Short Message Security System for Android-Based Mobile Phone Using Hill Cipher and Arithmetic Coding Algorithm

Tita Karlita, Isbat Uzzin Nadhori, Rani Dewi Ismawati
Informatics Engineering Department
Politeknik Elektronika Negeri Surabaya
pens.ac.id, rani_ismawati@ymail.com

***Abstract*— Since mobile phones have become a must-have item, sending messages using text-based messaging or Short Message Service (SMS) through mobile phones becomes something that is almost done every day. A short message is one form of communication services that are already widely used and widely known at this time because it is easy to use and cheap. Utilization of text characters as the media, not only provides ease of use, but also can cause delivery problems of the security of the message content. Another issue is due to the short message using a universal coding standard. Short message has been sent can be captured by the operator or by third parties using a tool that is not expensive. In the message contains the data sender and the recipient's phone number and the content of the message itself. So that short messages can be easily replaced and falsified by third parties. This triggered concerns because sometimes a short message is used to exchange confidential messages. To overcome these problems, then we made a short message security system for Android mobile phone using Hill Cipher algorithm for doing encryption and Arithmetic Coding algorithms for compression processes.**

Keywords*— *encryption, decryption, sms, android

I. INTRODUCTION

Today mobile phones have become a necessity for everyone. SMS (Short Message Service) is one of the favorite facility in mobile phone wich is rapid, easy use and cheap. SMS provides ease communication using text, so it is possible to intercepted by others. To overcome these problems required additional security on the SMS system. SMS service solutions above is using cryptographic methods. Cryptography is the art and science to secure a message (plaintext) into a hidden message (ciphertext). In this research created a security system short message or SMS-based android using encryption methods. Usually, the encryption method produces a larger character than the original message. in In this paper, we propose an secure encryption method and combine compression algorithm that produces secure message and the size of the data encrypted and decrypted not increased.

Previous research, Hill Cipher Cryptographic has been done Todd Douglas and Dustin Helliwell (1997) [8]. This research describes the encryption and decryption process using a 2x2 matrix. SMS encryption using AES algorithm has been done by Rohan Rayarikar [9]. The results showed no delay in the process of sending the message. Sri Rangarajan also made a SMS security system using symmetric key algorithm called Elliptic Curve Cryptograpgy (ECC). The main requirement ECC system is the sender and the recipient must be active at the same time and each should install ECC application. Tarek M. Mahmoud has also made security system combined with compression of SMS on the Symbian OS using the RSA algorithm [11]. Tarek research results show that the size of the data encrypted and decrypted not increased. At this research explains the process of encryption and decryption to 26 characters using a 2x2 matrix by Todd Douglas and Dustin Helliwell (1997) [8] combined with Aritmetic Coding compression algorithm by Maya Basoeki (2008) [6] which is implemented on Mobile Phone android

At this research, we propose a compression and decryption methods to produce secure message on the android mobile device. By using these methods, the size of the encrypted and decrypted message have not increased. The encryption process used Hill Cipher algorithms. Here we used two types of metrics, namely 2x2 and 3x3. And we used 95 characters for sending messages. For the compression process, we used Arithmetic Coding algorithms.

II. HILL CIPHER ALGORITHM

Hill Cipher is a cryptographic algorithm that implements modulo arithmetic. This cryptographic technique using a square matrix as a key used in the encryption and decryption process. Hill Cipher invented by Lester S. Hill in 1929. Hill Cipher not substitution method like other classical cryptography method, but using the matrix multiplication as the basis for encryption and decryption process. On the Hill Cipher, Plaintext divided into blocks of a certain size. Each character in a block of plaintext will affect the other characters in the process of encryption and decryption, so the same plaintext character will not produce the same character in the ciphertext, and produce same size between plaintext and chipertext

Basic Hill Cipher Methods is modulo matrix. In the implementation, Hill Cipher Method using matrix

multiplication and the matrix inverse. Hill Cipher key is nxn matrix where n is the block size. If n = 2, then the encryption is done every two characters. Matrix K as the key must be an invertible matrix, which has an inverse K-1, so that:

$$K.K^{-1} = I \quad (1)$$

The Key must have an inverse, because the matrix K-1 is a key used to decrypt it

A. *Hill Cipher Encryption Algorithm*

The stages of Encryption Hill Cipher is as follows:

1. Create a matrix K as the key size NxN

$$K_{nxn} = \begin{pmatrix} k_{11} & k_{12} & .... & k_{1n} \\ k_{21} & k_{22} & .... & k_{2n} \\ .... & .... & .... & .... \\ k_{n1} & k_{n2} & .... & k_{nn} \end{pmatrix} \quad (2)$$

2. Substitute alphabet with numerics code

   A → 1 , B → 2, C → 3, ……., Z → 0

3. Categorize numbers obtained into several blocks vector P whose length is equal to the size of the matrix "K".
4. Calculate Ciphertext (modulo 26) for each vector P

$$C = K \cdot P \pmod{26} \quad (3)$$

   C = Ciphertext, K = Key Matrix, and P = Plaintext
5. Restore each number in a vector C to the letter using the phase 2 to get the ciphertext

