3p13qLCt0Ki3A	Ben Woodward	Ghost (Acoustic)	Ghost - Acoustic	55	149610	FALSE	0.42	0.166	1	-17.235	1	0.0763	0.924	0.000006	0.101	0.267	77.489	4	acoustic
2	1iJBSr7s7jYXzM8EGcbK5b	Ingrid Michaelson;ZAYN	To Begin Again	To Begin Again	57	210826	FALSE	0.438	0.359	0	-9.734	1	0.0557	0.21	0	0.117	0.12	76.332	4	acoustic
3	6lfxq3CG4xtTiEg7opyCyx	Kina Grannis	Crazy Rich Asians (Original Motion Picture Sou...	Can't Help Falling In Love	71	201933	FALSE	0.266	0.0596	0	-18.515	1	0.0363	0.905	0.000071	0.132	0.143	181.74	3	acoustic
4	5vjLSffimiIP26QG5WcN2K	Chord Overstreet	Hold On	Hold On	82	198853	FALSE	0.618	0.443	2	-9.681	1	0.0526	0.469	0	0.0829	0.167	119.949	4	acoustic

Tabel 1. Data teratas dari dataset

3.2.2. Exploratory Data Analysis (EDA)

Tahap ini dilakukan untuk memahami dataset yang sudah bersih dengan lebih dalam sebelum dilakukan feature engineering. EDA dilakukan dengan *data cleaning* dan membuat visualisasi data seperti:

1) Distribusi genre dengan xticks
2) Distribusi genre terhadap setiap fitur dengan strip plot
3) Mencari tahu outlier data pada genre terhadap setiap fitur dengan box plot
4) Melihat distribusi dari data numerik dengan histogram

3.2.2.1. Data Cleaning

Setelah mendapatkan informasi dari dataframe, dilakukan pembersihan data dalam upaya untuk meningkatkan akurasi model dengan:

1) Drop lagu yang mengandung NA

Memastikan bahwa tidak ada row yang memiliki nilai null.

2) Drop lagu dengan *track_genre* yang tidak relevan

Menghapus genre yang tidak relevan berdasarkan *MusicMap.*

3) Drop lagu yang mengandung duplikat

Terdapat lagu duplikat yang dapat dilihat melalui nilai track_id yg sama dan yang membedakan lagu duplikat tersebut adalah genre yang berbeda. Oleh karena itu, dilakukan drop duplicate row supaya setiap lagu diwakili secara unik dalam dataset dan menghindari bias yang mungkin timbul dari informasi yang bertentangan.

4) Drop kolom/fitur yang tidak relevan

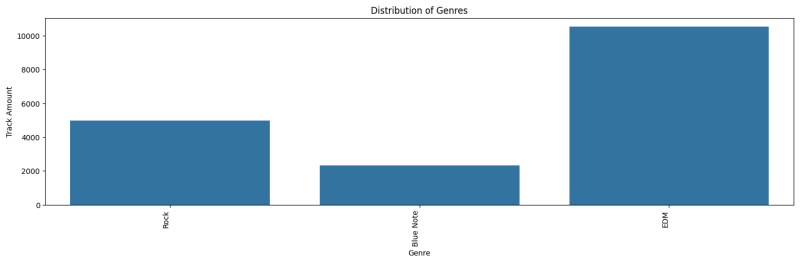
Menghapus kolom tidak relevan, seperti *Unnamed: 0*, *track_id*, *track_name*, *album_name*, dan *artists*.

5) Mengkategorikan *track_genre* yang memiliki kesamaan

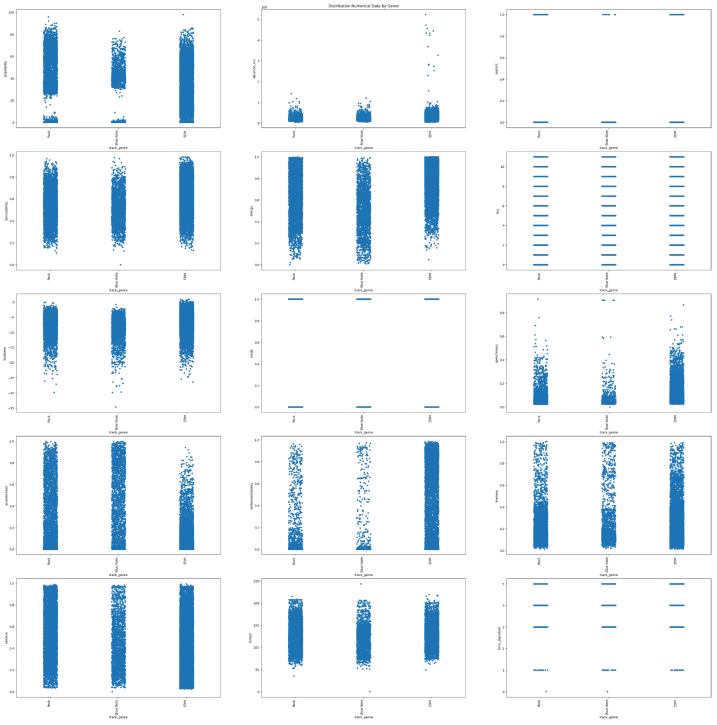
Mengelompokkan lagu berdasarkan 3 primary genre clusters berdasarkan *MusicMap* untuk mempermudah proses training.

