International Journal of Engineering and Techniques -Volume 4, Issue 5, Sept - Oct 2018

RESEARCH ARTICLE OPEN ACCESS

Artificial Intelligence to Measure the Percentage Level of Similarity of Text in Compare Two documents

Bambang Krismalela[1], Nana Supiana[2], Toni Fathoni[3]

1(Computer Science, Budi LuhurUniversity, and Jakarta)

2 (Computer Science, Budi LuhurUniversity, and Jakarta)

3 (Computer Science, Budi LuhurUniversity, and Jakarta)

Abstract:

In the world of education sometimes occur plagiatrisme practice or plagiarism results of the study. Plagiatrisme or what is often called the act of plagiarism is plagiarism or making bouquets, opinions, and so forth from others and make it as his own essays and opinion. As a student who is making scientific writing a thesis, an act of plagiarism in copying data (copy and paste). The existence of scientific writing thesis title equation between students make students perform copying data or text, thus causing the same scientific writing because it comes from the same data, it is also supported abundance of resources on the internet. To detect the degree of similarity data source documents and source code can be several approaches that are already widely in use. Because plagiarism against text documents difficult to avoid. Therefore, many created a system that can assist in the detection palgiarisme text documents such as MOSS, Tessy, JPlag, CopyCatch, etc. In this study will describe several methods of detecting plagiarism, as the solution of the problem of plagiarism that has happened so far. The results of the implementation of the system can create Artificial Intelligence a high percentage of the degree of similarity in comparing two documents because of the occurrence frequency of words that are similar decision table system.

Keywords — **plagiarism document, Artificial Intelligence, percentage, level of similarity.**

I. INTRODUCTION

In the world of education sometimes there is often the practice of plagiarism in research and scientific writing for students. Plagiarism or plagiarism according to Permendiknas, (Prevention and Overcoming Plagiarism in Universities, No 7, Article 1 paragraph 1 2010) Plagiarism or what is often referred to as plagiarism is plagiarism or retrieval of articles, opinions, and so on from others and making it appear as an essay and own opinion.

Plagiarism in the world of education, (UniversitasPendidikan Indonesia 2012: 1-15) is usually the case if like a student who is making scientific writing there is a plagiarism action in copying and copying thesis with the many internet facilities, making it easier for students to take plagiarism.

Acts of plagiarism often appear in various versions, there are those who take the entire document of someone else's work and call it their own work, some write back to publish it, some only use some of the work of others by combining several other people's works.

Writing scientific papers is the most frequent case of plagiarism in the world of education conducted by students and teachers (Pikiran Rakyat, 02/03/2012), because of the supporting technology and the similarity between the titles of scientific writing between students, making students copy the text or data (copy and paste) in scientific writing thesis so as to enable the occurrence of the same scientific writing because it comes from the same data as the abundance of all sources of information only by accessing the internet makes more and more types of plagiarism in all forms.

Various methods have been carried out by researchers to reduce plagiarism. To minimize the practice of plagiarism, it is necessary to detect writing a paper. Therefore it is necessary to make an algorithm in the form of an application that can

detect the similarity of a document with other documents that are used as a comparison.

II. RELATED WORK

In order to make it easier to understand the material related to the writing of scientific articles, the writer presents it simply as follows.

A. Artificial intelligence

Artificial Intelligence (AI) or artificial intelligence is a branch of computer science that concentrates on automating intelligent behavior.

These principles include the data structure used in the representation of knowledge, the algorithms needed to apply that knowledge, as well as the language and programming techniques used in implementing it. From the above definition, it can be concluded that artificial intelligence (Artificial Intelligence) is a science that learns how to make computers in which there are knowledge needed to apply it, so that this computer can do the work done by humans.

