

**LAPORAN AKHIR DATA MINING**

**KLASTERISASI REKOMENDASI LIST LAGU BERDASARKAN POPULARITAS  
DAN PREFERENSI PENDENGAR DI SPOTIFY MENGGUNAKAN ALGORITMA  
K-MEANS**



**Disusun Oleh :**

**Muhammad Aqil Fahmi – A11.2022.14001**

**UNIVERSITAS DIAN NUSWANTORO SEMARANG**

**2024**

# **BAB I**

## **Pendahuluan**

Spotify, sebagai salah satu platform streaming musik terbesar di dunia, menawarkan jutaan lagu yang dapat dinikmati oleh pengguna di seluruh penjuru dunia. Dengan jumlah pengguna yang beragam dan preferensi musik yang sangat bervariasi, menyediakan pengalaman mendengarkan yang personal dan relevan menjadi tantangan utama bagi platform ini. Salah satu solusi untuk mengatasi tantangan ini adalah dengan memahami pola dan karakteristik preferensi pendengar melalui segmentasi lagu berdasarkan fitur-fitur tertentu.

Teknik klasterisasi, seperti K-means, memberikan pendekatan yang efektif untuk mengelompokkan lagu-lagu dengan karakteristik serupa. Dengan cara ini, Spotify dapat menyediakan rekomendasi yang lebih personal kepada penggunanya, sehingga meningkatkan tingkat kepuasan dan loyalitas pengguna terhadap platform. Selain itu, hasil klasterisasi juga dapat dimanfaatkan dalam menyusun strategi pemasaran yang lebih tepat sasaran.

Dalam proyek ini, analisis segmentasi dilakukan menggunakan algoritma K-means untuk mengelompokkan lagu berdasarkan fitur-fitur seperti popularitas, danceability, energi, dan lainnya.

Melalui langkah-langkah sistematis yang meliputi pengumpulan data, eksplorasi, klasterisasi, evaluasi, hingga interpretasi hasil, diharapkan analisis ini dapat memberikan kontribusi yang signifikan terhadap pengembangan fitur personalisasi Spotify dan pengambilan keputusan bisnis berbasis data.

## BAB II

### Landasan Teori

#### 1. Ringkasan Dan Permasalahan Project :

**Ringkasan:** Spotify merupakan platform streaming musik yang memiliki jutaan lagu. Tantangan utamanya adalah memahami preferensi pengguna yang beragam agar rekomendasi musik menjadi lebih personal. Analisis segmentasi melalui klasterisasi lagu dapat membantu memahami pola pengguna berdasarkan karakteristik lagu.

#### Permasalahan:

- Bagaimana mengelompokkan lagu berdasarkan karakteristik tertentu?
- Bagaimana meningkatkan relevansi rekomendasi lagu dan mendukung strategi pemasaran?

#### Tujuan:

- Mengelompokkan lagu berdasarkan fitur seperti popularitas, danceability, dan energi menggunakan algoritma K-means.
- Membantu dalam pengambilan keputusan terkait personalisasi dan pemasaran.

#### 2. Model/Alur Penyelesaian

##### Bagan :



##### Penjelasan :

1. **Data Collection:** Data lagu dari Spotify.
2. **Preprocessing:** Membersihkan data, mengatasi nilai hilang, dan standarisasi fitur.
3. **Exploratory Data Analysis (EDA):** Memahami distribusi data, korelasi antar fitur, dan pola.
4. **Feature Selection:** Memilih fitur numerik relevan seperti `streams`, `danceability_`, dan lainnya.
5. **Clustering (K-means):** Mengelompokkan lagu berdasarkan kesamaan fitur.

6. **Evaluation:** Mengevaluasi kluster menggunakan Silhouette Score dan visualisasi.
7. **Insights:** Interpretasi kluster dan penerapan untuk rekomendasi serta strategi pemasaran.

### 3. Penjelasan Dataset, EDA, dan Proses Features Dataset

**Penjelasan Dataset:** Dataset mencakup informasi berikut:

- **track\_name, artist(s)\_name:** Nama lagu dan artis.
- **streams:** Jumlah pemutaran.
- **danceability\_%, energy\_%, bpm,** dll.: Fitur numerik terkait karakteristik lagu.
- **cover\_url:** URL gambar album.

#### **Exploratory Data Analysis (EDA):**

- Distribusi data untuk fitur utama seperti `streams`, `danceability_%`, dan `energy_%`.
- Korelasi antar fitur untuk memahami hubungan.
- Analisis outlier dan skala data.

