Sistem Pengamanan Pesan Singkat Untuk Mobile Phone Berbasis Android Menggunakan Algoritma Hill Cipher

Isbat Uzzin Nadhori, Tita Karlita, Rani Dewi Ismawati
Jurusan Teknik Informatika
Politekni Elektronika Negeri Surabaya
{isbat,tita}@pens.ac.id, rani_ismawati@ymail.com

Abstract— **Since mobile phone 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 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.**

Keywords: encryption, decryption, sms, android

Abstrak—**Sejak telepon genggam menjadi barang yang harus dimiliki setiap orang, pengiriman pesan berbasis teks berupa pesan singkat atau Short Message Service (SMS) melalui telepon genggam, menjadi sesuatu yang hampir dilakukan setiap hari. Pesan singkat merupakan salah satu bentuk layanan komunikasi yang sudah banyak digunakan dan dikenal luas saat ini karena mudah digunakan dan murah dari sisi biaya. Pemanfaatan karakter teks sebagai media, tidak hanya memberikan kemudahan dalam penggunaannya, tetapi juga dapat menimbulkan permasalahan yaitu keamanan isi pesan. Permasalahan lain dikarenakan pesan singkat menggunakan standard pengkodingan yang universal. Pesan singkat yang terkirim dapat ditangkap oleh operator maupun oleh pihak ketiga dengan menggunakan alat yang tidak mahal. Dalam pesan tersebut berisi data nomor telepon pengirim dan penerima serta isi dari pesan itu sendiri. Sehingga pesan singkat dapat dengan mudah diganti dan dipalsukan oleh pihak ketiga. Hal ini memicu kekhawatiran karena adakalanya pesan singkat digunakan untuk melakukan pertukaran pesan yang sifatnya rahasia. Untuk mengatasi permasalahan tersebut, maka dibuat sebuah sistem pengamanan pesan singkat berbasis android menggunakan algoritma enkripsi. Dalam penelitian ini digunakan algoritma Hill Cipher untuk proses enkripsi.**

Kata kunci : enkripsi, dekripsi, sms, android

I. PENDAHULUAN

Saat ini telepon genggam menjadi barang yang harus dimiliki setiap orang, pengiriman pesan singkat melalui telepon genggam, atau SMS, menjadi sesuatu yang kita lakukan setiap hari. Pesan SMS menjadi sangat umum seperti panggilan telepon karena cepat, murah dan menyenangkan. Penggunaan SMS memberikan kemudahan dalam berkomunikasi dengan memanfaatkan teks sebagai media. Pemanfaatan karakter teks sebagai media, tidak hanya memberikan kemudahan dalam penggunaannya, tetapi juga akan menimbulkan permasalahan. Salah satunya adalah rentan terhadap masalah keamanan isi SMS. Sehingga dibutuhkan Kriptografi yaitu seni dan ilmu untuk mengamankan sebuah pesan (*plaintext*) menjadi pesan yang tersembunyi (*ciphertext*).

Untuk mengatasi permasalahan layanan SMS tersebut, maka dibuat sebuah sistem pengamanan pesan singkat atau SMS berbasis android menggunakan metode enkripsi yang dapat memudahkan pengguna dalam menggunakan layanan SMS sekaligus mengamankan isi pesan dari layanan SMS.

Sebelumnya telah ada penelitian mengenai Kriptografi Hill Cipher. Pada penelitian ini menjelaskan mengenai proses enkripsi dan dekripsi untuk 26 karakter dengan menggunakan matriks 2x2 oleh Todd Douglas dan Dustin Helliwell (1997)[8]. Enkripsi pada SMS dengan menggunakan algoritma AES telah dilakukan oleh Rohan Rayarikar [9]. Hasil risetnya menunjukkan tidak adanya delay pada proses pengiriman pesan. Sri Rangarajan juga membuat suatu sistem pengamanan SMS dengan menggunakan kunci simetris yang disebut dengan algoritma *Elliptic Curve Cryptograpgy* (ECC). Syarat utama dalam menggunakan sistem ECC adalah kedua node yaitu pengirim dan pengguna harus aktif di saat yang sama dan masing-masing sudah menginstal aplikasi ECC. Tarek M. Mahmoud juga telah membuat sistem keamanan yang dipadukan dengan kompresi SMS di Symbian OS dengan menggunakan algoritma RSA [11]. Hasil penelitian Tarek menunjukkan bahwa ukuran data terenkripsi dan terdekripsi tidak meningkat. Sebelumnya juga telah ada penelitian mengenai Kriptografi Hill Cipher. Pada penelitian ini menjelaskan mengenai proses enkripsi dan dekripsi untuk 26 karakter dengan menggunakan matriks 2x2 oleh Todd Douglas dan Dustin Helliwell (1997)[8].

