STUDI PERBANDINGAN ALGORITMA RSA DAN ALGORITMA EL-GAMAL

Cindy Himawan[1], Toni Wibowo[2], Budi Sulityo[3], Rusdianto Roestam[4], Yuyu Wahyu[5], RB.Wahyu[6]

(1) President University, (Contact : cindy.himawan@gmail.com)
(2) President University, (Contact : toni.wibowo25@gmail.com)
(3) President University, (Contact : budi241@yahoo.com)
(4) President University, (Contact : rroestam@gmail.com)
(5) President University, (Contact : yuyuwahyusr@yahoo.com)
(6) President University, (Contact : rbw0101@yahoo.com)

Abstrak

Dengan kemajuan teknologi saat ini sangat memudahkan masyarakat untuk berkomunikasi. Namun, tingkat keamanan masih rendah sehingga memungkinkan informasi yang disampaikan dapat bocor kepada pihak-pihak yang tidak berkepentingan. Oleh karena itu diperlukan suatu software yang mampu melakukan enkripsi terhadap data teks, sehingga hanya orang yang dituju yang dapat mengetahui isi pesan setelah melalui proses dekripsi terlebih dahulu. Aplikasi enkripsi dan dekripsi pesan dapat meningkatkan tingkat keamanan pada layanan yang memerlukan kerahasiaan pesan. Hal ini dapat mengurangi bocornya informasi kepada pihak-pihak yang tidak berkepentingan. Untuk meningkatkan keamanan dalam aplikasi enkripsi dan dekripsi pesan, perlu digunakan algoritma yang handal. Banyak metode yang dapat digunakan untuk menyelesaikan masalah tersebut. Hingga akhirnya pada tahun 1976 muncul suatu sistem kriptografi baru, yaitu kriptografi kunci publik. Hingga saat ini ada beberapa algoritma kriptografi kunci publik yang sering digunakan yaitu Algoritma RSA dan Algoritma El-Gamal. Kedua algoritma tersebut memiliki perbedaan dalam proses pembangkitan kunci publik dan privat, serta perbedaan dalam proses enkripsi dan dekripsi. Pada penelitian ini, akan dicoba untuk melakukan studi perbandingan dari kedua algoritma tersebut, sehingga dapat diketahui kelebihan dan kekurangan dari kedua algoritma tersebut.

Key word : *Enkripsi, Dekripsi, kriptografi, Algoritma RSA, Algoritma El-Gamal.*

1. Pendahuluan

Dalam perkembangan teknologi sekarang ini, pertukaran informasi sudah sangat mudah dilakukan. Namun kebutuhan keamanan dan kerahasiaan dari suatu informasi semakin berkembang dari hari ke hari, dan user membutuhkan suatu aplikasi yang mampu menjaga keamanan dan kerahasiaan informasi tersebut.

Untuk mengatasi masalah tersebut maka diperlukan ilmu kriptografi, yang mempelajari teknik-teknik menyandikan suatu pesan dengan algoritma-algoritma tertentu. Salah satu metode yang digunakan adalah algoritma kunci. Metode ini tidak menumpukan keamanan pada algoritmanya, tetapi pada kerahasiaan kunci yang digunakan pada proses penyandian. Hingga saat ini ada beberapa algoritma kriptografi kunci publik yang sering digunakan yaitu Algoritma RSA dan Algoritma El-Gamal.

Algoritma RSA adalah algoritma kunci publik yang paling popular. Algoritma RSA dibuat oleh tiga orang peneliti dari MIT (Massachussets Institute of Technology), yaitu Ron Rivest, Adi Shamir, dan Len Adleman pada tahun 1976. Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi faktor-faktor prima.

Algoritma El-Gamal merupakan salah satu dari algoritma kunci. Algoritma ini dikembangkan pertama kali oleh Taher El-Gamal pada tahun 1985. Algoritma ini pada mulanya digunakan untuk digital signature, namun kemudian dimodifikasi sehingga juga bisa digunakan untuk enkripsi dan dekripsi. Keamanan algoritma ini terletak pada sulitnya menghitung logaritma diskrit.

