

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/272165855>

DETECTING PLAGIARISM JOURNAL WITH SHERLOCK ALGORITHM

Conference Paper · September 2014

CITATIONS

0

READS

524

3 authors, including:



Li Za

Universitas Mulawarman

39 PUBLICATIONS 31 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Distance [View project](#)



2016 FMIPA-UNMUL [View project](#)

DETECTING PLAGIARISM JOURNAL WITH SHERLOCK ALGORITHM

Heliza Rahmania Hatta¹, Muhammad Rasyid², Muhamad Azhari³

Computer Science, Faculty of Mathematics and Natural Science, Mulawarman University, Samarinda,
Indonesia^{1,2,3}

Email : heliza_rahmania@yahoo.com¹, rasyid37@gmail.com², ktob34@yahoo.com³

Abstract

Plagiarism is not allowed in the research journal. Therefore the similarity of research journal must be checked. Similarity is usually checked manually, so it needs long time for verification. The purpose of this research is to create a software for detecting similarity of research journal. Journal plagiarism detection software created in this research is done by implementing sherlock algorithm. Sherlock algorithm can detect document similarity by comparing similarity of each sentence inside a document with each other sentence in other documents. Sentence similarity detection is based on the same shared keyword between compared sentences. Result of test concluded that this software can detect similarity of research journal.

Keywords: Journal, Sherlock, Plagiarism

1 INTRODUCTION

Knowledge is growing with the expanding of information technology. This expansion provides information overload on media such as research journals. Implementation of information technology provide journals documentation in softcopy format. Journals documentation has a purpose to make the journal scans become evident in document format and can be used as research references.

Journals that will be documented need to be checked before regarding its scientific value, which is there must be no plagiarism in it. For knowing if there is plagiarism or not, there must be a check for similarity degree of journals.

Usually, detection process is done manually, by reading those journals one by one. This method of detection is not very efficient, because it needs very long time. Therefore there should be a computerized similarity journal documents detection software which can make the process for detecting similarity of research journals becomes faster.

This journal plagiarism detection software was implemented using Sherlock algorithm. Sherlock algorithm was used because this algorithm can detect document similarity by comparing similarity of each sentence inside a document with each other sentence in other documents. Therefore this algorithm is considered capable for detecting journal similarity [1].

According to the problem mentioned above, the main topic of this paper is to do research about journal similarity detection using sherlock algorithm. This research aims to create a computerized journal similarity detection using Sherlock algorithm.

2 STRUCTURE OF WRITING

The remainder of the paper is organized as follows : In section 3, we explain the proposed methods, observation and similarity detection process. In section 4, the software testing are explained. Finally, in section 5, our conclusions are outlined.

3 METHODOLOGY

3.1 Sherlock Algorithm

Sherlock algorithm is an algorithm for detecting plagiarism by comparing similarity between one sentence with other sentence. Sherlock algorithm indicates that if there are two sentences which have different sets of keywords then these two sentences have different content. The opposite is if two sentences have same sets of keywords then these two sentences have same content. Detection process is done by comparing each sentence in one document with each other sentence in another document [2].

Calculation of similarity score is done by getting the number of shared keyword from sentence A that found in sentence B and then divided it with total words in sentence A. Calculation similarity score of sentence B is done by dividing the number of shared keywords between sentence A and B with the total words in sentence B. Average score from this calculation is the sentence similarity score. If the calculation has a result of more than 80 then the compared sentences are indicated as similar [2].

Equation example of sherlock algorithm:

Sentence A : Software for detecting research document similarity

Sentence B : Computerized software for detecting research document similarity

$$\text{Similarity score} = \frac{\left(\frac{6}{6} * 100\right) + \left(\frac{6}{7} * 100\right)}{2} = \frac{100 + 85,7}{2} = 92,85$$

There are 6 shared keywords between sentence A and sentence B which are, “software”, “for”, “detecting”, “research”, “document”, “similarity”. In sentence A there are 6 words. In sentence B there are 7 words, so the average similarity score is 92,85. Sherlock algorithm flowchart is shown in Figure 1.