Pada kesempatan kali ini, penulis ingin mengembangkan algoritma tersebut agar dapat digunakan untuk mengamankan isi pesan singkat pada mobile device android. Dimana untuk proses enkripsi dilakukan dengan menggunakan dua tipe metriks, yakni 2x2 dan 3x3 dan menggunakan 95 karakter yang umum digunakan untuk pengiriman pesan.

II. URGENSI PENELITIAN

Berdasarkan riset di bidang industri telekomunikasi [12], SMS adalah layanan komunikasi *non-voice* yang paling sukses diantara beberapa layanan yang ditawarkan oleh operator telekomunikasi *mobile*. Tiap tahun penggunaan SMS sebagai media komunikasi semakin meningkat seiring dengan meningkatnya penggunaan telepon genggam atau *smartphone*. Di Indonesia pengguna ponsel pintar terus meningkat. Bahkan, sebuah lembaga riset menyebutkan bahwa Indonesia berada di peringkat kelima dalam daftar pengguna smartphone terbesar di dunia. Diantara pengguna *smartphone*, populasi Android telah lebih 1 miliar. Dalam data tersebut disebutkan pula Indonesia menduduki posisi 5 besar dengan pengguna aktif sebanyak 47 juta, atau sekitar 14% dari seluruh total pengguna ponsel.

Namun demikian terdapat celah keamanan dalam proses pengiriman SMS. Layanan SMS sangat tergantung dengan *Short Message Service Centre* (SMSC) yang biasanya ditangani oleh penyedia atau operator telekomunikasi tanpa keamanan yang memadai [10]. Dalam prosesnya, sebuah SMS terkirim tanpa enkripsi. Sebelum sampai ke tujuan. Pesan akan tersimpan di SMSC terlebih dahulu. Sehingga bisa diakses oleh pihak lain yang memiliki akses pada sistem.Sedangkan email dikirimkan oleh *Mail Transfer Agent* (MTA) dari asal ke tujuan dengan melalui proses *hopping* dari satu *node* ke *node* lainnya. Potensi penyadapan dapat terjadi di setiap *node* yang dilalui.

Sebagai solusi maka akan lebih aman jika dibuat sistem keamanan SMS untuk data yang bersifat rahasia sehingga tidak disalahgunakan oleh pihak yang tidak bertanggung jawab. Strategi yang dapat dilakukan untuk meningkatkan keamanan data, dalam hal ini berbentuk *plaintext*, adalah menerapkan algoritma enkripsi.

III. ALGORITMA HILL CIPHER

Hill Cipher merupakan penerapan aritmatika modulo pada kriptografi. Teknik kriptografi ini menggunakan sebuah matriks persegi sebagai kunci yang digunakan untuk melakukan enkripsi dan dekripsi. Hill Cipher diciptakan oleh Lester S. Hill pada tahun 1929. Hill Cipher tidak mengganti tiap abjad yang sama pada *plaintext* dengan abjad lainnya yang sama pada *ciphertext* karena menggunakan perkalian matriks pada dasar enkripsi dan dekripsinya. Plaintext yang akan diproses pada Hill Cipher akan dibagi menjadi blok-blok dengan ukuran tertentu. Setiap karakter *plaintext* pada satu blok akan mempengaruhi karakter lainnya dalam proses enkripsi maupun dekripsi, sehingga karakter *plaintext* yang sama belum tentu menjadi karakter *ciphertext* yang sama pula.

Dasar teknik Hill Cipher adalah modulo terhadap matriks. Dalam penerapannya, Hill Cipher menggunakan teknik perkalian matriks dan teknik *invers* terhadap matriks. Kunci Hill Cipher adalah matriks n x n dengan n merupakan ukuran blok. Jika n=2, maka enkripsi dilakukan setiap 2 karakter. Matriks K yang menjadi kunci ini harus merupakan matriks yang *invertible*, yaitu memiliki *invers* K-1, sehingga:

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

Kunci harus memiliki *invers* karena matriks K^{-1} adalah kunci yang digunakan untuk melakukan dekripsi.

