The Application of Text Compression to Short Message Service Using Huffman Table

Ahmad Affandi,[1] Saparudin,[2] and Erwin[3]

***Abstract*— Short Message Service (SMS) is a way of sending short messages in a quick and relatively cheap price. However, besides easiness provided, these SMSs limit the number of characters that can be sent by users. A message sent via SMS, has a maximum capacity of 140 bytes. This causes a person who wants to send a message long enough, consists of a number of characters will have difficulty. Although it is delivered, the message must be assembled into a number of SMS based on maximum capacity. Several ways can be performed to overcome this, one of them by doing compression. By using Huffman table an application text compression on SMS is made in order to compress and decompress when sending and receiving message. This application is made by J2ME and will run on mobile phones based on MIDP 2.0. The results of this study SMS text compression application with Huffman table overall is able to perform the compression process of an SMS text with an average compression ratio of 28.73%.**

***Keywords*— Huffman code, Huffman table, short message service, text compression.**

I. INTRODUCTION

Short Message Service (SMS) is one of the facilities of the mobile phone that has long been used by the public. Use of this SMS allows mobile phone users to exchange information. Through the SMS services, mobile phone users will be able to send short messages in quick time with a relatively cheap price compared with the use of telephone, without being limited by distance and time. So far the SMS become a favorite among mobile phone users. This can be seen from the results of a survey conducted by Nielsen Mobile in the U.S. in second quarter of 2008, show that cell phone subscribers in the United States are using SMS more than a phone conversation.

But besides the eases that has been granted, SMS service limits the number of characters that can be used by users. An SMS message consists of a maximum of 140 bytes, in other words, a message can contain 140 characters of the 8-bit, 160 7-bit characters or 70 characters for 16-bit Japanese, Mandarin and Korean language [1]. In doing sending an SMS a user can send messages over 140 bytes, but for that payment will be more than one. This happens because the message which is sent consists of more than one page and also the process of sending a message will be as many as the number of pages available.

Limitations in the number of characters is what makes the users of SMS to be selective in choosing words so that messages to be delivered is contained entirely within a single SMS. And not only that such limitations also pose a habit to write the message in a way abbreviated which will make of misunderstanding the meaning to receiver. This certainly will be a little inconvenient when users have to perform sending SMS ,so need to be made an application of compression to increase the amount of the character of the SMS that will be sent in single page SMS, by way of compression the text or content of messages on the SMS service.

Compression is the process of encoding data using a smaller number of bits, so that the smaller bits can represent the same information [2]. SMS compression applications is made in order to perform the compression process at the time of sending SMS messages and make the process of decompression when receiving message. The text or data compression will reduce the amount of memory used and accelerate of sending message.

In this study the process of compression and decompression using the Huffman tree will not be performed. This is because the receiver (decoder) will have difficulty to decompress the message if the information did not include the Huffman tree when sending message. Addition information on the Huffman tree will need a separate place so that the process of compression becomes less effective. This is reinforced from the results of research Lipesik Liliana VJ [4], which makes the conclusion that the process is less successful if the file contents of too little data so that the size of original file can be smaller than the size of the compressed file because of file compression results still need to save the Huffman tree.

The compression process will be applied to SMS text that has a small capacity, which is 140 bytes in one page SMS. So the process of compression with Huffman trees become less effective as an information exchange, Huffman tree will be accommodated in a table. This table is called Huffman table, storing the Huffman codes of SMS characters to be used. This table is static and will be used in applications, either the application sender (encoder) or the application receiver (decoder), as a reference in the process of compression and decompression on the SMS text. By using the Huffman table encoding and decoding process can be performed faster and smaller memory requirements than compared with using the Huffman tree [3].

[1]Ahmad Affandi was graduated from bachelor degree in informatics (S.Kom) , Faculty of Computer Science, University of Sriwijaya, Inderalaya, Ogan Ilir, Southern Sumatera.

[2]Saparudin is a senior lecturer in informatics. He is now with the Faculty of Computer Science, University of Sriwijaya, Inderalaya, Ogan Ilir, Southern Sumatera. (e-mail: saparudin@unsri.ac.id).

