BAB 4
HASIL PENELITIAN

4.1 RINGKASAN HASIL PENELITIAN

Penelitian ini berfokus pada penerapan *Feature Selection* dan *Hyperparameter Tuning* pada algoritma Random Forest untuk klasifikasi genre musik. *Dataset* yang digunakan untuk proses klasifikikasi menggunakan Spotify Tracks Dataset yang berisi fitur audio dari setiap lagu. *Feature Selection* (pemilihan fitur) dilakukan untuk meningkatkan kinerja, mengurangi overfitting, mengurangi waktu pelatihan, meningkatkan interpretabilitas model, dan mengatasi masalah dimensionalitas. Setelah dilakukannya *preprocessing* dan *Feature Selection,* atribut yang digunakan pada proses klasifikasi dapat dilihat pada Tabel 4.1.

Tabel 4.1 Atribut yang digunakan Klasifikasi

Atribut	Tipe Data
acousticness	float64
loudness	float64
energy	float64
instrumentalness	float64
danceability	float64
speechiness	float64
valence	float64
mode	int64
duration_ms	int64
time_signatue	int64

Selain itu, penerapan *Hyperparameter Tuning* dilakukan untuk menemukan set nilai *hyperparameter* yang paling baik dalam meningkatkan performa model. Metode yang digunakan untuk *Hyperparameter Tuning* yaitu Grid Search. Nilai dari parameter yang digunakan pada proses klasifikasi setelah dilakukannya optimalisasi dapat dilihat pada Tabel 4.2

Tabel 4.2 Nilai Parameter Random Forest Setelah Optimalisasi

Parameter	Nilai
max_depth	14
min_samples_leaf	1
n_estimators	125

Untuk menilai kinerja model menggunakan metrik evaluasi ROC AUC. ROC AUC Score diperoleh dari rata – rata nilai AUC yang dimiliki oleh setiap kelas. Perbandingan nilai ROC AUC dari model sebelum dan sesudah optimalisasi dapat dilihat pada Tabel 4.3.

Tabel 4.3 Perbandingan Nilai ROC AUC

Metode	Nilai ROC AUC
Random Forest	0.909
Random Forest + *Hyperparameter Tuning*	0.913
Percobaan Pada Data *Train Full*	0.917

Selisih nilai ROC AUC dari model pada saat sebelum dan sesudah dilakukannya optimalisasi pada metode Random Forest yaitu sebesar 0.004. Sedangkan perbedaan nilai ROC AUC menggunakan data *train full* sebesar 0.004. Menurut Tabel 2.2, nilai ROC AUC yang dimiliki oleh model masuk kedalam kategori *excellent*.

