JTBC: Jurnal Teknologi dan Bisnis Cerdas
Volume 2 Nomor 2, Juni 2026
e-ISSN : 3109-8924
DOI: 10.64476/jtbc.v2i2.69

DESIGN AND CONSTRUCTION OF A WEBSITE-BASED DIGITAL DOCUMENT CHECKING APPLICATION FOR STUDENT ASSIGNMENTS
RANCANG BANGUN APLIKASI PENGECEKAN DOKUMEN DIGITAL TUGAS MAHASISWA BERBASIS WEBSITE

Haeruddin[1]*, Gunawan Putra Wahdana[2], Dhimas Tribuana[3], Dayanti[4], Nur Inda[5]
Universitas Prof. DR. H. M. Arifin Sallatang[1,2], Akademi Sekretari Manajemen Publik[2], Universitas Patria Artha[3], Institut Teknologi dan Bisnis Muhammadiyah Polewali Mandar[5]
ayankkamasee@gmail.com[1], me.gunawanputra@gmail.com[2], d.tribuana@gmail.com[3], dayanti.fattah@gmail.com[4], nurinda@itbmpolman.ac.id[5]

ABSTRACT

The development of information technology has encouraged the submission of student assignments to shift from physical documents to digital documents. However, this convenience also creates new challenges, such as unstructured assignment archiving and the increasing potential for document similarity among students. This study aims to design and develop a website-based digital document checking application for student assignments that can manage documents and automatically display similarity percentages between files. The system was developed using the Waterfall method through requirement analysis, system design, implementation, testing, and evaluation stages. The technologies used include HTML, CSS, PHP, JavaScript, MySQL, XAMPP, and Sublime Text. The document checking process consists of document upload, text extraction, text preprocessing, word weighting using TF-IDF, and similarity measurement using the cosine similarity method. The data used consisted of 165 student assignment files, comprising 79 dataset files and 86 test files. Testing on five sample documents showed that the system was able to display various similarity percentages of 5%, 12%, 45%, 88%, and 92%, categorized as original, plagiarism indication, and high plagiarism. These results indicate that the application can assist lecturers in conducting an initial examination of similarity in student assignment documents. Therefore, the developed application has the potential to support more effective academic document management and improve academic integrity in higher education environments.

Keywords: *digital document, student assignment, website, TF-IDF, cosine similarity, plagiarism.*

ABSTRAK

Perkembangan teknologi informasi mendorong proses pengumpulan tugas mahasiswa beralih dari dokumen fisik ke dokumen digital. Namun, kemudahan tersebut menimbulkan tantangan baru, seperti pengelolaan arsip tugas yang kurang terstruktur dan meningkatnya potensi kemiripan dokumen antar mahasiswa. Penelitian ini bertujuan untuk merancang dan membangun aplikasi pengecekan dokumen digital tugas mahasiswa berbasis website yang mampu mengelola dokumen serta menampilkan persentase kemiripan antarfile secara otomatis. Sistem dikembangkan menggunakan metode Waterfall melalui tahapan analisis kebutuhan, perancangan, implementasi, pengujian, dan evaluasi. Teknologi yang digunakan meliputi HTML, CSS, PHP, JavaScript, MySQL, XAMPP, dan Sublime Text. Proses pengecekan dokumen dilakukan melalui tahapan unggah dokumen, ekstraksi teks, text preprocessing, pembobotan kata menggunakan TF-IDF, serta pengukuran kemiripan menggunakan metode cosine similarity. Data yang digunakan terdiri atas 165 file tugas mahasiswa, yaitu 79 file dataset dan 86 data uji. Hasil pengujian terhadap lima sampel dokumen menunjukkan bahwa sistem mampu menampilkan variasi persentase kemiripan sebesar 5%, 12%, 45%, 88%, dan 92%, dengan kategori orisinal, indikasi plagiat, dan plagiat tinggi. Hasil tersebut menunjukkan bahwa aplikasi dapat membantu dosen dalam melakukan pemeriksaan awal terhadap kemiripan dokumen tugas mahasiswa. Dengan demikian, aplikasi ini berpotensi mendukung pengelolaan dokumen akademik secara lebih efektif serta meningkatkan integritas akademik di lingkungan perguruan tinggi.

Kata Kunci: dokumen digital, tugas mahasiswa, website, TF-IDF, *cosine similarity*, plagiarisme.

INTRODUCTION

The development of information technology has encouraged significant changes in the learning process in higher education. Academic activities that were previously carried out conventionally have increasingly shifted into digital forms, including reference searching, assignment preparation, document submission, and the evaluation of student learning outcomes. Easy access to digital information sources provides major benefits for students because it allows them to obtain learning materials from various sources quickly and flexibly. However, on the other hand, this convenience also creates new challenges in maintaining academic integrity, particularly regarding the increasing potential for document similarity and plagiarism in student assignments.

