

Text clustering on computing publication using natural logarithm-based TF-IDF in K-means

Jiel Vayad Ramadhan, Widodo, Ressay Dwitias Sari

Department Informatics and Computer Engineering Education, Universitas Negeri Jakarta, Jakarta, Indonesia

Article Info

Article history:

Received Sep 23, 2025

Revised Aug 2, 2026

Accepted Aug 28, 2026

Keywords:

Computing publication

K-means

Logarithm

Text clustering

TF-IDF

ABSTRACT

Text clustering is essential for organizing large textual datasets, yet the impact of term frequency inverse document frequency (TF-IDF) logarithmic base selection remains underexplored. This study compares natural logarithm and base-10 logarithm in TF-IDF weighting applied to K-Means and hierarchical agglomerative clustering (HAC) of scientific publications from Universitas Negeri Jakarta across Information Systems, Computer Science, and Informatics Education departments. Clustering performance was evaluated using the Davies-Bouldin index (DBI), silhouette coefficient, and Calinski-Harabasz Index across $k=2-5$. Results show that both bases produce structurally equivalent clustering outcomes, with $k=2$ consistently optimal across all departments based on DBI and Silhouette, and consistent agreement between K-Means and HAC on this configuration. Stability analysis across five independent random seeds confirms zero variance at $k=2$ for two departments and the highest Silhouette score for the third. However, natural logarithm yields higher term weight magnitudes with wider contrast between dominant terms, enhancing qualitative interpretability of cluster themes. Thus, logarithm base selection does not affect clustering structure but meaningfully influences term weight representation, with natural logarithm preferred for thematic analysis. This provides clear, empirically grounded guidance for TF-IDF-based text clustering pipelines. This approach can be further extended to hybrid weighting techniques and additional clustering algorithms.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Jiel Vayad Ramadhan

Department Informatics and Computer Engineering Education, Universitas Negeri Jakarta

Rawamangun Muka Raya Street, Jakarta, Indonesia

Email: jielvayad261102@gmail.com

1. INTRODUCTION

Text clustering plays a crucial role in text mining by helping individuals comprehend large volumes of text and serving as a useful preparatory step that structures the corpus for further analysis [1]. This technique finds wide application across multiple areas, such as document management, sentiment analysis, and natural language processing [2]. By categorizing similar texts, it enhances the precision and relevance of subsequent analyses, allowing focus on more uniform data sets with particular features or themes [3]. A significant challenge in text clustering is how to accurately convert textual data into a numerical format that effectively reflects the semantic connections between words and documents. The term frequency-inverse document frequency (TF-IDF) method is one of the most popular techniques for achieving this, as it measures the significance of a word within a document compared to a broader collection of documents.

The TF-IDF method combines two essential components: TF, which measures how frequently a word appears in a document, and IDF, which evaluates how rare or common a word is across the entire document collection [4]. The IDF component is typically calculated using a logarithmic function to mitigate the impact of highly frequent words that may not convey significant meaning, such as stop words. However, the choice of the logarithmic base whether natural logarithm ($\ln$) or base-10 logarithm ($\log_{10}$) can significantly influence the weighting of terms and, consequently, the results of clustering algorithms. One widely used algorithm for text clustering is K-Means, an unsupervised learning method known for its simplicity and efficiency [5]. However, the performance of K-Means is highly dependent on the quality of the input data representation [6].

Prior studies have explored various methodologies for topic clustering by leveraging text mining and machine learning algorithms [7]. For instance, [8] conducted a subject clustering analysis and science mapping of COVID-19 research in the Scopus database, identifying three primary clusters (health, basic science, and clinical research) using scientometric methods such as word co-occurrence and network analysis. Another study [9] proposed a hybrid feature-based approach for clustering government science and technology projects using text mining and K-means, extracting key terms such as "Artificial Intelligence," "Deep Learning," and "Innovation," and showed that combining multiple textual features with optimized cluster numbers significantly improves topic identification accuracy. While these studies confirm the effectiveness of clustering for organizing domain-specific research landscapes, none of them examine how the choice of logarithmic base in TF-IDF affects clustering outcomes a gap that is particularly relevant in computing literature, where terminological specificity and rapid technological change are prevalent.

This study addresses that gap by clustering computing-related scientific publications using the K-Means algorithm, with hierarchical agglomerative clustering (HAC) as a comparison method, applied to TF-IDF implementations involving both natural logarithm ($\ln$) and base-10 logarithm ($\log_{10}$). The results are compared using evaluation metrics such as the Davies-Bouldin index (DBI), the silhouette coefficient, and the Calinski-Harabasz Index. The findings are expected to offer a deeper understanding of how the choice of logarithmic base in TF-IDF impacts the performance of text clustering, provide recommendations on the optimal logarithmic transformation for similar applications, and serve as a reference for developing more accurate text clustering methodologies. This paper has the following contributions,

- Empirical comparison of natural logarithm and base-10 logarithm variants of TF-IDF demonstrates that logarithmic base selection has no practical impact on K-Means and HAC outcomes for academic text corpora, allowing practitioners to choose a base based on interpretability needs rather than clustering performance concerns.
- A specialized dataset of scientific publications from three computing-related academic departments is compiled and pre-processed to support clustering analysis
- Research theme representation of computing-related academic departments at Universitas Negeri Jakarta is mapped through text clustering, providing institutional insights into departmental research landscapes.
- The effect of logarithm base selection on term weight magnitude is analyzed, demonstrating that natural logarithm produces wider weight contrast between terms, which enhances qualitative interpretability of dominant research themes per cluster without altering the underlying cluster structure.
- Clustering stability and robustness are evaluated across distinct paradigms by integrating multi-seed experiments (reporting mean $\pm$ standard deviation for internal validation metrics) with cross-algorithm comparison between K-Means and HAC, providing a reproducibility-oriented framework for reliable k selection in small-corpus academic text clustering.

2. METHOD

This study compares the effectiveness of using the natural logarithm and base 10 logarithm in TF-IDF calculations on text clustering results with the K-Means algorithm, aiming to identify the most optimal logarithmic approach for improving clustering quality. To achieve this, we conducted controlled experiments on various text datasets, applied both logarithmic variants in TF-IDF, and analyzed the clustering results using the K-Means algorithm with evaluation metrics such as the DBI and silhouette score. The following steps were carried out during this research.

2.1. Data collection

The data used in this study are the full texts of scientific publications by lecturers in the field of computer science at the State University of Jakarta indexed by Scopus. The components of the publication data include lecturer names and the content of their scientific publications published up to October 1, 2024. Process stage for clustering as shown in Figure 1.

