JIKA (Jurnal Informatika) Universitas Muhammadiyah Tangerang *P ISSN : 2549-0710*
Tangerang, January 2025, Vol 9, No 1 pp 16-25 *E ISSN : 2722-2713*

OPTIMALISASI ALGORITMA RANDOM FOREST -*FEATURE SELECTION* DAN *HYPERPARAMETER TUNING* KLASIFIKASI GENRE MUSIK

Fathur Fakhriza[1)], Dayat Subekti2[2)], Puji Winar Cahyo[3)]
[1,2,3] Informatika Fakultas Teknik dan Teknologi Informasi, Universitas Jenderal Achmad Yani Yogyakarta
Jl. Siliwangi Jl. Ringroad Barat, Area Sawah, Banyuraden, Kabupaten Sleman, Daerah Istimewa Yogyakarta
Co Responden Email: fakhrizafathur05@gmail.com

Article history
Received 07 Aug 2024
Revised 11 Oct 2024
Accepted 10 Nov 2024
Available online 31 Jan 2025

Keywords
Genres,
Random forest,
Feature selection,
Hyperparameter,
Tuning

Abstract

Listening to music is an important aspect of human life, but the subjective recognition of music genres adds complexity to the classification process. Therefore, a careful and reliable approach is needed to analyze and categorize music data. The Random Forest method is widely used in genre classification, requiring precise algorithm optimization through Feature Selection and Hyperparameter Tuning. The benefit of this research is to provide an understanding of the role of Feature Selection and Hyperparameter Tuning techniques in optimizing the performance of the Random Forest algorithm. By maximizing the algorithm's potential, the accuracy of music genre classification can be improved, which plays a crucial role in developing a more precise and accurate music recommendation system. This research begins with data collection, processed through preprocessing to obtain clean data. Features in the dataset are selected via Feature Selection to identify those that best represent music genre classes. The Random Forest method is then applied for classification, followed by Hyperparameter Tuning to find the optimal parameters. Testing results show that the Random Forest method achieved an ROC AUC value of 0.909. Optimization improved performance, raising the ROC AUC value to 0.913, indicating a model performance increase of 0.004 and placing it in the excellent evaluation category.

Riwayat
Diterima 07 Agu 2024
Revisi 11 Okt 2024
Disetujui 10 Nov 2024
Terbit online 31 Jan 2025

Kata Kunci
Genre,
Random forest,
Pemilihan fitur,
Hyperparameter,
Tuning

Abstrak

Mendengarkan musik merupakan aspek penting dari kehidupan manusia, namun pengenalan genre musik secara subjektif menambah kompleksitas dalam proses klasifikasinya. Oleh karena itu, diperlukan pendekatan yang teliti dan andal untuk menganalisis serta mengelompokkan data musik. Metode Random Forest banyak digunakan dalam klasifikasi genre musik, memerlukan optimalisasi algoritma yang presisi melalui *Feature Selection* dan *Hyperparameter Tuning*. Manfaat penelitian ini yaitu untuk memberikan pemahaman mengenai peran teknik *Feature Selection* dan *Hyperparameter Tuning* dalam mengoptimalkan performa algoritma Random Forest. Dengan memanfaatkan algoritma secara maksimal, akurasi klasifikasi genre musik dapat ditingkatkan, yang berperan penting dalam menciptakan sistem rekomendasi musik yang lebih tepat dan akurat. Penelitian ini diawali dengan pengumpulan data yang diolah dalam proses *preprocessing* untuk mendapatkan data yang bersih. Fitur-fitur dalam dataset dipilih melalui *Feature Selection* untuk mendapatkan fitur yang mampu merepresentasikan kelas genre musik. Metode Random Forest digunakan untuk klasifikasi, diikuti dengan *Hyperparameter Tuning* untuk mendapatkan parameter yang optimal. Hasil pengujian menunjukkan bahwa metode Random Forest memiliki nilai ROC AUC sebesar 0.909. Optimalisasi meningkatkan kinerja dengan nilai ROC AUC menjadi 0.913, menunjukkan peningkatan kinerja model sebesar 0.004 dan masuk kategori evaluasi yang *excellent*.

PENDAHULUAN

Mendengarkan musik merupakan bagian yang tidak bisa dipisahkan dari kehidupan manusia. Musik memegang peran utama sebagai bentuk hiburan yang sangat penting dalam kehidupan manusia. Musik memberikan wadah untuk mengekspresikan emosi dan perasaan yang dimiliki oleh seseorang

(Singhal et al., 2022). Dengan adanya kemajuan teknologi, seseorang dapat dengan mudah untuk mendengarkan musik. Mendengarkan musik dapat dilakukan dimanapun dan kapanpun dengan bantuan internet melalui aplikasi *streaming* seperti Spotify, AppleMusic, Tuneln, Youtube Music dan Soundcloud. Menurut data yang dilaporkan oleh We Are Social, sebanyak 50,3% penduduk Indonesia yang menggunakan internet mendengarkan lagu melalui layanan *streaming* musik pada kuartal III tahun 2022. Persentase tersebut mengalami peningkatan sebesar 2,8% dibandingkan dengan periode yang sama tahun sebelumnya, yang sebelumnya mencapai 47,5% (We Are Social, 2022).

