Knuth Morris Pratt Algorithm in Enrekang-Indonesian Language Translator

1st Desi Anggreani
Department of Electrical Engineering
State University of Malang
Malang, Indonesia
desiianggreanii@gmail.com

2nd Desy Pratiwi Ika Putri
Department of Electrical Engineering
State University of Malang
Malang, Indonesia
desypratiwi407@gmail.com

3rd Anik Nur Handayani
Department of Electrical Engineering
State University of Malang
Malang, Indonesia
aniknur.ft@um.ac.id

4th Huzain Azis
Faculty of Computer Science
Universitas Muslim Indonesia
Makassar, Indonesia
huzain.azis@umi.ac.id

***Abstract*—String match search as in the case of making a translator matching accuracy is essential. Therefore, there needs to be an implementation of a string matching algorithm that will help get accurate and optimal results. Knuth Morris Pratt is one algorithm to solve problems in the case of string matching. The workings of the KMP algorithm match character by character between pattern and text in the repository. The results of this study indicate that the accuracy of string matching with input in the form of letters and characters by mixing punctuation is 100%. By using 1063 vocabularies, the implementation of the Knuth Morris Pratt algorithm in the Indonesian enrekang language translator has an average processing time of 0.01901 milliseconds and average memory usage of 15,768 MB so that the KMP algorithm can be implemented optimally in the enrekang-Indonesian language translator.**

Keywords— KMP, translator, String Matching, Enrekang

I. INTRODUCTION (*HEADING 1*)

Enrekang language is a language of local clumps that has proximity to communications in the region of South Sulawesi and West Sulawesi. Enrekang in clusters Massenrempulu ' has a closeness to Bugis, Toraja, Tae ' (Luwu), and Mandar. Enrekang language as a regional language has a distinctive characteristic in grammar and word meaning. It makes Enrekang style different from other local languages in Indonesia[1]. Enrekang languages also have unique features, such as word suppression. Therefore, to preserve and introduce Bahasa Enrekang, it takes technology to understand and learn the literature[2].

With the development of today's technology, many online and offline applications to translate other languages into Bahasa Indonesia or vice versa with several methods such as *string matching*[3], or zoning[4]. However, current technological developments, especially the application of local language translators, have not been felt by the people in Enrekang Regency itself. The vocabulary of the regional languages from enrekang to English too. In terms of the Enrekang language repository, it is still challenging to find.

Many studies implement string matching problems (also called string search problems), and there are too many attempts to find better algorithms for this problem regarding the reduction of the complexity of time and space[5]. One of the most well-known algorithms is Knuth Morris Pratt[6] because it's effective in string matching method. Much of its implementation has led to the translation of translations, for example, English to Indonesian language translation[7]. Other studies have also explained using the same algorithm for the Mandailing-Indonesia language translator application, suggesting that Knuth Morris Pratt's algorithms are essential for the more natural word translation process.

This study to show implement the algorithm of Knuth Morris Pratt, making a regional language translator Enrekang to Indonesia and vice versa with attention to input in the form of characters and punctuation.

II. RELATED RESEARCH

Researcher[8] merge the Knuth Morris Pratt and Boyer Moore Hybrid algorithms to model knowledge management systems on employee competencies in a company. Similarly, the researcher[9] combines the Fuzzy and Knuth Morris Pratt algorithms for the Model knowledge management system on heavy metals content in oil palm plantations. This algorithm matching, string matching, can be done and taken, which has some similarities. So knowledge to determine the solutions to the problems associated with heavy metals that do not know can be found quickly. Besides, the researcher[10] on "A review: search visualization with Knuth Morris Pratt Algorithm" allows researchers or students to know how the KMP algorithm works and can be developed and implemented into many search processes.

Researches[11], [12], and [13] performed a parallel analysis on Knuth Morris Pratt Algorithm. Displays better algorithm performance and more efficient to use. Another case with[14] uses the Knuth Morris Pratt algorithm in genomic detection, which results in faster, efficient, and significant computing. Other researchers[15] introduced a parallelization and optimization approach for the KMP algorithm on a heterogeneous architecture based on multi-core GPU. Through overlapping calculations/communications, most data transfers have errors in calculations and also optimize the allocation of work items and workgroups.