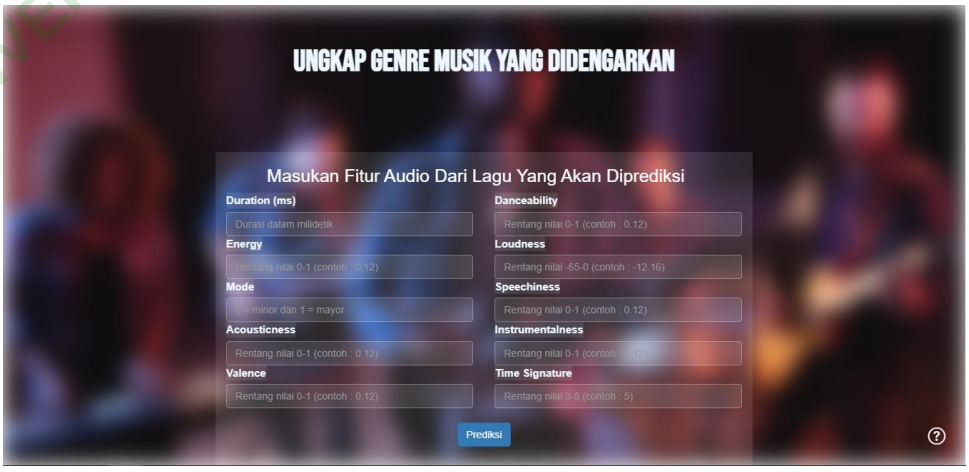
4.2 IMPLEMENTASI SISTEM

Sistem pada studi ini merupakan hasil akhir dari pengolahan serta pengujian data. Dalam implementasinya, penelitian ini memanfaatkan bahasa pemrograman Python dengan framework Flask. Arsitektur sistem terbagi menjadi dua komponen, yakni client dan server. Server bertanggung jawab sebagai penyedia layanan dalam aplikasi, sedangkan client bertugas mengirimkan permintaan dan menerima respons dari server. Komunikasi antara client dan server dilakukan melalui REST API, di mana client mengirimkan permintaan pada server melalui URL yang disediakan, lalu server memberikan respons kepada client berdasarkan permintaan yang

diterima. Flask dipilih karena merupakan micro framework Python yang cocok untuk arsitektur REST API. Ilustrasi arsitektur yang digunakan dalam penelitian ini ditampilkan pada Gambar 4.1.

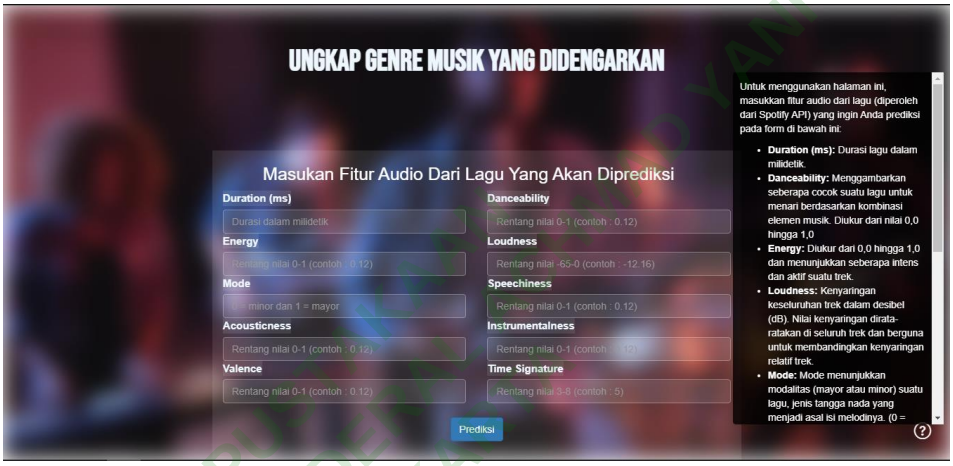
Gambar 4.1 Ilustrasi REST API

Sisi klien memiliki tanggung jawab untuk mengoptimalkan pengalaman pengguna dengan memastikan bahwa antarmuka aplikasi dirancang dengan baik dan mudah dipahami. Ini penting agar pengguna dapat dengan mudah berinteraksi dengan aplikasi dan mencapai tujuan tanpa hambatan. Dalam penelitian ini, aplikasi dikembangkan sebagai single-page application (SPA) sederhana karena fungsinya hanya untuk memprediksi genre musik dengan memasukkan fitur audio dari sebuah lagu. Oleh karena itu, desain antarmuka pengguna (UI) yang disajikan perlu sederhana dan mudah dipahami. Gambar 4.2 menunjukkan antarmuka pengguna aplikasi dalam penelitian ini.



Gambar 4.2 *Landing Page* Sistem

Komponen utama dalam antarmuka pengguna adalah *form input* yang digunakan untuk melakukan prediksi genre musik. *Form* ini memungkinkan pengguna untuk memasukkan fitur audio dari sebuah lagu yang akan digunakan untuk prediksi genre musik. Fitur audio dari sebuah lagu dapat dilihat dari Spotify API dan setiap form memiliki aturan pengisian yang dapat dilihat pada tombol *help* (?) yang terletak di pojok kanan bawah pada halaman sistem yang dapat dilihat pada Gambar 4.3.