3.2.2.2. Data Visualization

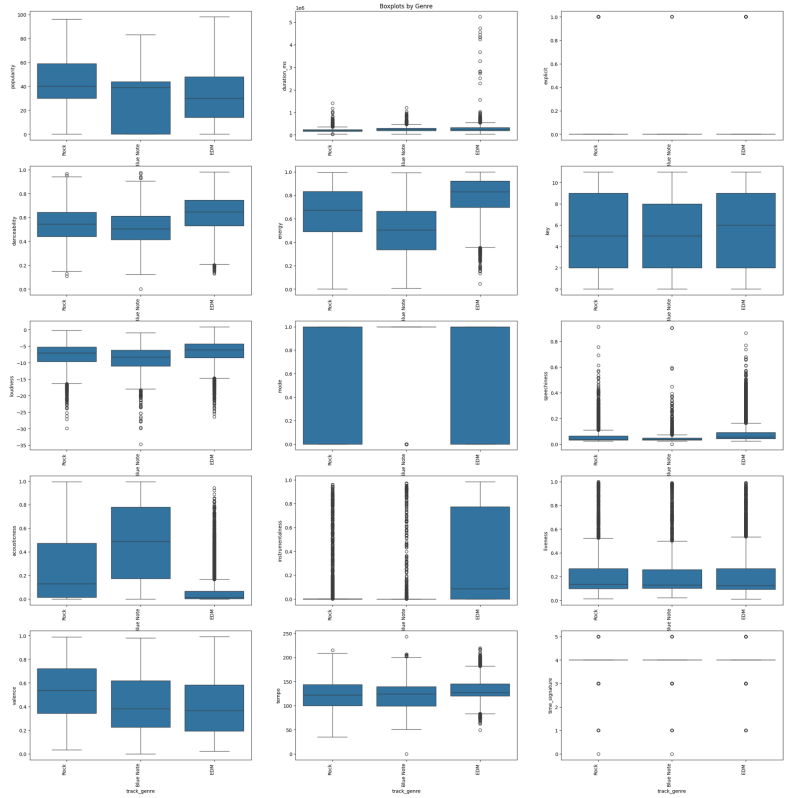
Visualisasi dari dataset adalah sebagai berikut.



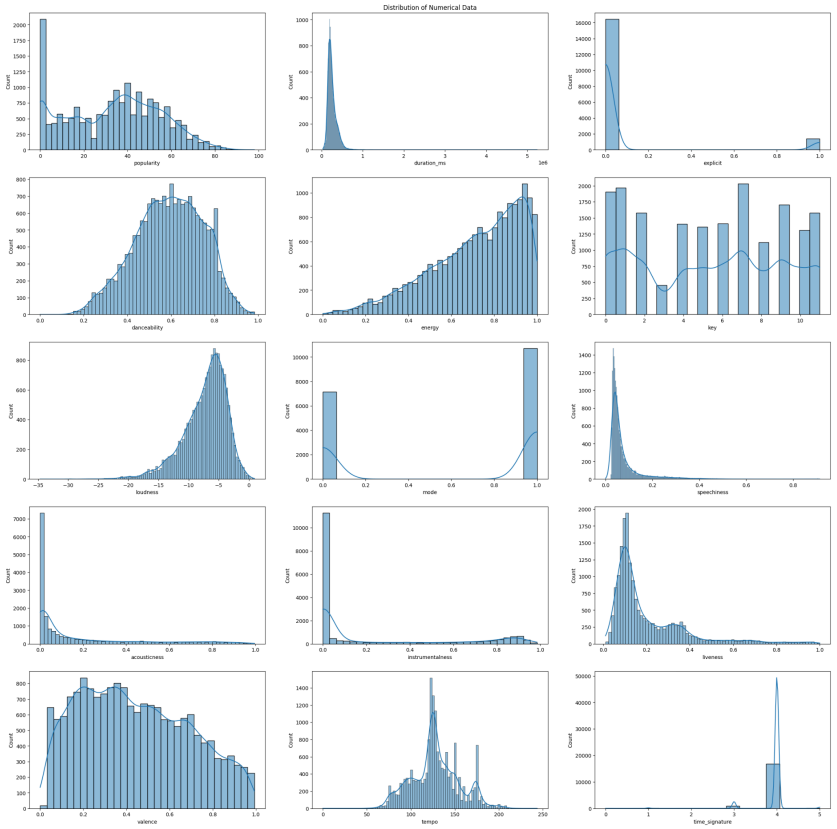
Gambar 9. Distribusi Genre



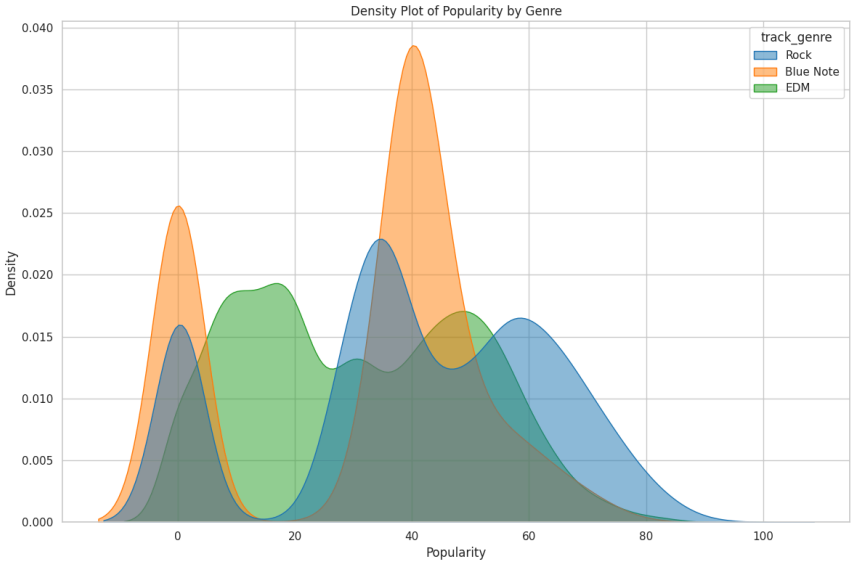
Gambar 10. Distribusi genre terhadap setiap fitur



