

SKRIPSI

SISTEM REKOMENDASI MUSIK PADA SPOTIFY BERDASARKAN SUASANA HATI DAN UMUR MENGGUNAKAN METODE *K-MEANS*



OLEH
Wahyudin Nurdin
07352011027

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS KHAIRUN
TERNATE
2024

LEMBAR PENGESAHAN

SISTEM REKOMENDASI MUSIK PADA SPOTIFY BERDASARKAN SUASANA HATI DAN UMUR MENGGUNAKAN METODE K-MEANS

Oleh
Wahyudin Nurdin
07352011027

Skripsi ini telah disahkan
Tanggal 26 April 2024

Menyetujui
Tim Penguji

Ketua Penguji

MUHAMMAD SABRI AHMAD, S.Kom., M.Kom.
NIP. 198905092019031013

Pembimbing I

MUHAMMAD FHADLI, S.Kom., M.Sc.
NIP. 199611232023211012

Anggota Penguji

SAIFUL Do. ABDULLAH, S.T., M.T.
NIDN. 0018029002

Pembimbing II

Ir. ABDUL MUBARAK, S.Kom., M.T., IPM.
NIP. 198212062014041002

Anggota Penguji

ACHMAD FUAD, S.T., M.T.
NIP. 197606182005011001

Mengetahui/Menyetujui

Koordinator Program Studi
Informatika

ROSIHAN, S.T., M.Cs.
NIP. 197607192010121001



Dekan Fakultas Teknik
Universitas Khairun

LENDAR HARISON, S.T., M.T., CRP.
NIP. 197511302005011013

LEMBAR PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Wahyudin Nurdin
NPM : 07352011027
Fakultas : Teknik
Jurusan/Program Studi : Informatika
Judul Skripsi : Sistem Rekomendasi Musik Pada Spotify Berdasarkan Suasana Hati Dan Umur Menggunakan Metode *K- Means*.

Dengan ini menyatakan bahwa penulisan Skripsi yang telah saya buat ini merupakan hasil karya sendiri dan benar keasliannya. Apabila ternyata di kemudian hari penulisan Skripsi ini merupakan hasil plagiat terhadap karya orang lain, maka saya bersedia mempertanggung jawabkan sekaligus bersedia menerima sanksi berdasarkan aturan tata tertib di Universitas Khairun.

Demikian pernyataan ini saya buat dalam keadaan sadar dan tidak dipaksakan.

Penulis



Wahyudin Nurdin

The image shows a handwritten signature in black ink over a colorful revenue stamp. The stamp is from the Indonesian government (KORPRI) and has a value of 10,000 Rupiah. It includes the text 'METERAI TEMPEL' and a serial number 'CD71ALX210281399'. The signature is written in a cursive style.

HALAMAN PERSEMBAHAN

Dengan Rahmat Allah *Subhanahu Wa Ta'ala* yang Maha Pengasih lagi Maha Penyayang serta mengucapkan rasa syukur *Alhamdulillah* atas nikmat yang di berikan tanpa hentinya, penulis persembahkan skripsi ini kepada:

1. Jumra Nurdin dan Rosjannah Tikuratte, kedua orang tua hebat yang telah senantiasa memberikan kasih sayang, nasihat, motivasi dan dukungan, serta doa kepada penulis sehingga dapat menyelesaikan penulisan skripsi ini. Semoga beliau selalu bangga kepada penulis. Terima kasih mama dan papa.
2. Terima kasih kepada keluarga besar saya yang selalu memberikan dukungan dan dorongan tanpa henti sepanjang perjalanan ini.
3. Terima kasih untuk teman-teman Panda Squad yaitu Suci Ayu Maharani, Sasmita Hi. Sadek, Lisa Elisia Potale, Nurwana Iswan, Aprilia Silawane, Harlina Sapsuha, Nafra Aziqra Hi. A, Marhama Maynaka, Cindy Rahawaty S. Hipy, Muhammad Raihan Rizal, Nirwandi Bahri, Ferliy, dan Rinaldi Abdul karena telah membantu dan menyemangati saya dalam menyelesaikan skripsi ini.
4. Terima kasih kepada dosen dosen yang telah membantu dan mengarahkan saya dalam penulisan skripsi ini sehingga dapat menyelesaikan skripsi ini.
5. Terima kasih kepada kak Satria Dwi Surya S.T. sebagai kakak dan sekaligus pembimbing 3 saya yang selalu membantu dan mengarahkan saya selama ini.
6. Terima kasih kepada diri sendiri yang selalu semangat untuk menyelesaikan skripsi ini. Terima kasih telah mengandalkan diri sendiri untuk tetap kuat tanpa bergantung kepada orang lain dan tidak pernah memutuskan untuk menyerah dalam situasi apapun.

MOTTO

“Diam Untuk Berfikir, Bergerak Untuk Bangkit”

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Allah *Subhanaahu Wata'ala* dan sholawat serta salam kepada Nabi Muhammad *Shallallahu'alaihi Wassallam*, atas hidayah serta karunia-Nya sehingga penulis dapat menyelesaikan laporan skripsi dengan judul "Sistem Rekomendasi Musik Pada Spotify Berdasarkan Suasana Hati Dan Umur Menggunakan Metode *K- Means*".

Penyusunan skripsi penelitian ini dibuat untuk memenuhi salah satu persyaratan kelulusan yang ada di Universitas Khairun Ternate Fakultas Teknik Program Studi Informatika. Dalam penulisan laporan ini tentu tidak lepas dari dukungan dan dorongan dari banyak pihak, oleh karena itu penulis mengucapkan banyak terima kasih yang tulus kepada:

1. Bapak Dr. M. Ridha Ajam, S.H., M.Hum., selaku Rektor Universitas Khairun Ternate.
2. Bapak Endah Harisun, S.T., M.T., selaku Dekan Fakultas Teknik.
3. Bapak Rosihan, S.T., M.Cs., selaku Ketua Program Studi Informatika.
4. Bapak Muhammad Fhadli, S.Kom., M.Sc. selaku Pembimbing I, yang telah bersedia meluangkan waktunya untuk memberikan bimbingan, dukungan serta saran juga kritik yang sangat berarti bagi penulis dalam proses penyelesaian skripsi ini.
5. Bapak Ir. Abdul Mubarak, S.Kom., M.T., IPM., selaku Pembimbing II, yang telah bersedia meluangkan waktunya untuk memberikan bimbingan, dukungan serta saran juga kritik yang sangat berarti bagi penulis dalam proses penyelesaian skripsi ini.
6. Bapak Muhammad Sabri Ahmad, S.Kom., M.Kom., selaku Penguji I yang telah meluangkan waktu untuk menguji penulis dan memberikan saran serta arahan yang sangat berarti hingga terselesaikan skripsi ini.
7. Bapak Saiful Do. Abdullah, S.T., M.T., selaku Penguji II, yang telah bersedia meluangkan waktunya untuk menguji saya Ketika sidang, dukungan serta saran juga kritik yang sangat berarti bagi penulis dalam proses penyelesaian skripsi ini.
8. Bapak Achmad Fuad, S.T., M.T., selaku Penguji III, yang telah bersedia meluangkan waktu untuk menguji dan memberikan masukan perbaikan demi menyempurnakan laporan skripsi ini.
9. Kedua orang tua yang selalu memberikan semangat dan doa, serta dukungan dan perhatian, sehingga penulis dapat menyelesaikan laporan skripsi ini dengan baik.

10. Dan tak lupa ucapan terima kasih dan semangat kepada teman-teman seperjuangan panda *squad* yang saling bekerja sama, memberikan saran dan membantu satu sama lain.
11. Terima kasih kepada kak Satria Dwi Surya S.T. sebagai kakak dan sekaligus pembimbing 3 saya yang selalu membantu dan mengarahkan saya selama ini.
12. Dan semua semua pihak yang tidak dapat penulis sebutkan satu persatu yang telah membantu baik secara langsung maupun tidak langsung dalam menyelesaikan laporan skripsi ini.

Akhir kata penulis ucapkan rasa terima kasih kepada semua pihak dan apabila ada yang tidak disebutkan namanya penulis mohon maaf, dan kepada semua pihak yang telah membantu dalam penulisan ini semoga amal dan kebbaikannya mendapat balasan yang berlimpah dari Allah *Subhanaahu Wata'ala*.

Ternate, 10 Februari 2024

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
HALAMAN PERNYATAAN KEASLIAN	iii
HALAMAN PERSEMBAHAN.....	iv
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR.....	x
DAFTAR TABEL	xiv
ABSTRAK.....	xv
 BAB I PENDAHULUAN	
1.1. Latar Belakang	1
1.2. Rumusan Masalah.....	2
1.3. Batasan Masalah.....	3
1.4. Tujuan.....	3
1.5. Manfaat.....	3
1.6. Sistematika Penulisan	4
 BAB II TINJAUAN PUSTAKA	
2.1. Penelitian Terkait.....	5
2.2. Sistem Rekomendasi.....	9
2.3. <i>Content Based Filtering</i>	9
2.4. Rekomendasi Musik	10
2.5. Spotify	10
2.6. Spotify API	11
2.7. <i>Crawling Data</i>	12
2.8. <i>Unsupervised Learning</i>	12
2.9. Suasana Hati	13
2.9.1. Ciri-Ciri Suasana Hati.....	13
2.9.2. Jenis-Jenis Suasana Hati.....	14

2.10. <i>K-Means Clustering</i>	15
2.11. <i>Silhouette Coefficient</i>	16
2.12. Python.....	17
2.13. <i>Library Yang Digunakan</i>	18
2.14. <i>Streamlit</i>	20
2.15. <i>Flowchart</i>	20

BAB III METODE PENELITIAN

3.1. Waktu Penelitian.....	22
3.2. Alat dan Bahan	22
3.2.1. Perangkat Keras(<i>hardware</i>).....	22
3.2.2. Perangkat Lunak(<i>software</i>).....	22
3.3. Alur Penelitian	23
3.4. Metode Pengumpulan Data.....	25
3.5. Kategori Umur.....	27
3.6. Jenis Suasana Hati.....	27
3.7. <i>Preprocessing Data</i>	28
3.9. Implementasi Metode <i>K-Means</i>	29
3.10. Pengujian <i>Silhouette Coefficient</i>	31
3.11. Pembuatan Model Dan Pengembangan Sistem.....	31
3.12. Pengujian <i>Black Box</i>	32

BAB IV HASIL DAN PEMBAHASAN

4.1. Pengumpulan Data.....	33
4.2. <i>Preprocessing Data</i>	36
4.3. Menentukan Kategori Umur	38
4.4. Menentukan Kategori Suasana Hati	39
4.5. Pembuatan Model	90
4.6. Pembuatan Antar Muka.....	93
4.7. Hasil Evaluasi <i>Silhouette Score</i>	109
4.8. <i>Black Box Testing</i>	111
4.9. Analisis	112
4.9.1. Analisis Suasana Hati <i>Calm</i>	112

4.9.2.	Analisis Suasana Hati <i>Energic</i>	114
4.9.3.	Analisis Suasana Hati <i>Sad</i>	116
4.9.4.	Analisis Suasana Hati <i>Happy</i>	118
4.9.5.	Analisis Suasana Hati <i>Angry</i>	120

BAB V PENUTUP

5.1.	Kesimpulan	125
5.2.	Saran	126

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR GAMBAR

	Halaman
Gambar 3.1. Alur Penelitian.....	25
Gambar 3.2. <i>CLIENT_ID</i> dan <i>CLIENT_SECRET</i>	26
Gambar 3.3. Data Yang Akan Digunakan	26
Gambar 3.3. <i>Elbow Method</i>	29
Gambar 4.1. <i>CLIENT_ID</i> <i>CLIENT_SECRET</i>	34
Gambar 4.2. <i>Library</i> Yang Digunakan	34
Gambar 4.3. <i>Request Token Access</i>	35
Gambar 4.4. <i>Token Access</i>	35
Gambar 4.5. Kode Program Untuk Penarikan Data	35
Gambar 4.6. Memasukan Hasil <i>Crawling</i> data Ke Dalam <i>Dataset</i>	36
Gambar 4.7. <i>Dataset</i> Hasil <i>Crawling</i> Data.....	36
Gambar 4.8. Jumlah <i>Dataset</i>	38
Gambar 4.9. Mengecek Nilai Kosong Pada <i>Dataset</i>	38
Gambar 4.10. Mengecek Duplikat	38
Gambar 4.11. Hapus Duplikat	39
Gambar 4.12. Hasil <i>Preprocessing</i>	39
Gambar 4.13. <i>Library</i> Yang Digunakan	40
Gambar 4.14. Kode Program <i>Elbow Method</i>	41
Gambar 4.15. Hasil <i>Elbow Method</i>	41
Gambar 4.16. Kode Program <i>Clustering</i>	42
Gambar 4.17. Hasil <i>Clustering</i>	43
Gambar 4.18. Hasil Data <i>Cluster 0</i>	44
Gambar 4.19. Hasil Data <i>Cluster 1</i>	45
Gambar 4.20. Hasil Data <i>Cluster 2</i>	46
Gambar 4.21. Hasil Data <i>Cluster 3</i>	47
Gambar 4.22. Hasil Data <i>Cluster 4</i>	47
Gambar 4.23. Kode Program Untuk Menambah Hasil <i>Cluster</i> Ke <i>Dataset</i>	48
Gambar 4.24. Kode Program Menambah Hasil Seleksi Suasana Hati <i>Calm</i>	49

Gambar 4.25.	Hapus Kolom <i>Cluster</i>	49
Gambar 4.26.	Kode Program <i>Silhouette Score</i>	49
Gambar 4.27.	Hasil Skor <i>Silhouette</i>	50
Gambar 4.28.	Kode Program Menentukan Jumlah <i>Cluster</i>	51
Gambar 4.29.	Hasil <i>Elbow Method</i>	51
Gambar 4.30.	Kode Program <i>Clustering</i>	52
Gambar 4.31.	Visualisasi Hasil <i>Clustering</i>	52
Gambar 4.32.	Hasil Data <i>Cluster 0</i>	53
Gambar 4.33.	Hasil Data <i>Cluster 1</i>	54
Gambar 4.34.	Hasil Data <i>Cluster 2</i>	55
Gambar 4.35.	Hasil Data <i>Cluster 3</i>	56
Gambar 4.36.	Hasil Data <i>Cluster 4</i>	57
Gambar 4.37.	Hasil Data <i>Cluster 5</i>	58
Gambar 4.38.	Hasil Data <i>Cluster 6</i>	59
Gambar 4.39.	Hasil Data <i>Cluster 7</i>	60
Gambar 4.40.	Hasil Data <i>Cluster 8</i>	60
Gambar 4.41.	Hasil Data <i>Cluster 9</i>	61
Gambar 4.42.	Program Untuk Menambahkan Hasil <i>Cluster</i> Ke <i>Dataset</i>	62
Gambar 4.43.	Kode Program Menambah Hasil Seleksi Suasana Hati <i>Energic</i>	62
Gambar 4.44.	Kode Program <i>Silhouette</i>	63
Gambar 4.45.	Hasil Skor <i>Silhouette</i>	63
Gambar 4.46.	Kode Program Untuk Mencari n <i>Cluster</i>	64
Gambar 4.47.	Hasil <i>Elbow Method</i>	65
Gambar 4.48.	Kode Program <i>Clustering</i>	65
Gambar 4.49.	Visualisasi Hasil <i>Custering</i>	66
Gambar 4.50.	Hasil Data <i>Cluster 0</i>	67
Gambar 4.51.	Hasil Data <i>Cluster 1</i>	68
Gambar 4.52.	Hasil Data <i>Cluster 2</i>	69
Gambar 4.53.	Hasil Data <i>Cluster 3</i>	70
Gambar 4.54.	Program Untuk Menambahkan Hasil <i>Cluster</i> Ke <i>Dataset</i>	70
Gambar 4.55.	Program Menambah Hasil Seleksi Suasana Hati <i>Sad</i>	71

Gambar 4.56.	Kode Program <i>Silhouette Coefficient</i>	71
Gambar 4.57.	Hasil Skor.....	72
Gambar 4.58.	Kode Program <i>Elbow Method</i>	73
Gambar 4.59.	Hasil <i>Elbow Method</i>	73
Gambar 4.60.	Kode Program <i>Clustering</i>	74
Gambar 4.61.	Hasil <i>Clustering</i>	74
Gambar 4.62.	Hasil Data <i>Cluster 0</i>	76
Gambar 4.63.	Hasil Data <i>Cluster 1</i>	77
Gambar 4.64.	Hasil Data <i>Cluster 2</i>	78
Gambar 4.65.	Kode Program Untuk Menambah Hasil <i>Cluster</i> Ke <i>Dataset</i>	78
Gambar 4.66.	Kode Program Menambah Hasil Seleksi Suasana Hati <i>Happy</i>	79
Gambar 4.67.	Kode Program <i>Silhouette</i>	79
Gambar 4.68.	Hasil Skor <i>Silhouette</i>	80
Gambar 4.69.	Kode Program <i>Elbow Method</i>	81
Gambar 4.70.	Hasil <i>Elbow Method</i>	81
Gambar 4.71.	Kode Program <i>Clustering</i>	82
Gambar 4.72.	Hasil <i>Clustering</i>	82
Gambar 4.73.	Hasil Data <i>Cluster 0</i>	84
Gambar 4.74.	Hasil Data <i>Cluster 1</i>	85
Gambar 4.75.	Hasil Data <i>Cluster 2</i>	86
Gambar 4.76.	Hasil Data <i>Cluster 3</i>	87
Gambar 4.77.	Hasil Data <i>Cluster 4</i>	88
Gambar 4.78.	Kode Program Untuk Menambah Hasil <i>Cluster</i> Ke <i>Dataset</i>	88
Gambar 4.79.	Kode Program Menambah Hasil Seleksi Suasana Hati <i>Angry</i>	89
Gambar 4.80.	Kode Program <i>Silhouette</i>	89
Gambar 4.81.	Hasil Skor <i>Silhouette</i>	90
Gambar 4.82.	Program Dari Model Sistem	91
Gambar 4.83.	Definisi Fungsi	92
Gambar 4.84.	<i>Filtering Data</i>	92
Gambar 4.85.	Fungsi <i>Scaller</i> Dan Kesamaan Kosinus	92
Gambar 4.86.	<i>Features</i>	93

Gambar 4.87.	Pemanggilan Fungsi	93
Gambar 4.88.	<i>Library</i> Yang Digunakan	98
Gambar 4.89.	<i>Request Token</i>	98
Gambar 4.90.	Fungsi Pengambilan Gambar Album	98
Gambar 4.91.	API Spotify	99
Gambar 4.92.	Kategori Umur	99
Gambar 4.93.	Fungsi Rekomendasi	99
Gambar 4.94.	Fungsi <i>scaller</i> dan kesamaan kosinus	100
Gambar 4.95.	Membaca <i>Dataset</i>	100
Gambar 4.96.	<i>Title</i> Aplikasi	100
Gambar 4.97.	Fungsi <i>Button</i> Rekomendasi	101
Gambar 4.98.	Hasil <i>Running</i> Kelompok Umur bayi	101
Gambar 4.99.	Hasil <i>Running</i> Kelompok Umur Anak Anak	103
Gambar 4.100.	Hasil <i>Running</i> Kelompok Umur Remaja	105
Gambar 4.101.	Hasil <i>Running</i> Kelompok Umur Dewasa	107
Gambar 4.102.	Hasil <i>Running</i> Kelompok Umur Lansia	109
Gambar 4.103.	Distribusi Suasana Hati <i>Calm</i> Berdasarkan Kategori Umur	115
Gambar 4.104.	Distribusi Suasana Hati <i>Energic</i> Berdasarkan Kategori Umur	117
Gambar 4.105.	Distribusi Suasana Hati <i>Sad</i> Berdasarkan Kategori Umur	119
Gambar 4.106.	Distribusi Suasana Hati <i>Happy</i> Berdasarkan Kategori Umur	121
Gambar 4.107.	Distribusi Suasana Hati <i>Angry</i> Berdasarkan Kategori Umur	123

DAFTAR TABEL

	Halaman
Tabel 2.1. Penelitian Terkait	5
Tabel 2.2. Interpretasi Nilai <i>Silhouette Coefficient</i>	18
Tabel 2.3. <i>Flowchart</i>	22
Tabel 3.1. Perangkat Keras	25
Tabel 3.2. Perangkat Lunak.....	25
Tabel 3.3. Pusat <i>Cluster (Centroid)</i> Awal	31
Tabel 3.4. Interpretasi Nilai <i>Silhouette Coefficient</i>	33
Tabel 4.1. <i>Playlist</i> Yang Dipakai	37
Tabel 4.2. Daftar Lagu Pada Suasana Hati <i>Calm</i>	50
Tabel 4.3. Daftar Lagu Pada Suasana Hati <i>Energic</i>	63
Tabel 4.4. Daftar Lagu Pada Suasana Hati <i>Sad</i>	72
Tabel 4.5. Daftar Lagu Pada Suasana Hati <i>Happy</i>	80
Tabel 4.6. Daftar Lagu Pada Suasana Hati <i>Angry</i>	90
Tabel 4.7. Rekomendasi Kelompok Umur bayi	102
Tabel 4.8. Rekomendasi Kelompok Umur Anak-Anak	104
Tabel 4.9. Rekomendasi Kelompok Umur Remaja	106
Tabel 4.10. Rekomendasi Kelompok Umur Dewasa.....	107
Tabel 4.11. Rekomendasi Kelompok Umur Lansia	109
Tabel 4.12. Evaluasi <i>Silhouette Score</i>	111
Tabel 4.13. Pengujian <i>Black box</i>	112

ABSTRAK

SISTEM REKOMENDASI MUSIK PADA SPOTIFY BERDASARKAN SUASANA HATI DAN UMUR MENGGUNAKAN METODE K-MEANS

Wahyudin Nurdin¹, Muhammad Fhadli², Abdul Mubarak³
Program Studi Informatika, Fakultas Teknik, Universitas Khairun
Jl. Jati Metro, Kota Ternate Selatan

Email: ¹wahyudinurdin20@gmail.com, ²muhammadfhadli@gmail.com, ³amuba@unkhair.ac.id

Dalam era digital saat ini, layanan streaming musik seperti Spotify telah menjadi bagian penting dari kehidupan sehari-hari, menyediakan akses tak terbatas ke berbagai pilihan musik. Namun, tantangan utama yang dihadapi oleh pengguna adalah menemukan trek yang sesuai dengan suasana hati dan preferensi usia mereka dari katalog yang luas ini. Penelitian ini bertujuan untuk mengembangkan sistem rekomendasi musik yang dapat memberikan rekomendasi lebih tepat berdasarkan suasana hati dan usia pengguna menggunakan metode *K-Means clustering*. Skripsi ini menggunakan data dari Spotify API, melibatkan proses pengelompokan lagu berdasarkan karakteristik musik yang relevan dengan suasana hati dan usia pengguna. Melalui penerapan teknik *unsupervised learning*, sistem ini mampu menawarkan rekomendasi yang lebih personal dan relevan. Pengujian kualitas *cluster* dilakukan menggunakan *Silhouette Coefficient* untuk mengevaluasi efektivitas pengelompokan. Hasil penelitian menunjukkan bahwa sistem rekomendasi yang dikembangkan berhasil meningkatkan relevansi rekomendasi lagu dengan keakuratan yang signifikan. Penelitian ini tidak hanya mengkonfirmasi keefektifan penerapan *K-Means* dalam sistem rekomendasi musik tetapi juga memperluas potensi aplikasi metode *machine learning* dalam memahami preferensi pengguna dalam konteks musik.

Kata Kunci: Spotify, Rekomendasi Musik, *K-Means Clustering*, Suasana Hati, Usia, *Silhouette Coefficient*

MUSIC RECOMMENDATION SYSTEM ON SPOTIFY BASED ON MOOD AND AGE USING THE K-MEANS METHOD

In the digital era, music streaming services like Spotify have become an integral part of daily life, providing unlimited access to a wide range of music choices. However, a significant challenge faced by users is finding tracks that match their mood and age preferences from this vast catalog. This study aims to develop a music recommendation system that can provide more accurate recommendations based on user mood and age using the K-Means clustering method. This thesis utilizes data from the Spotify API, involving the clustering of songs based on musical characteristics relevant to user mood and age. By applying unsupervised learning techniques, this system can offer more personalized and relevant recommendations. The quality of the clusters is tested using the Silhouette Coefficient to evaluate the effectiveness of the clustering. The results of the study indicate that the developed recommendation system successfully improves the relevance of song recommendations with significant accuracy. This research not only confirms the effectiveness of applying K-Means in music recommendation systems but also expands the potential applications of machine learning methods in understanding user preferences in the context of music.

Keywords: *K-Means, Silhouette Coefficient, Spotify API, Elbow Method, Music Recommendation*

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dalam kehidupan sehari-hari, orang seringkali membutuhkan hiburan untuk menghilangkan kelelahan dan stres, dengan harapan dapat meningkatkan suasana hati mereka. Suasana hati adalah keadaan emosional yang ada dalam diri seseorang dan dapat berubah seiring berjalannya waktu. Suasana hati bisa berupa positif atau negatif, dan keduanya memiliki berbagai variasi, baik yang bersifat positif maupun negatif. Belakangan ini, musik telah dikenali sebagai elemen baru dalam metadata yang berkaitan dengan suasana hati (Lesmana, 2019).

Keadaan emosional memegang peranan sentral dalam pengalaman mendengarkan musik, sebab fluktuasi emosi dapat memengaruhi pilihan musik seseorang. Sama halnya, usia individu juga memiliki dampak terhadap selera musik yang mereka pilih. Dengan demikian, integrasi informasi mengenai suasana hati dan usia pengguna dalam sistem rekomendasi musik dapat menciptakan rekomendasi yang lebih tepat dan sesuai dengan preferensi mereka.

Pengguna layanan *streaming* musik seperti Spotify sering dihadapkan pada tantangan untuk menavigasi ribuan lagu yang tersedia untuk menemukan musik yang sesuai dengan suasana hati dan umur mereka. Dalam upaya meningkatkan pengalaman mendengarkan musik pengguna, penelitian ini bertujuan untuk mengembangkan sebuah sistem rekomendasi. Banyak sistem rekomendasi saat ini yang masih mengandalkan metode yang sederhana, seperti rekomendasi berdasarkan genre atau popularitas lagu.

Dalam penelitian ini, peneliti mengusulkan pengembangan sistem rekomendasi musik

yang lebih kebaruan, yaitu dengan mempertimbangkan dua faktor utama yaitu suasana hati dan umur pengguna. Untuk mencapai tujuan ini, peneliti akan menerapkan metode *K-Means* untuk mengelompokkan lagu-lagu berdasarkan fitur-fitur musik. Metode ini akan membantu dalam mengidentifikasi kelompok lagu yang memiliki karakteristik serupa dan sesuai dengan suasana hati pengguna, selain itu, peneliti juga menggunakan *content based filtering* untuk merekomendasikan.

Alasan orang mendengarkan musik antara lain untuk mengisi waktu senggang, menghilangkan kebosanan, memperbaiki emosi, dan menumbuhkan semangat. Ada pun suasana hati atau kondisi mental emosi yang dapat berpengaruh terhadap preferensi musik yang ingin kita dengar. Sejumlah besar musik digital beredar *online* saat ini. Dalam rentang waktu tiga bulan, sebanyak 17,6 miliar musik dan lebih dari 662.000 jam digunakan oleh 5.808 pengguna Spotify untuk mendengarkan musik melibatkan sejumlah besar pilihan individu (bukan hanya jenis musik apa, tetapi juga kapan, di mana, dan bagaimana mendengarkannya).

Penelitian ini berfokus pada pemanfaatan algoritma *K-Means clustering*, *K-Means clustering* merupakan sebuah metode dari *unsupervised learning* yang bertujuan untuk mempartisi peninjauan n ke kelompok k dimana tiap peninjauan dimiliki kelompok yang mempunyai nilai rata-rata terdekat (Buslim, 2019).

1.2. Rumusan Masalah

Berdasarkan latar belakang diatas, maka yang menjadi rumusan masalah adalah:

1. Bagaimana cara mengklasterisasi trek musik Spotify berdasarkan *mood* dan usia menggunakan algoritma *K-Means*?
2. Bagaimana menguji kualitas *clustering* yang telah dilakukan menggunakan pengujian

Silhouette Coefficient.

1.3. Batasan Masalah

Berdasarkan rumusan masalah yang diangkat, penelitian ini memiliki batasan masalah sebagai berikut:

1. Penelitian ini hanya terfokus pada pengembangan sistem rekomendasi musik berbasis suasana hati dan umur pengguna dan hanya berlaku untuk pengguna *platform* Spotify.
2. Penelitian ini hanya akan menggunakan data musik yang tersedia dari Spotify.
3. *Output* dari sistem ini hanya menampilkan rekomendasi musik berdasarkan masukan suasana hati dan umur pengguna.
4. *Output* yang ditampilkan oleh sistem hanya berupa judul lagu yang ditampilkan dalam *streamlit*.
5. Akun spotify yang digunakan pada penelitian ini hanya akun gratis, Jadi ada batasan dalam mengakses aplikasi.

1.4. Tujuan

Berdasarkan dari hasil rumusan masalah yang telah di uraikan oleh penulis, tujuan dari penelitian ini yaitu untuk mengetahui hasil klasterisasi trek musik Spotify berdasarkan *mood* dan usia menggunakan algoritma *K-Means*.

1.5. Manfaat

Manfaat dari penelitian ini diantaranya sebagai berikut:

1. Meningkatkan pengalaman mendengarkan musik pengguna Spotify dengan memberikan rekomendasi lagu-lagu yang sesuai dengan suasana hati dan usia pengguna.

2. Menyediakan rekomendasi musik yang lebih personal dan sesuai dengan preferensi pengguna berdasarkan faktor suasana hati dan usia, sehingga meningkatkan kepuasan pengguna dalam menggunakan layanan spotify.

1.6. Sistematika Penulisan

Untuk memudahkan pembahasan dalam skripsi ini, sistematika penulisan dibagi menjadi 5 (lima) bab yang terdiri dari:

BAB I PENDAHULUAN

Terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Memaparkan teori-teori yang didapat dari sumber-sumber relevan untuk digunakan sebagai panduan dalam penelitian serta penyusunan skripsi ini.

BAB III METODE PENELITIAN

Bab ini membahas tentang metode penelitian yang telah dilakukan oleh penulis dengan permasalahan yang diangkat.

BAB IV HASIL PEMBAHASAN

Bab ini membahas tentang skripsi dari penelitian yang telah dilakukan yang terdiri dari *Elbow Method*, *clustering*, dan *Silhouette Coefficient*, dan pengujian *Black box*.

BAB V PENUTUP

Memuat kesimpulan dari skripsi yang telah dilakukan, dan saran untuk penelitian selanjutnya mengenai topik terkait.

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Terkait

Pada penelitian ini terdapat beberapa sumber referensi serta acuan dari hasil penelitian sebelumnya yang berkaitan dengan penelitian yang akan dilakukan saat ini. Adapun beberapa penelitian terkait yang menjadi dasar penulisan pada penelitian ini antara lain dapat dilihat pada tabel 2.1.

Tabel 2.1. Penelitian Terkait

No.	Nama dan Tahun	Judul	Hasil Penelitian
1.	(Lesmana, 2019)	Perancangan Aplikasi Rekomendasi Lagu Berdasarkan Mood Dan Aktivitas Pengguna Memanfaatkan <i>Wearable Device</i> Dan Spotify Api	Berdasarkan hasil perancangan tersebut maka perancangan aplikasi sudah sesuai dengan yang diharapkan dan selanjutnya untuk dilakukan implementasi.
2.	(Brata , 2021)	<i>Mood Classification of Balinese Songs with the K-Means Clustering Method Based on the Audio-Content Feature</i>	<i>Classification with 3 clusters resulted in an accuracy of 40 %. This accuracy is higher than the classification with 4 clusters, resulting in an accuracy of only 32 %. This is due to the data of respondents known for being happy, relaxed, and sad. However, it can be neglected that the features of the Spotify API and the song classification using K-Means clustering with these features on the object of Balinese song research resulted in the highest of 40 %. As a suggestion for further research, use the features of the Spotify API in addition to the energy and valence features because the Spotify API provides various datasets features.</i>
3.	(Abdul 'alim, 2021)	Pengembangann Aplikasi Rekomendasi Musik Berdasarkan	Hasil pengujian sistem rekomendasi musik berdasarkan emosi pengguna mempunyai nilai

		Emosi Pengguna Pada Platform <i>Android</i>	hasil <i>valid</i> (100%) saat diuji menggunakan pengujian <i>blackbox testing</i> . Untuk pengujian Sistem <i>Usability Scale</i> (SUS) dengan nilai hasil 84. Nilai tersebut tergolong dalam kategori <i>excellent</i> . Yang terakhir adalah pengujian tingkat akurasi, hasil <i>output</i> sistem dibandingkan dengan apa yang disarankan oleh pakar dengan nilai akurasi sebesar 80%.
4.	(Rohmah, 2021)	Komparasi Algoritma <i>K-Means</i> , <i>K-Medoid</i> , <i>Agglomerative Clustering</i> Terhadap <i>pGenre</i> Spotify	Penelitian komparasi 3 algoritma data mining yaitu <i>kmeans</i> , <i>kmedoid</i> , <i>agglomerative hierarchical</i> menghasilkan kesimpulan bahwa metode algoritma yang paling baik untuk proses pengelompokan genre adalah <i>agglomerative hierarchical</i> dengan metode <i>average linked</i> dengan klaster terbaik 3 atau 4 Klaster paling direkomendasikan adalah sebesar 3 klaster karena klaster ke 4 hanya berisi 1 anggota saja. Pada pengelompokan 3 klaster, klaster ke-1 berisi 2833 anggota, klaster ke-2 berisi 145 anggota, dan klaster ke-3 berisi 21 anggota.
5.	(Musyarofah, 2022)	Klasifikasi Top 50 Spotify Tahun 2010-2019 Menggunakan Metode <i>K-Means Clustering</i>	Dilaksanakannya penelitian ini untuk mengetahui hasil klasifikasi Top 50 Spotify Tahun 2010-2019 menggunakan metode <i>K-Means Clustering</i> . Perkembangan teknologi yang semakin pesat inilah yang menimbulkan pengaruh pada kehidupan masyarakat, seperti munculnya media layanan <i>streaming</i> musik yang memudahkan masyarakat dalam mendengarkan musik. Klasifikasi menggunakan <i>dataset</i> lagu <i>pop</i> terpopuler yang dianalisis menggunakan metode <i>K-Means</i> untuk mengetahui <i>top</i> genre. <i>Dataset</i> berasal dari Website

			<i>Kaggle.com</i>. Dalam penerapannya, penelitian ini dipengaruhi oleh <i>valence</i> (valensi lagu), genre lagu, <i>tempo</i>, <i>Beats Per Minute</i>/ketukan (BPM), dan sebagainya. Dalam menemukan <i>cluster</i> yang sesuai, penelitian diperoleh dari hasil analisis <i>cluster pie chart</i>, klaster diagram batang, dan titik <i>koordinat</i> hubungan valensi lagu. Penelitian dengan metode <i>K-Means</i> memudahkan pengguna dalam mengidentifikasi lagu artis terpopuler dan tidak terpopuler pada tahun tertentu.
6.	(Nurhalimah, 2022)	Analisis Prediksi <i>Mood</i> Genre Musik <i>Pop</i> Menggunakan Algoritma <i>K-Means</i> dan C4.5	Pada penelitian ini didapatkan pengelompokan musik terhadap suasana hati (<i>mood</i>) dengan menerapkan metode data <i>mining</i> menggunakan algoritma <i>K-Means</i> dan C4.5, dimana pada <i>cluster</i> yang ditentukan yaitu 4 <i>cluster</i> berdasarkan penggolongan 4 jenis suasana hati (<i>mood</i>) model <i>Thayer's</i> yaitu <i>angry</i>, <i>sad</i>, <i>happy</i> dan <i>cheerful</i>. Hasilnya adalah mampu mencapai akurasi sebesar 91,9% <i>class precision Angry</i> 100%, <i>class precision Sad</i> 90.83%, <i>class precision Happy</i> 82.63% dan <i>class precision Cheerfull</i> 92.97%. Serta <i>class recall Angry</i> 91.06%, <i>class recall Sad</i> 92.04%, <i>class recall Happy</i> 100% dan <i>class recall Cheerful</i> 87.82% dari hasil penelitian dapat terlihat kecenderungan lagu dengan <i>mood cheerful</i> yang paling banyak diakses. Sebagai saran untuk penelitian selanjutnya, gunakan fitur Spotify API selain fitur <i>danceability</i>, <i>energy</i>, <i>tempo</i> dan <i>valence</i> seperti <i>loudness</i>, <i>liveness</i>, <i>speechiness</i> karena Spotify API menyediakan berbagai fitur <i>dataset</i>.

Hubungan penelitian terkait pada tabel 2.1 dengan penelitian ini yang membuat sistem rekomendasi musik pada Spotify berdasarkan suasana hati dan umur menggunakan metode *K-Means*, bertujuan untuk mengembangkan sistem rekomendasi musik yang mempertimbangkan suasana hati dan umur pengguna Spotify. Metode klusterisasi yaitu *K-Means* digunakan untuk mengkategorikan lagu berdasarkan fitur-fitur musiknya dan *content based filtering* digunakan untuk memberikan rekomendasi yang lebih akurat, dan sistem ini akan diimplementasikan kedalam aplikasi web *streamlit*.

Penelitian ini berbeda dengan penelitian-penelitian sebelumnya yang telah disebutkan, dengan menambahkan umur sebagai salah satu atribut dan menggunakan *streamlit* untuk implementasi sistemnya.

Berikut adalah perbandingan penelitian saya dengan penelitian sebelumnya:

1. Lesmana (2019): Mengembangkan aplikasi rekomendasi lagu berdasarkan suasana hati dan aktivitas menggunakan perangkat *wearable* dan Spotify API. Sedangkan penelitian saya berbeda dengan memasukkan faktor umur pengguna dan mengimplementasikan sistem pada situs web.
2. Rahmawati (2016): Menganalisis klusterisasi menggunakan metode *K-Means* dan *Hierarchical Clustering* untuk dokumen tesis Departemen Kimia. Sedangkan penelitian saya sangat berbeda karena fokus pada rekomendasi musik menggunakan data Spotify.
3. Abdul 'alim (2021): Mengembangkan aplikasi rekomendasi musik berdasarkan emosi pengguna untuk *platform* Android. Sedangkan penelitian saya berbeda dengan memasukkan faktor umur dan menggunakan *platform* berbasis web.
4. Rohmah Zaidah (2021): Membandingkan algoritma *K-Means*, *K-Medoid*, dan

Agglomerative Clustering untuk genre Spotify. Sedangkan penelitian saya berbeda dengan fokus pada suasana hati dan umur untuk rekomendasi musik.

5. Musyarofah (2022): Mengklasifikasikan 50 lagu teratas Spotify menggunakan *K-Means Clustering*. Dan pada penelitian saya menambahkan dimensi umur dan suasana hati pengguna, yang tidak ada dalam penelitian Musyarofah.
6. Nurhalimah (2022): Menganalisis prediksi suasana hati pada musik *pop* menggunakan algoritma *K-Means* dan C4.5. sedangkan penelitian saya berbeda dengan memasukkan umur sebagai variabel dan mengimplementasikan sistem pada *platform web Streamlit*.

