Performance Comparison of K-Means and Hybrid Hierarchical–Partitioning Methods for Clustering Efficiency

Bowo Winarno[1,*], Budi Warsito[2], Bayu Surarso[3]

[1,2,3]*Doctoral Program of Information System, School of Postgraduate Studies, Diponegoro University, Semarang, 50275, Indonesia*

[1]*Department of Mathematics, Faculty of Mathematics and Natural Sciences, Sebelas Maret University, Surakarta, 57126, Indonesia*

[2]*Department of Statistics, Faculty of Science and Mathematics, Diponegoro University, Semarang, 50275, Indonesia*

[3]*Department of Mathematics, Faculty of Science and Mathematics, Diponegoro University, Semarang, 50275, Indonesia*

(Received: January 5, 2026; Revised: March 10, 2026; Accepted: May 20, 2026; Available online: June 28, 2026)

Abstract

Clustering is a fundamental technique in data analysis, particularly for exploring patterns in large-scale datasets. While K-Means is widely used for its simplicity and efficiency, its performance is highly sensitive to centroid initialization, which can affect both clustering quality and convergence speed. Hierarchical clustering methods, such as agglomerative and divisive approaches, provide more structured and deterministic initialization but incur higher computational cost. This study evaluates two hybrid models—Agglomerative K-Means and Divisive K-Means—where hierarchical clustering is used to initialize centroids, followed by K-Means refinement. This approach aims to reduce the limitations of random initialization while improving clustering stability and efficiency in large-scale data environments. Experiments on poverty data from Central Java Province show that hybrid methods accelerate K-Means convergence: Agglomerative K-Means reduced iterations to 2 (from 3 in standard K-Means), while Divisive K-Means converged in 1 iteration. Silhouette, Davies–Bouldin, and Calinski–Harabasz indices indicate that Agglomerative K-Means achieves the most compact and well-separated clusters, whereas Divisive K-Means performed worse than standard K-Means. Execution time measured only during the K-Means refinement phase shows that hybrids converge faster (Agglomerative: 2.04 ms; Divisive: 1.91 ms; K-Means: 116.68 ms), though this does not account for the hierarchical initialization cost. These findings provide practical insights into the trade-offs between clustering quality and computational efficiency when applying hybrid clustering methods. Overall, these results demonstrate that hybrid approaches can improve clustering stability and convergence efficiency, with Agglomerative K-Means providing the best balance between cluster quality and computational performance.

Keywords: Clustering, K-Means, Hybrid Hierarchical–Partitioning, Execution Time, Centroid Optimization

1. Introduction

In the era of big data, the rapid growth in the volume, variety, and complexity of data has posed significant challenges for data analysis techniques. One of the most widely used approaches to address these challenges is clustering, an unsupervised learning method that aims to uncover hidden structures and patterns in large, complex, and often unlabeled datasets. Clustering plays a crucial role in various domains, including healthcare, marketing, financial analytics, and socio-economic analysis, where accurate data segmentation is essential for informed decision making and policy development [1], [2], [3], [4], [5].

Among existing clustering techniques, K-Means has become one of the most popular algorithms due to its simplicity, scalability, and computational efficiency. It is particularly suitable for large datasets, making it a common choice in big data applications [6], [7], [8], [9]. However, despite these advantages, K-Means suffers from a critical limitation: its dependence on random centroid initialization. This limitation often leads to unstable clustering results, sensitivity to initial conditions, and convergence to local minima, especially when dealing with high-dimensional or complex datasets [6], [10], [11], [12], [13].

To address these challenges, researchers have proposed various improvements and alternative approaches. One promising direction is the integration of hierarchical clustering with partitioning methods, forming hybrid clustering techniques. These approaches aim to combine the strengths of hierarchical methods such as their ability to capture

***Corresponding author: Bowo Winarno (bowowinarno@students.undip.ac.id)**
DOI: https://doi.org/10.47738/jads.v7i3.1140

global data structure with the efficiency of partitioning algorithms like K-Means [14], [15], [16], [17], [18]. By using hierarchical clustering to generate more representative initial centroids, hybrid methods can improve clustering stability and reduce sensitivity to random initialization [19], [20], [21], [22].

However, despite the increasing adoption of hybrid clustering techniques, several issues remain unresolved. In particular, the trade-off between improved centroid initialization and additional computational overhead introduced by hierarchical processing has not been fully explored [15], [18], [23]. Furthermore, limited studies have provided a direct comparison between different hierarchical–partitioning strategies, such as agglomerative and divisive approaches, especially in terms of clustering efficiency and execution time [16], [20], [24].

Therefore, this study aims to compare the performance of standard K-Means with two hybrid approaches: Agglomerative K-Means and Divisive K-Means. The comparison focuses on execution time as a key performance metric, while also considering convergence behavior and clustering efficiency. By analyzing these methods, this research seeks to provide insights into the effectiveness of hierarchical-based centroid initialization and its impact on computational performance in large-scale data environments [7], [9], [25]. Consequently, the sensitivity of K-Means to initialization remains a significant challenge, particularly in big data environments where efficiency and robustness are equally important [13], [25], [26], [27], [28].

2. Literature Review

2.1. Theoretical Background of Clustering Methods

Clustering is a fundamental technique in unsupervised learning that aims to group data objects based on similarity, such that objects within the same cluster are more similar to each other than to those in different clusters [6], [7], [8], [9]. In the context of big data, clustering plays a crucial role in extracting meaningful patterns from large, complex, and often unlabeled datasets across various domains, including healthcare, marketing, and financial analytics [1], [2], [3], [4], [5].

Among various clustering techniques, K-Means has been widely adopted due to its simplicity, scalability, and computational efficiency [10], [11], [12], [13]. The algorithm partitions data into k clusters by minimizing intra-cluster variance, typically measured using the sum of squared errors. Despite its advantages, K-Means is highly sensitive to the initial placement of centroids, which are usually selected randomly. This limitation often leads to unstable clustering results and convergence to local minima, particularly in high-dimensional or complex datasets [10], [14], [15], [16], [17], [18].

Hierarchical clustering provides an alternative approach by constructing a nested hierarchy of clusters, typically represented in the form of a dendrogram [3], [6], [7], [9], [29]. This method does not require predefined cluster numbers and offers a global view of data structure. Agglomerative (bottom-up) and divisive (top-down) strategies are the two main types of hierarchical clustering. Although hierarchical methods provide better interpretability and more stable clustering structures, they suffer from high computational complexity, making them less suitable for large-scale datasets [2], [30], [31], [32], [33].

2.2. Literature on Hybrid Hierarchical-Partitioning Clustering

Recent studies have explored hybrid clustering approaches that integrate hierarchical and partitioning methods to overcome the limitations of individual techniques [34], [35], [36], [37], [38], [39], [40], [41], [42], [43]. In general, these approaches utilize hierarchical clustering to generate more representative initial centroids, which are subsequently refined using efficient partitioning algorithms such as K-Means.

This hybridization has been shown to improve clustering stability, reduce sensitivity to random initialization, and enhance overall clustering quality [25], [34], [40], [44], [45]. By leveraging the global structural insight of hierarchical clustering and the computational efficiency of K-Means, hybrid methods achieve a balance between accuracy and scalability, which is essential in big data environments. To assess these improvements, various cluster validity indices are commonly employed, including Silhouette Score, Davies–Bouldin Index, and Calinski–Harabasz Index, which measure cluster cohesion and separation [46].

Several variations of hybrid clustering have been proposed in the literature. For instance, agglomerative-based initialization methods generate initial clusters using bottom-up merging strategies, which are then refined using K-Means to optimize centroid positions. Conversely, divisive-based approaches employ a top-down splitting strategy to

form initial clusters before applying partitioning refinement. These complementary strategies allow hybrid methods to capture both global and local data structures more effectively [27], [34], [38].

However, not all hybrid approaches are equally effective for large-scale structured data. Density-based hybrid methods, while capable of identifying arbitrarily shaped clusters, often require complex parameter tuning and may perform poorly in high-dimensional or heterogeneous datasets [3], [8]. In contrast, hierarchical–partitioning hybrids based on agglomerative and divisive strategies provide more deterministic and structure-preserving initialization processes, making them more suitable for structured datasets such as socio-economic or regional data [27], [34], [38].

3. Methodology

3.1. Dataset

This study employs the poverty dataset of Central Java Province obtained from the Indonesian Central Bureau of Statistics (BPS) in 2024 (table 1) [47]. The dataset consists of records from 35 districts and municipalities within the province and includes nine socio-economic indicators representing multidimensional aspects of poverty.

