IMPLEMENTASI ALGORITMA AES DAN MODIFIKASI VIGENERE UNTUK PENGAMANAN PESAN SMS DENGAN NOMOR PENGIRIM DAN PENERIMA SEBAGAI KUNCI TAMBAHAN

Imam Prayogo Pujiono

Abstract - One of the factor in the rise of information interception is the weak security in the information exchange. One means of information exchange that is widely used is to use SMS (Short Message Service). Messages sent through SMS can be any information which is confidential and may not be noticed by the public. The sender of the message is often not aware that message sent have a low level of security from eavesdropping. To maintain the security and confidentiality of message, an application to secure the message is needed. In this research, the author uses AES (Advanced Encryption Standard) algorithm to encrypt the message which will be sent and the modification of Vigenere algorithm to generate a new encryption key which is generated from the inputed key combination, the number of sender and receiver. The number of the sender and the receiver that are used to generate the new encryption key is to ensure that the content of the message will be known only from the handphone of the receiver by inputing the correct key. From the results of this research, eventhough another people able to get the ciphertext and knows the key, ita bug gets the ciphertext messages (encryption) and knowing the key safety message, the message remains unreadable due to a bug in mobile, meaning the number of the sender or recipient of the message has changed.

Key words: Security application, SMS, Cryptography, AES, Vigenere.

I. PENDAHULUAN

Salah satu faktor maraknya penyadapan informasi adalah lemahnya pengamanan sarana pertukaran informasi yang ada. Informasi dapat didefinisikan sebagai data yang diolah menjadi bentuk yang berguna bagi para pemakainya [1]. Salah satu sarana pertukaran informasi yang banyak digunakan adalah dengan menggunakan SMS (*Short Message Service*).

SMS adalah teknologi yang memungkinkan pengiriman dan penerimaan pesan antar telepon seluler. Pesan yang dikirim berupa teks yang terdiri dari kata-kata atau nomor atau kombinasi *alphanumeric*. SMS pertama kali muncul di Eropa pada tahun 1992. Setelah itu diadopsi ke teknologi nirkabel seperti CDMA dan TDMA [2]. Salah satu kelebihan dari sarana pertukaran informasi menggunakan SMS yaitu merupakan sebuah fitur dasar pada sebuah *handphone* sehingga bisa digunakan semua *handphone* tanpa memerlukan instalasi.

Pesan yang dikirim melalui SMS dapat berupa informasi yang sifatnya rahasia dan tidak boleh diketahui oleh umum, namun masyarakat seringkali tidak menyadari bahwa informasi yang dikirim memiliki tingkat keamanan yang rendah dari penyadapan. Hal ini disebabkan pesan SMS yang dikirim masih dalam bentuk *plaintext* (pesan asli) antara stasiun *mobile* dan SMS *Center* yang menggunakan jaringan nirkabel. Sehingga isi SMS yang tersimpan dalam sistem operator jaringan dapat dengan mudah dibaca oleh pihak yang tidak berhak. Selain dari penyadapan pada SMS *Center*, juga banyak beredar aplikasi-aplikasi yang menawarkan kemampuan untuk bisa membaca isi SMS. Oleh karena itu, adalah sebuah kebutuhan untuk menyediakan enkripsi tambahan pada pesan yang ditransmisikan [3].

Untuk itu dibutuhkan sebuah sistem keamanan pada layanan SMS yang mampu menjaga integritas dan keamanan isi pesan agar isi pesan (pesan asli) hanya dapat dibaca maknanya oleh penerima aslinya, maka isi pesan sebelum dikirim melalui SMS terlebih dahulu harus dienkripsi dengan algoritma kriptografi, misalnya AES (*Advanced Encryption Standard*) dan Vigenere.

Oleh karena itu penulis mempunyai ide untuk membuat aplikasi pengamanan pesan dengan tema "Implementasi Algoritma AES dan Modifikasi Vigenere untuk Pengamanan Pesan SMS dengan Nomor Pengirim dan Penerima Sebagai Kunci Tambahan". Digunakannya AES pada pengamanan pesan dikarenakan algoritma AES merupakan algoritma pengamanan yang kuat, hal ini terbukti dari ditetapkannya algoritma AES oleh *National Institute of Standards and Technology* (NIST) pada November 2001 sebagai standar algoritma pengamanan di Amerika Serikat yang menggantikan algoritma Data Encryption Standard (DES), algoritma AES sendiri didesain sedemikian rupa sehingga cukup kuat terhadap kriptanalisis linier dan diferensial [4]. Sedangkan modifikasi algoritma Vigenere digunakan untuk membuat kunci baru yang dihasilkan dari kombinasi kunci yang diinputkan, nomor pengirim dan penerima, dipilihnya algoritma Vigenere untuk pembuatan kunci baru karena

algoritma ini dinilai simple, cukup kuat dan dapat dimodifikasi dengan memberikan tiga inputan. Nomor pengirim dan penerima yang digunakan dalam kunci enkripsi berfungsi untuk memastikan bahwa isi pesan hanya dapat dibaca maknanya pada *handphone* penerima asli dengan memasukkan kunci yang benar. Sehingga meskipun seorang penyadap mendapatkan *ciphertext* (pesan enkripsi) dan mengetahui kunci pengamanan pesan, pesan tetap tidak terbaca maknanya di *handphone* penyadap karena nomor pengirim atau penerima pesan sudah berubah.

