

# Penerapan Fuzzy C-Means pada Ketertarikan Musik Indonesia Melalui Tangga Nada

Muhammad Rafif Yufinza<sup>1</sup>, Raga Hammam Ramadhan<sup>2</sup>,  
Niaka Anggara Kusumawan<sup>3</sup>, Rafdy Ahmad Hawis<sup>4</sup>

<sup>1,2,3,4</sup>Program Studi Matematika, Universitas Brawijaya

Jl. Veteran, Ketawanggede, Lowokwaru, Kota Malang

e-mail: <sup>1</sup>[mralfifyuvinza@student.ub.ac.id](mailto:mralfifyuvinza@student.ub.ac.id), <sup>2</sup>[ragahammam97@student.ub.ac.id](mailto:ragahammam97@student.ub.ac.id),

<sup>3</sup>[niakaanggara@student.ub.ac.id](mailto:niakaanggara@student.ub.ac.id),

<sup>4</sup>[rafdy.hawis@gmail.com](mailto:rafdy.hawis@gmail.com)

## Abstrak

Musik merupakan salah satu sarana yang dapat digunakan oleh Masyarakat Indonesia untuk mengekspresikan perasaan secara bebas. Musik sendiri memiliki dua jenis kunci dalam tangga nada, yaitu kunci mayor dan minor. Fitur audio musik, terutama pada aplikasi zaman sekarang seperti Spotify yang dapat diakses melalui API, menjadikan sebagai sebuah penompang dalam memberikan sebuah rekomendasi lagu sesuai dengan tipe yang masyarakat suka. Namun, terkait dengan kunci mayor dan minor yang berada dalam keadaan bias (mayor murni, minor murni) dapat mendatangkan keambiguitas dalam pengelompokkan lagu yang disukai masyarakat.

Oleh karena itu, dengan mengolah data “Top 50 Indonesia” dari Spotify menggunakan algoritma Fuzzy C-means, dapat membantu dalam permasalahan pengelompokkan mayor dan minor melalui fitur audio yang telah disediakan. Berdasarkan hasil dari algoritma, didapatkan bahwa rata-rata Masyarakat Indonesia memiliki ketertarikan terhadap lagu minor. Melalui penelitian ini, dapat dikembangkan seperti menambahkan skala yang lebih besar terhadap data lagu, dan optimalisasi lebih lanjut.

**Kata kunci**— Mayor, Minor, Fuzzy C-Means,

## 1. PENDAHULUAN

### 1.1. Latar Belakang

Musik memiliki kekuatan emosional yang besar dan menjadi bagian penting dari budaya Indonesia. Dengan musik, masyarakat Indonesia dapat mengekspresikan perasaan atau emosinya secara bebas. Dalam musik, terdapat dua jenis kunci atau tuts, yaitu tangga nada mayor dan minor. Tangga nada mayor dan minor disusun sedemikian sehingga terbentuk suatu musik beremosi yang dapat dirasakan oleh pendengar musik tersebut. Kunci mayor dan minor diakui sebagai fitur musik utama dalam memunculkan respons emosional. Perbedaan antara kunci mayor dan minor berakar pada interval antara nada tonik dan tingkat ketiga dari skala. Kunci mayor umumnya dianggap terdengar cerah, bahagia, dan menggembirakan, sedangkan kunci minor diasosiasikan dengan emosi yang suram, melankolis, atau dramatis [1].

Zaman sekarang, layanan musik daring sudah terkenal dan bahkan sudah banyak layanan langganan berbayar untuk mendengar musik. Meskipun banyak pengunduhan yang ilegal, banyak layanan pengunduhan yang sah yang mengenakan biaya juga ada dan tersedia secara luas[2]. Salah satu aplikasi yang menyediakan langganan musik daring adalah Spotify. Spotify adalah aplikasi layanan musik, *audio book* dan podcast digital. Spotify memberikan fitur audio musik yang didapatkan dari pengguna aplikasi tersebut. Fitur audio di Spotify dapat diakses menggunakan *Application Programming Interface* (API) yang berfungsi sebagai ra bagi dua atau lebih program komputer untuk berkomunikasi satu sama lain. Sesuai dengan namanya, API adalah perangkat lunak antarmuka, yang berarti menyediakan layanan ke bagian perangkat

lunak lain. Protokol atau aturan pengiriman data yang digunakan oleh Spotify *Web API* adalah *hypertext transfer protocol* (HTTP) yang kebetulan sama dengan browser internet ca [2]. Dengan bebasnya akses fitur audio Spotify, data-data yang didapatkan dari pengguna yang sesuai persetujuan Spotify dapat digunakan oleh siapapun yang memiliki akses ke *Web Spotify*.