Table 1. Employs the poverty dataset of Central Java Province

Regency	Percentage of poor population (%)	Did Not/Have Not Completed Primary School (>15 Years Old)	Literacy Rate (15–55 Years Old)	Scol Participation Rate (13–15 Years Old)	Not Employed (>15 Years Old)	Employed in the Agricultural Sector (>15 Years Old)	Employed in the Informal Sector (>15 Years Old)	Per Capita Monthly Expenditure on Food Commodities	Using Private/ Shared Toilet
Cilacap	10.68	17.76	95.28	95.37	39.54	32.25	38.96	65.66	93.88
Banyumas	11.95	15.25	95.91	99.59	38.05	14.93	37	64.03	91.56
Purbalingga	14.18	18.45	93.87	94.77	32.05	19.04	39.89	63.98	94.16
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Semarang City	4.03	5.86	97.88	99.98	34.29	0.71	21.06	55.39	94.99
Pekalongan City	6.71	8.47	98.69	96.52	31.37	1.67	25.6	62.7	83.97
Tegal City	7.64	13.54	97.94	98.56	36.7	4.47	23.94	60.63	94.61

This dataset was selected because it represents a real-world case of high-dimensional, imbalanced, and unlabeled socio-economic data that requires accurate clustering to support regional poverty reduction policies and resource allocation [2], [5], [47]. In addition to its multidimensional nature, preliminary analysis indicates that the dataset exhibits heterogeneous distributions across variables, where certain regions show significantly higher values in indicators such as poverty rate, agricultural employment, and informal sector participation. This suggests the presence of clusters with varying densities and potentially imbalanced sizes, rather than perfectly uniform or spherical group structures.

In addition to the regional dataset, supplementary testing was conducted using benchmark datasets from the UCI Machine Learning Repository to ensure the generalizability of the proposed methods across different domains [11], [48]. Benchmark datasets are widely used to evaluate clustering algorithms under standardized conditions, allowing validation of consistency, accuracy, and adaptability [10], [12].

Before performing clustering, data preprocessing was conducted to handle missing or incomplete values. The dataset contained several entries with “NA” (Not Applicable), particularly in socio-economic indicators such as employment data. Rather than deleting records with missing values, which may lead to the loss of meaningful information and distortion of data distribution, this study applied imputation techniques to replace missing entries with estimated values derived from existing data patterns [49], [50], [51], [52].

Such characteristics are particularly relevant when applying K-Means, which assumes clusters to be spherical and relatively balanced in size due to its reliance on distance-based partitioning [10], [11]. In datasets like this, where socio-economic disparities vary significantly across regions, K-Means may produce suboptimal clustering results, such as biased centroid placement or improper grouping of regions with distinct profiles [10], [53]. This limitation motivates the need for more robust initialization strategies, such as those provided by hierarchical-based hybrid methods.

 ISSN 2723-6471

Moreover, the dataset reflects structured regional relationships, where geographically or economically similar districts may form natural groupings. This makes it well-suited for hierarchical clustering approaches, which can capture such global structures more effectively than purely partitioning-based methods [27], [34]. Therefore, the combination of hierarchical and partitioning techniques is particularly appropriate for this type of data.

Specifically, missing values were imputed using the mean of the corresponding variable, a widely adopted statistical method in unsupervised learning tasks [49], [51]. The choice of mean imputation in this study is motivated by several considerations. First, the dataset consists of numerical socio-economic indicators where the mean provides a simple yet effective representation of central tendency without introducing additional model complexity. Second, clustering algorithms such as K-Means rely on distance calculations, making mean imputation particularly suitable as it preserves the geometric structure of the data and avoids biasing cluster centroids [10], [53]. Third, compared to more complex imputation techniques, mean imputation offers lower computational cost, which is essential in the context of big data processing and scalability [50], [51].

Moreover, preliminary data inspection indicated that the proportion of missing values in the dataset was relatively small and randomly distributed, reducing the risk of systematic bias when applying mean substitution [49], [52]. In such conditions, mean imputation has been shown to provide stable and reliable results without significantly affecting clustering performance. Therefore, this approach ensures that the dataset remains complete, consistent, and suitable for subsequent clustering analysis while maintaining computational efficiency and methodological simplicity.

3.2. Methods

This section outlines the methodology, the hybrid hierarchical–partitioning approach, as illustrated in the flowchart in Figure 1.

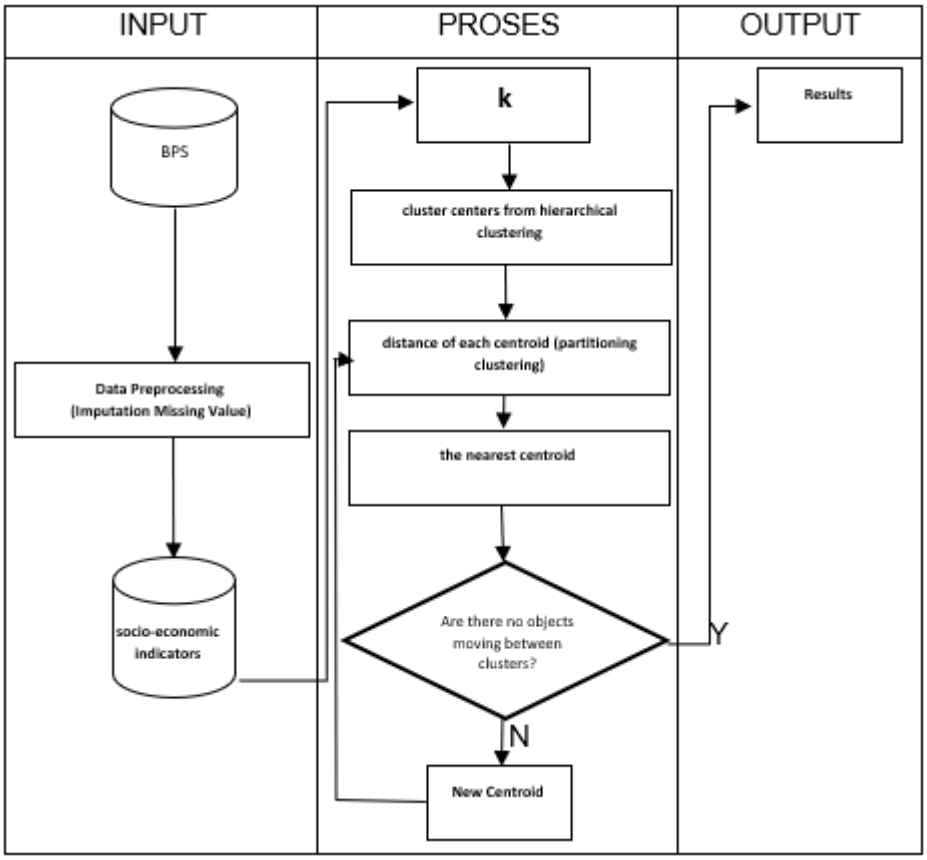

Figure 1. The hybrid hierarchical–partitioning flowchart

The K-Means algorithm is one of the most widely used partitioning methods in cluster analysis due to its simplicity and computational efficiency. Originally introduced by MacQueen (1967) [14], the algorithm aims to partition a dataset into *k* non-overlapping clusters such that objects within the same cluster are more similar to each other than to those in different clusters. The clustering process is based on minimizing the Within-Cluster Sum of Squared Errors (SSE), which serves as the objective function [11], [48]. The pseudocode of K-Means involves the following steps [10], [48]:

Algorithm 1. Standard K-Means Clustering

Input: Dataset X, number of clusters k
Output: Cluster labels
1. Initialize k centroids randomly
2. Repeat until convergence:
   a. Assign each data point to the nearest centroid
   b. Update centroids as the mean of assigned points
3. Return final cluster labels

The pseudocode above summarizes the iterative procedure of the standard K-Means algorithm, highlighting the two main phases: cluster assignment and centroid update. This process continues until convergence is achieved, ensuring that the final clusters minimize intra-cluster variance while maximizing separation between clusters.

Despite its efficiency, the performance of K-Means is strongly influenced by the initial centroid selection. Random initialization may result in poor clustering solutions, convergence to local minima, or instability across multiple runs [10], [16], [17]. This issue is especially pronounced in high-dimensional or large datasets, where suboptimal initialization can lead to longer execution times and reduced clustering quality [13], [53]. Various studies have confirmed that K-Means may require several iterations before stabilizing, and the final solution can vary significantly depending on the initial seeds [22], [25].