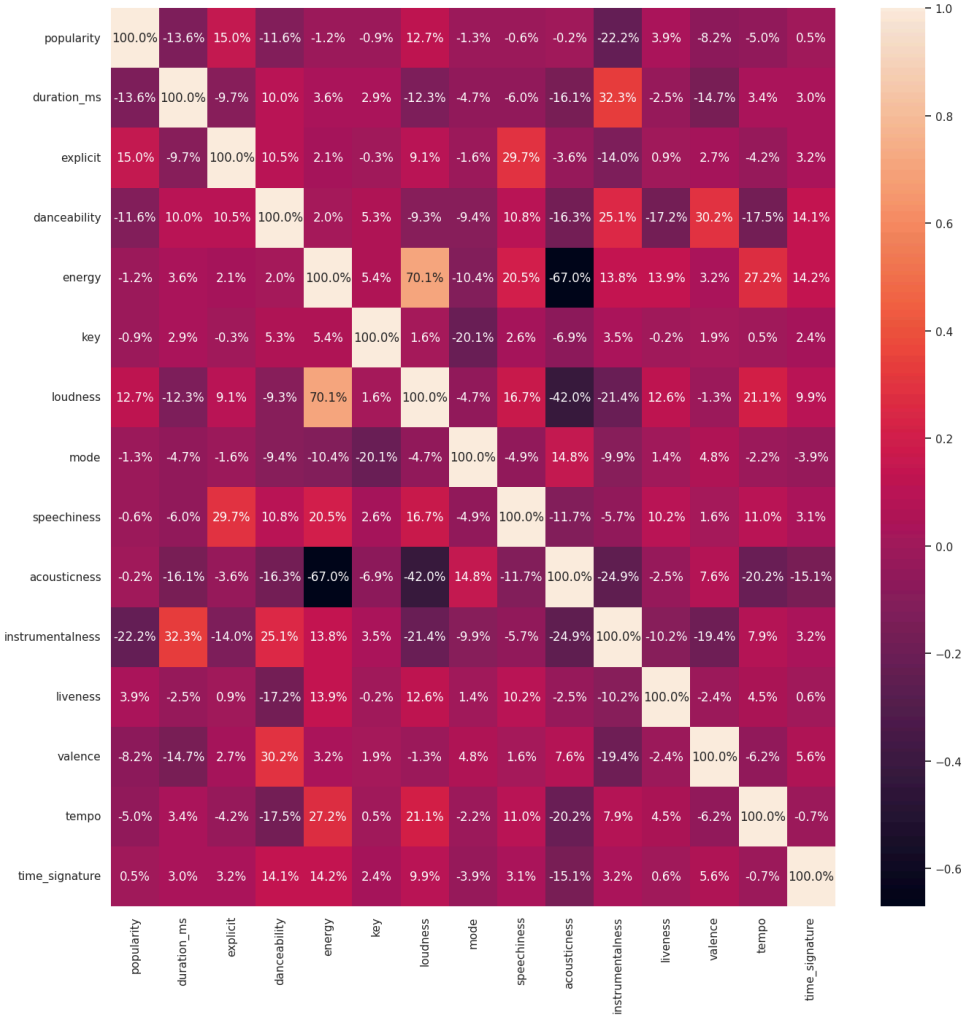
Gambar 11. Boxplot Genre



Gambar 12. Distribusi Data Numerik



Gambar 13. Plot Kepadatan Genre



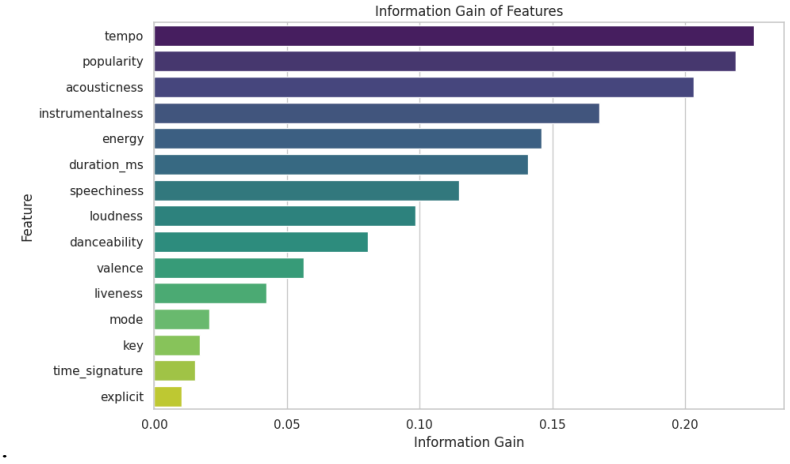
Gambar 14. Heatmap

3.2.3. Data Preprocessing

Tahap ini dilakukan dengan *feature selection, outlier handling, train test split, normalization*.

3.2.3.1. Feature Selection

Dilakukan dengan menggunakan *information gain*, yaitu dengan mengevaluasi gain dari setiap variabel dari target. Kalkulasi berdasarkan dari informasi mutual dari dua variabel yang random. Dari dilakukannya feature selection, dapat dilakukan reduksi fitur yang tidak relevan dengan menentukan threshold.

