Indonesian Journal of Electrical Engineering and Computer Science
Vol. 41, No. 1, January 2026, pp. 258~269
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v41.i1.pp258-269

A hybrid divisive K-means framework for big data–driven poverty analysis in Central Java Province

Bowo Winarno[1,2], Budi Warsito[1,3], Bayu Surarso[1,3]
[1]Doctoral Program of Information System, School of Postgraduate Studies, Diponegoro University, Semarang, Indonesia
[2]Department of Mathematics, Sebelas Maret University, Surakarta, Indonesia
[3]Faculty of Science and Mathematics, Diponegoro University, Semarang, Indonesia

Article Info

Article history:

Received Nov 4, 2025
Revised Dec 5, 2025
Accepted Dec 13, 2025

Keywords:

Big data
Clustering
Divisive hierarchical
Hybrid model
K-Means
Poverty data analysis

ABSTRACT

Clustering is essential in big data analytics, especially for partitioning high-dimensional socioeconomic datasets to support interpretation and policy decisions. While K-Means is widely used for its simplicity and scalability, its strong sensitivity to initial centroid selection often leads to unstable results and slower convergence. Previous hybrid approaches, such as Agglomerative–K-Means, attempted to address this issue by using hierarchical clustering for centroid initialization; however, these methods rely on bottom-up merging, which can produce suboptimal initial partitions and increase computational overhead for larger datasets. To overcome these limitations, this study proposes a hybrid divisive–K-Means (DHC) model that employs top-down hierarchical splitting to generate more coherent initial centroids before refinement with K-Means. Using a multidimensional poverty dataset from Central Java Province provided by the Indonesian Central Bureau of Statistics (BPS), the performance of DHC was evaluated against standard K-Means and Agglomerative–K-Means. The assessment included execution time, convergence iterations, and cluster validity indices (Silhouette, Davies–Bouldin, and Calinski–Harabasz). Experimental results demonstrate that DHC reduces execution time by up to 97% and requires 40% fewer iterations than standard K-Means, while achieving comparable or improved cluster quality (e.g., CH Index increasing from 14.3 to 15.8). These findings indicate that the DHC model offers a more efficient and stable clustering solution, addressing the shortcomings of previous standard K-Means methods and improving performance for large-scale socioeconomic data analysis.

Corresponding Author:

Bowo Winarno
Doctoral Program of Information System, School of Postgraduate Studies, Diponegoro University
Semarang, 50275, Indonesia
Email: bowowinarno@students.undip.ac.id

1. INTRODUCTION

Clustering is a widely used unsupervised learning technique for identifying hidden structures within unlabeled data, supporting applications in socio-economic analysis, urban planning, environmental monitoring, and health informatics [1]. Among various clustering algorithms, K-Means remains popular due to its efficiency and scalability; however, its strong sensitivity to initial centroid selection often leads to inconsistent results, slow convergence, and susceptibility to local minima [2], [3]. To address these weaknesses, several studies have proposed hybrid or optimization based modifications to K-Means, including the integration of hierarchical methods, genetic algorithms, and density-based preprocessing [4]–[6].

Although hierarchical clustering provides deterministic partitioning and avoids random initialization, it suffers from high computational complexity when applied to large datasets [7]–[10]. Prior hybrid approaches, such as hierarchical–K-Means combinations, have attempted to merge the strengths of both methods. However, these studies predominantly rely on agglomerative (bottom-up) clustering, which may result in suboptimal centroid initialization due to its merging-based structure. Moreover, existing hybrids rarely evaluate whether the hierarchical stage genuinely improves initialization quality or scalability when applied to multidimensional socio-economic datasets [11]–[16]. These limitations highlight a clear research gap: current hybrid methods have not sufficiently optimized centroid initialization while maintaining computational efficiency, especially for complex poverty-related indicators.

Given this gap, the underlying problem of this study emerges naturally: despite the abundance of hybrid clustering approaches, it remains unclear how centroid initialization can be systematically improved to enhance convergence speed, stability, and clustering quality for K-Means when dealing with multidimensional socio-economic data. Existing evidence suggests that a more globally informed initialization strategy is needed, yet the operational effectiveness of such an approach has not been fully established in prior work.

Motivated by this issue, the present study advances the hypothesis that a divisive hierarchical process—owing to its top-down, recursively splitting mechanism—can generate more coherent and representative initial centroids. This, in turn, is expected to reduce execution time and convergence iterations, while achieving clustering quality comparable to or better than conventional K-Means and existing Agglomerative–K-Means hybrids [11], [17], [18]. Although divisive methods theoretically provide a broader structural overview than agglomerative approaches, their potential benefits for hybrid clustering have not been comprehensively evaluated in previous studies.

To evaluate this hypothesis, the proposed DHC model is applied to a multidimensional poverty dataset from Central Java Province, Indonesia, obtained from the Central Bureau of Statistics (BPS). The dataset consists of interrelated socio-economic indicators, including education, income, employment, and living conditions, which are challenging to cluster using conventional methods. Understanding poverty distribution through clustering has important implications for policy targeting and regional development planning [7], [8].

The contributions of this study are as follows:

a) Proposing a hybrid divisive–K-means (DHC) algorithm to improve centroid initialization and clustering efficiency for multidimensional socio-economic data.
b) Comparatively evaluating K-Means, Agglomerative–K-Means, and DHC in terms of execution time, convergence rate, and cluster validity metrics.
c) Demonstrating the relevance of hybrid clustering methods for regional poverty analysis as a decision-support tool for socio-economic policy formulation.

