Plagiarism Checking Information System for Final Assignments of Malang State Polytechnic Using the Latent Semantic Analysis (LSA) Method

Bafian Atha Fiddin[1], Rieke Adriati Wijayanti[2*], Rizky Ardiansyah[3]

Digital Telecommunication Network, Department of Electrical Engineering, State Polytechnic of Malang, 65141, Indonesia.

[1] bafianfiddin12@gmail.com [2] riekeaw@polinema.ac.id [3] rizkyardiansyah@polinema.ac.id

Abstract— This study aims to design and develop a plagiarism checking information system for students' final assignments at the State Polytechnic of Malang using the Latent Semantic Analysis (LSA) method. Plagiarism in scientific papers is a serious problem that can damage academic integrity. Therefore, an effective and accurate system is needed to detect plagiarism in students' final assignment documents. The LSA method was chosen because of its ability to analyze semantic relationships between words and documents, thus detecting plagiarism that cannot be detected directly by word-based methods. This system was built to check the similarity between uploaded documents and documents in the campus database. Based on the test results, this system demonstrated a plagiarism detection accuracy of up to 80%, as well as an efficient processing time of between 13 and 14 seconds per document. Thus, this system is expected to improve supervision of the authenticity of students' final assignments and support academic integrity at the State Polytechnic of Malang. Recommendations for the development of this system are to expand file format support and improve the ability to detect more hidden plagiarism, such as paraphrasing.

Keywords— Plagiarism Detection, Latent Semantic Analysis (LSA), Final Project, Information System, Accuracy.

I. INTRODUCTION

The development of information and communication technology has brought significant changes to various aspects of life, including education. Malang State Polytechnic, as a higher education institution committed to producing quality graduates, faces challenges in maintaining the authenticity of students' academic work, especially in final assignments. One of the main challenges faced is the increasing number of plagiarism cases that can damage academic integrity. Based on data from the Ministry of Research, Technology, and Higher Education, plagiarism in scientific work in Indonesia has shown an increase from 2019 to 2022. According to research conducted by Budoyo, there were 808 cases of plagiarism in the academic realm in 2019 alone. Based on the results of research conducted by Hutabarat in 2021, it also showed that there were 0-27% of plagiarism cases among Library and Information Science students. Based on the proportion of plagiarized work: Light Plagiarism: Less than 30% of the content of the work is plagiarized. Moderate Plagiarism: Between 30%-70% of the content of the work is plagiarized. Total Plagiarism: More than 70% of the content of the work is plagiarized. Based on the pattern of plagiarism: Total Plagiarism: Plagiarizing the entire work of others [1][2].

Plagiarism, whether intentional or unintentional, threatens the quality of scientific work, primarily due to a lack of student awareness and easy access to digital information. Therefore, a system to detect and prevent plagiarism is needed[3]. The Latent Semantic Analysis (LSA) method was chosen because of its ability to analyze semantic relationships between words and documents, making it more effective than conventional methods in detecting hidden plagiarism[4].

With the LSA-based plagiarism checking information system, Malang State Polytechnic is expected to improve oversight of the authenticity of students' final assignments. This system not only helps detect plagiarism automatically and accurately, but also educates students about the importance of academic honesty and originality of scientific work. This aligns with the institution's vision of creating a distinguished academic environment that supports innovation[5].

Previous research, entitled "Application of the Laravel Framework in the Design and Construction of a Thesis Title Search Application for the Digital Telecommunication Networks Study Program," [6] aimed to design and implement a thesis title search application using the Laravel framework. This application performs data preprocessing to clean non-alphabetical data and remove common words (stopwords) to improve performance. Testing using GTmetrix showed that the application provides a good user experience, prevents duplicate theses titles, and improves search similarity using the Cosine Similarity and Jaccard Similarity algorithms.

Based on previous research, although systems for checking the similarity of student final assignment documents exist, weaknesses remain in the Cosine Similarity and Jaccard Similarity algorithms. Cosine Similarity only measures similarity based on vector orientation without understanding word meaning or the relationships between words[7]. This algorithm is sensitive to word distribution and is less effective for long documents because it doesn't understand the context between paragraphs. Meanwhile, Jaccard Similarity only compares elements within a set without considering word order, so it can't capture context or sentence structure[8]. Furthermore, Jaccard is less accurate for long documents with significantly

*Corresponding Author
E-ISSN: 2654-6531 P- ISSN: 2407-0807

different sizes. In contrast, the LSA method is superior because it can understand semantic relationships between words through matrix decomposition (SVD). It is also more reliable for long or complex documents.

