Rekomendasi Musik Pada Youtube Music Menggunakan Metode Decision Tree CART

Ririn Wanandi[1], Dea Rajwa Zahra Athaya[2], Torrilynn Farrell Zuriely[3], Anggraini Puspita Sari[4*]

[1,2,3,4]Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional "Veteran" Jawa Timur
[1]23081010136@student.upnjatim.ac.id, [2] 23081010013@student.upnjatim.ac.id,
[3] 23081010194@student.upnjatim.ac.id, [*4] anggraini.puspita.if@upnjatim.ac.id

Abstrak

Perkembangan teknologi digital telah mengubah cara masyarakat mengakses dan menikmati musik, khususnya melalui platform streaming seperti YouTube Music yang menawarkan ragam konten luas namun menimbulkan tantangan dalam menemukan musik sesuai preferensi pengguna. Tujuan penelitian ini adalah untuk mengembangkan sistem rekomendasi yang mampu memprediksi popularitas lagu secara akurat berdasarkan atribut-atribut utama. Penelitian ini mengimplementasikan algoritma Decision Tree Classification and Regression Tree (CART) untuk mengembangkan sistem rekomendasi musik yang dapat memprediksi popularitas lagu berdasarkan preferensi pengguna. Metode CART dipilih karena kemampuannya menangani data multidimensional dengan interpretabilitas tinggi dan dapat mengelola atribut kategorikal serta numerik secara bersamaan. Kontribusi utama dari penelitian ini adalah penerapan metode CART untuk klasifikasi musik populer yang dapat diintegrasikan secara *real-time* ke dalam sistem rekomendasi web. Dataset yang digunakan mencakup empat fitur utama yaitu performer, genre, mood, dan year sebagai parameter input untuk mengklasifikasikan lagu ke dalam kategori "Hit" atau "Not Hit". Proses penelitian meliputi tahapan pengumpulan data dari sumber terbuka, pra-pemrosesan data dengan pembersihan, transformasi, pelabelan, seleksi fitur, dan pembagian data dengan rasio 80:20, 70:30, dan 60:40 untuk training dan testing. Model Decision Tree CART kemudian diintegrasikan ke dalam aplikasi web berbasis framework Flask dengan antarmuka dropdown yang memungkinkan pengguna menginput empat parameter secara *real-time.* Evaluasi menggunakan metrik precision, recall, F1-score, dan accuracy menunjukkan performa yang sangat baik dengan tingkat akurasi mencapai 98,33%. Hasil ini membuktikan efektivitas metode CART dalam mengidentifikasi pola preferensi musik dan menghasilkan rekomendasi yang akurat, serta menunjukkan potensi implementasi sistem rekomendasi musik yang lebih canggih untuk meningkatkan pengalaman pengguna platform streaming musik.

Kata kunci: decision tree, klasifikasi, machine learning, rekomendasi musik, youtube music

Commented [A1]: Kata kunci ditulis dalam huruf kecil kecuali singkatan.

1. Pendahuluan

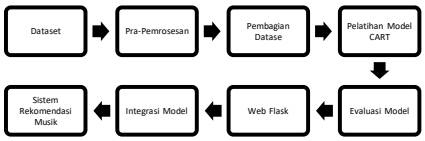
Perkembangan teknologi digital telah mengubah cara masyarakat mengakses musik, dengan YouTube Music menjadi salah satu platform utama penyedia jutaan lagu dari berbagai genre dan era (Fitradhi et al., 2023; Rachman & Voutama, 2024). Kompleksitas pemilihan lagu dari database besar mendorong kebutuhan akan sistem rekomendasi yang cerdas dan personal, karena setiap pengguna memiliki preferensi unik berdasarkan genre, artis, mood, dan periode waktu (Amin et al., 2023; Harahap & Harahap, 2025). Sistem rekomendasi konvensional sering gagal mengelola atribut kategorikal dan numerik secara simultan, sehingga menghasilkan prediksi yang kurang akurat (Hartono & Sutopo, 2024; Madani et al., 2024). Prediksi yang kurang akurat juga dilaporkan pada penelitian lain (Izzah & Widyastuti, 2017). Beberapa penelitian sebelumnya telah mengembangkan sistem rekomendasi musik dengan pendekatan yang beragam, seperti filtering, clustering, dan pemanfaatan NLP. Namun, pendekatan tersebut masih memiliki keterbatasan dalam hal akurasi dan fleksibilitas ketika menghadapi data musik yang bersifat heterogen dan kompleks. Selain itu, masih jarang penelitian yang mengkaji penggunaan metode klasifikasi berbasis decision tree secara khusus untuk prediksi popularitas lagu dengan input fitur gabungan seperti performer, genre, mood, dan year. Hal ini menjadi celah yang ingin diisi dalam penelitian ini.

Decision Tree dengan algoritma CART memiliki keunggulan dalam mengelola data musik yang kompleks dan multidimensional (Asrianda et al., 2024; Azwanti & Putria, 2024). CART mampu membagi dataset berdasarkan atribut paling relevan menggunakan kriteria optimal, serta memiliki interpretabilitas tinggi dan efisiensi komputasi yang baik (Mukhsinin et al., 2024; Mienye & Jere, 2024). Keunggulan ini juga didukung oleh studi lain (Sánchez-Moreno et al., 2020). CART juga mendukung pruning otomatis untuk menghindari overfitting, serta dapat menangani atribut kategorikal seperti performer dan genre maupun atribut numerik seperti year dengan optimal (Fiqri et al., 2025). Selain itu, dalam konteks evaluasi klasifikasi, model Decision Tree telah terbukti memberikan kinerja stabil pada domain lain seperti klasifikasi penyakit jantung, dengan metrik evaluasi seperti precision,

recall, dan F1-score yang menjadi indikator penting dalam memahami efektivitas model terhadap data yang tidak seimbang (Sari et al., 2023). Pemilihan CART dalam penelitian ini didasarkan pada kemampuannya mengolah empat fitur utama dalam dataset YouTube Music, yaitu performer, genre, mood, dan year (Indah Prabawati et al., 2019; Hakim et al., 2023). Struktur if-then yang dihasilkan CART meningkatkan transparansi dan kepuasan pengguna (Harjananto et al., 2021; Maulida et al., 2016). Oleh karena itu, penelitian ini bertujuan untuk membangun sistem rekomendasi musik berbasis web yang mampu memprediksi popularitas lagu dengan pendekatan klasifikasi menggunakan algoritma Decision Tree CART. Metode ini dinilai paling sesuai karena mampu menangani atribut musik yang bersifat heterogen dan memberikan interpretasi model yang mudah dipahami pengguna. Kontribusi utama dari penelitian ini adalah pengembangan prototipe sistem klasifikasi musik dengan akurasi tinggi yang dapat diintegrasikan secara real-time ke dalam antarmuka pengguna. Dibandingkan pendekatan sebelumnya yang berfokus pada filtering atau clustering, penelitian ini mengisi celah dengan menerapkan model klasifikasi berbasis decision tree yang mampu menghasilkan prediksi langsung terhadap status kepopuleran lagu (Hit atau Not Hit) secara lebih presisi dan dapat dijelaskan logikanya.