B. *Hill Cipher Decryption Algorithm*

The stages of decryption Hill Cipher is as follows:

1. Substitute alphabet with numerics code

   A → 1 , B → 2, C → 3, ……., Z → 0

2. The key used to decrypt the ciphertext into plaintext is the inverse matrix K_{nxn}
3. Calculate K^{-1} (invers) :

$$K\text{-}1 = \frac{1}{\text{Det } K} adj(K) \quad (4)$$

4. Calculate plaintext :

$$P = K^{-1} \cdot C \quad (5)$$

5. Restore each number in a vector P to the letter using the phase 2 to get the plaintext

III. ARITHMETIC CODING ALGORITHM

A. *Data Compression Algorithm*

Generally, data compression algorithms perform the replacement of one or more input symbols with a specific code. Contrast to these methods, Arithmetic Coding compression algorithm replaces one row of input symbols with a floating point numbers. The longer and more complex coded message, the more bits are required for this purpose. The process of converting a String Message to be Code Number. The output of the compression algorithm Arithmetic Coding is a number less than 1 and greater than or equal to 0

To produce these outputs, each symbol will be encoded (compressed) given a set of probability values. Example "DEA" will be encoded (compressed), and has defined the probability of a character, the possibility of appearing in a message is as follows.

After the probability of each character is known, the character will be given a certain range with a value between 0 and 1, appropriate with the existing probabilities. Based on Table 2.1, Table 2.2 is formed for the word "DEA".

Next, do the compression (encoding) using the following algorithm:

```
Set low = 0.0
Set high = 1.0
While (input symbol is still there) do
Take the symbol input
CR = high – low
High = low + CR*high_range (simbol)
Low = low + CR*low_range (simbol)
End While
Print (Low+High)/2
("Low" is output process Aritmetic algorithm Coding)
```

TABLE I. EXAMPLE PROBABILITY TABLE

Character	Probability
A	2/5
B	2/5
C	2/5
D	2/5
E	2/5

TABLE II. WORDS RANGE SIMBOL

Character	Probabilitas	Range
A	2/5	0.0-0.2
B	2/5	0.2-0.4
C	2/5	0.5-0.6
D	2/5	0.6-0.8
E	2/5	0.8-1.0

The principle of the algorithm is as follows:

1. Set initial value low and high low = 0 dan high = 1
2. Find the first character current_range= high –low
3. Set the new value high dan low, using formula:

   high = low + current_range x high_range (the first caracter)
   low = low + current_range x low_range (the first caracter)

4. Repeat 2-3 until the last character

After last character, you will get a range of new highs and lows. For the word "DEA", first we take the character "D". CR value is 1-0 = 1. High_range ('D') = 0.8, Low_range ('D') = 0.6. results :

high = 0.00 + 1.0*0.8 =0.8
low = 0.00 + 1.0*0.6 = 0.6

Then take characters “E”.
CR value is 0.8 – 0.6 = 0.2.
High_range(‘E’)= 1.0, Low_range (‘E’) = 0.8.
Results:
high = 0.6 + (0.2*1.0) = 0.8
low = 0.6 + (0.2*0.4) = 0.76

Then take characters “A”.
CR value is 0.8 – 0.76 = 0.04
High_range(‘A’)= 0.2, Low_range (‘A’) = 0.0.
Results:
high = 0.76 + (0.04*0.2) = 0.768
low = 0.76 + (0.04*0.0) = 0.76

This process obtained the value of output = (0.7600 + 0768 -) / 2 = 0764. This value is transmitted to bring the message "DEA"

B. Decompression Data Algorithm

Decompression is a process for changing the compression results into original form. A process for changing a code to Message (Langdon, Jr., 1984). The working principle arithmetic decoding algorithm is as follows:

1. Based on the code number is obtained, it can be determined the first character of the Message that was sent. By looking at the probability table. See the code number and range of characters generated

TABLE III. ENCODING PROCESS (COMPRESSION) FOR THE WORD "DEA"

Character	LOW	HIGH	CR
	0.0	1.0	1.0
D	0.6	0.8	0.2
E	0.76	0.8	0.04
A	0.76	0.768	

2. If it has been know the character first appeared, then the next is find code_range using the formula:
   code_range = high_range symbol - low_range symbol
3. Get the new value encoded_number using the formula:
   encoded_number = (high_range symbol - low_range symbol)/ code_range
4. Based on encoded_number determine the next character. Repeat step 2 until the process is complete.

Decompression process words "DEA", can be started from the determination of the value of the initial range of encryption results.

