



Indonesian Proverbs Application Design by Applying Android Horspool Boyer Moore Algorithm

Andri Agus Yahya, Nelly Astuti Hasibuan, Putri Ramadhani

Department of Computer Science STMIK Budi Darma, North Sumatra, Indonesia

Abstract – Proverbs are words or groups of words which states a purpose or will, circumstance of someone, something, or something that express behavior, actions and things about a person indirectly, but implicitly convey a message that can be understood by both the reader and the listener. String matching is an algorithm to search for all occurrences of short and long strings, and for short strings called patterns and long strings called text. Horspool Algorithm is a simplification of the Boyer-Moore algorithm which in search of text is to search in large text to find the first pattern. Because the text sought can be very large (allowing hundreds of thousands of characters) it is important to use more efficient techniques based on that case, a search application based on android is made to make it easier for someone to find proverbs because not everyone understands this and that makes it more practical and efficient in terms of its use without having to carry proverbs in print media such as books.

Keywords– Application, Proverbs, Horspool Algorithm, Android

1 INTRODUCTION

Proverbs should not be forgotten because they are characteristic of Indonesia, even though proverbs are old literature at least must be able to keep up with today's technological advances. Proverbs can also be interpreted as an expression that, although not direct, but implicitly convey something that can be understood by the reader or audience. Proverbs must be conveyed in an attractive style and manner because they function for the language of advice. The main purpose of the creation of proverbs is as a means of communication, education, and satire with subtle words so that people who are reprimanded are not easily offended whether they admonish older people or younger people.

For Example, if there is someone who wants to rebuke the attitude of someone older than him, then he uses a proverb that contains subtle innuendo that contains satire or advice for the attitude of the person who is reprimanded. Horspool Algorithm is a simplification of Boyer-Moore Algorithm made by R. Nigel Horspool. According to Horspool, R.N. (1980), the problem in this text search is to search in large text to find the first pattern. Because the text sought can be very large (allowing hundreds of thousands of characters) it is important to use more efficient techniques.

The Horspool algorithm works with methods similar to the Boyer-Moore algorithm but does not make jumps based on the characters in the pattern found to be unsuitable in the text. The Horspool algorithm has the rightmost character shift value from the window. At the stage of initial observation (preprocessing), the shift value will be calculated for all characters. At this stage, the pattern from right to left is compared to the compatibility or mismatch pattern. The rightmost character in the window is used as an index to perform shift values. In the case of incompatibility (characters not in the pattern) occur, the window is shifted by the length of a pattern. If not, the window is shifted according to the rightmost character on the pattern.

2 THEORY

2.1 Proverbs

Proverbs are sentences or groups of words that state a purpose or will, a situation from someone, or a matter that expresses behavior, actions and things about a person. Proverbs can also be interpreted as an expression that, although indirect, but implicitly convey something that can be understood by the reader or audience. Proverbs must be conveyed in an attractive style and manner because they function for the language of advice. The main purpose of creating proverbs is as a means of communication, education, and satire with a subtle word so that those who are reprimanded are not easily offended either rebuke older people or younger people. For example, if there is someone who wants to rebuke the attitude of someone older than him, then he uses a proverb that contains subtle innuendo that contains satire or advice for the attitude of the person who is reprimanded.



2.2 String Matching

String is characters of a certain length. Strings are actually not pure basic types because they are composed of elements of the character type. However, because string types are often used in programming, strings can be treated as basic types of strings in computer science that can be interpreted as sequences of characters. Although it is often also considered as abstract data that stores data value sequences, it is usually in the form of bytes which is an element that is used to form characters according to the agreed character encoding such as ASCII, or EBCDIC. String matching algorithm is a method used to find the accuracy or results of one or more given text patterns.

2.3 Boyer-Moore Horspool Algorithm

Boyer-Moore Horspool Algorithm is the simplification of *Boyer-Moore Algorithm* found by R. Nigel Horspool. According to Horspool, R.N. (1980), the problem in this text search is to search in large text to find the first pattern. Because the text sought can be very large (allowing hundreds of thousands of characters) it is important to use more efficient techniques. The Horspool algorithm works with methods similar to the Boyer-Moore algorithm but does not make jumps based on the characters in the pattern found to be unsuitable in the text.

3 RESULT AND DISCUSSION

System analysis is the learning of a system and its components as a prerequisite for system design / system design and specifications of a new system. Moving from the classical definition of the analysis of this system to a more contemporary one, system analysis is a term that collectively describes the early phases of system development.