Berdasarkan deskripsi singkat mengenai kedua algoritma kunci tersebut, dapat diperkirakan bahwa akan banyak perbedaan lainnya. Oleh karena perlu dilakukan studi perbandingan antara Algoritma RSA dan Algoritma El-Gamal.

2. Metodologi

2.1 Kriptografi

Kriptografi merupakan suatu ilmu yang mempelajari bagaimana cara menjaga agar data atau pesan tetap aman saat dikirimkan dari pengirim ke penerima tanpa mengalami gangguan dari pihak ketiga. Kriptografi sendiri berasal dari bahasa Yunani, crypto dan graphia. Crypto berarti secret (rahasia) dan graphia berarti writing (tulisan). Menurut terminologinya, Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat yang lain.

Prinsip-prinsip yang mendasari kriptografi yakni:

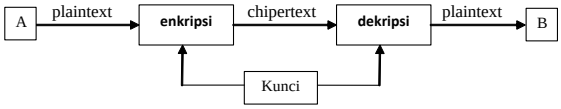
- Confidelity (kerahasiaan) yaitu layanan agar isi pesan yang dikirimkan tetap rahasia dan tidak diketahui oleh pihak lain (kecuali pihak pengirim, pihak penerima / pihak-pihak memiliki ijin). Umumnya hal ini dilakukan dengan cara membuat suatu algoritma matematis yang mampu mengubah data hingga menjadi sulit untuk dibaca dan dipahami.
- Data integrity (keutuhan data) yaitu layanan yang mampu mengenali/mendeteksi adanya manipulasi (penghapusan, pengubahan atau penambahan) data yang tidak sah (oleh pihak lain).
- • Authentication (keotentikan) yaitu layanan yang berhubungan dengan identifikasi. Baik otentikasi pihak-pihak yang terlibat dalam pengiriman data maupun otentikasi keaslian data/informasi.
- Non-repudiation (anti-penyangkalan) yaitu layanan yang dapat mencegah suatu pihak untuk menyangkal aksi yang dilakukan sebelumnya (menyangkal bahwa pesan tersebut berasal dirinya).

2.2 Algoritma Kriptografi

Algoritma Kriptografi adalah suatu fungsi matematis yang digunakan dalam proses enkripsi dan dekripsi. Dalam perkembangannya algoritma kriptografi terbagi menjadi dua macam yaitu: algoritma simetris (symmetric algorithms) dan algoritma asimetris (asymmetric algorithms).

Algoritma Simetris

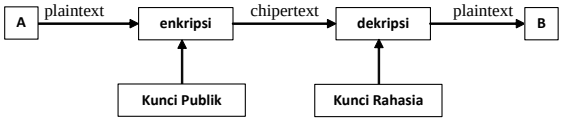
Algoritma Simetris adalah jenis algoritma kriptografi yang dalam proses enkripsi dan dekripsi menggunakan kunci yang sama. Algoritma ini mengharuskan pengirim dan penerima menentukan suatu kunci tertentu sebelum melakukan komunikasi. Keamanan algoritma simetris tergantung pada kunci tersebut, membocorkan kunci berarti bahwa orang lain dapat mengenkripsi dan mendekripsi pesan. Oleh karena itu algoritma simetris sering juga disebut dengan algoritma kunci rahasia, algoritma kunci tunggal, atau algoritma satu kunci.



Gambar 1. Diagram proses Algoritma Simetris

Algoritma Asimetris

Algoritma Asimetris atau yang lebih sering disebut dengan algoritma kunci publik menggunakan dua jenis kunci, yaitu kunci publik (publik key) dan kunci rahasia (secret key). Kunci publik merupakan kunci yang digunakan untuk mengenkripsi pesan dan bersifat umum, sehingga dapat diketahui oleh siapa saja. Sedangkan kunci rahasia digunakan untuk mendekripsi pesan dan bersifat rahasia, sehingga hanya dapat diketahui oleh orang yang memiliki otoritas.