Dalam musik, sering kali sebuah lagu tidak bersifat “murni mayor” atau “murni minor” dari segi persepsi emosional atau struktur nada yang berdasar dari data fitur-fitur audio musik, karena ada ambiguitas atau gradasi dari setiap bagian fitur numerik seperti *valence*, *energy*, *mode*, atau fitur lainnya. Oleh karena itu, dibutuhkan metode yang mampu menangani ambiguitas. Salah satu metode pengelompokan data yang menangani ambiguitas data yang dikenal adalah Algoritma *Fuzzy C-Means* (FCM) [3, 4]. FCM memungkinkan sebuah lagu untuk memiliki keanggotaan parsial terhadap lebih dari satu klaster, misalnya, 0.6 ke klaster mayor, 0.4 ke klaster minor. Dalam studi oleh Velmurugan, FCM digunakan sebagai metode klustering untuk mengidentifikasi pola dalam data tanpa label ground truth [5]. Hasilnya menunjukkan bahwa FCM efektif dalam mengelompokkan data berdasarkan kemiripan fitur, meskipun tanpa label yang jelas [5]. Algoritma *Fuzzy Clustering* menganggap bahwa setiap sampel memiliki hubungan afiliasi dengan setiap pusat klustering, yaitu fuzzy dari sampel tersebut [6]. FCM meminimalkan fungsi objektif yang memungkinkan titik-titik data menjadi bagian dari beberapa kelompok, yang mencerminkan ketidakpastian data. Prinsip utamanya adalah konsep keanggotaan, di mana asosiasi setiap titik data dengan sebuah klaster dinyatakan sebagai nilai antara 0 dan 1, sehingga memungkinkan hasil pengelompokan yang lebih fleksibel untuk data yang tidak pasti [7]. Dibandingkan metode lain, seperti *K-means* yang mengizinkan satu data point menjadi anggota penuh dari satu klaster, FCM memungkinkan satu data point memiliki derajat keanggotaan di beberapa klaster sekaligus yang cocok untuk klasifikasi preferensi musik karena mampu menangani ketidakpastian atau ambiguitas dan menghasilkan hasil klasterisasi yang lebih fleksibel dan representatif dibanding metode *hard-label* [8].

### 1.2 Rumusan Masalah

Adapun rumusan masalah, sebagai berikut:

- a. Bagaimana penerapan algoritma Fuzzy C-Means dalam klasifikasi musik Top 50 Indonesia pada *platform* Spotify berdasarkan tangga nada mayor atau minor?
- b. Dari data Top 50 Indonesia pada *platform* Spotify, apakah masyarakat Indonesia cenderung lebih tertarik atau menyukai musik dengan tangga nada mayor atau minor?
- c. Bagaimana tingkat keefektifan hasil penilaian tren ketertarikan musik berdasarkan tangga nada mayor atau minor.

### 1.3 Tujuan Penelitian

Penelitian ini akan mencoba mengklasifikasikan lagu-lagu “*Top 50 Spotify Indonesia*” berdasarkan tangga nada untuk mengetahui kecenderungan ketertarikan musikal masyarakat. Dengan menggunakan Algoritma Fuzzy C-Means (FCM), data fitur audio Spotify dapat di klasterisasi dengan fitur yang memiliki kemiripan yang direpresentasikan oleh titik-titik data yang serupa berdasarkan kemiripannya.

Adapun tujuan penelitian sebagai berikut:

- Menerapkan metode algoritma Fuzzy C-Means untuk mengelompokkan musik “*Top 50 Indonesia*” pada *platform* Spotify berdasarkan fitur audio dari Spotify.
- Menganalisis tren ketertarikan musik masyarakat Indonesia berdasarkan tangga nada mayor atau minor hasil pengelompokkan.
- Mengetahui tingkat keefektifan hasil tren ketertarikan melalui analisis tangga nada mayor atau minor.

### 1.4 Manfaat Penelitian

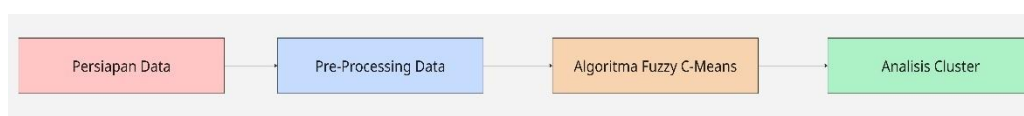
---

Adapun manfaat penelitian, sebagai berikut:

- Memberikan pemahaman pada studi musik digital dan sistem rekomendasi.
- Menyediakan wawasan tentang kecenderungan musikal masyarakat berbasis data.

## 2. METODE PENELITIAN

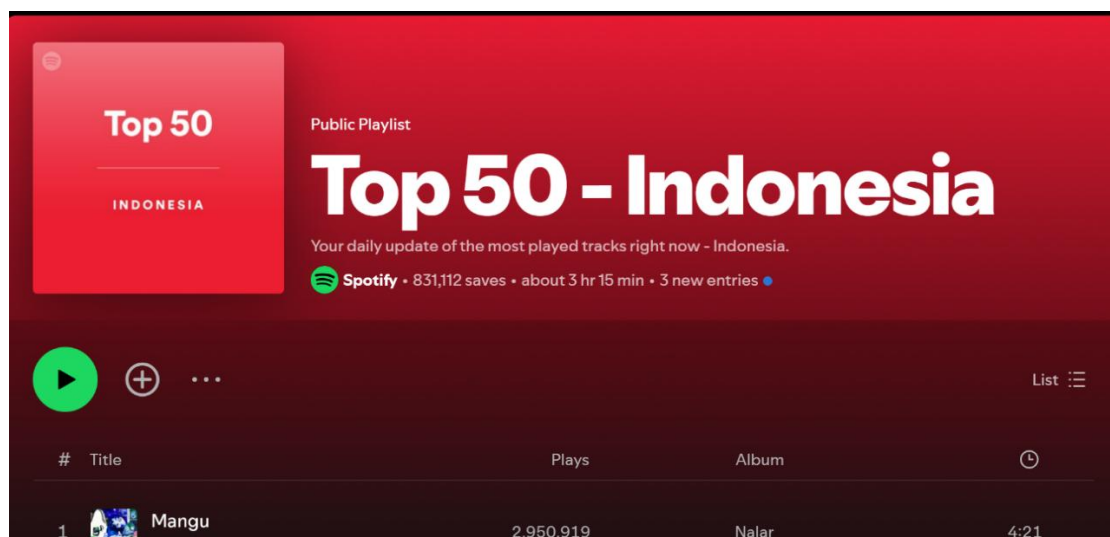
Pada bagian ini, akan dijelaskan beberapa proses yang dilakukan untuk mengetahui keakuratan klastering dalam dataset lagu pada aplikasi Spotify. Proses tersebut, dapat direpresentasikan melalui diagram berikut.



