Computer Science and Information Technologies
Vol. 6, No. 2, July 2025, pp. 147~158
ISSN: 2722-3221, DOI: 10.11591/csit.v6i2.p147-158

Classification and similarity detection of Indonesian scientific journal articles

Nyimas Sabilina Cahyani[1], Deris Stiawan[2], Abdiansah Abdiansah[1], Nurul Afifah[3], Dendi Renaldo Permana[1]
[1]Department of Computer Science, University of Sriwijaya, Palembang, Indonesia
[2]Department of Computer Engineering, University of Sriwijaya, Palembang, Indonesia
[3]Department of Informatics Engineering, University of Sriwijaya, Palembang, Indonesia

Article Info

Article history:

Received Feb 20, 2025
Revised Mar 24, 2025
Accepted May 23, 2025

Keywords:

Classification
Cosine similarity
GARUDA
Naïve Bayes
Similarity

ABSTRACT

The development of technology is accelerating in finding references to scientific articles or journals related to research topics. One of the sources of national aggregator services to find references is Garba Rujukan Digital (GARUDA), developed by the Ministry of Education, Culture, Research, and Technology (Kemendikbudristek) of the Republic of Indonesia. The naïve Bayes method classifies articles into several categories based on titles and abstracts. The system achieves an F1-score of 98%, which indicates high classification accuracy, and the classification process takes less than 60 minutes. Article similarity detection is done using the cosine similarity method, and a similarity score of 0.071 reflects the degree of similarity between the title and the abstract that has been concatenated, while a score close to 1 indicates a higher similarity. Searching for similar scientific articles based on title and abstract, sort articles based on the results of the highest similarity score are the most similar articles, and generating article categories. The results of the research show that the proposed method significantly improves the classification and search processes in GARUDA, as well as accurate and efficient similarity detection.

Corresponding Author:

Deris Stiawan
Department of Computer Engineering, University of Sriwijaya
Indralaya, Ogan Ilir 30662, Palembang, Indonesia
Email: deris@unsri.ac.id

1. INTRODUCTION

The development of technology is accelerating in aggregator services for scientific journal searches as a reference or bibliography in determining article writing topics. The aggregator service is a platform that collects and compiles information from various sources to provide easier and more organized access for its users, one of which is the national aggregator Garba Rujukan Digital (GARUDA) developed by the Ministry of Education, Culture, Research, and Technology (Kemendikbudristek) of the Republic of Indonesia. Also, it has a database and network connected to SINTA, Bima, Arjuna, PDDIKTI, Risbang, Scopus, and Rama. Research from all lecturers in Indonesia is collected and entered into the science and technology index (SINTA) portal used to measure and monitor the performance of scientific research conducted by researchers [1].

Naive Bayes is an algorithm used for classification based on Bayes theorem [2]. Classification is a technique to group data sets into multi-classes to obtain correct prediction and analysis results [3]. Classification with the support vector machine Linear Kernel method using 205 features obtained a poor accuracy rate of 58.3% [4]. Classification based on journal abstracts using the naive Bayes and naive Bayes

multinomial methods provides more accurate classification performance with faster and more efficient computing time compared to other classification methods [5], [6].

Natural language processing (NLP) to understand human language effectively and provide techniques for managing the classification of large amounts of text or documents with accurate results [7], [8]. Text classification using NLP can automatically learn from text, so that it can provide information from the text in an organized manner with the correct classification [9], [10]. NLP combines computer science, linguistics, and mathematics [11], NLP can be used for plagiarism detection [12]. This research uses text processing to classify articles and detect similarities based on titles and abstracts into specific categories. Text is converted to numerical form using the term frequency-inverse document frequency (TF-IDF) technique [13].

Cosine similarity in measuring the degree of similarity of scientific article citations with reference sources gives quite good results of 70%, and citations with no relationship with reference sources by 30% [14]. Cosine similarity to detect plagiarism in Bengali text content successfully determines similarity by comparing vectors in numerical values [15]. Measurement using cosine similarity by measuring two vectors and calculating the size of the cosine angle between them [16]. Matching scientific article titles using the cosine similarity and Jaccard similarity methods provides better cosine similarity performance results than Jaccard similarity in specific scenarios [17]. Cosine similarity is used for a book recommendation system to provide results relevant to the course topic with a precision of 0.7 and a recall of 0.73 [18]. Comparing three different methods, namely cosine similarity, Jaccard similarity, and Euclidean distance, to measure the similarity of two news articles in Hindi and English based on topic. The most accurate cosine similarity results compared to the other two methods, with an accuracy of 81.25%, recall of 100%, and F-measure of 76.92% [19].

The mapping related to the previous research based on the results of the literature research collection is illustrated in Figure 1 VOS Viewer naive Bayes and Figure 2 VOS Viewer cosine similarity. Figure 1 is the VOS Viewer for methods related to naive Bayes and previous research. Figure 2 is the VOS Viewer for methods related to cosine similarity and previous research. VOS Viewer is used for mapping the relationships between keywords, previous authors, and journals used in research. Researchers use it to identify emerging fields and look at interrelated scientific concepts. This research contributes to evaluating the classification of published articles in GARUDA by applying the naive Bayes method, and detecting similarities using the cosine similarity approach. The classification method using naive Bayes and the similarity detection method using cosine similarity, in this research produce an accurate and good model in classifying articles and measuring article similarity. This research aims to determine the performance of classifiers using the naïve Bayes method and to determine the performance of similarity detection using the cosine similarity method. The discussion in this research is as follows: section 2 discusses the methods used in determining references and datasets. Section 3 provides the results of classification, similarity detection, and search of articles. Section 4 provides conclusions and suggestions for further research.

