Journal of Information System Exploration and Research

https://joiser.shmpublisher.com

p-ISSN 2964-1160 | e-ISSN 2963-6361

Comparative Analysis of Data Normalization Effects on RFM-Based Customer Segmentation Using K-Means and DBSCAN

Nabilah Aulia Zahra[1*], Tyas Sekar Arum[2], Zerafica Adhyasti Dinar Patriawan[3], Dwika Ananda Agustina Pertiwi[4], Much Aziz Muslim[5], Yusuf Enril Fathurrohman[6]

[1,2,3,5]*Department of Computer Science, Universitas Negeri Semarang, Indonesia*
[4]*Faculty of Technology Management, Universiti Tun Hussein Onn Malaysia, Malaysia*
[6]*Department of Agribusiness, Universitas Muhammadiyah Purwokerto, Indonesia*
[6]*Doctoral School of Management and Business, University of Debrecen, Hungary*

DOI: https://doi.org/10.52465/joiser.v4i2.86
Received 17 June 2026; Accepted 02 July 2026; Available online 02 July 2026

Article Info

Keywords:
Customer segmentation;
RFM analysis;
Data normalization;
K-Means algorithm;
DBSCAN algorithm

Abstract

Customer segmentation is widely used to analyze customer transaction patterns and support effective business strategies. However, previous studies have reported inconsistent findings regarding the impact of data normalization on clustering quality across different datasets and algorithms. This study investigates the effect of data normalization on RFM-based customer segmentation using K-Means and DBSCAN. Two transaction datasets, Online Retail II and TransJakarta, were analyzed under three preprocessing scenarios: no normalization, Min-Max normalization, and Z-Score normalization. Clustering performance was evaluated using the Silhouette Score and Davies–Bouldin Index (DBI). For the Online Retail II dataset, K-Means achieved the best performance without normalization (Silhouette Score = 0.9845), while DBSCAN produced valid clusters only after Z-Score normalization. For the TransJakarta dataset, both algorithms performed best without normalization, whereas DBSCAN identified up to 20 clusters and noise points. These findings highlight that the effectiveness of normalization depends on dataset characteristics and the clustering algorithm used.

1. Introduction

The volume of transaction data generated by business activities and public services continues to increase. This situation encourages the use of data analysis techniques to gain a better understanding of customer behavior. One widely used approach is Recency, Frequency, and Monetary (RFM)-based customer segmentation, as it can describe customer interaction patterns through the time of the last transaction, the frequency of transactions, and the resulting transaction [1], [2], [3], [4]. To form groups

[*] ***Corresponding Author:***

Nabilah Aulia Zahra,
Department of Computer Science,
Universitas Negeri Semarang,
Sekaran, Gunungpati, Semarang, Indonesia 50229.
Email: nabilahuliaz27@students.unnes.ac.id

of customers with similar characteristics, various studies utilize clustering algorithms such as K-Means and DBSCAN [5], [6], [7].

Although the application of RFM-based customer segmentation has been widely reported in various domains, previous research has generally focused on comparing clustering algorithms or their application to specific [4]. Meanwhile, the influence of data preprocessing, particularly normalization, has been relatively rarely discussed in depth [8]**.** Differences in scale between RFM attributes can impact the cluster formation process, particularly in algorithms that rely on distance calculations [8]. Furthermore, results from various studies indicate that the benefits of normalization are not always consistent, so it is uncertain whether changes in clustering quality are more influenced by data characteristics or by the algorithmic mechanisms [5], [7], [8], [9].

Based on these conditions, this study analyzes the effect of normalization on RFM-based customer segmentation using the K-Means and DBSCAN algorithms. This analysis was conducted on two datasets with different transaction characteristics, namely Online Retail II and TransJakarta, using three pre-processing scenarios, namely without normalization, Min-Max normalization, and Z-Score normalization. Cluster quality was evaluated using the Silhouette Score and Davies–Bouldin Index (DBI) to obtain an overview of the relationship between pre-processing strategies, data characteristics, and clustering performance [7], [8].

The main contributions of this study are threefold. First, this study systematically compares three preprocessing scenarios, namely raw RFM data, Min-Max normalization, and Z-Score normalization, to evaluate their effects on customer segmentation quality. Second, unlike many previous studies that rely on a single dataset, this research investigates normalization effects across two datasets with different transaction characteristics, providing broader empirical evidence regarding the role of feature scaling in RFM-based clustering. Third, this study compares the behavior of a centroid-based clustering algorithm (K-Means) and a density-based clustering algorithm (DBSCAN) under identical preprocessing scenarios, enabling a deeper understanding of how normalization influences clustering performance across different clustering paradigms.

2. Literature Review

The Recency, Frequency, and Monetary (RFM) model is widely used in customer segmentation research to represent customer behavior through three measurable dimensions: the time elapsed since the last transaction (recency), the total number of transactions within a given period (frequency), and the cumulative transaction value generated by each customer (monetary). Overall, these three dimensions enable transaction data to be summarized into customer profiles that can be used for segmentation and behavioral analysis [1], [2].

Several studies have utilized RFM-based approaches in various domains. In the online retail sector, clustering techniques are used to identify purchasing patterns and customer groups based on transaction value [5]. The same approach is also applied to transportation services to analyze transaction activity and passenger usage patterns [9]. The results of these studies indicate that RFM-based clustering can be effectively utilized to map customer behavior and support the decision-making process [10].

K-Means and DBSCAN are two clustering algorithms widely used in RFM-based customer segmentation. K-Means works by dividing data into a predetermined number of clusters by iteratively minimizing the distance between the data and the cluster center. Because it uses Euclidean distance as the basis for clustering, this algorithm is sensitive to differences in scale between features. Variables with a larger range of values can dominate the distance calculation, thus affecting the resulting clusters. On the other hand, DBSCAN forms clusters based on data density and does not require a predetermined number of clusters. This algorithm is also capable of identifying data that does not fall into any cluster as noise [5], [6]. However, the quality of the clustering results is greatly influenced by the data distribution and the parameter selection used.

Before the clustering process is performed, data is generally normalized to reduce scale differences between variables. In RFM data, each attribute typically has a different value range. For example, the Monetary attribute typically has a much larger value than the Recency or Frequency attributes. This can impact clustering results, especially in algorithms that use Euclidean distance calculations such as K-Means. If data is not normalized, attributes with large values will more dominantly influence cluster formation [11]. Numerous studies have shown that the success of a normalization process varies from

case to case. Dataset characteristics, feature distribution, and the algorithmic mechanisms used in the clustering process influence its effectiveness [2], [9], [11].

In retail transaction datasets, the Monetary attribute often contains significantly larger values than other RFM dimensions due to a small number of customers making very high-value transactions. This can result in a highly skewed distribution and increase the influence of outliers during data analysis. Previous studies have also suggested using a logarithmic transformation before normalization as a way to reduce distribution skew and moderate the impact of extreme values. By compressing a large range of observations while preserving the relative order of data points, this approach can help create a more balanced feature distribution and support clustering algorithms that rely on distance- and density-based calculations [11], [12], [13].

Assessing clustering results is a crucial step in determining the quality of the resulting customer segmentation. In this process, internal evaluation metrics such as the Silhouette Score and Davies-Bouldin Index are often used to measure cluster quality. The Silhouette Score is used to assess the similarity of data within a cluster compared to other nearby clusters, while the Davies-Bouldin Index is used to measure the degree of compactness and separation between clusters [1], [5]. The combination of these two metrics allows cluster quality to be analyzed from different perspectives.

3. Method

This study analyzes the impact of data normalization on mutual customer segmentation based on RFM using quantitative methods through an experimental approach. The K-Means and DBSCAN algorithms were used for testing as clustering methods. This approach was chosen because it can make the testing process on various data processing techniques more focused and, at the same time, facilitate objective evaluation of cluster results through the use of standard assessment measurements [2], [6].

This research implemented several interconnected stages. The first stage began with data collection, followed by pre-processing and feature formation, followed by data normalization, clustering, and evaluation and analysis of the results. Each stage of this research was systematically designed to ensure the analysis remained focused and the experiment could be repeated with similar results. The general research flow can be seen in Figure 1.

