PERBANDINGAN KECEPATAN ALGORITMA KRIPTOGRAFI ASIMETRI

Megah Mulya[1,2)]
[1]Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Sriwijaya
[2]megahmulya@yahoo.com

ABSTRACT

Nowadays the issue of data security has evolved into a staple in every organization. Cryptographic is a common technique to secure the data.To determine which Cryptographic algorithms will be used in data security systems, besides of considering the strength, the speed is also important to be considered. This consideration of the speed will be more prefered if Cryptographic algorithms is used in the computer network environment. Therefore, besides of the speed, the strength of the cryptographic algorithm is also has to be considered. The measurement of cryptographic algorithm speed has been done by several parties, but commonly for symmetry type. Therefore in this research performed comparison of three speed popular asymmetric cryptographic algorithm, that are RSA, Elgamal and eliptic Curve. RSA, Elgamal and eliptic Curve these three are type of asymmetric cryptographic algorithm that has a public and private key pairs for encryption and decryption. In this study, all three algorithms are implemented in C + + and compiled with Borland C + + Builder. This study led to the conclusion of encryption processing speed sequence from the fastest to the slowest which are RSA, Elgamal and eliptic Curve. While the sequence of the fastest to the slowest for the decryption process is different from the order of the encryption process which are Elgamal, eliptic Curve and RSA.

*Keywords:*Elgamal, eliptic Curve, RSA, comparison performance.

I. PENDAHULUAN

Untuk menentukan algoritma Kriptografi yang akan digunakan dalam sistem keamanan data selain pertimbangan kekuatan terhadap serangan *Cryptanalis* dan *Brutforse* yang tidak kalah penting adalah pertimbangan kecepatan [**1**]. Pada saat ini terdapat berbagai macam algoritma Kriptografi simetri maupun asimetri. Jika suatu algoritma Kriptografi dipercaya kuat namun diketahui lamba dalam proses penyandiannya maka tidak akan dijadikan pilihan oleh pengguna. Pertimbangan kecepatan ini akan menjadi lebih diutamakan lagi jika pemakaian algoritma Kriptografi menyangkut jaringan komputer terutama pada arsitektur clien-server. Pada jaringan clien-server dengan jumlah clien besar maka lambatnya kinerja algoritma Kriptografi akan sangat signifikan menambah beban keja server. Jika banyak clien dalam saat yang bersamaan melakukan *handshake* (pada *hybrid cryptosystem*) maka akumulasi waktu penyadian akan sangat menjengkelkan pengguna saat login (bahkan dapat hang). Olehkarena itu maka pengguna dalam memilih algoritma Kriptografi akan juga sangat memperhatikan faktor kecepatan selain kekuatannya.

Pengukuran kecepatan algoritma kriptografi telah dilakukan oleh berberapa pihak namun pada umumnya untuk jenis simetri [**1**]-[**2**]. Sedangkan untuk algoritma asimetri pengukuran telah dilakukan oleh beberapa pihak namun tidak sebanyak pihak yang mengukur algoritma simetri. Pengukuran kecepatan algoritma asimetri pada umumnya juga sebatas RSA dengan sebuah algoritma lain [**3**]atau antar varian RSA [**4**]. Hal itu disebabkan karena algoritma simetrilah yang umumnya digunakan untuk penyadian data secara langsung. Padahal walaupun tidak digunakan untuk penyadian data secara langung algoritma Kriptografi asimetri akan dapat membebani kinerja server jika pada saat bersamaan terlalu banyak clien melakukan *handshake* secara bersamaan. Oleh karena itu pada penelitian ini akan dilakukan perbandingan kecepatan tiga algorirtma Kriptografi asimetri yang populer yaitu RSA, Elgamal dan Eliptic Curve.