Gambar 1 Flowchart Processing Data

### 2.1 Pengumpulan Data

Data yang akan digunakan sebagai pembuatan model pengelompokan kali ini adalah lagu terpopuler di Indonesia berdasarkan aplikasi Spotify, dimana dataset didapatkan dari API Spotify (Exportify). Dari setiap lagu yang ada dalam Spotify, memiliki berbagai fitur-fitur audio seperti, *Danceability*, *Energy*, *Key*, *Loudness*, dan lainnya pada tabel 1. Spotify, memiliki *playlist* (daftar lagu) yang dikelompokkan berdasarkan kepopuleran, genre, dan sebagainya. Dalam analisis kali ini, akan diambil 50 top lagu terpopuler berdasarkan *playlist* yang ada pada Spotify (dari tahun 2000-2025).



Gambar 2 Top 50 Indonesia (Sumber : Spotify)

Tabel 1 Atribut Audio [6]

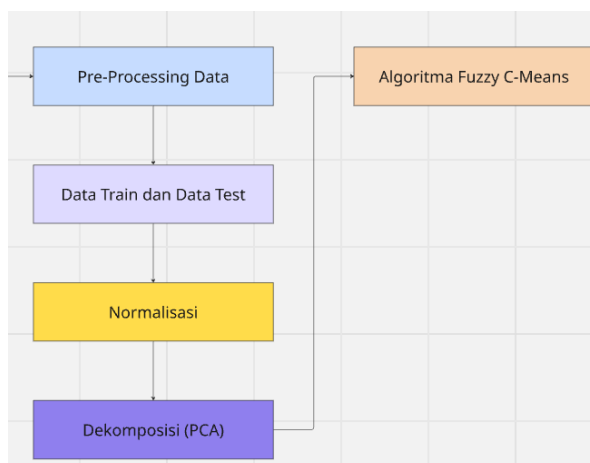
Atribut	Penjelasan
<i>Danceability</i>	Mendeskripsikan bagaimana sebuah lagu dapat dijadikan untuk berdansa
<i>Energy</i>	Lagu yang berenergi tinggi mempunyai rasa yang cepat, kuat, parau. Seperti contoh, lagu rock metal
<i>Key</i>	Kunci / tuts yang ada pada lagu tersebut. Menggunakan notasi yang berhubungan dengan tinggi suara nada
<i>Loudness</i>	Level intensitas suara dari lagu (dB)
<i>Mode</i>	Menyatakan skala dari tangga nada / modalitas pada lagu (mayor dan minor)
<i>Speechiness</i>	Menyatakan apakah lirik tersampaikan dengan jelas pada lagu
<i>Acousticness</i>	Merepresentasikan pada tingkat kepercayaan [0,1] bahwa sebuah lagu merupakan tipe akustik. Jika tingkat kepercayaan 1 maka sebuah lagu diyakini sebagai lagu akustik
<i>Instrumentalness</i>	Memprediksi apakah sebuah lagu tidak memiliki vocal
<i>Liveness</i>	Mendeteksi apakah adanya penonton langsung (biasanya terindikasi apakah lagu tersebut merupakan versi live panggung) pada lagu
<i>Valence</i>	Merepresentasikan apakah terdapat nada positif dalam lagu
<i>Tempo</i>	Menyatakan BPM (ketukan per menit) dalam lagu

Tabel 2 Dataset Top 50 Indonesia Spotify (10 anggota ; didapatkan pada 10/5/2025)

<i>Danceability</i>	<i>Energy</i>	<i>Key</i>	<i>Loudness</i>	<i>Mode</i>	<i>Speechiness</i>	<i>Acousticness</i>	<i>Instrumentalness</i>	<i>Liveness</i>	<i>Valence</i>	<i>Tempo</i>
0.534	0.295	7	-8.733	1	0.0238	0.64700	0.000003	0.0794	0.197	94.048
0.600	0.375	0	-10.668	1	0.0316	0.46500	0.000000	0.5420	0.401	135.816
0.486	0.613	9	-8.748	1	0.0394	0.05690	0.001390	0.1260	0.633	187.531
0.696	0.413	7	-8.548	1	0.0270	0.39600	0.000059	0.0998	0.344	133.905
0.457	0.630	2	-5.826	1	0.0268	0.18300	0.438000	0.0879	0.332	90.009
0.710	0.669	1	-8.961	1	0.0302	0.00619	0.000104	0.1240	0.826	119.993
0.832	0.825	8	-4.951	1	0.0601	0.11900	0.000000	0.1330	0.915	103.011
0.439	0.652	2	-5.403	1	0.0293	0.06240	0.372000	0.1260	0.247	130.029
0.720	0.786	11	-7.396	1	0.0319	0.45600	0.616000	0.1860	0.488	116.032
0.517	0.329	2	-10.699	1	0.0330	0.78500	0.000019	0.0823	0.148	119.863

## 2.2 Pre-Processing Data

Untuk menganalisis keakuratan klastering pada data tersebut, perlu dilakukan *pre-processing* data. Pada tahap ini, data akan dibagi menjadi data tes, dan data uji, lalu akan dinormalisasikan agar tidak terjadi *data leakage*, dan melakukan dekomposisi menjadi dimensi 2 yang akan dilakukan untuk menjalankan algoritma *Fuzzy C-Means*. Hal ini dapat divisualisasikan melalui *flowchart* berikut.



