# K-Means Centroid Optimization with Genetic Algorithm for Clustering Micro, Small, Medium Enterprises in Yogyakarta

Muhammad Faris Akbar[1*], Lisna Zahrotun[2]

[1,2] *Department of Industrial Engineering, Ahmad Dahlan University, Indonesia*

*corr-author: muhammad2100018169@webmail.uad.ac.id

**Abstract --K-Means is a widely used data clustering algorithm due to its simplicity and fast performance. However, the weakness of K-Means is in determining the cluster centroid randomly, which can result in suboptimal clustering results, especially since it tends to get stuck on local solutions. This research aims to overcome this weakness by integrating the Genetic Algorithms (GA) into the K-Means process, optimizing the initial centroid, and improving clustering quality. The method combines GA with K-Means on MSME data in Yogyakarta, where GA rearranges the cluster's initial centroid more optimally. The results showed that this method reduced the average value of the Davies-Bouldin Index (DBI) from 1,819 in conventional K-Means to 1,349 with GA integration, indicating an improvement in cluster quality by 25.9%. These results prove that integration of GA with K-Means improves clustering accuracy and improves cluster separation, as measured by a significant decrease in DBI value.**

**Keywords: Genetic Algorithm, K-Means, Optimization, Davies-Bouldin Index, MSMEs**

## I. INTRODUCTION

Micro, Small, and Medium Enterprises (MSMEs) play a crucial role in absorbing labor in Indonesia [1]. In 2023, 66 million MSMEs employed 117 million people, equivalent to 97% of the total national workforce and 61% of Indonesia's gross domestic product [2]. Despite their significant contributions, MSMEs often face major challenges, including limited access to capital, inadequate human resources, lack of management skills, and insufficient market insight [1]. Addressing these issues is essential, especially considering the increasing number of MSMEs. One effective approach to analyze and assist MSMEs is through clustering. Clustering is a technique that groups data objects into distinct clusters with high internal similarity [3,4]. It is often used for classifying unlabeled data with minimal supervision [5]. Previous research has extensively applied clustering to MSME data, though results have shown room for improvement.

For instance, researchers applied the Agglomerative Hierarchical Clustering (AHC) method using the Complete Linkage approach to analyze the distribution of service sector MSMEs in Yogyakarta City, forming 2 clusters with a silhouette coefficient score of 0.729, reflecting moderate cluster quality [6]. In another study, the Average Linkage approach of AHC was used to cluster fashion and craft sector MSMEs in Yogyakarta City, achieving silhouette coefficient scores of 0.64 and 0.65, respectively, suggesting moderate cluster structures [7]. Furthermore, the Single Linkage approach of AHC was utilized to cluster culinary and craft sector MSMEs in Yogyakarta City, with the culinary sector forming 2 clusters, achieving a silhouette score of 0.79, which indicates a strong cluster structure with high intra-cluster similarity and low inter-cluster similarity. The craft sector, on the other hand, formed 3 clusters with a silhouette score of 0.615, representing a medium structure with moderate cohesion and separation [8]. Meanwhile, the K-Medoids method was applied to cluster MSMEs in the culinary sector, resulting in 2 clusters with a silhouette coefficient score of 0.60, suggesting a weak to moderate structure with lower cluster distinctiveness [9]. While these clustering methods have provided valuable insights, further improvements can be made using optimization algorithms. Therefore, this study will apply the K-Means algorithm, optimized using Genetic Algorithms (GA), to enhance clustering performance.

K-Means algorithm, introduced by Mac Queen in 1967, is widely acknowledged for its scalability in clustering large datasets. However, its reliance on random initialization of cluster centroids often leads to poor accuracy and local optima [10,11]. To address this, GA offers a robust optimization solution inspired by the principles of natural evolution. Originally introduced by John Holland in 1970, GA applies selection, crossover, and mutation operations to refine centroids iteratively,

minimizing the risk of local optima and enhancing cluster quality. With its proven effectiveness in various applications, GA remains a powerful tool for optimizing clustering performance [12,13].

Several studies have demonstrated the effectiveness of using GA to optimize K-Means. One study applied GA to determine initial cluster centroids, significantly reducing the SSE value compared to conventional K-Means [1]. Another study showed that K-Means optimized with GA improved clustering quality by 54.9% for high-dimensional data and 52.4% for dimension-reduced data [4,14]. Additionally, in clustering clean water users in Riau Province, K-Means optimized with GA achieved a lower Davies-Bouldin Index (DBI) of 2.06894 compared to 2.164763 from conventional K-Means, demonstrating its superiority [11]. Based on the success of previous research, this study aims to determine the extent of Genetic Algorithm optimization in clustering MSME data in Yogyakarta to optimize the value of the cluster center point in the K-Means algorithm.

## II. METHOD

To achieve the research objectives, the activities of this study can be illustrated as shown in Fig. 1.

### A. Data Collection

The dataset utilized in this research comprises secondary data sourced from the Yogyakarta City Industry, Trade, and Cooperative Office, provided in Excel format. It includes information on 5 MSME sectors with 40 variables/attributes, aggregating 2.769 data from 2021 to 2022. For this study, however, only the service sector, which consists of 399 data, is considered.

### B. Data Preprocessing

Data preprocessing is an essential phase that encompasses techniques applied before implementing data mining methods. This stage is crucial for extracting meaningful insights from data [15]. The data preprocessing stages carried out in this study are as follows:

*Data Cleaning:* This involves the identification and rectification of incorrect or noisy data, as well as the removal of such data from the dataset. Key activities in data cleaning include removing irrelevant or duplicate entries, correcting structural errors, and addressing missing values [16].

*Standardization:* In the context of data preprocessing, standardization refers to the process of transforming values within a dataset to a uniform format. This process entails grouping similar values, assigning canonical values to each group, and substituting the original values to create a standardized dataset, facilitating more effective analysis [17]. Data standardization encompasses establishing standards related to the entire data value chain. These standards may pertain to various aspects, including the attributes of the data being collected, terminology, dataset structure, organization, and data storage and its application [18].

*Data Transformation:* This refers to the process of converting data from one format or structure to another. This transformation process is essential for ensuring that the data aligns more closely with the assumptions required for statistical inference procedures or for enhancing its interpretability [19].