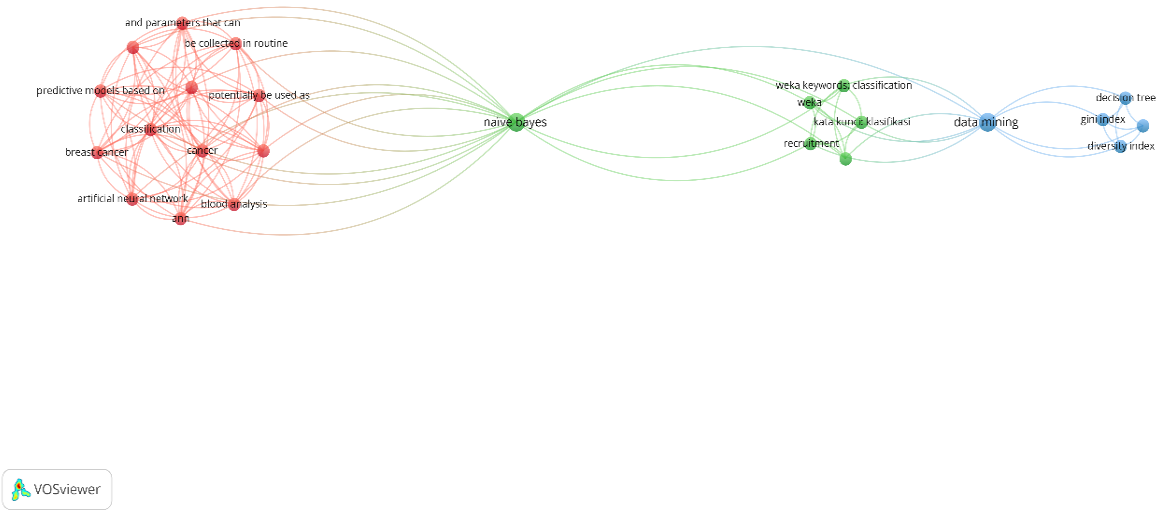


Figure 1. VOS Viewer for naïve Bayes

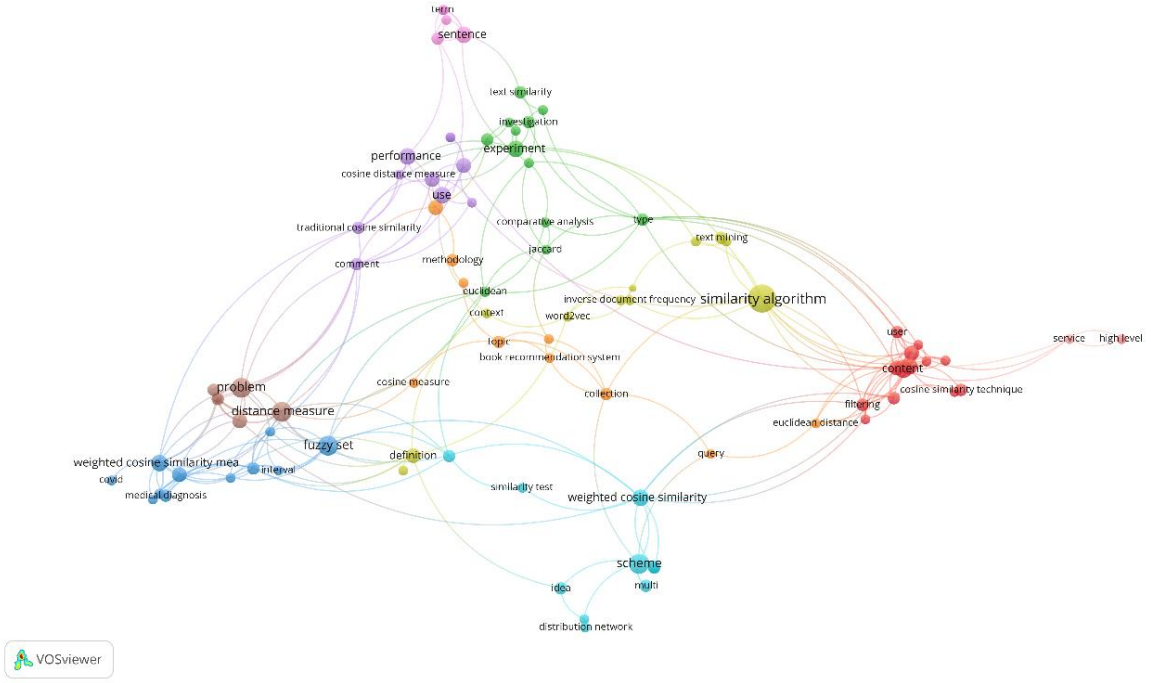


Figure 2. VOS Viewer for Cosine Similarity

2. METHOD

2.1. Research framework

This research requires a framework, research flow, and several stages. The theory used by previous research, the appropriate methodology to solve the main problem using the naive Bayes method in classification, the cosine similarity method in similarity detection, and the cosine similarity method in article search, the research framework is drawn in Figure 3. The research framework in Figure 3 describes the process, starting with data collection, data engineering, labeling using rule-based auto labeling, classification using naive Bayes, similarity detection using cosine similarity, and article search using cosine similarity. Data engineering is carried out to clean up duplicate data and consists of several stages: tokenization, concatenation, balancing, flattening data, and splitting. Rule-based auto-labeling is done by matching keywords in the text with predetermined categories. Classification was carried out to determine the category of text based on the probability of words, if the classification results are incorrect, go back to the data engineering process. Meanwhile, the correct classification results will proceed to the similarity detection process. Similarity detection and scientific article searches were carried out to analyze the results of cosine similarity to assess the similarity of articles based on the similarity score. The classification method using naive Bayes and the similarity detection method using Cosine Similarity produce an accurate and good model in classifying articles and measuring article similarity.

2.1.1. Data collection

The dataset collection technique obtained from GARUDA in the form of Excel is secondary data. There are 35,908 rows and 13 columns of Author ID, GARUDA_ID, OJS_IDENTIFIER, and GARUDA_DOI. AKREDITASI, GARUDA_TITLE, GARUDA_ABSTRACT, GARUDA JOURNAL, GARUDA_YEAR_PUBLISH, GARUDA_DATE_PUBLISH, GARUDA_CITE, GARUDA_URL, ORIGINAL_URL. This research used the title and abstract to classify, detect similarities and search for scientific articles. The data was processed by deleting duplicate data into 29,239 rows and 13 columns. Good classification results using naive Bayes, article similarity detection using cosine similarity, article search using cosine similarity, and research data are obtained and depicted in Figure 4.