II. METODE YANG DIUSULKAN

A. Tinjauan Studi

Sistem keamanan pesan SMS menggunakan kriptogfi sudah pernah dilakukan oleh beberapa orang sebelumnya, antara lain:

Tabel 1. Penelitian terkait

No	Nama Peneliti	Judul
1	Arif Dwiyanto, Mukhlisulfatih Latief, Rochmad M T.	Penerapan Algoritma AES 128 dan Vigenere Cipher pada Aplikasi Enkripsi Pesan Singkat Berbasis Android.
2	Rohan R, Sanket U, Priyanka P.	SMS Encryption Using AES Algorithm on Android
3	Muhammad Rafii	Implementasi Algoritma ECDH dan AES untuk Pengamanan Pesan SMS Pada Telepon Seluler Berbasis Android.

Arif Dwiyanto, Mukhlisulfatih Latief, Rochmad Mohammad Thohir Jassin dalam (Penerapan Algoritma AES 128 dan Vigenere Cipher pada Aplikasi Enkripsi Pesan Singkat Berbasis Android, 2014) melakukan penelitian untuk membuat aplikasi pengamanan pesan SMS dengan memanfaatkan Algoritma AES 128 dan Vigenere Cipher untuk melakukan enkripsi pada pesan SMS, kedua algoritma tersebut digunakan untuk melakukan enkripsi terhadap pesan SMS / plaintext. Dari penelitian tersebut didapat hasil bahwa Aplikasi dapat mengirimkan pesan terenkripsi dan dapat melakukan dekripsi kembali apabila kunci yang dimasukkan sudah sesuai.

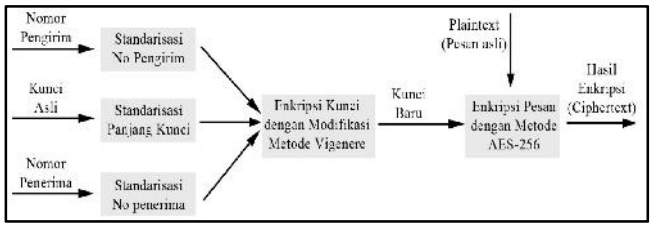
Rohan Rayarikar, Sanket Upadhyay, Priyanka Pimpale dalam (SMS Encryption using AES Algorithm on Android, 2012) melakukan penelitian untuk membuat aplikasi pengamanan pesan SMS dengan memanfaatkan Algoritma AES. Dari penelitian tersebut dihasilkan sebuah Aplikasi yang dapat digunakan untuk otentifikasi pengiriman pesan. Juga dimungkinkan mendeteksi jika pesan telah rusak atau dirusak selama transmisi. Yang terpenting dengan menggunakan aplikasi tersebut informasi sensitive pesan disimpan dengan aman dan tetap bersifat rahasia ketika perangkat diakses oleh penyadap.

Dalam tesis Muhammad Rafii seorang mahasiswa Universitas Gajah Mada dalam (Implementasi Algoritma ECDH dan AES untuk Pengamanan Pesan SMS Pada Telepon Seluler Berbasis Android, 2013) melakukan penelitian untuk membuat aplikasi pengamanan pesan SMS dengan memanfaatkan Algoritma AES dan ECDH untuk melakukan enkripsi pada pesan SMS. Dari penelitian tersebut dihasilkan sebuah aplikasi pengamanan pesan SMS berbasis Android yang mengimplementasikan algoritma AES dan ECDH pada proses enkripsinya.

Dari penelitian diatas penulis berusaha mengembangkan dari penelitian yang sudah ada. Oleh karenanya penulis menggunakan algoritma AES-256 dan modifikasi Vigenere guna mengamankan pesan, beserta menambahkan nomor pengirim dan penerima sebagai kunci tambahan dalam pengamanan pesan.

B. Metode AES dan modifikasi Vigener

Dalam penelitian ini penulis mengusulkan metode Advanced Encryption Standard (AES) dan modifikasi algoritma Vigenere yang digunakan untuk mengamankan pesan SMS. Digunakannya modifikasi dari gabungan kedua algoritma tersebut untuk memastikan keamanan pesan, AES sendiri merupakan standar enkripsi dengan kunci-simetris yang diadopsi oleh pemerintah Amerika Serikat. diumumkan oleh Institut Nasional Standar dan Teknologi (NIST) sebagai Standar Pemrosesan Informasi Federal (FIPS) pada tanggal 26 November 2001 setelah proses standardisasi selama 5 tahun, Berikut proses penyandian pesan / proses enkripsi yang terjadi dengan metode yang dikembangkang dari algoritma AES dan Vigenere:



Gambar 1. Proses enkripsi dengan metode yang diusulkan

Dari gambar diatas, terlihat bahwa pesan tidak hanya diamankan dengan menggunakan kunci pesan saja tetapi juga memasukkan nomor pengirim dan penerima untuk melakukan pengamanan sehingga meskipun seorang penyadap mendapatkan *ciphertext* (pesan enkripsi) dan mengetahui kunci pengamanan pesan, pesan tetap tidak terbaca maknanya di *handphone* penyadap karena nomor pengirim atau penerima pesan sudah berubah. Pada gambar 1 modifikasi algoritma Vigenere digunakan untuk mendapatkan sebuah kunci baru yang didapat dari gabungan kunci yang diinputkan, nomor pengirim dan nomor penerima yang terinput secara otomatis, kunci baru tersebut nantinya akan digunakan sebagai kunci dalam proses enkripsi dengan algoritma AES-256. Proses deskripsi yang terjadi di *handphone* penerima juga memiliki alur / tahapan yang sama seperti proses enkripsi di *handphone* pengirim, penerima hanya memasukkan kunci yang disepakati (yang sama) sebagai kunci yang diinputkan sedangkan nomor pengirim

dan penerima akan otomatis terinput saat proses pembuatan kunci baru.