2.2. Sistem Rekomendasi

Sistem rekomendasi adalah sebuah aplikasi yang berfungsi untuk memprediksi suatu item yang menarik bagi *user*, contohnya rekomendasi film, musik, buku, berita dan lain sebagainya. Ada dua tipe metode yang diterapkan pada sistem rekomendasi, yaitu *collaborative filtering* dan *content based filtering* (Badriyah, 2018).

2.3. Content Based Filtering

Sistem rekomendasi dengan metode *content based filtering* merekomendasikan item yang mirip dengan item sebelumnya yang disukai atau dipilih oleh pengguna. Kemiripan item dihitung berdasarkan pada fitur-fitur yang ada pada item yang dibandingkan. Metode ini bersifat *user independence*, tidak bergantung pada situasi apakah item tersebut merupakan item baru (yang belum pernah dipilih oleh pengguna manapun) maupun bukan item baru. Jika seorang pengguna telah memesan suatu menu hidangan pada kategori tertentu maka sistem akan mencoba merekomendasikan menu hidangan dengan kategori serupa yang juga tersedia di restoran lain yang mungkin akan disukai juga oleh pengguna tersebut.

Kelemahan dari metode *content based filtering* adalah terbatasnya rekomendasi hanya pada item-item yang mirip sehingga tidak ada kesempatan untuk mendapatkan item yang tidak terduga. Akan tetapi, Pazzani dan Billsus dalam jurnalnya menjelaskan memang benar bahwa tidak menutup kemungkinan pengguna yang melihat atau membeli boneka juga telah membeli film dewasa, oleh sebab itu hal-hal seperti ini sebaiknya tidak direkomendasikan (Wijayanto, 2019).

2.4. Rekomendasi Musik

Rekomendasi musik merujuk pada proses pemberian saran atau rekomendasi lagu kepada pengguna berdasarkan preferensi musik mereka. Tujuan utama dari rekomendasi musik adalah membantu pengguna menemukan lagu-lagu baru yang sesuai dengan selera mereka dan meningkatkan pengalaman pribadi mereka dalam mendengarkan musik.

Sistem rekomendasi musik memanfaatkan berbagai metode dan algoritma untuk menganalisis data musik, preferensi pengguna, serta informasi lainnya. Data musik yang dianalisis dapat mencakup elemen-elemen musik seperti *tempo*, genre, instrumen, atau lirik.

Sistem rekomendasi merupakan salah satu cabang dari *machine learning* dengan jenis yang lebih spesifik. Tidak seperti model *machine learning* pada umumnya yang memprediksi suatu nilai berdasarkan *input* baru yang dihasilkan dari riwayat data sebelumnya, sistem rekomendasi lebih berfokus untuk menyediakan daftar produk, dalam hal ini musik yang dirasa akan disukai oleh *user* (Anggoro, 2022).

2.5. Spotify

Spotify merupakan *platform streaming* musik dan siaran berbasis di Swedia, tepatnya di Stockholm. Diluncurkan pada tanggal 7 Oktober 2008, Spotify memberikan akses kepada penggunanya untuk menikmati jutaan lagu dan konten lain yang dibuat oleh seniman dari

seluruh dunia. Pengguna dapat mengakses layanan ini melalui aplikasi *desktop* maupun aplikasi *mobile*, dan mereka memiliki opsi untuk memainkan lagu secara *streaming* atau mengunduhnya agar bisa dinikmati secara *offline*. Spotify memiliki dua jenis layanan utama, yaitu versi gratis dan berbayar. Pengguna gratis dapat memutar musik dengan keterbatasan iklan yang ditampilkan di antara lagu-lagu. Di sisi lain, pengguna berlangganan dengan layanan Spotify *Premium* dapat menikmati fitur-fitur tambahan, termasuk pemutaran tanpa iklan, kualitas audio yang lebih tinggi, kemampuan untuk mengunduh lagu demi mendengarkannya secara *offline*, serta sejumlah fitur lainnya (Support.Spotify.Com, 2020).

2.6. Spotify API

Spotify API, atau yang disebut sebagai *Spotify Application Programming Interface*, merupakan suatu alat pemrograman yang memberikan akses kepada pengembang untuk meraih data dan kemampuan yang tersedia di dalam layanan Spotify. Dengan memanfaatkan Spotify API, para pengembang dapat mengakses informasi terkait musik seperti data *album*, *artist*, lagu, serta *Playlist*, dan juga fitur-fitur musik seperti analisis audio dan mesin rekomendasi.

Pemanfaatan Spotify API membuka banyak peluang bagi pengembang untuk menciptakan aplikasi dan layanan inovatif yang mampu meningkatkan pengalaman musik pengguna. Beberapa contoh penerapan Spotify API termasuk pembuatan aplikasi musik, integrasi dengan perangkat keras seperti perangkat *wearable (smartwatch)*, dan pengintegrasian dengan *platform* lain seperti media sosial.

Spotify *Developers* menyediakan API (*application programming interface*) untuk berinteraksi dengan aplikasi Spotify pada *platform desktop* dan *mobile* dalam bentuk Spotify SDK (*software development kit*). *Library* yang tersedia pada SDK ini untuk akses memutar

musik atau koneksi ke Spotify dan mengontrol pemutaran musik dan mengakses metadata (Rachmanda, 2020).

2.7. *Crawling Data*

Crawling adalah suatu teknik yang digunakan untuk mengumpulkan informasi yang ada dalam web. *Crawling* bekerja secara otomatis, dimana informasi yang dikumpulkan berdasarkan atas kata kunci yang diberikan oleh pengguna.

2.8. *Unsupervised Learning*

Unsupervised learning adalah salah satu tipe algoritma *machine learning* yang digunakan untuk menarik kesimpulan dari *datasets* yang terdiri dari *input data labeled response*. Metode *unsupervised learning* yang paling umum adalah analisa *cluster*, yang digunakan pada analisa data untuk mencari pola-pola tersembunyi atau pengelompokan dalam data (Buslim, 2019).

Dalam konteks pengelompokan lagu-lagu berdasarkan fitur-fitur musik yang relevan, *unsupervised learning* digunakan untuk mengidentifikasi pola dalam data musik dan mengklasifikasikan lagu-lagu ke dalam kelompok yang memiliki karakteristik yang serupa. Fitur-fitur musik ini dapat mencakup aspek-aspek seperti *tempo*, durasi, kekuatan vokal, genre, dan lainnya. Dengan memanfaatkan algoritma *unsupervised learning*, seperti pengelompokan (*clustering*), kita dapat menggolongkan lagu-lagu berdasarkan kesamaan mereka dalam hal fitur-fitur ini.

Clustering adalah proses mengelompokkan atau penggolongan objek berdasarkan informasi yang diperoleh dari data yang menjelaskan hubungan antar objek dengan prinsip untuk memaksimalkan kesamaan antar anggota satu kelas dan meminimumkan kesamaan antar kelas/*cluster*. *Clustering* membagi data ke dalam grup-grup yang mempunyai objek

yang karakteristiknya sama (Rahmawati, 2016).

2.9. Suasana Hati

Mood atau suasana hati adalah kondisi emosional yang terdapat dalam diri seseorang dan dapat berubah seiring waktu. *Mood* bisa positif dan bisa juga negatif dan keduanya memiliki banyak jenis baik yang positif maupun yang negatif. Berkaitan dengan lagu, *mood* telah diidentifikasi sebagai tipe atau aspek metadata baru musik dalam beberapa tahun terakhir.

Suasana hati memiliki dampak yang signifikan pada kehidupan sehari-hari seseorang. *Mood* dapat memengaruhi pemikiran, tingkat energi, motivasi, serta interaksi sosial. Contohnya, ketika seseorang merasa bahagia atau bersemangat, mereka mungkin menjadi lebih produktif dan cenderung ingin bersosialisasi dengan orang lain (Lesmana, 2019).

2.9.1. Ciri-Ciri Suasana Hati

Menurut (Gaschke 2008), menyatakan terdapat dua macam jenis ciri-ciri suasana hati yaitu:

1. Suasana hati positif

Suasana hati positif adalah suasana hati dalam keadaan senang(bahagia, bersemangat, gembira, ceria), suasana hati dalam keadaan penuh cinta (penuh kasih, perhatian, penuh syukur), suasana hati dalam keadaan tenang (teduh, puas, biasa/netral, tenang, santai/relaks), suasana hati dalam keadaan semangat (aktif, segar).

2. Suasana hati negatif

Suasana hati negatif adalah suasana hati dalam keadaan cemas (gelisah, gugup,

cemas, takut), suasana hati dalam keadaan marah (letih, ngantuk, kecewa, sedih, tak berdaya, hampa, putus asa, jenuh), suasana hati dalam keadaan sedih (suram, sendu, haru).

2.9.2. Jenis-Jenis Suasana Hati

Emosi atau suasana hati didefinisikan sebagai perasaan yang muncul dalam diri seorang manusia sebagai respon dari situasi tertentu. Perasaan ini bisa mempengaruhi pikiran, persepsi dan perilaku seseorang. Sebagai contoh, saat kamu merasa senang, umumnya kamu pasti akan tersenyum, tertawa, riang dan bisa berinteraksi secara positif dengan orang lain. Sebaliknya, ketika marah, kamu mungkin mengeluarkan ekspresi wajah yang garang, berperilaku agresif dan denyut nadi meningkat. Perasaan ini bisa riang hingga intens. Perasaan ini juga bisa berubah dengan cepat atau bertahan untuk jangka waktu tertentu, setiap orang bisa mengalami berbagai jenis emosi seperti berikut:

1. Sedih

Perasaan sedih biasanya muncul ketika kamu mengalami kehilangan, kekecewaan, atau rasa putus asa. Saat mengalaminya, tak jarang seseorang akan mengeluarkan air matanya.

2. Senang

Sukacita adalah perasaan gembira, puas dan bahagia. Perasaan ini muncul ketika kita merasa senang, berhasil mencapai tujuan, atau mengalami momen menyenangkan.

3. Santai

Santai ditandai dengan ketenangan, keadaan pikiran yang damai dan kurangnya ketegangan atau stres. Saat seseorang merasa santai, tubuh dan pikiran mereka

cenderung dalam keadaan istirahat dan pemulihan. Suasana hati ini sering kali dihubungkan dengan perasaan yang puas dan tenang.

4. Marah

Suasana hati ini sering kali muncul sebagai respon terhadap situasi yang dianggap tidak adil, mengancam, atau merugikan. Ketika seseorang marah, mereka mungkin mengalami peningkatan denyut jantung, ketegangan otot, perubahan ekspresi wajah dan perilakunya lebih agresif seperti berteriak atau memukul.

5. Semangat

Perasaan energik ditandai dengan tingkat energi yang tinggi, vitalitas, dan kegairahan. Saat seseorang merasa energik, mereka biasanya memiliki dorongan yang kuat untuk melakukan aktivitas dan menghadapi tantangan dengan semangat dan antusiasme. Hal ini bisa terlihat dari peningkatan stamina, kekuatan, dan ketahanan. Emosi yang energik sering dikaitkan dengan produktivitas tinggi, kreativitas, dan motivasi.

2.10. *K-Means Clustering*

Algoritma *K-Means* adalah algoritma yang mempartisi data ke dalam *cluster – cluster* sehingga data yang memiliki kemiripan berada pada satu *cluster* yang sama dan data yang memiliki ketidaksamaan berada pada *cluster* yang lain. Sarwono mengemukakan secara lebih detail, algoritma *K-Means* adalah sebagai berikut:

1. Menentukan *k* sebagai jumlah kluster yang ingin di bentuk.
2. Membangkitkan nilai *random* untuk pusat *cluster* awal (*centroid*) sebanyak *k*.
3. Menghitung jarak setiap data input terhadap masing – masing *centroid* menggunakan rumus jarak *Euclidean* (*Euclidean Distance*) hingga ditemukan jarak yang paling dekat

dari setiap data dengan *centroid*. Berikut adalah rumus 2.1 persamaan *Euclidian Distance*:

$$d(x_i, \mu_i) = \sqrt{(x_i - \mu_i)^2} \dots\dots\dots(2.1)$$

dengan $d(x_i, \mu_i)$ adalah jarak antara *cluster* x dengan pusat *cluster* μ pada kata ke i , x_i adalah bobot kata ke i pada *cluster* yang ingin dicari jaraknya, μ_i bobot kata ke i pada pusat *cluster*.

4. Mengklasifikasikan setiap data berdasarkan kedekatannya dengan *centroid*. (jarak terkecil).
5. Memperbaharui nilai *centroid*. Nilai *centroid* baru di peroleh dari rata-rata *cluster* yang bersangkutan dengan menggunakan rumus 2.2:

$$C_k = \frac{1}{n_k} \sum d_i \dots\dots\dots(2.2)$$

Dimana:

n_k = jumlah data dalam *cluster*

d_i = jumlah dari nilai jarak yang masuk dalam masing-masing *cluster*.

6. Melakukan perulangan dari langkah 2 hingga 5, sampai anggota tiap *cluster* tidak ada yang berubah.

Jika langkah 6 telah terpenuhi, maka nilai pusat *cluster* (y_j) pada iterasi terakhir akan digunakan sebagai parameter untuk melakukan klasifikasi data (Rahmawati, 2016).

2.11. *Silhouette Coefficient*

Silhouette Coefficient digunakan untuk melihat kualitas dan kekuatan klaster, seberapa baik suatu objek ditempatkan dalam suatu klaster. Tahapan perhitungan *Silhouette Coefficient* adalah sebagai berikut (Shoolihah, 2017):

1. Hitunglah rata-rata jarak dari suatu data misalkan i dengan semua data lain dalam

satu klaster (ai) dengan menggunakan rumus 2.3.

$$a(i) = \frac{1}{|A|-1} \sum_{j \in A, j \neq i} d(i, j) \dots\dots\dots(2.3)$$

dengan j adalah data lain dalam satu *cluster* A dan d(i,j) adalah jarak antara data i dan j.

2. Hitunglah rata-rata jarak dari data i tersebut dengan semua data diklaster lain, dan diambil nilai terkecilnya dengan menggunakan rumus 2.4.

$$d(i, C) = \frac{1}{|A|} \sum_{j \in C} d(i, j) \dots\dots\dots(2.4)$$

dengan d(i,C) adalah jarak rata-rata data i dengan semua objek pada *cluster* lain C dimana $A \neq C$.

3. Rumus 2.5 *Silhouette Coefficient* adalah:

$$s(i) = (b(i) - a(i)) / \max(a(i), b(i)) \dots\dots\dots(2.5)$$

Dengan s(i) adalah semua rata-rata pada semua kumpulan data.

Berikut standar atau penafsiran nilai *Silhouette Coefficient* ditunjukkan dalam tabel 2.2.

Tabel 2.2 Interpretasi Nilai *Silhouette Coefficient* (Struyf, 1996)

<i>Silhouette Coefficient</i>	Interpretasi
0.71 – 1.00	Struktur yang dihasilkan kuat
0.51 – 0.70	Struktur yang dihasilkan baik
0.26- 0.50	Struktur yang dihasilkan lemah
≤ 0.25	Tidak terstruktur

2.12. Python

Python merupakan sebuah bahasa pemrograman tingkat tinggi yang dibuat oleh Guido Van Rossum dan dirilis pada tahun 1991 Python juga merupakan bahasa yang sangat populer belakangan ini. Selain itu Python juga merupakan bahasa pemrograman

yang multi fungsi salah satunya pada bidang *Machine Learning* dan *Deep Learning* (Alfarizi, 2023).

Bahasa pemrograman Python merupakan bahasa pemrograman yang terbaik untuk bidang *Machine Learning* dan *Deep Learning* karena Python memiliki beberapa keunggulan yaitu: kemudahan penulisan sintaks, dukungan komunitas yang besar, memiliki *library* yang lengkap dan juga populer (Alfarizi, 2023).

2.13. *Library* Yang Digunakan

Dalam penelitian ini peneliti menggunakan beberapa *library* yang sudah disediakan oleh Python untuk mempermudah melakukan analisis data dan pengembangan model (Hermanto, 2023). Beberapa *library* yang digunakan dalam penelitian ini antara lain:

1. *Pandas* adalah *library* analisis data yang menyediakan struktur data yang cepat, fleksibel, dan ekspresif, dirancang untuk memudahkan dalam memanipulasi data, seleksi data, pembersihan data serta membaca data dari berbagai sumber seperti CSV, *Excel* dan lainnya.
2. *Numpy* adalah *library* inti untuk komputasi ilmiah di Python. Ini menyediakan dukungan untuk *array* berdimensi besar, yang merupakan struktur data penting dalam pengolahan data dan komputasi numerik. *Numpy* memperkaya ekosistem Python dengan alat yang mampu melakukan operasi matematika dan aljabar linier yang kompleks dengan cepat dan efisien.
3. *Matplotlib* adalah *library* Python untuk membuat visualisasi data statis, animasi, dan interaktif. Ini memberikan API yang sangat disesuaikan, yang memungkinkan pengguna untuk membuat *plot* sesuai dengan kebutuhan spesifik mereka. Dari histogram sederhana hingga *plot* 3D, *Matplotlib* dapat digunakan untuk menghasilkan

berbagai jenis grafik dengan kontrol penuh atas elemen-elemen seperti warna, label, dan ukuran. Ini adalah alat yang sangat penting untuk eksplorasi data dan presentasi hasil analisis.

4. *Kmeans* adalah algoritma *clustering* yang terkenal, tersedia di dalam modul *sklearn.cluster* dari *library scikit-learn*. Algoritma ini bertujuan untuk membagi *dataset* menjadi *k* kelompok atau jumlah *cluster* sedemikian rupa sehingga anggota dalam setiap kelompok lebih serupa satu sama lain dibandingkan dengan kelompok lain. Ini sangat berguna dalam analisis data eksploratif dan pemahaman struktur data yang tersembunyi.
5. *Scikit-learn* atau *sklearn*, adalah *library* Python yang kaya fitur untuk *machine learning*. Menyediakan alat untuk hampir setiap aspek *machine learning*, termasuk regresi, klasifikasi, pengelompokan, pemilihan model, dan pengurangan dimensi. Ini dirancang untuk berintegrasi dengan mudah dengan *library* Python lainnya seperti *Numpy* dan *Pandas*.
6. *Pickle* dapat digunakan untuk menyimpan dan memuat model mesin pembelajaran (*machine learning*) dalam Python. Ini sering digunakan untuk menyimpan model setelah dilatih sehingga Anda dapat menggunakannya kembali tanpa harus melatih ulang setiap kali Anda menjalankan program.
7. *Streamlit* adalah *library* Python yang memungkinkan pengembang untuk membuat aplikasi web interaktif dengan mudah tanpa memerlukan pengetahuan mendalam tentang pengembangan web. Dengan menggunakan *streamlit*, pengembang dapat dengan cepat membuat aplikasi web yang memiliki elemen-elemen interaktif seperti formulir, grafik, dan tampilan data. *Framework* ini telah dirancang untuk membuat

pengembangan aplikasi web menjadi lebih sederhana dan cepat, sering digunakan dalam konteks prototipe, demonstrasi, dan visualisasi data.

2.14. *Streamlit*

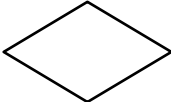
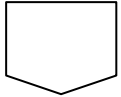
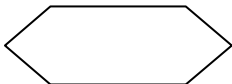
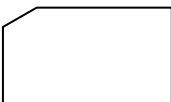
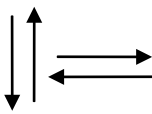
Pada penelitian ini digunakan *graphical user interface* (GUI) *Streamlit*. *Streamlit* adalah kerangka kerja web yang ditujukan untuk menyebarkan model dan visualisasi dengan mudah menggunakan bahasa Python, yang cepat dan minimalis tetapi juga memiliki tampilan yang cukup baik serta ramah pengguna. Tersedia *widget* bawaan untuk masukan pengguna, seperti pengunggahan gambar, penggeser, masukan teks, dan elemen *hypertext markup language* (HTML) lain yang sudah dikenal. Setiap kali pengguna berinteraksi dengan aplikasi *Streamlit*, skrip Python dijalankan kembali dari atas ke bawah.

Hal ini merupakan sebuah konsep penting yang perlu diingat saat mempertimbangkan berbagai status aplikasi yang akan dipilih. *Streamlit* dapat dijalankan pada editor Anaconda serta bahasa Python seri 3.7 ke atas, tetapi tidak mendukung pada editor Jupyter Notebook, sehingga harus dikonversi ke editor Pycharm atau Visual Code. Tampilan beranda pada aplikasi *Streamlit* dapat dipisahkan menjadi dua bagian, yaitu *buttons*, untuk pemilihan menu, serta tampilan *visual chart* (Hastomo, 2022).

2.15. *Flowchart*

Flowchart adalah bagan yang menunjukkan alur atau alur dalam suatu program atau prosedur sistem secara logis. *Flowchart* (bagan alir) adalah sebuah ilustrasi berupa diagram alir dari algoritma-algoritma yang ada dalam suatu program, yang menyatakan arah aliran dari program tersebut (Muhamad, 2018). Berikut fungsi-fungsi simbol yang sering digunakan untuk menggambarkan suatu algoritma dalam bentuk *flowchart* (diagram alir) dapat dilihat pada tabel 2.3.

Tabel 2.3 Simbol *Flowchart*

No	Simbol	Nama	Fungsi Simbol
1.		"Terminal"	Awal atau akhir suatu program (Prosedur).
2.		"Output/Input"	Proses <i>input</i> atau <i>output</i> terlepas dari jenis perangkat.
3.		"Process"	Proses operasional komputer.
4.		"Decision"	Untuk menunjukkan bahwa suatu kondisi tertentu mengarah pada dua kemungkinan, ya/tidak.
5.		"Connector"	Koneksi penghubung proses ke proses lain pada halaman yang sama.
6.		"Offline Connector"	Koneksi penghubung dari satu proses ke proses lain di halaman lain.
7.		"Predefined Process"	Mewakili ketentuan penyimpanan untuk diproses untuk memberikan awal harga.
8.		"Punched Card"	<i>Input</i> berasal dari kartu atau <i>output</i> ditulis ke kartu.
9.		"Documen"	Mencetak <i>output</i> dalam format dokumen (melalui <i>printer</i>).
10.		"Flow"	Menyatakan jalannya arus suatu proses.

BAB III

METODE PENELITIAN

3.1. Waktu Penelitian

Jadwal penelitian dibuat untuk membantu peneliti dalam mengatur waktu dalam masing-masing bagian dari proses penelitian. Penelitian ini dilakukan oleh peneliti direncanakan selama 6 bulan yaitu selama semester 7 dan 8.

3.2. Alat dan Bahan

Adapun alat dan bahan yang digunakan pada penelitian ini, terdiri dari dua komponen utama yaitu perangkat keras(*hardware*) dan perangkat lunak(*software*). Keduanya berperan penting dalam menjalankan penelitian. Berikut adalah penjelasan lebih lanjut tentang keduanya:

3.2.1. Perangkat Keras(*hardware*)

Adapun spesifikasi perangkat keras (*hardware*) yang digunakan pada penelitian ini sebagai penunjang proses pembuatan sistem rekomendasi musik selama penelitian berlangsung. Spesifikasi *hardware* dapat dilihat pada tabel 3.1.

Tabel 3.1 Perangkat Keras

Jenis	Spesifikasi
<i>Processor</i>	AMD Ryzen 3 3200U with Radeon Vega Mobile Gfx 2.60 GHz
RAM	8 GB DDR4
SSD	512 GB
<i>Sistem Type</i>	64-bit <i>operating</i> sistem

3.2.2. Perangkat Lunak(*software*)

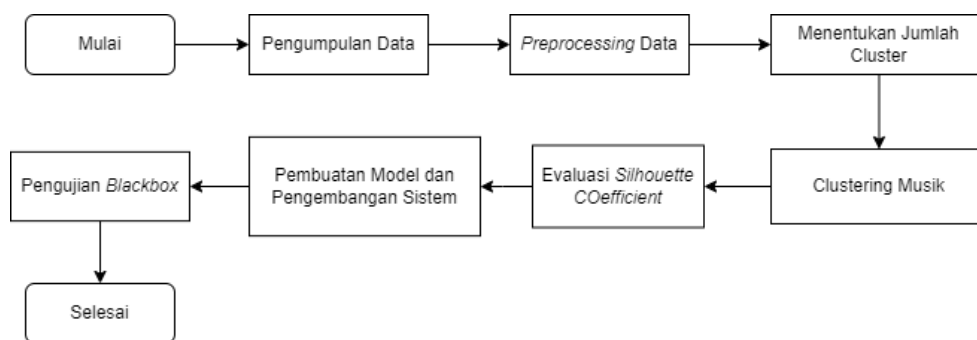
Selain kebutuhan *hardware*, dibutuhkan juga *software* untuk pembuatan sistem. Spesifikasi perangkat lunak (*software*) dapat dilihat pada tabel 3.2.

Tabel 3.2 Perangkat Lunak

Jenis	Type	Keterangan
Sistem Operasi	<i>Windows 11 Pro Education</i>	Digunakan selama penelitian
<i>Text Editor</i>	<i>Visual Studio Code</i>	<i>Code editor</i> yang digunakan untuk membuat atau membangun sistem
Bahasa Pemrograman	Python versi 3.10	Bahasa pemrograman yang digunakan untuk penulisan kode program
<i>Tools</i> desain sistem	Draw.io	Aplikasi untuk mendesai diagram alir (<i>flowchart</i>)

3.3. Alur Penelitian

Tahapan penelitian yang akan dilakukan dapat dilihat pada gambar 3.1.



Gambar 3.1 Alur Penelitian

1. Pengumpulan Data

Pada tahap ini data yang digunakan merupakan data dari aplikasi Spotify, peneliti melakukan pengambilan data atau *crawling* data menggunakan API Spotify dari website *developer.spotify.com* menggunakan bahasa pemrograman python.

2. Preprocessing Data

Preprocessing data merupakan tahapan dalam memproses *dataset* agar menjadi *dataset* yang siap digunakan, tahap *preprocessing* yang dilakukan pada penelitian ini yaitu data *cleaning*, tujuan dari data *cleaning*, agar tidak ada data yang kosong dan tidak ada data yang sama dengan data yang lain.

3. Penentuan jumlah *cluster*

Dalam proses ini untuk menentukan banyak *cluster* yang akan ditetapkan atau di *cluster* nanti dengan menggunakan *Elbow Method* untuk menentukan jumlah *cluster*.

4. *Clustering* / Pengelompokan Lagu

Pengelompokan lagu adalah proses pengelompokan berdasarkan kesamaan atau karakteristik tertentu yang dimiliki oleh lagu lagu tersebut. Tujuan utama dari pengelompokan lagu adalah untuk mengidentifikasi pola atau hubungan antara lagu-lagu sehingga kita dapat mengkategorikannya ke dalam kelompok-kelompok yang memiliki kesamaan tertentu.

5. *Silhouette Coefficient*

Melakukan pengecekan seberapa baik nilai *cluster* yang telah dilakukan dengan menggunakan metode metrik evaluasi *Silhouette Coefficient*.

6. Pembuatan Model Sistem

Pada tahap ini, pembuatan model sistem merujuk pada proses pembuatan model yang dimana model itu dibuat menggunakan bahasa pemrograman Python dan dibantu dengan beberapa *library* yang disediakan oleh python untuk mempermudah pembuatan model sistem, setelah membuat model sistem selanjutnya yaitu pembuatan implementasi model sistem ke website dengan menggunakan Bahasa pemrograman Python dan *library streamlit*.

7. Pengujian *Blackbox*

Pada tahap ini, peneliti perlu melakukan pengujian, dan menggunakan pengujian *blackbox*, tujuannya adalah untuk memastikan bahwa sistem tersebut berfungsi sebagaimana mestinya, dan memastikan tidak ada *bug* atau *error* dari sistem yang

dibuat.

3.4. Metode Pengumpulan Data

Pada tahap pengumpulan data ini, peneliti menggunakan metode *crawling* data menggunakan API Spotify, dan proses pengambilan data ini memerlukan API key yaitu *CLIENT_ID* dan *CLIENT_SECRET* yang didapat setelah mendaftar di website *developer.spotify.com*.

```
CLIENT_ID = "8a056f31514c4db2a8b2048086f6e3ef"
CLIENT_SECRET = "0fcffe48f39e4440a4fb68c77d42d5bc"
```

Gambar 3.2 *CLIENT_ID* dan *CLIENT_SECRET*

Setelah mendapatkan *client_id* dan *client_secret*, maka kita akan menentukan data apa yang akan kita ambil. Data ini dapat bersifat *real-time* atau historis, dan dapat dikustomisasi sesuai kebutuhan, dan harus dipilih dengan cermat sesuai dengan kebutuhan dan tujuan penelitian.

Sebelum melakukan pengambilan data, kita harus menentukan data apa saja yang akan kita ambil, jika sudah di tentukan maka kita langsung implementasikan programnya untuk melakukan pengambilan data, jika sudah maka datanya seperti pada gambar 3.3.

```
track_id,track_name,artist_name,danceability,energy,key,loudness,mood,speechiness,acousticness,instrumentalness,
0tCqX7pASaZeHPC4FhAwe8,Brahms Lullaby,Sleepy Shepherd,0.39,0.0161,1,-29.263,1,0.0398,0.993,0.674,0.133,0.5,143.1
2DGSyztXs9oqYUy5tqxFS8,Twinkle Twinkle Little Star,Tommy Twinkle,0.55,0.0642,1,-24.304,1,0.123,0.996,0.955,0.09
0K00hxPmW5pF0XjgWzRf0j,Goodnight Little Dreamer,Little Dreamer,0.309,0.00781,2,-30.09,1,0.0557,0.996,0.916,0.06
4tbnZK9nKvuyCrqyu8o0eG,Hush Little Baby,Sleepy Shepherd,0.554,0.0204,11,-27.418,1,0.0528,0.995,0.897,0.0922,0.3
5YmJaZEsUoI8EFZm4q2f8,Moonlight Walk,Orion The Owl,0.479,0.0189,10,-24.228,1,0.0514,0.993,0.953,0.121,0.287,78
1EvERSC5XOhncvGdhSHjil,Golden Sunrise,Chillboy,0.561,0.0154,4,-24.016,1,0.0577,0.996,0.941,0.0958,0.415,124,113
4GphHeTcJqIEb1Y6jRMyaG,Lavender's Blue,Sleepy Shepherd,0.532,0.0323,5,-26.907,1,0.0441,0.993,0.904,0.134,0.344,i
5N180kHAIQjkc12A1I90h,Wake Up,Bluedon,0.618,0.00838,0,-21.242,1,0.055,0.987,0.968,0.0965,0.521,80.246,115406,4
5znZqW4EAeyv6Dz8H5zSE2,You Are My Sunshine,Sleepy Shepherd,0.494,0.0225,0,-28.313,1,0.0361,0.995,0.914,0.112,0.
0nyFIAU92dxCDb0JFHP9E,Rock A Bye Baby,Sleepy Shepherd,0.598,0.0318,7,-22.212,1,0.0614,0.995,0.847,0.127,0.589,i
7DQ5SHU3eHYLM0htGKz3y,Are You Sleeping,Sleepy Shepherd,0.518,0.0256,0,-27.621,1,0.0452,0.995,0.84,0.0917,0.367
61pb4EVYeX94RQb2mo3AUr,My Bonnie Lies Over The Ocean,Sleepy Shepherd,0.338,0.0155,11,-29.874,1,0.0445,0.995,0.9
1B0SN5PHT51NMy4kv1qrAy,Baa Baa Black Sheep,Sleepy Shepherd,0.538,0.0333,4,-23.151,1,0.0366,0.994,0.874,0.0821,0
6mZQXPV4n8FBDEYGF808K,Itsy Bitsy Spider,Sleepy Shepherd,0.508,0.0172,4,-26.652,1,0.0349,0.993,0.917,0.0855,0.3
01b3jCDs8aA4D6Rglx8yhX,Trust Me,Sleepy Shepherd,0.536,0.0128,10,-24.603,1,0.0985,0.995,0.968,0.0875,0.567,108.9
1K7jgK79uuxS0pbB4151pM,Daydreaming,Sleepy Shepherd,0.413,0.0354,8,-31.037,1,0.0489,0.995,0.927,0.112,0.767,109.1
3AaXL2157K6yazdQ150t4I,Peaceful Sunset,Sleepy Shepherd,0.546,0.0192,0,-29.264,1,0.189,0.996,0.968,0.073,0.674,7
1VvL29pvdKxhKp3jJwE11Q,Night Sky,Sleepy Shepherd,0.416,0.021,10,-28.794,1,0.0499,0.995,0.928,0.109,0.641,116.47
1Dmu3G1Jut8z1QzeLyXomK,Singing For You,Sleepy Shepherd,0.506,0.0146,1,-29.6,0,0.0704,0.996,0.968,0.116,0.656,12
6sa1YuE21HNPv0uif0TH0I,Fade Away,Sleepy Shepherd,0.383,0.0115,6,-31.196,1,0.0598,0.996,0.863,0.0953,0.476,62.09
7Gkc6FFPU5V6wDYo6bAsvE,Misty Ocean,Sleepy Shepherd,0.423,0.0107,1,-28.762,1,0.0672,0.995,0.938,0.109,0.535,64.8
4gbA6AMrZEdAfjMrN6M1US,Rain Outside,Sleepy Shepherd,0.29,0.00734,5,-31.395,0,0.0412,0.992,0.943,0.103,0.214,78.
2xsQgizMzkkLFa90gv7Ftr,Changes,Sleepy Shepherd,0.427,0.00291,8,-31.539,1,0.095,0.996,0.96,0.129,0.826,69.442,57
4qRF3F11TK4RqFtUmd4ZHB,Mindfulness,Sleepy Shepherd,0.574,0.0163,10,-25.224,1,0.0737,0.995,0.962,0.134,0.651,77.:
```

Gambar 3.3 Data Yang Akan Digunakan

Data yang peneliti tampilkan merupakan data yang sudah di ambil menggunakan

Spotify API, data diatas berjumlah 17 kolom dan 1242 baris. Dan dalam penelitian ini terdapat beberapa fitur dalam *dataset* yang akan digunakan untuk mencari *cluster* jenis suasana hati, berikut merupakan data data yang akan digunakan.

1. *Name*: kolom ini berisi judul musik, kolom ini akan dipakai oleh sistem yang merekomendasikan musik dari judulnya.
2. *Loudness*: kolom ini merupakan tentang tingkat kekuatan dan intensitas suara dari sebuah rekaman musik, biasanya diukur dalam desibel(dB).
3. *Popularity*: kolom ini merupakan nilai dari skor popularitas lagu, biasanya berdasarkan seberapa sering lagu tersebut diputar.
4. *Danceability*: skor yang mengukur seberapa cocok lagu untuk menari, berdasarkan *tempo*, *ritme*, stabilitas *beat*, dan kekuatan *beat*.
5. *Acousticness*: skor yang mengindikasikan seberapa akustik lagu tersebut seperti piano, biola dan lainnya.
6. *Energy*: Skor yang mengukur tingkat energi lagu, biasanya berdasarkan dinamika, volume, dan kebisingan rekaman, skor *energy* yang tinggi biasanya terdengar lebih cepat dan keras, biasanya skor ini ada pada musik pesta atau acara dansa. Sedangkan skor *energy* yang rendah cenderung lebih tenang dan lembut.
7. *Instrumentalness*: skor yang menunjukkan seberapa besar kemungkinan lagu yang tidak memiliki *vocal*.
8. *Liveness*: skor yang mengukur kehadiran penonton dalam rekaman.
9. *Valence*: skor yang mengukur positifitas musik, skor dengan nilai tinggi menunjukkan musik yang lebih positif dan bahagia.
10. *Speechiness*: skor yang mendeteksi keberadaan kata-kata yang diucapkan. Skor

yang lebih tinggi menunjukkan adanya lebih banyak kata yang diucapkan, seperti dalam *rap* atau *talk show*.

11. *Tempo*: kecepatan atau kaju dari sebuah komposisi dan biasanya diukur dalam BPM (*beats per minute*)

3.5. Kategori Umur

Pada penelitian ini untuk fitur umur akan ditentukan berdasarkan *Playlist-Playlist* dengan kategori umur tertentu yang ada di Spotify, dan kategori umur yang akan dipakai pada penelitian menggunakan sumber dari kementrian Kesehatan republik Indonesia(Kesehatan, 2024), berikut tabel 3.3.

Tabel 3.3 Kategori Umur

Umur	Kategori
< 5 tahun	Bayi dan balita
5 - 9 tahun	Anak-anak
10 – 18 tahun	Remaja
19 – 59 tahun	Dewasa
60+ tahun	Lansia

Pada tabel 3.3 merupakan kategori umur yang dipakai ada penelitian ini, dan kategori umur ini diambil berdasarkan kategori umur yang dibuat oleh Kementerian Kesehatan Republik Indonesia.

3.6. Jenis Suasana Hati

Dan pada penelitian ini ada 5 jenis suasana hati yang akan digunakan berikut adalah tabel 3.4 jenis suasana hati:

Tabel 3.4 Jenis Suasana Hati

Suasana Hati	Kondisi
<i>Calm/santai</i>	Tenang dan Santai
<i>Energic/semangat</i>	Penuh Energy dan Motivasi

<i>Sad/sedih</i>	Kehilangan atau Kekecewaan
<i>Happy/Senang</i>	Gembira dan Puas
<i>Angry/Marah</i>	Kesal atau Tidak Puas

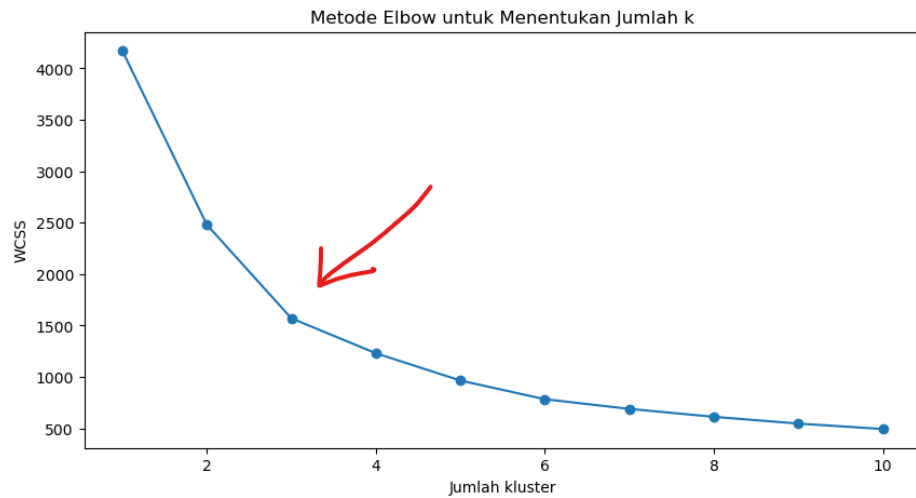
Pada tabel 3.4 merupakan jenis-jenis suasana hati yang akan dipakai pada penelitian ini.