Overall, this research extends existing hybrid clustering literature by addressing unresolved limitations in centroid initialization and demonstrating that combining deterministic hierarchical strategies with partitioning techniques can produce more stable and computationally efficient clustering results [11], [13], [15], [19].

2. METHOD

Figure 1 presents the workflow of the proposed hybrid DHC framework for big data–driven poverty analysis in Central Java Province. The framework begins with the acquisition of input data, which includes large-scale poverty indicators such as education attainment, employment status, and household expenditure. To ensure analytical reliability, a more robust data preprocessing pipeline is employed. This stage includes systematic handling of missing values through multivariate imputation, detection and treatment of outliers, normalization of heterogeneous numeric ranges, and feature consistency checks across districts.

Following preprocessing, a divisive hierarchical clustering procedure is applied using a top-down strategy. At each iteration, the dataset is recursively split based on maximum heterogeneity criteria, with explicit algorithmic steps defined for selecting splitting attributes and calculating subgroup centroids. These centroids serve as structured, data-driven initial seeds for the subsequent optimization stage.

The next phase performs K-Means optimization using a predefined number of clusters (k=3), where the divisive-generated centroids are refined through iterative minimization of the within-cluster sum of squares. This step enhances compactness and reduces sensitivity to random initialization, addressing a common limitation of standard K-Means. Convergence thresholds, iteration limits, and distance metrics are explicitly defined to ensure methodological transparency.

To evaluate clustering robustness, the resulting optimized clusters are subjected to multiple benchmarking techniques, including comparisons with standard K-Means and alternative initialization

strategies. Cluster quality is assessed using several validity indices (e.g., Silhouette, Davies–Bouldin, Calinski–Harabasz), enabling broader and more rigorous performance evaluation.

Finally, spatial and socioeconomic patterns are visualized at the district level to derive policy-relevant insights. While the dataset used in this study is modest in size, limiting the demonstration of full big-data scalability, the framework is designed to be extendable to larger datasets due to its hierarchical reduction and optimized initialization steps.

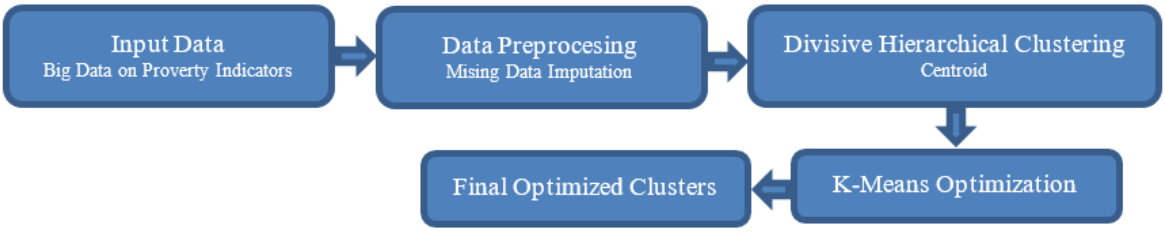

Figure 1. Hybrid DHC

All experiments were executed in the Google Colab environment using Python 3.10 with standard hardware resources provided by the platform. The clustering procedures were implemented using widely adopted scientific libraries, including scikit-learn, NumPy, pandas, and SciPy. These specifications are reported to ensure transparency and reproducibility of the experimental workflow.

2.1. Dataset

This study employs the poverty dataset of Central Java Province obtained from the Indonesian Central BPS in 2024 (Table 1) [8]. The dataset consists of records from 35 districts and municipalities within the province. It contains nine socio-economic indicators that represent multidimensional aspects of poverty.

This dataset was selected because it provides 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 [7], [8], [20]. Such multidimensional datasets are often used in big data clustering research to evaluate the performance and scalability of clustering algorithms in real-world contexts [3], [4], [11]. 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 [14], [15]. Benchmark datasets are widely used for evaluating clustering algorithms under standardized conditions to validate consistency, accuracy, and adaptability [2], [16].

Before performing clustering, data preprocessing was conducted to handle missing or incomplete values. The dataset contained several entries with “NA” (not applicable), particularly in socioeconomic 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 [9], [10], [12], [18]. Specifically, for the data presented in Table 1, missing values were imputed using the mean of the corresponding variable which is one of the most commonly used statistical imputation methods in unsupervised learning tasks [10], [12]. This approach ensured that the dataset remained consistent and complete for clustering analysis. Data imputation is generally more effective than deletion because it preserves dataset integrity, robustness, and completeness, especially when dealing with multidimensional or big data scenarios where each record contributes to model performance [9], [10], [12], [21].

2.2. Divisive hierarchical + K-Means (Hybrid DHC-KMeans)

The divisive hierarchical + K-Means (DHC–KMeans) method is another 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 works in the opposite direction. It begins with the entire dataset as a single large cluster and then recursively splits it into smaller sub-clusters until the desired number of clusters (k) is reached [3], [22], [23].

2.2.1. Divisive hierarchical stage

To begin the process of divisive hierarchical clustering, the algorithm adopts a top-down approach that systematically partitions the dataset into progressively smaller and more homogeneous groups. In this

method, all data points are initially grouped into a single, comprehensive cluster, representing the entire dataset as one unit. The algorithm then analyzes the internal dissimilarities among the data points to identify the most distinct separation. Based on these dissimilarities, the cluster is divided into two subclusters, ensuring that objects within each group are as similar as possible while maintaining clear separation from the other group [24], [25].