A. Algoritma Enkripsi Hill Cipher

Secara umum, tahapan-tahapan enkripsi Hill Cipher adalah sebagai berikut:

1. Buat matrik K yang dipakai sebagai kunci berukuran nxn.

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

2. Korespondenkan abjad dengan numerik

   A → 1 , B → 2, C → 3, ……., Z → 0

3. Kelompokkan barisan angka yang didapat kedalam beberapa blok vektor P yang panjangnya sama dengan ukuran matriks “K”.

4. Hitung Ciphertext (modulo 26) untuk tiap vektor P

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

   Dengan : C = *Ciphertext*
   K = Matriks kunci
   P = *Plaintext*

5. Kembalikan tiap angka dalam vektor sandi C ke huruf yang sesuai untuk mendapatkan *ciphertext* sandi.

B. Algoritma Dekripsi Hill Cipher

Secara umum, tahapan dekripsi Hill Cipher adalah sebagai berikut:

1. Korespondenkan abjad hasil enkripsi dengan numerik

   A → 1 , B → 2, C → 3, ……., Z → 0

2. Kunci yang digunakan untuk mendekrip *ciphertext* ke *plaintext* adalah *invers* dari matriks K_{nxn}

3. Hitung K^{-1} (*invers*) dengan cara:

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

   dengan nilai 1/Det K dalam mod 26

4. Hitung *plaintext* dengan cara:

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

5. Kembalikan tiap angka dalam vektor P ke huruf yang sesuai untuk mendapatkan *plaintext* kembali.

IV. METODE PENELITIAN

Dalam penelitian ini dibangun sistem enkripsi pada pesan singkat yang dapat memudahkan pengguna dalam menggunakan layanan SMS yang mengamankan isi

pesan. Untuk proses enkripsi digunakan algoritma Hill Cipher

A. *Blok Diagram*

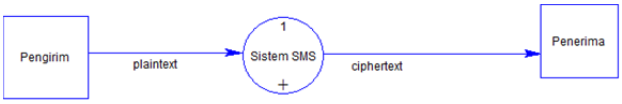
Blok diagram sistem secara keseluruhan dapat dilihat pada Gambar 1. Pengirim pesan dapat melakukan enkripsi pesan yang akan dikirim melalui layanan SMS, karena pesan yang diterima dalam keadaan terenkripsi, maka dari sisi penerima harus ada fitur dekripsi pesan supaya pesan aslinya bisa dibaca oleh penerima. Sehingga pada telepon genggam *receiver* (penerima) harus dibuat program dekripsi.

Gambar 1. Blok Diagram Sistem Pengamanan Pesan Singkat Diagram Sistem (Proses Enkripsi)

Aplikasi ini memiliki beberapa tahapan yang dapat dilakukan. Diawali dari menginstall aplikasi. Setelah aplikasi dibuka, untuk menuliskan isi pesan maka pilih menu New Message. Setelah semua selesai diketikkan dan akan dilakukan proses pengiriman, maka aplikasi akan memberikan pilihan menu enkripsi.

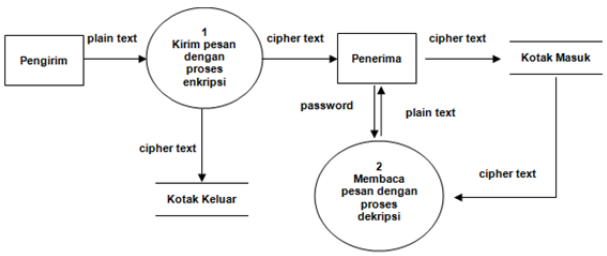
B. *Diagram Konteks*

Proses yang digambarkan pada level 0 seperti terlihat pada Gambar 2 adalah pengirim melakukan pengiriman pesan teks (*plaintext*) yang kemudian dilakukan proses enkripsi sehingga menghasilkan pesan teks dalam bentuk yang lain (*ciphertext*) dan dikirimkan ke penerima.