C. Standarisasi Nomor Pengirim dan Penerima

Proses ini dilakukan untuk menstandarkan panjang nomor pengirim dan penerima menjadi 32 karakter yang nantinya akan digunakan sebagai kunci tambahan. dipilihnya 32 karakter dikarenakan pada AES-256 bit panjang kuncinya adalah 32 karakter. Proses merubah panjang nomor pengirim / penerima menjadi 32 karakter dapat di ilustrasikan pada gambar dibawah ini:

Nomor Lama	0	8	9	9	0	0	2	1	0	7	5					
Nomor Baru	5	7	0	1	2	0	0	5	7	0	1	2	0	0	5	7
Nomor Baru Ke	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Nomor Baru	0	1	2	0	0	5	7	0	1	2	0	0	5	7	0	1
Nomor Baru Ke	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32

Gambar 2. Proses standarisasi nomor pengirim / penerima

Pada gambar diatas dimisalkan nomor yang akan dimasukkan ke proses standarisasi nomor adalah ‘08990021075’ makan proses dimulai dengan mengambil 7 nomor dari belakang untuk membuat nomor baru, kemudian untuk melengkapi nomor baru menjadi 32 karakter maka dilakukan pengambilan 7 nomor dari belakang pada nomor lama untuk ditambahkan ke nomor baru secara berulang sehingga nomor baru menjadi 32 karakter. Dipilihnya 7 nomor dari belakang karena 2 nomor paling depan rawan berubah misal ‘0’ menjadi ‘+62’ dan nomor ke 3 sampai ke 5 merupakan kode wilayah atau kode operator yang tidak bersifat unik, selain itu dipilihnya 7 nomor untuk mengantisipasi terdapatnya nomor telepon dengan panjang nomor yang pendek (nomor cantik) sehingga tidak terjadi error karena yang dibutuhkan hanya 7 nomor saja.

D. Standarisasi Panjang Kunci

Proses ini dilakukan untuk meyetandarkan panjang kunci menjadi 32 karakter, dipilihnya 32 karakter dikarenakan pada AES-256 bit panjang kuncinya adalah 32 karakter. Proses merubah panjang kunci menjadi 32 karakter dapat di ilustrasikan pada gambar dibawah ini:

Kunci Lama	i	m	a	m	p	r	a	y	o	g	o					
Kunci Baru	i	m	a	m	p	r	a	y	o	g	o	i	m	a	m	p
Kunci Baru Ke	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Kunci Baru	r	a	y	o	g	o	i	m	a	m	p	r	a	y	o	g
Kunci Baru Ke	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32

Gambar 3. Proses standarisasi panjang kunci

Pada gambar diatas dimisalkan kunci yang dimasukkan adalah ‘imamprayogo’, maka proses dimulai dengan menambahkan karakter kunci lama (kunci yang di inputkan) ke kunci baru secara berulang dimuali dari karakter awal kunci lama sampai panjang kunci baru menjadi 32 karakter.

E. Enkripsi Kunci dengan Modifikasi Metode Vigenere

Proses ini dilakukan untuk membuat kunci baru yang natinya akan digunakan pada enkripsi pesan menggunakan AES-256. Pada proses ini input berupa nomor pengirim, nomor penerima dan kunci yang di inputkan, dimana ketiga input tersebut telah dilakukan proses standarisasi sehingga memiliki panjang karakter yang sama (32 karakter). Proses enkripsi kunci dengan modifikasi metode Vigenere dapat di rumuskan sebagai berikut:

$(N1_{(n)}+N2_{(n+1)}+K1_{(n)}) \bmod 126 = K2_{(n)}$

Dimana N1 adalah nomor pengirim ke (n), N2 adalah nomor penerima ke (n + 1), K1 adalah kunci awal ke (n), dan K2 adalah kunci akhir (*output*) ke (n). Misalkan N1=4 (ascii ke 52), N2=5 (ascii ke 53) dan K1=i (ascii ke 105).

Sehingga menghasilkan K2: ‘T’ (ascii ke 84). Yang didapat dari: (4+5+i) mod 126 = K2, yaitu (52+53+105) mod 126 = 84 (‘T’).

Untuk lebih jelasnya proses enkripsi kunci dengan modifikasi metode Vigenere dapat di ilustrasikan pada proses perhitungan dibawah ini, dimisalkan nomor pengirim = ‘08990021075’, nomor penerima = ‘089619613216’, kunci awal = ‘imamprayogo’. Dan ke tiga input tersebut dianggap telah melalui proses standarisasi.