2. Metode

Metode penelitian menggunakan algoritma *Decision Tree CART* (Classification and Regression Tree) untuk membangun sistem rekomendasi musik berdasarkan preferensi pengguna dengan memanfaatkan dataset yang telah tersedia, dataset ini berisi informasi terkait lagu-lagu di YouTube Music, termasuk fitur seperti genre, artis, durasi, popularitas, dan label "hit" atau tidaknya lagu. Langkah-langkah dalam penelitian ini diawali dengan pengumpulan data lalu *preprocessing* data, seperti pembersihan data, transformasi data, pelabelan data, seleksi fitur, dan pembagian. Model yang telah dilatih kemudian diintegrasikan ke dalam aplikasi web menggunakan framework Flask, di mana pengguna dapat menginput preferensi mereka dan sistem akan menampilkan rekomendasi lagu berdasarkan hasil klasifikasi model CART. Alur keseluruhan penelitian meliputi: pemahaman dataset, preprocessing, pelatihan model CART, evaluasi model, dan pengembangan aplikasi web sebagai media penyajian rekomendasi.



Gambar 1. Diagram alur metode penelitian decision tree.

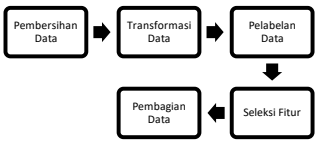
Gambar ini memperlihatkan tahapan penelitian mulai dari pengumpulan data, pra-pemrosesan, pelatihan model, evaluasi, hingga implementasi sistem berbasis web. Alur ini menunjukkan bahwa proses dimulai dari penyediaan dataset, kemudian dilakukan pembersihan, transformasi, pelabelan, seleksi fitur, pembagian data, pelatihan menggunakan algoritma CART, evaluasi performa, dan akhirnya integrasi model ke dalam antarmuka web Flask untuk digunakan oleh pengguna secara real-time.

2.1 Pengumpulan Data

Data dalam penelitian ini diperoleh dari dua sumber utama, yaitu Spotify Dataset dan YouTube Music**,** melalui repositori publik seperti Kaggle dan teknik *web scraping*. Dataset berisi 24.329 lagu**,** disimpan dalam format .csv untuk mempermudah analisis.
Variabel yang digunakan mencakup: *Video ID*, Judul Lagu, *Performer*, *Total Views*, *Year*, *Genre*, *Mood*, serta *Status* lagu sebagai label target (Hit/Not Hit), dengan ambang batas 500 juta views. Selain data sekunder, juga dikumpulkan data primer melalui survei terhadap 100 responden untuk validasi hasil rekomendasi. Proses pengumpulan dilakukan selama Maret–Mei 2024, dengan memperhatikan keseimbangan genre dan tahun rilis guna menghindari bias. Variabel *Performer*, *Genre*, *Mood*, dan *Year* digunakan sebagai fitur input untuk proses klasifikasi.

2.2 Pra-pemrosesan Data

Pra-pemrosesan data merupakan langkah dilakukan agar data siap diolah. Tahapan ini mencakup sejumlah langkah sistematis yang dirancang untuk meningkatkan kualitas data dan efektivitas proses klasifikasi. Diagram alur pra-pemrosesan berikut menggambarkan keseluruhan tahapan yang diterapkan dalam penelitian ini.



Gambar 2. Diagram Alur Pra Pemrosesan

Gambar ini menggambarkan tahapan rinci pra-pemrosesan data yang terdiri dari pembersihan data, transformasi atribut kategorikal dan numerik, pelabelan target, seleksi fitur, serta pembagian dataset. Diagram ini membantu memvisualisasikan proses penyiapan data sebelum dilakukan pelatihan model, sehingga kualitas dan representasi data menjadi optimal untuk klasifikasi.

Commented [A2]: Berikan deskripsi dalam 1 paragraf atau lebih di bawah setiap gambar pada naskah.

Langkah-Langkah yang diperlukan untuk melakukan pra-pemrosesan meliputi:

1. Pembersihan Data (*Data Cleaning*)

Pembersihan data merupakan proses untuk menghapus gangguan (noise) dan data yang tidak diperlukan atau tidak relevan Proses ini mencakup penghapusan data duplikat, pengisian nilai kosong, serta eliminasi karakter atau entri yang tidak relevan. Tujuannya adalah mengurangi noise agar model dapat belajar secara optimal dari data bersih.

2. Transformasi Data (*Data Transformation*)

Atribut kategorikal seperti *genre, mood, dan performer* diubah ke bentuk numerik menggunakan teknik Label Encoding. Sementara itu, atribut numerik seperti year dinormalisasi agar berada pada skala yang seragam dan tidak mendominasi fitur lain dalam proses pelatihan.

3. Pelabelan Data (*Data Labeling*)

Pelabelan data merupakan proses pemberian label pada data yang digunakan sebagai data latihSetiap lagu diberi label "*Hit*" jika terbukti populer (misalnya masuk chart YouTube Music), dan "*Not Hit*" jika tidak. Pelabelan dilakukan secara manual dan semi-otomatis, dengan menggabungkan informasi popularitas dan metadata lagu.

4. Seleksi Fitur (*Feature Selection*)

Seleksi fitur merupakan salah satu teknik paling penting dan umum digunakan dalam tahap pra-pemrosesan. Tujuan utamanya adalah untuk memilih fitur-fitur paling relevan dari sekumpulan fitur yang tersedia dalam data Untuk meningkatkan performa dan menghindari overfitting, dilakukan pemilihan fitur yang paling relevan. Fitur yang dipertahankan meliputi *performer, genre, mood, dan year,* karena secara empiris memiliki pengaruh terhadap preferensi pendengar berdasarkan studi terdahulu.

5. Pembagian Data (*Data Splitting*)

Pembagian Data (*data splitting*) merupakan proses membagi kumpulan data yang memiliki banyak entri ke dalam dua atau lebih subset, dengan tujuan memisahkan data untuk keperluan pelatihan dan pengujian model secara terstruktur Dataset dibagi ke dalam tiga skenario perbandingan rasio pembagian data, yaitu 80:20, 70:30, dan 60:40, guna mengevaluasi performa model pada masing-masing rasio. Hasil dari tiap skenario kemudian dibandingkan untuk menentukan rasio dengan tingkat akurasi tertinggi.