B. The Concept of Artificial Intelligence

There are several concepts that must be understood in artificial intelligence, including (Efrain Turban, 2010):

1. Turing Test, Intelligence Testing Method. Turing Test is an intelligence testing method created by Alan Turing. In this concept, the questioner (human) will be asked to distinguish which is the human answer and which is the computer answer. If it cannot distinguish, then Turing argues that the machine can be assumed to be intelligent
2. Symbolic Processing, The original computer was designed for numerical processing, while humans in thinking and solving problems were more symbolic. The important nature of AI is part of computer science that processes symbolically and non-algorithmically in solving problems
3. Heuristic, Heuristic is a strategy to selectively search problem spaces, which guides the search process that we do along the path that has the greatest likelihood of success.
4. Withdrawal of Conclusions (Inferencing) AI tries to make the machine have the ability to think or consider (reasoning). thinking ability (reasoning) including the process of inferencing based on facts and rules using hueristik or other search methods.
5. Pattern Matching, AI works with a pattern matching method that attempts to explain objects, events or processes, in logical or computational relationships.

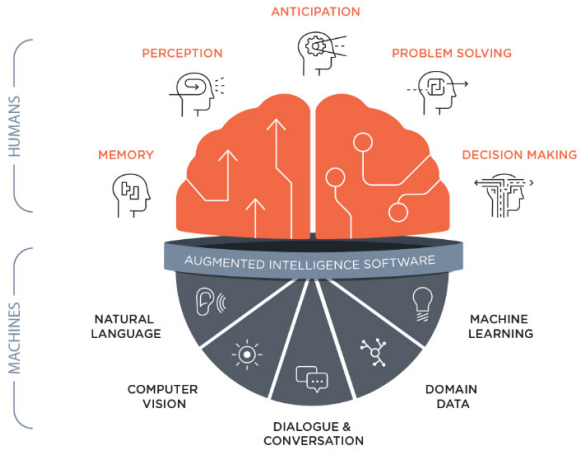

Fig 1. Aplications of Artificial Intelligence
Source: cognitivescale.com

C. Digital document

The document is an article that contains data and information. Usually, documents are written on paper and the information is written using ink using either hand or using electronic media (such as a printer).

Digital documents are any electronic information that is created, forwarded, transmitted, received, or stored in analog, digital, electromagnetic, optical, or the like, which can be viewed, displayed and or heard through a computer or electronic system, including but not limited to writing, sound or picture, map, design, photo or the like, letters, signs, numbers, access codes, symbols or perforations that have meaning or meaning or can be understood by people who are able to understand it. (Supriyono, 2013).

With the understanding that electronic documents are one of the digital library collections, the understanding of digital libraries can be a reference for electronic document management. In summary, it can be said that digital libraries not only provide electronic documents but also provide access to other available information sources. The term

attached refers to hardware, software and data related to the existence, and activities carried out in cyberspace. This term popularly shows the "place" where humans interact using computer networks, namely the internet.

There are at least five aspects that need to be considered in the management of electronic data, namely:

1. Format and security standardization;
2. Indexing and abstracting;
3. Providing links to other sources of information;
4. Analysis of access and citations;
5. Librarian readiness.

D. Plagiatrsm

Preceding the deeper discussion of the topic raised, the authors describe the definitions used in stating the act of plagiarism. Plagiarism is an act of abuse, theft or seizure, publishing, statement, or stating as one's own thoughts, ideas, writings or creations that actually belong to someone else (Rick Anderson, 2016). Plagiarism detection systems can be developed for:

1. Search for text data such as essays, articles, journals, research and so on.
2. Search for more structured text documents such as programming languages.

Types of Plagiarism, the following are the types of Plagiarism, namely:

1. Word-for-word plagiarism copies every word directly without being changed at all.
2. Plagiarism of authorship recognizing the work of another person as a result of his own work in a way to put his own name in place of the actual author's name.
3. Plagiarism of ideas recognize the thoughts or ideas of others.
4. Plagiarism of sources if an author uses quotations from other authors without specifying the source.

