Text Similarity Analysis for Evaluating Alignment Between Lesson Plans and Teaching Reports

Antonius Rachmat Chrismanto*[1], Willy Sudiarto Raharjo[2], Oscar Gilang Purnajati[3]
[1,2]Prodi Informatika, Fakultas Teknologi Informasi Universitas Kristen Duta Wacana, Yogyakarta, Indonesia
[3]Prodi Pendidikan Dokter, Fakultas Kedokteran Universitas Kristen Duta Wacana, Yogyakarta, Indonesia
e-mail: *[1]anton@ti.ukdw.ac.id, [2]willysr@ti.ukdw.ac.id, [3]oscargilang@staff.ukdw.ac.id

Abstract

RPS (Rencana Pembelajaran Semester, or called Lesson Plans) is a class activity planning document in the higher education learning process that includes learning outcomes, methods, learning strategy, and evaluation criteria. It is created by the lecturers in charge of the course and coordinated with the relevant department. This document needs to be monitored throughout the semester for its conformity with the implementation document (Borang Pelaksanaan Perkuliahan (BPP)). It was done manually through our eRPS system, but it requires a lot of effort and precision and is not time-efficient. This research focused on evaluating the effectiveness of several content-based text similarity methods to detect RPS conformity compared with the BPP, or called Teaching Reports document. The Boyer-Moore (B), Rabin-Karp (R), Jaccard (JC), Jaro-Winkler (JW), Smith-Waterman (SW), Knuth-Morris-Pratt (K), Levenehtein cosine similarity (C), Dice (D), Jaro (J), and Soundex (S) algorithms were evaluated in this paper. In the vector-based similarity method, TF-IDF was used. The evaluation of 11 string-matching algorithms across four scenarios demonstrated clear performance trends. Fuzzy algorithms (SW with accuracy 0,845–0,870, and JW with accuracy 0,840-0,850) achieved the highest accuracy in a single row of lecturer scenario, while exact/pattern-based algorithms (B, K, and S with accuracy 0,8625–0,8725) on a combination of all rows of lectures with minimal variance (≈0,005–0,015). Pre-processing benefits fuzzy algorithms (+2.5%) but is neutral for exact/pattern-based algorithms. The combined scenario improves the exact/phonetic algorithms (+6–7%) but reduces the fuzzy performance algorithm (−10–14%). The optimal thresholds were generally 40–50%, except for JW and J, which were 65%.

Keywords— Lesson Plans, Teaching Reports, Text Similarity algorithms, Class evaluation

1. INTRODUCTION

RPS (Rencana Pembelajaran Semester) / Lesson plan is a class activity planning document in the higher education learning process that includes learning outcomes, methods, learning strategy, and evaluation. It is created by the lecturers in charge of the course and coordinated with the department [1] as a plan before the course starts. In reality, there is always a deviation between the RPS and its implementation during the course for various reasons. These differences include content/schedule ordering, the content of the materials being taught, and the learning methodology.

Universitas Kristen Duta Wacana (UKDW) requires all lecturers to create RPS documents before the start of a new semester. These documents are reviewed and evaluated by Lembaga Pengembangan Akademik dan Inovasi Pembelajaran (LPAIP) to ensure that all the plans in the RPS document are implemented accordingly.

LPAIP has a web-based eRPS system (https://rps.ukdw.ac.id) that lecturers can use to create their RPS documents before a course begins, which must be validated by the head of the department. Currently, the eRPS is not equipped with a conformity detection module between the RPS and its implementation during a course. All lecture activities are monitored and logged in BPP (Borang Pelaksanaan Perkuliahan) / teaching report document in a different system (eClass). This complicates the monitoring process because these two documents are separated across different systems.

Text similarity is a subfield of computer science called Natural Language Processing (NLP), which aims to identify the level of similarity between two or more texts. Text similarity is commonly found in plagiarism [4], document similarity, news similarity, and other text-based content, such as social media [5], and others [6].

There are two main works on this project: 1) developing a system that can detect conformity level between RPS and BPP which can be integrated into eRPS and 2) measuring the accuracy and flexibility of several text similarity methods, including Boyer-Moore (B), Rabin-Karp (R), Jaccard (JC), Jaro Winkler (JW), Smith-Waterman (SW), Knuth Morris Prat (K), Lavensthein distance (L), Dice (D), Jaro (J) Soundex (S), and cosine similarity (C) algorithm. The eRPS system development was completed by other work [7] as part of an effort to help users improve their efficiency.

Text data is the most common form of data used in many applications. It is also one of the easiest forms to process. An example of an implementation that uses text data is a search engine. Using search engine technology, this study developed text mining, text analysis, text processing, text similarity, and information retrieval technology. There have been many applications that use these technologies, such as plagiarism detection, news similarity detection [8], spam detection [9], question duplications [10], question answering in the form of an essay [11], word similarity in class diagram generator applications in the software engineering field [12], and spam detection in social media platforms. A higher level of text similarity is referred to as contextual text similarity. Contextual text similarity is related to the proximity between texts that share the same meaning but have different structures, counts, numbers, positions, and lengths. To solve this problem, a more comprehensive understanding of semantic similarity, which is above lexical similarity, is required. Methods that go into the lexical similarity category are Cosine Similarity, Jaccard Similarity, Sørensen–Dice coefficient, and Levenshtein Distance. Methods that go into semantic text similarity are word/sentence embeddings [13-14], contextual language models, machine learning [15], and deep learning using Transformers [16].

2. RESEARCH METHODS

2.1. The Methodology

The prototyping method was used as the system development approach because it allowed us to showcase our work gradually over time to our users and revise it accordingly in each cycle. This study was conducted in several steps as follows.

1. Data collection: collected RPS documents from the year 2023, which were manually validated by the LPAIP as the main reference for the system. These documents are converted into an Excel document and then performed pre-processing [17], which includes tokenization, stemming [18-21], stopword removal [22], normalization [23-25], and cleaning steps. The final step was to split the dataset into training, validation, and test data based on the scenario.
2. System development: a web-based module using PHP/Python was used to implement all the methods (Boyer-Moore [26] [27], Rabin-Karp [28], Jaccard [29] [30], Jaro Winkler [31-33], Smith-Waterman [34], Khuth Morris Prat [35], Lavensthein distance [36], Dice Coefficient [37], Jaro [32], Soundex [38], and Cossine Similarity [39-42]). In the prototyping phase, the Jupyter Notebook was used as the development framework to assist us during testing.