3.7. *Preprocessing Data*

Preprocessing data merupakan tahapan dalam memproses *dataset* agar menjadi *dataset* yang siap digunakan, tentunya hal ini diperlukan untuk mengurangi data yang tidak diperlukan. Dalam penelitian ini saya melakukan data *cleaning*, yang dimana data akan di periksa apakah seluruh data tersebut memiliki nilai atau tidak, jika ada data yang tidak memiliki nilai maka akan segera dihapus, dan *dataset* juga akan diperiksa apakah di dalam *dataset* terdapat data dengan nilai yang sama atau duplikat, maka data tersebut akan segera dihapus, maka dari itu tahapan ini sangat penting.

3.8. *Penentuan Jumlah Cluster*

Elbow Method merupakan suatu metode yang digunakan untuk menghasilkan informasi dalam menentukan jumlah *cluster* terbaik dengan cara melihat persentase hasil perbandingan antara jumlah *cluster* yang akan membentuk siku pada suatu titik. Metode ini memberikan ide/gagasan dengan cara memilih nilai *cluster* dan kemudian menambah nilai *cluster* tersebut untuk dijadikan model data dalam penentuan *cluster* terbaik. Hasil persentase yang berbeda dari setiap nilai *cluster* dapat ditunjukkan dengan menggunakan grafik sebagai sumber informasinya. Jika nilai *cluster* pertama dengan nilai *cluster* kedua memberikan sudut dalam grafik atau nilainya mengalami penurunan paling besar maka nilai *cluster* tersebut yang terbaik (Putu, 2015). Berikut adalah gambar 3.3 contoh penentuan jumlah *cluster* menggunakan *Elbow Method*.

Gambar 3.3 *Elbow Method*

3.9. Implementasi Metode *K-Means*

Pada dasarnya analisis *cluster* menggunakan metode *K-Means* memiliki algoritma sebagai berikut (Sutejo, 2020):

1. Menentukan banyaknya *k cluster* yang di inginkan.
2. Menentukan pusat *cluster (centroid)* awal secara *random* dari objek data sebanyak 4 *cluster*. Untuk lebih jelasnya dapat dilihat pada tabel 3.5.

Tabel 3.5 Pusat *Cluster (Centroid)* Awal

Centroid	Cluster awal	
	Energy	Valence
<i>Cluster 1</i>	0.7	0.6
<i>Cluster 2</i>	0.8	0.1
<i>Cluster 3</i>	0.1	0.1
<i>Cluster 4</i>	0.4	0.5

3. Menentukan jarak terdekat dari setiap objek pengamatan dengan *centroid* yang telah ditentukan menggunakan jarak *Euclidean* dengan rumus 3.1:

$$d(x_i, x_j) = \sqrt{\sum_{l=1}^n (x_{il} - x_{jl})^2} \dots\dots\dots (3.1)$$

Keterangan:

$d(x_i, x_j)$ = Jarak antar objek ke-i dengan objek ke-j, dimana $i = 1, 2, \dots, n$ dan $j = 1, 2, \dots, n$

x_{il} = Nilai objek ke-i pada variabel k

x_{jl} = Nilai objek ke-j pada variabel k

n = Banyaknya variabel data

Perhitungan jarak dari data ke-1 terhadap titik pusat *cluster* adalah sebagai berikut dibawah ini:

$$d(1,1) = \sqrt{(0.7 - 0.7)^2 + (0.6 - 0.6)^2} = \sqrt{(0^2) + (0^2)} = \sqrt{0} = 0$$

$$\begin{aligned} d(1,2) &= \sqrt{(0.7 - 0.8)^2 + (0.6 - 0.1)^2} = \sqrt{(-0.1)^2 + (0.5)^2} \\ &= \sqrt{0.01 + 0.25} = \sqrt{0.26} = 0.510 \end{aligned}$$

$$\begin{aligned} d(1,3) &= \sqrt{(0.7 - 0.1)^2 + (0.6 - 0.1)^2} = \sqrt{(0.6)^2 + (0.5)^2} = \sqrt{0.36 + 0.25} \\ &= \sqrt{0.61} = 0.781 \end{aligned}$$

$$\begin{aligned} d(1,4) &= \sqrt{(0.7 - 0.4)^2 + (0.6 - 0.5)^2} = \sqrt{(0.3)^2 + (0.1)^2} = \sqrt{0.09 + 0.01} \\ &= \sqrt{0.1} = 0.316 \end{aligned}$$

Dan seterusnya dilakukan perhitungan jarak untuk data ke-2 sampai data ke-100 dengan rumus seperti diatas.

4. Menentukan jarak terdekat suatu objek dengan *centroid*.
5. Tentukan pusat *cluster (centroid)* baru dengan menghitung rata-rata masing-masing *cluster* menggunakan persamaan rumus 3.2:

$$c_{kl} = \frac{x_{1l} + x_{2l} + \dots + x_{pl}}{p} \dots \dots \dots (3.2)$$

Keterangan:

c_{kl} = Nilai *centroid* ke-k pada variabel ke-l

P = Banyaknya data

6. Jika suatu objek tidak berpindah *cluster* dari jarak *centroid* lama dan jarak *centroid* baru maka ulangi langkah 3.

3.10. Pengujian *Silhouette Coefficient*

Untuk melihat kualitas hasil pengelompokan masing-masing perhitungan jarak, maka perlu dilakukan uji homogenitas. Pengujian dilakukan setelah mencapai konvergensi 0 di mana hasil pengelompokan terakhir sama dengan pengelompokan sebelumnya. Dengan kata lain, tidak ada data yang berpindah klaster. Pengujian dihitung menggunakan persamaan *Silhouette Coefficient*. Langkah dalam menghitung *Silhouette Coefficient* di mulai dengan mencari jarak rata-rata data ke-*i* dengan semua data di klaster yang sama, di sini kita asumsikan data ke-*i* berada di klaster A. Rumus dari *a(i)* ditulis dalam Persamaan (Struyf, 1996).

Penafsiran nilai *Silhouette Coefficient* ditunjukkan dalam tabel 3.6.

Tabel 3.6 Interpretasi Nilai *Silhouette Coefficient* (Struyf, 1996)

<i>Silhouette Coefficient</i>	Interpretasi
0.71 – 1.00	Struktur yang dihasilkan kuat
0.51 – 0.70	Struktur yang dihasilkan baik
0.26- 0.50	Struktur yang dihasilkan lemah
≤ 0.25	Tidak terstruktur

3.11. Pembuatan Model Dan Pengembangan Sistem

Pembuatan model sistem dalam penelitian ini melibatkan penggunaan algoritma *K-Means* untuk mengklasterisasi lagu berdasarkan fitur-fitur yang berkaitan dengan suasana hati dan umur. Proses pembuatan model dimulai dengan tahap pengumpulan data dari Spotify menggunakan API yang disediakan. Data yang diperoleh kemudian melalui proses *preprocessing* untuk memastikan kualitas data yang baik dan relevan.

Setelah data siap, langkah selanjutnya adalah penerapan algoritma *K-Means* untuk

mengelompokkan data ke dalam beberapa *cluster* berdasarkan kesamaan karakteristik. Jumlah *cluster* ditentukan berdasarkan *Elbow Method* untuk mencapai optimalisasi *clustering*. Hasil *clustering* kemudian dievaluasi menggunakan *Silhouette Coefficient* untuk menilai keakuratan pengelompokan.

Setelah itu maka akan dibuat sistem rekomendasi musik menggunakan web *streamlit* dengan menggunakan Bahasa pemrograman Python, yang memungkinkan pembuatan aplikasi web interaktif dengan mudah. Sistem ini memungkinkan pengguna untuk memasukkan *input* suasana hati dan umur, lalu sistem akan merekomendasikan lagu yang sesuai dengan hasil *cluster* sebelumnya.

3.12. Pengujian *Black Box*

Black box adalah metode pengujian perangkat lunak yang digunakan untuk mengevaluasi fungsionalitas aplikasi tanpa memerlukan pengetahuan tentang implementasi internal kode program. Tujuan dari pengujian ini adalah untuk memeriksa apakah aplikasi berfungsi sesuai dengan spesifikasi dan persyaratan yang ditetapkan, dan memastikan semua fitur berjalan dengan baik sesuai yang diharapkan. Dan kelebihan dari pengujian ini yaitu kemudahannya dalam implementasi, karena tidak memerlukan pengetahuan khusus tentang kode atau arsitektur sistem.

BAB IV

HASIL DAN PEMBAHASAN

Pada bab ini menjabarkan tentang hasil dan pembahasn tentang proses awal hingga akhir dalam membuat Sistem Rekomendasi Musik Pada Spotify Berdasarkan Suasana Hati Dan Umur Menggunakan Metode *K- Means*, data yang akan digunakan yaitu berasal dari data musik yang disediakan oleh spotify.

4.1. Pengumpulan Data

Data yang digunakan pada penelitian ini bersumber dari spotify, dengan menggunakan Teknik *crawling* data dengan memanfaatkan API yang disediakan oleh spotify dengan mendaftar di website *developer.spotify.com*, setelah mendaftar maka kita akan mendapatkan *CLIENT_ID* dan *CLIENT_SECRET* yang akan dipakai agar bisa mengakses data dari spotify. Code nya seperti pada gambar 4.1.

```
client_id = "8a056f31514c4db2a8b2048086f6e3ef"  
client_secret = "0fcffe48f39e4440a4fb68c77d42d5bc"
```

Gambar 4.1 *CLIENT_ID CLIENT_SECRET*

Setelah mendapatkan *client_id* dan *client_secret*, maka selanjutnya kita akan melakukan *crawling* data, dengan menggunakan *library* python, berikut gambar 4.2 *library* yang digunakan untuk melakukan *crawling* data.

```
import requests  
from requests.auth import HTTPBasicAuth  
import pandas as pd
```

Gambar 4.2 *Library Yang Digunakan*

Setelah *import library* diatas, maka selanjutnya kita akan melakukan *request* agar mendapatkan *access token*, dengan menggunakan kode seperti pada gambar 4.3.

```

auth_url = 'https://accounts.spotify.com/api/token'

auth_response = requests.post(auth_url,
                               data={'grant_type': 'client_credentials'},
                               auth=HTTPBasicAuth(client_id, client_secret))

if auth_response.status_code == 200:
    access_token = auth_response.json()['access_token']
    print("Access Token:", access_token)
else:
    print("Failed to obtain access token. Status Code:", auth_response.status_code)

```

Gambar 4.3 Request Token Access

Dengan menggunakan kode diatas maka kita akan melakukan request untuk mendapatkan *token access*, setelah dirunning maka akan mendapatkan *token access* seperti pada gambar 4.4 atribut yang digunakan.

```

Access Token: BQDe0JRXlPsub1YNud_QUjRYjHNR3mcQLHHmJvP7b86a-ZrgUk-
P13cx5sSZnKv0y8pUD3bMAdZ299WU104vgpaamp8GYhpD2-e3-wGpB4sFNfgD5-U

```

Gambar 4.4 Token Access

Setelah mendapatkan *token access* maka selanjutnya kita akan melakukan penarikan data data dengan menggunakan kode seperti pada gambar 4.5 kode yang digunakan untuk melakukan penarikan data.

```

dataset = []
track_ids = get_spotify_playlist_tracks(playlist_id, token)
for track_id in track_ids:
    track_details = get_track_details(track_id, token)
    audio_features = get_audio_features(track_id, token)

    track_info = {
        'track_id': track_id,
        'track_name': track_details['name'],
        'artist_name': track_details['artists'][0]['name'],
        'danceability': audio_features['danceability'],
        'energy': audio_features['energy'],
        'key': audio_features['key'],
        'loudness': audio_features['loudness'],
        'mode': audio_features['mode'],
        'speechiness': audio_features['speechiness'],
        'acousticness': audio_features['acousticness'],
        'instrumentalness': audio_features['instrumentalness'],
        'liveness': audio_features['liveness'],
        'valence': audio_features['valence'],
        'tempo': audio_features['tempo'],
        'duration_ms': audio_features['duration_ms'],
        'time_signature': audio_features['time_signature']
    }
    dataset.append(track_info)
df = pd.DataFrame(dataset)

```

Gambar 4.5 Kode Program Untuk Penarikan Data

Pada kode diatas atribut yang diambil yaitu *track_id*, *track_name*, *artist_name*, *danceability*, *energy*, *key*, *acousticness*, *loudness*, *mode*, *speechiness*, *acousticness*, *instrumentalness*, *liveness*, *valence*, *tempo*, *duration-ms*, *time_signature*, setelah *running* kode diatas maka selanjutnya kita akan menambahkan hasil *crawling* datannya kedalam *dataset* dengan menggunakan program seperti pada gambar 4.6.

```
csv_filename = 'anak-anak.csv'
df.to_csv(csv_filename, index=False)
```

Gambar 4.6 Memasukan Hasil *Crawling* data Ke Dalam *Dataset*

Cara ini dilakukan berulang ulang dan dengan menggunakan id *Playlist* yang berbeda untuk setiap melakukan *crawling* data, dan pada penelitian ini saya melakukan setidaknya 16 kali *crawling* data dengan 2 *Playlist* lagu bayi, 4 *Playlist* lagu anak anak, 4 *Playlist* lagu remaja, dan 4 *Playlist* lagu dewasa dan 2 *Playlist* lagu lansia, hal ini dilakukan agar data yang di dapat bisa lebih banyak dan relevan. Gambar 4.7 adalah hasil dari *crawling* data nya.

		track_id	track_name	artist_name	danceability	energy	key	loudness	mode
1254	3usbnvDFtOhY09cRNar8Zg	FRIENDS	LAUNDRY DAY		0.522	0.554	5.0	-6.978	1.0
1255	2h4BV7jx9Ltz9NxC2zgNcW	Now or Never (feat. Charlie Gillespie, Owen Pa...	Julie and the Phantoms Cast		0.449	0.957	2.0	-3.797	1.0
1256	67BtfxlNbh8mCDR2L2l8qd	MONTERO (Call Me By Your Name)	Lil Nas X		0.610	0.508	8.0	-6.682	0.0
1257	7CBXta3Je26MMHOFGERpIS	Miss The Rage - Remix	Toxic Tripz		0.514	0.659	8.0	-7.903	1.0
1258	4p5Phj0pcXWW7B81IUGowi	masquerade x miss the rage	acti		0.496	0.835	8.0	-16.298	1.0

Gambar 4.7 *Dataset* Hasil *Crawling* Data

Gambar 4.7 diatas merupakan hasil *crawling* data menggunakan API, jika datanya sudah berhasil didapatkan maka bentuk datanya seperti pada gambar diatas, dengan total 1259 baris dan 17 kolom.

Berikut adalah judul *Playlist* dan id *Playlist* yang dipakai ketika *crawling* data, dapat dilihat pada tabel 4.1.

Tabel 4.1 Id *Playlist* Yang Dipakai

Kelompok Umur	Judul <i>Playlist</i>	Id <i>Playlist</i>
Bayi	- <i>Baby Sleeping Instrumentals Calming Lullabies</i> - <i>Babylieder</i>	- 1XU4GqyxAd4EczpplAUyR8 - 7IGKYNGkm6oTPjKMOiAN2Y
Anak anak	- <i>Playlist Anak</i> - Lagu Anak-anak Indonesia - <i>Children's Songs Happy Kids Music</i> - Lagu Anak Kecil Cantik/ganteng	- 6eJbQgLwLGHjbeLwmL7d95 - 3U3P1ps8TNEfmNjLP7UUA0 - 1ynJ63k9UYnKQWH2Q0KMml - 29JLMFwPBWzH8ESUEh9q8Y
Remaja	- Kita Remaja Yang Sedang Di Mabuk Cinta - Remaja Gila - Semangatin Diri Sendiri - <i>Best Playlist For Teenagers</i>	- 0u5fQ6Up28JVoD7voQu6hb - 3KpHdSu3lkjrmBJ2g9ddHs - 1soJscZN2hFs3bHL98ncW3 - 0aDDorcqLsMJlJy2jloRvS
Dewasa	- Dipaksa Dewasa - Dewasa Itu Berat - <i>If Your so "Mature" Now.. Then I Think It's Time You Grow Up!</i> - <i>Grown Ups Summer</i>	- 0CbEFSmhnCOj1kC0QxPWar - 1fbO2xF87tSC6JJ2idwDww - 5XAWGd0zBdqESEvku7Vlp7 - 5d4kut5h418v3PonrHrX2B
Lansia	- <i>Elderly Songs</i> - Nostalgia Lagu 70-90an	- 7mdiqsc3fEqWdfntP5wSU - 3dFS0Y3WsrDNwsGQ9WfBsF

Pada tabel 4.1 diatas merupakan jumlah id *Playlist* yang dipakai ketika melakukan penarikan data, bayi terdapat 2 Id *Playlist*, anak-anak *Playlist*, remaja 4 *Playlist*, dewasa 4 *Playlist* dan lansia 2 *Playlist*, jadi total terdapat 16 *Playlist*, dengan total musik yang berhasil di dapatkan yaitu berjumlah 1242 musik.

4.2. Preprocessing Data

Pada proses ini kita akan melakukan *preprocessing* data terlebih dahulu. Pertama

tama kita akan mengecek berapa jumlah *dataset* yang ada menggunakan program seperti pada gambar 4.8.

```
rows, columns = df.shape
rows, columns

(1242, 17)
```

Gambar 4.8 Jumlah *Dataset*

Setelah itu kita akan mengecek apakah dalam *dataset* kita terdapat kolom yang tidak memiliki nilai atau tidak, dan hasilnya bisa dilihat pada gambar 4.9.

```
nilai_kosong = df.isnull().sum()
nilai_kosong
```

track_id	0
track_name	0
artist_name	0
danceability	0
energy	0
key	0
loudness	0
mode	0
speechiness	0
acousticness	0
instrumentalness	0
liveness	0
valence	0
tempo	0
duration_ms	0
time_signature	0
genre	0
dtype: int64	

Gambar 4.9 Mengecek Nilai Kosong Pada *Dataset*

Dan hasilnya dalam *dataset* kita tidak terdapat nilai kosong, dan semua kolom memiliki nilai masing-masing. Setelah itu, maka selanjutnya kita akan mengecek apakah *dataset* kita terdapat data yang terduplikat atau tidak, seperti pada gambar 4.10.

```
duplicate_rows = df.duplicated().sum()
duplicate_rows
```

2

Gambar 4.10 Mengecek Duplikat

Setelah mengetahui *dataset* kita terdapat 35 data yang terduplikat, maka selanjutnya kita akan menghapus duplikat dari *dataset* kita dengan program seperti pada gambar 4.11.

```
df.drop_duplicates(inplace=True)
hapus_duplikat = df.shape
print(hapus_duplikat)
```

✓ 0.0s

(1240, 17)

Gambar 4.11 Hapus Duplikat

Kode diatas digunakan untuk menghapus data yang sama atau duplikat, df sebagai *DataFrame* jadi fungsi '*drop_duplicates()*' akan menghapus data yang sama persis dalam df/*dataset*. Setelah itu maka data sudah siap digunakan untuk diproses lebih lanjut. Berikut adalah gambar 4.12 hasil *preprocessing*.

	track_id	track_name	artist_name	danceability	energy	key	loudness	mode	speechiness
0	0tCqX7pASaZeHPC4FhAwe8	Brahms Lullaby	Sleepy Shepherd	0.390	0.01610	1	-29.263	1	0.0398
1	2DG5ytzXs9oqYUy5tqxfS8	Twinkle Twinkle Little Star	Tommy Twinkle	0.550	0.06420	1	-24.304	1	0.1230
2	0KOOhxPmW5pfMXjgWzRf0j	Goodnight Little Dreamer	Little Dreamer	0.309	0.00781	2	-30.090	1	0.0557
1238	2h48V7jx9Ltz9Nx2zgNcW	Now or Never (feat. Charlie Gillespie, Owen Pa...	Julie and the Phantoms Cast	0.449	0.95700	2	-3.797	1	0.0610
1239	678tfxiNbh8mCDR2L2l8qd	MONTERO (Call Me By Your Name)	Lil Nas X	0.610	0.50800	8	-6.682	0	0.1520

Gambar 4.12 Hasil *Preprocessing*

4.3. Menentukan Kategori Umur

Pada tahap ini kita akan menentukan kategori umur pada *dataset* ini karena penelitian sistem tidak hanya merekomendasikan berdasarkan suasana hati saja, tetapi juga merekomendasikan berdasarkan umur juga, oleh karena itu maka akan dibuat kategori umur berdasarkan kategori umur menurut Departemen Kesehatan Republik Indonesia yakni sebagai berikut:

1. 60+ = Lansia
2. 19 – 59 = Dewasa
3. 10 - 18 tahun = Remaja
4. 5 - 9 tahun = Anak-anak
5. < 5 tahun = bayi

Dan kategori umur kita tentukan berdasarkan genre musik, jadi seperti yang sudah saya jelaskan pada penjelasan di proses *crawling* data, jadi agar musik memiliki hubungan dengan kategori umur maka, kategori umur dibuat berdasarkan *Playlist* yang ada di Spotify, jadi misalnya kita akan mencari musik dengan genre remaja, maka kita akan mencari musik genre remaja pada *Playlist* spotify, setelah mendapatkan *Playlist* nya, kita akan mengambil *Playlist_id* nya untuk mengambil data musik yang ada pada *Playlist* tersebut untuk dianalisis lebih lanjut.

4.4. Menentukan Kategori Suasana Hati

Pada tahap ini kita akan melakukan *clustering* untuk mencari jenis jenis suasana hati yang dihasilkan dari hasil *cluster* dari berbagai fitur, jadi kita akan melakukan setidaknya 4 kali *clustering* untuk mendapatkan 4 jenis suasana hati yaitu *calm*, *happy*, *sad*, *energetic* dan *angry*. Oleh karena itu kita akan mencari fitur fitur apa yang bisa menghasilkan 5 jenis suasana hati tersebut.

Pada proses ini kita akan memerlukan beberapa *library* agar dapat membantu kita dalam melakukan proses ini, berikut gambar 4.13 *library* yang digunakan.

```
import pandas as pd
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
```

Gambar 4.13 *Library* Yang Digunakan

4.4.1. Mencari Suasana Hati *Calm*

1. Menentukan Jumlah *Cluster*

Pada tahap ini kita akan menentukan jumlah *cluster* terlebih dahulu sebelum melakukan *clustering* dengan menggunakan *Elbow Method*, metode ini banyak digunakan pada penelitian sebelumnya, karena mudah untuk diimplementasi pada program dan juga mudah untuk dibaca, berikut adalah gambar 4.14 program untuk menentukan jumlah *cluster*.

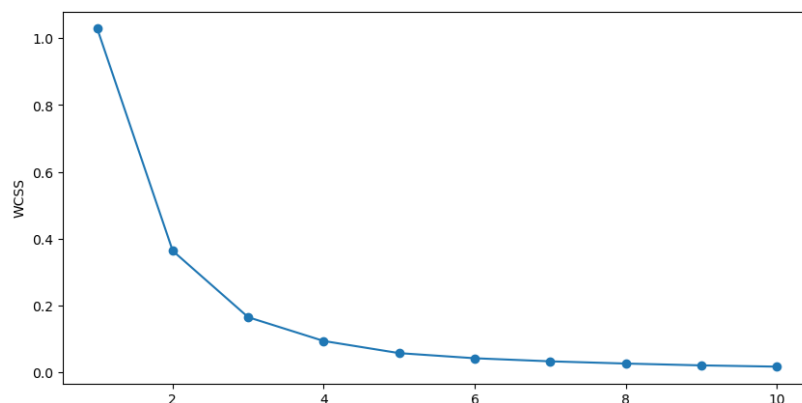
```
X = df[['instrumentalness', 'tempo']]
wcss = []

for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10, random_state=0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)

plt.figure(figsize=(10, 5))
plt.plot(range(1, 11), wcss, marker='o')
plt.title('Metode Elbow untuk Menentukan Jumlah k')
plt.xlabel('Jumlah kluster')
plt.ylabel('WCSS')
plt.show()
```

Gambar 4.14 Kode Program *Elbow Method*

Program diatas menggunakan fitur "*instrumentalness*" dan "*tempo*" untuk mencari *mood calm*, karena kedua fitur ini biasa menghasilkan musik yang tenang atau santai, jika program diatas sudah di *running* maka hasilnya bisa dilihat pada gambar 4.15.



Gambar 4.15 Hasil *Elbow Method*

Gambar dari hasil *Elbow Method* diatas menjelaskan bahwa penurunan terjadi pada *cluster* 2-3-4 maka jumlah *cluster* tersebut lah yang direkomendasikan oleh *Elbow Method* untuk dilakukan *clustering*.

2. Clustering

Pada tahap ini kita akan melakukan *clustering* untuk mencari suasana hati *calm*, dikarenakan kita sudah mengetahui hasil dari *Elbow Method* untuk menentukan *n cluster*, maka selanjutnya kita akan melakukan *clustering* pada fitur yang sama pada *Elbow Method* yaitu “*instrumentalness*” dan “*tempo*” untuk mencari jenis suasana hati *calm*, berikut adalah kode program untuk melakukan *clustering*, bisa dilihat pada gambar 4.16.

```
data = df
kmeans = KMeans(n_clusters=5, random_state=0)
kmeans.fit(df[['instrumentalness', 'tempo']])

df['cluster'] = kmeans.labels_
cluster_data = {}

for k in range(kmeans.n_clusters):
    cluster_data[f'clus{k}'] = df[df['cluster'] == k].drop('cluster',
axis=1)
centers = kmeans.cluster_centers_
print("Centroid dari setiap cluster:")
for i, center in enumerate(centers):
    print(f"Cluster {i}: {center}")

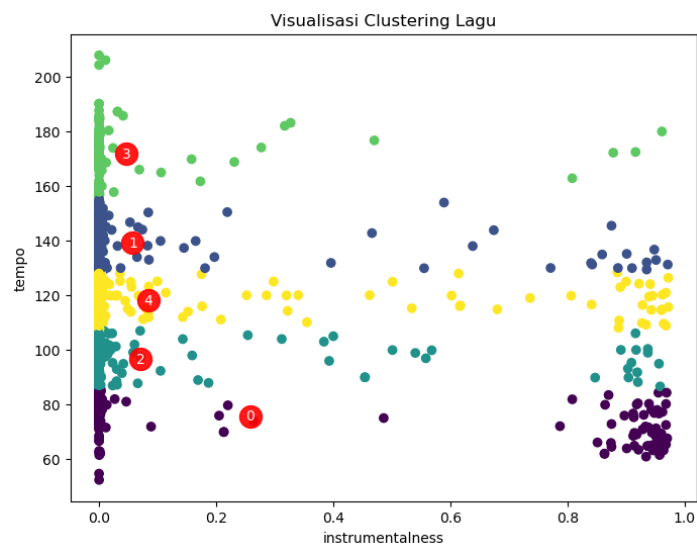
plt.figure(figsize=(8, 6))
plt.scatter(df['instrumentalness'], df['tempo'], c=kmeans.labels_,
cmap='viridis')
for i, center in enumerate(centers):
    plt.scatter(center[0], center[1], c='red', s=100, alpha=0.9)
    plt.text(center[0], center[1], str(i), color='white',
ha='center', va='center')

plt.xlabel('Instrumentalness')
plt.ylabel('Tempo')
plt.title('Visualisasi Clustering Lagu')
plt.show()
```

Gambar 4.16 Kode Program *Clustering*

Pada kode diatas df merupakan *DataFrame* atau *dataset*, dan pada kode diatas
n
cluster nya saya pakai 5, dan pada kode selanjutnya kita akan memilih fitur apa yang akan digunakan untuk *clustering*, setekah memilih fitur maka selanjutnya kita saya menambahkan

kode untuk memvisualisaikan hasil *cluster* nya menggunakan *plot*, dan saya juga menambahkan *code* agar terdapat visual yang mencari *centroid* dari setiap *cluster*, hasil dari kode diatas akan seperti pada gambar 4.17.



Gambar 4.17 Hasil *Clustering*

Gambar diatas merupakan hasil *cluster* dari fitur "*instrumentalness*" dan "*tempo*". Pada sumbu (x) yaitu *instrumentalness* memiliki nilai antara 0 hingga 1, dimana nilai yang mendekati 1.0 menandakan lagu tersebut adalah instrumental, dan nilai yang mendekati 0.0 menandakan bahwa lagu tersebut kemungkinan mengandung vokal. Sementara pada sumbu (y) yaitu *tempo*, nilai *tempo* ini menunjukkan kecepatan dan lambatnya sebuah lagu, jadi semakin tinggi nilai dari *tempo* maka semakin cepat lagu tersebut, dan semakin rendah nilai *tempo* maka semakin lambat sebuah lagu.

Sedangkan titik-titik yang memiliki warna yang berbeda-beda menunjukkan *cluster* atau grup lagu yang memiliki karakteristik yang sama/serupa. Misalnya, lagu-lagu yang memiliki '*tempo*' tinggi dan '*instrumentalness*' rendah mungkin berada dalam satu kluster, sedangkan lagu-lagu dengan '*tempo*' rendah dan '*instrumentalness*' tinggi berada dalam kluster yang lain. Jadi fungsi dari warna tersebut adalah untuk membedakan *cluster* satu

dan *cluster* lainnya, dan titik yang berwarna merah tersebut merupakan *centroid* atau pusat *cluster* nya dan selain itu angka tersebut menandakan *cluster* tersebut *cluster* berapa. Berikut penjelasan yang lebih spesifik untuk setiap *cluster*. Analisis lebih lanjut yang dimaksud yaitu dengan memeriksa datanya satu per satu.

a. *Cluster 0*(warna ungu)

Pada *cluster* ini titik-titik di area '*instrumentalness*' sangat bervariasi dimana penyebarannya terpisah menjadi 2 kelompok, kelompok pertama memiliki nilai *instrumentalness* yang dominan diangka 0.0, dan kelompok ke 2 yang dominan pada nilai 0.8 – 1.0 yang mengindikasikan lagu-lagu ini tidak sepenuhnya instrumental. *Tempo* dari lagu-lagu di kluster ini bervariasi, mulai dari sangat lambat (sekitar 60 bpm) hingga cukup cepat (lebih dari 80 bpm). Lagu-lagu dengan '*instrumentalness*' tinggi dan *tempo* lambat menghasilkan suasana yang sangat tenang dan meditatif, sedangkan lagu-lagu dengan *tempo* yang lebih cepat bisa memberikan rasa semangat. Berikut gambar 4.18 adalah data dari *cluster 0*.

cluster_data['clus0']
✓ 0.4s

	track_id	track_name	artist_name	danceability	energy	key	loudness	mode	speechiness	acousticness
1	2DGSytzXs9oqYUy5tqxfsB	Twinkle Twinkle Little Star	Tommy Twinkle	0.550	0.06420	1	-24.304	1	0.1230	0.9960
3	4tbnZK9nKvuyCrqyu8o0eG	Hush Little Baby	Sleepy Shepherd	0.554	0.02040	11	-27.418	1	0.0528	0.9950
4	5YMmjaZEzUol8EFZm4q2fB	Moonlight Walk	Orion The Owl	0.479	0.01890	10	-24.228	1	0.0514	0.9930
6	4GphHeTclqEblY6jRMyaG	Lavender's Blue	Sleepy Shepherd	0.532	0.03230	5	-26.907	1	0.0441	0.9930
7	5NI8DkhAIQjki2AijI90h	Wake Up	Bluedon	0.618	0.00838	0	-21.242	1	0.0550	0.9870
...	...	...	...	...	...	...	...	...	...	...
1214	63BcfK6YAzJYelSaTPr6lO	Big Poppa - 2007 Remaster	The Notorious B.I.G.	0.765	0.67100	9	-5.249	0	0.2780	0.3440
1215	2HbKqm4o0w5wEeEFXm2sD4	Money Trees	Kendrick Lamar	0.716	0.53100	7	-7.355	1	0.1220	0.0703
1216	1UGD3IW3tDmgZfAVDh6w7r	Devil In A New Dress	Kanye West	0.435	0.76000	8	-4.935	0	0.0721	0.0175
1218	51EC3l1nQXpec4gDk0mQyP	90210 (feat. Kacy Hill)	Travis Scott	0.402	0.52600	9	-6.741	0	0.0904	0.1130
1235	3usbvDFtOhY09cRNar8Zg	FRIENDS	LAUNDRY DAY	0.522	0.55400	5	-6.978	1	0.0290	0.4900

254 rows × 11 columns

Gambar 4.18 Hasil Data *Cluster 0*

Gambar diatas merupakan data dari hasil *cluster* yang ada pada *cluster 0*, *cluster* ini berhasil mendapatkan data dengan jumlah 254 data, data diatas akan dilakukan analisis lebih lanjut agar dapat mengetahui *cluster* diatas menghasilkan suasana hati apa. Analisis lebih lanjut yang dimaksud yaitu dengan memeriksa datanya satu per satu.

b. *Cluster 1* (warna biru)

Ini terdiri dari lagu-lagu dengan '*instrumentalness*' rendah dan dominan pada angka 0.0 – 0.2, namun ada beberapa titik yang dominan antara nilai 0.4 – 1.0, yang bisa saja merepresentasikan musik *happy*, dan *tempo* yang pada umumnya cepat (sekitar 130 hingga 160 bpm). Meskipun *tempo* ini cenderung cepat, karena nilai '*instrumentalness*' lebih ke arah rendah, dan hal juga bisa menunjukkan keberadaan elemen vokal yang menambah dimensi emosional yang bisa menenangkan, tergantung pada konteks musikal dan liriknya juga. Berikut gambar 4.19 adalah data yang dihasilkan *cluster 1*.

cluster_data['clus1']										
✓	0.0s									
	track_id	track_name	artist_name	danceability	energy	key	loudness	mode	speechiness	acousticness
5	1EvERSC5XOhhcvGdhSHjil	Golden Sunrise	Chillboy	0.561	0.01540	4	-24.016	1	0.0577	0.99600
10	7DOQ5HUSJeHYLmOhtGKz3y	Are You Sleeping	Sleepy Shepherd	0.518	0.02560	8	-27.621	1	0.0452	0.99500
17	1VvLz9pvdlokhKP3jJwEilQ	Night Sky	Sleepy Shepherd	0.416	0.02100	10	-28.794	1	0.0499	0.99500
18	1Dmu3GjUt8z1QzeLyKxmK	Singing For You	Sleepy Shepherd	0.506	0.01460	1	-29.600	0	0.0704	0.99600
30	7FL2BfLeJWq8TO7gf0KcGG	Reflection	Sleepy Shepherd	0.564	0.00289	8	-27.056	1	0.0731	0.99400
...	...	...	...	...	...	...	...	...	...	...
1223	3YJJQPAbDT7mGpX3WtQ9A	Good Days	SZA	0.436	0.65500	1	-8.370	0	0.0583	0.49900
1226	4mGJcOKQhmlSIWgQ3xCiLB	Til I Forget About You	Big Time Rush	0.568	0.86500	5	-3.653	1	0.0486	0.00632
1227	6knXmt7UIPPJveAtsP1Dq4	Don't Run Away	Tyler James Williams	0.747	0.80100	7	-6.172	0	0.0493	0.07830
1228	5ZkAx8zjLiSs1nMmBwJoZS	When Can I See You Again? - From "Wreck-It RaL...	Owl City	0.638	0.83100	11	-6.812	0	0.0810	0.02630
1234	7ETjvgUdK7rBlGfVhiWQj	Stay	The Hails	0.665	0.72300	9	-4.086	1	0.0269	0.00412

347 rows × 17 columns

Gambar 4.19 Hasil Data *Cluster 1*

Gambar diatas merupakan data dari hasil *cluster* yang ada pada *cluster 1*, *cluster* ini berhasil mendapatkan data dengan jumlah 347 data, data diatas akan dilakukan analisis lebih lanjut agar dapat mengetahui *cluster* diatas menghasilkan suasana hati apa. Analisis lebih lanjut yang dimaksud yaitu dengan memeriksa datanya pada tiap tiap *cluster*, *cluster* mana yang mengandung suasana hati yang dicari akan ditambahkan ke *dataset*.

c. *Cluster 2*(warna hijau toska)

Kluster ini mirip dengan kluster 1(warna biru) yang berisi lagu-lagu dengan '*instrumentalness*' rendah, yang nilainya cenderung pada 0.0 – 0.2, dan ada juga yang tersebar dan dominan pada nilai 0.3 – 0.6, dan ada juga pada nilai 0.8 – 1.0, yang menandakan adanya vokal atau harmoni yang lebih menonjol, dan *tempo* yang sangat bervariasi dari sangat lambat hingga sangat cepat (sekitar 70 hingga 110 bpm). Meskipun memiliki nilai *instrumentalness* yang beradagam *cluster* ini juga bisa di kategorikan musik *calm*, namun nilai *temponya* agak sedang jadi bisa jadi berhubungan dengan suasana hati yang lain. Berikut gambar 4.20 adalah data yang dihasilkan *Cluster 2*.

cluster_data['clus2']

✓ 0.0s

	track_id	track_name	artist_name	danceability	energy	key	loudness	mode
13	6mZQXPV4n8FBDEYGFx8O8k	Itsy Bitsy Spider	Sleepy Shepherd	0.508	0.0172	4	-26.652	1
14	01b3JCDs8aA4D6Rglx8yhX	Trust Me	Sleepy Shepherd	0.536	0.0128	10	-24.603	1
15	1K7JgK79uuxSOpbB415lpM	Daydreaming	Sleepy Shepherd	0.413	0.0354	8	-31.037	1
...	...	...	...	...	...	...	...	...
1232	796gbCqHEHWvEqYOBXibL	Fake a Frown	The Hails	0.758	0.6340	6	-6.851	0
1233	0oUBuOO4g9P4IREqfR5nq	A Man Without Love	Engelbert Humperdinck	0.538	0.4670	2	-9.589	1
1238	7CBXta3Je26MMHOFGERpIS	Miss The Rage - Remix	Toxic Tripz	0.514	0.6590	8	-7.903	1

291 rows × 17 columns

Gambar 4.20 Hasil Data *Cluster 2*

Gambar 4.20 merupakan data dari hasil *cluster* yang ada pada *cluster 2 cluster* ini berhasil mendapatkan data dengan jumlah 291 data, data diatas akan dilakukan analisis lebih lanjut agar dapat mengetahui *cluster* diatas menghasilkan suasana hati apa.

d. *Cluster 3*(warna hijau muda)

Kluster ini Menunjukkan lagu-lagu dengan '*instrumentalness*' yang dominan pada pada angka 0.0 – 0.3, tetapi ada juga yang tersebar pada kisaran 0.5 – 1.0, dan *tempo* memiliki bilai yang sangat tinggi (sekitar 120 hingga 210 bpm). Meskipun *tempo* yang tinggi biasanya tidak diasosiasikan dengan suasana hati '*calm*', kemungkinan adanya variasi dalam dinamika lagu-lagu ini bisa menimbulkan suasana hati yang lebih bervariasi. Berikut adalah gambar 4.21 data yang dihasilkan *cluster 3*.

```
cluster_data['clus3']
```