II. METHOD

A. *Type of Research*

This research is a research and development project aimed at designing and developing a Final Project Information System. This system is designed to simplify the management of students' final projects on campus, with various features that can increase efficiency, one of which is a plagiarism checker.

B. *The research stages*

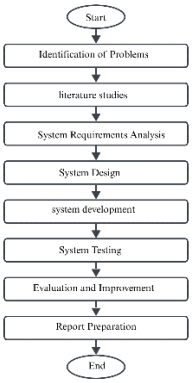

Figure 1. Flowchart Stages Research

The research phase begins with problem identification, which aims to understand the issues that exist in final project management on campus, particularly those related to plagiarism. At this stage, researchers identified the need for a system that can efficiently and accurately detect plagiarism in students' final projects. This identified problem will form the basis for developing a system designed to better support final project management, as shown in Fig. 1.

Once the problem is identified, the next step is to conduct a literature review. This stage is crucial for exploring various references and research related to technologies that can be used to detect plagiarism. One method used in this research is Latent Semantic Analysis (LSA), which is effective in measuring the level of similarity between text documents[9]. This literature review also includes a review of existing systems and other text analysis methods that could be used in the system being developed.

Next, researchers conducted a system requirements analysis, which aimed to gather information regarding the functional and non-functional requirements of the system to be built. Functional requirements include key features such as a plagiarism checking system, final project database management, and an easy-to-use user interface. Non-functional requirements cover aspects such as data security, system speed, and the ability to handle various final project document formats. This information will form the basis for more detailed system design.

In the system design stage, researchers design the system architecture and compile the features to be implemented. This design includes creating flowcharts, data structures, and selecting appropriate technology for system development. At this point, the plagiarism checking feature using the LSA method will be designed in detail, including the text data processing mechanism, document similarity calculation, and how the analysis results are presented to users[10].

The final stage is system testing, which aims to ensure that the system being built functions properly. Testing is conducted by comparing the results of the plagiarism analysis on the final project document with the data in the database. If the system functions as expected, Then, the system is evaluated and refined to ensure its accuracy and effectiveness. After testing and evaluation are complete, the researcher prepares a research report documenting the entire development process, testing, and research results.

C. *Use Case Diagram*

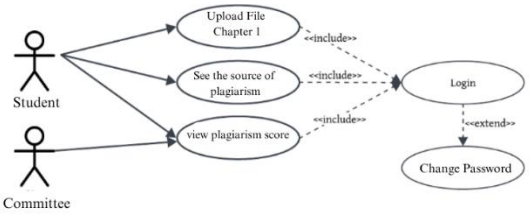

Figure 2. Use Case Diagram

In the Final Project Information System, students must first log in before accessing other features. Each student use case—Upload CHAPTER 1 File, View Plagiarism Score, and View Plagiarism Source File—has an include relationship with Login, indicating that the login process is a mandatory initial step. After successfully logging in, students can access features for checking their thesis documents for plagiarism, while the committee can view the plagiarism score results, as shown in Fig. 2.

Furthermore, Login has an extended relationship with Change Password, meaning students have the option to change their password after successfully logging in. This allows students to maintain the security of their accounts and update their passwords if necessary. Thus, this diagram illustrates that logging in is a mandatory step to access the system, with the additional option to change their password as an optional feature after logging in.

D. *Activity Diagram*

This activity diagram illustrates the system's workflow in detecting plagiarism in Chapter 1, uploaded by a student. The diagram consists of three main columns: STUDENT, SYSTEM, and DATABASE, which shows the interaction between the three in the plagiarism checking process. The following is an image of the activity diagram in this system:

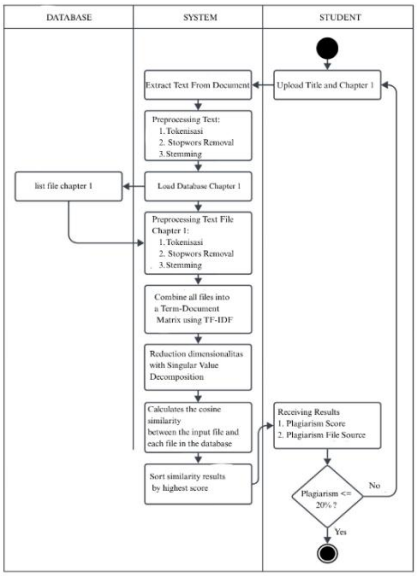

Figure 3. Activity Diagram

Figure 3 illustrates the system's workflow for detecting plagiarism in Chapter 1 uploaded by a student. The process begins with the student uploading the Chapter 1 document to the system, which serves as the starting point in the activity diagram. The uploaded document is processed by the system to detect potential plagiarism, with the aim of determining whether it contains similarities to other files in the database.

