APLIKASI KEAMANAN SMS MENGGUNAKAN ALGORITMA RIJNDAEL

Raisul Azhar[1], Kurniawan [2]

[1]Tenaga Pengajar S1 Teknik Informatika STMIK Bumigora Mataram
[2]Mahasiswa S1 Teknik Informatika STMIK Bumigora Mataram

[1]raisulazhar@yahoo.co.id, [2]kurni_awan65@gmail.com

ABSTRACT

Mobile phones and smartphones providing the basic features of SMS (Short Message Service). Messages sent through SMS facility sometimes confidential, so that the necessary mechanisms to protect messages from crime. Mechanism that can be used to protect the message is to use a cryptographic algorithm Rijndael. This algorithm is a cryptographic algorithm chipher 128-bit block that has the ability to protect confidential messages. This research resulted in an application that can be used to protect the SMS message when the message is sent and received by users. Based on research conducted on the length of different messages with the same key length is obtained that the length of the message used computing time takes longer than the length of the message is the same and different key lengths.

***Key words**: SMS, Cryptography Rijndael, Smartphone, Block Chipper*

I. PENDAHULUAN

1.1. Latar Belakang

Perkembangan perangkat telekomunikasi pada saat ini telah mengubah paradigma sistem telekomunikasi konvensional. Lahirnya smartphone membuat pengguna perangkat telekomunikasi dapat berkomunikasi dan bekerja layaknya sebuah komputer. Namun dibalik kemajuannya itu, smartphone masih tetap menyediakan fitur dasar sebuah pesawat telekomunikasi yaitu SMS (*Short Message Service).* SMS merupakan sebuah layanan yang banyak diaplikasikan pada sistem komunikasi tanpa kabel, memungkinkan dilakukannya pengiriman pesan dalam bentuk *alphanumeric* antara terminal pelanggan. [1] Sebagai fitur dasar dalam sistem komunikasi nirkabel, SMS memegang peranan penting bagi penggunannya dalam bertukar informasi. SMS biasanya digunakan untuk mengirimkan pesan yang bersifat pribadi dan umum seperti SMS Banking, transaksi bisnis, ataupun SMS biasa.

Mengingat pentingnya pesan yang terkandung dalam SMS membuat orang yang tidak bertanggung jawab ingin mengambil keuntungan dengan memanfaatkan kelemahan fitur SMS. SMS memiliki kelemahan dimana ia dibangun dengan sistem dan program yang sama, dan SMS bisa melakukan roaming jaringan setempat hingga ke jaringan asing. Komunikasi SMS memungkinkan pengiriman SMS spoofing dalam bentuk penyamaran ataupun manipulasi informasi seperti alamat atau data lainnya yang menyerupai pemakai pada umumnya. Ditambah lagi dengan berkembangnya perangkat keras dan perangkat lunak pada sistem komunikasi nirkabel yang melahirkan *smartphone* membuat kejahatan tersebut semakin mudah dilakukan.

Salah satu mekanisme yang bisa digunakan untuk melindungi pesan adalah kriptografi. Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dengan memanfaatkan teori matematika[2]. Kriptografi telah banyak di implementasikan untuk melindungi data-data rahasia seperti data perbankan, mesin *atm, e-comerce*, dan lain-lain. Salah satu algoritma kriptografi yang bisa digunakan untuk melindungi pesan SMS yaitu Algoritma Rijndael. Rijndael termasuk dalam jenis

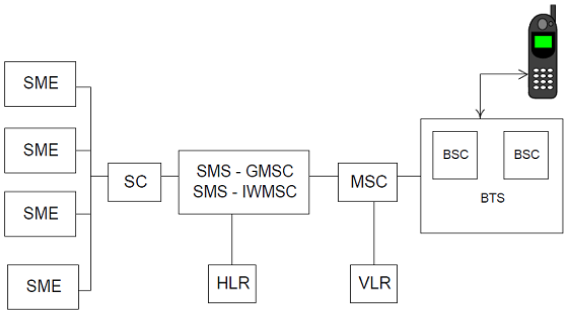
algoritma simetris dan cihpher *block*. Dengan demikian algoritma ini menggunakan kunci yang sama saat enkripsi dan dekripsi serta masukan dan keluarannya berupa blok dengan jumlah bit tertentu. Rijndael mendukung berbagai variasi ukuran blok dan kunci yang akan digunakan, namun Rijndael mempunyai ukuran blok dan kunci yang tetap sebesar 128, 192, 256 bit. Pemilihan ukuran blok data dan kunci akan menentukan jumlah proses yang harus dilalui untuk proses enkripsi dan dekripsi.[5]