Plagiarism is one form of academic misconduct that can disrupt the quality of learning and reduce trust in students' scientific work. In higher education, plagiarism does not only occur due to the intentional copying of other people's work, but can also be caused by students' limited understanding of writing ethics, paraphrasing skills, and proper citation practices. Gregory and Leeman (2021) emphasized that perceptions of plagiarism in academic environments are often influenced by context, intention, and individual understanding of the boundaries between legitimate source use and academic misconduct. Therefore, plagiarism prevention is not sufficient if it only relies on sanctions, but also requires system support that can help identify document similarity objectively.

In recent years, academic plagiarism has become increasingly complex along with the growing use of digital technology and artificial intelligence in assignment writing. Balalle and Pannilage (2025) explained that artificial intelligence has two sides in the context of academic integrity: it can support the learning process, but it also has the potential to facilitate academic misconduct if not used ethically. Similarly, Bittle and El-Gayar (2025) showed that the development of Generative Artificial Intelligence encourages higher education institutions to review plagiarism detection strategies, academic policies, and mechanisms for ensuring the integrity of student assignments. Therefore, educational institutions need a system that not only stores documents digitally but also helps detect document similarity levels as an initial step in maintaining the originality of student assignments.

The management of student assignment documents in digital form also requires an effective system. Without a structured system, lecturers may experience difficulties in archiving documents, retrieving assignment files, and comparing similarities among student documents. Manual similarity checking requires a long time, especially when the number of documents to be examined is large. This condition provides an important reason for developing a website-based application that can help with uploading, storing, managing, and checking student assignment documents more efficiently.

One approach that is widely used in document similarity measurement is the combination of Term Frequency-Inverse Document Frequency (TF-IDF) and cosine similarity. TF-IDF is used to assign weight to each word based on its frequency in a document and its uniqueness within a document collection. Meanwhile, cosine similarity is used to calculate the level of similarity between documents based on text vector representation. Halim and Lasut (2024) developed a web-based document plagiarism detection application using TF-IDF and cosine similarity. The results of their study showed that the system process included preprocessing, TF-IDF calculation, and cosine similarity calculation, with test results that were consistent between manual calculation and the application. These findings indicate that TF-IDF and cosine similarity are relevant for application in website-based academic document similarity checking systems.

In addition to TF-IDF and cosine similarity, other studies have also shown that string matching-based approaches can be used to detect document similarity. Zachrias and Gunawan (2025) developed a document similarity classification system using the Winnowing algorithm with the Jaccard Coefficient approach. Their study showed that the system was able to provide relevant results for Indonesian-language documents and could be

used to support academic plagiarism detection. However, string matching-based approaches have limitations when dealing with documents that have been paraphrased or semantically modified. Therefore, research on document checking applications needs to continue developing by considering user needs, document types, and the similarity measurement methods used.

In the academic context, document checking applications should not be positioned as tools that absolutely determine plagiarism, but rather as tools that provide initial indicators of document similarity levels. Pudasaini et al. (2025) explained that the development of Large Language Models makes plagiarism detection more challenging because generated or modified texts can resemble human writing. Therefore, detection systems need to be supported by academic interpretation, institutional policy, and lecturer verification of document content context. Thus, document similarity percentage results should be understood as initial considerations, not as the sole basis for determining academic violations.

Based on these problems, this study aims to design and develop a website-based digital document checking application for student assignments. The application was developed to assist lecturers in managing assignment documents, extracting text, processing document content, and displaying similarity percentages between documents. The system was designed using the Waterfall software development method with HTML, CSS, PHP, JavaScript, MySQL, XAMPP, and Sublime Text technologies. The data used consisted of student assignment documents in digital form, totaling 165 files comprising 79 dataset files and 86 test files.

The contribution of this study lies in the development of a website-based application that can be used as an initial checking tool for measuring the similarity level of student assignment documents. This application is expected to improve the effectiveness of academic document management, assist lecturers in verifying student assignments, and support efforts to strengthen academic integrity in higher education. In addition, this study provides practical contributions in applying text-processing-based methods to support a simple, structured, and usable digital document checking system in the learning context.

METHODS

This study uses a research and development approach with the Waterfall software development model. The Waterfall model was selected because it has systematic, sequential development stages and is suitable for building applications whose system requirements have been defined from the beginning. In the context of website-based application development, the Waterfall model can guide the work process from requirement analysis, system design, implementation, testing, to system evaluation. The use of the Waterfall model remains relevant in information system development because it provides a clear and documented workflow at each development stage (Putri et al., 2026).

The research stages consist of five main stages: requirement analysis, system design, implementation, testing, and evaluation. In the requirement analysis stage, the researchers identified the functional and nonfunctional requirements of the system. Functional requirements include the system's ability to upload student assignment documents, store document data, extract text, perform text preprocessing, calculate word weights, compare similarities between documents, and display similarity percentage results. Meanwhile, nonfunctional requirements include ease of use, processing speed, system compatibility with browsers, and data storage capability through a database.

The system design stage was carried out by preparing the application workflow, database design, and user interface design. The system was designed so that lecturers could upload student assignment documents in digital format, fill in student identities, and run the document similarity checking process. The interface was designed simply so that users could operate the system easily. This is in line with the principles of web-based application development, where the system should support process efficiency, ease of access, and clear

user interaction.

