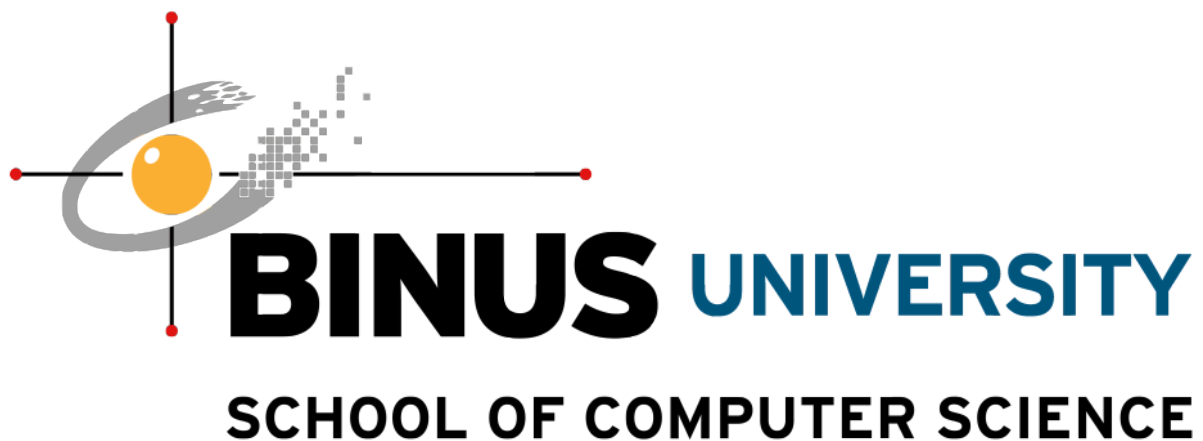


Laporan UAP Machine Learning

Personalized Music Classifier & Recommender AI



Kelompok 4

2702341360 – Ahya Muhammad Abiyu Salam

2702282490 - Kenneth Owen

2702359502 - Prince Lukas Marpaung

1. Pendahuluan.....	3
1.1 Latar belakang.....	3
1.2 Permasalahan yang diangkat.....	4
1.3 Tujuan dan Manfaat projek.....	4
1.4 Ruang Lingkup.....	4
2. Dataset dan Pemrosesan Data.....	5
2.1 Sumber Dataset.....	5
2.2 Struktur dan Deskripsi fitur.....	5
2.3 Pembersihan dan agregasi data.....	6
3. Pendekatan dan Methodology.....	6
3.1 Pendekatan.....	6
3.2 Methodology.....	7
4. Implementasi Aplikasi.....	7
4.1 Arsitektur Sistem Aplikasi.....	7
Komponen Utama:.....	8
4.2 Library dan tools yang digunakan.....	8
Backend (File Jupyter Notebook).....	8
Frontend (Streamlit).....	9
4.3 Interface dan interaksi.....	9
4.4 Alur kerja aplikasi.....	10
5. Evaluasi dan Analisis hasil.....	10
5.1 Distribusi Cluster.....	10
5.2 Analisa cluster.....	11
6. Pembagian Tugas Kelompok.....	11
6.1 pembagian tugas secara teknis.....	11
7. Kesimpulan dan Rekomendasi.....	12
7.1 Ringkasan temuan.....	12
8. Lampiran.....	12
8.1 Screenshot Demo.....	12
8.2 Contoh Output CSV.....	12

1. Pendahuluan

1.1 Latar belakang

Musik adalah sesuatu yang sudah melekat dalam kehidupan manusia sehari-hari, kerap mengisi berbagai kegiatan dan menjadi iringan setia. Meskipun musik atau lagu melimpah, seringkali sulit bagi pengguna untuk menemukan lagu yang benar-benar sesuai dengan mood atau aktivitas yang sedang mereka jalani. Beruntungnya, di era digital ini, data audio masif dari platform streaming seperti *Spotify* membuka peluang besar untuk analisis mendalam, memungkinkan kita mengidentifikasi karakteristik unik setiap lagu melalui klustering dan personalisasi. Oleh karena itu, proyek ini bertujuan mengembangkan aplikasi web inovatif yang memungkinkan pengguna mencari preferensi lagu berdasarkan mood atau aktivitas yang diinginkan. Dengan memanfaatkan teknik pemrosesan data dan kecerdasan buatan, kami akan menghubungkan karakteristik musik dengan suasana hati atau kegiatan, sehingga menciptakan pengalaman mendengarkan yang lebih efisien dan personal. Aplikasi ini diharapkan dapat memberikan solusi intuitif bagi pengguna untuk menemukan musik yang cocok untuk setiap momen, dengan adanya website ini akan meningkatkan kepuasan dan pengalaman mendengarkan mereka secara signifikan.