To mitigate these limitations, several improvements to the standard K-Means have been proposed, including K-Means++ [12], [23] and other advanced centroid initialization methods such as Adaptive initialization method for K-Means algorithm (AIMK) [20] and K-Means NANI: An improved clustering algorithm for optimal seed selection [21], which introduce probabilistic or geometry-aware initialization schemes to select centroids more strategically. However, even with such enhancements, K-Means remains sensitive to initialization and is less effective when clusters are non-spherical or imbalanced in size and density [1], [5], [11].

For these reasons, K-Means is often used as a baseline method in clustering research, against which new methods or hybrid approaches are evaluated [11], [27], [38]. In this study, standard K-Means serves as the benchmark for comparison with hybrid hierarchical–partitioning models, where the hierarchical stage is employed to improve centroid initialization and reduce the weaknesses inherent in the traditional algorithm [37], [42]. The Agglomerative Hierarchical + K-Means (AHC–KMeans) method is a hybrid clustering approach that combines the strengths of agglomerative hierarchical clustering and K-Means. It is designed to address one of the primary limitations of K-Means, namely the random initialization of centroids, which often leads to unstable clustering results and convergence to local minimum [14], [34], [38].

In the agglomerative hierarchical stage, the clustering process begins by treating each data point as an individual cluster. The algorithm then iteratively merges clusters based on a predefined linkage criterion until the desired number of clusters is achieved, forming a hierarchical structure known as a dendrogram [7], [9]. In this study, the agglomerative clustering is implemented using the Ward linkage method with Euclidean distance, as provided by the *Agglomerative Clustering* algorithm in scikit-learn. Ward's linkage minimizes the total within-cluster variance during each merging step, resulting in compact and relatively spherical clusters.

The selection of Ward's method is particularly important in this hybrid framework. Compared to other linkage strategies, such as single linkage (which is prone to chaining effects and noise sensitivity) and complete linkage (which may produce overly tight clusters), Ward's linkage provides a balanced clustering structure with improved cohesion and separation. Moreover, its objective function is consistent with the variance-minimization principle used in K-Means, making it highly compatible for centroid initialization [29], [48].

Once the hierarchical clustering process produces k clusters, the centroid of each cluster is computed as the mean of its member data points. These centroids serve as deterministic initial seeds for the K-Means algorithm, replacing the conventional random initialization. This hierarchical-based initialization significantly reduces randomness and improves the stability of the clustering process, minimizing the risk of convergence to poor local minima and enhancing clustering consistency [15], [18], [35].

With these improved initial centroids, the K-Means algorithm is subsequently applied to refine cluster assignments. Since the centroids are already positioned near representative regions of the data space, the algorithm starts from a

more informed initialization. This leads to faster convergence, as fewer iterations are required to reach stability. Additionally, the reduced iteration count contributes to lower computational time while improving the robustness and reproducibility of the final clustering results [10], [11], [28].

The hybrid hierarchical–partitioning approach offers several advantages over traditional clustering techniques. First, it produces more stable and reproducible centroids by eliminating randomness in the initialization phase [38], [41]. Second, it enables faster convergence during the K-Means stage due to better initial centroid placement [7], [27]. Third, it generally improves clustering quality, as indicated by higher Silhouette scores and lower Davies–Bouldin Index (DBI) values, reflecting stronger intra-cluster cohesion and clearer inter-cluster separation [11], [46].

Despite these advantages, the AHC–KMeans method also has certain limitations. The hierarchical stage introduces additional computational overhead, as constructing the dendrogram requires significant time and memory resources [2], [6]. This limitation becomes more pronounced for large datasets due to the inherent complexity of agglomerative clustering [3], [30]. However, in structured datasets such as socio-economic data, the improved initialization provided by Ward-based hierarchical clustering often outweighs the additional cost, resulting in more accurate and stable clustering outcomes, particularly for high-dimensional or non-uniform data distributions [32], [36], [39].

Algorithm 2. Hybrid Agglomerative–KMeans Clustering

Input: Dataset X, number of clusters k
Output: Cluster labels

1. Apply Agglomerative Hierarchical Clustering to dataset X
2. Merge data points iteratively until k clusters are formed
3. Compute initial centroids as the mean of data points in each cluster
4. Initialize K-Means using the computed centroids
5. Repeat until convergence:
   a. Assign each data point to the nearest centroid
   b. Update centroids as the mean of assigned points
6. Return final cluster labels

The pseudocode above describes a hybrid clustering approach in which agglomerative hierarchical clustering is first used to generate structured initial clusters. These clusters are then used to compute deterministic initial centroids for K-Means, reducing the randomness of initialization and improving convergence stability and clustering quality.

The Divisive Hierarchical + K-Means (DHC–KMeans) method is a hybrid clustering approach that combines divisive hierarchical clustering with K-Means. Unlike the agglomerative method, which starts from individual points and merges them step by step, the divisive approach follows a top-down strategy by initially treating the entire dataset as a single cluster and recursively partitioning it into smaller sub-clusters until the desired number of clusters (k) is reached [7], [9], [34].

This approach differs from conventional divisive hierarchical clustering, where splits are often determined using global dissimilarity measures or dimensionality reduction techniques such as PCA [29], [54]. While those methods can capture global data structures more explicitly, they typically introduce higher computational complexity. In contrast, the K-Means-based splitting strategy adopted in this study offers a practical balance between clustering accuracy and computational efficiency, particularly for high-dimensional datasets.

In this study, the divisive hierarchical stage is implemented using a recursive binary splitting strategy based on K-Means partitioning, rather than relying on traditional dissimilarity-based or projection-based methods. Specifically, at each iteration, the largest cluster (in terms of the number of data points) is selected and divided into two sub-clusters using the K-Means algorithm with k = 2 and Euclidean distance. This splitting process is repeated iteratively until the predefined number of clusters (k) is achieved. The use of K-Means as the splitting mechanism ensures that each partition is determined by minimizing intra-cluster variance, providing a computationally efficient and scalable alternative to more complex strategies such as exhaustive dissimilarity evaluation or Principal Component Analysis (PCA)-based splitting [10], [22], [55].

After the recursive splitting process produces k clusters, the centroid of each cluster is computed as the mean of its member data points. These centroids represent the central positions of their respective clusters and are used as

deterministic initial seeds for the subsequent K-Means refinement stage [17], [19], [35]. By deriving centroids from the hierarchical division, the initialization becomes more structured and less dependent on random selection, thereby improving clustering stability and reproducibility.

In the final stage, the K-Means algorithm is applied using the centroids obtained from the divisive hierarchical stage as initial seeds. Since these centroids already capture the underlying structure of the dataset, the algorithm converges more rapidly and requires fewer iterations compared to standard K-Means with random initialization [10], [22], [55]. This refinement process enhances the precision of cluster boundaries and improves overall clustering accuracy and consistency across multiple runs.

The integration of divisive hierarchical clustering with K-Means offers several advantages. First, the top-down splitting strategy enables early identification of globally significant cluster structures, resulting in more representative initial centroids [7], [38]. Second, the use of K-Means-based splitting ensures computational efficiency while maintaining consistency with the variance-minimization objective of the final clustering stage. Third, the hybrid approach reduces convergence time and improves clustering stability by providing well-informed initial conditions [11], [37], [39].

Despite these advantages, the DHC–KMeans method also has certain limitations. The recursive splitting process introduces additional computational overhead, particularly as the number of clusters increases [2], [6], [9]. Moreover, although the splitting process is efficient, it inherits some limitations of K-Means, such as sensitivity to data distribution and a tendency to favor spherical cluster shapes [13], [30]. Nevertheless, this hybrid framework is specifically designed to address the primary weakness of standard K-Means—its sensitivity to random centroid initialization—while maintaining a balance between computational efficiency and clustering performance [36], [38], [44].

Algorithm 3. Hybrid Divisive–KMeans Clustering

Input: Dataset X, number of clusters k
Output: Cluster labels

1. Initialize dataset X as a single cluster
2. While the number of clusters is less than k:
   a. Select the largest cluster
   b. Split the selected cluster into two sub-clusters using K-Means (k=2)
3. Compute centroids of the resulting kkk clusters
4. Initialize K-Means using the computed centroids
5. Repeat until convergence:
   a. Assign each data point to the nearest centroid
   b. Update centroids as the mean of assigned points
6. Return final cluster labels

The pseudocode above outlines a top-down hybrid clustering strategy, where divisive hierarchical clustering is performed through recursive partitioning. The resulting clusters provide structured initial centroids for the K-Means refinement stage, enabling faster convergence and improved clustering consistency compared to random initialization.