Gambar 2. Diagram Konteks (DFD level 0)

Pada level 1 seperti terlihat pada Gambar 3 pengirim mengirimkan SMS dengan proses enkripsi. Masukan yang diberikan ke sistem berupa *plaintext*. Pesan terenkripsi berupa *ciphertext* akan diterima penerima dalam "kotak masuk". Selanjutnya penerima akan membaca SMS yang masuk dengan terlebih dulu memasukkan *password*.



Gambar 3. Diagram Alur Data (DFD level 1)

C. *Enkripsi Pesan*

Proses enkripsi dilakukan dengan metode *Hill Cipher*. Kunci matriks yang digunakan pada aplikasi ini adalah matriks 2x2 dan 3x3 yang telah ditetapkan didalam aplikasi sebagai kunci statik. Ketika panjang pesan merupakan kelipatan dari 2, maka dimensi matriks kunci yang digunakan sesuai dengan panjang pesan. Artinya adalah pesan akan dibagi-bagi tiap 2 karakter untuk diproses dengan matriks kunci. Akan tetapi, ketika panjang pesan bukan merupakan kelipatan 2, maka akan ada 1 karakter yang tertinggal dan tidak ikut serta dalam pemrosesan dengan matriks kunci. Sehingga, untuk karakter terakhir tersebut akan dikirimkan sesuai dengan karakter aslinya. Nilai matriks harus bernilai positif, jika nilai indeks matriks negatif maka perlu dicari hasil matriks modulo. Nilai modulo bergantung pada jumlah karakter yang ada. Pada aplikasi ini, karakter dibagi dalam 3 inisialisasi, seperti terlihat pada *script* dibawah ini.

```
final String[] letter1={"A", "B", "C", "D", "E",
  "F" ,"G", "H", "I", "J", "K", "L", "M", "N",
"O", "P","Q", "R", "S", "T", "U", "V", "W", "X",
 "Y", "Z" , "a" , "b" , "c" , "d" , "e" , "f" ,
 "g" , "h" , "i" , "j" , "k" , "l" , "m" , "n" ,
 "o" , "p" , "q" , "r" , "s" , "t" , "u" , "v" ,
              "w" , "x" , "y" , "z"
   ,"0","1","2","3","4","5","6","7","8","9","
          ","."," ","'","?","!","\"","-
","(",")","@","/",":","_",";","+","&","%","*","=
              ","<",">","$","#"};

final String[] letter2={"A", "B", "C", "D", "E",
  "F" ,"G", "H", "I", "J", "K", "L", "M", "N",
"O", "P","Q", "R", "S", "T", "U", "V", "W", "X",
 "Y", "Z" , "a" , "b" , "c" , "d" , "e" , "f" ,
 "g" , "h" , "i" , "j" , "k" , "l" , "m" , "n" ,
 "o" , "p" , "q" , "r" , "s" , "t" , "u" , "v" ,
              "w" , "x" , "y" , "z"
   ,"0","1","2","3","4","5","6","7","8","9","
          ","."," ","'","?","!","\"","-
","(",")","@","/",":","_",";","+","&","%","*","=
","<",">","$","[","]","{","}","\\","~","^","#","
                |"};

final String[] letter3={"A", "B", "C", "D", "E",
  "F" ,"G", "H", "I", "J", "K", "L", "M", "N",
"O", "P","Q", "R", "S", "T", "U", "V", "W", "X",
 "Y", "Z" , "a" , "b" , "c" , "d" , "e" , "f" ,
 "g" , "h" , "i" , "j" , "k" , "l" , "m" , "n" ,
 "o" , "p" , "q" , "r" , "s" , "t" , "u" , "v" ,
              "w" , "x" , "y" , "z"
   ,"0","1","2","3","4","5","6","7","8","9","
          ","."," ","'","?","!","\"","-
","(",")","@","/",":","_",";","+","&","%","*","=
","<",">","$","[","]","{","}","\\","~","^","`","
                #","|"};
```

Inisialisasi karaktar (*letter*) ini dibagi dalam beberapa tipe. Untuk *letter1* terdapat 86 karakter, dimana 1 karakter akan mewakili 1 karakter sms. Untuk *letter2* terdapat 94 karakter. Dimana terdapat karakter-karakter yang ada pada *letter1* dan karakter-karakter yang satu karakternya mewakili 2 karakter dalam sms. Sedangkan *letter3* merupakan gabungan dari letter1 dan letter2 ditambah 1 karakter (`) yang mewakili 16 karakter sms.