Musik dapat dikategorikan berdasarkan genre yang dimiliki. Secara sederhana genre musik dapat diartikan sebagai tipe, pengkategorian, aliran atau jenis dari suatu musik (Navisa et al., 2021). Pengkategorian musik oleh penyedia layanan bertujuan untuk mempermudah pengelompokan musik yang akan dipresentasikan di platform digital penyedia layanan musik. Tampilan rekomendasi yang berbeda akan diberikan kepada setiap pendengar berdasarkan preferensi pada genre musik yang sering, jarang, atau bahkan tidak pernah didengarkan pada perangkat yang dimiliki. Hal ini terjadi karena adanya proses klasifikasi berdasarkan genre yang didengarkan oleh pengguna. Klasifikasi genre musik menjadi sebuah tantangan yang rumit karena adanya beragam variasi dalam genre musik dan juga memungkinan adanya perpaduan elemen dari berbagai genre (Ghosh et al., 2023). Genre musik yang bervariatif sering kali memiliki kemiripan satu sama lain. Hal ini dapat menyebabkan pendengar kesulitan mengidentifikasi dan mengkategorikan genre dari lagu yang mereka dengarkan. Faktor subjektif dalam mengidentifikasi genre musik juga menambah kompleksitas dalam proses klasifikasi. Oleh karena itu, diperlukan suatu pendekatan yang teliti dan kuat dalam menganalisis serta mengelompokkan data musik.

Dalam beberapa tahun terakhir, metode Random Forest telah menjadi salah satu metode yang diminati dalam klasifikasi genre musik. Hal ini disebabkan oleh kecakapannya dalam menangani *dataset* yang rumit dengan efektif dan memberikan kinerja yang memuaskan dalam berbagai situasi (Tanujaya et al., 2020). Random Forest adalah sebuah algoritma *ensemble* yang menggabungkan *multiple decision trees* untuk menghasilkan prediksi yang lebih akurat.

Algoritma *machine learning*, termasuk Random Forest terbukti memiliki efektivitas dalam tugas klasifikasi data musik. Untuk meningkatkan kinerja klasifikasi, diperlukan optimalisasi algoritma yang akurat dan pemilihan fitur yang sesuai (*Feature Selection*). Pemilihan fitur yang relevan dan eliminasi fitur yang tidak penting dapat meningkatkan tingkat akurasi klasifikasi dan mengurangi kompleksitas data, sehingga mengoptimalkan waktu komputasi (Khan et al., 2022).

Disamping itu, penyesuaian parameter (*Hyperparameter Tuning*) pada algoritma *machine learning* juga memiliki potensi untuk memengaruhi performa klasifikasi (Belete & Huchaiah, 2022). Oleh karena itu, *Hyperparameter Tuning* memungkinkan penyetelan parameter algoritma untuk meningkatkan adaptabilitasnya terhadap karakteristik data musik yang beragam.

Berdasarkan uraian permasalahan diatas, maka penelitian ini berfokus pada permasalahan optimalisasi *Feature Selection* dan *Hyperparameter Tuning* pada algoritma Random Forest agar mendapatkan model yang lebih baik.

Tujuan utama dari penelitian ini yaitu untuk menerapkan *Feature Selection* dan *Hyperparameter Tuning* pada optimalisasi algoritma Random Forest untuk klasifikasi genre musik serta untuk mengetahui evaluasi algoritma Random Forest sebelum dan sesudah menggunakan *Hyperparameter Tuning* untuk klasifikasi genre musik.

Manfaat dari penelitian ini:

1. Memberikan pengetahuan kepada pengguna mengenai klasifikasi genre musik.
2. Memberikan pengetahuan mengenai penggunaan *Feature Selection* dan *Hyperparameter Tuning* pada algoritma Random Forest untuk klasifikasi genre musik.
3. Mengetahui hasil evaluasi algoritma Random Forest sebelum dan sesudah menerapkan optimalisasis menggunakan

Hyperparameter Tuning untuk klasifikasi genre musik.

METODE PENELITIAN

Penelitian ini merupakan upaya untuk memperbaiki dan mengoptimalkan performa algoritma Random Forest dengan menerapkan pendekatan *Feature Selection* dan *Hyperparameter Tuning* dalam konteks klasifikasi genre musik. Langkah awal penelitian ini melibatkan pemahaman mendalam terhadap latar belakang dan tantangan yang dihadapi dalam klasifikasi genre musik, analisis terhadap proses-proses yang terlibat dalam pengklasifikasian tersebut, pengidentifikasian faktor-faktor krusial yang memengaruhi hasil klasifikasi, dan akhirnya merancang serta mengembangkan sistem yang mampu meningkatkan akurasi klasifikasi genre musik dengan memanfaatkan algoritma Random Forest yang telah dioptimalkan. Proses pengembangan sistem ini melibatkan serangkaian eksperimen yang mencakup pemilihan fitur-fitur yang relevan dengan genre musik, penyetelan *hyperparameter* dan evaluasi menyeluruh terhadap kinerja sistem yang dikembangkan.

1. Bahan dan Alat Penelitian

Penelitian ini menggunakan jenis data kuantitatif yang bersumber dari data sekunder. Data sekunder merujuk pada data yang telah dikumpulkan oleh pihak lain dan tidak terkait langsung dengan objek penelitian. Data tersebut diperoleh dari sumber kedua, yang biasanya tersedia di berbagai situs web yang menyediakan akses terhadap kumpulan data global. Informasi yang dikumpulkan dan dipublikasikan dalam situs data publik tersebut dapat digunakan oleh peneliti lain untuk melakukan eksperimen atau penelitian. Sumber data pada penelitian ini merupakan *dataset* Spotify Tracks Dataset dari website Kaggle yang dapat diakses melalui tautan https://www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset.