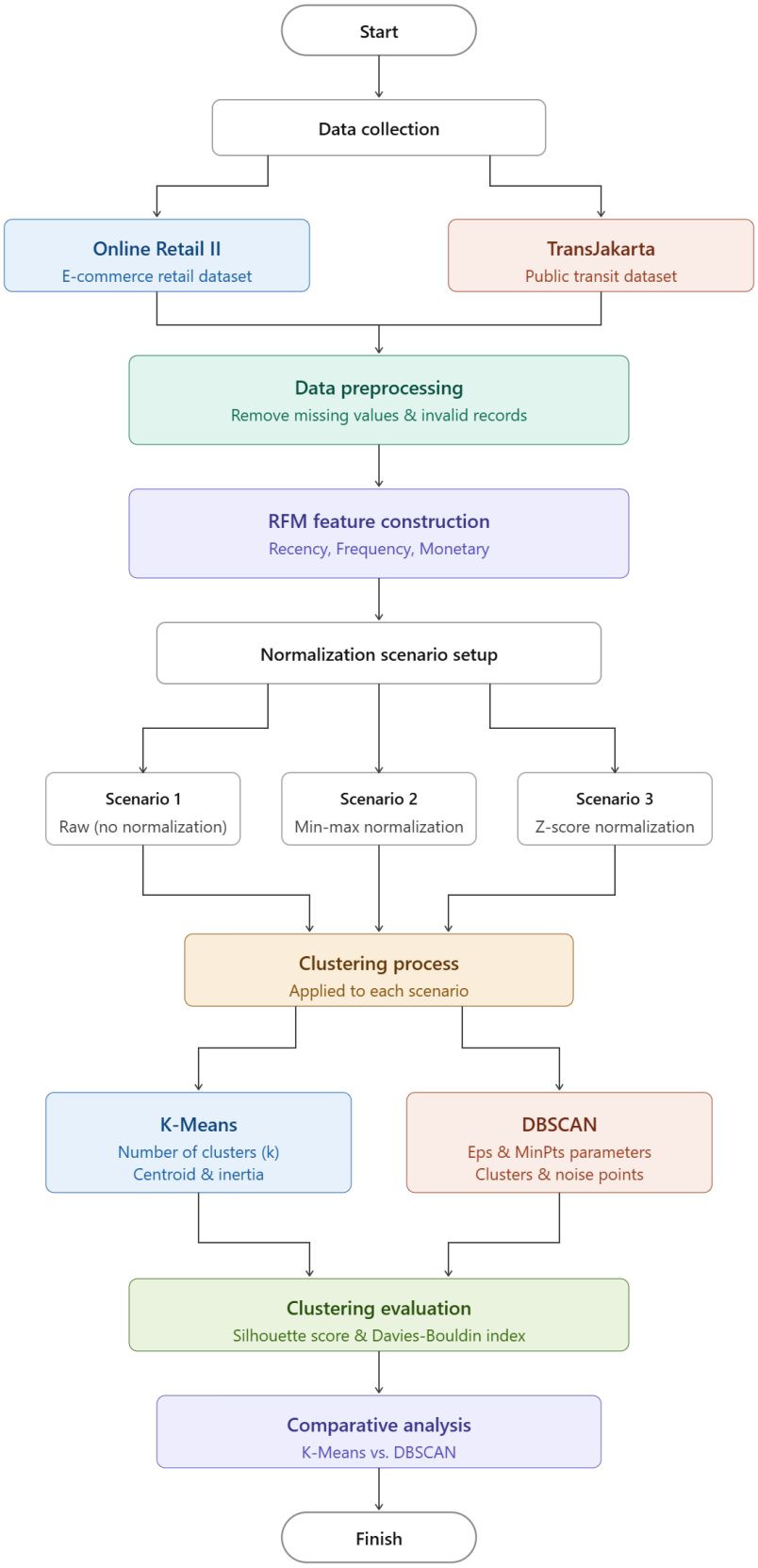

Figure 1. Research methodology workflow

3.1 Data collection

This study used two common datasets obtained from the Kaggle platform, namely Online Retail II (www.kaggle.com/datasets/ersany/online-retail-dataset/data) and TransJakarta (www.kaggle.com/code/kemalmaolana/transjakarta-eda/). Before data processing, the Online Retail II dataset contained 1,067,371 transaction histories from e-commerce retail activities. This dataset describes customer transaction behavior with varying transaction values. The TransJakarta dataset contained 37,900 transaction histories from public transportation services with stable transaction values.

The two datasets were selected based on their different characteristics. The Online Retail II dataset depicts customer behavior with varying spending levels, while the TransJakarta dataset depicts similar transaction patterns. These different characteristics provide an opportunity to assess the performance of clustering and normalization techniques on two datasets with different characteristics. Therefore, a comprehensive analysis of the effect of normalization on RFM-based customer segmentation can be conducted [5], [9].

3.2 Data preprocessing and RFM feature construction

Before the clustering process was carried out, a preprocessing phase was performed on both datasets to ensure clean and stable data. The preprocessing phase began with data cleaning, then handling incomplete data, and forming RFM features as the basis for the analysis. The process begins with checking for duplicate data and then removing them. Duplicate data can result in multiple recorded transactions, which affetcs the Frequency and Monetary values. Afterward, data with blank values in important attributes was excluded from the analysis. Incomplete data results in inaccurate feature calculations and decreases the quality of the clustering results.

Furthermore, transactions deemed invalid were filtered out so that this study only included data that accurately reflected customer activity. This stage aims to ensure stable data quality and reduce the potential for bias. Once the data is clean, the transaction data is transformed into three main RFM attributes. Recency measures the time interval since a customer's last transaction. Frequency indicates the number of times a customer made a transaction during the observation period. The Monetary value indicates the total transaction value generated by the customer. The selection of three attributes was based on their ability to concisely represent behavioral patterns and has been widely used in several customer segmentation studies [1], [9], [14].

The resulting RFM dataset was then used as the basis for the normalization and clustering stages.

3.3 Data normalization scenarios

To evaluate the impact of feature scale on clustering results, three different data conditions were compared in this study. These three conditions included the original RFM data, data after Min-Max normalization, and data after Z-Score normalization. The original RFM data served as the baseline, while the two normalization techniques were used to observe changes in clustering results after adjusting the feature scale.

The different numerical scales of the RFM attributes are the reason for normalization. Of the three features, the Monetary attribute typically has the highest value compared to Recency and Frequency. This causes the clustering process to be influenced by the larger value, resulting in an imbalance in the role of each feature [11], [15]

Equation (1) is used to perform Min-Max normalization, where the feature value scale is changed into the range 0 to 1 [11], [15]:

$$\boldsymbol{x'} = \frac{\boldsymbol{x - xmin}}{\boldsymbol{xmax - xmin}} \quad (1)$$

Where the original feature value is indicated by $\boldsymbol{x}$, the minimum value is indicated by $\boldsymbol{xmin}$, and the maximum value of the feature is indicated by $\boldsymbol{xmax}$.

Equation (2) is used for Z-Score normalization, which normalizes each feature based on its data mean and standard deviation [11], [15]**:**

$$\boldsymbol{z} = \frac{\boldsymbol{x - \mu}}{\boldsymbol{\sigma}} \quad (2)$$

With the original feature value denoted by $\boldsymbol{x}$, the feature mean denoted by $\boldsymbol{\mu}$, and the feature standard deviation denoted by $\boldsymbol{\sigma}$.

As a comparison parameter, this study still uses the original RFM data without normalization. For comparison, two commonly used normalization techniques are used: Min-Max and Z-Score, which reflect two different approaches to data scaling. The Min-Max normalization method is used to balance the range of values between features, so that each attribute is on an equal scale. Meanwhile, the Z-Score standardizes the values by considering the average and the degree of data distribution. By comparing the results on the original data with both normalization schemes, the impact of feature scale on clustering results can be more explicitly observed [2].

Transaction data with an uneven monetary distribution is recommended to use a logarithmic transformation. Using a logarithmic transformation can reduce the impact of excessively large values and make the data distribution more equal. In this study, the Monetary attribute in both datasets has a much wider range of values compared to the Recency and Frequency attributes. The different ranges indicate outliers and uneven data distribution.

However, logarithmic transformation was not applied because the focus of this study was to compare the effect of two commonly used normalization techniques, Min-Max and Z-Score, on clustering quality in the original RFM representation. By limiting the scenario to the raw data, Min-Max, and Z-Score, the effect of normalization can be evaluated more directly. The use of logarithmic transformation before normalization can be explored in future research.

3.4 Clustering process

The data from each normalization scenario were subsequently used in the clustering stage using K-Means and DBSCAN. For K-Means, Euclidean distance was used to measure the similarity between data points, following the standard approach commonly adopted in centroid-based clustering [6]. In order for each experiment to obtain consistent test results, the K-Means algorithm was run using random_state = 42 and n_init = 10. Meanwhile, other parameters remained standard from Scikit-learn, including the maximum number of iterations set at 300. To determine the optimal number of clusters, k values ranging from 2 to 10 were tested. The clustering quality at each k value was assessed using the Silhouette Score. Of all the experiments conducted, the k value with the highest Silhouette Score was selected as the optimal number of clusters for use in the dataset.