3.3. Evaluation Metrics

To comprehensively assess clustering performance, this study applies three categories of evaluation metrics: execution time, convergence iterations, and cluster validity indices**.** These metrics capture not only computational efficiency but also the stability and quality of clustering results, which are essential for evaluating clustering algorithms in big data environments [5], [10], [11], [25].

Execution time refers to the total amount of time taken by each clustering algorithm to complete the clustering process, measured in seconds. In this study, execution time was recorded using Python's built-in time function, which captures the duration from the initialization of the algorithm to its convergence. This metric directly evaluates computational efficiency, which is particularly critical in the context of big data analysis, where clustering methods must be both accurate and scalable [5], [7], [27]. Hybrid approaches, such as Agglomerative K-Means and Divisive K-Means, are expected to reduce overall computation time by improving centroid initialization, which leads to faster convergence despite the additional hierarchical overhead [25], [37], [42]. Previous studies have demonstrated that such hybrid

hierarchical–partitioning methods can achieve a balance between accuracy and efficiency, outperforming traditional K-Means in large-scale datasets [36], [39].

Convergence iterations represent the number of refinement steps K-Means requires to stabilize after centroid initialization. A lower number of iterations indicates that centroids were initialized closer to optimal positions, leading to faster convergence and reduced computational load [15], [17], [19]. In standard K-Means, poor centroid initialization can lead to multiple redundant iterations, increasing both execution time and the risk of suboptimal clustering [10], [11], [14]. In contrast, hybrid methods such as hierarchical-based or metaheuristic-assisted centroid initialization improve convergence by providing better starting centroids, thereby accelerating the stabilization process [18], [22], [23], [25]. This metric is essential for comparing the efficiency and stability between conventional and hybrid clustering algorithms, as it highlights the role of initialization in the optimization of clustering performance [17], [24], [35].

To evaluate the quality of the resulting clusters, three internal validation indices are employed: Silhouette Coefficient**,** Davies–Bouldin Index (DBI)**,** and Calinski–Harabasz Index (CH Index)**.** These indices measure cluster cohesion, separation, and variance structure to provide a comprehensive evaluation of clustering quality [28], [46], [48].

The Silhouette Coefficient is one of the most widely used internal validation indices for clustering evaluation. It provides a quantitative measure of how well each data point fits within its assigned cluster compared to other clusters. The index combines two key aspects of clustering quality: cohesion (the degree of similarity between a data point and other points in the same cluster) and separation (the degree of dissimilarity between a data point and points in the nearest neighboring cluster). For each data point *i*, the Silhouette value *s(i)* is defined as:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{1}$$

The Silhouette Coefficient is a widely used metric for evaluating the quality and validity of clustering results. It measures how similar an object is to its own cluster compared to other clusters, thereby assessing both cluster cohesion and separation. In this metric, *a(i)* represents the average distance between a data point *i* and all other points within the same cluster, which reflects the degree of cohesion. Meanwhile, *b(i)* denotes the minimum average distance between the point *i* and all points belonging to other clusters, representing the separation between clusters. The Silhouette Coefficient value ranges from –1 to +1. A value close to +1 indicates that the data point is well-matched to its own cluster and poorly matched to neighboring clusters, implying a well-defined cluster structure. A value near 0 suggests that the point lies on the boundary between clusters, while a value approaching –1 indicates that the point may have been incorrectly assigned to a cluster [48].

High Silhouette scores suggest compact and well-separated clusters, while low scores indicate overlap or weak structure [46], [48]. This metric is frequently used in comparative studies of clustering algorithms to evaluate the effectiveness of initialization and the optimal number of clusters [11], [25], [28]. The Davies–Bouldin Index (DBI) is an internal cluster validity metric that evaluates the average similarity between clusters by considering both the compactness within clusters and the separation between clusters. It was first introduced by Davies and Bouldin (1979) and has since been widely used for assessing clustering quality in unsupervised learning.

For each cluster i, the DBI is calculated as the average of the maximum similarity values between cluster i and all other clusters j. The similarity measure is defined as the ratio between the within-cluster scatter (how compact the cluster is) and the distance between cluster centroids (how far apart two clusters are). Mathematically, the DBI is expressed as:

$$DBI = \frac{1}{k}\sum_{i=1}^{k} \max_{j \neq i}\left(\frac{S_i + S_j}{M_{ij}}\right) \tag{2}$$

In the context of the Davies–Bouldin Index (DBI), several key parameters are used to quantify the compactness and separation of clusters. Here, k denotes the total number of clusters formed during the clustering process. The term S_i represents the average distance between all data points within cluster i and its corresponding centroid, which measures the intra-cluster compactness or cohesion. Meanwhile, M_{ij} refers to the distance between the centroids of clusters i and j, capturing the inter-cluster separation or distinctiveness between clusters. Together, these components form the basis for calculating the DBI, where lower index values indicate better clustering performance characterized by tighter clusters and greater separation between them. Lower DBI values indicate better clustering compact clusters and high separation [28], [48]. The DBI is particularly effective for imbalanced or variable-density datasets, making it a suitable index for evaluating hybrid clustering algorithms in big data analysis [5], [25], [46].

 ISSN 2723-6471

The Calinski–Harabasz Index (CH Index), also referred to as the Variance Ratio Criterion (VRC), is an internal clustering validation metric that evaluates the quality of a clustering structure based on the ratio of between-cluster dispersion to within-cluster dispersion. It was first proposed by Caliński and Harabasz (1974) and has since been widely adopted as a reliable measure for determining the optimal number of clusters in unsupervised learning. Mathematically, the CH Index is defined as:

$$CH(k) = \frac{Tr(B_k)}{Tr(W_k)} x \frac{N-k}{k-1} \quad (3)$$

In the formulation of the Calinski–Harabasz (CH) Index, several key parameters are used to evaluate the balance between cluster compactness and separation. Here, N denotes the total number of data points in the dataset, while k represents the number of clusters generated by the clustering algorithm. The term $Tr(B_k)$ refers to the trace of the between-cluster dispersion matrix, which quantifies the variance of cluster centroids relative to the overall mean of the dataset reflecting the degree of cluster separation. Meanwhile, $Tr(W_k)$ denotes the trace of the within-cluster dispersion matrix, representing the variance of data points within each cluster and thus measuring cluster compactness. The ratio between these two measures serves as the foundation of the CH Index, where a higher value indicates better-defined clusters characterized by strong separation and internal cohesion.

A higher CH Index value indicates better clustering quality, as it reflects clusters that are well-separated from each other (high between-cluster variance) and internally compact (low within-cluster variance). Unlike the Davies–Bouldin Index (DBI), where lower values are preferred, the CH Index favors higher values as a sign of optimal partitioning [25], [48]. Higher CH values indicate better clustering, signifying high inter-cluster variance and low intra-cluster variance [28], [46]. The CH Index is widely applied for model selection to determine the optimal number of clusters, complementing the Silhouette and DBI metrics [11], [28], [46]. Prior studies have confirmed its effectiveness in hybrid hierarchical–partitioning clustering**,** especially for medium to large-scale datasets, due to its sensitivity to both compactness and separation [25], [36], [41].

When combined with the Silhouette Coefficient and the Davies–Bouldin Index (DBI), the Calinski–Harabasz (CH) Index offers a comprehensive perspective for evaluating clustering performance from multiple dimensions. In addition to these validity indices, several complementary metrics are also considered to provide a more holistic assessment. Execution Time measures the computational efficiency of the algorithm, indicating how effectively the method handles data processing within a given runtime [7], [25], [27], [39]. Convergence Iterations reflect the effectiveness of centroid initialization and the overall algorithmic stability, where fewer iterations typically imply faster and more reliable convergence [15], [17], [18], [22]. Finally, the Cluster Validity Indices including Silhouette, DBI, and CH jointly assess the clustering accuracy and structural quality of the resulting partitions, ensuring that clusters are both cohesive and well-separated [11], [28], [46], [48]. This integrated evaluation framework ensures a balanced and objective comparison between standard K-Means and hybrid approaches, revealing the trade-offs between efficiency, stability, and cluster quality [5], [10], [11], [25], [28].