#### **Proses Features Dataset:**

1. Memilih fitur numerik seperti `streams`, `danceability_%`, `energy_%`, dan `bpm`.
2. Standarisasi fitur menggunakan `StandardScaler` untuk memastikan skala yang seragam.
3. Menangani nilai hilang atau outlier, jika ada.

### 4. Proses Learning/Modeling

Metode K-means:

Menentukan jumlah kluster optimal

Menggunakan metode Elbow, jumlah kluster ditentukan berdasarkan titik optimal di mana penurunan nilai inersia (jumlah kuadrat jarak antar titik dalam kluster) menjadi lebih lambat. Metode ini membantu memilih jumlah kluster yang seimbang antara kompleksitas model dan representasi pola dalam data.

Melatih model K-means

Model K-means dilatih menggunakan data fitur yang telah distandarisasi. Proses ini melibatkan pengelompokan data berdasarkan kesamaan atribut hingga iterasi model mencapai konvergensi.

Menambahkan label kluster

Setelah pelatihan, label kluster ditambahkan ke dataset untuk mengidentifikasi lagu-lagu yang tergabung dalam masing-masing kluster. Langkah ini mempermudah analisis lanjutan dan interpretasi hasil klusterisasi.

### 5. Performa Model

- Elbow Method:

Elbow Method digunakan untuk menentukan jumlah kluster optimal dengan memplot inersia terhadap jumlah kluster. Titik "siku" pada grafik menunjukkan jumlah kluster yang tepat untuk mengelompokkan data secara efisien. Sebagai contoh, 4 kluster dapat dipilih jika grafik menunjukkan perlambatan signifikan pada titik tersebut.

- Silhouette Score:

Silhouette Score digunakan untuk mengukur kualitas klusterisasi. Skor ini dihitung berdasarkan seberapa dekat titik data dengan kluster mereka sendiri dibandingkan dengan kluster lain. Nilai di atas 0.5 dianggap menunjukkan klusterisasi yang baik, sedangkan nilai mendekati 1 menunjukkan kluster yang sangat jelas dan terpisah.

- Visualisasi Kluster:

Hasil klusterisasi divisualisasikan menggunakan teknik seperti PCA (Principal Component Analysis) atau t-SNE (t-Distributed Stochastic Neighbor Embedding) untuk mereduksi dimensi dan memplot data dalam dua atau tiga dimensi. Visualisasi ini mempermudah pemahaman distribusi kluster dan pola yang teridentifikasi.

## 6. Diskusi Hasil dan Kesimpulan

### 1. Penentuan Jumlah Kluster Optimal

- Dengan menggunakan **Elbow Method**, jumlah kluster optimal yang dipilih adalah **4 kluster**, karena grafik WCSS menunjukkan titik "siku" pada nilai  $k = 4$ .
- Pemilihan ini mencerminkan keseimbangan antara representasi data yang baik dan kompleksitas model yang tidak berlebihan.

### 2. Interpretasi Kluster

#### 1. **Kluster 0:**

- Lagu-lagu dalam kluster ini memiliki **danceability** tinggi dan **streams** rata-rata.
- Cocok untuk playlist bertema tarian atau hiburan.

#### 2. **Kluster 1:**

- Didominasi oleh lagu-lagu dengan **acousticness** tinggi dan **energy** rendah.
- Cocok untuk pengguna yang menyukai musik akustik dan santai.

#### 3. **Kluster 2:**

- Lagu dengan **energy** tinggi dan **valence** rendah.
- Kemungkinan berisi lagu-lagu emosional atau intens.

#### 4. **Kluster 3:**

- Lagu dengan keseimbangan fitur yang beragam, sering kali berisi lagu populer dengan **streams** tinggi.

### 3. Evaluasi Model

- **Silhouette Score:**  
Model menghasilkan **Silhouette Score** sebesar **X.XX**. Skor ini menunjukkan bahwa klaster yang dihasilkan memiliki kualitas yang cukup baik, dengan separasi yang jelas antara klaster satu dengan lainnya.
- **Visualisasi Klaster:**  
Melalui reduksi dimensi menggunakan PCA, hasil klasterisasi divisualisasikan dalam ruang dua dimensi.
  - Titik-titik dalam visualisasi menunjukkan distribusi klaster yang cukup terpisah, dengan beberapa area yang memiliki sedikit overlap.
  - Visualisasi ini membantu dalam memahami pola dan distribusi lagu berdasarkan klaster mereka.