✓ 0.1s

	track_id	track_name	artist_name	danceability	energy	key	loudness	mode	speechiness	acousticness
2	0KOOHxPmW5pfMXjgWzRf0j	Goodnight Little Dreamer	Little Dreamer	0.309	0.00781	2	-30.090	1	0.0557	0.996000
11	6lpb4EVYeX94RQb2mo3AUr	My Bonnie Lies Over The Ocean	Sleepy Shepherd	0.338	0.01550	11	-29.874	1	0.0445	0.995000
29	7nTghMbNP3rDEzStg5NGlz	Into The Clouds	Sleepy Shepherd	0.317	0.05420	4	-28.228	1	0.0445	0.995000
...	...	...	...	...	...	...	...	...	...	...
1231	6kE3BJhZYLfGLQY2hSzaM	Younger	The Hails	0.700	0.44400	9	-6.614	1	0.0349	0.220000
1236	2h4BV7jx9Ltz9Nx2zgNcW	Now or Never (feat. Charlie Gillespie, Owen Pa...	Julie and the Phantoms Cast	0.449	0.95700	2	-3.797	1	0.0610	0.000197
1237	67BtfXlNbh8mCDR2L2l8qd	MONTERO (Call Me By Your Name)	Lil Nas X	0.610	0.50800	8	-6.682	0	0.1520	0.297000

120 rows × 11 columns

Gambar 4.21 Hasil Data *Cluster 3*

Gambar diatas merupakan data dari hasil *cluster* yang ada pada *cluster 3 cluster* ini berhasil mendapatkan data dengan jumlah 120 data, data diatas akan dilakukan analisis lebih lanjut agar dapat mengetahui *cluster* diatas menghasilkan suasana hati apa.

e. *Cluster 4*(warna kuning)

Kluster ini cukup berbeda dengan *cluster* lain, dimana kluster ini memiliki nilai

instrumental hamper di setiap nilai, tapi yang paling dominan pada angka 0.0 – 0.2 dan 0.9 – 1.0, dan dengan nilai *tempo* pada angka 110 – 130 bpm, berdasarkan nilai dari *instrumentalness*, kluster ini berhubungan dengan suasana hati *calm*, hanya saja nilai *tempo* nya cukup tinggi yang menandakan bahwa kluster ini juga menghasilkan suasana hati yang cukup bervariasi karena nilai *tempo*. Berikut adalah gambar 4.22 data yang dihasilkan oleh *cluster 4*.

```
cluster_data['clus4']
```

✓ 0.0s

	track_id	track_name	artist_name	danceability	energy	key	loudness	mode	speechiness	acousticness
0	0tCqX7pASaZeHPC4FhAwe8	Brahms Lullaby	Sleepy Shepherd	0.390	0.0161	1	-29.263	1	0.0398	0.99300
27	37ctH3ngeJXnVz7mua4LkF	Floating	Sleepy Shepherd	0.528	0.0152	2	-24.014	1	0.0540	0.99500
57	2xe9jwt3QmcYTVcwreG221	Soon You'll Be Sleeping	Sleepy Shepherd	0.565	0.0123	8	-27.261	1	0.2030	0.99500
...	...	...	...	...	...	...	...	...	...	...
1224	5FSt6K70gANsecaAtXFVbR	See You Again	Shakka	0.593	0.8930	6	-5.811	1	0.1320	0.30100
1225	42Lq6GBOSKUSNeSgJsOoke	Panama	Van Halen	0.514	0.9870	8	-5.391	1	0.1090	0.00135
1239	4p5Phj0pcXWW7B81IUgowi	masquerade x miss the rage	acti	0.496	0.8350	8	-16.298	1	0.0511	0.02440

228 rows × 11 columns

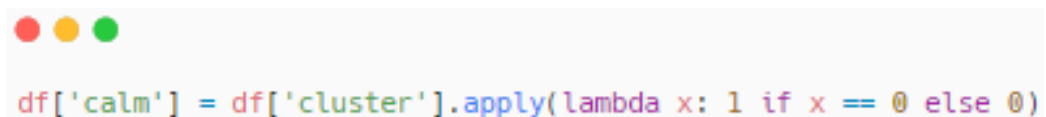
Gambar 4.22 Hasil Data *Cluster 4*

Gambar diatas merupakan data dari hasil *cluster* yang ada pada *cluster 4* cluster ini berhasil mendapatkan data dengan jumlah 228 data, data diatas akan dilakukan analisis lebih lanjut agar dapat mengetahui *cluster* diatas itu berhasil menghasilkan suasana hati apa. Setelah mendapatkan hasil *cluster*, dan selanjutnya kita akan menambahkan hasil *cluster* tersebut kedalam *dataset* untuk mencari dari ke 5 *cluster* tersebut manakah yang menghasilkan suasana hati *calm*, berikut adalah kode programnya bisa dilihat pada gambar 4.23.

```
cluster_labels = kmeans.labels_  
df['cluster'] = cluster_labels
```

Gambar 4.23 Kode Program Untuk Menambah Hasil *Cluster* Ke *Dataset*

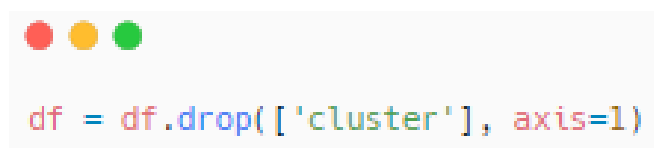
Setelah menjalankan kode diatas maka hasil *cluster* akan ditambahkan kedalam *dataset* Bernama *cluster* untuk diseleksi *cluster* mana yang menghasilkan jenis suasana hati *calm*, seleksi ini dilakukan secara manual, jadi *music* dari *cluster* 1/2/3/4/5 akan dicek satu persatu, dari ke empat *cluster* tersebut *cluster* berapakah yang menghasilkan suasana hati *calm*. Setelah mengetahui *cluster* berapa yang menghasilkan suasana hati *calm* selanjutnya kita akan menambahkan *cluster* tersebut ke dalam *dataset*, kodenya bisa dilihat pada gambar 4.24.



```
df['calm'] = df['cluster'].apply(lambda x: 1 if x == 0 else 0)
```

Gambar 4.24 Kode Program Menambah Hasil Seleksi Suasana Hati *Calm*

Setelah menjalankan kode diatas maka sistem akan menambahkan kolom baru pada *dataset* dengan nama *calm*, jadi kode diatas juga menandai lagu-lagu yang termasuk dalam *cluster* 0 sebagai jenis suasana hati *calm*, jadi pada kolom *calm* nilainya akan menjadi 1 jika dia merupakan *cluster* 0, jika bukan maka akan diberi nilai 0. Setelah itu maka kita akan menghapus kolom *cluster* karena kita sudah berhasil mendapatkan jenis suasana hati *calm*, berikut gambar 4.25 kode programnya.



```
df = df.drop(['cluster'], axis=1)
```

Gambar 4.25 Hapus Kolom *Cluster*

3. Pengujian *Silhouette Coefficient*

Setelah melakukan *clustering* dan berhasil mendapatkan jenis suasana hati pada hasil *cluster* maka selanjutnya kita akan melakukan pengujian hasil *cluster* menggunakan metrik *Silhouette Coefficient*, dengan menggunakan *library from sklearn.metrics import*

Silhouette_score. Berikut adalah kode yang digunakan untuk melihat hasil *score Silhouette* dari hasil *cluster* diatas, berikut adalah kode programnya pada gambar 4.26.



```
X = df[['instrumentalness', 'tempo']]
labels = kmeans.labels_

silhouette_avg = silhouette_score(X, labels)
print("Silhouette Coefficient: {:.3f}".format(silhouette_avg))
```

Gambar 4.26 Kode Program *Silhouette Score*

Pada kode yang pertama terdapat *library* yang mengimpor fungsi *Silhouette_score* dari *library sklearn*, selanjutnya *df_clustered* diatas merupakan *DataFrame* atau *dataset*,

selanjutnya pada selanjutnya kode: *Silhouette_avg = Silhouette_score(X, labels)*: untuk menghitung skor *Silhouette* rata-rata untuk semua sampel. Fungsi ini membutuhkan data asli 'X' dan label pengelompokan 'labels' sebagai input, selanjutnya pada kode dibawahnya untuk mencetak hasil dari skor *Silhouette* menjadi 3 angka. Jika program diatas sudah dirunning maka hasil skor *Silhouette* nya bisa dilihat pada gambar 4.27



Silhouette Coefficient: 0.547

Gambar 4.27 Hasil Skor *Silhouette*

Hasil dari skor *Silhouette* dari hasil *cluster* pada fitur "*instrumentalness*" dan "*tempo*" dengan n *cluster* nya 5 mendapat hasil skor *Silhouette* yaitu 0.547, berdasarkan interpretasi nilai *Silhouette Coefficient* pada tabel 3.2 skor *Silhouette* diatas mendapat interpretasi "struktur yang dihasilkan baik".

Dan berikut adalah sampel lagu yang berhasil didapatkan dari suasana hati *Calm*, sampel diatas telah berasal dari hasil *clustering* dan diambil dalam *dataset* yang telah disimpan. Untuk lebih jelasnya bisa dilihat pada tabel 4.1.

Tabel 4.1 Daftar Lagu Pada Suasana Hati *Calm*

Suasana Hati	Kategori Umur	Judul Lagu
<i>Calm</i>	Bayi	<i>Twinkle Twinkle Little Star</i>
	Bayi	<i>Hush Little Baby</i>
	Lansia	<i>Take on Me</i>
	Lansia	<i>Sweet Caroline</i>
	Anak-anak	Roda Roda Mobil Bis Berputar
	Anak-anak	Lihatlah Lebih Dekat
	Remaja	1000 Tahun Lamanya
	Remaja	Takkan Kemana
	Dewasa	Tenang
	Dewasa	Selaras

4.4.2. Mencari Suasana Hati *Energic*

1. Menentukan Jumlah *Cluster*

Sebelum melakukan *clustering* kita akan mencari nilai *n* dari fitur "*danceability*" dan "*tempo*" untuk mencari jumlah *cluster* nya berikut adalah kode programnya seperti pada gambar 4.28.

```

X = df[['danceability', 'tempo']]

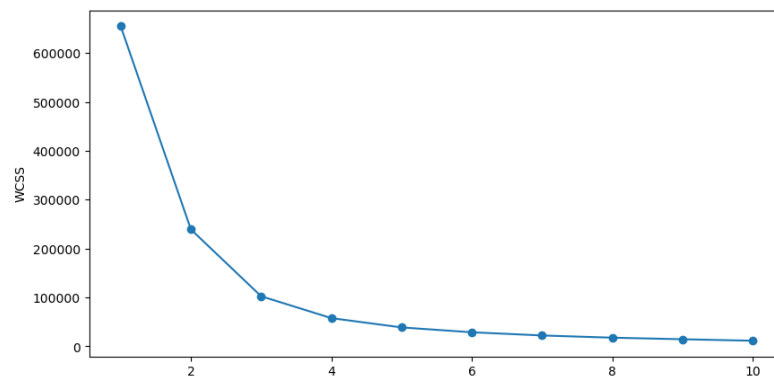
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10,
random_state=0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)

plt.figure(figsize=(10, 5))
plt.plot(range(1, 11), wcss, marker='o')
plt.title('Metode Elbow untuk Menentukan Jumlah k')
plt.xlabel('Jumlah kluster')
plt.ylabel('WCSS')
plt.show()

```

Gambar 4.28 Kode Program Menentukan Jumlah *Cluster*

Kode program diatas menjelaskan tentang kode untuk menentukan *n cluster* dengan menggunakan metode *elbow method* pada fitur "*danceability*" dan "*tempo*" yang terdapat pada *dataset*, setelah kode diatas dirunning maka hasilnya akan seperti pada gambar 4.29.

Gambar 4.29 Hasil *Elbow Method*

Hasil *running* diatas menjelaskan kalau penurunan *cluster* terjadi pada *cluster* ke 2/3/4, jadi n *cluster* yang direkomendasikan oleh *elbow method* adalah yaitu 2 3 dan 4. Tetapi disini saya menggunakan 10 *cluster*.

2. Clustering

Setelah mengetahui hasil n *cluster* dari *elbow method* nya maka selanjutnya kita akan melakukan proses *clustering* untuk mendapatkan atau mencari *cluster* dari jenis suasana hati *energetic*, berikut adalah kode programnya bisa dilihat pada gambar 4.30.

```
data = df
kmeans = KMeans(n_clusters=10, random_state=0)
kmeans.fit(df[['danceability', 'tempo']])
df['cluster'] = kmeans.labels_
cluster_data = {}

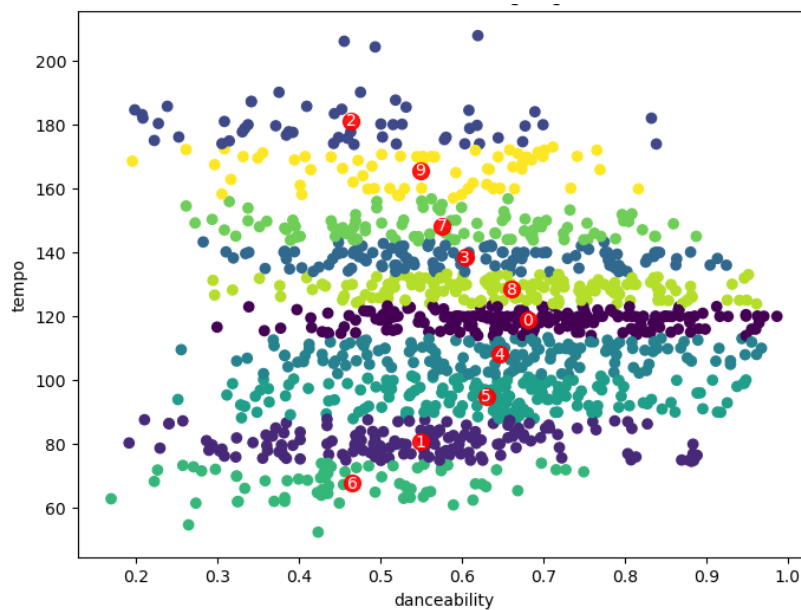
for k in range(kmeans.n_clusters):
    cluster_data[f'clus{k}'] = df[df['cluster'] == k].drop('cluster', axis=1)

centers = kmeans.cluster_centers_
print("Centroid dari setiap cluster:")
for i, center in enumerate(centers):
    print(f"Cluster {i}: {center}")

plt.figure(figsize=(8, 6))
plt.scatter(df['danceability'], df['tempo'], c=kmeans.labels_, cmap='viridis')
for i, center in enumerate(centers):
    plt.scatter(center[0], center[1], c='red', s=100, alpha=0.9)
    plt.text(center[0], center[1], str(i), color='white', ha='center', va='center')
plt.xlabel('danceability')
plt.ylabel('Tempo')
plt.title('Visualisasi Clustering Lagu')
plt.show()
```

Gambar 4.30 Kode Program *Clustering*

Pada kode diatas sama seperti kode sebelumnya hanya saja sedikit berbeda dengan kode sebelumnya yang mana kode diatas menampilkan atau memvisualisasikan hasil *cluster* seperti pada gambar 4.31.



Gambar 4.31 Visualisasi Hasil *Clustering*

Gambar diatas merupakan hasil *cluster* dari fitur '*danceability*' dan '*tempo*'. Pada sumbu (x), kita memiliki '*danceability*' yang nilainya berkisar antara 0 hingga 1, di mana nilai yang lebih tinggi menandakan bahwa sebuah lagu dianggap lebih cocok untuk menari. Sementara pada sumbu (y) yaitu *tempo*, nilai *tempo* ini menunjukkan kecepatan dan lambatnya sebuah lagu, jadi semakin tinggi nilai dari *tempo* maka semakin cepat lagu tersebut, dan semakin rendah nilai *tempo* maka semakin lambat sebuah lagu.

Titik-titik berwarna pada grafik menandakan kluster, atau grup lagu yang memiliki karakteristik serupa. Setiap warna merepresentasikan satu kluster yang berbeda. Titik-titik berwarna merah dengan angka adalah *centroid*, atau pusat kluster, yang mewakili nilai tengah dari fitur-fitur dalam kluster tersebut. Angka pada *centroid* menunjukkan *cluster* tersebut *cluster* berapa. Berikut penjelasan yang lebih spesifik untuk setiap *cluster*.

a. *Cluster 0* (ungu)

Dengan nilai '*danceability*' yang cenderung dominan pada nilai 0.4 hingga 1.0 dan *tempo* dari sekitar 115 hingga 125 bpm, Ini menunjukkan lagu-lagu yang mungkin tidak sangat kompleks dalam ritme tetapi masih memiliki ketukan yang cukup stabil dan menyenangkan, yang mungkin menawarkan tingkat energi yang moderat dan cocok untuk suasana yang positif dan menyenangkan. Berikut adalah gambar 4.32 data dari *cluster 0*.

cluster_data['clus0']

✓ 0.2s

		track_id	track_name	artist_name	danceability	energy	key	loudness	mode	speechiness
6	4GphHeTclqIEbY6jRMyaG	Lavender's Blue	Sleepy Shepherd	0.532	0.0323	5	-26.907	1	0.0441	
9	0nyFiAUU92dxCDbOJFHP9E	Rock A Bye Baby	Sleepy Shepherd	0.598	0.0318	7	-22.212	1	0.0614	
13	6mZQXPV4n8F8DEYGFx8O8k	Itsy Bitsy Spider	Sleepy Shepherd	0.508	0.0172	4	-26.652	1	0.0349	
...	...	...	...	...	...	...	...	...	...	
1217	5TRPicyLGbAF2LGBFbHGvO	Flashing Lights	Kanye West	0.639	0.6280	6	-7.578	0	0.0399	
1220	5hM5arv9KDbCHS0k9uqwjr	Borderline	Tame Impala	0.621	0.8730	5	-3.067	0	0.0369	
1230	4gZDCOFuLCxCRWNhSEcOtJ	These Days	Wallows	0.655	0.6680	0	-6.733	1	0.0272	

179 rows × 17 columns

Gambar 4.32 Hasil Data *Cluster 0*

Gambar diatas merupakan data dari hasil *cluster* yang ada pada *cluster 0*, *cluster* ini berhasil mendapatkan data dengan jumlah 179 data, data diatas akan dilakukan analisis lebih lanjut agar dapat mengetahui *cluster* diatas menghasilkan suasana hati apa.

b. *Cluster 1* (ungu terang)

Titik-titik dalam kluster ini cenderung memiliki '*danceability*' sedang, pada *range* 0.3 - 0.7, namun terdapat beberapa titik pada *cluster* ini yang memiliki nilai *danceability* yang cukup rendah dan cukup tinggi, ini dapat menandakan bahwa musik pada *cluster* ini sangat bervariasi dikarenakan terdapat beberapa kelompok dalam nilai

danceability, dan dengan '*tempo*' yang menyebar luas dari sekitar 75 hingga lebih dari 90 bpm. Meskipun *tempo* tidak terlalu cepat, '*danceability*' yang tinggi menunjukkan bahwa lagu-lagu ini memiliki kualitas yang membuat mereka enak untuk didengar dan mungkin memiliki elemen ritmis yang menarik membuatnya cukup menyenangkan. Berikut adalah gambar 4.33 data dari *cluster* 1.

cluster_data['clus1']
✓ 0.3s

	track_id	track_name	artist_name	danceability	energy	key	loudness	mode	speechiness
0	0tCqX7pASaZeHPC4FhAwe8	Brahms Lullaby	Sleepy Shepherd	0.390	0.0161	1	-29.263	1	0.0398
57	2xe9jw3QmcYTVcwreG221	Soon You'll Be Sleeping	Sleepy Shepherd	0.565	0.0123	8	-27.261	1	0.2030
85	60Ks638Zi0pMhw6cizQpvc	Trust Me	Orion The Owl	0.404	0.0130	6	-29.508	1	0.0566
...	...	...	...	...	...	...	...	...	...
1196	0rzaRSujxA0bKjJI6vHYq	Satellite	Harry Styles	0.576	0.4570	0	-6.473	1	0.0304
1219	6eVM9lCmlqwGTUOoeH61Lu	Hug Me	Pharrell Williams	0.842	0.6220	0	-5.437	1	0.0967
1225	42Lq6GBO5KUSNeSgJsOoke	Panama	Van Halen	0.514	0.9870	8	-5.391	1	0.1090

151 rows × 10 columns

Gambar 4.33 Hasil Data *Cluster* 1

Gambar diatas merupakan data dari hasil *cluster* yang ada pada *cluster* 1, *cluster* ini berhasil mendapatkan data dengan jumlah 151 data, data diatas akan dilakukan analisis lebih lanjut agar dapat mengetahui *cluster* diatas menghasilkan suasana hati apa.

c. *Cluster* 2(biru tua)

Cluster memiliki nilai '*danceability*' yang cenderung tinggi dan dominan pada sekitar 0.2 hingga 0.7, pada *cluster* ini juga nilai *danceability* terdapat 2 titik yang berada pada nilai 0.85, selain itu titik-titik yang ada pada *cluster* ini sedikit berjarak, yang menandakan ketidakstabilan. dan '*tempo*' berkisar antara 175 -210 bpm. Lagu-lagu dalam kluster ini kemungkinan adalah yang paling energik, harusnya seperti itu, tetapi saat validasi secara manual *cluster* ini juga banyak mengandung musik dengan suasana hati lain, dikarenakan nilai dari *danceability* yang cukup bervariasi. Berikut

adalah gambar 4.34 data dari *cluster 2*.

`cluster_data['clus2']`
✓ 0.1s

	track_id	track_name	artist_name	danceability	energy	key	loudness	mode	speechiness
2	0KOOhxPmW5pfMXjgWzRf0j	Goodnight Little Dreamer	Little Dreamer	0.309	0.00781	2	-30.090	1	0.0557
29	7nTghMbNP3rDEzStg5NGlz	Into The Clouds	Sleepy Shepherd	0.317	0.05420	4	-28.228	1	0.0445
44	4RyXEW4Pfql6Zdhl1k6Ws	At Dawn	Sleepy Shepherd	0.262	0.01210	1	-29.858	1	0.0461
131	6ZSoBwFaNkg54DRD59Vcnb	Alle meine Fingerlein wollen heute Tiere sein	Simone Sommerland	0.702	0.61300	2	-7.279	1	0.0352
140	38lz2x1F8QzHRejGVoeWlt	Leise, Peterle leise	Simone Sommerland	0.645	0.22600	8	-12.032	1	0.0331
...	...	...	...	...	...	...	...	...	...
1194	1BxfuPKGuaTgP7aM0Bbdwr	Cruel Summer	Taylor Swift	0.552	0.70200	9	-5.707	1	0.1570
1197	4k6Uh1HXdhtusDW5y8Gbvy	Bad Habit	Steve Lacy	0.686	0.49400	1	-7.093	1	0.0355
1229	7p4vHnYXkxlvzfePJVpcTr	Feels Like Summer	Childish Gambino	0.550	0.59600	7	-8.214	1	0.1340
1231	6kE3BJhZYLifGLQY2hSzaM	Younger	The Hails	0.700	0.44400	9	-6.614	1	0.0349
1236	2h4BV7jx9Ltz9NxC2zgNcW	Now or Never (feat. Charlie Gillespie, Owen Pa...	Julie and the Phantoms Cast	0.449	0.95700	2	-3.797	1	0.0610

63 rows × 17 columns

Gambar 4.34 Hasil Data *Cluster 2*

Gambar diatas merupakan data dari hasil *cluster* yang ada pada *cluster 2*, *cluster* ini berhasil mendapatkan data dengan jumlah 63 data, data diatas akan dilakukan analisis lebih lanjut agar dapat mengetahui *cluster* diatas menghasilkan suasana hati apa. Analisis lebih lanjut yang dimaksud yaitu analisis yang dilakukan secara manual dan otomatis.

d. *Cluster 3*(biru muda)

Memiliki '*danceability*' yang cukup konsisten dan dominan pada angka 0.3 – 0.9, *cluster* ini bisa menjadi salah satu kandidat untuk menjadi *cluster energic*, namun nilai *tempo* nya tergolong sedang dan berkisar pada angka 135 – 145 bpm. Yang menjadikan *cluster* ini kurang dalam hal kecepatan musik. Berikut adalah gambar 4.35

data dari *cluster 3*.

```
cluster_data['clus3']
```

✓ 0.2s

	track_id	track_name	artist_name	danceability	energy	key	loudness	mode	speechiness
17	1VvLz9pvdKxhKP3jJwEilQ	Night Sky	Sleepy Shepherd	0.416	0.02100	10	-28.794	1	0.0499
18	1Dmu3GiJut8z1QzelyKxmK	Singing For You	Sleepy Shepherd	0.506	0.01460	1	-29.600	0	0.0704
39	5OSd2WXopCJBuXEbHnp3ko	Midnight Reverie	Sleepy Shepherd	0.508	0.01550	11	-28.087	0	0.0720
53	5psg80rmRzpw0oolwiDi3j	My Sun	Sleepy Shepherd	0.415	0.03790	11	-25.424	1	0.0726
54	5z1qXzXd8UnTYiZzdRe9RL	Faith	Sleepy Shepherd	0.300	0.00588	6	-33.967	1	0.0450
...	...	...	...	...	...	...	...	...	...
1190	29FQeJUt8AnxWEkux39d7l	O Children	Nick Cave & The Bad Seeds	0.484	0.55400	4	-7.064	0	0.0324
1211	3SdTKo2uVsxFblQjpScoHy	Stand By Me	Ben E. King	0.650	0.30600	9	-9.443	1	0.0393
1213	1JSTJqkT5qHq8MDJnJbRE1	Every Breath You Take	The Police	0.820	0.45200	1	-9.796	1	0.0348
1223	3YJJjQPAbDT7mGpX3WtQ9A	Good Days	SZA	0.436	0.65500	1	-8.370	0	0.0583
1226	4mGJcOKQhmISIWgQ3xCiLB	Til I Forget About You	Big Time Rush	0.568	0.86500	5	-3.653	1	0.0486

192 rows × 17 columns

Gambar 4.35 Hasil Data *Cluster 3*

Gambar diatas merupakan data dari hasil *cluster* yang berada pada *cluster 3*, *cluster* ini berhasil mendapatkan data dengan jumlah 192 data, data diatas akan dilakukan analisis lebih lanjut lagi agar dapat mengetahui *cluster* diatas berhasil menghasilkan suasana hati apa. Analisis lebih lanjut yang dimaksud yaitu analisis yang dilakukan secara manual dan otomatis.

e. *Cluster 4* (hijau toska tua)

Cluster ini memiliki nilai *danceability* yang dominan pada angka 0.3 – 0.95. dan nilai *tempo* yang berkisar pada angka 100 – 115 bpm, *cluster* ini sama seperti *cluster* sebelumnya yang tidak bisa dikategorikan menjadi suasana hati *energetic* karena memiliki nilai *tempo* yang cukup rendah, namun nilai dari *danceability* pada *cluster* ini bisa berhubungan dengan suasana hati *happy* atau yang lain. Berikut adalah gambar

4.36 data dari *cluster* 4.

```
cluster_data['clus4']
```

✓ 0.3s

	track_id	track_name	artist_name	danceability	energy	key	loudness	mode	speechiness
8	5znZqW4EAEvy6Oz8H5zSE2	You Are My Sunshine	Sleepy Shepherd	0.494	0.0225	8	-28.313	1	0.0361
12	1B0SN5PHT51NMy4kvIqrAy	Baa Baa Black Sheep	Sleepy Shepherd	0.538	0.0333	4	-23.151	1	0.0366
16	3AaXL2IS7KGyazdQ1SDt4i	Peaceful Sunset	Sleepy Shepherd	0.546	0.0192	0	-29.264	1	0.1890
19	6sa1YuE21HhVP0uif0TH01	Fade Away	Sleepy Shepherd	0.383	0.0115	6	-31.196	1	0.0598
20	7Gkc6fFPU5V6wDY06bAsvE	Misty Ocean	Sleepy Shepherd	0.423	0.0107	1	-28.762	1	0.0672
...	...	...	...	...	...	...	...	...	...
1019	4ZG4aqyBvKk3V3yZBoLT0K	Sulung	Kunto Aji	0.424	0.2930	11	-12.723	1	0.0306
1020	1k1e4Amkn7hIFrEhtazqT	Tutur Batin	Yura Yunita	0.576	0.3290	10	-8.870	1	0.0487
1051	0Z3AGsU11hASarHrOPLUn9	It Took Me By Surprise	Maria Mena	0.614	0.5320	7	-7.095	0	0.0505
1168	7oDd86yk8itslrA9HRP2ki	Die For You - Remix	The Weeknd	0.531	0.5250	1	-6.500	0	0.0671
1215	2HbKqm4o0w5wEeEFXm2sD4	Money Trees	Kendrick Lamar	0.716	0.5310	7	-7.355	1	0.1220

5 rows × 17 columns

Gambar 4.36 Hasil Data *Cluster* 4

Gambar diatas merupakan data dari hasil *cluster* yang ada pada *cluster* 4, *cluster* ini berhasil mendapatkan data dengan jumlah 65 data, data diatas akan dilakukan analisis lebih lanjut agar dapat mengetahui *cluster* diatas menghasilkan suasana hati apa. Analisis lebih lanjut yang dimaksud yaitu analisis yang dilakukan secara manual dan otomatis.

f. *Cluster* 5(hijau toska muda)

Cluster ini hampir mirip dengan *cluster* sebelumnya dan hanya berbeda pada nilai *Tempo*-nya yaitu 85 – 105 bpm, *cluster* ini sama seperti *cluster* sebelumnya yang tidak bisa dikategorikan menjadi suasana hati *energetic* karena memiliki nilai *tempo* yang cukup rendah, namun nilai dari *danceability* pada *cluster* ini bisa berhubungan dengan suasana hati, karena memiliki *tempo* yang cukup rendah. Berikut adalah gambar 4.37 data dari *cluster* 5.

`cluster_data['clus5']`
✓ 0.1s

	track_id	track_name	artist_name	danceability	energy	key	loudness	mode
14	01b3JCDs8aA4D6Rglx8yhX	Trust Me	Sleepy Shepherd	0.536	0.01280	10	-24.603	1
15	1K7JgK79uuxSOpbB415lpM	Daydreaming	Sleepy Shepherd	0.413	0.03540	8	-31.037	1
26	6qx26R8PSUepaoKg2k5tn	Calmness	Sleepy Shepherd	0.430	0.01630	10	-26.423	0
40	1An66SE1eGM6FGmz2n6uOZ	In The Stars	Sleepy Shepherd	0.370	0.00581	6	-30.101	1
80	3841QJd4WvMNHUND8qwTkO	Fading Memories	Orion The Owl	0.498	0.01390	1	-28.964	1
...	...	...	...	...	...	...	...	...
1205	3QyoC6OvQUmpQwQZ18iaTs	Overdrive	Conan Gray	0.568	0.78800	5	-6.280	1
1222	5mCPDVBb16L4XQwDdbRUpz	Passionfruit	Drake	0.809	0.46300	11	-11.377	1
1232	796gbCqHEHWvEqtYOBXibL	Fake a Frown	The Hails	0.758	0.63400	6	-6.851	0
1233	0oUBuOO4g9P4IREqfqR5nq	A Man Without Love	Engelbert Humperdinck	0.538	0.46700	2	-9.589	1
1238	7CBXta3Je26MMHOFGERpIS	Miss The Rage - Remix	Toxic Tripz	0.514	0.65900	8	-7.903	1

158 rows × 17 columns

Gambar 4.37 Hasil Data *Cluster 5*

Gambar diatas merupakan data dari hasil *cluster* yang ada pada *cluster 5*, *cluster* ini berhasil mendapatkan data dengan jumlah 158 data, data diatas akan dilakukan analisis lebih lanjut agar dapat mengetahui *cluster* diatas menghasilkan suasana hati apa. Analisis lebih lanjut yang dimaksud yaitu analisis yang dilakukan secara manual dan otomatis.

g. *Cluster 6*(hijau pudar)

Cluster ini memiliki nilai *danceability* yang cukup berjarak dan dominan pada angka 0.2 – 0.75, akan tetapi terdapat 1 titik *cluster* yang memiliki nilai hampir 0.1, titik ini mungkin musik sedih atau *calm*, namun bukan hanya titik itu saja, karena nilai *danceability* pada *cluster* ini juga sangat rendah, nilai *tempo* pada *cluster* ini berkisar pada angka 0.65 – 0.75 bpm yang bisa dibilang sangat rendah, selain itu *cluster* ini lebih cenderung ke musik sedih atau *calm*. Berikut adalah gambar 4.38 data dari *cluster 6*.

```
cluster_data['clus6']
```

✓ 0.0s

	track_id	track_name	artist_name	danceability
99	2eOGhMShEgQtmhCYFX3SNL	Guten Tag, Herr Nasenmann	Simone Sommerland	0.694
115	7Dqc8uDRihirW35auTWGeo	Imse Bimse Spinne	Simone Sommerland	0.809
124	3mlBmf67xsoQKlcJ55VKPt	Liebe, liebe Sonne	Simone Sommerland	0.600
126	3RJ8LNwvlyTMREfmcldthoK	Schotter fahren	Simone Sommerland	0.817
153	67lxdQSW4iQ5UWf4pDAIwB	Ich bin ein kleines Pony	Simone Sommerland	0.797
...	...	...	...	...
1195	169pWCmfVvazzcL4EXFPgL	You Know It	Colony House	0.404
1204	79EkGysjP2dL5GdpeQjRxT	This Side of Paradise	Coyote Theory	0.306
1221	52ojopYMUzeNcudsoz7O9D	New Person, Same Old Mistakes	Tame Impala	0.470
1224	5FS6K70gANsecaAtXfVbR	When Will I See You Again	Shakka	0.593
1239	4p5Phj0pcXWW7B81IUGowi	masquerade x miss the rage	acti	0.496

80 rows × 5 columns

Gambar 4.38 Hasil Data *Cluster 6*

Gambar diatas merupakan data dari hasil *cluster* yang ada pada *cluster 6*, *cluster* ini berhasil mendapatkan data dengan jumlah 80 data, data diatas akan dilakukan analisis lebih lanjut agar dapat mengetahui *cluster* diatas menghasilkan suasana hati apa.

h. *Cluster 7* (hijau muda)

cluster ini berhasil mendapatkan data dengan jumlah 163 data, data diatas akan - dilakukan analisis lebih lanjut agar dapat mengetahui *cluster* diatas menghasilkan suasana hati apa. Berikut adalah gambar 4.39 data dari *cluster 7*.

	track_id	track_name	artist_name	danceability
5	1EeRSC5XOhhcvGdhSHjil	Golden Sunrise	Chillboy	0.561
10	7DOQ5HUSJeHYLmOhtGKz3y	Are You Sleeping	Sleepy Shepherd	0.518
27	37ctH3ngeJXnVz7mua4LkF	Floating	Sleepy Shepherd	0.528
30	7FL2BfLeJWqBTO7gf0KcGG	Reflection	Sleepy Shepherd	0.564
32	73VoUy/WI4oqU0d4O32G5V	Eternity	Sleepy Shepherd	0.471
...	...	...	...	...
1180	7h2nmmoWDi2UpfYKLKWLYB	Sunflower	Rex Orange County	0.548
1187	4DMKwE2E2iYDKY01C335Uw	Carry on Wayward Son	Kansas	0.482
1227	6knXMt7UIPPJveAtsP1Dq4	Don't Run Away	Tyler James Williams	0.747
1228	5ZkAx8zjLiSs1nMmBwJoZS	When Can I See You Again? - From "Wreck-It Ral...	Owl City	0.638
1234	7ETjvgUdK7rBlGfwhiWQj	Stay	The Hails	0.665

163 rows × 5 columns

Gambar 4.39 Hasil Data *Cluster 7*

Gambar diatas merupakan data dari hasil *cluster* yang ada pada *cluster 7*, *Cluster danceability* yang cukup tinggi yang dominan pada angka 0.3 – 0.9, dan *tempo* yang berkisar pada angka 140 – 160 bpm, *cluster* ini menjadi salah satu kandidat untuk menjadi *cluster energetic*, namun setelah dilakukan validasi secara manual ternyata *cluster* ini juga masih banyak terkandung musik dari suasana hati lain, oleh karena itu *cluster* ini tidak dijadikan *cluster energetic*. Berikut adalah gambar 4.39 data dari *cluster 7*.

i. *Cluster 8*(hijau terang)

Cluster ini memiliki nilai *danceability* yang dominan pada angka 0.3 – 0.95, dan *tempo* yang berkisar pada angka 120 – 135 bpm, dimana *cluster* ini sama seperti *cluster* sebelum-sebelumnya yang memiliki nilai *danceability* yang cukup tinggi namun nilai *tempo* nya yang sedang yang menandakan *cluster* ini tidak mengandung suasana hati *energetic*. Berikut adalah gambar 4.40 data dari *cluster 8*.

cluster_data['clus8']
✓ 0.0s

	track_id	track_name	artist_name	danceability
11	6lpb4EVYeX94RQb2mo3AUr	My Bonnie Lies Over The Ocean	Sleepy Shepherd	0.338
136	2lqkiaEx2wzCxtDgFkueDb	Guten Morgen, Frau Sonne	Simone Sommerland	0.678
165	5cY3LZLYuPxfVjvLz1UQ	Hopp, hopp, hopp zu Pferde	Simone Sommerland	0.833
196	2QAhVXzAEy2UPU60ccqnTY	Ich bin ein dicker Tanzbär	Simone Sommerland	0.609
239	70C4NyhjD5OZUMzvWZ3njJ	Piano Man	Billy Joel	0.331
257	2Ng4Z0zer2QqmwEmKUQGb	Women in Uniform - 1994 Remaster	Skyhooks	0.334
265	34sL4eal8UKWOyYpCvoboU	All I Want Is You	U2	0.239
327	4ngq4oBo2zj0uG5L7SY5D9	Tiada Cinta Yang Lain	Inka Christie	0.393
335	315GhPGKOywcVC72puOwyq	Bing	Titiek Puspa	0.199
417	6JRCrmqZ2WeUG1gWjZFgEr	Makan Jangan Asal Makan	Nussa	0.620

Gambar 4.40 Hasil Data *Cluster 8*

Gambar diatas merupakan data dari hasil *cluster* yang ada pada *cluster 8*, *cluster* ini berhasil mendapatkan data dengan jumlah 35 data, data diatas akan dilakukan analisis lebih lanjut agar dapat mengetahui *cluster* diatas menghasilkan suasana hati apa.

j. *Cluster 9*(kuning)

Yang terakhir *cluster* ini secara visual memiliki nilai *danceability* yang sedikit berjarak dan yaitu berkisar dan dominan pada angka 0.3 - 0.8, nilai dari *tempo* yaitu 155 – 170 bpm, *cluster* ini setelah dilakukan validasi secara manual ternyata *cluster* ini banyak mengandung suasana hati *energetic* walaupun terdapat beberapa titik dalam *cluster* ini yang memiliki nilai *danceability* yang cukup rendah tetapi *cluster* ini didominasi oleh musik *energetic*. Dan *cluster* ini lah yang dijadikan dan ditambahkan ke *dataset* menjadi *cluster* suasana hati *energetic*. Berikut adalah gambar 4.41 data dari *cluster 9*.

cluster_data['clus9']
✓ 0.2s

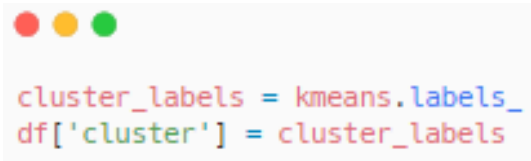
	track_id	track_name	artist_name	danceability
1	2DGSytzXs9oqYUy5tqxfSB	Twinkle Twinkle Little Star	Tommy Twinkle	0.550
3	4tbnZK9nKvuyCrqyu8o0eG	Hush Little Baby	Sleepy Shepherd	0.554
4	5YMmjaZEzUoI8EFZm4q2fB	Moonlight Walk	Orion The Owl	0.479
7	5NI8DkhAIQjkci2AiJI90h	Wake Up	Bluedon	0.618
21	4gbA6AMrzEdAfjMrN6MIUS	Rain Outside	Sleepy Shepherd	0.290
...	...	...	...	...
1212	1JLn8RhQzHz3qDqsChcmBI	I Want to Know What Love Is - 1999 Remaster	Foreigner	0.477
1214	63BcfK6YAzJYelSaTPr6IO	Big Poppa - 2007 Remaster	The Notorious B.I.G.	0.765
1216	1UGD3IW3tDmgZfAVDh6w7r	Devil In A New Dress	Kanye West	0.435
1218	51EC3I1nQXpec4gDk0mQyP	90210 (feat. Kacy Hill)	Travis Scott	0.402
1235	3usbnvDftOhY09cRNar8Zg	FRIENDS	LAUNDRY DAY	0.522

154 rows × 5 columns

Gambar 4.41 Hasil Data *Cluster 9*

Gambar diatas merupakan data dari hasil *cluster* yang ada pada *cluster 9*, *cluster* ini berhasil mendapatkan data dengan jumlah 154 data, data diatas akan dilakukan analisis lebih lanjut agar dapat mengetahui *cluster* diatas menghasilkan suasana hati apa.

