Idealis: Indonesia Journal Information System
Volume 8, Nomor 1, Januari 2025
ISSN 2684-7280 (*online*)
Halaman 74-83
available online at http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index

IMPLEMENTASI *MACHINE LEARNING* DALAM PENGELOMPOKAN MUSIK MENGGUNAKAN ALGORITMA *K-MEANS CLUSTERING*

Bhustomy Hakim[1*], Fergie Joanda Kaunang[2], Cornelius Susanto[3], Jonathan Salim[4], Reynaldi Indradjaja[5]
[1]Sistem Informasi, Fakultas Teknologi dan Desain, Universitas Bunda Mulia, Jakarta, Indonesia
[2,3,4,5] Informatika, Fakultas Teknologi dan Desain, Universitas Bunda Mulia, Jakarta, Indonesia
Email: [1*]bhakim@bundamulia.ac.id, [2]fkaunang@bundamulia.ac.id, [3]s322200140@student.ubm.ac.id, [4]s322200124@student.ubm.ac.id, [5]s322200126@student.ubm.ac.id
(* : corresponding author)

Abstrak-Musik merupakan bagian yang tak mungkin dapat hilang dari hidup semua orang. Banyak orang mendengarkan musik namun dengan preferensi yang berbeda karena tersedia banyak sekali jenis musik yang beragam. Banyak platform penyedia musik streaming berlomba membuat rekomendasi lagu yang sesuai dengan preferensi penggunanya namun masih sulit untuk mengelompokkan musik didalamnya. Penelitian ini bertujuan untuk menganalisis musik menggunakan algoritma *Clustering K-Means*, sebuah metode *unsupervised machine learning*, untuk mengelompokkan lagu berdasarkan fitur-fiturnya seperti tempo, nada, dan elemen-elemen lainnya. Penelitian ini dilakukan dalam konteks digitalisasi musik yang berkembang pesat, di mana platform streaming musik semakin populer dan memungkinkan personalisasi preferensi pengguna. Algoritma *K-Means* digunakan untuk menemukan pola dari berbagai genre musik, sehingga bisa memberikan wawasan mengenai tren musik dan preferensi pendengar. Penelitian ini melibatkan beberapa tahap utama, termasuk eksplorasi data (*Exploratory Data Analysis*/EDA), pengecekan *missing values* dan *outliers*, serta pemilihan fitur yang relevan. Selanjutnya, proses *clustering* dilakukan menggunakan algoritma *K-Means* dengan evaluasi melalui metode *Elbow* dan *Silhouette* untuk menentukan jumlah cluster yang optimal serta menilai kualitas clustering. Penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan sistem rekomendasi musik yang lebih baik dengan meningkatkan pengetahuan di bidang machine learning, khususnya dalam penerapan algoritma *K-Means* untuk pengelompokan data musik. Hasil evaluasi penelitian ini memiliki *Silhouette Scores* 0,53 untuk data *untrimmed* dan *trimmed* yang menunjukkan model yang dibuat sudah menunjukan kualitas *clustering* yang baik. Serta klaster yang terbentuk memiliki indicator utama yaitu fitur *duration_ms* yang dominan dari fitur lain seperti *tempo, valence, liveness,* dan *energy* yang menghasilkan kelompok terbaik untuk digunakan sebagai rekomendasi playlist.

Kata Kunci: Algoritma, Clustering, K-Means, Machine Learning, Musik

Abstract-*Music is an inseparable part of everyone's life. Many people listen to music but with different preferences because there are so many different types of music available. Many music streaming platforms compete to make song recommendations that suit their users' preferences but it is still difficult to group the music in them. This study aims to analyze music using the K-Means Clustering algorithm, an unsupervised machine learning method, to group songs based on their features such as tempo, tone, and other elements. This research was conducted in the context of the rapidly growing digitalization of music, where music streaming platforms are increasingly popular and allow for personalization of user preferences. The K-Means algorithm is used to find patterns from various music genres, so that it can provide insight into music trends and listener preferences. This study involves several main stages, including data exploration (Exploratory Data Analysis/EDA), checking for missing values and outliers, and selecting relevant features. Furthermore, the clustering process is carried out using the K-Means algorithm with evaluation through the Elbow and Silhouette methods to determine the optimal number of clusters and assess the quality of clustering. This research is expected to contribute to the development of a better music recommendation system by increasing knowledge in the field of machine learning, especially in the application of the K-Means algorithm for music data clustering.*

***Keywords**: Algorithm, Clustering, K-Means, Machine Learning, Music*

1. PENDAHULUAN

Musik merupakan sesuatu yang menyenangkan, mendatangkan keceriaan, mempinyai irama (ritme), *melody*, *timbre* (*tone colour*) tertentu untuk membantu tubuh dan pikiran saling bekerjasama [1]. Musik telah menjadi bagian tak terpisahkan dari kehidupan sehari-hari kita, musik berperan sebagai media yang kuat untuk ekspresi diri, koneksi emosional, dan identitas budaya. Musik memiliki banyak manfaat bagi kesehatan manusia dan dapat memberikan kekuatan mentalitas yang baik bagi pendengarnya. Di era teknologi, peran musik menjadi semakin penting, karena *platform streaming* digital dan *smartphone* semakin mempermudah akses dan konsumsi beragam konten musik [2].

Perkembangan *platform streaming* musik telah mengubah cara masyarakat mengakses dan menikmati musik di era digital. Sebelum 2015, layanan *streaming* masih tergolong baru, namun kini telah menjadi metode utama bagi banyak orang untuk mendengarkan musik secara *online*. *Platform* seperti *Spotify*, *Apple Music*, dan

Idealis: Indonesia Journal Information System
Volume 8, Nomor 1, Januari 2025
ISSN 2684-7280 (*online*)
Halaman 74-83
available online at http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index

YouTube Music menawarkan kemudahan dalam menjelajahi jutaan lagu dari berbagai genre dan artis [3]. Seiring kemajuan teknologi, algoritma canggih digunakan untuk mempersonalisasi preferensi pengguna, memungkinkan pengalaman mendengarkan yang lebih relevan dan disesuaikan dengan selera individu. Algoritma ini menganalisis kebiasaan mendengarkan pengguna dan memberikan rekomendasi musik yang diprediksi sesuai, menciptakan interaksi yang lebih mendalam dan personal antara pendengar dan konten musik yang tersedia. Transformasi ini tidak hanya memperkaya pengalaman pengguna tetapi juga membentuk industri musik secara keseluruhan. Genre dan sub-genre juga telah digunakan untuk mengkategorikan musik berdasarkan musik dan liriknya, sehingga memahami pola dan karakteristik musik melalui teknik *machine learning*, seperti algoritma *K-Means*, dapat memberikan wawasan berharga tentang evolusi tren musik dan preferensi pendengar, yang pada akhirnya membentuk cara mendengar dan menikmati musik dalam kehidupan yang semakin digital [4].