*Feature Selection:* This involves the identification and elimination of irrelevant and redundant information [15]. As a result of this process, 12 variables were selected from a pool of 40 available attributes. The selected variables include: last education, business activity, export commodity products, marketing objectives, land/building ownership status, electronic media facilities, government-assisted capital, public business credit loans, turnover per year, health insurance ownership, male labor, and female labor.

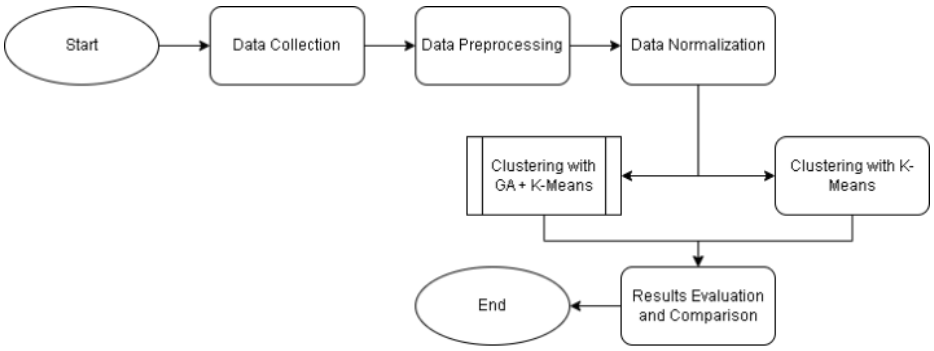


**Fig. 1 Research framework**

### C. Data Normalization

Data that has gone through the preprocessing stage will proceed to the normalization stage using the min-max normalization method. This technique converts the range of values in the dataset into a more interpretable scale, specifically from 0 to 1 [7]. Min-max normalization adjusts each feature to fit within a specified range, which is defined by two parameters: the lower limit and the upper limit [15]. The equation for min-max normalization is provided in (1).

$$Vn' = \frac{Vn - Min}{Max - Min} \quad (1)$$

Description:
$Vn'$ = Data
$Vn$ = Data value
$Min$ = Minimum value of data
$Max$ = Maximum value of data

### D. Clustering With K-Means

K-Means algorithm is one of the non-hierarchical cluster analysis methods in which the number of groups to be formed is predetermined [20,21]. While K-Means has no significant weaknesses, it is influenced by the initial data selection, which can sometimes lead to local optima. However, K-Means excels in rapid convergence and quality of clustering [1,4]. The stages of the K-Means algorithm are as follows:

- Determine the number of clusters, k.
- Establish the initial center for each cluster. This initial center is chosen from the existing data by shuffling the dataset. Next, calculate the central point for the subsequent cluster.
- Calculate the distance between the data points and the cluster centers using Euclidean distance. The outcomes of this distance calculation will be compared, and the closest distance will be selected to indicate that the data point belongs to the group with the nearest cluster center. The formula for Euclidean distance is provided in (2).

$$d(X_{2j}, X_{1j}) = \sqrt{\sum_{j=1}^{n} (X_{ki} - X_{kj})^2} \quad (2)$$

Description:
$d(X_{2j}, X_{1j})$ = Euclidean distance
$X_{ki}$ = Data to $i$ on the $k$ data attribute
$X_{kj}$ = Center point to $j$ on the $k$ data attribute

- Group the data. Once some data points are identified as close to one of the center points, they will be assigned to the class corresponding to that center point.
- Iterate and update the center point positions using (3).

$$V_{ij} = \frac{1}{N_i} \sum_{k=0}^{n} X_{kj} \quad (3)$$

Description:
$V_{ij}$ = Average point of the $i$ cluster for the $j$ variable
$N_i$ = Number of data that are members of the $i$ cluster
$X_{kj}$ = The $k$ data value in the cluster for the $j$ variable

- Repeat step 3 if there are still data points that switch groups or if the center point values change beyond a specified threshold value.

### E. Clustering With GA+K-Means

Genetic Algorithms (GA) are search algorithms that mimic the same natural selection mechanism as algorithms used in solving complex optimization problems. They are widely used in business, technical, scheduling, and other applications. GA starts by retrieving solutions based on a population that will be used to form a new population. A better population will be selected to form a new solution based on the best fitness value [1,22]. In general, the stages of the GA can be described as follows: Chromosome representation; Initialization of the initial population; Fitness value evaluation; Selection, Crossover, Mutation; Generating a new population; Criterion checking based on fitness value; Best individual.

In this study, GA will use the centroid of the K-Means algorithm as the initial chromosome in the centroid optimization stage. Fig. 2 shows the framework of K-Means centroid optimization with Genetic Algorithms.

*1) Chromosome Representation with K-Means Centroid:* At this stage, chromosomes are generated using the centroids obtained from the clustering results of the K-Means algorithm. The purpose is to maintain the cluster structure while facilitating the exploration of solution space.

*2) Initialization Centroid Population:* The initial chromosomes obtained will be redefined to align with the target population size of 500. This population initialization process aims to form a diverse and well-distributed set of initial cluster centers. This approach is very important as it enhances the exploration of the search space, facilitating more effective clustering results. Diversity in the centroid population prevents local optima and encourages a more comprehensive search for optimal clusters.

*3) Fitness Evaluation:* Once the population is generated, the next stage involves fitness evaluation using the Davies-Bouldin Index (DBI) as the fitness

function. This function assesses the validity of the clustering, where a lower DBI value indicates a better clustering structure. A penalty is introduced to prevent invalid data solutions from being grouped into a single cluster by giving a very high fitness value when only one unique cluster label exists. The fitness function equation is provided in (4).

$$Fitness = min(DBI) \quad (4)$$

Description:
$Fitness$ = Fitness value
$min(DBI)$ = Minimum value of Davies-Bouldin Index

*4) Selection Process:* The selection process utilizes the Tournament Selection method, featuring a tournament size of 3 individuals. This approach was chosen for its ability to adjust the selection bias dynamically according to the evolutionary process. When the population predominantly comprises individuals with low fitness values, the selection bias is heightened to accelerate convergence toward improved solutions. Conversely, when the population clusters around local optima, the selection bias is reduced to preserve genetic diversity within the population [23,24].