Gambar 3 Flowchart PreProcessing Data

### 2.2.1 Pemisahan Data Train dan Data Test

Untuk menganalisis keakuratan klaster, perlu mengubah data tersebut menjadi 2 data terpisah, yaitu data latih dan data uji. Pada tahap awal, data akan dipisah menjadi data 2 variabel yaitu  $x$  dan  $y$ , dimana  $x$  adalah data atribut dan  $y$  adalah data target. Dikarenakan fokus *clustering* adalah klastering atribut *mode* yaitu mayor (0) dan minor (1), maka  $y$  merupakan data atribut *mode* dan  $x$  merupakan atribut lainnya.

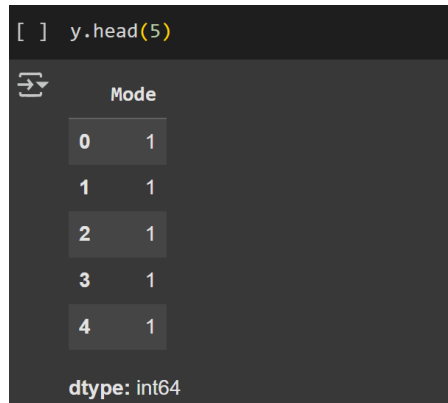
Buat X dan Y

```
[ ] x = Clean_Musik.drop(columns=['Mode'])
      y = Clean_Musik['Mode']
```

[ ] x.head(5)

	Danceability	Energy	Key	Loudness	Speechiness	Acousticness	Instrumentalness	Liveness	Valence	Tempo
0	0.534	0.295	7	-8.733	0.0238	0.6470	0.000003	0.0794	0.197	94.048
1	0.600	0.375	0	-10.668	0.0316	0.4650	0.000000	0.5420	0.401	135.816
2	0.486	0.613	9	-8.748	0.0394	0.0569	0.001390	0.1260	0.633	187.531
3	0.696	0.413	7	-8.548	0.0270	0.3960	0.000059	0.0998	0.344	133.905
4	0.457	0.630	2	-5.826	0.0268	0.1830	0.438000	0.0879	0.332	90.009

Gambar 4 Data x



```
[ ] y.head(5)
```

	Mode
0	1
1	1
2	1
3	1
4	1

dtype: int64

Gambar 5 Data Target y

Setelah data x dan y dipisah, akan dilakukan pemisahan data menjadi data latih dan data uji dengan `train_test_split` dari `sklearn.model_selection`. Fungsi *train test split* digunakan untuk memisahkan dataset menjadi 2 bagian, yaitu data latih dan data uji [9]. Berdasarkan penelitian [10], *train test split* yang ideal adalah 70% atau 80% training data dan 30% atau 20% testing data. Sehingga dilakukan *train test split* 80% data latih dan 20% data uji agar mendapatkan akurasi yang ideal.

```
[10] X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42, stratify=y)
```

Gambar 6 Train test Split x dan y

### 2.2.2 Normalisasi

Setelah data telah dipisah menjadi data latih dan data uji, data tersebut akan dilakukan *feature scaling*. Algoritma *Machine Learning* dan optimasi akan bekerja lebih optimal jika berada dalam skala yang sama [11]. Terdapat 2 macam jenis *feature scaling*, yaitu *Min-Max scaling* (Normalisasi) dan standarisasi. Dalam kasus ini, akan dilakukan normalisasi pada data tersebut. Normalisasi data, dapat dituliskan dengan formula berikut [11]:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

Dalam implementasinya, dapat menggunakan `MinMaxScaler` yang ada pada library `sklearn.preprocessing`.

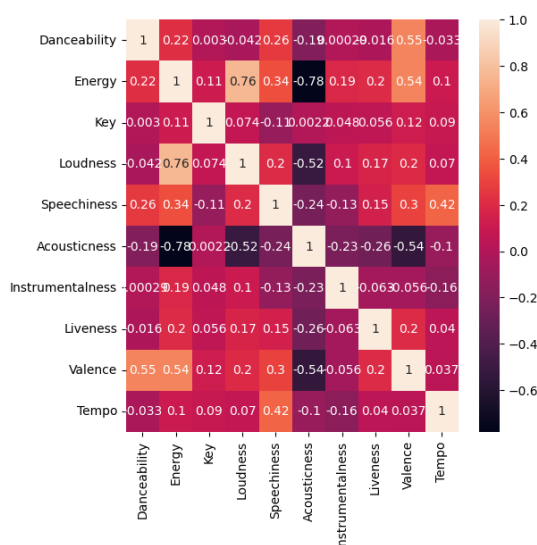
```
scaler = MinMaxScaler()
X_normalized_train = scaler.fit_transform(X_train)
X_normalized_test = scaler.transform(X_test)
```

Gambar 7 Normalisasi x train dan x test

Dalam normalisasi data, kita hanya menormalisasikan data `x_train` dan `x_test`, dikarenakan data x adalah data atribut, sedangkan data y adalah data target 0 atau 1. Data x sendiri memiliki nilai yang beragam pada atributnya, berdasarkan tabel 2. Sehingga, agar dapat bekerja dalam *Fuzzy C-Means*, diperlukan skala yang sama antar data atribut x. Sedangkan data y (*mode*), sudah dalam bentuk *encoding* (0,1).

### 2.2.3 Dekomposisi PCA

PCA atau *Principal Correlation Analysis* adalah teknik yang digunakan untuk mengurangi dimensi atribut dan untuk visualisasi data [12]. PCA digunakan ketika atribut data dalam klastering akan dikurangi dimensi menjadi dimensi yang lebih stabil, seperti dimensi 5 ke dimensi 2 dan sebagainya. Menurut [12], PCA bekerja dengan algoritma dimana data berukuran matrix  $X$  akan dikomposisikan menjadi ukuran baru  $W$  dengan properti yg mewakili. Hasil dari PCA akan mengambil korelasi yang tinggi antar atribut data. PCA melihat dari pola pada data yang diberikan lalu diolah menjadi variasi data, yang dimana variasi tersebut menyatakan atribut yang saling berkorelasi tinggi. Dari PCA ini kita dapat menganalisis dari klastering dengan dimensi yang lebih rendah, tetapi setiap atribut PCA akan mewakili seluruh dari atribut dimensi, melalui korelasi yang kuat.



