Aplikasi Enkripsi *Instant Messaging* Pada Perangkat *Mobile* Dengan Menggunakan Algoritma *Elliptic Curve Cryptography* (ECC)

Andreas Dwi Nugroho
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Bandung, Indonesia
andreasdwin@gmail.com

Rinaldi Munir
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Bandung, Indonesia
rinaldi@informatika.org

***Abstrak*—Saat ini aplikasi *instant messaging* sudah banyak dipakai oleh orang untuk berkomunikasi. Dengan semakin banyaknya penggunaan *instant messaging* maka perlu adanya keamanan pesan yang dikirimkan karena pesan tersebut rawan terhadap penyadapan. Salah satu cara untuk menjaga keamanan pesan tersebut adalah dengan memanfaatkan kriptografi untuk mengenkripsi pesan sehingga pesan tidak mudah untuk dibaca.**

Pada penelitian ini dilakukan pembangunan aplikasi *instant messaging* yang mengimplementasikan ECC untuk pengamanan pesan. Untuk mengenkripsi dan mendekripsi pesan digunakan algoritma ECC ElGamal. Aplikasi ini diimplementasikan pada perangkat *mobile* Android. Hasil pengujian aplikasi menunjukkan bahwa aplikasi ini dapat menjaga keamanan pesan dari tindakan penyadapan. Berdasarkan hasil proses penyadapan, didapat bahwa pesan yang didapat sudah dalam bentuk cipherteks sehingga tidak mudah dibaca.

***Kata kunci—instant messaging,* enkripsi, perangkat *mobile,* ECC**

I. LATAR BELAKANG

Saat ini, *instant messaging* merupakan salah satu bentuk layanan komunikasi yang sedang berkembang pesat. *Instant messaging* adalah sebuah layanan komunikasi yang memungkinkan penggunanya untuk mengirimkan pesan singkat secara *real time* melalui jaringan internet. Beberapa contoh aplikasi *instant messaging* saat ini antara lain Whatsapp, Blackberry Messenger (BBM), Line, dan lain sebagainya. Jika beberapa tahun yang lalu pengguna saling bertukar pesan melalui layanan SMS, sekarang ini banyak pengguna perangkat mobile yang tertarik menggunakan layanan instant messaging untuk saling berkomunikasi karena kelebihan yang dimilikinya. Banyaknya penggunaan layanan *instant messaging* kini sudah mengalahkan penggunaan layanan SMS. Menurut data perusahaan riset Informa, pada tahun 2012 jumlah pesan yang dikirimkan melalui layanan *instant messaging* mencapai 19 miliar melampaui jumlah pesan yang dikirimkan melalui SMS yang berjumlah 17,6 miliar [1].

Dengan semakin banyaknya penggunaan aplikasi *instant messaging* menyebabkan perlu adanya keamanan pesan terutama pesan yang bersifat rahasia. Pesan-pesan instan dalam aplikasi *instant messaging* sangat rawan terhadap penyadapan. Salah satu cara yang dapat digunakan untuk mengamankan pesan yang dikirim melalui aplikasi *instant messaging* adalah dengan memanfaatkan peran kriptografi untuk mengenkripsi pesan sebelum pesan tersebut dikirim. Dengan dienkripsinya pesan yang akan dikirim, maka pesan tersebut tidak akan mudah diketahui isinya oleh pihak yang tidak diinginkan.

Berdasarkan kuncinya, kriptografi dibedakan menjadi dua, yaitu kriptografi simetris dan kriptografi asimetris. Kekuatan pengamanan dengan sistem enkripsi sangat bergantung pada keamanan kunci. Karena kriptografi asimetris mempunyai kelebihan, yaitu distribusi kunci tidak perlu melalui saluran yang aman sebagaimana pada kriptografi simetris, maka digunakanlah kriptografi asimetris.

Pendekatan kriptografi kunci asimetris yang digunakan adalah ECC. ECC memiliki kelebihan yaitu memberikan tingkat keamanan yang sama dengan algoritma kriptografi kunci asimetris lainnya namun kunci yang digunakan yang lebih pendek.Dengan ukuran panjang kunci yang lebih pendek maka operasi komputasi kriptografi yang dilakukan lebih cepat dan penggunaan *resource* juga lebih kecil sehingga cocok digunakan untuk perangkat *mobile* yang memiliki keterbatasan *resource* [2].