RSA, Elgamal dan Eliptic Curve ketiganya merupakan algoritma kriptografi jenis simetri yang memiliki pasangan kunci privat dan publik untuk proses enkripsi dan dekripsi. RSA merupakan algoritma kriptografi asimetri yang paling populer. Tumpuan kekuatan dari

algoritma ini adalah pada kesulitan melakukan pemfaktoran bilangan yang besar menjadi faktor-faktor prima. Proses pemfaktoran dilakukan untuk mendapatkan kunci private/pribadi. Oleh karena itu algoritma ini akan tetap dinyatakan kuat asalkan belum ditemukan cara yang mangkus untuk pemfaktoran bilangan besar menjadi faktor-faktor prima [**5**]

II. RSA

Dari sekian banyak algoritma kriptografi kunci-publik yang pernah dibuat, algoritma yang paling populer adalah algoritma RSA. Algoritma RSA dibuat oleh 3 orang peneliti dari MIT (*Massachussets Institute of Technology*) pada tahun 1976, yaitu: Ron (R)ivest, Adi (S)hamir, dan Leonard (A)dleman. Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima [**5**].

Pada RSA terdapat tiga proses utama yaitu proses pembentukan kunci, proses enkripsi dan proses dekripsi. Prinsip pembentukan kunci untuk algoritma RSA adalah sebagai berikut:

- Pilih duabilangan prima yang sangat besarutamarandom: pdanq
- Hitungn danϕ(n): n =pqdanϕ(n) =(p-1) (q-1)
- Piliheinteger,1<e <φ (n) sedemikian rupa sehingga: gcd(e, ϕ(n)) =1
- Hitunglahd, 1<d <φ (n) sedemikian rupa sehingga: ed≡1(mod ϕ(n)) kuncipublik(n, e) dankunci pribadiadalah(n, d)

  nilai-nilaip, q, dan ϕ(n) adalah swasta eadalah eksponenpublik atauenkripsi dadalah eksponenpribadi ataudekripsi

sehingga pasangan {d,n} merupakan kunci privat/pribadi dan {e,n} adalah kunci publik [**5**].

Pada RSA algoritma enkripsi yang digunakan adalah sebagai berikut:

$$C = M^e \bmod n \quad (1)$$

Dimana M adalah pesan asli yang akan di jadikan chipertext

Proses dekripsi merupakan proses mengembalikan pesan yang telah di enkripsi ke bentuk awal pesan(plaintext),Algoritma untuk dekripsi adalah sebagai berikut:

$$M = C^d \bmod n \quad (2)$$

II. El Gamal

Algoritma ElGamal merupakan algoritma kriptografi asimetris.Pertama kali dipublikasikan oleh Taher ElGamal pada tahun 1985.Algoritma ini didasarkan atas masalah logaritma diskrit pada grup ZP*.Kekuatan algoritma ElGamal terletak pada kesulitan penghitungan logaritma diskret pada bilangan modulo prima yang besar sehingga upaya untuk menyelesaikan masalah logaritma ini menjadi sangat sukar.Algoritma Elgamal tidak dipatenkan.Tetapi, algoritma ini didasarkan pada algoritma Diffie – Hellman, sehingga hak paten algoritma Diffie – Hellman juga mencakup algoritma ElGamal.Karena hak paten algoritma Diffie – Hellman berakhir pada bulan April 1997, maka algoritma ElGamal dapat diimplementasikan untuk aplikasi komersil.

Algoritma Elgamal terdiri dari tiga proses, yaitu proses pembentukan kunci, proses enkripsi dan proses dekripsi. Algoritma ini merupakan cipher blok, yaitu melakukan proses enkripsi pada blok-blok plainteks dan menghasilkan blok-blok cipherteks yang kemudian dilakukan proses dekripsi, dan hasilnya digabungkan kembali menjadi pesan yang utuh dan dapat dimengerti. Untuk membentuk sistem kriptografi ElGamal, dibutuhkan bilangan prima *p* dan elemen primitif grup ZP*

Algoritma kriptografi Elgamal terdiri dari tiga proses, yaitu: proses pembangkitan kunci, enkripsi dan dekripsi. Proses Pembangkitan Kunci adalah merupakan proses yang dilakukan untuk memperoleh kunci publik yang akan digunakan pada proses enkripsi, hal ini tidak berlaku pada proses dekripsi yang mana dapat dilakukan tanpa proses pembangkitan kunci.