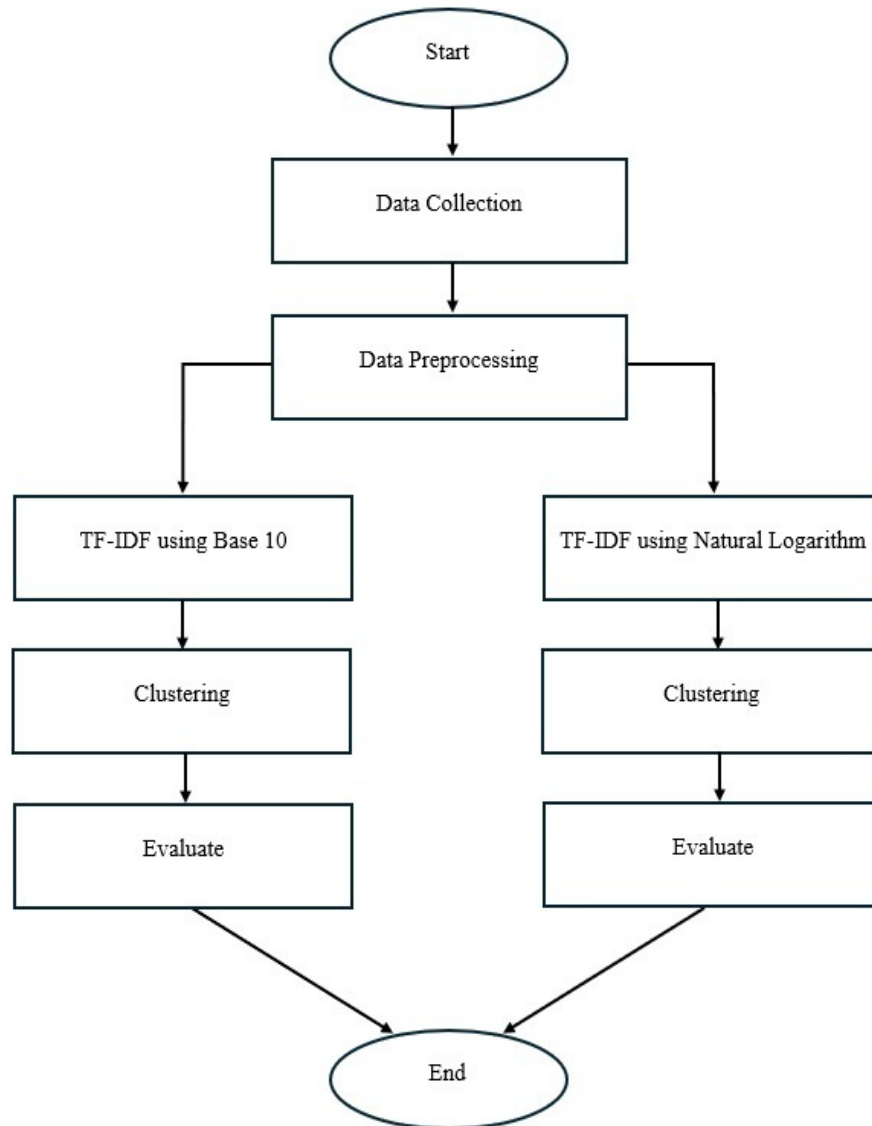


Figure 1. Process stage for clustering

2.2. Data preprocessing

The stage of data preprocessing is crucial and cannot be overlooked. Within this phase, we provide methods to remove implicit noise from text datasets [10]. Unimportant features are discarded, yet the feature spaces remain highly dimensional. Therefore, during preprocessing, applying specific thresholds helps reduce the feature space for each input document, considering the frequency of each term [11], [12]. The preprocessing workflow adopted in this study follows established text mining practices in R as documented by Silge and Robinson [13]. Data preprocessing performed on the data includes the following sequence of stages.

2.2.1. Cleansing (CL)

Cleansing (CL) involves eliminating all characters from a document, except for alphabetic letters. The purpose of this stage is to remove characters that are meaningless or unnecessary. Such characters include numbers, symbols like @ and #, website links, and emojis [14]. The CL process was implemented in R using the `str_replace_all()` function from the `stringr` package. Several cleaning operations were applied sequentially, including the removal of citation markers (e.g., [1], [2]), section headers such as "abstract", "keywords", and "references", URLs, non-alphanumeric characters, numeric digits, and possessive suffixes. The resulting text was further normalized using `str_squish()` to eliminate redundant whitespace and improve textual consistency.

2.2.2. Stopword removal

In information retrieval, stop words are commonplace terms that typically aren't factored into the indexing or searching of documents. These words, like "a," "an," "the," and "this" in English, lack significant meaning on their own and are instead assessed based on their frequency across a text collection. If a word appears with similar frequency across all texts, it's deemed "empty" and doesn't effectively differentiate texts. Generally, stop words are language-specific, encompassing prepositions, articles, and pronouns, forming what are often termed "anti-dictionaries" in search engine algorithms. Yet, in topic-specific text collections, even non-grammatical words might function as stop words if uniformly distributed, while some rare grammatical words could retain significance in certain contexts [15]. Stopword removal (SR) was performed using the `tm` package (version 0.7-11) with the standard English stopwords list obtained through `stopwords("english")`. To further reduce noise specific to scientific publications, a custom stopwords list containing 24 publication-related terms, including "table", "figure", "et", "al", "doi", "introduction", "conclusion", and "methodology", was incorporated into the filtering process.

2.2.3. Case folding

Case folding (CF) is a text preprocessing step that converts all letters to lowercase. As part of this process, the system also removes invalid characters (including numbers and punctuation marks) and eliminates uniform resource locators (URLs) to standardize the input data [16]. All text was converted to lowercase using the base R `tolower()` function prior to subsequent preprocessing stages. This transformation ensured that words differing only in letter capitalization were treated as identical terms during feature extraction.

2.2.4. Lemmatization

Lemmatization (LN) is a crucial process in the lexical processing of texts, aiming to standardize terms by returning them to their canonical form or common lexical entry, called a lemma. This treatment makes it possible to reduce the inflection of lexemes, such as verbs, nouns, and adjectives, by coding them in such a way as to refer to their canonical form recorded in the language's dictionaries. For example, the words "eat," "eats," "ate" would all be reduced to their common lemma "eat." This approach harmonizes variations in word form and thus facilitates their analysis, classification, and indexing in textual information systems. In summary, LN helps improve the consistency and precision of lexical analyses by normalizing terms according to their canonical form [17]. LN was carried out using the `textstem` package (version 0.1.4) through the `lemmatize_words()` function. This procedure reduced inflected word forms to their canonical representations; for example, "running" and "runs" were normalized to "run", while "studies" was transformed into "study". Such normalization helps improve lexical consistency across documents and reduces feature sparsity.

2.2.5. Tokenization

Tokenization (TN) involves the process of dividing words into smaller parts or tokens, often involving separating letters along with punctuation. Typically, these tokens correspond to words or elements. It's crucial to differentiate between types and tokens. In a text, a token refers to a sequence of characters that acts as a meaningful semantic unit. A type, on the other hand, is the set of all tokens that share the same sequence of letter [18]. The cleaned and lemmatized text was subsequently divided into individual tokens using the `strsplit()` function. Tokens consisting of a single character were discarded, and residual ordinal suffixes such as "st", "nd", "rd", and "th" were removed. To reduce feature dimensionality and eliminate infrequent terms, a minimum term frequency threshold of 2 was applied before constructing the TF-IDF representation.