[3]Erwin is a senior lecturer in informatics. He is now with the Faculty of Computer Science, University of Sriwijaya, Inderalaya, Ogan Ilir, Southern Sumatera. (e-mail: erwin@unsri.ac.id).

II. LITERATURE REVIEW

A. Short Message Service

SMS is one of the facilities of GSM and CDMA standards used for sending and receiving of text messages from a cellular phone. In order to use the SMS feature, users need to complete a cell phone with SIM card (Subscriber Identity Module) of GSM and CDMA service providers that support SMS. An SMS message is not sent directly from mobile phone to the mobile phone receiver but message will be delivered prior to the Short Message Service Center (SMSC). When the destination mobile phone is not active, the system will delay the delivery of messages to the destination mobile phone until mobile phone is active again. In the event of failure of delivery of messages that are temporary (e.g. mobile phone destination is not active) will be resend the message, unless it is enforced the rule that a message has exceeded a certain time limit should be removed and otherwise failed messages.

An SMS message has a maximum length of 140 bytes or 160 characters in the ASCII 7-bit encoding which this format is the standard format used on the SMS [1]. SMS messages in 8-bit encoding has a maximum length of 140 characters and is usually used to send smart message (smart messaging) as a picture or a ringtone and sending data via OTA (over the air) for setting up WAP. Messages are delivered in Arabic, Korean, Chinese or other papers with 16-bit encoding format, then writing an SMS is limited to 70 characters [1].

A standard message containing 160 characters or less will be counted as a single SMS package. For a concatenated message that containing more than 160 characters, each 153 characters will be counted as a single SMS, for seven other characters used as a marker (tag) numbers are part of each section. With the technique of concatenated SMS messaging, although the message contains more than 160 characters, each message sent and equipment consist of 160 characters, only with the marker number of earlier, when an SMS is received by cellular phone that supports concatenated messaging, then some SMS was going directly combined into one long message [5].

B. Huffman Table

In this study the compression that will be applied to the SMS text that has a small capacity of 140 bytes for each page of text. This is consistent with the conclusion that produced on Liliana and Lipesik VJ [4] research and the Huffman tree creation process is removed then this may save time so that the process of encoding process can be performed more quickly. To replace the information on the Huffman tree then created a table (Huffman table), which contains the Huffman codes of the characters default GSM to be used. This table is static and will be used by the application, whether the application sender (encoder) or receiver (decoder), as reference to make the process of encoding / decoding of the text messages.

Huffman table is made by determining the Huffman code from the SMS characters. Determination of Huffman codes are based on assumptions, that is performed by giving a short code for the character that is often accessed so even vice versa. Lowercase that a chance occurrence is more often given Huffman code is shorter, large letters have a smaller chance of occurrence given a longer Huffman code, as well as frequently accessed punctuation such as: Question mark, commas, periods, etc. also given shorter Huffman code, and the last for numbers and other symbols are rare and never even accessed will have a longer Huffman code. With conditions in the determination of the Huffman codes, Huffman code on one of the characters should not be a prefix for the Huffman code on other characters.

In the Huffman tables that used in this research consists of 130 symbols which is the default characters that exist in the GSM 7-bit. Some Huffman table to be used as a reference for compression and decompression in this study can be seen in Table 1. Following:

TABEL I. HUFFMAN TABLE

No.	Huffman Code	Symbol
1	1100	Space
2	1101	a
3	1110	n
4	11110	i
5	10111	e
6	10110	t
7	10101	r
8	10100	k
9	10011	u
10	10010	s
11	10001	m
12	10000	g
13	011110	.
14	011101	?
15	011100	b
16	011011	c
17	011010	d
18	011001	h
19	011000	j
20	010111	l
21	010110	o
22	010101	p
23	010100	y

Huffman table within the program will be formed into two separate shelters, which are formed in the two arrays, namely: an array of letters (containing the characters SMS) and an array of code (containing the Huffman code). For clearer illustration can be seen from this example: *"saya ada di rumah"*