Pembentukan kunci terdiri atas pembentukan kunci publik dan kunci rahasia. Pada proses ini dibutuhkan sebuah bilangan prima *p* yang digunakan untuk membentuk grup Zp*, elemen primitif α dan sembarang $a \in \{0,1,..., p-2\}$. Kunci publik algoritma ElGamal terdiri atas pasangan 3 bilangan (p,α ,β). Ketiga bilangan tersebut memenuhi persamaan:

$$\beta = \alpha^a \bmod p \quad (3)$$

yang digunakan sebagai kunci rahasia adalah bilangan *a*.

Proses enkripsi menggunakan kunci publik (p,α ,β) dan sebuah bilangan integer acak *k* (*k*

$\in\{0,1,\ldots, p-1\}$) yang dijaga kerahasiaannya oleh penerima yang mengenkripsi pesan. Untuk setiap karakter dalam pesan dienkripsi dengan menggunakan bilangan k yang berbeda-beda. Satu karakter yang direpresentasikan dengan menggunakan bilangan bulat ASCII akan menghasilkan kode dalam bentuk blok yang terdiri atas dua nilai (r, t).

Proses Dekripsi dari cipherteks ke plainteks menggunakan kunci rahasia a yang disimpan kerahasiaanya oleh penerima pesan. Teorema yang digunakan adalah diberikan (p,α ,β) sebagai kunci public dan a sebagai kunci rahasia pada algoritma ElGamal. Jika diberikan cipherteks (r, t), maka:

$$M = t\,(ra)\text{ -1 mod } p \text{ (a)} \qquad (4)$$

dengan$M$ adalah plainteks.

IV. ELLIPTIC CURVE

Elliptic Curve Cryptography adalah kriptografi kunci publik.yang didasarkan atas himpunan yang mengasosiasikan kunci dan operasi kriptografi yang digunakan.hanya pengguna yang cocok yang dapat menggunakan privat key yang sesuai tetapi kunci publik yang digunakan disebarkan kepada pihak yang akan mengirimkan data kepada pemilik private key. Beberapa algoritma kunci publik menyediakan pendefinisian konstanta yang akan disebarkan ke semua bagian yang ikut berpartisipasi dalam komunikasi. Domain parameter di ECC adalah salah satu contoh dari konstanta tersebut. Kriptografi kunci publik tidak seperti algoritma kunci privat yang tidak menyediakan privat key ke seluruh pengguna tetapi lebih lambat dibanding algoritma kunci privat [**5**]

ECC adalah salah satu pendekatan algoritma kriptografi kunci publik berdasarkan pada struktur aljabar dari kurva ellips pada daerah finite.Penggunaan elliptic curve pertama kali dicetuskan oleh Neal Koblitz dan Viktor S Miller pada tahun 1985. Elliptic Curve juga digunakan pada beberapa algoritma pemfaktoran integer yang juga diaplikasikan dalam kriptografi seperti Lenstra Elliptic Curve Factorization[**6**]

ECC adalah teknologi yang sangat efisien untuk PKI (*Public Key Infrastructur).*Keamanan dari sistem kunci publik yang menggunakan elliptic curve berdasarkan kesulitan dalam komputasi algoroitma diskrit pada group dengan poin pada elliptic curve yang didefinisokan atas finite field. Beberapa pondasi matematika dari ECC adalah aritmatika modular, groups dan finite field yang di dalamnya terdapat groups, order group an generator, subgroup, finite field, dan *The Discrete Logarithm Problem* (DLP).

