Penggabungan Algoritma Kriptografi Simetris dan Kriptografi Asimetris untuk Pengamanan Pesan

Andreas Dwi Nugroho (13511051)[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13511051@std.stei.itb.ac.id

***Abstrak*—Makalah ini membahas tentang rancangan penggabungan algoritma kriptografi simetris dan kriptografi asimetris. Dengan adanya penggabungan tersebut, dapat diperoleh skema pengamanan pesan yang menyediakan kelebihan dari kedua kriptografi tersebut yaitu proses enkripsi dan dekripsi yang cepat serta tidak perlu adanya distribusi kunci yang aman. Berdasarkan rancangan yang diusulkan, diimplementasikan rancangan tersebut dan dilakukan pengujian dengan membandingkan kinerjanya dengan sebuah algoritma kriptografi asimetris. Hasil pengujian menunjukan kinerja gabungan kedua jenis kriptografi ini lebih baik.**

***Kata kunci*—kriptografi simetris, kriptografi asimetris, ECC, AES, penggabungan**

I. PENDAHULUAN

Seiring dengan berkembangnya teknologi, layanan komunikasi semakin canggih dan memberikan kemudahan dalam komunikasi. Seperti misalnya layanan komunikasi yang sedang berkembang pesat saat ini yaitu *instant messaging*. Banyak orang yang menggunakan perangkat *mobile* untuk berkomunikasi menggunakan layanan *instant messaging*.

Dengan semakin banyaknya pertukaran pesan yang terjadi maka perlu adanya keamanan pesan terutama jika pesan tersebut bersifat privat atau rahasia. Salah satu cara yang dapat digunakan untuk mengamankan kerahasiaan pesan adalah dengan memanfaatkan peran kriptografi.

Kriptografi merupakan seni dan ilmu untuk menjaga keamanan pesan. Dalam menjaga keamanan pesan, kriptografi mengubah pesan jelas (plainteks) ke dalam bentuk pesan sandi (cipherteks). Enkripsi menjadikan pesan yang telah disandikan tersebut tidak dapat dimengerti dan dipahami isinya oleh pihak lain yang tidak berkepentingan. Untuk mengembalikan pesan sandi ke dalam pesan jelas dilakukan dengan proses dekripsi.

Kriptografi dapat dibedakan menjadi dua buah berdasarkan kunci yang ada, yaitu kriptografi simetris dan kriptografi asimetris. Kriptografi simetris memiliki kelebihan dimana proses enkripsi dan dekripsi berjalan cepat. Namun salah satu kelemahan dari kriptografi simetris adalah permasalahan distribusi kunci, karena kunci harus dikirim melalui saluran yang aman. Apabila seseorang telah mengetahui kunci enkripsi maka ia dengan mudah dapat mendekripsi pesan dan mengetahui isi dari pesan tersebut.

Sedangkan dalam kriptografi asimetris tidak ada kebutuhan mengirim kunci privat sebagaimana pada kriptografi simetris. Namun kriptografi asimetris memiliki kelemahan yaitu proses enkripsi dan dekripsi membutuhkan waktu yang lebih lama dan *resource* yang lebih besar. Oleh karena kebanyakan algoritma kriptografi asimetris kurang cocok untuk perangkat dengan keterbatasan *resource*.

Maka dari itu, dibutuhkan cara lain yang memberikan waktu enkripsi dan dekripsi yang cepat dan tidak adanya permasalah distribusi kunci yang aman. Dengan menggabungkan algoritma kriptografi simetris dan kriptografi asimetris memungkinkan untuk melakukan proses pengamanan pesan yang memanfaatkan kelebihan dari dua jenis kriptografi tersebut. Kelebihan yang bisa didapatkan antara lain proses enkripsi dan dekripsi yang cepat dan tidak perlu adanya distribusi kunci yang aman. Rancangan yang diusulkan berdasarkan hal tersebut yaitu pesan dienkripsi dengan sebuah kunci menggunakan kriptografi simetris dan selanjutnya kunci tersebut dienkripsi menggunakan kriptografi asimetris dengan kunci publik penerima.