By using Table I, then do the encoding of the message. The first word was "s", first the system will perform a search into the array letters and when the letter "s" was found the next step is to match the number sequence on the array code. The point is that in this case the letter "s" was ranked 10th in the array of letters and then the system will search the sequence is the 10th in an array code and binary code for the letter "s" was found, namely: "10010". So the next encoding process shifted to the next letter until forming a series of

code like as follow:

“10010110101010011011100110101101011011100011010
11110110010101100111000111010110011”

The next step is to calculate the number of characters from the series of bits above. Due to the encoding that was formed in Binary message format (binary message), then such discussion in the previous section automatic calculations performed using 8-bit encoded data. In other words how the calculation is done by dividing the number of bits with a series of eight, so we get the number of characters compression. However, the amount of the above circuit is 81 bits, 7 bits are needed again for the number of bits to be rounded to be divided by eight, if not then the process of calculating the amount of compression character will be difficult to be determined and the system will experience an error in calculating the number of bytes that must be submitted and message cannot be sent. So that made a way in a program that is adding bits "11111111". In this case the required seven bits for eight can be divided so that bits that are added is "1111111" when necessary for example in the case of another three points to be divided by eight then added bits "111". So the end result of a series of bits in this encoding process is as follows:

“10010110101010011011100110101101011011100011010
11110110010101100111000111010110011111111”

And the message is ready to be sent to the recipient's mobile phone number. Once the message is received on a mobile phone receiver, we perform the decoding process (return from a binary message into a text message that can be read by the recipient).

To make the process of decoding the first thing to do is read these bits. Reading of the bits is based on the array code and starts reading from the first bit of the bit "1", then searched and matched to the array code and if not found, we perform the shift by 1 bit, so that it becomes "10" again sought the matching bits of the array code and if not found continue to be a shift, so in this case, bit "10010" is found in the array and the same code as in the encoding process. The next step is the matching number sequence on the array of letters based on the number sequence on the array code; it was found and converted into the character “s”.

Next starts again reading the next bits until everything has been on-decoding everything. In the end the remaining bits in this case is "1111111" because the bits are not located in an array of bits, the code will be given a blank value. Decoding process has been successfully carried out and can be read by the recipient.

C. Sending and Receiving in J2ME

The application of SMS compression in this study is a stand-alone application, which means this application seperates with the existing standard SMS application on mobile phones. Thus, this application must be installed on the sender's mobile phone and receiver’s mobile phone, so that the process of compression and decompression can be performed by both parties.

In the case of sending a message on the MIDP packages to handle SMS is handled by an optional package that is WMA (Wireless Messaging API). At the WMA 2.0, there are three forms of message delivery [6] namely:

1) Binary messages (binary message) that is shaped binary message that sent via SMS. Message of this type uses 8-bit encoded data with a maximum amount of data per SMS pages 140 bytes or 133 bytes if the port number is included.
2) Text message is message in the form of text sent via SMS. In this type of message if the data used in the GSM 7-bit format, the maximum number of characters in one SMS page is as much as 160 characters or 152 characters if the port number is included. If the data used in UCS-2 format, the maximum number of characters is as much as 70 characters or 66 characters if the port number is included.
3) Multipart message is a multi-media message that is sent via MMS.

To receive an SMS message, WMA using the URL "sms: / /" as the SMS protocol identifier. In the WMA format, there are two addresses to send messages [7] namely:

1) Format sms:// <phone number>. SMS messages sent using this format will be immediately catched by inbox and will not be accepted by the WMA application.
2) Format sms:// < phone number >: <Port>. This format allows you to send SMS to other mobile phone applications that enable the WMA as a recipient of an SMS. Port allows for communications between applications WMA. If the port is not included, then the SMS received will go into standard mobile phone inbox so that WMA will never receive the SMS.

In this study, sending and receiving SMS conducted in the form of binary format. The process of compression or decompression of the SMS text is performed by manipulating the bits in the character of the SMS which will be sent based on the Huffman table that contains the binary code for each character that was created earlier.