Once the file is uploaded, the system begins a series of steps to process the document. The first step is Text Extraction, in which the system extracts text from the student's uploaded document. This extraction process allows the system to obtain text content for further analysis. Once the text is extracted, the document undergoes Text Preprocessing, which includes several steps: tokenization (breaking the text into words), stopword removal (removing common, meaningless words), and stemming (converting words to their base form).

Next, the system loads the existing Chapter 1 database to compare it with the student's uploaded document. This process is performed by loading all Chapter 1 files in the database relevant to the plagiarism analysis. All Chapter 1 files in the database also undergo a similar text preprocessing process, including tokenization, stopword removal, and stemming. This ensures that all compared documents have a consistent format and structure for accurate plagiarism analysis[11].

After the preprocessing process is complete, the system constructs a Term-Document Matrix using the TF-IDF (Term Frequency-Inverse Document Frequency) method. This method represents each document in vector form based on the word frequency within the document. Using TF-IDF, the system can assess the importance of each word in the document and compare it with words in other documents, including those in the database[12].

Once the Term-Document Matrix is formed, the next step is dimensionality reduction using the Singular Value Decomposition (SVD) method. This method eliminates noise (less important words) and increases efficiency in calculating the similarity level between documents.

The next step is Cosine Similarity, where the system calculates the similarity between the student's document and each document in the database. Cosine Similarity is a method used to measure the similarity between two text vectors. This similarity score provides an indication of how similar the student's document is to documents in the database, which is important for detecting plagiarism[13]. The system then sorts the similarity scores, sorting the results by the highest score, displaying the files with the highest similarity first.

After the calculation and sorting process is complete, students will receive results from the system. These results consist of two main pieces of information: the Plagiarism Score and the Plagiarism Source File. The plagiarism score indicates the similarity level of the student's document to the files in the database, calculated as a percentage. The plagiarism source file lists files with a high similarity to the uploaded document.

The system then evaluates the results and issues a Plagiarism Decision based on the plagiarism score. If the plagiarism score is less than or equal to 20%, the document is deemed plagiarism-free. Conversely, if the plagiarism score is greater than 20%, the document is considered plagiarized, and students are asked to review their submission.

E. Data Flow Diagram

The Data Flow Diagram (DFD) presented illustrates the data flow within this system. The DFD is divided into two levels: Level 0 and Level 1.

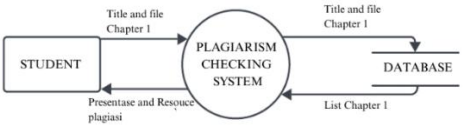

Figure 4. Data Flow Diagram Level 0

A Level 0 Data Flow Diagram (DFD) depicts the general data flow between external entities and the main processes within the system. This Level 0 DFD contains three main components: Students, the System, and the Database. This DFD provides a simple overview of the interactions between the components within the system, without including technical details. The focus is on the data flow between the external entities and the main process that checks documents for plagiarism.

The Student entity serves as the external actor that inputs documents into the system. Students submit documents containing documents to be tested for plagiarism. These documents are further processed by the system to determine the level of similarity or plagiarism with other documents. The data flow from the Student to the System consists of input documents to be tested for plagiarism.

The System process serves as the main administrator within the system. Upon receiving a document from a Student, the System processes it by storing it in the Database and proceeding with the plagiarism check. This process includes text processing and calculating the similarity between the uploaded document and other documents in the system. The

System then accesses the data stored in the Database to perform the plagiarism check.

The database is responsible for storing and managing all document data uploaded by students. Documents uploaded by students are stored in the database for further reference and comparison. The system can access this database to retrieve documents needed for the plagiarism checking process. The database serves as a central repository, allowing the system to examine other potentially relevant documents.

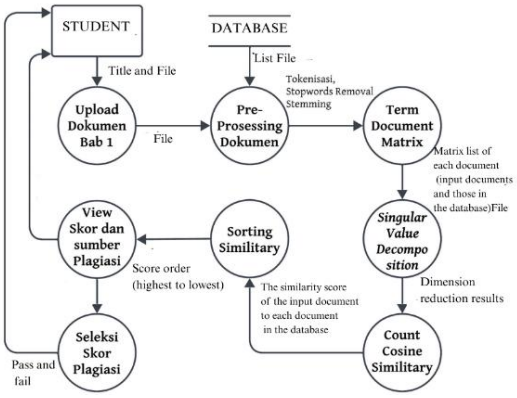

Figure 5. Data Flow Diagram Level 1

In DFD Level 1, the system is explained in more detail through the data flow between the Student Entity, the processes within the System Process, and the Final Project Database. The student acts as a user submitting a document for plagiarism testing. The uploaded document is received by the Document Upload Process in Chapter 1, which is responsible for receiving and initiating the document processing flow. Once the document is received, the data flow then proceeds to the Preprocessing Process, where it undergoes steps such as tokenization, stopword removal, and stemming for further analysis[14].