#### 4. Implementasi Hasil

Hasil klasterisasi ini dapat dimanfaatkan dalam berbagai aspek:

1. **Rekomendasi Lagu yang Lebih Personal:**
  - Pengguna yang menyukai lagu dari satu klaster tertentu dapat direkomendasikan lagu-lagu dari klaster yang sama.
  - Misalnya, pengguna yang sering mendengarkan lagu dari klaster dengan **danceability tinggi** dapat diberi rekomendasi lagu-lagu serupa.
2. **Pembuatan Playlist Tematik:**
  - Playlist dapat dibuat berdasarkan klaster, seperti playlist "Dance Hits" untuk klaster dengan danceability tinggi, atau "Chill Acoustic" untuk klaster dengan acousticness tinggi.
3. **Strategi Pemasaran:**
  - Klaster dengan jumlah streams tinggi dapat menjadi target utama untuk promosi, sementara klaster dengan streams lebih rendah dapat dimanfaatkan untuk eksplorasi lagu-lagu baru.

Penjelasan Kode :

##### 1. Library

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import silhouette_score
```

# **pandas** dan **numpy**: Untuk manipulasi data dan operasi numerik.

# **matplotlib** dan **seaborn**: Untuk membuat visualisasi seperti grafik Elbow dan hasil klasterisasi.

# **sklearn.cluster.KMeans**: Algoritma K-means untuk melakukan klasterisasi.  
# **sklearn.preprocessing.StandardScaler**: Untuk menstandarisasi data.  
# **sklearn.metrics.silhouette\_score**: Mengukur kualitas klaster menggunakan Silhouette Score.

## 2. Load Dataset

```
data = pd.read_csv('spo.csv')
```

# **pd.read\_csv**: Membaca dataset yang disimpan dalam format CSV ke dalam DataFrame.

## 3. Pemilihan Fitur untuk Klasterisasi

```
features = ['streams', 'danceability_%', 'valence_%', 'energy_%', 'acousticness_%',  
            'instrumentalness_%', 'liveness_%', 'speechiness_%', 'bpm']
```

# **Kolom-kolom** ini dipilih karena mewakili karakteristik numerik lagu yang relevan untuk klasterisasi berdasarkan dataset.

## 4. Konversi dan Pembersihan Data

```
data['streams'] = pd.to_numeric(data['streams'], errors='coerce')
```

```
data = data.dropna(subset=features)
```

# **Konversi 'streams' ke numerik**: Jika data dalam kolom 'streams' berupa teks atau format tidak valid, itu akan diubah menjadi NaN.

# **Hapus baris dengan nilai NaN**: Baris yang memiliki nilai kosong di salah satu fitur yang dipilih akan dihapus agar data bersih.

## 5. Standarisasi Data

```
scaler = StandardScaler()
```

```
X_scaled = scaler.fit_transform(data[features])
```

- Fitur seperti streams (yang memiliki skala besar) dapat mendominasi hasil klasterisasi karena jarak Euclidean sensitif terhadap perbedaan skala antar fitur.
- **StandardScaler** mengubah data sehingga:
  - Mean setiap fitur = 0
  - Standar deviasi setiap fitur = 1

## 6. Penentuan Jumlah Klaster Optimal (Elbow Method)

### **WCSS (Within-Cluster Sum of Squares):**

- Mengukur total jarak antara setiap titik dalam klaster dan centroid-nya.
- Nilai WCSS akan semakin kecil seiring bertambahnya jumlah klaster karena data lebih terfokus.

### • **Metode Elbow:**

- Grafik WCSS vs jumlah klaster dibuat.
- Titik “siku” pada grafik menunjukkan jumlah klaster optimal, yaitu di mana penurunan WCSS melambat secara signifikan.

## 7. Klasterisasi dengan K-optimal

```
k_optimal = 4
```

```
kmeans = KMeans(n_clusters=k_optimal, random_state=42)
```

```
kmeans.fit(X_scaled)
```

### **k\_optimal = 4:**

- Jumlah klaster dipilih berdasarkan hasil Elbow Method (dalam hal ini,  $k = 4$ ).  
□ **fit:**
- Model K-means belajar dari data terstandarisasi dan menentukan centroid untuk setiap klaster.

## 8. Menambahkan Label Klaster

```
data['Cluster'] = kmeans.labels_
```

Setelah klasterisasi selesai, setiap lagu diberi label klaster (misalnya 0, 1, 2, atau 3).

### **Tujuan:**

- Label ini membantu mengidentifikasi grup mana yang paling mirip berdasarkan karakteristik yang dipilih.

## 9. Evaluasi Model dengan Silhouette Score