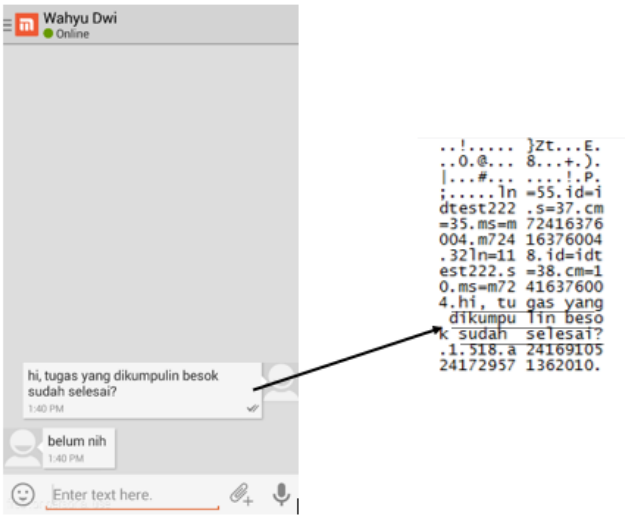
II. DASAR TEORI

A. Analisis Keamanan Aplikasi IM

Seiring dengan semakin banyaknya penggunaan layanan *instant messaging* sebagai sarana komunikasi, maka aspek keamanan menjadi sangat penting untuk dipertimbangkan. Hal itu dikarenakan pihak lain yang tidak berkepentingan dapat memata-matai komunikasi yang yang dilakukan. Sebagian besar alat komunikasi yang mudah digunakan oleh orang tidak mempertimbangkan aspek keamanan yang tinggi. Menurut Electronic Frontier Foundation (2015), terdapat 7 macam

kriteria dalam menentukan keamanan dalam aplikasi *instant messaging* [3]. Ketujuh kriteria harus dipenuhi untuk mendapatkan tingkat keamanan yang tinggi dari segala aspek. Salah satu kriteria tersebut adalah apakah komunikasi yang dilakukan antar pengguna telah terenkripsi selama pengiriman dalam jalur komunikasi. Saat ini, hampir semua aplikasi IM telah memenuhi kriteria tersebut. Hal ini setidaknya bisa mencegah terjadinya serangan penyadapan (*eavesdropping*) sehingga pihak ketiga tidak mudah untuk membaca pesan dalam komunikasi yang dilakukan.

Salah satu contoh aplikasi yang belum menerapkan kriteria tersebut yaitu Mxit. Setelah dilakukan penyadapan dengan menangkap paket data menggunakan Wireshark, hasil yang didapatkan menunjukkan bahwa komunikasi yang dilakukan dengan aplikasi dapat dengan mudah dibaca oleh orang lain. Hasil percobaan tersebut dapat dilihat pada Gambar 1.



Gambar 1. Isi percakapan dan hasil penyadapan

Hal tersebut tentunya sangat berbahaya jika pesan bersifat rahasia. Oleh karena itu, perlu adanya enkripsi pesan sehingga pesan tidak mudah dibaca oleh orang lain yang tidak berkepentingan.

B. *Elliptic Curve Cryptography*

Elliptic Curve Cryptography (ECC) merupakan salah satu pendekatan kriptografi kunci asimetris yang mendasarkan keamanannya pada persoalan logaritma diskrit dari kurva eliptik bidang terbatas. Salah satu kegunaan ECC yaitu skema enkripsi yang menggunakan algoritma ECC ElGamal. Pendekatan enkripsi-dekripsi tersebut dapat diimplementasikan dengan menggunakan kurva eliptik pada bidang terbatas F_p. Kurva eliptik tersebut didefinisikan dengan sebuah persamaan dalam bentuk:

$$y^2 = x^2 + ax + b \ (mod\ p) \qquad (1)$$

yang memenuhi:

$$4a^3 + 27b^2 \ (mod\ p) \neq 0 \qquad (2)$$

Dalam persoalan logaritma diskrit dari kurva eliptik, diberikan P dan Q yang merupakan dua buah titik di kurva eliptik, carilah integer k sedemikian sehingga $Q = kP$. Secara komputasi sulit untuk menemukan k jika k adalah bilangan yang besar. Bilangan k merupakan logaritma diskrit dari Q dengan basis P. Pada ECC, Q adalah kunci publik, k adalah kunci privat, dan P adalah sembarang titik pada kurva eliptik.