Perkembangan pesawat smartphone saat ini banyak menggunakan system operasi android, namum penggunaan android hanya sebatas pemanfaatan aplikasi-aplikasi yang terdapat didalamnya. Pemanfaatan android belum dilakukan secara maksimal seperti untuk aplikasi keamanan. Berdasarkan hal tersebut peneliti tertarik untuk meneliti aplikasi keamanan *short message service* (sms) menggunakan algoritma kriptografi rijndael berbasis android. Aplikasi ini mempunyai kemampuan untuk melakukan enkripsi pesan SMS sebelum dikirim serta dapat melakukan dekripsi pesan SMS yang terenkripsi yang telah diterima. Sehingga aplikasi ini bermanfaat dalam membantu kerahasiaan suatu pengiriman dan penerimaan SMS.

1.2. Short Message Service

SMS merupakan sebuah layanan yang bersifat *none real time* dimana sebuah *short message* dapat di *submit* ke suatu tujuan, tidak peduli apakah tujuan tersebut aktif atau tidak. Bila dideteksi bahwa tujuan tidak aktif, maka sistem akan menunda pengiriman ke tujuanhingga tujuan aktif kembali. Pada dasarnya sistem SMS akan menjamin *delivery* dari suatu *short message*hingga sampai tujuan. Kegagalan pengiriman yang bersifat sementara seperti tujuan tidak aktif akan selalu teridentifikasi sehingga pengiriman ulang *short message* akan selalu dilakukan kecuali apabila diberlakukuan aturan bahwa *short message* yang telah melampaui batas waktu tertentu harus dihapus dan dinyatakan gagal kirim.[2]

Layanan SMS dibangun dari berbagai *entitas*yang saling terkait dan mempunyai fungsi dan tugas masing-masing. Tidak ada satupun dalam sistem SMS yang dapat bekerja secara *parsial*. *Entitas* dalam jaringan SMS ini disebut juga elemen SMS. Di bawah ini merupakan arsitektur SMS dengan beberapa elemen-elemen yang saling terkait :



Gambar.1. Arsitektur SMS

Elemen-elemen dasar pada jaringan SMS :

1. SME (*Short message Entity*), merupakan tempat penyimpanan dan pengiriman *message* yang akan dikirimkan ke MS tertentu.
2. SC (*Service Centre*), bertugas untuk menerima *message* dari SME dan melakukan *forwarding* ke alamat MS yang dituju.
3. SMS-GMSC (*Short message Service Gateway SMC*), melakukan penerimaan*message*dari SC dan memeriksa parameter yang ada. Selain itu GMSC juga mencari alamat MS yang dituju dangan bantuan HLR, dan mengirimkannya kembali ke MSC yang dimaksud.
4. SMS IWMSC (*Short Message Service Interworking MSC*), berperan dalam SMS *Message Origiating*, yaitu menerima pesan dari MSC.
5. *Home Location Register* (HLR) merupakan sebuah database yang digunakan sebagai tempat penyimpanan permanen data dan profil pelanggan. Bila diminta oleh SMSC, maka HLR dapat memberikan informasi *routing* dari pelanggan tertentu. HLR juga dapat memberikan informasi status tujuan apakah aktif atau tidak.
6. *Visitor Location Register* (VLR) merupakan sebuah database tempat menyimpan informasi sementara yang berisi data pelanggan dari sebuah HLR yang *roaming* pada HLR lain.
7. MSC merupakan sebuah sistem yang melakukan fungsi *switching* dan mengontrol panggilan telepon dalam sebuah jaringan komunikasi bergerak. MSC inilah yang akan mengirimkan sebuah *short message* kesuatu tujuan tertentu melalui *base station* yang sesuai.