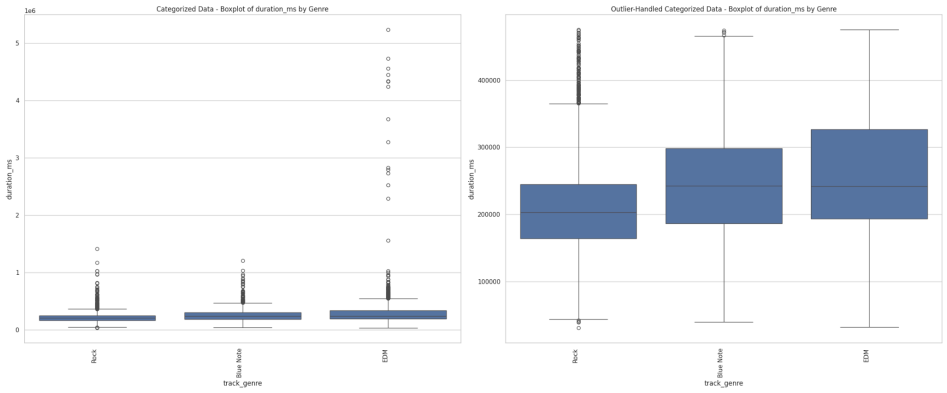


Gambar 15. Information Gain

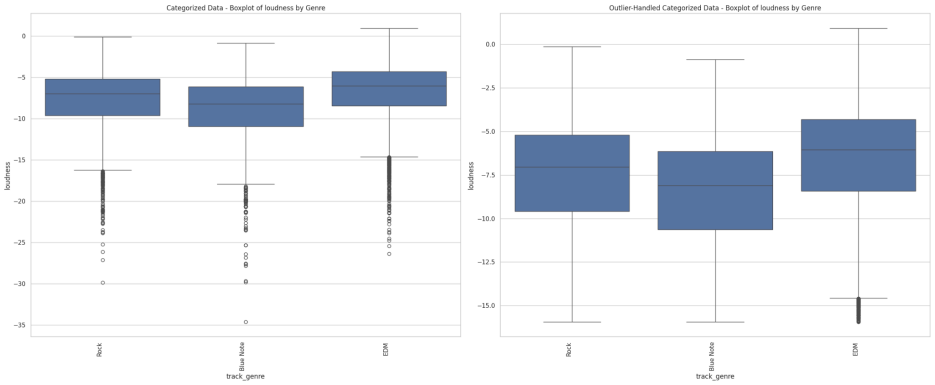
Dengan menggunakan Information Gain sebagai kriteria untuk seleksi fitur, kita dapat secara sistematis mengidentifikasi dan menghapus fitur yang memiliki kontribusi rendah dalam memprediksi variabel target. Proses ini tidak hanya meningkatkan efisiensi dan kinerja model tetapi juga membantu dalam membangun model yang lebih sederhana dan lebih mudah diinterpretasikan. Dengan menetapkan threshold dan menghapus fitur yang berada di bawah ambang batas tersebut, kita memastikan bahwa model hanya mempertahankan fitur-fitur yang benar-benar relevan untuk analisis dan prediksi, sehingga menghasilkan hasil yang lebih akurat dan dapat diandalkan. Dalam final project ini, kami menggunakan threshold 0.05 dan drop fitur *explicit*, *time_signature*, *key*, *mode*, *liveness*.

3.2.3.2. Outlier Handling

Dataset yang digunakan memiliki banyak outlier setiap fiturnya sehingga diperlukan handling untuk meminimalisir outlier. Metode yang digunakan untuk outlier handling adalah IQR (Interquartile Range) method, yaitu mengukur variabilitas dengan membagi dataset menjadi beberapa kuartil.



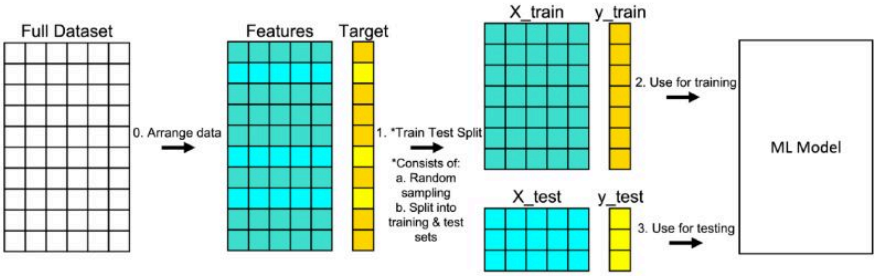
Gambar 16. Boxplot Comparison (duration_ms)



Gambar 17. Boxplot Comparison (loudness)

3.2.3.3. Train Test Split

Sebuah prosedur validasi model untuk mensimulasikan bagaimana sebuah model bekerja pada data yang baru atau tidak terlihat.



Gambar 18. Train Test Split

Prosedur dilakukan dengan memisahkan dataset menjadi dua bagian, training set dan testing set, dengan random sampling.

```
Train set: (14267, 10) (14267,)
Test set: (3567, 10) (3567,)
```

Gambar 19. Train Test Split Result

3.2.3.4. Normalization

Normalisasi dilakukan untuk membuat model menjadi lebih akurat. Umumnya, normalisasi memiliki dua metode, yaitu Z-Score (standarisasi) dan Min-Max Scaling. Pada dataset ini, dilakukan normalisasi menggunakan Min-Max Scaling. Normalisasi Min-Max dilakukan dengan transformasi data original secara linear. Nilai maksimum dan minimum data diambil dan setiap data akan diubah menggunakan formula sebagai berikut.

$$X_{normalisasi} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Formula ini bekerja dengan mengurangi nilai minimum dari nilai original untuk menentukan seberapa jauh nilai original dari nilai minimum lalu

nilai tersebut dibagi oleh range dari variabel (perbedaan antara nilai minimum dan maksimum).

Melakukan normalisasi data setelah membagi dataset menjadi set pelatihan (train) dan set pengujian (test) adalah langkah krusial dalam proses pembelajaran mesin untuk memastikan keakuratan dan validasi model. Ketika kita membagi data menjadi set pelatihan dan pengujian, penting untuk hanya menghitung parameter normalisasi (seperti nilai minimum dan maksimum dalam kasus MinMaxScaler) dari data pelatihan saja. Ini menghindari kebocoran data (data leakage), dimana informasi dari set pengujian, yang seharusnya tetap tidak diketahui selama pelatihan, secara tidak sengaja digunakan. Kebocoran data dapat membuat model tampak bekerja lebih baik daripada yang sebenarnya, karena model telah mempelajari pola dari data pengujian yang seharusnya menjadi data baru dan tidak dikenal selama pelatihan.

Dengan melakukan normalisasi hanya pada data pelatihan dan kemudian menerapkan parameter ini pada data pengujian, kita menjaga skala dan distribusi data pengujian konsisten dengan data pelatihan. Hal ini mencerminkan skenario dunia nyata di mana model kita akan dihadapkan pada data baru yang belum pernah dilihat sebelumnya. Selain itu, ini memastikan bahwa model tidak mendapatkan keuntungan yang tidak adil dengan memiliki akses ke informasi skala dari data pengujian selama pelatihan. Praktik ini juga mendorong generalisasi yang lebih baik, karena model dilatih dan diuji dalam kondisi yang lebih realistis dan tidak dipengaruhi oleh skala atau distribusi yang berbeda antara data pelatihan dan pengujian. Dengan menjaga integritas proses pelatihan dan evaluasi melalui normalisasi yang tepat, kita membangun model yang lebih robust dan mampu bekerja dengan baik pada data baru, yang merupakan tujuan utama dari pembelajaran mesin.

3.2.3.5. Label Encoding

Mengubah variabel kategorikal menjadi representasi numerikal agar dapat dipahami oleh algoritma model yang akan digunakan. Label encoding dilakukan dengan menggunakan Label Encoder. Data yang sudah di-*encode* akan dijadikan to_categorical atau One Hot Encoding untuk digunakan saat processing saat ANN. One-hot encoding memastikan bahwa kategori diperlakukan secara independen dan setara. Dalam ANN, ini membantu jaringan mengenali kategori secara unik tanpa menganggap ada urutan atau hierarki antar kategori. ANN bekerja dengan baik dengan data numerik dalam bentuk vektor atau matriks. One-hot encoded vectors cocok untuk input layer dalam ANN karena setiap kategori direpresentasikan secara eksplisit tanpa ada implikasi hubungan numerik antara kategori.