Algoritma *K-Means* adalah teknik *unsupervised machine learning* yang dianggap salah satu algoritma *machine learning* yang paling kuat dan populer di kalangan peneliti [5]. Algoritma ini dapat digunakan untuk mengelompokan musik berdasarkan fitur-fiturnya, sehingga memungkinkan untuk menemukan pola musik yang serupa dari berbagai genre musik. Metode ini melibatkan ekstrasi fitur-fitur musik seperti tempo, nada, dan lain lain untuk membagi musik yang memiliki kemiripan kedalam satu kelompok yang sama [6]. Penelitian ini diharapkan dapat memberikan wawasan yang berharga untuk meningkatkan pengetahuan pada bidang *machine learning* terutama pada algoritma *K-Means*, dan juga berguna untuk pengembangan sistem pengelompokan lagu yang lebih lanjut.

2. METODE PENELITIAN

2.1. *Data Preprocessing*

Pada tahap ini, *data preprocessing* merupakan langkah awal yang penting dalam setiap penelitian berbasis data dengan mengubah data mentah ke dalam bentuk yang lebih mudah dipahami. Ini bertujuan untuk menyiapkan data dan mengolah data agar dapat digunakan pada tahap *Building K-Means Model* (*K-Means* membutuhkan data yang sudah diolah agar algoritmanya dapat bekerja dengan lancar dan optimal). Berikut ini adalah tahapan-tahapan pada *Data Preprocessing* yaitu:

a) EDA (*Exploratory Data Analysis*)
EDA adalah langkah awal yang memungkinkan untuk menjelajahi dan memahami data yang sudah dimiliki sebelum memulai analisis yang lebih mendalam [7]. Dalam EDA, dilakukan sejumlah aktivitas, seperti membuat visualisasi, mengidentifikasi pola atau trend dalam data, dan mengecek apakah ada data yang hilang (*NaN*, *string* yang kosong, dan data-data lain yang tidak valid) atau tidak konsisten. Analisis data ini akan mengidentifikasi tipe data, kategori data, dan jumlah data dalam dataset yang disediakan. Hasil analisis tersebut akan disajikan dalam bentuk diagram untuk visualisasi yang lebih jelas dan informatif.

b) *Checking Missing Value*
Memeriksa nilai yang hilang (*missing values*) dalam *dataset* sangat penting karena dapat mempengaruhi hasil analisis dan model [8]. Setelah proses analisis selesai, dilakukan pemeriksaan menyeluruh terhadap data. Pemeriksaan ini bertujuan untuk mendeteksi keberadaan data kosong. Jika ditemukan data kosong, jumlahnya akan ditampilkan.

c) *Checking Skewness*
Setelah melakukan *checking missing value*, langkah berikutnya yang dilakukan adalah menghitung *skewness* sepanjang baris dengan mengabaikan baris dengan nilai yang hilang [9]. Jika nilainya lebih dari nol maka termasuk *skewness* positif yang dimana data akan miring ke kiri dan jika nilainya kurang dari nol maka termasuk *skewness* negatif yang dimana data akan miring ke kanan.

d) *Handling Outliers*
Outlier adalah data yang nilainya jauh berbeda dari data lain dalam satu set data. Kemunculannya bisa disebabkan oleh berbagai hal seperti kesalahan saat pengumpulan data, kesalahan pengukuran, atau peristiwa yang tidak biasa. Terdapat dua tahap di bagian *handling outliers* yaitu *checking outliers* dan *impute outliers* [10]. Dalam *checking outliers*, pertama-tama yang dilakukan adalah membuat *boxplot* agar bisa mengetahui kolom data mana yang memiliki *outliers* (jika di dalam *boxplot* terdapat lingkaran maka itu menandakan datanya sudah menyimpang) dan selanjutnya dilakukan menemukan indeks *outliers* dengan cara menggunakan metode *Interquartile Range* (IQR), yang dimana rumusnya adalah IQR=Q3-Q1. Q1 sebagai batas bawah dan Q3 sebagai batas atas. Setelah itu, mencari indeks *outliers* per kolom data untuk mengetahui apakah ada data yang mempunyai *index outliers* atau tidak ada yang mempunyai indeks *outliers*. Dalam *impute outliers*, data yang dianggap menyimpang akan dihapus dengan cara diubah menjadi nilai *null*. Data yang sudah dijadikan nilai *null* maka akan diisi oleh *mean* atau *median* sesuai dengan *skewness*/kemiringannya. Jika *skewness* nya di antara -2 sampai +2 maka data yang kosong akan

Idealis: Indonesia Journal Information System
Volume 8, Nomor 1, Januari 2025
ISSN 2684-7280 (*online*)
Halaman 74-83
available online at http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index

diisi dengan *mean*, dan jika *skewness* nya di luar batas itu maka data yang kosong akan diisi dengan *median*.

e) *Feature Selection*
Dalam *Feature Selection*, digunakan *variance threshold*. *Threshold* digunakan dalam *Feature Selection* ini dikarenakan untuk menghilangkan data yang memiliki kemiripan yang tinggi agar nantinya pada saat model *clustering* (*K-Means*) menjadi relevan [11]. Pada saat proses ini, dapat dilakukan dengan mengubah *dataset* ke dalam bentuk *heatmap*. *Heatmap* ini berbentuk kotak dengan menampilkan nilai-nilai data yang ada di kolom data tersebut. Jika nilai pada kotak tersebut mempunyai hasil negatif dan positif, maka data akan dianggap berpengaruh, tetapi jika data mendekati angka nol maka data tersebut dianggap tidak berpengaruh dan harus dihapus dikarenakan memiliki kolerasi yang rendah terhadap data lainnya [12]. Setelah ini, sudah diketahui bahwa data mana yang berisi dari *dataset* (data *original/untrimmed*) dan data mana yang sudah dihapus pada saat proses pemilihan data (*data trimmed*).