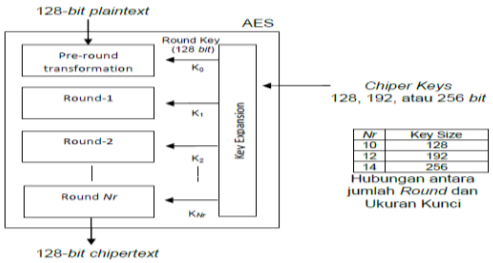
8. *Base Station Sistem* (BSS) Merupakan kesatuan sistem yang bertanggung jawab mengatur*transmisi* sinyal *elektromagnetik* untuk membawa data dari MSC ke perangkat telepon bergerak.*Base Station* terdiri dari *Base Station Controler* (BSC) dan *Base Tranceiver Station* (BTS) dan juga dikenal dengan nama *cell cite* atau sederhananya *cell*. Sebuah BSC biasanya menangani satu atau lebih BTS dan bertanggung jawab menangani pelanggan saat berpindah dari satu *cell* ke *cell* lainnya.
9. *Mobile Device* merupakan perangkat yang mempunyai kemampuan mengirimkan dan menerima *short message*, biasanya berupa telepon seluler dengan teknologi digital. Akan tetapi, saat ini jenis terminal berkembang sesuai aplikasi dan kebutuhan seperti POS, laptop dan *Personal Digital Assistant* [2].

1.3. Algoritma Rijndael

Algoritma kriptografi bernama Rijndael pertama kali di desain oleh Vincent Rijmen dan John Daemen asal Belgia, dan keluar sebagai pemenang dalam kontes algoritma kriptografi pengganti DES yang diadakan oleh NIST (National Institutes of Standards and Technology) milik pemerintah Amerika Serikat pada 26 November 2001. Algoritma Rijndael kemudian dikenal dengan Advanced Encryption Standard (AES). Setelah mengalami beberapa proses standardisasi oleh NIST, Rijndael kemudian diadopsi menjadi standard algoritma kriptografi secara resmi pada 22 Mei 2002. Pada 2006, AES merupakan salah satu algoritma terpopuler yang digunakan dalam kriptografi kunci simetrik.[4]

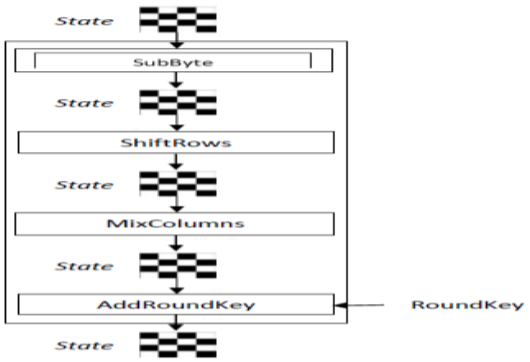
NIST memberikan spesifikasi ukuran blok harus 128-*bit* dan dapat memiliki ukuran panjang kunci 128-*bit*, 192-*bit*, dan 256-*bit*. Persyaratan lainnya adalah seluruh rancangan algoritma harus publik (tidak dirahasiakan). Pengumuman dibuat secara internasional untuk mendapat respon dari seluruh dunia.

AES adalah *non-feistel cipher* yang mengenkripsi dan mendepenelitian blok data 128-*bit*. AES menggunakan 10, 12, atau 14 putaran. Panjangkunci dapat berukuran 128-*bit*, 192-*bit*, atau 256-*bit* tergantung dari jumlah putaran. Gambar 2.4 menunjukan desain secara umum untuk algoritma enkripsi (disebut *cipher*). Pada gambar 2.4 *Nr* mendefinisikan jumlah putaran dan ukuran panjang kunci. Hal ini yang mengklasifikasikan 3 versi AES yaitu AES-128, AES-192, dan AES-256.[4]