4. Results and Discussion

4.1. Execution Time Comparison

Table 2 presents the 35 data points of the clustering results show the assigned cluster labels for each method: K-Means predominantly assigns most points to cluster 2, with some points in clusters 0 and 1; Agglomerative K-Means shows more variation across clusters 0, 1, and 2; while Divisive K-Means produces a pattern very similar to Agglomerative K-Means, indicating comparable cluster assignments between the two hybrid approaches.

Table 2. 35 data points of clustering results

Method		Regency Cluster Assignment																																	
Kmeans	:	2	2	2	0	2	2	0	0	2	2	1	2	2	0	2	0	2	2	1	1	2	2	2	2	2	2	0	2	0	1	1	1	1	1	1
Agglomerative-Kmeans	:	2	2	1	1	1	2	1	1	1	2	2	1	2	1	1	1	2	2	0	2	2	2	1	2	1	2	1	2	1	0	0	0	0	0	0
Divisive-Kmeans	:	2	2	1	1	1	2	1	1	1	2	2	1	2	1	1	1	2	2	0	2	2	2	1	2	1	2	1	2	1	0	0	0	0	0	0

Table 3 presents the execution time of the three clustering methods: standard K-Means, Agglomerative K-Means, and Divisive K-Means. The results are reported in seconds and represent the average of multiple experimental runs to reduce the effect of random variations. All experiments were conducted on a personal computer equipped with an Intel

ISSN 2723-6471

Core i5-1035G1 CPU and 8 GB of RAM. Figure 2 presents the clustering results in the form of a geographic cluster map, illustrating the spatial distribution of clusters across districts and municipalities in Central Java Province for three different methods: standard K-Means, Hybrid Agglomerative–KMeans, and Hybrid Divisive–KMeans.

Each subfigure in figure 2 represents a different clustering approach, where regions are color-coded into three clusters (Cluster 0, Cluster 1, and Cluster 2). The map visualization allows for intuitive interpretation of how each method groups regions based on socio-economic similarity. In the K-Means result, cluster assignments appear more scattered and less spatially coherent, indicating sensitivity to random centroid initialization. Some neighboring regions are assigned to different clusters, suggesting weaker structural consistency.

In contrast, the Hybrid Agglomerative–KMeans method produces more spatially consistent clusters, where geographically adjacent regions tend to belong to the same cluster. This indicates that hierarchical initialization helps capture the underlying structure of the data more effectively, leading to improved cluster stability and cohesion.

Similarly, the Hybrid Divisive–KMeans method shows a more balanced distribution of clusters across the region. The clusters appear more structured compared to standard K-Means, although slight fragmentation is still observed in certain areas. This reflects the influence of the recursive splitting mechanism, which captures global data structure but may still be affected by local variations.

Overall, the cluster map visualization in figure 2 demonstrates that hybrid hierarchical–partitioning approaches produce more interpretable and spatially meaningful clustering patterns compared to standard K-Means. This visual evidence supports the quantitative evaluation results, particularly in terms of improved clustering stability and efficiency.

Table 3. Execution time comparison of clustering methods

Method	Execution Time (ms)	Iterations to Convergence	Silhouette Score	DBI	CH Index
Standard K-Means	116.68	3	0.1903	1.4189	15.4015
Agglomerative K-Means (AHC)	2.04	2	0.2189	1.3275	15.5569
Divisive K-Means (DHC)	1.91	1	0.1436	1.6802	11.9842

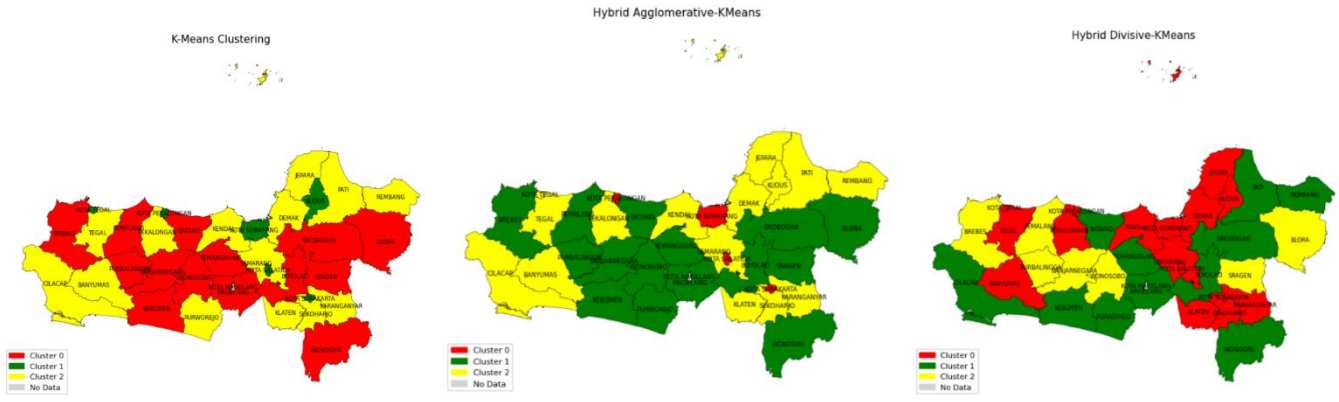

Figure 2. the result of the hybrid clustering

From the table 3, it can be observed that both hybrid approaches (AHC–KMeans and DHC–KMeans) required fewer iterations to converge compared to the standard K-Means. The reduced number of iterations translated into lower execution times, despite the additional overhead introduced by the hierarchical initialization phase. Moreover, the hybrid methods achieved slightly better Silhouette Scores and lower Davies–Bouldin Index values, indicating improved cluster quality. In addition, the Calinski–Harabasz (CH) Index values for the hybrid methods were notably higher than those of the standard K-Means. Since a higher CH Index reflects clusters that are both compact and well-separated, this further confirms that the hybrid approaches produced more reliable and well-defined clustering structures. Taken together, these results demonstrate that the integration of hierarchical initialization with K-Means not only improves computational efficiency but also enhances the overall quality and stability of the clustering results.

4.2. Cluster Characteristics Analysis

The Cluster Characteristics Analysis aims to provide a detailed interpretation of the clustering results by examining the unique socio-economic features represented in each cluster. This analysis helps to understand how regions with similar attributes are grouped together and reveals the underlying patterns that distinguish one cluster from another. The results of the Cluster Characteristics Analysis are presented in figure 2.

The clustering results obtained using the K-Means method reveal distinct socio-economic characteristics across the three identified clusters. Each cluster represents regions with similar poverty, education, employment, and living condition profiles, enabling a clearer interpretation of regional socio-economic patterns.

Cluster 0 – Moderate-Poverty Rural Regions. Cluster 0 is characterized by a relatively high average poverty rate of 12.44%, accompanied by a high proportion of individuals who did not complete primary education (17.54%) and a relatively low literacy rate (92.67%). The school participation rate is also lower (94.96%) compared to other clusters. In terms of employment, this cluster shows strong dependence on the agricultural sector (29.56%) and a high proportion of informal workers (46.16%). The average per capita food expenditure is moderate (62.05), while toilet usage remains relatively high (94.64%). These quantitative indicators confirm that Cluster 0 represents rural regions with moderate-to-high poverty levels and limited educational attainment, such as Banjarnegara, Wonosobo, and Grobogan.

Cluster 1 – Urbanized and High-Literacy Areas. Cluster 1 exhibits the most favorable socio-economic conditions, with the lowest poverty rate (6.35%) and the lowest proportion of individuals without primary education (6.86%). This cluster also has the highest literacy rate (98.12%) and school participation rate (99.12%), indicating strong educational outcomes. Agricultural employment is very low (5.24%), while informal employment is also relatively low (24.19%). Despite having slightly lower per capita food expenditure (58.46), toilet usage remains high (89.59%). These metrics indicate that Cluster 1 represents urban and economically advanced regions, such as Semarang City, Salatiga City, and Magelang City.

Cluster 2 – Transitional Semi-Urban Areas. Cluster 2 represents an intermediate group, with a moderate poverty rate (9.70%) and a moderate proportion of individuals without primary education (13.84%). The literacy rate (95.02%) and school participation rate (98.33%) are higher than Cluster 0 but lower than Cluster 1. Employment characteristics show a balanced structure, with moderate agricultural employment (18.23%) and informal sector participation (35.50%). Per capita expenditure (61.99) and toilet usage (95.20%) indicate relatively stable living conditions. Thus, this cluster can be interpreted as semi-urban regions undergoing economic transition, such as Banyumas, Klaten, and Kendal. A detailed comparison of socio-economic indicators for each cluster is provided in table 4.