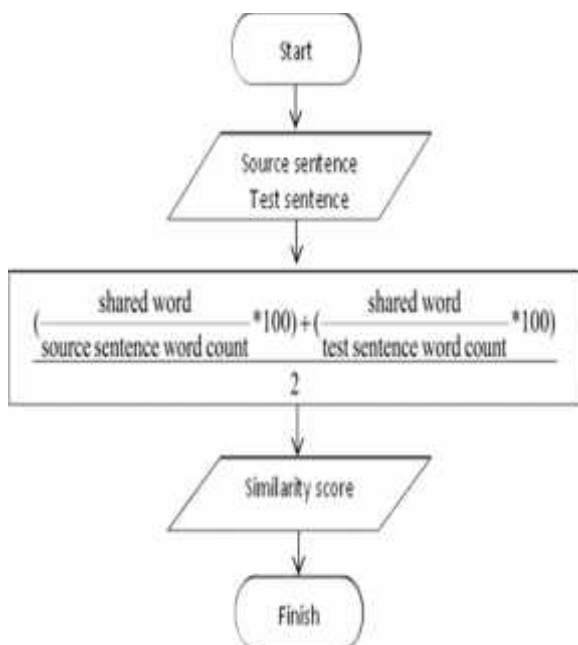


Figure 1 Sherlock Algorithm Flowchart

3.2 Similarity Percentage Calculation

Similarity percentage is the comparison degree of percentage similarity between tested documents, Figure 2. This similarity will give a result of a score which will be a reference for determining percentage similarity degree on a tested document. The number of similarity percentage is affected by the similarity degree from tested document. If similarity percentage is higher, then similarity degree will be higher [3]. Percentage similarity calculation between two documents using equation [4] :

$$\text{Similarity}(\%) = \frac{\text{Total amount of detected sentences}}{\text{Total amount of sentences in document}} * 100\%$$

(1)

Equation for calculating similarity percentage by dividing number of similar sentences detected with total sentences in document. There are 3 way for determining similarity between documents [5]:

1. Testing result lower than 30% means those document considered has little plagiarism.
2. Testing result between 30-70% means those document considered has moderate plagiarism.
3. Testing result more than 70% means those document considered has heavy plagiarism.

3.3 Observation

Observation aimed for collecting data that needed before software implementation. Data collected were research journal documents obtained from the internet in doc and docx formats. This research used 15 journal documents as test documents and 25 journal documents as source documents. Journal documents that were collected mostly were journals with similar topic.

3.4 Similarity Detection Process

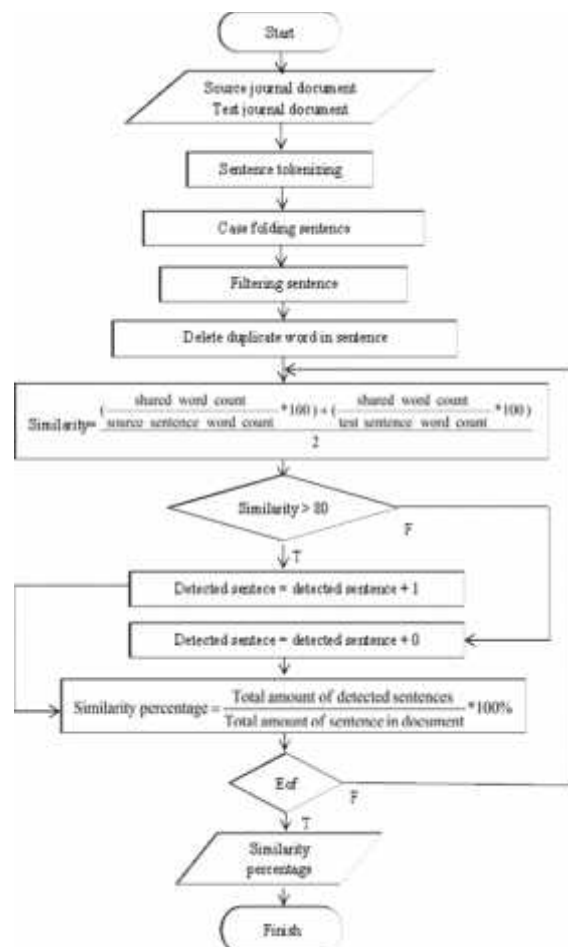


Figure 2 Similarity Detection Flowchart

Firstly user input a test document as a journal with doc or docx format and journal's year into preprocessing step. Preprocessing aimed to get keywords from document. Preprocessing step of a document consists of :