In the search application proverbs require string matching algorithms because in the proverb many characters are searched, therefore string stringing algorithms are needed to make it easier to find the desired number in large numbers. There are two stages in matching strings using the Horspool algorithm, namely:

1. Preprocessing stage

At this stage an observation pattern is observed for the text to construct a bad-match table that contains the shift value when a mismatch between the pattern and the text occurs. Systematically, the steps performed by Horspool algorithm in the preprocessing stage are:

- a. The Horspool algorithm matches the right-hand character on the pattern.
- b. Each character in the pattern is added to the bad-match table and the value of the shift is calculated.
- c. Characters that are on the pattern tip are not counted and are not used as the right character of the same character as them.
- d. If there are two characters in common and one of them is not the right-hand character, then the character with the largest index is calculated by the value of the shift.
- e. The Horspool algorithm stores the length of the pattern as the length of the shift value by default if the characters in the text are not found in the pattern.
- f. The value of the shift that will be used can be searched by calculating the length of the pattern minus the last index of the character minus 1, for each character, value = $m-i-1$.

2. Search phase

Systematically, the steps performed by Horspool algorithm in the preprocessing stage are:

- a. Comparison of the rightmost character of the pattern against the window is done.
- b. Bad-match tables are used to pass characters when a mismatch occurs.
- c. When there is a mismatch, the rightmost character in the window serves as a basis for determining the distance of the shift to be performed.
- d. After matching (whether the results match or do not match) a right-hand shift is made in the window
- e. This procedure is repeated until the window is at the end of the text or when the pattern matches the text.
- f. The following is an example of a text search for Indonesian proverbs using the Boyer Moore Horspool algorithm.

Systematically, the steps performed by Horspool algorithm in the preprocessing stage are

- a. Comparison of the rightmost character of the pattern against the window is done.
- b. Bad-match tables are used to pass characters when a mismatch occurs.
- c. When there is a mismatch, the rightmost character in the window serves as a basis for determining the distance of the shift to be performed.
- d. After matching (whether the results match or do not match) a right-hand shift is made in the window



- e. This procedure is repeated until the window is at the end of the text or when the pattern matches the text.

The following is an example of a text search for Indonesian proverbs using the Boyer Moore Horspool algorithm.

Pattern : TALAS
T A L A S
0 1 2 3 4
Pattern : TALAS
Index : 0 1 2 3 4

Each character in the patter is added to the bad-match table and the shift value is calculated. Here is a bad-match table.

Table 1. *Bad-match Making*

Pattern	Index	Value
T	0	4
L	2	2
A	3	1
S	4	5
*	-	5

Explanation :

- The point of value is calculated based on the formula: $Value = m - i - 1$
where : m = Numbers of Character
 i = Index
- The last index on the pattern is not calculated of the point of its value
- The point of the value of the unknown character has the last value of the numbers of the patterns
- If there are two characters in common and one of them is not the right-hand character, then the character with the largest index is calculated by the value of its shift.
 - Pattern T (Value = $5 - 0 - 1 = 4$)
 - Pattern L (Value = $5 - 2 - 1 = 2$)
 - Pattern A (Value = $5 - 3 - 1 = 1$)
 - Pattern S is the last *pattern*, so the point of its value is not calculated
 - * = unknown character.

Based on what seen on table 1 above, Bad-match initial initialization is performed. Each text and pattern are given a value of m and i , where m is the length of the pattern and i as the index.

Table 2. Iteration of The First *Boyer-Moore Horspool Algorithm*

m	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
P	R	A	G	A	I	_	A	I	R	_	D	I	_	D	A	U	N	_	T	A	L	A	S
T	T	A	L	A	S																		
i	0	1	2	3	4																		

In table 2, there is a mismatch between the "I" character and the "S" character in the rightmost window of the text. In a bad-match table, the shear value of the "I" character is 5 because the "I" character is an unknown character in making a bad-match table. So, there is a right shift on the window 5 times. This is seen in the Table 3.

Table 3. Iteration of The Second *Boyer-Moore Horspool Algorithm*

m	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
P	B	A	G	A	I	_	A	I	R	_	D	I	_	D	A	U	N	_	T	A	L	A	S
T																							
i																							



In table 3, the incompatibility returns between the "_" character with the "S" character in the rightmost window of the text. In a bad-match table, the "_" character value is 5 because the "_" character is an unknown character in making a bad-match table. So, there is a right shift on the window 5 times. This is shown in Table 4.

Table 4. Iteration of The Third *Boyer-Moore Horspool* Algorithm

m	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
P	B	A	G	A	I	_	A	I	R	_	D	I	_	D	A	U	N	_	T	A	L	A	S
T											T	A	L	A	S								
l											0	1	2	3	4								

In table 4, incompatibility occurs between the "A" character with the "S" character in the rightmost window of the text. In a bad-match table, the sliding value of the "A" character is 1 because the character "A" is a known character in making a bad-match table. So, do the right shift in the window 1 time. This is shown in Table 5.

Table 5. Iteration of The Fourth *Boyer-Moore Horspool* Algorithm

m	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
P	B	A	G	A	I	_	A	I	R	_	D	I	_	D	A	U	N	_	T	A	L	A	S
T											T	A	L	A	S								
l											0	1	2	3	4								

In the table 5, incompatibility occurs between the "U" character with the "S" character in the rightmost window of the text. In a bad-match table, the "U" character is 5 because the "U" character is an unknown character in making a bad-match table. So, there is a right shift on the window 5 times. This is shown in Table 6.