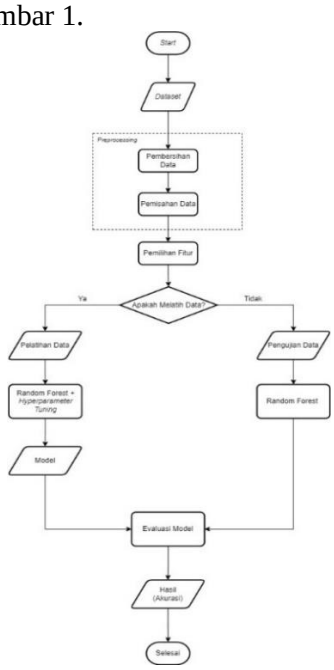
Peralatan yang diperlukan dalam penelitian ini adalah sebuah komputer dengan spesifikasi yang memadai untuk menjalankan sistem operasi dan perangkat lunak pengembangan, serta memiliki koneksi internet yang stabil.

Sistem Operasi dan program-program aplikasi yang dipergunakan dalam dalam pengembangan aplikasi ini adalah:

1. Sistem Operasi: Windows 10 atau lebih baru.
2. Bahasa pemrograman: Python dengan *library* yang digunakan diantaranya Numpy, Pandas, Matplotlib, Seaborn, Scikit-Learn, dan Flask.
3. *Code editor* : Visual Studio Code atau yang lebih baik.
4. *Web browser* : Google Chrome atau yang lebih baik.
5. Lingkungan pengembangan interaktif : Google Colab.

2. Jalan Penelitian

Penelitian ini dilakukan menggunakan bahasa pemrograman Python dengan memerlukan beberapa tahapan agar memudahkan penelitian untuk membangun mekanisme atau langkah-langkah yang harus dilakukan dalam penelitian ini. Untuk mengetahui lebih jelasnya maka akan gambarkan dalam bentuk *flowchart* pada Gambar 1.



Gambar 1 Flowchart jalan penelitian

Secara umum, penelitian ini dimulai dengan pengumpulan data yang akan digunakan untuk proses klasifikasi. Setelah itu data yang telah dikumpulkan dilakukan pembersihan data untuk meningkatkan kualitas data agar lebih akurat. Kemudian atribut-atribut penting yang dapat merepresentasikan kelas dipilih menggunakan *Feature Selection*. Data yang telah dibersihkan kemudian dibagi menjadi data *train, test* dan *validation*. Selanjutnya data tersebut dilakukan klasifikasi dengan menggunakan metode

Random Forest serta menerapkan *Hyperparameter Tuning*. Evaluasi performa model dengan menggunakan metrik evaluasi ROC AUC.

Berdasarkan *flowchart* pada Gambar 1, jalan penelitian yang akan dilakukan pada penelitian ini yaitu sebagai berikut :

2.1. *Dataset*

Data yang digunakan pada penelitian ini berupa kumpulan fitur audio dari sebuah lagu yang diperoleh dari situs Kaggle melalui tautan https://www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset. Data tersebut berisikan 114.000 lagu yang memiliki atribut track_id, artists, album_name, track_name, popularity, duration_ms, explicit, danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valance, tempo, time_signature dan track_genre.

2.2. *Preprocessing*

Preprocessing merupakan serangkaian proses untuk membersihkan, mengubah, dan menyesuaikan data mentah sebelum dimasukkan ke dalam model klasifikasi dengan tujuan untuk meningkatkan akurasi dan kualitas model klasifikasi (Ma et al., 2022). Adapun langkah yang dilakukan pada tahap ini meliputi :

1. *Cleaning data*, pembersihan data dilakukan untuk memastikan data siap untuk diproses oleh mesin (Avci et al., 2023). Berikut beberapa langkah *cleaning data* yang dilakukan pada dataset berbentuk numerik :
    a. *Check missing value*, dilakukan untuk melihat data apakah memiliki nilai yang hilang atau kosong. Jika telah ditemukan data yang kosong maka akan dilakukan tindakan untuk mengatasinya seperti menghapus atribut yang memiliki nilai yang kosong.
    b. *Class-aware data cleaning*, proses ini bertujuan untuk memilih nilai kelas yang akan digunakan pada proses klasifikasi.
2. *Splitting data*, merupakan proses memisahkan data menjadi tiga kategori yaitu data *training*, *validation*, dan *testing* (Azmi et al., 2023).

2.3. Tahap pemilihan fitur (*Feature Selection)*

Feature Selection merupakan proses pemilihan subset fitur yang paling relevan dan informatif dari *dataset*. Tujuannya adalah untuk meningkatkan kinerja model dengan mengurangi dimensi data, kompleksitas, serta meningkatkan interpretasi hasil (Setiadi et al., 2020). Dengan memilih fitur-fitur yang paling signifikan, proses ini dapat mempercepat waktu komputasi dan meningkatkan akurasi prediksi model.

2.4. Tahap perancangan sistem

Implementasi algoritma dilakukan dalam bentuk program modular yang telah dirancang menggunakan bahasa pemrograman Python dengan menggunakan beberapa *library* seperti Pandas, Numpy, Matplotlib dan Sci-kit Learn. Aplikasi ini dijalankan pada platform *cloud computing* gratis yang diberikan oleh Google, yang dikenal sebagai Google Colab. Platform ini memungkinkan pengguna untuk mengakses lingkungan Jupyter Notebook secara *online* dan menjalankan program Python tanpa memerlukan instalasi Python atau *library* yang diperlukan pada komputer lokal. Selain itu, Google Colab menyediakan sumber daya komputasi yang cukup besar, seperti CPU, GPU, dan TPU yang dapat digunakan untuk melatih model dengan efisien.