Gambar 2. Diagram proses Algoritma Asimetris

2.3 Algoritma 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. Pemfaktoran dilakukan untuk memperoleh kunci pribadi. Selama pemfaktoran bilangan besar menjadi faktor-faktor prima belum ditemukan algoritma yang lebih efektif, maka selama itu pula keamanan algoritma RSA tetap terjamin. Kunci pada RSA mencakup dua buah kunci, yaitu public key dan private key. Public key digunakan untuk melakukan enkripsi, dan dapat diketahui oleh orang lain. Sedangkan private key tetap dirahasiakan dan digunakan untuk melakukan dekripsi.

2.4 Algoritma El-Gamal

Algoritma El-Gamal adalah algoritma kriptografi kunci-publik. Pertama kali dipublikasikan oleh Taher El-Gamal pada tahun 1985. Algoritma ini pada mulanya digunakan untuk digital signature, namun kemudian dimodifikasi sehingga juga bisa digunakan untuk enkripsi dan dekripsi. Kekuatan algoritma ini terletak pada sulitnya menghitung logaritma diskrit. 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 adalah probabilistik, yang berarti bahwa sebuah plaintext tunggal dapat dienkripsi

menjadi beberapa chipertext yang mungkin, yang konsekuensinya adalah proses enkripsi ElGama yang umum menghasilkan 2:1 perluasan pada ukuran dari plaintext ke chipertext. Algoritma El-Gamal terdiri dari tiga proses, yaitu proses pembentukan kunci, proses enkripsi dan proses dekripsi.

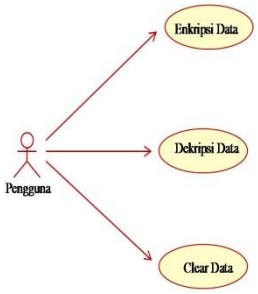
3. Analisis Sistem

3.1 Tinjauan Sistem

Software yang akan dikembangkan digunakan untuk melakukan simulasi proses kriptografi dengan Algoritma RSA dan Algoritma El-Gamal. Software yang dikembangkan ini akan dibagi menjadi beberapa frame yang berbeda sesuai dengan masing masing algoritma yang digunakan untuk mempermudah saat melakukan pembandingan hasil dari proses kriptografi. Software ini harus mampu melakukan proses pembentukan kunci berdasarkan inputan data dari pengguna serta mampu melakukan proses enkripsi dan dekripsi terhadap pesan text yang di input oleh pengguna. Software yang akan dikembangkan juga akan menampikan analisa waktu proses enkripsi dan dekripsi serta menampilkan perbedaan jumlah karakter yang dihasilkan pada proses tersebut.

3.2 Analisis Use Case

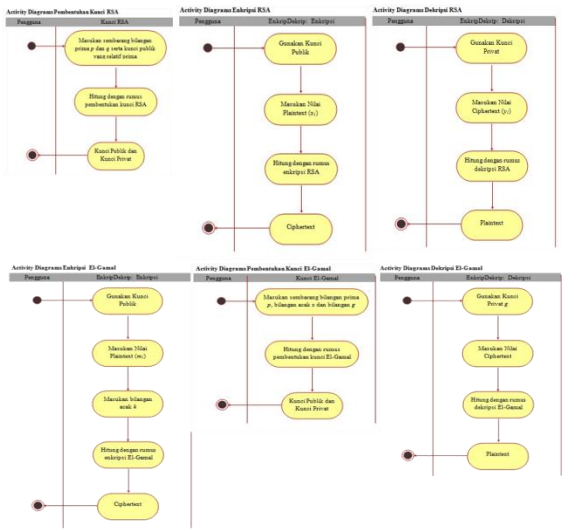
Use case adalah gambaran fungsionalitas dari suatu sistem, sehingga pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun. Gambaran Diagram Use case dibawah menjelaskan bahwa terdapat satu aktor yaitu pengguna. Pengguna ini akan melakukan tiga proses yaitu enkripsi data, dekripsi data dan clear data setelah melakukan kedua proses sebelumnya.