Table 6. Iteration of The Fifth *Boyer-Moore Horspool* Algorithm

m	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
P	B	A	G	A	I	_	A	I	R	_	D	I	_	D	A	U	N	_	T	A	L	A	S
T																			T	A	L	A	S
l																			0	1	2	3	4

In the table 6, re-mismatch occurs between the "L" character and the "S" character in the rightmost window of the text. In a bad-match table, the sliding value of the "L" character is 2 because the "L" character is a known character in making bad-match tables. So, there is a right shift on the window twice. This is shown in Table 7.

Table 7. Iteration of The Sixth *Boyer-Moore Horspool* Algorithm

m	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
P	B	A	G	A	I	_	A	I	R	_	D	I	_	D	A	U	N	_	T	A	L	A	S
T																			T	A	L	A	S
l																			0	1	2	3	4

In the table 7, the window is at the end of the text and all patterns match the text. All character matching using the Horspool algorithm has been completed and stopped at the sixth iteration.

4 IMPLEMENTATION

The display of Indonesian proverb layout is a display that forms or describes the front page design which contains several buttons including the Indonesian Proverb and About. Where Indonesian Proverbs contains several Layouts to search for Indonesian proverbs, but before carrying out the search process for Indonesian proverbs, there are some Indonesian proverbs that appear in the search layout and about contains the identity of the author.



Figure 1. The Display of Indonesian Proverbs Layout

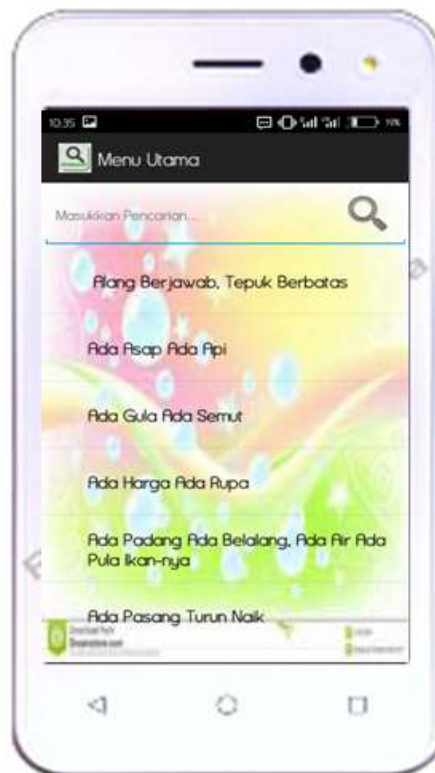


Figure 2. The Display of the First Menu Layout



Figure 3. The Display of Indonesian Proverbs Search Process



Figure 4. The Display of Indonesian Proverbs Meaning Layout

5 CONCLUSION

In the design process and the creation of an Indonesian proverb search application program by implementing this Android-based Boyer Moore Horspool algorithm, there are some conclusions that the author can convey as a result of the evaluation of system development in Cthis thesis. The conclusions are as follows:

1. The process of searching for Indonesian proverbs is needed because of the limitations of smartphone screens, while Indonesian proverbs are sought very much.
2. The Boyer Moore Horspool algorithm is used in the search process for Indonesian proverbs.



3. The Indonesian proverb application has been designed using the Android operating system and uses the Eclipse 4.2 Juno application and can run well on Android 2.2 version of Gingerbread up to the Android 5.1 version of Lollipop.

REFERENCES

- [1] Al-Bahra, Analisis dan Desain Sistem Informasi. Bandung: Informatika, 2005.
- [2] S. P. Ph. D. Roger, Rekayasa Perangkat Lunak. Yogyakarta: Andi, 2012.
- [3] R. Munir, Algoritma dan Pemrograman Dalam Bahasa Pascal dan C. Bandung: Informatika, 2007.
- [4] Kadir, A, Algoritma dan Pemrograman Menggunakan C. Yogyakarta: Andi Offset, 2012.
- [5] Rosa A. S. M. Salahuddin, Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek. Bandung: Informatika, 2013.
- [6] H. N. Safaat, Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android. Bandung: Informatika, 2014.
- [7] M. M. Hilmi, Buku Pintar Android. Jakarta: Elex Media Komputindo, 2015.
- [8] Nugroho, Perancangan dan Implementasi Sistem Basis Data. Yogyakarta: Andi Offset, 2009.
- [9] S. Muhammad, "Implementasi Algoritma String Matching Dalam Pencarian Surah dan Ayat Dalam Al-Quran Berbasis Web,"
- [10] Indonesian Journal on Networking, vol. 6 no 2, no. 2302-5700, p. 70, 2017.