Keunggulan dari kriptosistem kurva elips adalah proses transformasi *plaintext* menjadi titik-titik dalam kurva elips sebelum dilakukan enkripsi. Proses enkripsinya dilakukan dengan menggunakan aturan penjumlahan pada kurva elips. Proses ini tentunya akan memberikan tingkat keamanan yang lebih baik.

Kriptosistem kurva elips memberikan tingkat keamanan yang lebih baik dibandingkan dengan algoritma asimetris lainnya seperti RSA. Hasil tinjauan pustaka memperlihatkan untuk tingkat keamanan yang sama (MIPS tahun yang sama) kriptosistem kurva elips memerlukan jumlah bit kunci yang jauh lebih sedikit dibandingkan dengan RSA atau DSA. Hal ini tentunya kriptosistem kurva elips dapat menjadi pilihan yang baik untuk membangun sistem kriptografi yang memiliki tingkat kemanan yang tinggi [**3**].

V. HASIL DAN PEMBAHASAN

Data yang digunakan dalam penelitian ini adalah berupa data teks dalam ukuran byte. Data tersebut akan dienkripsi dan didekripsi dengan diukur waktunya dengan menggunakan timer yang ada di dalam operating sistem.

Pada penelitian ini untuk mencapai tujuan maka dilakukan langkah-langkah sebagai berikut:

Mengukur kecepatan enkripsi dan dekripsi

Untuk mengukur kecepatan algoritma RSA, ElGamal dan Eliptic curve Penelitian ini direncanakan menggunakan tiga standar yang harus dibuat sama didalam setiap eksperimennya agar didapatkan hasil yang konsisten dan obyektif. Ketiga standar tersebut adalah parameter sistem, faktor eksperimen dan prosedur simulasi [**2**].

Parameter sistem

Eksperimen eksekusi program simulasi akan dilakukan pada komputer dengan spesifikasi perangkat keras dan perangkat yang sama. Eksperimen juga dilakukan beberapa kali agar hasil pengukuran waktu konsisten mengingat

kecepatan ekseskusi program dapat berubah-ubah sesuai dengan ketidak stabilan kesibukan prosesor.

Faktor Eksperimen

Pada penelitian ini faktor penelitian yang diuji adalah kecepatan algoritma. Kecepatan yang dimaksud adalah meliputi kecepatan proses enkripsi dan kecepatan proses dekripsi. Setiap algoritma akan diuji pada proses enkripsi dan dekripsi terhadap berbagai ukuran data (dalam byte).

Prosedur Simulasi

Program simulasi akan mengeksekusi modul algoritma untuk mengenkripsi dan mendekripsi berbagai data dan dicatat waktu yang diperlukan. Dari catatan waktu yang diperlukan untuk mengenkripsi dan mendekripsi akan dapat diketahui kecepatan dan karakteristik algoritma.

Program Simulasi menerima dua input yaitu algoritma dan ukuran blok data. Setelah eksekusi sukses, data yang dihasilkan akan terenkripsi dan sekaligus didekripsi yang akan ditampilkan pada GUI. Melalui tampilan GUI akan dapat dipastikan bahwa proses enkripsi dan dekripsi berjalan dengan normal dengan cara membandingkan data asli dengan data terdekripsi.

Membuat perbandingan

Untuk melakukan perbandingan kecepatan dari ketiga algoritma kriptografi dalam penelitian ini maka hasil pengukuran waktu untu enkripsi dan dekripsi terhadap data kemudian diplot kedalam grafik kartesian antara ukuran data versus waktu yang dibutuhkan. Dari gravfik tersebut maka terlihat perbandingan kecepatan sekaligus karakteristik ketiga algoritma tersebut. Perbandingan tersebut dilakukan dengan menerapkan panjang kunci yang sama yaitu 8 digit bilangan bulat.

