JITS (Journal of Informatics, Technology and Science)
Vol 1, No 1, August 2025, pp. 26-30
ISSN: xxxx - xxxx (Online Media)
DOI: -
https://jurnal.katersipublisher.com/index.php/JITS/index

Implementation of the Boyer Moore Algorithm in a Web-Based Accounting Terminology Dictionary Application

Azlan

STMIK Triguna Dharma, Medan, Indonesia

Corresponding Email: Azlansaja19@gmail.com

Abstract. The Boyer Moore algorithm is one of the most efficient string matching algorithms, known for using bad character and good suffix heuristics to speed up the process of searching for patterns in text. This study aims to implement the Boyer Moore algorithm in a web-based accounting dictionary application to improve efficiency in searching for terms that match the keywords used by users. The research method used is qualitative. The results of this study conclude that the Boyer-Moore algorithm is capable of providing efficient and accurate results in accounting term searches. By utilizing Occurrence Heuristic and Mismatch Heuristic, this algorithm speeds up the pattern matching process without checking the entire text, making it ideal for application in web-based accounting term dictionary applications that require fast, accurate, and responsive searches.

Keywords: Implementation, Boyer Moore Algorithm, Dictionary Application, Accounting Terms

1. INTRODUCTION

Searching for terms in a digital dictionary is a key aspect of many applications, especially in the field of accounting, where academics, practitioners, and researchers often need quick definitions for various accounting terms. Web-based accounting dictionaries generally contain thousands of entries, ranging from basic terms such as assets, liabilities, and equity to more specific terms related to auditing, taxation, accounting standards, and corporate finance. With digital access to these dictionaries, system response time for searching terms becomes a key factor in the user experience.

Conventional search methods often use linear searches that analyze each entry individually. This method is simple but inefficient when the number of entries is very large, when search terms are used frequently, or when the type of search is repetitive. Delayed responses can lead to a poor user experience and a decline in productivity. Therefore, the development of more efficient string matching methods is urgently needed.

Boyer-Moore is a string matching algorithm that is relevant for solving pattern search problems in text, especially if the pattern is relatively short compared to very long text [1]. Boyer-Moore uses the bad character rule and good suffix rule heuristics in the process of determining how far the pattern can be shifted when a mismatch occurs, so that most strings in the text can be skipped without being checked directly [2].

Several previous studies have applied Boyer-Moore as a solution to solve the problem of searching for words in large databases. For example, research by [3] on data search on MSME websites, research by [4] on text search in work report information systems, and research by [5] on searching for library book titles. Based on the results of the previous studies mentioned above, it can be concluded that Boyer-Moore can facilitate the word search process compared to the general linear search method.

Thus, this study proposes the implementation of the Boyer-Moore algorithm in accounting terminology dictionary applications. The main objective is to make it easier for academics, practitioners, and researchers to find the meanings of accounting terms in web-based accounting terminology dictionaries. The benefits of this study are expected to help developers of dictionary applications, e-learning systems, and accounting reference platforms to provide fast, accurate, and responsive term search services.

JITS (Journal of Informatics, Technology and Science)
Vol 1, No 1, August 2025, pp. 26-30
ISSN: xxxx - xxxx (Online Media)
DOI: -
https://jurnal.katersipublisher.com/index.php/JITS/index

2. METHODOLOGY

2.1 Research Stages

The research method used by the author in this study is qualitative. The stages of research used are as follows:

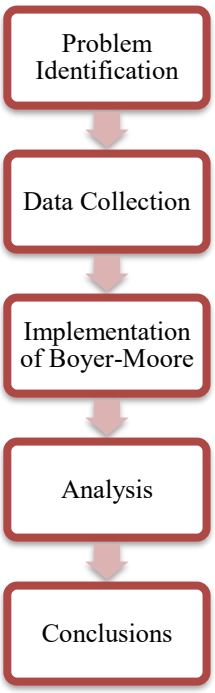

Figure 1. Research Stages

2.2 String Matching

String matching is the process of finding one or more occurrences of a pattern (substring) within a longer text [6]. This process is a basic operation in computer science and is widely used in various fields, such as search engines, DNA analysis, text editors, and spam filtering [7], [8].

2.3 Boyer-Moore

The Boyer-Moore algorithm is a classic string search algorithm that efficiently finds all occurrences of a specific pattern in a text. It does this by traversing sections of text using heuristics rather than checking each character sequentially. It uses two main heuristics, namely the bad character rule and the good suffix rule, to determine how far the pattern can be shifted when a mismatch occurs, thereby speeding up the search process [1], [9].

2.4 Accounting Glossary

An accounting dictionary is a reference book containing a collection of accounting terms and their explanations, which is used to help students, accountants, auditors, entrepreneurs, and business people understand accounting concepts [10].