1. Tokenizing document into a list of sentence using dot separator (.)
2. Case folding sentence by changing each capital letter into lower case.
3. Filtering sentence process for stopword removal.
4. Deleting duplicate word to get keywords from sentence.

The next step is similarity detection of each sentence inside document using sherlock algorithm. If the similarity score was more than 80, then the total number of detected sentences will be accumulated. Percentage similarity was calculated by dividing the number of detected sentences with the total sentences in document.

4 RESULTS AND DISCUSSIONS

Testing software was done by detecting journal similarity. Testing process was done by comparing similarity between test journal document with each source journal. This aimed to test software for detecting journal similarity.

Testing was done by inputting journal document. Journal can be inputted by pressing browse button. Software will show a dialog for choosing documents as seen in figure 3.

Figure 3 is the dialog shown for choosing journal document. User can choose one of the journals listed and then pressing open button. Document that can be inputted are documents with doc and docx format.

Figure 4 is the input journal result. Document that had been inputted automatically went through preprocessing step. The preprocessing result and preprocessing time can be seen for examination. The next step was detecting document percentage similarity by comparing with each document in the database. Document comparing step was done by pressing detection button.

The next step was similarity detection by comparing test journal with each journal in the database. If the detection process is finished, the result of journal similarity is ranked from highest percentage into lowest percentage as shown in figure 5.

Figure 6 showed the detailed detection result. In detailed detection, the software detected similarity by comparing each sentence in test journal with each other sentence in journal chosen by user. If there are similar sentences found, it will be marked by red color.

Journal database form is shown in figure 7. User can manage data in journal database. User can input, edit and delete journal data. Input data is done by pressing button for input data, and then software showed form for input journal data as shown in figure 8.

Editing and deleting can be done by choosing the journal data first. If user presses edit button, software will show a form as shown on figure 8 that consists of journal data which is going to be edited. User can edit journal data and then press button for saving data. Journal data can be deleted by choosing data journal which is going to be deleted and then pressing button to delete data.

