Implementation of the Levenshtein Distance Method and Similarities in Checking the Equal Content of the Document Text

Yo Ceng Giap, Ma'mun Johari, Pebrianto Indrawan, Dedih

Abstract ***The development of information technology is growing rapidly and giving positive and negative impact. One positive impact is the ease of searching and exchanging information. Ease that is often misused by some people in select jobs. The purification is one of the negative information technologies. Therefore plagiarism detection needs to be done to reduce plagiarism and raise awareness to get the work of others. The method used by researchers in checking the similarity of the contents of the text of the document is Levenshtein Distance and Similarities. The design of this application uses a storyboard consisting of the initial access page, the main menu and the account creation menu created using Microsoft Visual Studio 2010 as the main software and the presence of Visual Basic language coding as an application builder. Testing the system using the white box method by analyzing the flow of applications and black boxes by testing all the buttons on the application and using a questionnaire to know the opinions of users about the applications that have been made. Based on the result of research can be concluded that with existence this plagiarism detection application helps to improve understanding and awareness which has made it easier for users to get information about plagiarism detection in order not to take action, a solution for users to check the percentage of plagiarism, contained in the contents of the document text owned by the user or even in the web browser internet, who want to check the results of the difference percentage by the user.***

Index Terms**: *Applications, Information Technology, Text Document, Detection, Plagiarism, Research***

1. INTRODUCTION

Technological developments in managing information develop rapidly and provide good things, it is an easy way to get information. The convenience is often misused to complete a job as one of the things that are not good from the development of technology.Information technology makes it easy to store a document that is efficient and easy to find in terms of retrieving text content, the document retrieval content is often used to complete the tasks through copy-paste-modify techniques without the need to learn and explore the material first.[1]

Revised Manuscript Received on December 22, 2018.

Yo Ceng Giap Buddhi Darma University,Indonesia
Ma'mun Johari STMIK Muhammadiyah Banten,Indonesia
Pebrianto Indrawan, STMIK Muhammadiyah Banten,Indonesia
Dedih STMIK Kharisma Karawang,Indonesia Corresponding author
E-mail:cenggiap@buddhidharma.ac.id

Plagiarism or retrieval of text content of documents is easy to do, simply by copy-paste-modify the contents of the text of the document or a portion of the entire contents can be said that the document is the result of retrieving text content from other documents [2].Taking the contents of the text on plagiarism is one thing that needs to be avoided. To avoid this by the work of others, by respecting and giving the award before writing the paper. It is known that information knowledge is developed based on one thought previously made. So that for anyone at the time of writing a paper, must mention the source of the information is an act of respecting and acknowledging in appreciation for an article. Things that need to be considered for the honesty of science with the aim of not reducing the value of a written work. Mention correctly, the source of thoughts made by others to use in writing a quote, the quote is to see a part of someone's written work.

with the scope: checking similarities with the Levenstein Distance method is one method to check the similarity of the text content of documents, similarities as a percentage formula used to get the results of checking the similarity of the text content of the document. This study uses the contents of documents or articles published on the internet by copy-pasting or in the form of files made into .txt, .doc, .pdf from text, and aims to: check the similarity of the text content of the document to reduce the level of plagiarism and find out the percentage of plagiarism.

II. LITERATURE REVIEW

A. Plagiarism

Plagiarism is a criminal act that often occurs in the world of education. Plagiarism itself comes from the Latin word (Plagiarus) which means kidnapper and (Plagiare) which means stealing. So, simply plagiarism means taking the idea of the idea of the contents of a person's sentence and used it as the result of his own work without including the source where an author quoted it. [3]

Types of plagiarism based on the classification include:

This type of plagiarism is based on stolen aspects, namely the category of idea plagiarism, content plagiarism, word plagiarism, sentence, paragraph, and total plagiarism. The classification is based on intentional or not plagiarism which is deliberate plagiarism and accidental plagiarism.

Based on plagiarism patterns namely word for word plagia

Retrieval Number:F10070476S519/19 ©BEIESP

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

rism and mosaic plagiarism.
Classification is based on the proportion value or percentage of words, sentences, paragraphs hijacked, namely[2]:

1. Mild plagiarism, plagiarism which is the amount of proportion or percentage of words, sentences, paragraphs hijacked do not exceed 30 percent (<30%).
2. Moderate plagiarism, plagiarism that amounts to proportions or percentages of words, sentences, paragraphs that are hijacked between 30-70 percent.
3. Severe plagiarism, plagiarism which amounts to proportions or percentages of words, sentences, paragraphs that are hijacked by more than 70 percent (> 70%).