E. Winnowing Algorithm

Winnowing algorithm is a fingerprinting document algorithm that is used to detect copies of documents using hashing techniques (Schleime et al. 2014). To hash a document using k-gram, the length of the substring k where k is the value selected by the user. The document will be divided into possible k-grams and then the k-gram will be hashed. To select a fingerprint from a hashed result, the division is done using the w window, and the smallest value is selected.

From each window, the minimum or smallest hash value is selected. If there is a minimum value of more than one value, then choose from the right-hand window. Then save all the results of the selected hash which is the document fingerprint. Given a document, wanting to find the same substring among the documents, the properties that are carried out are:

1. If there is the same string whose length is equal to the length t, where t is the guarantee of the specified value threshold, then matching is detected.
2. Cannot detect multiple matches if shorter than the threshold disturbance, k.

The constant value t and $k \leq t$ are selected by the user. Avoid matching the same string below the value of the threshold value by considering hash k-grams.

Input from the document fingerprinting process is a text file. Then the output will be a set of hash values called fingerprint. This fingerprint will be used as a basis for comparison between text files that have been entered. One of the prerequisites of the plagiarism detection algorithm is whitespace insensitivity, and Winnowing algorithm has fulfilled these prerequisites, namely removing all irrelevant characters such as punctuation marks, spaces and other characters, so that later only characters in the form of letters or numbers will be processed further.

Broadly speaking, the following concepts of Winnowing algorithm work:

1. Removal of whitespace insensitivity.
2. Formation of gram circuits with size k.
3. Calculation of hash values.
4. Divide into certain windows.
5. Selection of several hash values into document fingerprinting.

F. Rabin-Karp Algorithm

The Rabin-Karp algorithm is a string search algorithm found by Michael Rabin and Richard Karp. This algorithm uses hashing to find a substring in a text (Junaidi, Fifit Alfiah, 2014: 3).

Hashing is a method that uses a hash function to convert a data type into several simple integers. It is called the "string search" algorithm and not "string matching" like Knuth-Morris-Pratt or Boyer-Moore because the Rabin-Karp algorithm does not aim to find strings that match the input string, but instead find patterns that match the input text. .

For text with length n and pattern with length m, the best computing time is O (n), while the worst is O ((n-m + 1) m). Steps in the Rabin Karp algorithm:

1. Eliminates punctuation and changes to source text and the words you want to search for words without capital letters.
2. Divide the text into grams which are determined by the k-gram value.
3. Finding the hash value with the rolling hash function of each gram formed.
4. Search for the same hash value between 2 texts.
5. Determine the equation 2 pieces of text with the Dice's Similarity Coefficient equation.

G. Levenshtein Algorithm

Levenshtein Distance was made by Vladimir Levenshtein in 1965, an edit distance calculation is obtained from the matrix used to calculate the number of string differences between two strings. 3 The calculation of the distance between these two strings is determined by the minimum number of change operations to make string A become string B.

Levenshtein algorithm, or often referred to as Levenshtein Distance or Edit distance is a search algorithm for the number of string differences found by Vladimir Levenshtein, a Russian scientist, in 1965. This algorithm is widely used in various fields, for example search engines, spell checking), speech recognition, dialect pronunciation, DNA analysis, counterfeit detection, and others.

Basically, this algorithm calculates the minimum amount of transformation of a string into another string which includes replacement, deletion, and insertion. This algorithm is used to optimize the search because it is very inefficient if it searches for each combination of string operations. Therefore, this algorithm is classified as a dynamic program in finding the minimum value. To calculate the distance (edit distance) a matrix (n + 1) × (m + 1) is used where n is the length of the string s1 and m is the length of the string s2.