Gambar 2. Ukuran Panjang Kunci

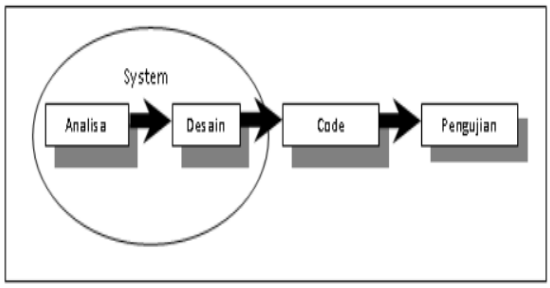
Struktur dari setiap putaran pada saat enkripsi ditunjukan gambar 2.. setiap putaran, kecuali putaran terakhir, terdapat empat tranformasi yang dapat dibalikkan. Namun pada putaran terakhir hanya terdapat tiga transformasi. [4]. Setiap transformasi menggunakan *state* dan menghasilkan *state* lain yang digunakan untuk transformasi selanjunya atau putaran selanjutnya. Sebelum putaran dimulai (*pre-round*) hanya ada satu transformasi (*AddRoundKey*), saat putaran terakhir hanya terdapat tiga transformasi (tanpa transformasi *MixColumns*).



Gambar 3. Tranformasi Mix Columns

II. METODOLOGI

Metodologi penelitian ini digunakan sebagai pedoman peneliti dalam pelaksanaan penelitian agar hasil yang dicapai tidak menyimpang dari tujuan yang sudah ditetapkan adalah dengan menggunakan metodologi waterfall, dimana metodologi waterfall ini mempunyai beberapa tahapan yakni: analisa, desain , pengujian dan impelemntasi.



Gambar 4. Metodologi Penelitian

1. Tahap Analisa

Langkah ini merupakan analisa terhadap suatu kebutuhan sistem yang dibangun. Pengumpulan data dalam tahap ini bisa dilakukan dengan beberapa cara antara lain wawancara atau studi literature. Pada tahapan ini intinya merupakan tahap dimana inisialisasi pendefinisian masalah untuk penyelesaian teknis pengembangan perangkat lunak mulai dilakukan. Disini hal yang dilakukan adalah menganalisa kebutuhan user akan keamanan data dari Aplikasi SMS yang digunakan pada smartphone.

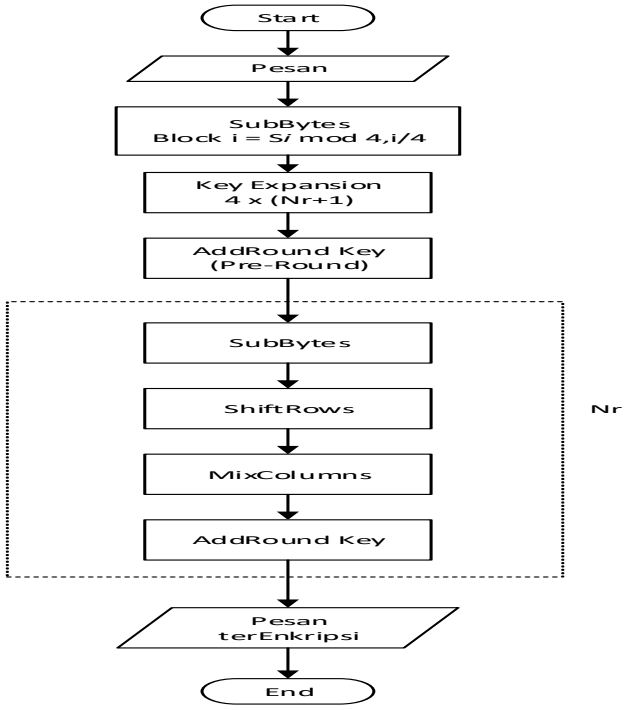
2. Tahap Desain Sistem

Pada Tahapan dimana dilakukan perancangan sistem terhadap solusi dari permasalahan yang ada dengan menggunakan perangkat pemodelan sistem seperti diagram alir data (dataflowdiagram), diagram hubungan entitas (Entity Relationship Diagram), struktur dan perancangan interface dari aplikasi yang dibangun.