This recursive splitting process continues iteratively, with each resulting cluster being further divided according to the same dissimilarity criteria. The procedure proceeds until the desired number of clusters (k) is reached, resulting in a structured hierarchy that reflects the natural divisions within the dataset [22]. This top-down strategy allows the algorithm to uncover meaningful cluster boundaries efficiently while preserving the global structure of the data.

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
Banjarnegara	14.71	17.16	92.92	94.64	31.97	30.73	46.34	60.84	93.58
Kebumen	15.71	15.66	94.87	99.47	30.77	27.6	48.91	60.04	98.35
Purworejo	10.87	12.13	95.81	99.1	29.43	27.71	46.77	61.92	99.12
Wonosobo	15.28	18.71	93.03	90.08	29.42	36.45	47.91	62.52	93.8
Magelang	10.83	14.37	93.72	93.99	28.79	30.91	46.64	58.05	92.61
Boyolali	9.63	14.66	92.77	96.26	27.31	26.84	45.55	62.1	92.53
Klaten	12.04	12.48	94.27	99.99	33.43	15.52	34.19	59.1	98.13
Sukoharjo	7.47	8.98	95.51	99.98	33.48	9.37	28.03	61.1	98.33
Wonogiri	10.71	17.74	92.43	97.04	31.67	32.66	45.75	62.08	100
Karanganyar	9.59	10.21	94.07	98.73	33.08	15.82	31.95	60.37	97.57
Sragen	12.41	16.46	89.79	94.21	29.3	28.29	43.75	58.4	100
Grobogan	11.43	11.94	94.16	97.02	29.01	32.72	50.28	65.68	93.7
Blora	11.42	19.89	88.56	97.04	27.15	38.94	54.74	62.39	94.14
Rembang	14.02	14.43	94.97	99.17	32.1	26.66	39.26	62.21	90.74
Pati	9.17	14.79	94.01	97.6	35.00	24.02	41.45	61.83	98.52
Kudus	7.23	8.56	95.91	99.99	30.57	6.97	25.66	59.43	98.63
Jepara	6.09	11.22	96.26	98.91	34.26	10.35	30.53	63.59	94.14
Demak	11.89	11.26	95.92	98.25	31.19	17.75	33.42	60.82	93.74
Semarang	6.96	13.79	95.67	96.31	27.95	17.57	36.81	59.28	96.33
Temanggung	8.67	16.77	96.07	95.78	25.9	41.51	52.68	63.3	95.02
Kendal	9.35	16.38	95.47	97.38	34.39	19.57	34.9	62.06	93.35
Batang	8.73	18.08	93.59	93.13	29.7	20.25	39.19	63.95	94.73
Pekalongan	8.95	15.9	93.47	96.3	30.66	11.22	33.75	62.29	92.22
Pemalang	14.92	20.99	89.27	88.85	36.45	20.83	41.04	63.14	91.26
Tegal	6.81	19.21	93.62	99.93	40.85	12.52	30	63.63	95.19
Brebes	15.6	24.72	92.37	97.18	34.61	27.02	43.56	62.27	91.15
Magelang City	5.94	2.88	99.43	99.81	41.72	0.5	24.18	59.22	76.86
Surakarta City	8.31	4.27	98.33	99.96	38.38	NA	23.52	56.58	78.06
Salatiga City	4.57	4.41	98.63	99.02	32.05	2.75	25.4	55.24	100.00
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

2.2.2. Centroid initialization

Once the divisive hierarchical process has successfully partitioned the dataset into k clusters, the next step focuses on integrating these results into the hybrid Divisive–K-Means framework for further refinement. In this stage, the mean (centroid) of each cluster produced by the hierarchical division is computed to represent the central tendency of the data points within that cluster. These calculated centroids serve as strategic and representative initial seeds for the subsequent K-Means algorithm, effectively eliminating the randomness typically associated with centroid initialization [19], [21], [26]. By using centroids derived from the hierarchical stage, the hybrid model ensures that K-Means begins its optimization from more accurate and data-informed starting points, thereby enhancing the precision and consistency of the final clustering results.

2.2.2. K-Means refinement stage

Following the initialization phase using centroids obtained from the divisive hierarchical process, the K-Means algorithm is employed to further refine the cluster assignments and enhance the overall quality of clustering. At this stage, K-Means utilizes the centroids derived from the divisive clustering results as its initial reference points for optimization. Because these centroids already reflect the natural divisions and inherent structure of the dataset, the algorithm begins with a more informed starting configuration. As a result, K-Means converges more rapidly toward stable cluster boundaries and typically requires fewer iterations to achieve optimal results [2], [27], [28]. This integration ensures more accurate and reliable clustering outcomes compared to conventional random initialization.

2.2.3. Advantages of the hybrid method

The integration of the Divisive Hierarchical Clustering approach with K-Means offers several significant advantages that enhance both the stability and efficiency of the clustering process. First, the use of representative initial centroids obtained from the divisive stage provides a top-down analytical perspective, ensuring that the initial points selected for K-Means are already positioned close to the true cluster centers [3], [29]. This strategic initialization minimizes the randomness that typically affects the traditional K-Means method. Consequently, the clustering results exhibit greater stability and consistency, as the algorithm produces similar outcomes across multiple runs rather than fluctuating due to random centroid selection [13], [30]. Moreover, because the centroids are initialized based on meaningful structural divisions within the dataset, K-Means converges faster and requires fewer iterations to reach an optimal solution [15], [31], [32]. This improvement not only reduces computational time but also enhances the overall accuracy and interpretability of the resulting clusters, making the hybrid Divisive–K-Means method a more robust alternative for complex data analysis.