2.2. *Split Training and Testing Data*

Pada tahap ini, *split training and testing* data akan dibagi menjadi dua data yaitu data *untrimmed* (data *original*) dan data *trimmed* (data yang sudah dihapus pada saat proses pemilihan data). Data *untrimmed* mempresentasikan realitas data sedangkan data *trimmed* memberikan akurasi yang lebih baik [13]. Data *untrimmed* dan data *trimmed* akan dibagi menjadi *training set* yang dimana biasanya 80% dari data dan *testing set* yang biasanya 20% dari data secara terpisah. Lalu pada saat *training set* akan menggunakan algoritma *K-Means* untuk membagi data menjadi beberapa *cluster* sesuai dengan yang diatur oleh diri sendiri. Setelah model *K-Means* dilatih, model tersebut kemudian digunakan untuk memprediksi *cluster* dari *training set* dan menghasilkan label *cluster* untuk setiap sampel dalam *testing set* [14].

2.3. *Building K-Means Model*

Pada tahap ini, *building K-Means* model merupakan data yang telah dikelompokkan melalui proses pemilahan sebelumnya akan diolah menggunakan model *K-Means* [15]. *K-Means* adalah algoritma *clustering* tanpa pengawasan yang mengelompokkan data ke dalam sejumlah *k cluster* tertentu tanpa memerlukan label atau target variabel. Proses ini mencakup beberapa tahap penting. Pertama, menentukan nilai *k*, yaitu jumlah *cluster* yang diinginkan. Memilih nilai *k* yang optimal seringkali memerlukan eksperimentasi dan evaluasi menggunakan metrik tertentu seperti metode *Elbow* dan *Silhouette Scores* [16]. Setelah itu, inisialisasi *centroid*, yaitu titik pusat yang mewakili setiap *cluster*, dilakukan secara acak. Kemudian iterasi hingga konvergensi dengan langkah-langkah yaitu menetapkan setiap titik data ke *cluster* dengan *centroid* terdekat dan memperbarui *centroid* setiap *cluster* menjadi rata-rata dari titik data yang ditetapkan padanya. Setelah menemukan *centroid* dan membentuk *clustering*, hasilnya akan divisualisasikan dalam bentuk *plot K-Means* untuk menganalisis kualitas *clustering*. *Clustering* yang dianggap bagus adalah ketika data tidak bercampur-campur dan memiliki kelompoknya sendiri. Analisis lebih lanjut dilakukan dengan menghitung jarak dari tiap data ke *centroid* masing-masing dan jarak ini dijumlahkan untuk membentuk *total error*.

Nilai *centroid* yang baru ditemukan kemudian diambil dari rata-rata nilai dari setiap kelompok, seperti yang ditunjukkan oleh (1) [17]:

$$C_k = \frac{1}{n_k} \sum d \tag{1}$$

Dengan *Euclidean distance* yang digunakan untuk mengukur jarak antara dua *centorid* yang berbeda dalam garis *Euclidean*. Berikut rumus yang digunakan seperti pada (2), yaitu:

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

2.4. Evaluation (Shilloutte Score)

Silhouette Score adalah pengukuran yang digunakan untuk validasi dan evaluasi apakah jumlah *cluster* sudah baik dan stabil. *Silhouette Score* menghitung rata-rata dari semua data dalam setiap *cluster*, di mana nilai yang dihasilkan merupakan selisih antara nilai separasi dan kompak, yang kemudian dibagi dengan nilai maksimum antara kedua nilai tersebut. Rumus untuk mencari *Silhouette Score* dapat dilihat pada (3) [18]:

$$s(j) = \frac{b(j) + a(j)}{max\{a(j), b(j)\}} \tag{3}$$

Idealis: Indonesia Journal Information System
Volume 8, Nomor 1, Januari 2025
ISSN 2684-7280 (*online*)
Halaman 74-83
available online at http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index

3. HASIL DAN PEMBAHASAN

Penelitian ini menggunakan *dataset* dengan jumlah data 586.673 data. Dikarenakan jumlah data yang sangat besar, maka penelitian ini hanya menggunakan sebanyak sekitas 2.000 data dari proses Data Cleaning yang dilakukan untuk menangani *missing values*. Serta dilakukan teknik *Oversampling Minor Class + Undersampling Major Class* untuk membuat data lebih *dense* untuk siap dilakukan *clustering*. *Dataset* terdiri dari 20 kolom fitur (*feature*) yang terdiri dari 5 kolom yang berisi *feature* dengan tipe data *string* atau *object*, 3 kolom yang berisi *feature* dengan tipe data *kategorikal*, dan 12 kolom yang berisi *feature* dengan tipe data numerikal.

Tahapan awal pembuatan model *machine learning* yaitu melakukan *data preprocessing* yang dimulai dengan melakukan *Exploratory Data Analysis* (EDA) untuk memahami *dataset* secara mendalam sebelum melakukan pemodelan, lalu setelah data sudah dianalisis, akan dilakukan pengecekan data kosong (*missing value*), menghapus *outlier* yang berada didalam disetiap kolom, dan menyeleksi fitur (*feature selection*). *Feature selection* yang digunakan pada penelitian ini adalah menggunakan *variance threshold* dan *heatmap*.

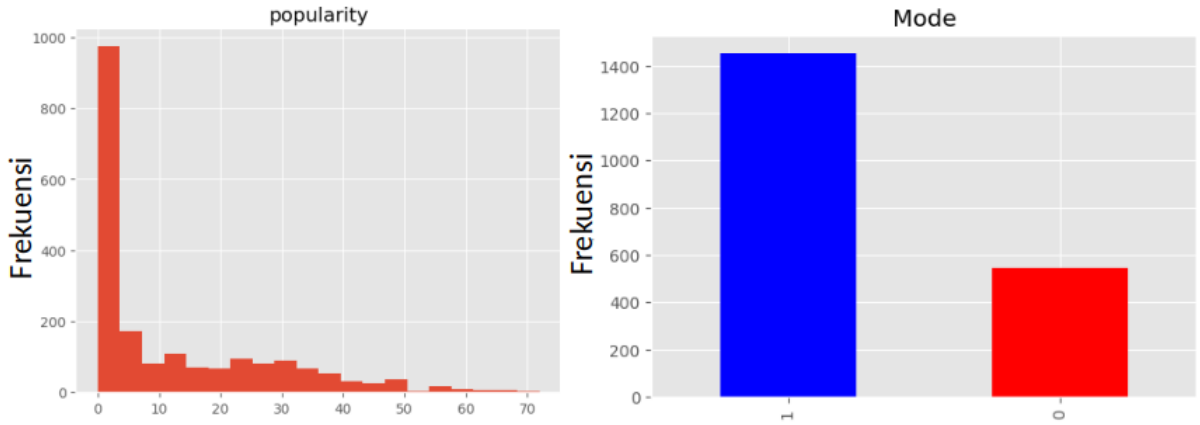
Setelah itu kedua *dataset* yang telah diolah dan dipotong menggunakan *feature selection* akan dimasukan kedalam model algoritma *K-Means* dan dievaluasi perbedaan hasil *clustering* dari data yang diolah saja (*untrimmed*), dan data yang diolah dan dipotong (*trimmed*).