On the software created with J2ME storage of all information or messages are stored in non-volatile memory (fixed memory) called the Record Management System (RMS). RMS is a storage management system of record that refers to a table with a collection of records. In the RMS does not have primary keys and foreign keys can be defined. Primary key on the RMS has automatically defined as an integer record id.

Record on the RMS is stored as an array of bytes that how it works based on the record (row of data). RMS has the orientation of a simple database record so it does not recognize the fields (columns of data) as a common database, so the data need to be mapped first.

And also J2ME does not provide an API to access the inbox and outbox SMS standard application on mobile phone. Due to the limitations that made an inbox and outbox artificial separated with the inbox and outbox SMS standard application on mobile phone. In this final project, software development, there are three types of SMS message storage in the form of tables in the RMS are:

1) Table inbox (keep messages received that have been decompressed)
2) Table outbox (save the messages sent or outgoing)
3) Table drafts (save the messages are stored before sending)

D. Information Compression

In the discussion in the previous explanation can be seen that the application of compression can compress SMS can either words or messages in order to save memory and load more characters in one SMS page. Therefore in this application will be made a form which will contain information about the results of compression sms, so users can see how much a given compression ratio, the number of characters that can be loaded after compressing, the number of pages in the SMS after compressing etc. So with the info compression facility is expected to be a validation or consideration of the users in sending a message and also useful as additional information for users about the performance of text compression using the Huffman table from the results shown in the form of compression ratio and percentage increase. The compression-info info that is displayed is as follows:

1) The number of initial messages: the number of initial message before compression
2) The number of initial characters: the number of initial characters before compression
3) Number of message compression: the number of messages after compression that will be sent
4) The amount of compression of characters: the number of characters after compression that will be sent
5) Compression ratio: the percentage of SMS compression ratio, can be calculated by equation as follows [8]:

$$\text{Rasio} = \left(1 - \left(\text{compressed\_size} \;/\; \text{raw\_size}\right)\right) * 100\%$$

As the foregoing discussion that the application of this final compression is performed by Binary Message mode, which means messages are sent in the form of binary and message of this type uses 8-bit encoded data with a maximum amount of data per SMS pages of 140 bytes or 133 bytes if the port number included. So the equation to get compressed_size is:

$$\text{compressed\_size} = \text{The number of compressed characters} * 8$$

While for raw_size using Text Message mode, which means messages are sent in text. These messages use the GSM-7 bit format with a maximum number of characters in one SMS page is as much as 160 characters or 152 characters if the port number is included. So the equation is obtained in calculating raw_size are as follows:

$$\text{raw\_size} = \text{The number of initial characters} * 7$$

III. RESULT AND DISCUSSION

Tests conducted with the aim to find out how much influence the compression process to the number of characters and number of pages generated SMS. The testing process was conducted in order to calculate the ratio between the data before it is done with the data compression process after the compression process.

In this case compressed_size represents the number of bits of compression and raw_size character represents the number of bits initial character (before the compressed data). For visual interest the results of testing applications will be presented using the visualization of the emulator. In this test will show information related to the amount of compression SMS initial message, the number of initial characters, the number of message compression, the number of characters compression, the compression ratio and compression results. Testing applications on the SMS text using Huffman table can be seen in the Figure 1 and Figure 2 below.

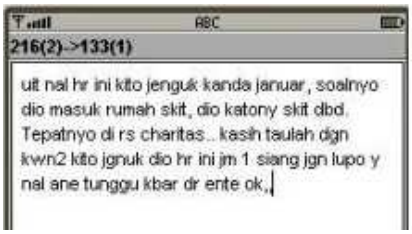

Figure 1. Original Message

In this study the process of compression first will be performed on 15 different pieces lower case letter of SMS, which are consist of 5 pieces of SMS with the number of 1 page, 5 pieces of SMS with the number of 2 pages and 5 pieces of SMS with the number of 3 pages. The language used for writing SMS has the characteristics of many acronyms and many use non-standard language. The results of this test can be seen in the Table II below.

