Fuzzy Knuth Moris Pratt Algorithm for Knowledge Management System Model on Knowledge Heavy Metal Content in Oil Plants

Ermatita[1*]
[1]Department of Information Systems
Faculty of Computer Science
Universitas Sriwijaya
Inderalaya, Indonesia
[1]ermatitaz@yahoo.com

Dedik Budianta[2]
[2]Department of Soil Science
Faculty of Agriculture
Universitas Sriwijaya
Inderalaya, Indonesia
[2]dedik_budianto@yahoo.com

Abstract--Knowledge Management system is a system that can help to accommodate and store knowledge. Storage of useful knowledge for the business processes necessary to sustain existing knowledge. Knowledge management system can accommodate knowledges, so that if needed at any time can be reused. Storage of knowledge in this system, to get it back to do a search string to facilitate the search for knowledge needed. String search method in this study with Fuzzy Algorithm Knuth Morris pratt. In the study, Fuzzy Knuth Morris Pratt algorithm used for modeling knowledge management system in the heavy metal content of oil crops. Application of fuzzy search string vague where the string matched only have similar characters determined by the method Knuth Morriss Pratt. The analysis of the heavy metal content can be performed to determine the content of heavy metal in the oil palm. By knowing the content of heavy metals in soil, plants and oil palm fresh fruit bunches then mitigation solutions can be determined accurately. The results of this research is the search string matching method with Fuzzy Algorithm Knuth Morriss Pratt on the model of Knowledge Management System that stores a variety of knowledge and experience about the metal content and solutions to overcome them. Novelty in this research is the application of Knuth Morris pratt algorithm combined with fuzzy knowledge on modeling heavy metal content in the palm. So the search for the Knowledge Management System can be done more quickly and accurately.
Keywords--Knuth Morris Pratt; Knowledge Management; KMS.

I. INTRODUCTION

Palm oil production is highly dependent on various factors, including the type of soil, seed varieties, climate and the technology applied. Increased production of oil palm trees need to identify the various barriers that can lead to low productivity of oil palm. The presence of heavy metals in soil, plants and fresh fruit bunches of oil palm can hamper productivity.

Production of palm oil will be disrupted by the heavy metal. Thus in this study the quest for knowledge about the content of heavy metals in the oil palm trees and the solution. Knowledge from the analysis of heavy metal content in palm oil, will be stored in a knowledge management system. To access the knowledge in the knowledge management system is needed a model that can help speed up the search for knowledge in a knowledge management system. Therefore in this study will be conducted modeling using Fuzzy Algorithm Knuth Morris Pratt (KMP). This algorithm to model the search for knowledge in the knowledge management system with string matching. Research in performing string matching has been done by several researchers, among others [1] has conducted research string matching to search text in Arabic. His research has been reducing the time to search in parallel systems. Another study conducted by [2] has conducted research on composite method in string matching, research has lead to the conclusion that the algorithm Knuth Morris Pratt takes faster than the search string matching method to another. [3] in research have KMP as part of the concept of parallelization improve the performance of the algorithm. Research by [4] combining algorithm Knuth-Morris-Pratt and the Boyer Moor. The results of this study can be applied to the search string for the better. From the results of previous research in this study will model the Knowledge Management System to string matching in search of knowledge about the content of heavy metals in oil palm plantations by Fuzzy Knuth-Morris-Pratt Model.

II. LITERATURE REVIEW

2.1 Knowledge Management

Knowledge Management is a discipline with an integrated approach to the identification, management and distribution of all information assets of an organization. Knowledge Management function works to increase the organization's ability to learn from their environment and incorporate knowledge within an organization to create, collect, preserve and disseminate knowledge of the organization [5]. It is also defined as an effort towards the

increasement of value of knowledge along the organization includes, the way in encouraging communicating, giving the chance to learn, and promoting the sharing of better knowledge [6]. Knowledge Management has important functions which are divided into four (4) as follows [7]:

a) Identify the key asset of the knowledge in the company.
b) Reflecting on what the organization knows.
c) Sharing all knowledge to anyone who needs it.
d) Applying the use of knowledge to improve performance
e) organization.
f) Triggered Knowledge Management (The 24 Drivers of KM) makes KM into things that can not be ignored in a business is divided into 6 groups of Error [8].