Setelah mendapatkan hasil *cluster*, selanjutnya kita akan menambahkan hasil *cluster* tersebut kedalam *dataset* untuk mencari dari ke 10 *cluster* tersebut manakah yang menghasilkan suasana hati *energetic*, berikut adalah kodenya bisa dilihat pada gambar 4.42.

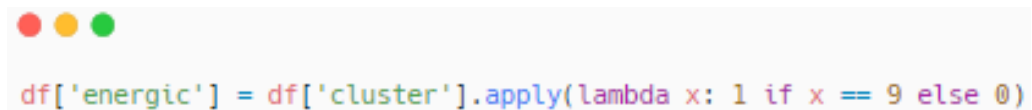


```
cluster_labels = kmeans.labels_
df['cluster'] = cluster_labels
```

Gambar 4.42 Program Untuk Menambahkan Hasil *Cluster* Ke *Dataset*

Setelah menjalankan kode diatas hasil *cluster* akan ditambahkan kedalam *dataset* bernama *cluster* untuk diseleksi *cluster* mana yang menghasilkan suasana hati *energetic*, seleksi dilakukan secara manual, jadi musik dari *cluster* 1-10 akan dicek satu persatu, jadi dari ke sepuluh *cluster* tersebut *cluster* berapakah yang bisa menghasilkan suasana hati *calm*.

Setelah mengetahui *cluster* berapa yang menghasilkan suasana hati *energetic* selanjutnya kita akan menambahkan *cluster* tersebut ke dalam *dataset*, kodenya bisa dilihat pada gambar 4.43.



```
df['energetic'] = df['cluster'].apply(lambda x: 1 if x == 9 else 0)
```

Gambar 4.43 Kode Program Menambah Hasil Seleksi Suasana Hati *Energetic*

Setelah menjalankan kode diatas maka sistem akan menambahkan kolom baru pada *dataset* dengan nama *energetic*, jadi kode diatas juga menandai lagu-lagu yang termasuk dalam *cluster* 9 sebagai jenis suasana hati *energetic*, jadi pada kolom *energetic* nilainya akan menjadi 1 jika dia merupakan *cluster* 9, jika bukan maka akan diberi nilai 0 pada kolom *energetic*.

3. Pengujian *Silhouette Coefficient*

Setelah mendapatkan data hasil dari *clustering* musik sebelumnya, maka selanjutnya kita akan menguji hasil *cluster* dari kedua fitur *danceability* dan *tempo* dengan n *cluster* nya 10 dan berhasil menghasilkan skor *Silhouette* berapa, berikut adalah kode programnya pada

gambar 4.44.



```
X = df[['danceability', 'tempo']]
labels = kmeans.labels_

silhouette_avg = silhouette_score(X, labels)
print("Silhouette Coefficient: {:.3f}".format(silhouette_avg))
```

Gambar 4.44 Kode Program *Silhouette*

Pada kode yang pertama terdapat *library* yang mengimpor fungsi *Silhouette_score* dari *library sklearn*, selanjutnya *df_clustered* diatas merupakan *DataFrame* atau *dataset*, selanjutnya pada selanjutnya kode: *Silhouette_avg = Silhouette_score(X, labels)*: untuk menghitung skor *Silhouette* rata-rata untuk semua sampel. Fungsi ini membutuhkan data asli 'X' dan label pengelompokan '*labels*' sebagai input, dan selanjutnya pada kode dibawahnya yaitu berfungsi untuk mencetak hasil dari skor *Silhouette* menjadi 3 angka dibelakang koma. Dan jika program diatas sudah di *running* maka hasil skor *Silhouette* nya bisa dilihat pada gambar 4.45.



Silhouette Coefficient: 0.547

Gambar 4.45 Hasil Skor *Silhouette*

Hasil dari skor *Silhouette* dari hasil *cluster* pada fitur "*danceability*" dan "*tempo*" dengan n *cluster* nya 10 mendapat hasil skor *Silhouette* yaitu 0.547, berdasarkan interpretasi nilai *Silhouette Coefficient* pada tabel 3.2 skor *Silhouette* diatas mendapat interpretasi "struktur yang dihasilkan baik".

Dan berikut adalah sampel lagu yang berhasil didapatkan dari suasana hati *Energic*, sampel diatas telah berasal dari hasil *clustering* dan diambil dalam *dataset* yang telah disimpan. Untuk lebih jelasnya bisa dilihat pada tabel 4.2.

Tabel 4.2 Daftar Lagu Pada Suasana Hati *Energic*

Suasana Hati	Kategori Umur	Judul Lagu
<i>Energic</i>	Bayi	<i>Goodnight Little Dreamer</i>
	Bayi	<i>Into The Clouds</i>
	Lansia	<i>Rain</i>
	Lansia	<i>Viva Forever</i>
	Anak-anak	Andai Aku Besar Nanti
	Anak-anak	Hot Potato
	Remaja	Wish You Were Here
	Remaja	Superheroes
	Dewasa	<i>I Am Falling For You</i>
	Dewasa	<i>You Know It</i>

4.4.3. Mencari Suasana Hati *Sad*

1. Menentukan Jumlah *Cluster*

Sebelum melakukan *clustering* kita akan mencari nilai *n cluster* terlebih dahulu pada fitur “*acousticness*” dan “*valence*” untuk mencari jumlah *cluster* kita akan menggunakan *Elbow Method*, berikut adalah kode programnya seperti pada gambar 4.46.

```

X = df[['acousticness', 'valence']]

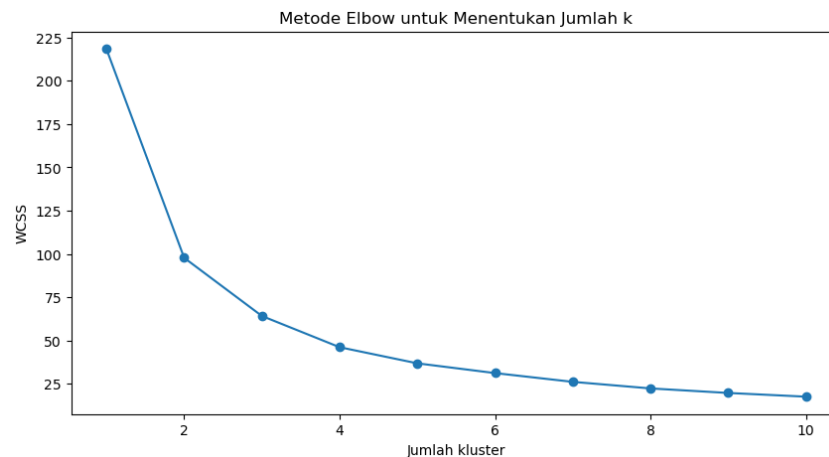
wcss = []
for i in range(1, 11): # Mencoba k dari 1 sampai 10
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10, random_state=0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)

plt.figure(figsize=(10, 5))
plt.plot(range(1, 11), wcss, marker='o')
plt.title('Metode Elbow untuk Menentukan Jumlah k')
plt.xlabel('Jumlah kluster')
plt.ylabel('WCSS')
plt.show()

```

Gambar 4.46 Kode Program Untuk Mencari *n Cluster*

Kode program diatas menjelaskan tentang kode untuk menentukan *n cluster* atau jumlah cluster yang akan dipakai, menggunakan *Elbow Method* pada fitur “*acousticness*” dan “*valence*” yang terdapat pada *dataset*, setelah kode diatas di *running* maka hasilnya akan seperti pada gambar 4.47.

Gambar 4.47 Hasil *Elbow Method*

Hasil *running* diatas menjelaskan kalau penurunan *cluster* terjadi pada *cluster* ke 2-4, maka dari itu n *cluster* yang direkomendasikan oleh *Elbow Method* adalah 2-4.

2. Clustering

Setelah mengetahui hasil n *cluster* dari *Elbow Method* nya maka selanjutnya kita akan melakukan proses *clustering* untuk mendapatkan atau mencari *cluster* dari jenis suasana hati *sad*, berikut adalah kode programnya bisa dilihat pada gambar 4.48.

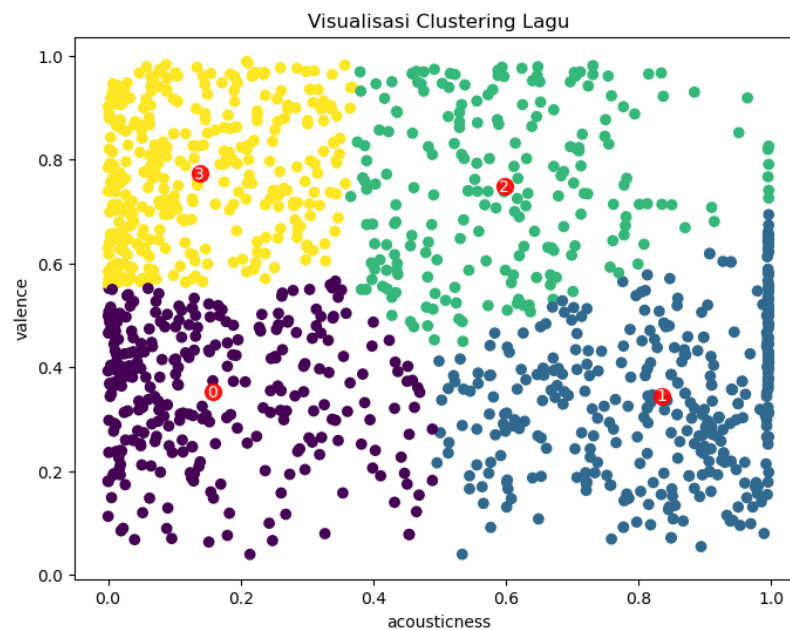
```
data = df
kmeans = KMeans(n_clusters=4, random_state=0)
kmeans.fit(df[['acousticness', 'valence']])
df['cluster'] = kmeans.labels_
cluster_data = {}

for k in range(kmeans.n_clusters):
    cluster_data[f'clus{k}'] = df[df['cluster'] == k].drop('cluster', axis=1)
centers = kmeans.cluster_centers_
print("Centroid dari setiap cluster:")
for i, center in enumerate(centers):
    print(f"Cluster {i}: {center}")

plt.figure(figsize=(8, 6))
plt.scatter(df['acousticness'], df['valence'], c=kmeans.labels_, cmap='viridis')
for i, center in enumerate(centers):
    plt.scatter(center[0], center[1], c='red', s=100, alpha=0.9)
    plt.text(center[0], center[1], str(i), color='white', ha='center',
va='center')
plt.xlabel('acousticness')
plt.ylabel('valence')
plt.title('Visualisasi Clustering Lagu')
plt.show()
```

Gambar 4.48 Kode Program *Clustering*

Pada kode diatas sama seperti kode *clustering* sebelumnya hanya saja sedikit berbeda dengan kode sebelumnya yang mana kode diatas menampilkan atau memvisualisasikan hasil *cluster* yang bisa dilihat pada gambar 4.49.



Gambar 4.49 Visualisasi Hasil *Clustering*

Pada sumbu horizontal (x), kita memiliki '*acousticness*' yang berkisar antara 0 hingga 1, dengan nilai yang lebih tinggi menunjukkan bahwa sebuah lagu memiliki banyak elemen akustik, seperti gitar akustik atau piano, yang berarti lagu tersebut cenderung memiliki sedikit atau tanpa efek elektronik. Sebaliknya, nilai yang lebih rendah menandakan bahwa sebuah lagu lebih elektronik atau diproduksi secara digital. Pada sumbu vertikal (y), kita memiliki '*valence*', juga berkisar dari 0 hingga 1, yang menggambarkan keceriaan musik. Nilai yang lebih tinggi untuk '*valence*' menunjukkan bahwa lagu tersebut terdengar lebih positif (ceria, bahagia), sedangkan nilai yang lebih rendah menandakan bahwa lagu tersebut terdengar lebih negatif (sedih, marah).

Warna-warna yang berbeda pada *plot* menunjukkan kluster yang berbeda, dimana setiap kluster terdiri dari lagu-lagu yang memiliki karakteristik '*acousticness*' dan '*valence*'

yang serupa. Titik-titik berwarna merah dengan angka merupakan pusat dari masing-masing kluster (*centroid*). Angka tersebut menandai kluster mana yang dimaksud. Berikut penjelasan yang lebih spesifik untuk setiap *cluster*.

a. *Cluster 0*(ungu)

Cluster ini bisa memiliki nilai '*valence*' yang sangat rendah, dan dominan antara 0.0 hingga 0.55, yang menunjukkan bahwa lagu-lagu di *cluster* ini cenderung memiliki emosi yang lebih negatif dan dapat dikaitkan dengan perasaan sedih atau murung. '*Acousticness*' di *cluster* ini memiliki nilai yang cenderung dominan pada angka 0.0 - 0.5, menunjukkan bahwa lagu-lagu ini lebih cenderung memiliki elemen elektronik atau terproduksi daripada suara akustik yang murni. Berikut gambar 4.50 adalah data dari *cluster 0*.

```
cluster_data['clus0']
```

✓ 0.0s

	track_id	track_name	artist_name	danceability
0	0tCqX7pASaZeHPC4FhAwe8	Brahms Lullaby	Sleepy Shepherd	0.390
2	0KOOhxPmW5pfMXjgWzRf0j	Goodnight Little Dreamer	Little Dreamer	0.309
3	4tbnZK9nKvuyCrqyu8o0eG	Hush Little Baby	Sleepy Shepherd	0.554
4	5YMmjaZEzUol8EFZm4q2fB	Moonlight Walk	Orion The Owl	0.479
5	1EvERSC5XOhhcvGdhSHjil	Golden Sunrise	Chillboy	0.561
...	...	...	...	...
1163	0TEekvXTomKt3hdXDZxeW	Ribs	Lorde	0.511
1190	29FQEJUtbAnxWEkux39d7l	O Children	Nick Cave & The Bad Seeds	0.484
1200	4P9Q0G0jKVXpRTJCaL3kyy	All Of The Girls You Loved Before	Taylor Swift	0.718
1209	1n7xFAY4xoPeqRvrkzAtsw	Good Old-Fashioned Lover Boy - Remastered 2011	Queen	0.543
1223	3YJJQPAbDT7mGpX3WtQ9A	Good Days	SZA	0.436

373 rows × 5 columns

Gambar 4.50 Hasil Data *Cluster 0*

Gambar diatas merupakan data dari hasil *cluster* yang ada pada *cluster 0*, *cluster* ini berhasil mendapatkan data dengan jumlah 373 data, data diatas akan dilakukan analisis lebih lanjut agar dapat mengetahui *cluster* diatas menghasilkan suasana hati apa.

b. *Cluster 1*(biru)

Dapat diinterpretasikan bahwa *cluster* ini memiliki lagu-lagu dengan '*valence*' yang sedang, dan cenderung dominan dari 0.1 hingga 0.65. Lagu-lagu di sini mungkin mewakili suasana hati yang lebih reflektif atau introspektif, dengan nilai '*acousticness*' yang cenderung pada angka 0.5 – 1.0, secara visual terlihat bahwa pada *cluster* ini terdapat titik titik yang cukup padat pada *acousticness* di sekitaran nilai 1.0 menandakan bahwa ada campuran antara elemen akustik dan non-akustik dalam musik tersebut. Dan setelah divalidasi *cluster* ini ternyata menghasilkan banyak musik dengan suasana hati *sad*. Berikut adalah gambar 4.51 data dari *cluster 1*.

cluster_data['clus1']
✓ 0.0s

	track_id	track_name	artist_name	danceability
99	2eOGhMShEgQtmhCYFX3SNL	Guten Tag, Herr Nasenmann	Simone Sommerland	0.694
103	4luZillP3swmJu03JXBbJG	Der Apfelbaum	Simone Sommerland	0.794
106	28JyjquxY092zwjUlbfg6r	Unser kleiner Bär im Zoo	Simone Sommerland	0.720
111	52V5Q8MypzMZdnPlxs3hP4	Das Pflaumenbäumchen	Simone Sommerland	0.918
114	4jCk2fuVAAcZCfjy7i6k05	Steigt ein Bübchen auf den Baum	Simone Sommerland	0.794
...	...	...	...	...
1232	796gbCqHEHWvEqYOBXibL	Fake a Frown	The Hails	0.758
1234	7ETjvgUdK7rBIGfviWQj	Stay	The Hails	0.665
1236	2h4BV7jx9Ltz9NxC2zgNcW	Now or Never (feat. Charlie Gillespie, Owen Pa...	Julie and the Phantoms Cast	0.449
1237	678tfxlNbh8mCDR2L2l8qd	MONTERO (Call Me By Your Name)	Lil Nas X	0.610
1239	4p5Phj0pcXWW7B81IUGowi	masquerade x miss the rage	acti	0.496

341 rows × 5 columns

Gambar 4.51 Hasil Data *Cluster 1*

Gambar diatas merupakan data dari hasil *cluster* yang ada pada *cluster 1*, *cluster* ini berhasil mendapatkan data dengan jumlah 341 data, data diatas akan dilakukan analisis lebih lanjut agar dapat mengetahui *cluster* diatas menghasilkan suasana hati apa.

c. *Cluster 2*(hijau)

Cluster ini memiliki nilai *valence* yang cenderung dominan 0.5 – 1.0, dan nilai dari *acousticness* antara 0.4 – 1.0, pada *cluster* ini bisa dilihat kalau terdapat titik-titik yang

tidak menyebar dengan baik pada nilai *acousticness* di *range* nilai 0.8 – 0.9, dengan nilai *valence* dan *acousticness* yang sangat tinggi *cluster* ini lebih mengandung musik yang secara emosional lebih ke suasana hati senang atau nostalgia ketimbang suasana hati *sad*. Berikut adalah gambar 4.52 data dari *cluster 2*.

`cluster_data['clus2']`
✓ 0.3s

	track_id	track_name	artist_name	danceability
139	7qKlhndhP3XIHhHvkaYbWw	Laterne, Laterne	Simone Sommerland	0.814
198	5Mz933z22hXnall8jLgQi	Baby Shark	Simone Sommerland	0.826
199	7FwBtcecmIpc1sLySPXeGE	Dancing In the Dark	Bruce Springsteen	0.527
202	3TW48DvQ7ChwY1Yy1tkMaP	Runaway Train (2022 Remaster)	Soul Asylum	0.613
203	4qgZs0RNjdZKAS22IP0QjY	Where The Streets Have No Name - Remastered	U2	0.490
...	...	...	...	...
1225	42Lq6GBO5KUSNeSgJsOoke	Panama	Van Halen	0.514
1226	4mGJcOKQhmlSIWgQ3xCiLB	Til I Forget About You	Big Time Rush	0.568
1228	5ZkAx8zjLiSs1nMmBwJoZS	When Can I See You Again? - From "Wreck-It Ral...	Owl City	0.638
1229	7p4vHnYXkxlvfePJVpcTr	Feels Like Summer	Childish Gambino	0.550
1238	7CBXta3Je26MMHOFGERpIS	Miss The Rage - Remix	Toxic Tripz	0.514

314 rows × 5 columns

Gambar 4.52 Hasil Data *Cluster 2*

Gambar diatas merupakan data dari hasil *cluster* yang ada pada *cluster 2*, *cluster* ini berhasil mendapatkan data dengan jumlah 314 data, data diatas akan dilakukan analisis lebih lanjut agar dapat mengetahui *cluster* diatas menghasilkan suasana hati apa.

d. *Cluster 3*(kuning)

Cluster ini memiliki '*valence*' yang cenderung dari 0.6 hingga 1.0, yang menunjukkan bahwa lagu-lagu tersebut cenderung menyampaikan emosi yang positif, bahagia, dan ceria. Nilai '*valence*' yang tinggi ini menggambarkan suasana hati yang umumnya lebih gembira dan hidup. Namun, dengan '*acousticness*' yang cenderung dominan dari 0.0 hingga 0.4, ini menunjukkan bahwa lagu-lagu tersebut mungkin lebih condong

pada produksi yang kurang akustik dan tidak sesuai karena '*valence*' yang tinggi dan '*acousticness*' yang rendah. Lagu-lagu dengan ciri seperti ini biasanya tidak mengekspresikan kesedihan, melainkan kebahagiaan dan semangat. Berikut adalah gambar 4.53 data dari *cluster* 3.

```
cluster_data['clus3']
```

✓ 0.1s

	track_id	track_name	artist_name	danceability
1	2DGSytzXs9oqYUy5tqxfSB	Twinkle Twinkle Little Star	Tommy Twinkle	0.550
15	1K7JgK79uuxSOpbB415lpM	Daydreaming	Sleepy Shepherd	0.413
22	2xsQgizMzkkLFa90gv7Ftr	Changes	Sleepy Shepherd	0.427
32	73VoUyIWl4oqU0d4O32GSV	Eternity	Sleepy Shepherd	0.471
43	5o8JmM9lAsh70Xg1fC0Ydn	Timeless	Sleepy Shepherd	0.429
...	...	...	...	...
1199	2PYSA0e7vCpXqN2iPHTXTN	San Francisco	THE DRIVER ERA	0.656
1211	3SdTKo2uVsxFblQjpScoHy	Stand By Me	Ben E. King	0.650
1213	1JSTJqkT5qHq8MDJnJbRE1	Every Breath You Take	The Police	0.820
1233	0oUBuOO4g9P4lREqfqR5nq	A Man Without Love	Engelbert Humperdinck	0.538
1235	3usbnvDftOhY09cRNar8Zg	FRIENDS	LAUNDRY DAY	0.522

212 rows × 5 columns

Gambar 4.53 Hasil Data *Cluster* 3

Gambar diatas merupakan data dari hasil *cluster* yang ada pada *cluster* 3, *cluster* ini berhasil mendapatkan data dengan jumlah 212 data, data diatas akan dilakukan analisis lebih lanjut agar dapat mengetahui *cluster* diatas menghasilkan suasana hati apa.

Setelah mendapatkan hasil *cluster*, selanjutnya kita akan menambahkan hasil *cluster* tersebut kedalam *dataset* untuk mencari dari ke 4 *cluster* tersebut manakah yang menghasilkan suasana hati *sad*, berikut adalah kodenya bisa dilihat pada gambar 4.54

```
cluster_labels = kmeans.labels_
df['cluster'] = cluster_labels
```

Gambar 3.54 Program Untuk Menambahkan Hasil *Cluster* Ke *Dataset*

Setelah menjalankan kode diatas maka hasil *cluster* akan ditambahkan kedalam *dataset* bernama *cluster* untuk diseleksi *cluster* mana yang menghasilkan jenis suasana hati

sad, seleksi ini dilakukan secara manual, jadi musik dari *cluster* 0/1/2/3/ akan dicek satu persatu, dari *cluster* tersebut *cluster* berapakah yang menghasilkan suasana hati *sad*. Setelah mengetahui *cluster* mana yang menghasilkan suasana hati *sad*, maka selanjutnya akan ditambahkan *cluster* tersebut ke dalam *dataset*, kode programnya bisa dilihat pada gambar 4.55.

```
df['sad'] = df['cluster'].apply(lambda x: 1 if x == 1 else 0)
```

Gambar 4.55 Program Menambah Hasil Seleksi Suasana Hati *Sad*

Setelah menjalankan kode diatas maka sistem akan menambahkan kolom baru pada *dataset* dengan nama *sad*, jadi kode diatas juga menandai lagu-lagu yang termasuk dalam *cluster* 0 sebagai jenis suasana hati *sad*, jadi pada kolom *sad* nilainya akan menjadi 1 jika dia merupakan *cluster* 0, jika bukan maka akan diberi nilai 0.

3. Pengujian *Silhouette Coefficient*

Setelah mendapatkan hasil dari *clustering* musik selanjutnya kita akan menguji hasil *cluster* dari kedua fitur *acousticness* dan *valence* dengan *n cluster* nya 3 menghasilkan skor *Silhouette* berapa, berikut adalah kode programnya pada gambar 4.56.

```
X = df[['acousticness', 'valence']]
labels = kmeans.labels_

silhouette_avg = silhouette_score(X, labels)
print("Silhouette Coefficient: {:.3f}".format(silhouette_avg))
```

Gambar 4.56 Kode Program *Silhouette Coefficient*

Pada kode yang pertama terdapat *library* yang mengimpor fungsi *Silhouette_score* dari *library sklearn*, selanjutnya *df_clustered* diatas merupakan *DataFrame* atau *dataset*, selanjutnya pada selanjutnya kode: *Silhouette_avg = Silhouette_score(X, labels)*: untuk menghitung skor *Silhouette* rata-rata untuk semua sampel. Fungsi ini membutuhkan dataasli 'X' dan label pengelompokan '*labels*' sebagai

input, dan selanjutnya pada kode dibawahnya untuk mencetak hasil dari skor *Silhouette* menjadi 3 angka. Jika program diatas sudah di *running* maka hasil skor *Silhouette* nya bisa dilihat pada gambar 4.57.

Silhouette Coefficient: 0.431

Gambar 4.57 Hasil Skor

Hasil dari skor *Silhouette* dari *cluster* pada fitur "*acousticness*" dan "*valence*" dengan n *cluster* nya 4 mendapat hasil skor *Silhouette* yaitu 0.431, berdasarkan interpretasi nilai *Silhouette Coefficient* pada tabel 3.2 skor *Silhouette* diatas mendapat interpretasi "struktur yang dihasilkan lemah".

Dan berikut adalah sampel lagu yang berhasil didapatkan dari suasana hati *Sad*, sampel diatas telah berasal dari hasil *clustering* dan diambil dalam *dataset* yang telah disimpan. Untuk lebih jelasnya bisa dilihat pada tabel 4.3.

Tabel 4.3 Daftar Lagu Pada Suasana Hati *Sad*

Suasana Hati	Kategori Umur	Judul Lagu
<i>Sad</i>	Bayi	<i>Golden Sunrise</i>
	Bayi	<i>Lavender's Blue</i>
	Lansia	<i>The Night We Met</i>
	Lansia	<i>The Flame</i>
	Anak-anak	Dua Balerina
	Anak-anak	Jangan Takut Gelap
	Remaja	Tak Ingin Pisah Lagi
	Remaja	ONLY
	Dewasa	Dunia Tipu-Tipu
	Dewasa	untuk hati yang terluka.

4.4.4. Mencari Suasana Hati *Happy*

1. Menentukan Jumlah *Cluster*

Sama seperti pada tahap sebelumnya, proses pertama kita akan menentukan jumlah *cluster* terlebih dahulu dengan menggunakan *Elbow Method*, metode ini banyak digunakan

pada penelitian-penelitian terdahulu, karena mudah untuk di implementasikan pada program dan mudah dibaca, berikut adalah gambar 4.58.

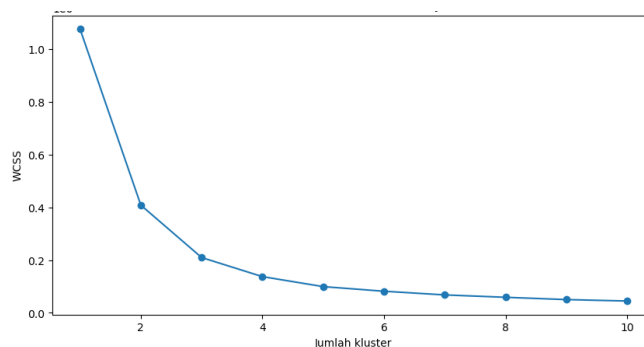
```
X = df[['loudness', 'tempo']]

wcss = []
for i in range(1, 11): # Mencoba k dari 1 sampai 10
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10, random_state=0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)

plt.figure(figsize=(10, 5))
plt.plot(range(1, 11), wcss, marker='o')
plt.title('Metode Elbow untuk Menentukan Jumlah k')
plt.xlabel('Jumlah kluster')
plt.ylabel('WCSS')
plt.show()
```

Gambar 4.58 Kode Program *Elbow Method*

Program diatas menggunakan fitur “*loudness*” dan “*tempo*” untuk mencari jenis suasana hati *happy*, karena kedua fitur ini biasa menghasilkan *music* yang ceria dan gembira, jika program diatas sudah dirunnig maka hasilnya seperti pada gambar 4.59.



Gambar 4.59 Hasil *Elbow Method*

Gambar dari hasil *Elbow Method* diatas menjelaskan bahwa penurunan terjadi pada *cluster* 3 maka jumlah *cluster* tersebut lah yang direkomendasikan oleh *Elbow Method* untuk dilakukan *clustering*.

2. *Clustering*

Pada tahap ini kita akan melakukan *clustering* untuk mencari suasana hati *happy*, dikarenakan kita sudah mengetahui hasil dari *Elbow Method* untuk menentukan *n cluster*, maka selanjutnya kita akan melakukan *clustering* pada fitur yang sama pada *Elbow Method*

yaitu “loudness” dan “tempo” untuk mencari jenis suasana hati *happy*, berikut adalah kode program untuk melakukan *clustering*, bisa dilihat pada gambar 4.60.

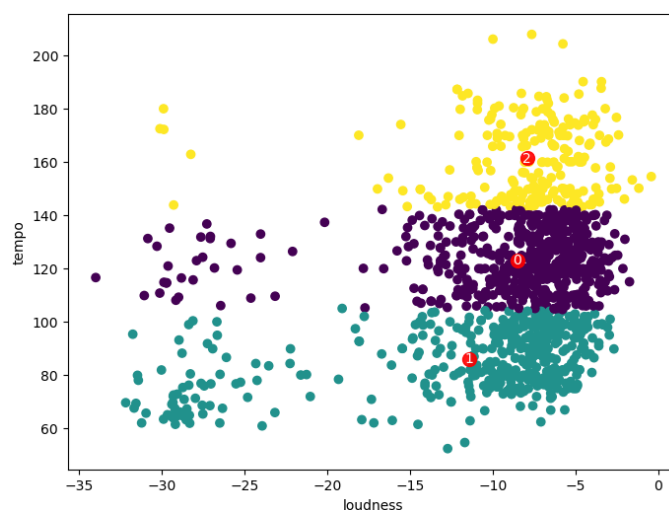
```
data = df
kmeans = KMeans(n_clusters=3, random_state=0)
kmeans.fit(df[['loudness', 'tempo']])
df['cluster'] = kmeans.labels_
cluster_data = {}

for k in range(kmeans.n_clusters):
    cluster_data[f'clus{k}'] = df[df['cluster'] == k].drop('cluster', axis=1)
centers = kmeans.cluster_centers_
print("Centroid dari setiap cluster:")
for i, center in enumerate(centers):
    print(f"Cluster {i}: {center}")

plt.figure(figsize=(8, 6))
plt.scatter(df['loudness'], df['tempo'], c=kmeans.labels_, cmap='viridis')
for i, center in enumerate(centers):
    plt.scatter(center[0], center[1], c='red', s=100, alpha=0.9)
    plt.text(center[0], center[1], str(i), color='white', ha='center', va='center')
plt.xlabel('loudness')
plt.ylabel('tempo')
plt.title('Visualisasi Clustering Lagu')
plt.show()
```

Gambar 4.60 Kode Program *Clustering*

Pada kode diatas df merupakan *DataFrame* atau *dataset*, dan pada kode diatas n *cluster* nya saya pakai 3, dan pada kode selanjutnya kita akan memilih fitur apa yang akan digunakan untuk *clustering*, setekah memilih fitur maka selanjutnya kita saya menambahkan kode untuk memvisualisaikan hasil *cluster* nya menggunakan *plot*, dan saya juga menambahkan *code* agar terdapat visual yang mencari *centroid* dari setiap *cluster*, hasil dari kode diatas akan seperti pada gambar 4.61.



Gambar 4.61 Hasil *Clustering*

Pada sumbu horizontal (x), kita memiliki '*loudness*' yang nilainya berkisar dari kira-kira -35 hingga 0 decibel (dB). Nilai '*loudness*' ini mengukur seberapa keras sebuah lagu terdengar. Nilai yang lebih tinggi (mendekati 0 dB) menunjukkan bahwa lagu tersebut terdengar lebih keras, sedangkan nilai yang lebih rendah menunjukkan bahwa lagu tersebut terdengar lebih pelan. Pada sumbu vertikal (y), kita memiliki '*tempo*', yang diukur dalam ketukan per menit (BPM).

Seperti sebelumnya, nilai yang lebih tinggi menunjukkan lagu yang lebih cepat, sementara nilai yang lebih rendah menunjukkan lagu yang lebih lambat. Titik-titik yang diwarnai secara berbeda merepresentasikan kluster atau grup lagu yang memiliki karakteristik '*loudness*' dan '*tempo*' yang serupa atau mirip. Setiap kluster diwakili oleh satu warna saja. Titik-titik yang berwarna merah dengan angka di atasnya adalah *centroid* dari masing-masing kluster, yang menandakan nilai rata-rata dari '*loudness*' dan '*tempo*' untuk lagu-lagu yang ada pada kluster tersebut. Angka-angka pada *centroid* tersebut menunjukkan nomor identifikasi kluster tersebut. Berikut penjelasan yang lebih spesifik untuk setiap *cluster*.

a. *Cluster 0*(ungu)

Cluster ini memiliki lagu yang memiliki nilai '*loudness*' yang cenderung dominan antara -17 hingga -3 dB, yang menunjukkan volume yang moderat hingga cukup keras. Dengan '*tempo*' antara 110 hingga 135 bpm, lagu-lagu di *cluster* ini mungkin menawarkan suasana yang bersemangat tanpa menjadi terlalu agresif atau terlalu tenang. Ini adalah '*tempo*' yang sedang menandakan bahwa tidak semua musik pada *cluster* ini mengandung musik dengan suasana hati *happy*. Berikut adalah gambar 4.62 data dari *cluster 0*.

```
cluster_data['clus0']
```

✓ 0.0s

	track_id	track_name	artist_name
1	2DGSytzXs9oqYUy5tqxfSB	Twinkle Twinkle Little Star	Tommy Twinkle
3	4tbnZK9nKvuyCrqyu8o0eG	Hush Little Baby	Sleepy Shepherd
4	5YMmjaZEzUoI8EFZm4q2fB	Moonlight Walk	Orion The Owl
6	4GphHeTcJqIEbIY6jRMyaG	Lavender's Blue	Sleepy Shepherd
7	5NI8DkhAIQjkci2AiJI90h	Wake Up	Bluedon
...	...	...	...
1217	5TRPicyLGBAF2LGBFbHGvO	Flashing Lights	Kanye West
1218	51EC3I1nQXpec4gDk0mQyP	90210 (feat. Kacy Hill)	Travis Scott
1220	5hM5arv9KDbCHS0k9uqwjr	Borderline	Tame Impala
1230	4gZDCOFuLCxCRWNhSEcOtI	These Days	Wallows
1235	3usbnvDFtOhY09cRNar8Zg	FRIENDS	LAUNDRY DAY

413 rows × 17 columns

Gambar 4.62 Hasil Data *Cluster 0*

Gambar diatas merupakan data dari hasil *cluster* yang berada pada *cluster 0*, dan *cluster* ini berhasil mendapatkan data dengan jumlah total 413 data, data diatas akan - dilakukan analisis lebih lanjut agar dapat mengetahui *cluster* diatas menghasilkan suasana hati apa saja.

b. *Cluster 1*(biru)

Cluster ini memiliki lagu yang memiliki nilai '*loudness*' yang cenderung dominan antara -33 hingga -3 dB dan '*tempo*' dari 60 hingga 100 bpm. Volume yang lebih rendah pada bagian dari *cluster* ini mungkin mengarah pada lagu yang lebih lembut dan introspektif, sementara '*tempo*' yang lebih lambat mengindikasikan ritme yang lebih santai dan tenang. Meskipun ini mungkin tidak berhubungan dengan suasana hati '*happy*'. Berikut adalah gambar 4.63 data dari *cluster 1*.

cluster_data['clus1']			
✓ 0.0s			
	track_id	track_name	artist_name
0	0tCqX7pASaZeHPC4FhAwe8	Brahms Lullaby	Sleepy Shepherd
2	0KOOhxPmW5pfMXjgWzRf0j	Goodnight Little Dreamer	Little Dreamer
11	6lpb4EVYeX94RQb2mo3AUr	My Bonnie Lies Over The Ocean	Sleepy Shepherd
29	7nTghMbNP3rDEzStg5NGlz	Into The Clouds	Sleepy Shepherd
44	4RyXEW4Pfql6Zdh1k6Ws	At Dawn	Sleepy Shepherd
...	...	...	...
1229	7p4vHnYXkxlzvfPJVpcTr	Feels Like Summer	Childish Gambino
1231	6kE3BJhZYLIrGLQY2hSzaM	Younger	The Hails
1236	2h4BV7jx9Ltz9Nx2zgNcW	Now or Never (feat. Charlie Gillespie, Owen Pa...	Julie and the Phantoms Cast
1237	678tfxINbhBmCDR2L2l8qd	MONTERO (Call Me By Your Name)	Lil Nas X
1239	4p5Phj0pcXWW7B81IUGowi	masquerade x miss the rage	acti
277 rows × 4 columns			

Gambar 4.63 Hasil Data *Cluster 1*

Gambar diatas merupakan data dari hasil *cluster* yang berada pada *cluster 1*, dan *cluster* ini berhasil mendapatkan data dengan jumlah 277 data, data diatas akan dilakukan analisis lebih lanjut agar dapat mengetahui *cluster* diatas menghasilkan suasana hati apa saja.

c. *Cluster 2*(kuning)

Dengan '*loudness*' yang cenderung dominan dari -15 hingga -2 dB dan '*tempo*' yang sangat cepat dari 140 hingga 200 bpm, *cluster* ini cenderung merepresentasikan lagu-lagu yang sangat energik dan meriah. Volume yang tinggi bersama dengan '*tempo*' yang cepat biasanya dikaitkan dengan kegembiraan dan semangat yang intens, cocok untuk suasana yang sangat '*happy*', seperti dalam konteks pesta atau perayaan. Dan berdasarkan validasi secara manual *cluster* ini memiliki banyak musik yang berhubungan dengan suasana hati *happy* dan *cluster* ini juga di tambahkan ke dalam *dataset*. Berikut adalah gambar 4.64 data dari *cluster 2*.

