

PLAGIARISM DETECTION IN RESEARCH OF LECTURERS AND STUDENTS AT BUDI DARMA UNIVERSITY APPLYING THE TURBO ALGORITHM BOYER MOORE

Alwin Fau ¹, Nelly Astuti Hasibuan ²

Universitas Budi Darma, Universitas Budi Darma

Email alwinfau@gmail.com nelly.ahsb@gmail.com

Article Info

Received, 01/08/2022

Revised, 29/08/2022

Accepted, 30/08/2022

Research conducted by students or lecturers is a form of creativity and also a sense of sharing knowledge and experience with others. Research for a lecturer or lecturer is something that must be done by the lecturer because it is part of the lecturer's duties. Research for students is a step that must be carried out by every student at the time of completing studies in each study program that is being carried out. The form of research produced by students and lecturers is in the form of scientific articles. Scientific articles which are the result of research by lecturers and students will be published in certain journals depending on the researcher. Before the research of lecturers and students is published in a journal, the article needs to be checked first to avoid plagiarism in the research. Budi Darma University always has problems with that checking every article being researched by lecturers and students. This study will discuss how the processes and stages are carried out in checking lecturers and students. This research will produce an application that can be used by Budi Darma University in checking scientific articles. The results of checking plagiarism with the Turbo Boyer algorithm can detect more than 75%.

Keywords: String Matching, Research, Turbo Boyer Moore, Searching

1. Introduction

With the rapid development of technology today, the world is governed by the development of this technology. When viewed from the side of technological developments, humans today cannot be separated from technology. The development of this technology is part of human life. One form or part of technology that cannot be separated from human life is gadgets and the internet. The development of technology has triggered many people to do some research on the development of the technology. Anyone can conduct research in various fields, both in the fields of biology, economics, and technology. The results of each research carried out will be published in various existing media. Budi Darma University is a private university (PTS) where students and lecturers will conduct research every year. This research is conducted because it is the duty and obligation of lecturers and students. The results of the research conducted. The results of research conducted by lecturers and students are in the form of articles. The articles resulting from the research are then published in certain journals. When researchers publish articles to journals, the articles must be checked first to avoid plagiarism [1][2]. Budi Darma University experienced several obstacles in checking each article which was the result of research by lecturers and students. So for that it is necessary to hold an application that can help check articles that are the results of student and lecturer research. In connection with the plagiarism rules, if it is known that the articles

produced by students are the result of plagiarism from other people's research, then the rules or sanctions that will be given to students are cancellation of diplomas and bad track records for lecturers.

2. Algorithm

The term algorithm comes from the name of an Arab author named Abu Ja'far Mohammed bin Musa al Khwarizmi (790-840 years), who is very well known as the father of Algebra [9][10]. An algorithm is a well-defined computation that takes some values as input and produces some values which are called outputs. So, an algorithm is a series of computational steps that transform input into output. Judging from the origin of the word, the word "algorithm" has a quite strange history.

2.1 String Matching

String Matching is the process of searching for all occurrences of the query, which is then called a pattern into a longer string (text) [3][4][5]. String Matching can be formulated as follows:

$$x = x(0 \dots m-1) \dots \dots \dots (3.1)$$

$$y = y(0 \dots n-1) \dots \dots \dots (3.2)$$

Information :

x is pattern

m is the length of pattern

y is text

n is the length of the text

The two strings consisting of a set of characters are called the alphabet, denoted by (σ) and have a size of ($|\sigma|$). String Matching is divided by or arranged into two, namely Exact Matching and Heuristic or Statistical Matching [6].

2.2 Extra Matching

Exact Matching is used to find patterns that come from text from one text. An example of an exact matching search is the search for the word "student" in the sentences "I am a student" or "I am a student". The system will give the result that the first sentence contains the word "student" while the second sentence does not, even though the words student and student are synonymous, the exact matching algorithm is classified into three parts according to the direction of the search, namely [7]:

- Reading direction from left to right. Algorithms that fall into this category are Bruce Force, and Knuth Morris Pratt (developed by Knuth, Morris, and Pratt).
- The reading process is from right to left. Algorithms that fall into this category are Boyer Moore, which were later developed into Turbo Boyer-Moore, Tuned Boyer-Moore, and Zhu-Takaoka algorithms.

2.3 Turbo Boyer Moore Algorithm

The Turbo Boyer-Moore algorithm was introduced by Bob Boyer and J.S. Moore in 1977. In this algorithm, word matching starts from the last character of the keyword to its initial character. If there is a difference between the last character of the keyword and the matched word, the characters in the matched word snippet will be checked one by one. This matter. The Boyer Moore algorithm is a string search algorithm that searches by comparing a letter with the letters in the pattern being searched for, and shifting the pattern until its position is the same as the text being searched for and comparing the words [11].