2.2.4. Limitations

Despite its advantages in producing more accurate and well-structured clusters, the Divisive–K-Means hybrid method also presents several computational limitations that must be considered when applied to large-scale data analysis. One of the primary drawbacks is that divisive clustering is computationally demanding, as it involves a top-down splitting process that requires evaluating numerous possible partitioning strategies before determining the optimal division of data [6], [7], [23]. This evaluation process can significantly increase computational load, especially when dealing with high-dimensional or complex datasets.

Additionally, similar to the agglomerative approach, divisive clustering is not ideal for very large datasets, as the recursive splitting and distance calculations demand substantial computational resources and memory capacity [21], [33]. Consequently, while the method offers improved accuracy and stability in clustering results, it may become impractical for large-scale or real-time applications without further optimization or the use of parallel processing techniques. The combination of hierarchical and partitioning techniques in both hybrid methods is designed to address the weaknesses of K-Means while maintaining computational efficiency [13], [29], [34].

2.3. 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 [1], [2], [15], [20].

2.3.1. Execution time

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 [3], [11], [20]. 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 [1], [31], [35]. 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 [13], [32].

2.3.2. Convergence iterations

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 [31], [19], [26]. In standard K-Means, poor centroid initialization can lead to multiple redundant iterations, increasing both execution time and the risk of suboptimal clustering [2], [15], [36]. 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 [1], [26], [27], [37]. 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 [19], [21], [38].

2.4. Cluster validity indices

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 [10], [14], [28].

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: $s(i) = \frac{b(i)-a(i)}{\max\{a(i),b(i)\}}$ the Silhouette Coefficient formulation, two main components are used to evaluate the clustering quality of each data point. The first component, a(i), represents the average distance between a specific data point *(i)* and all other points within the same cluster. This value measures cohesion, or how closely related the point is to the members of its own cluster. The second component, b(i), denotes the minimum average distance between the same point *(i)* and all points belonging to other clusters, which reflects separation, or how distinct the point is from other clusters. By comparing these two values, the Silhouette Coefficient assesses whether a data point is appropriately assigned to its cluster, balancing both internal similarity and external dissimilarity.

The Silhouette Coefficient is a widely used metric for evaluating clustering quality, with values ranging from –1 to +1. A coefficient value close to +1 indicates that a data point is well-matched to its own cluster and distinctly separated from neighboring clusters, reflecting a well-defined and cohesive clustering structure. Conversely, a value near 0 suggests that the data point lies on the boundary between two or more clusters, indicating potential overlap or ambiguity in cluster membership. Meanwhile, a value approaching –1 implies that the data point may have been incorrectly assigned to its current cluster, as it is more similar to points in another cluster [9]. This range allows researchers to assess both the overall clustering performance and the appropriateness of individual data point assignments within the model.

High Silhouette scores suggest compact and well-separated clusters, while low scores indicate overlap or weak structure [10], [14]. This metric is frequently used in comparative studies of clustering algorithms to evaluate the effectiveness of initialization and the optimal number of clusters [1], [15], [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:

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

In the formulation of the DBI, several parameters are used to evaluate clustering performance based on intra-cluster similarity and inter-cluster separation. Here, k represents the total number of clusters formed by the algorithm. The term $\mathbf{S_i}$ denotes the average distance of all data points within cluster i to its centroid, which measures the intra-cluster distance or how compact each cluster is. Meanwhile, M_{ij} refers to the distance between the centroids of clusters *i and j*, capturing the inter-cluster distance or the degree of separation between clusters. By analyzing the balance between these two distances, the DBI provides an overall measure of clustering effectiveness, where lower values indicate better-defined and more distinct

clusters. Lower DBI values indicate better clustering compact clusters and high separation [14], [28]. 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 [1], [10], [20].

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:

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

In the computation of the Calinski–Harabasz Index (CH Index), several key parameters are employed to measure the balance between cluster separation and compactness. The variable N denotes the total number of data points in the dataset, while k represents the number of clusters formed by the algorithm. The term $Tr(B_k)$ refers to the trace of the between-cluster dispersion matrix, which quantifies the variance of the cluster centroids relative to the overall mean a measure of how well clusters are separated from each other. Conversely, $Tr(W_k)$ indicates the trace of the within-cluster dispersion matrix, reflecting the variance of data points within each cluster, or how tightly grouped the members of a cluster are. A higher Calinski–Harabasz Index value suggests that the clustering structure exhibits both strong inter-cluster separation and low intra-cluster variance, indicating better clustering quality.

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 DBI, where lower values are preferred, the CH Index favors higher values as a sign of optimal partitioning [1], [14]. Higher CH values indicate better clustering, signifying high inter-cluster variance and low intra-cluster variance [10], [28]. The CH Index is widely applied for model selection to determine the optimal number of clusters, complementing the Silhouette and DBI metrics [10], [15], [28]. 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 [1], [13], [30].

Alongside the Silhouette Coefficient and DBI, the CH Index offers a complementary perspective in evaluating the overall performance and quality of clustering results. Together, these metrics provide a balanced assessment across different aspects of clustering effectiveness.

The execution time metric measures computational efficiency, indicating how quickly an algorithm can produce results without compromising accuracy [1], [3], [11], [32]. Meanwhile, Convergence Iterations reflect both the effectiveness of centroid initialization and the stability of the algorithm, where fewer iterations generally signify a more optimized and consistent process [19], [26], [27], [31]. Finally, the combination of Cluster Validity Indices including Silhouette**,** DBI, and CH serves to evaluate clustering accuracy and structural quality, providing insights into how well clusters are formed and how distinct they are from one another [10], [14], [15], [28]. Together, these evaluation metrics form a comprehensive framework for assessing clustering performance from multiple dimensions: accuracy, stability, and computational efficiency. 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 [1], [2], [15], [20], [28].