3.1. Data Preprocessing

Pengolahan data mentah yang telah dipotong menjadi 2.000 data akan dilakukan pada tahap ini yang dimulai dari *Exploratory Data Analysis*, *check missing value*, *handling outliers*, *feature selection*.

3.1.1. Exploratory Data Analysis (EDA)

EDA dilakukan untuk memahami secara mendalam karakteristik setiap kolom fitur pada *dataset*. Hasil dari analisis menujukan adanya 3 kolom fitur dengan data kategorikal, dan 12 kolom fitur dengan data numerikal. Contoh data kategorikal dapat dilihat pada Gambar 1a yang memiliki banyak variasi dan data numerikal pada Gambar 1b yang hanya memiliki 2 variasi.



Gambar 1. Hasil *Exploratory Data Analysis*

3.1.2. *Handling Outliers*

Tahap ini dilakukan untuk memeriksa data-data yang terlalu berbeda dari data lainnya dalam sebuah *dataset* (*outlier*). Mengidentifikasi dan menangani *outliers* adalah bagian penting dalam *data preprocessing* untuk memastikan pemodelan yang lebih akurat.

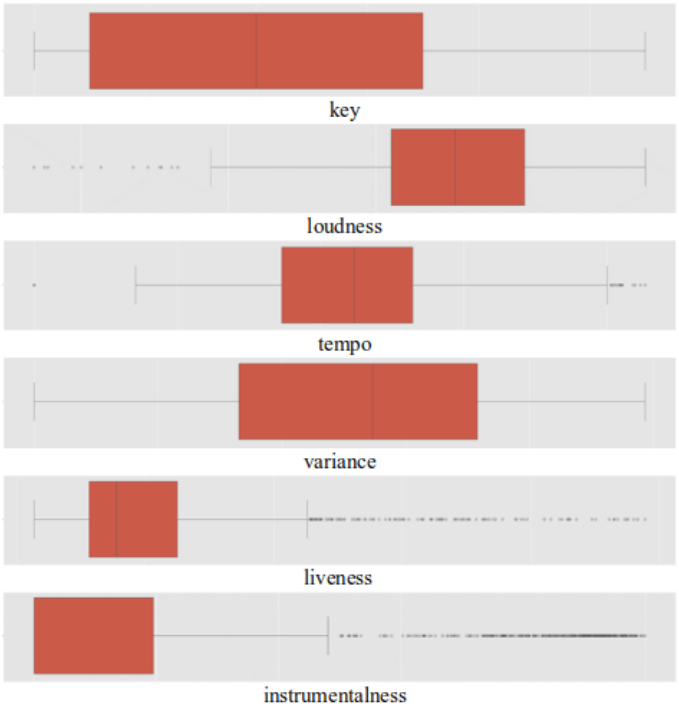
3.1.3. *Checking Outliers*

Pemeriksaan *outlier* dapat dilakukan dengan menampilkan data menggunakan *boxplot* untuk menggambarkan bentuk data dari kolom tertentu. Hasil dari pemeriksaan *outlier* data setiap kolom fitur terdapat pada Gambar 2.

Berdasarkan Gambar 2, *boxplot* menunjukan adanya *outlier* pada kolom fitur *'loudness', 'tempo', 'liveness', 'instrumentalness', 'time_signature', 'energy', 'popularity', 'explicit', 'danceability', 'duration_ms'*. Data-data kolom fitur yang memiliki *outlier* dapat dipastikan menggunakan cara lain yaitu *Interquartile Range*

Idealis: Indonesia Journal Information System
Volume 8, Nomor 1, Januari 2025
ISSN 2684-7280 (*online*)
Halaman 74-83
available online at http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index

(IQR). Hasil dari IQR dapat menunjukan seluruh data yang berada diluar dari batas bawah dan batas atas dari sebuah kolom fitur. Contoh untuk hasil dari IQR yang memiliki outlier, serta yang tidak memiliki IQR.



Gambar 2. Hasil IQR

3.1.4. *Impute Outlier*

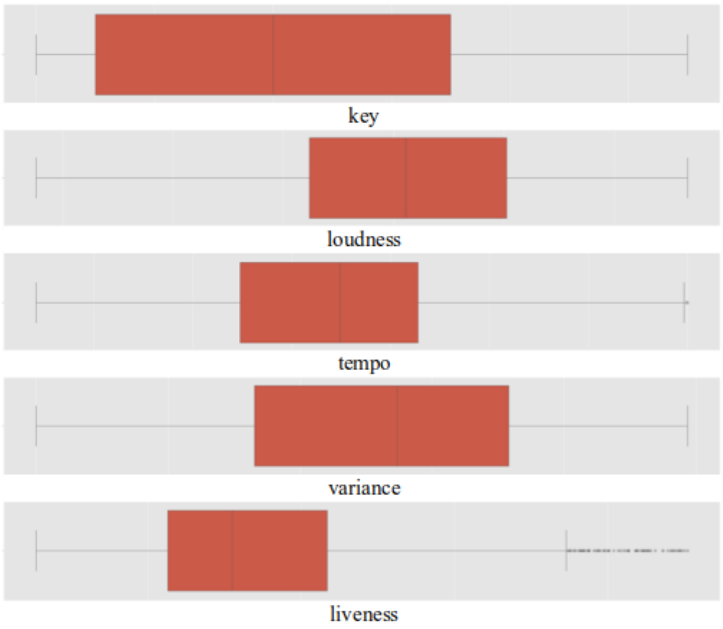
Setelah menentukan tiap kolom fitur yang memiliki *outlier*. *Outlier* akan dihapus dan diubah menjadi data kosong atau *NaN*, lalu akan diperiksa kemiringan dari tiap kolom untuk menentukan data akan diiisi menggunakan perintah *mean()* yang memiliki syarat $-2 < x < 2$ atau *median()* dengan syarat $x > 2$ atau $x < -2$.

```
key                0.300724
loudness          -0.054201
tempo              0.366253
valence           -0.186028
liveness           1.011570
instrumentalness   4.325029
acousticness      -0.995227
speechiness        0.905250
mode              -1.022675
time_signature     0.000000
energy             0.787301
explicit           0.000000
popularity         1.132650
danceability      -0.595138
duration_ms        0.333162
dtype: float64
```