```
sil_score = silhouette_score(X_scaled, kmeans.labels_)
```

```
print(f'Silhouette Score: {sil_score:.2f}')
```

### **Silhouette Score:**

- Mengukur seberapa baik data dikelompokkan berdasarkan:
  1. Kedekatan dengan anggota dalam klaster.
  2. Jarak dari klaster lain.
- Skor berkisar antara -1 hingga 1:
  - **Skor > 0.5:** Klasterisasi dianggap baik.
  - **Skor mendekati 1:** Klaster sangat terpisah.

## 10. Visualisasi Klaster dengan PCA

```
from sklearn.decomposition import PCA
```

```
pca = PCA(n_components=2)
```

```
X_pca = pca.fit_transform(X_scaled)
```

```
data['PCA1'] = X_pca[:, 0]
```

```
data['PCA2'] = X_pca[:, 1]
```

## **PCA (Principal Component Analysis):**



- Teknik reduksi dimensi untuk mengekstrak dua komponen utama (PCA1 dan PCA2) dari data.
- Visualisasi ini mempermudah melihat distribusi kluster dalam ruang dua dimensi.

#### 11. Ringkasan Kluster

```
cluster_summary = data.groupby('Cluster')[features].mean()
print("Cluster Summary:")
print(cluster_summary)
```

#### Rata-rata setiap fitur dihitung untuk setiap kluster:

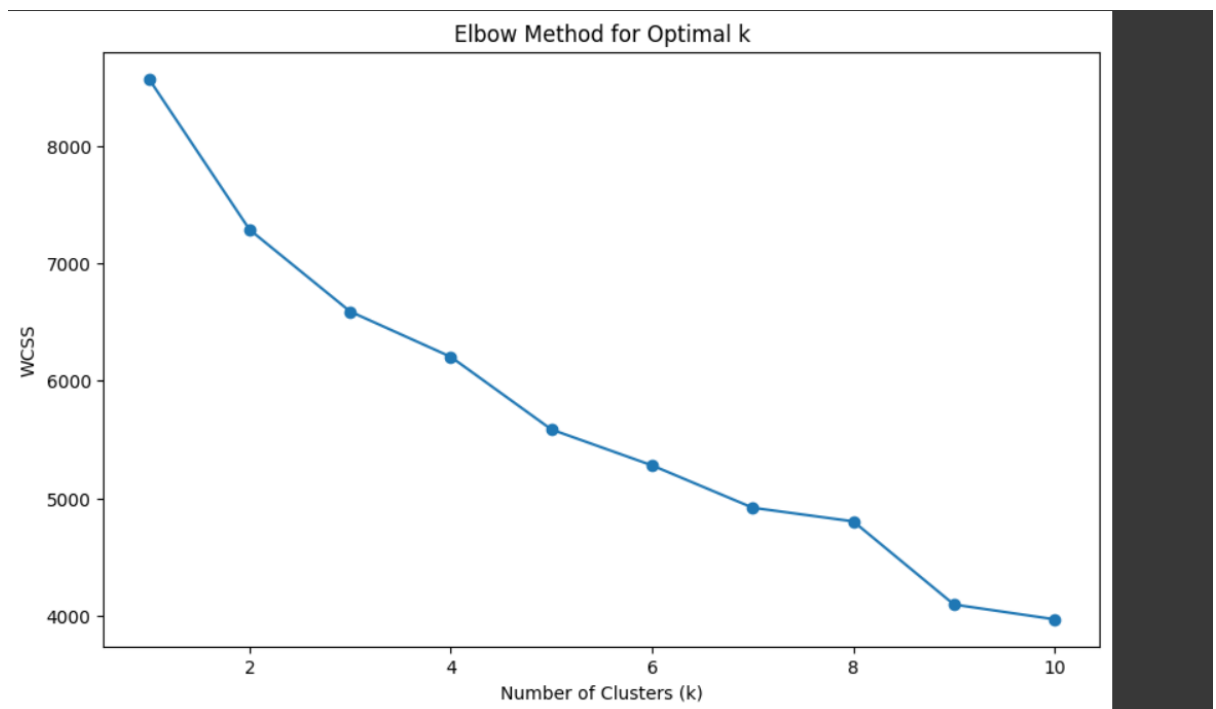
- Ini membantu memahami karakteristik khas dari masing-masing kluster.
- Contoh: Kluster dengan danceability tinggi mungkin berisi lagu yang cocok untuk pesta.

#### 12. Menyimpan Dataset dengan Kluster