Nomor Pengirim	5	7	0	1	2	0	0	5	7	0	1	2	0	0	5	7	
Nomor Penerima	6	1	2	3	1	6	9	6	1	2	3	1	6	9	6	1	2
Kunci Awal	i	m	a	m	p	r	a	y	o	g	o	i	m	a	m	p	
Kunci Akhir	Q	X	F	Q	Z	]	I	a	Z	L	S	S	X	I	U	[	
Kunci Akhir Ke	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	

Nomor Pengirim	0	1	2	0	0	5	7	0	1	2	0	0	5	7	0	1	
Nomor Penerima		3	1	6	9	6	1	2	3	1	6	9	6	1	2	3	6
Kunci Awal	r	A	y	o	g	o	i	m	a	m	p	r	a	y	o	g	
Kunci Akhir	W	E	c	Z	O	W	T	R	E	W	[	Z	I	d	T	P	
Kunci Akhir Ke	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	

F. Enkripsi Pesan dengan Metode AES

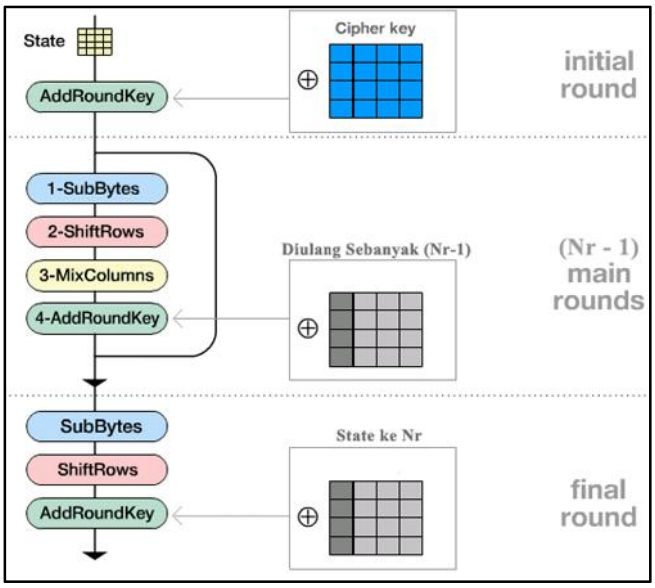
Advance Encryption Standard (AES) disosialisasikan oleh *National Institute of Standards and Technology* (NIST) pada November 2001. NIST mengumumkan bahwa AES digunakan sebagai pengganti dari algoritma enkripsi Data Encryption Standard (DES) yang telah lama dan kurang aman [13]. AES telah menjadi block cipher dimana dapat memproses block 128 bit dari input yang berupa *plaintext* dalam suatu waktu. AES juga telah mendukung pengaturan kunci 128, 192 dan 256 bit serta lebih efisien daripada DES [14].

AES mempunyai panjang block 128 bits dengan panjang kunci yang diperbolehkan adalah 128, 192 dan 256 bits. AES adalah *Cipher* iterasi dengan perputaran yang dinotasikan dengan Nr ditentukan oleh panjang kunci. Pada tabel 2 ini ditampilkan perbandingan panjang kunci dengan proses yang dilalui untuk masing-masing masukan [15].

Tabel 2. Perbandingan panjang kunci dan proses

	Jumlah Key	Besar Blok (Nb)	Jumlah Round (Nr)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Proses enkripsi pada algoritma AES terdiri dari 4 jenis tranformasi *bytes*. Yaitu *SubByte*, *ShiftRows*, *Mixcolumns*, dan *AddRoundKey*. Pada awal proses enkripsi input yang telah dikopikan ke dalam *state* akan mengalami tranformasi *byte AddRoundKey*. Kemudian *state* akan mengalami tranformasi *SubByte*, *ShiftRows*, *Mixcolumns*, dan *AddRoundKey* secara berulang-ulang sebanyak Nr. Proses ini dalam algoritma AES disebut *round function.* Round yang terakhir sedikit berbeda degan round sebelumnya dimana pada round ini *state* tidak mengalami tranformasi *MixColumns*, lebih jelasnya dapat dilihat pada gambar 4 [15].



Gambar 4. Proses enkripsi AES

Dari gambar diatas secara umum proses enkripsi AES terdiri dari empat langkah [12]:

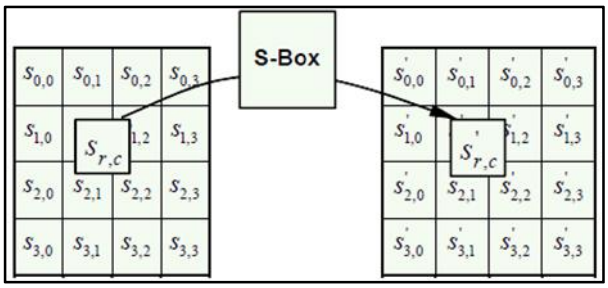
1. *SubByte*: Operasi ini merupakan suatu operasi subtitusi *nonlinier* yang beroperasi secara mandiri pada setiap *byte* dengan menggunakan tabel subtitusi S-Box (Tabel 3).

Tabel 3. S-Box – nilai subtitusi *byte* ($S_{r,c}$) [15]

S	Y															
	0	1	2	3	4	5	6	7	8	9	A	B	C	d	E	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	Fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	Af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	Cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	Eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	Ed	20	fc	b1	5b	6a	cb	Be	39	4a	4c	58	cf
6	d0	ef	Aa	Fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
X 7	51	a3	40	8f	92	9d	38	f5	bc	b6	Da	21	10	ff	f3	d2
8	cd	0c	13	Ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	Dc	22	2a	90	88	46	ee	b8	14	De	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	Ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	Ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	04	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	Ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

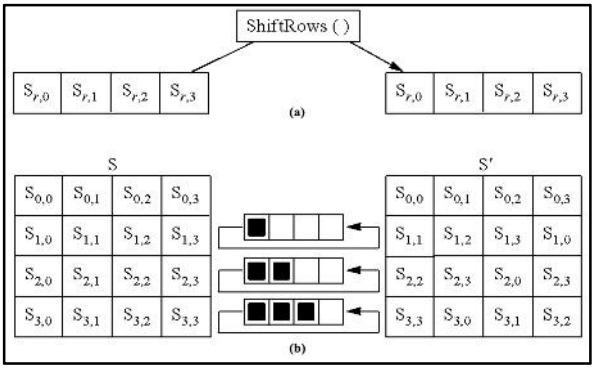
Cara pensubtitusian adalah sebagai berikut: untuk setiap byte array *state*, misalkan S[r,c] = xy, yang dalam hal ini xy adalah digit heksadesimal dari nilai S[r,c], maka nilai subtitusinya dinyatakan dengan S'[r,c] adalah elemen di dalam S-box yang merupakan perpotongan baris x dengan kolom y.