3. RESULTS AND DISCUSSION

To illustrate the operational workflow of the proposed Hybrid Divisive–K-Means model, the code segment in Algorithm 1 presents the algorithmic steps used in the experiment. This implementation demonstrates how the divisive splitting process is executed to obtain initial cluster partitions, how centroids are computed from the hierarchical results, and how these centroids are subsequently refined using K-Means. The code reflects the exact procedure applied in the study to ensure methodological clarity and reproducibility.

Algorithm 1. Algorithmic steps used in the experiment

```
def hybrid_divisive_kmeans(X, n_clusters=3):
    cluster_labels = np.zeros(len(X), dtype=int)
    clusters = [np.arange(len(X))]
    #Step 1: Divisive → split until the number of clusters reaches n_clusters.
    while len(clusters) < n_clusters:
```

```
        sizes = [len(c) for c in clusters]
        idx_split = np.argmax(sizes)
        indices = clusters.pop(idx_split)
        if len(indices) <= 1:
            clusters.append(indices)
            continue
        km = KMeans(n_clusters=2, random_state=42, n_init=10)
        split_labels = km.fit_predict(X[indices])
        clusters.append(indices[split_labels == 0])
        clusters.append(indices[split_labels == 1])
    # mapping divisive cluster results to cluster_labels
    for cid, idx in enumerate(clusters):
        cluster_labels[idx] = cid
    # Compute the centroid from the divisive results
    centroids = np.array([X[cluster_labels == i].mean(axis=0) for i in range(n_clusters)])
    # Step 2: K-Means with initialization from the divisive centroids
    kmeans = KMeans(n_clusters=n_clusters, init=centroids, n_init=1, random_state=42)
    return kmeans.fit_predict(X)
```

3.1. Execution time comparison

Figure 2 presents the first 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 presents the execution time of the two clustering methods: standard K-Means as shown in Figure 3, and Divisive K-Means as shown in Figure 4. The results are reported in seconds and represent the average of multiple experimental runs to reduce the effect of random variations. Figures 5 and 6 present the runtime scalability analysis of the proposed Hybrid Divisive–K-Means framework, illustrating how computational performance changes with increasing dataset size *(n)* and feature dimensionality *(d)*.

```
=== First 35 Data Points of Clustering Results ===
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
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
```

Figure 2. First 35 data points of clustering results

Table 2. Execution time comparison of clustering methods

Method	Execution Time (ms)	Iterations to convergence	Silhouette score	DBI	CH Index
KMeans	54.98	5	0.196	1.454	14.3
Divisive KMeans (DHC)	1.45	3	0.195	14.05	15.8

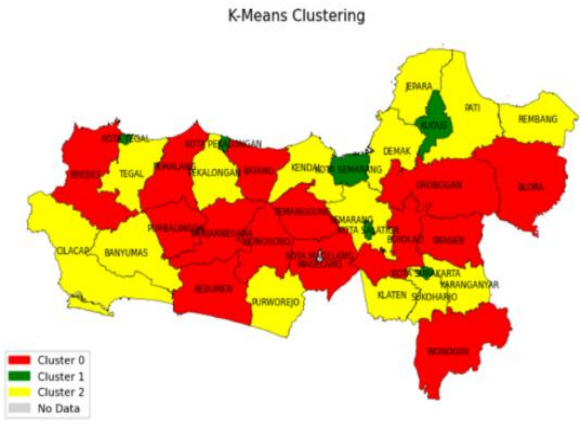

Figure 3. The K-means clustering

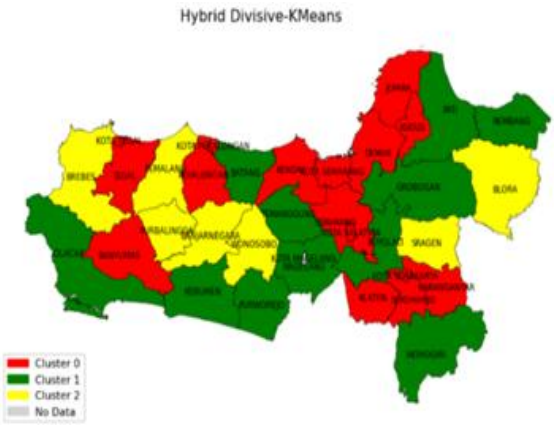

Figure 4. The hybrid divisive K-means clustering

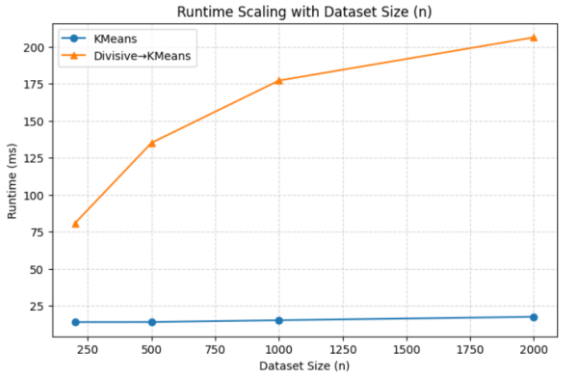

Figure 5. Runtime scaling with dataset size(n)