DBSCAN was chosen as a density-based clustering technique because it allows clustering without having to determine the number of clusters first [15], [16]. The selection of the eps parameter in this study was determined with the help of the k-distance graph stated by Ester et al. [15] The way to carry out this method is by sorting the distance to the kth nearest neighbor, then observing the change points on the graph as a basis for determining the most appropriate eps value.

Based on this, parameter searches were conducted in stages for each normalization scheme using an empirical grid search approach. Parameter selection was conducted in two stages of testing. The first stage used a wider range of eps values, namely 0.5; 1.0; 1.5; 2.0; 3.0; and 5.0 to obtain an overview of the parameter areas with high potential. The second stage carried out a more focused search using eps values of 0.1; 0.3; 0.5; 0.7; and 1.0 to obtain more precise parameters. All eps values at each stage were tested with min_samples values of 2,5 and 10.

For each normalization scheme, the best eps value is determined based on observations during the grid search process. Experimental results show that very small eps values tend to create many data points that are considered noise, while very large eps values tend to combine most of the data into too few clusters. Therefore, the best eps value is selected based on the setting that produces valid clustering results with the highest Silhouette Score.

In the DBSCAN algorithm, the selection of the min_samples value depends on the number of dimensions used in the data to be analyzed. One parameter often used as a reference is d + 1, with d indicating the number of features used [15], [17]. The feature space in this study includes three RFM attributes, so the min_samples value directed by this parameter is around 4.

However, considering the initial test results using min_samples = 3, the clustering results obtained on both datasets were more consistent. With this value, clusters can be formed well without increasing the number of noisy data points. Increasing the min_samples value also tends to increase the number of noisy points, resulting in less optimal clustering results. Therefore, the value 3 was chosen for all tests. Furthermore, the parameter search process was implemented the same on both datasets to ensure consistent and easy replication of comparisons.

3.5 Clustering evaluation

To measure the quality of clustering results in this study, two internal metrics were used, namely the Silhouette Score and the Davies-Bouldin Index (DBI) [18]. These two metrics were chosen for use in this study because they can directly assess cluster quality without requiring class labels. The assessment is conducted by observing how the data is combined and the clarity of the resulting cluster separation. This study employed an unsupervised learning approach, therefore, these metrics were considered appropriate for assessing the quality of the clustering results.

Silhouette Score menunjukkan tingkat kesesuaian suatu titik terhadap cluster tempatnya berada dan dibandingkan dengan cluster terdekat lainnya [19]. Koefisien siluet untuk setiap data $\boldsymbol{i}$ dihitung menggunakan Persamaan (3):

$$\boldsymbol{s(i)} = \frac{\boldsymbol{b(i)-a(i)}}{\mathbf{max}\boldsymbol{(a(i), b(i))}} \quad (3)$$

In this calculation, $\boldsymbol{a(i)}$ indicates the average distance between data point i and all other points located in the same cluster, thus describing the level of closeness within the cluster. Conversely, the value of $\boldsymbol{b(i)}$ indicates the average of the smallest distance between data point i and other points in the closest cluster, thus describing the level of separation between clusters. All Silhouette Score values are obtained by calculating the average value of $\boldsymbol{s(i)}$ from all data points in the dataset. Silhouette Scores range from -1 to 1. Values closer to 1 indicate that a data point has a valid match with its cluster and is well separated from other clusters. Values closer to 0 indicate that the point is located in the border area between clusters. Meanwhile, values below 0 indicate that the data is likely placed in an inappropriate cluster. Therefore, a larger Silhouette Score value is used in this study as a sign that the clustering results have better quality with a clear level of separation.

The Davies-Bouldin Index (DBI) is used to measure the quality of clustering results by looking at two main factors, namely the level of data compactness and the level of separation from other clusters [18]. The DBI calculation is based on the relationship between distribution within a cluster and the distance between cluster centers, which is shown in Equation (4):

$$\boldsymbol{DBI} = \frac{\mathbf{1}}{\boldsymbol{n}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \mathbf{max}\ \boldsymbol{j \neq i} \left(\frac{\boldsymbol{Si + Sj}}{\boldsymbol{Mij}} \right) \quad (4)$$

In this calculation, the number of clusters formed is indicated by $\boldsymbol{k}$. The average distance between cluster members $\boldsymbol{i}$ and the cluster centroid is indicated by $\boldsymbol{Si}$, and is used to indicate the level of cluster cohesion. $\boldsymbol{Mij}$ indicates the distance between the centroid of cluster $\boldsymbol{i}$ and the centroid of cluster $\boldsymbol{j}$, indicating the degree of separation between clusters.

Each cluster is calculated by comparing the ratio $\boldsymbol{(Si + Sj)/Mij}$ to all other clusters. The highest ratio is then selected because it indicates the highest level of similarity with other clusters. The DBI value is then obtained by averaging all maximum ratios across all clusters.

Unlike the Silhouette Score, which has a specific value range, the DBI does not have a specific maximum value. A small DBI value indicates good clustering quality because it is denser and more clearly separated. A high DBI value indicates a significant distribution of data within the cluster.

The Silhouette Score and DBI are used together to obtain a more comprehensive picture of the quality of clustering results. The Silhouette Score is used to measure the accuracy of each data point's location within its cluster by observing the comparison of its distance to other clusters, so that the degree of separation between clusters can be clearly seen. On the other hand, the DBI is used to measure clustering quality by considering the level of cohesion within the cluster and the distance between one cluster and another.

Combining both metrics simultaneously allows for a more comprehensive assessment of clustering results. If the results from both metrics differ, clustering quality cannot be assessed based on a single metric alone. When this occurs, the presentation of the results should be supplemented by observing the characteristics and profiles of each cluster for greater accuracy and unbiased analysis.

3.6 Comparative analysis

A comparative analysis was conducted to observe the different clustering results for each experimental scheme tested. This study focused primarily on comparing the performance of K-Means and DBSCAN across various datasets and measuring the extent to which the normalization techniques impacted the clustering results.

Comparisons were made for all normalization schemes between one another, starting from the original data, data through Min-Max normalization, and data through Z-Score normalization. These

were then applied to the two clustering algorithms used in this research, Through this analysis, the research aims to determine which combination of methods produces the most accurate and effective results for each dataset.

3.7 Implementation details

The entire analysis series in this study was run using Python 3.12.13 through the Google Colab platform. Several libraries were used to facilitate various stages of data processing, namely pandas, NumPy, and Scikit-learn 1.6.1. Pandas was used for data management and preparation, Numpy for numerical calculations, and Scikit-learn for normalization, clustering, and measuring clustering results.

The entire analysis series used the same rules for each dataset to obtain rationally comparable results. This process began with data cleaning, feature formation and normalization, the clustering stage, and the evaluation of the results.

To ensure the stability of the research results, the initialization process for K-Means was set to random values using random_state = 42 and n_init = 10, as described in section 3.4. Furthermore, to allow for future research to be replicated and developed, all test settings used, such as parameters, normalization methods, evaluation metrics, library versions, and testing platforms, must also be recorded in detail.

4. Results and Discussion

This stage presents the results and discussion.

4.1 Data preprocessing results

Based on the data preprocessing results, the number of records in the two datasets experienced different changes. In the Online Retail II dataset, the initial data of 1,067,371 records was reduced to 1,055,238 records after removing duplicates. After handling missing values, the number of records decreased to 812,368 records and then again to 793,609 records after removing invalid transactions. The significant reduction in the missing value handling stage indicates that the primary problem in this dataset stems from incomplete transaction data [2], [14], [20].

In the TransJakarta dataset, the number of records remained at 37,900 records after removing duplicates. After handling missing values, the number of records decreased to 36,893 records, then decreased to 20,245 records after filtering out invalid transactions. In contrast to Online Retail II, the largest reduction in this dataset occurred due to the large number of transactions that did not meet validity criteria. Table 1 provides a concise overview of the record redcuctions across each stage of data cleaning.

These results indicate that the two datasets have different data quality characteristics. Retail Online II was more affected by missing values, while TransJakarta was more affected by invalid transactions. Nevertheless, the cleaning process successfully produced data that was more suitable for use in RFM feature formation, allowing the resulting customer representation to serve as the basis for clustering analysis in the next stage

Table 1. Summary of preprocessing results on the Online Retail II and TransJakarta datasets

Dataset	Initial Data	After Duplicate Removal	After Missing Value Handling	After Invalid Transaction Filtering
Online Retail II	1,067,371	1,055,238	812,368	793,609
TransJakarta	37,900	37,900	36,893	20,245

4.2 RFM feature construction results

After the pre-processing stage, transaction records are converted into Recency, Frequency, and Monetary (RFM) features, which are then used to describe customer behavior at the individual level. This transformation is necessary because the clustering process is based on customer characteristics, not individual transaction records [1], [2], [14].