2.3 Perancangan *Wireframe* dan Implementasi Sistem Berbasis Flask

Setelah model klasifikasi selesai dibangun, tahap selanjutnya adalah perancangan antarmuka pengguna berbasis web. Desain awal dilakukan melalui pembuatan wireframe untuk menggambarkan alur dan tampilan sistem. Komponen utama dalam wireframe meliputi:

- *Dropdown* untuk memilih *Performer, Genre, Mood, dan Year*.
- Tombol "*Predict*" untuk memulai proses prediksi.
- Tampilan hasil yang mencakup : Prediksi status lagu *Hit* atau *Not Hit*, Daftar lagu serupa sebagai rekomendasi, dan Nilai *confidence score* dari hasil prediksi.

Gambar 3. Wireframe output berbasis web

Gambar ini menampilkan rancangan awal antarmuka sistem rekomendasi musik berbasis web. Dalam wireframe ditunjukkan letak komponen utama seperti dropdown untuk input performer, genre, mood, dan year, tombol "Predict" untuk memulai prediksi, serta area output untuk menampilkan hasil klasifikasi, daftar rekomendasi lagu serupa, dan confidence score. Setelah *wireframe* selesai, tahap implementasi dilakukan menggunakan framework Flask yang ringan dan fleksibel, serta mendukung integrasi dengan model machine learning berbasis Python. Sistem terdiri dari tiga bagian utama:

- Frontend: Dibangun dengan HTML, CSS, dan JavaScript untuk menampilkan antarmuka sesuai desain wireframe.
- Backend: Menggunakan Flask untuk menerima input pengguna, menjalankan model klasifikasi, dan mengembalikan hasil prediksi secara real-time.
- Model *Machine Learning*: Menggunakan algoritma Decision Tree (CART) yang telah dilatih sebelumnya untuk melakukan klasifikasi berdasarkan input dari pengguna.

3. Hasil dan Pembahasan

3.1 Pengumpulan Data

Pengumpulan data pada penelitian ini ditunjukkan pada Gambar 4, yang mencakup kombinasi dataset dari Spotify dan YouTube Music dengan total 24.329 entri lagu. Dataset tersebut kemudian dipersiapkan melalui tahap pra-pemrosesan untuk memastikan

bahwa data bersih, relevan, dan siap digunakan dalam pelatihan model klasifikasi.

No	youtubeId	itemLabel	performerLabel	youtubeViews	year	genreLabel	mood	hit_label
1	JGwWNGJdvx8	Shape of You	Ed Sheeran	5534218047	2017	short film, dance-pop, musical film	energetic	Hit
2	RgKAFK5djSk	See You Again	Charlie Puth	5016965543	2015	hip hop music, pop rap	energetic	Hit
3	OPf0YbXqDm0	Uptown Funk	Mark Ronson	4362269292	2014	soul, funk, boogie, disco-pop	energetic	Hit
4	lYvFa2ARD24	Uptown Funk	Mark Ronson	4362269292	2014	soul, funk, boogie, disco-pop	energetic	Hit
5	k85mRPqvMbE	Axel F	Crazy Frog	3694049867	2005	dance-pop	energetic	Hit
24329	gKTk_iZT_OA	Fall Into The Sun	Ike Moriz	1	2002	Britpop, pop rock	relaxed	Not a Hit

Total Baris: 24329
Index Baris Terakhir: 24328

Gambar 4. Dataset Musik YouTube

Gambar ini memperlihatkan tampilan awal dataset yang digunakan dalam penelitian, berisi informasi seperti ID video, judul lagu, performer, jumlah views, tahun rilis, genre, mood, serta label target (Hit/Not Hit). Dataset ini menjadi sumber utama untuk pelatihan dan pengujian model CART.

3.2 Pra-pemrosesan Data

Tahap awal dalam pembangunan sistem rekomendasi musik ini adalah proses pra-pemrosesan data untuk memastikan bahwa data yang digunakan telah bersih, relevan, dan siap diolah. Dataset berasal dari kombinasi Spotify Dataset dan YouTube Music yang diperoleh melalui Kaggle dan scraping, berisi 24.329 entri lagu dengan variabel utama: performer, genre, mood, year, dan status lagu (Hit atau Not Hit). Langkah-langkah pra-pemrosesan meliputi:

- Data Cleaning: Langkah awal dalam pra-pemrosesan adalah pembersihan data (data cleaning) untuk memastikan bahwa data yang digunakan dalam pelatihan model telah bersih dan siap diolah. Proses ini mencakup deteksi dan penanganan nilai kosong (*missing values*), penghapusan entri duplikat, serta koreksi terhadap data yang tidak valid atau tidak konsisten. Proses data cleaning ditunjukkan dalam Gambar 5 dan 6, dimana adanya proses sebelum dan sesudah dilakukannya data cleaning.

```
Jumlah data sebelum cleaning: 24329
Missing value per kolom:
 youtubeId          0
itemLabel          0
performerLabel   532
youtubeViews       0
year               0
genreLabel      5385
mood               0
hit_label          0
dtype: int64
```

Gambar 5. Sebelum data cleaning

Gambar ini menunjukkan contoh data mentah sebelum dilakukan pembersihan. Terlihat adanya entri yang mengandung duplikasi, ketidakkonsistenan format, dan potensi nilai kosong yang dapat mengganggu proses analisis.

```
Missing value per kolom setelah cleaning:
 youtubeId        0
itemLabel        0
performerLabel   0
youtubeViews     0
year             0
genreLabel       0
mood             0
hit_label        0
dtype: int64
```

Gambar 6. Setelah data cleaning

Gambar ini menunjukkan dataset setelah melalui proses pembersihan. Seluruh data duplikat dan nilai kosong telah dihapus atau diisi, serta format penulisan setiap atribut telah diseragamkan sehingga siap diproses pada tahap selanjutnya. Berdasarkan hasil pemeriksaan terhadap dataset, tidak ditemukan adanya nilai kosong pada seluruh kolom**,** termasuk kolom youtubeId, judul, *performerLabel, youtubeViews*, *year, genreLabel, mood,* dan hit_label. Informasi ini ditunjukkan melalui ringkasan statistik setelah proses pembersihan, sebagaimana disajikan pada Gambar 6. Selain itu, format data telah dipastikan sesuai dengan tipe data yang diperlukan, di mana variabel kategorikal dan numerik telah terstruktur secara tepat. Dengan demikian, dataset dinyatakan layak untuk dilanjutkan ke tahap transformasi dan pelabelan.