2.3. TF-IDF

TF-IDF is an essential numerical metric used in natural language processing and information retrieval. It quantifies how important a term is within a document collection or corpus [19]. This approach is based on the idea that the importance of a term is influenced by its frequency in individual documents and how uncommon it is across the whole document collection [20]. By weighting the words in a text or set of texts, the TF-IDF value grows in proportion to how frequently a word appears in the corpus. This quantization helps create a mathematical representation of each text in the corpus [21]. A word that appears numerous times in the corpus is considered less significant compared to a word that appears only a few times; in TF-IDF, the key factor is the result of combining these two statistics [22].

TF quantifies how often a word appears in a given document, calculated using (1) [23],

$$TF(w, d) = \frac{\text{word } w \text{ frequency in document } d}{\text{total number of words in document } d} \quad (1)$$

Meanwhile, IDF measures the uniqueness of a term across a document collection; terms that frequently appear in many documents receive lower IDF values, whereas terms that occur less frequently are assigned higher IDF values. The first approach is based on the base-10 logarithm, as shown in (2) [24],

$$IDF(w, C) = \log^{10} \left(\frac{\text{total number of document in the collection } C}{\text{number of document containing word } w} \right) \quad (2)$$

where as the second approach uses the natural logarithm, as presented in (3) [13];

$$IDF(w, C) = \ln \left(\frac{\text{total number of document in the collection } C}{\text{number of document containing word } w} \right) \quad (3)$$

the TF-IDF score of a term in a document is calculated using the following (4) [25],

$$TF - IDF (w, d, C) = TF (w, d) * IDF(w, C) \quad (4)$$

2.4. K-Means

K-Means clustering is widely regarded as the most prominent and widely used algorithm among available clustering methods due to its simplicity and computational efficiency [26]. The algorithm operates by specifying k cluster centroids, where k is a predefined input parameter that must not exceed the total number of data points. Initially, K-Means randomly initializes these centroids and iteratively optimizes their positions by minimizing within-cluster variance [27]. To improve clustering stability and reduce sensitivity to centroid initialization, the k-means++ initialization strategy was employed. The algorithm was executed with nstart = 10, indicating ten independent runs with different centroid seeds, while the maximum number of iterations for each run was set to iter.max = 300. In addition, five fixed random seeds (42, 123, 456, 789, 1,011) were used across independent runs to ensure stability and reproducibility of the experimental results. Five seeds were considered sufficient for this purpose given the small corpus size per department (34–50 documents) and the internal use of nstart = 10, which already mitigates single-run initialization variance by selecting the best solution among ten restarts per seed; the outer five-seed layer therefore serves to confirm cross-run convergence stability rather than to characterize a wide variance distribution, which would otherwise require a substantially larger seed count. As a distance-based algorithm, K-Means employs Euclidean distance metrics, thus fundamentally restricting its application to spherically separable data distributions. This theoretical constraint requires that cluster elements exhibit relatively uniform and symmetric dispersion patterns around their respective centroids for effective performance [28].

2.5. Hierarchical agglomerative clustering (HAC)

HAC is a deterministic, bottom-up clustering framework that begins by treating each document as an isolated single-element cluster and iteratively merges the most proximate pairs to construct a hierarchical tree structure known as a dendrogram [29]. Because the merging sequence relies strictly on pairwise distances rather than randomized centroid initialization, the algorithm produces entirely consistent and reproducible cluster hierarchies across executions without requiring random seeds or multiple restarts. To measure cluster proximity, this study employs Ward's minimum variance criterion, which selectively combines clusters to minimize the overall increase in within-cluster variance [30]. The linkage criterion was applied using Euclidean distance calculated on TF-IDF document vectors, establishing distance metric alignment with K-Means. To partition the dataset into k distinct clusters for comparative evaluation, the resulting dendrogram was cut at the corresponding tree depth. Computationally, the clustering procedure was executed in R using the hclust() function with method = "ward.D2", ensuring exact mathematical adherence to Ward's minimum variance objective on squared Euclidean distances.

2.6. Davies-Bouldin index

The DBI serves as an effective cluster evaluation metric that simultaneously measures cluster compactness and separation [31]. In implementation, the DBI calculates the average distance between objects and their cluster centroid to assess density, while also computing distances between centroids to evaluate separation between clusters [32]. The DBI is formally defined as (5) [33],

$$DB = \frac{1}{c} \sum_{i=1, i \neq j}^c \text{Max}\{R_{ij}\} \quad (5)$$

2.7. Silhouette coefficient

The silhouette coefficient serves as a crucial clustering evaluation metric that assesses both intra-cluster cohesion and inter-cluster separation [34]. The silhouette coefficient calculation requires several

steps. First, for each object i within a cluster, we calculate the mean intra-cluster distance $a(i)$, representing the average proximity between the object and all other members of its assigned cluster. Second, for all neighboring clusters, we compute the mean inter-cluster distance, where $b(i)$ represents the minimum average distance from object i to all members of any other cluster c [35]. These measures are combined to produce the silhouette value for each data point i (6), which quantifies how well the point is grouped within its assigned cluster.

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad (6)$$

This formulation quantifies the appropriateness of a point's cluster assignment, where values approaching +1 indicate optimal placement (strong cohesion and clear separation), values near 0 suggest boundary ambiguity, and negative values imply misclassification. The silhouette coefficient's effectiveness stems from its dual consideration of both internal cluster density and external isolation [36].

2.8. Calinski-Harabasz index

To complement the structural validation provided by the DBI and silhouette coefficient, this study incorporates the Calinski–Harabasz (CH) index, also recognized as the variance ratio criterion [37]. Rather than focusing strictly on pairwise sample separations, the CH index evaluates the overall partition quality by assessing the ratio of the between-cluster sum of squares (SS_b) to the within-cluster sum of squares (SS_w). Here, SS_b quantifies the dispersion of cluster centroids relative to the global data center, whereas SS_w measures the internal compactness of observations around their designated centroids [38]. Mathematically, the index is formulated as follows (7),

$$CH = \frac{\left(\frac{SS_b}{K-1}\right)}{\left(\frac{SS_w}{N-K}\right)} = \frac{SS_b}{SS_w} \times \frac{N-K}{K-1} \quad (7)$$

where K denotes the total number of clusters and N represents the sample size. Higher CH values reflect well-separated, dense clusters, whereas lower scores point to overlapping or poorly structured boundaries [39]. Within this empirical setup, the CH index was evaluated alongside the DBI and silhouette coefficient across all candidate k values to provide multi-perspective validation. The silhouette coefficient served as the primary decision criterion due to its bounded scale and dual sensitivity to intra-cluster cohesion and inter-cluster separation. Meanwhile, DBI and the CH index acted as complementary metrics to confirm the structural stability of the chosen partition.