Gambar 8 Matriks Korelasi Data x

Dari korelasi tersebut, PCA akan mencari dimensi nilai yang sesuai dengan korelasi tersebut. PCA digunakan untuk menentukan nilai optimal kluster dalam dataset dengan menganalisis proporsi dari variansi dari setiap komponen utama, dari tiap variabel, dan korelasinya [12]. Berikut implementasi dari data yang dibentuk PCA.

```
pca = PCA(n_components=2)
X_pca_train = pca.fit_transform(X_normalized_train)
X_pca_test = pca.transform(X_normalized_test)
```

X\_pca\_train

```
array([[ -0.19408419,  0.11695171],
       [ -0.55943342, -0.16063889],
       [  0.6905731 , -0.32063291],
       [ -0.15665834,  0.4402988 ],
       [  0.45121325,  0.48542555],
       [ -0.38605167,  0.58501033],
       [  0.16815159, -0.46732433],
       [ -0.9139005 ,  0.25455466],
       [  0.58871937,  0.04474056],
```

Gambar 9 Dekomposisi x dalam PCA

### 2.3 Algoritma Fuzzy C-Means (FCM)

Ketika suatu algoritma klastering fungsi tujuan mampu meminimalkan *error function* maka biasa disebut C-Means, di mana  $c$  menunjukkan jumlah kelas atau jumlah klaster. Jika kelas-kelas tersebut dibolehkan bersifat fuzzy, maka dapat digunakan algoritma klastering Fuzzy C-Means (FCM) [13].

Perhatikan himpunan  $n$  vektor  $X = (x_1, x_2, \dots, x_n)$  dengan  $2 \leq c \leq n$  yang akan dikelompokkan ke dalam  $c$  klaster. Setiap vektor  $x_k \in \mathbb{R}^s$  direpresentasikan oleh  $s$  pengukuran bernilai riil yang mewakili fitur-fitur objek  $x_i$ . Matriks keanggotaan yang dikenal sebagai matriks partisi fuzzy digunakan untuk menggambarkan matriks keanggotaan fuzzy. Himpunan matriks partisi fuzzy ( $c \times n$ ) dinotasikan dengan  $M_{fc}$  sebagaimana didefinisikan pada (2)

$$M_{fc} = \{W \in \mathbb{R}^{cn} | W_{ik} \in [0, 1], \forall i, k; \sum_{i=1}^c W_{ik} = 1, \forall k; 0 < \sum_{k=1}^n W_{ik} < n, \forall i\} \quad (2)$$

di mana  $1 \leq i \leq c, 1 \leq k \leq n$ .

Dari definisi di atas, dapat diketahui bahwa elemen-elemen dapat masuk ke dalam lebih dari satu klaster dengan tingkat keanggotaan yang berbeda. Total derajat keanggotaan setiap elemen dinormalisasi menjadi 1, sehingga tidak ada satu pun klaster yang memuat seluruh titik data [14].

Fungsi tujuan seperti yang ditunjukkan pada persamaan (3) dari algoritma FCM dapat dihitung dengan menggunakan nilai keanggotaan dan jarak Euclidian (4)

$$J_m(W, P) = \sum_{\substack{1 \leq k \leq n \\ 0 \leq i \leq c}} (w_{ik})^m (d_{ik})^2 \quad (3)$$

di mana

$$d_{ik} = ||x_k - p_i|| \quad (4)$$

Di mana  $m \in (1, +\infty)$  adalah parameter yang mendefinisikan tingkat *fuzziness* dari klaster yang dihasilkan dan  $d_{ik}$  adalah jarak Euclidean dari objek  $x_k$  ke pusat klaster  $p_i$  [14].

Proses meminimalkan fungsi tujuan  $J_m$  dalam algoritma FCM dilakukan dengan cara memperbarui matriks partisi secara iteratif menggunakan persamaan (5) dan (6) [15].

$$P_i = \frac{\sum_{k=1}^n w_{ik}^m x_k}{\sum_{k=1}^n w_{ik}^m} \quad (5)$$

$$W_{ik} = \frac{1}{\sum_{j=1}^c \left( \frac{d_{ik}^{(b)}}{d_{jk}^{(b)}} \right)^{\frac{2}{m-1}}} \quad (6)$$

Metode FCM, termasuk fungsi keanggotaan pada Persamaan (7) dan langkah-langkah algoritmanya, dijelaskan secara rinci dalam [14].

$$\mu_{i,j} = \left[ \sum_{t=1}^c \left( \frac{||x_j - v_i||_A}{||x_j - v_t||_A} \right)^{\frac{2}{m-1}} \right]^{-1} \quad (7)$$

Nilai keanggotaan sampel ke- $j$  pada klaster ke- $i$  dinyatakan dengan  $\mu_{i,j}$ , sedangkan banyak klaster dilambangkan dengan  $c$ . Sampel ke- $j$  dilambangkan  $x_j$  dan titik pusat klaster ke- $i$  dilambangkan  $v_i$ . Notasi  $|| \cdot ||_A$  menunjukkan norma.

Berikut algoritma FCM:

1. Inisialisasi jumlah klaster  $c$ .



2. Pilih metrik hasil kali dalam norma Euklides dan metrik pembobotan (*fuzziness*).
3. Inisialisasi prototipe kluster  $P^{(0)}$  dengan penghitung iterasi  $b = 0$ .
4. Hitung matriks partisi  $W^b$  menggunakan persamaan (6).
5. Perbarui pusat kluster fuzzy  $W^{(b+1)}$  dengan persamaan (5).
6. Jika  $\|P^{(b)} - P^{(b+1)}\| < \varepsilon$  maka hentikan, jika tidak, ulangi langkah 2—4.