JITS (Journal of Informatics, Technology and Science)
Vol 1, No 1, August 2025, pp. 26-30
ISSN: xxxx - xxxx (Online Media)
DOI: -
https://jurnal.katersipublisher.com/index.php/JITS/index

3. RESULTS AND DISCUSSION

3.1 Results

In this study, the case study was completed using the Boyer-Moore algorithm, which matches patterns with text that has a database related to accounting terms, where the pattern used is "Loss" and the text used is "Profit and Loss Statement." The results of implementing the Boyer-Moore algorithm in the case study can be seen in the following description:

Table 1. Occurence Heuristic (OH) dan Mismatch Heuristic (MH)

Index	0	1	2	3
Pattern	R	U	G	I
OH	3	2	1	0
MH	4	4	4	4

a. First string match
In the first string matching stage, the character I is matched with O in Text. However, Table 2 shows that in this first string matching, there is no match and O does not exist in the pattern, so a shift must be performed. The shift for the next matching is performed in 4 steps (equal to the MH value).

Table 2. First String Match

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Text	L	A	P	O	R	A	N		L	A	B	A		R	U	G	I
Pattern	R	U	G	I													

b. Second string matching
In the second string matching stage, the character I is matched with Space in Text. However, Table 3 shows that in the second string matching, there is also no match and Space is not in the pattern, so a shift must be performed. The shift for the next matching is performed in 4 steps (equal to the MH value).

Table 3. Second String Matching

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Text	L	A	P	O	R	A	N		L	A	B	A		R	U	G	I
Pattern					R	U	G	I									

c. Third string matching
In the third string matching stage, the character I is matched with A in Text. However, Table 4 shows that in the third string matching, there is also no match and A is not in the pattern, so a shift must be performed. The shift for the next match is performed by 4 steps (equal to the MH value).

Table 4. Third String Matching

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Text	L	A	P	O	R	A	N		L	A	B	A		R	U	G	I
Pattern									R	U	G	I					

d. Fourth string matching
In the fourth string matching stage, the character I is matched with G in Text. However, Table 5 shows that in the fourth string matching, there is also no match and G is in the pattern, so a shift must be made. The shift for the next match is made by 1 step (equal to the OH value of G).

Table 5. Fourth String Matching

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Text	L	A	P	O	R	A	N		L	A	B	A		R	U	G	I
Pattern													R	U	G	I	

JITS (Journal of Informatics, Technology and Science)
Vol 1, No 1, August 2025, pp. 26-30
ISSN: xxxx - xxxx (Online Media)
DOI: -
https://jurnal.katersipublisher.com/index.php/JITS/index

e. Fifth string match
In the fifth string matching stage, the character I is matched with I in Text. Table 5 shows that in the fifth string matching, there is a match between I and I. Therefore, the next matching is between the characters G and G. Since the matching of the characters G and G also results in a match, another matching is performed between the characters U and U. The matching of the characters U and U also results in a match. Next, another match is performed between the character R and R. In this last match, a match is also found between the pattern character and the text. Thus, the string matching process is complete because the pattern character as a whole has been found to match in the text.

Table 6. Fourth String Matching

Index	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
Text	L	A	P	O	R	A	N		L	A	B	A		R	U	G	I
Pattern														R	U	G	I

3.2 Discussion

Based on the results of the implementation, it can be seen that the Boyer-Moore algorithm is capable of efficiently matching patterns in texts containing accounting terms. In this case study, the pattern used is the word "Loss", while the text being searched for is "Profit and Loss Statement". The matching process was carried out through systematic stages using two main approaches in the Boyer-Moore algorithm, namely Occurrence Heuristic (OH) and Mismatch Heuristic (MH).

From the results presented in Tables 2 to 6, it can be seen that the Boyer-Moore algorithm does not match characters sequentially from the beginning of the text, but rather starts from the end of the pattern and shifts based on heuristic values. This approach saves computing time because not all characters in the text need to be checked one by one. In the first to third stages, there are consecutive mismatches between the last character of the pattern (I) and characters in the text that do not match (O, Space, and A). Because these characters are not found in the pattern, the algorithm shifts four steps based on the Mismatch Heuristic value (MH = 4).

This step illustrates the main strength of Boyer-Moore, which is its ability to "skip" parts of the text that clearly cannot contain the pattern, thereby speeding up the search process. Compared to simple matching algorithms such as Brute Force, which match characters one by one from the beginning of the text, Boyer-Moore can significantly reduce the number of character comparisons.

In the fourth match, the algorithm found a mismatch between the character I in the pattern and the character G in the text. However, because G is present in the pattern, the shift made is smaller, only one step, based on the Occurrence Heuristic value (OH = 1). This adaptive shift shows that the algorithm is not only efficient in cases of large mismatches (characters not present in the pattern), but also flexible in adjusting the shift when some relevant characters are found.

This stage proves that the combination of two heuristics, namely the bad character rule (OH) and the good suffix rule (MH), provides a balance between accuracy and efficiency in the matching process. In accounting terminology dictionary systems that have hundreds to thousands of terms, mechanisms such as this are very important for maintaining real-time term search performance.