*5) Crossover Operation:* The crossover process utilizes the Two-Point Crossover method, with a crossover probability of 0.8. This approach is selected based on the length of the chromosomes, the nature of the solution space, and the unique characteristics of the individuals involved. By exchanging genetic material between parent chromosomes, the crossover operation promotes a more thorough exploration of the solution space and aids in creating enhanced offspring.

*6) Mutation Operation:* The mutation process utilizes the Gaussian Mutation method, characterized by a normal distribution with a mean of 0 and a standard deviation of 0.1. The mutation probability per gene is set at 0.2, with a mutation value of 0.2. This mutation strategy is chosen for its proven effectiveness in self-adaptive Genetic Algorithms [25]. The mutation operation introduces controlled random variations in centroid positions, thereby preserving population diversity and reducing the risk of premature convergence.

*7) New Centroid Population:* After the selection, crossover, and mutation processes are completed, a new centroid population is created. This population then undergoes another round of fitness evaluation, and the iterative process is repeated until a stopping condition is met. The termination criteria are either reaching the desired fitness value or surpassing the population size limit of 50 generations.

*8) Selection of the Best Individual (New Centroid):* Upon the conclusion of the iterative process, the chromosome exhibiting the highest fitness value is selected as the new centroid. This optimized centroid serves as the refined cluster center, enhancing the clustering process and overall solution quality.

### *F. Result Comparison and Evaluation*

After completing the clustering process with K-Means and GA+K-Means, the next step is to compare and evaluate the results by analyzing both methods' cluster distribution, Davies-Bouldin Index (DBI) values, and computation time. Davies-Bouldin Index (DBI) is a cluster validity method that maximizes the inter-cluster distance and minimizes the distance between points in the cluster [13]. The foundation of the DBI method is the value of cohesion and separation [3]. The smaller the DBI value produced, the more optimal the cluster results [26]. The steps in calculating the DBI value are as follows:

- Calculate the Sum of Square Within Cluster (SSW) value by using (5).

$$SSWi = \frac{1}{mi} \sum_{j=1}^{mi} d(xj, ci) \quad (5)$$

Description:
$SSWi$ = Average sum of distances of all data points in the cluster $i$ to the cluster center $ci$
$xj$ = The $j$-th data point in the cluster $i$
$ci$ = Cluster centroid $i$
$mi$ = Number of data points in the cluster $i$
$d(xj, ci)$ = Distance between data point $xj$ and cluster center $ci$

- Calculate the Sum of Square Between Clusters (SSB) value by using (6).

$$SSBi,j = d(ci, cj) \quad (6)$$

Description:
$SSBi,j$ = Sum of squares between cluster $i$ and $j$ values
$d(ci, cj)$ = Distance between cluster center $ci$ and cluster center $cj$

- Calculate the Ratio value by using (7).

$$Ri,j = \frac{SSWi + SSWj}{SSBi,j} \quad (7)$$

Description:
$Ri,j$ = Ratio of dispersion within clusters $i$ and $j$ to the distance between clusters $i$ and $j$
$SSWi$ = Value of Sum of Square Within cluster $i$
$SSWj$ = Value of Sum of Square Within cluster $j$
$SSBi,j$ = Value of Sum of Squares between clusters for clusters $i$ and $j$

- Calculate the Davies-Bouldin Index value by using (8).

$$DBI = \frac{1}{k}\sum_{i=1}^{k} \max_{i \neq j}(Ri, j) \quad (8)$$

Description:
$DBI$ = Davies-Bouldin Index value
$k$ = Total number of cluster
$\max_{i \neq j}(Ri, j)$ = The largest ratio value between cluster $i$ and $j$, which is calculated for cluster $j$ with $i \neq j$

## III. RESULT AND DISCUSSION

In the results and discussion stages, the clustering results, the Davies-Bouldin index value, and the execution time of the two algorithms in clusters 2 to 10 will be compared. However, before conducting the tests, it is necessary to normalize the pre-processed data. The results of this normalization stage can be seen in Fig. 3.

### A. Genetic Algorithm Stage Results

The results of the Genetic Algorithm stage will explain the results of each algorithm process that occurs in this study. Using an example of 2 clusters, the following explains each process stage.

*1) Chromosome Representation with K-Means Centroid:* As explained in the method stage, the chromosome representation in this study will use the centroid generated by K-Means as the initial initialization of the chromosome. For example, if we have 2 centroids, then we will have a chromosome that combines the 2 centroids. An example of this chromosome initialization process can be seen in Table I.

*2) Initialization Centroid Population:* After the initial chromosomes are obtained, the next step is reinitializing them to create a population of chromosomes. Table II provides an overview of this chromosome population.