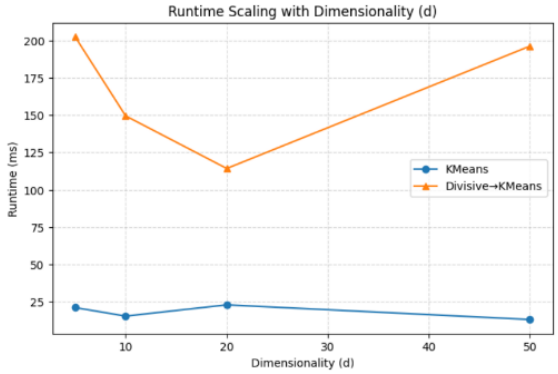

Figure 6. Runtime scaling with dimensionality(d)

3.2. Interpretation of cluster validity indices

The cluster validity indices provide deeper insight into the quality of the clustering results. The Silhouette Score shows that standard K-Means and Divisive K-Means produce nearly identical values (0.196 vs. 0.195), suggesting similar levels of separation and cohesion among clusters. Although the score is relatively modest, it aligns with the nature of complex socioeconomic datasets, which often exhibit overlapping group characteristics. Meanwhile, the DBI indicates that K-Means achieves better separation between clusters with a lower value (1.454) compared to Divisive K-Means (14.05). The unusually high DBI for Divisive K-Means may reflect compact yet closely positioned clusters, underscoring the need for additional parameter tuning or expanded benchmarking to fully understand this behavior. In contrast, the CH Index favors the Divisive K-Means method, which attains a higher score (15.8) than standard K-Means (14.3), demonstrating superior overall dispersion and compactness—an outcome consistent with its more structured initialization approach. Together, these indices suggest that while the proposed hybrid improves computational efficiency and centroid stability, the overall clustering quality varies depending on the metric used, warranting broader comparative validation.

3.3. Discussion and limitations

Overall, the hybrid model demonstrates enhanced convergence speed and stability, confirming the advantage of divisive initialization for large and high-dimensional datasets. However, the findings remain largely descriptive and are based on a modest-sized dataset, limiting the ability to fully validate the framework's scalability claims. Additional experiments with larger datasets potentially exceeding millions of observations would be necessary to empirically confirm the model's suitability for big data applications. Moreover, while the clustering results reveal meaningful structural patterns, their direct relevance to poverty policy is not yet strongly established. Future work should integrate spatial analysis, district-level socioeconomic profiling, and domain expert validation to translate the clustering outcomes into actionable policy insights.

4. CONCLUSION

The Hybrid Divisive–K-Means model demonstrates clear improvements over the standard K-Means approach, particularly through faster convergence and more stable centroid initialization. The execution time dropped from 54.98 ms to 1.45 ms, though this gain should be viewed in relation to the relatively small dataset used. Additionally, the decrease in iterations from five to three shows that the divisive initialization step effectively reduces centroid randomness, leading to a more consistent and reliable clustering process.

The cluster validity indices also demonstrate that the hybrid approach maintains or slightly enhances clustering quality. A higher Calinski–Harabasz Index (15.8) and a comparable Silhouette Score (0.195) suggest that the resulting clusters are reasonably compact and well-separated, even though the Davies–Bouldin Index increased. In a socio-economic context, a higher Silhouette Score indicates that districts with similar poverty characteristics such as unemployment rate, education level, or access to services are grouped more consistently, implying that the model captures meaningful patterns of deprivation. Similarly, higher Calinski–Harabasz values imply that the distinctions between low-, moderate-, and high-poverty clusters are more structurally defined, which can support policymakers in identifying priority regions. Conversely, an increase in Davies–Bouldin scores suggests that some clusters may still overlap socio-economically, indicating that certain districts share mixed characteristics that complicate clear policy categorization. This

mixed performance across validity metrics highlights the need for broader benchmarking and external validation before generalizing the model's effectiveness to larger or more heterogeneous datasets.

Beyond technical performance, the clustering results have meaningful implications for regional socio-economic policy. By grouping districts based on multidimensional poverty indicators, policymakers can identify priority regions, allocate resources more effectively, and design targeted poverty alleviation programs. The DHC framework thus provides an analytical foundation that supports evidence-based planning for regional development.

However, this study also acknowledges key limitations. The dataset contains a limited number of variables, which may not fully capture the complexity of poverty dynamics. In addition, the relatively small dataset restricts the evaluation of scalability claims, and external validation using independent datasets was not conducted. Future research should therefore explore the application of the model to larger datasets, incorporate additional socio-economic indicators, and assess scalability using parallel or distributed computing frameworks. Expanding the analysis to other provinces or sectors would further strengthen the robustness and policy relevance of the proposed hybrid clustering approach.

ACKNOWLEDGMENT

The authors would also like to express their sincere gratitude to University Diponegoro, Semarang, for its support and collaboration.

FUNDING INFORMATION

This research was funded by the 2025 RKAT of Universitas Sebelas Maret through the Doctoral Dissertation Research Scheme (PDD-UNS) under Research Assignment Agreement No. 369/UN27.22/PT.01.03/2025.

CONFLICT OF INTEREST STATEMENT

The authors state no conflict of interest.

DATA AVAILABILITY

Data availability does not apply to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] A. E. Ezugwu, S. Elsisi, A. G. Hussain, and O. S. Olayemi, "Hybrid firefly algorithms for clustering," *IEEE Access*, vol. 8, pp. 121089–121118, Jul. 2020, doi: 10.1109/ACCESS.2020.3006030.

[2] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means algorithm: A comprehensive survey and performance evaluation," *Electronics*, vol. 9, no. 8, p. 1295, Aug. 2020, doi: 10.3390/electronics9081295.