Gambar 3. Nilai kemiringan fitur

Berdasarkan hasil pemeriksaan kemiringan fitur pada Gambar 3, maka dapat menentukan bahwa kolom fitur yang akan diisi menggunakan perintah *median()* adalah *instrumentalness*, dan kolom fitur yang akan diisi menggunakan perintah *mean()* adalah *'loudness', 'tempo', 'liveness', 'time_signature', 'energy', 'popularity', 'explicit', 'danceability', 'duration_ms'*. Hasil dari *impute outlier* dapat dilihat menggunakan *boxplot.*

Idealis: Indonesia Journal Information System
Volume 8, Nomor 1, Januari 2025
ISSN 2684-7280 (*online*)
Halaman 74-83
available online at http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index

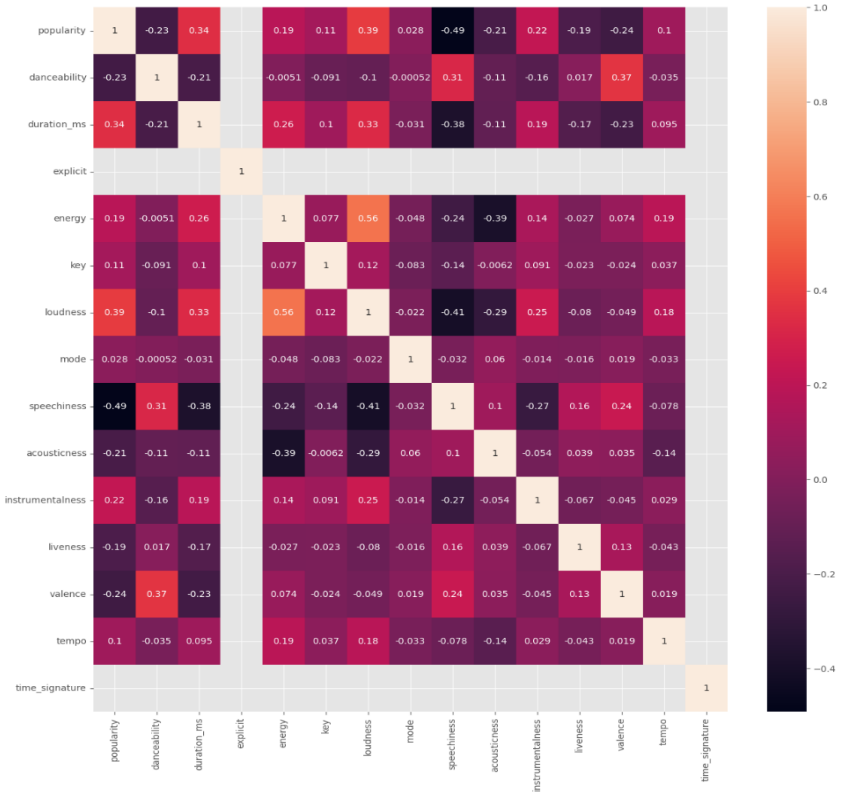


Gambar 4. Hasil IQR *Impute Outlier*

Hasil dari *impute outlier* yang dapat dilihat pada Gambar 4, menunjukan bahwa sebagian besar *outlier* telah hilang dan *dataset* sudah dapat gunakan pada tahap selanjutnya yaitu *feature selection*.

3.1.5. *Feature Selection*

Tahap ini dilakukan pada *dataset* untuk memotong kolom fitur yang tidak berdampak signifikan terhadap kualitas *dataset*.



Gambar 5. Gambar *Heatmap* untuk *Feature Selection*

Idealis: Indonesia Journal Information System
Volume 8, Nomor 1, Januari 2025
ISSN 2684-7280 (*online*)
Halaman 74-83
available online at http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index

Feature yang akan dihapus (di-*drop*) adalah *feature* yang memiliki rata-rata nilai *heatmap* terendah. Berdasarkan perhitungan, *feature* yang akan dihapus (*drop*) berdasarkan *heatmap* adalah *feature 'key', 'mode', 'liveness', 'valence', dan 'tempo'*. Selanjutnya adalah *feature selection* menggunakan *variance threshold*. *Variance threshold* digunakan untuk melakukan *feature selection* pada data yang bertipe kategorikal.

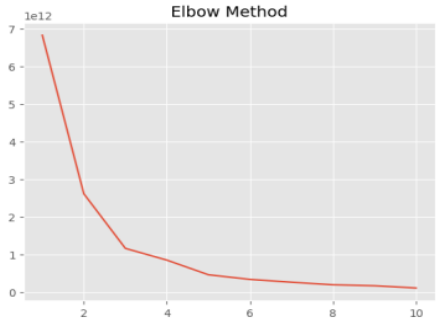
```
Feature: popularity, Variance: 2.09e+02
Feature: duration_ms, Variance: 3.41e+09
Feature: explicit, Variance: 0.00e+00
Feature: danceability, Variance: 2.63e-02
Feature: energy, Variance: 3.19e-02
Feature: key, Variance: 1.22e+01
Feature: loudness, Variance: 3.20e+01
Feature: mode, Variance: 1.98e-01
Feature: speechiness, Variance: 1.62e-01
Feature: acousticness, Variance: 9.17e-02
Feature: instrumentalness, Variance: 4.15e-03
Feature: liveness, Variance: 7.82e-03
Feature: valence, Variance: 5.85e-02
Feature: tempo, Variance: 9.20e+02
Feature: time_signature, Variance: 0.00e+00
```

Gambar 6. *Variance Threshold* fitur

Berdasarkan hasil perhitungan *variance threshold* pada Gambar 6, *feature* '*explicit'* dan *'time_signature'* memiliki nilai 0 (nol). Hal ini menunjukan bahwa kedua *feature* tersebut tidak memiliki dampak signifikan terhadap *dataset*. Sehingga, *feature* yang dihapus pada *dataset* ini adalah *'key', 'mode', 'liveness', 'valence', 'explicit', 'time_signature', dan 'tempo'*. Sehingga *dataset* yang telah dipotong hanya memiliki kolom fitur.