Implementasinya sebagai berikut:

```
# Fuzzy C-Means implementation
class FuzzyCMeans:
    def __init__(self, n_clusters, m=2, max_iter=100, tol=1e-4):
        self.n_clusters = n_clusters
        self.m = m
        self.max_iter = max_iter
        self.tol = tol

    def fit(self, X):
        n_samples, n_features = X.shape
        # Initialize membership matrix with random values and normalize
        self.U = np.random.rand(n_samples, self.n_clusters)
        self.U = self.U / np.sum(self.U, axis=1, keepdims=True)
        self.centroids = np.zeros((self.n_clusters, n_features))

        for _ in range(self.max_iter):
            # Update centroids
            self.centroids = (self.U ** self.m).T @ X / np.sum(self.U ** self.m, axis=0)[:, None]

            # Calculate distances between samples and centroids
            distances = pairwise_distances(X, self.centroids)

            self.U_old = self.U.copy()
            # Update membership matrix
            self.U = 1 / (distances ** (2 / (self.m - 1)))
            self.U = self.U / np.sum(self.U, axis=1, keepdims=True)

            # Check convergence
            if np.linalg.norm(self.U - self.U_old) < self.tol:
                break

    def predict(self, X):
        distances = pairwise_distances(X, self.centroids)
        # Return index of closest centroid
        return np.argmin(distances, axis=1)
```

Gambar 10 Implementasi Fuzzy C-Means

### 3. HASIL DAN PEMBAHASAN

Pada subbab ini dijelaskan langkah-langkah penerapan algoritma Fuzzy C-Means FCM untuk mengklusterisasi lagu-lagu dalam daftar Top 50 Indonesia di Spotify berdasarkan ciri tangga nada (Mayor atau Minor). Hasil akhirnya akan menunjukkan bagaimana algoritma FCM mampu membedakan dua kategori tangga nada tersebut.

Dengan menggunakan parameter sebagai berikut:

1. Jumlah kluster  $C = 2$  (Mayor dan Minor).
2. Fuzzifier  $m = 2$ .
3. Toleransi konvergensi  $\epsilon = 10^{-4}$ .
4. Maksimum iterasi sebesar 100.

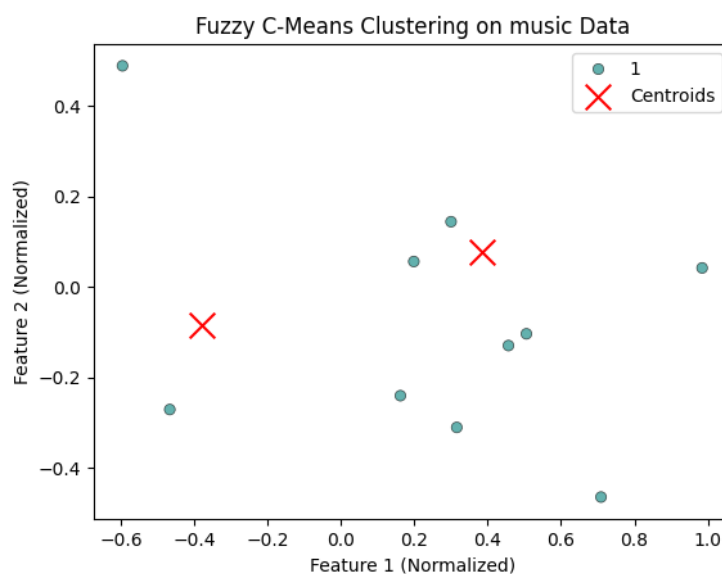
Kemudian diperoleh persentasi akurasi sebesar 90% seperti yang terlihat pada gambar (10).

Percentage Accuracy on Test Set: 90.00%

Gambar 11 Persentasi akurasi

Pada gambar (11) diplot titik-titik lagu (data) pada bidang dua dimensi (Fitur 1 pada sumbu X, Fitur 2 pada sumbu Y), diwarnai sama (karena semua diberi label “1” pada plot awal), tetapi pusat kluster diberi tanda silang (“x” berwarna merah) untuk memudahkan visualisasi pusat kluster. Interpretasi plot sebagai berikut:

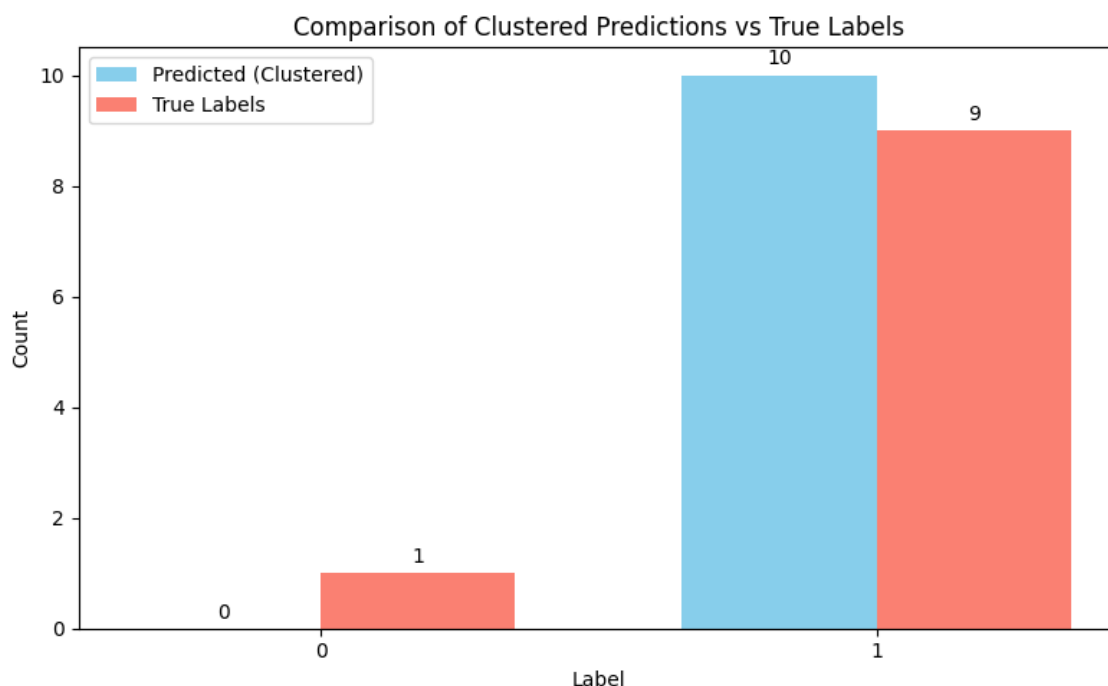
- Centroid Kiri ( $c_1 \approx [-0,40, -0,07]$ ): Ini adalah titik pusat kluster untuk lagu-lagu yang bertangga nada minor. Karena  $mode = 1$  (minor) dinormalisasi menjadi angka negatif di sumbu Y, sebagian besar lagu di sekitar titik ini memang memiliki mode 1.
- Centroid Kanan ( $c_2 \approx [0,44, 0,06]$ ): Ini adalah titik pusat kluster untuk lagu-lagu yang bertangga nada mayor. Nilai  $mode = 0$  (mayor) dinormalisasi menjadi angka positif di sumbu Y, sehingga hampir semua lagu di sekitarnya memiliki mode 0.
- Pola klastering menunjukkan:
  1. Lagu-lagu dengan  $value\ mode = 0$  (Mayor) cenderung dikelompokkan di kluster sebelah kanan.
  2. Lagu-lagu dengan  $value\ mode = 1$  (Minor) cenderung dikelompokkan di kluster sebelah kiri.



Gambar 12 Visualisasi hasil

Berdasarkan perhitungan akurasi seperti pada gambar (10), 90% lagu berhasil diklasifikasikan dengan benar sebagai mayor atau minor. Dengan demikian, FCM berhasil memisahkan mayor dan minor dengan tingkat keberhasilan 90 % pada data uji.

Berdasarkan hasil prediksi dengan FCM, didapatkan bahwa kluster 1 bernilai lebih tinggi daripada kluster 0, yang dimana hal ini juga didukung oleh *True Label* yang menyatakan kluster 1 lebih tinggi dari kluster 0. Sehingga, dapat di prediksi bahwa Indonesia rata rata menyukai lagu yang berkluster 1, yang didalam konteks ini adalah lagu dengan tangga nada minor.



Gambar 13 Perbandingan kluster prediksi dan data test

Tingkat keefektifan penilaian tren ketertarikan musik berdasarkan tangga nada mayor atau minor dapat dievaluasi melalui penerapan algoritma *Fuzzy C-Means* (FCM) terhadap data fitur audio Spotify, khususnya dari *playlist "Top 50 Indonesia"*. Efektivitas ini ditunjukkan melalui beberapa poin berikut:

1. Kemampuan Menangani Ambiguitas

FCM dapat menangani lagu yang tidak sepenuhnya termasuk dalam satu kategori mayor atau minor secara *absolute*. Ini memberikan hasil yang lebih representatif terhadap kenyataan persepsi musik yang seringkali bersifat campuran. Sebagai contoh, sebuah lagu bisa memiliki keanggotaan 60% pada kluster mayor dan 40% pada minor.

2. Akurasi Klasifikasi

Hasil dari pemrosesan data menunjukkan bahwa metode FCM berhasil mengklasifikasikan lagu dengan akurasi yang tinggi, seperti terlihat dalam kesimpulan bahwa kluster 1 (minor) memiliki nilai lebih dominan dan sesuai dengan label asli (*true label*), yang menunjukkan preferensi pendengar.

3. Dukungan Data dan Visualisasi

Proses pra-pemrosesan seperti normalisasi dan PCA membantu meningkatkan efektivitas model FCM dalam menghasilkan pemisahan kluster yang signifikan. Hal ini memperkuat validitas analisis tren ketertarikan berdasarkan mode mayor atau minor.

#### 4. Relevansi Hasil dengan Tren Nyata

Penelitian ini menemukan bahwa masyarakat Indonesia, dalam periode data yang diambil, lebih cenderung menyukai lagu dengan tangga nada minor, yang terkonfirmasi dari hasil dominasi klaster minor dalam hasil FCM.

Penilaian tren ketertarikan musik berdasarkan tangga nada mayor atau minor menggunakan metode FCM tergolong efektif karena mampu mengelola ambiguitas data audio secara fleksibel, menghasilkan klasifikasi yang akurat, serta mencerminkan preferensi musikal masyarakat Indonesia secara *data-driven*. Keefektifan ini diperkuat oleh teknik *preprocessing* dan validasi terhadap label sebenarnya.