3. Tahap Penulisan Code Program

Fase dimana dilakukan penambahan fitur dari aplikasi yang sudah ada menjadi aplikasi jadi yang sudah terintegrasi dengan metode enkripsi yang digunakan sesuai keinginan user. Dan juga dilakukan

Tahapan pemeriksaan eksekusi bagian program yang dibuat apakah sesuai dengan analisa yang didapat. Pada tahap penulisan code program ini diterapkan algortima kriftografi rijndael dengan beberapa tahapan proses yang harus dilakukan. Metode yang diterapkan dengan melakukan inkripsi dan dekripsi dari pesan pesan SMS yang dikirimkan dan diterima oleh user. Penerapan Algortima Rijndael yang digunakan beroperasi pada blok 128-bit dengan kunci 128-bit, dengan melakukan beberapa tahapan proses sesuai dengan diagram alir sebagai berikut:



Gambar 5. Proses Inkripsi Pesan SMS

```
// encryption SMS
cipher.init(Cipher.ENCRYPT_MODE,
key);

byte[] cipherText = new
byte[cipher.getOutputSize(input.length)];
int ctLength = cipher.update(input, 0,
input.length, cipherText, 0);
ctLength += cipher.doFinal(cipherText,
ctLength);
System.out.println(new String(cipherText));
System.out.println(ctLength);
```

Pada Diagram alir tersebut diterapkan untuk pengamanan SMS dengan rincian proses yang akan dilakukan sebagai berikut:

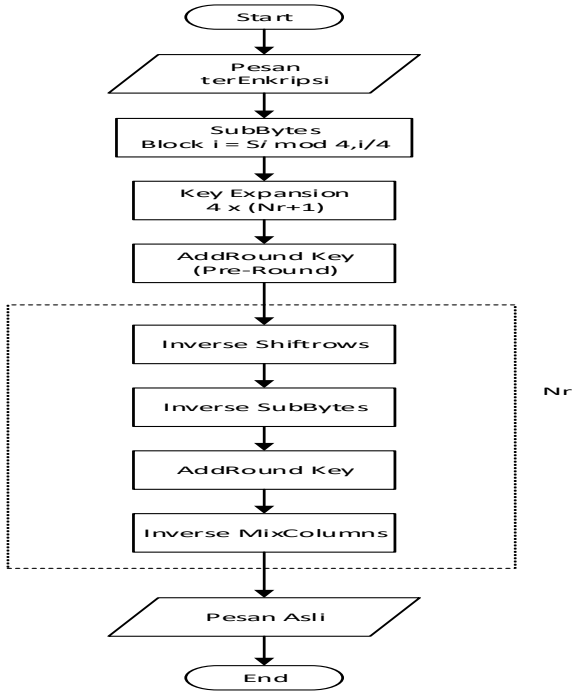
1. *Proses AdRoundKey*
   melakukan *XOR* antara *state* awal (plainteks) dengan *cipher key*. Tahap ini disebut juga *initial round*.
2. Proses Putaran sebanyak *Nr* – 1 kali. Proses yang dilakukan pada setiap putaran adalah:
   - *SubBytes*: substitusi *byte* dengan menggunakan tabel substitusi (*S-box*).
   - *ShiftRows*: pergeseran baris-baris *array state* secara *wrapping*.
   - *MixColumns*: mengacak data di masing-masing kolom *array state.*

3. Proses *AddRoundKey*: melakukan XOR antara *state* sekarang di dalam *round key*.
4. *Final round*: proses untuk putaran terakhir:
   - o *SubBytes*
   - o *ShiftRows*
   - *AddRoundKey*

Selama kalkulasi plainteks menjadi cipherteks, status yang digunakan dari data disimpan di dalam s*array of bytes* dua dimensi, state, yang berukuran NROWS dan NCOLS. [5]

- Untuk blok data 128-bit, ukuran state adalah 4 4.
- Elemen *array* state diacu sebagai *S*[*r*,*c*], 0 £ *r* < 4 dan 0 £ *c* < *Nb* (*Nb* adalah panjang blok dibagi 32.
- Pada *AES*-128, *Nb* = 128/32 = 4