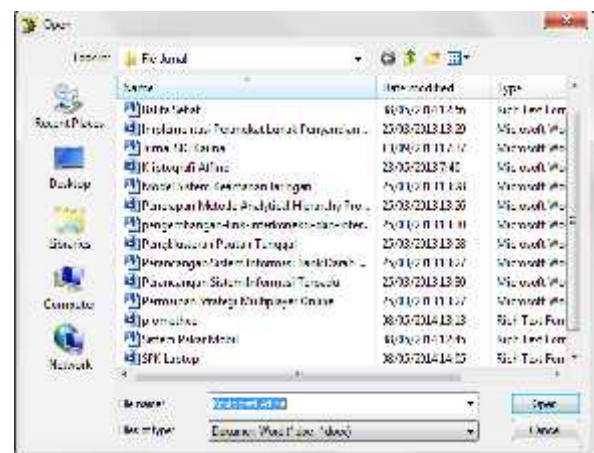


Figure 3 Input Journal

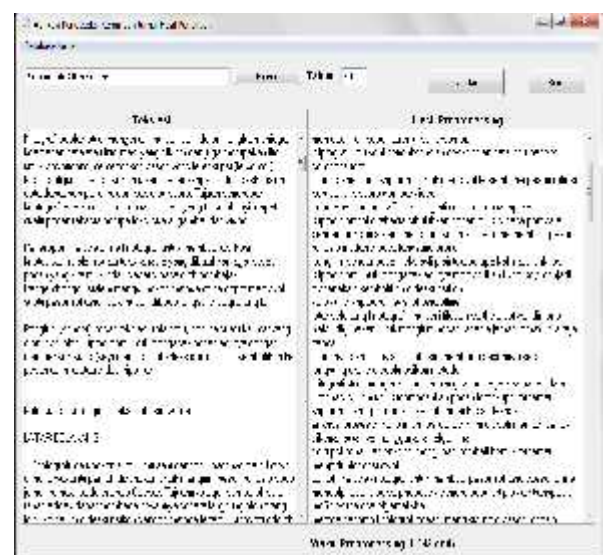
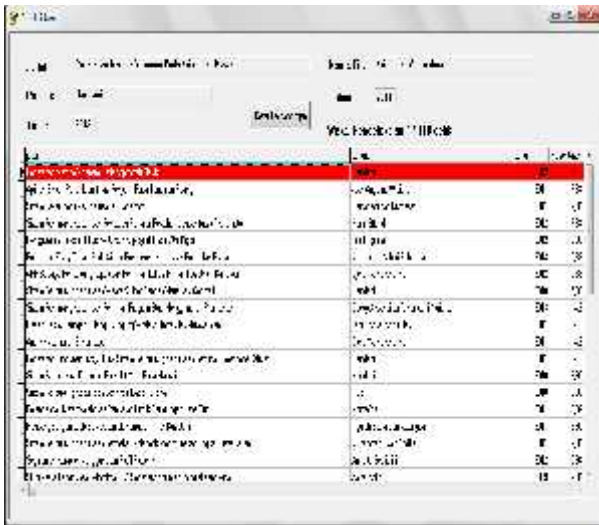


Figure 4 Input Journal Result

ARTICLES OF BALI INTERNATIONAL SEMINAR ON SCIENCE AND TECHNOLOGY (BISSTECH) II 2014
 “Fundamental and Applied Research for Industrial Sustainability: Food, Agrochemical, and Information and
 Communication Technology (ICT)”
 September 2-4, 2014, BALI-INDONESIA



No	Judul	Waktu	Persentase
1	Deteksi kemiripan dokumen	13,59	54,90
2	Deteksi kemiripan dokumen	13,59	54,90
3	Deteksi kemiripan dokumen	13,59	54,90
4	Deteksi kemiripan dokumen	13,59	54,90
5	Deteksi kemiripan dokumen	13,59	54,90
6	Deteksi kemiripan dokumen	13,59	54,90
7	Deteksi kemiripan dokumen	13,59	54,90
8	Deteksi kemiripan dokumen	13,59	54,90
9	Deteksi kemiripan dokumen	13,59	54,90
10	Deteksi kemiripan dokumen	13,59	54,90
11	Deteksi kemiripan dokumen	13,59	54,90
12	Deteksi kemiripan dokumen	13,59	54,90
13	Deteksi kemiripan dokumen	13,59	54,90
14	Deteksi kemiripan dokumen	13,59	54,90
15	Deteksi kemiripan dokumen	13,59	54,90
16	Deteksi kemiripan dokumen	13,59	54,90
17	Deteksi kemiripan dokumen	13,59	54,90
18	Deteksi kemiripan dokumen	13,59	54,90
19	Deteksi kemiripan dokumen	13,59	54,90
20	Deteksi kemiripan dokumen	13,59	54,90