III. METHODOLOGY

This research discusses about how to compare two(2) Document so that can be check the level of similarity of words and sentences or usually called plagiarism in the world of education. Here is the explanation the method for this research:

A. String Matching Classification

Strings are arrays of characters (numbers, alphabets or other characters) and are usually presented as array data structures. Strings can be words, phrases or sentences. Whereas string matching is interpreted as a problem to find a string character arrangement pattern in another string or part of the text content. String matching in Indonesian is known as string matching. (Stoimen, 2012).

A string search, also called string matching, is an algorithm to search for all occurrences of a short string pattern [0 ... n-1] called a longer string pattern text [0 ... m-1] called text. (Stoimen, 2012).

The operation of changing this string can be to change one letter to another, delete one letter from the string, or enter one letter into the string. These operations are used to calculate the number of differences needed to consider a string's match with the source string.

String Matching Framework in (Stoimen, 2012) search string matching is formulated as follows:

1. A text (text), which is a (long) string whose length is n characters.
2. Pattern, which is a string with length m.

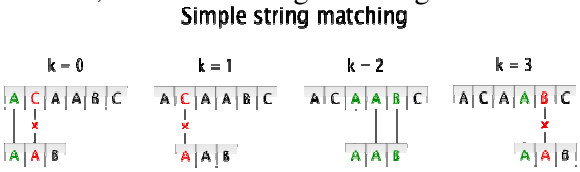

Fig 2. Illustration of Matching Algorithm (Source: Deeksha, Shenoy. *Articles:Boolean Text Search Queries And Their Processing*, 2011, may 20)

String matching (string matching) can be broadly divided into two, namely:

1. Exact string matching, is matching strings precisely with the arrangement of characters in matching strings having the number and

sequence of characters in the same string. Example: the step word will show compatibility only with the step word.

2. Inexact string matching or Fuzzy string matching, is a string matching in vague, meaning matching strings where matching strings have similarities where both have different character arrangements (maybe the number or sequence) but the strings have similarities both textual or writing similarities (approximate string matching) or phonetic string matching. Inexact string matching can still be divided into two, namely:
   a. String matching based on the similarity of writing (approximate string matching) is a string matching with the basis of similarity in terms of writing (number of characters, arrangement of characters in the document). The level of similarity is determined by the difference between writing two strings that are compared and the level of similarity is determined by the programmer. Example: compiler with the compiler, has the same number of characters but there are two different characters. If these two character differences can be tolerated as a writing error, the two strings are said to be suitable.
   b. Phonetic string matching is a string matching with basic similarities in terms of pronunciation even though there are differences in the writing of the two strings compared. Step example, with steppe, sttep, stepp, stepe. Exact string matching is useful if the user wants to search for strings in a document that is exactly the same as the input string. But if the user wants a search string that is close to the input string or there is an error writing the input string or search object document, then inexact string matching is useful.

B. Hashing

Hashing is a way to transform a string into a unique value with a fixed length (fixed-length) that serves as a marker of the string. The function to generate this value is called a hash function, while the resulting value is called a hash value. The use of hashing in a database search, if it is not hashed, the search will be performed by character-per-character on names whose length varies and there are 26 possibilities for each character. But the search will be more efficient after being hashed because the possibility of each number is different. The hash value is generally described as a fingerprint which is a short string consisting of letters and numbers that appear to be random (binary data written in hexadecimal).

C. Rolling Hash Method

Rolling Hash is a function that is used to generate a hash value from a gram circuit. In the beginning the Rolling Hash method was used in the Rabin-Karp Algorithm where this method was used to compare hashing values of all k-grams into a long string. However, the hashing process on each string along k will spend a long computing time if the k value is large (Schleimer, et al. 2014). For that Rabin Karp uses Rolling Hash where the H hash function ($c_1 \dots c_k$) is defined as follows:

Rolling Hashing Formula:

$$C1 * b^{k-1} + C2 * b^{k-2} + \dots + C_{k-1} * b + Ck$$

Information:
c: character ascii value
b: base (prime number)
k: lots of characters
To benefit from rolling hash the next gram hash value H ($c_2 \dots c_{k+1}$) can be done by:

$$H(c_2 \dots c_{k+1}) = (H(c_1 \dots c_k) - C_1 * b^{(k-1)}) * b + c_{k+1}$$

The formula for searching for the 2nd to nth hash. In the hash calculation of the nth gram, the hash value to gram n-1 is reduced by the first character value of gram n-1 then added with the last character value of the nth gram. That way there is no need to iterate from the first to the last index to calculate the hash value for the 2nd gram until the last. This can certainly save computing time when calculating the hash value of a gram.