Nilai akhir matriks akan bergantung pada jumlah karakter yang ada. Dikarenakan terdapat tiga tipe inisialisasi, maka matriks yang memiliki indeks negatif akan dimodulokan dengan tiga *length* yang berbeda dan tiap-tiap *length* ini akan didefinisikan sebagai kode *letter* enkripsi yang nantinya akan berpengaruh pada proses dekripsi, antara lain:

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

Berikut penjabaran matrik yang digunakan pada proses enkripsi:

1. **Matriks 2x2** (*berlaku untuk semua length letter)*

$$K = \begin{pmatrix} 4 & 3 \\ 3 & 2 \end{pmatrix}$$

2. **Matriks 3x3** (*inisialisasi awal)*

$$K = \begin{pmatrix} 1 & 3 & -1 \\ -2 & -5 & 1 \\ 1 & 5 & -2 \end{pmatrix}$$

3. **Matriks 3x3** (*length letter=95)*

$$K = \begin{pmatrix} 96 & 98 & 94 \\ 93 & 90 & 96 \\ 96 & 100 & 93 \end{pmatrix}$$

4. **Matriks 3x3** (*length letter=94)*

$$K = \begin{pmatrix} 95 & 97 & 93 \\ 92 & 89 & 95 \\ 95 & 99 & 92 \end{pmatrix}$$

5. **Matriks 3x3** (*length letter=86)*

$$K = \begin{pmatrix} 87 & 89 & 85 \\ 84 & 81 & 87 \\ 87 & 91 & 84 \end{pmatrix}$$

D. Dekripsi Pesan

Dekripsi dilakukan dengan pengubahan kunci matriks, yaitu kunci matriks *invers* dari matriks pada proses enkripsi, dimana matrik inver harus memenuhi syarat sebagai berikut:

$$K = \begin{pmatrix} 4 & 3 \\ 3 & 2 \end{pmatrix} \quad \begin{pmatrix} -2 & 3 \\ 3 & -4 \end{pmatrix}$$

$$K^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Sehingga untuk matriks kunci pada proses dekripsi adalah:

1. **Matriks 2x2**(*inisialisasi awal)*

$$K^{-1} = \begin{pmatrix} -2 & 3 \\ 3 & -4 \end{pmatrix}$$

2. **Matriks 3x3**(*inisialisasi awal)*

$$K^{-1} = \begin{pmatrix} 5 & 1 & -2 \\ -3 & -1 & 1 \\ -5 & -2 & 1 \end{pmatrix}$$

3. **Matriks 2x2**(*length letter=95)*

$$K^{-1} = \begin{pmatrix} 93 & 98 \\ 98 & 91 \end{pmatrix}$$

4. **Matriks 2x2**(*length letter=94)*

$$K^{-1} = \begin{pmatrix} 92 & 97 \\ 97 & 90 \end{pmatrix}$$

5. **Matriks 2x2**(*length letter=86)*

$$K^{-1} = \begin{pmatrix} 84 & 89 \\ 89 & 82 \end{pmatrix}$$

6. **Matriks 3x3**(*length letter=95)*

$$K^{-1} = \begin{pmatrix} 100 & 96 & 93 \\ 92 & 94 & 96 \\ 90 & 93 & 96 \end{pmatrix}$$

7. **Matriks 3x3**(*length letter=94)*

$$K^{-1} = \begin{pmatrix} 99 & 95 & 92 \\ 91 & 93 & 95 \\ 89 & 92 & 95 \end{pmatrix}$$

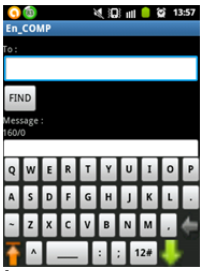
8. **Matriks 3x3**(*length letter=86)*

$$K^{-1} = \begin{pmatrix} 91 & 87 & 84 \\ 83 & 85 & 87 \\ 81 & 84 & 87 \end{pmatrix}$$

V. HASIL EKSPERIMEN

Tujuan dari pengujian aplikasi ini adalah untuk mengetahui tingkat keberhasilan dan resposibilitas aplikasi. Pengujian dilakukan dengan menggunakan Android Mobile Phone untuk menguji hasil pengiriman dan penerimaan pesan.