- Label Encoding: Proses label encoding dilakukan untuk mengubah data kategorikal menjadi representasi numerik agar dapat diproses oleh algoritma *machine learning*. Dalam konteks penelitian ini, kolom yang dikodekan meliputi genreLabel, mood, dan performerLabel. Proses Label Encoding ditunjukkan dalam Gambar 7 dan 8.

```
Sebelum Encoding:

  genreLabel                              mood       performerLabel
0  short film, dance-pop, musical film  energetic    Ed Sheeran
1             hip hop music, pop rap   energetic  Charlie Puth
2       soul, funk, boogie, disco-pop  energetic   Mark Ronson
3       soul, funk, boogie, disco-pop  energetic   Mark Ronson
4                           dance-pop  energetic    Crazy Frog
```

Gambar 7. Sebelum label encoding

Gambar ini memperlihatkan nilai atribut kategorikal seperti genre, mood, dan performer yang masih berbentuk teks. Format ini belum dapat diproses langsung oleh algoritma machine learning.

Setelah Encoding:

	genreLabel	genreLabel_enc	mood	mood_enc	performerLabel	performerLabel_
0	short film, dance-pop, musical film	1678	energetic	0	Ed Sheeran	1472
1	hip hop music, pop rap	978	energetic	0	Charlie Puth	907
2	soul, funk, boogie, disco-pop	1719	energetic	0	Mark Ronson	3273
3	soul, funk, boogie, disco-pop	1719	energetic	0	Mark Ronson	3273
4	dance-pop	448	energetic	0	Crazy Frog	1074

Gambar 8. Sebelum label encoding

Gambar ini menunjukkan hasil transformasi atribut kategorikal menjadi representasi numerik menggunakan label encoding. Proses ini memungkinkan model CART mengolah data secara efektif tanpa kehilangan makna kategori.
Sebelum encoding, nilai dalam kolom-kolom tersebut berupa teks, seperti *genre "soul, funk, boogie, disco-pop"*, *mood "energetic"*, dan p*erformer* seperti "*Ed Sheeran*" atau "*Charlie Puth*". Nilai-nilai ini kemudian dikonversi menjadi angka integer unik yang merepresentasikan masing-masing kategori. Misalnya, *mood "energetic"* dikodekan sebagai 0, sedangkan nama artis seperti *"Charlie Puth"* dan *"Mark Ronson"* dikodekan masing-masing sebagai 927 dan 3273. Transformasi ini ditunjukkan dalam Gambar 7 dan 8, yang memperlihatkan perubahan data sebelum dan sesudah proses encoding. Hasil encoding menjamin bahwa data kategorikal dapat digunakan dalam proses pelatihan model tanpa

Commented [A3]: Berikan garis tepi pada setiap gambar.

kehilangan makna kategorisnya. Proses ini juga mendukung efisiensi pemrosesan dan konsistensi dalam representasi fitur.

- Normalization: Proses normalisasi dilakukan untuk memastikan bahwa nilai pada atribut numerik memiliki skala yang seragam dan tidak mendominasi atribut lainnya dalam proses pelatihan model. Dalam penelitian ini, kolom numerik yang dinormalisasi adalah kolom *year*, yang merepresentasikan tahun rilis lagu. Sebelum dinormalisasi, nilai *year* memiliki rentang data yang cukup besar, misalnya antara tahun 2005 hingga 2017. Perbedaan skala ini dapat memengaruhi kinerja model dalam melakukan pemisahan data selama proses klasifikasi. Oleh karena itu, dilakukan proses normalisasi menggunakan metode Min-Max Scaling, sehingga nilai *year* dikonversi ke dalam rentang antara 0 dan 1. Hasil transformasi ini ditampilkan dalam Gambar 9 dan 10, yang menunjukkan perubahan nilai *year* menjadi *year_norm* dengan skala seragam, misalnya tahun 2017 menjadi 0.9965 dan tahun 2005 menjadi 0.9906. Dengan skala yang sudah dinormalisasi ini, seluruh fitur numerik memiliki bobot yang seimbang dan tidak mendistorsi hasil pelatihan model klasifikasi.

Sebelum Normalisasi:

	year
0	2017
1	2015
2	2014
3	2014
4	2005

Gambar 9. Sebelum normalisasi

Gambar ini menunjukkan nilai tahun rilis lagu (year) dalam bentuk asli. Skala tahun yang bervariasi berpotensi mendominasi fitur lain dalam proses pembelajaran jika tidak dinormalisasi.

Setelah Normalisasi:

	year	year_norm
0	2017	0.996543
1	2015	0.995556
2	2014	0.995062
3	2014	0.995062
4	2005	0.990617

Gambar 10. Sebelum label encoding

Gambar ini memperlihatkan hasil normalisasi nilai year ke dalam rentang 0–1 menggunakan metode Min-Max Scaling. Dengan skala seragam, model dapat menyeimbangkan bobot setiap fitur numerik dan kategorikal.

- *Feature Selection*: Tahap *feature selection* bertujuan untuk memilih fitur yang paling relevan dan berpengaruh terhadap variabel target guna meningkatkan akurasi dan efisiensi model klasifikasi. Dalam penelitian ini, fitur yang dipilih berdasarkan relevansi terhadap status lagu (*Hit* atau Not *Hit*) adalah: *genreLabel*_enc (hasil encoding *genre*), *mood*_enc (hasil encoding *mood*), performerLabel_enc (hasil encoding *performer*), dan *year*_norm (hasil normalisasi tahun rilis lagu). Keempat fitur tersebut terbukti memiliki kontribusi signifikan dalam membedakan lagu berdasarkan popularitasnya, sebagaimana ditunjukkan dalam Gambar 11 dan 12 yang menampilkan subset data hasil seleksi fitur. Masing-masing fitur telah melalui proses encoding atau normalisasi sebelumnya, sehingga siap digunakan dalam pelatihan model. Sementara itu, variabel target yang akan diprediksi adalah hit_label_enc, yaitu representasi numerik dari status lagu yang telah dikategorikan sebagai *Hit* atau *Not Hit*. Struktur data yang terorganisir ini memastikan bahwa input dan output model tersusun secara konsisten dan optimal untuk proses klasifikasi menggunakan algoritma CART.

Fitur yang digunakan untuk pelatihan model:

	genreLabel_enc	mood_enc	performerLabel_enc	year_norm
0	1678	0	1472	0.996543
1	978	0	907	0.995556
2	1719	0	3273	0.995062
3	1719	0	3273	0.995062
4	448	0	1074	0.990617

Gambar 11. Sebelum *Feature Selection*

Gambar ini menampilkan dataset sebelum proses seleksi fitur, di mana masih terdapat atribut yang tidak relevan atau memiliki pengaruh kecil terhadap target klasifikasi.