It can be concluded that the writer based on the above understanding that Plagiarism is an act of theft of ideas from the contents of sentences that have been quoted by someone but not used the writing of the source name after being taken from the original source then it can be said that the person did the plagiarism of a written work from an existing source of information.

a. Similarity

Similarity is a condition or property that can be measured between two or more texts, which determines the level of similarity between the two texts. Similarities can range from 0% (no relationship at all) to 100% (documents are identical). Also note that two similar texts do not need to share content, not word by word or expressed in other words. They may only discuss the same topic or be written in the same language. [4]
It can be concluded that the author based on the above understanding that Similirarity is one of the ways or formulas after obtaining the results of calculating the diff distance from the levensthtein distance which will later be used in determining the percentage of similarity values of the two documents using the text content equation.[5, 17]

b. Document

The term documentation of the word document, which is in language The Dutch are called documents, in English called documents. If we will use English so the term document can be said work (documents) and nouns (documents). Verbs to document meaning Provide documents, prove by showing document. As a noun, the document means information, data recorded or sold in the vehicle complete its meaning For learning, testimony, research, recreation and the like. Thus, documents can have different connotations slightly different scope. [6,16, 18]
It can be concluded that the author is based on the above understanding that the document is one of the things where information from sources is needed by an agency, organization, or country. Without documents we will lose the data needed for the needs of a group's activity activities in the future.
One of the document functions is as a reminder that is stored in various forms, it needs an efficient and effective retrieval system. Efficient here means that a situation when the completion of a job is carried out accurately and accurately without wasting time, effort and costs. The effective meaning is a condition in choosing the method and equipment that is used appropriately so that the desired goal can be achieved with satisfactory results.[7]

c. Text mining

Text mining to accommodate data in the form of text where data is usually revealed from documents, and issued to find documents that can be analyzed from the connection between documents. Text mining is the application of concepts and techniques in data mining to find patterns in text, namely a process of text analysis to extract useful information for a particular purpose. Based on the irregularity of the text data structure, the text mining process requires several stages which essentially are preparing for the text to be changed to be more structured.[8]
Documents examined in this system are documents with the extension .doc, .pdf and .txt. Users can enter original documents and comparison documents that will be calculated the level of similarity in the text content of the document. The output of this system is the percentage of the level of similarity in the text content of the document. Text mining must be able to fill, extract and use this information, both in the form of keywords and semantics.[9]
Text Mining is the process of making text and then finding or analyzing the patterns in it. The goal is what other people call. Similarity in text documents uses text mining which will ultimately lead to the detection of plagiarism. Basically words and phrases that are part of unstructured data become numerical numbers by connecting data structures and then comparing them for further action. [10]

d. Levenshtein distance

Levenshtein distance is an algorithm designed in 1965 by Russian scientist Vladimir Levenshtein. The *Levenshtein distance* between two strings is given by the minimum number of operations needed to transform one string into the other, where an operation is an insertion, deletion, or substitution of a single character. [9]
Levenshtein distance is a string matrix (sentence) used to bathe the difference or distance between two sentences. The value of the distance between two sentences is obtained from the minimum number in the modified operation that is needed in the transformation of a sentence into another sentence.[11]
These operations are : [15]

1. Insertion operation,
2. Deletion operation,
3. Subtitution operations.

Levenshtein distance is an algorithm that is used in detecting the similarity between two sentenced sentences as whether a writer acts plagiarism.
Operations on Levenshtein Distance, there are three types of operations, namely: [12]

1. Insertion Operation
   This insertion operation is a meaningful character inserted into a string (sentence). For example the sentence 'disrit' becomes the sentence 'discrete', the insertion of the character 'k' at the end of the string is carried out. This insertion is done in the middle of the string, but can be inserted anywhere at the beginning or end of the sentence. Illustration:

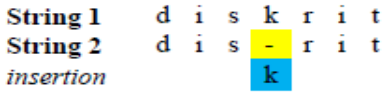


Fig.1: Insertion Operation

2. Deletion Operation
The deletion operation in a character is done to eliminate characters from a string (sentence). For example, the 'mathematical' sentence of the last character is removed so that it becomes the phrase 'math'. In the 'n' character deletion operation. Illustration:

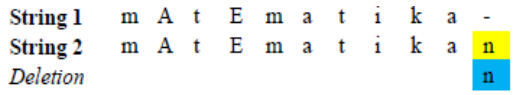


Fig.2: Deletion Operation

3. Subtitution Operation
Operation Changing to a character is the process of exchanging a character from another character's operation. For example the author writes the phrase 'set' to 'set'. In this case the 'g' character found at the beginning of the string is replaced with the letter 'h'. Illustration:

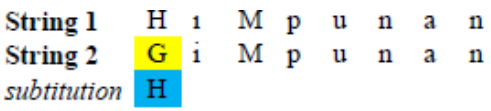


Fig.3 Subtitution Operation

III. METHOD

The steps of the Levenshtein distance algorithm in getting a distance value:
String (sentence) Suppose S = Initial sentence, and T = Target sentence [13]
Step 1: Enter
a) Calculate the length of the initial sentence as S and the target sentence as T, for example m and n
b) Then make a matrix measuring 0 ... m rows and 0 ... n columns
c) Enter the first line with 0 ... n
d) Enter the first column with 0 ... m

Step 2: Process
a) Check S [i] for 1 <i <n
b) Check T [j] for 1 <j <m
c) If S [i] = T [j], then the entry is the value that is located exactly at the top of the left, ie d [i, j] = d [i-1, j-1]
d) If S [i] ≠ T [j], then the entry is d [i, j] minimum from:
1. The value found above it, plus one, is d [i, j-1] +1
2. The value contained right in the left, plus one, is d [i-1, j] +1
3. Located right at the diagonal above the left, plus one, is d [i-1, j-1] +1

Step 3: The results of entering the matrix in line i and column j, that is
a) d [i, j] Step 2 is then repeated so that the input d [m, n] is found.

The *Levenshtein Distance* algorithm can be implemented in the programming language with the help of the *pseudocode* [16]:

```
int LevensgteinDistance (char s[1…m], char t[1…n])
    //d is a table with m+1 rows and n+1 columns
    declare int d[0…m, 0…n]
        for I from 0 to m
            D[i,0]:=i
        for j from 0 to n
            D[0,j]:=j
        for i from 1 to m
            For j from 1 to n
                If s[i]=t[j] then d[i,j]:=d[i-1,j-1]
                    Else
                D[I,j]:=minimum(
                    d[i-1,j]+1, //deletion
                    d[i,j-1]+1, //insertion
                    d[i-1-j-1]+1, //substitution
        return d[m,n]
```

Weight or avalue *Similarity Levenshtein Distance* determines the calculation of its *similarity* after it can determine the value *distance* of the two documents that are compared. Then do an equation in determining the weight or avalue *Similarity*, namely:[12]

$$Plagiarized \quad Value = \{1 - \frac{diff}{max(cs,st)}\} * 100 \quad (1)$$

Description:
CS = *Source String*
ST = Target *String*
*Similarity = Similarity value / Plagiarized Val*ue
Diff = Levenshtein Distance
Max(CS, ST) = The longest *string* value

With d [m, n] the *distance* value, which lies in a row to m and a column to n, S becomes the length of the initial *strin*g, T becomes the length of the target *string*, and Max (S, T) is the largest long string between the initial *strings* and target *strings*.[14]
The assumed weight or *Similarity* value is in the range of 0 (zero) to 100 (suppose) percent, which means the value of 100 is the maximum value that shows the two words. This design is enabled so that it can be used to measure the weight or *Similarity* values between two *strings* (sentences) based on the character arrangement.

Table.1: Levenshtein Distance Sample 1

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

	i (x)	B	P	K	M	A	K	A	N	N	A	S	I	T	D
j(x)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
B	1	0	1	2	3	4	5	6	7	8	9	10	11	12	13
P	2	1	0	1	2	3	4	5	6	7	8	9	10	11	12
K	3	2	1	0	1	2	3	4	5	6	7	8	9	10	11
M	4	3	2	1	0	1	2	3	4	5	6	7	8	9	10
A	5	4	3	2	1	0	1	2	3	4	5	6	7	8	9
K	6	5	4	3	2	1	0	1	2	3	4	5	6	7	8
A	7	6	5	4	3	2	1	0	1	2	3	4	5	6	7
N	8	7	6	5	4	3	2	1	0	1	2	3	4	5	6
B	9	8	7	6	5	4	3	2	1	1	2	3	4	5	6
A	10	9	8	7	6	5	4	3	2	3	1	2	3	4	5
S	11	10	9	8	7	6	5	4	3	4	5	1	2	3	4
O	12	11	10	9	8	7	6	5	4	5	6	7	2	1	2
T	13	12	11	10	9	8	7	6	5	6	7	8	9	2	1
D	14	13	12	11	10	9	8	7	6	7	8	9	10	11	2