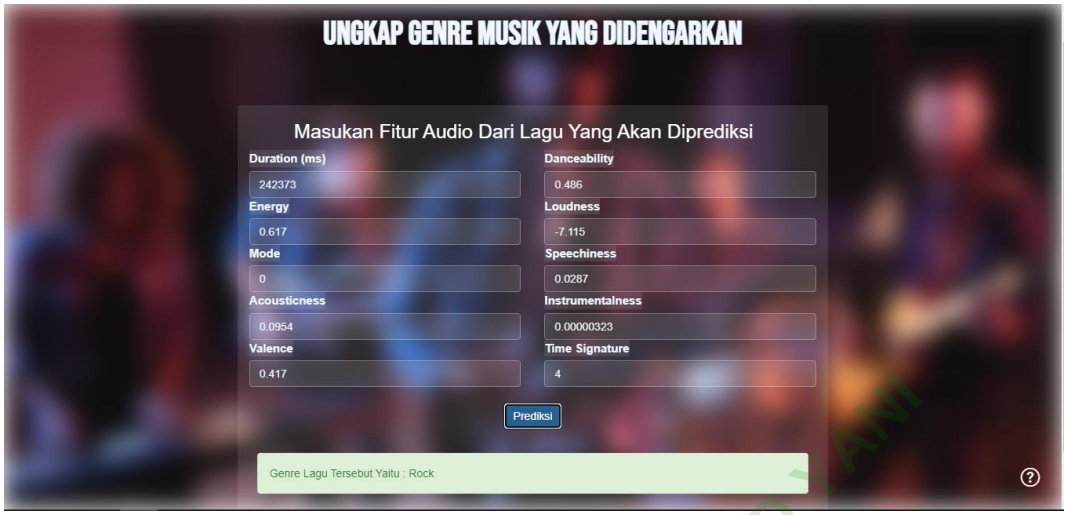
Gambar 4.3 Petunjuk Penggunaan Sistem

Untuk memulai proses analisis, pengguna harus memasukan fitur audio kedalam form yang dapat diperoleh dari Spotify API. Contoh data yang diperoleh dapat dilihat pada Gambar 4.4.

artist	song	duration_ms	danceability	energy	loudness	mode	speechiness	acousticness	instrumentalness	valence	time_signature	genre
Bon Jovi	It's My Life	224493	0.551	0.913	-4.063	0	0.0466	0.0263	1.35E-05	0.544	4	rock, metal
blink-182	All The Small Things	167066	0.434	0.897	-4.918	1	0.0488	0.0103	0	0.684	4	rock, pop
Destiny's Child	Say My Name	271333	0.713	0.678	-3.525	0	0.102	0.273	0	0.734	4	pop, R&B
LeAnn Rimes	I Need You	229826	0.478	0.736	-7.124	1	0.0367	0.02	9.58E-05	0.564	4	pop, country
Next	Wifey	243666	0.829	0.652	-8.693	0	0.108	0.067	0	0.726	4	hip hop, pop, R&B
3 Doors Down	Kryptonite	233933	0.545	0.865	-5.708	0	0.0286	0.00664	1.10E-05	0.543	4	pop, rock, metal
Steps	It's the Way You Make Me Feel	197360	0.573	0.665	-5.081	1	0.0239	0.108	2.09E-06	0.347	4	pop, Dance/Electronic
Katie Melua	The Closest Thing to Crazy	252466	0.562	0.219	-13.2	1	0.0312	0.856	0.000296	0.106	4	pop, easy listening, jazz
M83	Midnight City	243960	0.507	0.729	-5.399	0	0.0393	0.0182	1.40E-06	0.272	4	rock, pop, metal, Dance/Electronic
Coldplay	Viva La Vida	242373	0.486	0.617	-7.115	0	0.0287	0.0954	3.23E-06	0.417	4	rock, pop

Gambar 4.4 *Dataset* Pengujian Sistem

Dari data pada Gambar 4.4, terdapat lagu yang memiliki beberapa genre seperti lagu dari Coldplay yang berjudul “Viva La Vida”memiliki genre rock dan pop. Untuk mencegah ambiguitas genre, sistem ini dapat menentukan genre yang lebih dominan dari lagu tersebut seperti pada Gambar 4.5.



Gambar 4.5 *Output* Prediksi Genre Musik

Hasil prediksi dari sistem menunjukan bahwa lagu dari Coldplay yang memiliki judul "Viva La Vida" memiliki genre Rock. Setelah proses selesai, sistem akan memberikan hasil prediksi yang jelas dan informatif mengenai genre dari sebuah lagu yang telah diprediksi.