The implementation stage was carried out by translating the system design into a website-based application. The technologies used include HTML and CSS to build the page structure and appearance, PHP as the server-side programming language, JavaScript to support page interactivity, and MySQL as the database for storing document information. In addition, development was carried out using XAMPP as the local server and Sublime Text as the code editor. These technologies were selected according to the needs of a web-based system that requires document management and structured data storage.

The data used in this study consisted of student assignment documents in digital format. Based on the research data, the number of documents used was 165 files, consisting of 79 dataset files and 86 test files. The documents used were student assignments from the same course and topic so that the document similarity comparison process could be carried out more relevantly. The use of documents with the same assignment context is important because text similarity measurement becomes more meaningful when the compared documents are in the same domain or topic.

The document checking process was carried out through several stages. The first stage was document upload, which is the process of entering student assignment files into the system. After the document was successfully uploaded, the system performed text extraction to retrieve the text content from the digital document. The next stage was text preprocessing, which included case folding, tokenizing, filtering, and stemming. This stage aimed to clean the text, standardize word forms, remove irrelevant words, and produce a text representation that was ready for computational processing. Preprocessing stages such as case folding, filtering, stopword removal, and stemming are widely used in TF-IDF and cosine similarity-based systems because they can improve the quality of text representation before the similarity calculation process is performed (Albab et al., 2026).

After the preprocessing stage, the system performed word weighting using the Term Frequency-Inverse Document Frequency or TF-IDF method. TF-IDF is used to measure the importance of a word in a document based on the frequency of word occurrence in the document and its distribution across the document collection. This approach is widely used in natural language processing and document similarity measurement because it can represent text in numerical vector form. Widianto et al. (2024) explained that TF-IDF can be used to represent documents in a high-dimensional vector space, capture unique characteristics of document content, and reduce the influence of common words that appear in many documents.

The next stage was measuring document similarity using the cosine similarity method. This method is used to calculate the closeness between two documents based on the angle between document vectors. The greater the cosine similarity value, the higher the similarity level between two documents. The combination of TF-IDF and cosine similarity has been widely used in the development of document similarity detection systems because it can produce similarity values in the form of scores or percentages that are easy to interpret. Halim and Lasut (2024) showed that web-based plagiarism detection applications can use TF-IDF and cosine similarity to generate document similarity percentages through preprocessing, weighting, and similarity calculation.

In general, the document checking workflow in this system is as follows:

1. The user uploads a student assignment document.
2. The system extracts text from the document.
3. The system performs text preprocessing.
4. The system calculates word weights using TF-IDF.
5. The system calculates document similarity using cosine similarity.
6. The system displays the document similarity percentage.
7. The checking result is stored in the database.

System testing was conducted to determine whether the developed application could perform its main functions according to the research objectives. Testing was carried out on the document upload feature, data storage, text extraction, preprocessing process, similarity calculation, and similarity percentage result display. In addition, testing was also conducted using several sample documents with different levels of similarity. In the initial manuscript, the system was tested using five sample documents that produced various categories, namely original, plagiarism indication, and high plagiarism.

The testing results were used to assess the system's ability to assist in the initial examination of student assignment document similarity. In this study, the similarity percentage value was not directly used to determine whether a document constituted plagiarism, but rather as an initial indicator that required further verification by lecturers. This approach is important because text similarity does not always indicate academic misconduct, especially when there are quotations, technical terms, or standard sections that commonly appear in academic documents. Recent studies on plagiarism detection also emphasize that detection systems should be understood as supporting tools because text-based approaches still have limitations in detecting paraphrases, sentence structure changes, or more complex semantic similarities (Sajid et al., 2025).

Through this method, the study produced a website-based application that can help lecturers manage student assignment documents and automatically display similarity levels between documents. The developed system is not intended to replace lecturers' academic judgment, but rather to serve as a tool to accelerate the initial identification process for documents with high similarity levels.

RESULTS AND DISCUSSION

System Implementation

This study produced a website-based digital document checking application for student assignments designed to assist lecturers in managing and checking the similarity level of student assignment documents. The system was built with key features such as document upload, assignment data storage, text extraction, text preprocessing, word weighting using TF-IDF, and document similarity measurement using cosine similarity.

The application was developed using HTML, CSS, PHP, JavaScript, and MySQL. The system runs through a local server using XAMPP. In general, the system workflow begins with the process of uploading student assignment documents, extracting document content, processing text, calculating word weights, calculating similarity values, and displaying document similarity percentage results.

Based on the manuscript design, this application has a document upload page that allows lecturers to enter student names, select assignment files, and upload documents into the system. The uploaded documents are then processed by the system to calculate their similarity level with other documents stored in the database.