2.2 Method of Search String Matching

According to Black [9] string is the array of characters (numbers, alphabet, or other characters) is usually presented as a data structure array. String can be a word, phrase, or sentence. While the string matching according to Black [9] is defined as a problem to find the pattern arrangement of the character string in another string or part of the contents of the text.

III. RESEARCH METHODOLOGY

3.1Analisis knowledge

This research will be carried out the mixing of the experience, values, contextual information and vision experts who provide evaluation framework related to new experiences and information.

Modeling knowledge management system is done by:

a) Kowledge discovery system: the search for knowledge of data and information
b) Knowledge capture system: the process of gaining knowledge from explicit or tacit knowledge
c) Knowledge sharing system: the process of distributing knowledge [10].
d) Modeling Knowledge Management System with String Matching and Fuzzy Algorithm Knuth-Morris-pratt (KMP)

IV. MODELING KNOWLEDGE MANAGEMENT SYSTEM WITH FUZZY KNUTH-MORRIS- PRATT (KMP) ALGORITHM

4.1 Search String Matching

String matching is one string search method that uses a process approach to pattern of the search string. String matching method is inexact matching. The concept of this method of searching for the same string and string approached by another string. A key concept of this search is how to decide that a search string has similarities with a string accommodated in the dictionary, although not exactly in the order of his character. To decide 'sameness' is used a function termed the similarity function. Various methods have been developed to determine the similarity function. This function will be in charge of deciding string search results if it is found the results of approach (approximation). Inexact string matching or also called Fuzzy string matching, a matching string cryptically, that matching string where the string that matched similarities both with the arrangement of a different character (possibly the number or sequence) but the string has a resemblance both similarities textual / writing (approximate string matching) or similarity utterance (phonetic string matching). Inexact string matching is still divided into two, namely:

a. Matching string based on the similarity of writing (approximate string matching) is a matching string with a basic similarity in terms of writing (number of characters, the arrangement of the characters in the document). The degree of similarity is determined by whether or not far different from writing two strings are compared or the value of the degree of similarity is determined by the programmer (programmer).

   Example: a computer with a compiler, have the same number of characters, but there are two different characters. If the difference in these two characters can be tolerated as a typing error in the two strings is said to be suitable.

b. String matching based on similarity of pronunciation (phonetic string matching) is a matching string with a basic similarity in terms of pronunciation even though there are differences in the writing of the two strings are compared.
   Example: step by steb of posts in the pronunciation is different but so similar that the two strings are considered suitable. Another example is a step, with steppe, sttep, stepp, stepe.

Exact string matching helpful if the user wants to search for a string in a document that exactly matches the input string. But if the user wants the search string is approached with the input string or an error occurred writing string input and document search object, then the inexact string matching beneficial. Some exact string matching algorithms, among others: the algorithm Knuth-Morris Pratt, Bayer-Moore, and others - others.

4.2 Knuth-Morris-Pratt algorithm (KMP)

KMP algorithms perform initial process (preprocessing) of the pattern P by calculating the outskirts function. In some literature called overlapping functions, function failure, prefix functions, and so on. This function

indicates the largest possible shift P by using comparisons that formed before the search string. Thus, we can get through a shift or a string comparison that is not useful, as the brute force algorithm.

Calculating the periphery function is the first step and the most important steps of the algorithm KMP, because the results of the calculation function of periphery will indicate a shift to do pattern. With the function of this suburb unnecessary shift can be prevented. This function depends only on the outskirts of the characters in the pattern, therefore we can do calculations before the string search process begins. KMP method of calculating the outskirts function b (j) is defined as the size of the longest prefix string of P that is a suffix of P [1..j].