2.5. Evaluasi model

Tahap evaluasi dilaksanakan pada akhir proses klasifikasi. Tahap ini bertujuan untuk menguji model yang telah dilakukan pengujian. Metrik evaluasi yang digunakan pada penelitian ini yaitu ROC AUC (Yang et al., 2022). Langkah – langkah dalam proses evaluasi yaitu sebagai berikut.

1. Gunakan model yang telah dilatih untuk memprediksi probabilitas kelas positif pada data pengujian.
2. Hitung nilai *True Positive Rate* (TPR) dan *False Positive Rate* (FPR) pada berbagai *threshold*.
3. Plot TPR *versus* FPR untuk berbagai *threshold* untuk melihat kurva ROC.
4. Hitung *Area Under Curve* (AUC) dari ROC *Curve* untuk mendapatkan skor ROC AUC.
5. Interpretasi nilai ROC AUC membantu memahami seberapa baik model dapat membedakan antara kelas positif dan negatif.

HASIL DAN PEMBAHASAN

1. Hasil Pengumpulan Data

Penelitian ini menggunakan jenis dataset public yang bersumber dari website Kaggle, yaitu Spotify Tracks Dataset yang berisi kumpulan audio feature dari sebuah lagu yang bersumber dari Spotify API. Data tersebut berisikan 114.000 lagu yang memiliki atribut track_id, artists, album_name, track_name, popularity, duration_ms, explicit, danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valance, tempo, time_signature dan track_genre seperti pada Gambar 2.

Gambar 2 *Dataset preview*

2. Hasil Pengolahan Data

Proses pengolahan data dilakukan sebelum tahap penerapan algoritma klasifikasi yang disebut *preprocessing*. Adapun langkah–langkahnya yaitu sebagai berikut.

2.1. *Cleaning* data

Cleaning data dilakukan dengan tujuan untuk memastikan data siap untuk diproses oleh mesin. Terdapat beberapa tahapan yang dilakukan pada data numerik yaitu sebagai berikut.

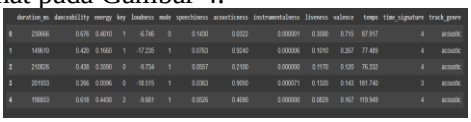
1. *Check missing value*

Check Missing Value merupakan proses yang bertujuan untuk melihat data apakah memiliki nilai yang hilang atau kosong. Pada *dataset* yang akan digunakan, terdapat beberapa atribut yang memiliki nilai kosong yang dapat dilihat dari Gambar 3.

```
Unnamed: 0          0
track_id            0
artists             1
album_name          1
track_name          1
popularity          0
duration_ms         0
explicit            0
danceability        0
energy              0
key                 0
loudness            0
mode                0
speechiness         0
acousticness        0
instrumentalness    0
liveness            0
valence             0
tempo               0
time_signature      0
track_genre         0
dtype: int64
```

Gambar 3 Hasil *check missing value*

Berdasarkan gambar Gambar 3 atribut yang memiliki nilai kosong yaitu artists, album_name dan track_name. Selain itu, beberapa atribut berisikan identitas sebuah lagu dan memiliki nilai yang unik diantaranya yaitu Unnamed: 0, track_id, popularity dan explicit. Atribut tersebut kemudian dihapus karena tidak berkaitan dengan genre musik. Hasil setelah *Check Missing Value* dapat dilihat pada Gambar 4.



Gambar 4 *Dataset preview* setelah *check missing value*

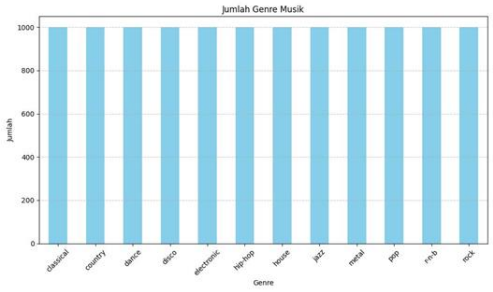
2. *Class-aware* Data *Cleaning*

Pada *dataset* tersebut, terdapat beberapa genre yang merupakan turunan dari sebuah genre dan genre yang merupakan identitas suatu daerah. Contohnya yaitu genre heavy-metal yang merupakan turunan dari metal, hard-rock yang merupakan turunan dari rock dan turkish yang merupakan ciri khas dari lagu yang berasal dari negara Turki. Maka dari itu untuk mengurangi kompleksitas dari model, data yang memiliki nilai genre tersebut dihapus dan hanya menyisakan data yang memiliki genre umum. Adapun genre yang akan digunakan pada tahap klasifikasi ditunjukan pada Gambar 5.

```
['classical' 'country' 'dance' 'disco' 'electronic' 'hip-hop' 'house'
 'jazz' 'metal' 'pop' 'r-n-b' 'rock']
```

Gambar 5 Hasil *class-aware data cleaning*

Setelah dilakukannya *class-aware data cleaning* jumlah dari data yang akan digunakan untuk proses klasifikasi menjadi 1.200 data dengan masing – masing genre memiliki 1.000 data yang dapat dilihat dari Gambar 6.



Gambar 6 Hasil *cleaning data*

2.2. *Splitting data*

Splitting data merupakan proses memisahkan data menjadi data *training*, *validation* dan *testing*. Data dibagi menjadi data *train full* dan *test* dengan perbandingan 80% untuk data *train full* serta 20% untuk data *test*. Data *test* diambil dari *dataset* sebanyak 2.400 data. Data *train full* digunakan untuk uji validasi model yang telah dibuat. Selain itu data *train full* digunakan untuk membagi data

train dan data *validation* dengan perbandingan 75% : 25%. Jumlah dari data *training* yaitu 7.200 data dan data *validation* 2.400 data. Hasil dari *splitting data* dapat dilihat dari Tabel 1.