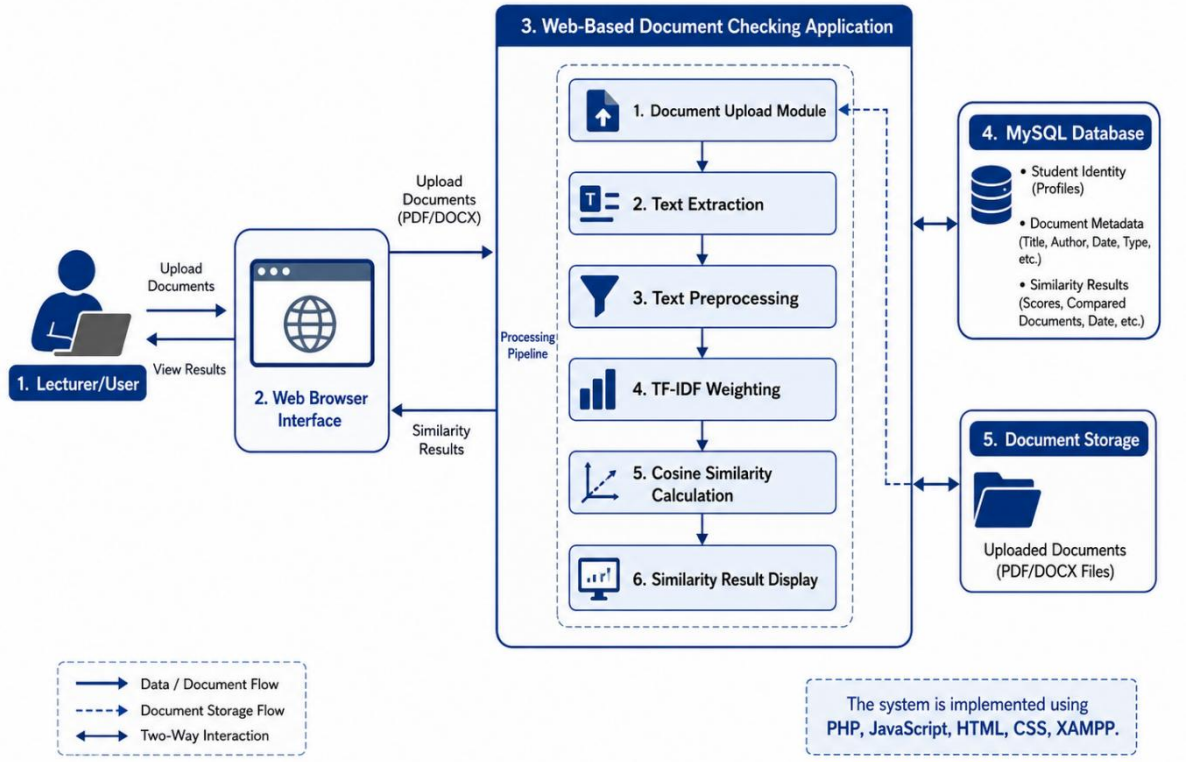

Figure 1. Workflow of the Digital Document Checking System

Figure 1 shows the workflow of the digital document checking application for student assignments. The process begins with document file upload, followed by file format checking by the system. If the file format is valid, the system extracts text from the document. The extracted text is then processed through preprocessing stages, namely case folding, tokenizing, filtering, and stemming. After that, the system performs word weighting using TF-IDF and calculates similarity values between documents using cosine similarity. The final result of this process is the document similarity percentage displayed to the user.

Document Similarity Checking Process

The document similarity checking process is carried out through several main stages. The first stage is document upload, where the lecturer enters student assignment files into the system. The files used can be digital documents, such as PDF or DOCX. After the document is successfully uploaded, the system extracts text to retrieve the document content to be analyzed.

The next stage is text preprocessing. At this stage, the extracted text is processed so that it is ready to be used in document similarity calculations. The preprocessing stages include:

1. Case folding, which converts all letters into a uniform format.
2. Tokenizing, which breaks the text into word units.
3. Filtering, which removes irrelevant or common words.
4. Stemming, which converts words into their root forms.