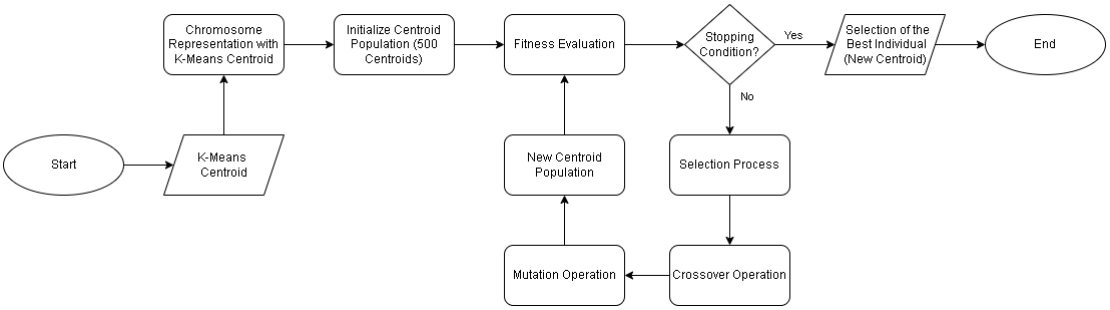


**Fig. 2 K-Means centroid optimization framework with Genetic Algorithms**

| | Last Education | Business Activities | Export Commodity Products | Marketing Objectives | Land/Building Ownership Status | Electronic Media Facilities | Government-Assisted Capital | Public Business Credit Loans | Turnover Per Year | Health Insurance Ownership | Male Labour | Female Labour |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.8 | 0.5 | 0.0 | 0.250 | 1.000000 | 0.333333 | 0.0 | 0.00 | 0.000000 | 0.0 | 0.2 | 0.2 |
| 1 | 0.0 | 0.0 | 0.0 | 0.125 | 0.000000 | 0.833333 | 0.0 | 0.00 | 0.000000 | 0.5 | 0.1 | 0.0 |
| 2 | 0.6 | 0.5 | 0.0 | 0.000 | 0.000000 | 0.333333 | 0.0 | 0.00 | 0.000000 | 0.0 | 0.0 | 0.0 |
| 3 | 1.0 | 0.5 | 0.0 | 0.875 | 0.333333 | 0.333333 | 1.0 | 0.00 | 0.333333 | 0.0 | 0.1 | 0.0 |
| 4 | 1.0 | 0.0 | 0.0 | 0.125 | 0.333333 | 0.833333 | 0.0 | 0.00 | 0.333333 | 0.0 | 0.5 | 0.3 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 380 | 0.6 | 0.5 | 0.0 | 0.750 | 0.333333 | 0.166667 | 0.0 | 0.00 | 0.000000 | 0.0 | 0.0 | 0.3 |
| 381 | 0.4 | 0.5 | 0.0 | 0.000 | 0.333333 | 0.333333 | 0.0 | 0.75 | 0.000000 | 0.5 | 0.0 | 0.0 |
| 382 | 1.0 | 1.0 | 0.0 | 0.000 | 1.000000 | 0.833333 | 1.0 | 0.00 | 0.000000 | 0.0 | 0.1 | 0.0 |
| 383 | 0.6 | 0.5 | 0.0 | 0.125 | 0.666667 | 0.833333 | 1.0 | 0.00 | 0.000000 | 0.0 | 0.1 | 0.0 |
| 384 | 1.0 | 0.0 | 0.0 | 0.000 | 0.000000 | 0.166667 | 1.0 | 1.00 | 0.333333 | 0.5 | 0.0 | 0.0 |

**Fig. 3 Normalized data**

TABLE I
RESULTS OF THE CHROMOSOME INITIALIZATION STAGE

| No | Cluster | Centroids | Chromosomes |
|---|---|---|---|
| 1 | 0 | [0.605, 0.291, 0, 0.229, 0.954, 0.585, 0.266, 0.243, 0.197, 0.339, 0.084, 0.057] | [0.605, 0.291, 0, 0.229, 0.954, 0.585, 0.266, 0.243, 0.197, 0.339, 0.084, 0.057, 0.569, 0.210, 0, 0.193, |
| 2 | 1 | [0.569, 0.210, 0, 0.193, 0.157, 0.585, 0.208, 0.287, 0.184, 0.309, 0.067, 0.04] | 0.157, 0.585, 0.208, 0.287, 0.184, 0.309, 0.067, 0.049] |

TABLE II
RESULTS OF THE INITIALIZATION CENTROID POPULATION STAGE

| No | Index | Value |
|---|---|---|
| 1 | 0 | [0.605, 0.291, 0, 0.229, 0.954, 0.585, 0.266, 0.243, 0.197, 0.339, 0.084, 0.057, 0.569, 0.210, 0, 0.193, 0.157, 0.585, 0.208, 0.287, 0.184, 0.309, 0.067, 0.049] |
| 2 | 1 | [0.821, -0.231, 0.346, 0.158, 0.999, 0.388, 1.576, 0.197, 0.003, 0.769, 0.288, 0.987, 0.783, -0.008, -0.035, -0.020, 0.788, 0.549, 0.907, 0.346, 0.179, 0.380, 0.185, -0.151] |
| 3 | 2 | [0.661, -0.271, 0.318, 0.036, 1.098, 0.552, 1.726, 0.304, -0.018, 0.693, 0.218, 0.246, 0.672, 0.109, 0.000, 0.094, 0.899, 0.513, 0.843, 0.440, 0.277, 0.111, 0.149, -0.034] |

*3) Fitness Evaluation:* Each chromosome generated in the population will be evaluated for its fitness value. Table III shows an example of each chromosome fitness value in the population.

*4) Selection Process:* Chromosomes with the best fitness value will be selected as parent chromosomes of as many as the number of tournament size, namely 3. Because the fitness evaluation used DBI, the best chromosome will be selected with the smallest DBI value. Table IV provides an example of the results of this selection process.

*5) Crossover Operation:* At the crossover stage, all chromosomes, including the chromosome used as the parent chromosome, will be crossed. Because the value used is quite large, at 0.8, 80% of all chromosomes may be crossed. Table V provides an example of the results of this crossover process.

*6) New Centroid Population*: Populations that have passed the mutation process will recalculate their fitness. Chromosomes with poor fitness values will be replaced with child chromosomes (offspring) obtained from the selection process, crossover, and mutation of parent chromosomes so that a new population can be obtained. This new population will be the basis for the next generation to get more optimal results. An example of a representation of this new population can be seen in Table VII

*7) Best Individual:* To get the best individual, the chromosome with the best fitness value is split by dividing its length by the number of clusters and features in the data. Table VIII shows an example of splitting a chromosome into a center point.

TABLE III
RESULTS OF THE FITNESS EVALUATION STAGE

| No | Index | Value | Fitness |
|---|---|---|---|
| 1 | 0 | [0.605, 0.291, 0, 0.229, 0.954, 0.585, 0.266, 0.243, 0.197, 0.339, 0.084, 0.057, 0.569, 0.210, 0, 0.193, 0.157, 0.585, 0.208, 0.287, 0.184, 0.309, 0.067, 0.049] | 0.594 |
| 2 | 1 | [0.821, -0.231, 0.346, 0.158, 0.999, 0.388, 1.576, 0.197, 0.003, 0.769, 0.288, 0.987, 0.783, -0.008, -0.035, -0.020, 0.788, 0.549, 0.907, 0.346, 0.179, 0.380, 0.185, -0.151] | 0.594 |
| 3 | 2 | [0.661, -0.271, 0.318, 0.036, 1.098, 0.552, 1.726, 0.304, -0.018, 0.693, 0.218, 0.246, 0.672, 0.109, 0.000, 0.094, 0.899, 0.513, 0.843, 0.440, 0.277, 0.111, 0.149, -0.034] | 0.594 |