3. RESULT AND DISCUSSION

The dataset used in this study was collected based on lecturer names registered in the Higher Education Database (Pangkalan Data Pendidikan Tinggi, PDDikti). After collection, the dataset underwent a preprocessing stage to ensure data quality and improve the accuracy of subsequent analysis. The preprocessing was conducted using RStudio, a programming tool based on the R language, which provides a robust environment for text processing and analysis. Several preprocessing techniques were applied, as detailed in Table 1, to refine the textual data by removing noise and irrelevant information. This step is crucial for improving the overall quality of text analysis, ensuring that only meaningful and relevant information is retained for further processing. Proper preprocessing plays a vital role in sentiment analysis, as it allows models to focus on essential linguistic features, thereby enhancing their ability to accurately classify sentiment within the text. The preprocessing techniques applied in this study include CL, SR, CF, LN, and TN.

Table 1 presents the preprocessed text data prepared for feature representation. Subsequently, the dataset was processed using TF-IDF transformation - a statistical weighting technique that quantifies term relevance based on both intra-document frequency and inter-document rarity within the corpus. This mechanism automatically enhances weights for discriminative terms that frequently appear in specific documents but remain uncommon across the document collection while reducing the contribution of generic terms. The resulting weighted vector representation creates an optimized feature space for machine learning modeling by preserving semantically meaningful terms and minimizing noise interference. For the initial analysis, we employed base-10 logarithm transformation in the TF-IDF calculation, with representative results displayed in Table 2. Prior to TF-IDF vectorization, the text corpus was evaluated across departments to measure document length reduction and matrix sparsity. Preprocessing reduced average document lengths by 43.40% for information system technology (46 documents; 4,258 terms), 45.75% for informatics and

computer engineering education (50 documents; 3,799 terms), and 44.01% for computer science (34 documents; 3,306 terms). This token reduction was driven primarily by stopword filtering and minimum term-frequency thresholds. Consequently, the resulting TF-IDF matrices displayed high sparsity levels of 91.61%, 91.01%, and 91.03%, respectively. These high zero-entry proportions reflect the high vocabulary-to-document ratio typical of specialized academic text corpora.

Table 1. Result of preprocessing excerpt

CL	SR	CF	LN	TN
Privacy Preserving Data	Privacy Preserving	privacy-preserving	privacy preserve	"privacy" "preserve"
Publishing with	Data Publishing	data publishing	datum publish	"datum" "publish"
Multiple Sensitive	Multiple Sensitive	multiple sensitive	multiple sensitive	"multiple" "sensitive"
Attributes based on	Attributes based	attributes based	attribute base	"attribute" "base"
Overlapped Slicing	Overlapped Slicing	overlapped slicing	overlap slice	"overlap" "slice"
Sentiment Analysis for	Sentiment Curriculum	sentiment	Sentiment	"sentiment" "curriculum"
Curriculum of	Independent Learning	curriculum	curriculum	"independent" "learn"
Independent Learning	based Naïve Bayes	independent	independent learn	"base" "naïve" "bayes"
based on Naïve Bayes	Laplace Estimator	learning based naïve	base naïve bayes	"laplace" "estimator"
with Laplace Estimator	Curriculum	bayes laplace	laplace estimator	"curriculum"
Curriculum		estimator curriculum	curriculum	
Simple Distribution of	Simple Distribution	simple distribution	Simple distribution	"simple" "distribution"
Sensitive Values for	Sensitive Values	sensitive values	sensitive value	"sensitive" "value"
Multiple Sensitive	Multiple Sensitive	multiple sensitive	multiple sensitive	"multiple" "sensitive"
Attributes in Privacy	Attributes Privacy	attributes privacy	attribute privacy	"attribute" "privacy"
Preserving Data	Preserving Data	preserving data	preserve datum	"preserve" "datum"

Table 2. Result of TF-IDF with base-10 logarithm

Term	Doc1	Doc2	Doc3	Doc4
aim	0.0034	0.0000	0.0000	0.0000
produce	0.0030	0.0000	0.0013	0.0000
augment	0.0186	0.0000	0.0000	0.0000
reality	0.0146	0.0000	0.0000	0.0000
android	0.0186	0.0000	0.0000	0.0000
base	0.0006	0.0000	0.0001	0.0000
historical	0.0246	0.0000	0.0000	0.0000
site	0.0293	0.0000	0.0000	0.0000

For more in-depth analysis, TF-IDF weights were additionally computed using natural logarithm transformation, with complete results presented in Table 3. This secondary calculation enables direct comparison between logarithmic bases by examining their effects on term weight distributions, feature space sparsity, and cluster separation metrics.

Table 3. Result of TF-IDF with natural logarithm

Term	Doc1	Doc2	Doc3	Doc4
aim	0.0077	0.0009	0.0000	0.0000
produce	0.0068	0.0013	0.0000	0.0000
augment	0.0429	0.0000	0.0000	0.0000
reality	0.0337	0.0000	0.0000	0.0000
android	0.0429	0.0000	0.0000	0.0000
base	0.0014	0.0002	0.0000	0.0000
historical	0.0565	0.0000	0.0000	0.0000
site	0.0674	0.0000	0.0000	0.0000

Tables 2 and 3 display the term-weighted dataset generated through TF-IDF transformation, where each value quantitatively represents term significance across documents. Following feature weighting, clustering was performed using the K-Means algorithm with $nstart = 10$ and five independent random seeds (42, 123, 456, 789, 1011) to ensure initialization stability, with HAC applied as a deterministic comparison method. Cluster quality was evaluated using three validation metrics: the DBI, the silhouette coefficient, and the CH index. Because the natural logarithm ($\ln$) and base-10 logarithm ($\log_{10}$) differ only by a constant scaling factor ($\ln(x) = \log_{10}(x) \times 2.303$), both bases yield structurally identical cluster assignments and identical DBI, Silhouette, and Calinski-Harabasz scores. Therefore, Tables 4-6 present the combined evaluation results for both logarithmic bases and both clustering methods, reported as mean $\pm$ standard

deviation (SD) across five seeds for K-Means. The optimal k for each department is determined based on the highest silhouette coefficient and lowest DBI, as presented in the subsequent Tables.