Dalam kriptografi kunci asimetris, harus ditentukan terlebih dahulu nilai parameter yang akan digunakan dan telah disepakati oleh pihak yang akan berkomunikasi. Parameter yang digunakan dalam ECC yaitu nilai a dan b, bilangan prima p dalam persamaan kurva eliptik bidang terbatas serta titik generator G yang dipilih dari kurva eliptik. Pendekatan enkripsi dengan ECC ini dapat dijelaskan dalam contoh kasus misalnya Alice ingin mengirim pesan yang terenkripsi kepada Bob [4].

1) *Pembangkitan Kunci Privat dan Kunci Publik*

Bob membangkitkan kunci privat n_B dengan cara memilih bilangan acak yang nilainya diantara [1,p-1]. Dengan kunci privat tersebut, Bob membangkitkan kunci publik $P_B = n_B \cdot G$.

2) *Enkripsi*

Misalnya pesan yang akan dikirim adalah pesan m. Alice meng-*encode* pesan m menjadi sebuah titik, P_m, dari kurva eliptik. Lalu memilih bilangan acak k yang nilai diantara [1,p-1]. Alice menghasilkan cipherteks, C_m, yang terdiri dari pasangan titik $C_m = \{(kG), (P_m+kP_B)\}$ dimana G adalah titik generator dan P_B adalah kunci publik Bob.

3) *Dekripsi*

Untuk melakukan dekripsi cipherteks C_m, Bob mula-mula mengalikan titik pertama dari cipherteks dengan kunci privatnya n_B dan kemudian mengurangkan titik kedua dari cipherteks dengan hasil perkalian tersebut.

$$P_m + kP_B - n_B(kG) = P_m + k(n_BG) - n_B(kG) = P_m$$

Lalu Bob men-*decode* P_m menjadi pesan m semula.

C. Penelitian Terkait

Terdapat beberapa penelitian terkait yang berhubungan dengan keamanan pada *instant messaging*. Dalam [5] dilakukan perancangan arsitektur yang aman untuk aplikasi *private instant messenger*. Arsitektur tersebut menggunakan modul *secure* yang berfungsi untuk melindungi data berupa pesan yang ditransmisikan dengan cara melakukan enkripsi pada pesan tersebut. Modul tersebut mengimplementasikan algoritma hashing (SHA) dengan enkripsi asimetris dalam proses enkripsi pesan.

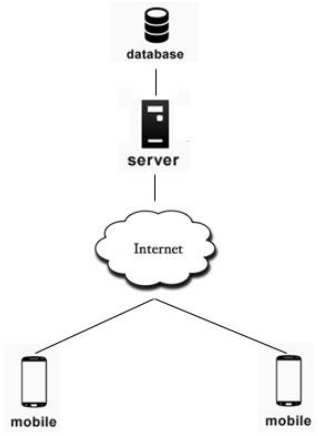
Dalam [6] dibahas mengenai rancangan metode untuk mengamankan pesan *instant messaging* dengan menggunakan sebuah metode otentifikasi yang efisien. Otentifikasi tersebut dilakukan untuk menjamin validitas data sehingga orang yang tidak berkepentingan tidak dapat mengubah isi pesan asli yang dikirimkan. Metode otentifikasi ini mengimplementasikan algoritma *Hyper Elliptic Curve Cryptosystem* (HECC) untuk membangkitkan dan memverifikasi tanda tangan digital dari pesan.