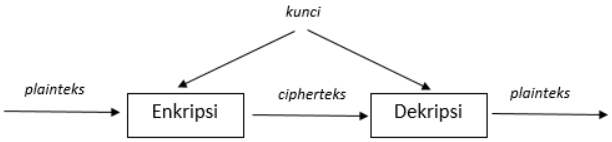
II. DASAR TEORI

A. Kriptografi Simetris

Kriptografi simetris adalah kriptografi yang menggunakan kunci yang sama untuk proses enkripsi dan dekripsi. Kriptografi simetris mengharuskan pengirim dan penerima menyetujui satu kunci tertentu sebelum dapat berkomuniksai secara aman. Keamanan kriptografi simetris bergantung pada kerahasiaan kunci. Jika kunci berhasil dipecahkan maka pesan yang terenkripsi dapat didekripsi dengan mudah.

Kelebihan dari kriptografi simetris adalah proses enkripsi dan dekripsi yang cepat. Namun kriptografi ini juga memiliki kelemahan, yaitu permasalahan distribusi kunci dan efisiensi jumlah kunci. Contoh algoritma kriptografi simetris adalah DES, Twofish, AES (Rijndael),

Blowfish, GOST, dan lain-lain.



Gambar 1 – Skema Kriptografi Simetris

AES (Advanced Encryption Standard)

AES adalah standar enkripsi atau dekripsi blok cipher dengan menggunakan kunci dengan panjang kunci 128 bit sampai 256 bit dengan step 32 bit. Panjang kunci dan ukuran blok dapat dipilih secara inpedenden. Penggunaan panjang kunci yang sering digunakan adalah 128-bit dan 256-bit. AES menggunakan algoritma Rjindael yang beroperasi dalam orientasi byte. AES memiliki kunci internal yang berbeda di setiap iterasinya yang disebut dengan *round key*. Kunci ini dibangkitkan dengan menggunakan *key schedule*.

Algoritma Rjindael dengan kunci 128-bit dijabarkan seperti berikut.

- *Initial round* - AddRoundKey: Melakukan xor terhadap plainteks dan key.
- *N-round* : melakukan putaran sebanyak N kali dengan melakukan 4 hal berikut :
  - SubBytes : Substitusi menggunakan S-Box berukuran 16x16
  - ShiftRow : pergeseran baris-baris *array state* secara *wrapping*
  - MixColumns : mengacak data di masing-masing kolom *array state*
  - AddRoundKey : melakukan xor antara *state* dengan *round key*
- *Final round* : melakukan SubBytes, ShiftRow, dan AddRoundKey

Banyaknya iterasi pada N-round bergantung kepada panjang kunci. Kunci dengan panjang 128-bit memiliki iterasi sebanyak 10 kali. Sedangkan kunci dengan panjang 256-bit memiliki iterasi sebanyak 14 kali.

SubBytes adalah substitusi dengan menggunakan S-Box. S-Box dirancang agar ai,j tidak sama dengan S(ai,j) dan hasil xor keduanya tidak sama dengan 0xFF. Pembangunan S-Box ini dilakukan menggunakan invers multiplikatif.

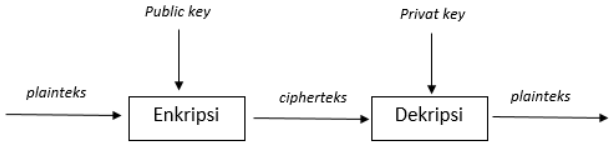
ShiftRow adalah pergeseran baris-baris pada array state sebanyak i kali ke kiri. Baris pertama (i=0) tidak digeser. Baris kedua (i=1) digeser sebanyak 1 kali. Begitu juga seterusnya hingga baris keempat (i=3).