Table 4. Clustering evaluation results of information system technology

Method	k	DBI Mean ($\pm$ SD)	Silhouette Mean ($\pm$ SD)	Calinski–Harabasz Mean ($\pm$ SD)
K-Means	2	0.4080930 ($\pm$ 0.0000000)	0.4688537 ($\pm$ 0.0000000)	5.1654 ($\pm$ 0.0000)
K-Means	3	0.4215313 ($\pm$ 0.0000000)	0.4535762 ($\pm$ 0.0000000)	5.1326 ($\pm$ 0.0000)
K-Means	4	0.5165466 ($\pm$ 0.1960943)	0.3891831 ($\pm$ 0.1267412)	5.1237 ($\pm$ 0.0399)
K-Means	5	0.9718514 ($\pm$ 0.1143528)	0.2064943 ($\pm$ 0.0052189)	5.1502 ($\pm$ 0.0835)
HAC	2	0.4080930	0.4688537	5.1654
HAC	3	0.4215313	0.4535762	5.1326
HAC	4	0.4288506	0.4458635	5.1415
HAC	5	0.7672909	0.1971584	5.2996

Table 5. Clustering evaluation results of computer science

Method	k	DBI Mean ($\pm$ SD)	Silhouette Mean ($\pm$ SD)	Calinski–Harabasz Mean ($\pm$ SD)
K-Means	2	0.4264496 ($\pm$ 0.0000000)	0.4490391 ($\pm$ 0.0000000)	4.4633 ($\pm$ 0.0000)
K-Means	3	0.4490735 ($\pm$ 0.0000000)	0.4245789 ($\pm$ 0.0000000)	4.3077 ($\pm$ 0.0000)
K-Means	4	0.5322483 ($\pm$ 0.0000000)	0.3160234 ($\pm$ 0.0000000)	4.4825 ($\pm$ 0.0000)
K-Means	5	0.6037305 ($\pm$ 0.2071797)	0.3508514 ($\pm$ 0.0087281)	4.4868 ($\pm$ 0.0317)
HAC	2	0.4264496	0.4490391	4.4633
HAC	3	0.4490735	0.4245789	4.3077
HAC	4	0.5322483	0.3160234	4.4825
HAC	5	0.5110769	0.3547547	4.5010

Table 6. Clustering evaluation results of informatics and computer engineering education

Method	k	DBI Mean ($\pm$ SD)	Silhouette Mean ($\pm$ SD)	Calinski–Harabasz Mean ($\pm$ SD)
K-Means	2	0.9752791 ($\pm$ 0.7109114)	0.3644531 ($\pm$ 0.0748777)	4.0564 ($\pm$ 0.2603)
K-Means	3	0.9606198 ($\pm$ 0.4554485)	0.2127458 ($\pm$ 0.1871338)	4.2637 ($\pm$ 0.0275)
K-Means	4	1.1018889 ($\pm$ 0.0293502)	0.1180301 ($\pm$ 0.0090615)	4.4331 ($\pm$ 0.0394)
K-Means	5	1.0272017 ($\pm$ 0.1244644)	0.1560625 ($\pm$ 0.0021119)	4.6878 ($\pm$ 0.0977)
HAC	2	0.4561028	0.4191360	4.2465
HAC	3	0.4617010	0.4177406	4.2938
HAC	4	1.0697374	0.1081037	4.4763
HAC	5	0.9715395	0.1570070	4.7315

Tables 4-6 present clustering evaluation results across all three departments for both K-Means and HAC. For information system technology and computer science, DBI and Silhouette values are perfectly stable at k = 2 and k = 3 (SD = 0.000), confirming that the optimal cluster structure is consistently reproduced regardless of initialization. At k = 4 and k = 5, standard deviations increase substantially, indicating sensitivity to initialization at higher cluster counts. For informatics and computer engineering education, elevated SD at k = 2 (DBI SD = 0.711) reflects greater textual heterogeneity in that corpus, yet the silhouette coefficient remains highest at k = 2 (0.364) across all k values, supporting it as the optimal configuration. In contrast, the CH index increases with k in most cases, typically peaking at k = 5 for both K-Means and HAC across computer science and informatics and computer engineering education, with the exception of K-Means in information system technology, where k = 2 narrowly remains highest (5.1654 vs. 5.1502 at k = 5).

This divergence from DBI and Silhouette is consistent with the CH index's known tendency to favor larger cluster counts on sparse, high-dimensional TF-IDF representations, where between-cluster dispersion continues to grow with k while within-cluster cohesion offers diminishing returns. Given this limitation, k selection in this study is based primarily on DBI and Silhouette, with the CH index reported as a supplementary metric rather than a decision criterion. HAC results for DBI and Silhouette closely mirror the K-Means findings at k = 2 and k = 3 across departments, reinforcing the robustness of the selected cluster structure independent of the clustering algorithm used. The consistent degradation of DBI and Silhouette at k $\geq$ 4, combined with the relatively small corpus sizes of each department, empirically supports the selection of k = 2 as optimal for all three departments. Qualitative validation of the cluster structures is provided through top-term analysis in Tables 7-9. Because K-Means and HAC yielded closely comparable cluster structures at k = 2 across all three departments, top-term results for both methods are presented as a single combined ranking per department rather than reported separately.

Table 7. Top 10 TF-IDF terms per cluster of information system technology (k = 2, Optimal)

Rank	Cluster 1 Term	Ln TF-IDF	Log10 TF-IDF	Cluster 2 Term	Ln TF-IDF	Log10 TF-IDF
1	node	0.0109812	0.0047691	rubber	0.1852568	0.0804560
2	sensitive	0.0102671	0.0044589	tyre	0.1852568	0.0804560
3	bus	0.0073479	0.0031911	stacker	0.1852568	0.0804560
4	packet	0.0071060	0.0030861	loader	0.1852568	0.0804560
5	student	0.0070553	0.0030641	gantry	0.1517175	0.0658901
6	anatomy	0.0069124	0.0030020	delivery	0.1128516	0.0490108
7	attribute	0.0062328	0.0027069	smoothness	0.1011450	0.0439267
8	image	0.0060015	0.0026064	receive	0.0866926	0.0376501
9	shoot	0.0060009	0.0026061	side	0.0789395	0.0342830
10	cluster	0.0056545	0.0024557	variable	0.0722962	0.0313978

Table 8. Top 10 TF-IDF terms per cluster of computer science (k = 2, Optimal)

Rank	Cluster 1 Term	Ln TF-IDF	Log10 TF-IDF	Cluster 2 Term	Ln TF-IDF	Log10 TF-IDF
1	owner	0.0084646	0.0036761	namber	0.2635358	0.1144522
2	happiness	0.0081503	0.0035396	fruit	0.2246329	0.0975568
3	learner	0.0079125	0.0034363	leave	0.1844751	0.0801165
4	alumnus	0.0075430	0.0032759	flower	0.0928650	0.0403308
5	bus	0.0073375	0.0031867	armpit	0.0903551	0.0392407
6	contour	0.0068684	0.0029829	weight	0.0753667	0.0327313
7	learn	0.0063264	0.0027475	height	0.0736031	0.0319654
8	business	0.0062439	0.0027117	plant	0.0722236	0.0313663
9	msme	0.0060461	0.0026258	hardness	0.0652565	0.0283405
10	assurance	0.0056856	0.0024692	lenght	0.0614917	0.0267055

Table 9. Top 10 TF-IDF terms per cluster of informatics and computer engineering education (k = 2, Optimal)