Pada penelitian [7] dilakukan perancangan arsitektur untuk *instant messaging* yang aman dilengkapi dengan sebuah fitur *self-message-destructing* untuk informasi yang sensitif. Dalam arsitektur tersebut, sistem akan mengirimkan pesan *self-destructible* kepada penerima dalam bentuk pesan terenkripsi. Maksud dari pesan *self-destructible* tersebut adalah pengguna bisa menentukan batasan waktu, frekuensi, dan lokasi yang dikirimkan bersama pesan tersebut dan jika batasan tersebut dipenuhi maka kunci yang digunakan untuk deskripsi akan dihapus sehingga pesan enkripsi yang dikirim tidak bisa dideskripsi untuk bisa dibaca. Pesan *self-destructible* tersebut terdiri dari isi pesan yang terenkripsi oleh sebuah kunci sementara, batasan yang ditentukan oleh pengguna, serta kunci sementara yang digunakan untuk mendeskripsi isi pesan. Batasan dan kunci sementara dalam pesan tersebut juga dienkripsi dengan menggunakan kunci publik penerima pesan. Dalam proses enkripsi yang dilakukan, isi pesan dienkripsi menggunakan enkripsi simetris AES 128-bit dan menggunakan enkripsi asimetris OpenPGP dengan RSA 1024-bit untuk mengenkripsi kunci AES dan batasan yang dimasukkan oleh pengguna dalam sebuah pesan *self-destructible*.

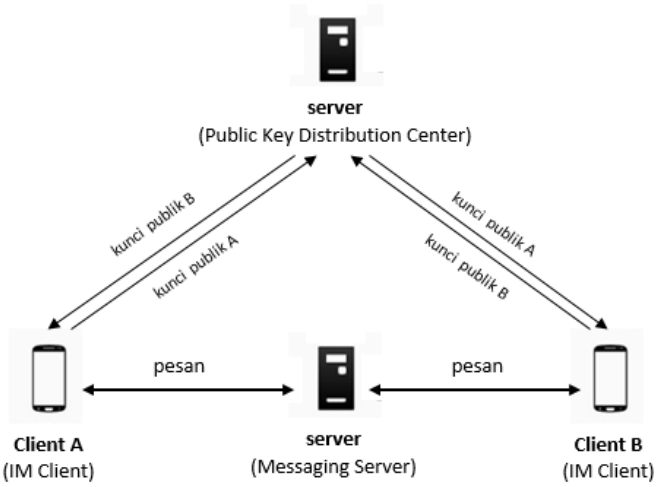
III. ARSITEKTUR SISTEM

Secara umum, arsitektur yang digunakan dalam pembangunan perangkat lunak yaitu *client-server*. Arsitektur ini dapat dilihat pada Gambar 2.

Dalam arsitektur sistem ini, ada 2 buah komponen utama yaitu *client* dan *server*. Pada aplikasi *instant messaging* yang mengimplementasikan enkripsi, *client* merupakan sebuah aplikasi *instant messaging* (*IM Client*) yang berjalan di perangkat *mobile*. Sedangkan *server* mempunyai 2 fungsi yaitu sebagai *Messaging Server* dan *Public Key Distribution Center* yang menggunakan *database* untuk menyimpan kumpulan kunci publik. Setiap fungsi ini memiliki peran masing-masing seperti yang digambarkan pada arsitektur yang lebih detail (Gambar 3).



Gambar 2. Arsitektur umum sistem



Gambar 3. Arsitektur detail dari sistem

Berdasarkan arsitektur yang lebih detail, terdapat 3 bagian utama, yaitu:

A. *IM Client*

IM Client mempunyai fungsi utama yaitu mengirim dan menerima pesan. Selain itu, dalam mengimplementasikan enkripi-dekripsi, *IM Client* berfungsi untuk membangkitkan pasangan kunci privat dan kunci publik serta melakukan enkripsi dan dekripsi pesan. Setelah membangkitkan pasangan kunci, *IM Client* bertugas untuk mengirimkan kunci publik ke *Public Key Distribution Center* untuk disimpan di *server*. Dalam proses pengiriman pesan, *IM Client* akan mengenkripsi pesan tersebut terlebih dahulu sebelum dikirim dengan menggunakan kunci publik penerima. Kunci publik penerima tersebut diperoleh dengan cara meminta ke *Public Key Distribution Center*. Sedangkan ketika menerima pesan, *IM Client* akan mendekripsi pesan terenkripsi yang diterimanya dengan kunci publik yang telah dibangkitkan.