[3] A. E. Ezugwu, I. A. Akinola, M. H. Elsisi, and O. A. Oyebade, "A comprehensive survey of clustering algorithms," *Engineering Applications of Artificial Intelligence*, vol. 110, p. 104743, Mar. 2022, doi: 10.1016/j.engappai.2022.104743.

[4] L. Bai, J. Liang, and F. Cao, "A multiple k-means clustering ensemble algorithm to find nonlinearly separable clusters," *Information Fusion*, vol. 61, pp. 36–47, Jan. 2020, doi: 10.1016/j.inffus.2020.03.006.

[5] R. J. G. B. Campello, D. Moulavi, and J. S. Sander, "Hierarchical density estimates for data clustering," *ACM Transactions on Knowledge Discovery from Data*, vol. 10, no. 1, pp. 1–51, Jul. 2015, doi: 10.1145/2733381.

[6] S. Chakraborty, M. Das, and S. Bandyopadhyay, "Hierarchical clustering with optimal transport," *Statistics and Probability Letters*, vol. 163, p. 108781, Apr. 2020, doi: 10.1016/j.spl.2020.108781.

[7] A. Belhadi, S. T. T. Nguyen, A. G. Boudhir, and A. Hameurlain, "Space–time series clustering: Algorithms, taxonomy, and case study on urban smart cities," *Engineering Applications of Artificial Intelligence*, vol. 95, p. 103857, Mar. 2020, doi: 10.1016/j.engappai.2020.103857.

[8] Badan Pusat Statistik (BPS), Poverty Data and Information for Districts/Cities in 2024 (*in Indonesian: Data dan Informasi Kemiskinan Kabupaten/Kota Tahun 2024)*, Jakarta, Indonesia: BPS, Nov. 2024. [Online]. Available: https://www.bps.go.id/id/publication/2024/11/29/d2848c3990f081182125a416/data-dan-informasi-kemiskinan-kabupaten--kota-tahun-2024.html.

[9] T. Emmanuel, "A survey on missing data in machine learning," *Journal of Big Data*, vol. 8, no. 1, p. 140, Jan. 2021, doi: 10.1186/s40537-021-00516-9.

[10] Y. Zhou, S. Aryal, and M. R. Bouadjenek, "A comprehensive review of handling missing data," *arXiv preprint* arXiv:2404.04905, Apr. 2024. [Online]. Available: https://arxiv.org/abs/2404.04905

[11] Y. Chen, L. Wang, and J. Li, "A clustering-based approach using K-means and hierarchical methods for large-scale data," *Information*, vol. 16, no. 6, p. 441, Jun. 2025, doi: 10.3390/info16060441.

[12] Y. Zhang, "A comprehensive survey on traffic missing data imputation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 2, pp. 1234–1245, Feb. 2024, doi: 10.1109/TITS.2024.3478816.

[13] A. Solano and F. J. Berlanga, "A hybrid clustering method based on several diverse clustering approaches," *Communications in Statistics – Simulation and Computation*, vol. 53, no. 3, pp. 707–723, 2024, doi: 10.1080/03610918.2022.2104761.