3. Testing and evaluation: Accuracy and flexibility level evaluation methods were used in this study. The accuracy evaluation method was implemented using the following accuracy metrics: precision, recall, and F1-score. From these components, an accuracy evaluation performance benchmark of several text similarity methods used in the RPS study case. The second evaluation was conducted to test the flexibility of text similarity methods against the possibility of typos in both RPS and BPP document content. The main reference is the document validated by LPAIP UKDW in 2024 for the academic year 2023/2024 for all departments and for all odd and even semesters.

The detailed workflow of the development approach is presented in Table 1.

2.2. Table 1. Development Workflow

No.	Phase	Input	Process and Tools	Output
1.	Data Gathering	RPS and BPP document from 2023	Convert to Excel	Excel documents
2.	Data Preprocessing and Cleaning	Excel documents for RPS and BPP	Preprocessing (tokenisation, stemming, stopwords removal, normalization) and cleaning	Clean dataset
3.	Scenario Generation	Clean dataset	Train, Validation, Test Split	Dataset split by train, validation, and test based on scenario
4.	Methods Implementation	Dataset split by train, validation, and test based on scenario	*Text similarity methods*: Boyer-Moore (B), Robin-Karp (R), Jaccard (JC), Jaro-Winkler (JW), Smith-Waterman (SW), Knuth Morris Prat (K), Levenshtein distance (L), Dice (D), Jaro (J), Soundex (S), and cosine similarity (C).	
5.	Design and Prototyping	-	Development program/module using Python / PHP	Program/module
6.	Evaluation and Analysis	*Result*	Evaluation: accuracy and flexibility metrics	Comparison of the best methods for evaluation

2.3. System Testing Plan

Two evaluation methods are used: accuracy and flexibility evaluation methods. The accuracy evaluation method is based on precision, recall, and F1-Score, while the flexibility evaluation method uses scenario-based testing against the dataset. The details of the testing plan are presented in Table 2. The baseline dataset is the RPS and BPP document without any pre-processing actions. The dataset with pre-processing means that it goes through stemming, stopword removal, and normalization of duplicate content and typos. Next, we compared the material content in each lecture with all lectures combined for one whole semester. From Table 2, we can see the result for each algorithm performance used: Boyer-Moore (B), Rabin-Karp (R), Jaccard (JC), Jaro Winkler (JW), Smith-Waterman (SW), Knuth Morris Prat (K), Lavensthein distance (L), Dice (D), Jaro (J) Soundex (S), and cosine similarity (C).

Table 2. System Testing Plan

Scenario	Accuracy
	All text similarity algorithms
RPS/BPP Dataset without pre-processing (each lecture) with thresholds 40, 50, 65, 85%	… %
RPS/BPP Dataset with pre-processing (single row of lecture) with thresholds 40, 50, 65, 85%	... %

.

RPS/BPP Dataset without pre-processing (all rows of lectures combined) with thresholds of 40, 50, 65, and 85%	… %
RPS/BPP Dataset with pre-processing (all rows of lectures combined) with thresholds 40, 50, 65, 85%	… %
Average	… %

2.4. System Evaluation

Several evaluation methods are used to determine the accuracy of the system. In the text similarity detection system, the accuracy of the methods is tested against the dataset. Test data were obtained from the dataset using K-Fold Validation [20], [21], [43], [44]. K-Fold Validation works by splitting the entire dataset into three parts: training, validation, and testing. In the RPS implementation dataset, there was no training data; therefore, we only experimented using the test dataset. The accuracy, precision, and F1-score are measured. The flexibility of the method is tested using data that had not been tested during system development.

In the system testing, the conformity level between the content from the RPS is evaluated and its implementation in the BPP document using the confusion matrix in Table 3.

Table 3. Confusion Matrix

		Prediction Class	
		Negative	Positive
Real Class	Negative	True Negative (TN)	False Negative (FN)
	Positive	False Positive (FP)	True Positive (TP)

Based on the confusion matrix in Table 4, it is conducted further tests to obtain the accuracy, precision, recall, and F1-score using Equations 1 – 4 [45]:

Accuracy = (TN + TP) / (TN + FP + FN + TP) (1)

Recall = TP / (FP + TP) (2)

Precision = TP / (FN + TP) (3)

F1-Score = 2 * TP / (2 * TP + FP + FN) (4)

In addition to the evaluation metrics above, a scenario-based evaluation was implemented to determine the flexibility of the text similarity methods used. The scenarios will use several combinations to determine the best combination, which can be used for the next prototype cycle.

3. RESULT AND DISCUSSION

3.1. Requirements gathering and Dataset Analysis

In this phase, all the RPS and BPP documents are collected that would be used by the LPAIP and Puspindika units at UKDW. The RPS document was retrieved from the online eRPS system (https://rps. ukdw. ac. id/). The collected data were in Excel format, which had the following columns: materi_rps, sub_cpmk_rps, and a link to the finalized PDF. From Puspindika, the BPP documents are collected after the courses were completed at the end of the semester. The collected data consists of the following fields: kdsemester, kode, grup, prodi, nik, tatap_muka, tanggal_pertemuan, pukul, topik_pertemuan, topik, keterangan_tambahan, metode_pembelajaran, bentuk_pembelajaran, media_pembelajaran, media_lain, dosen_pengampu, and jumlah_hadir. An example of complete data from the accounting department is provided in Table 4.