2.1.2. Data engineering

Data engineering is collecting and preparing datasets from the Ministry of Education and Culture's GARUDA, cleaning duplicate data, and labeling. The stages carried out in data engineering are illustrated in Figure 5. The stages of data engineering are shown in Figure 5. The first stage is tokenizing, the second stage is concat the data to combine the text data between the title and abstract columns to generate a new column

with the name CONCAT_DATA. The third stage is balancing data to balance data between minority and majority classes. We already tried to use an unbalanced dataset, but we got unsatisfactory results and negatively impacted the performance of machine learning algorithms [20], [21]. The fourth stage flattens the data to change the structure of a multi-dimensional array into a one-dimensional array. The last stage splits the data to separate the dataset into two subsets: training and test data. In this study, a random split method is used by sampling data to ensure that refraction against different data characteristics does not affect the data modeling process. After performing the data engineering stage, labeling is depicted in Figure 6.

The category labeling in Figure 6 describes the results of the auto label use rule-based auto. There is a new column with the name of the category. The categories that have been determined in this study are nine categories, namely other, management information systems, decision support systems, sales information systems, customer relationship management, marketing information systems, financial information systems, executive information systems, and human resources information systems.

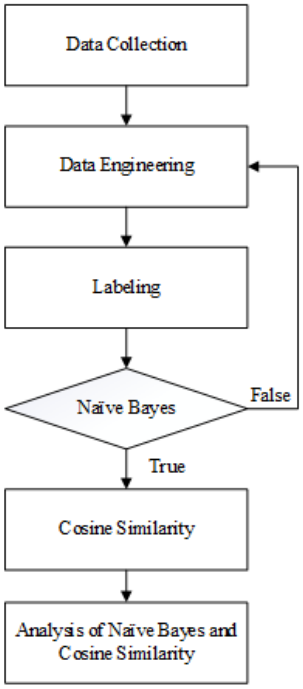


Figure 3. Research framework

Author ID	GARUDA_ID	OJS_IDENTIFIER	GARUDA_DOI	AKREDITA	GARUDA_TITLE
281	2346381	oai:ojs.ejournal.unp.ac.id:article/112194	DOI: 10.24036/jtev.v7i1.112194	-	Rancang Bangun Sistem Pengisia
281	2041984	oai::article/571	DOI: 10.28989/angkasa.v12i2.571	4	DC Motor Control Using Laborat
281	1150621	oai:ojs.publikasilmiah.unwahas.ac.id:article/2355	-	-	PENGHITUNG JUMLAH ORANG D
281	1913592	oai:ojs.jurnal.unissula.ac.id:article/3024	DOI: 10.30659/ei.2.2.%p	-	Penerapan Sistem Informasi Mor
281	550834	oai:ojs.portalgaruda.org:article/1152	DOI: 10.11591/eecsi.v3.1152	-	Joule-Thief Circuit Performance
281	558049	oai:ojs.portalgaruda.org:article/1152	DOI: 10.11591/eecsi.v3.1152	-	Joule-Thief Circuit Performance
283	1230817	oai:senatik.stta.ac.id:article/359	DOI: 10.28989/senatik.v5i0.359	-	Three-Phase Power Data Loggir
283	536052	oai:ojs.portalgaruda.org:article/410	DOI: 10.12928/jti.v6i1.	-	Indonesian Articles Recommend
283	1186558	oai:ojs.publikasilmiah.unwahas.ac.id:article/2034	DOI: 10.36499/jim.v13i2.2034	-	MONITORING JARAK JAUH DAN
283	1913667	oai:ojs.jurnal.unissula.ac.id:article/3062	DOI: 10.30659/ei.2.2.%p	-	Pengembangan Sistem Informasi
283	1913592	oai:ojs.jurnal.unissula.ac.id:article/3024	DOI: 10.30659/ei.2.2.%p	-	Penerapan Sistem Informasi Mor
283	352893	oai:ojs.publikasilmiah.unwahas.ac.id:article/1177	-	-	APLIKASI SENSOR PIR UNTUK SIS
283	113635	oai:publikasi.dinus.ac.id:article/189	-	-	DIREKTORI ONLINE PENELITIAN
283	1525248	oai:ojs.jurnal.uii.ac.id:article/2926	-	-	PERANCANGAN DIREKTORI BAH
283	1912144	oai:ojs.jurnal.unissula.ac.id:article/64	-	-	APLIKASI INTERAKTIF PEMBELAJ
283	1353559	oai:jurnal.unimus.ac.id:article/483	DOI: 10.26714/me.2.1.2009.%p	-	SIMULATOR PEMBANGKITAN SK
285	2034716	oai:journals.usm.ac.id:article/2989	DOI: 10.26623/elektrika.v13i1.2989	-	Optimasi Kualitas Jaringan WLAN
285	2160081	oai:jurnal.unimus.ac.id:article/6386	DOI: 10.26714/me.14.1.2021.32-41	-	KENDALI SISTEM PENGABUTAN
285	2260126	oai:ojs.ejournal.pnc.ac.id:article/726	DOI: 10.35970/infotekmesin.v12i2.726	-	Analisa Unjuk Kerja Software Det
285	1990002	oai:ojs.ejournal.undip.ac.id:article/29981	DOI: 10.14710/transmisi.22.4.135-141	-	ANALISA PEMODELAN ARUS TRA
285	1205910	oai:ojs.172.16.7.13:article/640	DOI: 10.25077/jnte.v8n2.640.2019	3	Desain dan Implementasi Akuisis
285	554786	oai:jurnal.untirta.ac.id:article/3308	DOI: 10.36055/setrum.v7i1.3308	-	Aplikasi Mobile Untuk Pencegaha
285	1186946	oai:ojs.portalgaruda.org:article/1646	DOI: 10.11591/eecsi.v5.1646	-	Co-channel Interference Monito
285	1215308	oai:ojs.portalgaruda.org:article/1646	DOI: 10.11591/eecsi.v5.1646	-	Co-channel Interference Monito
285	1913790	oai:ojs.jurnal.unissula.ac.id:article/1626	DOI: 10.30659/ei.2.1.15-20	-	Pemanfaatan E-Ktp Untuk Penga
285	550653	oai:ojs.portalgaruda.org:article/1128	DOI: 10.11591/eecsi.v3.1128	-	The Performance of SISO in Wire
285	557851	oai:ojs.portalgaruda.org:article/1128	DOI: 10.11591/eecsi.v3.1128	-	The Performance of SISO in Wire
285	1353562	oai:jurnal.unimus.ac.id:article/2061	DOI: 10.26714/me.9.1.2016.%p	-	IMPLEMENTASI SISTEM KOMUN
5988482	1304447	oai:ojs.www.iaescore.com:article/14960	DOI: 10.11591/ijece.v10i4.pp3441-3450	-	Fuzzy logic applications for data
5988482	536050	oai:ojs.portalgaruda.org:article/405	DOI: 10.12928/jti.v6i1.	-	COLOUR DETECTOR TOOL USIN