Table 4. Comparison of Socio-Economic Indicators K-Means Clusters

Indicator	Cluster 0	Cluster 1	Cluster 2
Poverty Rate	12.44	6.35	9.70
No Primary Education	17.54	6.86	13.84
Literacy Rate	92.67	98.12	95.02
School Participation	94.96	99.12	98.33
Agricultural Employment	29.56	5.24	18.23
Informal Employment	46.16	24.19	35.50
Food Expenditure	62.05	58.46	61.99
Toilet Usage	94.64	89.59	95.20

The clustering results obtained using the Hybrid Agglomerative-KMeans method reveal distinct socio-economic characteristics across the three identified clusters. Each cluster represents regions with similar poverty, education, employment, and living condition profiles, enabling a clearer interpretation of regional socio-economic patterns.

Cluster 0 – Urban Core Group. Cluster 0 demonstrates the lowest poverty rate (5.91%) and the lowest percentage of individuals without primary education (5.18%). It also has the highest literacy rate (98.59%) and school participation rate (99.06%). Agricultural employment is minimal (5.05%), and informal sector participation is relatively low (23.95%). However, this cluster shows slightly lower toilet usage (86.78%) compared to others, despite having a lower

per capita expenditure (57.83). Overall, these indicators confirm that Cluster 0 represents urban core regions with strong educational attainment and diversified economies, such as Salatiga City and Semarang City.

Cluster 1 – Rural-Agrarian Cluster. Cluster 1 has the highest poverty rate (12.34%) and the highest proportion of individuals without primary education (17.18%), along with the lowest literacy rate (92.88%). It also shows strong reliance on agriculture (29.43%) and the highest informal employment rate (46.20%). Interestingly, this cluster records the highest per capita food expenditure (62.04), but this does not offset its structural disadvantages. These quantitative findings indicate that Cluster 1 represents rural and agriculturally dependent regions with lower socio-economic conditions, such as Banjarnegara, Kebumen, and Wonosobo.

Cluster 2 – Emerging Semi-Urban Cluster. Cluster 2 exhibits moderate poverty levels (9.32%) and a moderate percentage of individuals without primary education (13.58%). The literacy rate (95.22%) and school participation rate (98.40%) suggest relatively good educational access. Employment indicators show moderate agricultural (15.93%) and informal sector participation (33.32%), reflecting a diversified economic structure. Per capita expenditure (61.74) and toilet usage (95.13%) are also relatively high. Therefore, this cluster can be interpreted as emerging semi-urban regions transitioning toward more developed economic structures, such as Banyumas, Demak, and Rembang.

A detailed comparison of socio-economic indicators for each cluster is provided in table 5.

Table 5. Comparison of Socio-Economic Indicators Hybrid Agglomerative-KMeans Clusters

Indicator	Cluster 0	Cluster 1	Cluster 2
Poverty Rate	5.91	12.34	9.32
No Primary Education	5.18	17.18	13.58
Literacy Rate	98.59	92.88	95.22
School Participation	99.06	95.05	98.40
Agricultural Employment	5.05	29.43	15.93
Informal Employment	23.95	46.20	33.32
Food Expenditure	57.83	62.04	61.74
Toilet Usage	86.78	94.94	95.13

The clustering results obtained using the Hybrid Divisive-KMeans method reveal distinct socio-economic characteristics across the three identified clusters. Each cluster represents regions with similar poverty, education, employment, and living condition profiles, enabling a clearer interpretation of regional socio-economic patterns.

Cluster 0 – Developed Urban Centers. Cluster 0 shows relatively low poverty (7.97%) and lower educational deprivation (10.75%), with a high literacy rate (96.29%) and school participation (98.78%). Agricultural employment is relatively low (10.66%), and informal sector participation is also moderate (29.41%). Per capita expenditure (60.32) and toilet usage (92.80%) indicate relatively good living standards. These metrics suggest that Cluster 0 represents developed urban or semi-urban regions with relatively strong socio-economic conditions.

Cluster 1 – Middle-Class Transition Regions. Cluster 1 has a moderate poverty rate (10.95%) and moderate educational attainment, with literacy at 94.33%. Agricultural employment remains relatively high (29.38%), and informal sector participation is also high (45.04%). Per capita expenditure (62.44) and toilet usage (95.38%) are relatively high, indicating improving living conditions. This cluster reflects regions undergoing economic transition with mixed rural–urban characteristics.

Cluster 2 – High-Poverty Agrarian Regions. Cluster 2 exhibits the highest poverty rate (14.07%) and the highest proportion of individuals without primary education (19.48%), along with the lowest literacy rate (91.40%). It also shows high dependence on agriculture (28.76%) and the informal sector (45.32%). Although per capita expenditure (61.93) is relatively similar to other clusters, the overall socio-economic indicators suggest structural disadvantages. Thus, Cluster 2 represents high-poverty rural regions with limited educational attainment and strong dependence on traditional economic sectors, such as Banjarnegara, Purbalingga, and Brebes. A detailed comparison of socio-economic indicators for each cluster is provided in table 6.

Table 6. Comparison of Socio-Economic Indicators Hybrid Divisive-KMeans Clusters

Indicator	Cluster 0	Cluster 1	Cluster 2
Poverty Rate	7.97	10.95	14.07
No Primary Education	10.75	15.30	19.48
Literacy Rate	96.29	94.33	91.40
School Participation	98.78	96.72	93.82
Agricultural Employment	10.66	29.38	28.76
Informal Employment	29.41	45.04	45.32
Food Expenditure	60.32	62.44	61.93
Toilet Usage	92.80	95.38	94.01

4.3. Discussion

The experimental results demonstrate that employing hierarchical clustering techniques to initialize centroids prior to executing K-Means substantially improves clustering performance. By selecting more representative initial centroids, the hybrid approach reduced the number of convergence iterations and lowered overall execution time. While the hierarchical phase introduces additional computational overhead due to dendrogram construction, this cost is offset by the reduced refinement steps required in K-Means. These results indicate that hierarchical initialization enhances clustering efficiency, stability, and convergence speed.

The observed improvements are consistent with prior studies emphasizing the importance of centroid initialization. Osei-Bryson [38] showed that hybrid hierarchical–K-Means models produce more consistent results than conventional K-Means [14]. Cabrera [15] and Khan [17] confirmed that advanced initialization strategies reduce iterations and accelerate convergence. Akhter et al. [37] demonstrated that an O(N log N) hybrid merging approach balances initialization cost and clustering efficiency. Chen et al. [27] reported that hierarchical–partitioning combinations are particularly advantageous for large-scale datasets, as the reduced iteration count compensates for dendrogram computation. Similarly, Ikotun et al. [11] and Das and Mitra [46] found that hybrid initialization improves cluster compactness and separation, reflected in higher Silhouette and lower Davies–Bouldin scores. While Bai et al. [1] and Solano and Berlanga [36] highlighted hybrid approaches' ability to handle non-spherical data structures, Zhang et al. [30] and Salehi et al. [40] cautioned about memory and scalability challenges in extremely large datasets.

Overall, the current experiments confirm that hierarchical initialization enhances K-Means performance by stabilizing clustering outcomes and improving computational efficiency. The findings reinforce the conclusions of fast hybrid methods [37], [39] and advanced centroid optimization strategies [17], [19], demonstrating that intelligent centroid selection remains a critical factor in hybrid clustering frameworks.

The results highlight a trade-off between clustering quality and execution time. Standard K-Means is computationally efficient for small datasets but may converge slowly or produce unstable clusters for larger, complex datasets. Hybrid approaches mitigate these issues by combining the global structure capture of hierarchical clustering with K-Means' refinement efficiency, improving both cluster validity and execution speed.

However, the dendrogram construction in hierarchical clustering can become a bottleneck for extremely large datasets. Future research could explore optimization strategies, such as parallelized hierarchical methods, approximate dendrogram construction, or hybrid centroid selection algorithms, to enhance scalability without sacrificing cluster quality. Additionally, further investigation into the applicability of hybrid methods for high-dimensional or non-spherical data could extend their practical utility.

This study is subject to several limitations related to the dataset, scalability, and algorithmic assumptions. First, the dataset used in this research consists of socio-economic data from Central Java, which can be categorized as a medium-sized dataset and may not fully represent more complex, large-scale, or high-dimensional data environments. Additionally, the dataset is primarily numerical and relatively well-structured, which may favor clustering performance compared to more heterogeneous or unstructured data. From a scalability perspective, although the hybrid hierarchical–K-Means approach improves clustering quality and stability, the hierarchical stage introduces additional computational overhead, particularly in terms of time and memory complexity, which may limit its applicability to very large datasets.