Figure 5 Detection result

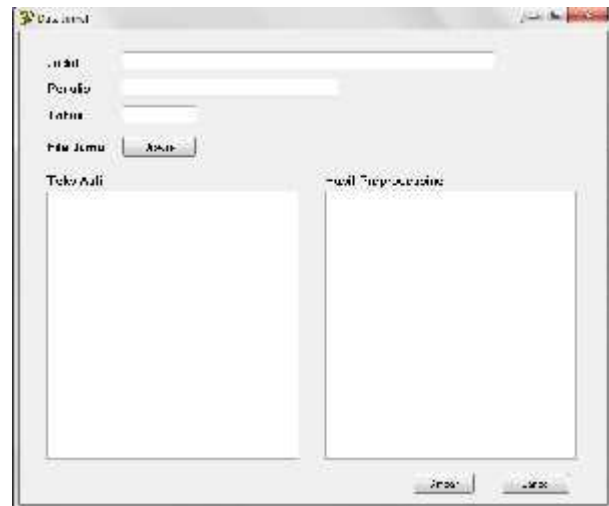
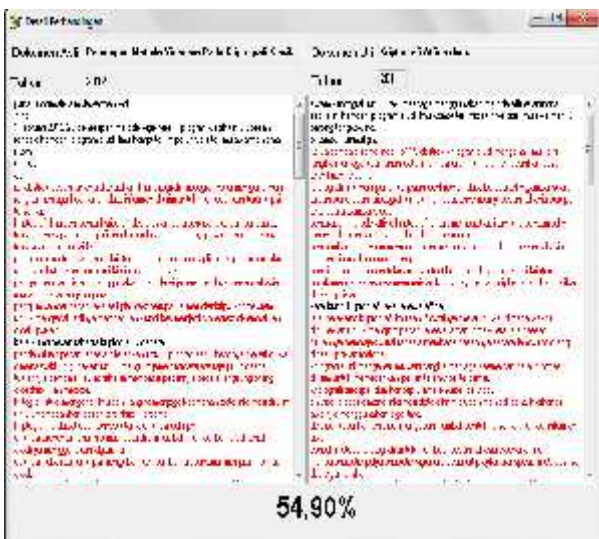
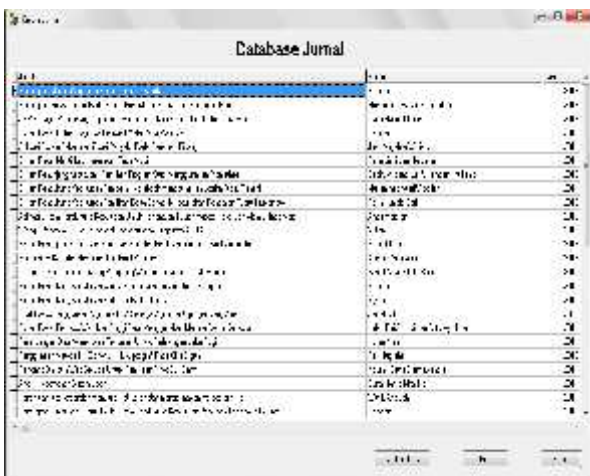


Figure 8 Input Document Into Database



No	Judul	Waktu	Persentase
1	Deteksi kemiripan dokumen	13,59	54,90
2	Deteksi kemiripan dokumen	13,59	54,90
3	Deteksi kemiripan dokumen	13,59	54,90
4	Deteksi kemiripan dokumen	13,59	54,90
5	Deteksi kemiripan dokumen	13,59	54,90
6	Deteksi kemiripan dokumen	13,59	54,90
7	Deteksi kemiripan dokumen	13,59	54,90
8	Deteksi kemiripan dokumen	13,59	54,90
9	Deteksi kemiripan dokumen	13,59	54,90
10	Deteksi kemiripan dokumen	13,59	54,90
11	Deteksi kemiripan dokumen	13,59	54,90
12	Deteksi kemiripan dokumen	13,59	54,90
13	Deteksi kemiripan dokumen	13,59	54,90
14	Deteksi kemiripan dokumen	13,59	54,90
15	Deteksi kemiripan dokumen	13,59	54,90
16	Deteksi kemiripan dokumen	13,59	54,90
17	Deteksi kemiripan dokumen	13,59	54,90
18	Deteksi kemiripan dokumen	13,59	54,90
19	Deteksi kemiripan dokumen	13,59	54,90
20	Deteksi kemiripan dokumen	13,59	54,90