Gambar 3. Diagram Use case

3.3 Activity Diagrams

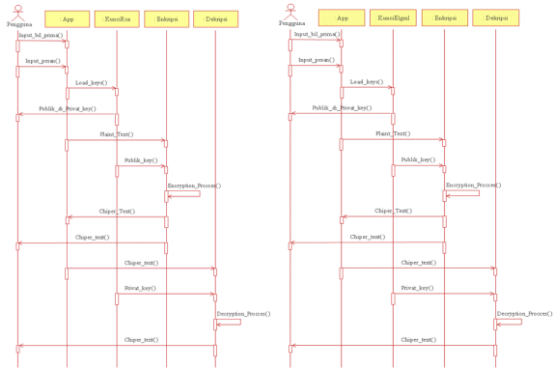
Berikut adalah gambaran activity diagrams yang digunakan dalam merancang software simulasi kriptografi algoritma RSA dan Algoritma El-Gamal.



Gambar 4. Activity Diagrams

3.4 Squence Diagrams

Sequence diagrams adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi diantara obyek-obyek tersebut.

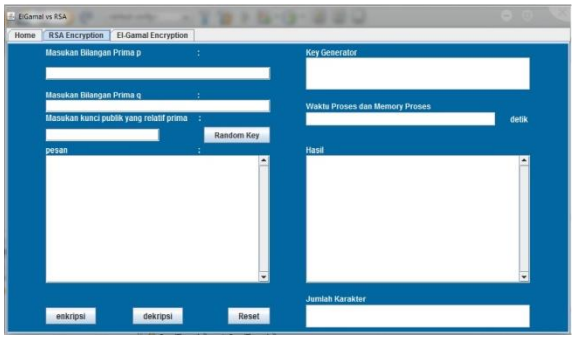


Gambar 5. Activity diagrams

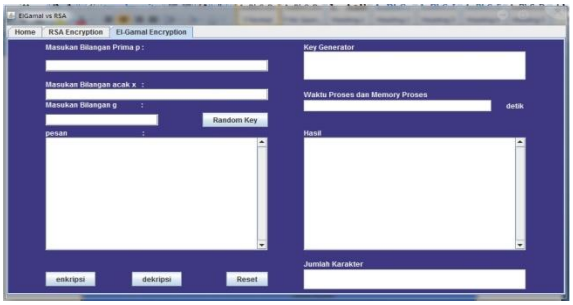
4. Pembahasan

Software yang akan dikembangkan digunakan untuk melakukan simulasi proses kriptografi dengan Algoritma RSA dan Algoritma El-Gamal. Software mempunyai 3 pilihan tab yaitu Home, RSA Encryption dan El-Gamal Encryption.

Gambar 6. Tampilan Awal



Gambar 7. Tampilan Algoritma RSA



Gambar 8. Tampilan Algoritma El-Gamal

Software yang dikembangkan mampu menampilkan parameter-parameter kinerja yang akan dibandingkan meliputi waktu proses, perubahan jumlah karakter, penggunaan CPU dan Memory.

5. HASIL PENGUJIAN

5.1 Skenario Pengujian

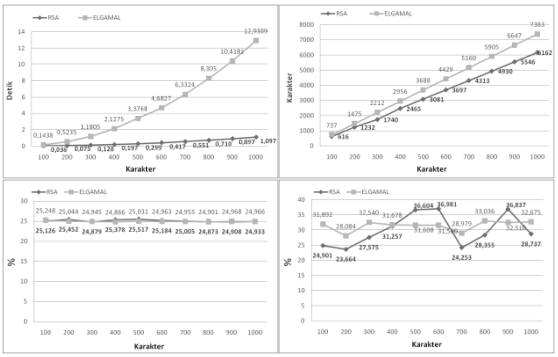
Pengujian dibagi menjadi beberapa tahap yaitu :

1. Pengujian terhadap perbedaan waktu proses, jumlah karakter chipertext, penggunaan CPU dan Memory akibat pengaruh perbedaan jumlah karakter pesan.
2. Pengujian terhadap perbedaan waktu proses, jumlah karakter chipertext, penggunaan CPU dan Memory akibat pengaruh perbedaan besaran kunci yang akan dibuat.
3. Membandingkan karakteristik chipertext hasil proses kriptografi dari masing-masing Algoritma.