D. Principle of Rabin-Karp Algorithm

Basically, the Rabin-Karp algorithm will compare the hash values of input strings and substrings in the text. If the same, then it will be compared once again to the characters. If not the same, then the substring will shift to the right. The

main key to the performance of this algorithm is the efficient calculation of the substring hash value at the time of shifting. The following is an example of how the Rabin-Karp algorithm works. Given input "cab" and text "aabbcaba". The hash function used for example will add the sequential value of each letter in the alphabet (a = 1, b = 2, etc.) and do modulo with the hash value obtained from "cab" is 0 and the first three characters in the text that is "aab" are 1.

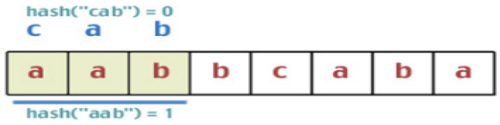

Fig 3. Early Fingerprint

The comparison results are not the same, then the substring in the text will shift one character to the right. The algorithm does not recalculate the substring hash value. This is where what is called rolling hash is to reduce the value of the characters that come out and add the value of the incoming character so that the time complexity is relatively constant at each shift.

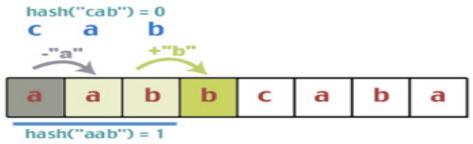

Fig 4. Shift Fingerprint

After shifting, the hash value is obtained from the fingerprint "abb" (abb = aab - a + b) to two (2 = 1 - 1 + 2).

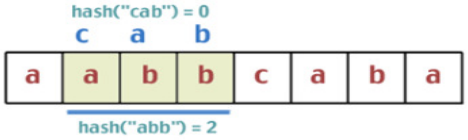

Fig 5. Second comparison

The comparison results are also not the same, so the shift is done. Likewise with the third comparison. In the fourth comparison, the same hash value is obtained.

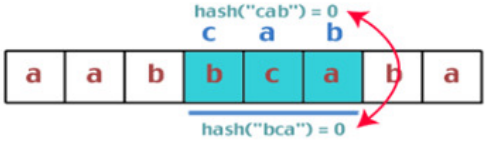

Fig 6. The fourth comparison (same hash value)

Because the hash value is the same, a character string is compared per character between "bca" and "cab". The result is that the two strings are not the same. Back substring shifts to the right.

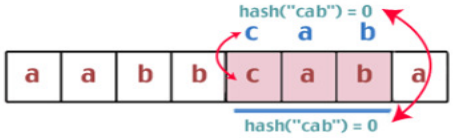

Fig7. The fifth comparison (string found)

In the fifth comparison, the two hash values and string forming characters match, so the solution is found. From the calculation results, the time complexity required is O (m + n) where m is the length of the input string and n is the number of loops performed to find a solution. This result is far better than the time complexity obtained using the brute-force algorithm, O (mn).

E. Document Extraction

The text that will be carried out by the mining text process generally has several characteristics including having a high dimension, there is noise in the data, and there is a text structure that is not good. The method used in learning a text data, is to first determine the features that represent each word for each feature in the document. Before determining the features that represent, a preprocessing stage is needed in general in mining text on documents, namely case folding, tokenizing, filtering, stemming, tagging and analyzing.