Target (y) yang akan diprediksi:

0	0
1	0
2	0
3	0
4	0

Name: hit_label_enc, dtype: int64

Gambar 12. Sesudah *Feature Selection*

Gambar ini memperlihatkan dataset yang hanya memuat fitur relevan, yaitu performer, genre, mood, dan year yang telah diolah. Dataset hasil seleksi ini digunakan untuk pelatihan model CART.

- *Data Splitting*: Untuk mengevaluasi performa model secara menyeluruh, data dibagi ke dalam tiga skenario pembagian, yaitu rasio 80:20**,** 70:30, dan 60:40, yang masing-masing merepresentasikan proporsi antara data pelatihan (*training*) dan data pengujian (*testing*). Proses pembagian dilakukan menggunakan parameter test_size dengan nilai 0.2, 0.3, dan 0.4, sesuai dengan kebutuhan tiap skenario. *Split data* ditunjukkan dalam Gambar 13, 14, dan 15.

```
Train-Test Split 80:20 (test_size = 0.2)
Ukuran X_train: (15058, 4)
Ukuran X_test : (3765, 4)
Ukuran y_train: (15058,)
Ukuran y_test : (3765,)
```

Gambar 13. *Split data* 80:20

Gambar ini menampilkan pembagian dataset menjadi 80% data latih dan 20% data uji. Rasio ini menghasilkan performa terbaik dengan akurasi dan F1-score yang seimbang.

```
Train-Test Split 70:30 (test_size = 0.3)
Ukuran X_train: (13176, 4)
Ukuran X_test : (5647, 4)
Ukuran y_train: (13176,)
Ukuran y_test : (5647,)
```

Gambar 14. *Split data* 70:30

Gambar ini menunjukkan pembagian dataset menjadi 70% data latih dan 30% data uji, dengan performa cukup baik namun sedikit di bawah rasio 80:20.

```
Train-Test Split 60:40 (test_size = 0.4)
Ukuran X_train: (11293, 4)
Ukuran X_test : (7530, 4)
Ukuran y_train: (11293,)
Ukuran y_test : (7530,)
```

Gambar 15. *Split data* 60:40

Gambar ini menampilkan pembagian dataset menjadi 60% data latih dan 40% data uji, yang menghasilkan akurasi tinggi namun performa kurang optimal dalam mengenali kelas minoritas.

- Pada rasio 80:20 (test_size = 0.2), data latih (X_train) terdiri dari 30.580 entri**,** sementara data uji (X_test) sebanyak 7.635 entri**.** Jumlah entri target (y_train dan y_test) mengikuti pembagian yang sama.
- Untuk rasio 70:30 (test_size = 0.3), diperoleh X_train sebanyak 31.176 entri dan X_test sebanyak 5.647 entri**.**
- Sementara itu, skenario 60:40 (test_size = 0.4) menghasilkan X_train sebanyak 11.293 entri dan X_test sebanyak 7.530 entri**.**

Pembagian ini dilakukan untuk mengidentifikasi pengaruh proporsi data terhadap akurasi dan kestabilan performa model, khususnya dalam mengenali kelas minoritas. Hasil pengujian dari masing-masing skenario dievaluasi lebih lanjut pada tahap berikutnya untuk menentukan rasio pembagian terbaik dalam konteks klasifikasi popularitas lagu.

3.3 Evaluasi Model Berdasarkan Split Data

- Split Data 60:40

Berdasarkan hasil evaluasi pada skenario pembagian data 60% untuk pelatihan dan 40% untuk pengujian**,** model menghasilkan akurasi sebesar 98,22%**.** Meskipun angka akurasi tampak tinggi, hasil ini perlu dianalisis lebih dalam melalui metrik lain seperti precision**,** recall**,** dan F1-score**,** terutama pada kelas minoritas (*Hit*). Hasil dari split data 60:40 ditunjukkan dalam Gambar 16.

```
=== EVALUASI CONFUSION MATRIX UNTUK BERBAGAI SPLIT DATA ===

Train-Test Split 60:40 (test_size = 0.4)
Confusion Matrix:
[[  17   67]
 [  67 7379]]
Akurasi: 98.22%
Classification Report:
              precision    recall  f1-score   support

         Hit       0.20      0.20      0.20        84
  Not a Hit       0.99      0.99      0.99      7446

    accuracy                           0.98      7530
   macro avg       0.60      0.60      0.60      7530
weighted avg       0.98      0.98      0.98      7530
```

Gambar 16. Hasil skenario 60:40

Gambar ini menunjukkan hasil evaluasi model pada rasio 60:40, di mana akurasi tinggi tetapi recall kelas Hit rendah akibat ketidakseimbangan data.

Meskipun model menunjukkan akurasi tinggi secara keseluruhan (98,22%), hal ini tidak mencerminkan performa sebenarnya, terutama dalam mengenali lagu-lagu populer (Hit). Hal ini disebabkan oleh ketidakseimbangan data, di mana jumlah lagu Not Hit jauh lebih besar daripada Hit. Akibatnya, model cenderung bias dan kurang mampu mengenali kelas minoritas secara efektif. Oleh karena itu, akurasi tidak boleh menjadi satu-satunya indikator performa model, dan perlu mempertimbangkan F1-score pada kelas minoritas untuk evaluasi yang lebih adil. Dalam kasus ini, pembagian data 60:40 menunjukkan hasil yang kurang optimal.

- Split Data 70.30

Pada skenario ini, data dibagi menjadi 70% untuk pelatihan dan 30% untuk pengujian, menghasilkan akurasi sebesar 98,34%. Model memperlihatkan peningkatan performa dalam mengenali lagu yang termasuk kategori *Hit*. Hasil dari split data 70:30 ditunjukkan dalam Gambar 17.

```
Train-Test Split 70:30 (test_size = 0.3)
Confusion Matrix:
[[  16   46]
 [  48 5537]]
Akurasi: 98.34%
Classification Report:
              precision    recall  f1-score   support

         Hit       0.25      0.26      0.25        62
  Not a Hit       0.99      0.99      0.99      5585

    accuracy                           0.98      5647
   macro avg       0.62      0.62      0.62      5647
weighted avg       0.98      0.98      0.98      5647
```

Gambar 17. Hasil skenario 70:30

Gambar ini menampilkan hasil evaluasi pada rasio 70:30, dengan peningkatan recall dan F1-score untuk kelas Hit dibandingkan skenario 60:40.
Model menunjukkan performa yang lebih seimbang dibandingkan skenario sebelumnya (60:40). Meskipun nilai recall dan F1-score kelas *Hit* masih tergolong rendah, terdapat peningkatan yang signifikan. Model mulai dapat mengenali lebih banyak lagu populer secara benar, meskipun sebagian besar prediksi tetap didominasi oleh kelas *Not Hit*.