MixColumns adalah pengkombinasian setiap kolom pada array state dengan melakukan perkalian terhadap sebuah matriks.

AddRoundKey adalah operasi xor terhadap array state dengan kunci yang bersangkutan. Pada initial round, operasi xor dilakukan menggunakan kunci masukan pengguna. Sedangkan pada N-Round digunakan kunci internal yang dibangun dari Key Schedule.

B. Kriptografi Asimetris

Kriptografi asimetris merupakan kriptografi yang menggunakan kunci yang berbeda untuk proses enkripsi dan dekripsi pesan. Pada kriptografi asimetris, kunci dibuat sepasang yaitu satu kunci untuk enkripsi dan satu kunci untuk dekripsi. Kunci untuk enkripsi dapat diketahui oleh publik dan bersifat tidak rahasia sehingga dinamakan kunci publik. Sedangkan kunci untuk dekripsi bersifat rahasia dan hanya diketahui oleh si pemilik kunci sehingga dinamakan kunci privat. Contoh algoritma dari kriptografi asimetris adalah RSA, ElGamal, dan Elliptic Curve Cryptography (ECC). Kriptografi asimetris mendasarkan keamanannya pada persoalan matematis yang dianggap sulit dipecahkan, sehingga secara komputasi hampir tidak mungkin menurunkan kunci privat dari kunci publik.



Gambar 2 – Skema Kriptografi Asimetris

Elliptic Curve Cryptography

Elliptic Curve Cryptography (ECC) merupakan salah satu pendekatan kriptografi asimetris yang mendasarkan keamanannya pada persoalan logaritma diskrit dari kurva eliptik bidang terbatas. Algoritma ECC dapat digunakan pada perangkat yang memiliki keterbasan *resource* seperti kemampuan penyimpanan dan pemrosesan. Selain itu, ECC memiliki keuntungan dengan panjang kunci yang lebih kecil dibandingkan kriptografi asimetris lainnya dapat menyediakan tingkat keamanan yang relatif sama.

ECC ditemukan oleh Neal Koblitz dan Victor Miller. ECC menggantikan Zp* pada logaritma diskrit dengan sekelompok titik pada kurva elips. Sebuah kurva elips E pada Zp didefinisikan dengan persamaan

$$y2 = x3 + ax + b$$

dengan a,b ε Zp dan $4a3 + 27b2 \neq 0 (\text{mod } p)$, dengan titik O yang dinamakan point pada titik tak hingga. Set E(Zp) terdiri dari titik (x,y), x, y ε Zp, yang memenuhi persamaan diatas.

Setiap nilai dari a dan b memberikan kurva elips yang berbeda. *Public key* adalah titik dari kurva dan *private key* adalah bilangan random. Kunci privat didapatkan dari memilih secara random bilangan integer bukan nol pada grup order *n*. *Public key* didapatkan dengan mengalikan kunci privat dengan titik G pada kurva. *Public key* didapatkan dengan rumus Q=dG, dengan d adalah jumlah titik G pada kurva. Lalu *public key* dipilih secara random dari perhitungan Q pada titik G tersebut.

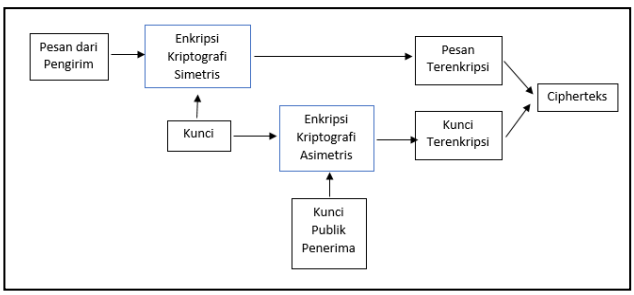
III. RANCANGAN SOLUSI

Dalam rancangan solusi yang diusulkan, digunakan penggabungan kriptografi simetris dengan kriptografi asimetris. Awalnya plainteks dienkripsi menggunakan algoritma kriptografi simetris dengan sebuah kunci. Kemudian kunci yang digunakan dalam proses enkripsi plainteks tersebut dienkripsi menggunakan algoritma kriptografi asimetris.