Sedangkan pada proses dekripsi terhadap pesan SMS yang telah dikrimkan oleh user, dilakuakn proses dekripsi dengan diagram alir sebagai berikut:



Gambar.6. Proses Dekripsi Pesan SMS

```
// decryption SMS
cipher.init(Cipher.DECRYPT_MODE, key);
byte[] plainText = new byte[cipher.getOutputSize(ctLength)];
int ptLength = cipher.update(cipherText, 0, ctLength, plainText, 0);
ptLength += cipher.doFinal(plainText, ptLength);
System.out.println(new String(plainText));
System.out.println(ptLength);
}
```

4. Tahap Pengujian Program

Tahapan akhir dimana sistem yang baru diuji kemampuan dan keefektifannya sehingga didapatkan kekurangan dan kelemahan sistem yang kemudian dilakukan pengkajian ulang dan perbaikan terhadap aplikasi sehingga menjadi lebih baik dan sempurna. Pengujian dilakukan dengan pengecekan apakah plainteks yang diperoleh pada saat dekripsi sama dengan plainteks yang sebelum dienkripsikan, dimana akan diuji peng-*input*-an plainteks dengan berbagai variasi karakter yang dimasukkan pada pesan SMS (Short Message Service)

III. HASIL DAN PEMBAHASAN

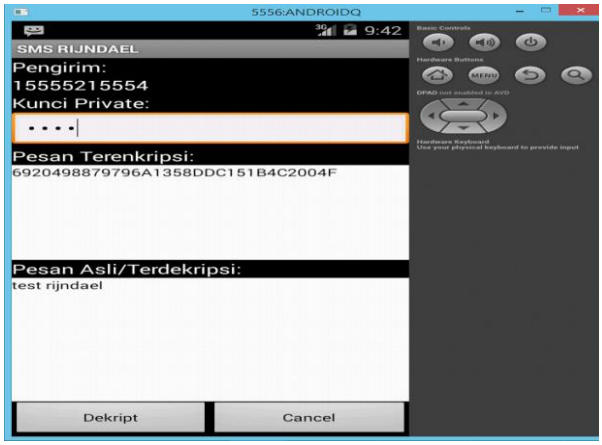
3.1. Hasil Aplikasi

Untuk mengamankan pesan SMS dengan menggunakan algortima rijndael dilakukan dengan proses inkripsi dan depenelitian pada pesan yang akan dikirimkan. Berikut adalah tampilan pesan terkirim setelah dilakukan proses depenelitian:

Gambar 7. Interface Pesan Terkirim

Gambar 7. merupakan tampilan yang akan muncul pada saat pesan diterima. Disana terlihat bahwa pesan yang diterima merupakan pesan yang sudah terenkripsi. Dan pengirimnya dari emulator 5554 yang dapat dilhat pada 4 digit terakhir di pengirim. Untuk dapat membaca pesan yang terenkripsi maka perlu dilakukan proses dekripsi terlebih dahulu. Proses dekripsi dilakukan dengan

memasukkan kunci privat yang sudah disepakati antara pengirim dan penerima.



Gambar 8. Inerface Membaca Pesan

Untuk melakukan uji coba pada Mobile diperlukan 2 mobile phone yang sudah terinstal aplikasi SMS rijndael. Pada kasus ini terdapat 2 mobile phone yang digunakan yaitu sony dan samsung dimana sony yang digunakan sebagai sisi pengirim. Dan samsung yang digunakan sebagai sisi penerima

Gambar 9. Interface Sisi Penerima

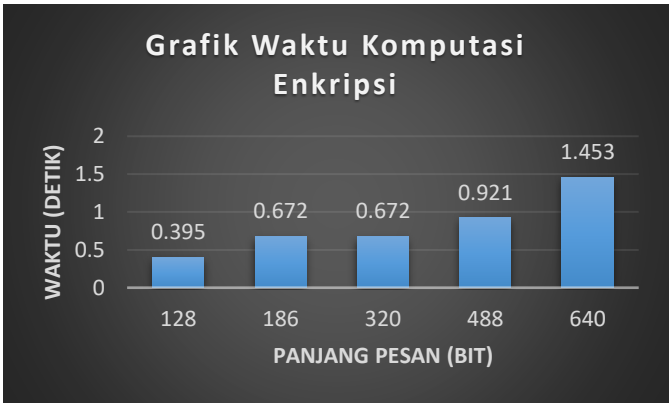
Untuk dapat membaca pesan yang terenkripsi maka perlu dilakukan proses dekripsi terlebih dahulu. Proses dekripsi dilakukan dengan memasukkan kunci privat yang sudah disepakati antara pengirim dan penerima.