For more details, the following is given an example to calculate the function outskirts of a pattern P = xlnxls. For the record, use a value of 0 (zero) as the initial index string on this issue.

The prefix of P is

□, x, xl, xln, xlnx, xlnxl

Suffix of P is

□, s, ls, xls, nxls, lnxls

Description: □ = empty string

Value function outskirts b (j) for each character in P are shown in Table 1.

TABLE. 1. VALUE FUNCTION

J	0	1	2	3	4	5
P*(j)*	X	1	n	x	1	s
b*(j)*	0	0	0	1	2	0

Before calculating the outskirts of the function, the authors define a first array placeholder periphery functions.Algorithm calculate the outskirts function itself is as follows:

```
procedure calculate(input m : integer, P : array[1..m] of char,
output b : array[1..m] of integer)
{calculate  b[1..m] untuk pattern P[1..m]}
Declaration :
k, q : integer
Algorithm:
b[1] ← 0
q ← 2
k ← 0
for q ← 2 to m do
  while((k > 0) and (P[q] ≠ P[k+1]))do
     k ← b[k]
  endwhile
            if P[q] = P[k+1] then
            k ← k+1
            endif
            b[q] ← k
     endfor
```

After completing the counting process outskirts function, then the next stage is the process of matching patterns or keywords with less using Fuzzy algorithm KMP pattern matching phase.Systematically, the steps - the steps taken by the algorithm Knuth-Morris-Pratt during the match string is:

1. Knuth-Morris-Pratt algorithm begin to match the pattern at the beginning of the text.
2. From left to right, these algorithms will match the character-by-character pattern with the character in the corresponding text, until one of the following conditions are met:
    a. Characters in the pattern and in the text being compared do not match (mismatch).
    b. All the characters in the pattern match. Then the algorithm will inform the invention in this position.
3. The algorithm then shift pattern based on the table, and then repeat steps 2 through pattern at the end of the text.The next stage was to match the pattern P Text T= xlnxlnxls.

```
i : 0 1 2 3 4 5 6 7 8
T : x l n x l n x l s

j : 0 1 2 3 4 5
P : x l n x l s
```

The first step is compare left end pattern P with the left end of the text T. The characters at position 0 to 4 equal (match), but in the position i = j = 5 mismatch occurs, the text n T s on pattern P. As there is a mismatch, we do shift pattern P with the number of shifts in accordance with the value of the outskirts of the prefix pattern P match. In this case, the prefix match is xlnxl with length l = 5. Values for the outskirts of the longest string P [0..4] is b (4) = 2. The amount of shift is l - b = 5-2 = 3. So , pattern P is shifted to the right as far as 3 characters and subsequent comparison carried out starting at the position j = l - b - 1 = 5-2 - 1 = 2.

```
i : 0 1 2 3 4 5 6 7 8
T : x l n x l n x l s

j : 0 1 2 3 4 5
P : x l n x l s
```

The complete KMP algorithm is as follows:

```
procedure KMPSearch(input m: integer, n: integer, input P:
array[1..m] of char, input T: array[1..n] of char, output idx : integer)

{Search for fit pattern p in text T with Knuth-Morris-Pratt Algorithm.
If p is found, the location will be kept and change of idx

Declaration

i, j : integer

ketemu : Boolean
b : array[1..m] of integer
procedure HitungPinggiran(input m: integer, P: array[1..m] of char,
output b : array[1..m] of integer)
{counting b value[1..m] to pattern P[1..m]}
Algoritma:
HitungPinggiran(m,P, b)
i ← 1
j ← 0
ketemu ← false
while (i≤ n and not ketemu) do
   while ((j > 0) and (P[j+1] ≠ T[i])) do
            j ← b[j]
endwhile
if P[j+1] = T[i] then
            j ← j+1
endif
if j = m then
            ketemu ← true
     else
            i ← i+1
     endif
     endwhile
     if ketemu then
        idx ← i-m+1
     {note : if index of array starts from 0, idx←i-m}
     else
       idx ← -1
endif
```