1. Input: 0.7640 (This value is in the range of 0.6-08, so that the resulting character is "D")
2. CR = high_range (“D”) = low_range (“D”)
   = 0.8 -0.6 =0.2
   encoded_numer = (0.764-0.6)/0.2 = 0.82 (“character “E”)
3. CR = high_range (“E”) = low_range (“E”)
   = 1.0 -0.8 =0.2
   encoded_numer = (0.82-0.8)/0.2 = 0.2 (“character “A”)

So from 0764 input values can be generated message "DEA".

IV. RESEARCH METHODS

In this research, we develop secure SMS service using encryption and compression method. Hill Cipher algorithm used for the encryption and Arithmetic Coding algorithms used for compression.

A. Blok Diagram

The Security Message System Block Diagram as shown in Figure 1. Using this application The sender can perform encryption or compression process messages and the recipient using this appplications for decompression and or decryption a received message.

Step application usage is as follows: Sender wrote the message using New Message menu, finish writing a message, press the button to send the message, before sending the message, the application will provide a menu selection encryption and/or compression.

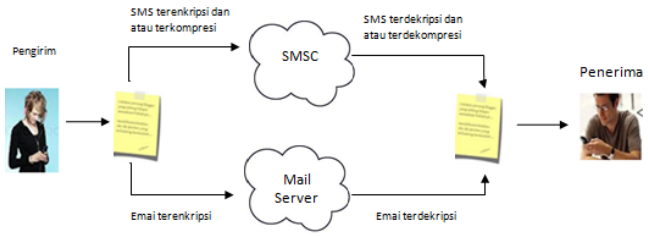

Fig. 1. Security Message System Block Diagram

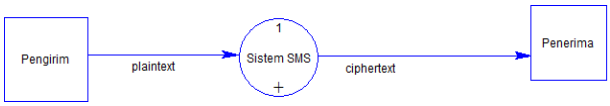

Fig. 2. Contex Diagram (DFD level 0)

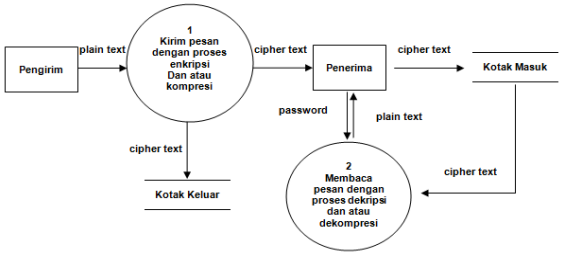

Fig. 3. Data Flow Diagram (DFD level 1)

B. Contex Diagram

Level 0 Context Diagram as shown in Figure 2. The sender sending a text message (plaintext), then sms system do the encryption and or compression process, the results are sent to the recipient.

Level 1 context diagram as shown in Figure 3. The sender sent an SMS, then sms system do the encryption and or compression process, results chipertext sent to the recipient. Recipient receive in the "inbox". To be able to read the message, the recipient must enter the password, and the next sms system do the decryption and decompression process to generate original message.

C. Encrytion and Decryption Message

The encryption process using Hill Cipher. Key matrix used in this application is the 2x2 and 3x3 matrix that has been set in the application as a static key. When the length of a message is a multiple of 2, then the dimension of the key matrix is used appropriate with the length of the message. It means the message will be divided every two characters to be processed with the key matrix. However, when the length of the message is not a multiple of 2, then there will be one character left and did not participate in the processing of the key matrix. The last character will be delivered according to the original character. Matrix values must be positive, if the value of index matrix is negative, then search modulo results. The value of Modulo depends on the number of characters available.

In this application, SMS character divided into 3 types, adapted to the character on the android keyboard. letter1 there are 86 characters, where the first character will represent one character sms. letter2 there are 94 characters consisting of the characters that exist in letter1 and characters which one character represents two characters in the text. letter3 a combination of letter1 and letter2 plus 1 character (`) which represent 16 characters sms.

The final value matrix depend on the number of available characters. Because there are three types of initialization, then the matrix which has a negative index would modulo with three different lengths and each length will be defined as a letter code encryption will affect the decryption process, as follows:

- Length Letter1 = 86 (letter code=1)
- Length Letter1 = 94 (letter code=2)
- Length Letter1 = 95 (letter code=3)

Decryption is done by changing the encryption key matrix into the matrix inverse.

D. Compression dan Decompression Message

Compression done with create dictionary table containing the value range of each character that relies on a probability value of each character. Value range is inserted into the body text as a reference to the decompression process. Range high and low for each character entered into a database which will be called back when the calculation process.

Initials range value each character is taken from the difference in value between the value on the character and value of the previous character. An example is the range for the character "a" or kar [97]:

```
kar[96]="0.3385";
kar[97]="0.4272";

range_high ("a")= "0.4272";
range_low ("a") ="0.3385";
```

Compression process begins by dividing the message into several parts (N). Where N is the value segment, as shown in Figure 4.

Value 'N' is the number 10,15, and 20 are used as the value segment. the message will be broken down by multiples of N. Whereas 'n' is the number of characters to be compressed. 'x' is the number of code number compression results obtained from x = n / N. If n / N is not equal to zero then x will be x + 1.