Figure 4. Research dataset

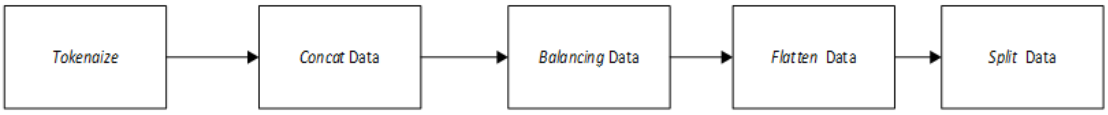


Figure 5. Stages of data engineering

AKREDITASI	GARUDA_TITLE	GARUDA_ABSTRACT	GARUDA_JOURNAL	GARUDA_YEAR_PUBLISH	GARUDA_DATE_PUBLISH	GARUDA_CITE	GARUDA_URL	ORIGINAL_URL	KATEGORI
-	Rancang Bangun Sistem Pengisian dan Penutup Bo...	Dunia industri saat ini tidak dapat lagi dipis...	JTEV (Jurnal Teknik Elektro dan Vokasional)	2021	2021-06-09	-	https://garuda.kemdikbud.go.id/documents/detai...	http://ejournal.unp.ac.id/index.php/jtev/artic...	Lainnya
4	DC Motor Control Using Laboratory Virtual Inst...	Motor has an important role in everyday life. ...	Angkasa: Jurnal Ilmiah Bidang Teknologi	2020	2020-11-02	-	https://garuda.kemdikbud.go.id/documents/detai...	https://ejournals.itda.ac.id/index.php/angkasa...	Lainnya
-	PENGHITUNG JUMLAH ORANG DALAM RUANG DENGAN SEN...	Seseorang yang berada dalam ruang akan melakuk...	Prosiding SNST Fakultas Teknik	2018	2018-08-29	-	https://garuda.kemdikbud.go.id/documents/detai...	https://publikasiilmiah.unwahas.ac.id/index.ph...	Lainnya
-	Penerapan Sistem Informasi Monitoring Tugas Ak...	Banyaknya mahasiswa yang sedang mengerjakan tu...	TRANSISTOR Elektro dan Informatika	2017	2018-07-27	-	https://garuda.kemdikbud.go.id/documents/detai...	http://jurnal.unissula.ac.id/index.php/EI/arti...	Lainnya
-	Joule-Thief Circuit Performance for Electricit...	The alternative energy such as battery as powe...	Proceeding of the Electrical Engineering Compu...	2016	2016-12-01	-	https://garuda.kemdikbud.go.id/documents/detai...	http://journal.portalgaruda.org/index.php/EECS...	Lainnya

Figure 6. Category labeling

2.1.3. Labeling

Labeling in this research using a rule-based method is carried out based on predetermined keywords. This method categorizes the labels automatically, the specified keywords must be relevant to the title and abstract of the article. The labeling process in this study changes words to lowercase letters. Labels based on predefined keywords, the result is that if there are keywords, they will automatically enter the category labels that have been determined, if there are no keywords, they will automatically enter other labels.

2.1.4. Classification using naive Bayes

Classification is carried out after the dataset pre-processing stage is completed, and a classification model is carried out using the naïve Bayes method. Process steps to measure classification performance by calculating accuracy, recall, precision, and F1-score [22]. Calculating accuracy can use (1).

$$accuracy = \frac{true\ positive + true\ negative}{true\ positive + true\ negative + false\ positive + false\ negative}\ x\ 100\% \quad (1)$$

Calculating the precision can be done using (2).

$$precission\ = \frac{true\ positive}{true\ positive + false\ positive} \quad (2)$$

Calculating recalls can be done using (3).

$$recall\ = \frac{true\ positive}{true\ positive + false\ negative} \quad (3)$$

Calculating the F1-score can be done using (4).

$$\mathrm{F1} - score = 2\ x\ \frac{precission * recall}{precission + recall} \quad (4)$$

Information:
True positive: the amount of correctly classified positive data.
True negative: the amount of correctly classified negative data.
False positive: the amount of negative data that is incorrectly classified as positive.
False negative: the amount of positive data that is incorrectly classified as negative.

2.1.5. Article similarity detection using cosine similarity

The detection of similarity in articles using the cosine similarity method is very appropriate to evaluate how much similarity between classes and the result is in the form of vector angle parameters. Calculating similarity has a range value from 0-1. The calculation formula for the similarity results uses (5).

$$Cos\theta = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2}\sqrt{\sum_1^n b_i^2}} \quad (5)$$

Information:
a_i and b_i : components of two vectors A and B.
$\Sigma a_i b_i$: dot product (multiplication of dots) between two vectors.
Σa_i^2 and Σb_i^2: the length (magnitude) of each vector.
n: unique word count.

This research uses vector weighting to calculate cosine similarity in vector form using TF-IDF. TF-IDF is a statistical method used to measure the level of importance of a term in a document [23]. TF indicates the frequency of the word appearing in the document; the higher the TF value is considered the more relevant the word is in the document [24]. The word weight is obtained from the multiplication of TF and IDF [25]. This calculation begins by determining two vectors, namely article 1 and article 2, which represent the object being compared, in this research, based on the title and abstract that have been concatenated. The first step in the calculation process is to determine the dot product of the two vectors, which is obtained by adding the result of the multiplication of each corresponding element in each vector. In the second step, the length or magnitude of each vector is calculated by taking the square root of the sum of the squares of each element in that vector. The cosine similarity value is then obtained by dividing the dot product result by the product multiplication of the lengths of the two vectors, according to the equation. The results of this calculation are in the range of 0 to 1, where values close to 1 indicate a high degree of similarity, values close to 0 indicate the absence of a significant relationship, and values close to -1 indicate that the two vectors have opposite directions. The similarity detection stages are depicted in Figure 7.