It's Known:
i (x) = BPKMAKANNASITD and
j (y) = BPKMAKANBASOTD

CS = *Source String* = i (x) ,
ST = Target *String* = j (y)
Similarity = *Similarity value* / *Plagiarized Val*ue
Diff = *Levenshtein Distance* = 2 (Difference)
Max (CS, ST) = The longest *string* value = 14
*Plagiarized Val*ue = $\{1 - \frac{2}{max\,(14)}\} * 100 = 0{,}86$ / 85,71%.

Table 2: Levenshtein Distance Sample 2

	i(x)	B	A	R	U
j(y)	0	1	2	3	4
B	1	0	1	2	3
A	2	1	0	1	2
T	3	2	1	1	2
U	4	3	2	2	1

It's Known:
i (x) = BARU and j (y) = BATU

CS = *Source String* = i (x),
ST = *Target String* = j (y)
Similarity = *Similarity value* / *Plagiarized Val*ue
Diff = *Levenshtein Distance* = 1 (Difference)
Max(CS, ST) = The longest *string* value = 4

*Plagiarized Val*ue = $\{1 - \frac{1}{max\,(4)}\} * 100 = 0{,}75$ / 75%.

IV. RESULTS

Detecting the similarity of the text content of this document is a system that will detect and give a percentage of the similarity between the first or original documents and the comparison documents tested. Figure 4 is a flowchart that will explain the flow of the program.

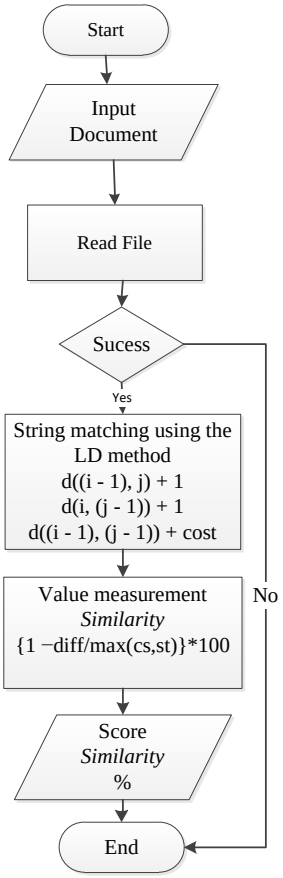


Fig.4: Flowchart Flow of Program

To begin using detection in common text content of the document, the user must know the process flow methods of the program as follows:

The first time the user inputs a file, the inputted file can be a Pdf, Doc and txt file, after that the file will be read first, the file that is read will be checked for status, if it is appropriate then the calculation will be done using the Levenstein Distance method for generate the similarity value.

At this stage the document entered will be read by character characters. The document entered is

Documents from the First sentence or original
Informatics Engineering is the best department

Documents from Comparative sentences
Informatics Engineering is the best major this year

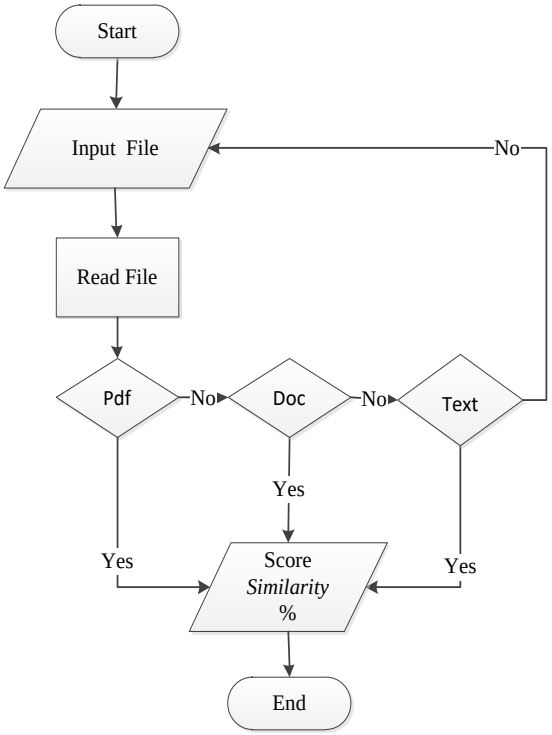


Fig.6: Flowchart Reading File

To begin using detection in common text content of the document, the user must know the flow of the process of reading the file as follows:

The first time the user inputs a file, the inputted file can be in the form of Pdf, Doc and txt files, then checking the text format that will be checked for similarity, after that the system will provide similarity score information

Table 3. Scenario Test Results

No	Scenario	Long sentence Character section A	Long sentence Character section B	Difference Number of Characters	Presentase
1	A1: Many on the street out of the seller's factory hood B1: Many found the seller's factory hood	51	36	18	64.71%
2	A2: Eat with family B2: Eat with family in the morning	15	30	15	50%
3	A3: Gasoline drives up the vehicle B3: Gasoline has asked for help	30	27	16	46.67%
4	A4: Are you done B4: I asked you if you're done	12	26	16	38.46%
5	A5: The best information technology recently B5: The best information technology just yesterday	40	46	11	76.09%
6	A6: The storm out there is very big B6: The storm will come out there is very large	31	43	14	67.44%
7	A7: Fish eat grass B7: Fish eat grass pellets	14	22	8	63.64%
8	A8: Mother cooked rice this morning B8: Mother cooked chiken this morning	31	33	4	87.88%
9	A9: Dad fishing fish in the lake B9: Dad fishing at sea this afternoon	28	33	16	51.52%
10	A10: Sister reads picture books B10: Brother read magazine books	26	27	12	55.56%

Information Scenario Test Result Percentage Check, done to find out the Levenshtein Distance formula and Similarity can produce a percentage result with a formula that is given information that is easily understood by the author to explain that the difference in the number of characters is used for the calculation of the Levenshtein Distance formula. Then Similarity to calculate the results.

Here's how to test it how to calculate it:
With the formula (1) :

The tester will take two examples from table 3 Scenario Test Results and then calculate them:
Is known : CS = A5 and B5 = ST

Table.4: Scenario Test Results point 5

5	A5: The best information technology recently B5: The best information technology just yesterday	40	46	11	76.09%

A5: The best information technology recently
B5: The best information technology just yesterday

A5 = 40 character lengths
B5 = 46 character lengths
Diff = 11 different characters

Plagiarized Value = $\{1 - \frac{11}{max(46)}\} * 100 = 76.09\%$ kesamaannya.

In known : CS= A9 dan B9= ST
The scenario taken and the percentage will be calculated:

Table.5: Scenario Test Results point 9

9	A9: Dad fishing fish in the lake B9: Dad fishing at sea this afternoon	28	33	16	51.52%

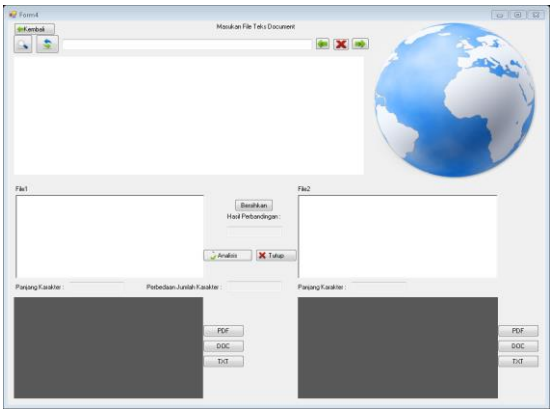
A9: Dad fishing fish in the lake
B9: Dad fishing at sea this afternoon

A5 = 28 character lengths
B5 = 33 character lengths
Diff = 16 different characters

Plagiarized Value = $\{1 - \frac{16}{max(33)}\} * 100 = 51.52\%$ kesamaannya.

Implementation Syste

Fig.7: Implementation System



The user can enter files in the form of Pdf, Doc, Txt, Pdf files that must be copied into an empty column if they want to check, what percentage of the difference is compared from the similarity of the text content of the document you want to find out whether the plagiarism or not is in the text .

V. DISCUSSION

Users should know and understand plagiarism well by searching for information in various ways and trying to avoid plagiarism and also getting used to making scientific papers properly and correctly from the beginning of

college.Motivate the academic community to be able to emphasize honesty in every activity, specifically academic activities related to campus.

VI. CONCLUSIONS

This application helps improve understanding and awareness which has made it easier for users to obtain information about plagiarism detection so as not to take action, a solution for users to be able to check the percentage of plagiarism, contained in the text content of documents owned by users or even those on an internet web browser, who want to check the results of the difference in percentage by the user.