3.2.4. Model Training

Training pada dataset dilakukan dengan model yang sudah ditentukan sebelumnya, yaitu SVM (Support Vector Machine), ANN (Artificial Neural Network), dan XGBoost (Extreme Gradient Boosting).

4. HASIL DAN PEMBAHASAN

4.1. Skenario Pengujian

Perbandingan model dilakukan dengan membandingkan akurasi, presisi, recall, dan F1-score dari setiap model.

4.1.1. Perbandingan Model Berdasarkan Metode Klasifikasi

Dari ketiga metode klasifikasi, berikut merupakan kompilasi laporan semua pengujian klasifikasi. Setiap metode klasifikasi diuji dengan menggunakan *cross validation* dan normalisasi, serta diuji dengan tanpa menggunakan *cross validation* dan normalisasi. Gambar berikut merupakan urutan dari tingkat hasil pengujian yang terbaik hingga terendah.

	Model	Akurasi	Presisi	Recall_list	F1-Score
1	XGB Unnormalized	0.872442	0.87104	0.872442	0.87121
2	XGB CV Normalized	0.872442	0.87104	0.872442	0.87121
3	XGB CV Unnormalized	0.872442	0.87104	0.872442	0.87121
4	XGB Hyperparameter Tuned	0.872442	0.87104	0.872442	0.87121
5	XGB Normalized	0.869078	0.866977	0.869078	0.867313
6	SVM Normalized	0.826184	0.820881	0.826184	0.822152
7	SVM CV Normalized	0.826184	0.820881	0.826184	0.822152
8	ANN Normalized	0.821419	0.825159	0.821419	0.820826
9	ANN CV Normalized	0.818335	0.815037	0.818335	0.813286
10	SVM Unnormalized	0.58873	0.757873	0.58873	0.436327
11	SVM CV Unnormalized	0.58873	0.757873	0.58873	0.436327
12	ANN Unnormalized	0.58873	0.757873	0.58873	0.436327
13	ANN CV Unnormalized	0.58873	0.757873	0.58873	0.436327

Tabel 2. Kompilasi Classification Report

4.1.1.1. Classifier XGBoost

Berikut merupakan hasil pengujian dengan metode klasifikasi XGBoost.

	Model	Akurasi	Presisi	Recall_list	F1-Score
1	XGB Unnormalized	0.872442	0.87104	0.872442	0.87121
2	XGB CV Normalized	0.872442	0.87104	0.872442	0.87121
3	XGB CV Unnormalized	0.872442	0.87104	0.872442	0.87121
4	XGB Hyperparameter Tuned	0.872442	0.87104	0.872442	0.87121
5	XGB Normalized	0.869078	0.866977	0.869078	0.867313

Tabel 3. Classification Report Classifier XGBoost

Pengujian menggunakan metode klasifikasi XGBoost menunjukkan evaluasi matriks yang paling tinggi dibandingkan dengan model lain dalam dataset ini. Meskipun penggunaan dataset yang dinormalisasi dan tidak dinormalisasi tidak menunjukkan perbedaan signifikan dalam performa model secara keseluruhan, terdapat perbedaan yang mencolok dalam hasil evaluasi matriks antara XGBoost Normalized dan XGBoost CV Normalized. Perbedaan ini disebabkan oleh cara keduanya melakukan proses normalisasi data training. XGBoost Normalized melakukan normalisasi sekali pada seluruh data training sebelum model dilatih, sedangkan XGBoost CV Normalized menggunakan normalisasi yang diterapkan secara terpisah pada setiap fold selama proses cross-validation. Pendekatan XGBoost CV Normalized memastikan bahwa model dievaluasi menggunakan data training yang dinormalisasi secara lokal untuk setiap fold, yang membantu dalam menghasilkan evaluasi yang lebih stabil dan mewakili kemampuan model untuk generalisasi terhadap data yang belum pernah dilihat sebelumnya. Di sisi lain, XGBoost Normalized mungkin lebih rentan terhadap overfitting pada subset data tertentu yang digunakan dalam satu train-test split, karena normalisasi yang dihitung berdasarkan seluruh data training. Dengan demikian, penggunaan cross-validation dan normalisasi lokal pada setiap fold dalam XGBoost CV Normalized memberikan keunggulan dalam menghasilkan model yang lebih robust dan evaluasi matriks yang lebih konsisten dan dapat diandalkan.

Hyperparameter tuning dilakukan menggunakan GridSearchCV dengan hasil parameter terbaik

- 'colsample_bytree': 0.8,
- 'gamma': 0.5,
- 'max_depth': 7,
- 'min_child_weight': 5,
- 'subsample': 0.8

Ketika menggunakan Hyperparameter Tuning pada XGBoost, evaluasi matriks yang dihasilkan tetap sama dengan model XGBoost lainnya. Hal ini bisa disebabkan oleh beberapa faktor. Pertama, parameter default XGBoost sering kali sudah dioptimalkan untuk kinerja yang baik di berbagai jenis dataset, sehingga perubahan parameter tidak memberikan peningkatan signifikan. Kedua, nilai parameter yang Anda eksplorasi dalam GridSearchCV mungkin terlalu dekat dengan nilai default XGBoost. Karena parameter yang dipilih tidak cukup berbeda dari default, efek tuning menjadi minimal, dan hasil evaluasi model tetap sama.**F1 Score**: Mencapai sekitar 87.01%, yang merupakan rata-rata harmonis antara presisi dan recall setelah proses standardisasi.

4.1.1.2. Classifier SVM

Berikut merupakan hasil pengujian dengan metode klasifikasi SVM.