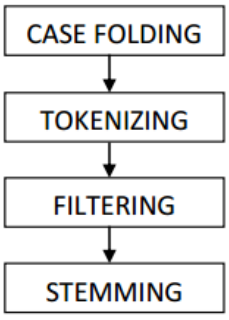

Fig8. Stage of Preprocessing

F. Case folding and Tokenizing

Case folding is changing all letters in a document into lowercase letters. Only the letters 'a' up to the letter 'z' are accepted. Characters other than letters are removed and are considered delimiter. The tokenizing or parsing stage is the stage of cutting

the input string based on each word that composes it.

Fig 9. Tokenizing process

G. Filtering

Filtering is the stage of taking important words from the token results. Can use the stoplist algorithm (throw out less important words) or wordlist (save important words). Stoplist or stopword are non-descriptive words that can be discarded in the bag-of-words approach. Examples of stopwords are 'the', 'and', 'on', 'from' and so on.

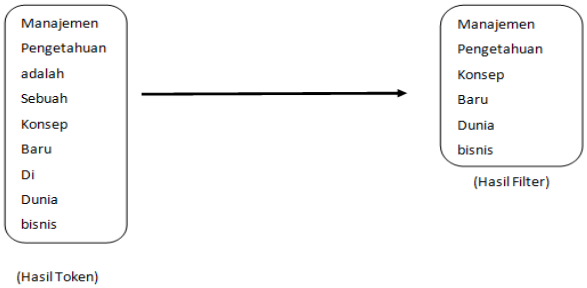

Fig 10. Filtering Process

H. Stemming

Stemming stage is the stage of finding the root word of each word resulting from filtering. At this stage the process of returning various forms of words is carried out into the same representation. This stage is mostly used for English text and is more difficult to apply to Indonesian texts. This is because Indonesian does not have a permanent standard form formula.

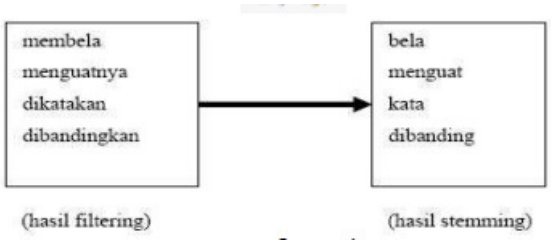

Fig 11. Stemming process

I. Rabin-Karp Complexity

The Rabin Karp algorithm has complexity O (nm) where n, of course, is the length of the text, while m is the length of the pattern. So where does it compare to brute force match? Well, the rough complexity of the suitable style is O (nm), so there doesn't seem to be much profit in performance. But it is considered that the complexity of Rabin-Karp is O (n + m) in practice, and that makes it a little faster, as shown in the graph below:

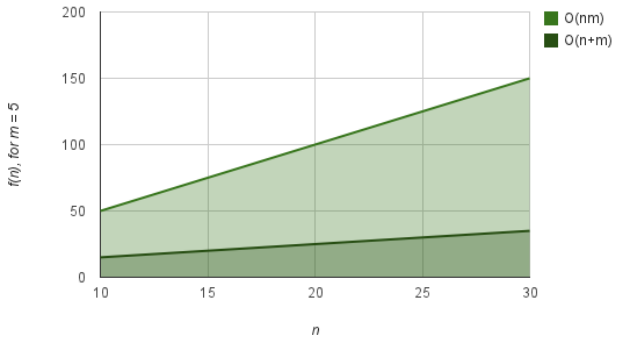

Fig 12. Rabin-Karp's complexity O (nm), but yield O (n + m) (Stoimen, 2012)

J. Empirical Method of Levenshtein Distance

The Levenshtein Distance process is done by making a matrix of two words that are compared (the wrong word with the standard word). From each wrong word, the distance is found with all the standard words in the database and the Levenshtein Distance value is obtained.