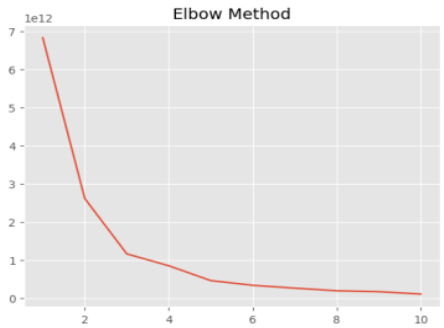
3.2. Building Model

Selanjutnya adalah membuat model *machine learning* menggunakan algoritma *K-Means* kepada *dataset* yang sudah diolah (*untrimmed*) dan *dataset* yang sudah diolah dan dipotong (*trimmed*). *K-Means* merupakan model *unsupervised*, yang berarti harus ditentukan jumlah klaster (*cluster*) yang terbaik bagi data tersebut. Jumlah *cluster* terbaik dapat ditentukan menggunakan *Elbow Method* dan *Silhouette Scores*. Hasil dari *Elbow Method* dan *Silhouette Scores* dapat dilihat pada gambar 7a, b, c, dan d.

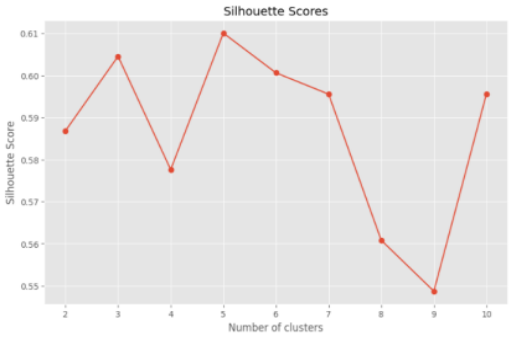


(a) Grafik *Elbow Method Untrimmed*

(b) Grafik *Silhouette Scores Untrimmed*



(c) Grafik *Elbow Method Trimmed*



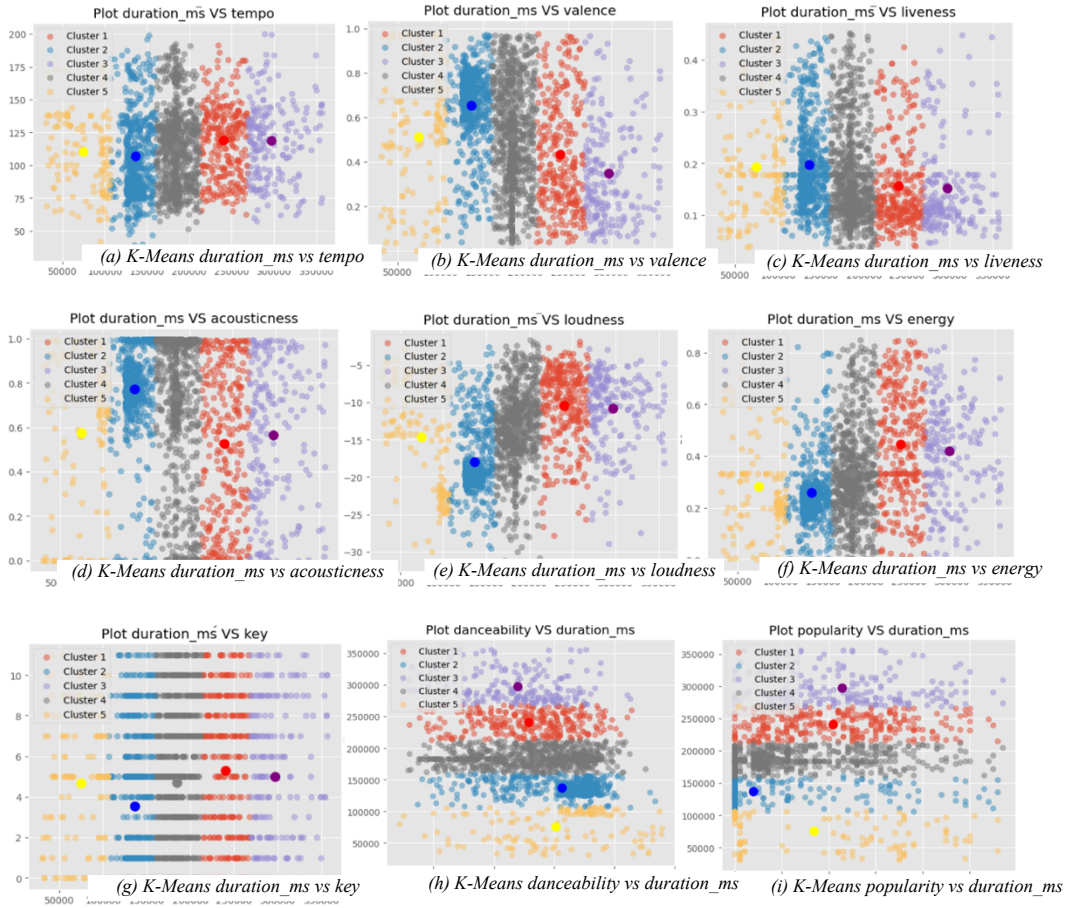
(d) Grafik *Silhouette Scores Trimmed*

Gambar 7. Grafik hasil *Elbow Method* dan *Silhouette Scores*

Idealis: Indonesia Journal Information System
Volume 8, Nomor 1, Januari 2025
ISSN 2684-7280 (*online*)
Halaman 74-83
available online at http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index

Berdasarkan Gambar 7a, b, c, dan d, nilai *k* (jumlah *cluster*) yang terbaik adalah 5 untuk kedua *dataset* (data *untrimmed* dan data *trimmed*). Jumlah *cluster* 5 merupakan jumlah *cluster* disebabkan tidak perubahan yang signifikan pada grafik setelah *k* bernilai pada grafik *Elbow Method*. Grafik *Sihlouette Score* juga menunjukan hasil yang sama, yaitu jumlah *cluster* terbaik adalah 5 *cluster*.