- The Looking-Glass Technique

The Looking-Glass Technique compares a final character in the word w with a character in the text s . If the characters are the same then the character window will run backward on both strings and

recheck both characters [10]. Searching for a string match in the text with the pattern to be searched by moving or sliding it until the text string is complete.

b. The Character-Jump Technique

Character-jump Technique acts when a comparison between two different characters

3. Results and Discussion

To search and detect similarities using the Turbo Boyer Moore algorithm, it can be seen that the steps are as follows:

Example:

Text : deteksi plagiasi penelitian algoritma mahasiswa boyer

pattern : boyer

The first step is to calculate the BmBc table and the BmGs table.

Table 1. BmBc dan BmGs

index	0	1	2	3	4
Pattern	b	o	y	e	r
Pergesearn BmBc	4	3	2	1	0
Pergeseran BmGs	4	4	4	4	1

The first stage of pattern detection can be seen in the process below:

a. Pettern Match 1

Table 2 Pettern Match 1

index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
text	d	e	t	e	k	s	i		p	l	a	g	i	a	s	i		p	e	n
petern	b	o	y	e	r															

index	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
text	e	l	i	t	i	a	n		a	l	g	o	r	i	t	m
petern																

index	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53
text	a		m	a	h	a	s	i	s	w	a		b	o	y	e	r
petern																	

In the first pattern meeting, there is a difference between index n, namely k and r, so: Shift by BmBc $[n] - m + (\text{bottom index} + 1) = 5 - 5 + 5 = 5$, Shift by BmGs $[4] = 1$, So the pettern shift is 5 (maximum value of both calculations).

b. Pettern Match 2

Table 3. Pettern Match 2

index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
-------	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----

text	d	e	t	e	k	s	i	p	l	a	g	i	a	s	i	p	e	n
petern					b	o	y	e	r									

index	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
text	e	l	i	t	i	a	n		a	l	g	o	r	i	t	m
petern																

index	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53
text	a		m	a	h	a	s	i	s	w	a		b	o	y	e	r
petern																	

In the second pattern meeting, there is a difference between index n, namely l and r, so: Shift by BmBc $[n] - m + (\text{bottom index} + 1) = 5 - 5 + 5 = 5$, Shift by BmGs $[4] = 1$, So the pettern shift is 5 (maximum value of both calculations).

c. Petter Match 3

Table 4. Pettern Match 3

index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
text	d	e	t	e	k	s	i		p	l	a	g	i	a	s	i		p	e	n
petern											b	o	y	e	r					

index	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
text	e	l	i	t	i	a	n		a	l	g	o	r	i	t	m
petern																

index	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53
text	a		m	a	h	a	s	i	s	w	a		b	o	y	e	r
petern																	

In the third pattern meeting, there is a difference between index n, namely s and r, so: Shift by BmBc $[n] - m + (\text{bottom index} + 1) = 5 - 5 + 5 = 5$, Shift by BmGs $[4] = 1$, So the pettern shift is 5 (maximum value of both calculations).

d. Pettern Match 4

Table 5. Pettern Match 4

index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
text	d	e	t	e	k	s	i		p	l	a	g	i	a	s	i		p	e	n
petern																b	o	y	e	r

index	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
text	e	l	i	t	i	a	n		a	l	g	o	r	i	t	m
petern																

index	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53
text	a		m	a	h	a	s	i	s	w	a		b	o	y	e	r
petern																	

In the fourth pattern meeting, there is a difference between index n, namely n and r, so: Shift by BmBc [n] – m + (bottom index + 1) = 5 – 5 + 5 = 5, Shift by BmGs [4] = 1, So the pettern shift is 5 (maximum value of both calculations)

e. Pettern Match 5

Table 6. Pettern Match 5

index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
text	d	e	t	e	k	s	i		p	l	a	g	i	a	s	i		p	e	n
petern																				

index	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
text	e	l	i	t	i	a	n		a	l	g	o	r	i	t	m
petern	b	o	y	e	r											

index	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53
text	a		m	a	h	a	s	i	s	w	a		b	o	y	e	r
petern																	

In the fifth pattern meeting, there is a difference between index n, namely i and r, so: Shift by BmBc [n] – m + (bottom index + 1) = 5 – 5 + 5 = 5, Shift by BmGs [4] = 1, So the pettern shift is 5 (maximum value of both calculations)

f. Pettern Match 6

Table 7. Pettern Match 6

index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
text	d	e	t	e	k	s	i		p	l	a	g	i	a	s	i		p	e	n
petern																				

index	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
text	e	l	i	t	i	a	n		a	l	g	o	r	i	t	m
petern						b	o	y	e	r						

index	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53
text	a		m	a	h	a	s	i	s	w	a		b	o	y	e	r
petern																	