In the Online Retail II dataset, the RFM construction process generated 5,878 customer records. According to Table 2, the average values for Recency, Frequency, and Monetary were 201.3319, 6.2894, and 3,008.7548, respectively. Among these three attributes, Monetary had the largest range of values, with a minimum value of 2.9500 and a maximum value of 608,821.6500. The significant variation in this attribute indicates significant differences in transaction value between customers. Some customers

generate only relatively small transaction values, while others contribute significantly larger transactions. This condition indicates an imbalance in scale between features that has the potential to affect the clustering process, especially in algorithms that use distance calculations as the basis for grouping.

Table 2. Descriptive statistics of RFM features on the Online Retail II dataset

Statistic	Recency	Frequency	Monetary
Count	5,878	5,878	5,878
Mean	201.3319	6.2894	3,008.7548
Std	209.3387	13.0094	14,731.9241
Min	1	1	2.9500
25%	26	1	345.1150
50%	96	3	887.3900
75%	380	7	2,298.0050
Max	739	398	608,821.6500

In the TransJakarta dataset, the RFM construction process resulted in 1,356 customer records. Based on Table 3, the average values of Recency, Frequency, and Monetary were 7.0597, 14.9299, and 73,451.6962, respectively. Compared to the Online Retail II dataset, a lower Recency value indicates that TransJakarta customers tend to make transactions within a timeframe closer to the observation period. Furthermore, a relatively high Frequency value indicates a more repetitive service usage pattern. However, the Monetary attribute still has a fairly wide range of values, from 3,500.000 to 800,000.000. This variation indicates a significant difference in the accumulated transaction value between customers. Similar to the Online Retail II dataset, significant scale differences in the Monetary attribute have the potential to affect clustering results if the data is used without a normalization process.

Table 3. Descriptive statistics of RFM features on the TransJakarta dataset

Statistic	Recency	Frequency	Monetary
Count	1,356	1,356	1,356
Mean	7.0597	14.9299	73,451.6962
Std	6.6726	17.3683	137,191.4319
Min	1	1	3,500.0000
25%	2	2	7,000.0000
50%	3	3	14,000.0000
75%	12	40	140,000.0000
Max	30	40	800,000.0000

Overall, the RFM construction process successfully transformed transaction data analysis into customer representations ready for clustering. The Online Retail II dataset contains transactions with a wide range of values, in contrast to the TransJakarta dataset, where service usage tends to be more regular and repetitive. Despite their different characteristics, both datasets display a much larger range of Monetary values compared to Recency and Frequency. This small scale has the potential to affect the clustering process, especially in algorithms that rely on distance calculations [11]. Therefore, the next stage applies several normalization methods to mitigate the effect of feature scaling on the quality of the resulting clusters.

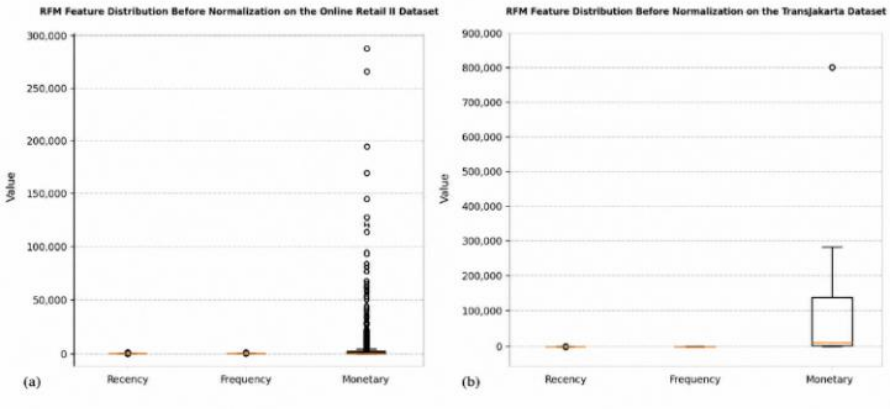

Figure 2. Boxplot distribution of Recency, Frequency, and Monetary (RFM) features before normalization in (a) the Online Retail II dataset and (b) the TransJakarta dataset.

4.3 Data Normalization Scenario Results

Based on the characteristics of RFM features that have different value ranges between attributes, three data scenarios were applied to the Online Retail II and TransJakarta datasets, namely without normalization, Min-Max normalization, and Z-Score normalization. The application of these three scenarios aims to evaluate the effect of feature scaling on clustering results. The difference in scale is a concern because algorithms such as K-Means and DBSCAN use distance information in the cluster formation process, so that attributes with a larger value range can have a more dominant influence [21]. This condition is commonly found in RFM data, especially in Monetary attributes which often have a much larger value range than other attributes [11].

In the unnormalized scenario, the original values of each attribute are retained so that the data characteristics remain unchanged. This scenario was used as a reference for comparing clustering results with the normalized data. Based on Figure 2, the Monetary attribute in both datasets has a much larger range of values than Recency and Frequency. Large differences in feature scales can cause monetary attributes to have a stronger influence during the clustering process.

To reduce the scale gap between features, RFM data was normalized using the Min-Max and Z-Score methods [22]. Min-Max normalization transforms feature values into a range of 0 to 1, while Z-Score normalization adjusts values based on the mean and standard deviation of each feature. Examples of the transformation results using both normalization methods for the Online Retail II and TransJakarta datasets are presented in Table 4.

Table 4. Example of RFM feature transformation under Min-Max and Z-Score normalization scenarios

Dataset	Original RFM Values	Min-Max Normalization	Z-Score Normalization	Interpretation
Online Retail II	Recency = 326, Frequency = 12, Monetary = 77,556.46	Recency = 0.4404, Frequency = 0.0277, Monetary = 0.1274	Recency = 0.5956, Frequency = 0.4390, Monetary = 5.0607	Min-Max places all features within the 0–1 range, while Z-Score shows that the Monetary value is far above the dataset average.
TransJakarta	Recency = 3, Frequency = 40, Monetary = 140,000	Recency = 0.0690, Frequency = 1.0000, Monetary = 0.1714	Recency = -0.6086, Frequency = 1.4440, Monetary = 0.4853	Min-Max shows that Frequency reaches the maximum observed value, while Z-Score shows the relative position of each feature from its mean.

Based on Table 4, Min-Max normalization successfully transformed all RFM attributes into a uniform value range, reducing the scale differences between features. Meanwhile, Z-Score normalization

describes the relative position of a value relative to the average of the corresponding feature. For example, the Monetary attribute in the Online Retail II dataset has a Z-Score of 5.0607, indicating that it is well above the average for Monetary. In the TransJakarta dataset, the Frequency value reached 1.0000 after Min-Max normalization, indicating that this data has the highest frequency value in the dataset.

Based on the transformation results, Min-Max and Z-Score normalization successfully reduced the scale differences between RFM attributes, especially for the Monetary attribute which has the largest value range. However, equalizing the feature scales does not automatically result in better cluster quality. The effect of normalization can vary depending on the data characteristics and the algorithm mechanism used. Therefore, the three data scenarios, namely without normalization, Min-Max normalization, and Z-Score normalization, were then used in the clustering process using K-Means and DBSCAN to compare the quality of the resulting clusters in each condition.

4.4 Clustering results using K-Means

This section presents the clustering result obtained using the K-Means algorithm on the datasets analyzed in this study.

4.4.1 K-Means results on the Online Retail II dataset

To determine the optimal number of clusters, values of k ranging from 2 to 10 were tested. The Silhouette Score was then used to evaluate the quality of each clustering configuration [23].