	Model	Akurasi	Presisi	Recall_list	F1-Score
1	SVM Normalized	0.826184	0.820881	0.826184	0.822152
2	SVM CV Normalized	0.826184	0.820881	0.826184	0.822152
3	SVM Unnormalized	0.58873	0.757873	0.58873	0.436327

4	SVM CV Unnormalized	0.58873	0.757873	0.58873	0.436327

Tabel 4. Classification Report Classifier SVM

Evaluasi matriks model Support Vector Machine (SVM) dengan berbagai skenario menunjukkan pentingnya normalisasi data dalam meningkatkan efektivitas model. Berdasarkan hasil evaluasi, model SVM yang menggunakan data yang dinormalisasi, baik dengan maupun tanpa cross-validation menghasilkan evaluasi matriks yang jauh lebih tinggi dibandingkan dengan model yang menggunakan data tidak dinormalisasi. Misalnya, akurasi dan recall model pada data yang dinormalisasi mencapai 0.826184, sementara pada data yang tidak dinormalisasi hanya 0.588730. Hal ini menunjukkan bahwa normalisasi membantu model dalam mengenali dan memisahkan kelas-kelas data secara lebih efektif. Sebaliknya, model yang dilatih pada data yang tidak dinormalisasi mengalami penurunan performa yang signifikan, terutama terlihat pada metrik recall dan F1-Score, meskipun presisinya lebih tinggi. Presisi yang lebih tinggi pada model tanpa normalisasi mungkin menunjukkan bahwa model lebih jarang memprediksi kelas positif, sehingga lebih selektif, tetapi ini tidak diimbangi dengan kemampuan deteksi keseluruhan yang baik.

Selain itu, hasil menunjukkan bahwa tidak ada perbedaan kinerja yang signifikan antara model yang menggunakan cross-validation dan yang tidak, baik pada data yang dinormalisasi maupun yang tidak. Ini mengindikasikan bahwa cross-validation tidak memberikan keuntungan tambahan dalam konteks ini, terutama ketika data sudah dinormalisasi. Kesimpulannya, normalisasi data sebelum melatih model SVM sangat penting karena membantu model dalam belajar dari fitur yang seimbang dan menghindari bias yang disebabkan oleh skala yang berbeda pada fitur input. Oleh karena itu, untuk mencapai performa model yang optimal, normalisasi data adalah langkah yang sangat direkomendasikan dalam proses preprocessing untuk model dengan metode SVM.

4.1.1.3. Classifier ANN

Berikut merupakan hasil pengujian dengan metode klasifikasi ANN.

	Model	Akurasi	Presisi	Recall_list	F1-Score
1	ANN Normalized	0.821419	0.825159	0.821419	0.820826
2	ANN CV Normalized	0.818335	0.815037	0.818335	0.813286
3	ANN Unnormalized	0.58873	0.757873	0.58873	0.436327
4	ANN CV Unnormalized	0.58873	0.757873	0.58873	0.436327

Tabel 5. Classification Report Classifier ANN

Evaluasi performa model Artificial Neural Network (ANN) menunjukkan pentingnya normalisasi data dalam meningkatkan hasil evaluasi. Model ANN dengan data yang dinormalisasi, baik dengan atau tanpa cross-validation, memiliki kinerja yang jauh lebih baik dalam hal akurasi, presisi, recall, dan F1-Score dibandingkan model yang tidak dinormalisasi. Misalnya,

akurasi model ANN Normalized mencapai 0.821419 dan ANN CV Normalized adalah 0.818335, sedangkan akurasi model yang tidak dinormalisasi hanya 0.588730.

Normalisasi membantu model memproses fitur dengan lebih efektif karena skala yang seimbang, sementara model yang tidak dinormalisasi kesulitan mengenali pola akibat skala fitur yang berbeda, menghasilkan performa buruk. Meskipun presisi dan recall pada model yang tidak dinormalisasi tinggi, F1-Score yang rendah menunjukkan ketidakseimbangan dan banyak kesalahan prediksi. Cross-validation pada data yang dinormalisasi sedikit menurunkan performa metrik evaluasi tetapi membuat model lebih robust terhadap overfitting. Sebaliknya, pada data yang tidak dinormalisasi, cross-validation tidak memperbaiki kelemahan yang ada.

4.1.2. Perbandingan Model Berdasarkan Normalization atau Unnormalized

	Model	Akurasi	Presisi	Recall_list	F1-Score
1	XGB CV Normalized	0.872442	0.87104	0.872442	0.87121
2	XGB Normalized	0.869078	0.866977	0.869078	0.867313
3	SVM Normalized	0.826184	0.820881	0.826184	0.822152
4	SVM CV Normalized	0.826184	0.820881	0.826184	0.822152
5	ANN Normalized	0.821419	0.825159	0.821419	0.820826
6	ANN CV Normalized	0.818335	0.815037	0.818335	0.813286

Tabel 6. Classification Report Normalization

	Model	Akurasi	Presisi	Recall_list	F1-Score
1	XGB Unnormalized	0.872442	0.87104	0.872442	0.87121
2	XGB CV Unnormalized	0.872442	0.87104	0.872442	0.87121
3	SVM Unnormalized	0.58873	0.757873	0.58873	0.436327
4	SVM CV Unnormalized	0.58873	0.757873	0.58873	0.436327
5	ANN Unnormalized	0.58873	0.757873	0.58873	0.436327
6	ANN CV Unnormalized	0.58873	0.757873	0.58873	0.436327

Tabel 7. Classification Report Classifier Unnormalized

Model XGBoost (XGB) yang menggunakan normalisasi (XGB Normalized dan XGB CV Normalized) menunjukkan hasil evaluasi matriks yang sangat serupa. XGB CV Normalized dan XGB Normalized memiliki akurasi yang tinggi sekitar 0.87, dengan presisi, recall, dan F1-Score yang juga cukup tinggi, yaitu sekitar 0.87. Hal ini menunjukkan bahwa normalisasi data memberikan dampak positif pada kinerja XGBoost, memungkinkan model untuk belajar dengan baik dari data yang memiliki skala yang seragam.

Model SVM (SVM Normalized dan SVM CV Normalized) dan ANN (ANN Normalized dan ANN CV Normalized) juga menunjukkan performa yang baik saat menggunakan normalisasi. Mereka memiliki akurasi yang lebih rendah dibandingkan dengan XGBoost, tetapi masih cukup baik, berkisar antara 0.82 hingga 0.82. Presisi, recall, dan F1-Score untuk kedua model ini juga menunjukkan performa yang seimbang, menandakan kemampuan model untuk mengklasifikasikan dengan baik antara kelas positif dan negatif.

Model XGBoost (XGB Unnormalized dan XGB CV Unnormalized), meskipun tidak menggunakan normalisasi, tetap menunjukkan hasil evaluasi matriks yang identik dengan model yang menggunakan normalisasi. Ini menandakan bahwa XGBoost cenderung lebih toleran terhadap variasi dalam skala fitur ketika menggunakan dataset yang tidak dinormalisasi.