Decompression process also created initialization value for each character. Where this values are adjusted from the value of each character in the compression process. It's show the Arithmetic Coding algorithm is static algorithm.

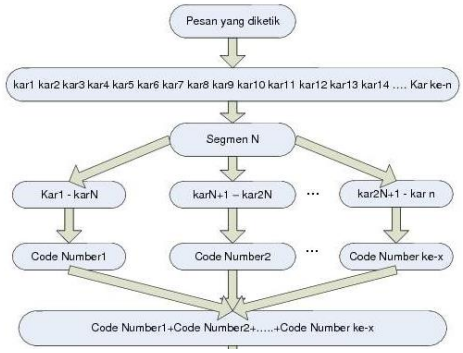

Fig. 4. Message processing scheme in the process of compression

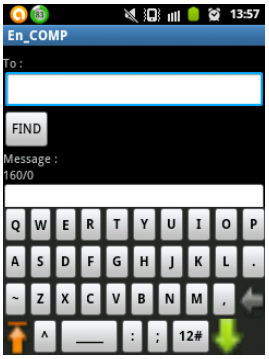

Fig. 5. New Message Menu

TABLE IV. TESTING ENCRYPTION PROCESS (CHARACTER VARIATION)

VARIASI KATA	JUM. KAR	JUM. HAL	ENKRIPSI (RK)		DEKRIPSI (RT)	
			RASIO%	JUM.HAL	RASIO%	JUM.HAL
HURUF NON KAPITAL	90	1	102.2222	1	100	1
	276	2	100.7246	2	100	2
	826	6	100.2421	6	100	6
HURUF KAPITAL	90	1	102.2222	1	100	1
	276	2	100.7246	2	100	2
	826	6	100.2421	6	100	6
GABUNGAN	90	1	102.2222	1	100	1
	276	2	100.7246	3	100	2
	826	6	100.2421	6	100	6

V. RESULT

The purpose of this application testing is to determine success rate and responsibilityy applications. Experiments were done using Android Mobile Phone to test the results of sending and receiving messages.

In this application, writing messages use a special keypad as shown in Figure 5, because there are some characters in the original keypad android can not be processed by this application

There are two types of experiments: based on the variation of characters and based on the length of the message. The purpose of the experiment as follows:

- To Know the effect of variations in the character againts of ratio rate for each process of sending and receiving message
- To Know the effect of the length of the message againts of the ratio rate for each process sending and receiving message.

Delivery Ratio (RK) is the ratio of the number of characters successfully sent divided by the number of characters written. Information on the number of characters written and were successfully sent can be found in the menu Information message. The smaller the acceptance ratio it's better, it's mean characters successfully sent almost same as the number of characters plain text/original character.

Acceptance Ratio (RT) is the ratio of the number of correct characters (the result of decryption characters is equal to written character) divided by the total number of sent characters.

Calculation of The Delivery ratio and Acceptance ratio in% is as follows:

$$RK : \frac{\text{The number of sent characters}}{\text{The number of written character}} \times 100\% \quad (6)$$

$$RT: \frac{\text{The number of correct character}}{\text{Total number of characters the message}} \times 100\% \quad (7)$$

A. *Testing Based on Variation Characters*

In the testing process, the variation of characters divided into three groups: lower-case characters (non-capital), capital letters, and a combination of letters (capital and non-capital), numbers, and symbols. This is to determine the effect of different types of characters in the result of sending and receiving messages in each message delivery process. Experiment focused on variations of words / characters are written and the results of its acceptance ratio.

1) Encryption Process (Based on Character Variation)

Results of testing the encryption process based on the variation of characters are shown in Table 1. Result shows that the ratio of encryption has increased over 100%. It's means that the total length of characters that is sent greater than the number of characters that are written by the sender. But Variations letters (capital or non-capital) has no effect because all of the letters represent one character. No additional value to each character.

Contrast in "GABUNGAN", characters consists of variations of special characters. Results showed, there were differences encryption and decryption results. Because there are some characters symbols that represent more than one character. For example quote characters (") or ([). Where one character has a bit length such as 2 characters.

Table 2 shown the results of Delivery Ratio (RK) and Acceptance Ratio (RT) in each variation of the word. For all variations of the word it produces the same RK is 101.063%. For the best variation RK result on variations non-capital letters. Where has the largest acceptance ratio, ie 19.74289301%. From three types of variations of the word, all RT produces a value of 100%. It means that messages sent and received are the same.