Penggunaan algoritma dalam enkripsi tersebut ditentukan berdasarkan karateristik setiap algoritma kriptografi serta pesan yang dienkripsi menggunakan algoritma tersebut. Algoritma kriptografi simetris memiliki kelebihan yaitu waktu dan *resource* yang dibutuhkan untuk melakukan enkripsi lebih sedikit dibandingkan dengan algoritma asimetris. Namun algoritma simetris memiliki kelemahan dalam keamanan distribusi kunci, sedangkan algoritma asimetris tidak.

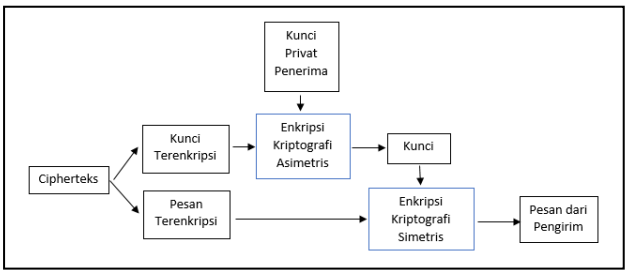
Berdasarkan hal tersebut, plainteks dienkripsi menggunakan algoritma kriptografi simetris karena pada umumnya ukuran plainteks lebih besar dibandingkan ukuran kunci. Semakin besar ukuran plainteks maka semakin lama proses enkripsi dan *resource* yang digunakan. Maka dari itu, untuk proses enkripsi plainteks digunakan algoritma kriptografi simetris karena kelebihan yang dimilikinya. Namun ketika menggunakan algoritma kriptografi simetris, maka distribusi kunci harus melalui saluran yang aman. Untuk menjaga kerahasiaan kunci maka kunci tersebut juga perlu dienkripsi. Kunci dienkripsi menggunakan algoritma kriptografi asimetris karena ukuran kunci biasanya tidak terlalu besar sehingga tidak terlalu berpengaruh terhadap waktu dan *resource* yang digunakan ketika melakukan proses enkripsi. Selain itu, karena menggunakan algoritma kriptografi asimetris maka tidak perlu memperhatikan keamanan distribusi kunci. Kunci yang digunakan untuk enkripsi pesan plainteks akan dienkripsi juga menggunakan kunci publik penerima yang bersifat publik sehingga tidak perlu menjaga kerahasiaan kunci publik penerima yang digunakan dalam algoritma asimetris.

Dari rancangan ini, maka cipherteks yang dihasilkan merupakan gabungan dari pesan yang terenkripsi menggunakan algoritma kriptografi simetris dan kunci yang terenkripsi menggunakan algoritma kriptografi asimetris. Secara umum, skema enkripsi pada rancangan solusi ini ditunjukkan pada Gambar 3.