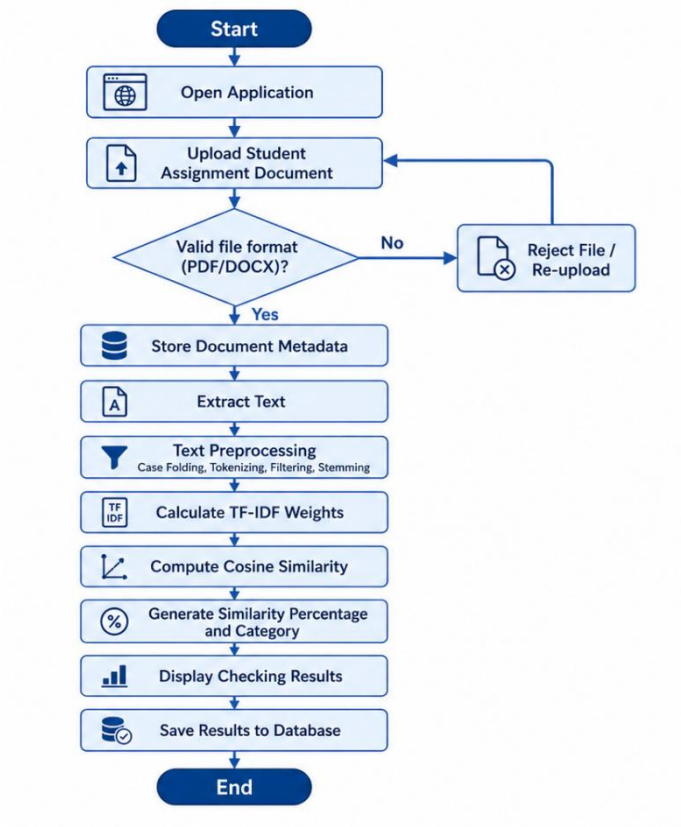

Figure 2. System Flowchart

After the preprocessing stage, the system performs word weighting using the TF-IDF method. This method is used to assign weight to each word based on its frequency of occurrence in a document and its uniqueness in the document collection. Words that frequently appear in one document but rarely appear in other documents receive higher weights.

The final stage is similarity value calculation using cosine similarity. This method calculates the closeness between documents based on the angle between document vectors. The higher the cosine similarity value, the higher the similarity level between two documents. The calculation results are then converted into percentage form so that lecturers can understand them more easily as system users.

Application Interface Implementation

One of the main displays in the system is the student assignment upload page. On this page, lecturers can enter student identities, select assignment files, and upload documents into the system. The interface is designed simply so that lecturers can use it easily without requiring complex technical skills.

Figure 3 shows the student assignment upload page in the developed application. The page provides fields for student name, a file selection button, and a button to submit data to the system. After the lecturer uploads a file, the system stores the document in the database and processes its content for similarity checking. This interface is designed simply so that the student assignment document management process can be carried out practically and efficiently.

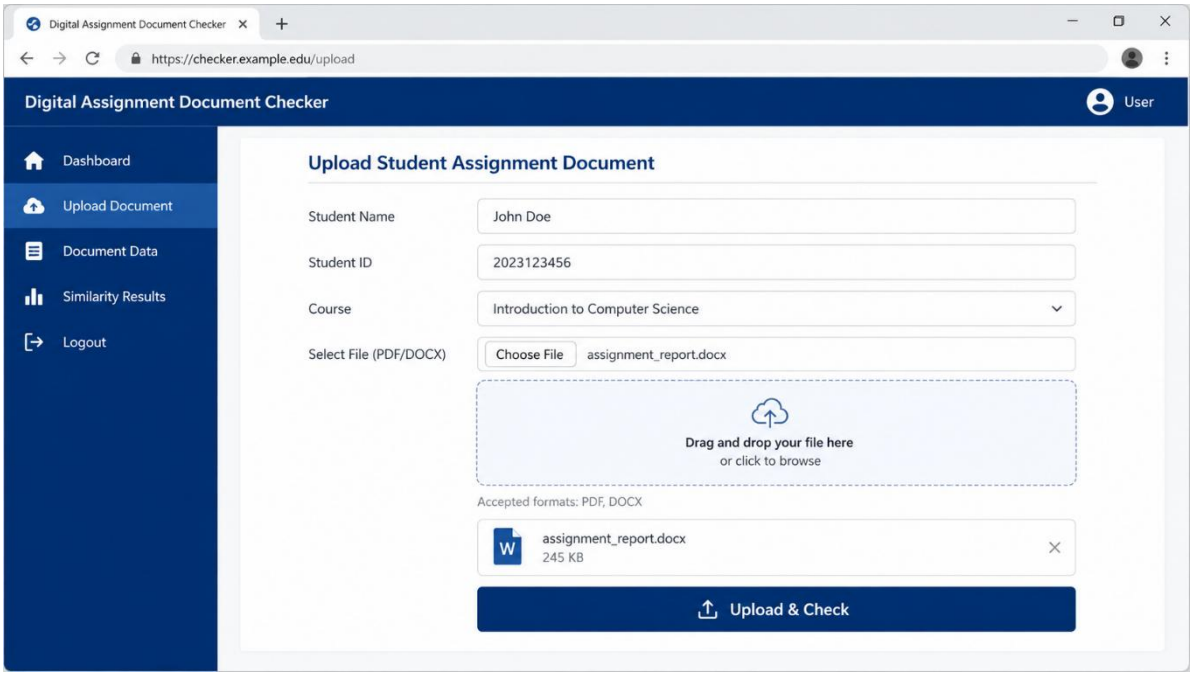

Figure 3. Student Assignment Upload Page

System Testing Results

System testing was carried out using five sample student assignment documents with different similarity levels. The testing aimed to determine the system's ability to read documents, calculate word counts, process documents, and display similarity percentages. The testing results showed that the system was able to provide various document similarity values, ranging from original, plagiarism indication, to high plagiarism categories.