TABLE V. AVERAGE RATIO OF TESTING VARIATION WORD

VARIASI KATA	HURUF NON KAPITAL (%)	HURUF KAPITAL (%)	GABUNGAN (%)
RATA-RATA RASIO PENGIRIMAN (RK)(JUML. KAR)	101,0630	101,0630	101,0630
RATA-RATA RASIO PENERIMAAN (RT)(JUML. KAR)	100	100	100

TABLE VI. TESTING COMPRESSION PROCESS SEGMENT = 10, 15, AND 20

VARIASI KATA	JUM. KAR	JUM. HAL	KOMPRESI (RK) Segmen=10		DEKOMPRESI (RT) Segmen=10		KOMPRESI (RK) Segmen=15		DEKOMPRESI (RT) Segmen=15		KOMPRESI (RK) Segmen=20		DEKOMPRESI (RT) Segmen=20	
			RASIO %	JUM. HAL	RASIO%	JUM.HAL	RASIO %	JUM.HAL	RASIO%	JUM.HAL	RASIO %	JUM.HAL	RASIO%	JUM.HAL
HURUF NON KAPITAL	90	1	67,77778	1	16,66667	1	47,77778	1	15,55556	1	41,11111	1	14,44444	1
	276	2	63,4058	2	19,2029	2	43,84058	1	14,13043	2	32,97101	1	14,49275	2
	826	6	61,13801	4	17,67554	6	41,52542	3	15,61743	6	31,35593	2	12,83293	6
			Rata-rata		17,8484		Rata-rata		15,1011		Rata-rata		13,9234	
HURUF KAPITAL	90	1	67,77778	1	16,66667	1	47,77778	1	12,22222	1	41,11111	1	12,22222	1
	276	2	63,4058	2	19,2029	2	43,84058	1	14,49275	2	32,97101	1	10,86957	2
	826	6	61,13801	4	17,67554	6	41,52542	3	11,50121	6	31,35593	2	9,443099	6
			Rata-rata		15,3496		Rata-rata		12,7387		Rata-rata		10,845	
GABUNG AN	90	1	67,77778	1	16,66667	1	47,77778	1	15,55556	1	41,11111	1	8,888889	1
	276	2	63,4058	2	19,2029	2	43,84058	1	14,49275	2	32,97101	1	14,85507	2
	826	6	61,13801	4	17,67554	6	41,52542	3	14,89104	6	31,35593	2	12,2276	6
			Rata-rata		16,8335		Rata-rata		14,9798		Rata-rata		11,9905	

2) Compression Process (Based on character Variation)

The compression process results based on character variation shown in Table 3. The smallest average value of truth (the acceptance ratio/ decompression) on compression process for the segment = 10, 15, and 20 is in the variation of the word capital letters. The best acceptance ratio is a variation on the word non-capital letters.

3) Compression-Encryption Process (Based on Character Variation)

This experiment started with compression process and then the results will be encrypted. Experiment Results shown in Table 4. The result is almost the same as in experiment b (compression process- *Based on character Variation*). The best RT still dominated by a variation of non-capital letters. But in this process there is a difference in the number of pages on the sending message. When the number of characters is 826 which generates 6 page message, after this process, the number of pages increases, because specially in this encryption process using matrix on the letter 3 (len = 95).

Seen on the average value of the ratio of shipments, value for segment 10 is equal to the ratio value of the delivery segment 10 in the compression process, because the ratio of acceptance process includes a compression process, will depend on the value of a given segment. For segment 20, the percentage of truth is smaller than on segment 10 and 15. It can be concluded that compression with a larger segment will produce less accurate compare with using a smaller segment.

TABLE VII. TESTING COMPRESSION-ENCRYPTION PROSES SEGMEN=10, 15, AND20

VARIASI KATA	JUM. KAR	JUM. HAL	KOMPRESI-ENKRIPSI Segmen=10		DEKRIPSI-DEKOMPRESI Segmen=10		KOMPRESI-ENKRIPSI Segmen=15		DEKRIPSI-DEKOMPRESI Segmen=15		KOMPRESI-ENKRIPSI Segmen=15		DEKRIPSI-DEKOMPRESI Segmen=15	
			RASIO %	JUM. HAL	RASIO%	JUM. HAL	RASIO %	JUM. HAL	RASIO%	JUM. HAL	RASIO %	JUM .HAL	RASIO%	JUM. HAL
HURUF NON KAPITAL	90	1	67,7778	1	16,66667	1	67,77778	1	15,55556	1	67,7778	1	14,44444	1
	276	2	63,4058	2	19,2029	2	63,4058	2	14,13043	2	63,4058	2	14,49275	2
	826	6	61,138	8	17,67554	6	61,13801	6	15,61743	6	61,138	4	12,83293	6
			Rata-rata		17,8484		Rata-rata		15,1011		Rata-rata		13,9234	
HURUF KAPITAL	90	1	67,7778	1	16,66667	1	67,77778	1	12,22222	1	67,7778	1	12,22222	1
	276	2	63,4058	2	15,21739	2	63,4058	2	14,49275	2	63,4058	2	10,86957	2
	826	6	61,138	8	14,16465	6	61,13801	6	11,50121	6	61,138	4	9,443099	6
			Rata-rata		15,3496		Rata-rata		12,7387		Rata-rata		10,845	
GABUNG AN	90	1	67,7778	1	15,55556	1	67,77778	1	15,55556	1	67,7778	1	8,888889	1
	276	2	63,4058	2	17,75362	2	63,4058	2	14,49275	2	63,4058	2	14,85507	2
	826	6	61,138	8	17,19128	6	61,13801	6	14,89104	6	61,138	4	12,2276	6
			Rata-rata		16,8335		Rata-rata		14,9798		Rata-rata		11,9905	