- Split Data 80.20

Pada pembagian data 80% pelatihan dan 20% pengujian**,** model menghasilkan akurasi 98,33%**.** Ini merupakan skenario dengan performa terbaik secara keseluruhan**.** Hasil dari split data 80:20 ditunjukkan dalam Gambar 18.

```
Train-Test Split 80:20 (test_size = 0.2)
Confusion Matrix:
[[  13   28]
 [  35 3689]]
Akurasi: 98.33%
Classification Report:
              precision    recall  f1-score   support

         Hit       0.27      0.32      0.29        41
  Not a Hit       0.99      0.99      0.99      3724

    accuracy                           0.98      3765
   macro avg       0.63      0.65      0.64      3765
weighted avg       0.98      0.98      0.98      3765
```

Gambar 18. Hasil skenario 70:30

Gambar ini menunjukkan hasil evaluasi terbaik pada rasio 80:20, dengan akurasi tinggi dan F1-score tertinggi untuk kelas Hit, menunjukkan keseimbangan performa.
Meskipun akurasinya sedikit lebih rendah dari 70:30 (selisih 0,01%), skenario ini memberikan nilai F1-score tertinggi untuk kelas *Hit***.** Hal ini menunjukkan bahwa proporsi pelatihan data yang lebih besar memberikan manfaat signifikan dalam mengenali pola lagu-lagu populer (*Hit*). Oleh karena itu, split 80:20 dianggap sebagai konfigurasi paling optimal**,** dengan akurasi tinggi dan performa klasifikasi yang relatif seimbang.

3.4 Evaluasi Akurasi Model

Evaluasi performa model dilakukan dengan menerapkan sejumlah metrik evaluasi yang umum digunakan dalam pengujian algoritma klasifikasi, yaitu precision**,** recall**,** F1-score**,** dan accuracy**.** Metrik-metrik ini digunakan untuk mengukur sejauh mana model mampu melakukan klasifikasi secara tepat terhadap data uji, khususnya dalam membedakan antara kelas *Hit* dan *Not Hit.* Evaluasi ini penting untuk mengetahui kinerja model tidak hanya dari sisi akurasi keseluruhan, tetapi juga kemampuan model dalam mengenali kelas minoritas yang relevan. Berikut ini disajikan dalam Tabel 1 yaitu hasil klasifikasi model terhadap data uji berdasarkan masing-masing rasio pembagian data yang telah digunakan.

Tabel 1. Hasil Akurasi (Precision, Recall, F1-Score, Support)

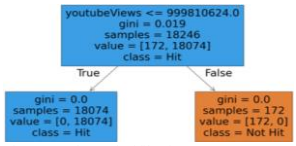
Split	Accuracy	Precision	Recall	F1-Score
60:40	98,22	0,20	0,20	0,20
70:30	98,34	0,25	0,26	0,99
80:20	**98,33**	**0,27**	**0,32**	**0,29**

Dalam evaluasi model, dilakukan perbandingan terhadap tiga skenario pembagian data, yaitu 60:40, 70:30, dan 80:20 yang tunjukkan dalam Table 1. Berdasarkan hasil klasifikasi, rasio 80:20 menghasilkan akurasi tertinggi sebesar 98,30%, disertai nilai precision, recall, dan F1-score yang lebih seimbang dibandingkan dua skenario lainnya. Hal ini menunjukkan bahwa proporsi 80% data latih dan 20% data uji memberikan performa paling optimal dalam memprediksi status lagu secara akurat.

3.5 Visualisasi Pohon Keputusan

Berikut adalah visualisasi Decision Tree CART yang digunakan dalam proses klasifikasi ditunjukkan pada Gambar 19.

Gambar ini memperlihatkan struktur pohon keputusan hasil pelatihan model CART. Terlihat bahwa atribut youtubeViews menjadi pemisah utama untuk menentukan status lagu Hit atau Not Hit.



Gambar 19. Pohon Klasifikasi (*Decision Tree CART*)

Visualisasi pohon keputusan digunakan untuk menggambarkan proses klasifikasi dalam model *Decision Tree CART* (Classification and Regression Tree) yang dikembangkan pada penelitian ini. Model ini secara otomatis membagi dataset berdasarkan atribut yang paling informatif terhadap label target, yaitu apakah sebuah lagu tergolong "*Hit*" atau "*Not Hit*". Pada Gambar 4 ditunjukkan struktur pohon keputusan yang dihasilkan oleh model dengan kedalaman maksimum 3. Berdasarkan hasil pelatihan, atribut yang paling menentukan adalah jumlah view di YouTube (*youtubeViews*). Akar pohon membagi data berdasarkan nilai ambang sebesar 999.810.624 *views*. Lagu dengan jumlah views di bawah atau sama dengan nilai tersebut diklasifikasikan sebagai *Hit*, sementara lagu dengan views lebih tinggi diklasifikasikan sebagai *Not Hit***.**

Setiap simpul (node) pada pohon menyajikan informasi sebagai berikut:

- gini: mengukur ketidakmurnian (impurity) pada node tersebut. Nilai mendekati 0 menunjukkan bahwa data pada node sebagian besar berasal dari satu kelas.
- samples: jumlah total data pada node tersebut.
- value: distribusi jumlah data untuk masing-masing kelas.
- class: hasil klasifikasi pada node berdasarkan mayoritas kelas.

Pohon menunjukkan bahwa model dapat memisahkan data dengan sangat baik berdasarkan atribut youtubeViews saja, dengan nilai Gini 0,0 pada masing-masing daun (leaf node). Hal ini mengindikasikan bahwa model mampu menghasilkan prediksi yang sangat murni, tanpa ambiguitas kelas di level akhir.

Visualisasi ini memberikan gambaran intuitif mengenai proses pengambilan keputusan oleh model, serta membantu peneliti dan pengguna dalam memahami logika klasifikasi yang digunakan sistem. Dengan hanya satu pemisah utama, model memberikan hasil klasifikasi yang sederhana namun efektif, sesuai dengan distribusi data dalam dataset.

3.6 Framework Flask

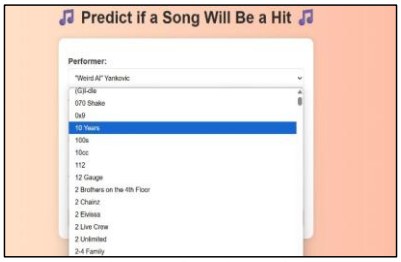
Hasil implementasi dari penelitian ini berupa sebuah sistem rekomendasi musik berbasis web yang mampu memberikan daftar lagu kepada pengguna berdasarkan empat input utama, yaitu *performer, genre, mood,* dan *year.* Sistem ini dirancang menggunakan framework Python Flask sebagai

antarmuka web, dan Decision Tree CART sebagai algoritma inti untuk klasifikasi dan prediksi.