1.2 Permasalahan yang diangkat

Pengguna sering kesulitan mencari lagu yang sesuai dengan suasana hati atau aktivitas mereka di antara jutaan pilihan musik. Mereka menghabiskan waktu mencari, namun sering gagal menemukan komposisi yang tepat untuk berbagai momen. Ini tidak hanya membuat pencarian jadi tidak efisien, tapi juga mengurangi pengalaman mendengarkan yang seharusnya lebih personal. Akibatnya, pengguna cenderung terjebak pada daftar putar yang itu-itu saja, sehingga mereka tidak menemukan musik baru yang berpotensi jadi rekomendasi pribadi. Kami akan mengatasi masalah ini dengan menciptakan sistem cerdas yang mempermudah pengguna menemukan musik yang mereka inginkan, sekaligus membuka akses ke ragam lagu yang lebih luas.

1.3 Tujuan dan Manfaat proyek

Proyek ini memiliki tujuan utama untuk mengelompokkan lagu ke dalam klaster berdasarkan kesamaan fitur audio, seperti tempo, energy, dan instrumentasi. Selanjutnya, kami

mengembangkan sebuah website interaktif yang memanfaatkan pengelompokan ini untuk memberikan rekomendasi musik. Rekomendasi tersebut akan didasarkan pada mood atau aktivitas yang diinginkan pengguna dan memastikan setiap saran lagu benar-benar relevan dengan suasana hati atau kegiatan yang sedang mereka lakukan.

Manfaat dari proyek ini akan meningkatkan pengalaman mendengarkan musik bagi penggemar lagu. Website ini tidak hanya akan menyediakan rekomendasi cerdas, tetapi juga memungkinkan pengguna untuk menentukan fitur audio sendiri dan mendapatkan lagu yang sesuai dengan parameter input mereka. Selain itu, kami akan menyertakan fitur untuk menambahkan lagu langsung ke playlist pribadi, sehingga pengguna dapat mengelola koleksi musik mereka dengan mudah. Secara keseluruhan, proyek ini bertujuan untuk membuat penemuan musik lebih personal, efisien, dan menyenangkan dan memberikan jalan bagi eksplorasi musik yang lebih luas dan disesuaikan dengan preferensi pengguna.

1.4 Ruang Lingkup

Proyek ini memiliki ruang lingkup yang terfokus untuk memastikan pengembangan yang efisien dan hasil yang optimal. Data yang digunakan akan terbatas pada fitur-fitur audio yang berasal dari dataset *Spotify*, memastikan relevansi dan aksesibilitas data. Untuk pemodelan, proyek ini secara spesifik akan menggunakan algoritma *clustering* K-Means untuk mengelompokkan lagu berdasarkan kesamaan karakteristik audio mereka. Selanjutnya, demo aplikasi akan dibangun menggunakan Streamlit, sebuah *framework* yang cocok untuk pembuatan aplikasi web interaktif berbasis Python, dengan visualisasi data yang diimplementasikan menggunakan Plotly untuk menyajikan hasil analisis dan rekomendasi secara menarik dan informatif.

2. Dataset dan Pemrosesan Data

2.1 Sumber Dataset

Dataset yang digunakan dalam proyek ini diperoleh dari platform Kaggle, sebuah repositori data dan *notebook* yang populer. Meskipun diunduh dari Kaggle, sumber asli data tersebut adalah Spotify, penyedia layanan *streaming* musik global. Dataset ini sangat komprehensif, karena berisi informasi tentang ribuan lagu lengkap dengan fitur-fitur audionya. Fitur-fitur audio ini mencakup parameter seperti tempo, *danceability*, *energy*, *instrumentalness*, dan lain-lain, yang sangat krusial untuk analisis dan pengelompokan musik berdasarkan karakteristik musik.

2.2 Struktur dan Deskripsi fitur

Sistem yang akan dikembangkan dalam proyek ini dirancang dengan beberapa fitur utama untuk memberikan pengalaman pengguna yang intuitif dan personal dalam pencarian musik. Struktur aplikasi web akan mengintegrasikan elemen-elemen berikut:

1. Profil Pengguna: Fitur ini memungkinkan pengguna untuk membuat dan mengelola profil pribadi.
2. Pencarian Lagu Berdasarkan Mood dan Aktivitas: Fitur yang menjadi inti dari website ini, di mana pengguna dapat memilih preferensi lagu berdasarkan *mood* atau aktivitas yang ingin mereka lakukan. Sistem akan memproses input dari pengguna dan merekomendasikan lagu-lagu dari klaster yang relevan.
3. Finetuning Preferensi Audio (Kustomisasi): Untuk memberikan kontrol yang lebih spesifik, aplikasi akan menyediakan fitur yang memungkinkan pengguna untuk menyesuaikan atau '*finetune*' parameter audio yang sesuai dengan keinginan mereka. Finetune disini berupa pengaturan slider untuk *tempo*, *energy*, atau *danceability* dan lain-lain, sehingga rekomendasi yang diberikan menjadi sangat spesifik.
4. Penambahan Lagu ke Playlist: Pengguna akan memiliki kemampuan untuk menambahkan lagu yang mereka temukan dan sukai ke dalam *playlist* pribadi. Fitur ini mendukung manajemen koleksi musik pengguna dan memungkinkan mereka untuk menyimpan serta mengakses lagu-lagu favorit dengan mudah.