Rank	Cluster 1 Term	Ln TF-IDF	Log10 TF-IDF	Cluster 2 Term	Ln TF-IDF	Log10 TF-IDF
1	sensitive	0.0126942	0.0055130	rca	0.2194244	0.0952948
2	report	0.0100155	0.0043497	rsa	0.2068858	0.0898494
3	attribute	0.0085790	0.0037258	key	0.1456479	0.0632541
4	character	0.0074759	0.0032467	encryption	0.0940390	0.0408406
5	weapon	0.0069331	0.0030110	symmetric	0.0877697	0.0381179
6	biographical	0.0065286	0.0028353	algorithm	0.0868250	0.0377076
7	student	0.0061286	0.0026616	encrypt	0.0815005	0.0353952
8	ontology	0.0059686	0.0025921	hybrid	0.0809528	0.0351574
9	engine	0.0057905	0.0025148	network	0.0053784	0.0023358
10	document	0.0053549	0.0023256	decryption	0.0689619	0.0299498

Tables 7-9 present the top 10 TF-IDF terms for each cluster at the optimal $k = 2$ configuration, shown for both logarithmic bases. The term rankings are identical across $\ln$ and $\log_{10}$, as both bases produce the same cluster assignments; the quantitative difference in TF-IDF values ($\ln$ values are approximately $2.303 \times$ larger) does not alter the relative ranking or thematic interpretation of terms.

The experimental results confirm that base-10 and natural logarithm implementations of TF-IDF produce structurally equivalent clustering outcomes across all three departments at $k = 2$. The practical difference between the two bases is therefore not in cluster structure, which is identical, but in the interpretive value of term weights for qualitative analysis. In the natural logarithm representation, the top-weighted term in a cluster centroid may have a value of 0.185, whereas its base-10 counterpart yields 0.080. The wider numerical spread of natural logarithm weights (by a factor of $\ln(10) \approx 2.303$) makes thematic contrasts more visually distinct when examining top terms per cluster, as shown in Tables 7-9. This practical advantage in human-interpretable theme extraction constitutes the key differentiating factor between the two logarithmic bases in this study's context.

The top terms presented in Tables 7-9 provide qualitative validation of the clustering results. For information system technology (Table 7), Cluster 1 spans a broad mix of networking and sensor-related terms ("node", "packet", "bus") alongside data-sensitivity and imaging terms ("sensitive", "anatomy", "image"), reflecting the department's heterogeneous computing research portfolio, while Cluster 2 consists of a single outlier document concentrated on port-handling equipment terms ("rubber", "tyre", "stacker", "gantry"). For Computer Science (Table 8), Cluster 1 reflects business- and education-oriented data mining themes ("owner", "happiness", "learner", "alumnus", "msme"), while Cluster 2 is a single outlier document centered on plant and fruit measurement terms ("namber", "fruit", "leave", "flower", "hardness"). For informatics and computer engineering education (Table 9), Cluster 1 centers on data-sensitivity, ontology, and software-related themes ("sensitive", "report", "attribute", "ontology", "engine"), while Cluster 2 is a single

outlier document concentrated on cryptography terms ("rca", "rsa", "key", "encryption", "hybrid"). Across all three departments, the singleton second cluster consistently isolates one thematically distinct document from the main corpus, a pattern expected given the small dataset sizes; this suggests that the clusters are semantically coherent and not arbitrary groupings, serving as preliminary qualitative validation of the internal evaluation metrics in the absence of external ground-truth labels.

The finding that logarithmic base selection does not alter cluster structure extends prior TF-IDF-based clustering literature, which has largely treated the choice between $\ln$ and $\log_{10}$ as an implementation detail rather than an empirical question. Studies such as [8] and [9] demonstrated the effectiveness of K-Means-based text clustering in biomedical and governmental domains but did not examine whether logarithmic base selection influences their reported cluster structures. The present results suggest that their choice of logarithmic base likely did not materially affect their clustering outcomes either, reinforcing the generalizability of this study's conclusion beyond the computing publication domain. At the same time, this study's identification of natural logarithm as preferable for qualitative term-weight interpretation despite producing identical cluster assignments adds a practical dimension absent from prior work: the logarithmic base choice matters not for clustering accuracy, but for how clearly analysts can read dominant themes from top-weighted terms, a consideration particularly relevant for small, heterogeneous corpora such as departmental research portfolios where thematic interpretability is the primary analytical goal.

These findings should nonetheless be interpreted in light of several limitations. The dataset is drawn from a single university (Universitas Negeri Jakarta), which may limit the generalizability of the findings to computing publications from other institutions with different research emphases or publication practices. Cluster validity was also assessed using internal validation metrics (DBI, silhouette coefficient, and CH index) in the absence of external ground-truth labels, meaning that the qualitative top-term analysis in Tables 7-9 serves as the primary form of validation beyond these internal metrics; consequently, the reported optimal k values reflect statistical cluster compactness and separation rather than a verified alignment with human-annotated topic categories.

4. CONCLUSION

This study investigated whether the choice of logarithmic base in TF-IDF weighting often treated as a minor implementation detail in prior text clustering literature significantly impacts clustering outcomes. The empirical findings confirm that this parameter is structurally inconsequential. Across all three computing-related departments at Universitas Negeri Jakarta and for k values ranging from 2 to 5, natural logarithm and base-10 formulations yielded structurally equivalent clustering outcomes under K-Means, closely mirrored by HAC. Furthermore, $k = 2$ emerged as the optimal configuration across all departments, with zero variance across five independent random seeds for two of the three departments and consistently high stability for the third. Aligning empirical evidence with our initial premise, these results directly fulfill Contributions 1 and 5, demonstrating that logarithmic base selection can be safely decided on interpretability grounds rather than clustering-performance concerns within a reproducible, cross-algorithm validation framework.

The structural equivalence of the two logarithm bases is mathematically guaranteed for IDF-only log transformations, as the constant multiplicative factor between $\ln$ and $\log_{10}$ cancels in ratio-based metrics. The practical distinguishing factor is term weight magnitude: natural logarithm produces weights approximately 2.303 times larger than base-10, generating wider numerical contrast between dominant and peripheral terms in cluster centroids. This enhanced contrast improves the qualitative interpretability of top terms per cluster, making natural logarithm preferable for institutional research theme mapping and similar human-interpretable analytical tasks. Qualitative top-term analysis per cluster confirms that the resulting groupings are semantically coherent, representing identifiable research themes within each department.

For future work, applying this methodology to larger, multi-institutional corpora with expert-annotated ground-truth labels would enable external validation metrics such as ARI or V-measure to complement the internal metrics used here, addressing the reliance on internal validation and single-university scope noted as limitations of the present study. Extending the corpus to multilingual publication datasets would further test whether the observed structural equivalence between logarithmic bases generalizes beyond English-dominant academic text, particularly for institutions publishing in mixed-language venues. Beyond scaling the dataset, future work could also explore transformer-based embeddings (e.g., BERT or SciBERT) as an alternative to TF-IDF term weighting, given their capacity to capture contextual and semantic relationships that frequency-based representations inherently cannot. Relatedly, topic modeling approaches such as BERTopic, which combine transformer embeddings with density-based clustering, offer a promising direction for comparison against the centroid- and linkage-based

methods (K-Means and HAC) used in this study, potentially yielding richer and more interpretable thematic structures for large-scale academic publication corpora.