Table 4. Dataset example from the Accounting Department

kdsemester	20241	20241
kode	AK1113	AK1113
grup	A	A
prodi	Prodi Akuntansi	Prodi Akuntansi
nik	202304612	202304612
tatap_muka	1	2
tanggal_pertemuan	2024-08-26	2024-09-02
pukul	14:30	14:30
topik_pertemuan		Weygandt, Kimmel, and Kieso (2018). Chapter 1
topik		
judul	Pertemuan ke-1	Pertemuan ke-2
keterangan_tambahan		
gabungan_pertemuan		Weygandt, Kimmel, and Kieso (2018). Chapter 1
materi_rps	Accounting Universe	Weygandt, Kimmel, and Kieso (2018). Chapter 1
sub_cpmk_rps	Mahasiswa dapat mengklasifikasikan runtutan dunia akuntansi	Mahasiswa mampu mengidentifikasi, mempraktikan, dan menyatakan aktivitas akuntansi, pengguna data akuntansi, serta mampu mengklasifikasikan data transaksi ke dalam persamaan dasar akuntansi.
kesesuaian	Tidak	Ya
metode_pembelajaran	Kuliah/Transfer Knowledge (TCL);	Kuliah/Transfer Knowledge (TCL); Small Group Discussion;
bentuk_pembelajaran	Tatap Muka	Tatap Muka
media_pembelajaran	eClass;	eClass;
media_lain	-	-
dosen_pengampu	Albertus Henri Listyanto Nugroho, S, ;	Albertus Henri Listyanto Nugroho, S, ;
jumlah_hadir	21	20
link	https://rps.ukdw.ac.id/archives/0_20241_AK1113_A.pdf	https://rps.ukdw.ac.id/archives/0_20241_AK1113_A.pdf

The collected 25685 rows of data from the odd semester of 2024, the even semester of 2023, and the odd semester of 2023. The details of the statistical data from the dataset profile are presented in Table 5.

Table 5. Dataset profile

Department name	Total data count	Number of courses	Yes Label	No Label
Humaniora	885 data (blank: 4 bpp 269 rps)	10	387	478
Akuntansi	2332 data (blank: 46 bpp 327 rps)	63	1720	612
Arsitektur	2587 data (blank: 171 bpp 286 rps)	43	2076	511
Biologi	1526 data (blank: 29 bpp 394 rps)	51	750	775
Desain Produk	1143 data (blank: 41 bpp 126 rps)	40	813	330
Filsafat Keilahian	2419 data (blank: 22 bpp 218 rps)	99	2002	417

Informatika	4387 data (blank 507 bpp 724 rps)	87	2567	1820
Manajemen	3784 data (blank: 137 bpp 826 rps)	67	2107	1627
Pasca Teologi	91 data (blank: 0 bpp 56 rps)	6	29	62
Pendidikan Bahasa Inggris	1645 data (blank 37 bpp 275 rps)	49	1451	194
Sistem Informasi	1814 data (blank: 11 bpp 375 rps)	43	1296	518
Humanitas	1192 data (blank 10 bpp 145 rps)	19	582	610
PPB	1640 data (blank 161 bpp 1429 rps)	12	367	1273
Puspindika	218 data (blank 2 bpp 102 rps)	9	91	127

The results are summarized in Table 6. It shows that there are many blank rows, mostly in the RPS documents. This means that some lecturers did not fill the RPS document properly, with an average of 27,47%. There were blank entries in the BPP document as well, but the percentage was lower (3.21%).

Table 6. EDA from the Dataset

Department name	**Total data count**	**BPP blank entries (%)**	**RPS blank entries (%)**	**Yes Label (%)**	**No Label (%)**
Humaniora	885 data (blank: 4 bpp 269 rps)	0,45	30,40	43,73	54,01
Akuntansi	2332 data (blank: 46 bpp 327 rps)	1,97	14,02	73,76	26,24
Arsitektur	2587 data (blank: 171 bpp 286 rps)	6,61	11,06	80,25	19,75
Biologi	1526 data (blank: 29 bpp 394 rps)	1,90	25,82	49,15	50,79
Desain Produk	1143 data (blank: 41 bpp 126 rps)	3,59	11,02	71,13	28,87
Filsafat Keilahian	2419 data (blank: 22 bpp 218 rps)	0,91	9,01	82,76	17,24
Informatika	4387 data (blank 507 bpp 724 rps)	11,56	16,50	58,51	41,49
Manajemen	3784 data (blank: 137 bpp 826 rps)	3,62	21,83	55,68	43,00
Pasca Teologi	91 data (blank: 0 bpp 56 rps)	0,00	61,54	31,87	68,13
Pendidikan Bahasa Inggris	1645 data (blank 37 bpp 275 rps)	2,25	16,72	88,21	11,79
Sistem Informasi	1814 data (blank: 11 bpp 375 rps)	0,61	20,67	71,44	28,56
Humanitas	1192 data (blank 10 bpp 145 rps)	0,84	12,16	48,83	51,17
PPB	1640 data (blank 161 bpp 1429 rps)	9,82	87,13	22,38	77,62
Puspindika	218 data (blank 2 bpp 102 rps)	0,92	46,79	41,74	58,26
RATA-RATA		**3,21786**	**27,4764**	**58,5314**	**41,2086**

3.2. Dataset Labeling

After obtaining the dataset, the next step was to label it with two possible entries: YES and NO. The Yes label indicates that the material contents matched between the RPS and BPP. No label indicates that it does not match the criteria. After one month of manual work in labeling those data, obtained the following results: the Yes label had an average of 58,5% from all datasets, while the No label had an average of 41,3%. From these results, it can conclude that the level of conformity between the RPS and BPP is approximately 58.5%. This result was then compared with the automation system using various algorithms described in the previous section.

3.2. Data Cleaning

Data cleaning was performed by removing all extra white spaces and unknown characters and changing them into a single space. It also merged three columns in the BPP materials into a single column, whereas in the RPS, merged two columns, namely, the material and sub-CPMK. This is because in the BPP document, some lecturers often do not implement the course based on their planning document and overwrite it with new text that is manually inputted, while for RPS, some lecturers often use sub CPMK instead of the material content.

Once all the data were cleaned and verified, multiple algorithms were implemented. Once the dataset was completed and labeled, it was saved in XLSX format and became the main dataset, which was processed in the next phase, that is, conformity detection, using the planned algorithms. To process these files, Python libraries are used, including OpenPyXL and Pandas.