2.3 Pembersihan dan agregasi data

Proses pembersihan dan agregasi data merupakan langkah krusial untuk memastikan kualitas dan kesiapan data sebelum analisis lebih lanjut. Dalam proyek ini, langkah-langkah yang akan diambil meliputi:

1. Duplikasi Dihapus: Data akan melalui tahap pembersihan untuk menghapus entri yang terduplikasi. Ini penting untuk menghindari bias dalam analisis dan memastikan setiap lagu merepresentasikan data yang unik.
2. Fitur Numerik Dinormalisasi: Seluruh fitur numerik yang akan digunakan untuk proses *clustering* akan dinormalisasi menggunakan *scaler*. Normalisasi ini bertujuan untuk menyamakan skala fitur-fitur yang memiliki rentang nilai berbeda, sehingga tidak ada satu fitur pun yang mendominasi perhitungan jarak dalam algoritma *clustering*.
3. Menggunakan Kolom yang Berguna untuk *Clustering*: Data akan difilter, dan kolom-kolom yang tidak relevan atau tidak digunakan untuk proses *clustering* akan *didrop*. Ini membantu mengurangi dimensi data, meningkatkan efisiensi komputasi, dan memfokuskan analisis pada informasi yang penting untuk pembentukan klaster.

3. Pendekatan dan Methodology

3.1 Pendekatan

Proyek ini mengadopsi pendekatan berbasis *clustering* untuk mengelompokkan lagu. Setelah melalui tahap *preprocess* data yang meliputi pembersihan dan normalisasi, kami hanya menggunakan informasi fitur audio yang relevan dan diperlukan untuk klastering. Informasi ini esensial karena merefleksikan karakteristik unik setiap lagu seperti tempo, *danceability*, energi, dan lainnya. Untuk teknik *clustering*, kami secara spesifik akan menggunakan algoritma K-Means. K-Means dipilih karena efisiensinya dalam mengelompokkan data besar dan kemampuannya untuk mengidentifikasi pola tersembunyi berdasarkan kedekatan fitur-fitur audio, sehingga memungkinkan kami untuk mengorganisir lagu ke dalam klaster-klaster yang representatif untuk berbagai *mood* dan aktivitas.

3.2 Methodology

Metodologi yang digunakan dalam proyek ini terdiri dari beberapa tahapan berikut:

1. Pengumpulan Data: Tahap awal melibatkan perolehan dataset Spotify dari Kaggle. Pada tahap ini, kami memahami secara mendalam struktur dataset serta jenis-jenis fitur audio yang tersedia, seperti tempo, *danceability*, dan energi.
2. Preprocess Data: Data yang terkumpul akan melalui proses *preprocess* yang komprehensif. Ini mencakup pembersihan data dari entri duplikat, normalisasi fitur numerik untuk menyamakan skalanya, dan pengecekan distribusi data untuk mengidentifikasi anomali atau pola penting.
3. Clustering Lagu: Selanjutnya, kami akan menerapkan algoritma K-Means untuk mengelompokkan lagu berdasarkan kemiripan fitur audio mereka. Proses *clustering* ini bertujuan untuk mengidentifikasi klaster lagu dengan karakteristik akustik yang serupa.
4. Analisis Hasil Clustering: Klaster yang terbentuk kemudian akan dianalisis secara mendalam. Ini melibatkan pemeriksaan distribusi genre dan karakteristik dari setiap klaster, serta pemberian label interpretatif yang dapat mewakili *mood* atau aktivitas tertentu untuk setiap klaster.
5. Pengembangan Aplikasi: Kami akan mengembangkan aplikasi interaktif menggunakan Streamlit. Aplikasi ini akan menjadi antarmuka utama bagi pengguna untuk mendapatkan rekomendasi lagu berdasarkan *mood* atau aktivitas yang mereka pilih, memanfaatkan hasil *clustering* yang telah dilakukan.
6. Evaluasi Sistem: Tahap terakhir adalah evaluasi menyeluruh terhadap sistem. Evaluasi ini mencakup penilaian kualitas hasil *clustering* menggunakan metrik seperti *Silhouette Score*, *Davies-Bouldin Score*, dan *Calinski-Harabasz Score*.