ACKNOWLEDGMENTS

This research was supported by Doctoral Research Grant, Universitas Negeri Jakarta under the Research Agreement No. 38/DOKTOR/LPPM/III/2026. I gratefully acknowledge the institution’s guidance and support throughout the completion of this study.

FUNDING INFORMATION

The authors state no funding is involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Jiel Vayad Ramadhan	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
Widodo	✓	✓				✓			✓	✓		✓		✓
Ressy Dwitias Sari				✓		✓	✓			✓		✓	✓	

- C : Conceptualization
- M : Methodology
- So : Software
- Va : Validation
- Fo : Formal analysis
- I : Investigation
- R : Resources
- D : Data Curation
- O : Writing - Original Draft
- E : Writing - Review & Editing
- Vi : Visualization
- Su : Supervision
- P : Project administration
- Fu : Funding acquisition

CONFLICT OF INTEREST STATEMENT

The Authors state no conflict of interest.

DATA AVAILABILITY

The publication metadata used in this study were retrieved from the Scopus database for three computing-related study programs at Universitas Negeri Jakarta. Derived data (the compiled TF-IDF document-term matrices and clustering results) supporting the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] Y. Rong and Y. Liu, “Staged text clustering algorithm based on K-means and hierarchical agglomeration clustering,” *Proceedings of 2020 IEEE International Conference on Artificial Intelligence and Computer Applications, ICAICA 2020*, pp. 124–127, 2020, doi: 10.1109/ICAICA50127.2020.9182394.

[2] B. Kumar, U. K. Tiwari, D. C. Dobhal, and H. S. Negi, “User story clustering using k-means algorithm in agile requirement engineering,” *Proceedings of International Conference on Computational Intelligence and Sustainable Engineering Solution, CISES 2022*, pp. 1–5, 2022, doi: 10.1109/CISES54857.2022.9844390.

[3] A. Petukhova, J. P. Matos-Carvalho, and N. Fachada, “Text clustering with large language model embeddings,” *International Journal of Cognitive Computing in Engineering*, vol. 6, pp. 100–108, 2025, doi: 10.1016/j.ijcce.2024.11.004.

[4] R. Dounia, K. Yasssine, and E. N. Noureddine, “Exploring text representation impact on K-means based arabic text documents clustering,” *2022 International Conference on Intelligent Systems and Computer Vision, ISCV 2022*, 2022, doi: 10.1109/ISCV54655.2022.9806067.

[5] A. K. Abasi, A. T. Khader, M. A. Al-Betar, S. Naim, Z. A. A. Alyasser, and S. N. Makhadmeh, “A novel hybrid multi-verse optimizer with K-means for text documents clustering,” *Neural Computing and Applications*, vol. 32, no. 23, pp. 17703–17729, 2020, doi: 10.1007/s00521-020-04945-0.

[6] M. D. R Wahyudi, “Evaluation of TF-IDF algorithm weighting scheme in the qur’an translation clustering with k-means algorithm,” *Journal of Information Technology and Computer Science*, vol. 6, no. 2, pp. 117–129, 2021, doi: 10.25126/jitecs.202162295.

[7] J. Yun, S. Ahn, and J. Y. Lee, “Return to basics: clustering of scientific literature using structural information,” *Journal of Informetrics*, vol. 14, no. 4, 2020, doi: 10.1016/j.joi.2020.101099.

[8] M. M. Amiri, T. Nasiri, and P. Mehdizadeh, “Subjects clustering analysis and science mapping on COVID-19 researches in scopus database,” *Journal of Military Medicine*, vol. 22, no. 6, pp. 663–669, 2020, doi: 10.30491/JMM.22.6.663.