978-1-7281-8131-8/20/$31.00 ©2020 IEEE

Researcher[16] on "combined string searching algorithm based on Knuth Morris Pratt and Boyer-Moore algorithm." Display the string search algorithm by combining the advantages of the two algorithms, and the second algorithm gives rise to higher effectiveness than the first algorithm. That way, this algorithm is used for text searches in English and Russian. While the researcher[17] on the "Implementation and Analysis of Zhu-Takaoka Algorithm and Knuth Morris Pratt Algorithm for Dictionary of computer applications Based on Android." Displays a comparison between the two algorithms in the dictionary application of the computer terminal and the ones that show the algorithm Zhu-Takaoka is more optimal used in the matching string based on the time used in the search process.

III. METHOD

A. Data Mining

In the manufacture of Enrekang language translator-Indonesia and vice versa, the essential thing is the vocabulary of the language Enrekang itself. In this study, there was 1065 vocabulary gathered from several sources[18]. In the glossary, it was starting from the prefix letter A until the prefix letter Z. From table 1 will be seen in some samples of vocabulary that exist in this study.

TABLE I. ENREKANG REGIONAL VOCABULARY DATA SET SAMPLE

Word	Translate
Aje	Kaki
Ala	Ambil
Allo	Matahari
Ballo'	Bagus
Ba'te	Sangria
Ba'ta	Jalan
Canik	Manis
Carikki'	Sangat Dingin
Dikka	Kasihan
Doko	Kurus
Edda'	Tidak
Gaja	Sangat
Indo'	Ibu
Jabe	Manja
Kabiri	Dimarah
Lako'	Mungkin
Madosa	Berdosa
Nasu	Masak
Olo	Depan
Pai'	Pahit
Rampan	Jatuh
Sai	Lama
Tallo	Telur
Undi	Ikut
…	…
…	…
Wai	Air
Yara	Jika