4. Implementasi Aplikasi

4.1 Arsitektur Sistem Aplikasi

Aplikasi rekomendasi musik ini dirancang dengan arsitektur yang terstruktur, menghubungkan beberapa komponen utama dalam sebuah alur kerja yang efisien. Pipeline sistem ini dapat digambarkan sebagai berikut:

Start → Dataset → Preprocessing (Cleaning, Scaler) → Process Model (K-Means) → Simpan Model (Pickle) → Load Model (Streamlit) → Interaksi & Visualisasi

Komponen Utama:

1. Backend: Bagian *backend* adalah inti dari sistem ini, di mana pemrosesan data dan logika rekomendasi dijalankan. Ini terdiri dari:
 - Preprocessing Data: Proses ini melibatkan normalisasi fitur menggunakan *scaler* dari *scikit-learn* untuk menyamakan skala data numerik, diikuti dengan aplikasi algoritma *clustering* K-Means.
 - Model K-Means: Model ini bertanggung jawab untuk mengelompokkan lagu-lagu berdasarkan kemiripan fitur audionya.
 - Logika Filtering dan Klasifikasi: Modul ini akan memproses input pengguna (mood atau aktivitas) dan menggunakan hasil *clustering* untuk lagu yang paling relevan.
2. Penyimpanan Model: Untuk efisiensi, model K-Means dan hasil *clustering* akan disimpan dalam satu file .pkl (Pickle). File ini akan berisi objek-objek penting seperti *kmeans_model* itu sendiri, *scaler* yang digunakan untuk normalisasi, *df_with_clusters* (dataframe yang telah ditambahkan informasi klaster), *cluster_centers* (titik pusat setiap klaster), dan *cluster_genre_distribution* (distribusi genre di setiap klaster). Penyimpanan ini memungkinkan model dimuat dengan cepat saat aplikasi dijalankan tanpa perlu melatih ulang setiap kali.
3. Frontend: Antarmuka pengguna (UI) aplikasi dibangun menggunakan Streamlit. *Frontend* ini berperan penting dalam menyajikan UI interaktif kepada pengguna, memungkinkan mereka untuk memilih *mood* atau aktivitas, *finetuning* parameter audio, dan melihat rekomendasi lagu secara visual. Streamlit memudahkan proses pengembangan *frontend* yang responsif dan mudah digunakan, berinteraksi langsung dengan logika *backend* yang telah dimuat.

Dengan arsitektur ini, aplikasi mampu menerima data mentah, memprosesnya untuk mengidentifikasi pola musik, dan menyajikannya dalam bentuk rekomendasi yang personal dan interaktif kepada pengguna.

4.2 Library dan tools yang digunakan

Pengembangan proyek ini didukung oleh berbagai *library* dan *tools* Python yang esensial, terbagi menjadi kebutuhan *backend* (pemrosesan model) dan *frontend* (antarmuka aplikasi).

Backend (File Jupyter Notebook)

Pada sisi *backend*, kami memanfaatkan *library* untuk persiapan data, pemodelan *machine learning*, dan evaluasi:

- *os, pandas, numpy*: Digunakan untuk operasi sistem file, manipulasi data tabular, dan komputasi numerik.
- *sklearn.preprocessing.StandardScaler*: Untuk normalisasi fitur agar data siap untuk *clustering*.
- *sklearn.cluster.KMeans*: Implementasi algoritma K-Means sebagai inti dari model *clustering* kami.
- *pickle*: Digunakan untuk menyimpan dan memuat model *machine learning* yang telah dilatih, memungkinkan penggunaannya kembali tanpa *training* ulang.
- *matplotlib.pyplot, seaborn*: Untuk visualisasi data dan analisis eksplorasi hasil *clustering*.
- *datetime*: Untuk keperluan *timestamp* atau pencatatan waktu.
- *sklearn.metrics*: Menyediakan metrik evaluasi model *clustering* seperti *silhouette_score*, *davies_bouldin_score*, dan *calinski_harabasz_score* untuk menilai kualitas klaster.

Frontend (Streamlit)

Pada sisi *frontend*, *library* ini mendukung pembangunan antarmuka pengguna interaktif:

- *streamlit*: *Framework* utama untuk membangun aplikasi web interaktif dengan cepat dan mudah menggunakan Python.
- *pickle*: Digunakan kembali di *frontend* untuk memuat model dan data hasil *clustering* yang telah disimpan.
- *pandas, numpy*: Digunakan untuk manipulasi data yang akan ditampilkan atau diproses di *frontend*.
- *plotly.express, plotly.graph_objects, plotly.subplots, plotly.figure_factory*: Kumpulan *library* Plotly untuk menciptakan visualisasi data yang dinamis, interaktif, dan menarik, seperti *scatter plots*, *bar charts*, atau *heatmaps* untuk menampilkan hasil rekomendasi dan analisis.