3.3. Development Phase

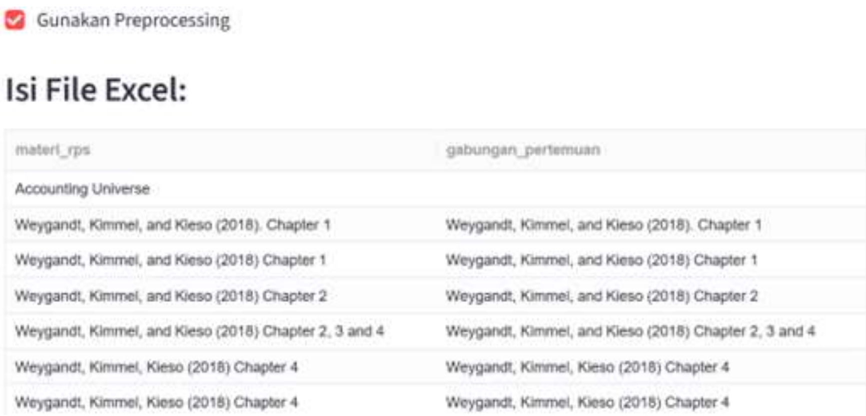
In the development phase, a program using Python is developed. The system is divided into several modules.

1. Dataset processing
   This module uses OpenPyXL (https://openpyxl.readthedocs.io/en/stable/) and Pandas (https://pandas.pydata.org/) to read, parse, and load XLSX files into memory. The final data were placed in a Pandas DataFrame format.
2. Pre-processing
3. For preprocessing, several operations are performed: multiple space removal, case folding conversion, punctuation mark removal, word normalization, and repetitive normalization.
4. Text similarity algorithm.
   Several algorithms were implemented in the functions.
   - Boyer-Moore (B), Rabin-Karp (R), Jaccard (JC), Jaro-Winkler (JW), Smith-Waterman (SW), Knuth Morris Prat (K), Lavenstein distance (L), Jaro (J), Dice (D), Soundex (S), and cosine similarity (C).
   - A function was created to call all the algorithms using the input from the dataset. The output of this function was saved in XLSX format so that it could be further processed for similarity percentage, along with its accuracy.
5. Metrics evaluation calculation for each algorithm, which measures the accuracy level for each data label, and saves the results.
6. Web-based publishing of the system.
7. Pipeline to upload to the GitHub repository for the source code management.
8. Streamlit repository setup for web access.
9. Desktop application as an alternative to a client application.

The front page of the system is presented in Figure 1. Users can upload RPS and BPP in XLSX format, and they will be processed by the system, as shown in Figure 2.

.

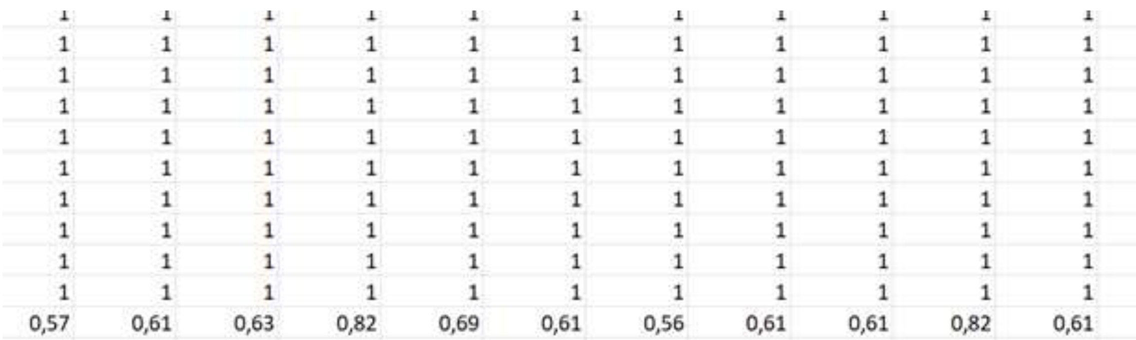
Figure 1. Front page of the system



materi_rps	gabungan_pertemuan
Accounting Universe	
Weygandt, Kimmel, and Kieso (2018). Chapter 1	Weygandt, Kimmel, and Kieso (2018). Chapter 1
Weygandt, Kimmel, and Kieso (2018) Chapter 1	Weygandt, Kimmel, and Kieso (2018) Chapter 1
Weygandt, Kimmel, and Kieso (2018) Chapter 2	Weygandt, Kimmel, and Kieso (2018) Chapter 2
Weygandt, Kimmel, and Kieso (2018) Chapter 2, 3 and 4	Weygandt, Kimmel, and Kieso (2018) Chapter 2, 3 and 4
Weygandt, Kimmel, Kieso (2018) Chapter 4	Weygandt, Kimmel, Kieso (2018) Chapter 4
Weygandt, Kimmel, Kieso (2018) Chapter 4	Weygandt, Kimmel, Kieso (2018) Chapter 4

Figure 2. The user uploaded an Excel document to the system.

Once it has been processed, the conformity level between the RPS and BPP is shown along with its accuracy, as shown in Figure 3.



1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
0,57	0,61	0,63	0,82	0,69	0,61	0,56	0,61	0,61	0,82	0,61

Figure 3. Conformity level from the XLS file and RPS/BPP accuracy

Using the data analysis results, it gained insights into the last three semesters from the RPS and BPP documents.

1. Some lecturers did not make any RPS documents, or they were late in their submission. Another possibility is that it is a new course, so it was entered after the submission date for the RPS had passed.
2. There are 3% blank rows in the BPP documents and 27.4% in the RPS documents. It is also found that there were 58,5% matched rows and 41,5% unmatched rows. Therefore,

the conformity level using human manual verification for the RPS and BPP in the last three semesters was approximately 60%.

3. There are a lot of lecturers who did not write the RPS document properly:
    a. Some did not enter the material contents
    b. Some only fill in the materials with reference materials
4. The lecturers did not fill in the BPP form properly.
    a. A lot of them did not fill in the materials contents
    b. There were many differences between RPS planning and its implementation in the BPP.

3.4. Evaluation

The accuracy and conformity with a predefined threshold were measured. The results were compared with those of the given labels. Table 7 lists the parameters used for this evaluation.

Table 7. Evaluation environment parameters

Environment Parameter	Value / Information
Reference data count	25685
Golden Dataset	Yes: 16250, No: 9433
Scenario	Without Preprocessing, With Preprocessing, Single row, Combined row, using Threshold 40, 50, 65, 85%.
Number of Algorithms	11 algorithms (Boyer-Moore (B), Rabin-Karp (R), Jaccard (JC), Jaro-Winkler (JW), Smith-Waterman (SW), Knuth Morris Prat (K), Lavenstein distance (L), Dice (D), Jaro (J), Soundex (S), and cosine similarity (C))

Based on the system developed, the conformity and accuracy outputs are based on several algorithms with and without pre-processing (Tables 8 and 9) and with pre-processing (Tables 10 and 11, respectively). These algorithms were tested using a 50% threshold value. These algorithms can be categorized as exact algorithms, such as Boyer-Moore (B), Knuth-Morris-Pratt (K), Rabin-Karp (R), and Soundex (S) and fuzzy algorithms such as Jaccard (JC), Dice (D), Cosine (C), Jaro (J), Jaro-Winkler (JW), Smith-Waterman (SW), and Levenshtein (L).