- In the sixth pattern meeting, there is a difference between index n, namely g and r, so: Shift by BmBc $[n] - m + (\text{bottom index} + 1) = 5 - 5 + 5 = 5$, Shift by BmGs $[4] = 1$, So the pettern shift is 5 (maximum value of both calculations).
- At the seventh pattern meeting, there is a difference between index n, namely m and r, so: Shift by BmBc $[n] - m + (\text{bottom index} + 1) = 5 - 5 + 5 = 5$, Shift by BmGs $[4] = 1$, So the pettern shift is 5 (maximum value of both calculations)
- In the eighth pattern meeting, there is a difference between index n, namely h and r, so: Shift by BmBc $[n] - m + (\text{bottom index} + 1) = 5 - 5 + 5 = 5$, Shift by BmGs $[4] = 1$ So the pettern shift is 5 (maximum value of both calculations)
- At the ninth pattern meeting, there is a difference between index n, namely w and r, so: Shift by BmBc $[n] - m + (\text{bottom index} + 1) = 5 - 5 + 5 = 5$, Shift by BmGs $[4] = 1$ So the pettern shift is 5 (maximum value of both calculations)
- At the tenth pattern meeting, there is a difference between index n, namely y and r, and in the text pad there is an index that exceeds the number of patterns, then the pattern will be shifted as much as the number exceeds the pettern so that: Shift by BmBc $[n] - m + (\text{bottom index} + 1) = 5 - 5 + 1 = 2$, Shift by BmGs $[4] = 1$, So the pettern shift is 2 (maximum value of both calculations)

g. Petter Match 11

Table 8. Pettern Match 11

index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
text	d	e	t	e	k	s	i		p	l	a	g	i	a	s	i		p	e	n
petern																				

index	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
text	e	l	i	t	i	a	n		a	l	g	o	r	i	t	m
petern																

index	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53
text	a		m	a	h	a	s	i	s	w	a		b	o	y	e	r
petern													b	o	y	e	r

At the eleventh pattern meeting, there is no need to shift again because it has arrived at the last index and has found the pattern in accordance with.

4. Conclusions

Based on the results of the tests carried out above, it proves that the Turbo Boyer Moore algorithm can detect the similarity of text in scientific papers by matching characters between pattern and task by matching characters starting from the right side of the pattern. If the first character does not match the text

then the shift will be made based on the number of patterns. The process of detecting the similarity of texts in scientific works of students and lecturers can be carried out according to the steps of the Turbo Boyer Moore algorithm. Looking at the performance results, it can be said that the Turbo Boyer Moore algorithm can detect plagiarism in scientific papers by eliminating some special characters to improve pattern search results with text. The results of the Turbo Boyer Moore algorithm will search until all the patterns in the text match.

Reference

- [1] S. D. Nasution and Mesran, "Goldbach Codes Algorithm for Text Compression," *IJournals Int. J. Softw. Hardw. Res. Eng.*, vol. 4, no. December, pp. 43–46, 2016.
- [2] J. Martina and B. Panjaitan, "Penerapan Algoritma Fibonacci Codes Pada Kompresi Aplikasi Audio Mp3 Berbasis Dekstop," vol. 1, no. 1, pp. 27–33, 2021.
- [3] M. A. Latif, S. D. Nasution, and Pristiwanto, "Analisa Perbandingan Algoritma Rice Codes Dengan Algoritma Goldbach Codes Pada Kompresi File Text Menggunakan Metode Exponential," *Maj. Ilm. INTI (Informasi dan Teknol. Ilmiah)*, vol. 13, no. 1, pp. 28–33, 2018.
- [4] E. Hariska, "Perancangan Aplikasi Kompresi File Gambar Menggunakan Algoritma Additive Code," vol. 5, pp. 193–202, 2021, doi: 10.30865/komik.v5i1.3671.
- [5] N. Aftikasyah, "Penerapan Algoritma Yamamoto ' s Recursive Code Untuk Mengkompresi File Dokumen," vol. 5, pp. 255–263, 2022, doi: 10.30865/komik.v5i1.3716.
- [6] D. Salomon, *Handbook of Data Compression*. London: Springer, 2007.
- [7] M. Yogie, "PENERAPAN ALGORITMA GOLDBACH CODES PADA KOMPRESI FILE GAMBAR TERENKRIPSI VIGENERE CIHPER," *J. Pelita Inform.*, vol. 7, pp. 81–85, 2018.
- [8] S. D. Nasution, G. L. Ginting, M. Syahrizal, and R. Rahim, "Data Security Using Vigenere Cipher and Goldbach Codes Algorithm," *Int. J. Eng. Res. Technol.*, vol. 6, no. 01, pp. 360–363, 2017.
- [9] D. Salomon and G. Motta, *HandBook Of Data Compression*. Springer, 2010.
- [10] R. Y. TANJUNG, "Perancangan aplikasi kompresi file dokumen menggunakan algoritma additive code," UNIVERSITAS BUDIDARMA MEDAN, 2020.