After the document has been processed through the Preprocessing stage, the prepared document data is sent to the Term Document Matrix Process. Here, the document is converted into a matrix format that allows for further similarity calculations between documents. This process is crucial for converting unstructured documents into a format usable for similarity calculations, making it easier for the system to compare the uploaded document with other documents in the Database.

Once the Term-Document Matrix is formed, the next step is dimensionality reduction using the Singular Value Decomposition (SVD) method. Dimensionality reduction using SVD simplifies the matrix by reducing the number of dimensions or components of the Term-Document Matrix while retaining the most significant information. This allows the system to eliminate noise (less important words) and increase the efficiency of calculating the similarity level between documents[15].

The next stage is the Cosine Similarity Calculation Process, which receives the prepared document matrix. Here, the system calculates the similarity score between the uploaded document and other documents in the database[16]. The results of this calculation are forwarded to the Highest Similarity Score Sorting Process, which sorts the results based on the highest similarity score. This aims to provide a clear comparison of the level of plagiarism between the uploaded document and other documents.

After the similarity scores are sorted, the final result is forwarded to the Plagiarism Score Display Process, which displays the results of the similarity analysis to students. This process provides feedback to students regarding whether their documents contain plagiarism and the extent of the similarity. Furthermore, the documents are further processed through the Plagiarism Pass Process, which checks whether the documents pass the plagiarism check according to the system's predetermined threshold.

Overall, DFD Level 1 describes how the system processes and manages and processes student-uploaded documentation to detect plagiarism in detail. The processes involved, from document upload, preprocessing, document matrix formation, similarity calculation, to scoring and plagiarism checking, work together to produce accurate results.

III. RESULTS AND DISCUSSION

A. Design and Implementation Results

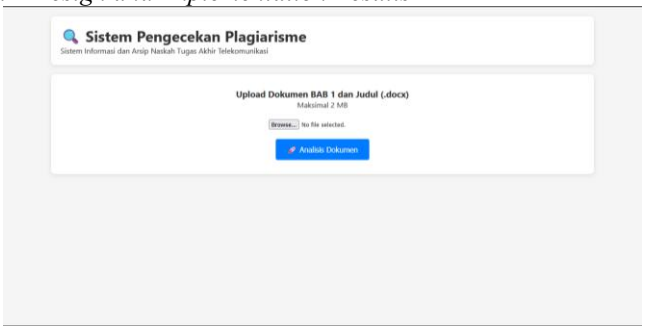

Figure 6. Plagiarism Check Page

Figure 6. shows the interface of a plagiarism detection system for checking .docx final assignments. This system, named "Telecommunication Final Assignment Manuscript Information and Archive System," is designed to check for plagiarism in Telecommunication Engineering final assignments. Users can upload their final assignment files (title and first chapter) with a file size limit of 2 MB, using the Browse button. Once the file is uploaded, the " Analyze Document" button starts the plagiarism analysis process using the Latent Semantic Analysis (LSA) algorithm, which compares the meaning and structure of sentences to accurately detect plagiarism.

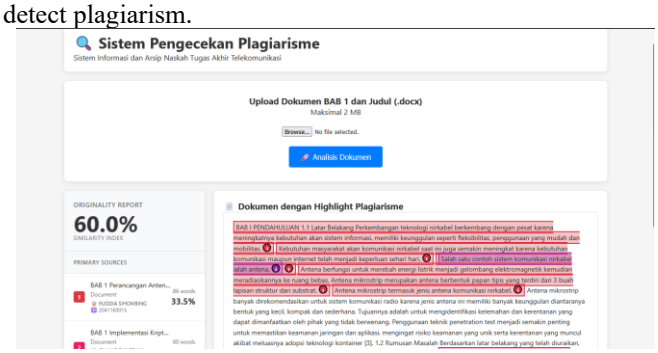

Figure 7. Plagiarism check results page

Figure 7. This is the result of a plagiarism check generated by the plagiarism detection system, specifically for checking Chapter 1 and the title of the final project. This system has several important features relevant to ensuring the originality of a thesis.

The check results will be displayed in the Originality Report, which shows the Similarity Index, or percentage of similarity between the uploaded document and other sources. In the image, a figure of 60.0% indicates a 60% similarity between the uploaded document and other sources in the system database. This provides an initial overview of the level of originality of the document being checked.