For example, given the following example:

Text : Leave - soil

Pattern : age

The first step is to calculate the fringe function of the keyword (pattern), as shown in Table 2

TABLE 2. SUBURB FUNCTIONS 1

j	1	2	3	4
P(j)	A	g	e	
B(j)	0	0	1	0

Description:

j : index from P(j)
P(j) : Longest prefix size pattern
B(j) : Suburb funtion
Text : Characters entered
Pattern : Key word
m : index from text.

The process of calculating the fringe functions :

```
B[1] = 0
B[2]   q<= m
       B[3]    q<=m
        2 <= 4 (ya)
              3 <= 4 (ya)
          k > 0
       k > 0                                  0  >  0
       (No)                                     0
       > 0 (No)
          P[k+1]              =              P[q]
       P[k+1] = P[q]
          P[0+1]              =              P[2]
       P[0+1] = P[3]
            P[1]              =              P[2]
       P[1]    = P[3]
             U = M (No)
       U = U (yes)
            B[q] = k
                  k = k+1
            B[2] = 0
                  k = 0+1
              q = q+1
            k = 1
              q = 2+1
            B[q] = k
              q = 3
                  B[3] = 1
                        q = q + 1
                     q = 3 + 1
                      q = 4

B[4]     q<= m
       B[5]    q<=m
         4 <= 4 (yes)
              5 <= 4 (No)                  k > 0
                                         Finish


         1 > 0 (yes)

          P[k+1] = P[q]
          P[1+1] = P[4]
            P[2] = P[4]
                M = R (No)
            k = B[k]

           k = B[1]
           k = 0
              B[q] = k
              B[4] =   0
        q = q+1
    q = 4+1                                  q = 5
```

The second step is the process of matching patterns (keywords) as shown in the Table. 3:

TABLE 3. STEP 1 PATTERN MATCHING PROCESS

m	1	2	3	4	5	6	7	8	9	10
Text	l	e	a	v	e		s	o	i	l
Pattern	a	g	e							
j	1	2	3	4						

Results matching the above there are no characters that match the pattern with the text, the shift is done as far as one character to the right.

TABLE 4. STEP 2 PATTERN MATCHING PROCESS

m	1	2	3	4	5	6	7	8	9	10
Text	l	e	a	v	e		S	o	i	l
Pattern		a	g	e						
j		1	2	3	4					

There are no characters that match the pattern with the text, then do another shift to the right as far as one character.

TABLE 5. STEP 3 PATTERN MATCHING PROCESS

m	1	2	3	4	5	6	7	8	9	10
Text	l	e	a	v	e		s	o	i	l
Pattern			a	g	e					
j			1	2	3	4				

In Table. 5 over the last character that matches the pattern with the text, the amount of the shift pattern is determined by the periphery of the corresponding P prefix. In the example above, the corresponding prefix is U, with a length L = 1. The longest Fringe A B (j) it is 0, then the distance shift is L - B (j) = 1-0 = 1. So, pattern P is shifted as far as 1 character.

TABLE 6. STEP 2. PATTERN MATCHING PROCESS

m	1	2	3	4	5	6	7	8	9	10
Text	l	e	a	v	E		S	o	i	l
Pattern				a	G	e				
j				1	2	3	4			

There is no matching character between pattern and text, then the shift is done as far as one character to the right. This is continued until pattern matching is complete.

TABLE7. STEP 2 PATTERN MATCHING PROCESS

m	1	2	3	4	5	6	7	8	9	10
Text	l	e	a	v	e		s	o	i	l
Pattern					a	g	e			
j					1	2	3	4		

Note : There is no matching character between pattern and text, then the shift is done as far as one character to the right.