There are 3 main types of operations that this algorithm can do, namely:

a. The conversion operation character Character conversion operation is an operation to swap a character with another character string for example, the author wrote "yamg" to "a". In this case the character "m" is replaced with the letter "n".
b. Character Addition Operations Adding characters means adding characters to a string. For example the string "to" becomes a string "to", the addition of the character "a" at the end of the string. Adding characters is not only done at the end of the word, but can be added at the beginning or inserted in the middle of the string.
c. Character Removal Operation Character deletion operations are performed to eliminate characters from a string. For example, the last string

"barur" character is omitted so that it becomes a 'new' string. In this operation, the "r" character is deleted. To determine Levenshtein Distance between two words we need the following matrix equation:

$$lev\ a,b(i,j) = \begin{cases} \max(i,j) & if\ \min(i,j) = 0 \\ \min \begin{cases} lev\ a,b(i-1,j)+1 \\ lev\ a,b(i,j-1)+1 \\ lev\ a,b(i-1,j-1)+1(ai \neq bj) \end{cases} & otherwise \end{cases}$$

lev a, b = is the matrix lev a, b
i = is a matrix row
j = is a matrix column

While to calculate the similarity value is:

$$Similarity = \left\{1 - \frac{edit\ distance}{maxLength(str1,str2)}\right\}.$$

Editdistance is the result of the preposesing that has been done earlier or Levenshtein distance maxLength is the number of strings from the longest word.

Approximate String Matching is a technique for matching patterns to strings by means of approach, the performance of this method does not have to be similar to the actual enough with just the approach. In this approach, there are three types of operations that are used to transform a string into another string. These operations include deletion, insertion and replacement operations. These operations are used to calculate the number of differences needed to consider a string's match with the source string. The amount of the difference is obtained from the sum of all changes that occur from each operation. The use of these differences is applied in various algorithms, such as Hamming, Levenshtein, Damerau-Levenshtein, Jaro-Winkler, Wagner-Fischer, and others.

IV. RESULT AND DISCUSSION

A. *System Flowchart Design*

Design is carried out to determine the details of the algorithm that will be stated in a program. Procedural design on Artificial Intelligence systems to compare the level of similarity of 2 documents is described using a flowchart. Flowchart procedural design data processing application consists of a system process flowchart, Preprocessing flowchart, read document flowchart, document info flowchart, flowchart analyze result paper, similarity level flowchart, and flowchart showing a graph of similarity percentage.

The explanation and description of each flowchart are as follows:

1.Flowchart of the system process.

The system process flowchart describes the steps taken by the user to perform the process of detecting the data similarity of a document to the application being built. The flowchart description of the system process can be seen in figure below.

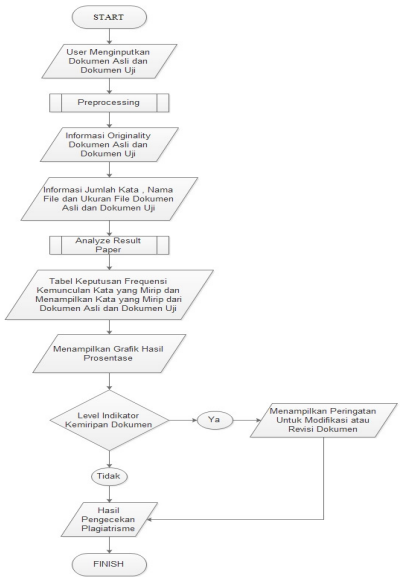

Fig 13. Artificial Intelligence system flowchart

2. Flowchart Preprocessing

Prepocessingflowcharts describe the steps of the system when the user uploads documents. The description of the preprocessing system upload flowchart can be seen in this figure.