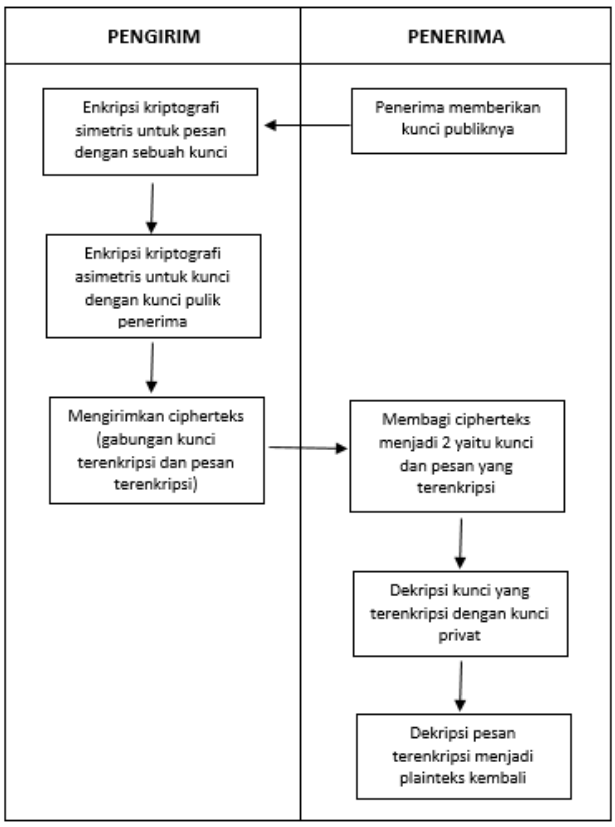
Gambar 3 - Skema Enkripsi

Sedangkan untuk proses dekripsi pada rancangan solusi ini, awalnya cipherteks dibagi menjadi 2 bagian yaitu kunci yang terenkripsi dan pesan yang terenkripsi. Kunci yang terenkripsi tersebut kemudian didekripsi menggunakan algoritma kriptografi asimetris dengan kunci privat penerima. Dari proses dekripsi ini akan dihasilkan sebuah kunci. Selanjutnya kunci tersebut digunakan dalam algoritma kriptografi simetris untuk mendekripsi pesan yang terenkripsi. Dan hasil dari proses dekripsi tersebut berupa plainteks dari pesan yang dikirimkan oleh seorang pengirim. Skema proses dekripsi ditunjukkan pada Gambar 4.



Gambar 4 - Skema Dekripsi

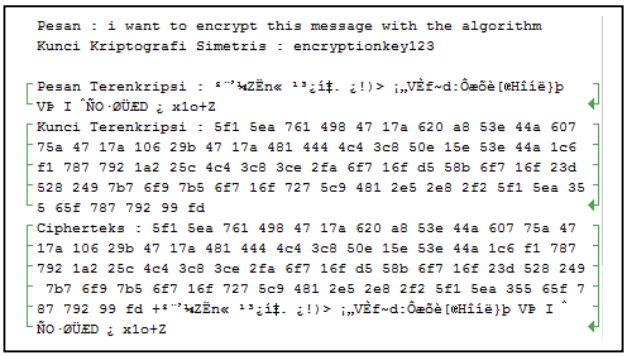
Berikut ini adalah skema prosedur pertukaran pesan dengan menggunakan rancangan solusi ini.



Gambar 5 - Skema Pertukaran Pesan

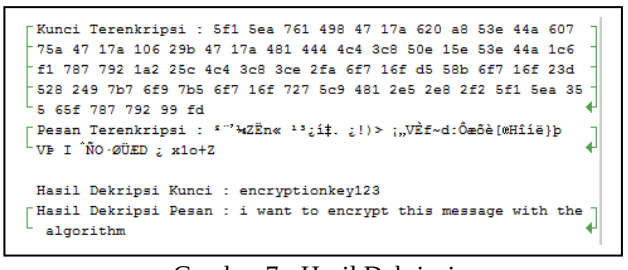
IV. IMPLEMENTASI DAN PEMBAHASAN HASIL

Berdasarkan rancangan yang diusulkan, dilakukan implementasi algoritma tersebut dengan menggunakan bahasa pemrograman Java. Algoritma kriptografi simetris yang diimplementasikan yaitu algoritma AES. Sedangkan algoritma kriptografi asimetris yang diimplementasikan adalah ECC. Setelah dilakukan implementasi, selanjutnya dilakukan pengujian. Pengujian tersebut dilakukan dengan mengukur kecepatan dan ukuran yang dihasilkan dari melakukan enkripsi serta dekripsi yang menggunakan gabungan algoritma ECC dan AES dengan algoritma ECC seluruhnya. Contoh hasil pengujian proses enkripsi dan dekripsi ditunjukkan pada Gambar dan Gambar