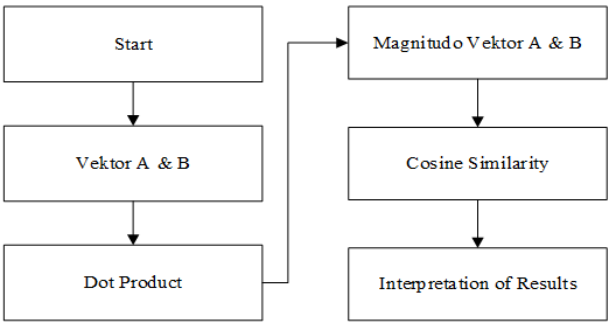


Figure 7. Stages of cosine similarity

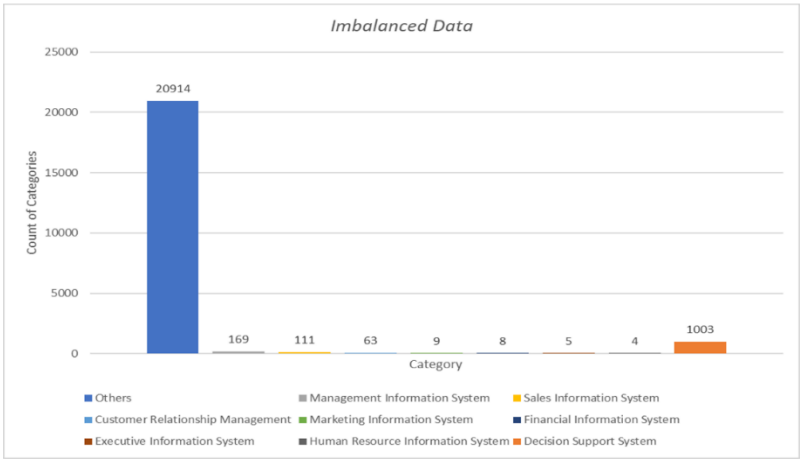
3. RESULTS AND DISCUSSION

This section describes the results and discussion of the data balancing process. The results of data balancing are illustrated in Table 1, and the data difference graph is illustrated in Figure 8. Figure 8 shows a graph of the data classification results, Figure 8(a) graphs that provide classification results using imbalanced data, and Figure 8(b) graphs that provide classification results using balanced data using the random over-sampling (ROS) method. The final stage discusses the results of classification using the naive Bayes method, the results of article similarity detection, and the results of scientific article search using the cosine similarity method. Table 1 describes the dataset that has done category auto-labeling based on the keywords that have been defined. There are two columns in the number of datasets: the number before and the number after. The previous number results from auto-labeling without using the over-sampling method. The number after is the result of auto-labeling with data balancing using ROS, which has a way of working by identifying minority classes and majority classes. Duplicate data from the minority classes of 20,914 classes to be balanced with the majority class [26]. Figure 8(a) depicts imbalanced data with other categories as many as 20,914, count of appointment information system is 169, decision support system is 1,003, sales information system is 111, customer relationship management is 63, marketing information system is 9, financial information system is 8, executive information system is 5, and

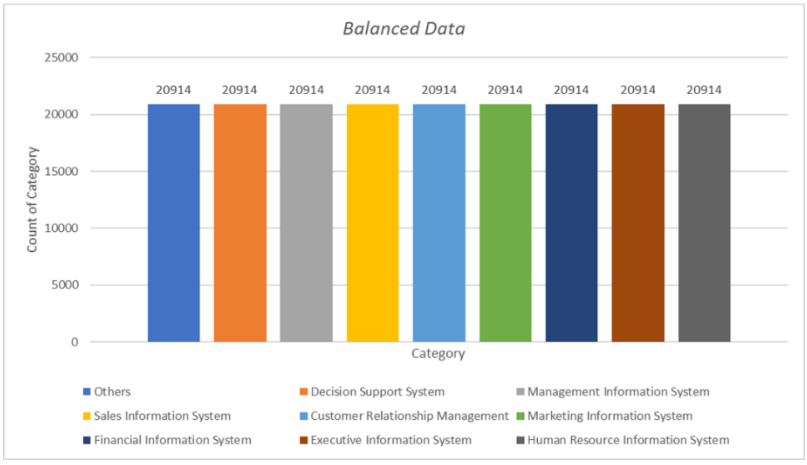
human resources information system is 4. Figure 8(b) illustrates the balanced data that has been carried out using ROS, and all categories are balanced as many as 20,914 in each category.

Table 1. Balancing data using ROS

Category	Total	
	Before ROS	After ROS
Other	20,914	20,914
Management information systems	169	20,914
Decision support system	1,003	20,914
Sales information system	111	20,914
Customer relationship management	63	20,914
Marketing information system	9	20,914
Financial information system	8	20,914
Executive information systems	5	20,914
Human resource information systems	4	20,914



(a)



(b)

Figure 8. Graph of the data classification results: (a) imbalanced data and (b) balanced data

3.1. Classification using naive Bayes

This classification conducted one experiment with 80% of the training data and 20% of the test data from 29,239 rows. The results of the naive Bayes classification model test can be seen in Table 2, which uses imbalanced data, and Table 3, which uses balanced data. The results of two tests with different data gave excellent F1-score accuracy results. The test results using imbalanced data did not provide good classification results, and no category labels were detected correctly. The test result using balanced data gives good classification results and correctly detects category labels.

The classification results in Table 2 show the impact of imbalanced data on model performance. Based on the classification results, the other category has a precision value of 0.94, a recall of 1.00, and an F1-score of 0.96, with the total data reaching 4,197 samples. This suggests that the model tends to classify most of the data into other categories caused by unbalanced data. In contrast, other categories, such as management information systems, marketing information systems, and decision support systems, have an F1 score of 0.00, indicating that the model cannot easily recognize data in those categories. Low macro average values include precision 0.11, recall 0.12, and F1-score 0.12. It shows that the model is inaccurate in classifying classes with a few samples. The weighted average is higher because the majority class influences it. Although the model has an overall accuracy of 0.94, this value cannot indicate that the model classifies well due to refraction towards the majority class. To improve the model's performance in classifying minority classes, strategies such as oversampling are needed so that the labels of class categories are more balanced and the classification results are more appropriate.