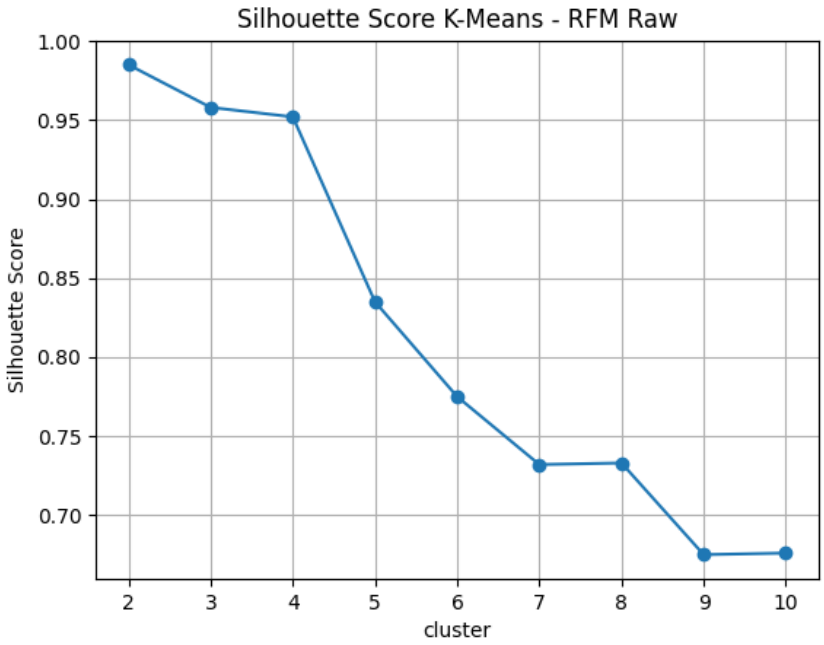

Figure 3. Silhouette score vs. number of clusters on the Online Retail dataset

Figure 3 shows the Silhouette Score values for various numbers of clusters across all normalization scenarios. In all three scenarios, the highest value was obtained when k = 2. As the number of clusters increased, the Silhouette Score tended to decrease, indicating that the cohesion and separation between clusters became less optimal. These results indicate that customer patterns in the Online Retail II dataset are more accurately represented by two main segments. Detailed evaluation results are presented in Table 5. The RFM data without normalization performed best with a Silhouette Score of 0.9845 and a Davies–Bouldin Index (DBI) of 0.3279. These values indicate that members within the same cluster share similar characteristics, while differences between clusters remain clear. The data normalized using Min-Max resulted in a Silhouette Score of 0.7225 and a DBI of 0.4304. Meanwhile, Z-Score normalization resulted in a Silhouette Score of 0.8958 and a DBI of 0.7450. Based on the DBI criterion, a lower value indicates more compact and well-separated clusters [11].

Although the optimal number of clusters remained the same, evaluation results showed differences between normalization scenarios. This is due to the Monetary attribute, which has a much larger range of values than Recency and Frequency, making it more dominant in distance calculations in the raw

data. Normalization balanced the scales of the three attributes, reducing the dominance of Monetary, and causing changes in cluster composition and evaluation scores. However, the overall segmentation pattern still produced the same optimal solution. A similar effect of feature scaling on distance-based clustering methods has also been reported in previous research. [14].

Table 5. Best K-Means results on the Online Retail II dataset

Scenario	Number of Clusters	Silhouette Score	DBI
Raw	2	0.9845	0.3279
Min-Max	2	0.7225	0.4304
Z-Score	2	0.8958	0.7450

4.4.2 K-Means results on the TransJakarta dataset

For the TransJakarta dataset, Silhouette Scores were evaluated for k values ranging from 2 to 10, as shown in Figure 4. This metric is used to identify cluster configurations with the best balance between cluster cohesion and separation, where higher values indicate better clustering quality.

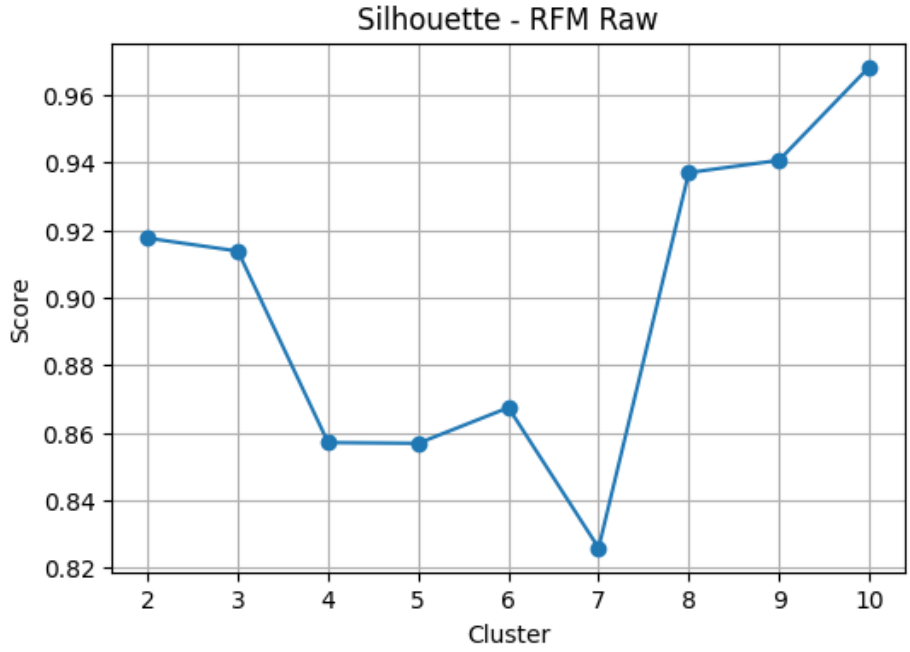

Figure 4. Silhouette score vs. number of clusters on the TransJakarta dataset

In Figure 4, the Silhouette Score fluctuates across different k values, indicating that the clustering performance is more sensitive to the choice of k, compared to the Online Retail II dataset. The best results on the TransJakarta dataset are obtained at higher k values, indicating a more diverse customer behavior pattern. Therefore, these two values were used as the most appropriate number of clusters for each scenario. A summary of the clustering results for each normalization scenario is shown in Table 6. For the unnormalized RFM data (Raw RFM), the most optimal configuration was obtained when the number of clusters was set at 10, resulting in a Silhouette Score of 0.9683 and a DBI of 0.1753. These results indicate that the formed clusters have a high degree of cohesion and clear separation between clusters. On the other hand, the application of Min-Max normalization produced the best performance at k = 9 with a Silhouette Score of 0.8662 and a DBI of 0.2047. Meanwhile, Z-Score normalization also produced an optimal number of clusters of 9, with a Silhouette Score of 0.8767 and a DBI of 0.2000. We also checked the validity of the clustering outcome using another internal clustering validation index known as Davies–Bouldin Index (DBI). This index rates clusters based on their compactness; the lower the score, the better the clustering result [19], [24], [25].

The results indicate that normalization influenced not only clustering quality but also the optimal number of clusters identified in the TransJakarta dataset. These findings also support the fact that feature scaling affects the performance of clustering especially with distance based algorithms like K-Means where features with different scales can distort the distance computations [12].

Table 6. Best K-Means results on the TransJakarta dataset

Scenario	Number of Clusters	Silhouette Score	DBI
Raw	10	0.9683	0.1753
Min-Max	9	0.8662	0.2047
Z-Score	9	0.8767	0.2000

4.5 Discussion of K-Means results across scenarios

Although the raw RFM data yielded the highest Silhouette Score, these results should be interpreted with caution. Table 2 shows that the Monetary attribute has a significantly larger range of values than Recency and Frequency. Because K-Means relies on Euclidean distance, attributes with larger numerical scales contribute more to the distance calculation. Consequently, customer segmentation in the raw dataset may be influenced primarily by spending behavior rather than by a balanced combination of all RFM dimensions. Several previous studies have revealed that differences in scale between features can affect the structure of the formed clusters and the evaluation value in distance-based clustering methods [11], [12], [13]. Therefore, the very high Silhouette Score (0.9845) obtained from the raw Online Retail II dataset needs to be interpreted with caution. This value does not necessarily reflect a more meaningful customer segmentation because Monetary attributes have a much larger range of values and allow for the dominance of distance calculations between data.

The application of Min-Max and Z-Score normalization reduces the scale differences among RFM attributes. While the evaluation score decreases after normalization, the contributions of Recency, Frequency, and Monetary become more balanced during cluster formation [14]. The shrinkage of the evaluation score reflects the more balanced contributions of all RFM attributes after normalization. Here, it is seen that a lower evaluation score does not necessarily indicate a worse segmentation quality, but rather reflects a more even merging of the clustering structure across all RFM dimensions. Similar observations have been reported in previous studies, which emphasized that normalization can change the cluster structure and evaluation results by reducing the dominance of large-scale features [2], [5].

4.5.1 Customer segment profile analysis

Figure 5 presents the average RFM values of the two clusters generated by K-Means.