B. *Messaging Server*

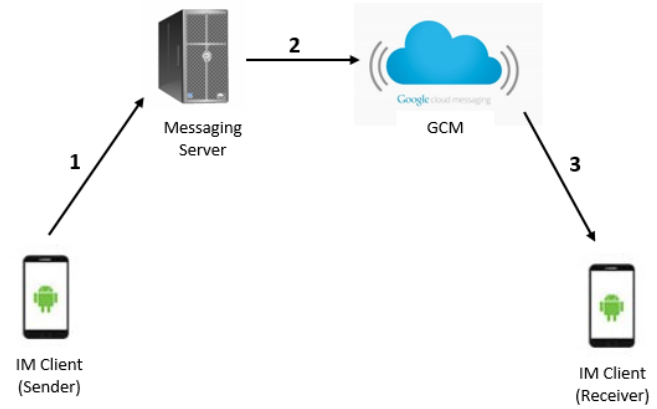
Messaging Server mempunyai fungsi utama yaitu meneruskan pesan yang dikirim oleh *IM Client* pengirim ke *IM Client* penerima. Pesan yang diterima oleh *Messaging Server* yang akan diteruskan ke *IM Client* merupakan pesan yang sudah terenkripsi.

C. *Public Key Distribution Center*

Public Key Distribution Center (PKDC) merupakan pusat penyimpanan dan distribusi kunci publik. PKDC akan menerima kunci publik yang dibangkitkan dan dikirimkan oleh *IM Client* lalu menyimpan kunci publik tersebut untuk bisa didistribusikan ke *IM Client* lain yang membutuhkan. PKDC akan mendistribusikan kunci publik ke *IM Client* yang meminta dan membutuhkan kunci publik untuk keperluan proses enkripsi pesan yang dilakukan oleh *IM Client*.

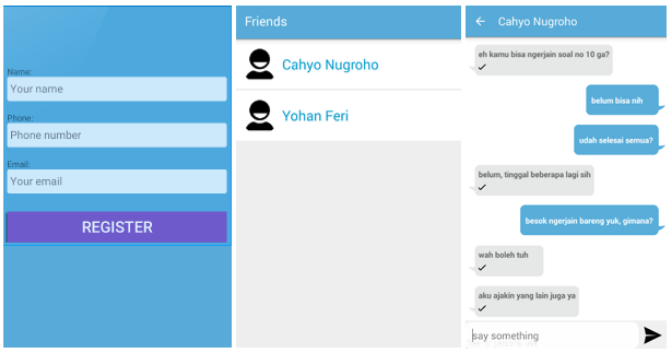
IV. IMPLEMENTASI

Implementasi dilakukan dengan menggunakan bahasa pemrograman Java untuk *IM Client* yang berupa aplikasi *instant messaging* yang berjalan pada perangkat Android. Sedangkan untuk aplikasi yang berjalan di *server* diimplementasikan dengan menggunakan bahasa pemrograman PHP dan basis data MySQL. Dalam implementasi ini, digunakan bantuan layanan Google Cloud Messaging (GCM). Dengan menggunakan layanan ini maka *server* dapat mengirimkan data dari *server* ke perangkat *mobile* dengan *platform* Android. Hal ini digunakan oleh *server* untuk meneruskan pesan yang dikirim ke penerimanya. Proses pengiriman pesan dengan bantuan layanan GCM ini dapat dilihat pada Gambar 4.



Gambar 4. Proses pengiriman pesan dengan layanan GCM

Beberapa tampilan antarmuka dari hasil implementasi aplikasi *instant messaging* dapat dilihat pada Gambar 5.