Table 2. Classification results using imbalanced data

	Precision	Recall	F1-score	Support
Other	0.94	1.00	0.96	4,197
Customer relationship management	0.00	0.00	0.00	10
Executive information systems	0.00	0.00	0.00	3
Financial information system	-	-	-	-
Management information systems	0.00	0.00	0.00	36
Marketing information system	0.00	0.00	0.00	1
Sales information system	0.00	0.00	0.00	21
Human resource information systems	0.00	0.00	0.00	3
Decision support system	0.00	0.00	0.00	187
Accuracy			0.94	4,458
Macro avg	0.11	0.12	0.12	4,458
Weighted avg	0.88	0.94	0.91	4,458

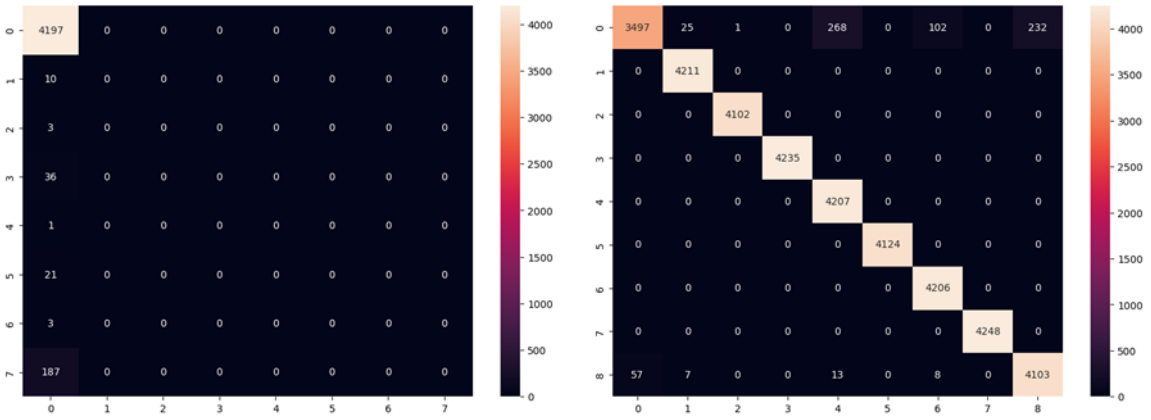
Table 3. Classification results using balanced data

	Precision	Recall	F1-score	Support
Other	0.98	0.85	0.91	4,125
Customer relationship management	0.99	1.00	1.00	4,211
Executive information systems	1.00	1.00	1.00	4,102
Financial information system	1.00	1.00	1.00	4,235
Management information systems	0.94	1.00	0.97	4,207
Marketing information system	1.00	1.00	1.00	4,124
Sales information system	0.97	1.00	0.99	4,206
Human resource information systems	1.00	1.00	1.00	4,248
Decision support system	0.95	0.98	0.96	4,188
Accuracy			0.98	37,646
Macro avg	0.98	0.98	0.98	37,646
Weighted avg	0.98	0.98	0.98	37,646

The classification results illustrated in Table 3 show the classification results obtained using balanced data, namely category label data that has been balanced using ROS. The model's performance was evaluated based on three primary measurements: precision, recall, and F1-Score, which showed the accuracy and consistency of the model in classifying data. The analysis results show that almost all classes have precision and recall above 0.94, and the model can perform classification with minimal error rate, without refraction that impacts certain classes. The overall Accuracy value reached 0.98, indicating the model has excellent prediction performance. The macro average and weighted average values, each valued at 0.98, also indicate the model has a balanced performance across categories. Thus, correctly applying data balancing can improve the model's performance compared to the imbalanced data condition, where some categories previously had a lower F1-score. These results show that the data balancing strategy can reduce refraction in classification and improve accuracy.

The confusion matrix results from the classification model test using naive Bayes are shown in Figure 9(a) confusion matrix with imbalanced data and Figure 9(b) confusion matrix with balanced data. Figure 9(a) shows that the model tends to classify data into only one dominant category, with many other classes having a near-zero prediction count. This shows that the model is biased towards the majority class, so it cannot recognize the patterns of the minority classes well. Figure 9(b) of the confusion matrix for balanced data shows a more even distribution of predictions along the diagonal of the matrix. The model can classify samples into appropriate classes with fewer errors. A comparison of these two matrices shows that

relevant data balancing improves model performance by reducing bias against majority classes and enabling more accurate classification across categories.



(a) (b)

Figure 9. Confusion matrix for (a) imbalanced data and (b) balanced data

3.2. Article similarity detection using Cosine Similarity

This process detects similarities using the title and abstract of Article 1 and Article 2. After the experiment, the similarity detection score was obtained as 0.071, shown in Table 4. Table 4 shows the results of the similarity detection analysis between two articles based on the calculation of similarity scores. Article 1 is titled "Web-based decision support system assessment..." which focuses on the implementation of a decision support system in the context of assessment in a village, while Article 2 is entitled "Design and build automatic bottle filling and capping system based on bottle height ..." which discusses automation systems in the manufacturing industry. The calculation results showed that the similarity score between the two articles was 0.071, which indicates a very low level of similarity. This value indicates that the two articles significantly differ in topic, terminology, and content. Thus, the similarity detection method is proven to distinguish articles with different topics well. Based on a similarity score range of 0 to 1, provide a good score to detect article similarities. In this research, the highest score was used to obtain an accurate and relevant article according to the research topic.

Table 4. Similarity detection results

Article	Title	Brief abstract
1	Web-based decision support system assessment ...	Pringsari Village is one of the villages in the sub-district .... Where is each village ...
2	Design and build automatic bottle filling and capping systems based on bottle height...	Today's industrial world can no longer be separated by the problem of automation for various production facilities. …
Score cosine similarity		0.071

3.3. Article search using cosine similarity

This process searches for scientific articles that are similar to the main article based on the title and abstract that have been concatenated. The value range is 0 to 1. The results of the search display a table containing the columns garuda title, garuda abstract, similarity score, category, and predicted category. The predicted category is the wrong category label column in classifying the category, as shown in Figure 10.

The search results of articles in Figure 10 show the results of the similarity score calculation, where the article with the highest score has the most significant level of similarity with the main article. The article with the highest similarity score is placed first in the search results. This process compares the search text with the title and abstract from the available dataset. Next, the search results are sorted by similarity score in descending order so that the most common documents appear first. The results of this study show that the top five documents have the highest level of similarity to the keywords of web-based decision support systems.