The main reason to used a variety of algorithms was the non-existence of a single algorithm that works for short-to-mid text, especially for RPS and BPP documents, which also include a mix of both English and Indonesian. The contents of the RPP and BPP documents vary greatly depending on each lecturer's style; therefore, an appropriate algorithm is needed for the text matching process. By comparing multiple algorithms with different scenarios, we aim to obtain the best results from multiple algorithms.

Table 8. Conformity percentage between RPS/BPP without pre-processing (Threshold 50%)

Jaccard	**Boyer Moore**	**Rabin Karp**	**Jaro Winkler**	**Smith Waterman**	**KMP**	**Levenshtein**	**Cosine similarity**	**Dice**	**Jaro**	**Soundex**
43,54	55,7	49,22	**64,76**	60,09	53,7	45,38	46,6	48,37	62,4	56,66

Table 9. Accuracy between RPS/BPP without pre-processing (between 0 - 100)

Jaccard	**Boyer Moore**	**Rabin Karp**	**Jaro Winkler**	**Smith Waterman**	**KMP**	**Levenshtein**	**Cosine similarity**	**Dice**	**Jaro**	**Soundex**
77	80	81	**85**	**87**	81	80	82	83	**85**	81

Table 10. Conformity percentage between RPS/BPP with pre-processing (Threshold 50%)

Jaccard	**Boyer Moore**	**Rabin Karp**	**Jaro Winkler**	**Smith Waterman**	**KMP**	**Levenshtein**	**Cosine similarity**	**Dice**	**Jaro**	**Soundex**
44,37	56,07	49,08	**65,27**	60,45	55,08	45,66	46,45	49,1	62,81	57,68

Table 11. Accuracy between RPS/BPP with pre-processing (between 0 - 100)

Jaccard	Boyer Moore	Rabin Karp	Jaro Winkler	Smith Waterman	KMP	Levenshtein	Cosine similarity	Dice	Jaro	Soundex
78	81	81	85	**88**	81	75	82	83	85	82

Furthermore, more test scenarios conducted without pre-processing with multiple thresholds of 40%, 50%, 65%, and 85%, using data from a single row (TS40, TS50, TS65, TS85), with pre-processing with the same threshold (PS40, PS50, PS65, PS85), and all combined data from multiple rows without pre-processing (TG40, TG50, TG65, TG85), and with pre-processing (PG40, PG50, PG65, and PG85). The results are presented in Table 12. The highlights are as follows:

- TS scenario (without pre-processing, single row): Smith-Waterman has the highest accuracy, 0,845, followed by Jaro-Winkler (0,840) and Jaro (0,8125). It can be seen that exact algorithms (Boyer-Moore, Knuth-Morris-Pratt, Rabin-Karp, and Soundex) have stable accuracy but lower than that of the fuzzy algorithm.
- PS scenario (with pre-processing, single row): Smith-Waterman has the highest accuracy, 0,870, followed by Jaro Winkler (0,850) and Jaro (0,8225). Pre-processing had a significant effect on Smith-Watterman and Jaccard (both accuracy increased by +2,5%).
- TG (without pre-processing, combined): Soudex has an accuracy of 0,865, similar to Boyer-Moore (0,8625), similar to Knuth-Morris-Prat (0,8625), followed by Rabin-Karp (0,8475). The combined scenario strengthens the exact/phonetic algorithms, whereas the fuzzy algorithm weakens them.
- PG (with pre-processing, combined): Boyer-Moore / Knuth-Morris-Prat / Soundex has 0,8725 accuracy (the top three with the highest and most robust scores), followed by Rabin Karp (0,850), and Cosine similarity (0,830).

Meanwhile, the impacts of the pre-processing and combined scenarios are as follows:

- Pre-processing and TS scenario (PS−TS): Swith-Waterman's accuracy is increased by +0.025, Jaccard increased by +0.025; Jaro / Jaro-Winkler/ Dice increased by +~0.01; Boyer Moore / Knuth-Morris-Prat / Rabin-Karp / Soundex are increased by -0,01, Cosine -0,012. Pre-processing is more useful for fuzzy algorithms than for precise algorithms.
- Combined scenario (in TG−TS): Boyer Moore +0.070, Knuth-Morris-Prat +0.065, Rabin-Karp +0.068, Soundex +0.060, but decreased for Smith-Waterman -0.142, Jaro-0.115, Jaro Winkler -0.097, and Jaccard -0.085. The combined scenario was found to be beneficial in the exact/phonetic algorithm but not in the fuzzy algorithm.
- Pre-processing in the combined scenario (PG−TG): improved in Jaro / Jaro Winkler / Dice / Jaccard (+0.043 - 0.050), but not significant in the exact algorithm (~+0.01).

From the comprehensive analysis of all algorithms used, several things: for a single-row scenario (TS/PS), it is recommended to use Smith-Waterman or Jaro-Winkler when there is a lot of noise or typos in the data, and it can also use pre-processing for consistency. For the combined scenario (TG/PG), Boyer-Moore, Knuth-Morris-Pratt, or Soundex should be used for fast and consistent search results when the data are merged. The best threshold is between 40% and 50%; for the Jaro-Winkler or Jaro algorithm, it can consider increasing the threshold up to 65%. The cosine (C) and Levenshtein (L) algorithms exhibited moderate performance; however, they were sensitive to the threshold; therefore, they were suitable for use as a baseline.