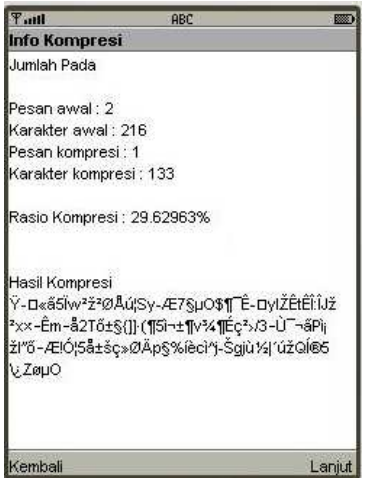

Figure 2. Information Compression

TABEL II. SMS COMPRESSION ON FIRST TEST

No.	The number of initial characters	The number of compressed characters	The number of initial page	The number of compressed page	Raw_size (The number of initial characters * 7) bit	Compressed_size (The number of compressed characters * 8) bit	Compression ration (%)
1	118	74	1	1	826 bit	592 bit	28,33 %
2	44	27	1	1	308 bit	216 bit	29,87 %
3	61	39	1	1	427 bit	312 bit	26,93 %
4	131	83	1	1	917 bit	664 bit	27,59 %
5	78	50	1	1	546 bit	400 bit	26,74 %
6	194	122	2	1	1358 bit	976 bit	28,13 %
7	202	126	2	1	1414 bit	1008 bit	28,71 %
8	216	133	2	1	1512 bit	1064 bit	29,63 %
9	213	131	2	1	1491 bit	1048 bit	29,71 %
10	197	122	2	1	1379 bit	976 bit	29,22 %
11	338	211	3	2	2366 bit	1688 bit	28,86 %
12	401	254	3	2	2807 bit	2032 bit	27,61 %
13	384	237	3	2	2688 bit	1896 bit	29,46 %
14	345	211	3	2	2478 bit	1688 bit	30,10 %
15	415	254	3	2	2905 bit	2032 bit	30,05 %

Bases on the result of system testing SMS compression on Table 4.14 can be concluded that compression ratio averages is 28,73%, for more explanation the result of compression can be looked on Figure 3 below.

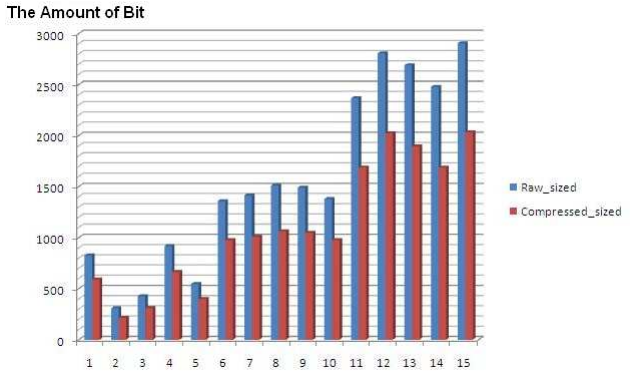

Figure 3. The first compression test based on raw_size

Based on Figure 3 the test results of compression system using Raw_sized SMS (SMS initial number of bits) ranging between 308 bits to 917 bits, in this case indicates that although the compression continues to be one page, but the results Compressed_sized (number of bits SMS compression) becomes lower ie ranging between 216 bits to 664 bits.

While the testing of compression systems using Raw_sized SMS (SMS initial number of bits) ranging from 1358 bit to 1512 bit capable of producing Compressed_sized (number of bits SMS compression), ie ranging between 976 bits to 1064 bits, so that in this test with the number of initial pages to 2 pages compressed into 1 pages.

And for testing the compression system using Raw_sized SMS (SMS initial number of bits) ranging from 2366 bit to 2905 bit capable of producing Compressed_sized (number of bits SMS compression), ie ranging from 1688 bits to 2032 bits, so that in this test with the number of initial pages 3 pages can be compressed into 2 pages.