These results are displayed in a table that includes the title columns, abstract, similarity score, original category, and predicted category. The similarity score obtained ranged from 0.39 to 0.26, which indicates a difference in the level of similarity of documents. The original and predicted categories showed a

reasonably high match, indicating that the model can recognize and group documents quite well based on text similarity analysis.

```
search = "Decision Support System"
searched_data = data.copy()
searched_data['SIMILARITY_SCORE'] = searched_data.apply(lambda row: calculate_similarity(doc1=search, doc2=f"{row['GARUDA_TITLE']} {row['GARUDA_ABSTRACT']}"), axis=1)
searched_data = searched_data.sort_values(by='SIMILARITY_SCORE', ascending=False).reset_index(drop=True)

# Line 1 - 5
searched_data[['GARUDA_TITLE', 'GARUDA_ABSTRACT', 'SIMILARITY_SCORE', 'CATEGORY', 'CATEGORY_PREDICTED']].iloc[0:5, :]
```

Figure 10. Article search results

Table 5. Article search results

	GARUDA_TITLE	GARUDA_ABSTRACT	SIMILARITY_SCORE	CATEGORY	CATEGORY_ PREDICTED
0	Decision support system for recipient selection…	Decision support system as a system…	0.391596	Decision support system	Decision support system
1	Decision support system for new student admission…	A decision support system is a system…	0.295967	Decision support system	Decision support system
2	Decision support system for employee recruitment…	Current information technology developments…	0.275679	Decision support system	Decision support system
3	Decision support system for restaurant selection…	The abstract of this research aims to help…	0.269314	Decision support system	Decision support system
4	Decision support system for employee performance evaluation…	Employee performance evaluation decision support system…	0.267959	Decision support system	Decision support system

4. CONCLUSION

The naive Bayes method used for classification provides a good level of F1-score accuracy by using balanced data of 98% and imbalanced data of 94%. The classification process takes less than 60 minutes to process and classify the article categories. The cosine similarity method used for similarity detection and search for the main article with other articles gave a good similarity detection score of 0.071. In contrast, a similarity score close to 1 indicates a higher similarity in searching for articles relevant to the specified research topic. Searches for scientific articles that are similar to the main article provide excellent search results. The research results show that the proposed method significantly improves the classification and search processes in GARUDA, and provides accurate and efficient similarity detection. Further research can be developed by adding more datasets in the scientific field, not only in the Indonesian language, and by applying features to find articles similar to the GARUDA.

ACKNOWLEDGMENTS

The authors would like to thank the Comnets Research Group, University of Sriwijaya, Indonesia for providing full support for their research necessary.

FUNDING INFORMATION

Authors state no funding 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
Nyimas Sabilina Cahyani	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Deris Stiawan		✓		✓		✓	✓		✓	✓				
Abdiansah Abdiansah		✓		✓		✓	✓		✓	✓				
Nurul Afifah	✓	✓	✓				✓	✓		✓				
Dendi Renaldo Permana	✓	✓	✓				✓	✓		✓				

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The authors confirm that the data supporting the findings of this study are available within the article [and/or its supplementary materials].

REFERENCES

[1] L. Lukman *et al.*, "Proposal of the s-score for measuring the performance of researchers, institutions, and journals in Indonesia," *Science Editing*, vol. 5, no. 2, pp. 135–141, 2018, doi: 10.6087/KCSE.138.

[2] M. M. Saritas and A. Yasar, "Performance analysis of ANN and naive Bayes classification algorithm for data classification," *International Journal ofIntelligent Systems and Applications in Engineering*, vol. 7, no. 2, pp. 88–91, 2019, doi: 10.18201//ijisae.2019252786.

[3] A. S. Osman, "Data mining techniques: review," *International Journal of Data Science Research*, vol. 2, no. 1, pp. 1–4, 2019.

[4] F. R. Lumbanraja, E. Fitri, Ardiansyah, A. Junaidi, and R. Prabowo, "Abstract classification using support vector machine algorithm (case study: abstract in a computer science journal)," *Journal of Physics: Conference Series*, vol. 1751, no. 1. 2021, doi: 10.1088/1742-6596/1751/1/012042.

[5] S. Latif, U. Suwardoyo, and E. A. W. Sanadi, "Content abstract classification using naive Bayes," *Journal of Physics: Conference Series*, vol. 979, no. 1, 2018, doi: 10.1088/1742-6596/979/1/012036.

[6] B. Parlak and A. K. Uysal, "On classification of abstracts obtained from medical journals," *Journal of Information Science*, vol. 46, no. 5, pp. 648–663, 2020, doi: 10.1177/0165551519860982.

[7] I. C. Chang, T. K. Yu, Y. J. Chang, and T. Y. Yu, "Applying text mining, clustering analysis, and latent dirichlet allocation techniques for topic classification of environmental education journals," *Sustainability*, vol. 13, no. 19, 2021, doi: 10.3390/su131910856.

[8] K. Yasaswi, V. K. Kambala, P. S. Pavan, M. Sreya, and V. Jasmika, "News classification using natural language processing," in *Proceedings of 3rd International Conference on Intelligent Engineering and Management, ICIEM 2022*, 2022, pp. 63–67, doi: 10.1109/ICIEM54221.2022.9853174.

[9] N. N. Qomariyah, A. S. Araminta, R. Reynaldi, M. Senjaya, S. D. A. Asri, and D. Kazakov, "NLP text classification for COVID-19 automatic detection from radiology report in Indonesian language," in *2022 5th International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2022*, 2022, pp. 565–569, doi: 10.1109/ISRITI56927.2022.10053077.

[10] N. Kumar, R. R. Suman, and S. Kumar, "Text classification and topic modelling of web extracted data," in *2021 2nd Global Conference for Advancement in Technology, GCAT 2021*, 2021, pp. 1–8, doi: 10.1109/GCAT52182.2021.9587459.

[11] Y. Kang, Z. Cai, C. W. Tan, Q. Huang, and H. Liu, "Natural language processing (NLP) in management research: a literature review," *Journal of Management Analytics*, vol. 7, no. 2, pp. 139–172, 2020, doi: 10.1080/23270012.2020.1756939.