TABLE VIII. TESTING ENCRYPTION-COMPRESSION PROCESS SEGMEN=10, 15, DAN 20

VARIASI KATA	JUM. KAR	JUM. HAL	ENKRIPSI-KOMPRESI Segmen=10		DEKOMPRESI DEKRIPSI Segmen=10		ENKRIPSI-KOMPRESI Segmen=15		DEKOMPRESI-DEKRIPSI Segmen=15		ENKRIPSI-KOMPRESI Segmen=15		DEKOMPRESI-DEKRIPSI Segmen=15	
			RASIO %	JUM .HAL	RASIO %	JUM. HAL	RASIO %	JUM. HAL	RASIO %	JUM. HAL	RASIO %	JUM. HAL	RASIO %	JUM. HAL
HURUF NON KAPITAL	90	1	67,778	1	0	1	47,778	1	1,1111	1	41,111	1	1,1111	1
	276	2	63,406	2	1,4493	2	43,841	1	1,4493	2	32,971	1	1,4493	2
	826	6	61,138	4	1,6949	6	41,525	3	1,4528	6	31,356	2	1,3317	6
			Rata-rata		1,048		Rata-rata		1,338		Rata-rata		1,297	
HURUF KAPITAL	90	1	67,778	1	0	1	47,778	1	1,1111	1	41,111	1	1,1111	1
	276	2	63,406	2	1,8116	2	43,841	1	1,8116	2	32,971	1	1,087	2
	826	6	61,138	4	1,3317	6	41,525	3	0,8475	6	31,356	2	1,3317	6
			Rata-rata		1,048		Rata-rata		1,257		Rata-rata		1,177	
GABUNG AN	90	1	67,778	1	2,2222	1	47,778	1	1,1111	1	41,111	1	1,1111	1
	276	2	63,406	2	1,4493	2	43,841	1	1,4493	2	32,971	1	1,4493	2
	826	6	61,138	4	1,5739	6	41,525	3	1,3317	6	31,356	2	1,5739	6
			Rata-rata		1,748		Rata-rata		1,297		Rata-rata		1,378	

4) Encryption-Compression Process (Based on Variation Character)

This experiment started with encryption process and then the results will be compressed. Experiment results shown in the table 5. Experiment result based on word variation produce small rate rasio. It's happen because the process begins with decompression, where ratio rate also low, as shown in Table 3. So that the decryption process, which a follow of the decompression process is also affected. Because decryption process input derived from decompression process output.

Table 6 shown ratio rate sending and receiving on variation character. The best rate on variation non-capital character, gets the value of 19.74289301%. 19.74289301%. It's happen because non-capital letters have greater probability than the word variations on capital letters and combination of letters, numbers, and symbols.

Sending ratio, each variation of the word have the same probability. It's happen because the sending ratio is dependent

on the number of segments given (to process involving the compression process).

TABLE IX. AVERAGE RATIO ON VARIATION CHARACTER TEST

VARIASI KARAKTER	HURUF NON KAPITAL(%)	HURUF KAPITAL(%)	GABUNGAN (%)
RASIO PENGIRIMAN RATA-RATA	61,17	61,17	61,17
RASIO PENERIMAAN RATA-RATA	19,74	18,14	19,2

TABLE X. TESTING ENCRYPTION PROCESS (CHARACTER LENGTH)

Panjang Pesan yang ditulis	Pesan yang dikirimkan	Jumlah Halaman	Panjang Pesan yang diterima	Rasio Pengiriman (Enkripsi) (%)	Rasio Penerimaan (Dekripsi) (%)
12	14	1	12	116,6667	100
78	80	1	78	102,5641	100
150	152	2	150	101,3333	100
224	226	2	224	100,8929	100
346	348	3	346	100,578	100
383	385	3	383	100,5222	100
401	403	3	401	100,4988	100
463	465	4	463	100,432	100
511	513	4	511	100,3914	100
827	829	6	827	100,2418	100

B. Testing Based on Long Character

This experiment was conducted testing based on length of sending messages with the encryption process.

1) Encryption Process (Based on Character Length)

The result of encryption process shown in tabel 7. This application uses two types of matrices, 2x2 and 3x3 dimension. If the message length is divisible by 3 or the result modulo is 1 character, used 3x3 matrix. If the message length is not divisible by 3 or modulo result is more than 1 character, used 2x2 matrix.

On the encryption process, the length of messages sent over 2 characters than the number of characters written. Two additional character is a character code used in the encryption process. The first character is the code for the type of process, an example of the encryption process coded 2, so that the receiving application can determine what process has been used by the sender. While the second character is the character for the type of letter is used, the code 1 for the letter 1, 2 letter code 2, and 3 letter code 3 as described in the previous chapter.