Pada aplikasi ini, untuk penulisan pesan digunakan menu *keypad* khusus seperti terlihat pada Gambar 4. Hal ini bertujuan untuk menyaring karakter-karakter yang akan dimasukkan oleh pengguna. Hal ini dikarenakan tidak semua karakter yang ada pada *keypad* asli android dapat diproses pada aplikasi ini



Gambar 4. Menu New Message

Pengujian ini dibagi menjadi dua tipe pengujian, yakni pengujian berdasarkan variasi karakter dan pengujian berdasarkan panjang pesan. Selama proses pengujian, diamati respon aplikasi yang berhubungan dengan waktu eksekusi kegiatan enkripsi dan dekripsi. Adapun hasil dari pengujian ini mempunyai tujuan sebagai berikut:

- Mengetahui pengaruh variasi karakter terhadap tingkat rasio masing-masing proses pengiriman dan penerimaan.
- Mengetahui pengaruh panjang pesan terhadap tingkat rasio masing-masing proses pengiriman dan penerimaan.

Rasio Pengiriman (RK) yang dimaksud dalam pengujian ini adalah rasio jumlah karakter pesan yang terkirim dibagi dengan jumlah karakter pesan yang dituliskan. Informasi jumlah karakter yang dituliskan dan yang terkirim dapat diketahui dari form Informasi pesan yang akan muncul setelah pengguna mengirimkan pesannya. Dalam hal ini semakin kecil rasio penerimaan berarti semakin bagus karena semakin mendekati jumlah karakter plain teks.

Rasio Penerimaan (RT) yang dimaksudkan dalam pengujian ini adalah rasio jumlah karakter yang benar (karakter hasil dekripsi sama dengan karakter yang dituliskan oleh pengirim) dibagi dengan jumlah karakter total yang dikirimkan. Perlu diketahui bahwa jumlah karakter yang dituliskan oleh pengirim pasti sama dengan jumlah karakter yang dibaca oleh penerima.

Persamaan untuk menghitung Rasio Pengiriman dan Rasio Penerimaan dalam % adalah sebagai berikut:

$$RK : \frac{\text{Jumlah karakter yang terkirim}}{\text{Jumlah karakter yang dituliskan}} \times 100\% \quad \text{.....(6)}$$

$$RT: \frac{\text{Jumlah karakter yang benar}}{\text{Jumlah karakter yang total pesan}} \times 100\% \quad \text{.....(7)}$$

A. Pengujian Berdasarkan Variasi Karakter

Pada proses pengujian ini variasi karakter dibagi menjadi tiga golongan, antara lain karakter huruf kecil (non kapital), huruf kapital, dan gabungan antara huruf (kapital dan non kapital), angka, dan simbol. Hal ini dimaksudkan untuk mengetahui pengaruh perbedaan jenis karakter terhadap hasil pengiriman dan penerimaan pesan pada tiap-tiap proses pengiriman pesan. Pengujian ini lebih ditekankan pada rasio penerimaannya yang dipengaruhi oleh variasi kata/karakter yang dituliskan.

a) Proses Enkripsi (Berdasarkan Variasi Karakter)

Hasil pengujian proses enkripsi berdasarkan pada variasi karakter ditunjukkan pada Tabel 1. Berdasarkan data hasil pengujian pada Tabel 1, terlihat bahwa Rasio Enkripsi mengalami pembengkakan diatas 100%. Hal ini dapat diartikan bahwa jumlah panjang karakter yang dikirimkan lebih besar dibandingkan dengan jumlah karakter yang dituliskan oleh pengirim. Pada variasi kata huruf (kapital maupun non kapital) jumlah halaman pesan yang dikirimkan bernilai sama. Hal ini dikarenakan semua huruf mewakili 1 karakter. Sehingga tidak ada penambahan nilai pada tiap karakter. Penambahan terjadi karena adanya 2 karakter tambahan yang dijadikan sebagai kode untuk proses dekripsi. Hal ini akan diulas lebih lanjut pada pengujian Enkripsi berdasarkan panjang karakter.