Furthermore, documents with plagiarism highlights will display sections of text deemed similar to other sources. Sections of text suspected of plagiarism will be highlighted in red to make it easier for users to see which parts of the document need improvement. For example, in the image, Chapter 1 Introduction is highlighted in red due to its similarity to other documents.

In the Primary Sources section, the system will display primary sources detected as similar to the uploaded document. In this figure, the two main sources detected are Chapter 1 Antenna Design with a similarity of 33.5% and Chapter 1 Cryptography Implementation with a similarity of 23.3%. With this information, users can verify the level of originality of their thesis and identify areas that need improvement if plagiarism is found. This system is very helpful in ensuring that the thesis is written in accordance with academic principles and is free from plagiarism.

B. System Database

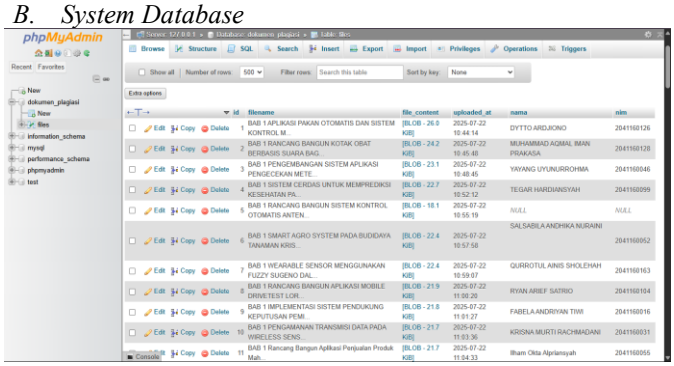

Figure 8. Database system

Figure 8. The following figure shows the PHPMyAdmin interface used to manage a database named dokumen_plagiasi (document_plagiarism) within a plagiarism checking system for thesis documents. This interface shows that PHPMyAdmin is used to manage the files table, which stores various information about thesis files uploaded by the system administrator. Several important columns in this table include id, which uniquely identifies the file; filename, which indicates the file name or title of the thesis document; and file_content, which stores the file contents in BLOB (Binary Large Object) format. This data will then be processed by the system to detect the level of similarity with other documents already stored in the database. Additionally, the uploaded_at column records the file upload time, while the name and student ID columns record the identity of the student who uploaded the document.

This interface also provides three action buttons in each row: Edit, Copy, and Delete. This feature gives administrators the flexibility to manage files directly, such as correcting data, duplicating, or deleting files if an upload error occurs. Furthermore, PHPMyAdmin also provides search and sorting features based on specific attributes, such as file name or upload time. This feature is very helpful in simplifying the search for specific files, especially when the number of documents stored in the system is quite large. Overall, the use of PHPMyAdmin in this system is crucial for supporting the efficient and systematic management of thesis documents. This system ensures that every uploaded file is properly managed and checked for originality through a plagiarism detection process that aligns with the principles of academic integrity.Database Structure

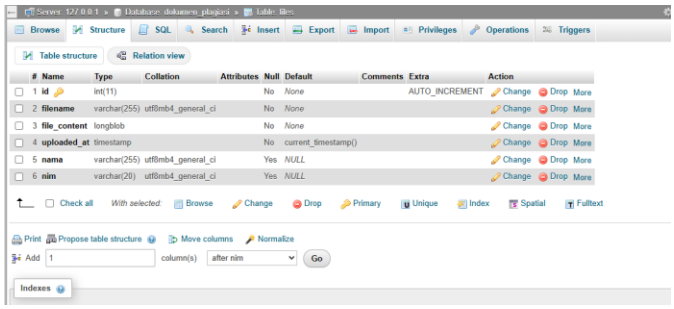

Figure 9. Database structure

Figure 9. The files table in the dokumen_plagiasi database functions to store thesis files uploaded by the admin for plagiarism checking. This table consists of several important columns, namely id as the primary key with type int(11) and the AUTO_INCREMENT attribute to provide a unique identity for each file, filename of type varchar(255) which is required to store the file name, file_content of type longblob to store the contents of binary files such as .docx or .pdf, uploaded_at of type timestamp with a default value of current_timestamp() to record the upload time, and the optional name and student ID columns to store the identity of the uploading student. This table design is equipped with a BTREE primary index so that it supports fast data searches, is flexible for development, and is able to manage academic documents systematically.

C. Testing the Latent Semantic Analysis Method

This test used 104 documents, titles, and chapters from the 2020/2021 intake, which had been uploaded to the system database as comparisons. Students then uploaded documents with mild plagiarism, documents with severe plagiarism, and documents without plagiarism. After all documents were uploaded, the system analyzed them using the Latent Semantic Analysis (LSA) method to detect the level of content similarity between the test documents and the comparison documents in the database, as shown in Table I.