SMS compression system further testing will be conducted on 5 pieces of data an SMS with the text type which consists of all upper case, which consists of 2 pieces of SMS with the number of pages 1, 2 SMS with the number of pages 2 and 2 pieces of SMS with the number of page 3. To more clearly seen from the Table III below.

TABEL III. SMS COMPRESSION ON SECOND TEST

No.	The number of initial characters	The number of compressed characters	The number of initial page	The number of compressed page	Raw_size (The number of initial characters * 7) bit	Compressed_size (The number of compressed characters * 8) bit	Compression ration (%)
1	114	98	1	1	798 bit	784 bit	1,75 %
2	177	152	2	2	1239 bit	1216 bit	1,86 %
3	222	190	2	2	1554 bit	1330 bit	2,19 %
4	306	265	3	3	2142 bit	2120 bit	1,03 %
5	419	370	3	3	2933 bit	2960 bit	-0,92 %

Based on the results of the testing system for data compression SMS SMS is composed of capital letters in Table 4.15 to conclude that the average compression ratio that is equal to 1.18% for more details, the result of compression can be seen in Figure 4 below.

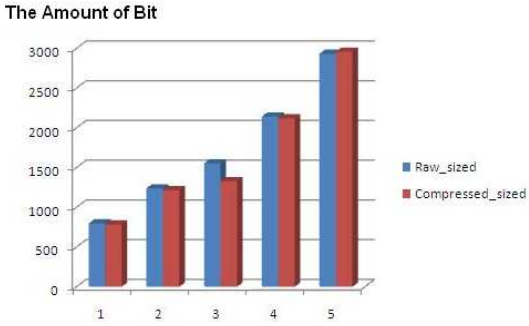

Figure 4. The second compression test based on raw_size

Based on Figure 4 the test results of compression system using Raw_sized SMS (SMS initial number of bits) in the SMS data consisting of all uppercase ranging between 798 bits to 2933 bits, and gives results in Compressed_sized (number of bits SMS compression), ie, ranging between 784 bits up to 2960 bits, shows that the difference between Raw_sized and Compressed_sized is very small. In this test can be proved that the pages aren't compressed which consisting of 2 pages and 3 pages. It can be concluded that the SMS data consisting of all uppercase less effective for compressed.

IV. CONCLUSION

Huffman table can be used in the compression text on SMS services. Generally Huffman table has an average compression ratio of 28.73 %. Overall SMS compression testing system by using the Huffman table will produce good compression if the compressed data to be composed of the characters with a shorter length code (lower case letter) contained in the Huffman table, and vice versa if the SMS data consists of characters which have a longer code (data consisting of all upper case) contained in the table Huffman compression results become less effective.

REFERENCES

[1] Aprianto, D. 2007. “Performansi Modifikasi LZW (Lempel-Ziv-Welch) Untuk Kompresi Teks.” [unpublished thesis], Department of Informatics, Bandung Institute of Technology.

[2] Ayuningtyas, N. 2008. “Implementasi Kode Huffman Dalam Kompresi Teks.” [unpublished thesis], Department of Informatics, Bandung Institute of Technology.

[3] Hashemian, R. 2005. “Direct Huffman Code and Decoding Using The Table of Code-Lengths.” [unpublished thesis], Northen Illionis University.

[4] Liliana, Lipesik, V.J. 2006. “Pembuatan Perangkat Lunak Untuk Kompresi File Text Dengan Menggunakan Huffman Tree.” [unpublished thesis], Faculty of Industry Technology, Kristen Petra University.

[5] JSR 120 Expert Group. 2002. *Wireless Messaging API for Java™ 2 Micro Edition*. Sun Microsystem Inc.

[6] Ortiz, C.E. 2005. *The Wireless Messaging API, Sun Microsystem*, Inc. [online]Available: http://developers.sun.com/mobility/ midp/articles/ wma2/

[7] Mardiono, T. 2006. *Membangun Solusi Mobile Business Dengan Java*. Elex Media Komputindo, Jakarta.

[8] Nelson, M., Jean, L.G., 1995, *The Data Compression Book Second Edition*, IDG Books Worldwide, Inc, Cambridge.