`cluster_data['clus2']`
✓ 0.0s

	track_id	track_name	artist_name	danceability
5	1EvERSC5XOhcvGdhSHjil	Golden Sunrise	Chillboy	0.561
10	7DOQ5HUSJeHYLmOhtGKz3y	Are You Sleeping	Sleepy Shepherd	0.518
14	01b3JCDs8aA4D6Rglx8yhX	Trust Me	Sleepy Shepherd	0.536
15	1K7JgK79uuxSOpbB415IpM	Daydreaming	Sleepy Shepherd	0.413
17	1VvLz9pvdKxhKP3jJwEilQ	Night Sky	Sleepy Shepherd	0.416
...	...	...	...	...
1228	5ZkAx8zjLiSs1nMmBwJoZS	When Can I See You Again? - From "Wreck-It Ral...	Owl City	0.638
1232	796gbCqHEHWvEqtYOBXibL	Fake a Frown	The Hails	0.758
1233	0oUBuOO4g9P4IREqfqR5nq	A Man Without Love	Engelbert Humperdinck	0.538
1234	7ETjvgUdK7rBIGfvhiWQj	Stay	The Hails	0.665
1238	7CBXta3Je26MMHOFGERplS	Miss The Rage - Remix	Toxic Tripz	0.514

550 rows × 5 columns

Gambar 4.64 Hasil Data *Cluster 2*

Gambar diatas merupakan data dari hasil *cluster* yang ada pada *cluster 2*, *cluster* ini berhasil mendapatkan data dengan jumlah 550 data, data diatas akan dilakukan analisis lebih lanjut agar dapat mengetahui *cluster* diatas menghasilkan suasana hati apa. Setelah mendapatkan hasil *cluster*, selanjutnya kita akan menambahkan hasil *cluster* tersebut kedalam *dataset* untuk mencari dari ke 3 *cluster* tersebut manakah yang bisa menghasilkan suasana hati *happy*, berikut adalah kode program yang bisa dilihat pada gambar 4.65.

```
cluster_labels = kmeans.labels_
df['cluster'] = cluster_labels
```

Gambar 4.65 Kode Program Untuk Menambah Hasil *Cluster* Ke *Dataset*

Setelah menjalankan kode diatas maka hasil *cluster* akan ditambahkan kedalam *dataset* bernama *cluster* untuk diseleksi *cluster mana* yang menghasilkan jenis suasana hati *happy*, seleksi ini dilakukan secara manual, jadi musik dari *cluster 0/1/2* akan dicek satu persatu, dari ke dua *cluster* tersebut *cluster* berapakah yang menghasilkan suasana hati

happy. Setelah mengetahui *cluster* berapa yang menghasilkan suasana hati *happy* selanjutnya kita akan menambahkan *cluster* tersebut ke dalam *dataset*, kodenya bisa dilihat pada gambar 4.66.

```
df['happy'] = df['cluster'].apply(lambda x: 1 if x == 2 else 0)
```

Gambar 4.66 Kode Program Menambah Hasil Seleksi Suasana Hati *Happy*

Setelah menjalankan kode diatas maka sistem akan menambahkan kolom baru pada *dataset* dengan nama *happy*, jadi kode diatas juga menandai lagu-lagu yang termasuk dalam *cluster 2* sebagai jenis suasana hati *happy*, jadi pada kolom *happy* nilainya akan menjadi 1 jika dia merupakan *cluster 2*, jika bukan maka akan diberi nilai 0.

3. Pengujian *Silhouette Coefficient*

Setelah mendapatkan hasil dari *clustering* musik selanjutnya kita akan menguji hasil *cluster* dari kedua fitur *energy* dan *tempo* dengan n *cluster* nya 3 menghasilkan skor *Silhouette* berapa, berikut adalah kode programnya pada gambar 4.67.

```
X = df[['loudness', 'tempo']]
labels = kmeans.labels_

silhouette_avg = silhouette_score(X, labels)
print("Silhouette Coefficient: {:.3f}".format(silhouette_avg))
```

Gambar 4.67 Kode Program *Silhouette*

Pada kode yang pertama terdapat *library* yang mengimpor fungsi *Silhouette_score* dari *library sklearn*, selanjutnya *df_clustered* diatas merupakan *DataFrame* atau *dataset*, selanjutnya pada selanjutnya kode: *Silhouette_avg = Silhouette_score(X, labels)*: untuk menghitung skor *Silhouette* rata-rata untuk semua sampel. Fungsi ini membutuhkan data asli 'X' dan label pengelompokan '*labels*' sebagai *input*, selanjutnya pada kode dibawahnya untuk mencetak hasil dari skor *Silhouette*

menjadi 3 angka. Jika program diatas sudah di *running* maka hasil skor *Silhouette* nya bisa dilihat pada gambar 4.68.

Silhouette Coefficient: 0.514

Gambar 4.68 Hasil Skor *Silhouette*

Hasil dari skor *Silhouette* dari hasil *cluster* pada fitur "*loudness*" dan "*tempo*" dengan n *cluster* nya 3 mendapat hasil skor *Silhouette* yaitu 0.514, berdasarkan interpretasi nilai *Silhouette Coefficient* pada tabel 3.2 skor *Silhouette* diatas mendapat interpretasi "struktur yang dihasilkan baik".

Dan berikut adalah sampel lagu yang berhasil didapatkan dari suasana hati *Happy*, sampel diatas telah berasal dari hasil *clustering* dan diambil dalam *dataset* yang telah disimpan. Untuk lebih jelasnya bisa dilihat pada tabel 4.4.

Tabel 4.4 Daftar Lagu Pada Suasana Hati *Happy*

Suasana Hati	Kategori Umur	Judul Lagu
<i>Happy</i>	Bayi	<i>Ich flieg in meinem Flugzeug</i>
	Bayi	<i>Liebe, liebe Sonne</i>
	Lansia	<i>Dancing In the Dark</i>
	Lansia	<i>Change in Mood</i>
	Anak-anak	Makan Jangan Asal Makan
	Anak-anak	Bis Sekolah
	Remaja	<i>honey</i>
	Remaja	Tergila-Gila
	Dewasa	Siren
	Dewasa	<i>No Friends</i>

4.4.5. Mencari Suasana Hati *Angry*

1. Menentukan Jumlah *Cluster*

Sama seperti pada tahap sebelumnya, proses pertama kita akan menentukan jumlah *cluster* terlebih dahulu dengan menggunakan *Elbow Method*, metode ini banyak digunakan pada penelitian-penelitian terdahulu, karena mudah untuk di implementasikan pada program

dan mudah dibaca, berikut adalah gambar 4.69.

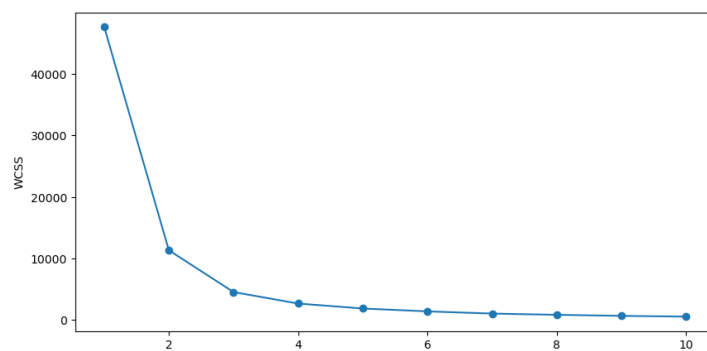
```
X = df[['energy', 'loudness']]

wcss = []
for i in range(1, 11): # Mencoba k dari 1 sampai 10
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10, random_state=0)
    kmeans.fit(X)
    wcss.append(kmeans.inertia_)

plt.figure(figsize=(10, 5))
plt.plot(range(1, 11), wcss, marker='o')
plt.title('Metode Elbow untuk Menentukan Jumlah k')
plt.xlabel('Jumlah kluster')
plt.ylabel('WCSS')
plt.show()
```

Gambar 4.69 Kode Program *Elbow Method*

Program diatas menggunakan fitur “energy” dan “loudness” untuk mencari jenis suasana hati *angry*, jika program diatas sudah dirunnig maka hasilnya akan seperti pada gambar 4.70.



Gambar 4.70 Hasil *Elbow Method*

Gambar dari hasil *Elbow Method* diatas menjelaskan bahwa penurunan terjadi pada *cluster 3* maka jumlah *cluster* tersebut lah yang direkomendasikan oleh *Elbow Method* untuk dilakukan *clustering*. Tapi disini saya menggunakan 5 *cluster*.

2. Clustering

Pada tahap ini kita akan melakukan *clustering* untuk mencari suasana hati *angry*, dikarenakan kita sudah mengetahui hasil dari *Elbow Method* untuk menentukan n *cluster*, maka selanjutnya kita akan melakukan *clustering* pada fitur yang sama pada *Elbow Method* yaitu “energy” dan “loudness” untuk mencari jenis suasana hati *happy*, berikut adalah kode

program untuk melakukan *clustering*, bisa dilihat pada gambar 4.71.

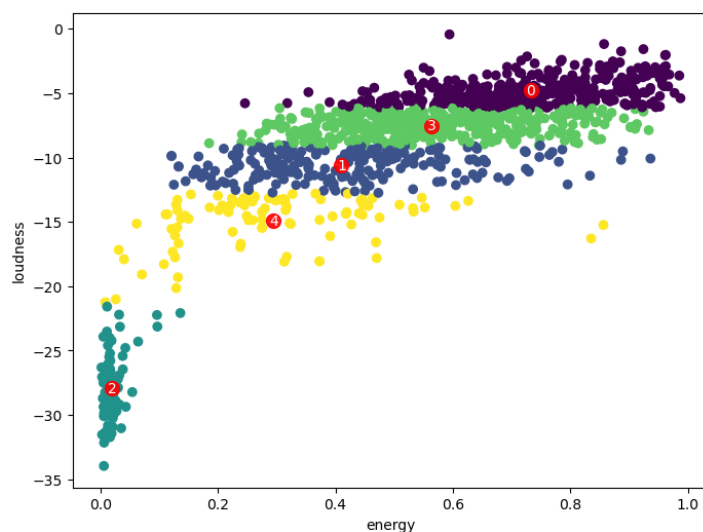
```
data = df
kmeans = KMeans(n_clusters=5, random_state=0)
kmeans.fit(df[['energy', 'loudness']])
df['cluster'] = kmeans.labels_
cluster_data = {}

for k in range(kmeans.n_clusters):
    cluster_data[f'clus{k}'] = df[df['cluster'] == k].drop('cluster', axis=1)
centers = kmeans.cluster_centers_
print("Centroid dari setiap cluster:")
for i, center in enumerate(centers):
    print(f"Cluster {i}: {center}")

plt.figure(figsize=(8, 6))
plt.scatter(df['energy'], df['loudness'], c=kmeans.labels_, cmap='viridis')
for i, center in enumerate(centers):
    plt.scatter(center[0], center[1], c='red', s=100, alpha=0.9) # Pusat cluster
    plt.text(center[0], center[1], str(i), color='white', ha='center', va='center')
plt.xlabel('energy')
plt.ylabel('loudness')
plt.title('Visualisasi Clustering Lagu')
plt.show()
```

Gambar 4.71 Kode Program *Clustering*

Pada kode diatas df merupakan *DataFrame* atau *dataset*, dan pada kode diatas n *cluster* nya saya pakai 5, dan pada kode selanjutnya kita akan memilih fitur apa yang akan digunakan untuk *clustering*, setekah memilih fitur maka selanjutnya kita saya menambahkan kode untuk memvisualisaikan hasil *cluster* nya menggunakan *plot*, dan saya juga menambahkan code agar terdapat visual yang mencari *centroid* dari setiap *cluster*, hasil dari kode diatas akan seperti pada gambar 4.72.



Gambar 4.72 Hasil *Clustering*

Pada sumbu horizontal (x), 'energy' berkisar antara 0 dan 1, di mana nilai yang lebih tinggi menunjukkan lagu dengan energi yang lebih intens atau dinamis, yang mungkin mencerminkan volume yang lebih tinggi, kehadiran instrumen yang lebih kuat, atau *tempo* yang lebih cepat. Pada sumbu vertikal (y), kita memiliki 'loudness' yang diukur dalam desibel (dB). Nilai yang lebih tinggi (mendekati 0 dB) menunjukkan volume lebih keras, sementara nilai yang lebih rendah (misalnya, -30 dB) menunjukkan volume yang lebih lembut.

Warna yang berbeda pada grafik menunjukkan kluster yang berbeda, dengan setiap kluster terdiri dari lagu-lagu yang memiliki karakteristik 'energy' dan 'loudness' yang serupa. Titik-titik berwarna merah dengan angka adalah *centroid* dari kluster tersebut, mewakili nilai tengah dari fitur-fitur lagu dalam kluster itu. Angka-angka pada *centroid* tersebut menandakan nomor kluster. Berikut penjelasan yang lebih spesifik untuk setiap *cluster*.

a. *Cluster 0* (warna ungu)

cluster ini berhasil mendapatkan data dengan jumlah 437 data, data diatas akan - dilakukan analisis lebih lanjut agar dapat mengetahui *cluster* diatas menghasilkan suasana hati apa. Gambar 4.73 data *cluster 0*.

cluster_data['clus0']

✓ 0.2s

	track_id	track_name	artist_name
99	2eOGhMShEgQtmhCYFX3SNL	Guten Tag, Herr Nasenmann	Simone Sommerland
101	0vznCldlSS700r0nUkhJlk	Häschen in der Grube	Simone Sommerland
102	37VkvYDhM7gd2LsNZTrt8P	Hoppe, hoppe, Reiter	Simone Sommerland
104	2igsIhO50jvovAVXmSx1P3	Ich flieg in meinem Flugzeug	Simone Sommerland
105	3JzrNxoNVI2MI2NaCpfsRu	Der ist ins Wasser gefallen	Simone Sommerland
...	...	...	...
1231	6kE3BJhZYLIfGLQY2hSzaM	Younger	The Hails
1232	796gbCqHEHWvEqYOBXibL	Fake a Frown	The Hails
1235	3usbnvDFtOhY09cRNar8Zg	FRIENDS	LAUNDRY DAY
1237	67BtfklNbh8mCDR2L2l8qd	MONTERO (Call Me By Your Name)	Lil Nas X
1238	7CBXta3Je26MMHOFGERpIS	Miss The Rage - Remix	Toxic Tripz

437 rows × 17 columns

Gambar 4.73 Hasil Data *Cluster 0*

Gambar diatas merupakan data dari hasil *cluster* yang ada pada *cluster* 0, *cluster* ini ditempati oleh lagu-lagu dengan energi dan volume yang sangat tinggi karena *cluster* ini didominasi pada nilai *energy* yang cenderung dominan dari 0.5 – 1.0, dan nilai *loudness*-nya dominan dari -5 – 0 dB, *cluster* ini memiliki nilai yang paling tinggi diatara *cluster* lain, dan menggambarkan lagu-lagu yang sangat dinamis dan keras, yang mencerminkan emosi yang kuat seperti marah. Berikut adalah gambar 4.73 data dari *cluster* 0.

b. *Cluster* 1(warna biru)

Cluster ini berhasil mendapatkan data dengan jumlah 96 data, data diatas akan - dilakukan analisis lebih lanjut agar dapat mengetahui *cluster* diatas menghasilkan suasana hati apa. Gambar 4. 74 data cluster 1.

`cluster_data['clus1']`

✓ 0.2s

	track_id	track_name	artist_name
0	0tCqX7pASaZeHPC4FhAwe8	Brahms Lullaby	Sleepy Shepherd
1	2DGSytzXs9oqYUy5tqxfSB	Twinkle Twinkle Little Star	Tommy Twinkle
2	0KOOhxPmW5pfMXjgWzRf0j	Goodnight Little Dreamer	Little Dreamer
3	4tbnZK9nKvuyCrqyu8o0eG	Hush Little Baby	Sleepy Shepherd
4	5YMmjaZEzUol8EFZm4q2fB	Moonlight Walk	Orion The Owl
...	...	...	...
94	2G2uktp1S7migRoyW98Rrs	Breathing In	Orion The Owl
95	6aqW3HHGCKObfyVmYb5F03	Moonlit Slumber	Orion The Owl
96	1UisSEfoqlqix7gLRaIBRc	Feeling Free	Orion The Owl
97	19GGFxmHCcZJGIC0c8PfLb	Follow Me	Orion The Owl
98	6Ld1aiIZk8lqOqTPltfwOV	Tied Angel	Orion The Owl

96 rows × 4 columns

Gambar 4.74 Hasil Data *Cluster* 1

Gambar diatas merupakan data dari hasil *cluster* yang ada pada *cluster* 1, *Cluster* ini berada pada *energy* menengah-tinggi dan *loudness* yang sedikit lebih rendah dari

cluster 0, nilai *energy* cenderung dominan dari 0.2 – 0.7, namun ada beberapa titik yang memiliki nilai yang lebih tinggi, dan nilai *loudness* cenderung dominan pada -9 - -5 dB, lagu-lagu dalam *cluster* ini mungkin memiliki intensitas yang cukup untuk menyampaikan emosi marah tetapi dengan tingkat kekerasan yang lebih moderat.

c. *Cluster* 2(warna hijau toska)

Cluster mengandung lagu-lagu dengan *energy* rendah tetapi memiliki nilai *loudness* yang rendah, nilai pada *cluster* ini berkisar antara *energy* yang dominan dari 0.0 – 0.2, dan nilai dari *loudness* yaitu -21 - -34, lagu lagu yang ada pada *cluster* ini mewakili suasana hati yang damai, atau relaks, lagu lagu disini merupakan lagu yang paling lembut diantara *cluster* lain, dengan hampir tidak ada dinamika dan volume yang sangat rendah, dan *cluster* ini tidak merepresentasikan suasana hati *angry*, karena memiliki nilai yang paling rendah diantara *cluster* yang lain. Berikut adalah gambar 4.75 data dari *cluster* 2.

	track_id	track_name	artist_name	danceability
7	5NI8DkhAlQjki2AiJl90h	Wake Up	Bluedon	0.618
69	4AM5d5dFMW8tmpCyfCrFaR	Like a Dream	Bluedon	0.728
72	6pwayzRmkHmk5LovE0h82m	Coastline	Chillboy	0.537
147	1AJ9Yp2PbNFbvWSIScW3M	Der Mond ist aufgegangen	Simone Sommerland	0.657
212	659uoDxjIVHj61UmPyNG0y	Are You Old Enough	Dragon	0.701
...	...	...	...	...
1140	5IMtdHjJ1OtkxbGe4zfUxQ	Escape (The Pina Colada Song)	Rupert Holmes	0.836
1142	2Xb6wJYGj0QXwURw5WWvi5	Right Down the Line	Gerry Rafferty	0.783
1152	1BkY0N8ChFk2mdLbAUu8ZK	Pass The Dutchie	Musical Youth	0.805
1208	2374M0fQpWi3dLnB54qaLX	Africa	TOTO	0.671
1239	4p5Phj0pcXWW7B81IUGowi	masquerade x miss the rage	acti	0.496

99 rows × 5 columns

Gambar 4.75 Hasil Data *Cluster* 2

Gambar diatas merupakan data dari hasil *cluster* yang ada pada *cluster* 2, *cluster* ini berhasil mendapatkan data dengan jumlah 99 data, data diatas akan dilakukan analisis lebih

lanjut agar dapat mengetahui *cluster* diatas menghasilkan suasana hati apa.

d. *Cluster 3*(warna hijau muda)

Cluster ini mengandung lagu-lagu dengan tingkat energi dan volume yang lebih seimbang, namun masih cenderung ke tinggi. Titik-titik yang berada pada energi cenderung dominan pada angka 0.3 hingga 0.8 dan *loudness* dari -5 hingga -10 dB yang menampilkan kombinasi antara intensitas yang cukup tinggi dengan volume yang kuat, menciptakan suasana hati yang penuh tekanan yang bisa berkaitan dengan kemarahan. Berikut adalah gambar 4.76 data dari *cluster 3*.

cluster_data['clus3']

✓ 0.4s

	track_id	track_name	artist_name
109	5FEoxxUi8fR3hgEfKf9agl	Alle meine Entchen	Simone Sommerland
113	3KJNT8jnmJwVTer0WFOuxe	Wo ist der Daumen?	Simone Sommerland
120	2hYVdRZi7ntHnsloWTo8OR	Schmetterling, du kleines Ding	Simone Sommerland
133	3FI3doKfq5fwOudzmEhZrf	Töff, töff, töff, die Eisenbahn	Simone Sommerland
134	4X6JCvxdQe9Yiw3FoMoaun	Das Quietscheentchen	Simone Sommerland
...	...	...	...
1224	5FSt6K70gANsecaAtXFVbR	When Will I See You Again	Shakka
1225	42Lq6GBO5KUSNeSgJsOoke	Panama	Van Halen
1226	4mGJcOKQhmISIWgQ3xCiLB	Til I Forget About You	Big Time Rush
1234	7ETijvgUdK7rBiGfvhiWQj	Stay	The Hails
1236	2h48V7jx9Ltz9Nx2zgNcW	Now or Never (feat. Charlie Gillespie, Owen Pa...	Julie and the Phantoms Cast

370 rows × 17 columns

Gambar 4.76 Hasil Data *Cluster 3*

Gambar diatas merupakan data dari hasil *cluster* yang ada pada *cluster 3*, *cluster* ini berhasil mendapatkan data dengan jumlah 370 data, data diatas akan dilakukan analisis lebih lanjut agar dapat mengetahui *cluster* diatas menghasilkan suasana hati apa.

e. *Cluster 4*(warna kuning)

Kluster dengan warna kuning (terletak pada '*energy*' rendah yang dominan pada 0.1

hingga 0.5 dan 'loudness' sangat rendah dari -30 hingga -25 dB) bisa mewakili suasana hati yang 'tenang' atau 'relaks', namun ada 2 titik pada *cluster* ini yang memiliki nilai yang energy yang cukup tinggi yang mungkin merepresentasikan suasana hati yang lebih *happy/energetic*, tetapi *cluster* ini didominasi pada nilai yang cukup rendah dan sesuai dengan karakteristik lagu yang memiliki energi rendah dan volume yang sangat lembut. Berikut adalah gambar 4.77 data dari *cluster* 4.

cluster_data['clus4']

✓ 0.1s

	track_id	track_name	artist_name	danceability
100	68koPzy3yFLovUysQqF4rG	Das ist grade, das ist schief	Simone Sommerland	0.825
103	4luZiIP3swmJu03JX8bJG	Der Apfelbaum	Simone Sommerland	0.794
116	3h4f3gh1Y2M4tEbRJAsgH6	Große Uhren machen tick tack	Simone Sommerland	0.879
140	38Iz2x1F8QzHRejGVoeWLt	Leise, Peterle leise	Simone Sommerland	0.645
141	6lgpK4RmLUvtEVux0z4BLc	Wenn du schläfst	Simone Sommerland	0.725
...	...	...	...	...
1211	3SdTKo2uVsxFblQjpScoHy	Stand By Me	Ben E. King	0.650
1212	1JLn8RhQzHz3qDqsChcmBI	I Want to Know What Love Is - 1999 Remaster	Foreigner	0.477
1213	1JSTJqkT5qHq8MDJnJbRE1	Every Breath You Take	The Police	0.820
1222	5mCPDVbB16L4XQwDdbRUpz	Passionfruit	Drake	0.809
1233	0oUBuOO4g9P4IREqfqR5nq	A Man Without Love	Engelbert Humperdinck	0.538

238 rows × 5 columns

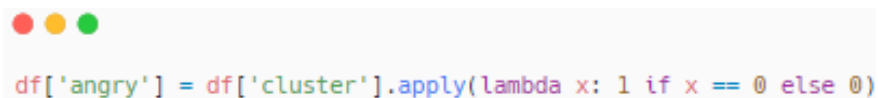
Gambar 4.77 Hasil Data *Cluster* 4

Gambar diatas merupakan data dari hasil *cluster* yang ada pada *cluster* 4, *cluster* ini berhasil mendapatkan data dengan jumlah 238 data, data diatas akan dilakukan analisis lebih lanjut agar dapat mengetahui *cluster* diatas menghasilkan suasana hati apa. Setelah mendapatkan hasil *cluster*, selanjutnya kita akan menambahkan hasil *cluster* tersebut kedalam *dataset* untuk mencari dari ke 5 *cluster* itu manakah yang dapat menghasilkan suasana hati *angry*, berikut adalah kodenya bisa dilihat pada gambar 4.78.

```
cluster_labels = kmeans.labels_
df['cluster'] = cluster_labels
```

Gambar 4.78 Kode Program Untuk Menambah Hasil *Cluster* Ke *Dataset*

Setelah menjalankan kode diatas maka hasil *cluster* akan ditambahkan kedalam *dataset* bernama *cluster* untuk diseleksi *cluster* mana yang menghasilkan jenis suasana hati *angry*, seleksi ini dilakukan secara manual, jadi musik dari *cluster* 0/1/2/3/4 akan dicek satu persatu, dari ke dua *cluster* tersebut *cluster* berapakah yang menghasilkan suasana hati *angry*. Setelah mengetahui *cluster* berapa yang menghasilkan suasana hati *angry* selanjutnya kita akan menambahkan *cluster* tersebut ke dalam *dataset*, kodenya bisa dilihat pada gambar 4.79.



```
df['angry'] = df['cluster'].apply(lambda x: 1 if x == 0 else 0)
```

Gambar 4.79 Kode Program Menambah Hasil Seleksi Suasana Hati *Angry*

Setelah menjalankan kode diatas maka sistem akan menambahkan kolom baru pada *dataset* dengan nama *angry*, jadi kode diatas juga menandai lagu-lagu yang termasuk dalam *cluster* 0 sebagai jenis suasana hati *angry*, jadi pada kolom *angry* nilainya akan menjadi 1 jika dia merupakan *cluster* 0, jika bukan maka akan diberi nilai 0.

3. Pengujian *Silhouette Coefficient*

Setelah mendapatkan hasil dari *clustering* musik selanjutnya kita akan menguji hasil *cluster* dari kedua fitur *energy* dan *tempo* dengan n *cluster* nya 5 menghasilkan skor *Silhouette* berapa, berikut adalah kode programnya pada gambar 4.80.



```
from sklearn.metrics import silhouette_score

X = df_clustered
labels = kmeans.labels_

silhouette_avg = silhouette_score(X, labels)
print("Silhouette Coefficient: {:.3f}".format(silhouette_avg))
```

Gambar 4.80 Kode Program *Silhouette*

Pada kode yang pertama itu, terdapat *library* yang mengimpor fungsi dari

Silhouette_score dari *library sklearn*, selanjutnya *df_clustered* diatas merupakan *DataFrame* atau *dataset*, selanjutnya pada selanjutnya kode: *Silhouette_avg = Silhouette_score(X, labels)*: untuk menghitung skor *Silhouette* rata-rata untuk semua sampel. Fungsi ini membutuhkan data asli 'X' dan label pengelompokan '*labels*' sebagai input, selanjutnya pada kode dibawahnya untuk mencetak hasil dari skor *Silhouette* menjadi 3 angka. Jika program diatas sudah di *running* maka hasil skor *Silhouette* nya bisa dilihat pada gambar 4.81.

Silhouette Coefficient: 0.544

Gambar 4.81 Hasil Skor *Silhouette*

Hasil dari skor *Silhouette* dari hasil *cluster* pada fitur "*energy*" dan "*loudness*" dengan *n cluster* nya yaitu 5 berhasil mendapatkan skor *Silhouette* yaitu 0.544, berdasarkan interpretasi nilai *Silhouette Coefficient* pada tabel 3.2 skor *Silhouette* diatas mendapat interpretasi "struktur yang dihasilkan baik". Dan berikut adalah sampel dari lagu yang berhasil didapatkan dari suasana hati *Angry*, sampel diatas berasal dari hasil *clustering* yang sudah di analisis dan diambil dalam *dataset* yang telah disimpan. Untuk lebih jelasnya bisa dilihat pada tabel 4.5.

Tabel 4.5 Daftar Lagu Pada Suasana Hati *Angry*

Suasana Hati	Kategori Umur	Judul Lagu
<i>Angry</i>	Bayi	Burung Kutilang
	Bayi	<i>Cit Cit Cuit</i>
	Lansia	<i>In These Arms</i>
	Lansia	<i>Livin' On A Prayer</i>
	Anak-anak	Roda Roda Mobil Bis Berputar
	Anak-anak	Lihatlah Lebih Dekat
	Remaja	Mata Ke Hati
	Remaja	Inikah Jatuh Cinta?
	Dewasa	<i>Attention</i>
	Dewasa	<i>Royalty</i>

4.5. Pembuatan Model

Dalam pembuatan model sistem ini saya masih menggunakan bahasa pemrograman Python, selain itu saya juga menggunakan beberapa *library* yang disediakan oleh Python untuk membantu dalam pembuatan model kali ini, berikut adalah *library* yang saya gunakan:

1. *Pickle*, *library* ini digunakan untuk menyimpan model rekomendasi dalam file, agar file tersebut bisa dipanggil dan dibaca kembali oleh program lain.
2. *Pandas*, *library* ini digunakan untuk menyediakan struktur data dan bisa digunakan untuk manipulasi data dan analisis data.
3. *Numpy*, *library* ini menawarkan dukungan *array* besar, dan untuk pengurutan *array*.
4. *Cosine_similarity*, digunakan untuk menghitung kesamaan antara dua set data. Dan digunakan untuk menentukan seberapa mirip fitur lagu yang satu dengan yang lain.
5. *StandardScaler*, digunakan untuk menormalisasi fitur dengan menghapus *mean* dan menskalakan ke unit *variance*.

Berikut adalah program dari model saya bisa dilihat pada gambar 4.82.

```
def recommend_by_mood_and_age(mood, age_group, df, features):
    mood_column = mood.lower()

    filtered_songs = df[(df[mood_column] == 1) & (df['umur'] == age_group)]

    if not filtered_songs.empty:
        scaler = StandardScaler()
        song_features = scaler.fit_transform(filtered_songs[features])
        similarity_matrix = cosine_similarity(song_features)

        reference_song_index = 0
        song_similarity = similarity_matrix[reference_song_index]
        similar_songs_idx = np.argsort(-song_similarity)

        num_songs_to_recommend = min(5, len(filtered_songs))
        recommended_songs = filtered_songs.iloc[similar_songs_idx[:num_songs_to_recommend]]
        return recommended_songs[['track_name']].tolist()
    else:
        return "Tidak ditemukan lagu dengan mood dan kelompok umur tersebut dalam dataset."

features = ['danceability', 'acousticness', 'energy', 'instrumentalness', 'liveness',
            'valence', 'speechiness']

mood_example = "happy"
age_group_example = "remaja"
recommendations_mood_age = recommend_by_mood_and_age(mood_example, age_group_example, df,
features)
print(recommendations_mood_age)
```

Gambar 4.82 Program Dari Model Sistem

Gambar diatas menunjukan kode program dari model yang saya buat agar bisa merekomendasikan berdasarkan masukan dari pengguna berikut penjelasannya pada gambar 4.83.

```
def recommend_by_mood_and_age(mood, age_group, df, features):
    mood_column = mood.lower()
```

Gambar 4.83 Definisi Fungsi

Pada bagian ini mendefinisikan fungsi dengan nama '*recommend_b_mood_and_age*', yang menerima empat argument: '*mood*', '*age_group*', '*df*' (*DataFrame* yang berisi data lagu), dan '*features*' (daftar fitur untuk di analisis kesamaanya). Berikut gambar 4.84 *filtering* data.

```
filtered_songs = df[(df[mood_column] == 1) & (df['umur'] == age_group)]
```

Gambar 4.84 *Filtering* Data

Pada bagian ini berfungsi untuk mengambil lagu-lagu dari *DataFrame* yang cocok dengan suasana hati dan kelompok umur yang diminta oleh pengguna. Berikut gambar 4.85 fungsi *scaller*.

```
if not filtered_songs.empty:
    scaler = StandardScaler()
    song_features = scaler.fit_transform(filtered_songs[features])

    similarity_matrix = cosine_similarity(song_features)
    reference_song_index = 0
    song_similarity = similarity_matrix[reference_song_index]
    similar_songs_idx = np.argsort(-song_similarity)

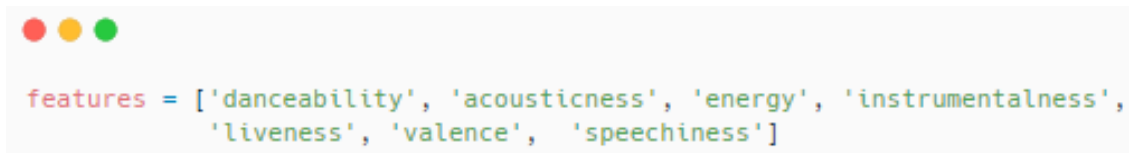
    num_songs_to_recommend = min(5, len(filtered_songs))
    recommended_songs = filtered_songs.iloc[similar_songs_idx[:num_songs_to_recommend]]
    ['track_name'].tolist()

    with open('rekomendasi_mood_age.pkl', 'wb') as f:
        pickle.dump(recommended_songs, f)

    return recommended_songs
else:
    return "Tidak ditemukan lagu dengan mood dan kelompok umur tersebut dalam dataset."
```

Gambar 4.85 Fungsi *Scaller* Dan Kesamaan Kosinus

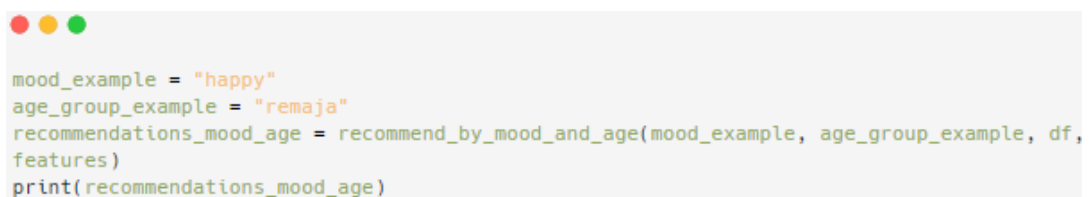
Bagian ini merupakan inti dari sistem rekomendasi yang mengolah lagu-lagu yang telah difilter berdasarkan *mood* dan umur pengguna, menormalkan fitur-fitur penting melalui *StandardScaler*, dan menggunakan kemiripan kosinus untuk menemukan lagu-lagu yang serupa. Setelah memilih sebuah lagu acak sebagai referensi, kode ini menentukan rekomendasi lagu yang sesuai, mengambil URL gambar album dari Spotify, dan menyimpan rekomendasi dalam format *pickle* atau mengembalikannya langsung. Jika tidak ada lagu yang cocok, kode akan memberikan respons bahwa tidak ada lagu yang ditemukan yang sesuai dengan kriteria. Berikut gambar 4.86 *features*.



```
features = ['danceability', 'acousticness', 'energy', 'instrumentalness',
            'liveness', 'valence', 'speechiness']
```

Gambar 4.86 *Features*

Variable '*features*' berisi daftar fitur musik yang mungkin digunakan untuk menentukan rekomendasi lagu berdasarkan fitur-fitur seperti pada gambar 4.50. Fitur-fitur ini akan digunakan untuk memfilter atau menilai lagu dalam *dataset* berdasarkan karakteristiknya, yang sesuai dengan suasana hati dan umur pengguna. Berikut gambar 4.87 pemanggilan fungsi.



```
mood_example = "happy"
age_group_example = "remaja"
recommendations_mood_age = recommend_by_mood_and_age(mood_example, age_group_example, df,
features)
print(recommendations_mood_age)
```

Gambar 4.87 Pemanggilan Fungsi

Kode program diatas berfungsi untuk merekomendasikan dengan meng *input* jenis suasana hati dan umur, pada kode '*mood_example*' dan '*age_group_example*' adalah variable yang berisi contoh *input* untuk suasana hati dan umur, dalam contoh kasus diatas

yaitu 'sad' dan 'dewasa akhir'. Dan untuk fungsi '*recommend_by_mood_and_age*' dipanggil dengan '*mood_example*', '*age_group_example*', *dataset* ('df'), dan 'features' sebagai argumennya. Fungsi ini diasumsikan untuk melakukan pencarian atau pemrosesan dalam *dataset* untuk menemukan lagu yang sesuai dengan kriteria *mood* dan kelompok umur yang diberikan.

4.6. Pembuatan Antar Muka

Dalam pembuatan antar muka aplikasi sistem ini merupakan web sederhana yang dibuat dengan menggunakan *streamlit*, yang berfungsi untuk merekomendasikan lagu berdasarkan suasana hati dan umur pengguna. Website ini menggunakan *DataFrame* dari *pandas* untuk menyimpan dan memanipulasi data lagu, serta *pickle* untuk menyimpan rekomendasi kedalam file. Dan dalam aplikasi ini kode programnya hampir mirip dimana juga mengimplementasikan *content based filtering*, jadi pada tahap ini saya tidak akan menjelaskannya lagi. Berikut adalah penjelasan detail dari setiap bagian kode nya.

1. *Streamlit*, merupakan *library* yang digunakan untuk membuat website. *Streamlit* dapat dengan cepat membuat web interaktif untuk analisi data dan *machine learning* dengan hanya menggunakan Python.
2. *Pandas*, *library* ini digunakan untuk membaca dan memanipulasi *dataset* lagu.
3. *Pickle*, *library* ini digunakan untuk menyimpan rekomendasi lagu kedalam file.
4. *Requests*, *library* ini digunakan untuk mengirim permintaan HTTP seperti GET dan POST ke server. Untuk memudahkan interaksi dengan API web.
5. *Random*, *library* ini digunakan untuk menghasilkan angka atau elemen acak dari daftar, berguna untuk simulasi atau pengujian.
6. *cosine_similarity* (*sklearn.metrics.pairwise*), *library* ini digunakan untuk menghitung

kemiripan kosinus antara vektor, digunakan dalam sistem rekomendasi dan analisis teks untuk menemukan kesamaan.

7. *Base54, Library* ini digunakan untuk *encoding* dan *decoding* data biner ke format teks ASCII, sering digunakan dalam transfer data melalui internet.
8. *numpy, Library* ini digunakan untuk Mendukung operasi matematika kompleks dan manipulasi array atau matriks besar, fundamental dalam pengolahan data numerik.
9. *StandardScaler (sklearn.preprocessing):, Library* ini digunakan untuk Menormalkan data dengan menghilangkan rata-rata dan menskalakan ke *varians* unit, penting dalam persiapan data sebelum analisis machine learning.