Gambar 5. Antarmuka aplikasi

V. PENGUJIAN

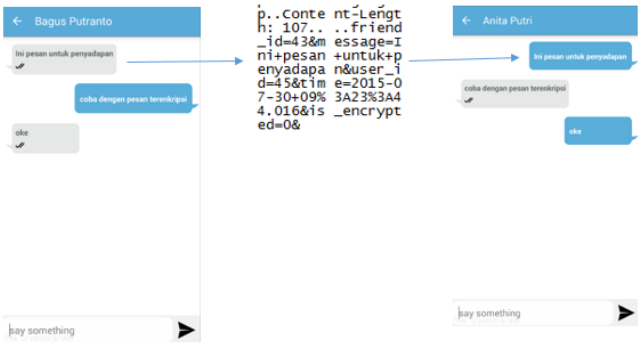
Pengujian dibagi menjadi 2 bagian yaitu pengujian terhadap penyadapan dan pengujian kinerja. Pengujian ini dilakukan pada telepon selular Android dengan tipe Asus Zenfone C yang memiliki spesifikasi sistem operasi Android 4.4.2, RAM 1 GB, dan CPU Dual-core 1.2 GHz. Serta sebuah laptop yang dipakai untuk menangkap paket data yang dikirim.

A. Pengujian Penyadapan

Pengujian ini dilakukan untuk mengetahui apakah pesan yang dikirimkan sudah dalam bentuk terenkripsi atau bentuk pesan yang mudah dibaca jika aplikasi mengimplementasikan enkripsi. Pengujian ini dilakukan dengan menggunakan bantuan perangkat lunak bernama Wireshark untuk menangkap paket data yang melewati sebuah jaringan.

Pengujian dilakukan dengan mengirimkan sebuah pesan melalui aplikasi dan Wireshark akan menangkap paket data yang dikirimkan selama proses pesan tersebut dikirim ke *server*.

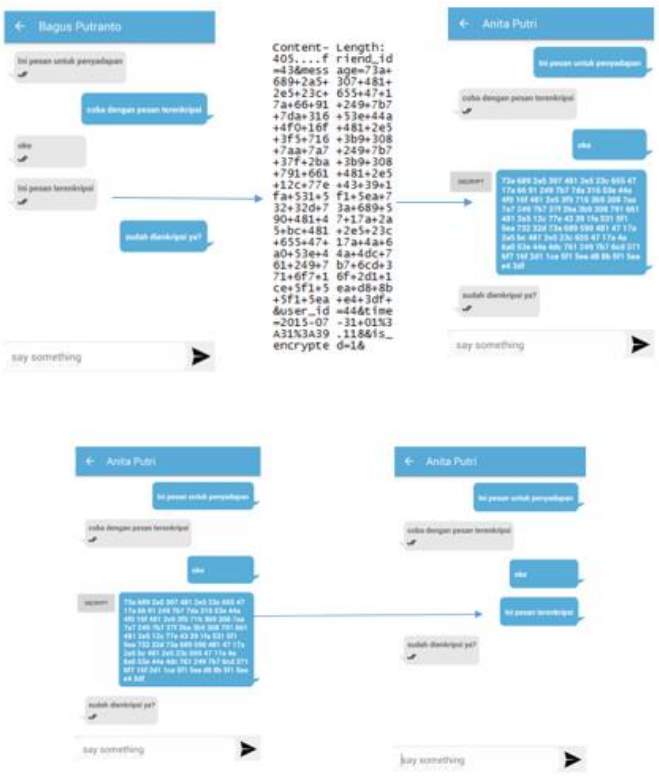
Untuk mengetahui apakah pesan yang dikirimkan sudah dalam bentuk cipherteks atau tidak maka pengujian yang dilakukan dibagi menjadi 2 macam, yaitu ketika modul enkripsi diaktifkan dan ketika modul enkripsi tidak diaktifkan. Hal ini bertujuan untuk melihat perbedaan di antara kedua macam pengujian tersebut. Pada proses pengujian pertama, modul enkripsi tidak diaktifkan sehingga pesan yang dikirim seharusnya berupa pesan yang mudah dibaca. Hasil pengujian ini dapat dilihat pada Gambar 6.



Gambar 6. Pengujian tanpa enkripsi

Pengujian selanjutnya dilakukan dengan mengaktifkan modul enkripsi. Karena modul enkripsi diaktifkan maka seharusnya pesan yang dikirim sudah dalam bentuk cipherteks. Hasil pengujian ini dapat dilihat pada Gambar 7.