4.3 PEMBAHASAN

4.3.1 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 4.6.

	Unnamed: 0	track_id	artists	album_name	track_name	popularity	duration_ms	explicit	danceability	energy	...	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	time_signature	track_genre
0	0	5SuOikwiRyPMVoIQDJUgSV	Gen Hoshino	Comedy	Comedy	73	230666	False	0.676	0.461	...	-6.746	0	0.1430	0.0322	0.000001	0.3580	0.7150	87.917	4	acoustic
1	1	4qPNDBW1i3p13qLCt0Ki3A	Ben Woodward	Ghost (Acoustic)	Ghost - Acoustic	55	149610	False	0.420	0.166	...	-17.235	1	0.0763	0.9240	0.000006	0.1010	0.2670	77.489	4	acoustic
2	2	1UBSr7s7jYXzAREGcdKSb	Ingrid Michaelson;ZAYN	To Begin Again	To Begin Again	57	210826	False	0.438	0.3590	...	-9.734	1	0.0557	0.2100	0.000000	0.1170	0.1200	76.332	4	acoustic
3	3	6fhq3CG4xtTEg7oqvCyx	Kina Grannis	Crazy Rich Asians (Original Motion Picture Sou...	Can't Help Falling In Love	71	201933	False	0.266	0.0596	...	-18.515	1	0.0363	0.9050	0.000071	0.1320	0.1430	181.740	3	acoustic
4	4	5vjLSffimIP26QG5WcN2K	Chord Overstreet	Hold On	Hold On	82	198853	False	0.618	0.443	...	-9.681	1	0.0526	0.4690	0.000000	0.0829	0.1670	119.949	4	acoustic
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
113995	113995	2C3TZjDRiAzdyViavDJ217	Rainy Lullaby	#mindfulness - Soft Rain for Mindful Meditatio...	Sleep My Little Boy	21	384999	False	0.172	0.2350	...	-16.393	1	0.0422	0.6400	0.928000	0.0863	0.0339	125.995	5	world-music
113996	113996	1hIzfL4IBtNGWRYPOCGPw	Rainy Lullaby	#mindfulness - Soft Rain for Mindful Meditatio...	Water Into Light	22	385000	False	0.174	0.1170	...	-18.318	0	0.0401	0.9940	0.976000	0.1050	0.0350	85.239	4	world-music
113997	113997	6xDzfSoqDjuNsISVPSQyX	Cesária Evora	Best Of	Miss Perfumado	22	271466	False	0.629	0.3290	...	-10.895	0	0.0420	0.8670	0.000000	0.0839	0.7430	132.378	4	world-music
113998	113998	2a6sXL2tIV4hSd1VTdrfLs	Michael W. Smith	Change Your World	Friends	41	283893	False	0.587	0.5060	...	-10.889	1	0.0297	0.3810	0.000000	0.2700	0.4130	135.960	4	world-music
113999	113999	2hETkH7cOfqmz3LqZDHZf5	Cesária Evora	Miss Perfumado	Barbincor	22	241826	False	0.526	0.487	...	-10.204	0	0.0725	0.6810	0.000000	0.0893	0.7080	78.198	4	world-music

114000 rows × 21 columns

Gambar 4.6 *Dataset Preview*

4.3.2 Hasil Pengolahan Data

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

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.