Tabel 1 Hasil splitting data

Genre	***Train***	***Validation***	***Testing***
House	621	201	178
Metal	617	183	200
Hip–hop	615	198	187
R–n–b	613	183	204
Jazz	608	217	175
Disco	602	204	194
Country	601	209	190
Rock	601	199	200
Classical	592	185	223
Pop	587	207	206
Dance	587	203	210
Electronic	556	211	233
Jumlah	**7.200**	**2.400**	**2.400**

2.3. Label encoder

Label Encoder merupakan proses untuk mengubah label yang tadinya berbentuk kategorikal menjadi numerik. Pada saat proses klasifikasi, algoritma dari *data mining* membutuhkan input dalam bentuk numerik. Namun pada penelitian ini, kelas yang terdapat pada dataset memiliki nilai kategorikal. Oleh karena itu, dibutuhkannya proses *Label Encoder*. Proses *Label Encoder* dilakukan menggunakan *library* dari Python yaitu Scikit-learn. Hasil dari *Label Encoder* dapat dilihat pada Tabel 2.

Tabel 2 Hasil label encoder

Genre	**Hasil *Label Encoder***
Classical	0
Country	1
Dance	2
Disco	3
Electronic	4
Hip-hop	5
House	6
Jazz	7
Metal	8
Pop	9
R-N-B	10
Rock	11

2.4. Feature Transformation

Feature Transformation merupakan proses mempersiapkan fitur – fitur dari *dataset* untuk digunakan pada saat proses pembelajaran mesin. Beberapa algoritma data mining memerlukan input berbentuk matriks bukan *dataframe*. Pada penelitian ini, fitur – fitur pada *dataframe* diubah menjadi representasi matriks menggunakan DictVectorizer. DictVectorizer merupakan sebuah transformer yang terdapat dalam library scikit-learn yang berfungsi untuk mengubah objek *dictionary* atau *dataframe* pandas menjadi matriks fitur yang digunakan untuk melatih model pembelajaran mesin. Cara kerja DictVectorizer adalah dengan mengonversi setiap *dictionary* atau baris dalam *dataframe* menjadi vektor fitur di mana setiap fitur direpresentasikan oleh pasangan (fitur, nilai). Nilai-nilai kategorikal kemudian dikodekan menjadi nilai numerik menggunakan beberapa metode *encoding* seperti *one-hot encoding* atau *binary encoding* yang dapat dilihat pada Tabel 3.

Tabel 3 Output dictvectorizer dan pembentukan input

Input	Output dari DictVectorizer dan pembentukan input data
Data *training*	[[3.20000e-01 6.79000e-01 2.46333e+05 ... 1.16269e+02 4.00000e+00 6.79000e-01] [9.49000e-01 8.71000e-02 5.86080e+05 ... 8.60150e+01 4.00000e+00 3.46000e-02] [3.50000e-03 6.51000e-01 1.78420e+05 ... 1.09019e+02 4.00000e+004.14000e-01] ... [5.12000e-02 6.92000e-01 1.84390e+05 ... 1.22984e+02 4.00000e+00 6.14000e-01][1.86000e-02 5.77000e-01 1.80150e+05 ... 1.60036e+02 4.00000e+00 5.43000e-01] [7.20000e-01 7.74000e-01 2.10466e+05 ... 1.03464e+02 4.00000e+004.21000e-01]]
Data validation	[[1.77000e-03 3.27000e-01 3.92600e+05 ... 1.25232e+02 4.00000e+00 2.18000e-01] [5.84000e-01 5.06000e-01 2.16419e+05 ... 1.39912e+02 4.00000e+00 2.21000e-01] [2.90000e-02 7.10000e-01 3.07014e+05 ... 1.40994e+02 4.00000e+00 7.84000e-01] ... [5.13000e-01 4.29000e-01 2.90400e+05 ... 1.49966e+02 3.00000e+00 7.43000e-01] [7.20000e-02 6.59000e-01 2.56760e+05 ... 1.22013e+02 4.00000e+00 4.54000e-01] [7.40000e-04 4.49000e-01 1.67450e+05 ... 1.41900e+02 4.00000e+00 6.94000e-01]]

Data *testing*	[[7.33000e-01 5.79000e-01 1.31733e+05 ... 5.13000e-02 4.00000e+00 8.36000e-01] [2.23000e-04 5.70000e-01 2.74480e+05 ... 2.78000e-02 4.00000e+004.76000e-01] [2.32000e-01 4.69000e-01 2.83000e+05 ... 3.26000e-02 4.00000e+00 5.29000e-01] ... [7.89000e-02 8.12000e-01 1.92477e+05 ... 5.54000e-02 4.00000e+00 5.73000e-01] [5.55000e-01 3.69000e-01 2.28013e+05 ... 4.00000e-02 3.00000e+00 1.48000e-01] [3.69000e-02 6.53000e-01 2.16281e+05 ... 6.85000e-02 4.00000e+006.69000e-01]]

3. Hasil Pemilihan Fitur (*Feature Selection*)

Feature Selection digunakan untuk memilih fitur – fitur yang relevan dengan target prediksi secara individual. Pemilihan fitur dilakukan dengan cara menghitung statistik untuk setiap fitur secara terpisah dan memilih fitur-fitur yang paling signifikan berdasarkan statistik tersebut (Al-Tashi et al., 2020). Dengan pendekatan ini, hanya fitur-fitur yang memiliki pengaruh besar terhadap target yang dipertahankan untuk model.