Berdasarkan hasil pengujian yang telah dilakukan, terdapat perbedaan bentuk pesan hasil penyadapan ketika modul enkripsi tidak diaktifkan dan ketika modul enkripsi diaktifkan. Jika modul enkripsi tidak diaktifkan, maka pesan yang disadap dapat dengan mudah untuk dibaca. Sedangkan jika modul enkripsi diaktifkan, maka pesan yang didapat dari proses penyadapan berupa cipherteks sehingga tidak mudah untuk dibaca dan perlu didekripsi terlebih dahulu agar pesan tersebut mudah dibaca.



Gambar 7. Pengujian dengan enkripsi

B. Pengujian Kinerja

Pengujian ini dilakukan untuk mengetahui pengaruh dari jumlah tambahan waktu (*overhead*) yang digunakan untuk proses enkripsi dan dekripsi dalam pengiriman dan penerimaan pesan terhadap kinerja aplikasi. Data uji yang digunakan dalam pengujian ini terdapat 2 jenis yaitu pesan dengan jumlah karakter tertentu dan parameter kurva eliptik bidang terbatas. Data tersebut digunakan untuk mengetahui pengaruh dari panjangnya pesan dan besarnya parameter algoritma terhadap waktu enkripsi dan dekripsi. Data uji ini dapat dilihat pada Tabel I dan Tabel II.

Pengujian ini dilakukan dengan menghitung waktu *overhead* dari eksekusi operasi enkripsi dan dekripsi pesan. Pengujian dilakukan untuk setiap pesan dengan menggunakan semua parameter kurva eliptik bidang terbatas.

Dalam pengujian ini, penghitungan dari penggunaan waktu dilakukan ketika memanggil fungsi enkripsi dan dekripsi. Hasil pengujian ini dapat dilihat pada Tabel III yang berisi informasi hasil rata-rata *overhead* proses enkripsi dan dekripsi.

TABLE I. DATA UJI BERDASARKAN PANJANG PESAN

No	Isi Pesan	Panjang Pesan
1	Ini adalah isi pesan	20 karakter
2	Pesan ini merupakan salah satu data dari data uji.	50 karakter
3	Bagian ini merupakan sebuah contoh dari pesan uji yang panjangnya paling tinggi.	80 karakter

TABLE II. DATA UJI BERDASARKAN PARAMETER ALGORITMA

No	Parameter Kurva Eliptik Bidang Terbatas			
	a	***b***	***p***	***G***
1	9	7	2011	(251,157)
2	-3	64210519e59c80e70fa7e9ab72243049feb8deecc146b9b1	6277101735386680763835789423207666416083908700390324961279	(188da80eb03090f67cbf20eb43a18800f4ff0afd82ff1012, 07192b95ffc8da78631011ed6b24cdd573f977a11e794811)

TABLE III. HASIL PENGUJIAN RATA-RATA *OVERHEAD*

No Pesan	Rata-Rata *Overhead* (Parameter Nomor 1)		Rata-Rata *Overhead* (Parameter Nomor 2)	
	Enkripsi	***Dekripsi***	***Enkripsi***	***Dekripsi***
1	222,4 ms	42,1 ms	3669,6 ms	1682,9 ms
2	448,9 ms	147,6 ms	9378 ms	4209,5 ms
3	844 ms	196 ms	14949,3 ms	6847 ms

Dari hasil pengujian yang diperoleh, semakin panjang pesan maka waktu yang digunakan untuk proses enkripsi dan dekripsi juga semakin besar. Hal ini disebabkan karena semakin panjang pesan maka proses enkripsi dan dekripsi yang dilakukan juga semakin banyak sehingga waktu yang digunakan juga semakin besar. Selain itu, nilai parameter yang digunakan juga mempengaruhi jumlah *overhead* proses enkripsi dan dekripsi. Semakin besar nilai parameter yang digunakan maka tingkat keamanan akan semakin meningkat. Namun semakin besar nilai parameter yang digunakan maka *overhead* dari proses enkripsi dan dekripsi juga semakin besar. Hal tersebut dikarenakan semakin besar nilai parameter yang digunakan maka operasi komputasi dalam fungsi enkripsi dan dekripsi akan melibatkan bilangan yang sangat besar dan waktu yang lebih lama untuk memprosesnya.