TABLE 1
RESULTS OF THE LATENT SEMANTIC ANALYSIS METHOD TEST

Title	Output	category	Expected results	Conclusion
Chapter 1 implementation of smart security and fire alert system in museum	24%	light	light	Succeed
CHAPTER 1 analysis comparison of RSTP and MSTP for restore network on service metronet in the GPON network of AH Building	72%	heavy	heavy	Succeed
Chapter 1 smart farming innovation in increase productivity agriculture studies case ward lapajung	0%	Not any plagiarism	Not any plagiarism	Succeed

In Document 1, which is entitled *"Chapter 1 implementation of smart security and fire alert system in museum"* , the output produced is 24%, which shows plagiarism light . System succeed detect plagiarism with category light , suitable as expected , although level plagiarism classified as low . In Document 2, entitled *"CHAPTER 1 analysis comparison of RSTP and MSTP for restore network on service metrotren in the GPON network of AH Building"* , the output obtained is 72%, with category weight , shows enough plagiarism significant . System capable identify plagiarism with accurate and provide the right category . While that , in Document 3, which is entitled *"Chapter 1 smart farming innovation in increase productivity agriculture studies case ward lapaiung "* , the resulting output is 0%, which indicates No There is plagiarism . System succeed with appropriate identify that document This free from plagiarism . In overall , results testing show that system detection plagiarism functioning with Good in detect plagiarism in various level similarities , both in documents with plagiarism light , heavy , or without plagiarism .

D. Testing Accuracy Detection Plagiarism

Testing Accuracy Detection Plagiarism is a process for measure how much Good system or detection model plagiarism in classify document based on level similarity (plagiarism). Table II shows the accuracy of measurement to what extent the system can identify whether A document own plagiarism or No.

TABLE II.
TEST RESULTS ACCURACY

Title	Output	System	Tester
CHAPTER 1 analysis comparison of RSTP and MSTP for restore network on service metronet in the GPON network of AH Building	72%	There is Plagiarism	There is Plagiarism
CHAPTER 1 Detection Vehicle Traffic Violations Two-Wheeled Motorbikes Using the Convolutional Neural Network Method in Bogor City	0%	n't any plagiarism	There is plagiarism
CHAPTER 1 System Watering Plant Automatic Based IoT Using DHT 11, DS 18B20 and Soil Moisture Sensors	0%	n't any plagiarism	n't any plagiarism
CHAPTER 1 Utilization of the Internet of Things (IoT) in Monitoring Indoor Air Pollution Emissions from Carbon Dioxide (CO2), Particulate Matter (PM), and Volatile Organic Compounds (VOC)	86%	There is plagiarism	There is plagiarism
Chapter 1 implementation of smart security and fire alert system in museum	24%	There is plagiarism	There is plagiarism

Table II shows the comparison results similarity system with tester so obtained results testing as following :

Data used :

- True Positive (TP): 2 documents detected plagiarism and really contain plagiarism .
- True Negative (TN): 2 documents that are not detected plagiarism and indeed No contain plagiarism .
- False Positive (FP): 1 document detected plagiarism but Actually No There is plagiarism .
- False Negative (FN): 0 (None) detected documents No There is plagiarism but Actually There is plagiarism).

$$Accuracy = \frac{2(TP) + 2(TN)}{2(TP) + 2(TN) + 1(FP) + 0(FN)} =$$

So, accuracy detection plagiarism is 0.8 or 80%

Testing system detection plagiarism produce five documents with different results . Document first , entitled "CHAPTER 1 analysis comparison of RSTP and MSTP for restore network on service metrotren in the GPON network of Building AH", has an output of 72%, which indicates existence plagiarism . Document This categorized as True Positive (TP) because plagiarism detected with correct . Document second , "CHAPTER 1 Detection Vehicle Traffic Violations Two-Wheeled Motorbikes Using the Convolutional Neural Network Method in the Bogor City Region", with an output of 0%, no contain plagiarism and also detected with right , so enter True Negative (TN) category .

Document third , entitled "CHAPTER 1 System Watering Plant Automatic IoT Based Using DHT 11, DS 18B20 and Soil Moisture Sensors ", also shows 0% output, indicating No There is plagiarism , and detected with Correct as True Negative (TN). Document fourth , "CHAPTER 1 Utilization of the Internet of Things (IoT) in Monitoring Indoor Air Pollution Emissions from Carbon Dioxide (CO2), Particulate Matter (PM), and Volatile Organic Compounds (VOC)", shows an output of 86%, which means existence successful plagiarism detected , so that categorized as True Positive (TP).