Pemilihan fitur diperoleh dengan menggunakan SelectKBest dari modul sklearn.feature_selection. Karena fitur – fitur pada *dataset* bersifat numerik dan target bersifat kategorik, maka nilai statistik yang digunakan yaitu ANOVA F-value. Dalam kode tersebut, selector.fit_transform (X_train, y_train) digunakan untuk menghitung nilai statistik (F-value) untuk setiap fitur dalam X_train dan memilih 10 fitur terbaik berdasarkan F-value tersebut. Nilai statistik setiap fitur dari pemilihan fitur dapat dilihat pada Tabel 4. Dari nilai statistik tersebut, dapat diperoleh 10 fitur terbaik yang akan digunakan dalam proses klasifikasi yaitu *acousticness, loudness, energy, instrumentalness, danceability, speechiness, valance, mode, duration_ms* dan *time_signature.*

Tabel 4 Nilai statistik pemilihan fitur

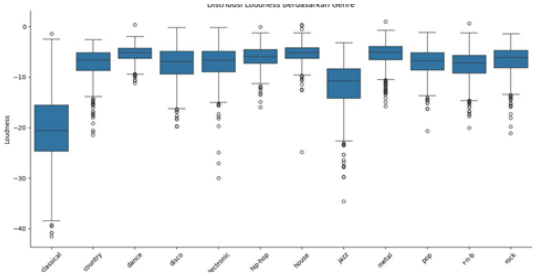
Feature	*Score*
acousticness	874.2342442541395
Loudness	864.6219462737918
energy	651.8890200909747
instrumentalness	482.6167773900606
danceability	407.9110070929281
speechiness	81.68841348259006
valance	79.25318043285179
mode	54.82117296466751
duration_ms	31.054608423734535
time_signature	27.404249690461114
tempo	27.188882675720464
liveness	22.22262916531269
key	4.7548938946093395

4. Visualisasi Data

Untuk mendalami hubungan antara genre musik dan fitur-fitur audio, dilakukan analisis visual menggunakan *heatmap*. *Heatmap* menyajikan representasi visual dari nilai rata-rata fitur audio berdasarkan genre, sehingga memudahkan identifikasi pola serta perbedaan signifikan antar genre musik seperti pada Gambar 7. Dalam *heatmap*, warna yang lebih terang atau lebih gelap (atau dalam beberapa kasus, warna yang berbeda sepenuhnya) menunjukkan nilai yang lebih tinggi atau lebih rendah (Nivethithaa & Vijayalakshmi, 2021).

Gambar 7 *Heatmap* rata - rata fitur berdasarkan genre

Heatmap yang terdapat pada Gambar 4.11 menunjukan kecenderungan nilai rata - rata pada fitur yang memiliki nilai *float* terhadap genre. Semakin gelap warna yang dimiliki maka semakin tinggi juga nilai rata – rata dari fitur tersebut. Sebagai contoh genre Classical memiliki nilai rata – rata pada fitur *acousticness* yang lebih tinggi dibandingkan dengan genre lain dan genre Metal memiliki nilai *energy* yang lebih tinggi dari genre lain. Selain itu, distribusi fitur *loudness* memiliki perbedaan nilai yang signifikan yang dapat dilihat pada Gambar 8. Genre Classical memiliki distribusi *loudnes* yang rendah diantara genre lain dengan nilai antara -25 – (-15).



Gambar 8 Distribusi *loudness* berdasarkan genre

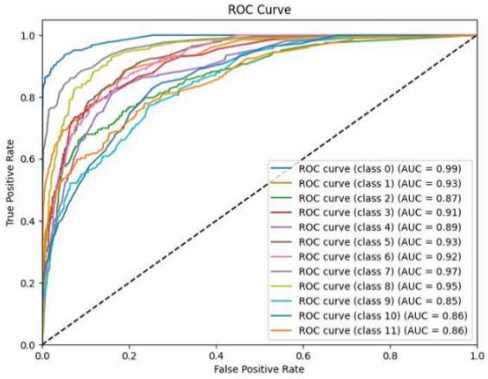
5. Hasil *Mining* Data

Hasil *mining* diperoleh dengan melakukan eksperimen pada algoritma Random Forest tanpa optimalisasi dan menggunakan *Hyperparameter Tuning* menggunakan Grid Search. Kemudian akan diimplementasikan pada *dataset* Spotify Tracks Dataset yang berisi kumpulan fitur audio dari setiap lagu.

5.1. Random Forest Tanpa Optimalisasi

Percobaan yang pertama diimplementasikan pada algoritma Random Forest tanpa menggunakan *Hyperparameter Tuning*. Data yang digunakan pada proses klasifikasi menggunakan 7.200 data untuk *training* dan 2.400 data untuk *testing*. Pada saat sebelum optimalisasi, nilai dari setiap parameter dari algoritma Random Forest diatur secara default.

Hasil dari klasifikasi tersebut diukur menggunakan nilai ROC AUC. ROC merupakan singkatan dari "*Receiver Operating Characteristic*". Sedangkan AUC merupakan singkatan dari "*Area Under the Curve*". Jadi nilai ROC AUC mengukur luas dibawah kurva ROC. Kurva ROC adalah grafik yang menunjukan kinerja model klasifikasi pada berbagai ambang batas. Kurva ROC dibuat dengan memplot TPR (*True Positive Rate*) melawan FPR (*False Positive Rate*) pada berbagai ambang batas klasifikasi yang dapat dilihat pada Gambar 9.