5.2 Hasil Pengujian

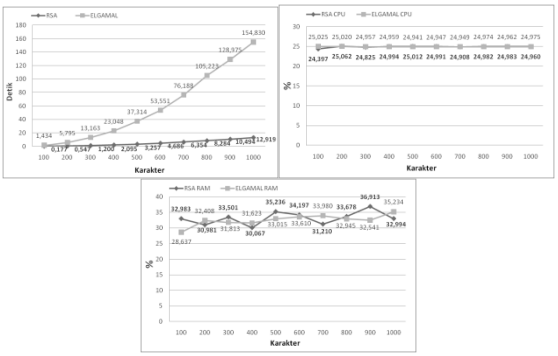
Pada bagian ini akan dijelaskan hasil pengujian terhadap sitem meliputi perbandingan hasil waktu proses, perubahan jumlah karakter, penggunaan CPU dan Memory serta perbedaan karakteristik hasil proses pada tiap Algoritma yang digunakan.

5.2.1 Pengaruh Jumlah Karakter Pesan Terhadap Waktu Proses, Jumlah Karakter Chipertext, Penggunaan CPU Dan Memory



Gambar 9. Perbandingan Proses Enkripsi Akibat Pengaruh Jumlah Karakter Pesan

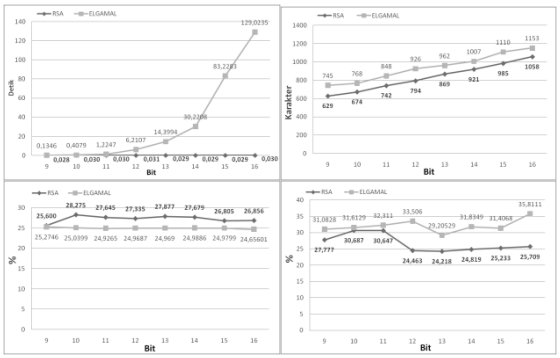
Berdasarkan grafik diatas terlihat waktu proses enkripsi dari algoritma El-Gamal membutuhkan waktu yang jauh lebih lama dan menghasilkan chipertext lebih besar yaitu 7,37-7,38 kali lebih besar dari jumlah pesan sedangkan algoritma RSA menghasilkan chipertext 5,8- 6,16 kali lebih besar dari jumlah pesan. Penggunaan CPU pada proses enkripsi dari masing-masing algoritma relatif seimbang yaitu berada pada kisaran 25%. Rata-rata penggunaan Memory algoritma RSA sedikit lebih unggul karena menggunakan Memory yang lebih sedikit yaitu sebesar 30,473% dibandingkan dengan algoritma El-Gamal yang menggunakan Memory rata-rata sebesar 31,407%.



Gambar 10. Perbandingan Proses Dekripsi Akibat Pengaruh Jumlah Karakter Pesan

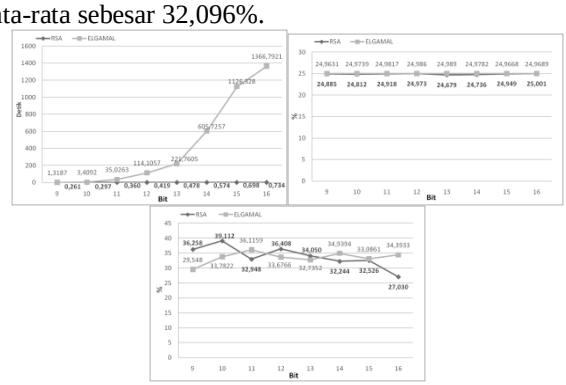
Berdasarkan grafik diatas terlihat waktu proses dekripsi dari algoritma El-Gamal membutuhkan waktu yang jauh lebih lama. Penggunaan CPU pada proses enkripsi dari masing-masing algoritma relatif simbang yaitu berada pada kisaran 25%. Apabila dilihat dari rata-rata penggunaan Memory dari masing-masing algoritma relatif simbang yaitu berada pada kisaran 33%.