Based on the experiment, it can be concluded that the encryption process has the message acceptance ratio of 100%. Written messages received 100%. There are two additional characters as code application system.

2) Compression Process (based on Character Length)

Experiment result depends on the segment value. Given segment value 10, 15, and 20. The results as shown in Table 8.

TABEL 8 EXPERIMENT RESULT OF COMPRESSION PROCESS ON SEGMEN 10, 15, AND 20 (BASED ON CHARACTER LENGTH)

JUM. KAR	JUM. HAL	Rasio Pengiriman (Kompresi) (%)						Rasio Penerimaan (Dekompresi) (%)		
		10	JUM. HAL	15	JUM. HAL	20	JUM. HAL	10	15	20
18	1	105.56	1	105.56	1	_	_	22.22	16.67	_
54	1	79.63	1	57.41	1	46.30	1	14.81	12.96	12.96
136	1	66.91	1	49.26	1	36.03	1	19.85	17.65	18.38
219	2	63.47	1	44.29	1	33.33	1	17.81	16.89	14.16
277	2	63.18	2	43.68	1	32.85	1	19.49	16.61	10.47
386	3	62.44	2	42.23	2	32.90	1	18.39	15.03	13.21
470	4	61.49	2	42.34	2	32.13	1	20.21	15.74	14.89
742	5	61.59	3	41.37	3	31.67	2	16.71	13.48	13.48
826	6	61.14	4	41.53	3	31.36	2	17.68	15.62	12.83
Rata-rata(%)		70.53		53.27		35.03		18.69	15.63	13.94

TABLE XI. EXPERIMENT RESULT COMREPSSION-ENCRYPTION PROCESS ON SEGMENT 10, 15, AND 20 (BASED ON CHARACTER LENGTH)

JUM. KAR	JUM. HAL	Rasio Kompresi-Enkripsi (%)						Rasio Dekripsi-Dekompresi (%)		
		10	JUM. HAL	15	JUM. HAL	20	JUM. HAL	10	15	20
18	1	105.56	1	105.56	1	_	_	22.22	16.67	_
54	1	79.63	1	57.41	1	46.30	1	24.07	11.11	16.67
136	1	66.91	2	49.26	1	36.03	1	21.32	15.44	12.50
219	2	63.47	3	44.29	2	33.33	2	17.81	14.16	10.96
277	2	63.18	3	43.68	2	32.85	2	18.41	13.00	15.88
386	3	62.44	4	42.23	3	32.90	2	18.39	15.03	13.21
470	4	61.49	5	42.34	3	32.13	3	20.21	15.74	14.89
742	5	61.59	7	41.37	5	31.67	4	16.71	13.48	13.48
826	6	61.14	8	41.53	6	31.36	4	17.68	15.62	12.83
Rata-rata (%)		70.53		53.27		35.03		19.89	14.33	13.94

Based on experiment result can be conclude, the result is the lowest delivery ratio sequence segment 20, segment 15 and segment 10. average of the best compression results achieved in the segment 20. In addition, the ratio of message delivery between segments 15 and 20 is smaller than the ratio of message delivery between segments 10 and 15. It will also affect the difference in the number of pages the message that will be sent.

Receiver Ratio from the highest to the lowest, is the segment 10, 15, and 20. From this result can be conclude the average results of decompression that is closer to the original message is a segment 10, 15, and 20.

3) Compression-Encryption Process Segmen 10, 15, and 20 (Based on Character Length)

This experiment started with compression process and then the results will be encrypted. Experimen result shown in Tabel 9. Best results are on the segment 20. In certain circumstances the results of compression-encryption produce message length is greater than the original message. It occurs in messages that use letter3 character type, in which one character represented by two characters.

In the reverse process, decryption process-decompression, the best results are on the segment 10. This is the same as the previous test results. Sequence best test results in 10 segments, 15 segments, and segment 20.

TABLE XII. EXPERIMENT RESULT ENCRYPTION-COMPRESSION PROCESS ON SEGMEN 10, 15, AND 20 (BASED ON CHARACTER LENGTH)

JUM. KAR	JUM. HAL	Rasio Enkripsi-Kompresi (%)						Rasio Dekompresi-Dekripsi (%)		
		10	JUM. HAL	15	JUM. HAL	20	JUM. HAL	10	15	20
18	1	105.56	1	105.56	1			0.00	5.56	
54	1	79.63	1	57.41	1	46.30	1	1.85	1.85	0.00
136	1	66.91	1	49.26	1	36.03	1	0.74	1.47	0.00
219	2	63.47	1	44.29	1	33.33	1	0.91	1.37	2.28
277	2	63.18	2	43.68	1	32.85	1	1.08	0.72	1.08
386	3	62.44	2	42.23	2	32.90	1	1.30	0.52	1.81
470	4	61.49	2	42.34	3	32.13	1	2.13	2.55	1.49
742	5	61.59	3	41.37	3	31.67	2	1.21	0.81	2.16
826	6	61.14	4	41.53	3	31.36	2	1.69	1.45	1.33
Rata-rata (%)		70.53		53.27		35.03		1.14	1.15	1.26