Gambar 9 Kurva ROC sebelum optimalisasi

Berdasarkan kurva ROC tersebut, nilai ROC AUC diperoleh berdasarkan nilai rata – rata dari nilai AUC setiap kelas. Selain itu, nilai ROC AUC juga dapat diperoleh menggunakan modul roc_auc_score dari *library* scikit-learn. Nilai ROC AUC dari hasil klasifikasi tersebut yaitu 0.909.

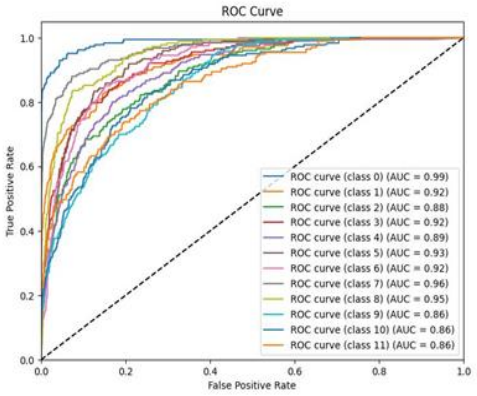
5.2. Penerapan *Hyperparameter Tuning*

Eksperimen yang kedua diterapkan pada algoritma Random Forest dengan optimalisasi menggunakan *Hyperparameter Tuning*. *Hyperparameter Tuning* dilakukan menggunakan Grid Search. Proses ini mengeksplorasi berbagai kombinasi parameter dalam ruang pencarian yang ditentukan untuk mengidentifikasi set parameter optimal berdasarkan performa model (Matin, 2023). Yang ditunjukan pada Tabel 5.

Tabel 5 Hasil hyperparameter tuning

Hyperparameter	*Value / Range*	*Optimal Hyperparameter*
max_depth	5, 10, 15, 20	15
min_samples_leaf	1, 3, 5, 10	1
n_estimators	10 - 200	160

Selama proses klasifikasi nilai dari setiap parameter yang digunakan yaitu hasil dari *Hyperparameter Tuning*. Selain itu model dievaluasi menggunakan metrik evaluasi ROC AUC. Kurva ROC dapat dilihat pada Gambar 10.



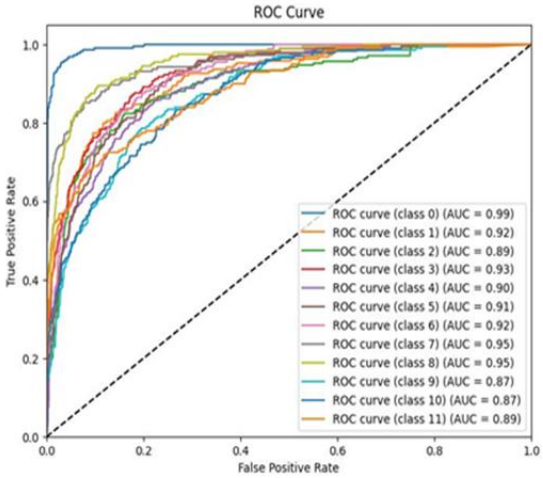
Gambar 10 Kurva ROC setelah optimalisasi

Berdasarkan kurva ROC pada Gambar 4.15, nilai ROC AUC dari model Random

Forest setelah dilakukannya Hyperparameter Tuning yaitu sebesar 0.913.

5.3. Percobaan Menggunakan Data *Train Full*

Percobaan selanjutnya diterapkan pada algoritma Random Forest dengan optimalisasi menggunakan *Hyperparameter Tuning* pada data *train full*. Data tersebut diperoleh dari pembagian *dataset* dengan data *testing*. Perbandingan dari pembagian data tersebut yaitu sebesar 80% untuk data *train full* dan 20% untuk data *test*. Setelah itu, model dievaluasi menggunakan metrik evaluasi ROC AUC dengan kurva ROC seperti pada Gambar 11.



Gambar 11 Kurva ROC Menggunakan Train Full

Berdasarkan kurva ROC pada Gambar 11, nilai ROC AUC dari model Random Forest yaitu sebesar 0.917 dan termasuk kedalam kategori *excellent* (Corbacioglu & Aksel, 2023).

KESIMPULAN

Penelitian ini menggunakan algoritma Random Forest untuk mengelompokan genre musik berdasarkan ciri-ciri yang diambil dari data musik. Manfaat penelitian ini yaitu untuk memberikan pemahaman mengenai peran teknik *Feature Selection* dan *Hyperparameter Tuning* dalam mengoptimalkan performa algoritma Random Forest. Dengan memanfaatkan algoritma secara maksimal, akurasi klasifikasi genre musik dapat ditingkatkan, yang berperan penting dalam menciptakan sistem rekomendasi musik yang lebih tepat dan akurat. Dengan menerapkan strategi seleksi fitur, penelitian ini berhasil menemukan subset fitur yang paling relevan dan informatif untuk memisahkan genre musik. Selain itu, dengan melakukan *Hyperparameter Tuning* penelitian ini berupaya menemukan kombinasi parameter yang optimal untuk Random Forest, termasuk jumlah pohon (n_estimators), kedalaman maksimum pohon (max_depth), dan ukuran minimum sampel di *leaf node* (min_samples_leaf).