4.3 Interface dan interaksi

Interface aplikasi berbasis web ini dirancang menggunakan Streamlit, terbagi menjadi lima tab utama, masing-masing dengan fungsionalitas spesifik:

1. Mood Selector: Memungkinkan pengguna memilih *mood* melalui emoji dan menyesuaikan *slider* fitur audio seperti tempo, energi, atau *danceability*, kemudian sistem merekomendasikan lagu berdasarkan parameter tersebut.
2. Smart Recommendations: Memberikan rekomendasi lagu berdasarkan aktivitas spesifik (misalnya "Workout" atau "Chill"), dilengkapi filter tambahan seperti genre, *danceability*, atau *energy* untuk kesesuaian maksimal.
3. Song Classifier: Fitur unik di mana pengguna dapat menginput nilai fitur audio secara manual, dan sistem akan memprediksi klaster lagu tersebut, memberikan wawasan tentang kategorinya.
4. Cluster Analysis: Menampilkan visualisasi mendalam seperti *radar chart*, *heatmap*, dan distribusi genre per klaster, membantu pengguna memahami pengelompokan lagu.
5. My Playlist: Berfungsi sebagai pusat manajemen koleksi musik pribadi, di mana pengguna dapat menyimpan lagu favorit dan mengekspor *playlist* ke format CSV.

4.4 Alur kerja aplikasi

Aplikasi ini dirancang dengan alur interaksi pengguna yang intuitif dan terarah, memastikan pengalaman pencarian musik yang mulus dan personal. Berikut adalah tahapan alur kerja utama yang akan dilalui pengguna:

- Onboarding: Pada awal penggunaan, pengguna akan diminta untuk mengisi informasi dasar seperti nama, preferensi genre, dan *mood* favorit, yang akan menjadi fondasi awal untuk personalisasi rekomendasi.
- Pilih Aktivitas / Moods: Pengguna dapat menentukan konteks spesifik mendengarkan lagu mereka, memilih *mood* seperti "santai" atau "semangat", atau aktivitas seperti "bekerja" atau "berolahraga".
- Sistem Memproses: Setelah menerima input *mood* atau aktivitas, sistem akan segera memproses dan menyesuaikan preferensi tersebut dengan klaster lagu yang paling relevan, memanfaatkan model *clustering* yang telah dilatih pada fitur-fitur audio.
- Rekomendasi Ditampilkan: Berdasarkan hasil pemrosesan, daftar lagu akan ditampilkan kepada pengguna, diurutkan berdasarkan tingkat kesesuaiannya dengan *mood* atau aktivitas yang telah dipilih.
- Pengguna Menyimpan *Playlist*: Pengguna memiliki opsi untuk menyimpan lagu-lagu favorit yang ditemukan ke dalam *playlist* pribadi, dan juga dapat mengunduh *playlist* tersebut untuk kemudahan akses di kemudian hari.

5. Evaluasi dan Analisis hasil

5.1 Distribusi Cluster

Setelah melakukan *clustering* menggunakan algoritma K-Means, lagu-lagu dalam dataset berhasil dikelompokkan ke dalam beberapa klaster yang merepresentasikan kategori musik berdasarkan kesamaan fitur audio. Setiap klaster ini kemudian diberi label interpretatif yang mencerminkan karakteristik dominan lagu-lagu di dalamnya, serta menunjukkan jumlah lagu yang termasuk dalam klaster tersebut. Berdasarkan hasil pengelompokan, ditemukan distribusi klaster sebagai berikut:

- *Instrumental / Cinematic*: Klaster ini berisi **23.971 lagu** yang cenderung memiliki karakteristik instrumental atau cocok untuk suasana sinematik. Lagu-lagu dalam klaster ini kemungkinan besar didominasi oleh melodi tanpa vokal atau memiliki *soundscape* yang luas.
- *Dance / Groove*: Klaster ini merupakan yang terbesar dengan **61.408 lagu**. Seperti namanya, klaster ini diisi oleh lagu-lagu dengan ritme yang kuat, *danceability* tinggi, dan cocok untuk suasana yang energik.
- *Upbeat / Alternative*: Dengan **49.576 lagu**, klaster ini kemungkinan besar mencakup lagu-lagu yang ceria, penuh semangat (*upbeat*), atau memiliki genre alternatif.
- *Vocal / Dramatic*: Klaster ini berisi **31.578 lagu**, menunjukkan kumpulan lagu yang menonjolkan elemen vokal yang kuat atau memiliki nuansa dramatis dan emosional.
- *Podcast / Spoken Word*: Klaster terkecil dengan **9.981 lagu**, yang secara spesifik mengindikasikan bahwa terdapat konten audio yang lebih berfokus pada narasi, seperti *podcast* atau *spoken word*, dibandingkan dengan musik tradisional.