[12] N. Malik, A. Bilal, M. Ilyas, S. Razzaq, F. Maqbool, and Q. Abbas, "Plagiarism detection using natural language processing techniques," *Technical Journal, University of Engineering and Technology (UET)*, vol. 26, no. 1, pp. 2313–7770, 2021.

[13] A. Hizqil and Y. Ruldeviani, "Sentiment analysis of online licensing service quality in the energy and mineral resources sector of the Republic of Indonesia," *Computer Science and Information Technologies*, vol. 5, no. 1, pp. 63–71, 2024, doi: 10.11591/csit.v5i1.pp63-71.

[14] U. Mardatillah, W. B. Zulfikar, A. R. Atmadja, I. Taufik, and W. Uriawan, "Citation analysis on scientific articles using Cosine Similarity," in *Proceeding of 2021 7th International Conference on Wireless and Telematics, ICWT 2021*, 2021, pp. 1–4, doi: 10.1109/ICWT52862.2021.9678402.

[15] A. Islam, E. Rahman, A. A. Chowdhury, and M. A. N. Mojumder, "A deep learning approach to detect plagiarism in Bengali textual content using similarity algorithms," in *Proceedings of IEEE InC4 2023-2023 IEEE International Conference on Contemporary Computing and Communications*, 2023, vol. 1, pp. 1–5, doi: 10.1109/InC457730.2023.10262998.

[16] P. Y. Ristanti, A. P. Wibawa, and U. Pujianto, "Cosine Similarity for title and abstract of economic journal classification," in *Proceeding-2019 5th International Conference on Science in Information Technology: Embracing Industry 4.0: Towards Innovation in Cyber Physical System, ICSITech 2019*, 2019, pp. 123–127, doi: 10.1109/ICSITech46713.2019.8987547.

[17] T. P. Rinjeni, A. Indriawan, and N. A. Rakhmawati, "Matching scientific article titles using Cosine Similarity and Jaccard Similarity algorithm," *Procedia Computer Science*, vol. 234, pp. 553–560, 2024, doi: 10.1016/j.procs.2024.03.039.

[18] V. Nuipian and J. Chuaykhun, "Book recommendation system based on course descriptions using Cosine Similarity," in *ACM International Conference Proceeding Series*, 2023, pp. 273–277, doi: 10.1145/3639233.3639335.

[19] R. Singh and S. Singh, "Text similarity measures in news articles by vector space model using NLP," *Journal of The Institution of Engineers (India): Series B*, vol. 102, no. 2, pp. 329–338, 2021, doi: 10.1007/s40031-020-00501-5.

[20] A. S. Dina, A. B. Siddique, and D. Manivannan, "Effect of balancing data using synthetic data on the performance of machine learning classifiers for intrusion detection in computer networks," *IEEE Access*, vol. 10, pp. 96731–96747, 2022, doi: 10.1109/ACCESS.2022.3205337.

[21] P. Mooijman, C. Catal, B. Tekinerdogan, A. Lommen, and M. Blokland, "The effects of data balancing approaches: a case study," *Applied Soft Computing*, vol. 132, 2023, doi: 10.1016/j.asoc.2022.109853.

[22] H. Chen, S. Hu, R. Hua, and X. Zhao, "Improved naive Bayes classification algorithm for traffic risk management," *Eurasip Journal on Advances in Signal Processing*, no. 1, 2021, doi: 10.1186/s13634-021-00742-6.

[23] Z. Liu, J. Zhu, X. Cheng, and Q. Lu, "Optimized algorithm design for text similarity detection based on artificial intelligence and natural language processing," *Procedia Computer Science*, vol. 228, pp. 195–202, 2023, doi: 10.1016/j.procs.2023.11.023.

[24] S. W. Kim and J. M. Gil, "Research paper classification systems based on TF-IDF and LDA schemes," *Human-centric Computing and Information Sciences*, vol. 9, no. 1, 2019, doi: 10.1186/s13673-019-0192-7.

[25] H. Yuan, Y. Tang, W. Sun, and L. Liu, "A detection method for android application security based on TF-IDF and machine learning," *PLoS ONE*, vol. 15, pp. 1–19, 2020, doi: 10.1371/journal.pone.0238694.
[26] C. Padurariu and M. E. Breaban, "Dealing with data imbalance in text classification," *Procedia Computer Science*, vol. 159, pp. 736–745, 2019, doi: 10.1016/j.procs.2019.09.229.

BIOGRAPHIES OF AUTHORS

Nyimas Sabilina Cahyani received an S.Kom. degree in Information Systems from STMIK GI MDP in 2017. She is a Master's student in Computer Science at Universitas Sriwijaya. She can be contacted at email: arisabil.ns@gmail.com.

Deris Stiawan received a Ph.D. degree in Computer Engineering from Universiti Teknologi Malaysia, Malaysia. He is currently a Professor at the Department of Computer Engineering, Faculty of Computer Science, Universitas Sriwijaya. His research interests include computer networks, intrusion detection/prevention systems, heterogeneous networks, and intelligent systems. He can be contacted at email: deris@unsri.ac.id.

Abdiansah Abdiansah received a Dr. degree in Computer Science from Universitas Gadjah Mada, Yogyakarta. He is currently Lecturer and Head of the D-III Study Program in Informatics Management, Universitas Sriwijaya. His research interests include artificial intelligence, natural language processing, and intelligent tutoring system. He can be contacted at email: abdiansah.unsri@gmail.com.

Nurul Afifah received the Bachelor Degree in Computer Engineering and master's degrees in Computer Science at 2014 and 2019. She joined Universitas Sriwijaya as a Lecturer in January 2020. She is currently a Researcher at COMNETS RG. Her research interests include information security, IoT system and security, blockchain and machine learning. She can be contacted at email: nurul@unsri.ac.id.

Dendi Renaldo Permana received a B.Com. degree in the information system program at University Riau. His main focus during career was initially software engineer and machine learning engineer. Then he received a scholarship called the *magister menuju doktor untuk sarjana unggul* (PMDSU) in 2023 to continue his master and doctoral studies in computer science at Universitas Sriwijaya with focus research in the field of cyber threat intelligence. He can be contacted at email: dendi.renaldo@gmail.com.