Misalnya S[0,0]=19, maka S'[0,0]=d4.



Gambar 5. Subtitusi *Byte* [15]

2. *ShiftRows*: merupakan langkah permutasi yang dieksekusi lewat pergeseran secara *wrapping* (siklis) pada 3 baris terakhir dari array state. Jumlah pergeseran bergantung pada niali baris (r). Baris r = 1 digeser sejauh 1 *byte*, baris r = 2 digeser sejauh 2 byte, dan baris r = 3 digeser sejauh 3 byte. Baris r = 0 tidak digeser.



Gambar 6. *Shift Rows* [15]

3. *MixColumns*: mengoperasikan *state* kolom demi kolom. Operasi ini dilakukan pada *state* kolom, dengan mengkonversikan setiap kolom sebagai polynomial. Kolom dianggap sebagai polynomial pada GF(2^8). Tranformasi ini dapat digambarka pada gambar 8 dengan perkalian matrik seperti persamaan pada gambar 7.

$$S'_{0,c} = ([02] \bullet S_{0,c}) \oplus ([03] \bullet S_{1,c}) \oplus S_{2,c} \oplus S_{3,c}$$

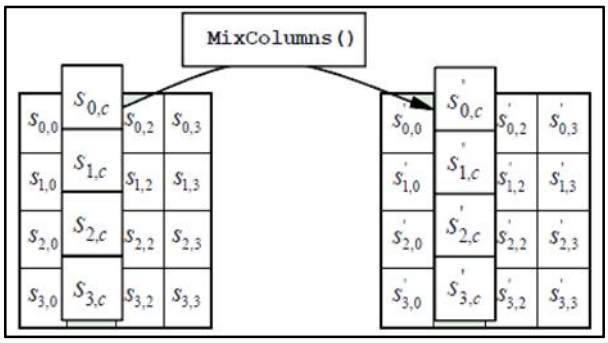
$$S'_{1,c} = S_{0,c} \oplus ([02] \bullet S_{1,c}) \oplus ([03] \bullet S_{2,c}) \oplus S_{3,c}$$

$$S'_{2,c} = S_{0,c} \oplus S_{1,c} \oplus ([02] \bullet S_{2,c}) \oplus ([03] \bullet S_{3,c})$$

$$S'_{3,c} = ([03] \bullet S_{0,c}) \oplus S_{1,c} \oplus S_{2,c} \oplus ([02] \bullet S_{3,c})$$

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} x \begin{bmatrix} S\,0,c \\ S\,1,c \\ S\,2,c \\ S\,3,c \end{bmatrix} = \begin{bmatrix} S'\,0,c \\ S'\,1,c \\ S'\,2,c \\ S'\,3,c \end{bmatrix}$$

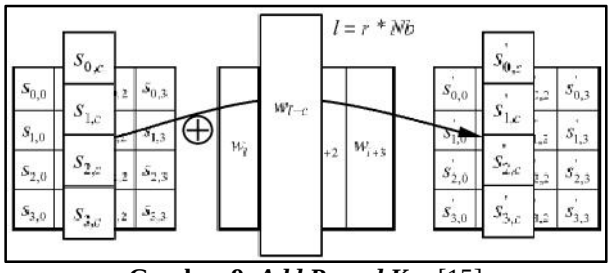
Gambar 7. Persamaan matriks *Mix Columns* [15]



Gambar 8. *Mix Columns* [15]

4. *AddRoundKey*: Transformasi ini melakukan operasi XOR terhadap sebuah *round key* dengan *array state*, dan hasilnya sebagai *array state* yang baru. Nilai awal dari *round key* adalah w_0, untuk *round* = 0, ditambahkan pertama kali pada *cryptographic round*. Kemudian setiap round untuk $1 \leq round \leq Nr$, kemudian 32-bit yang berbeda pada round-key w_i ditambahkan.

$[S'_{0,c}, S'_{1,c}, S'_{2,c}, S'_{3,c}] = [S_{0,c}, S_{1,c}, S_{2,c}, S_{3,c}] \oplus [w_{round * Nb+c}]$
Untuk $0 \leq c < Nb$.