TABLE IV
CHROMOSOME SELECTION RESULTS

| No | Index | Value | Fitness | Best Fitness |
|---|---|---|---|---|
| 1 | 0 | [0.605, 0.291, 0, 0.229, 0.954, 0.585, 0.266, 0.243, 0.197, 0.339, 0.084, 0.057, 0.569, 0.210, 0, 0.193, 0.157, 0.585, 0.208, 0.287, 0.184, 0.309, 0.067, 0.049] | 0.594 | |
| 2 | 1 | [0.821, -0.231, 0.346, 0.158, 0.999, 0.388, 1.576, 0.197, 0.003, 0.769, 0.288, 0.987, 0.783, -0.008, -0.035, -0.020, 0.788, 0.549, 0.907, 0.346, 0.179, 0.380, 0.185, -0.151] | 0.594 | 0.594 |
| 3 | 2 | [0.661, -0.271, 0.318, 0.036, 1.098, 0.552, 1.726, 0.304, -0.018, 0.693, 0.218, 0.246, 0.672, 0.109, 0.000, 0.094, 0.899, 0.513, 0.843, 0.440, 0.277, 0.111, 0.149, -0.034] | 0.594 | |

TABLE V
CHROMOSOME CROSSOVER RESULT

| No | Index 1 | Index 2 | Before Crossover | | After Crossover | |
|---|---|---|---|---|---|---|
| | | | Chromosome 1 | Chromosome 2 | Chromosome 1 | Chromosome 2 |
| 1 | 0 | 1 | [0.588, 0.250, 0.000, 0.163, 0.678, 0.552, 0.977, 0.308, 0.167, 0.366, 0.055, 0.041, 0.593, 0.263, 0.000, 0.230, 0.629, 0.589, 0.218, 0.009, 0.183, 0.309, 0.081, 0.052] | [0.584, 0.256, 0.000, 0.213, 0.625, 0.577, 0.298, 0.812, 0.212, 0.368, 0.072, 0.058, 0.592, 0.261, 0.000, 0.216, 0.646, 0.595, 0.032, 0.247, 0.200, 0.316, 0.085, 0.058] | [0.588, 0.250, 0.000, 0.213, 0.625, 0.577, 0.298, 0.812, 0.212, 0.368, 0.072, 0.058, 0.592, 0.261, 0.000, 0.216, 0.646, 0.589, 0.218, 0.009, 0.183, 0.309, 0.081, 0.052] | [0.584, 0.256, 0.000, 0.163, 0.678, 0.552, 0.977, 0.308, 0.167, 0.366, 0.055, 0.041, 0.593, 0.263, 0.000, 0.230, 0.629, 0.595, 0.032, 0.247, 0.200, 0.316, 0.085, 0.058] |

TABLE VI
CHROMOSOME MUTATION RESULT

| No | Index | Before Mutation | After Mutation |
|---|---|---|---|
| 1 | 0 | [0.588, 0.250, 0.000, 0.213, 0.625, 0.577, 0.298, 0.812, 0.212, 0.368, 0.072, 0.058, 0.592, 0.261, 0.000, 0.216, 0.646, 0.589, 0.218, 0.009, 0.183, 0.309, 0.081, 0.052] | [0.597, 0.109, 0.000, 0.213, 0.625, 0.577, 0.298, 0.812, 0.212, 0.368, 0.072, 0.058, 0.592, 0.261, 0.000, 0.251, 0.646, 0.589, 0.218, -0.006, 0.183, 0.309, 0.081, 0.052] |
| 2 | 3 | [0.584, 0.250, 0.000, 0.163, 0.629, 0.595, 0.032, 0.247, 0.200, 0.316, 0.085, 0.058, 0.584, 0.263, 0.000, 0.163, 0.678, 0.552, 0.977, 0.308, 0.167, 0.366, 0.085, 0.058] | [0.584, 0.443, 0.000, 0.163, 0.616, 0.595, 0.032, 0.306, 0.200, 0.316, 0.085, 0.058, 0.584, 0.263, 0.000, 0.163, 0.659, 0.552, 1.021, 0.308, 0.167, 0.366, 0.085, 0.266] |
| 3 | 4 | [0.584, 0.250, 0.000, 0.163, 0.678, 0.552, 0.977, 0.247, 0.200, 0.316, 0.085, 0.058, 0.584, 0.250, 0.000, 0.163, 0.678, 0.552, 0.977, 0.308, 0.167, 0.366, 0.055, 0.041] | [0.584, 0.250, 0.000, -0.017, 0.678, 0.552, 1.038, 0.247, 0.176, 0.316, 0.085, 0.058, 0.584, 0.250, 0.000, 0.163, 0.678, 0.552, 0.977, 0.234, 0.180, 0.366, 0.055, 0.041] |

TABLE VII
NEW CENTROID POPULATION RESULT TABLE

| No | Index | Value |
|---|---|---|
| 1 | 0 | [0.584, 0.250, 0.000, 0.163, 0.678, 0.552, 0.977, 0.308, 0.167, 0.366, 0.055, 0.041, 0.593, 0.263, 0.000, 0.230, 0.629, 0.595, 0.032, 0.247, 0.200, 0.316, 0.085, 0.058] |
| 2 | 1 | [0.573, 0.075, 0.000, 0.141, 0.528, 0.544, 0.272, 0.306, 0.154, 0.335, 0.053, 0.048, 0.628, 0.623, 0.000, 0.361, 0.859, 0.667, 0.187, 0.173, 0.267, 0.312, 0.127, 0.065] |
| 3 | 2 | [0.608, 0.076, 0.000, 0.192, 0.492, 0.575, 0.092, 0.277, 0.183, 0.310, 0.071, 0.054, 0.567, 0.529, 0.000, 0.249, 0.857, 0.600, 0.466, 0.237, 0.205, 0.353, 0.088, 0.054] |