Table 12. The accuracy results from all test scenarios

Algorithm	Accuracy															
	TS40	TS50	TS65	TS85	PS40	PS50	PS65	PS85	TG40	TG50	TG65	TG85	PG40	PG50	PG65	PG85
Jaccard	0,81	0,71	0,73	0,64	0,82	0,78	0,74	0,65	0,78	0,71	0,6	0,46	0,82	0,77	0,67	0,49
Boyer	0,83	**0,87**	0,77	0,7	0,83	0,81	0,78	0,71	**0,87**	**0,87**	**0,87**	0,84	**0,87**	**0,88**	**0,88**	0,86
Rabin	0,84	0,85	0,77	0,66	0,84	0,81	0,76	0,67	0,84	0,85	0,85	0,85	0,84	0,86	0,85	0,85

Jaro_winkler	0,84	0,82	**0,88**	0,82	0,84	0,85	**0,88**	0,83	0,68	0,82	0,85	0,62	0,84	0,84	0,85	0,62
Smith_waterman	**0,88**	0,78	**0,87**	**0,85**	**0,88**	**0,88**	0,87	**0,85**	0,8	0,78	0,7	0,53	0,81	0,79	0,71	0,53
Kmp	0,84	**0,87**	0,77	0,71	0,84	0,81	0,78	0,72	**0,87**	**0,87**	**0,87**	0,84	**0,87**	**0,88**	**0,88**	0,86
Levenshtein	0,81	0,78	0,69	0,6	0,81	0,75	0,69	0,6	0,82	0,78	0,65	0,49	0,83	0,78	0,64	0,49
Cosine	0,84	0,86	0,77	0,67	0,84	0,82	0,77	0,66	**0,87**	0,86	0,85	0,73	**0,87**	0,87	0,85	0,73
Dice	0,85	0,81	0,79	0,69	0,85	0,83	0,8	0,7	0,85	0,81	0,73	0,53	**0,87**	0,84	0,78	0,6
Jaro	0,84	0,82	**0,87**	0,72	0,84	0,85	**0,88**	0,72	0,68	0,82	0,83	0,46	0,84	0,84	0,83	0,46
Soundex	0,84	**0,87**	0,79	0,72	0,84	0,82	0,79	0,73	0,86	**0,87**	**0,87**	**0,86**	0,86	**0,88**	**0,88**	**0,87**
AVG	**0,83818**	0,821818	0,790909	0,707273	**0,839091**	0,819091	0,794545	0,712727	0,810909	**0,821818**	0,788182	0,655455	**0,847273**	0,839091	0,801818	0,669091

Based on the test results shown in Table 12, the algorithm performance is influenced by the characteristics of the RPS and BPP texts themselves. Exact matching algorithms, such as Boyer-Moore (B) and Knuth-Morris-Pratt (K), demonstrated the highest and most stable accuracy in the combined text (TG/PG) scenario, as they could identify phrases and text that consistently appeared between lesson plans and implementations in long texts. The Rabin-Karp (R) algorithm performed fairly well, but slightly lower due to its sensitivity to small changes in the text. Conversely, token-based and character distance algorithms such as Jaccard (J), Dice (D), and Levenshtein (L) produced lower accuracy because they were less able to represent similar meanings when there were text variations, additional words, or differences in sentence structure between the RPS and the BPP.

However, in the short text scenario for a single text without pre-processing, fuzzy similarity algorithms such as Jaro-Winkler (JW) and Smith-Waterman (SW) performed better because they were able to tolerate typos and text variations common in BPP. The Soundex (S) and cosine similarity (C) algorithms provide fairly stable results, especially in handling spelling variations and general topic similarities, but they are not optimal without semantic pre-processing. Based on the overall results, Boyer Moore (B) and Knuth-Morris-Pratt (K) are recommended as algorithms used in the implementation of RPS systems on combined texts, while Jaro Winkler (JW) or Smith Waterman (SW) are more suitable for analysis in single short texts scenario, so that the combination approach is the most effective solution for learning evaluation needs according to the case study.

Also, based on the test results, the algorithm's sensitivity to the data length significantly impacted the accuracy level in the comparison between RPS and BPP. Exact matching algorithms, such as Boyer-Moore (B) and Knuth-Morris-Pratt (K), demonstrated optimal performance as the text length increased. In combined text scenarios (TG/PG), these two algorithms exploited recurring text occurrences and consistent text terms, resulting in increased accuracy and more stable results. Conversely, for short texts and single texts, the performance of these algorithms was more sensitive to text variations, as even small character differences could immediately lead to a match failure.

Conversely, fuzzy-based algorithms such as Jaro-Winkler (JW) and Smith-Waterman (SW) demonstrated more stable performance on short texts, as they were designed to handle small differences, typos, and variations in character order. For longer texts, the sensitivity of these algorithms decreases because of the increased computational complexity and diminishing influence of local similarity on the overall similarity score. Token and vector-based algorithms such as Jaccard (J), Dice (D), and cosine similarity (C) demonstrated intermediate sensitivity to text length; their performance was relatively stable on long texts but declined on short texts owing to the limited number of tokens available for comparison. This means that the choice of algorithm must be adjusted to the length of the data, where exact matching algorithms are more effective

.

for long texts, whereas fuzzy-based algorithms are more suitable for short texts that contain text variations.

4. CONCLUSION

The text similarity methods are implemented to evaluate the consistency level between the RPS and BPP documents for each semester separately. This system can help reduce the verification process in terms of time and increase overall efficiency. It only takes approximately 15 min to evaluate all running classes in one semester. In the TS/PS scenarios, SW (0.845 -0 .870) and JW (0.840 - 0.850) were top performers. In the TG/PG scenarios, B/K/S (~0.8625 - 0.8725) achieved the best results with excellent stability (std ~0.005–0.015), whereas pre-processing yields significant gains for SW and Jaccard (JC), whereas aggregation benefits exact/phonetic algorithms but harms fuzzy similarity metrics. For production use, PG + B/K/S and PS + SW/JW provided high and stable accuracies. Finally, for future work, it is planned to expand the evaluation to use an ensemble algorithm (from exact and fuzzy algorithms) and adaptive threshold tuning.

5. ACKNOWLEDGMENTS

This work was supported by the Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana, Yogyakarta, Indonesia, via the LPPM UKDW Research Scheme with Grant No. 121/D.02/LPPM/202.

REFERENCES

[1] S. S. Kusumawardani *et al.*, *Panduan Penyusunan Kurikulum Pendidikan Tinggi Mendukung Merdeka Belajar-Kampus Merdeka Menuju Indonesia Emas*, V (2024). Jakarta: Direktorat Jenderal Pendidikan Tinggi, Riset, dan Teknologi Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi, 2024.