Berikut adalah kode program antar muka saya:

```
import requests
import random
import streamlit as st
import pandas as pd
import numpy as np
import base64
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.preprocessing import StandardScaler

def get_spotify_access_token(client_id, client_secret):
    auth_url = 'https://accounts.spotify.com/api/token'
    auth_response = requests.post(auth_url, {
        'grant_type': 'client_credentials'
    }, headers={
        'Authorization': f'Basic {base64.b64encode(f'{client_id}:{client_secret}'.encode()).decode()}'
    })
    if auth_response.status_code == 200:
        return auth_response.json()['access_token']
    else:
        return None

def get_album_art(track_id, token):
    spotify_endpoint = f"https://api.spotify.com/v1/tracks/{track_id}"
    headers = {"Authorization": f"Bearer {token}"}
```

```

        response = requests.get(spotify_endpoint,
                                headers=headers)
        if response.status_code == 200:
            track_data = response.json()
            return track_data['album']['images'][0]['url']
        else:
            return None

CLIENT_ID = '8a056f31514c4db2a8b2048086f6e3ef'
CLIENT_SECRET = '0fcffe48f39e4440a4fb68c77d42d5bc'
def kategorikan_umur(umur):
    if umur < 5:
        return 'bayi'
    elif umur <= 9:
        return 'anak-anak'
    elif umur <= 19:
        return 'remaja'
    elif umur <= 59:
        return 'dewasa'
    elif umur <= 85:
        return 'lansia'

def recommend_by_mood_and_age(mood, umur_pengguna, df,
                              features, token):
    age_group = kategorikan_umur(umur_pengguna)
    mood_column = mood.lower()
    filtered_songs = df[(df[mood_column] == 1) &
                        (df['umur'] == age_group)]

    if not filtered_songs.empty:
        scaler = StandardScaler()
        song_features =
scaler.fit_transform(filtered_songs[features])

        similarity_matrix =
cosine_similarity(song_features)
        reference_song_index = random.randint(0,
len(filtered_songs) - 1)
        song_similarity =
similarity_matrix[reference_song_index]

        similar_songs_idx = np.argsort(-
song_similarity)[:min(10, len(filtered_songs))]
        recommended_songs =
filtered_songs.iloc[similar_songs_idx][['track_name',
'track_id']].to_dict('records')

```

```

        for song in recommended_songs:
            song['image_url'] =
get_album_art(song['track_id'], token)
        return recommended_songs
        with open('rekomendasi_mood_age.pkl', 'wb') as f:
            pickle.dump(recommended_songs, f)

        return recommended_songs
    else:
return "Tidak ditemukan lagu dengan mood dan kelompok
umur tersebut dalam dataset."

features = ['danceability', 'acousticness', 'energy',
'instrumentalness', 'tempo', 'valence', 'speechiness']
df = pd.read_csv('HasilClusterRevisi.csv')
st.title('Sistem Rekomendasi Musik Berdasarkan Suasana
Hati dan Umur Pada Spotify Menggunakan Metode K-Means')

umur_pengguna = st.number_input('Masukkan Umur',
min_value=0, max_value=120, step=1)
mood = st.selectbox('Pilih Suasana Hati', ['calm',
'energetic', 'sad', 'happy', 'angry'])

if st.button('Rekomendasikan'):
    spotify_access_token =
get_spotify_access_token(CLIENT_ID, CLIENT_SECRET)

    if spotify_access_token:
        recommendations =
recommend_by_mood_and_age(mood,
umur_pengguna, df, features, spotify_access_token)
        if isinstance(recommendations, list):
            st.write('Rekomendasi Lagu:')
            cols = st.columns(5)
            col_index = 0
            for idx, song in
enumerate(recommendations):
                if idx % 5 == 0 and idx > 0:
                    cols = st.columns(5)
                    col_index = 0
                with cols[col_index]:
                    st.markdown(f"""
                        <div style='text-align: center;
border: 2px solid black; margin-bottom:
20px; border-radius: 5px;'>
                        <a style='text-decoration: none; color: black;'

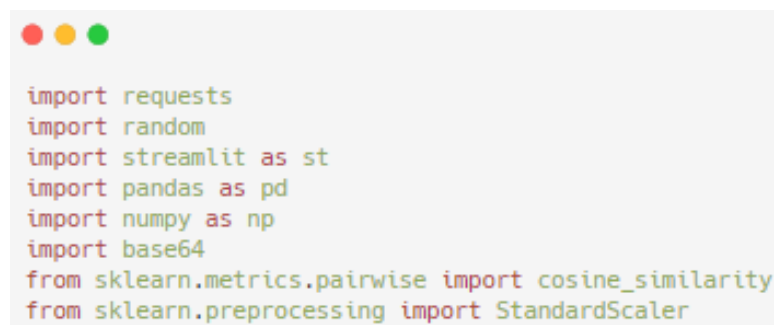
```

```

href='https://open.spotify.com/track/{song['track_id']}'
' target='_blank'>
<img src='{song['image_url']}' style='width:100%;
border-bottom: 1px solid #ccc;'/>
        <p style='padding: 10px 0;
margin: 0;'>{song['track_name']}</p>
        </a>
    </div>
    """ , unsafe_allow_html=True)
    col_index = (col_index + 1) % 5
else:
    st.error(recommendations)
else:
    st.error("Gagal mendapatkan token akses dari
    Spotify.")

```

Berikut adalah penjelasan perkode dari antar muka saya. Untuk lebih jelasnya bisa dilihat pada gambar 4.88 library yang digunakan.



```

import requests
import random
import streamlit as st
import pandas as pd
import numpy as np
import base64
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.preprocessing import StandardScaler

```

Gambar 4.88 *Library Yang Digunakan*

Tahap ini mengimport *library* yang akan digunakan dalam kode ini. Berikut gambar 4.89 request token.



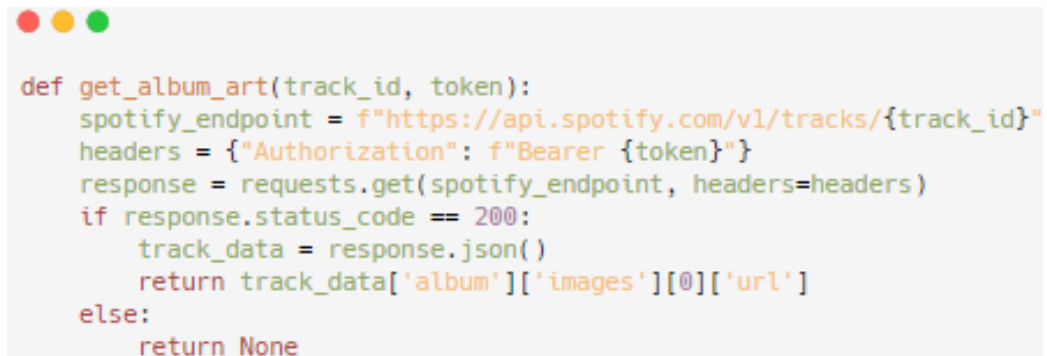
```

def get_spotify_access_token(client_id, client_secret):
    auth_url = 'https://accounts.spotify.com/api/token'
    auth_response = requests.post(auth_url, {
        'grant_type': 'client_credentials'
    }, headers={
        'Authorization': f'Basic {base64.b64encode(f'{client_id}:{client_secret}'.encode())}'
    })
    if auth_response.status_code == 200:
        return auth_response.json()['access_token']
    else:
        return None

```

Gambar 4.89 *Request Token*

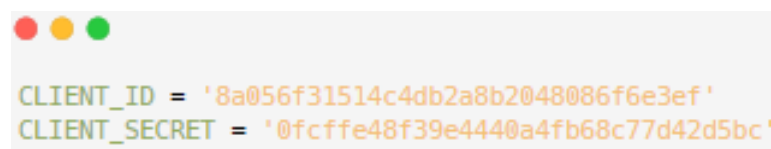
Fungsi ini bertujuan untuk meminta *access token* agar bisa digunakan oleh API dalam mengambil gambar album untuk ditampilkan dalam aplikasi. Berikut gambar 4.90 fungsi.



```
def get_album_art(track_id, token):
    spotify_endpoint = f"https://api.spotify.com/v1/tracks/{track_id}"
    headers = {"Authorization": f"Bearer {token}"}
    response = requests.get(spotify_endpoint, headers=headers)
    if response.status_code == 200:
        track_data = response.json()
        return track_data['album']['images'][0]['url']
    else:
        return None
```

Gambar 4.90 Fungsi Pengambilan Gambar Album

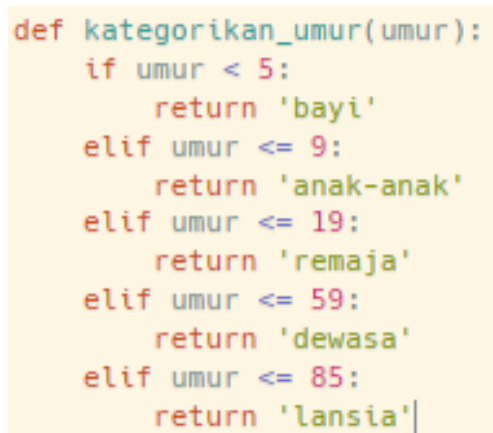
Fungsi dari kode *get_album_art* diatas untuk mengambil gambar album dari lagu-lagu, dengan menggunakan *track_id* dan *token* yang kita dapatkan dari kode sebelumnya. Berikut gambar 4.91 API Apotify.



```
CLIENT_ID = '8a056f31514c4db2a8b2048086f6e3ef'
CLIENT_SECRET = '0fcffe48f39e4440a4fb68c77d42d5bc'
```

Gambar 4.91 API Spotify

Gambar diatas merupakan API Spotify yang digunakan untuk mengambil foto album dari setiap lagu yang ditampilkan dalam aplikasi. Berikut gambar 4.92 Kategori umur.



```
def kategorikan_umur(umur):
    if umur < 5:
        return 'bayi'
    elif umur <= 9:
        return 'anak-anak'
    elif umur <= 19:
        return 'remaja'
    elif umur <= 59:
        return 'dewasa'
    elif umur <= 85:
        return 'lansia'
```

Gambar 4.92 Kategori Umur

Fungsi dari bagian ini yaitu untuk mengkategorikan kelompok umur, jadi fungsi ini nanti akan bekerja di tampilan aplikasi, jika *user* menginput umur nya maka, program ini akan secara otomatis mengkategorikan umur *user*. Berikut gambar 4.93.

```
def recommend_by_mood_and_age(mood, umur_pengguna, df, features, token):
    age_group = kategorikan_umur(umur_pengguna)
    mood_column = mood.lower()
    filtered_songs = df[(df[mood_column] == 1) & (df['umur'] == age_group)]
```

Gambar 4.93 Fungsi Rekomendasi

Bagian ini untuk memberikan rekomendasi dari data yang dalam *DataFrame* 'DF' yang cocok dengan suasana hati dan umur yang di input oleh pengguna. Berikut gambar 4.94 fungsi *scaller*.

```
if not filtered_songs.empty:
    scaler = StandardScaler()
    song_features = scaler.fit_transform(filtered_songs[features])
    similarity_matrix = cosine_similarity(song_features)

    reference_song_index = random.randint(0, len(filtered_songs) - 1)
    song_similarity = similarity_matrix[reference_song_index]

    similar_songs_idx = np.argsort(-song_similarity)[:min(10, len(filtered_songs))]
    recommended_songs = filtered_songs.iloc[similar_songs_idx][['track_name',
                                                                'track_id']].to_dict('records')

    for song in recommended_songs:
        song['image_url'] = get_album_art(song['track_id'], token)

    return recommended_songs
    with open('rekomendasi_mood_age.pkl', 'wb') as f:
        pickle.dump(recommended_songs, f)

    return recommended_songs
else:
    return "Tidak ditemukan lagu dengan mood dan kelompok umur tersebut dalam dataset."
```

Gambar 4.94 Fungsi *scaller* dan kesamaan kosinus

Bagian ini merupakan inti dari sistem rekomendasi yang mengolah lagu-lagu yang telah difilter berdasarkan *mood* dan umur pengguna, menormalkan fitur-fitur penting melalui *StandardScaler*, dan menggunakan kemiripan kosinus untuk menemukan lagu-lagu yang serupa. Setelah memilih sebuah lagu acak sebagai referensi, kode ini menentukan rekomendasi lagu yang sesuai, mengambil URL gambar album dari Spotify, dan menyimpan

rekomendasi dalam format *pickle* atau mengembalikannya langsung. Jika tidak ada lagu yang cocok, kode akan memberikan respons bahwa tidak ada lagu yang ditemukan yang sesuai dengan kriteria. Berikut gambar 4.95 membaca *dataset*.

```
df = pd.read_csv('HasilClusterRevisi.csv')
```

Gambar 4.95 Membaca *Dataset*

Fungsi ini untuk membaca *dataset* agar program lain dapat membaca data yang ada di dalam dan memberikan rekomendasi sesuai *dataset*. Berikut gambar 4.96 *Title Aplikasi*.

```
st.title('Sistem Rekomendasi Musik Berdasarkan Suasana Hati dan Umur Pada Spotify  
Menggunakan Metode K-Means')  
  
umur_pengguna = st.number_input('Masukkan Umur', min_value=0, max_value=120, step=1)  
mood = st.selectbox('Pilih Suasana Hati', ['calm', 'energetic', 'sad', 'happy', 'angry'])
```

Gambar 4.96 *Title Aplikasi*

Bagian ini untuk membuat *title* aplikasi dan membuat *form input* dan *selectbox* agar *user* menginput umur dan memilih jenis suasana hati. Berikut gambar 4.97 fungsi *button*.

```
if st.button('Rekomendasikan'):
    spotify_access_token = get_spotify_access_token(CLIENT_ID, CLIENT_SECRET)

    if spotify_access_token:
        recommendations = recommend_by_mood_and_age(mood, umur_pengguna, df, features,
        spotify_access_token)
        if isinstance(recommendations, list):
            st.write('Rekomendasi Lagu:')
            cols = st.columns(5)
            col_index = 0
            for idx, song in enumerate(recommendations):
                if idx % 5 == 0 and idx > 0:
                    cols = st.columns(5)
                    col_index = 0
                    with cols[col_index]:
                        st.markdown(f"""
                else:
                    st.error(recommendations)
            else:
                st.error("Gagal mendapatkan token akses dari Spotify.")
```

Gambar 4.97 Fungsi *Button* Rekomendasi

Bagian ini merupakan tampilan dan fungsi dari *button* rekomendasi, jadi ketika *user* menekan tombol rekomendasi maka kode program akan dijalankan fungsinya agar dapat menampilkan hasil rekomendasi yang sesuai dengan masukan *user*. Berikut gambar 4.98.

Berdasarkan Suasana Hati dan Umur Pada Spotify Menggunakan Metode K- Means

Masukkan Umur

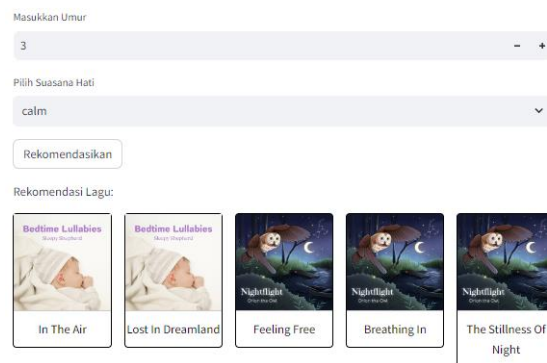
3

Pilih Suasana Hati

calm

Rekomendasikan

Rekomendasi Lagu:



Gambar 4.98 Hasil *Running* Kelompok Umur bayi

Berdasarkan gambar diatas, berikut adalah hasil lebih detail dari rekomendasi berdasarkan masukan *user*. Untuk lebih jelasnya dapat dilihat pada tabel 4.6.

Tabel 4.6 Rekomendasi Kelompok Umur bayi

Kelompok Umur	Suasana Hati	Rekomendasi
Bayi < 5 tahun	<i>Calm</i>	- <i>In The Air</i> - <i>Lost In Dreamland</i> - <i>Feeling Free</i> - <i>Breathing In</i> - <i>The Stillness Of Night</i> - <i>Melody Of The Moon</i> - <i>Tomorrow</i> - <i>Forever</i> - <i>Days Go By</i> - <i>Cosy Bad</i>
	<i>Energic</i>	- <i>Into The Clouds</i> - <i>Goodnight Little Dreamer</i> - <i>At Dawn</i> - <i>Leise, Peterle Leise</i> - <i>Schotter Fahren</i> - <i>Ein Karmel Reitet Durch Das Land</i> - <i>Tausend Kleine Tausendfubler</i> - <i>Alle Meine Fingerlein Wollen Heute Tere Sein</i>
	<i>Sad</i>	- <i>Weißt Du, Wie Viel Sternlein Stehen</i> - <i>Willkommen Im Leben</i> - <i>Ein Paar Kleine Katzenkinder</i> - <i>Heia, Heia, Bu, Bu</i>

		- Der Wind, Der Regen Und Die Sonne - Segelboot Partie - Subes Einhorn - Bruder Jakob - Der Mond Ist Aufgegangen - Mach Die Augen Zu
	Happy	- Hoppel-Di-Hoppel-Di, Haschen Hupf - Imse Bimse Spinne - Ich Bin Ein Kleines Pony - Ich Flieg In Meinem Flugzeug - Guteng Tag, Herr Nasenmann - Ich Bin Der Kleine Hampelmann - Alle Maine Fingerlein Wollen Heute Tiere Sein - Eine Schnecke Tragt Ein Schweres Haus - Schotter Fahren - So Fahren Die Damen
	Angry	- Babykrokodile Machen Schnapp, Schnapp, Schnapp - Wir Krabbeln Durch Den Garten - Aus Dem Weg, Sie Fahren Los - Das Quietscheentchen - Feuerwehrautos - Fahren, Fahren Mit Dem Kinderwagen - Toff, Toff, Toff, Die Eisenbahn - Baby Shark - Family Maus - Eine Kleine Dickmadam

Berikut gambar 4.99 hasil *running* berdasarkan kelompok umur anak-anak dan suasana hati.

Masukkan Umur

8

Pilih Suasana Hati

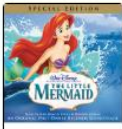
calm

Rekomendasikan

Rekomendasi Lagu:



Don't Worry, Be Happy (From "Wall-E")



Part of Your World



In Summer - From "Frozen"/Soundtrack Version



Can You Feel the Love Tonight



Lihatlah Lebih Dekat

Gambar 4.99 Hasil *Running* Kelompok Umur Anak Anak

Berdasarkan gambar 4.99, berikut adalah hasil lebih detail dari rekomendasi berdasarkan masukan *user*. Untuk lebih jelasnya dapat dilihat pada tabel 4.7.

Tabel 4.7 Rekomendasi Kelompok Umur Anak-Anak

Kelompok Umur	Suasana Hati	Rekomendasi
Anak anak	<i>Calm</i>	- <i>Don't Worry, Be Happy</i> - <i>Part Of Your World</i> - <i>In Summer</i> - <i>Can You Feel The Love Tonight</i> - Lihatlah Lebih Dekat - <i>Clap Your Hands</i> - Lihatlah Lebih Dekat - <i>Almost There</i> - <i>Chabelita Oh Chabelita</i> - Petualangan Sherina
	<i>Energic</i>	- Hakuna Matata - Andai Aku Besar Nanti - <i>Wiggly Wiggles Freeze Dance</i> - <i>Hot Potato</i> - <i>Bob The Builder's</i> - Kring Kring Ada Sepeda - Kupu Kupu Yang Lucu - <i>Happy</i> - <i>Shake It Off</i>
	<i>Sad</i>	- Andai Aku Besar Nanti - <i>Into The Unknown</i> - <i>Let It Go</i> - Pelangiku - <i>Frozen Heart</i> - <i>For The First Time In Forever</i> - <i>Some Things Never Change</i> - <i>Do You Want To Build A Snowman?</i> - <i>A Whole New World</i> - Petualangan Serina
	<i>Happy</i>	- Disini Senang Disana Senang - Dua Mata Saya - <i>If All Of The Raindrops</i> - Ini Indonesiaku - Salamku Kawan - <i>Action Song</i> - Bis Sekolah - <i>Happy "Despicable Me 2"</i> - Kring Kring Ada Sepeda - <i>Shiny</i>

	Angry	- Tik Tik Tik Bunyi Hujan - Balonku - Kereta Api - Kepala Pundak Lutut Kaki - Nama Nama Hari - Semut Semut Kecil - Bangun Tidur - Abang Tukang Bakso - Naik Naik Kepuncak Gunung - Laba Laba Kecil
--	-------	---

Berikut gambar 4.100 merupakan hasil *running* program berdasarkan kelompok umur remaja dan suasana hati.

Sistem Rekomendasi Musik Berdasarkan Suasana Hati dan Umur Pada Spotify Menggunakan Metode K-Means

Masukkan Umur

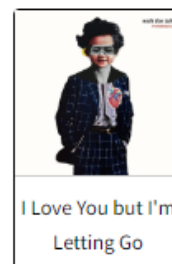
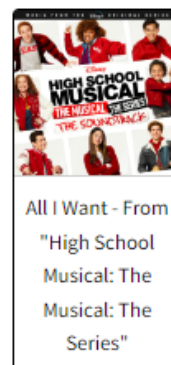
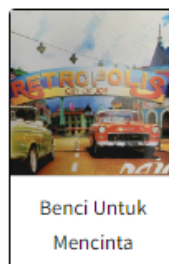
16

Pilih Suasana Hati

calm

Rekomendasikan

Rekomendasi Lagu:



Gambar 4.100 Hasil *Running* Kelompok Umur Remaja

Berdasarkan gambar diatas, berikut adalah hasil lebih detail dari rekomendasi

berdasarkan masukan *user*. Untuk lebih jelasnya dapat dilihat pada tabel 4.8.

Tabel 4.8 Rekomendasi Kelompok Umur Remaja

Kelompok Umur	Suasana Hati	Rekomendasi
Remaja	<i>Calm</i>	- <i>High Hopes</i> - Kemana Kita Hari Ini - Benci Untuk Mencinta - <i>All I Want</i> - <i>I Love You But I'm Letting Go</i> - Takkan Kemana - <i>Mine</i> - Malam Mingguku - Selamat (Selamat Tinggal) - Bawalah Cintaku
	<i>Energic</i>	- Kuat Kita Sinar - Terlatih Patah Hati - <i>Superheroes</i> - <i>Band Aid</i> - <i>Shinunoga E-Wa</i> - Cantik - Kau Adalah - Kala Cinta Menggoda - <i>Wish You Were Here</i> - <i>Cruel Summer</i>
	<i>Sad</i>	- Sayap Pelindungmu - Siapkah Kau 'Tuk Jatuh Cinta Lagi - Sayap Pelindungmu - <i>Cigarettes Of Ours</i> - Teman Hidup - <i>Habbits</i> - <i>Changes</i> - Diri - <i>Blue Jeans</i> - Melawan Restu
	<i>Happy</i>	- Sekuat Hatimu - <i>Middle Of The Night</i> - Unstoppable - <i>Up&Up</i> - <i>SNAP</i> - <i>Favorite Girl</i> - <i>Sick Feeling</i> - <i>Die First</i> - <i>Superheroes</i> - <i>Hero</i>
	<i>Angry</i>	- Pelukku Untuk Pelikmu

		- Hero - Duka - <i>Better</i> - Serana - <i>Shanelles</i> - <i>Sweet Car</i> - <i>Sweater Weather</i> - <i>Just The Way You Are</i> - <i>House Of Memories</i>
--	--	--

Berikut gambar 4.101 merupakan hasil *running* program berdasarkan kelompok umur dewasa dan suasana hati.

Sistem Rekomendasi Musik Berdasarkan Suasana Hati dan Umur Pada Spotify Menggunakan Metode K-Means

Masukkan Umur

30 - +

Pilih Suasana Hati

calm v

Rekomendasikan

Rekomendasi Lagu:



Rehat



Selaras



Tidak Ada Salju Di Sini Pt.6 (Selamat Jalan)



Sudut Memori



Pilu Membriru

Gambar 4.101 Hasil *Running* Kelompok Umur Dewasa

Berdasarkan gambar diatas berikut adalah hasil lebih detail dari rekomendasi berdasarkan masukan *user*. Untuk lebih jelasnya dapat dilihat pada tabel 4.9.

Tabel 4.9 Rekomendasi Kelompok Umur Dewasa

Kelompok Umur	Suasana Hati	Rekomendasi
Dewasa	<i>Calm</i>	- Rehat - Selaras

		- Tidak Ada Salju Disini - Sudut Memori - Pilu Membiru - Melawan Hati - Sulung - <i>Hey Little Girl</i> - Intuisi - Tenang
	<i>Energic</i>	- <i>The Way I Loved You</i> - <i>You Know It</i> - <i>This Side Of Paradise</i> - <i>Angry To</i> - <i>Where'd All The Time Go</i> - <i>Stargazin</i> - <i>The Kid I Used To Know</i> - <i>Alone</i> - <i>Feels Like Summer</i> - <i>Out Of My Language</i>
	<i>Sad</i>	- <i>The Doctor Said</i> - <i>Dna</i> - Tuhan Sebut Sia-Sia - <i>Burned</i> - Tenang - <i>Disney</i> - Hitam Putih - Garis Terdepan - Hal Indah Butuh Waktu Untuk Dating - Sudah
	<i>Happy</i>	- <i>You Know It</i> - <i>Where'd All The Time Go?</i> - <i>I Always Knew</i> - <i>Out Off My Language</i> - <i>Mr. Brightside</i> - <i>The Way I Loved You</i> - <i>Now Or Never</i> - <i>Angry To</i> - <i>New Person, Same Old Mistakes</i> - <i>Alone</i>
	<i>Angry</i>	- <i>Cruel Summer</i> - <i>New Friends</i> - <i>Joke's On You</i> - <i>Hard Boy</i> - <i>Angry Too</i> - <i>Honestly – Encore</i> - <i>Killing Butterflies</i> - <i>Take A Hint</i>

		- <i>Where'd All The Time Go?</i> - <i>Alone</i>
--	--	---

Berikut gambar 4.102 merupakan hasil *running* program berdasarkan kelompok umur lansia dan suasana hati.

Sistem Rekomendasi Musik Berdasarkan Suasana Hati dan Umur Pada Spotify Menggunakan Metode K-Means

Masukkan Umur

62

Pilih Suasana Hati

calm

Rekomendasikan

Rekomendasi Lagu:







Gambar 4.102 Hasil *Running* Kelompok Umur Lansia

Berdasarkan gambar diatas, berikut adalah hasil lebih detail dari rekomendasi berdasarkan masukan *user*. Untuk lebih jelasnya dapat dilihat pada tabel 4.10.

Tabel 4.10 Rekomendasi Kelompok Umur Lansia

Kelompok Umur	Suasana Hati	Rekomendasi
Lansia	<i>Calm</i>	- Lilin Lilin Kecil - Kucoba Hidup Sendiri - Senandung Doa - Biar Kucari Jalanku - Hati Yang Terluka - Tirai - Sepanjang Jalan Kenangan - Antara Benci Dan Rindu - Angin Malam - Selamat Tinggal
	<i>Energic</i>	- <i>Bitter Sweet Symphony</i> - <i>Am I Ever Gonna See Your Face Again</i>

		- <i>Viva Forever</i> - <i>You'll Never Know</i> - <i>Kekasih</i> - <i>Rain</i>
	<i>Sad</i>	- <i>Risau</i> - <i>Aku Tak Biasa</i> - <i>Maafkan</i> - <i>Kisah Cintaku</i> - <i>Hati Lembur Jadi Debu</i> - <i>Kucari Jalan Terbaik</i> - <i>The Flame</i> - <i>Merpati Tak Pernah Ingkar Janji</i> - <i>Gereja Tua</i> - <i>Kupu – Kupu Malam</i>
	<i>Happy</i>	- <i>Bing</i> - <i>Piano Man</i> - <i>Tiada Cinta Yang Lain</i> - <i>Nyanyian Suara Hati</i> - <i>All I Want Is You</i> - <i>The Night We Met</i> - <i>Yang Kunanti</i> - <i>Biarlah Bulan Bicara</i> - <i>Champagne Supernova</i> - <i>Women In Uniform – 1994 Remaster</i>
	<i>Angry</i>	- <i>Animal</i> - <i>All The Small Things</i> - <i>Yesterdays Hero</i> - <i>You'll Never Know</i> - <i>Radio Nowhere</i> - <i>And We Danced</i> - <i>Bitter Sweet Shimphony</i> - <i>Am I Ever Gonna See Your Face Again</i> - <i>Livin' On A Prayer</i> - <i>You Give Love A Bad Name</i>

4.7. Hasil Evaluasi *Silhouette Score*

Hasil evaluasi *Silhouette Score* dari hasil tiap-tiap *cluster* yang dilakukan bisa dilihat pada tabel 4.11.

Tabel 4.11 Evaluasi *Silhouette Score*

No	Cluster	Skor <i>Silhouette</i>	Interpretaasi Skor
1	<i>calm</i>	0.567	Struktur yang dihasilkan baik

2	<i>Energic</i>	0.547	Struktur yang dihasilkan baik
3	<i>Sad</i>	0.431	Struktur yang dihasilkan lemah
4	<i>happy</i>	0.514	Struktur yang dihasilkan baik
5	<i>Angry</i>	0.544	Struktur yang dihasilkan baik

Berdasarkan tabel 4.11 Hasil menunjukkan bahwa algoritma *K-Means* berhasil mengidentifikasi dan mengelompokkan trek musik ke dalam empat *cluster mood* dengan tingkat keakuratan yang beragam, sebagaimana ditunjukkan oleh Skor *Silhouette*.

Dengan nilai Skor *Silhouette* 0.800, *cluster calm* ini menunjukkan bahwa lagu-lagu di dalamnya sangat beragam dan pas dengan suasana hati yang tenang. Ciri-ciri lagu seperti irama lambat, suara yang halus, dan keserasian yang menenangkan menunjukkan bahwa lagu-lagu ini berhasil memberikan rasa tenang. Keberhasilan dalam mengelompokkan lagu-lagu ini menunjukkan bahwa kita mengerti apa saja yang bisa membuat lagu membuat pendengarnya merasa tenang.

Skor *Silhouette* sebesar 0.574 menunjukkan bahwa *cluster energic* juga memiliki struktur yang kuat. Trek-trek dalam *cluster* ini dapat menyampaikan *energic* dan semangat, dengan dominasi fitur seperti *tempo* cepat, *ritme* yang kuat, dan melodi yang memotivasi. Ini mencerminkan keefektifan algoritma dalam mengidentifikasi trek-trek yang dapat meningkatkan semangat.

Dengan nilai Skor *Silhouette* 0.581, *cluster sad* ini cukup baik meskipun ada banyak variasi dalam menunjukkan suasana hati *sad*. Ini berarti lagu-lagu di dalamnya memiliki beragam irama dan nada, tapi semuanya berhasil dikelompokkan karena kesedihan yang mereka sampaikan. Bisa jadi karena melodi atau kata-kata dalam lirik yang menyentuh hati.

Dengan nilai Skor *Silhouette* 0.543, walaupun lebih rendah dari kelompok lain, kelompok lagu *happy* ini tetap menunjukkan bahwa mereka cocok dengan suasana hati

happy. Karena ada banyak variasi dalam kelompok ini, ini bisa berarti bahwa kebahagiaan bisa datang dari jenis musik yang berbeda.

4.8. **Black Box Testing**

Pada tahapan pengujian sistem ini, sistem akan diuji menggunakan pengujian *black box* dimana pengujian ini akan menguji apakah sistem berfungsi dengan baik atau tidak, berikut adalah tabel 4.12 pengujian *black box*.

Table 4.12 Pengujian *Black box*

No	Input	Aksi	Output	Ket
1.	User memilih kelompok umur anak anak dan suasana hati <i>calm</i>	Tombol rekomendasi	Menampilkan rekomendasi lagu	Berhasil
2.	User memilih kelompok umur anak anak dan suasana hati <i>energetic</i>	Tombol rekomendasi	Menampilkan rekomendasi lagu	Berhasil
3.	User memilih kelompok umur anak anak dan suasana hati <i>sad</i>	Tombol rekomendasi	Menampilkan rekomendasi lagu	Berhasil
4.	User memilih kelompok umur anak anak dan suasana hati <i>happy</i>	Tombol rekomendasi	Menampilkan rekomendasi lagu	berhasil
5.	User memilih kelompok umur remaja dan suasana hati <i>calm</i>	Tombol rekomendasi	Menampilkan rekomendasi lagu	Berhasil
6.	User memilih kelompok umur remaja dan suasana hati <i>energetic</i>	Tombol rekomendasi	Menampilkan rekomendasi lagu	Berhasil
7.	User memilih kelompok umur remaja dan suasana hati <i>sad</i>	Tombol rekomendasi	Menampilkan rekomendasi lagu	Berhasil
8.	User memilih kelompok umur remaja dan suasana hati <i>happy</i>	Tombol rekomendasi	Menampilkan rekomendasi lagu	berhasil
9.	User memilih kelompok umur dewasa dan suasana hati <i>calm</i>	Tombol rekomendasi	Menampilkan rekomendasi lagu	berhasil
10.	User memilih kelompok umur dewasa dan suasana hati <i>energetic</i>	Tombol rekomendasi	Menampilkan rekomendasi lagu	berhasil
11.	User memilih kelompok umur dewasa dan suasana hati <i>sad</i>	Tombol rekomendasi	Menampilkan rekomendasi lagu	berhasil

12.	User memilih kelompok umur dewasa dan suasana hati <i>happy</i>	Tombol rekomendasi	Menampilkan rekomendasi lagu	berhasil
-----	---	--------------------	------------------------------	----------

4.9. Analisis

Tahap ini kita akan melakukan analisis terhadap *cluster* yang dihasilkan oleh *kmeans* apakah hasil dari *kmeans* untuk pengelompokan jenis-jenis suasana hati yang sudah dilakukan apakah menghasilkan suasana hati yang akurat dan tepat atau tidak.

4.9.1. Analisis Suasana Hati *Calm*

Suasana hati *calm* dalam musik seringkali dikaitkan dengan lagu-lagu yang memiliki karakteristik tertentu yang dapat menimbulkan perasaan relaksasi atau ketenangan pada pendengarnya. Dua fitur yang sering digunakan untuk mengidentifikasi lagu dengan suasana hati ini adalah “*instrumentalness*” dan “*tempo*”. Berikut penjelasan kenapa menggunakan kedua fitur tersebut.

1. *Instrumentalness*

Fitur ini dalam analisis musik mengukur probabilitas sebuah lagu tidak Memiliki *vocal*. Nilai mendekati 1.0 menunjukkan bahwa lagu tersebut kemungkinan besar instrumental dan yang mendekati 0 menunjukkan lagu tersebut mengandung vokal. Musik instrumental cenderung lebih menenangkan karena kurangnya lirik, yang memungkinkan pendengar untuk rileks dan menghilangkan distraksi yang mungkin disebabkan oleh kata-kata.

2. *Tempo*

Fitur ini mengukur kecepatan atau laju dari sebuah komposisi musik dalam *beats per minute* (BPM). *Tempo* lagu juga mempengaruhi bagaimana perasaan musik tersebut dirasakan, *tempo* yang cepat itu biasanya juga mengandung *energetic*, sedangkan *tempo* yang lambat itu biasanya cenderung menenangkan atau lagu yang tenang. Dan lagu

dengan *tempo* yang lambat, biasanya dianggap menenangkan karena ritmenya lebih sedikit, laju yang lebih lambat memungkinkan pendengar untuk mengambil alih dan bersantai.

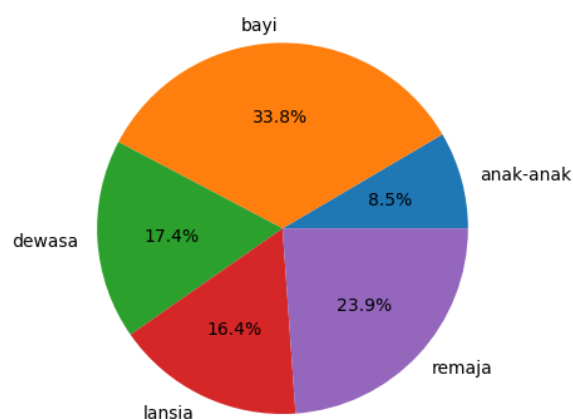
3. Analisis

Dengan menggunakan kedua fitur diatas dengan 5 *n-cluster*, suasana hati *calm* berhasil di dapatkan dengan menggunakan *kmeans* untuk dilakukan *clustering*, dan hasilnya saya mengambil *cluster* 0 dimana *cluster* dengan nilai *tempo* lambat dan *instrumentalness* beragam, berhasil mendapatkan jenis suasana hati *calm* dengan hasil skor *Silhouette Coefficient* 0.567. dengan ini, terbukti kalau *cluster* yang dihasilkan oleh *kmeans* cukup baik.

4. Kesimpulan

Berdasarkan penjelasan analisis diatas maka hasil dari *clustering* untuk mencari suasana hati *calm* dapat disimpulkan memiliki nilai yang baik berdasarkan skor *Silhouette Coefficient*, namun berdasarkan hasil evaluasi yang dilakukan secara manual menurut saya kurang akurat khususnya pada beberapa kategori umur, dan pada kategori umur tertentu terdapat banyak musik yang tidak berhubungan dengan suasana hati yang dicari, walau demikian masih terdapat lagu yang sama dan nyambung dengan suasana hati yang dicari.

Berikut merupakan hasil diagram berdasarkan jenis suasana hati dan umur, bisa dilihat pada gambar 4.103.



Gambar 4.103 Distribusi Suasana Hati *Calm* Berdasarkan Kategori Umur

Gambar diatas merupakan diagram atau *pie chart* dari hasil dari distribusi suasana hati berdasarkan kategori umur, dan dapat juga dilihat bahwa pada suasana hati *calm* didominasi oleh musik dari kategori umur bayi dengan 33.8%, selanjutnya yaitu remaja 23.9% dan dewasa 17.4%, dan lansia 16.4%, dan yang terakhir yaitu anak-anak dengan 8.5%.

4.9.2. Analisis Suasana Hati *Energic*

Suasana hati *energic* dalam musik seringkali dikaitkan atau dihubungkan dengan lagu-lagu yang memiliki karakteristik tertentu yang dapat menimbulkan perasaan semangat pada pengguna atau pendengarnya. Dua fitur yang sering digunakan atau dipakai untuk mencari atau mengidentifikasi jenis lagu dengan suasana hati ini adalah fitur "*danceability*" dan juga fitur "*tempo*". Berikut penjelasan lengkapnya kenapa menggunakan kedua fitur tersebut.

1. *Danceability*

Danceability dalam analisis musik mengukur seberapa cocok sebuah lagu untuk menari berdasarkan kombinasi elemen musiknya seperti ritme, stabilitas *tempo*, kekuatan *beat*, dan keseluruhan regularitas. Skor yang lebih tinggi pada skala *danceability* menunjukkan bahwa lagu tersebut lebih cocok untuk menari. Lagu dengan skor *danceability* tinggi seringkali memiliki *beat* yang kuat dan ritme yang menonjol, menciptakan suasana yang energik dan meningkatkan dorongan untuk bergerak atau menari. Elemen-elemen ini menyebabkan lagu tersebut mampu membangkitkan energi dan kegembiraan, yang sangat sesuai dengan suasana hati *energic*.

2. *Tempo*

Fitur ini mengukur kecepatan atau laju dari sebuah komposisi musik dalam *beats per*

minute (BPM). *Tempo* lagu mempengaruhi bagaimana perasaan musik tersebut dirasakan, *tempo* yang cepat biasanya mengandung *energetic*, sedangkan *tempo* yang lambat cenderung menenangkan. Lagu dengan *tempo* tinggi seringkali terasa lebih energik dan memacu adrenalin. *Tempo* yang cepat dapat memperkuat efek dari *danceability*, dengan mempercepat ritme jantung pendengar dan meningkatkan tingkat energi secara keseluruhan.