Gambar 20. Tampilan halaman utama sistem prediksi lagu

Gambar ini menampilkan antarmuka web utama sistem rekomendasi, dengan form input berupa dropdown untuk performer, genre, mood, dan year.

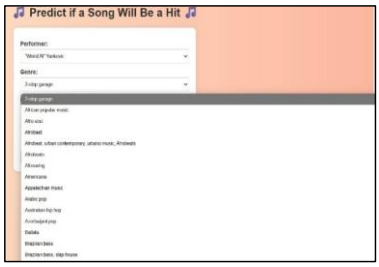
Pada antarmuka ini, pengguna diberikan kemudahan untuk secara intuitif memilih dan mengisi keempat parameter utama yang dibutuhkan dalam proses prediksi. Antarmuka dirancang sedemikian rupa agar ramah pengguna, dengan tampilan yang jelas dan interaktif. Pada tahap ini, sistem secara otomatis menyajikan empat elemen *dropdown* utama, masing-masing merepresentasikan parameter yang diperlukan, yaitu *genre* musik, suasana (*mood*), nama *performer*, dan tahun rilis lagu:



Gambar 21. Opsi *dropdown performer*

Gambar ini menunjukkan daftar pilihan performer (penyanyi atau grup musik) yang tersedia dalam sistem untuk dijadikan input prediksi.

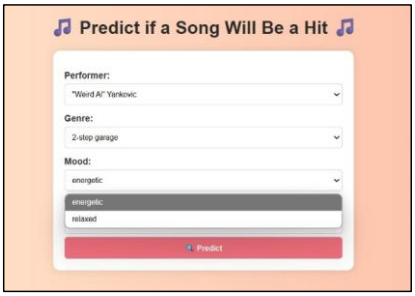
- *Performer*: Nama penyanyi atau grup musik yang membawakan lagu.



Gambar 22. Opsi *dropdown genre*

Gambar ini menampilkan daftar kategori musik, mulai dari genre populer seperti pop, rock, hingga sub-genre spesifik.

- Genre: Kategori musik dari lagu yang dimaksud, seperti pop, rock, hip-hop, afrobeat, blues, jazz, hingga sub-genre spesifik seperti 2-step garage.



Gambar 23. Opsi *dropdown mood*

Gambar ini memperlihatkan daftar pilihan suasana atau mood lagu, misalnya energetic atau relaxed, yang mempengaruhi hasil prediksi.

- *Mood*: Suasana atau nuansa emosional dari sebuah lagu merepresentasikan persepsi pendengar terhadap karakter emosional lagu tersebut, seperti energetic untuk menggambarkan perasaan semangat atau penuh energi, serta relaxed untuk mencerminkan suasana santai atau tenang saat mendengarkan lagu.

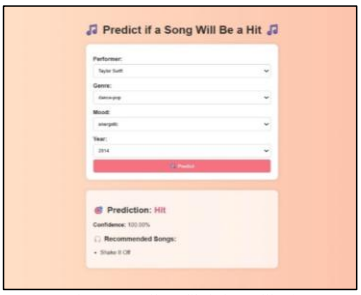
Gambar 24. Opsi *dropdown year*

Gambar ini menampilkan input tahun rilis lagu, yang menjadi salah satu parameter dalam model klasifikasi.

- *Year*: Tahun rilis lagu. Sistem ini dirancang untuk menerima tahun dalam format numerik.

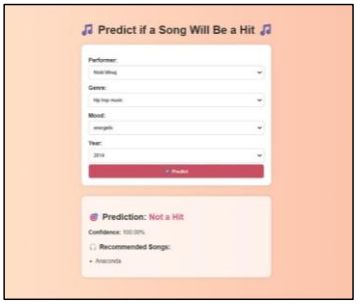
Setelah pengguna mengisi keempat parameter yang disediakan pada antarmuka sistem, yaitu performer, genre, mood, dan year, maka pengguna dapat menekan tombol “Predict” untuk memulai proses prediksi. Ketika tombol ini diaktifkan, sistem akan mengolah input yang diberikan dengan menggunakan model klasifikasi Decision Tree CART yang telah dilatih sebelumnya. Model ini akan menganalisis kombinasi dari keempat fitur tersebut

dan menelusuri jalur keputusan pada struktur pohon untuk menentukan keluaran prediksi. Selanjutnya, sistem akan menghasilkan output berupa informasi apakah lagu yang dimaksud memiliki potensi untuk dikategorikan sebagai lagu "Hit" atau "Not Hit". Hasil prediksi ini ditampilkan secara real-time pada antarmuka pengguna, dilengkapi dengan rekomendasi lagu serupa dan nilai confidence score dari model, sehingga pengguna dapat memperoleh gambaran yang lebih jelas mengenai tingkat keyakinan sistem terhadap hasil klasifikasi tersebut.



Gambar 25. Hasil prediksi sistem jika *hit*

Gambar ini memperlihatkan tampilan output sistem ketika hasil prediksi model menunjukkan lagu berstatus "Hit". Output dilengkapi daftar lagu serupa dan confidence score.



Gambar 26. Hasil prediksi sistem jika *not hit*

Gambar ini menunjukkan tampilan output sistem ketika lagu diprediksi berstatus "Not Hit", disertai rekomendasi lagu lain dan nilai confidence score dari model.

4. Kesimpulan

Penelitian ini telah berhasil mengembangkan sistem rekomendasi musik berbasis web menggunakan algoritma Decision Tree CART yang mampu mengklasifikasikan lagu sebagai "Hit" atau "Not Hit" berdasarkan input pengguna berupa genre, mood, performer, dan tahun rilis. Model ini memberikan hasil evaluasi akurasi tinggi sebesar 98,33% dengan konfigurasi pembagian data terbaik pada rasio 80:20, menunjukkan bahwa metode CART efektif dalam menangani data musik yang kompleks dan beragam. Sistem yang dirancang juga memiliki antarmuka interaktif yang memudahkan pengguna dalam memberikan input dan menerima hasil prediksi secara real-time.

Sebagai tindak lanjut, pengembangan sistem ini dapat diarahkan pada integrasi data real-time dari platform musik seperti YouTube Music dan Spotify agar sistem mampu beradaptasi secara dinamis terhadap tren musik terkini. Selain itu, eksplorasi terhadap algoritma lain seperti Random Forest atau model berbasis deep learning juga dapat dilakukan guna meningkatkan akurasi dan cakupan rekomendasi. Implementasi sistem pada skala pengguna yang lebih luas serta penyempurnaan antarmuka dan pengalaman pengguna juga menjadi fokus pengembangan selanjutnya untuk menghasilkan sistem rekomendasi musik yang lebih cerdas, adaptif, dan responsif terhadap kebutuhan pengguna.