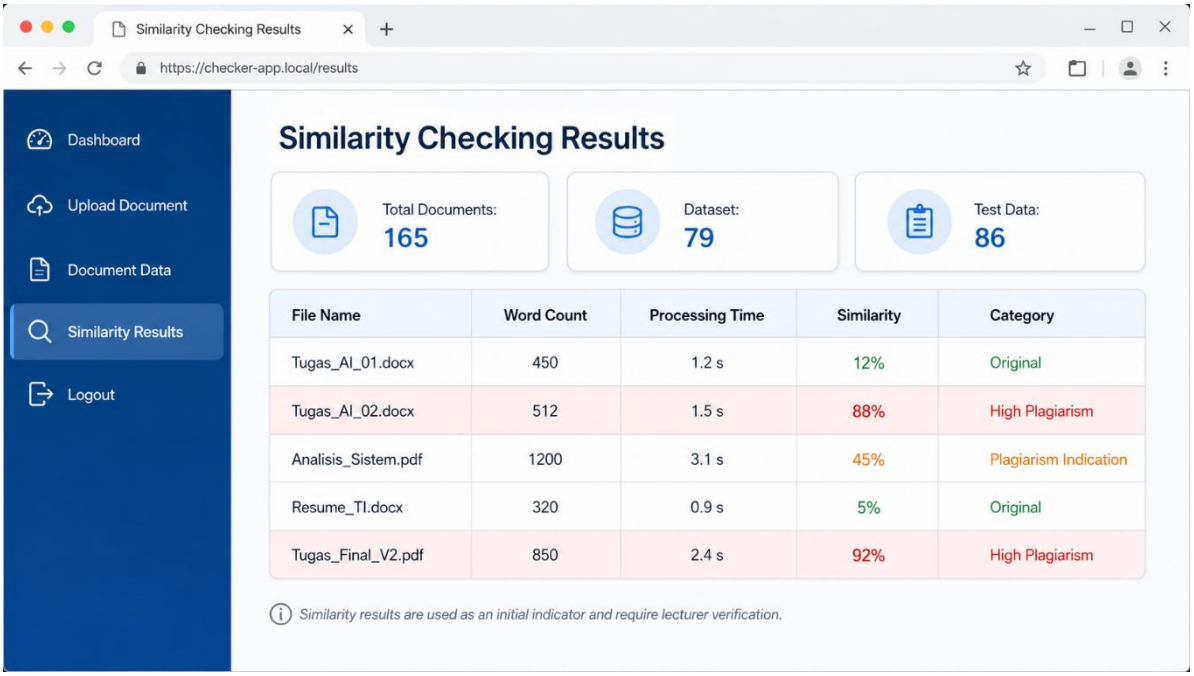

Figure 4. Document Similarity Testing Results

Table 1. Document Similarity Testing Results

No.	File Name	Word Count	Processing Time	Similarity	Description
1	Tugas_AI_01.docx	450	1.2 seconds	12%	Original
2	Tugas_AI_02.docx	512	1.5 seconds	88%	High Plagiarism
3	Analisis_Sistem.pdf	1,200	3.1 seconds	45%	Plagiarism Indication
4	Resume_TI.docx	320	0.9 seconds	5%	Original
5	Tugas_Final_V2.pdf	850	2.4 seconds	92%	High Plagiarism

Based on Table 1, the system was able to display document similarity percentages with varied results. The document Resume_TI.docx had the lowest similarity level, namely 5%, and was therefore categorized as original. The document Tugas_AI_01.docx had a similarity level of 12% and was also categorized as original. Meanwhile, the document Analisis_Sistem.pdf had a similarity level of 45%, so it was categorized as a plagiarism indication and required further examination by the lecturer.

The other two documents, Tugas_AI_02.docx and Tugas_Final_V2.pdf, had very high similarity levels, namely 88% and 92%, respectively. These values indicate that both documents had high textual similarity to other documents in the database, and were therefore categorized as high plagiarism.

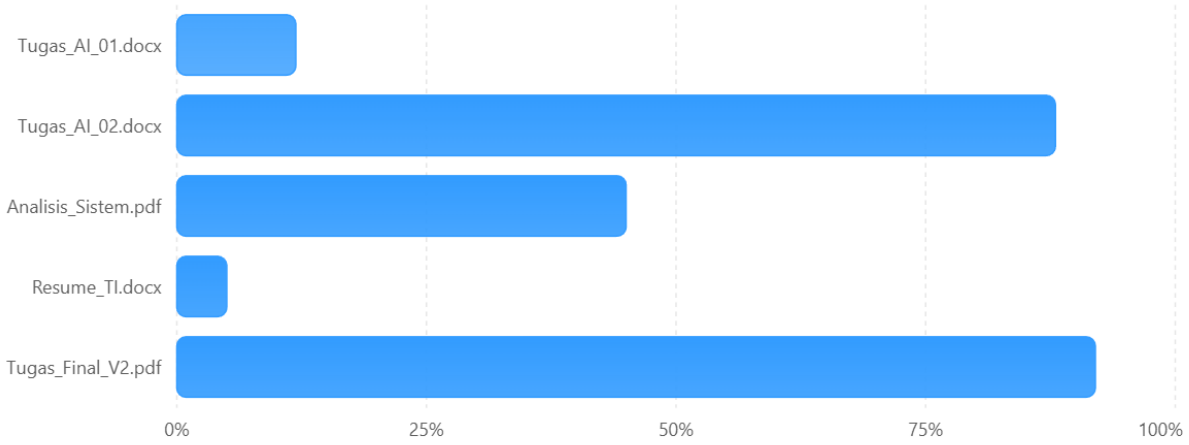

Figure 5. Visualization of Document Similarity Percentage

Figure 5 shows the visualization of similarity percentages for five sample student assignment documents. The chart shows that two documents had very high similarity levels, namely Tugas_Final_V2.pdf at 92% and Tugas_AI_02.docx at 88%. Meanwhile, Resume_TI.docx and Tugas_AI_01.docx had low similarity levels. This visualization helps lecturers identify documents that require further attention.