Distribusi ini memberikan gambaran awal tentang segmen-segmen musik yang ada dalam dataset dan menjadi dasar bagi sistem rekomendasi untuk mengelompokkan lagu berdasarkan *mood* atau aktivitas pengguna.

5.2 Analisa cluster

Kualitas hasil *clustering* dievaluasi menggunakan tiga metrik utama yang memberikan gambaran komprehensif tentang seberapa baik klaster terbentuk. Berikut adalah hasil dan interpretasi dari masing-masing metrik:

- *Silhouette Score*: 0.6610 Mengukur seberapa mirip suatu objek dengan klaster tempatnya berada dibandingkan dengan klaster lain (rentang -1 hingga 1). Semakin mendekati 1 maka akan semakin baik, pada proyek ini nilai *Silhouette Score* 0.6610 menunjukkan klaster baik.
- *Davies-Bouldin Index*: 0.5171 Mengukur rasio rata-rata antara kesamaan di dalam klaster dengan jarak antar-klaster. Semakin mendekati 0 maka akan semakin baik, pada proyek ini nilai *Davies-Bouldin Index* 0.5171 menunjukkan klaster cukup baik.

- *Calinski-Harabasz Index*: 202660.7729 Mengukur rasio varians antar-klastrer dengan varians di dalam klastrer. Semakin tinggi nilainya maka akan semakin baik, pada proyek ini nilai *Calinski-Harabasz Index* 202660.7729 menunjukkan klastrer sangat baik.

6. Pembagian Tugas Kelompok

6.1 pembagian tugas secara teknis

Dalam proyek ini, pembagian tugas dilakukan untuk memanfaatkan keahlian masing-masing anggota tim secara efektif:

- Ahya Muhammad Abiyu Salam: Bertanggung jawab penuh atas pengembangan model *clustering* dan aplikasi Streamlit. Ini mencakup implementasi algoritma K-Means, *preprocessing* data, serta desain dan implementasi interface pengguna interaktif.
- Kenneth Owen: Fokus mencari dataset yang relevan dan komprehensif, serta penyusunan laporan akhir yang mendokumentasikan seluruh proses dan hasil proyek.
- Prince Lukas :membantu penulisan dalam pembuatan laporan.

7. Kesimpulan

7.1 Ringkasan temuan

Proyek pengembangan web pencarian musik berdasarkan *mood* dan aktivitas ini telah menghasilkan beberapa temuan kunci yang signifikan. Pertama, melalui proses *clustering* K-Means pada dataset *Spotify*, kami berhasil mengelompokkan ratusan ribu lagu ke dalam klastrer-klastrer yang bermakna berdasarkan kesamaan fitur audio mereka, seperti "Instrumental / Cinematic" dan "Dance / Groove". Pembentukan klastrer ini menjadi fondasi utama bagi sistem rekomendasi.