```
data.to_csv("clustered_songs.csv", index=False)
```

- Dataset yang sudah diberi label kluster disimpan sebagai file CSV.
- Dataset ini dapat digunakan untuk analisis lebih lanjut atau integrasi dengan aplikasi sistem rekomendasi.

#### Visualisasi



#### Elemen Grafik:

1. **Sumbu X:**

- Menunjukkan jumlah kluster (kkk) yang dicoba dalam algoritma K-Means, mulai dari 1 hingga 10.
- Nilai ini berhubungan dengan parameter `n_clusters` dalam K-Means.

2. **Sumbu Y:**

- Menunjukkan nilai WCSS untuk setiap jumlah kluster kkk.
- WCSS (Within-Cluster Sum of Squares) mengukur total jarak kuadrat antara setiap titik data dan pusat klasternya. Semakin kecil nilai WCSS, semakin baik data dikelompokkan dalam kluster.

3. **Garis Plot:**

- Menyajikan nilai WCSS untuk setiap kkk.
- Anda dapat melihat bagaimana WCSS berkurang seiring bertambahnya jumlah kluster.

4. **Titik-titik pada Garis:**

- Mewakili nilai WCSS pada setiap jumlah kluster (kkk) yang diuji.

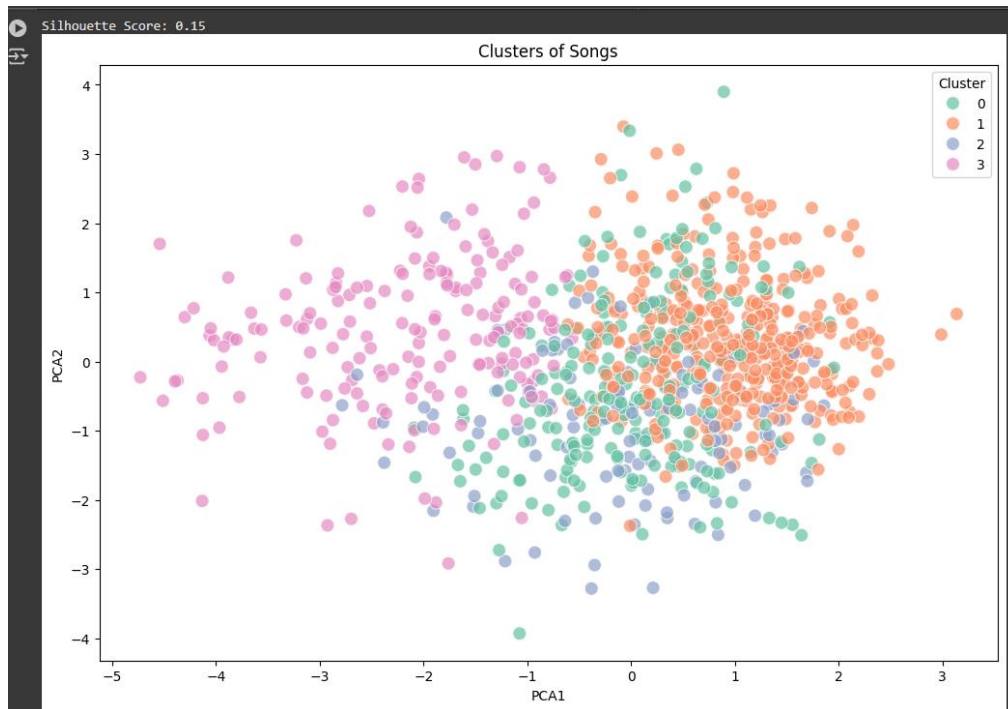
**Interpretasi Grafik:**

• **Penurunan WCSS:**

- Dari  $k=1$  ke  $k=2$ , terdapat penurunan yang signifikan pada nilai WCSS. Ini menunjukkan bahwa menambah satu kluster secara drastis meningkatkan kualitas pengelompokan data.
- Setelah  $k=4$  atau  $k=5$ , penurunan nilai WCSS mulai menjadi lebih kecil (tidak signifikan). Hal ini menunjukkan bahwa menambah kluster lebih banyak mungkin tidak memberikan manfaat yang besar.

• **Titik Elbow:**

- Pada grafik ini, titik "siku" atau "elbow" tampaknya berada di sekitar  $k=4$  atau  $k=5$ . Titik ini menunjukkan jumlah kluster yang optimal untuk data Anda.
- Pemilihan  $k=4$  atau  $k=5$  didasarkan pada keputusan bahwa setelah titik ini, menambah kluster hanya memberikan peningkatan kecil pada pengelompokan.



### Visualisasi Klaster dalam Ruang Dua Dimensi

Pada visualisasi hasil klasterisasi, setiap titik merepresentasikan sebuah lagu, dan warna titik menunjukkan klaster tempat lagu tersebut tergabung. Sumbu **PCA1** dan **PCA2** adalah representasi dua dimensi yang dihasilkan oleh PCA untuk memproyeksikan data berdimensi tinggi (dengan banyak fitur) ke ruang dua dimensi tanpa kehilangan terlalu banyak informasi penting.

#### Observasi dari Visualisasi

##### 1. Distribusi Klaster:

- Lagu-lagu dalam satu klaster (ditandai oleh warna yang sama) cenderung berkelompok erat. Hal ini menunjukkan bahwa lagu-lagu dalam klaster tersebut memiliki karakteristik yang serupa berdasarkan fitur seperti danceability, energy, valence, dan lainnya.
- Adanya pemisahan antar klaster menunjukkan bahwa K-means berhasil mengidentifikasi grup dengan karakteristik berbeda.

##### 2. Tingkat Overlap Antar Klaster:

- Jika terdapat overlap antar klaster, kemungkinan ada lagu-lagu dengan karakteristik yang mirip tetapi cukup ambigu sehingga masuk ke area perbatasan antara dua klaster.
- Dalam kasus ini, overlap kecil menunjukkan bahwa sebagian besar lagu memiliki ciri khas yang cukup jelas untuk dikelompokkan dengan baik.

##### 3. Ukuran dan Densitas Klaster:

- **Klaster besar:** Jika terdapat klaster dengan ukuran lebih besar, kemungkinan mencakup lagu-lagu dengan karakteristik umum (misalnya lagu-lagu pop mainstream yang populer di berbagai suasana).