Gambar 6 - Hasil Enkripsi

Dalam pengujian proses enkripsi digunakan pesan dengan panjang 48 byte dan kunci untuk algoritma kriptografi simetris sepanjang 16 byte. Terlihat bahwa cipherteks yang dihasilkan merupakan gabungan antara kunci yang terenkripsi dengan algoritma kriptografi simetris (AES) dan pesan yang terenkripsi dengan algoritma kriptografi asimetris (ECC). Dari hasil yang diperoleh, panjang cipherteks menjadi lebih panjang karena ditambah dengan panjang kunci yang terenkripsi. Panjang cipherteks dari algoritma kriptografi simetris sama dengan panjang plainteks. Sedangkan dalam algoritma kriptografi asimetris, panjang cipherteks biasanya lebih panjang dibandingkan dengan panjang plainteks.



Gambar 7 - Hasil Dekripsi

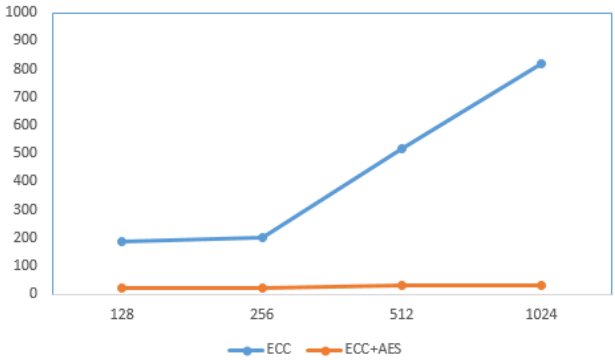
Untuk pengujian proses dekripsi, awalnya cipherteks dibagi menjadi dua bagian yaitu kunci terenkripsi dan pesan terenkripsi. Kemudian dilakukan dekripsi kunci dan pesan yang akan menghasilkan pesan atau plainteks awal.

Selanjutnya dilakukan pengujian untuk membandingkan kinerja dari algoritma saat melakukan proses enkripsi dan dekripsi antara ECC dan gabungan ECC-AES. Pengujian tersebut dilakukan dengan melakukan enkripsi dan dekripsi suatu pesan dengan berbagai ukuran panjangnya dan kemudian menghitung waktu yang dibutuhkan untuk melakukan hal tersebut serta ukuran cipherteks yang dihasilkan. Untuk hasil pengujian waktu kecepatan enkripsi ditunjukkan pada Tabel I.

TABLE I. WAKTU ENKRIPSI

Panjang Pesan	Waktu Enkripsi	
	ECC	ECC + AES (*16-byte key*)
128 bytes	188 ms	21 ms
256 bytes	202 ms	21 ms
512 bytes	518 ms	30 ms
1024 bytes	821 ms	31 ms
2048 bytes	15625 ms	91 ms

Dari tabel hasil pengujian tersebut, berikut adalah grafik perbandingan waktu kecepatan proses enkripsi antara keduanya.



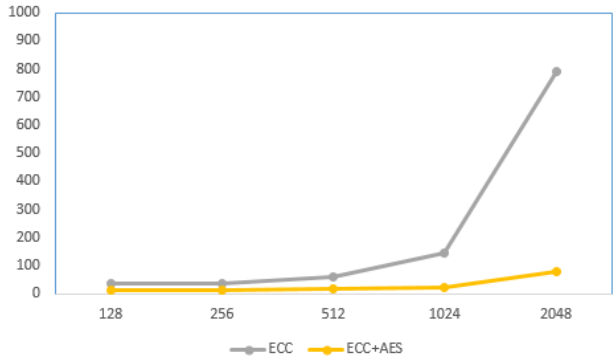
Gambar 8 - Perbandingan kecepatan enkripsi

Hasil pengujian waktu kecepatan dekripsi ditunjukkan pada Tabel II

TABLE II. WAKTU DEKRIPSI

Panjang Pesan	Waktu Dekripsi	
	ECC	ECC + AES (*16-byte key*)
128 bytes	34 ms	10 ms
256 bytes	37 ms	10 ms
512 bytes	60 ms	16 ms
1024 bytes	144 ms	22 ms
2048 bytes	792 ms	76 ms

Dari tabel hasil pengujian tersebut, berikut adalah grafik perbandingan waktu kecepatan proses dekripsi antara keduanya.