[2] A. Chrismanto *et al.*, *Panduan Penyusunan Kurikulum Berorientasi KKNI dengan Pendekatan Pendidikan Berbasis Capaian (Outcome-based Education/OBE)*. Yogyakarta: Universitas Kristen Duta Wacana, 2024.

[3] LPAIP, *Panduan Penyusunan Kurikulum Berorientasi KKNI Universitas Kristen Duta Wacana*, 1st ed. Yogyakarta: Lembaga Pengembangan Akademik dan Inovasi Pembelajaran Universitas Kristen Duta Wacana, 2016. doi: QADW-1200-PA-16.030.001.

[4] M. AL-Smadi, Z. Jaradat, M. AL-Ayyoub, and Y. Jararweh, "Paraphrase identification and semantic text similarity analysis in Arabic news tweets using lexical, syntactic, and semantic features," *Inf Process Manag*, vol. 53, no. 3, pp. 640–652, 2017, doi: 10.1016/j.ipm.2017.01.002.

[5] A. Debnath, N. Pinnaparaju, M. Shrivastava, V. Varma, and I. Augenstein, "Semantic Textual Similarity of Sentences with Emojis," in *Companion Proceedings of the Web Conference 2020*, New York, NY, USA: ACM, Apr. 2020, pp. 426–430. doi: 10.1145/3366424.3383758.

[6] G. Majumder, P. Pakray, A. Gelbukh, and D. Pinto, "Semantic Textual Similarity Methods, Tools, and Applications: A Survey," *Computación y Sistemas*, vol. 20, no. 4, pp. 647–665, Dec. 2016, doi: 10.13053/cys-20-4-2506.

.

[7] T. A. Firdaus, R. H. Putra, F. Arifandi, M. K. Anam, and L. Lathifah, "Implementasi Sistem Rencana Pembelajaran Semester Berbasis Web ntuk Mempermudah Proses Pembelajaran," *Jurnal Teknoinfo*, vol. 17, no. 1, pp. 156–169, Jan. 2023, doi: 10.33365/JTI.V17I1.2348.

[8] D. Sebastian, A. Chrismanto, and W. Raharjo, "Implementasi Algoritma Okapi Bm25 dan K-Means untuk Mencari Relevansi Artikel Pada Beberapa Situs Berita," Yogyakarta: Universitas Kristen Marantha Bandung, 2011. Accessed: Jan. 16, 2025. [Online]. Available:https://www.researchgate.net/publication/263814545_IMPLEMENTASI_ALGORITMA_OKAPI_BM25_DAN_K-MEANS_UNTUK_MENCARI_RELEVANSI_ARTIKEL_PADA_BEBERAPA_SITUS_BERITA

[9] A. R. Chrismanto, A. K. Sari, and Y. Suyanto, "Critical Evaluation On Spam Content Detection in Social Media," *Journal of Theoretical and Applied Information Technology (JATIT)*, vol. 100, no. 8, pp. 2642–2667, 2022, [Online]. Available: http://www.jatit.org/volumes/Vol100No8/29Vol100No8.pdf

[10] L. Wang, L. Zhang, and J. Jiang, "Duplicate Question Detection With Deep Learning in Stack Overflow," *IEEE Access*, vol. 8, pp. 25964–25975, 2020, doi: 10.1109/ACCESS.2020.2968391.

[11] A. R. Lahitani, "Automated Essay Scoring menggunakan Cosine Similarity pada Penilaian Esai Multi Soal," *Jurnal Kajian Ilmiah*, vol. 22, no. 2, pp. 107–118, May 2022, doi: 10.31599/JKI.V22I2.1121.

[12] R. Delima, A. Rachmat, and C. #2, "Otomatisasi Pembentukan Class Diagram dengan Pendekatan Metode Pemrosesan Teks dan Algoritma CombineTF," *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, vol. 10, no. 1, pp. 120–127, Apr. 2024, doi: 10.26418/JP.V10I1.72518.

[13] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 3980–3990. doi: 10.18653/v1/D19-1410.

[14] A. Talman, A. Yli-Jyrä, and J. Tiedemann, "Sentence embeddings in NLI with iterative refinement encoders," *Nat Lang Eng*, vol. 25, no. 4, pp. 467–482, Jul. 2019, doi: 10.1017/S1351324919000202.

[15] Haihua Chen, Lei Wu, Jiangping Chen, Wei Lu, Junhua Ding, A comparative study of automated legal text classification using random forests and deep learning, Information Processing & Management, Volume 59, Issue 2, 2022, 102798, ISSN 0306-4573, https://doi.org/10.1016/j.ipm.2021.102798.

[16] Yang X, He X, Zhang H, Ma Y, Bian J, Wu Y, Measurement of Semantic Textual Similarity in Clinical Texts: Comparison of Transformer-Based Models, JMIR Med Inform 2020;8(11):e19735, doi: 10.2196/19735

[17] C. P. Chai, "Comparison of text pre-processing methods," *Nat Lang Eng*, vol. 29, no. 3, pp. 509–553, May 2023, doi: 10.1017/S1351324922000213.

.

[18] A. Jabbar, S. Iqbal, M. I. Tamimy, A. Rehman, S. A. Bahaj, and T. Saba, "An Analytical Analysis of Text Stemming Methodologies in Information Retrieval and Natural Language Processing Systems," *IEEE Access*, vol. 11, pp. 133681–133702, 2023, doi: 10.1109/ACCESS.2023.3332710.

[19] Rianto, A. B. Mutiara, E. P. Wibowo, and P. I. Santosa, "Improving the accuracy of text classification using stemming method, a case of non-formal Indonesian conversation," *J Big Data*, vol. 8, no. 1, pp. 1–16, Dec. 2021, doi: 10.1186/S40537-021-00413-1/FIGURES/6.

[20] R. Feldman and J. Sanger, *The Text Mining Handbook*. Cambridge University Press, 2006. doi: 10.1017/cbo9780511546914.

[21] C. C. Aggarwal, "Data Mining: The Textbook," *Springer International Publishing*, pp. 285–344, 2015.

[22] M. Kranti and V. Ghag, "Comparative Analysis of Effect of Stopwords Removal on Sentiment Classification," *IEEE International Conference on Computer, Communication and Control (IC4-2015)*, pp. 2–7, 2015.