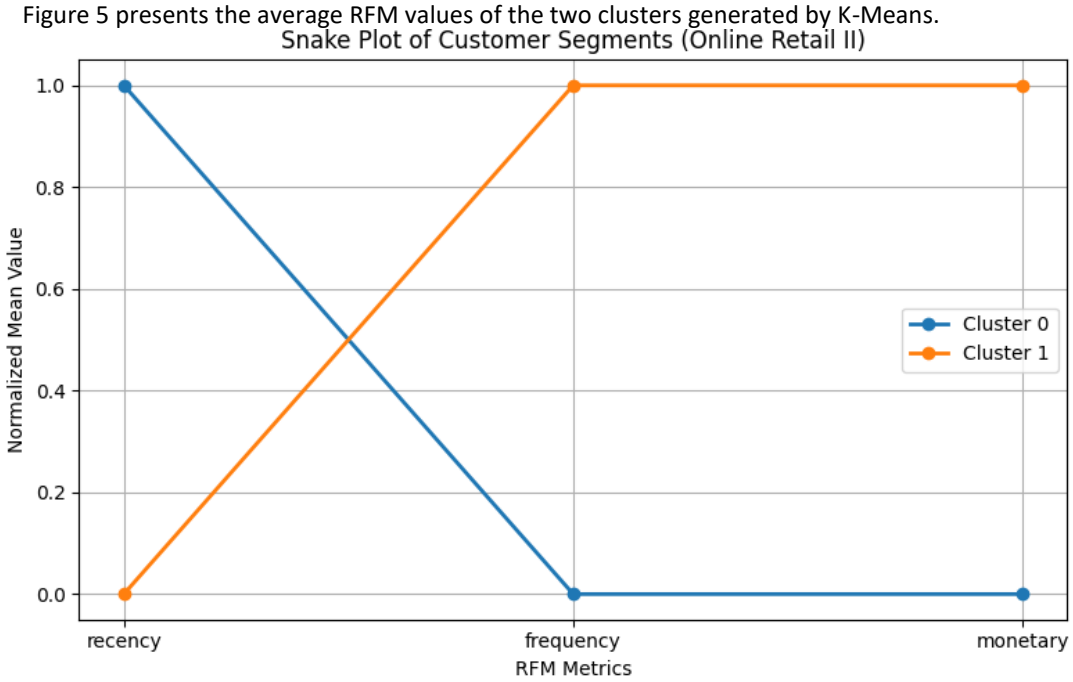

Figure 5. Snake plot of average normalized RFM values for customer segments in the Online Retail II dataset

Based on Figure 5, Cluster 0 consists of customers with high Recency values (92.70 days), low Frequency (4.17 transactions), and low Monetary Value (1,733.85). In contrast, Cluster 1 has low Recency values (6.25 days), high Frequency (61.13 transactions), and high Monetary Value

(172,453.10). The quite clear differences in these three attributes indicate that k = 2 can distinguish customers with different activities and transaction values in the Online Retail II dataset.

A similar segmentation pattern was also observed in the TransJakarta dataset. In both datasets, K-Means successfully separated customers based on their activity level and transaction value. Customers in one cluster tended to have lower transaction frequency and monetary value, while customers in the other cluster exhibited higher activity and spending behavior. These results demonstrate that K-Means can consistently identify meaningful differences in customer behavior across datasets.

4.6 Clustering results using DBSCAN

The clustering performance of the DBSCAN algorithm is evaluated using two different datasets. The results for each dataset are presented in the following subsections.

4.6.1 Clustering Results on the Online Retail II Dataset

The best clustering results for the Online Retail II dataset were obtained after data normalization using Z-Score and the application of DBSCAN parameters with eps = 1.0 and min_samples = 3. This process resulted in three data clusters. Based on the evaluation results, the resulting clustering had a Silhouette Score of 0.8036 and a DBI value of 2.1800. These results also indicate that parameter selection influences the clustering process. At eps = 1.0 and min_samples = 3, DBSCAN can identify areas with higher densities than the normalized data and separate them into three clusters. Using a smaller eps value tends to result in fragmented clusters or increase the amount of data identified as noise. Conversely, a larger eps value often causes observations to be grouped into one cluster, thus reducing the effectiveness of the clustering process.

In table 7 shown that a Silhouette Score of 0.8036 indicates that the separation between clusters was fairly good. On the other hand, the DBI value remained relatively high at 2.1800, showing that the data points within each cluster were still quite diverse. As a result, the clusters could be distinguished from one another, but their internal structure was not entirely compact. This condition may be related to the wide range of transaction values present in the dataset [1], [18].

Only the best result is discussed in this section because the comparison was carried out across all normalization scenarios. From the three scenarios tested, valid clusters were obtained only in the Z-Score normalization scenario. In the RFM Raw and Min-Max scenarios, most of the clustering results consisted of a single cluster, while several configurations produced undefined Silhouette Score values (NaN), making the evaluation metrics unavailable.

These results show that the performance of DBSCAN on the Online Retail II dataset was highly influenced by the scaling of the input features. When the differences in feature values were too large, the density patterns required by DBSCAN became more difficult to identify. This explains why meaningful clustering structures were not formed in several scenarios without appropriate normalization.

Table 7. Best DBSCAN results on the Online Retail II dataset

Scenario	Number of Clusters	Silhouette Score	DBI	eps	min_samples
Z-Score	3	0.8036	2.1800	1.0000	3

4.6.2 Clustering Results on the TransJakarta Dataset

The test results show that the DBSCAN algorithm is able to form valid clusters in all normalization scenarios applied to the TransJakarta dataset. On RFM data without normalization, the best results were obtained when using the parameters eps = 2.0 and min_samples = 3. With this combination of parameters, DBSCAN produced 20 clusters and 6 noise points. The evaluation results showed a Silhouette Score of 0.9614 and a Davies-Bouldin Index (DBI) of 0.3638. Although the Monetary attribute has a wider range of values compared to Recency and Frequency, eps = 2.0 can still detect data density. This is indicated by the elbow point on the k-distance graph that is located around this value, therefore DBSCAN can be used as a suitable area. Furthermore, transactions on TransJakarta are also more similar because they use relatively similar fares. This results in local density patterns still being detectable despite the different scales between features.

This is reflected in the formation of 20 clusters and the high silhouette score obtained in the raw data scenario. Therefore, the selected eps value appears to be sufficient to capture dense regions in the TransJakarta dataset despite differences in feature scale. The high Silhouette Score value followed by a relatively low DBI value indicates that the formed clusters have a good level of synchronization and

fairly clear boundaries between clusters. However, the presence of several noise points indicates that there are some data that are not included in any cluster [1], [18].

As shown in table 8, when Min-Max normalization was applied, the clustering changed. DBSCAN then created three clusters with no noise points, giving a Silhouette Score of 0.7822 and a DBI of 0.2172. Surprisingly, while the clusters were less separated, they were more compact based on the DBI value. This result suggests that Min-Max normalization reduced the effect of large-scale attributes, allowing observations with similar characteristics to be grouped more closely.

As a result, cluster compactness improved, although separation between clusters was lower than in the raw data scenario.

After applying Z-Score normalization, the clustering results formed only two clusters. The Silhouette Score decreased to 0.7035, while the DBI increased to 0.2419. While these metrics are not significantly better than before, these results are still sufficient to describe the data patterns. The smaller number of clusters indicates that the data has become more concentrated after normalization, resulting in a simpler clustering structure. Furthermore, the influence of extreme values is also reduced in the clustering process.

Thus, the normalization scheme affects the number of clusters generated by DBSCAN. However, a valid clustering structure is obtained in all scenarios. The density patterns in the TransJakarta dataset remain quite distinguishable despite changes in feature scaling. Despite the differences in evaluation scores, DBSCAN was still able to produce valid clustering structures across all preprocessing scenarios.

Table 8. Best DBSCAN results on the TransJakarta dataset

Scenario	Number of Clusters	Silhouette Score	DBI	eps	min_samples	Noise
Raw	20	0.9614	0.3638	2.0000	3	6
Min-Max	3	0.7822	0.2172	0.5000	3	0
Z-Score	2	0.7035	0.2419	3.0000	3	0

4.7 Discussion of DBSCAN results across normalization scenarios

Tests conducted on both datasets show that the data characteristics and the parameter settings for eps and min_samples have an impact on the DBSCAN results [17]. These results are in line with the research of Bushra et al. [7] which explains that DBSCAN is susceptible to changing parameters. They reported that DBSCAN is highly sensitive to the choice of parameters eps and minPts. DBSCAN forms clusters based on data density, determined by the parameters eps and minPts. Therefore, changes in feature scale can alter the density patterns in the dataset, ultimately affecting the clustering results and evaluation scores obtained. This observation is consistent with the results of Online Retail II, where valid clusters were obtained only after applying Z-Score normalization and selecting an appropriate parameter configuration.