Berdasarkan visualisasi data pada Gambar 8, terdapat beberapa pasang data yang memiliki penyebaran data yang rapih, mayoritas dari visualisasi yang rapih tersebut memiliki *feature 'duration_ms'* didalamnya. Visualisasi ini membuktikan bahwa *feature 'duration_ms'* memiliki dampak yang besar terhadap *feature* lainnya pada *dataset* yang diberikan. Hal ini juga dapat dilihat dari visualisasi data untuk data yang *trimmed*.



(a) K-Means duration_ms vs tempo *(b) K-Means duration_ms vs valence* *(c) K-Means duration_ms vs liveness*

(d) K-Means duration_ms vs acousticness *(e) K-Means duration_ms vs loudness* *(f) K-Means duration_ms vs energy*

(g) K-Means duration_ms vs key *(h) K-Means danceability vs duration_ms* *(i) K-Means popularity vs duration_ms*

Gambar 8. Hasil *K-Means Clustering* per fitur

Berdasarkan Gambar 8a sampai dengan 8i, terdapat beberapa pasang data yang memiliki penyebaran data yang rapih, mayoritas dari visualisasi yang rapih tersebut juga memiliki *feature 'duration_ms'* didalamnya. Visualisasi diatas memiliki jumlah visualisasi data rapih yang lebih sedikit. Hal ini disebabkan karena beberapa *feature* yang tidak memiliki dampak signifikan pada *heatmap* sudah di-*drop*.

3.3. Evaluasi

Berdasarkan *Silhouette Scores*, model yang dibuat mendapatkan angka 0,53 untuk data *untrimmed* dan *trimmed*. *Silhouette Scores* memiliki rentang nilai dari -1 sampai 1, berarti hasil ini menjelaskan bahwa model yang telah dibuat sudah menunjukan kualitas *clustering* yang baik

3.4. Hasil Analisis

Berdasarkan penelitian yang dilakukan, penelitian ini dapat berguna untuk diimplementasikan ke dunia nyata (*real case scenario*). Penelitian ini sangat membantu bagi aplikasi *streaming* musik seperti *Spotify* atau *Youtube Music*. Berikut adalah beberapa hasil analisis yang dapat diterapkan:

1. Rekomendasi musik. Hasil *clustering* pada *dataset* menunjukkan bahwa fitur *duration_ms* merupakan faktor utama dalam pengelompokan lagu. Namun, fitur lain seperti *tempo, valence, liveness,* dan *energy*

Idealis: Indonesia Journal Information System
Volume 8, Nomor 1, Januari 2025
ISSN 2684-7280 (*online*)
Halaman 74-83
available online at http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index

juga berkontribusi dalam menentukan preferensi pengguna. Lagu-lagu dibagi menjadi lima *cluster* berdasarkan durasi sebagai berikut :

1. *Cluster 1* (Kuning): Lagu dengan durasi pendek (50.000–100.000 ms), memiliki tempo cepat dan energy tinggi, cocok untuk aktivitas olahraga atau motivasi. Biasanya lagu-lagu di *cluster* ini juga memiliki *valence* tinggi (*mood booster*).
2. *Cluster 2* (Biru): Lagu dengan durasi menengah-pendek (100.000–150.000 ms), sering kali memiliki *acousticness* tinggi dan *liveness* rendah, cocok untuk suasana santai seperti mendengarkan di rumah atau perjalanan panjang.
3. *Cluster 3* (Abu-Abu): Lagu dengan durasi sedang (150.000–200.000 ms), memiliki keseimbangan antara *danceability* dan *popularity*, sering kali ditemukan dalam lagu-lagu pop *mainstream*.
4. *Cluster 4* (Merah): Lagu dengan durasi panjang-menengah (200.000–250.000 ms), memiliki *loudness* tinggi dan *key* yang bervariasi, sering digunakan dalam genre seperti *rock*.
5. *Cluster 5* (Ungu): Lagu dengan durasi panjang (300.000–350.000 ms), memiliki kombinasi tempo lambat dan *acousticness* tinggi, cocok untuk lagu-lagu atau *soundtrack* dramatis.

Dengan memahami *clustering* ini, *platform* musik dapat merekomendasikan lagu berdasarkan preferensi durasi pengguna seperti pengguna yang sering mendengarkan lagu pendek akan diberikan rekomendasi lagu-lagu yang memiliki durasi yang serupa dari *cluster* yang sama.

2. *Cluster* yang terbentuk juga dapat membantu dalam menciptakan *playlist* otomatis berdasarkan durasi. Sebagai contoh, *playlist "Mood Booster"* dapat berisi lagu-lagu dari *Cluster* 1 dan 2, sedangkan *playlist "Dramatic Vibes"* dapat berisi lagu-lagu dari *Cluster* 5. Hal ini memastikan pengalaman mendengarkan yang lebih akurat dengan pergantian music yang sesuai antar lagu.
3. Analisis tren musik. Mengelompokan fitur *duration_ms* sebagai indikator utama, platform dapat menganalisis tren musik dari waktu ke waktu. Misalnya, peningkatan jumlah lagu dalam *Cluster* 3 dan 4 mungkin menunjukkan minat pendengar terhadap lagu-lagu dengan durasi sedang. Data ini dapat membantu pembuat musik untuk menyesuaikan karya mereka dengan tren musik di pasar saat ini.

4. KESIMPULAN

Penelitian ini berhasil menunjukan dan membuat model *machine learning* menggunakan algoritma *K-Means* dapat diterapkan dengan efektif mengelompokan musik berdasarkan fitur (*feature*) yang terdapat pada sebuah musik. Algoritma *K-Means* mampu mengelompokan musik-musik yang sudah diberikan pada *dataset* berdasarkan fitur yang memiliki dampak yang signifikan dengan algoritma *K-Means* menggunakan *feature selection*. Penelitian ini juga menunjukan bahwa algoritma *K-Means* merupakan algoritma yang efisien dan dapat diimplementasikan dengan *dataset* yang diberikan. Penelitian ini menunjukan bahwa algoritma *K-Means* merupakan solusi yang dapat digunakan untuk aplikasi musik atau *platform streaming*. Implementasi *machine learning* dalam pengelompokan lagu menggunakan algoritma *K-Means* membuka peluang untuk inovasi dalam industri musik digital. Dengan pengelompokan yang akurat dan efisien, sistem rekomendasi musik dapat dibuat lebih personal dan relevan bagi pengguna. Penelitian ini memberikan dasar yang kuat untuk pengembangan lebih lanjut, termasuk eksplorasi kombinasi algoritma *K-Means* dengan teknik *machine learning* lainnya untuk meningkatkan performa dan akurasi pengelompokan.