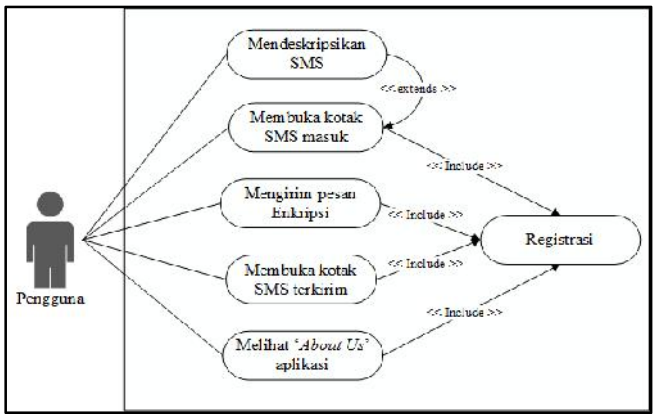


Gambar 9. *Add Round Key* [15]

III. IMPLEMENTASI

A. Use Case Diagram Aplikasi

Use-case diagram merupakan model diagram UML (*Unified Modeling Language*) yang digunakan untuk menggambarkan requirement fungsional yang diharapkan dari sebuah sistem. Use-case diagram menekankan pada "siapa" melakukan "apa" dalam lingkungan sistem perangkat lunak yang akan dibangun. Hal-hal yang dapat dilakukan pengguna terhadap sistem yang dibangun pada penelitian ini dapa dilihat pada diagram use case dibawah ini:



Gambar 10. Use Case diagram aplikasi

Pada gambar 10, yang dimaksud dengan pengguna adalah pengirim atau penerima, dimana pengguna dapat menjadi pengirim dan penerima pesan.

B. Implementasi Modifikasi Algoritma Vigenere

Algoritma Vigenere seperti yang dijelaskan pada landasan teori adalah algoritma kriptografi yang kuncinya simetris, yaitu kunci pada saat enkripsi dan deskripsi sama. Dalam modifikasi Algoritma Vigenere, algoritma ini digunakan untuk menghasilkan sebuah kunci baru dimana kunci tersebut akan digunakan dalam proses enkripsi dengan algoritma AES. Potongan program modifikasi Algoritma Vigenere yang digunakan untuk membuat kunci baru dapat dilihat pada gambar dibawah ini:

```
public static String Vigenere (String kunci, String npengirim, String npenerima){
    npengirim=standarnomer(npengirim);
    npenerima=standarnomer(npenerima);
    kunci=expandkey(kunci);

    String k="";

    for(int j=0;j<kunci.length();j++){
        int newC=0;
        int i=j+1;
        if(i==kunci.length()){i=0;}
        newC = (newC + ((npengirim.charAt(j) + npenerima.charAt(i) + kunci.charAt(j)) %126));
        if(newC < 32){
            newC=newC+32;
        }
        k = k + (char) newC;
    }
    return k;
}
```

Gambar 11. Potongan *Source Code* Modifikasi Algoritma Vigenere

Pada gambar 11 dapat dilihat bahwa dalam proses menghasilkan kunci baru setiap huruf / karakter kunci ke (n) dijumlahkan dengan huruf / karakter nomor pengirim ke (n) dan nomor penerima ke (n+1) kemudian dilakukan modulo dengan angaka 126, sehingga menghasilkan sebuah kunci baru.

C. Implementasi Algoritma AES

Algoritma AES seperti yang dijelaskan pada landasan teori adalah algoritma kriptografi yang kuncinya simetris, yaitu kunci pada saat enkripsi dan deskripsi sama, dalam penggunaanya untuk mengenkripsi pesan maupun mendeskrisi pesan, sebuah kunci simetris haruslah dibuat terlebih dahulu. Potongan program untuk mengenkripsi pesan dengan algoritma AES dapat dilihat pada gambar dibawah ini:

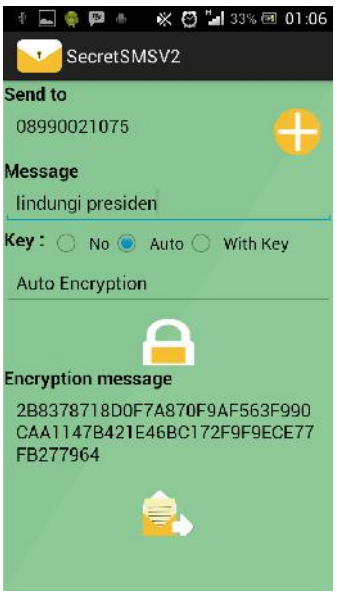
```
public String enkripsiaes(String message, String key, String npeng, String npen){
    String sms="",sms2="",sms3="",temp="";
    String kunci1="",kunci2="";

    sms=expandplaintext(message);
    key=Vigenere(key,npeng,npen);

    for(int i=0;i<16;i++){
        kunci1=kunci1+key.charAt(i);
        kunci2=kunci2+key.charAt(i+16);
    }
    kunci1=StringtoHex(kunci1);
    kunci2=StringtoHex(kunci2);
    for(int i=0;i<sms.length()/16;i++){
        temp="";
        for(int j=0;j<16;j++){
            temp=temp+sms.charAt(j + (i*16));
        }
        temp=StringtoHex(temp);
        sms2=addRoundKey(temp, kunci1);
        for(int k=1;k<14;k++){
            sms2=subbyte(sms2.toLowerCase());
            sms2=shifrow(sms2);
            sms2=mixcolom(sms2);
            if(k==1 && i==0){
                kunci1=KunciNRound(kunci2, 1);
                sms2=addRoundKey(sms2,kunci1);
            }
            else{
                kunci1=KunciNRound(kunci1, k);
                sms2=addRoundKey(sms2,kunci1);
            }
        }
        sms2=subbyte(sms2.toLowerCase());
        sms2=shifrow(sms2);
        kunci1=KunciNRound(kunci1, 14);
        sms2=addRoundKey(sms2,kunci1);

        sms3=sms3+sms2;
    }
    return sms3;
}
```

Gambar 12. Potongan *Source Code* Enkripsi Algoritma AES

Pada gambar diatas dapat dilihat bahwa untuk mengenkripsi pesan SMS, sistem membutuhkan input berupa teks pesan, kunci pesan, nomor pengirim dan nomor penerima.