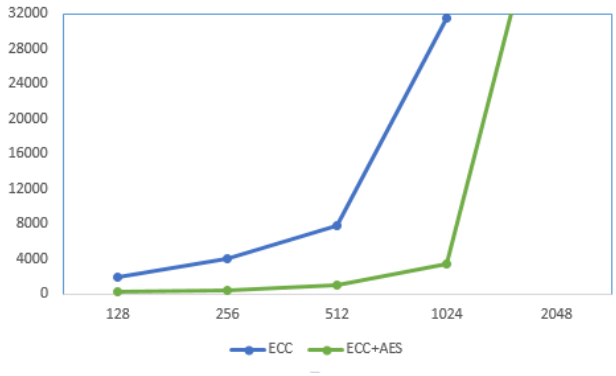
Gambar 9 - Perbandingan kecepatan dekripsi

Sedangkan hasil pengujian ukuran hasil enkripsi untuk keduanya ditunjukkan pada Tabel III

TABLE III. UKURAN CIPHERTEKS

Panjang Pesan	Ukuran Cipherteks	
	ECC	ECC + AES (*16-byte key*)
128 bytes	1950 bytes	263 bytes
256 bytes	3937 bytes	446 bytes
512 bytes	7853 bytes	1049 bytes
1024 bytes	31422 bytes	3480 bytes
2048 bytes	503093 bytes	51963 bytes

Berdasarkan tabel hasil pengujian tersebut, berikut adalah grafik perbandingan ukuran cipherteks hasil enkripsi antara keduanya.



Gambar 10 - Perbandingan ukuran hasil enkripsi

Dalam kriptografi, semakin panjang pesan, maka semakin banyak waktu yang digunakan untuk melakukan enkripsi dan dekripsi serta semakin besar ukuran hasil cipherteks. Perbandingan waktu enkripsi antara ECC dengan gabungan ECC dan AES menunjukkan bahwa terdapat perbedaan yang cukup signifikan. Untuk ECC, waktu kecepatan enkripsi terjadi peningkatan yang tinggi seiring dengan bertambahnya panjang pesan. Sedangkan pada gabungan ECC dan AES tidak terjadi peningkatan yang tinggi. Hal ini dikarenakan enkripsi pada ECC melibatkan operasi komputasi bilangan yang sangat besar. Karena digunakan untuk enkripsi seluruh isi pesan, maka waktu enkripsi meningkat secara tajam seiring bertambahnya jumlah pesan yang dienkripsi. Sedangkan pada gabungan ECC dan AES, operasi bilangan besar hanya dilakukan pada enkripsi kunci AES yang digunakan untuk mengenkripsi isi pesan. Karena panjang kunci biasanya lebih kecil daripada panjang pesan, maka semakin bertambahnya panjang pesan tidak akan terlalu berpengaruh pada waktu enkripsi. Peningkatan waktu enkripsi tersebut sebagian besar dipengaruhi oleh waktu enkripsi dengan AES yang memang lebih cepat dibandingkan dengan ECC.

Berdasarkan hasil perbandingan waktu dekripsi juga menunjukan bahwa waktu kecepatan dekripsi gabungan ECC dan AES lebih cepat dibandingkan dengan ECC. Hal ini dikarenakan pada ECC dekripsi pesan seluruhnya menggunakan ECC (kriptografi asimetris), sedangkan pada gabungan ECC dan AES dekripsi pesan dilakukan dengan AES (kriptografi simetris). Algoritma kriptografi simetris memiliki kelebihan yaitu waktu dekripsi lebih cepat dibandingkan dengan algoritma kriptografi asimetris.