However , the document The fifth , entitled "Chapter 1 implementation of smart security and fire alert system in museum", has an output of 24%, indicating existence plagiarism . Although thus , the document This Actually free from plagiarism , and the system incorrectly detected it , so that

categorized as False Positive (FP). There is no documents included in False Negative (FN) category , because all documents that should be detected plagiarism succeed recognized by the system .

In a way overall , results testing shows 2 True Positive (TP) documents , 2 True Negative (TN) documents , and 1 False Positive (FP) document .

E. Process Time Testing

Process Time Testing in system detection plagiarism used For evaluate how much fast system can processing documents and provide results detection plagiarism . This process time important For ensure that system can functioning with efficient, as shown in Table III.

TABLE III
PROCESSING TIME TEST RESULTS

Title	Size	Output	Time
CHAPTER 1 Application Voice - Based Stock Management and Medication Scheduler For Blind	20.2 kb	94%	14.36 seconds
CHAPTER 1 System Watering Plant Automatic Based IoT Using DHT 11, DS 18 B20 and Soil Moisture Sensors	15.3 kb	0%	13.17 seconds
Chapter 1 implementation of smart security and fire alert system in museum	15.7 kb	24%	13.36 seconds
CHAPTER 1 analysis comparison of RSTP and MSTP for restore network on service metronet in the GPON network of AH Building	17 kb	72%	13.74 seconds
CHAPTER 1 Detection Vehicle Traffic Violations Two-Wheeled Motorbikes Using the Convolutional Neural Network Method in Bogor City	14.7 kb	0%	13.13 seconds

Test results show that system detection plagiarism Work with Good in detect plagiarism in various level similarities . In addition , the system also shows efficiency in matter time processing , size document influence how much fast documents are analyzed by the system , the more big document so will the longer the time analysis required by the system . All tested documents processed in reasonable time , ranging between 13 seconds up to 14 seconds , showing that system can handle document with different sizes without decline significant performance .

F. Testing file size above 2 Megabytes

Testing This done For ensure whether system can reject more files from 2 MB with give message obvious error to user, as shown in Fig. 10.

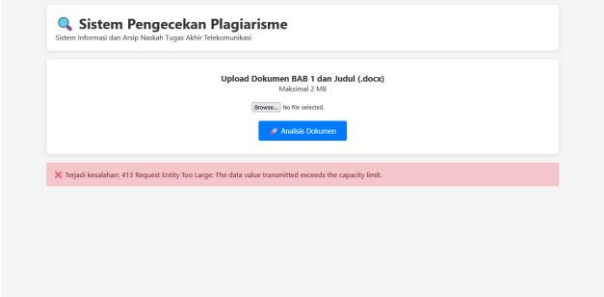

Figure 10. Upload more files from 2 Megabytes

From the picture the can obtained that system can reject more files from 2 MB with give message obvious error to users , namely *"413 Request Entity Too Large"* . This message show that the data sent exceeds the capacity limit specified by the system , so that the file upload process cannot be completed. can continued . This is give bait right back to users For reduce their file size so they can uploaded and analyzed . The system has succeed limit file size that can be processed , maintain performance system to remain optimal and avoid burden excessive .

IV. CONCLUSION

Accuracy Detection Plagiarism : System succeed detect plagiarism with OK , OK on documents containing plagiarism light , heavy , or free from plagiarism . System show accuracy detection enough plagiarism high , with results testing accuracy reached 80%. This is show that system can detect plagiarism with level good reliability in various case . Runtime Performance: Testing process time shows that system can processing document with varying sizes in reasonable time . Processing time range between 13 seconds up to 14 seconds per document , which indicates that system can functioning in a way efficient although with size different documents . System Benefits : System This designed For increase supervision to authenticity task end students at the State Polytechnic of Malang. With use LSA method , system capable analyze similarities semantics between document , which makes it more effective in detect plagiarism Recommendation Development **:** Although system Already show satisfactory results , development more carry on required For increase ability system in handle hidden plagiarism , such as paraphrase or compilation different sentences but own same meaning .

REFERENCES

[1] C. Pasaribu, M. R. Nurmaliza, N. N. Azizah, P. H. Panggayuh, R. Fitrianto, and R. Wasir, “Menghadapi Plagiarisme : Menjaga Kejujuran Akademik Dalam Penelitian Kesehatan Di Era Digital,” J. Kesehat. Tambusai, vol. 5, no. 4, pp. 13079–13087, 2024, doi: 10.31004/jkt.v5i4.37829.