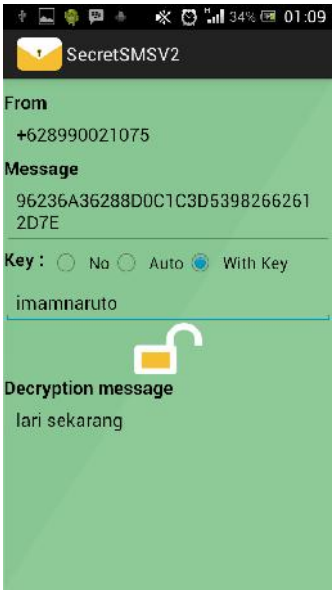
IV. HASIL & PEMBAHASAN

A. Antar Muka Aplikasi



Gambar 13. Antar muka Tulis Pesan

Pada gambar 13 Apabila pengguna ingin menulis pesan baru maka pengguna dapat memilih metode enkripsi yang tersedia dan dapat memasukan sebuah kunci pengamanan pesan. Pada gambar diatas ikon lingkaran dengan simbol plus dibagian dalam berguna untuk menampilkan kontak telepon yang tersimpan, sedangkan ikon gembok tertutup digunakan untuk melakukan proses enkripsi terhadap sebuah pesan dan ikon amplop pesan digunakan untuk mengirim pesan.



Gambar 14. Antar muka Baca pesan

Pada gambar 14 apabila pengguna ingin mendeskripsikan pesan pertama-tama pengguna harus memilih mode deskripsi (No Key, Auto Key atau With Key) dan memasukan kunci pesan setelah itu pengguna harus menekan ikon gembok terbuka untuk mendeskripsikan sebuah pesan.

B. Pembahasan Modifikasi Algoritma Vigenere

Pada bagian ini yang akan diuji dan dibahas tentang proses pembuatan kunci baru menggunakan modifikasi Algoritma Vigenere yang nantinya kunci tersebut akan digunakan pada proses enkripsi pesan menggunakan algoritma AES. Misalkan kunci pesan yang diinput adalah 'imamprayogo', nomor pengirim adalah '08990021075' dan nomor penerima adalah '089619613216' maka proses pembuatan kunci baru adalah sebagai berikut:

a. Nomor pengirim, nomor penerima dan kunci yang diinputkan masuk ke dalam proses standarisasi. Proses standarisasi nomor pengirim dengan nomor '08990021075' adalah sebagai berikut:

Nomor Lama	0	8	9	9	0	0	2	1	0	7	5					
Nomor Baru	5	7	0	1	2	0	0	5	7	0	1	2	0	0	5	7
Nomor Baru Ke	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Nomor Baru	0	1	2	0	0	5	7	0	1	2	0	0	5	7	0	1
Nomor Baru Ke	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32

Proses standarisasi kunci yang di inputkan (*expandkey*) dengan kunci 'imamprayogo' adalah sebagai berikut:

Kunci Lama	i	m	a	m	p	r	a	y	o	g	o					
kunci Baru	i	m	a	m	p	r	a	y	o	g	o	i	m	a	m	p
Kunci Baru Ke	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Nomor Baru	r	a	y	o	g	o	i	m	a	m	p	r	a	y	o	g
Nomor Baru Ke	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32

Dari proses standarisasi tersebut dihasilkan nomor pengirim baru '57012005701200570120057012005701', nomor penerima baru '61231696123169612316961231696123', dan kunci baru 'imamprayogoimamprayogoimamprayog', dimana ketiganya memiliki panjang 32 karakter.

b. Melakukan penjumlahan ascii dari setiap karakter ke (n) nomor pengirim hasil standarisasi dengan karakter ke (n+1) nomor penerima hasil standarisasi dan karakter ke (n) kunci hasil standarisasi kemudian hasil dari penjumlahan tersebut di modulo dengan angka 126, sehingga akan dihasilkan sebuah ascii dari kunci baru ke (n).

Dari proses penjumlahan ascii nomor pengirim, penerima dan kunci hasil standarisasi kemudian dilakukan modulo dengan angka 126 maka didapatkan sebuah teks sebagai kunci baru yaitu 'QXFQZ]IaZLSSXIU[WEcZOWTREW[ZIdTP'.

C. Pembahasan Algoritma AES

Pada bagian ini akan dibahas tentang enkripsi dan deskripsi pesan menggunakan algoritma AES-256 bit. Untuk kunci yang akan digunakan pada proses enkripsi dan deskripsi adalah 'QXFQZ]IaZLSSXIU[WEcZOWTREW[ZIdTP' atau kunci yang didapat dari hasil proses pembuatan kunci dengan modifikasi algoritma Vigenere sedangkan pesannya (plaintext) adalah 'Lindungi Presiden'. maka proses enkripsinya adalah sebagai berikut:

a. Plaintext diubah kedalam bentuk hexa:

L	i	n	d	u	n	g	i		P	r	e	s	i	d	e	n
4c	69	6e	64	75	6e	67	69	20	50	72	65	73	69	64	65	6e