a. *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 4.7.

```
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 4.7 Hasil *Check Missing Value*

Berdasarkan gambar Gambar 4.7 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 music. Hasil setelah *Check Missing Value* dapat dilihat pada Gambar 4.8.

	duration_ms	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	time_signature	track_genre
0	230666	0.676	0.4610	1	-6.746	0	0.1430	0.0322	0.000001	0.3580	0.715	87.917	4	acoustic
1	149610	0.420	0.1660	1	-17.235	1	0.0763	0.9240	0.000006	0.1010	0.267	77.489	4	acoustic
2	210826	0.438	0.3590	0	-9.734	1	0.0557	0.2100	0.000000	0.1170	0.120	76.332	4	acoustic
3	201933	0.266	0.0596	0	-18.515	1	0.0363	0.9050	0.000071	0.1320	0.143	181.740	3	acoustic
4	198853	0.618	0.4430	2	-9.681	1	0.0526	0.4690	0.000000	0.0829	0.167	119.949	4	acoustic

Gambar 4.8 *Dataset Preview* Setelah *Check Missing Value*

b. *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 4.9.

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

Gambar 4.9 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 4.10.

Gambar 4.10 Hasil *Cleaning* Data

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 *val* 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 4.4.

Tabel 4.4 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

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 4.5.

Tabel 4.5 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

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 4.6.

Tabel 4.6 *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]]

4.3.3 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. Dengan pendekatan ini, hanya fitur-fitur yang memiliki pengaruh besar terhadap target yang dipertahankan untuk digunakan dalam model. Kode dari *Feature Selection* dapat dilihat dibawah ini.

```
selector = SelectKBest(score_func=f_classif, k=10)
X_selected = selector.fit_transform(X_train, y_train)