Namun, untuk model SVM dan ANN (SVM Unnormalized, SVM CV Unnormalized, ANN Unnormalized, dan ANN CV Unnormalized), hasil evaluasi matriks menunjukkan kinerja yang jauh lebih rendah dibandingkan dengan model yang menggunakan normalisasi. Akurasi, presisi, recall, dan F1-Score semua memiliki nilai yang rendah sekitar 0.58. Hal ini menunjukkan bahwa model-model ini mengalami kesulitan dalam memahami pola dari data yang memiliki skala yang berbeda-beda, yang menghasilkan performa yang buruk dalam klasifikasi.

Secara keseluruhan, hasil evaluasi ini menggarisbawahi pentingnya normalisasi dalam preprocessing data sebelum melatih model. Normalisasi memungkinkan model untuk belajar dengan lebih efektif dan menghasilkan hasil yang lebih andal, terutama untuk model yang sensitif terhadap skala fitur seperti SVM dan ANN. Sedangkan untuk XGBoost, meskipun normalisasi memberikan keuntungan dalam konsistensi performa, model tersebut juga mampu menghasilkan hasil yang baik bahkan tanpa normalisasi, menunjukkan keunggulannya dalam menangani data dengan skala yang bervariasi.

4.1.3. Perbandingan Model Berdasarkan *Cross Validation* atau *Non Cross Validation*

	Model	Akurasi	Presisi	Recall_list	F1-Score
1	XGB CV Normalized	0.872442	0.87104	0.872442	0.87121
2	XGB CV Unnormalized	0.872442	0.87104	0.872442	0.87121
3	SVM CV Normalized	0.826184	0.820881	0.826184	0.822152
4	ANN CV Normalized	0.818335	0.815037	0.818335	0.813286
5	SVM CV Unnormalized	0.58873	0.757873	0.58873	0.436327
6	ANN CV Unnormalized	0.58873	0.757873	0.58873	0.436327

Tabel 8. Classification Report Classifier Cross Validation

	Model	Akurasi	Presisi	Recall_list	F1-Score
1	XGB Unnormalized	0.872442	0.87104	0.872442	0.87121
2	XGB Hyperparameter Tuned	0.872442	0.87104	0.872442	0.87121
3	XGB Normalized	0.869078	0.866977	0.869078	0.867313
4	SVM Normalized	0.826184	0.820881	0.826184	0.822152
5	ANN Normalized	0.821419	0.825159	0.821419	0.820826
6	SVM Unnormalized	0.58873	0.757873	0.58873	0.436327
7	ANN Unormalized	0.58873	0.757873	0.58873	0.436327

Tabel 9. Classification Report Classifier Non Cross Validation

Penggunaan Cross-Validation (CV) memberikan stabilitas dan konsistensi performa model machine learning. Model XGBoost, baik yang menggunakan CV maupun tidak, menunjukkan akurasi tinggi sekitar 87.24%, dengan sedikit penurunan pada model yang dinormalisasi tanpa CV (86.91%). Sementara itu, model SVM dan ANN menunjukkan performa yang lebih baik ketika menggunakan CV dibandingkan tanpa CV. SVM CV Normalized memiliki akurasi 82.62%, yang sama dengan model SVM Normalized tanpa CV, menunjukkan konsistensi yang baik. Namun, ANN CV Normalized menunjukkan akurasi 81.83%, sedikit lebih tinggi daripada ANN Normalized tanpa CV (82.14%), mengindikasikan sedikit penurunan tanpa CV. Pada model yang tidak dinormalisasi, baik SVM maupun ANN menunjukkan performa yang sangat rendah (58.87%) dengan nilai F1-Score yang juga rendah, menandakan bahwa normalisasi data sangat penting untuk performa yang baik. Secara keseluruhan, CV membantu mengevaluasi model dengan lebih akurat, menghindari overfitting, dan memberikan hasil yang lebih stabil terutama pada dataset yang lebih bervariasi.

4.2. Analisis Hasil Pengujian

Secara keseluruhan, model XGB menunjukkan konsistensi yang tinggi di seluruh variasi dengan nilai Akurasi, Presisi, Recall, dan F1-Score mencapai sekitar 87% untuk setiap metrik. Bahkan dalam skenario tanpa normalisasi dan tanpa cross-validation, XGBoost tetap menunjukkan performa yang unggul. Hal ini mengindikasikan bahwa XGBoost memiliki kemampuan adaptif yang kuat terhadap dataset, tanpa sangat bergantung pada proses normalisasi atau validasi silang untuk mencapai kinerja optimal. Keunggulan ini mungkin berasal dari kemampuan XGBoost dalam menangani perbedaan skala antar fitur dan optimalisasi melalui teknik gradient boosting yang memperbaiki kesalahan model secara iteratif.

Model Support Vector Machine (SVM), terlihat bahwa penggunaan normalisasi dan cross-validation memberikan pengaruh signifikan terhadap kinerjanya. Variasi SVM yang dinormalisasi dan menggunakan CV menghasilkan akurasi dan F1-Score sekitar 82-83%, menunjukkan peningkatan yang stabil dibandingkan dengan versi tanpa normalisasi yang hanya mencapai sekitar 59%. Normalisasi membantu SVM dalam mengatasi perbedaan skala antar fitur, yang penting untuk algoritma berbasis jarak

seperti SVM. Cross-validation juga membantu meningkatkan generalisasi model, mengurangi risiko overfitting dengan memastikan bahwa model dievaluasi pada berbagai subset data.

Model Artificial Neural Network (ANN) menunjukkan pola performa yang mirip dengan SVM. Variasi ANN dengan normalisasi dan CV mencapai akurasi dan F1-Score sekitar 82%, sementara versi tanpa normalisasi hanya mencapai sekitar 59%. Normalisasi tampaknya sangat penting untuk model ANN, karena membantu konvergensi jaringan dalam proses pelatihan dengan mengatur nilai input dalam skala yang konsisten. Validasi silang juga memberikan keuntungan dengan memastikan evaluasi yang lebih robust terhadap kinerja model.