[2] A. Pratiwi, D. Program, S. Jaringan, T. Digital, T. Elektro, and P. N. Malang, “Program Studi Jaringan Telekomunikasi Digital ~ Politeknik Negeri Malang Program Studi Jaringan Telekomunikasi Digital ~ Politeknik Negeri Malang,” pp. 198–201, 2019.

[3] M. S. Wahyuni, D. Setiawan, and T. Syahputra, “Sistem Temu Kembali Informasi Dengan Latent Semantic Analisys Pada Kesamaan Tugas Akhir Mahasiswa,” *J. Tek.*, vol. 1, no. 2, p. 46, 2021, doi: 10.54314/teknisi.v1i2.655.

[4] K. Fauzan Navaro, S. Rosetya Wardhana, and R. K. Hapsari, “Deteksi Plagiarisme Artikel Jurnal menggunakan Latent Semantic Analysis (LSA),” *Semin. Nas. Sains dan Teknol. Terap. XI*, pp. 1–8, 2023.

[5] N. Fitrianingsih, M. Asfi, D. Prasetyo, R. P. Kusuma, and M. A. Sulhan, “Deteksi Tingkat Kemiripan Judul Menggunakan Algoritma Oliver Pada Sistem Informasi

Pengajuan Skripsi," *J. Media Inform. Budidarma*, vol. 6, no. 4, p. 2148, 2022, doi: 10.30865/mib.v6i4.4409.

[6] E. Ernawati, Anindito, and R. N. P. Atmojo, "Istem Pendeteksi Plagiarisme Untuk Tugas Akhir," *Humaniora*, vol. 5, no. 1, pp. 541–549, 2014.

[7] M. Duggan, D. R. Roderick, and J. Sieburg, "Data bases," *Proc. 1970 25th Annu. Conf. Comput. Cris. How Comput. are Shap. our Futur. ACM 1970*, pp. 1–7, 1970, doi: 10.1145/1147282.1147284.

[8] A. Permatasari and S. Suhendi, "Rancang Bangun Sistem Informasi Pengelolaan Talent Film berbasis Aplikasi Web," *J. Inform. Terpadu*, vol. 6, no. 1, pp. 29–37, 2020, doi: 10.54914/jit.v6i1.255.

[9] M. R. S. Alfarizi, M. Z. Al-farish, M. Taufiqurrahman, G. Ardiansah, and M. Elgar, "Penggunaan Python Sebagai Bahasa Pemrograman untuk Machine Learning dan Deep Learning," *Karya Ilm. Mhs. Bertauhid (KARIMAH TAUHID)*, vol. 2, no. 1, pp. 1–6, 2023.

[10] *H. Saputro, U. Baturaja, and J. A. Yani, "Jurnal Informatika dan Komputer(JIK)," Jik, vol. 12, no. 2, p. 83, 2021.*

[11] F. Fataruba, "Penerapan Metode Cosine Similarity Untuk Pengecekan Kemiripan Jawaban Ujian Siswa," *J. Mhs. Tek. Inform.*, vol. 2, no. 2, pp. 88–95, 2018.

[12] A. Hidayat, A. Yani, Rusidi, and Saadulloh, "Membangun Website Sma Pgri Gunung Raya Ranau Menggunakan Php Dan Mysql," *JTIM J. Tek. Inform. Mahakarya*, vol. 2, no. 2, pp. 41–52, 2019.

[13] Y. Riadi Silitonga, Munawar, and I. Noor Hapsari, "Analisis Dan Penerapan Datamining Untuk Mendeteksi Berita Palsu (Fake News) Pada Social Media Dengan Memanfaatkan Modul Scikit Learn," *Undergrad. Theses Inf. Syst.*, 2019.

[14] E. N. Hartiwati, J. S. Informasi, and F. I. Komputer, "phpMyAdmin," *Defin. Guid. to MySQL5*, vol. 5, no. 1, pp. 87–116, 2006, doi: 10.1007/978-1-4302-0071-0_6.

[15] R. Irsyad, "Penggunaan Python Web Framework Flask untuk Pemula," *Khazanah Inform. J. Ilmu Komput. dan Inform.*, vol. 4, no. 1, pp. 8–15, 2018.

[16] Putri Elfa Mas`udia, Rieke Adriati Wijayanti, Moh. Abdullah Anshori, Lis Diana Mustafa, Mochammad Sarosa, and Yani Ratnawati, "Pembuatan Website, Instagram Dan Akun Shopee Sebagai Media Marketing Produk Anyaman Bambu Di Desa Duwet," *Cakrawala J. Pengabdi. Masy. Glob.*, vol. 2, no. 3, pp. 220–226, 2023, doi: 10.30640/cakrawala.v2i3.1506.