feature_scores = selector.scores_

for feature, score in zip(dv.get_feature_names_out(),
feature_scores):
    print(f"Feature: {feature}, Score: {score}")
```

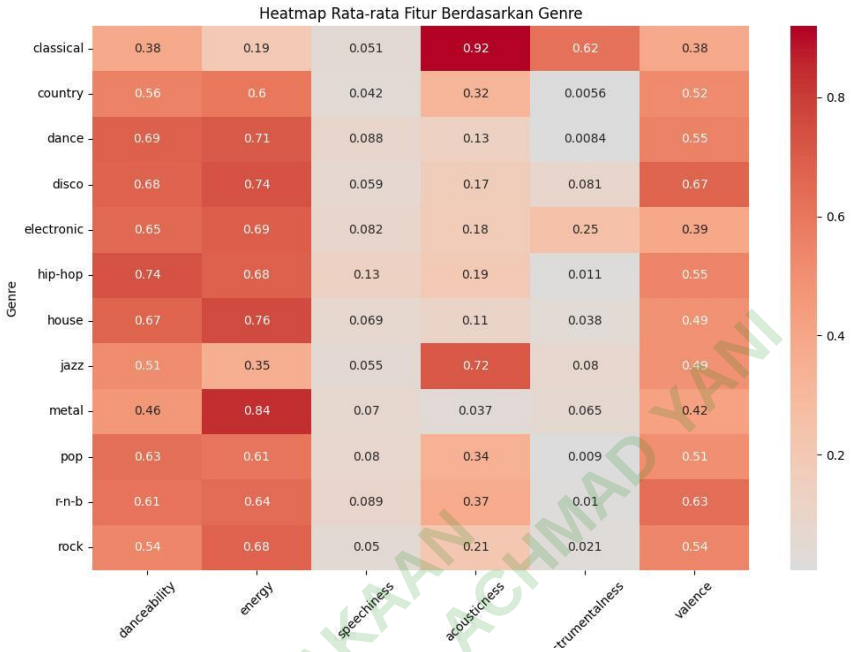
Kode diatas menggunakan SelectKBest dari modul sklearn.feature_selection. Karena fitur – fitur pada dataset bersifat numerik dan target bersifat kategorik, maka nilai statistik yang digunakan yaitu ANOVA F-value. Dalam kode tersebut, selector.fit_transform(X_train, y_train) digunakan untuk menghitung 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 4.7. 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 4.7 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

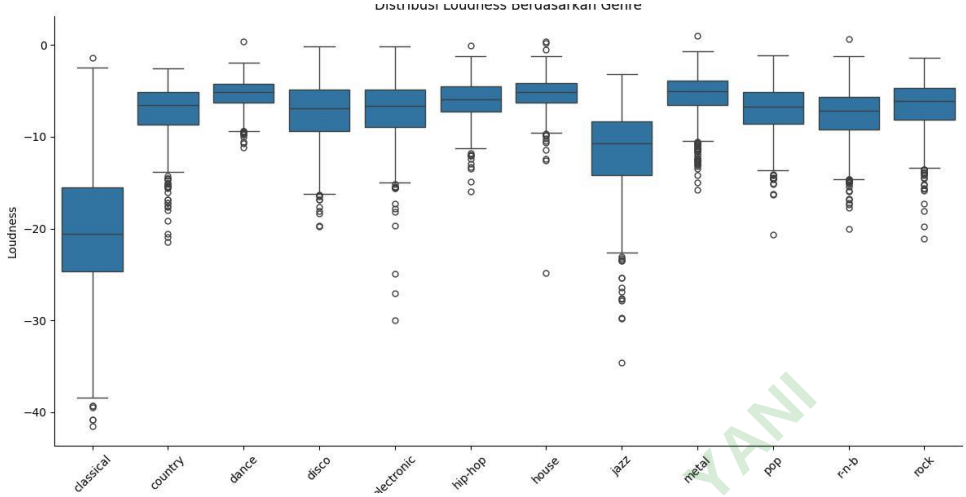
4.3.4 Visualisasi data

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 music seperti pada Gambar 4.11. 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.



Gambar 4.11 *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 4.12. Genre Classical memiliki distribusi loudnes yang rendah diantara genre lain dengan nilai antara -25 – (-15).



Gambar 4.12 Distribusi *Loudnes* Berdasarkan Genre

4.3.5 Hasil *Mining* Data

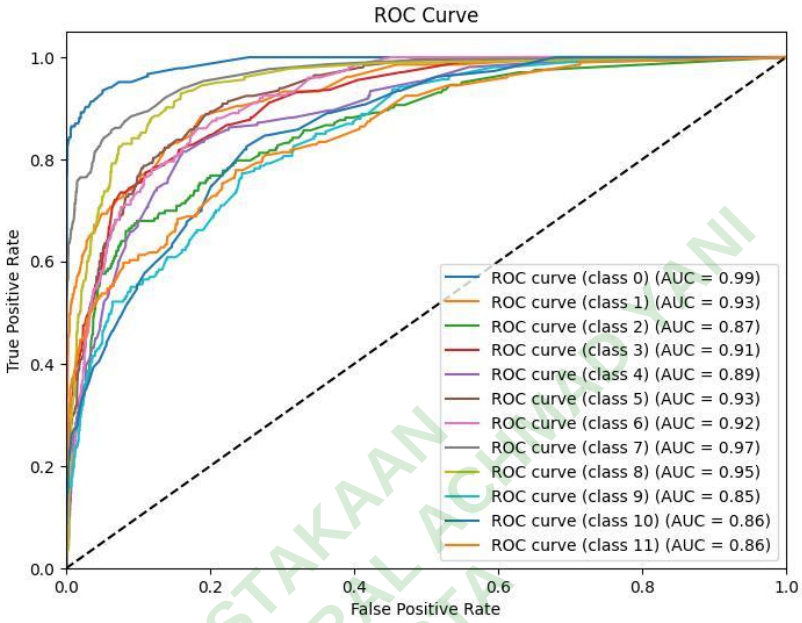
Hasil mining diperoleh dengan melakukan eksperimen pada algoritma Random Forest tanpa optimalisasi dan menggunakan *Hyperparameter Tuning* menggunakan Grid Search. Kemudian akan diimplementasikan pada *dataset* Spotify Tracks Dataset yang berisi kumpulan fitur audio dari setiap lagu.

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 4.13.



Gambar 4.13 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 Score dari hasil klasifikasi tersebut yaitu 0.909.

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 yang ditunjukan pada Tabel 4.8.

Tabel 4.8 Hasil *Hyperparameter Tuning*

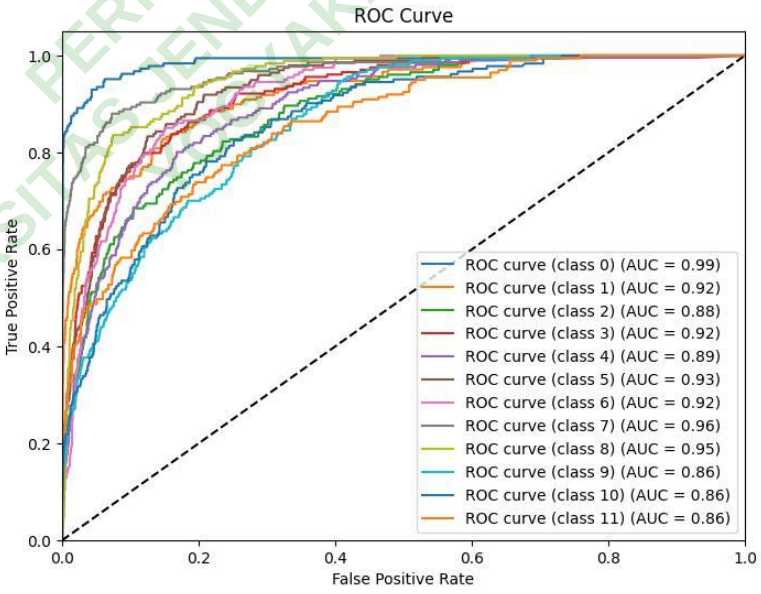
Hyperparameter	*Value / Range*	*Optimal Hyperparameter*
max_depth	5, 10, 15, 20	15
min_samples_leaf	1, 3, 5, 10	1
n_estimators	10 - 200	160

Konfigurasi model Random Forest setelah dilakukannya *Hyperparameter Tuning* dapat dilihat pada Gambar 4.14.