Secara keseluruhan, hasil pengujian ini menunjukkan bahwa XGBoost memiliki performa yang konsisten dan kuat di seluruh variasi, sementara SVM dan ANN menunjukkan peningkatan dalam performa ketika menggunakan normalisasi. Evaluasi lebih lanjut dan penyesuaian parameter mungkin diperlukan untuk memaksimalkan performa masing-masing model sesuai dengan konteks dan tujuannya.

Pengujian lebih lanjut menunjukkan bahwa kinerja model dengan cross validation lebih optimal dibandingkan dengan model yang tidak menggunakan cross validation. Cross validation membantu dalam memastikan bahwa model tidak overfitting terhadap data pelatihan dan memberikan gambaran yang lebih akurat tentang performa model pada data yang tidak terlihat. Selain itu, kinerja model yang dinormalisasi lebih optimal daripada model yang tidak dinormalisasi. Normalisasi membantu dalam mempercepat proses pelatihan dengan menghindari nilai yang terlalu besar atau terlalu kecil. Dengan normalisasi, model ANN dan SVM dapat mencapai tingkat akurasi dan F1-Score yang lebih tinggi, menunjukkan bahwa praktik ini sangat penting untuk model pembelajaran mesin.

5. KESIMPULAN

Model yang memiliki nilai akurasi paling baik adalah XGBoost. Dari hasil ini, jelas bahwa model XGBoost menunjukkan keunggulan dalam konsistensi performa, baik dalam kondisi dengan atau tanpa normalisasi dan cross-validation. Hal ini menegaskan fleksibilitas dan kekuatan XGBoost dalam menangani berbagai situasi data yang kompleks. Sebaliknya, SVM dan ANN menunjukkan ketergantungan yang lebih besar pada normalisasi dan *cross validation* untuk mencapai kinerja optimal. Normalisasi dan Cross Validation terbukti penting untuk mengatasi masalah skala fitur pada SVM dan ANN. Secara keseluruhan, hasil ini memberikan informasi bagaimana setiap model bereaksi terhadap tahap-tahap pada proses preprocessing dan *cross validation*. Final project ini dapat dikembangkan lagi dengan menggunakan pengujian atau metode lain untuk pembuatan sistem rekomendasi lagu berdasarkan genre bagi pengguna.

DAFTAR PUSTAKA

Akter, M., Sultana, N., Noori, S. R. H., & Hasan, M. Z. (2023). Bangla Song Genre Recognition using Artificial Neural Network. *IAES International Journal of Artificial Intelligence (IJ-AI)*, 13(2), 2413-2422. https://doi.org/10.11591/ijai.v13.i2.pp2413-2422

Bansal, M., Goyal, A., & Choudhary, A. (2022). A Comparative Analysis of K-Nearest Neighbor, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory Algorithms in Machine Learning. *Decision Analytics Journal*, 3, 100071. https://doi.org/10.1016/j.dajour.2022.100071

B N, S., & Akki, C. B. (2021). Sentiment Prediction using Enhanced XGBoost and Tailored Random Forest. *International Journal of Computing and Digital Systems, 10*(1). http://journals.uob.edu.bh

Haryandri, Endlessia R. A. (2020). Tinjuan Bentuk Karya Solo Piano *"The Dancer"* Ciptaan Levi Gunardi. *Jurnal Penelitian Musik*. 1 (1), 62-74.

Hendri, D., Nadha, D., Basri, F. K., Wajdi, M. F., & Nadhirah, N. (2024). Comparation of Decision Tree Algorithm, Naive Bayes, K-Nearest Neighbors on Spotify Music Genre. *Indonesian Journal of Applied Technology and Innovation Science*, 1(1), 47-53. https://doi.org/10.57152/IJATIS.v1i1.1219

Herdian, C., Kamila, A., & Budidarma, I. G. A. M. (2024). Studi Kasus Feature Engineering untuk Data Teks: Perbandingan Label Encoding dan One-hot Encoding pada Metode Linear Regresi. *Technologia*, 15(1), 94. http://dx.doi.org/10.31602/tji.v15i1.13457

Lee, Ga, Alzamil, Lubna, Doskenov, Bakhtiyar, dan Termehchy, Arash. (2021). A Survey on Data Cleaning Methods for Improved Machine Learning Model Performance.

Muren, J. (2019). Classification of Music Genres with Extreme Gradient Boosting. *Bachelor Thesis*, Stockholm University. Retrieved from http://www.math.su.se

Pramono, M.. (2021). Hubungan Minat pada Musik dengan Kreativitas. Surabaya, Indonesia: Universitas 17 Agustus 1945 Surabaya.

Prasetiyowati, M. I., Maulidevi, N. U., & Surendro, K. (2021). Determining Threshold Value on Information Gain Feature Selection to Increase Speed and Prediction Accuracy of Random Forest. *Journal of Big Data, 8*, 84. https://doi.org/10.1186/s40537-021-00472-4

Sagala, N. T. M., & Aryatama, F. Y. (2022). Exploratory data analysis (EDA): A Study of Olympic Medallist. *SISTEMASI: Jurnal Sistem Informasi*, 11(3), 578-587. http://sistemasi.ftik.unisi.ac.id

Setiadi, D. R. I. M., Susanto, A., Rahardwika, D. S., Mulyono, I. U. W., Rachmawanto, E. H., Astuti, E. Z., Sari, C. A., & Fahmi, A. (2020). Effect of Feature Selection on the Accuracy of Music Genre Classification using SVM Classifier. In *2020 International Seminar on Application for Technology of Information and Communication (iSemantic)*. Semarang, Indonesia: Dian Nuswantoro University. https://doi.org/10.1109/iSemantic50169.2020.9234245

Si, T., Bagchi, J., & Miranda, P. B. C. (2022). Artificial Neural Network training using metaheuristics for medical data classification: An experimental study. *Expert Systems with Applications, 193*, 116423. https://doi.org/10.1016/j.eswa.2021.116423

Unwin, A. (2020). Why Is Data Visualization Important? What Is Important in Data Visualization? Harvard Data Science Review, 2(1). https://doi.org/10.1162/99608f92.8ae4d525

Vodithala, S., Gudimalla, V., Bhavani, Y., Madadi, P., & Waseem, M. S. (2023). Recommendation of Algorithm for Efficient Retrieval of Songs from Musical Dataset. *International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC)*, 11(5s), 276. https://doi.org/10.17762/ijritcc.v11i5s.6654