Furthermore, the proposed method relies on several assumptions, including the use of Euclidean distance as the similarity metric, the expectation that clusters are relatively compact and spherical in shape (as required by K-Means), and a predefined number of clusters (k). These assumptions may not always hold in real-world scenarios with irregular cluster shapes or unknown cluster structures. Therefore, caution should be exercised when applying this method to large-scale, high-dimensional, or highly complex datasets, and further optimization or adaptation may be necessary to ensure its practical effectiveness.

5. Conclusion

This study compared the performance of standard K-Means with two hybrid approaches, namely Agglomerative K-Means and Divisive K-Means. The experimeintelntal results demonstrate that both hybrid methods outperform standard K-Means in terms of centroid stability and convergence behavior. By utilizing hierarchical clustering to determine initial centroids, the hybrids reduce sensitivity to random initialization and achieve more consistent clustering outcomes. In terms of execution time, the hybrid approaches also provide advantages, particularly for medium and large-scale datasets. Although hierarchical initialization introduces an initial overhead, this cost is compensated by faster convergence during the K-Means refinement phase, resulting in improved overall computational efficiency. Overall, the results confirm that integrating hierarchical initialization with K-Means enhances clustering performance and stability. Future work could focus on exploring variations of hierarchical initialization techniques or optimization strategies within the hybrid framework to further improve efficiency and scalability, while remaining consistent with the methods and analyses presented in this study.

6. Declarations

6.1. Author Contributions

Conceptualization: B.W., B.W., and B.S.; Methodology: B.W. and B.W.; Software: B.W.; Validation: B.W., B.W., and B.S.; Formal Analysis: B.W., B.W., and B.S.; Investigation: B.W.; Resources: B.W.; Data Curation: B.W.; Writing Original Draft Preparation: B.W.; Writing Review and Editing: B.W., B.W., and B.S.; Visualization: B.W.; Supervision: B.W. and B.S.. All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

"The dataset used in this study is publicly available on Kaggle at: https://www.kaggle.com/datasets/ziya07/student-engagement-using-biosensor-technology".

6.3. Funding

This research was funded by the 2026 RKAT of Universitas Sebelas Maret through the Doctoral Dissertation Research Scheme (PDD-UNS) under Research Assignment Agreement No. 362/UN27.22/PT.01.03/2026. The authors would also like to express their sincere gratitude to University Diponegoro, Semarang, for its support and collaboration.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

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.

References

[1] L. Bai, J. Liang, and F. Cao, "A multiple k-means clustering ensemble algorithm to find nonlinearly separable clusters," *Inf. Fusion*, vol. 61, no. Jan., pp. 36–47, Jan. 2020, doi: 10.1016/j.inffus.2020.03.006.

[2] A. Belhadi et al., "Space–time series clustering: Algorithms, taxonomy, and case study on urban smart cities," *Eng. Appl. Artif. Intell.*, vol. 95, no. Mar., pp. 1–14, Mar. 2020, doi: 10.1016/j.engappai.2020.103857.

[3] R. J. G. B. Campello, D. Moulavi, and J. S. Sander, "Hierarchical density estimates for data clustering, visualization, and outlier detection," *ACM Trans. Knowl. Discov. Data*, vol. 10, no. 1, pp. 1–51, Jul. 2015, doi: 10.1145/2733381.

[4] A. A. Wani, "Comprehensive analysis of clustering algorithms: Exploring challenges and future directions in data mining," *PeerJ Comput. Sci.*, vol. 10, no. 2024, pp. 1–20, 2024, doi: 10.7717/peerj-cs.2286.

[5] J. Singh and D. Singh, "A comprehensive review of clustering techniques in artificial intelligence for knowledge discovery: Taxonomy, challenges, applications and future prospects," *Adv. Eng. Informatics*, vol. 62, no. 2024, pp. 1–15, 2024, doi: 10.1016/j.aei.2024.102799.

[6] S. Chakraborty, M. Das, and S. Bandyopadhyay, "Hierarchical clustering with optimal transport," *Stat. Probab. Lett*., vol. 163, no. Apr., pp. 1–8, Apr. 2020, doi: 10.1016/j.spl.2020.108781.

[7] A. E. Ezugwu et al., "A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications," *Eng. Appl. Artif. Intel*l., vol. 110, no. Mar., pp. 1–24, Mar. 2022, doi: 10.1016/j.engappai.2022.104743.

[8] C. X. Gao et al., "An overview of clustering methods with guidelines for application in mental health research," *Psychiatry Res*., vol. 331, no. 2023, pp. 1–12, 2023, doi: 10.1016/j.psychres.2023.115265.

[9] S. Zhou et al., "A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions," *arXiv,* vol. 2022, no. Jun., pp. 1–35, 2022, doi: 10.48550/arXiv.2206.07579.

[10] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means algorithm: A comprehensive survey and performance evaluation," *Electronics,* vol. 9, no. 8, pp. 1–35, Aug. 2020, doi: 10.3390/electronics9081295.

[11] A. M. Ikotun et al., "K means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data," *Inf. Sci.*, vol. 622, no. Sep., pp. 178–210, Sep. 2023, doi: 10.1016/j.ins.2022.11.139.

[12] A. Kapoor and A. Singhal, "A comparative study of K Means, K Means++ and fuzzy C Means clustering algorithms," *Comput. Intell. Commun. Technol.*, vol. 2017, no. Feb., pp. 1–6, Feb. 2017, doi: 10.1109/CIACT.2017.7977272.

[13] H. Hu, J. Liu, and Y. Song, "An effective and adaptable K Means algorithm for big data clustering," *Appl. Soft Comput*., vol. 132, no. Aug., pp. 1–12, Aug. 2023, doi: 10.1016/j.patcog.2023.109404.

[14] J. MacQueen, "Some methods for classification and analysis of multivariate observations," *Math. Stat. Probab*., vol. 1967, no. 1967, pp. 281–297, 1967.

[15] A. Fahim, "Finding the number of clusters in data and better initial centers for K-means algorithm," *Int. J. Intell. Syst. Appl.*, vol. 12, no. 6, pp. 1–20, Dec. 2020, doi: 10.5815/ijisa.2020.06.01.

[16] R. Waboke et al., "Centroid initialization in k means clustering using GATCAM," *Sci. World J*., vol. 18, no. 1, pp. 1–10, 2023.

[17] A. A. Khan et al., "K Means centroids initialization based on differentiation between instances attributes," *Int. J. Intell. Syst*., vol. 2024, no. 2024, pp. 1–15, 2024, doi: 10.1155/2024/7086878.

[18] J. Preeti and K. Deep, "Automatic centroid initialization in k means using artificial hummingbird algorithm," *Neural Comput. Appl.*, vol. 37, no. 5, pp. 3373–3398, 2024, doi: 10.1007/s00521-024-10764-4.

[19] K. Yang, M. M. Amiri, and S. R. Kulkarni, "Greedy centroid initialization for federated K means," *Inf. Sci. Syst*., vol. 2023, no. Mar., pp. 1–6, Mar. 2023, doi: 10.1109/CISS56502.2023.10089666.

[20] J. Yang, X. Fang, and Y. Zhang, "Adaptive initialization method for K Means algorithm (AIMK)," *Front. Artif. Intell*., vol. 4, no. Aug., pp. 1–12, Aug. 2021, doi: 10.3389/frai.2021.740817.

[21] M. Liu et al., "Enhanced PSO based clustering algorithm with hybrid approach for population replacement and empty cluster correction," *Egypt. Inform. J*., vol. 32, no. Dec., pp. 1–15, Dec. 2025, doi: 10.1016/j.eij.2025.100814.

[22] S. Zhao et al., "Optimizing cluster centroids with improved quadratic interpolation: An adaptive K means algorithm," *J. Comput. Appl. Math*., vol. 473, no. Feb., pp. 1–15, Feb. 2026, doi: 10.1016/j.cam.2025.116921.

[23] Y. Ping et al., "Beyond k Means++: Towards better cluster exploration with geometrical information," *Pattern Recognit*., vol. 146, no. Feb., pp. 1–14, Feb. 2024, doi: 10.1016/j.patcog.2023.110036.