#### 4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan mengenai penerapan algoritma Fuzzy C-Means (FCM) untuk mengelompokkan lagu-lagu populer Indonesia berdasarkan tangga nada mayor dan minor, dapat diambil beberapa kesimpulan penting sebagai berikut:

1. Algoritma Fuzzy C-Means terbukti efektif dalam menangani ambiguitas fitur audio musik, terutama dalam konteks klasifikasi berdasarkan tangga nada. Dengan pendekatan keanggotaan fuzzy, setiap lagu dapat memiliki derajat keanggotaan pada lebih dari satu klaster, yang mencerminkan kenyataan bahwa karakteristik emosional musik tidak bersifat mutlak, tetapi spektrum yang bertingkat.
2. Dari hasil klasterisasi terhadap 50 lagu terpopuler Spotify Indonesia, diperoleh dua klaster utama. Berdasarkan analisis rata-rata nilai fitur mode, klaster dengan nilai rata-rata mode mendekati 1 diinterpretasikan sebagai klaster minor, sedangkan klaster dengan rata-rata mendekati 0 diinterpretasikan sebagai klaster mayor.
3. Distribusi klaster menunjukkan bahwa mayoritas lagu populer di Indonesia saat ini cenderung berada dalam klaster minor. Ini mengindikasikan bahwa masyarakat Indonesia memiliki ketertarikan lebih tinggi terhadap lagu-lagu dengan nuansa melankolis atau emosional, yang secara musikal direpresentasikan oleh tangga nada minor.
4. Proses *preprocessing* seperti normalisasi dan reduksi dimensi menggunakan PCA berperan penting dalam meningkatkan efektivitas klasterisasi. Normalisasi memastikan semua fitur berada dalam skala yang setara, sedangkan PCA memungkinkan visualisasi hasil klaster secara lebih informatif.

#### 5. SARAN

1. Penelitian ini menggunakan 50 lagu dari daftar “*Top 50 Indonesia*” sebagai sampel. Untuk mendapatkan hasil yang lebih representatif, disarankan agar penelitian selanjutnya menggunakan dataset yang lebih besar, baik dari segi jumlah lagu, variasi genre, maupun rentang waktu yang lebih luas.
  2. Untuk menilai keunggulan *Fuzzy C-Means* secara lebih objektif, perlu dilakukan perbandingan dengan metode lain seperti *K-Means*, *DBSCAN*, atau *Gaussian Mixture Models*. Hal ini penting untuk memahami kekuatan dan kelemahan relatif dari masing-masing pendekatan.
-

## DAFTAR PUSTAKA

- [1] G. Carraturo, V. Pando-Naude, M. Costa, P. Vuust, L. Bonetti and d. E. Brattico, "The major-minor mode dichotomy in music perception," *Physics of Life Reviews*, vol. 52, pp. 80-106, 2025.
- [2] D. A. G. Machmudin, M. Novita and d. G. Ardaneswari, "Analysis of Spotify's Audio Features Trends using Time Series Decomposition and Vector Autoregressive (VAR) Model".
- [3] J. M. Q. L. J. S. M. T. C. L. D. W. Y. Q. Y. W. d. F. W. W. Zhao, ""Comparison and application of SOFM, fuzzy c-means and k-means clustering algorithms for natural soil environment regionalization in China,"," *Environmental Research*, vol. 216, p. 114519, 2023.
- [4] A. Amirah, "Penerapan Algoritma Fuzzy C-Means (FCM) untuk Klasifikasi Musik Berdasarkan Emosi Menggunakan Parameter Audio," *Skripsi*, 2017.
- [5] S. Velmurugan, "Performance Analysis of K-Means and Fuzzy C-Means Clustering Algorithms for Statistical Distributions," *Symmetry*, vol. 15, p. 1679, 2023.
- [6] L. Liu, S. Z. Sun, X. Y. H. Yu and a. D. Zhang, "A modified Fuzzy C-Means (FCM) Clustering algorithm and its application on carbonate fluid identification," *Journal of Applied Geophysics*, vol. 129, p. 28–35, 2016.
- [7] C. Wu and d. S. Pan, "Fuzzy C-Poincaré Fréchet means clustering in hyperbolic space,," *Expert Systems with Applications*, pp. 128-245, 2025.
- [8] Bezdek and J.C, "Pattern Recognition with Fuzzy Objective Function Algorithms," Plenum Press, New York, 1981.
- [9] M. R. Dalida, L. B. Aquino, W. C. Hod, R. A. Agapor, S. L. Huyo-A and d. G. A. Sampedro, "Music Mood Prediction Based on Spotify's Audio Features Using Logistic Regression," in *Proc. 2022 IEEE 14th Int. Conf. Humanoid, Nanotechnol., Inf. Technol., Commun. Control, Environ. Manag. (HNICEM)*, 2022.
- [10] V. K. a. O. K. A. Gholamy, "A Pedagogical Explanation," UTEP Computer Science Technical Reports, 2018.
- [11] M. A. Ouameur, M. Caza-Szoka and D. dan Massicotte, "Machine learning enabled tools and methods for indoor localization using low power wireless network," *Internet of Things*, vol. 12, pp. 100-300, 2020.
- [12] G. S. M. Khamis, "Utilizing Fuzzy C-Means Clustering and PCA in Public Health: A Machine Learning Approach to Combat CVD and Obesity," Preprints, 2024.
- [13] J. Lázaro, J. Arias, J. L. Martín and C. Cuadrado, " Modified Fuzzy C-Means Clustering Algorithm for Real-Time Applications," in *Field Programmable Logic and Application (FPL 2003)*, Lisbon, Portugal, 2003.
- [14] J. Nayak, B. Naik and H. Behera, "Fuzzy C-Means (FCM) Clustering Algorithm: A Decade Review from 2000 to 2014," in *Computational Intelligence in Data Mining - Volume 2*, New Delhi, Springer India, 2015, p. 133–149.
- [15] H. Izakian, A. Abraham and V. Snášel, "Fuzzy clustering using hybrid fuzzy c-means and fuzzy particle swarm optimization," in *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)* , Coimbatore, India, 2009.