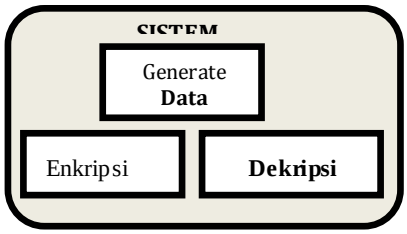
Lingkungan Implementasi Perangkat Lunak

Lingkungan implementasi perangkat lunak yang dibangun meliputi lingkungan perangkat keras dan perangkat lunak dengan spesifikasi sebagai berikut :

1. Perangkat Keras
   a. Processor Intel Core i3 2,27 GHz
   b. RAM 2 GB
   c. Hard Disk 500 GB
   d. VGA 512 MB
2. Perangkat lunak
   a. Sistem operasi windows 7 64 *bit*
   b. *Compiler* Borland C++ Builder
   c.

Implementasi Perangkat Lunak

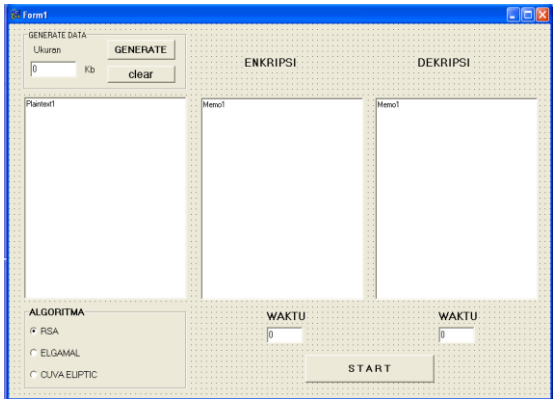
Bahasa yang pemrograman yang digunakan untuk implementasi perangkat lunak ini adalah C++ dengan *compiler* Borland C++ Builder.



Gambar 1 Model sistem perangkat lunak

Hasil pemodelan sistem perangkat lunak dapat dilihat pada gambar 1.

Perancangan antar muka perangkat lunak dapat dilihat pada gambar 2.



Gambar 2. Disain antar muka perangkat lunak

Eksekusi Perangkat Lunak

Hasil eksekusi perangkat lunak untuk beberapa nilai ukuran data dalam kilo byte dapat diperoleh lamanya waktu enkripsi dan dekripsi. Lamanya waktu proses diukur dari ketiga algoritma dengan panjang kunci yang sama yaitu 8 digit dan dengan berbagai ukuran data dalam *byte*. Oleh karena kinerja prosesor komputer tidak stabil selama proses pengukuran waktu maka waktu enkripsi dan dekripsi diambil dari nilai rata-rata. Waktu diukur secara berulang-ulang sebanyak masing-masing 5 kali kemudian diambil nilai rata-ratanya.

Untuk ukuran data yang kecil yaitu kurang dari 10 Kb waktu yang dibutuhkan untuk proses enkripsi dan dekripsi terkadang tidak terukur karena sangat kecil yaitu kurang dari 1 ms. Oleh karena itu ukuran data yang digunakan dalam percobaan diambil dari 10 Kb sampai

100 Kb.

Lamanya waktu enkripsi dan dekripsi algoritma RSA, Elgamal dan Kurva eliptik disajikan dalam tabel 1.

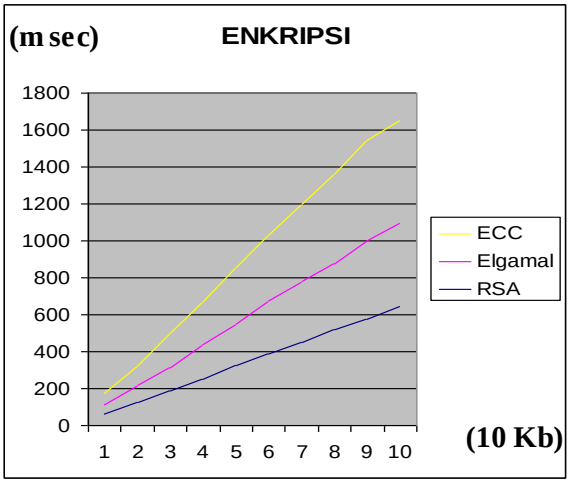
Tabel 1. Tabel data waktu enkripsi dan dekripsi