System Processing Time

In addition to displaying similarity percentages, the system also shows the processing time for each document. Based on the testing results, processing time ranged from 0.9 seconds to 3.1 seconds. Documents with a larger number of words tended to require longer processing time. For example, Analisis_Sistem.pdf had 1,200 words and required 3.1 seconds of processing time. Meanwhile, Resume_TI.docx, which had 320 words, required only 0.9

seconds.

These results indicate that the number of words in a document affects processing time. The more words that must be extracted and weighted, the greater the time required by the system to produce similarity values. Nevertheless, the processing time shown was still considered fast for initial checking of student assignment documents.

Table 2. Relationship Between Word Count and Processing Time

File Name	Word Count	Processing Time
Resume_TI.docx	320	0.9 seconds
Tugas_AI_01.docx	450	1.2 seconds
Tugas_AI_02.docx	512	1.5 seconds
Tugas_Final_V2.pdf	850	2.4 seconds
Analisis_Sistem.pdf	1,200	3.1 seconds

Based on Table 2, there is a tendency that documents with higher word counts require longer processing time. This is reasonable because the system must perform more processes, starting from text extraction, word tokenization, irrelevant word removal, TF-IDF weighting, to cosine similarity calculation.

The results of this study show that the website-based digital document checking application for student assignments is able to perform the main functions required for document management and checking. The system can receive document uploads, extract text, process words, calculate word weights, and display document similarity percentages. Therefore, this application can serve as a supporting tool for lecturers in conducting an initial examination of potential similarity in student assignments.

The application of TF-IDF and cosine similarity methods proved suitable for the needs of a text-based document checking system. TF-IDF helps the system distinguish words with important weights from common words that frequently appear in many documents. This is important because academic documents often contain similar common words, such as technical terms, conjunctions, or academic phrases. With TF-IDF weighting, the system can focus more on words that contribute significantly to distinguishing document content.

Meanwhile, cosine similarity helps the system calculate closeness between documents based on text vector representation. The calculation results in percentage form make it easier for lecturers to understand the similarity level of documents. A low similarity percentage can indicate that a document is relatively original, while a high similarity percentage can indicate that the document requires further examination.

Nevertheless, the similarity percentage results cannot be used as the sole basis for determining that a document constitutes plagiarism. Text similarity can occur due to the use of the same technical terms, similar assignment formats, proper quotations, or uniform assignment instructions. Therefore, this system is more appropriately positioned as an initial detection tool, not as a final determinant of academic misconduct.

The testing results show that documents with high similarity levels, such as Tugas_AI_02.docx and Tugas_Final_V2.pdf, can be directly marked as documents requiring deeper examination by lecturers. This makes the assignment checking process more efficient because lecturers can prioritize documents with high similarity percentages. Conversely, documents with low similarity levels can be processed more quickly because they have relatively low similarity risk.

In terms of system effectiveness, this application provides benefits in two main aspects. First, the system helps manage student assignment documents in a more structured manner. Documents that were previously scattered across various storage media can be stored and managed through a website-based system. Second, the system helps automate the document similarity checking process, thereby reducing lecturers' workload in conducting manual checking.

However, the developed system still has several limitations. First, the system's ability is highly dependent on the quality of text extraction from documents. If the document consists of scanned images, the system requires OCR support so that the text can be read. Second, the system only compares documents with files available in the database, so it cannot yet check similarity with external sources from the internet. Third, TF-IDF and cosine similarity methods are stronger in detecting textual similarity, but still limited in detecting paraphrases or sentence changes with the same meaning.

Based on these limitations, future development can be directed toward integration with OCR technology, expansion of the document database, and the application of semantic approaches based on natural language processing or machine learning. Thus, the system will not only be able to detect word similarity, but also analyze semantic closeness between documents.

The results of this study have practical implications for higher education institutions, particularly in the management of student assignments and the improvement of academic integrity. The developed application can be used as a support system for lecturers to conduct an initial examination of student assignment documents. With this system, the checking process is no longer fully manual, making the examination time more efficient.

In addition, this application can also serve as an educational medium for students to be more careful in preparing academic assignments. Students can understand that submitted digital documents may be checked for similarity levels. This can encourage students to pay more attention to writing originality, citation use, and academic ethics.