Daftar Pustaka:

Amin, M. I. F. R., Amartharizqi, M. R., Sofi, F. A., & Anggraini Puspita Sari. (2023). Sistem Rekomendasi Musik Berdasarkan Preferensi Pengguna Dengan Menggunakan Metode Natural Language Processing (NLP). *Seminar Nasional Informatika Bela Negara (SANTIKA)*, *3*, 129–133. http://santika.upnjatim.ac.id/submissions/index.php/santika/article/view/276%0Ahttp://santika.upnjatim.ac.id/submissions/index.php/santika/article/download/276/118

Appalanidu H, R. C., Kumar Badhan, A., Pushpa, B., Jhansi Rani, A., Sai Dharani, A., & Venkata Sai Sindhuja, M. (2021). Music Recommendation System With Advanced Classification. *International Journal of Engineering Research & Technology (IJERT)*, *10*(07). www.ijert.org

Asrianda, Damayanti, I., & Maryana. (2024). The Use Of Classification And Regression Tree (CART) Algorithm In Predicting Rice Yield Based On The Amount Of Subsidized Fertilizer Usage In Lokseumawe City. *Jatilima : Jurnal Multimedia Dan Teknologi Informasi*, *06*(02), 183–198.

Azwanti, N., & Putria, N. E. (2024). *Analisis Kepuasan Customer pada Sdtechnology Computer dengan Algoritma Decision Tree*. http://journal.aptikomkepri.org/index.php/JDDAT

Fiqri, M., Bin, F., Lizen, S., & Ikrom, M. (2025). Implementation of Supervised Learning Algorithm on Spotify Music Genre Classification. *IJATIS: Indonesian Journal of Applied Technology and Innovation Science*, *2*(1), 7–12.

Fitradhi, N. R., Firman Hidayat, M., Saputro, T. W., Alifian, M. G., & Sari, A. P. (2023). Laporan Final Project Machine Learning Rekomendasi Musik Spotify Menggunakan Metode K-

Means. *Seminar Nasional Informatika Bela Negara (SANTIKA)*, *3*, 2747–0563.

Hakim, A. R., Dewi Marini Umi Atmaja, Amat Basri, & Andri Ariyanto. (2023). Performance Analysis of Classification and Regression Tree (CART) Algorithm in Classifying Male Fertility Levels with Mobile-Based. *Tech-E*, *7*(1), 10–20. https://doi.org/10.31253/te.v7i1.2110

Harahap, L. S., & Harahap, D. M. R. (2025). REKOMENDASI CERDAS: BIG DATA DALAM MEDIA DAN HIBURAN UNTUK PENGALAMAN YANG LEBIH PERSONAL. *Jurnal Sistem Dan Teknologi Informasi (JSTI)*, *7*(1), 89–99.

Harjananto, D. Y., Kartika Dewi, R., & Brata, K. C. (2021). Pengembangan Sistem Rekomendasi Musik berdasarkan Waktu berbasis Android. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *5*(5), 1729–1733. http://j-ptiik.ub.ac.id

Hartono, A. D., & Sutopo, J. (2024). Implementasi Algoritma Upper Confidence Bound Untuk Sistem Rekomendasi Musik. *Jurnal Algoritma*, *21*(2), 82–90. https://doi.org/10.33364/algoritma/v.21-2.2095

Indah Prabawati, N., Widodo, & Ajie, H. (2019). Kinerja Algoritma Classification And Regression Tree (Cart) dalam Mengklasifikasikan Lama Masa Studi Mahasiswa yang Mengikuti Organisasi di Universitas Negeri Jakarta. *PINTER : Jurnal Pendidikan Teknik Informatika Dan Komputer*, *3*(2), 139–145. https://doi.org/10.21009/pinter.3.2.9

Izzah, A., & Widyastuti, R. (2017). Prediksi Harga Saham Menggunakan Improved Multiple Linear Regression untuk Pencegahan Data Outlier. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, *2*(3), 141–150. https://doi.org/10.22219/kinetik.v2i3.268

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 (Switzerland)*, *11*(21). https://doi.org/10.3390/electronics11213518

Madani, M. A., Helmi, G., & Hendra, Y. (2024). Sistem Rekomendasi Musik Menggunakan Machine Learning. *J-TECH : Journal Technology of Computer*, *1*(1), 40–49.

Maulida, I., Suyatno, A., & Hatta, H. R. (2016). Seleksi Fitur Pada Dokumen Abstrak Teks Bahasa Indonesia Menggunakan Metode Information Gain. *Jurnal SIFO Mikroskil*, *17*(2), 249–258. https://doi.org/10.55601/jsm.v17i2.379

Mienye, I. D., & Jere, N. (2024). A Survey of Decision Trees: Concepts, Algorithms, and Applications. *IEEE Access*, *12*, 86716–86727. https://doi.org/10.1109/ACCESS.2024.3416838

Mukhsinin, D. A., Rafliansyah, M., Ibrahim, S. A., Rahmaddeni, R., & Wulandari, D. (2024). Implementasi Algoritma Decision Tree untuk Rekomendasi Film dan Klasifikasi Rating pada Platform Netflix. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, *4*(2), 570–579. https://doi.org/10.57152/malcom.v4i2.1255

Rachman, M. D., & Voutama, A. (2024). Implementasi Algoritma K-Means Dalam Sistem Rekomendasi Musik Menggunakan Python. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *8*(3), 3857–3862. https://doi.org/10.36040/jati.v8i3.9635

Rasyid, A., Gilbijatno, S., Pramudya, A. W., Prasetyo, D., & Informatika, T. (2025). Implementasi Algoritma Decision Tree CART untuk Deteksi Dini. *PROSIDING SEMINAR NASIONAL TEKNOLOGI DAN SAINS TAHUN*, *4*, 440–445.

Sánchez-Moreno, D., Zheng, Y., & Moreno-García, M. N. (2020). Time-aware music recommender systems: Modeling the evolution of implicit user preferences and user listening habits in a collaborative filtering approach. *Applied Sciences (Switzerland)*, *10*(15), 1–33. https://doi.org/10.3390/APP10155324

Sari, A.P, Suzuki, H., Kitajima, T., Yasuno, T., Arman Prasetya, D., & Rabi', Abd. (2021). Deep convolutional long short-term memory for forecasting wind speed and direction. SICE Journal of Control, Measurement, and System Integration, 14(2), 30–38. https://doi.org/10.1080/18824889.2021.1894878