Figure 6 Detailed Detection Result



No	Judul	Waktu	Persentase
1	Deteksi kemiripan dokumen	13,59	54,90
2	Deteksi kemiripan dokumen	13,59	54,90
3	Deteksi kemiripan dokumen	13,59	54,90
4	Deteksi kemiripan dokumen	13,59	54,90
5	Deteksi kemiripan dokumen	13,59	54,90
6	Deteksi kemiripan dokumen	13,59	54,90
7	Deteksi kemiripan dokumen	13,59	54,90
8	Deteksi kemiripan dokumen	13,59	54,90
9	Deteksi kemiripan dokumen	13,59	54,90
10	Deteksi kemiripan dokumen	13,59	54,90
11	Deteksi kemiripan dokumen	13,59	54,90
12	Deteksi kemiripan dokumen	13,59	54,90
13	Deteksi kemiripan dokumen	13,59	54,90
14	Deteksi kemiripan dokumen	13,59	54,90
15	Deteksi kemiripan dokumen	13,59	54,90
16	Deteksi kemiripan dokumen	13,59	54,90
17	Deteksi kemiripan dokumen	13,59	54,90
18	Deteksi kemiripan dokumen	13,59	54,90
19	Deteksi kemiripan dokumen	13,59	54,90
20	Deteksi kemiripan dokumen	13,59	54,90

Figure 7 Journal Database

Table 1 showed performance testing of software for detecting journal similarity. Detection time is the time needed for comparing similarity between test document with each source document in database. According to table 1, the average detection time of this software for comparing with 25 source documents is 13,59 second, so the average time needed for detecting one document is 0-54 second.

According to table 1, document size did not affect the detection time. There were larger documents size with faster time and smaller documents size with slower time. This happened because documents had been through preprocessing step before the similarity detection.

Similarity percentage in table 1 was the highest percentage picked from similarity detection result. Although some of tested documents have similar topic with source document, most of documents tested had a score below 30% percentage similarity because there are few similar sentences detected. However there is also test document with similarity percentage higher than 30% detected because of high sentence similarity.

Table 1 Performance Testing

Test Document	Size (KB)	Time (s)	Similarity Percentage
Test Document 1	2675	13,01	7,22 %
Test Document 2	947	7,61	11,65 %
Test Document 3	2336	8,45	15,72 %
Test Document 4	410	7,16	54,90 %

Test Document 5	405	15,89	10,93 %
Test Document 6	1659	28,34	19,42 %
Test Document 7	1194	14,43	13,84 %
Test Document 8	651	12,62	24,04 %
Test Document 9	985	16,77	20,83 %
Test Document 10	1167	7,69	11,81 %
Test Document 11	1641	13,08	9,00 %
Test Document 12	751	13,46	14,80 %
Test Document 13	577	11,15	15,03 %
Test Document 14	1759	17,84	10,07 %
Test Document 15	1678	16,38	15,44 %

- [4] Mutiara, A.B. and Agustina, S. "Anti Plagiarism Application With Algorithm Karp-Rabin". Thesis in Gunadarma University, Jakarta, 2008.
- [5] Sastroasmoro, S. "Beberapa Catatan tentang Plagiarisme". Departemen Ilmu Kesehatan Anak Fakultas Kedokteran Universitas Indonesia, Jakarta, 2010.

Testing results indicated that although some journals had similar topic did not mean that it had plagiarism inside. Therefore this software have detailed detection feature for identification and examination purpose to minimize this problem.

5 CONCLUSIONS

According to software implementation, it was concluded that Journal plagiarism detection software using Sherlock algorithm can detect similarity of research journals with an average detecting time of 13-59 second. This software can rank journal similarity from the highest percentage into the lowest percentage. Although some journals have similar topic, it did not mean that it had high plagiarism, therefore this software can show detailed detection result by showing red marked similar sentences for easier identification. This research used 25 source documents for sample. If there are more source documents, then the chance and effectiveness for finding similar journals will be higher, but it will take slower detection time.

6 REFERENCES

- [1] Rasyid, M. "Pendeteksian Kemiripan Jurnal Hasil Penelitian Dengan Menggunakan Algoritma Sherlock". Skripsi Ilmu Komputer, Universitas Mulawarman, 2014.
- [2] White, D. R. and Joy, M. S. "Sentence-Based Natural Language Plagiarism Detection". ACM Journal of Education Resources, Vol 4 No. 4 pp.1-20, 2004.
- [3] Surahman, A. M. "Perancangan Sistem Penentuan Similarity Kode Program Pada Bahasa C Dan Pascal". Jurnal Sistem dan Teknologi Informasi Universitas Tanjungpura, Vol 1 No. 1, 2013.