The differences in data distribution and density in the two datasets cause different clustering results because DBSCAN forms clusters based on data density. Therefore, the quality of the cluster results obtained is directly influenced by the changing distribution patterns [16], [18].

In the Online Retail II dataset, the best performance was achieved after data normalization using the Z-Score method with an eps value of 1.0 and a min_samples value of 3. Based on this configuration, DBSCAN formed three clusters with a Silhouette Score of 0.8036 and a DBI of 2.1800. The relatively high Silhouette Score indicates that objects in each cluster have a good level of similarity and are quite separate from other clusters. However, the relatively high DBI values indicate that data variation within the clusters is not completely reduced [1], [18]. Furthermore, the results indicate that changing the normalization method affects clustering quality. The above explanation shows that the normalization process plays a role in forming data density patterns. Data density is then used by DBSCAN to determine clusters [26]. The results show that DBSCAN is more sensitive to feature scaling than K-Means on the Online Retail II dataset. Without normalization, large differences in the scale of RFM attributes can allow for deviations in the data density distribution, making it difficult for DBSCAN to identify meaningful dense regions.

On the other hand, in the Raw and Min-Max schemes, DBSCAN tends to produce a single, vague cluster. This indicates that the data density patterns cannot be clearly separated. This result indicates that changes in feature scale make it difficult for DBSCAN to form valid clusters.

The DBSCAN algorithm on the TransJakarta dataset produced valid clusters across all normalization schemes. The highest Silhouette Score was obtained for the original data, with a score of 0.9614 and a DBI of 0.3638. However, the results were divided into 20 clusters with some noise, resulting in a somewhat fragmented structure. This demonstrates that accurate segmentation results are not always due to a high evaluation score.

Furthermore, DBSCAN produced three noise-free clusters after Min-Max normalization, with a Silhouette Score of 0.7822 and a DBI of 0.2172. Using the Z-Score scheme, two clusters were formed, with a Silhouette Score of 0.7035 and a DBI of 0.2419. Despite the decrease in the Silhouette Score, the clustering results appeared neater and less fragmented.

In general, evaluation scores are not always improved through normalization, but rather tend to update the cluster structure to make it more stable and easier to summarize. The results of the research that has been conducted also support the results of previous research which explains that normalization plays a greater role in forming clusters, not only making the evaluation metric value increase immediately [13], [27].

The essence of this research is to prove that different feature scales and how the distribution of data density is formed can affect the performance of DBSCAN. Clusters in the Online Retail II dataset that met the evaluation criteria could only be obtained after Z-Score normalization, meaning that cluster structure formation could be disrupted by differences in feature scale.

In the TransJakarta dataset, the clustering results were valid across all normalization schemes. This indicates that the data distribution is more stable and therefore less susceptible to changes in feature scale.

From these results, it can be concluded that normalization does not always improve evaluation scores. Its effect depends heavily on the combination of dataset characteristics, scaling method, and DBSCAN's eps and min_samples parameter settings. Therefore, careful selection of normalization techniques, eps values, and min_samples values is necessary to obtain meaningful clustering results that accurately reflect the underlying data structure [6], [18], [25].

4.8 Comparative analysis of K-Means and DBSCAN

The test results show various differences when applying K-Means and DBSCAN to RFM data as shown in table 9. Before carrying out the clustering process, most K-Means have determined the number of clusters so that it obtains a more stable clustering. Finally, the formed cluster structure becomes more stable, especially when the data has a nearly uniform distribution [6].

On the one hand, the test results also show one drawback of K-Means that was noticed during the test, namely that there is no procedure related to separating data that deviates from the main cluster. All existing data will remain placed in one of the existing clusters, including observations that can be considered outliers, and ultimately the clustering results obtained are affected by the presence of extreme data.

The DBSCAN procedure differs from K-Means in that the cluster formation process is based on the level of data density, not a predetermined number of clusters [15]. With this approach, the algorithm can detect areas with high data concentration, while simultaneously separating sparsely distributed data and classifying it as noise if necessary. On the TransJakarta dataset, this approach shows fairly consistent clustering performance. Various configurations tested can obtain high Silhouette Score values and minimal DBI values. This indicates that the formed clusters have fairly clear boundaries and denser and more similar cluster members [6], [18].

No similar conditions were found in the Online Retail II dataset. During the testing process, valid clustering results were only obtained with certain parameter settings. Meanwhile, several other parameter settings failed to produce good-quality clustering. This indicates that DBSCAN's performance is more susceptible to changes in data distribution, and therefore, more careful clustering parameter settings are essential to obtain optimal clustering results that can be used for analysis.

Table 9. Overall summary of clustering results

Dataset	Scenario	Algorithm	Number of Clusters	Silhouette Score	DBI	Notes
Online Retail II	Raw	K-Means	2	0.9845	0.3279	Highly separated clusters; however, unnormalized data may introduce scale bias
Online Retail II	Min-Max	K-Means	2	0.7225	0.4304	Cluster quality decreases due to uniform rescaling of distribution
Online Retail II	Z-Score	K-Means	2	0.8958	0.7450	Clusters are relatively stable, but inter-cluster separation weakens
TransJakarta	Raw	K-Means	10	0.9683	0.1753	Well-formed clusters, but a higher number of clusters tends to emerge
TransJakarta	Min-Max	K-Means	9	0.8662	0.2047	More balanced clusters after Min-Max normalization
TransJakarta	Z-Score	K-Means	9	0.8767	0.2000	Stable results with good intra-cluster compactness
Online Retail II	Z-Score	DBSCAN	3	0.8036	2.1800	DBSCAN successfully forms clusters, yet high intra-cluster variance persists
TransJakarta	Raw	DBSCAN	20	0.9614	0.3638	Clusters are highly distinct, but sensitive to eps parameter settings
TransJakarta	Min-Max	DBSCAN	3	0.7822	0.2172	More compact clusters with minimal noise
TransJakarta	Z-Score	DBSCAN	2	0.7035	0.2419	Simpler cluster structure with reduced data complexity

The results of this study reinforce several previous studies that explain that the feature scaling process can impact clustering performance, particularly in algorithms that use distance calculations as the basis for clustering [11], [12]. Several studies related to RFM-based customer segmentation also show that clustering techniques can effectively divide customer groups based on transaction patterns [1], [5], [9], [14]. However, most previous studies focused primarily on segmentation results, clustering algorithm comparisons, or the use of a single dataset [1], [5], [9].

This study attempts to provide a different approach by comparing the performance of RFM data without normalization, Min-Max normalization, and Z-Score normalization using two datasets with different transaction characteristics and two clustering algorithms. The clustering results obtained from both datasets indicate that normalization does not always produce better performance. In some experiments, particularly those involving K-Means, the original data yielded higher Silhouette Scores and lower Davies-Bouldin Index values than the normalized versions. This finding suggests that the clustering structure present in the raw data is sufficient to support cluster formation without additional scaling.

When DBSCAN was applied, the clustering results obtained differed from those obtained with K-Means. The DBSCAN clustering results showed that each normalization scheme had greater changes and its performance was more susceptible to changes that occurred in the data after the pre-processing stage. This can be seen in the Online Retail II dataset, where when using Z-Score normalization, clusters that met all evaluation criteria were successfully obtained. Meanwhile, with other normalization methods, DBSCAN struggled to form clusters with adequate quality. These results indicate that DBSCAN's success is greatly influenced by the distribution and density of the data formed after the pre-processing stage is completed.

Test results show that the benefits of normalization in the clustering process vary across problems. The effect of normalization on clustering quality can vary depending on the characteristics of the dataset being analyzed and the algorithm used. Therefore, it is recommended that pre-processing stages not be applied with the same approach. Adjustments to the dataset's conditions when selecting the appropriate technique are necessary for more accurate clustering results.

The results of this study help address the problem discussed in the Introduction, namely the limited research comparing the effects of several normalization methods on datasets with different transaction characteristics and the use of different clustering algorithms. The test results indicate that clustering quality is influenced by a combination of data characteristics, normalization methods, and the algorithm used.

During the testing process, the algorithm that produced valid clustering results across all datasets and normalization schemes was K-Means. This indicates that K-Means tends to be more stable over changing data scales. However, because each data item must be assigned to a cluster, K-Means lacks a specific method for distinguishing outliers or noise.

In contrast, DBSCAN differs in that it can automatically detect and separate noise without requiring an initial number of clusters. However, clustering results are more dependent on the selected parameters and data distribution patterns. This can be seen in the Online Retail II dataset, where clustering results that met the evaluation criteria were only found in the Z-Score normalization scenario. Meanwhile, in the TransJakarta dataset, all normalization schemes produced valid clusters.