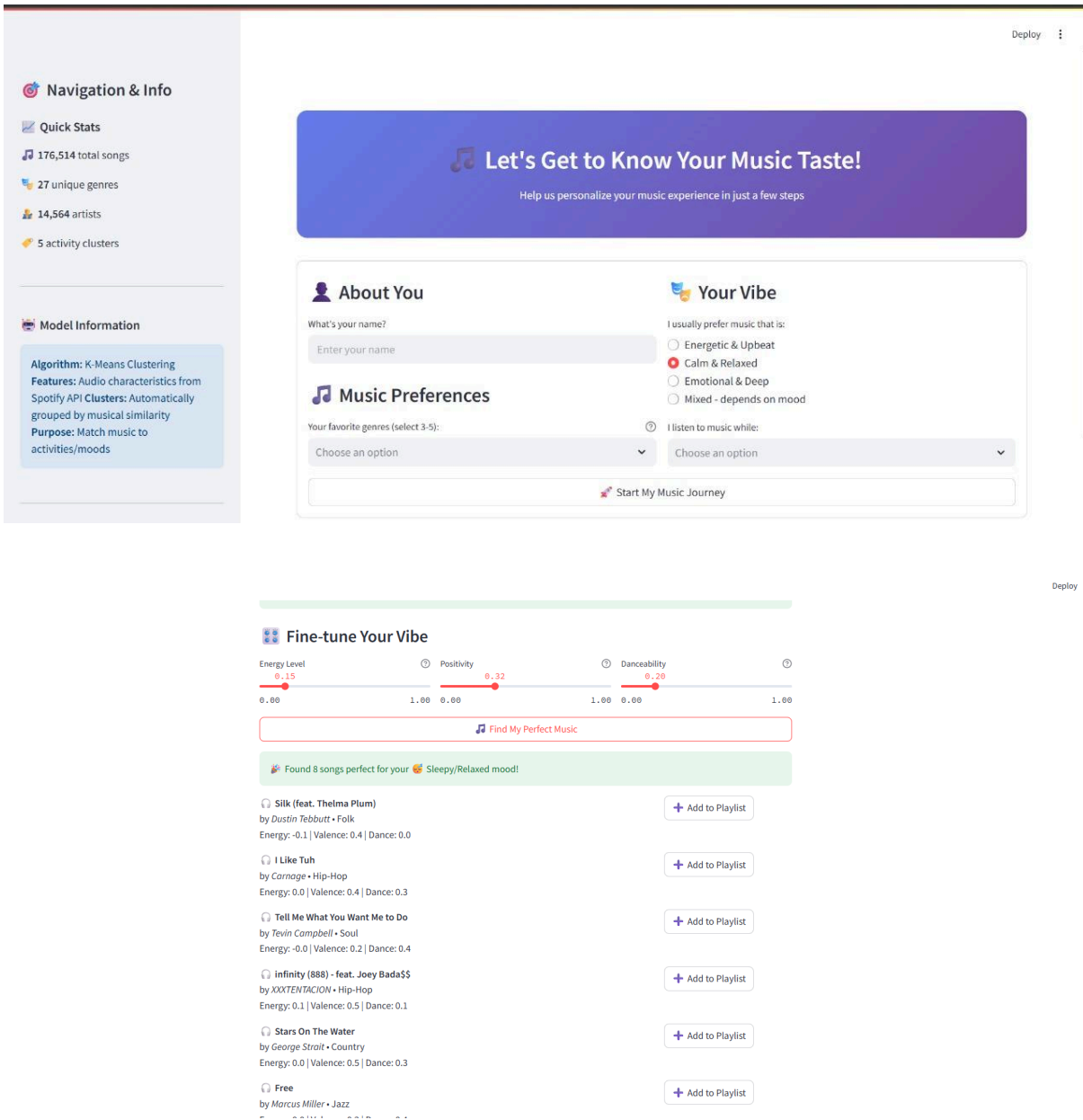
Kedua, evaluasi kualitas *clustering* menggunakan metrik *Silhouette Score* (0.6610), *Davies-Bouldin Index* (0.5171), dan *Calinski-Harabasz Index* (202660.7729) menunjukkan bahwa model *clustering* yang terbentuk memiliki kualitas yang baik hingga sangat baik. Hal ini mengindikasikan bahwa klastrer-klastrer yang dihasilkan kohesif di dalamnya dan terpisah dengan jelas satu sama lain, yang sangat penting untuk efektivitas rekomendasi.

Ketiga, pengembangan aplikasi web interaktif menggunakan Streamlit berhasil menyediakan antarmuka yang intuitif bagi pengguna. Fitur-fitur seperti *Mood Selector*, *Smart Recommendations*, *Song Classifier*, *Cluster Analysis*, dan *My Playlist* terintegrasi dengan

baik, memungkinkan pengguna untuk mencari lagu berdasarkan *mood* atau aktivitas, menyesuaikan preferensi audio, dan mengelola koleksi musik mereka. Secara keseluruhan, temuan ini menunjukkan bahwa pendekatan berbasis *clustering* dan pengembangan aplikasi web dapat secara efektif mengatasi permasalahan pencarian musik yang relevan sesuai preferensi pengguna.

8. Lampiran

8.1 Screenshot Demo



Welcome back, lukas! 🎵

Your music taste: Jazz, Pop, Hip-Hop • Preferred vibe: Calm & Relaxed

Mood Selector

Smart Recommendations

Song Classifier

Cluster Analysis

My Playlist

Visual Mood Selector

Tell us how you're feeling, and we'll find the perfect music for you!

How are you feeling right now?