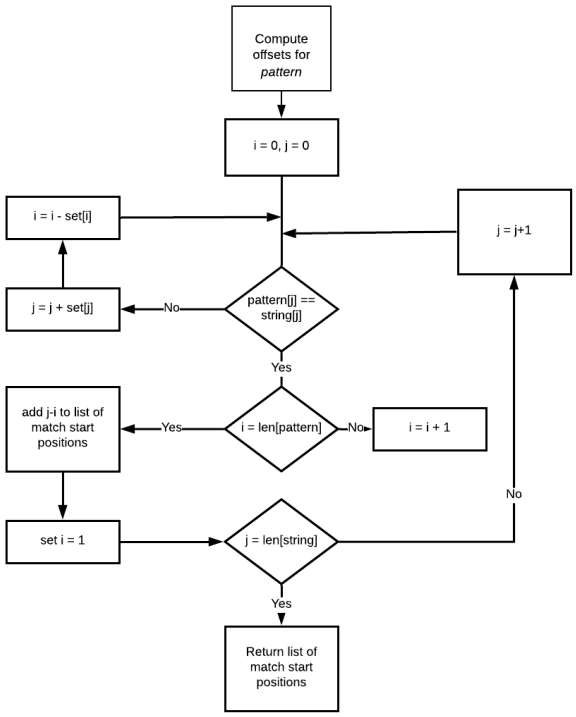

Fig. 1. *Flowchart Knuth Morris Pratt*

B. Knuth Morris Pratt Algorithm

Algoritma *Knuth-Morris-Pratt* has a per-shift calculation follows, when a mismatch occurs when the *pattern* and text are aligned $[i..i+n-1][i..i+n-1]$, be considered the first mismatch occurs between the text $[i+j][i+j]$ and *model* $[j][j]$, With $0 < j < n0 < j < n$. Meaning, the text $[i..i+j-1] = pattern[0..j-1]$ $[i..i+j-1] = pattern[0..j-1]$ and $a = teks\ [i+j]$ $a = teks\ [i+j]$ Not equal to $b = pattern[j]$ $b = pattern[j]$ [19]. When shifting, it is very reasonable if a prefix of the vv *pattern* will be the same as some of the ends of some text. U *pattern* can be shifted to align the vv prefix with the suffix. The search phase can be done in $O(m+n)$ $O(m+n)$. *Knuth Morris Pratt* 's algorithm performs most text character comparisons $2n-12n-1$ During the search phase. Delays (the maximum number of comparisons for a single text character) are limited by $\log_{\Phi}(m)\log_{\Phi}(m)$ where $\Phi\Phi$ Is the *Golden ratio* $\left(\Phi = \frac{1+\sqrt{5}}{2}\right)\left(\Phi = \frac{1+\sqrt{5}}{2}\right)$ [20].

Based on the flowchart above, the Knuth Morris Pratt algorithm performs the initial process by initializing the variables i and j as a place to perform shift calculations when there is no match between pattern and string. Furthermore, the pattern and string matching will be carried out, and if there is a condition, if the two are matched, it will be stored in the new variable as the matching result. Conversely, when there is no match, a shift will be carried out from left to right.

Algorithm Knuth Morris Pratt:

Step 1: Enter *query* the word query to search. With

examples $P = PatternP = Pattern$ the pattern of the wording used as an example pattern of text to be searched

$$T = Text\ or\ document\ title$$

$T = Text\ or\ document\ title$

Step 2: The KMP algorithm starts matching the pattern from the left direction. To fulfil $pattern[i] = teks[j]pattern[i] = teks[j]$. When $pattern[i]pattern[i]$ Not equal to $teks[j]teks[j]$, Then the value $i = i - offsets[i]$ $i = i - offsets[i]$ and value $j = j + offsets[i]$ $j = j + offsets[i]$.

Step 3: If $pattern[i] = teks[j]$ $pattern[i] = teks[j]$, Then the value ii will be checked $i = len(pattern)i = len(pattern)$. If not equal, then it will shift one step $(i = i + 1)$ $(i = i + 1)$. If $i = len(pattern)i = len(pattern)$ character will be saved and to the value $i = 1i = 1$

Step 4: Then, check the value $j = len(text)$ $j = len(text)$. If not the same, then $j = j + 1$ $j = j + 1$ And go back to step 2. If $j = len(text)$ $j = len(text)$, Then the output will be.

Example:
Target: ALA
Text: AJE, ALA
Pattern: ALA

Work steps:

Step 1

Pattern	A	L	A
Text 1	A	J	E
Text 2	A	L	A

Step 1: For the first character in the pattern, there is a match with the first character in text one, and then the role will be saved and move from left to right, i.e., the second character in the pattern and text one.

Step 2

Pattern	A	L	A
Text 1	A	J	E
Text 2	A	L	A

Step 2: For the second character in the pattern, there is no match with the second character in text 1, and the model will start matching back from the first character with the second character in text one.

Step 3

Pattern		A	L	A
Text 1	A	J	E	
Text 2	A	L	A	

Step 3: For the first character in the pattern, there is no match with the second character in text 1, the model will start matching again from the first character with the third character in text one.

Step 4

Pattern			A	L	A
Text 1	A	J	E		
Text 2	A	L	A		

Step 4: Re-check for the first character in the pattern is not a match with the third character in the text one because it does not meet the matching text at one, so that pattern will begin to match the return of the first character to the first character in the text two.

Step 5

Text 1	A	J	E
Pattern	A	L	A
Text 2	A	L	A

Step 5: For the first character in the pattern, there is a match with the first character in text two, then the role will be saved and move to the right, the second character in the pattern and text two.

Step 6

Text 1	A	J	E
Pattern	A	L	A
Text 2	A	L	A

Step 6: For the second character in the pattern, there is a match with the second character in text two, and then the role will be saved again and move to the right, the third character in the pattern and text two.