TABLE VIII
RESULT OF SPLITTING THE CHROMOSOME INTO CENTROIDS

| No | Chromosomes | Cluster | Centroids |
|---|---|---|---|
| 1 | [0.627, 0.351, 0.429, 0.123, 0.915, 0.576, 1.533, 0.32, 0.168, 0.458, 0.387, 0.096, 0.584, 0.401, -0.009, 0.258, 0.629, 0.552, 0.762, 0.268, 0.167, 0.33, 0.055, 0.058] | 1 | [0.627, 0.351, 0.429, 0.123, 0.915, 0.576, 1.533, 0.32, 0.168, 0.458, 0.387, 0.096] |
| | | 2 | [0.584, 0.401, -0.009, 0.258, 0.629, 0.552, 0.762, 0.268, 0.167, 0.33, 0.055, 0.058] |

### *B. Cluster Distribution of K-Means and GA+K-Means*

The cluster distribution result images illustrate the outcomes of the K-Means and GA+K-Means algorithms applied to the clustered data. Each figure represents the data distribution based on the formed clusters, with each centroid indicated by a distinctive symbol, such as a black cross (X). The varying colors of the data points in the graphs correspond to the resulting clusters, facilitating a clearer visualization of group patterns.

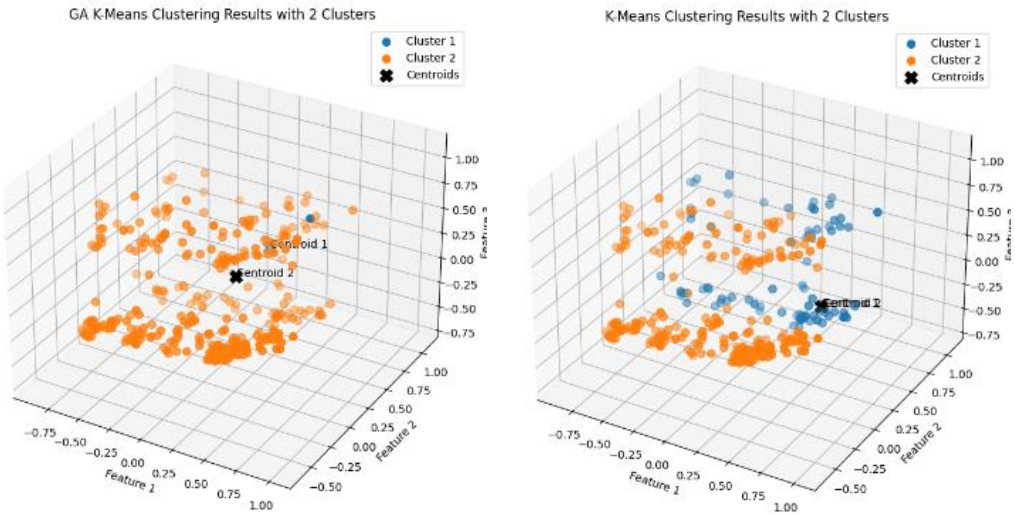


**Fig. 4 Cluster result of K-Means and GA+K-Means with 2 cluster**

### *C. Comparison of Davies-Bouldin Index Values*

A clustering experiment using the K-Means and GA+K-Means algorithms was conducted to compare the Davies-Bouldin Index (DBI) values for cluster counts ranging from 2 to 10. DBI is a widely used metric for evaluating clustering quality, where lower values indicate better clustering performance by reflecting higher intra-cluster similarity and greater inter-cluster separation.

Fig. 5 shows the DBI value obtained through the evaluation of both algorithms. K-Means algorithm achieves the lowest DBI value at 5 clusters of 1.660, followed by 6 clusters of 1.710 and 7 clusters of 1.765, indicating that K-Means performs relatively well with a moderate number of clusters. However, for smaller numbers of clusters, such as 2 and 3, the DBI values are 1.867 and 1.988, respectively, indicating less than optimal grouping quality. This limitation is likely due to the sensitivity of K-Means to the placement of the initial centroid, which often results in local optimization and poor separation when dealing with a small number of clusters.

In contrast, the GA+K-Means algorithm, which combines GA to optimize centroid initialization, shows a significant improvement in clustering quality. The lowest DBI value was obtained in 2 clusters of 0.594, followed by 10 clusters of 1.310 and 9 clusters of 1.341. Even in 5 clusters, where K-Means performed best, GA+K-Means lowered the DBI to 1.404, outperforming the conventional K-Means, which was only 1.660. On average, K-Means has a DBI value of 1,820, while GA+K-Means achieves a much lower average DBI of 1,349, which indicates a 25.9% decrease in DBI overall. This improvement demonstrates the effectiveness of GA in refining centroid initialization, which leads to consistently better clustering results.

Further analysis of the number of small clusters consisting of 2 to 4 clusters shows that GA+K-Means provides a substantial performance improvement. For example, in 2 clusters, GA+K-Means reduces the DBI from 1,867 to 0,594, while in 3 clusters, the DBI drops from 1,988 to 1,415. This reduction shows that GA+K-Means effectively reduces the tendency of K-Means to produce clusters that are not well separated in small numbers of clusters. Meanwhile, for larger numbers of clusters, namely 5 to 10 clusters, GA+K-Means still outperforms K-Means, although the difference in DBI values is not as noticeable as in smaller clusters.

Overall, although K-Means performs quite well for moderate numbers of clusters, it has difficulty producing high-quality groupings when the number of clusters is small. In contrast, GA+K-Means consistently outperforms K-Means in almost all cluster counts, with the most significant improvement observed in cases with a low number of clusters, such as 2 to 4 clusters. The lower DBI values obtained through various experiments confirm that GA+K-Means is superior in achieving high-quality grouping.

### *D. Execution Time Comparison*

A comparison of execution time between the K-Means and GA+K-Means algorithms was performed to evaluate the computational efficiency of the two algorithms in completing the clustering process.