From the system development perspective, this study shows that simple web technology can be used to build a functional document checking application. By utilizing PHP, MySQL, TF-IDF, and cosine similarity, the system can provide sufficiently informative results to support academic processes in higher education.

CONCLUSION

Based on the research results, the website-based digital document checking application for student assignments was successfully designed and developed to assist in the management and examination of document similarity levels. The system was developed using the Waterfall method through the stages of requirement analysis, design, implementation, testing, and evaluation. The main features produced include document upload, student assignment data storage, text extraction, text preprocessing, word weighting using TF-IDF, and document similarity measurement using cosine similarity.

The testing results showed that the system was able to process student assignment documents and display similarity percentages in several categories, namely original, plagiarism indication, and high plagiarism. In testing five sample documents, the system produced varied similarity levels, namely 5%, 12%, 45%, 88%, and 92%. Documents with low similarity levels were categorized as original, while documents with high similarity levels required further examination by lecturers. These results indicate that the application can be used as an initial supporting tool in identifying documents with high similarity potential.

The developed application can also help lecturers manage student assignment documents in a more structured manner. Through a website-based system, the upload, storage, and document checking processes can be carried out more practically compared with manual checking. Therefore, this system has the potential to improve the effectiveness and efficiency of the student assignment checking process, especially in classes with a large number of documents.

Nevertheless, the similarity percentage results displayed by the system cannot be used as the sole basis for determining plagiarism. These results should be positioned as initial indicators that still require academic verification by lecturers, particularly to distinguish similarities caused by plagiarism, the use of similar technical terms, uniform assignment formats, or legitimate quotations. In addition, the system still has limitations because it only compares

documents with data stored in the database and does not yet include external internet sources. Overall, this study shows that the application of TF-IDF and cosine similarity in a website-based application can support the similarity checking process for digital student assignment documents. For future development, the system can be improved by adding OCR technology to read scanned documents, integrating external source searching, and applying semantic similarity or machine learning-based approaches so that the system can better detect semantic similarity and paraphrasing.

ACKNOWLEDGMENT

The authors would like to express their gratitude to all parties who supported the implementation of this research.

REFERENCES

Balalle, H., & Pannilage, S. (2025). Reassessing academic integrity in the age of AI: A systematic literature review on AI and academic integrity. *Social Sciences & Humanities Open, 11*, 101299. https://doi.org/10.1016/j.ssaho.2025.101299

Bittle, K., & El-Gayar, O. (2025). Generative AI and academic integrity in higher education: A systematic review and research agenda. *Information, 16*(4), 296. https://doi.org/10.3390/info16040296

Foltýnek, T., Meuschke, N., & Gipp, B. (2020). Academic plagiarism detection: A systematic literature review. *ACM Computing Surveys, 52*(6), 1–42. https://doi.org/10.1145/3345317

Gregory, A., & Leeman, J. (2021). *On the perception of plagiarism in academia: Context and intent*. arXiv. https://arxiv.org/abs/2104.00574

Halim, J., & Lasut, D. (2024). Document plagiarism detection application using web-based TF-IDF and cosine similarity methods. *bit-Tech, 7*(2), 202–213. https://doi.org/10.32877/bt.v7i2.1697

Kaur, N. (2024). A review on string-based text similarity techniques in computational analysis. *International Journal of Intelligent Systems and Applications in Engineering*. https://ijisae.org/index.php/IJISAE/article/view/7629

Meuschke, N., Stange, V., Schubotz, M., Kramer, M., & Gipp, B. (2019). Improving academic plagiarism detection for STEM documents by analyzing mathematical content and citations. In *Proceedings of the 19th ACM/IEEE Joint Conference on Digital Libraries (JCDL)* (pp. 120–129). IEEE. https://doi.org/10.1109/JCDL.2019.00026

Navaro, K. F., Wardana, S. R., & Hapsari, R. K. (2023). Journal article plagiarism detection using Latent Semantic Analysis (LSA). *Prosiding Seminar Nasional Sains dan Teknologi Terapan*. https://ejurnal.itats.ac.id/sntekpan/article/view/5204

Pudasaini, S., Miralles-Pechuán, L., Lillis, D., & Llorens Salvador, M. (2025). Survey on AI-generated plagiarism detection: The impact of large language models on academic integrity. *Journal of Academic Ethics, 23*, 1137–1170. https://doi.org/10.1007/s10805-024-09576-x

Sadhin, I. H., Hassan, T., & Nayim, M. A. M. (2024). Plagiarism detection using artificial intelligence. *International Journal of Computer and Information System, 5*(2), 102–108. https://doi.org/10.29040/ijcis.v5i2.170

Virginia, C., & Alamsyah, D. (2026). Plagiarism detection in English academic documents using a

lexical semantic hybrid and support vector machine. *INOVTEK Polbeng – Seri Informatika, 11*(1), 96–107. https://doi.org/10.35314/2zz12581

Zachrias, U., & Gunawan, W. (2025). Classification of document similarity using Winnowing algorithm with Jaccard Coefficient approach. *Science of Information & Technology Applied*. https://ejournal.bacadulu.net/index.php/sinta/article/view/87