- [9] C. Y. Chuang, C. H. Lin, R. Q. Lin, and C. Y. Fan, "The generation of hybrid feature-based research topics using text mining and k-means algorithms for government science and technology projects," *PICMET 2024 - 2024 Portland International Conference on Management of Engineering and Technology: Technology Management in the Artificial Intelligence Era, Proceedings*, 2024, doi: 10.23919/PICMET64035.2024.10652997.
- [10] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: a survey," *Information (Switzerland)*, vol. 10, no. 4, 2019, doi: 10.3390/info10040150.
- [11] R. Kumbhar, S. Mhamane, H. Patil, S. Patil, and S. Kale, "Text document clustering using k-means algorithm with dimension reduction techniques," pp. 1222–1228, 2020, doi: 10.1109/icces48766.2020.9137928.
- [12] M. H. Ahmed, S. Tiun, N. Omar, and N. S. Sani, "Short text clustering algorithms, application and challenges: a survey," *Applied Sciences (Switzerland)*, vol. 13, no. 1, 2023, doi: 10.3390/app13010342.
- [13] J. Silge and D. Robinson, "Text mining with R: a tidy approach," *O'Reilly Media*, 2017.
- [14] R. Wirawan, E. Krisnanik, and A. Arista, "Text mining for news forecasting on the turnback hoax website," *International Journal on Informatics Visualization*, vol. 8, no. 1, pp. 96–106, 2024, doi: 10.62527/joiv.8.1.1939.
- [15] O. Chabih, S. Sbai, M. R. C. Louhdi, and H. Behja, "Unraveling the parameters of k-means clustering for enhanced textual analysis," 2024, doi: 10.1109/UNet62310.2024.10794714.
- [16] J. Rejito, A. Athariq, and A. S. Abdullah, "Application of text mining employing k-means algorithms for clustering tweets of Tokopedia," in *Journal of Physics: Conference Series*, 2021, vol. 1722, no. 1, doi: 10.1088/1742-6596/1722/1/012019.
- [17] S. Dhanalakshmi, S. Sathiyabama, and D. Ayyamuthukumar, "A novel text clustering approach based on bacterial colony optimization," in *7th International Conference on Electronics, Communication and Aerospace Technology, ICECA 2023 - Proceedings*, 2023, pp. 1847–1853, doi: 10.1109/ICECA58529.2023.10395857.
- [18] R. K. Ibrahim, S. R. M. Zeebaree, K. Jacksi, M. A. M. Sadeeq, H. M. Shukur, and A. Alkhayyat, "Design a clustering document based semantic similarity system using TFIDF and k-mean," in *4th International Iraqi Conference on Engineering Technology and Their Applications, IICETA 2021*, 2021, pp. 87–93, doi: 10.1109/IICETA51758.2021.9717942.
- [19] S. Kulshrestha and D. Santani, "Leveraging TF-IDF matrix for document clustering with k-means algorithm," *International Journal of Scientific Research and Modern Technology (IJSRMT)*, vol. 3, no. 10, pp. 39–44, 2024, doi: 10.38124/ijrmt.v3i10.61.
- [20] D. F. Surianto and D. F. Surianto, "Enhancing k-means clustering for journal articles using TF-IDF and LDA feature extraction," *Brilliance: Research of Artificial Intelligence*, vol. 4, no. 2, pp. 964–972, 2025, doi: 10.47709/brilliance.v4i2.5547.
- [21] R. G. Garcia, B. B. An, D. V. Nô, C. Zepeda, and R. Martínez, "Comparison of clustering algorithms in text clustering tasks," *Computacion y Sistemas*, vol. 24, no. 2, pp. 429–437, 2020, doi: 10.13053/CyS-24-2-3369.
- [22] J. Zhou, Z. Ye, S. Zhang, Z. Geng, N. Han, and T. Yang, "Investigating response behavior through TF-IDF and Word2vec text analysis: A case study of PISA 2012 problem-solving process data," *Heliyon*, vol. 10, no. 16, 2024, doi: 10.1016/j.heliyon.2024.e35945.
- [23] E. Hassan *et al.*, "A hybrid k-means++ and particle swarm optimization approach for enhanced document clustering," *IEEE Access*, vol. 13, pp. 48818–48840, 2025, doi: 10.1109/ACCESS.2025.3535226.
- [24] Y. D. Pramudita, I. O. Suzanti, Saifuddin, M. Syarief, and F. Solihin, "Automatic text summarization of madura tourism articles using TF-IDF and K-medoid clustering," *Proceeding - IEEE 8th Information Technology International Seminar, ITIS 2022*, pp. 168–172, 2022, doi: 10.1109/ITIS57155.2022.10009983.
- [25] A. E. Widjaja, "Text mining application with k-means clustering to identify sentiments and popular topics: a case study of the three largest online marketplaces in Indonesia," *Journal of Applied Data Sciences*, vol. 4, no. 4, pp. 441–453, Dec. 2023, doi: 10.47738/jads.v4i4.134.
- [26] J. Duo, P. Zhang, and L. Hao, "A k-means text clustering algorithm based on subject feature vector," *Journal of Web Engineering*, vol. 20, no. 6, pp. 1935–1946, 2021, doi: 10.13052/jwe1540-9589.20612.
- [27] A. Pugazhenthil and L. S. Kumar, "Selection of optimal number of clusters and centroids for k-means and fuzzy c-means clustering: a review," 2020, doi: 10.1109/ICCCS49678.2020.9276978.
- [28] S. Al Hasan, W. Ruiqin, and M. G. Hussain, "Clustering analysis of bangla news articles with TF-IDF & CV using mini-batch k-means and k-means," in *Proceedings - 2022 IEEE International Conference on Cybernetics and Computational Intelligence, CyberneticsCom 2022*, 2022, pp. 17–22, doi: 10.1109/CyberneticsCom55287.2022.9865339.
- [29] F. Jánhez-Martino, R. Alaiz-Rodríguez, V. González-Castro, E. Fidalgo, and E. Alegre, "Classifying spam emails using agglomerative hierarchical clustering and a topic-based approach," *Applied Soft Computing*, vol. 139, 2023, doi: 10.1016/j.asoc.2023.110226.
- [30] W. Li, Y. Liu, K. Deng, and X. Wu, "POS-HC: a part-of-speech hierarchical clustering approach for normative texts partition," Feb. 28, 2024, doi: 10.20944/preprints202402.1575.v1.
- [31] D. A. Tarigan, "Optimization of the k-means clustering algorithm using Davies Bouldin index in iris data classification," *Media Online*, vol. 4, no. 1, pp. 545–552, 2023.
- [32] N. E. H. Castañeda, R. A. G. Hernández, Y. Ledeneva, and A. N. H. Castañeda, "Evolutionary automatic text summarization using cluster validation indexes," *Computacion y Sistemas*, vol. 24, no. 2, pp. 583–595, 2020, doi: 10.13053/CyS-24-2-3392.
- [33] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, 1979, doi: 10.1109/TPAMI.1979.4766909.
- [34] H. B. Tambunan, D. H. Barus, J. Hartono, A. S. Alam, D. A. Nugraha, and H. H. H. Usman, "Electrical peak load clustering analysis using K-means algorithm and silhouette coefficient," in *Proceeding - 2nd International Conference on Technology and Policy in Electric Power and Energy, ICT-PEP 2020*, 2020, pp. 258–262, doi: 10.1109/ICT-PEP50916.2020.9249773.
- [35] O. Chabih, S. Sbai, M. R. C. Louhdi, and H. Behja, "Optimizing text clustering: a methodological approach for determining the optimal number of clusters," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 13, no. 3, pp. 103–111, Jun. 2024, doi: 10.30534/ijatcse/2024/021332024.
- [36] M. Jamalain and H. Vahdat-Nejad, "Effective clustering of big text data: empirical evaluation and comparative analysis," *Egyptian Informatics Journal*, vol. 32, 2025, doi: 10.1016/j.eij.2025.100812.
- [37] C. Drago, "Text mining in economics and health economics using stata," 2024.
- [38] L. E. Aik, T. W. Choon, and M. S. Abu, "K-means algorithm based on flower pollination algorithm and Calinski-Harabasz index," in *Journal of Physics: Conference Series*, 2023, vol. 2643, no. 1, doi: 10.1088/1742-6596/2643/1/012019.
- [39] M. Kossakov, A. Mukasheva, G. Balbayev, S. Seidazimov, D. Mukammejanova, and M. Sydybayeva, "Quantitative comparison of machine learning clustering methods for tuberculosis data analysis," in *Engineering Proceedings*, 2024, vol. 60, no. 1, doi: 10.3390/engproc2024060020.

BIOGRAPHIES OF AUTHORS

Jiel Vayad Ramadhan     received his first degree in Informatics and Computer Engineering Education from Universitas Negeri Jakarta, Indonesia, in 2025. He graduated as one of the top students in his program and is currently working as a Python tutor at Algorithmics. His main research interests include machine learning, clustering, computer vision, data mining, and artificial intelligence. He can be contacted at email: jielvayad261102@gmail.com.



Widodo     is an Assistant Professor at the Department of Informatics Education, Universitas Negeri Jakarta. He received his Ph.D. in Computer Science from the University of Indonesia in 2020, his M.Sc. in Computer Science from the University of Indonesia in 2004, and his B.Sc. in Information Systems from Gunadarma University in 1999. His research interests include Privacy Preserving Data Publishing, Natural Language Processing, Classification, and Clustering. He can be contacted at email: widodo@unj.ac.id.



Ressy Dwitias Sari     received her first degree in Informatics from Institut Teknologi Telkom, Indonesia, in 2011, and her master's degree in Information Technology from Universitas Indonesia, Indonesia, in 2015. She is currently a Lecturer at Universitas Negeri Jakarta, Indonesia. Her main research interests include information technology, artificial intelligence, computer systems, databases, and technology-enhanced learning. She can be contacted at email: ressydwitiassari@unj.ac.id.