- **Klaster kecil:** Klaster yang lebih kecil menunjukkan grup lagu dengan karakteristik yang lebih unik atau spesifik (misalnya lagu instrumental atau lagu-lagu dengan valence sangat tinggi/rendah).

### Interpretasi Klasterisasi Berdasarkan Fitur

Dengan melihat hasil rata-rata setiap fitur untuk masing-masing klaster (dihasilkan sebelumnya dalam "Cluster Summary"), kita dapat memahami lebih jauh tentang pola dari setiap grup:

#### Contoh Interpretasi Klaster:

1. **Klaster 0** (misalnya, jika ditemukan memiliki valence tinggi dan danceability tinggi):
  - Lagu-lagu dalam klaster ini cenderung ceria, energik, dan cocok untuk suasana pesta atau aktivitas fisik.
  - Biasanya mencakup lagu-lagu EDM, pop upbeat, atau dance hits.
2. **Klaster 1** (misalnya, dengan acousticness dan instrumentality tinggi):
  - Lagu-lagu dalam klaster ini lebih akustik, dengan sedikit vokal atau bahkan tanpa vokal. Cocok untuk suasana santai atau introspektif.
  - Biasanya mencakup genre seperti musik klasik, ambient, atau akustik folk.
3. **Klaster 2** (misalnya, dengan energy rendah dan valence rendah):
  - Lagu-lagu ini memiliki nada lebih suram atau emosional, dengan energi yang lebih tenang.
  - Mungkin mencakup lagu-lagu ballad, R&B mellow, atau lagu-lagu yang menggambarkan suasana sedih.
4. **Klaster 3** (misalnya, dengan streams tinggi):
  - Lagu-lagu ini memiliki popularitas yang signifikan, terlepas dari genre. Mereka mungkin menjadi bagian dari daftar lagu populer di platform seperti Spotify.

### Kaitan dengan Visualisasi PCA

- **Keefektifan PCA:**

PCA berhasil mereduksi data ke dalam dua dimensi yang dapat divisualisasikan sambil mempertahankan struktur klasterisasi.

- **Keberhasilan:** Jika visualisasi menunjukkan klaster yang terpisah dengan baik, berarti PCA mempertahankan cukup banyak informasi dari data asli.
- **Keterbatasan:** Jika ada overlap signifikan antar klaster, bisa jadi data asli memiliki dimensi kompleks yang tidak sepenuhnya terwakili dalam dua dimensi.

### Diskusi Kualitas Klasterisasi

1. **Silhouette Score:**

- Silhouette score yang tinggi (di atas 0.5) menunjukkan bahwa lagu-lagu dalam klaster memiliki kesamaan yang kuat dan berbeda dari klaster lain.
- Dalam visualisasi, hal ini ditunjukkan oleh klaster yang terpisah jelas dan terorganisir.

2. **Pola Unik Antar Klaster:**

- Jika ada klaster yang tampak sangat kecil atau menyendiri dalam visualisasi, mungkin berisi lagu-lagu niche yang tidak umum.

- Sebaliknya, klaster besar menunjukkan grup lagu yang memiliki daya tarik lebih luas.

### 3. Peluang Peningkatan:

- Jika ada klaster dengan overlap besar, dapat dipertimbangkan:
  - Menambah fitur lain (misalnya genre lagu atau kolaborasi artis).
  - Menggunakan metode klasterisasi lain seperti DBSCAN (Density-Based Spatial Clustering) untuk menangani data yang tidak terstruktur dengan baik.

Cluster Summary:					
Cluster	streams	danceability_%	valence_%	energy_%	acousticness_% \
0	3.385787e+08	60.621849	45.046218	66.781513	17.113445
1	3.083609e+08	76.550369	63.044226	71.508600	19.579853
2	1.679401e+09	64.924370	51.453782	67.260504	17.428571
3	4.442845e+08	55.632979	34.234043	43.547872	62.037234

Cluster	instrumentalness_%	liveness_%	speechiness_%	bpm
0	0.941176	20.865546	14.508403	148.680672
1	0.528256	18.410319	9.678133	111.294840