[24] N. Bajpai, J. H. Paik, and S. Sarkar, "Balanced seed selection for K means clustering with determinantal point process," *Pattern Recognit*., vol. 164, no. 2025, pp. 1–15, 2025, doi: 10.1016/j.patcog.2025.111548.

[25] A. E. Ezugwu et al., "A comparative performance study of hybrid firefly algorithms for automatic data clustering," *IEEE Access*, vol. 8, no. 2020, pp. 121089–121118, 2020, doi: 10.1109/ACCESS.2020.3006173.

[26] A. Qtaish et al., "Optimization of K means clustering method using hybrid capuchin search algorithm," *J. Supercomput.*, vol. 79, no. 2023, pp. 15066–15091, 2023, doi: 10.1007/s11227-023-05540-5.

[27] H. He, "A clustering algorithm based on grids for core data and adjacency relationships for edge data," *Sci. Rep.*, vol. 15, no. 2025, pp. 1–12, 2025, doi: 10.1038/s41598-025-00532-2.

[28] A. M. Ikotun, F. Habyarimana, and A. E. Ezugwu, "Benchmarking validity indices for evolutionary K means clustering performance," *Sci. Rep.*, vol. 15, no. 2025, pp. 1–18, 2025, doi: 10.1038/s41598-025-08473-6.

[29] F. Ros and S. Guillaume, "A hierarchical clustering algorithm and an improvement of the single linkage criterion to deal with noise," *Expert Syst. Appl.*, vol. 128, no. Aug., pp. 96–108, Aug. 2019, doi: 10.1016/j.eswa.2019.03.031.

[30] R. Haripriya et al., "Decentralized big data mining: Federated learning for clustering youth tobacco use in India," *J. Big Data*, vol. 11, no. 2024, pp. 1–20, 2024, doi: 10.1186/s40537-024-01042-0.

[31] A. A. Amer et al., "Neighboring-aware hierarchical clustering: A new algorithm and extensive evaluation," *Int. J. Semantic Web Inf. Syst.*, vol. 20, no. 1, pp. 1–24, 2024, doi: 10.4018/IJSWIS.346377.

[32] O. E. Uchenna and M. O. Olusola, "Overview of agglomerative hierarchical clustering methods," *Br. J. Comput. Netw. Inf. Technol.*, vol. 7, no. 2, pp. 14–23, 2024, doi: 10.52589/BJCNIT-CV9POOGW.

[33] G. Mishra and S. K. Yadav, "A fast hybrid clustering technique based on local nearest neighbor using MST," *Expert Syst. Appl.*, vol. 125, no. Jul., pp. 143–152, Jul. 2019, doi: 10.1016/j.eswa.2019.01.025.

[34] M. Vichi and A. M. Candia, "Hierarchical means clustering," *J. Classif.*, vol. 39, no. 2, pp. 465–489, 2022, doi: 10.1007/s00357-022-09419-7.

[35] B. A. Pimentel and A. C. P. L. F. de Carvalho, "A meta-learning approach for recommending the number of clusters," *Knowl.-Based Syst.*, vol. 195, no. Feb., pp. 1–15, Feb. 2020, doi: 10.1016/j.knosys.2020.105682.

[36] B. Zhou, B. Lu, and S. Saeidlou, "A hybrid clustering method based on the several diverse basic clustering and meta-clustering aggregation technique," *Cybern. Syst.*, vol. 2022, no. Aug., pp. 203–229, Aug. 2022, doi: 10.1080/01969722.2022.2110682.

[37] M. M. Akhter and S. K. Mohanty, "A fast O(N log N) time hybrid clustering algorithm using the circumference proximity based merging technique for diversified datasets," *Eng. Appl. Artif. Intell.*, vol. 125, no. 2023, pp. 1–18, 2023, doi: 10.1016/j.engappai.2023.106737.

[38] K. M. Osei-Bryson, "A hybrid clustering algorithm: Combining K-means and hierarchical methods," *Comput. Oper. Res.*, vol. 34, no. 4, pp. 1017–1031, Apr. 2007, doi: 10.1016/j.cor.2005.06.017.

[39] G. Mishra and S. K. Mohanty, "A fast hybrid clustering technique based on local nearest neighbor using minimum spanning tree," *Expert Syst. Appl.*, vol. 132, no. Oct., pp. 28–43, Oct. 2019, doi: 10.1016/j.eswa.2019.04.048.

[40] A. Salehi and M. Khedmati, "Hybrid clustering strategies for effective oversampling and undersampling in multiclass classification," *Sci. Rep.*, vol. 15, no. 1, pp. 1–15, Jan. 2025, doi: 10.1038/s41598-024-84786-2.

[41] O. Manjang et al., "Anchor model-based hybrid hierarchical federated learning with overlap SGD," *IEEE Trans. Mobile Comput.*, vol. 23, no. 12, pp. 12540–12557, 2024, doi: 10.1109/TMC.2024.3414999.

[42] S. Pourahmad et al., "Does determination of initial cluster centroids improve the performance of K-means clustering algorithm? Comparison of three hybrid methods by genetic algorithm, minimum spanning tree, and hierarchical clustering in an applied study," *Comput. Math. Methods Med.*, vol. 2020, no. Aug., pp. 1–17, Aug. 2020, doi: 10.1155/2020/7636857.

[43] S. A. Mousavian Anaraki and A. Haeri, "Soft and hard hybrid balanced clustering with innovative qualitative balancing approach," *Inf. Sci.*, vol. 613, no. Oct., pp. 786–805, Oct. 2022, doi: 10.1016/j.ins.2022.09.044.

[44] L. Yin et al., "An improved hierarchical clustering algorithm based on the idea of population reproduction and fusion," *Electronics*, vol. 11, no. 17, pp. 1–15, Aug. 2022, doi: 10.3390/electronics11172735.

[45] N. Farhan and S. Rizvi, "An interference-managed hybrid clustering algorithm to improve system throughput," *Sensors*, vol. 22, no. 4, pp. 1–18, Feb. 2022, doi: 10.3390/s22041598.

[46] A. M. Ikotun, F. Habyarimana, and A. E. Ezugwu, "Cluster validity indices for automatic clustering: A comprehensive review," *Heliyon*, vol. 11, no. 2025, pp. 1–25, 2025, doi: 10.1016/j.heliyon.2025.e41953.

[47] Statistics Indonesia (BPS), Data and Information on Poverty at the Regency/City Level, 2024, BPS Journal of Indonesian Statistics, vol. 2024, no. Nov., pp. 1–120, Nov. 2024.

[48] G. Gan, C. Ma, and J. Wu, "Data clustering: Theory, algorithms, and applications," *SIAM Rev.*, vol. 2020, no. 2020, pp. 1–350, 2020, doi: 10.1137/1.9780898718348.

[49] Y. Zhang et al., "A comprehensive survey on traffic missing data imputation," *IEEE Trans. Intell. Transp. Syst*., vol. 25, no. 2, pp. 1234–1245, Feb. 2024, doi: 10.1109/TITS.2024.3478816.

[50] T. Emmanuel, "A survey on missing data in machine learning," *J. Big Data*, vol. 8, no. 1, pp. 1–37, Jan. 2021, doi: 10.1186/s40537-021-00516-9.

[51] Y. Zhou, S. Aryal, and M. R. Bouadjenek, "A comprehensive review of handling missing data: Exploring special missing mechanisms," *arXiv*, vol. 2024, no. Apr., pp. 1–28, Apr. 2024.

[52] A. Mirzaei, "Missing data in surveys: Key concepts, approaches, and applications," *Res. Social Adm. Pharm*., vol. 18, no. 2, pp. 2308–2316, Feb. 2022, doi: 10.1016/j.sapharm.2021.09.008.

[53] R. C. de Amorim and V. Makarenkov, "On k-means iterations and Gaussian clusters," *Neurocomputing*, vol. 553, no. Jul., pp. 1–12, Jul. 2023, doi: 10.1016/j.neucom.2023.126547.

[54] Y. Ma et al., "A multi-stage hierarchical clustering algorithm based on centroid of tree and cut edge constraint," *Inf. Sci*., vol. 557, no. Jan., pp. 194–219, Jan. 2021, doi: 10.1016/j.ins.2020.12.016.

[55] Y. Zeng et al., "A centroid guided cluster transformation for dynamic multi-objective optimization algorithm," *IEEE Congr. Evol. Comput*., vol. 2025, no. 2025, pp. 1–8, 2025, doi: 10.1109/CEC65147.2025.11043010.