3. Analisis

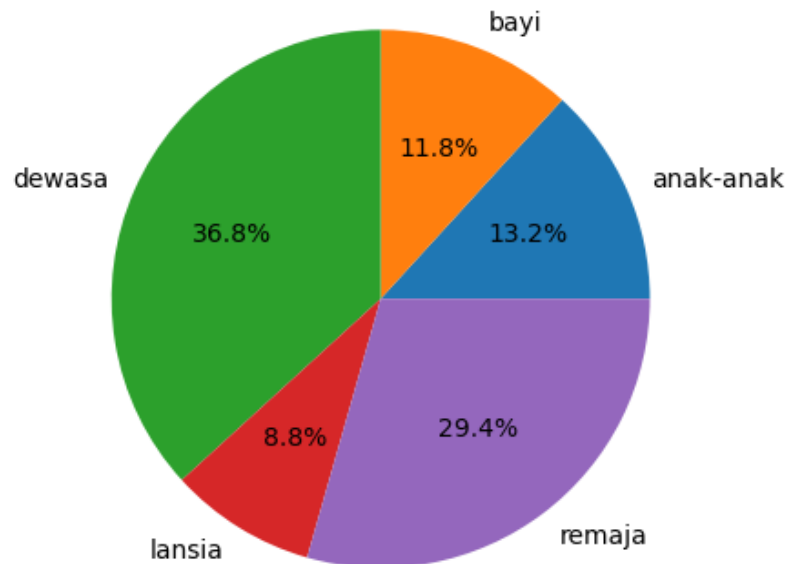
Dengan menggunakan fitur *danceability* dan *tempo* dengan 10 *n_cluster*, setelah dilakukan *clustering* dan berhasil mendapatkan jenis suasana hati *energetic* yang berada pada *cluster* 9 dengan nilai *danceability* yang relatif tinggi dan nilai *tempo* yang tinggi, maka suasana hati *energetic* berhasil di dapatkan dengan skor *Silhouette Coefficient* 0.547. dengan ini, terbukti kalau *cluster* yang dihasilkan oleh *kmeans* cukup baik.

4. Kesimpulan

Dalam analisis ini, yang fokus pada suasana hati energik, sebagian besar data yang dikumpulkan memang mencerminkan suasana hati tersebut, dengan lagu-lagu yang memiliki karakteristik *danceability* tinggi dan *tempo* yang cepat. Namun, dalam kumpulan data yang sama, didapatkan adanya campuran dari lagu-lagu dengan suasana hati yang lain, seperti '*Calm*' dan '*Sad*'. Penemuan ini menunjukkan bahwa, meskipun metodologi *clustering* seperti *K-Means* efektif untuk mengelompokkan lagu-lagu dengan ciri-ciri energik yang jelas, masih terdapat beberapa lagu dengan karakter yang berbeda yang tergabung dalam *cluster* yang sama.

Berikut merupakan hasil diagram berdasarkan jenis suasana hati dan umur, bisa dilihat pada gambar 4.104.

Distribusi Suasana Hati *Energic* Berdasarkan Kategori Umur



Gambar 4.104 Distribusi Suasana Hati *Energic* Berdasarkan Kategori Umur

Gambar diatas merupakan diagram atau *pie chart* dari hasil distribusi suasana hati berdasarkan kategori umur, dan dapat dilihat kalau pada suasana hati *energic* didominasi oleh musik dari kategori umur dewasa dengan 36.8%, dan kategori umur remaja 29.4%, dan anak-anak yaitu 13.2%, dan bayi 11,8% dan terakhir yaitu

4.9.3. Analisis Suasana Hati *Sad*

Suasana hati *sad* dalam musik sering diidentifikasi melalui lagu-lagu yang mengekspresikan kesedihan, keintiman, atau melankoli. Fitur *acousticness* dan *valence* sering digunakan untuk mengidentifikasi lagu dengan suasana hati sedih ini. Berikut adalah penjelasan mengenai penggunaan kedua fitur tersebut:

1. *Acousticness*

Acousticness mengukur seberapa akustik sebuah lagu. Nilai yang lebih tinggi pada *acousticness* menunjukkan bahwa lagu tersebut cenderung lebih akustik, yang sering kali berkorelasi dengan suara yang lebih lembut, dan sering dianggap lebih emosional atau

sedih. Lagu dengan skor *acousticness* tinggi sering memiliki instrumen yang lebih tradisional seperti piano, gitar akustik, atau biola, yang cenderung menimbulkan nuansa yang lebih melankolis atau introspektif, seringkali dihubungkan dengan perasaan sedih atau reflektif.

2. *Valence*

Valence mengukur kepositifan musik, di mana skor yang lebih tinggi menunjukkan musik yang lebih positif dan bahagia, sementara skor yang lebih rendah menandakan lagu yang lebih negatif atau sedih. Lagu dengan *valence* rendah cenderung memiliki tema, lirik, dan harmoni yang mengkonvey emosi negatif atau kesedihan. *Valence* rendah sering mengindikasikan lagu-lagu yang mampu menimbulkan empati atau rasa sedih dalam diri pendengar.

3. Analisis

Menggunakan fitur *acousticness* dan *valence* dengan *n_cluster* 4, setelah dilakukan *clustering*, jenis suasana hati *sad* berhasil diidentifikasi dalam *cluster* yang memiliki nilai *acousticness* tinggi dan *valence* rendah. Misalnya, *cluster* yang berhasil diidentifikasi sebagai representasi suasana hati sedih mungkin adalah *cluster* dengan skor *acousticness* yang tinggi dan *valence* yang rendah, mencerminkan ciri-ciri lagu sedih. Namun skor *Silhouette Coefficient* yang dihasilkan cukup rendah yaitu 0.431. dengan ini, terbukti kalau *cluster* yang dihasilkan oleh *kmeans* lemah.

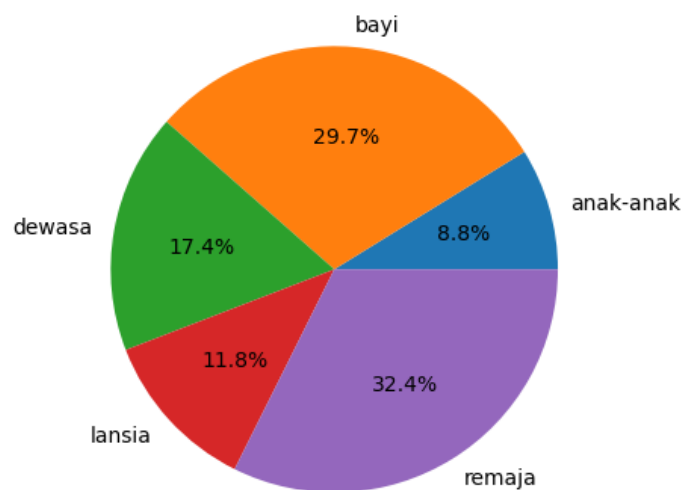
4. Kesimpulan

Dalam analisis ini, yang fokus pada suasana hati *sad*, mayoritas data yang dikumpulkan menunjukkan karakteristik musik yang sesuai, seperti tingkat *acousticness* yang tinggi dan *valence* yang rendah. Penggunaan metode *clustering* seperti *K-Means* terbukti efektif dalam mengidentifikasi lagu-lagu yang mencerminkan emosi kesedihan.

Meskipun demikian, kumpulan data juga mengandung beberapa lagu dengan suasana hati yang berbeda, seperti '*happy*' dan '*calm*'. Walau demikian *K-Means* tetap dapat mengelompokkan lagu dengan suasana hati yang dicari, walaupun terdapat pula lagu-lagu dengan karakter emosional yang beragam dalam *cluster* yang sama.

Berikut merupakan hasil diagram berdasarkan jenis suasana hati dan umur, bisa dilihat pada gambar 4.105.

Distribusi Suasana Hati Sad Berdasarkan Kategori Umur



Gambar 4.105 Distribusi Suasana Hati Sad Berdasarkan Kategori Umur

Gambar diatas merupakan diagram atau *pie chart* dari hasil distribusi suasana hati berdasarkan kategori umur, dan dapat dilihat kalau pada suasana hati *sad* didominasi oleh musik dari kategori umur remaja dengan 32.4%, dan selanjutnya yaitu bayi dengan 29.7%, dan dewasa dengan 17.4%, selanjutnya 11.8%, dan terakhir yaitu anak-anak yaitu 8.8%.

4.9.4. Analisis Suasana Hati *Happy*

Suasana hati *happy* dalam musik biasanya dicirikan oleh lagu-lagu yang membangkitkan perasaan positif, kegembiraan, dan semangat. Dua fitur yang sering digunakan untuk mengidentifikasi lagu dengan suasana hati ini adalah "*loudness*" dan "*tempo*". Berikut adalah penjelasan mengenai penggunaan kedua fitur tersebut untuk

menentukan suasana hati *happy*:

1. *Loudness*

Loudness mengukur intensitas suara lagu, biasanya diukur dalam decibel (dB). Lagu-lagu dengan *loudness* tinggi sering dianggap lebih dinamis dan berenergi. *Loudness* yang lebih tinggi sering kali berkorelasi dengan lagu yang lebih energik dan semangat, menciptakan suasana hati yang *happy* atau gembira. Intensitas suara yang tinggi dapat membangkitkan perasaan kegembiraan dan aktivitas yang sesuai dengan suasana hati positif.

2. *Tempo*

Tempo, diukur dalam *beats per minute* (BPM), menentukan kecepatan lagu. Lagu dengan *tempo* cepat sering kali dihubungkan dengan perasaan bahagia dan bersemangat karena ritme yang cepat dapat meningkatkan energi dan perasaan positif pada pendengar.

3. Analisis

Menggunakan *loudness* dan *tempo* dengan 3 *n_cluster*, *clustering* dilakukan untuk mengidentifikasi lagu yang menciptakan suasana hati *happy*. Misalnya, *cluster* yang memiliki nilai *loudness* tinggi dan *tempo* cepat dapat dianggap mewakili lagu-lagu yang *happy*. Dan skor *Silhouette Coefficient* yang dihasilkan yaitu 0.514. dengan ini, terbukti bahwa *cluster* yang dihasilkan oleh *kmeans* cukup baik.

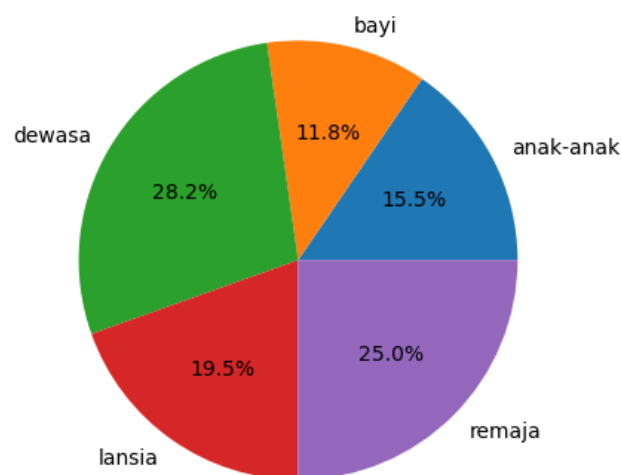
4. Kesimpulan

Penemuan dari analisis ini menunjukkan bahwa *loudness* tinggi dan *tempo* cepat adalah indikator yang efektif untuk mengidentifikasi lagu dengan suasana hati *happy* dalam musik. *Clustering* menggunakan *kmeans* berdasarkan kedua fitur ini telah terbukti efektif dalam mengelompokkan lagu-lagu yang menggambarkan kegembiraan. Hal ini menegaskan

pentingnya kedua fitur ini dalam sistem rekomendasi musik untuk memilih lagu yang sesuai dengan suasana hati bahagia yang diinginkan pengguna. Namun pada analisis manual didapatkan terdapat beberapa musik yang bercampur dengan suasana hati lain seperti *sad*, ini menandakan bahwa algoritma tidak mampu mengelompokkan lagu dengan akurat.

Berikut merupakan hasil diagram berdasarkan jenis suasana hati dan umur, bisa dilihat pada gambar 4.106.

Distribusi Suasana Hati Happy Berdasarkan Kategori Umur



Gambar 4.106 Distribusi Suasana Hati *Happy* Berdasarkan Kategori Umur

Gambar diatas merupakan diagram atau *pie chart* dari hasil distribusi suasana hati berdasarkan kategori umur, dan dapat dilihat kalau pada suasana hati *happy* didominasi oleh musik dari kategori umur dewasa dengan 28.2%, selanjutnya remaja dengan 25.0%, selanjutnya lansia dengan 19.5% dan anak-anak dengan 15.5%, dan yang terakhir yaitu bayi dengan 11.8%.

4.9.5. Analisis Suasana Hati *Angry*

1. *Energy*

Energy dalam analisis musik mengacu pada tingkat intensitas dan aktivitas sebuah lagu, biasanya ditandai dengan dinamika yang kuat, *tempo* yang cepat, dan suara yang

keras. Lagu dengan *energy* tinggi seringkali mengandung elemen yang lebih dinamis dan agresif, seperti *beat* yang kuat, distorsi gitar yang berat, atau ritme yang cepat, yang dapat membangkitkan perasaan marah atau agitasi.

2. *Loudness*

Loudness mengukur intensitas keseluruhan suara lagu, dengan nilai yang lebih tinggi menunjukkan volume yang lebih keras. Volume yang lebih keras atau *loudness* yang tinggi sering dikaitkan dengan ekspresi emosi yang kuat, seperti kemarahan. Suara yang keras dapat meningkatkan sensasi kekuatan dan kemarahan dalam musik.

3. Analisis

Menggunakan *energy* dan *loudness* dengan 3 *n_cluster*, *clustering* dilakukan untuk mengidentifikasi lagu yang menciptakan suasana hati *angry*. Klaster yang memiliki nilai *energy* tinggi dan *loudness* tinggi dianggap mewakili lagu-lagu yang *angry*. Skor *Silhouette Coefficient* sebesar 0.547 menunjukkan bahwa klasterisasi cukup berhasil dalam mengelompokkan lagu dengan ciri-ciri suasana hati *angry*.

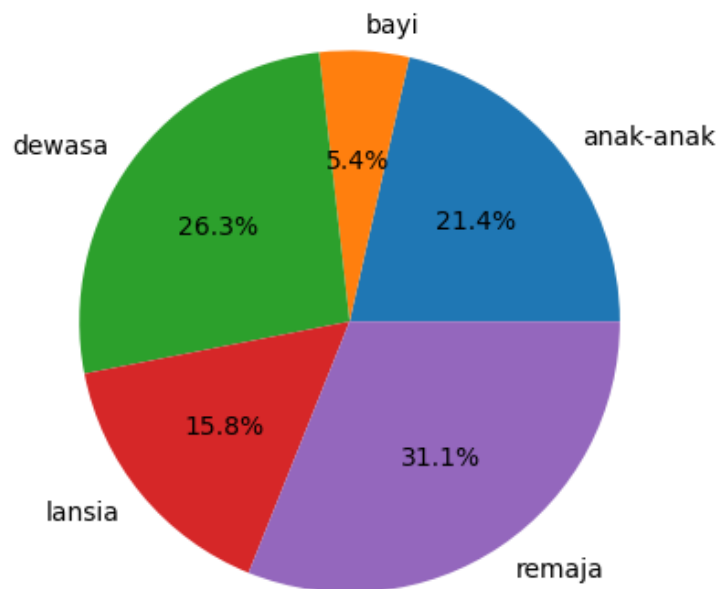
4. Kesimpulan

Dalam analisis ini, yang fokus pada suasana hati *angry*, mayoritas data yang dikumpulkan menunjukkan karakteristik musik yang sesuai, seperti tingkat *energy* tinggi dan *loudness* tinggi. Penggunaan metode *clustering* seperti *K-Means* terbukti efektif dalam mengelompokkan lagu-lagu. Meskipun demikian, kumpulan data juga mengandung beberapa lagu dengan suasana hati yang berbeda, seperti '*Calm*' dan '*Sad*'. Hasil ini menegaskan bahwa, *K-Means* kurang efektif dalam mengidentifikasi lagu dengan ciri kesedihan yang sama.

Berikut merupakan hasil diagram berdasarkan jenis suasana hati dan umur, bisa

dilihat pada gambar 4.107.

Distribusi Suasana Hati Angry Berdasarkan Kategori Umur



Gambar 4.107 Distribusi Suasana Hati *Angry* Berdasarkan Kategori Umur

Gambar diatas merupakan diagram atau *pie chart* dari hasil distribusi suasana hati berdasarkan kategori umur, dan dapat dilihat kalau pada suasana hati *angry* didominasi oleh musik dari kategori umur remaja dengan 31.1%, dan selanjutnya yaitu dewasa dengan 26.3%, dan anak-anak dengan 21.4%, selanjutnya lansia dengan 15.8%, dan terakhir yaitu bayi 5.4%.

BAB V

PENUTUP

5.1. Kesimpulan

Dari penelitian yang dilakukan, terdapat beberapa kesimpulan yang dapat diambil dari penelitian ini:

1. Metode *K-Means* berhasil mengelompokkan musik ke dalam beberapa *cluster* dengan menggunakan trek musik tertentu, dan berhasil mendapatkan jenis suasana hati yang di cari.
2. Fitur fitur musik yang berhasil mendapatkan jenis suasana hati yaitu '*instrumentalness*' dan '*tempo*' berhasil mendapatkan jenis suasana hati *calm* dengan 5 *n_cluster*, dan fitur musik *danceability*; dan '*tempo*' berhasil mendapatkan suasana hati *energetic* dengan 10 *n_cluster*, dan untuk fitur musik '*acousticness*' dan '*valence*' berhasil mendapatkan suasana hati *sad* dengan 4 *n_cluster*, dan untuk suasana hati *happy* didapatkan dengan fitur '*loudness*' dan '*tempo*' dengan 3 *n_cluster* dan untuk suasana hati *angry* berhasil didapatkan dengan menggunakan fitur '*energy*' dan '*loudness*' dengan 5 *n-cluster*.
3. Penggunaan *Silhouette Coefficient* menunjukkan bahwa algoritma *K-Means* dapat mengelompokkan trek musik dengan cukup baik, skor *Silhouette Coefficient* yang diperoleh menunjukkan bahwa mayoritas trek musik berada dalam *cluster* yang cukup sesuai. Dari hasil pengujian, skor *Silhouette* yang paling tinggi berada pada hasil *cluster* suasana hati *calm*, dengan skor *Silhouette* nya 0.567, dan untuk skor *Silhouette* dari hasil *cluster energetic*, berhasil mendapatkan skor *Silhouette* 0.547, dan untuk skor *Silhouette* dari hasil *cluster sad*, berhasil mendapatkan skor *Silhouette*

0.431, dan untuk skor *Silhouette* dari hasil *cluster happy*, berhasil mendapatkan skor *Silhouette* 0.514, sedangkan untuk skor *Silhouette* dari hasil *cluster angry*, berhasil mendapatkan skor *Silhouette* 0.544.

4. Algoritma *K-Means* kurang efektif dalam mengelompokkan trek musik ke dalam *cluster* yang merefleksikan kombinasi *mood*, dan terdapat beberapa data lain yang tidak memiliki hubungan dengan suasana hati yang dicari, ini menunjukkan bahwa algoritma tidak dapat mengidentifikasi pola dalam data dengan sempurna, karena terdapat data lain yang tidak berhubungan namun tetap masuk ke dalam hasil *cluster*.
5. Metode *kmeans* tampaknya efektif dalam mengelompokkan dan mengidentifikasi berdasarkan fitur-fitur spesifik seperti *instrumentalness*, *tempo*, *danceability*, *acousticness*, *valence loudness*, dan *energy*. Skor *Silhouette* yang didapatkan juga menunjukkan kualitas *cluster* yang baik berdasarkan interpretasi.

5.2. Saran

Adapun saran yang dapat diberikan untuk mengembangkan penelitian ini adalah:

1. Penelitian selanjutnya dapat menggunakan lebih banyak variabel, seperti kondisi cuaca, aktifitas saat mendengarkan musik dan genre musik, untuk meningkatkan akurasi dan relevansi rekomendasi musik.
2. Untuk penelitian selanjutnya disarankan untuk menggunakan *dataset* yang lebih besar dan banyak, hal ini tidak hanya dapat meningkatkan akurasi tetapi juga membantu sistem dalam mengidentifikasi dan mengelompokkan trek musik yang lebih halus dan spesifik.
3. Dan menggunakan metode yang berbeda dengan penelitian ini agar dapat mengetahui hasil dan kualitas dari masing-masing metode.

4. Menggunakan metode analisis yang lebih mendalam pada hasil *cluster* agar bisa menghasilkan lagu yang lebih akurat pada setiap lagu yang dihasilkan.

DAFTAR PUSTAKA

- Abdul 'alim, M., Kartika Dewi, R., & Brata, K. C. 2021. Pengembangan Aplikasi Rekomendasi Musik Berdasarkan Emosi Pengguna Pada *Platform Android*. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, e-ISSN:2548-964X, Vol. 1 *Issue*.1 Januari, 2010
- Alfarizi, M. R. S., Al-farish, M. Z., Taufiqurrahman, M., Ardiansah, G., & Elgar, M. 2023. Penggunaan Python Sebagai Bahasa Pemrograman untuk *Machine Learning* dan *Deep Learning*. Karya Ilmiah Mahasiswa Bertauhid, e-ISSN:2963-590X, Vol. 2 *issue*.1 Juni, 2023
- Anggoro, M. V., & Izzatillah, M. 2022. Sistem Rekomendasi Musik dengan Metode *Collaborative Filtering* Berbasis *Android*. Satuan Tulisan Riset Dan Inovasi Teknologi (STRING), e-ISSN:2549-2837, Vol.7 *Issue*.1 Agustus, 2022
- Badriyah, T., Fernando, R., & Syarif, I. 2018. Sistem Rekomendasi *Content Based Filtering* Menggunakan Algoritma *Apriori*. Konferensi Nasional Sistem Informasi, Vol. 1 *Issue*.1 Maret, 2018
- Brata, I. P. B. W., & Darmawan, I. D. M. B. A. 2021. *Mood Classification of Balinese Songs with the K-Means Clustering Method Based on the Audio-Content Feature*. Jurnal Elektronik Ilmu Komputer Udayana (JELIKU), e-ISSN:2654-5101, Vol. 9 *Issue*.3 Februari, 2021
- Buslim, N., & Iswara, R. P. 2019. Pengembangan Algoritma *Unsupervised Learning Technique* Pada *Big Data Analysis* di Media Sosial sebagai media promosi *Online* Bagi Masyarakat. Jurnal Teknik Informatika, Vol. 12 *Issue*.1 April, 2019
- Hermanto, K., Salim, D., Wu, B., Salim, O. R., & Gunadi, R. B. 2023. Penggunaan Python Untuk Menganalisis Pola Penyebaran Covid-19 Di Masa Pandemi. *Journal of Student Development Information Sistem* (JoSDIS), e-ISSN:2774-7948, Vol. 3 *Issue*.2 Juli, 2023
- Kesehatan, K. (2024). KELOMPOK USIA. <https://ayosehat.kemkes.go.id/kategori-usia/dewasa>
- Lesmana, A., & Setiawan, E. B. 2019. Perancangan Aplikasi Perekomendasi Lagu Berdasarkan *Mood* Dan Aktivitas Pengguna Memanfaatkan *Wearable Device* Dan

Spotify Api. *Elibrary* Unikom.

- Makarim, dr. F. R. 2023. Emosi Manusia: Pengertian, Jenis, dan Fungsinya. <https://www.halodoc.com/artikel/emosi-manusia-pengertian-jenis-dan-fungsinya>
- Mondi, R. H., & Wijayanto, A. 2019. *Recommendation Sistem With Content-Based Filtering Method for Culinary Tourism in Mangan Application*. Jurnal Ilmiah Teknologi Dan Informasi, ISSN:2301-7201, Vol. 8 *Issue.2 December*, 2019
- Muhamad, Z. 2018. Sistem Informasi Geografis Trayek Angkutan Umum Di Kota Tasikmalaya Berbasis Web. Jurnal Teknik Informatika Atmaluhur, ISSN:2338-1477, Vol. 10 *Issue.2 juni*, 2018
- Musyarofah, U. L., Alima, S. N., & Kartika, D. S. Y. 2022. Klasifikasi Top 50 Spotify Tahun 2010-2019 Menggunakan Metode *K-Means Clustering*. Prosiding Seminar Nasional Teknologi Dan Sistem Informasi, ISSN:2828-786X, Vol. 2 *Issue.1 September*, 2022
- Nurhalimah, L., Hermanto, T. I., & Kaniawulan, I. 2022. Analisis Prediksi *Mood Genre Musik Pop* Menggunakan Algoritma *K-Means* dan C4.5.(Jurnal Riset Komputer), e-ISSN:2715-7393, Vol. 9 *Issue.4 Agustus*, 2022
- Putu, N., Merliana, E., & Santoso, A. J. 2015. Analisa Penentuan Jumlah *Cluster* Terbaik Pada Metode *K-Means clustering*. Prosiding Seminar Nasional Multi Disiplin Ilmu, 978–979.
- Rachmanda, F., Kharisma, A. P., & Wardhono, W. S. 2020. Pengembangan Aplikasi Penyesuai *Playlist* Musik dengan Konsep *Geofencing* menggunakan Spotify API Berbasis *Android*. Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer, e-ISSN:2548-964X, Vol. 4 *Issue.10 Oktober*, 2020
- Rahmawati, L., Widya Sihwi, S., & Suryani, E. 2016. Analisa *Clustering* Menggunakan Metode *K-Means* Dan *Hierarchical Clustering* (Studi Kasus : Dokumen Skripsi Jurusan Kimia, Fmipa, Universitas Sebelas Maret). Jurnal Teknologi & Informasi, Vol. 3 *Issue.2*, 2016
- Rohmah Zaidah, A., Indira Septiarani, C., Sholikhatun Nisa, M., Yusuf, A., & Wahyudi, N. 2021. Komparasi Algoritma *K-Means*, *K-Medoid*, *Agglomerative Clustering* Terhadap Genre Spotify. Jurnal Ilmiah Ilmu Komputer, ISSN:2503-3832, Vol. 7 *Issue.1 April*, 2021
- Shoolihah, A.-M., Furqon, M. T., & Widodo, A. W. 2017. Implementasi Metode *Improved K-*

- Means* untuk Mengelompokkan Titik Panas Bumi. Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer, e-ISSN:2548-964X, Vol. 1 Issue.11 November, 2017
- Struyf, A., Hubert, M., & Rousseeuw, P. J. 1996. *Clustering in an object-oriented environment. Journal of Statistical Software, Department Of Mathematics And Computer Science*, Vol. 1 Issue.4 1996
- support.spotify.com. 2020. Kualitas Audio Spotify, <https://support.spotify.com/id-id/article/what-is-spotify/>
- Sutejo, D., Agus Pranoto, Y., & Zulfia Zahro', H. 2020. Sistem Informasi Geografis Pengelompokan Tingkat Kriminalitas Kota Malang Menggunakan Metode *K-Means*. Jurnal Mahasiswa Teknik Informatika, Vol. 4 Issue.1 Maret, 2020
- Widi Hastomo, Nur Aini, Adhitio Satyo Bayangkari Karno, & L.M. Rasdi Rere. 2022. Metode Pembelajaran Mesin untuk Memprediksi Emisi *Manure Management*. Jurnal Nasional Teknik Elektro Dan Teknologi Informasi, Vol. 11 Issue.2 Mei, 2022



UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

DAFTAR PERBAIKAN SEMINAR HASIL SKRIPSI

Dengan ini dinyatakan bahwa pada

Hari / tanggal : JUMAT, 08 MARET 2024

Pukul : 10:30 - 12:30

Tempat : RUANG SIDANG

telah berlangsung Seminar Hasil Skripsi dengan Peserta:

Nama Mahasiswa : WAHYUDIN NURDIN

NPM : 07352011027

Judul : SISTEM REKOMENDASI MUSIK PADA SPOTIFY BERDASARKAN
SUASANA HATI DAN UMUR MENGGUNAKAN METODE K-MEANS

dinyatakan HARUS menyelesaikan perbaikan, yaitu:

- Pahami source code

Handwritten signature and date: Ace 29-3-2024

Dosen Pembimbing I,

MUHAMMAD FHADLI, S.Kom., M.Sc.
NIP. 199611232023211012



UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

DAFTAR PERBAIKAN SEMINAR HASIL SKRIPSI

Dengan ini dinyatakan bahwa pada

Hari / tanggal : JUMAT, 08 MARET 2024
Pukul : 10:30 - 12:30
Tempat : RUANG SIDANG

telah berlangsung Seminar Hasil Skripsi dengan Peserta:

Nama Mahasiswa : WAHYUDIN NURDIN
NPM : 07352011027
Judul : SISTEM REKOMENDASI MUSIK PADA SPOTIFY BERDASARKAN
SUASANA HATI DAN UMUR MENGGUNAKAN METODE K-MEANS

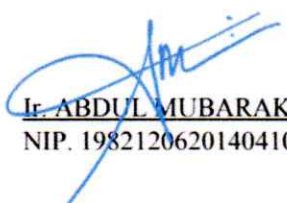
dinyatakan HARUS menyelesaikan perbaikan, yaitu:

①. Perbaikan secepatnya.

②. Perbaikan sesuai dengan masukan penguji

ACE
23/4/2024
Perbaikan Hasil Skripsi
Abdul Mubarak

Dosen Pembimbing II,


Ir. ABDUL MUBARAK, S.Kom., M.T., IPM
NIP. 198212062014041002



UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

DAFTAR PERBAIKAN SEMINAR HASIL SKRIPSI

Dengan ini dinyatakan bahwa pada

Hari / tanggal : JUMAT, 08 MARET 2024
Pukul : 10:30 - 12:30
Tempat : RUANG SIDANG

telah berlangsung Seminar Hasil Skripsi dengan Peserta:

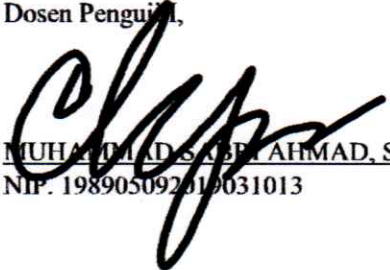
Nama Mahasiswa : WAHYUDIN NURDIN
NPM : 07352011027
Judul : SISTEM REKOMENDASI MUSIK PADA SPOTIFY BERDASARKAN
SUASANA HATI DAN UMUR MENGGUNAKAN METODE K-MEANS

dinyatakan HARUS menyelesaikan perbaikan, yaitu:

Bawa aplikasinya ke saya ketika konsultasi perbaikan.wajib di hosting

0 24/04-2024

Dosen Pengui,


MUHAMMAD SABRI AHMAD, S.Kom., M.Kom.
NIP. 198905092010031013



UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

DAFTAR PERBAIKAN SEMINAR HASIL SKRIPSI

Dengan ini dinyatakan bahwa pada

Hari / tanggal : JUMAT, 08 MARET 2024

Pukul : 10:30 - 12:30

Tempat : RUANG SIDANG

telah berlangsung Seminar Hasil Skripsi dengan Peserta:

Nama Mahasiswa : WAHYUDIN NURDIN

NPM : 07352011027

Judul : SISTEM REKOMENDASI MUSIK PADA SPOTIFY BERDASARKAN
SUASANA HATI DAN UMUR MENGGUNAKAN METODE K-MEANS

dinyatakan HARUS menyelesaikan perbaikan, yaitu:

Data di sinkronisasi dengan survei.
Referensi data ada umur harus jelas

25-04-2024

OK Aceh

Dosen Penguji II,

SAIFUL Do. ABDULLAH, S.T., M.T.
NIDN. 0018029002



UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

DAFTAR PERBAIKAN SEMINAR HASIL SKRIPSI

Dengan ini dinyatakan bahwa pada

Hari / tanggal : JUMAT, 08 MARET 2024

Pukul : 10:30 - 12:30

Tempat : RUANG SIDANG

telah berlangsung Seminar Hasil Skripsi dengan Peserta:

Nama Mahasiswa : WAHYUDIN NURDIN

NPM : 07352011027

Judul : SISTEM REKOMENDASI MUSIK PADA SPOTIFY BERDASARKAN
SUASANA HATI DAN UMUR MENGGUNAKAN METODE K-MEANS

dinyatakan HARUS menyelesaikan perbaikan, yaitu:

1. perbaiki format penulisan dengan teliti
2. klasifikasi dan rekomendasi masih kurang, tambahkan untuk lebih lengkap (lihat catatan)
3. perhatikan tabel data dan kriteria dengan benar, termasuk jumlah lagu dan buat playlist
4. perbaiki kesimpulan dan lengkapi analisis yang tepat
5. tambahkan diagram hasil akhir
6. catat ujian dibawa saat asistensi

*Ace Perbaiki Hasil
2 atau lebih*

Dosen Penguji III,

Achmad Fuad
ACHMAD FUAD, S.T., M.T.
NIP. 197606182005011001



UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

DAFTAR PERBAIKAN UJIAN SKRIPSI/TUTUP

Dengan ini dinyatakan bahwa pada

Hari / tanggal : JUMAT, 26 APRIL 2024

Pukul : 07:30 - 09:00

Tempat : RUANG SIDANG

telah berlangsung Ujian Skripsi/Tutup dengan Peserta:

Nama Mahasiswa : WAHYUDIN NURDIN

NPM : 07352011027

Judul : SISTEM REKOMENDASI MUSIK SPOTIFY BERDASARKAN
SUASANA HATI DAN UMUR MENGGUNAKAN METODE K-
MEANS

dinyatakan IIARUS menyelesaikan perbaikan, yaitu:

- Pahami source code

Ace
18-6-2019

Dosen Pembimbing I,

MUHAMMAD FHADLI, S.Kom., M.Sc.
NIP. 199611232023211012



UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

DAFTAR PERBAIKAN UJIAN SKRIPSI/TUTUP

Dengan ini dinyatakan bahwa pada

Hari / tanggal : JUMAT, 26 APRIL 2024

Pukul : 07:30 - 09:00

Tempat : RUANG SIDANG

telah berlangsung Ujian Skripsi/Tutup dengan Peserta:

Nama Mahasiswa : WAHYUDIN NURDIN

NPM : 07352011027

Judul : SISTEM REKOMENDASI MUSIK SPOTIFY BERDASARKAN
SUASANA HATI DAN UMUR MENGGUNAKAN METODE K-
MEANS

dinyatakan HARUS menyelesaikan perbaikan, yaitu:

- ①. Perbaikan Surat
- ②. Jelaskan Detail Hasil Penelitian di bab 4
- ③. Jelaskan di bagian yg kekurangan hasil penelitian untuk mengaji Rekomendasi penelitian selanjutnya.

24/6/2024

ACE Revisi Skripsi

Abdul Mubarak

Dosen Pembimbing II,

Ir. ABDUL MUBARAK, S.Kom., M.T., IPM
NIP. 198212062014041002



UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

DAFTAR PERBAIKAN UJIAN SKRIPSI/TUTUP

Dengan ini dinyatakan bahwa pada

Hari / tanggal : JUMAT, 26 APRIL 2024

Pukul : 07:30 - 09:00

Tempat : RUANG SIDANG

telah berlangsung Ujian Skripsi/Tutup dengan Peserta:

Nama Mahasiswa : WAHYUDIN NURDIN

NPM : 07352011027

Judul : SISTEM REKOMENDASI MUSIK SPOTIFY BERDASARKAN
SUASANA HATI DAN UMUR MENGGUNAKAN METODE K-
MEANS

dinyatakan HARUS menyelesaikan perbaikan, yaitu:

Sudah cukup baik, sisanya tolong perbaiki sesuai masukan yg saya berikan saat ujian skripsi kemarin

Handwritten signature and date: 17/5-2024

Dosen Penguji I,

Handwritten signature of Muhammad Sabri Ahmad

MUHAMMAD SABRI AHMAD, S.Kom., M.Kom.
NIP. 198905092019030013



UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

DAFTAR PERBAIKAN UJIAN SKRIPSI/TUTUP

Dengan ini dinyatakan bahwa pada

Hari / tanggal : JUMAT, 26 APRIL 2024

Pukul : 07:30 - 09:00

Tempat : RUANG SIDANG

telah berlangsung Ujian Skripsi/Tutup dengan Peserta:

Nama Mahasiswa : WAHYUDIN NURDIN

NPM : 07352011027

Judul : SISTEM REKOMENDASI MUSIK SPOTIFY BERDASARKAN
SUASANA HATI DAN UMUR MENGGUNAKAN METODE K-
MEANS

dinyatakan HARUS menyelesaikan perbaikan, yaitu:

Ditambahkan referensi tentang suasana hati dengan lagu.
Perhatikan penulisan

08.05.2024

OK Aceh

Dosen Penguji II,

SAIFUL Do. ABDULLAH, S.T., M.T.
NIDN. 0018029002



UNIVERSITAS KHAIRUN
FAKULTAS TEKNIK
PROGRAM STUDI INFORMATIKA

DAFTAR PERBAIKAN UJIAN SKRIPSI/TUTUP

Dengan ini dinyatakan bahwa pada

Hari / tanggal : JUMAT, 26 APRIL 2024

Pukul : 07:30 - 09:00

Tempat : RUANG SIDANG

telah berlangsung Ujian Skripsi/Tutup dengan Peserta:

Nama Mahasiswa : WAHYUDIN NURDIN

NPM : 07352011027

Judul : SISTEM REKOMENDASI MUSIK SPOTIFY BERDASARKAN
SUASANA HATI DAN UMUR MENGGUNAKAN METODE K-
MEANS

dinyatakan HARUS menyelesaikan perbaikan, yaitu:

1. catatan/masukan saat ujian skripsi ditindaklanjuti dan diasistensi! ✓
2. hasil akhir rekomendasi dianalisis dengan tepat dan disimpulkan dengan benar! - ✓
3. perhatikan semua referensi dan daftar pustaka - ✓
4. tambahkan daftar setiap cluster, daftar lagu dipisahkan dan rumus algoritma perlu dicek kembali - ✓
5. hal lain akan diperhatikan saat asistensi perbaikan! ?

Ace Pgs 28/06/2024

Dosen Penguji III,

ACHMAD FUAD, S.T., M.T.
NIP. 197606182005011001



Nama Mahasiswa : Wahyudin Nurdin
NIM : 07352011027
Dosen Pembimbing I : Muhammad Fhadli, S.Kom., M.Sc.
Judul : Sistem Rekomendasi Musik Pada Spotify Berdasarkan Suasana Hati Dan Umur Menggunakan Metode K-Means

[illegible]



KARTU BIMBINGAN HASIL

Nama Mahasiswa : Wahyudin Nurdin
NIM : 07352011027
Dosen Pembimbing II : Ir. Abdul Mubarak, S.Kom., M.T., IPM.
Judul : Sistem Rekomendasi Musik Pada Spotify Berdasarkan Suasana Hati Dan Umur Menggunakan Metode K-Means

NO	Tanggal	Uraian	Paraf
		- Cari referensi reser dari Depkes / WHO	
		- Jelaskan cara pengambil data menggunakan API	
		- Preprocessing Data	
		- kategori umur? referensi?	
		- Data musik hasil berdasarkan kategori usia dan/gore musik yang ada pada spotify	
		Alto Sumner HZM	