Step 7

Text 1	**A**	**J**	**E**
Pattern	**A**	**L**	**A**
Text 2	**A**	**L**	**A**

Step 7: For the third character in the pattern, there is a match with the third character in Text two, and then the role will be saved. Because the characters in the model and the characters in text two have a 100% match, the matching process stops then the saved character as a string matching has a 100% match.

IV. RESULT AND DISCUSSION

The results of the KMP algorithm, in general, the process carried out is matching string matching characters to get a match from the first index to the last index. The string results that match integrity are in table 2. Words that match the characters and punctuation are in table 3, and finally, the KMP implementation by finding the execution time and memory usage in table 4.

TABLE I. ANALYSIS OF STRING MATCHING WITH CHARACTERS

Work	Translate	KMP
Ala	Ambil	Accepted
Allo	Matahari	Accepted
Barah	kenyang	Accepted
Ceba	Monyet	Accepted
Dale	Jagung	Accepted
Gaja	Sangat	Accepted
Jabe	Manja	Accepted
Jama	kerja	Accepted
Toda	juga	Accepted
Tuo	hidup	Accepted
Yara	Jika	Accepted
Gaja	Sangat	Accepted
Jabe	Manja	Accepted
Jama	kerja	Accepted
Toda	juga	Accepted
Tuo	hidup	Accepted
Yara	Jika	Accepted
Gaja	Sangat	Accepted
Jabe	Manja	Accepted
Accuracy		100%

Based on table 2 above, matching strings with input in the form of characters without punctuation combinations obtained 100% accuracy. Likewise, table 3, with the implementation of the KMP algorithm and input testing in the way of styles combined with the results of punctuation, shows the same accuracy.

TABLE II. ANALYSIS OF STRING MATCHING WITH CHARACTERS AND READ

Work	Translate	KMP
Uki'	Tulis	Accepted
Ti-tungo	Terbalik	Accepted
Na-tumang	Diakibatkan	Accepted
Tuka'	naik	Accepted
To'Tok	Lubang	Accepted
Tonna anu	Dulu	Accepted
Ma-sipa'	Jahat	Accepted
Lado'	Jomblo	Accepted
Kantoro'	Kantor	Accepted
Doi'	Uang	Accepted
Uki'	Tulis	Accepted
Ti-Tungo	Terbalik	Accepted
Na-Tumang	Diakibatkan	Accepted
Tuka'	naik	Accepted
To'Tok	Lubang	Accepted
Tonna Anu	Dulu	Accepted
Ma-Sipa'	Jahat	Accepted
Lado'	Jomblo	Accepted
Accuracy		100%

TABLE III. ANALYSIS OF EXECUTION TIME AND USE OF MEMORY

Work	Translate	Time(ms)	Memory(MB)
Mamma	Tidur	0.02571	15.991
Ambe	Bapak	0.02056	15.969
Alli	Beli	0.01542	15.968
Uran	Hujan	0.02056	15.502
To'	Pohon	0.02056	15.513
Tallo	Telur	0.02056	15.481
Sipa'	Sikat	0.01541	15.981
Petada	Meminta	0.02056	15.331
Meki	Mari	0.01542	15.969
Bale	Ikan	0.01543	15.969
Mamma	Tidur	0.02571	15.991
Ambe	Bapak	0.02056	15.969
Alli	Beli	0.01542	15.968
Uran	Hujan	0.02056	15.502
To'	Pohon	0.02056	15.513
Tallo	Telur	0.02056	15.481
Sipa'	Sikat	0.01541	15.981
Petada	Meminta	0.02056	15.331
Everage		0.01901 ms	15.768 MB

V. CONCLUSION

The enrekang-Indonesian translator can use a string matching algorithm for their implementation. The results of testing the classification algorithm, according to KMP, were found to be 100% able to translate the words entered. From a performance point of view, the execution time in the implementation is 0.01901 ms, and the memory algorithm usage is only 15.765 MB.

To improve the quality of future research, researchers need to do a combination of string algorithms that can translate a sentence so that the research carried out is more complex.