Ukuran (Kb)	RSA		Elgamal		ECC	
	Enkripsi (ms)	Dekripsi (ms)	Enkripsi (ms)	Dekripsi (ms)	Enkripsi (ms)	Dekripsi (ms)
10	63	141	47	16	62	16
20	125	281	94	16	109	47
30	188	422	125	16	187	47
40	250	562	187	15	234	63
50	328	688	219	16	312	78
60	390	859	282	16	359	110
70	453	969	328	15	422	109
80	516	1141	360	16	484	141
90	578	1266	421	15	547	156
100	641	1391	453	16	563	172

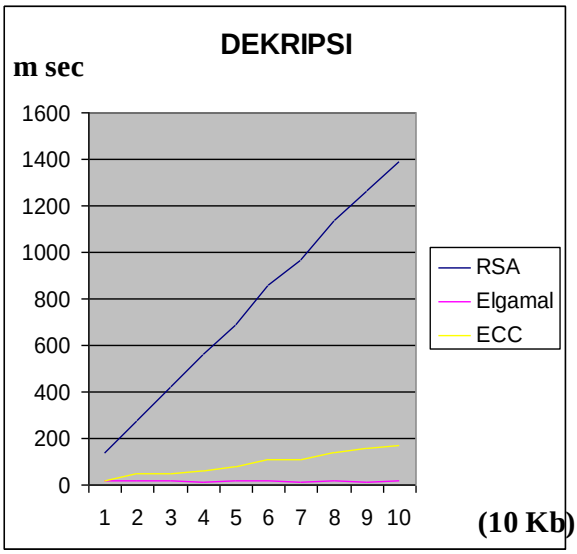
Dari data tabel 1.tersebut telah dibuat grafik waktu enkripsi dan waktu dekripsi seperti yang terlihat pada gambar 3. dan gambar 4**.**

Dari kedua grafik tersebut terdapat kesimpulan bahwa ketiga algoritma kecapatan proses enkripsi dan dekripsi menurun secara linier terhadap ukuran data. Pada proses enkripsi urutan kecepatan dari yang paling besar ke yang paling kecil adalah RSA, Elgamal dan ECC (kurva eliptik). Semakin besar ukuran data yang dienkripsi maka perbedaan kecepatan ketiga algoritma semakin tajam.

Dari grafik dekripsi pada gambar 3.terlihat bahwa urutan kecepatan proses dekripsi ketiga algoritma dari yang tercepat ke yang terlambat adalah ECC(kurva eliptik), Elgamal dan RSA. Semakin besar ukuran data yang dienkripsi maka perbedaan kecepatan enkripsi ketiga algoritma semakin tajam. Untuk RSA kecepatannya sangat menurun dibandingkan kedua algoritma yang lain jika ukuran data semakin besar. Sedangkan untuk algoritma Elgamal dan ECC kecepatan dekripsi menurun tetapi tidak terlalu signifikan.



Gambar 3. Grafik enkripsi RSA, Elgamal dan Kurva eliptik ukuran (Kb) data versus waktu (ms)



Gambar 4. Grafik dekripsi RSA, Elgamal dan Kurva eliptik ukuran data (Kb) versus waktu (ms)

Performansi relatif proses enkripsi dan dekripsi dari ketiga algoritma tersebut diperbandingkan secara kualitatif dapat dilihat pada tabel 2.