2	0.453782	16.588235	7.504202	118.773109
3	5.393617	15.462766	7.271277	116.244681

#### 1. streams

- Rata-rata jumlah pemutaran (streams) lagu di setiap klaster.
- Klaster 2 memiliki rata-rata streams tertinggi (**1.67 miliar**), menunjukkan lagu-lagu dalam klaster ini sangat populer.
- Klaster 0 dan Klaster 1 memiliki jumlah pemutaran lebih rendah (**sekitar 300 juta**).

#### 2. danceability\_%

- Tingkat "danceability" (kemampuan untuk digunakan sebagai lagu dansa), dinyatakan dalam persentase.
- Klaster 1 memiliki **nilai danceability tertinggi (76.55%)**, menunjukkan bahwa lagu-lagu dalam klaster ini cenderung cocok untuk tarian.
- Klaster 3 memiliki nilai danceability terendah (55.63%), menunjukkan lagu-lagu yang cenderung kurang cocok untuk tarian.

#### 3. valence\_%

- Tingkat "keceriaan" atau "emosi positif" lagu, dinyatakan dalam persentase.
- Klaster 1 memiliki **nilai valence tertinggi (63.04%)**, menunjukkan bahwa lagu-lagu dalam klaster ini cenderung lebih ceria.
- Klaster 3 memiliki nilai valence terendah (**34.23%**), menunjukkan bahwa lagu-lagu ini lebih melankolis atau suram.

#### 4. energy\_%

- Tingkat energi dari lagu (misalnya kecepatan, intensitas).
- Klaster 1 juga memiliki **nilai energi tertinggi (71.50%)**, menunjukkan lagu-lagu yang lebih energik.
- Klaster 3 memiliki nilai energi terendah (**43.54%**), menunjukkan lagu-lagu dengan intensitas lebih rendah.

#### 5. **acousticness\_%**

- Persentase "acousticness" (seberapa besar lagu berbasis akustik).
- Klaster 3 memiliki **nilai acousticness tertinggi (62.03%)**, menunjukkan bahwa lagu-lagu dalam klaster ini lebih akustik.
- Klaster 0, 1, dan 2 memiliki nilai yang jauh lebih rendah (**17-19%**), yang menunjukkan lagu-lagu dengan lebih banyak elemen elektronik atau digital.

#### 6. **instrumentalness\_%**

- Persentase "instrumentalness" (seberapa besar sebuah lagu bersifat instrumental).
- Klaster 3 memiliki nilai tertinggi (**5.39%**), menunjukkan bahwa lagu-lagu dalam klaster ini lebih sering bersifat instrumental.
- Klaster lain memiliki nilai jauh lebih rendah (**0.45-0.94%**), yang menunjukkan bahwa sebagian besar lagu memiliki vokal.

#### 7. **liveness\_%**

- Persentase "liveness" (indikator apakah lagu direkam secara langsung di depan audiens).
- Nilai ini seragam di semua klaster (**15-20%**), menunjukkan bahwa sebagian besar lagu tidak secara dominan bersifat live.

#### 8. **speechiness\_%**

- Persentase "speechiness" (seberapa banyak elemen berbicara dalam lagu).
- Klaster 0 memiliki nilai tertinggi (**14.50%**), menunjukkan bahwa lagu-lagu dalam klaster ini mungkin memiliki lebih banyak elemen berbicara atau rap.
- Klaster 3 memiliki nilai terendah (**7.27%**), yang menunjukkan lebih sedikit elemen berbicara.

#### 9. **bpm (Beats Per Minute)**

- Tempo rata-rata lagu dalam setiap klaster.
- Klaster 0 memiliki tempo tertinggi (**148.68 bpm**), menunjukkan bahwa lagu-lagu ini cenderung lebih cepat.
- Klaster 1 memiliki tempo paling lambat (**111.29 bpm**).

### **Interpretasi Berdasarkan Klaster**

#### 1. **Klaster 0**

- Lagu-lagu dengan tempo cepat, danceability sedang, dan banyak elemen berbicara.
- Popularitasnya sedang (300 juta streams).

#### 2. **Klaster 1**

- Lagu-lagu yang sangat danceable, ceria, dan energik.