Hasil pengujian ukuran hasil enkripsi juga menunjukan perbedaan antara keduanya. Cipheteks hasil enkripsi algoritma kriptografi asimetris biasanya lebih panjang dibandingkan dengan plainteks, sedangkan ukuran cipherteks dari enkripsi algoritma kriptografi simetris biasanya sama dengan ukuran plainteks. Karena pada ECC seluruh pesan dienkripsi dengan ECC dan pada gabungan ECC-AES seluruh pesan dienkripsi dengan AES maka ukuran cipherteks hasil enkripsi pada ECC lebih besar dibandingkan dengan hasil enkripsi pada gabungan ECC dan AES.

Dari hasil pengujian, maka penggunaan gabungan ECC dan AES memiliki kinerja yang lebih baik dibandingkan dengan ECC. Oleh karena itu, rancangan ini dapat digunakan untuk lingkungan penggunaan yang memiliki keterbatasan *resource* serta cocok digunakan untuk enkripsi pesan dengan ukuran yang besar. Perbandingan kedua hal tersebut pada penelitian berikutnya masih dapat dikembangkan, seperti menggunakan algoritma kriptografi simetris dan asimetris yang berbeda dengan yang digunakan dalam penelitian ini.

Dari penggabungan algoritma kriptografi simetris dan asimetris ini, diperoleh dua buah keuntungan. Pertama, karena menggunakan algoritama kriptografi simetris maka proses enkripsi dan dekripsi berjalan lebih cepat serta ukuran cipherteks yang dihasilkan tidak mengalami

peningkatan yang tajam seiring bertambahnya ukuran plainteks. Kedua, kelemahan dari algoritma kriptografi simetris terkait dengan keamanan distribusi kunci diatasi dengan memanfaatkan algoritma kriptografi asimetris. Kerahasiaan kunci dijaga dengan melakukan enkripsi pada kunci tersebut. Enkripsi kunci dilakukan dengan menggunakan algoritma kriptografi asimetris yang tidak membutuhkan distribusi kunci yang aman karena kunci bersifat publik serta cocok digunakan untuk enkripsi kunci yang panjangnya tidak terlalu besar sehingga tidak terlalu mempengaruhi kinerja.

V. KESIMPULAN

Kesimpulan yang diperoleh yaitu dengan adanya penggabungan algoritma kriptografi simetris dengan kriptografi asimetris dapat melakukan pengamanan pesan dengan cepat dan tanpa perlu distribusi kunci yang aman. Kriptografi simetris memiliki kelebihan dimana proses enkripsi/dekripsi lebih cepat namun harus menjaga keamanan kunci. Kriptografi asimetris memiliki kelebihan dimana tidak membutuhkan distribusi kunci yang aman namun proses enkripsi/dekripsi membutuhkan resource yang lebih besar. Dari hasil penggabungkan keduanya dapat diperoleh algoritma kriptografi yang lebih cepat dan tidak memerlukan distribusi kunci yang aman.

Berdasarkan hasil pengujian, gabungan ECC (kriptografi asimetris) dan AES (kriptografi simetris) memiliki kinerja yang lebih baik dibandingkan dengan penggunaan ECC keseluruhan dalam aspek kecepatan enkripsi-dekripsi dan ukuran cipherteks yang dihasilkan. Hal ini tentunya sangat sesuai digunakan untuk perangkat dengan resource yang terbatas.

REFERENSI

[1] Rijndael Algorithm (Advanced Encryption Standard) AES: https://www.lri.fr/~fmartignon/documenti/systemesecurite/5-AES.pdf, tanggal akses: 11 Maret 2015.

[2] Munir, Rinaldi. Diktat dan Materi Kuliah Kriptografi. Bandung: Penerbit Informatika, 2006.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 11 Mei 2015

Andreas Dwi Nugroho (13511051)