4C	75	20	73
69	6E	50	69
6E	67	72	64
64	69	65	65

State 1

6E	20	20	20
20	20	20	20
20	20	20	20
20	20	20	20

State 2

Pada state diatas terlihat bahwa plaintext dijadikan 2 buah state yang mana setiap statenya terdiri dari 16 byte, dan apabila didalam state ada ruang kosong maka isi dari state tersebut diberi isian 20 atau 'spasi', dipilihnya spasi untuk melengkapi isi dari state menjadi 16 byte karena

dalam sebuah SMS, hilangnya (dihapusnya) spasi di akhir pesan tidak merubah makna dari sebuah pesan.

b. Kunci diubah kedalam bentuk hexa:

Q	X	F	Q	Z	]	I	a	Z	L	S	S	X	I	U	[
51	58	46	51	5a	5d	49	61	5a	4c	53	53	58	49	55	5b

State 1

W	E	c	Z	O	W	T	R	E	W	[	Z	I	d	T	P
57	45	63	5a	4f	57	54	52	45	57	5b	5a	49	64	54	50

State 2

c. Melakukan proses penjadwalan kunci, yang nantinya akan digunakan pada proses Addroundkey di setiap roundnya.

d. Pada algoritma AES proses enkripsi terdiri dari 4 tahapan yaitu:
1. Subbyte
2. Shiftrow
3. Mixcoloumns
4. Addroundkey

e. Proses enkripsi dilakukan tiap state dengan alur seperti yang di jelaskan pada subbab II.F tentang Enkripsi Pesan dengan Metode AES.

V. PENUTUP

A. Kesimpulan

Berdasarkan hasil pengembangan aplikasi ini dan pembahasan pada bab-bab yang tersaji sebelumnya pada laporan ini, maka dapat ditarik kesimpulan sebagai berikut:

1. Sebuah perangkat lunak yang mengimplementasikan algoritma enkripsi-deskripsi AES dan Vigenere dengan nomor pengirim dan penerima sebagai kunci enkripsi tambahan berhasil dibangun. Perangkat lunak tersebut dapat melakukan pengiriman pesan SMS (dengan memberi kunci terhadap pesan atau memberi kunci secara otomatis) dan pesan hanya dapat dibaca di *handphone* penerima.
2. Kekurangan dari implementasi algoritma AES untuk enkripsi adalah pesan menjadi lebih panjang dari sebelumnya dikarenakan karakteristik dari algoritma AES dimana setiap statenya harus berisi 16 karakter per state.

B. Saran

Beberapa hal yang perlu dikembangkan dari penelitian ini adalah:

1. Pada pengembangan selanjutnya diharap dapat menerapkan algoritma kompresi terhadap *ciphertext* yang dihasilkan agar panjang pesan dapat di minimalkan.

REFERENCES

[1] H. Jogiyanto, Metodologi Penelitian Sistem Informasi, Yogyakarta: Andi, 2008.

[2] Medani, A., Gani, A., Zakaria, O., Zaidan, A. A., Zaidan, B. B., "*Review of mobile short message service security issues and techniques toward the solution*," *Scientific Research an Essay*, vol. 6, pp. 1147-1165, 2011.

[3] Chavan, R. R., Sabness, M.,"*Secured Mobile Messaging*" *in International Conference on Computing, Electronics and Electrical Technologies* (ICCEET), Mumbai, 2012.

[4] A. R. Amin, Studi Block Cipher Serpent dan Rijndael, Bandung: Institut Teknologi Bandung, 2006.

[5] Asoke K. Talukder, *Mobile Computing*, 2005.

[6] Wahana komputer, Pengembangan Aplikasi Sistem Informasi Akademik Berbasis SMS dengan Java, Jakarta: Salemba Infotek, 2005.

[7] Menezes A. J., Van Oorschot, P. C., Vanstone, S. A., *Handbook of Applied Cryptography*, USA: CRC Press, 2007.

[8] Rangga W, Implementasi Algoritma RC6 Untuk Enkripsi SMS Pada Telepon Seluler, Bandung: Institut Teknologi Bandung, 2010.

[9] R. Munir, Kriptografi, Bandung: Informatika, 2006.

[10] Talbot, Jhon, Welsh, *Complexity and Cryptography*, USA: Cambridge University Press, 2006.

[11] B. Schneier, *Applied Cryptography, Second Edition : Protocol, Algorithms and Source Code in C*, USA: John Willey and Sons, Inc, 1996.

[12] W. Stalling, *Cryptography and Network Security Principles and Practices*, New Jersey: 4thPrentice, 2005.

[13] Jhonson S., Dennis S., *Cryptography for Developer*, Syngress: Rockland, 2007.

[14] Sumitra, "*Comparative Analysis of AES and DES security Algorithm*," *International Journal of Scientific and Research Publication*, vol. 3, no. 1, pp. 1-5, 2013.

[15] National Institute of Standards and Technology, "*Announcing the ADVANCED ENCRYPTION STANDART* (AES)," in Federal Information Processing Standards Publication, USA, 2001.

[16] Latipun, Psikologi Eksperimen, Malang: UMM Press, 2002.

[17] A. Barent, "www.adambarent.com," 13 Agustus 2003. [Online]. Available: www.adambarent.com/AESbyExample.pdf. [Accessed 22 Januari 2015].

[18] Muhammad Rafii, Implementasi Algoritma ECDH dan AES untuk Pengamanan Pesan SMS Pada Telepon Seluler Berbasis Android, Yogyakarta: Universitas Gadjah Mada, 2013.