Berbeda dengan variasi kata pada “Gabungan”. Terlihat bahwa terjadi perbedaan jumlah halaman. Hal ini dikarenakan terdapat beberapa karakter simbol yang mewakili lebih dari 1 karakter. Sebagai contoh karakter *quote* (“) atau ([). Dimana 1 karakter ini memiliki panjang bit seperti 2 karakter.

Pada Table 2 diperlihatkan hasil rasio rata-rata untuk Rasio Pengiriman (RK) dan Rasio Penerimaan (RT) pada masing-masing variasi kata. Untuk semua variasi kata ternyata menghasilkan RK yang sama yaitu 101,063%. Untuk RK terbaik variasi kata terbaik untuk keseluruhan proses terdapat pada variasi kata huruf non kapital. Dimana memiliki rasio penerimaan terbesar, yakni 19.74289301 %. Dari ketiga tipe variasi kata tersebut juga terlihat semuanya memiliki prosentase dekripsi yang direpresentasikan dengan RT sebesar 100%. Artinya bahwa pesan yang dikirim dan diterima adalah sama.

TABEL 1 PENGUJIAN PROSES ENKRIPSI (VARIASI KARAKTER)

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

TABEL 2 RASIO RATA-RATA PENGUJIAN VARIASI KATA

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

B. Pengujian Berdasarkan Panjang Pesan

Pada pengujian ini dilakukan pengujian berdasarkan panjang pesan untuk pengiriman pesan dengan proses enkripsi.

a) Proses Enkripsi (Berdasarkan Panjang Pesan)

Pada proses pengujian ini dilakukan melakukan proses enkripsi atau merahasiakan isi pesan. Dimana pada aplikasi ini telah didefinisikan 2 tipe matriks, yaitu 2x2 dan 3x3. Ketika panjang pesan habis dibagi 3, atau dapat dibagi 3 dengan menyisakan 1 karakter, maka matriks yang digunakan adalah 3x3. Sedangkan jika panjang pesan tidak habis dibagi 3 dan jika dibagi 3 akan menyisakan lebih dari 1 karakter, maka matriks yang digunakan adalah matriks 2x2. Dengan asumsi bahwa panjang sms akan habis jika dibagi 2 dan akan menyisakan 1 karakter jika dibagi 2. Tabel 3 adalah tabel enkripsi berdasarkan beberapa panjang karakter:

Dilihat dari data hasil uji coba untuk proses enkripsi, dapat disimpulkan bahwa jika dilihat dari panjang karakternya, maka panjang pesan atau jumlah karakter yang dikirimkan ada jauh lebih besar sebanyak 2 karakter dibandingkan dengan jumlah karakter yang dituliskan. Dimana 2 karakter tambahan tersebut adalah karakter kode yang digunakan pada proses enkripsi. Karakter pertama merupakan kode untuk tipe proses. Dimana untuk proses enkripsi diberi kode 2, sehingga sistem pada

penerima dapat mengetahui proses apa yang telah digunakan oleh pengirim. Kemudian karakter kedua adalah adalah karakter untuk jenis letter yang digunakan. Letter yang dimaksud adalah kumpulan beberapa karakter yang digolongkan berdasarkan jumlah bitnya, yakni kode 1 untuk letter 1, kode 2 letter 2, dan kode 3 letter 3 yang telah dijelaskan pada bab sebelumnya.

TABEL 3 PENGUJIAN PROSES ENKRIPSI (PANJANG PESAN)

Panjang Pesan yang ditulis	Pesan yang dikirimkan	Jum. Halaman	Panjang Pesan yang diterima	Rasio Enkripsi (%)	Rasio 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

Jumlah karakter yang dikirimkan belum tentu sama dengan jumlah halaman pesan. Dimana pada dasarnya 1 halaman akan mewakili 160 karakter. Akan tetapi pada kenyataanya, pada sistem android tidak semua karakter akan mewakili 1 karakter. Terdapat beberapa karakter yang mewakili 2 karakter seperti karakter kurung suku ([). Ada pula karakter yang jika dituliskan untuk pertama kali, karakter ini akan mewakili 16 karakter, seperti karakter (`). Hal inilah yang menyebabkan penulis membagi karakter dalam beberapa tipe *letter*. *Letter* 1 akan berisi karakter yang hanya mewakili 1 karakter. *Letter* 2 berisikan karakter pada *letter* 1 ditambahkan dengan karakter-karakter yang mewakili 2 karakter. Sedangkan *letter* 3 adalah gabungan dari *letter* 1, *letter* 2, dan karakter yang dapat mewakili 16 karakter.