```
RandomForestClassifier
RandomForestClassifier(max_depth=15, n_estimators=125, random_state=1)
```

Gambar 4.14 Konfigurasi Random Forest Setelah Optimalisasi

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 4. 15.



Gambar 4.15 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.

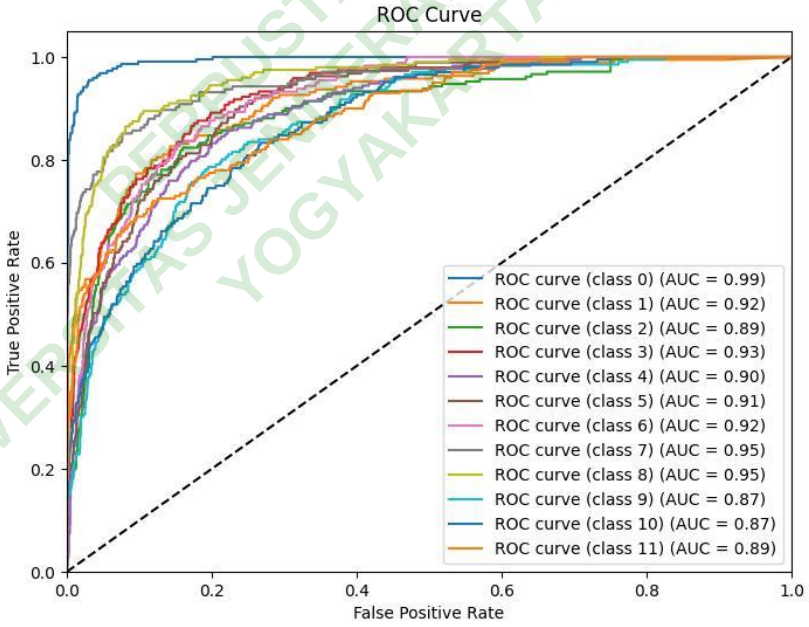
3. Percobaan Menggunakan Data *Train Full*

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*. Nilai yang digunakan pada parameter model Random Forest yang digunakan yaitu hasil dari *Hyperparameter Tuning* dengan konfigurasi seperti Gambar 4.16.

```
RandomForestClassifier
RandomForestClassifier(max_depth=15, n_estimators=125, random_state=1)
```

Gambar 4.16 Konfigurasi Random Forest Untuk Validasi

Setelah itu, model dievaluasi menggunakan metrik evaluasi ROC AUC dengan kurva ROC seperti pada Gambar 4.17.



Gambar 4.17 Kurva ROC Menggunakan *Train Full*

Berdasarkan kurva ROC pada Gambar 4.17, nilai ROC AUC dari model Random Forest yaitu sebesar 0.917. Menurut Tabel 2.2 dengan nilai ROC AUC diatas 0.9 maka model termasuk kedalam kategori *excellent*.