[23] D. Sebastian and K. A. Nugraha, "Text normalization for Indonesian abbreviated word using crowdsourcing method," *2019 International Conference on Information and Communications Technology, ICOIACT 2019*, pp. 529–532, 2019, doi: 10.1109/ICOIACT46704.2019.8938463.

[24] N. Hanafiah, A. Kevin, C. Sutanto, Fiona, Y. Arifin, and J. Hartanto, "Text Normalization Algorithm on Twitter in Complaint Category," *Procedia Comput Sci*, vol. 116, pp. 20–26, 2017, doi: 10.1016/j.procs.2017.10.004.

[25] S. Dutta, T. Saha, S. Banerjee, and S. K. Naskar, "Text normalization in code-mixed social media text," *2015 IEEE 2nd International Conference on Recent Trends in Information Systems, ReTIS 2015 - Proceedings*, no. c, pp. 378–382, 2015, doi: 10.1109/ReTIS.2015.7232908.

[26] R. S. Boyer and J. S. Moore, "A fast string searching algorithm," *Commun ACM*, vol. 20, no. 10, pp. 762–772, Oct. 1977, doi: 10.1145/359842.359859.

[27] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. MIT Press, 2009. Accessed: Jan. 18, 2025. [Online]. Available: https://www.amazon.com/Introduction-Algorithms-3rd-MIT-Press/dp/0262033844

[28] R. M. Karp and M. O. Rabin, "Efficient Randomized Pattern-Matching Algorithms.," *IBM J Res Dev*, vol. 31, no. 2, pp. 249–260, 1987, doi: 10.1147/RD.312.0249.

[29] P. Jaccard, "Etude de la distribution florale dans une portion des Alpes et du Jura," in *Bulletin de la Societe Vaudoise des Sciences Naturelles 37 (142)*, 1901, pp. 547–579. Accessed: Jan. 18, 2025. [Online]. Available: https://www.researchgate.net/publication/225035806_Etude_de_la_distribution_florale_dans_une_portion_des_Alpes_et_du_Jura

[30] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of Massive Datasets*, 3rd ed. Cambridge University Press, 2020. Accessed: Jan. 18, 2025. [Online]. Available:

https://www.amazon.com/Mining-Massive-Datasets-Jure-Leskovec-dp-1108476341/dp/1108476341

[31] M. A. Jaro, "Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida," *J Am Stat Assoc*, vol. 84, no. 406, p. 414, Jun. 1989, doi: 10.2307/2289924.

[32] O. Rozinek and J. Mares, "Fast and Precise Convolutional Jaro and Jaro-Winkler Similarity," *Conference of Open Innovation Association, FRUCT*, pp. 604–613, 2024, doi: 10.23919/FRUCT61870.2024.10516360.

[33] W. E. Winkler, "String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage.," 1990.

[34] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences," *J Mol Biol*, vol. 147, no. 1, pp. 195–197, Mar. 1981, doi: 10.1016/0022-2836(81)90087-5.

[35] D. E. Knuth, Jr. James H. Morris, and V. R. Pratt, "Fast Pattern Matching in Strings," *https://doi.org/10.1137/0206024*, vol. 6, no. 2, pp. 323–350, Jul. 2006, doi: 10.1137/0206024.

[36] "Levenshtein Distance: Information theory, Computer science, String (computer science), String metric, Damerau?Levenshtein distance, Spell checker, Hamming distance | Guide books | ACM Digital Library." Accessed: Aug. 20, 2025. [Online]. Available: https://dl.acm.org/doi/10.5555/1822502

[37] A. Carass *et al.*, "Evaluating White Matter Lesion Segmentations with Refined Sørensen-Dice Analysis," *Scientific Reports 2020 10:1*, vol. 10, no. 1, pp. 1–19, May 2020, doi: 10.1038/s41598-020-64803-w.

[38] D. Pinto, D. Vilariño, Y. Alemán, H. Gómez, N. Loya, and H. Jiménez-Salazar, "The Soundex Phonetic Algorithm Revisited for SMS Text Representation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7499 LNAI, pp. 47–55, 2012, doi: 10.1007/978-3-642-32790-2_5.

[39] G. Salton, *Automatic Text Processing: The Transformation Analysis and Retrieval of Information by Computer*, Hardcover Edition. Addison-Wesley, 1989. Accessed: Jan. 19, 2025. [Online]. Available: https://www.amazon.com/Automatic-Text-Processing-Transformation-Addison-Wesley/dp/0201122278

[40] 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, Apr. 2021, doi: 10.1007/S40031-020-00501-5/METRICS.

[41] H. Khatter, N. Goel, N. Gupta, and M. Gulati, "Movie Recommendation System using Cosine Similarity with Sentiment Analysis," *Proceedings of the 3rd International Conference on Inventive Research in Computing Applications, ICIRCA 2021*, pp. 597–603, Sep. 2021, doi: 10.1109/ICIRCA51532.2021.9544794.

[42] A. Widianto, E. Pebriyanto, F. Fitriyanti, and M. Marna, "Document Similarity Using Term Frequency-Inverse Document Frequency Representation and Cosine Similarity,"

.

Journal of Dinda : Data Science, Information Technology, and Data Analytics, vol. 4, no. 2, pp. 149–153, Aug. 2024, doi: 10.20895/DINDA.V4I2.1589.

[43] A. Srivastava and M. Sahami, "Text Mining: Classification, Clustering, and Applications (Chapman & Hall/CRC Data Mining and Knowledge Discovery Series)," p. 328, 2009, Accessed: Sep. 03, 2021. [Online]. Available: http://www.amazon.com/Text-Mining-Classification-Clustering-Applications/dp/1420059408

[44] S. M. Weiss, N. Indurkhya, T. Zhang, and F. J. Damerau, *Text mining: Predictive methods for analyzing unstructured information*. Springer New York, 2005. doi: 10.1007/978-0-387-34555-0.

[45] A. R. Chrismanto, A. Afiahayati, Y. Sari, A. K. Sari, and Y. Suyanto, "Spam Comments Detection on Instagram Using Machine Learning and Deep Learning Methods," *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, vol. 13, no. 1, p. 46, Aug. 2022, doi: 10.24843/LKJITI.2022.v13.i01.p05.