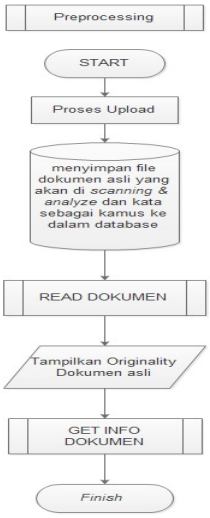

Fig 14. Flowchart preprocessing upload documents

B. ***Database Design***

Database design is used as a data storage media used in applications and databases help programming in displaying data.

The normal form of a relational database is achieved through several stages called the normalization process. Unnormalized steps, First Normal Form (1NF), Second Normal Form (2NF) to the Third Normal Form (3NF) form. Author will discuss in section 3NF directly to shorten.

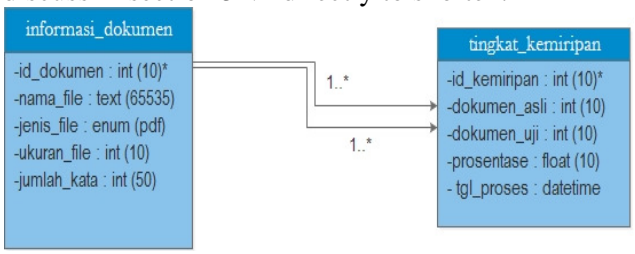

Fig 15. *Third Normal Form*(3NF)

Can be explained Third Normal picture Form (3NF) is a table that has been normal and to be used in the system to be built and consists of 2 tables, namely table information_documents and similarity tables.

C. ***Evaluation of system***

1. Display from the "Scanning File" menu to go to the document scanning page that will be analyzed

Fig 16. Check Plagiatrsm or similarity documents

Based on fig. 16 above the system is able to display document originality and document analyze result.

2. Display from the "Report" menu to go to the history page

Fig 17. Report and history of system

Based on fig. 17 above the system is able to display history pages of any files that have been analyzed analyze on the system.

V. CONCLUSIONS

In this paper conducted by this author has 3 point conclusions, that is:

1. Can compare documents to measure the percentage level of similarity of writing by using a file scanning system to produce a percentage analysis of the level of similarity between documents.

2. The system can produce a percentage level of similarity that is used optimally to determine how much the similarity of a document, and can know the status level similarity and be able to provide messages for modification if the level of similarity is high.
3. Able to reduce the action of plagiarism in the academic because before publishing or publishing will be checked first the authenticity or Originality of a document with the comparison document by a similarity checker system.

ACKNOWLEDGMENT

Thanks to all noble university postgraduate computer science lecturers who have guided and shared their knowledge to our BambangKrismalela, Nana Supiana and Toni Fathoni so that they are able to complete postgraduate studies well and complete scientific writing as a graduation requirement.

REFERENCES

1. *Efraim. Turban, Jay E.Aronson, Ting Peng. Liang, Decision Support Systems and Intelligent System, Issue 7, Vol 2, Yogyakarta: Andi Offset.*
2. *Supriyono.Management of Print, Electronic Journal and special materials in the Libraryof UGM, Yogyakarta: Universitas Gajah Mada, 2013.*
3. *Anderson. Rick, The difference between copyright infringement and plagiarism - and why it matters, ACORN: The Journal of Perioperative Nursing in Australia, Vol. 29, No. 4, Summer 2016: 50-51.*
4. *Schleimer. Saul, Daniel S. Wilkerson, and Alex Aiken, Rabin Karp: Local Algorithms for Document Fingerprinting, retrieved 26 05 Januari 2014. http://www.theory.stanford.edu/~aiken/publications/papers/sigmod03.pdf.*
5. *Junaidi, Fifit Alfiah, Collaborative Methode Model DalamMembandingkanDokumenUntukMengukurProsentaseKemiripan, ISSN: 2355-'941 KNSI 2014, Februari 2014.*
6. *Stoimen, computer-algorithms-rabin-karp-string-searching. retrieved 24 November 2014, http://www.stoimen.com/blog/2012/04/02/computer-algorithms-rabin-karp-string-searching/.*