Tabel 2. Tabel perbandingan kualitatif kecepatan relatif ketiga algoritma

Algoritma	Enkripsi	Dekripsi
RSA	Cepat	lambat
Elgamal	Sedang	Sangat cepat
Eliptic Curve	lambat	Cepat

Kesimpulan

Penelitian ini telah menghasilkan perangkat lunak untuk mengukur dan membandingkan kecepatan enkripsi dan dekripsi tiga algoritma Kriptografi asimetri yaitu RSA, El Gamal dan Eliptic Curve. Dari hasil pengukuran waktu enkripsi dan dekripsi serat setelah dilakukan perbandiungan maka didapat kesimpulan bahwa urutan kecepatan enkripsi dari yang terbesar adalah RSA, Elgamal lalu *Eliptic Curve*. Sedangkan urutan kecepatan dekripsi berbeda dari kecepatan enkripsi yaitu dari yang terbesar adalah Elgamal, *Eliptic Curve* lalu RSA. Kesimpulan lainnya adalah kecepatan enkripsi dan dekripsi dari ketiga algoritma menurun secara linier terhadap ukuran data.Khusus untuk Elgamal dan *Eliptic Curve* penurunan kecepatan dekripsi terhadap ukuran data tidak signifikan sedangkan untuk RSA sangat signifikan.Hasil penelitian ini dapat ditingkatkan obyektifitasnya jika implementasi program algoritma Kriptografi yang berupa librari (pustaka) yang sudah populer. Selain itu untuk lebih mempertajam hasil pengukuran perbedaan kecepatan ketiga algoritma disarankan agar percobaan dilakukan dengan variasi panjang kunci

VI. KESIMPULAN DAN SARAN

Penelitian ini telah menghasilkan perangkat lunak untuk mengukur dan membandingkan kecepatan enkripsi dan dekripsi tiga algoritma Kriptografi asimetri yaitu RSA, El Gamal dan Eliptic Curve. Dari hasil pengukuran waktu enkripsi dan dekripsi serat setelah dilakukan perbandiungan maka didapat kesimpulan bahwa urutan kecepatan enkripsi dari yang terbesar adalah RSA, Elgamal lalu *Eliptic Curve*. Sedangkan urutan kecepatan dekripsi berbeda dari kecepatan enkripsi yaitu dari yang terbesar adalah Elgamal, *Eliptic Curve* lalu RSA. Kesimpulan lainnya adalah kecepatan enkripsi dan dekripsi dari ketiga algoritma menurun secara linier terhadap ukuran data.Khusus untuk Elgamal dan *Eliptic Curve* penurunan kecepatan dekripsi terhadap ukuran data tidak signifikan sedangkan untuk RSA sangat signifikan.Hasil penelitian ini dapat ditingkatkan obyektifitasnya jika implementasi program algoritma Kriptografi yang berupa librari (pustaka) yang sudah populer. Selain itu untuk lebih mempertajam hasil pengukuran perbedaan kecepatan ketiga algoritma disarankan agar percobaan dilakukan dengan variasi panjang kunci

DAFTAR PUSTAKA

[1] Simar P. Singh and Raman Maini, "Comparison Data Encryption Algorithm," *International Journal of Computer Science and Communication*, vol. 2, no. 1, January-June 2011.

[2] A. Nadeem, "A Performance Comparison of Data Encryption Algorithms, Information and Communication Technologies," in *First International Conference on Date of Conference*, 2005.

[3] B. Kute Vivek, "A Software Comparison Of RSA an ECC," *International Journal of Computer Science and Applications*, 2009.

[4] A Mamun, Mohammad Motaharul Islam, S.M. Mashihure Romman, and A.H. Salah Uddin Ahmad, "Performance Evaluation of Several Efficient RSA Variants," *International Journal of Computer Science and Network Security*, vol. 8, no. 7, July 2008.

[5] B. Schneier, *Applied Cryptography : Protocols, Algorithms and Source Code in C*, 4th ed.: John Wiley & Sons, Inc, 1996.

[6] Craig Larman, *Appliying UML and Patterns" An Introduction To Object-Oriented Analysis and Design and Iterative Development*.: Prentice Hall PTR, 2004.

[7] Ivar Jacobson, Grady Booch, and James Rumbaugh, *The Unified Software Development Process*.: Addison-Wesley, 2004.