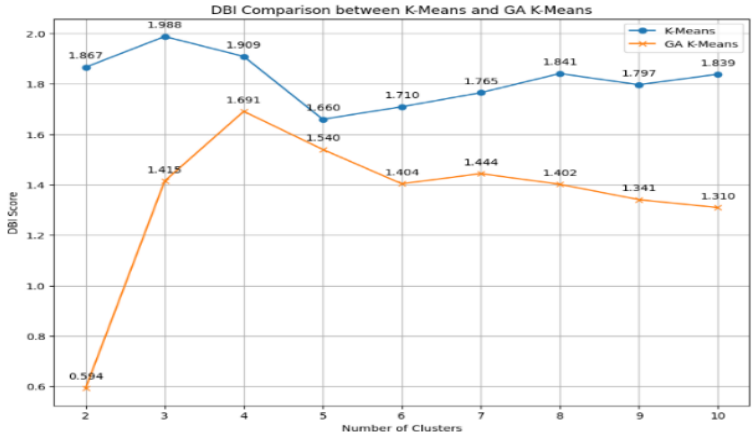


**Fig. 5 Comparison graph of DBI values of K-Means and GA+K-Means**

TABLE IX
COMPARISON TABLE OF K-MEANS AND GA+K-MEANS EXECUTION TIME

| Cluster | Execution Time (Seconds) | |
|---|---|---|
| | K-Means | GA+K-Means |
| 2 | 0.004939079284667969 | 70.59516525268555 |
| 3 | 0.007584333419799805 | 77.88833594322205 |
| 4 | 0.006617546081542969 | 86.11230182647705 |
| 5 | 0.0054967403411865234 | 101.06457328796387 |
| 6 | 0.00721430778503418 | 114.71563839912415 |
| 7 | 0.006398916244506836 | 130.05429530143738 |
| 8 | 0.005509853363037109 | 145.1785683631897 |
| 9 | 0.01973438262939453 | 164.50517296791077 |
| 10 | 0.014146566390991211 | 182.5505576133728 |
| Average | 0.008626858393351236 | 119.18495655059814 |

The results, as shown in Table IX, show a striking difference in execution duration. The K-Means algorithm shows a much faster execution time across the entire number of clusters tested, with durations ranging from 0.0049 to 0.0197 seconds and an average of 0.0086 seconds. This computational efficiency highlights the ability of K-Means to process data quickly and generate clusters, as it does not involve additional optimization steps.

In contrast, the GA+K-Means algorithm shows a much longer execution time, ranging from 70.59 to 182.55 seconds, with an average of 119.18 seconds. This increase in duration is due to the additional optimization step using Genetic Algorithms (GA), which requires more iterations and complex computations to determine the optimal centroid position. In addition, the GA+K-Means algorithm shows an increase in execution time along with an increase in the number of clusters. For example, with 2 clusters, GA+K-Means takes 70.59 seconds, while with 10 clusters, the duration increases to 182.55 seconds.

In summary, although GA+K-Means produces more optimal grouping results in terms of cluster separation, its execution efficiency is substantially lower than K-Means. Therefore, choosing the right algorithm must involve a careful trade-off between grouping quality and execution time, depending on the system's specific requirements and computing capacity.

## IV. CONCLUSION

Based on the research results, optimization of the K-Means algorithm using Genetic Algorithms (GA) has been proven to improve the clustering quality of MSMEs data in Yogyakarta. The integration of GA successfully reduced the average Davies-Bouldin Index (DBI) value by 25.9% compared to conventional K-Means, indicating a significant improvement in cluster quality. Especially for small clusters, such as 2 clusters, GA+K-Means resulted in a DBI of 0.594, much lower than the conventional K-Means with a DBI value of 1.867. However, this advantage is offset by the disadvantage in terms of execution time. The average execution time of the GA+K-Means method reaches 119.185 seconds, much higher than the conventional K-Means, which only takes an average of 0.009 seconds. This increase in

execution time is due to the iterative process in GA, which includes selection, crossover, and mutation, which require more computational resources. For future research, it is recommended that execution time constraints be overcome by optimizing the iteration process or utilizing better computing infrastructure. Additionally, testing with larger datasets, exploring alternative GA methods, and adding more variables can help expand the application of this method. Furthermore, hybrid approaches combining GA with other optimization techniques, such as Particle Swarm Optimization (PSO) or Differential Evolution (DE), may improve efficiency and accuracy. Lastly, utilizing parallel processing or Graphics Processing Unit (GPU) acceleration could reduce the high computational cost of GA iterations.

## ACKNOWLEDGEMENT

The researcher would like to express his deepest gratitude to all parties who have helped, especially to Mrs. Lisna Zahrotun, S.T., M.Cs., as the supervisor, for her guidance, knowledge, and support in obtaining the data and insights needed during the research process.

## REFERENCES

[1] B. Khusul Khotimah, F. Irhamni, and T. Sundarwati, "A GENETIC ALGORITHM FOR OPTIMIZED INITIAL CENTERS K-MEANS CLUSTERING IN SMEs," *J Theor Appl Inf Technol*, vol. 15, no. 1, 2016, [Online]. Available: www.jatit.org

[2] Y. Ansori and C. Wulandari, "CRISP-DM Method On Indonesian Micro Industries (UMKM) Using K-Means Clustering Algorithm," *MATICS: Jurnal Ilmu Komputer dan Teknologi Informasi (Journal of Computer Science and Information Technology)*, vol. 14, no. 2, pp. 35–40, Oct. 2022, doi: 10.18860/mat.v14i2.13760.

[3] R. Kesuma Dinata, H. Novriando, N. Hasdyna, S. Retno, J. Hadari Nawawi, and K. Barat, "JEPIN (Jurnal Edukasi dan Penelitian Informatika) Reduksi Atribut Menggunakan Information Gain untuk Optimasi Cluster Algoritma K-Means," *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, 2020.

[4] R. Kurniati, O. Arsalan, and Y. Ramadhana, "Initial Centroid Determination Using Genetic Algorithm in Data Clustering," *Jurnal Generic*, vol. Vol 13 No 1 (2021), 2021.

[5] G. J. Oyewole and G. A. Thopil, "Data clustering: application and trends," *Artif Intell Rev*, vol. 56, no. 7, pp. 6439–6475, Jul. 2023, doi: 10.1007/s10462-022-10325-y.

[6] L. Zahrotun, S. Hadi Nugroho, U. Linarti, and A. Hendri Soleliza Jones, "Analisis Persebaran UMKM Bidang Jasa Menggunakan Metode AHC Complete Linkage," *KESATRIA: Jurnal Penerapan Sistem Informasi (Komputer & Manajemen)*, vol. 4, no. 2, pp. 255–265, 2023.