TABLE XIII. AVERAGE RATIO OF PTRCESS INVOLVING COMPRESSION

SEGMEN	10	15	20
RASIO PENGIRIMAN RATA-RATA (%)	70.53	53.27	35.03
RASIO PENERIMAAN RATA-RATA (%)	13.24	10.60	9.71

4) Encryption-Compression Process On Segmen 10, 15, and 20 (Based On Character Length)

This experiment started with encryption process and then the results will be compression. Experimen result shown in Tabel 10. Results delivery ratio the same as in the compression process. Not only seen in the ratio of delivery (encryption-compression ratio) but also looks at the number of the desired character. When analyzed, this happens because the process is done first is the encryption process. Where in the process is the number of characters of the original message will be the same as the number of characters the message from encryption.

In the process of decompression and decryption, ratio acceptance message is very small when compared with other processes. This happens on all characters long.

For testing based on the length of character shown in Table 11. The best compression results in 20 segments (35.03%). the smaller the percentage of compression, the better the compression process. In the process of decompression, the best results in the segment 10 (13,24%). Results of the decompression process be good, when the ratio is greater value, which indicates the ratio of the suitability of the character of the results of decompression and original character

VI. CONCLUSION

- Encryption and decryption results have an accuracy of 100%.
- Word variations from non-capital letters produce the most good decompression percentage at 19.74%. It occurs because the probability value for a non-capital letters larger than the other variation.
- The result of compression with variation characters (capital letters, non capital, numbers, and symbols) has the same percentage (61.17%) for each type of word variation. While the decompression process results have an accuracy between 18-19%.
- Variations of non-capital letters decompression produce the most good percentage on 19.74%. It's happens because the probability value for a non-capital letters larger than the other variation.
- For the results of compression process based on the length of characters, the accuracy of the best compression ratio contained in the segment 20 (35.03%), but it has the lowest decompression percentage 70.53%
- The ratio of the decompression process based on the length of characters, the best results is in the segment of 10 (13:24%), but the results of compression in this segment got the lowest compression (70.53%)
- The Combinations Arithmetic Coding compression and encryption Hill Cipher can increase the length of a message sent up to 25% from the original message.
- When the number of messages exceed 3 pages (480 characters), the message is not converted to MMS. This is because in this application are not given the limits on the number of characters and the number of pages the message.

REFERENCES

[1] A. Fatih, "Mengatasi SMS yang Berubah Menjadi MMS pada Android", 2013.

[2] A. Kamaludin, "Aplikasi Enkripsi Citra dengan Menggunakan Hill yang di-Modifikasi dan didukung Self Invertiblematriks Menggunakan Java2SE". Bekasi : Universitas Gunadarma Bekasi, 2010.

[3] B. P. Silalahi, F. Bukhari, S. Nurhudayani, "Pengkodean Aritmetik untuk Kompresi Data Teks", Bandung : Institute Pertanian Bogor, 2006.

[4] I. Ika, "Aplikasi Matriks dalam Kriptografi Hill Cipher". Singaraja, 2011.

[5] I. H. Putro, P. Santoso, M. Basoeki, "Aplikasi Java Mobile untuk Kompresi Layanan Pesan Singkat". Surabaya : Universitas Kristen Petra Surabaya, 2010.

[6] M. Basoeki, "Aplikasi Kompresi SMS dengan menggunakan Metode Arithmetic Coding pada Mobile Phone Berbasi JAVA. Surabaya : Universitas Kristen Petra Surabaya", 2008.

[7] T. Bray, M. Murphy, "Access Message Inbox without Content Url", 2010

[8] T. Douglas, D. Helliwell, "Hill Ciphers", College of the Redwoods, 1997.

[9] R. Rayarikar, S. Upadhyay, P. Pimpale, "SMS Encryption using AES Algorith on Android", International Journal of Computer Applications, (0975-8887), Volume 50-No.19, July 2012.

[10] "Short Message Service Security", 2008, The Government of the Hongkong Special Administrative Region, 2012.

[11] T. M. Mahmoud, B. A. Abdel-latef, A. A. Ahmed & A.M. Mahfouz, "Hybrid Compression Encryption Technique for Securing SMS", Tarek M Mahmoud, Bahgat A. Abdel-latef, Awny A. Ahmed & Ahmed M Mahfouz International Journal of Computer Science and Security (IJCSS, Volume (3): Issue (6). January 2010.

[12] R.J. de Lange, Executive Vice President Global Product Solutions, Tekelec, "Future of SMS", http://www.messaging.telecom2.com/articles/sms_trends_future_of_sms_messaging_tekelec_solutions.ht, downloaded on April 28th 2014.