TABLE 8. STEP 2 PATTERN MATCHING PROCESS

m	1	2	3	4	5	6	7	8	9	10
Text	l	e	a	v	e	E	o	i	l	
Pattern						A	g	e		
j						1	2	3	4	

Note : There is no matching character between pattern and text, then the shift is done as far as one character to the right.

TABLE 9. STEP 2 PATTERN MATCHING PROCESS

m	1	2	3	4	5	6	7	8	9	10
Text	l	e	a	v	e	s	o	i	l	
Pattern							a	g	e	
j							1	2	3	4

Note : From the 8th process it is known that the letter U pattern of the word is sought not the same as the characters in the text, and a pattern matching process was stopped because the remaining length of the text characters are not the same pattern again. Then the system will provide a response to the search that has been done by responding that "the word you are looking for was not found".

This process is known that pattern letter D on the keywords that are searched have found a counterpart to the characters in the text, then the match continues to letters AUN to the characters in the text, to the end and finally letters LEAF has been matched or find a counterpart in sequence without missmatch any characters with the characters in the text. This match will also be made to the text which has similarities. If the similarity of character telahditemukan, matching process is stopped and the algorithm will inform the invention in this position, and the system will respond based on keywords. If there is a mismatch, but there are words that are similar, then the text can be used as keywords. Furthermore, it can be sought in knowledge in KMS explanation and knowledge related to the keyword.

V. CONCLUSION

Fuzzy Knuth Morris Pratt algorithm is an algorithm that combines fuzzy and Knuth-Morris-Pratt algorithm. With this algorithm matching string matching can be done and taken which has some similarities. So knowledges to determine the solution to the problems associated with heavy metals that have no knowledge can be found quickly. So that the problems associated with heavy metals in oil crops can tackle as soon as possible. The results of this modeling will be the basis for the development of a knowledge management system

REFERENCES

[1] I. M. Abu-Zaid and E. K. El-Rayyes, “Parallel search using KMP algorithm in arabic string,” *Int. J. Sci. Technol.*, vol. 2, no. 7, 2012.

[2] N. R. Dalal and P. Jadhav, “A Composite Algorithm for String Matching,” pp. 68–74, 2015.

[3] A. Rasool and N. Khare, “Parallelization of KMP String Matching Algorithm on Different SIMD

architectures : Multi- Core and GPGPU ' s," vol. 49, no. 11, pp. 26–28, 2012.

[4] R. Y. Tsarev, A. S. Chernigovskiy, E. A. Tsareva, V. V Brezitskaya, A. Y. Nikiforov, and N. A. Smirnov, "Combined string searching algorithm based on knuth-morris-pratt and boyer-moore algorithms," in *IOP Conference Series: Materials Science and Engineering*, 2016, vol. 122, no. 1, p. 12034.

[5] C. Koina, "Librarians are the ultimate knowledge managers?," *Aust. Libr. J.*, vol. 52, no. 3, pp. 269–272, 2003.

[6] C. McInerney, "Knowledge management and the dynamic nature of knowledge," *J. Assoc. Inf. Sci. Technol.*, vol. 53, no. 12, pp. 1009–1018, 2002.

[7] C. Davidson and P. Voss, *Knowledge management: An introduction to creating competitive advantage from intellectual capital*. Tandem, 2002.

[8] A. Tiwana, *The knowledge management toolkit: practical techniques for building a knowledge management system*. Prentice Hall PTR, 2000.

[9] M. Syaroni and R. Munir, "Pencocokan String Berdasarkan Kemiripan Ucapan (Phonetic String Matching) dalam Bahasa Inggris," vol. 2, pp. 1–6.

[10] I. Becerra-Fernandez and J. SABHERWAL, "Knowledge Management: Systems and Processes, New York: ME Sharpe." Inc, 2010.