[7] M. Faishal, R. Juniardi, L. Zahrotun, U. Linarti, and A. Hendri Soleliza Jones, "Data Mining Pengelompokan UMKM di Bidang Fashion dan Kerajinan Kota Yogyakarta Menggunakan AHC Average Linkage," *JUMANJI*, vol. 7, no. 2, pp. 2598–8069, 2023.

[8] L. Zahrotun, Y. R. Amanatullah, U. Linarti, and A. H. Soleliza Jones, "Strategy for improving and empowering MSMEs through grouping using the AHC method," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 13, no. 1, pp. 130–136, Feb. 2024, doi: 10.32736/sisfokom.v13i1.2021.

[9] U. Linarti, A. Rahmawati, A. Hendri Soleliza Jones, and L. Zahrotun, "Penerapan Metode K-Medoids Guna Pengelompokan Data Usaha Mikro, Kecil dan Menengah (UMKM) Bidang Kuliner Di Kota Yogyakarta," *Jurnal Ilmu Komputer dan Sistem Informasi (JIKOMSI)*, vol. 7, no. 1, pp. 37–45, 2024.

[10] L. Zheng, L. Haiyan, L. Ce, L. Qingyu, and L. Gang, "Research on K-Means Clustering Optimization Algorithm Based on Machine Learning," *Hans Journal of Data Mining*, vol. 12, no. 01, pp. 20–26, 2022, doi: 10.12677/hjdm.2022.121003.

[11] Taslim, D. Toresa, D. Jollyta, D. Suryani, and E. Sabna, "Optimasi K-Means dengan Algoritma Genetika untuk Target Pemanfaat Air Bersih Provinsi Riau," *Indonesian Journal of Computer Science*, vol. 10, no. 1, Jul. 2022, doi: 10.33022/ijcs.v10i1.3064.

[12] M. E. Al Rivan and R. A. Sonaru, "Perbandingan Metode K-Means dan GA K-Means untuk Clustering Dataset Heart Disease Patients," *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 9, no. 3, pp. 2585–2597, Sep. 2022, doi: 10.35957/jatisi.v9i3.2799.

[13] Hendrik, Kusrini, and Kusnawi, "OPTIMASI PENENTUAN SENTROID AWAL PADA K-MEANS UNTUK MENINGKATKAN HASIL EVALUASI DAVIES-BOULDIN INDEX," *Jurnal Informatika Teknologi dan Sains (JINTEKS)*, vol. 6, no. 1, 2024.

[14] Y. Ramadhana and M. Ihsan Jambak, "The Influence of Optimization of the k-Means Algorithm with Genetic Algorithm on the Results of High Dimension Data Clustering," *Indonesian Journal of Computer Science Attribution*, vol. 13, no. 1, p. 302, 2024.

[15] S. García, S. Ramírez-Gallego, J. Luengo, J. M. Benítez, and F. Herrera, "Big data preprocessing: methods and prospects," *Big Data Anal*, vol. 1, no. 1, Dec. 2016, doi: 10.1186/s41044-016-0014-0.

[16] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," *Global Transitions Proceedings*, vol. 3, no. 1, pp. 91–99, Jun. 2022, doi: 10.1016/j.gltp.2022.04.020.

[17] S. P. Kandel, Z. Asgar, W. Zheng, and P. J. Vander Broek, "Standardizing values of a dataset," US 10,824,606 B1, Nov. 03, 2020 Accessed: Dec. 17, 2024. [Online]. Available: https://patentimages.storage.googleapis.com/3c/fc/fd/3469d295ce2e73/US10824606.pdf

[18] M. S. Gal and D. L. Rubinfeld, "Data standardization," *New York University Law Review*, vol. 94, no. 4, pp. 737–770, Oct. 2019, doi: 10.2139/ssrn.3326377.

[19] S. Roy, P. Sharma, K. Nath, D. K. Bhattacharyya, and J. K. Kalita, "Pre-processing: A data preparation step," in *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics*, vol. 1–3, Elsevier, 2018, pp. 463–471. doi: 10.1016/B978-0-12-809633-8.20457-3.

[20] A. S. Sukamto, W. Setiawan, and E. E. Pratama, "JEPIN (Jurnal Edukasi dan Penelitian Informatika) Data Mining untuk Pengelompokan Saham pada Sektor Energi dengan Metode K-Means," *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, Apr. 2023.

[21] A. A. Arrosyad, A. I. Purnamasari, and I. Ali, "IMPLEMENTASI ALGORITMA K-MEANS CLUSTERING UNTUK ANALISIS PERSEBARAN UMKM DI JAWA BARAT," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 3, 2024.

[22] R. Li and L. A. Kazakovtsev, "COMPARATIVE STUDY OF MUTATION OPERATORS IN THE GENETIC ALGORITHMS FOR THE K-MEANS PROBLEM," *Facta Universitatis, Series: Mathematics and Informatics*, p. 1091, Feb. 2020, doi: 10.22190/fumi20040911.

[23] Y. Fang and J. Li, "A review of tournament selection in genetic programming," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2010, pp. 181–192. doi: 10.1007/978-3-642-16493-4_19.

[24] S. Prayudani, A. Hizriadi, E. B. Nababan, and S. Suwilo, "Analysis Effect of Tournament Selection on Genetic Algorithm Performance in Traveling Salesman Problem (TSP)," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Jul. 2020. doi: 10.1088/1742-6596/1566/1/012131.

[25] O. Bell, "Applications of Gaussian Mutation for Self Adaptation in Evolutionary Genetic Algorithms," *Journal of Machine Learning in Fundamental Sciences*, Jan. 2022, [Online]. Available: http://arxiv.org/abs/2201.00285

[26] I. Firman Ashari, R. Banjarnahor, D. R. Farida, S. P. Aisyah, A. P. Dewi, and N. Humaya, "Application of Data Mining with the K-Means Clustering Method and Davies Bouldin Index for Grouping IMDB Movies," *Journal of Applied Informatics and Computing (JAIC)*, vol. 6, no. 1, pp. 2548–6861, 2022, [Online]. Available: http://jurnal.polibatam.ac.id/index.php/JAIC