Sleepy/Relaxed

Happy/Upbeat

Energetic/Workout

Sad/Melancholic

Focus/Study

Party/Social

Chill/Morning

Night/Intimate

Selected mood: 🌙 Sleepy/Relaxed

Fine-tune Your Vibe

Navigation & Info

lukas

Blues

Mixed - depends on mood

Reset Profile

Quick Stats

176,514 total songs

27 unique genres

14,564 artists

5 activity clusters

Model Information

Algorithm: K-Means Clustering

Features: Audio characteristics from Spotify API

Clusters: Automatically grouped by musical similarity

Purpose: Match music to activities/moods

Input Fitur Audio

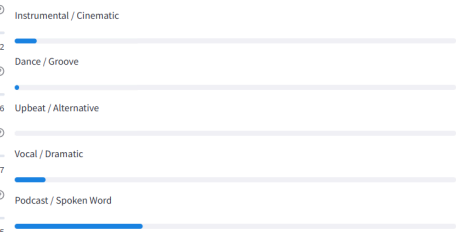


Prediksi Real-time

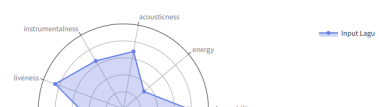
Prediksi: Podcast / Spoken Word

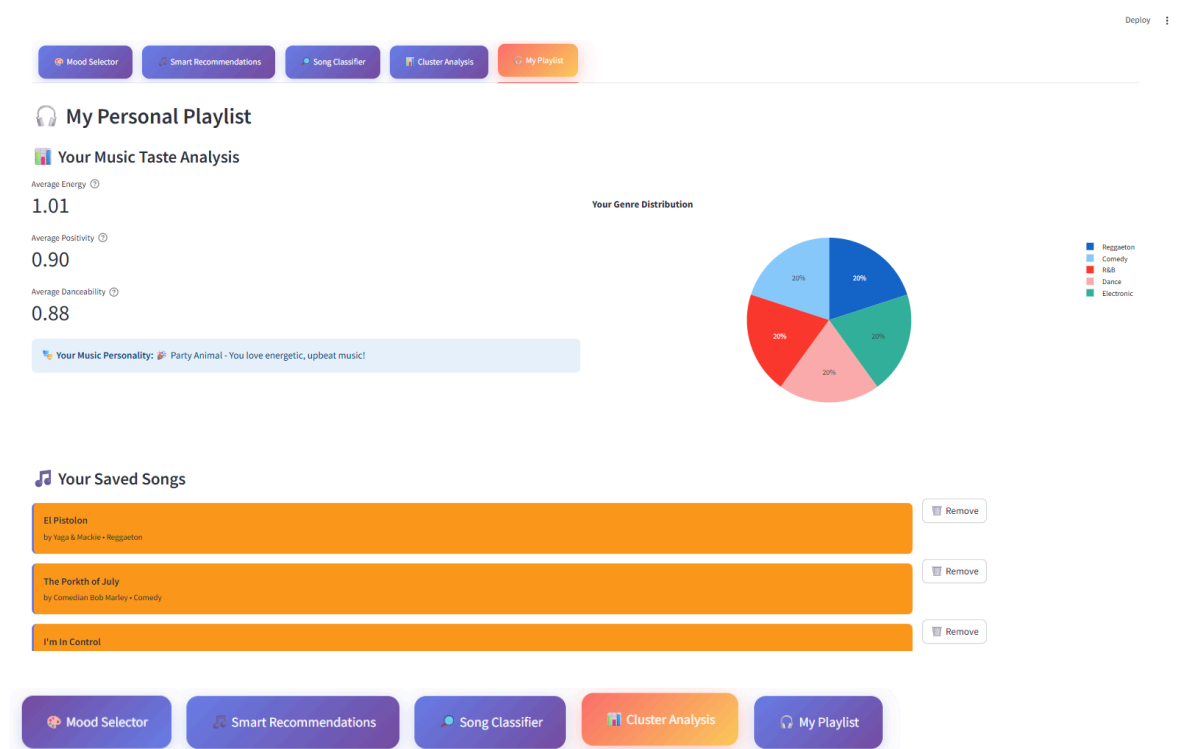
Confidence: 29.1%

Similarity Score



Profil Audio: Input Lagu

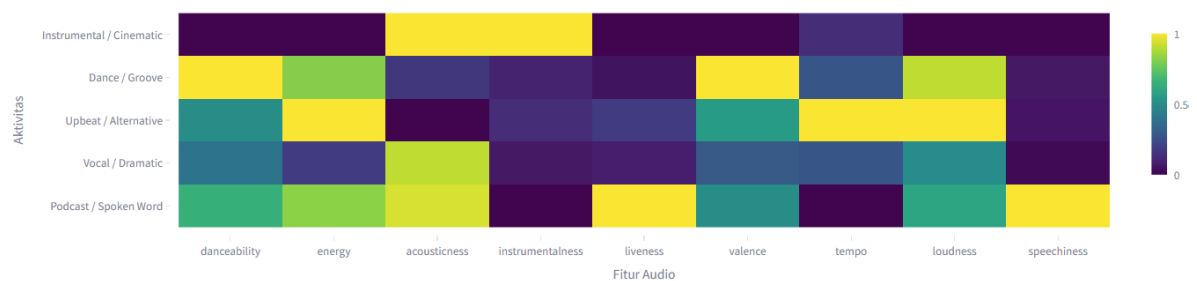




Advanced Cluster Analysis

Perbandingan Karakteristik Cluster

Heatmap Karakteristik Cluster (Normalized)



8.2 Contoh Output CSV

25	Movie	Jean Claude Pourquoi a 15C1P0LX	0	0.669	0.707	71299	0.696	0 C	0.197	-10.472	Major	0.028	114.752	4-Apr	0.853
26	Movie	Henri Salvi Quand je n 15nBPcHo	5	0.706	0.526	181441	0.298	0 F	0.143	-7.287	Major	0.0621	156.35	4-Apr	0.396