DAFTAR PUSTAKA

[1] A. S. Marwi, I. R. Lubis, Y. Sinurat, S. W. Ulfa, and T. H. B. Nainggolan, "PENGARUH MEDIA MUSIK DAN LAGU DALAM PEMBELAJARAN BIOLOGI," *Sinar Dunia: Jurnal Riset Sosial Humaniora dan Ilmu Pendidikan*, vol. 2, no. 1, pp. 74–86, Jan. 2023, doi: 10.58192/sidu.v2i1.507.
[2] D. Hesmondhalgh, "Streaming's Effects on Music Culture: Old Anxieties and New Simplifications," *Cult Sociol*, vol. 16, no. 1, pp. 3–24, Mar. 2022, doi: 10.1177/17499755211019974.
[3] Y. Chen, "Automatic Classification and Analysis of Music Multimedia Combined with Hidden Markov Model," *Advances in Multimedia*, vol. 2021, pp. 1–7, Dec. 2021, doi: 10.1155/2021/7824001.
[4] A. P. Thenata and M. Suryadi, "Machine Learning Prediction of Anxiety Levels in the Society of Academicians During the Covid-19 Pandemic," *Jurnal Varian*, vol. 6, no. 1, pp. 81–88, Nov. 2022, doi: 10.30812/varian.v6i1.2149.
[5] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means Algorithm: A Comprehensive Survey and Performance Evaluation," *Electronics (Basel)*, vol. 9, no. 8, p. 1295, Aug. 2020, doi: 10.3390/electronics9081295.
[6] D. B. Rarasati, "A Grouping of Song-Lyric Themes Using K-Means Clustering," *JISA(Jurnal Informatika dan Sains)*, vol. 3, no. 2, Dec. 2020, doi: 10.31326/jisa.v3i2.658.
[7] F. Apit, "MACHINE LEARNING UNTUK PENDIDIKAN : MENGAPA DAN BAGAIMANA," *Jurnal Informatika Dan Tekonologi Komputer*, vol. 1, no. 3, pp. 57–62, Nov. 2021.

[8] S. Naeem, A. Ali, S. Anam, and M. M. Ahmed, "An Unsupervised Machine Learning Algorithms: Comprehensive Review," *International Journal of Computing and Digital Systems*, vol. 13, no. 1, pp. 911–921, Apr. 2023, doi: 10.12785/ijcds/130172.

[9] Md. K. Hasan, Md. A. Alam, S. Roy, A. Dutta, Md. T. Jawad, and S. Das, "Missing value imputation affects the performance of machine learning: A review and analysis of the literature (2010–2021)," *Inform Med Unlocked*, vol. 27, p. 100799, 2021, doi: 10.1016/j.imu.2021.100799.

[10] A. Alabrah, "An Improved CCF Detector to Handle the Problem of Class Imbalance with Outlier Normalization Using IQR Method," *Sensors*, vol. 23, no. 9, p. 4406, Apr. 2023, doi: 10.3390/s23094406.

[11] N. Anwar, F. Adikara, R. Setiyati, R. Satria, and A. Satriawan, "Data Mining Menggunakan Metode Algoritma Apriori Pada Vending Machine Product Display," *JBASE - Journal of Business and Audit Information Systems*, vol. 4, no. 2, Aug. 2021, doi: 10.30813/jbase.v4i2.3004.

[12] M. A. H. Umar and B. Sitohang, "ANALISIS FAKTOR-FAKTOR YANG MEMENGARUHI KEPUTUSAN PEMBELIAN PAKET WISATA MENGGUNAKAN MODEL KLASIFIKASI DECISION TREES, RANDOM FOREST DAN K-NEAREST NEIGHBOURS," *Journal of Social and Economics Research*, vol. 6, no. 2, pp. 25–39, Aug. 2024, doi: 10.54783/jser.v6i2.590.

[13] C. S. D. B. Sembiring, L. Hanum, and S. P. Tamba, "PENERAPAN DATA MINING MENGGUNAKAN ALGORITMA K-MEANS UNTUK MENENTUKAN JUDUL SKRIPSI DAN JURNAL PENELITIAN (STUDI KASUS FTIK UNPRI)," *Jurnal Sistem Informasi dan Ilmu Komputer Prima(JUSIKOM PRIMA)*, vol. 5, no. 2, pp. 80–85, Feb. 2022, doi: 10.34012/jurnalsisteminformasidanilmukomputer.v5i2.2393.

[14] M. Guntara and N. Lutfi, "Optimasi Cacah Klaster pada Klasterisasi dengan Algoritma KMeans Menggunakan Silhouette Coeficient dan Elbow Method," *JuTI "Jurnal Teknologi Informasi,"* vol. 2, no. 1, p. 43, Aug. 2023, doi: 10.26798/juti.v2i1.944.

[15] D. Haversyalapa, S. Puspasari, and R. Gustriansyah, "KLASTERISASI PIXEL CITRA KOLEKSI FOTO MUSEUM MONPERA DENGAN METODE K-MEANS PADA APLIKASI AUGMENTED REALITY," *IDEALIS: InDonEsiA journaL Information System*, vol. 7, no. 2, pp. 189–199, Jun. 2024, doi: 10.36080/idealis.v7i2.3175.

[16] A. R. Lashiyanti, I. R. Munthe, and F. A. Nasution, "Optimisasi Klasterisasi Nilai Ujian Nasional dengan Pendekatan Algoritma K-Means, Elbow, dan Silhouette," *Jurnal Ilmu Komputer dan Sistem Informasi (JIKOMSI)*, vol. 6, no. 1, pp. 14–20, Mar. 2023, [Online]. Available: https://ejournal.sisfokomtek.org/index.php/jikom/article/view/1550

[17] F. M. Ilyas and S. S. Priscila, "An Optimized Clustering Quality Analysis in K-Means Cluster Using Silhouette Scores," 2024, pp. 49–63. doi: 10.4018/979-8-3693-1355-8.ch004.

[18] R. F. T. Wulandari and D. Anubhakti, "IMPLEMENTASI ALGORITMA SUPPORT VECTOR MACHINE (SVM) DALAM MEMPREDIKSI HARGA SAHAM PT. GARUDA INDONESIA TBK," *IDEALIS: InDonEsiA journaL Information System*, vol. 4, no. 2, pp. 250–256, Jul. 2021, doi: 10.36080/idealis.v4i2.2847.