Gambar 10. Proses Dekripsi **Pesan Mobile**

Pada gambar 10. dapat dilihat bahwa setelah memasukan kunci privat dan menekan tombol dekript didapatkan pesan asli seperti yang dikirim. Proses dekripsi akan berhasil apabila kunci privat yang dimasukkan sama dengan kunci privat yang digunakan pengirim untuk mengenkripsi pesan.

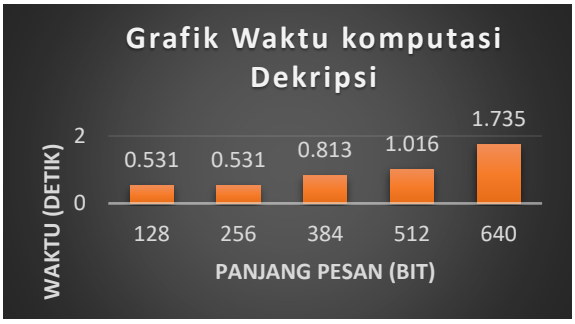
3.2. Hasil Pengujian Pesan SMS

Testing merupakan tahap yang dilakukan setelah menyelesaikan tahap pembuatan (Assembly) dengan cara menjalankan sebuah aplikasi atau program yang telah dibuat dan melihatnya kembali apakah ada kesalahan atau tidak. Testing terlebih dahulu dilakukan dilingkungan pembuat sendiri dimana pengujian aplikasi dilakukan. Berikut adalah grafik yang akan memperlihatkan pengaruh panjang pesan terhadap waktu komputasi.



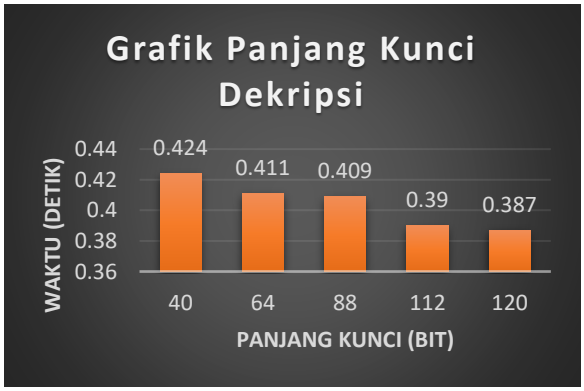
Gambar 11. Grafik Panjang Pesan Terhadap Waktu Komputasi Enkripsi

Dari gambar 11 dan 12 dapat dilihat bahwa semakin panjang pesan maka waktu komputasi akan semakin lama. Hal ini dibuktikan dengan bentuk grafik yang semakin naik.



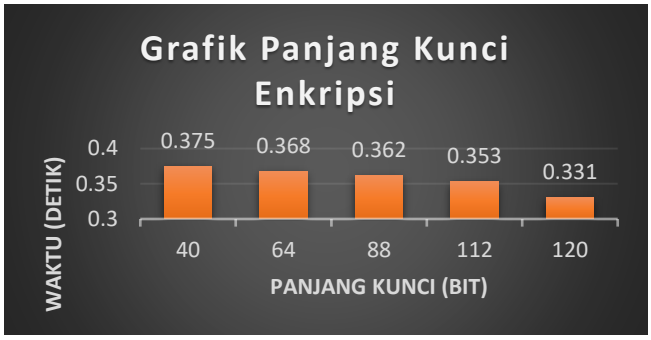
Gambar 12. Grafik Panjang Pesan Terhadap Waktu Komputasi Dekripsi