Sehingga pada saat dilakukan proses enkripsi, sistem aplikasi ini akan melakukan *scanning* terlebih dahulu terhadap isi pesan yang dituliskan. Hal ini dimaksudkan agar sistem dapat memilih *letter* mana yang akan digunakan. Hal ini bertujuan agar dapat mengurangi adanya pembengkakan kapasitas hasil proses enkripsi.

Jika dilihat dari hasil pengujian pada Tabel 3, dapat disimpulkan bahwa untuk proses enkripsi pesan, sistem aplikasi ini memiliki rasio penerimaan yang sangat baik yaitu 100%. Dimana pesan yang dituliskan oleh pengirim dapat dibaca dengan baik di sisi penerima, meskipun terdapat 2 karakter tambahan, atau bisa saja akan terjadi pembengkakan jumlah halaman yang akan dikirimkan mengingat perbedaan kapasitas pada masing-masing karakternya. Selama proses pengujian, tidak terdapat delay time yang berarti yang terjadi pada proses enkripsi dan dekripsi. Dengan kata lain, aplikasi berjalan normal sama seperti menulis dan membuka pesan singkat dengan menggunakan aplikasi SMS bawaan smartphone.

VI. KESIMPULAN

Kesimpulan dari uji coba aplikasi ini antara lain :

1. Aplikasi ini dapat bekerja dengan baik pada *mobile phone* Android dan memiliki respon dan waktu eksekusi normal, dengan mengkombinasikan bagian dari aplikasi ini dengan beberapa bagian fitur yang telah ada pada *mobile phone*.
2. Dengan memanfaatkan SQLite database pada *platform* android, pengguna dapat menyimpan dan menghapus beberapa item pada kotak masuk (*inbox*) mapun kotak keluar (*outbox*).
3. Pembuatan *keypad* pada aplikasi ini dapat menjadi batasan pengguna untuk menggunakan karakter-karakter input yang telah didefinisikan pada aplikasi, sehingga mengurangi tinggat kesalahan ketika karakter yang diinputkan tidak terdefinisikan oleh aplikasi.
4. Hasil enkripsi dan dekripsi memiliki akurasi prosentase 100%. Hal ini dapat disimpulkan karena antara proses enkripsi dan dekripsi memiliki kunci matriks yang sama.
5. Berdasarkan proses pengujian terhadap panjang karakter pada aplikasi ini, ketika jumlah pesan melebihi 3 halaman (480 karakter), pengirim tidak membutuhkan paket data (karena tidak terkonversi ke MMS) untuk melakukan proses pengiriman seperti pada aplikasi pesan yang telah ada pada *mobile* device android. Hal ini dikarenakan pada aplikasi ini tidak diberikan batasan mengenai jumlah karakter dan jumlah halaman pesan

REFERENCES

[1] Aik Fatih, “Mengatasi SMS yang Berubah Menjadi MMS pada Android”, 2013.

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

[3] Bib Paruhun Silalahi, Fahren Bukhari, Soliha Nurhudayani, “Pengkodean Aritmetik untuk Kompresi Data Teks”, Bandung : Institute Pertanian Bogor, 2006.

[4] Ika, “Aplikasi Matriks dalam Kriptografi Hill Cipher”. Singaraja, 2011.

[5] Iwan Handoyo Putro, Petrus Santoso, Maya Basoeki, “Aplikasi Java Mobile untuk Kompresi Layanan Pesan Singkat”. Surabaya : Universitas Kristen Petra Surabaya, 2010.

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

[7] Tim Bray, Mark Murphy, “Access Message Inbox without Content Url”, 2010

[8] Todd Douglas, Dustin Helliwell, “Hill Ciphers”, College of the Redwoods, 1997.

[9] Rohan Rayarikar, Sanket Upadhyay, Priyanka 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] Tarek M Mahmoud, Bahgat A. Abdel-latef, Awny A. Ahmed & Ahmed 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] Ronald 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, diunduh pada 28 April 2014.