In the fifth matching stage, it was found that all characters in the pattern ("R", "U", "G", "I") were successfully matched sequentially with the text ("R", "U", "G", 'I') at the end of the string "Profit and Loss Report". This condition indicates that the algorithm successfully found the complete pattern in the text. This process ends the search iteration because all characters in the pattern have been matched.

These results show that the Boyer-Moore algorithm works very efficiently in the context of searching for accounting terms, which are generally specific and relatively short. Matching only requires five iterations to find the correct result, which is much faster than conventional approaches that may require checking every character position in the text.

In the context of developing a web-based accounting dictionary application, the use of the Boyer-Moore algorithm is highly relevant because it speeds up the process of searching for terms when users enter specific keywords. For example, when a user types the word "Loss," the system can efficiently check thousands of accounting terms stored in the database and display the most relevant results with a fast response time.

In addition, accounting terms often vary in length, such as "Profit and Loss," "Revenue," or "Fixed Assets," and can be processed optimally using the adaptive shifting mechanism of the Boyer-Moore algorithm. This can improve the user experience when searching for terms interactively within the application.

JITS (Journal of Informatics, Technology and Science)
Vol 1, No 1, August 2025, pp. 26-30
ISSN: xxxx - xxxx (Online Media)
DOI: -
https://jurnal.katersipublisher.com/index.php/JITS/index

4. CONCLUSION

Based on the results of implementation and analysis, the Boyer-Moore algorithm has proven to be efficient in searching for accounting terms in text. Through the Occurrence Heuristic and Mismatch Heuristic mechanisms, the matching process can be carried out with a minimal number of character comparisons. This algorithm is able to find patterns quickly without checking all characters in the text, making it very suitable for application in web-based accounting term dictionaries that require fast, accurate, and responsive searches for various lengths of keywords or terms entered by users.

References

[1] 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, p. 012034, Apr. 2016, doi: 10.1088/1757-899X/122/1/012034.

[2] S. M. Morsy and D. Nashat, "VB-ARP : Boyer – Moore Majority Voting Algorithm Based Defense," vol. 4, no. 4, pp. 172–177, 2024.

[3] A. Aman Bangsa, B. Pramono, and L. Bahtiar Aksara, "Penerapan String Matching Menggunakan Algoritma Boyer Moore untuk Mencari Data Pada Website UMKM di Konawe Selatan," *Animator*, vol. 2, no. 1, pp. 18–25, 2024, [Online]. Available: https://animator.uho.ac.id/index.php/journal/article/view/41

[4] M. Kahfi Djardjani and D. Arwin Dermawan, "Penerapan Algoritma Boyer Moore untuk Pencarian Teks dalam Sistem Informasi Laporan Kerja di Fakultas Vokasi," *JMI (Jurnal Manaj. Inform.*, vol. 16, no. 2, p. 1, 2024, [Online]. Available: https://ejournal.unesa.ac.id/index.php/jurnal-manajemen-informatika/article/view/60962

[5] Isnawati, D. Handayani, and A. Noeman, "a. Implementation of Boyer-Moore," *J. Inform. Inf. Secur.*, vol. 3, no. 1, pp. 49–58, 2022, doi: 10.31599/jiforty.v3i1.1182.

[6] S. S. Abdul-Jabbar, A. K. Farhan, A. A. Abdelhamid, and M. E. Ghoneim, "Razy: A String Matching Algorithm for Automatic Analysis of Pathological Reports," *Axioms*, vol. 11, no. 10, pp. 1–15, 2022, doi: 10.3390/axioms11100547.

[7] F. A. T. Tobing, A. Chandra, and R. Nainggolan, "Penerapan Algoritma Knuth Morris Pratt (KMP) Pada Pencarian Data Di SQL Like Operators," *J. Widya*, vol. 3, no. 1, pp. 50–58, 2022, [Online]. Available: https://doi.org/10.54593/awl.v3i1.82%0Ahttps://jurnal.amikwidyaloka.ac.id/index.php/awl/article/view/82/

[8] L. Sernicula, M. Manalo, G. A. Niebre, R. Anastacio, J. Alagos, and J. Sermeno, "Analysis of the Enhanced Boyer-Moore Search Algorithm for a Desktop PC Search Engine," *J. Innov. Technol. Converg.*, vol. 5, no. 2, pp. 45–56, Dec. 2023, doi: 10.69478/JITC2023v5n2a05.

[9] A. Mubarak, M. Ikhsan, and A. Putri, "Penerapan Web Semantik Berdasarkan Ontologi Pada Pencarian Judul Skripsi Dengan Algoritma Boyer-Moore," *J. Ilm. Komputasi*, vol. 23, no. 2, pp. 189–200, 2024, doi: 10.32409/jikstik.23.2.3587.

[10] R. Annisa, "Perancangan Kamus Istilah Akuntansi Berbasis Desktop Dengan Metode Interpolation Search," *Evolusi*, vol. 5, no. 1, pp. 45–52, 2017.