Pada hasil pengujian Variasi panjang kunci terhadap panjang pesan yang sama dapat diketahui panjang kunci 40 *bit* dengan panjang pesan 22 *byte* (186 *bit*).Berikut adalah grafik Pada hasil pengujian dengan menerapkan sejumlah 5 variasi panjang kunci dan panjang pesan yang berbeda terlihat bahwa waktu komputasi yang dihasilkan akan semakin sedikit/turun seiring dengan semakin banyaknya jumlah panjang kunci yang dimasukkan. Hal ini dikarenakan oleh proses penambahan bit pengganjal yang banyak pada saat panjang kunci lebih sedikit.



Gambar 14. Grafik Panjang Pesan Terhadap Waktu Komputasi Dekripsi

Berdasarkan analisis dan uji coba pada gambar 12 dimana dilakukan pengujian terhadap panjang pesan berbeda dengan panjang kunci yang yang sama didapatkan hasil yang menunjukkan bahwa semakin panjang pesan yang digunakan maka waktu komputasi yang dibutuhkan akan semakin lama. Sedangkan pada poin 3.2 dimana dilakukan pengujian dengan panjang pesan sama dan panjang kunci yang berbeda didapatkan hasil yang menunjukkan bahwa waktu komputasi yang diperlukan akan yang akan memperlihatkan pengaruh panjang kunci terhadap waktu komputasi.



Gambar 13. Grafik Panjang Kunci Terhadap Waktu Komputasi Enkripsi

semakin sedikit seiring dengan semakin panjang kunci yang dimasukkan.

IV. KESIMPULAN

Berdasarkan hasil penelitian yang dilakukan, maka dapat ditarik kesimpulan sebagai berikut:

1. Algoritma kriptografi Rijndael merupakan algoritma kriptografi yang sangat sensitive, dimana tiap karakter masukan akan menghasilkan keluaran yang berbeda sehingga sangat baik untuk mengamankan pengiriman SMS yang bersifat rahasia pada *mobile smartphone*.
2. Panjang kunci yang dimasukkan pada algorithma kriptogrfi Rijndael akan mempengaruhi jumlah round dan keluaran hasil enkripsi yang akan dikirimkan melalui SMS, serta akan berpengaruh terhadap waktu komputasi proses enkripsi dan dekripsi SMS.
3. Pengiriman SMS dengan mengunakan panjang kunci yang sama (128 bit) dan panjang pesan yang berbeda akan berbeda (dari rentang 128 bit sampai 640 bit) menghasilkan waktu komputasi yang berbeda-beda. Dimana semakin panjang pesan, maka waktu yang dibutuhkan untuk proses enkripsi dan dekripsi akan semakin lama.
4. Pengiriman SMS dengan menggunakan panjang pesan yang sama (186 bit) dan panjang kunci yang berbeda (mulai dari 40 bit sampai 120) bit menghasilkan waktu komputasi yang relative sama. Hal ini dikarenakan panjang kunci yang masih dalam cakupan 128, sehingga dalam proses enkripsi

dan dekripsi panjang kunci yang digunakan tetap 120 bit dengan jumlah putaran 10.

DAFTAR PUSTAKA

[1] Rosidi, Romzi Imron. (2004). "*Membuat Sendiri SMS Gateway Berbasis Protokol SMPP"*. Andi Yogyakarta

[2] Defni, Rahmayun Indri. (2014). "*Enkripsi SMS (Short Message Service) Pada Telepon Seluler Berbasis Android Dengan Metode RC6".* Jurnal Momentum Vol. 16 No. 1 ISSN : 1693-752X.

[3] Surian, Didi. (2006). "*Algoritma Kriptografi AES Rijndael"*. Jurnal Teknik Elektro TESLA Vol. 8 No. 2, 97 – 101.

[4] Ariyus, Dony. (2008). "*Pengantar Ilmu Kriptografi"*. Yogyakarta : Andi.

[5] Irawan, Arif Fajar. (2013). "*Sistem Keamanan Pesan pada Android Gingerbread (2.3.4) dengan Algoritma LUC".* Universitas Jember.