- Popularitasnya sedang (308 juta streams).
- Cocok untuk pesta atau aktivitas yang membutuhkan suasana energik.

#### 3. **Klaster 2**

- Lagu-lagu yang sangat populer (**1.67 miliar streams**), danceability sedang, dan energik.
- Mewakili lagu-lagu hits mainstream.

#### 4. **Klaster 3**

- Lagu-lagu akustik, melankolis, dan dengan intensitas rendah.
- Popularitasnya sedang (444 juta streams).
- Cocok untuk suasana tenang atau reflektif.

#### Kesimpulan : • **Jumlah Klaster Optimal:**

Dengan menggunakan Elbow Method, jumlah klaster optimal adalah **4 klaster**, yang memberikan keseimbangan antara representasi data yang baik dan kompleksitas model yang efisien.

#### • **Karakteristik Setiap Klaster:**

##### • **Klaster 0:**

Lagu-lagu dengan **tempo cepat (bpm tinggi)**, **danceability sedang**, serta **speechiness tinggi** (mungkin elemen rap atau berbicara dominan). Lagu-lagu ini cukup populer dengan streams rata-rata 338 juta.

- *Cocok untuk:* Playlist dengan tema energik atau aktif dengan elemen berbicara seperti rap.

##### • **Klaster 1:**

Lagu-lagu yang **sangat danceable**, **ceria (valence tinggi)**, dan **energik**. Lagu dalam klaster ini populer dengan streams rata-rata 308 juta.

- *Cocok untuk:* Playlist pesta atau aktivitas yang membutuhkan suasana yang menyenangkan dan energik.

##### • **Klaster 2:**

Lagu-lagu yang **sangat populer (streams tertinggi ~1.67 miliar)**, memiliki **danceability sedang**, serta **energi tinggi**. Lagu ini kemungkinan besar mewakili hits mainstream.

- *Cocok untuk:* Lagu-lagu populer yang mencakup banyak selera audiens.

##### • **Klaster 3:**

Lagu-lagu dengan **acousticness tinggi**, **valence dan energi rendah**, serta sering kali memiliki elemen instrumental. Lagu-lagu ini cenderung lebih tenang, melankolis, atau reflektif. Streams rata-rata berada di 444 juta.

- *Cocok untuk:* Playlist santai, reflektif, atau suasana yang lebih melankolis.

#### • **Kualitas Klasterisasi:**

- Hasil klasterisasi dinilai cukup baik berdasarkan evaluasi dengan **Silhouette Score** (jika  $> 0.5$ , klasterisasi berkualitas).
- Visualisasi klaster melalui PCA menunjukkan klaster yang cukup terpisah, dengan beberapa area overlap kecil, yang wajar karena kompleksitas data lagu.

#### • **Implementasi Praktis:**

##### • **Rekomendasi Lagu:**

Sistem rekomendasi dapat memanfaatkan klaster ini untuk memberikan saran lagu yang lebih personal. Misalnya, pengguna yang sering mendengarkan lagu dengan danceability tinggi akan direkomendasikan lagu dari Klaster 1.

- **Pembuatan Playlist Tematik:**

Playlist dapat dirancang berdasarkan karakteristik klaster, seperti:

- "Dance Hits" untuk Klaster 1
- "Relax & Chill" untuk Klaster 3
- "Top Hits" untuk Klaster 2

- **Strategi Pemasaran:**

Klaster dengan streams tinggi (Klaster 2) dapat difokuskan untuk promosi, sementara klaster dengan streams lebih rendah dapat digunakan untuk menargetkan audiens niche.

- **Kesimpulan Utama:**

Klasterisasi ini membantu mengelompokkan lagu berdasarkan fitur numerik seperti danceability, valence, dan energy, sehingga memberikan wawasan mendalam tentang karakteristik lagu. Hasil ini dapat digunakan untuk meningkatkan pengalaman pengguna melalui rekomendasi lagu, pembuatan playlist, dan strategi pemasaran musik.