REFERENCES

[1] R. K. Hondro, Z. A. Hsb, and R. D. Sianturi, "Aplikasi Penerjemahan Bahasa Mandailing-Indonesia," *JURIKOM (Jurnal Ris. Komputer)*, vol. 3, no. 4, pp. 49–53, 2016.

[2] E. Hussain, A. Hannan, and K. Kashyap, "A Zoning based Feature Extraction method for Recognition of Handwritten Assamese Characters," *Int. J. Comput. Sci. Technol.*, vol. 8491, no. April 2019, pp. 226–228, 2015.

[3] A. Fatah *et al.*, "Application of knuth-morris-pratt algorithm on web based document search," *J. Phys. Conf. Ser.*, vol. 1175, no. 1, 2019.

[4] H. W. Herwanto, A. N. Handayani, K. L. Chandrika, and A. P. Wibawa, "Zoning Feature Extraction for Handwritten Javanese Character Recognition," *ICEEIE 2019 - Int. Conf. Electr. Electron. Inf. Eng. Emerg. Innov. Technol. Sustain. Futur.*, pp. 264–268, 2019.

[5] H. Azis, R. D. Mallongi, D. Lantara, and Y. Salim, "Comparison of Floyd-Warshall Algorithm and Greedy Algorithm in Determining the Shortest Route," *Proc. - 2nd East Indones. Conf. Comput. Inf. Technol. Internet Things Ind. EIConCIT 2018*, pp. 294–298, 2018.

[6] S. A. Sini, "Enhanced Pattern Matching Algorithms for Searching Arabic Text Based on Multithreading Technology," pp. 0–6, 2019.

[7] X. Lu, "The Analysis of KMP Algorithm and its Optimization," 2019.

[8] R. Apriyadi, "Knuth Morris Pratt - Boyer Moore Hybrid Algorithm for Knowledge Management System Model on Competence Employee in Petrochemical Company," pp. 201–206, 2019.

[9] E. Ermatita and D. Budianta, "Fuzzy Knuth Moris Pratt Algorithm for Knowledge Management System Model on Knowledge Heavy Metal Content in Oil Plants," pp. 188–192, 2017.

[10] R. Rahim, I. Zulkarnain, and H. Jaya, "A review: Search visualization with Knuth Morris Pratt algorithm," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 237, no. 1, 2017.

[11] U. S. Alzoabi, N. M. Alosaimi, A. S. Bedaiwi, and A. M. Alabdullatif, "Parallelization of KMP String Matching Algorithm," pp. 5–7.

[12] P. Cao and S. Wu, "Parallel Research on KMP Algorithm," pp. 4252–4255, 2011.

[13] S. Aygün, E. Olcay, and L. Kouhalvandi, "Python Based Parallel Application of Knuth – Morris – Pratt Algorithm," 2016.

[14] L. S. Riza, A. B. Rachmat, and T. Hidayat, "Genomic Repeat Detection Using the Knuth-Morris-Pratt Algorithm on R High-Performance-Computing Package," vol. 11, no. 1, 2019.

[15] S. Park, D. Kim, N. Park, and M. Lee, "High Performance Parallel KMP Algorithm on a Heterogeneous Architecture," *2018 IEEE 3rd Int. Work. Found. Appl. Self* Syst.*, pp. 65–71, 2018.

[16] 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," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 122, no. 1, 2016.

[17] H. Handrizal, A. Budiman, and D. R. Ardani, "Implementation and Analysis Zhu-Takaoka Algorithm and Knuth-Morris-Pratt Algorithm for Dictionary of Computer Application Based on Android," *IJISTECH (International J. Inf. Syst. Technol.*, vol. 1, no. 1, p. 8, 2017.

[18] K. Pengantar, "Kamus Massenremp ulu – Indonesia (Dialek Duri)."

[19] M. B. Sri, R. Bhavsar, and P. Narooka, "String Matching Algorithms," vol. 7, no. 3, pp. 23769–23772, 2018.

[20] C. and Charras and T. Lecroq, "Handbook of Exact String Matching Algorithms," p. 238, 2004.