Nilai ROC AUC dari model dengan *Hyperparameter Tuning* sebesar 0.913. Berdasarkan hasil evaluasi menggunakan metrik ROC AUC yang diperoleh dari hasil klasifikasi menggunakan algoritma Random Forest tanpa optimalisasi didapatkan sebesar 0.909. Kemudian diterapkan metode Grid Search untuk *Hyperparameter Tuning* diperoleh nilai ROC AUC sebesar 0.913. Selain itu, pada saat proses validasi model diperoleh nilai ROC AUC sebesar 0.917. Dari uraian tersebut, dapat disimpulkan bahwa kombinasi model Random Forest dengan *Feature Selection* dan *Hyperparameter Tuning* menunjukan peningkatan dalam nilai ROC AUC dibandingkan dengan model Random Forest tanpa optimalisasi dan termasuk dalam kategori *excellent*.

Grid Search digunakan untuk mencari *hyperparameter* optimal pada model Random Forest. Namun proses ini terbatas pada ruang *hyperparameter* yang telah ditetapkan sebelumnya yang mungkin tidak mencakup semua kombinasi *hyperparameter* yang paling optimal. Selain itu Grid Search kurang dapat menyesuaikan diri dengan tugas tuning yang rumit atau ketika ada ketergantungan antara *hyperparameter*.

REFERENSI

Al-Tashi, Q., Abdulkadir, S. J., Rais, H. M., Mirjalili, S., & Alhussian, H. (2020). Approaches to Multi-Objective Feature Selection: A Systematic Literature Review. *IEEE Access*, *8*, 125076–125096. https://doi.org/10.1109/ACCESS.2020.3007291

Avci, C., Budak, M., Yagmur, N., & Balcik, F. (2023). Comparison between random forest and support vector machine algorithms for LULC classification. *International Journal of Engineering and Geosciences*, *8*(1), 1–10. https://doi.org/10.26833/ijeg.987605

Azmi, B. N., Hermawan, A., & Avianto, D. (2023). Analisis Pengaruh Komposisi Data Training dan Data Testing pada Penggunaan PCA dan Algoritma Decision Tree untuk Klasifikasi Penderita Penyakit Liver. *JTIM : Jurnal Teknologi Informasi Dan Multimedia*, *4*(4), 281–290.

Belete, D. M., & Huchaiah, M. D. (2022). Grid search in hyperparameter optimization of machine learning models for prediction of HIV/AIDS test results. *International Journal of Computers and Applications*, *44*(9), 875–886. https://doi.org/10.1080/1206212X.2021.1974663

Corbacioglu, Ş., & Aksel, G. (2023). Receiver operating characteristic curve analysis in diagnostic accuracy studies: A guide to interpreting the area under the curve value. *Turkish Journal of Emergency Medicine*, *23*(4), 195.

Ghosh, P., Mahapatra, S., Jana, S., & Kr. Jha, R. (2023). A Study on Music Genre Classification using Machine Learning. *International Journal of Engineering Business and Social Science*, *1*(04), 308–320. https://doi.org/10.58451/ijebss.v1i04.55

Khan, F., Tarimer, I., Alwageed, H. S., Karadağ, B. C., Fayaz, M., Abdusalomov, A. B., & Cho, Y.-I. (2022). Effect of Feature Selection on the Accuracy of Music Popularity Classification Using Machine Learning Algorithms. *Electronics*, *11*(21), 3518. https://doi.org/10.3390/electronics11213518

Matin, I. M. M. (2023). Hyperparameter Tuning Menggunakan GridsearchCV pada Random Forest untuk Deteksi Malware. *MULTINETICS*, *9*(1), 43–50. https://doi.org/10.32722/multinetics.v9i1.5578

Ma, Z., Cui, S., & Joe, I. (2022). An Enhanced Proximal Policy Optimization-Based Reinforcement Learning Method with Random Forest for Hyperparameter Optimization. *Applied Sciences*, *12*(14), 7006. https://doi.org/10.3390/app12147006

Navisa, S., Hakim, L., & Nabilah, A. (2021). Komparasi Algoritma Klasifikasi Genre Musik pada Spotify Menggunakan CRISP-DM. *Jurnal Sistem Cerdas*, *4*(2), 114–125. https://doi.org/10.37396/jsc.v4i2.162

Nivethithaa, K. K., & Vijayalakshmi, S. (2021). Survey on Data Mining Techniques, Process and Algorithms. *Journal of Physics: Conference Series*, *1947*(1), 012052.

Setiadi, D. R. I. M., Rahardwika, D. S., Rachmawanto, E. H., Sari, C. A., Susanto, A., Mulyono, I. U. W., Astuti, E. Z., & Fahmi, A. (2020). Effect of Feature Selection on The Accuracy of Music Genre Classification using SVM Classifier. *2020 International Seminar on Application for Technology of Information and Communication (ISemantic)*, 7–11. https://doi.org/10.1109/iSemantic50169.2020.9234222

Singhal, R., Srivatsan, S., & Panda, P. (2022). Classification of Music Genres using Feature Selection and Hyperparameter Tuning. *Journal of Artificial Intelligence and Capsule Networks*, *4*(3), 167–178. https://doi.org/10.36548/jaicn.2022.3.003

Tanujaya, L. B. C., Susanto, B., & Saragih, A. (2020). The Comparison of Logistic Regression Methods and Random Forest for Spotify Audio Mode Featurre Classification. *Indonesian Journal of Data and Science*, *1*(3). https://doi.org/10.33096/ijodas.v1i3.16

We Are Social. (2022). *Presentase Pengguna Streaming Musik Di Indonesia*. Https://Dataindonesia.Id/Internet/Detail/503-Warga-Ri-Gunakan-Streaming-Musik-Pada-Kuartal-Iii2022.

Yang, Z., Xu, Q., Bao, S., Cao, X., & Huang, Q. (2022). Learning With Multiclass AUC: Theory and Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(11), 7747–7763. https://doi.org/10.1109/TPAMI.2021.3101125