C. Analisis Hasil Pengujian

Melalui pengujian dengan penyadapan, hasil yang diperoleh menunjukkan bahwa pesan yang disadap berupa pesan dalam bentuk terenkripsi ketika modul enkripsi diaktifkan karena sebelum pesan tersebut dikirimkan, pesan dienkripsi terlebih dahulu. Oleh karena itu, selama pengiriman pesan dari aplikasi ke *server* dan dari *server* diteruskan ke aplikasi penerima, pesan tersebut sudah dalam bentuk terenkripsi. Karena pesan telah terenkripsi maka keamanan pesan akan lebih terjaga dari tindakan penyadapan.

Selain itu, berdasarkan hasil pengujian kinerja diketahui bahwa kinerja aplikasi dipengaruhi oleh jumlah *overhead* yang timbul dari proses enkripsi dan dekripsi. Besarnya jumlah *overhead* yang terjadi ditentukan oleh panjang pesan yang dienkripsi dan didekripsi serta juga ditentukan oleh besarnya bilangan yang digunakan dalam parameter algoritma enkripsi dan dekripsi.

VI. KESIMPULAN

Aplikasi *instant messaging* dapat dibangun sesuai dengan rancangan arsitektur sistem yang terdiri dari *IM Client* dan *IM Server*. IM *Client* berfungsi untuk mengirim dan menerima pesan. Sedangkan *IM Server* berfungsi untuk meneruskan pesan dari pengirim ke penerima. Dalam mengimplementasikan teknik enkripsi ECC pada aplikasi ini, proses pembangkitan kunci serta proses enkripsi-dekripsi dilakukan oleh *client*. Sedangkan untuk distribusi kunci dilakukan oleh *server*.

Pengujian keamanan pesan yang dikirimkan melalui aplikasi *instant messaging* yang dibangun dilakukan dengan penyadapan. Dari hasil penyadapan tersebut dapat diketahui pesan dalam bentuk terenkripsi atau tidak. Dengan terenkripsinya pesan maka keamanan pesan akan lebih terjaga dari tindakan penyadapan.

Penggunaan algoritma enkripsi ECC akan membutuhkan waktu tambahan (*overhead*) untuk melakukan eksekusi fungsi enkripsi dan dekripsi. Kinerja aplikasi dipengaruhi oleh besarnya *overhead*. *Overhead* dari proses enkripsi dan dekripsi ditentukan oleh panjang pesan dan nilai parameter yang digunakan dalam algoritma enkripsi-dekripsi.

REFERENSI

[1] Meyer, D. (2013, April 29). *Chat apps have overtaken SMS by message volume, but how big a disaster is that for carriers?* Diambil dari Gigaom: https://gigaom.com/2013/04/29/chat-apps-have-overtaken-sms-by-message-volume/, Tanggal akses : 15 November 2014.

[2] Certicom. (2004). *An Elliptic Curve Cryptography (ECC) Primer.* The Certicom 'Catch the Curve' White Paper Series.

[3] Electronic Frontier Foundation. (2015, June 12). *Secure Messaging Score Board*. Diambil dari Electronic Frontier Foundation: https://www.eff.org/secure-messaging-scorecard, Tanggal akses : 18 Agustus 2015.

[4] Rangarajan, S., Nellutla, S. R., & Nellutla, V. K. (2013). *Securing SMS using Cryptography. International Journal of Computer Science and Information Technologies,* (pp. 285 - 288).

[5] Yusof, M. K., Usop, S.M., & Abidin, A. F. (2001). *Designing a Secure Architecture for Private. International Conference on Computer Science and Information Technology*. Pattaya.

[6] Wanda, P., Selo, Hantono, B. S. (2014). *Efficient Message Security Based Hyper Elliptic Curve Cryptosystem (HECC) for Mobile Instant Messenger. International Conference on Information Technology, Computer and Electrical Engineering.* Semarang.

[7] Tung, T.-Y., Lin, L., & Lee, D. T. (2012). *Pandora Messaging: An Enhanced Self-Message-Destructing Secure Instant. International Conference on Advanced Information Networking and Applications Workshops.*