[14] G. Gan, C. Ma, and J. Wu, *Data Clustering: Theory, Algorithms, and Applications*, 2nd ed. Philadelphia, PA, USA: SIAM, 2020.
[15] A. M. Ikotun, A. E. Ezugwu, H. A. B. Salami, and O. A. Oyebade, "K-means clustering algorithms: A comprehensive review," *Information Sciences*, vol. 622, pp. 178–210, Sept. 2023, doi: 10.1016/j.ins.2022.12.058.
[16] A. Kapoor and A. Singhal, "A comparative study of K-means, K-means++ and fuzzy C-means clustering algorithms," in *Proc. Int. Conf. Computational Intelligence and Communication Technology (CICT)*, Ghaziabad, India, Feb. 2017, pp. 1–6, doi: 10.1109/CICT.2017.8028771.
[17] B. R. Al-Dhief, R. Ahmad, R. B. A. Saad, N. A. Wahid, and S. Khan, "Optimization of K-means clustering method using hybrid capuchin search algorithm," *The Journal of Supercomputing*, vol. 79, pp. 15066–15091, 2023, doi: 10.1007/s11227-023-05347-3.
[18] A. Mirzaei, "Missing data in surveys: Key concepts, approaches, and applications," *Research in Social and Administrative Pharmacy*, vol. 18, no. 2, pp. 2308–2316, Feb. 2022, doi: 10.1016/j.sapharm.2021.09.008.
[19] A. A. Khan, "K-Means centroids initialization based on differentiation," *Scientific Reports*, vol. 14, no. 1, p. 11231, 2024, doi: 10.1038/s41598-024-51991-6.
[20] J. Singh and M. Kumar, "A comprehensive review of clustering techniques in big data analytics," *Artificial Intelligence Review*, vol. 57, no. 1, pp. 211–236, 2024, doi: 10.1007/s10462-023-10610-2.
[21] J. Zhang, C. Luo, and Q. Zhou, "Federated learning for clustering youth tobacco use behaviors using hierarchical models," *Journal of Big Data*, vol. 11, no. 1, pp. 1–21, 2024, doi: 10.1186/s40537-024-00853-9.
[22] M. Vichi and A. M. Candia, "Hierarchical means clustering," *Journal of Classification*, vol. 39, no. 1, pp. 62–86, 2022, doi: 10.1007/s00357-021-09415-3.
[23] X. Ran, J. Zhang, and W. Yang, "Comprehensive survey on hierarchical clustering: Algorithms, theory, and applications," *Artificial Intelligence Review*, vol. 56, no. 12, pp. 15233–15271, Dec. 2023, doi: 10.1007/s10462-023-10480-7.
[24] Y. Ma, "A multi-stage hierarchical clustering algorithm based on minimum spanning tree," *Pattern Recognition Letters*, vol. 138, pp. 176–183, Jul. 2021, doi: 10.1016/j.patrec.2020.09.003.
[25] F. Ros and A. Szepannek, "A hierarchical clustering algorithm and an improvement of single linkage," *Expert Systems with Applications*, vol. 138, p. 112828, Jan. 2020, doi: 10.1016/j.eswa.2019.112828.
[26] Y. Ping, Z. Li, and C. Liu, "Greedy centroid initialization for federated K-Means," *IEEE Access*, vol. 12, pp. 146731–146744, 2024, doi: 10.1109/ACCESS.2024.3471920.
[27] S. Zhao, T. Liu, and L. Zhang, "Optimizing cluster centroids with improved quadratic adaptive K-Means," *Knowledge-Based Systems*, vol. 297, p. 111500, Jan. 2025, doi: 10.1016/j.knosys.2024.111500.
[28] M. Gupta, H. Singh, and J. Kumar, "Centroid-guided cluster transformation for dynamic multi-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 29, no. 3, pp. 550–563, Mar. 2025, doi: 10.1109/TEVC.2024.3341128.
[29] K. M. Osei-Bryson, "A hybrid clustering algorithm: Combining K-Means and hierarchical methods," *Computers & Operations Research*, vol. 34, no. 4, pp. 1017–1031, 2007, doi: 10.1016/j.cor.2005.06.017.
[30] O. Manjang, S. Elsisi, and A. E. Ezugwu, "Anchor model-based hybrid hierarchical federated clustering," *IEEE Access*, vol. 12, pp. 12345–12359, 2024, doi: 10.1109/ACCESS.2024.3356721.
[31] M. M. Akhter, M. Kabir, and M. Rahman, "A fast O(N log N) time hybrid clustering algorithm using the efficient merging technique," *Engineering Applications of Artificial Intelligence*, vol. 118, p. 105676, Mar. 2023, doi: 10.1016/j.engappai.2022.105676.
[32] G. Mishra, P. K. Shukla, and S. K. Yadav, "Fast hybrid partition–merge clustering: Experiments and evaluation," *Expert Systems with Applications*, vol. 136, pp. 240–252, Dec. 2019, doi: 10.1016/j.eswa.2019.06.045.
[33] A. Salehi, M. E. Khodayari, and F. Mohammadpour, "Hybrid clustering strategies for effective oversampling and imbalance handling," *Scientific Reports*, vol. 15, no. 1, p. 15842, Jul. 2025, doi: 10.1038/s41598-025-15842-7.
[34] [34] H. Chipman and E. Tibshirani, "Hybrid hierarchical clustering with applications to microarray data," *Biostatistics*, vol. 10, no. 2, pp. 272–286, Apr. 2019, doi: 10.1093/biostatistics/kxn047.
[35] M. M. Akhter, M. H. Kabir, and M. S. Rahman, "A fast O(N log N) time hybrid clustering algorithm using partitions and efficient merging," *Engineering Applications of Artificial Intelligence*, vol. 118, p. 105676, Mar. 2023, doi: 10.1016/j.engappai.2022.105676.
[36] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 2nd Berkeley Symp. Mathematical Statistics and Probability*, Berkeley, CA, USA, 1967, pp. 281–297.
[37] Y. Ping, Z. Li, and C. Liu, "Beyond K-Means++: Towards better cluster exploration with local geometry," *Pattern Recognition*, vol. 157, p. 110871, May 2024, doi: 10.1016/j.patcog.2024.110871.
[38] R. C. de Amorim and M. Hennig, "On k-means iterations and Gaussian clusters," *Pattern Recognition Letters*, vol. 171, pp. 44–50, Feb. 2023, doi: 10.1016/j.patrec.2023.01.005.

BIOGRAPHIES OF AUTHORS

Bowo Winarno, S.Si., M.Kom is a lecturer at the Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Sebelas Maret, Surakarta, Indonesia. He earned his Bachelor's degree in Mathematics from Universitas Sebelas Maret in 2005 and his Master's degree in Computer Science from Universitas Gadjah Mada in 2008. His teaching and research interests include data mining, spatial analysis, decision support systems, and e-learning technology. He has also been actively involved in community service projects and academic training programs aimed at improving digital literacy and education quality. He can be contacted at email: bowowinarno@staff.uns.ac.id.

Dr. Budi Warsito is a lecturer in the Department of Statistics at Diponegoro University (UNDIP) in Semarang, Indonesia, whose prolific research portfolio encompasses over 150 publications and 600+ citations as of this profile. ResearchGate His work spans diverse areas including machine learning applications in healthcare, environmental science (wastewater and air pollution), decision support systems and clustering algorithms, reflecting a strong interdisciplinary orientation at the intersection of data science, environmental engineering and systems analysis. He can be contacted at email: budiwarsito@lecturer.undip.ac.id.

Drs. Bayu Surarso, M.Sc., Ph.D. is a faculty member at the Department of Mathematics, Universitas Diponegoro (UNDIP), Indonesia. He earned his Bachelor's degree in Mathematics at UNDIP and completed his Master's and Doctoral studies at Hiroshima University. His areas of expertise include algebra, combinatorics, and mathematical logic. He can be contacted at email: bayus@lecturer.undip.ac.id.