REFERENCES

1. Irianto, W.A., 2014, Penentuan Tingkat Plagiarisme Dokumen Penelitian Menggunakan Centroid Linkage Hierarchical Method (CLHM), Jurnal. Program Teknologi Informasi dan Ilmu Komputer. Universitas Brawijaya. Malang.
2. H. A. Na'Firul, "Aplikasi Pendeteksi Kemiripan Isi Teks Dokumen Menggunakan Metode Levenshtein Distance," in semanTIK, 2016.
3. Sastroasmoro, S., 200, Beberapa Catatan Tentang Plagiarisme, Majalah Kedokteran Indonesia, Volume : 57, Nomor : 8, Agustus 2007
4. Juan, M. Tores, Moreno, G. Sierra, P. Peinl, A German Corpus for Similarity Detection Tasks, International Journal of Computational Linguistics and Applications, vol. 5, no. 2, 2014, pp. 9–24
5. Shik Kang Seung, Word Similarity Calculation by Using the Edit Distance Metrics with Consonant Normalization. J Inf Process Syst, Vol.11, No.4, pp.573~582, December 2015
6. J. L. Yudhy, S. Alicia and J. Agustinus, "Rancang Bangun Aplikasi Deteksi Kemiripan Dokumen Teks Menggunakan Algoritma Ratcliff atau Obershelp," E-Journal Teknik Informatika, 2017
7. Nugrohoadhi Agung, 2015, Pengorganisasian Dokumen Dalam kegiatan Kepustakawanan, Jurnal Ilmu Perpustakaan , Informasi dan Kearsipan Khizanah Al-Hikah, Vol 3, No. 1, pp1-10
8. Junedy, Richard. 2014. Perancangan Aplikasi Deteksi Kemiripan Isi Dokumen Teks dengan Menggunakan Metode Leveshtein Distance. Jurnal Pelita Informatika Budi Darma Vol. VII No.2, Jurusan Teknik Informatika, STMIK Budi Darma, Medan.
9. Kumar, L. & Bhatia, P.K., 2013. Text Mining: Concepts, Process and Applications. Journal of Global Research in Computer Science, Vol. 4, No. 3, pp. 36-39.
10. Saini Anu, Bahl Ankita, Kumari Supriya, Singh Mitali, Plagiarism Checker: Text Mining, International Journal of Computer Applications (0975 – 8887), Volume 134 – No.3, January 2016
11. P. P. B. and P. A. S., "Analisis Kinerja Algoritma Levenshtein Distance Dalam Mendeteksi Kemiripan Dokumen Teks," Jurnal Logika, p. 131–133, 2016.
12. Harlian, M (2018) Text Mining, Htttp://iwanarif.lecturer.pens.ac.id/kuliah/dm/6Text%20/Mining.pdf, Diakses pada 2 Mei 2018.
13. Haldar, Rishin. dan Mukhopadhyay, D. 2011. Levenshtein Distance Technique in Dictionary Lookup Methods: An Improved Approach. Diakses tanggal 2 Mei 2018, dari Cornell University Library (http://arxiv.org/abs/1101.1232).
14. Winarsono, D., D.O. Siahaan dan U. Yuhana. 2009. Sistem Penilaian Otomatis Kemiripan Kalimat Menggunakan Syntactic Semantic Similarity pada Sistem ELearning. Jurnal Ilmiah KURSOR Menuju Solusi Teknologi Informasi Volume 5 Nomor 2, Jurusan Teknik Informatika, ITS, Surabaya.
15. Andriyani, N.M., 2010 , Implementasi Algoritma Levenshtein Distance dan Metode Empiris untuk menampilkan saran perbaikan kesalahan pengetikan dokumen berbahasa Indonesia, Skripsi, Teknik Informatika, Universitas Udayana, Bali.
16. Sulistyo-Basuki, Teknik dan Jasa Dokumentasi. Jakarta: Gramedia, Pustaka Utama. 1992.
17. Kabir Mashud, Similarity Matching Techniques For Fault Diagnosis in Automotive Infotainment Electronics, IJCSI International Journal of Computer Science Issues, Vol. 3, 2009.
18. Lydia, E. L., Kumar, P.K., Shankar, K., Lakshmanaprabu, S.K., Vidhyavathi, R.M., Maseleno, A., Charismatic Document Clustering through Novel K-Means Non-negative Matrix Factorization (KNMF) Algorithm using Key Phrase Extraction, International Journal of Parallel Programming, Springer, 2018, pp. 1-19.