Overall, the research results show that K-Means is superior in terms of clustering stability across various data conditions. Conversely, DBSCAN is superior in handling noise and outliers, but requires more careful parameter selection because the characteristics of the dataset used affect the quality of the clustering results.

5. Conclusion

The results show that the effect of normalization on RFM-based customer segmentation depends on both dataset characteristics and the clustering algorithm used. In K-Means, the best clustering performance was achieved on raw data for the Online Retail II dataset, whereas normalization contributed to a more balanced representation of RFM attributes. In contrast, DBSCAN was more sensitive to feature scaling, resulting in different clustering structures across normalization scenarios. These findings highlight the important role of normalization in shaping cluster structures rather than consistently improving clustering evaluation metrics.

This study contributes empirical evidence that the effectiveness of normalization in RFM-based customer segmentation is both dataset-dependent and algorithm-dependent. By comparing three preprocessing scenarios (raw, Min-Max, and Z-Score normalization) across two datasets with different transaction characteristics and two clustering algorithms (K-Means and DBSCAN), the results provide practical guidance for selecting appropriate preprocessing strategies in customer segmentation applications. Nevertheless, this study is limited to two datasets, two normalization methods, and two clustering algorithms. Future research may investigate additional preprocessing techniques, such as logarithmic transformation and robust scaling, as well as alternative clustering methods, including HDBSCAN, Gaussian Mixture Models, and Spectral Clustering, using datasets from broader domains.

Credit Authorship Contribution Statement

Nabilah Aulia Zahra : Conceptualization, Methodology, Software, and Project administration.**Tyas Sekar Arum:** Software and Writing – original draft. **Zerafica Adhyasti Dinar Patriawan**: Writing – review & editing. **Dwika Ananda Agustina Pertiwi**: Validation. **Much Aziz Muslim**: Supervision.**Yusuf Enril Faturrohman:** Investigation & Resources.

Declaration of Competing Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Use of Artificial Intelligence

During the preparation of this manuscript, the authors used artificial intelligence (AI) solely to improve the language, grammar, and readability of the manuscript. AI was not used to develop the research methodology, perform data preprocessing, implement the clustering algorithms, conduct the experiments, analyze the results, or draw the conclusions. All scientific content was developed, reviewed, and approved by the authors. The authors take full responsibility for the content of this manuscript.

References

[1] G. Aslantaş, M. Gençgül, M. Rumelli, and M. Özsaraç, "Customer Segmentation Using K-Means Clustering Algorithm and RFM Model K- Means Kümeleme Algoritması ve RFM Modeli Kullanarak Müşteri Segmentasyonu," vol. 25, no. 74, pp. 491–503, 2023, doi: 10.21205/deufmd.2023257418.

[2] P. Li, C. Wang, and J. Wu, "An E-commerce Customer Segmentation Method based on RFM Weighted K-means," *2022 Int. Conf. Manag. Eng. Softw. Eng. Serv. Sci.*, pp. 61–68, 2022, doi: 10.1109/ICMSS55574.2022.00017.

[3] A. L. Corrêa Vianna Filho, L. de Lima, and M. Kleina, "RFM model customer segmentation from a graph theory perspective," *Qual. Quant.*, 2026, doi: 10.1007/s11135-026-02784-0.

[4] C. Rungruang, P. Riyapan, A. Intarasit, K. Chuarkham, and J. Muangprathub, "RFM model customer segmentation based on hierarchical approach using FCA," *Expert Syst. Appl.*, vol. 237, no. PB, p. 121449, 2024, doi: 10.1016/j.eswa.2023.121449.

[5] J. M. John and O. Shobayo, "An Exploration of Clustering Algorithms for Customer Segmentation in the UK Retail Market," pp. 809–823, 2023.

[6] B. G. Keller, "Clustering—Basic concepts and methods," 2022.

[7] A. A. Bushra, D. Kim, Y. Kan, and G. Yi, "AutoSCAN: automatic detection of DBSCAN parameters and efficient clustering of data in overlapping density regions," *PeerJ Comput. Sci.*, vol. 10, 2024, doi: 10.7717/peerj-cs.1921.

[8] I. Izonin, R. Tkachenko, N. Shakhovska, B. Ilchyshyn, and K. K. Singh, "A Two-Step Data Normalization Approach for Improving Classification Accuracy in the Medical Diagnosis Domain," *Mathematics*, vol. 10, no. 11, pp. 1–18, 2022, doi: 10.3390/math10111942.

[9] C. Wong, "Exploring Customer Segmentation in E-Commerce using RFM Analysis with Clustering Techniques," vol. 12, no. 3, pp. 97–125.

[10] H. J. Wilbert, A. F. Hoppe, A. Sartori, S. F. Stefenon, and L. A. Silva, "and External Indices for Customer Segmentation from Retail Data," 2023.

[11] C. Wongoutong, "The impact of neglecting feature scaling in k-means clustering," pp. 1–19, 2024, doi: 10.1371/journal.pone.0310839.

[12] A. Saxena, M. Prasad, A. Gupta, N. Bharill, and O. Prakash, "Neurocomputing," *Neurocomputing*, vol. 267, pp. 664–681, 2017, doi: 10.1016/j.neucom.2017.06.053.

[13] I. Niño-adan, I. Landa-torres, E. Portillo, and D. Manjarres, "Engineering Applications of Artificial Intelligence Influence of statistical feature normalisation methods on K-Nearest Neighbours and K-Means in the context of industry 4 . 0," *Eng. Appl. Artif. Intell.*, vol. 111, no. March, p. 104807, 2022, doi: 10.1016/j.engappai.2022.104807.

[14] P. Anitha and M. M. Patil, "RFM model for customer purchase behavior using K-Means algorithm," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 5, pp. 1785–1792, 2022, doi:

10.1016/j.jksuci.2019.12.011.
[15] M. Ester, H. Kriegel, X. Xu, and D.- Miinchen, “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise,” 1996.
[16] M. Hahsler, M. Piekenbrock, and D. Doran, “dbscan : Fast Density-Based Clustering with R,” vol. 91, no. 1, 2019, doi: 10.18637/jss.v091.i01.
[17] M. Ester, “DBSCAN Revisited , Revisited : Why and How You Should (Still) Use DBSCAN,” vol. 42, no. 3, 2017.
[18] O. Arbelaitz, I. Gurrutxaga, and J. Muguerza, “An extensive comparative study of cluster validity indices,” vol. 46, pp. 243–256, 2013, doi: 10.1016/j.patcog.2012.07.021.
[19] P. J. Rousseeuw, “Silhouettes : a graphical aid to the interpretation and validation of cluster analysis,” vol. 20, pp. 53–65, 1987.
[20] S. Romero and K. Becker, “A framework for event classification in tweets based on hybrid semantic enrichment,” *Expert Syst. Appl.*, vol. 118, pp. 522–538, 2019, doi: 10.1016/j.eswa.2018.10.028.
[21] V. Krishna, “ScienceDirect ScienceDirect NSE Stock Stock Market Market Prediction Prediction Using Using Deep-Learning Deep-Learning Models Models,” *Procedia Comput. Sci.*, vol. 132, no. Iccids, pp. 1351–1362, 2018, doi: 10.1016/j.procs.2018.05.050.
[22] N. El, M. Oudghiri, R. El, N. El, M. Oudghiri, and R. El, “ScienceDirect Vehicle Vehicle lateral lateral dynamics dynamics estimation estimation using using unknown unknown input input observer observer,” *Procedia Comput. Sci.*, vol. 148, pp. 502–511, 2019, doi: 10.1016/j.procs.2019.01.063.
[23] M. Gagolewski, M. Bartoszuk, and A. Cena, “Are cluster validity measures (in) valid ?,” *Inf. Sci. (Ny).*, vol. 581, pp. 620–636, 2021, doi: 10.1016/j.ins.2021.10.004.
[24] H. Zhong, H. Zhang, and F. Jia, “on Collaborative Computing EAI Endorsed Transactions Analysis and improvement of evaluation indexes for clustering results,” pp. 1–8, doi: 10.4108/eai.9-10-2017.163211.
[25] P. Fränti and S. Sieranoja, “How much can k-means be improved by using better initialization and repeats ?,” vol. 93, pp. 95–112, 2019, doi: 10.1016/j.patcog.2019.04.014.
[26] R. Corizzo, G. Pio, M. Ceci, and D. Malerba, “DENCAST : distributed density - based clustering for multi - target regression,” *J. Big Data*, 2019, doi: 10.1186/s40537-019-0207-2.
[27] S. N. Auliani and R. Novita, “The Indonesian Journal of Computer Science,” vol. 13, no. 4, pp. 5115–5132, 2024.