5.2.2 Pengaruh Besaran Kunci Terhadap Waktu Proses, Jumlah Karakter Chipertext, Penggunaan CPU Dan Memory



Gambar 11. Perbandingan Proses Enkripsi Akibat Pengaruh Besaran Kunci

Berdasarkan grafik diatas terlihat waktu proses enkripsi dari algoritma El-Gamal membutuhkan waktu yang jauh lebih lama dan chipertext hasil proses enkripsi menggunakan algoritma El-Gamal jauh lebih besar dibandingkan menggunakan algoritma RSA. Perbedaan jumlah chipertext berkisar antara 9-18%. Dapat disimpulkan besaran kunci dapat mempengaruhi jumlah chipertext yang dihasilkan saat proses enkripsi. Algoritma RSA sebesar 6-10% dan Algoritma El-Gamal sebesar 3-10%. Penggunaan CPU pada proses enkripsi menggunakan algoritma RSA jauh lebih tinggi dibandingkan penggunaan CPU pada proses enkripsi menggunakan algoritma El-Gamal. Algoritma El-Gamal rata-rata menggunakan CPU sebesar 25% sedangkan algoritma RSA rata-rata sebesar 27%. Apabila dilihat dari rata-rata penggunaan Memory algoritma RSA sedikit lebih unggul karena menggunakan Memory yang lebih sedikit yaitu sebesar 26,694% dibandingkan dengan algoritma El-Gamal yang menggunakan Memory rata-rata sebesar 32,096%.



Gambar 12. Perbandingan Proses Dekripsi Akibat Pengaruh Besaran Kunci

Berdasarkan grafik diatas terlihat waktu proses dekripsi dari algoritma El-Gamal membutuhkan waktu yang jauh lebih lama. Penggunaan CPU pada proses enkripsi dari masing-masing algoritma relatif simbang yaitu berada pada kisaran 25%. Apabila dilihat dari rata-rata penggunaan Memory dari masing-masing algoritma relatif simbang yaitu berada pada kisaran 33%.

5.2.3 Perbandingan Chipertext Hasil Proses Enkripsi

Berdasarkan Tabel 5.1 terlihat bahwa chipertext yang dihasilkan oleh proses enkripsi menggunakan algoritma RSA selalu sama. Sedangkan chipertext yang dihasilkan oleh proses enkripsi menggunakan algoritma El-Gamal selalu berubah. Hal ini terjadi karena pada proses enkripsi menggunakan El-Gamal terdapat penambahan bilangan acak k sehingga meskipun pesan dan besaran kunci yang digunakan sama, chipertext yang dihasilkan akan berbeda dan akan semakin sulit untuk dipecahkan. Dapat disimpulkan algoritma El-Gamal memiliki keunggulan pada tingkat keamanan yang lebih tinggi.

TABEL 5.1. Perbandingan Chipertext Hasil Proses Enkripsi

RSA					
No.	Pembentukan Kunci (Bit)	PESAN	JUMLAH PESAN (Karakter)	ENKRIPSI	
				CHIPERTEXT	JUMLAH CHIPERTEXT (Karakter)
1	9	INI PESAN RAHASIA.	18	90367 103932 90367 9382 107821 18477 71206 84814 103932 9382 4317 84814 73507 84814 71206 90367 84814 21213	108
2	9	INI PESAN RAHASIA.	18	90367 103932 90367 9382 107821 18477 71206 84814 103932 9382 4317 84814 73507 84814 71206 90367 84814 21214	108
3	9	INI PESAN RAHASIA.	18	90367 103932 90367 9382 107821 18477 71206 84814 103932 9382 4317 84814 73507 84814 71206 90367 84814 21215	108
4	9	INI PESAN RAHASIA.	18	90367 103932 90367 9382 107821 18477 71206 84814 103932 9382 4317 84814 73507 84814 71206 90367 84814 21216	108
5	9	INI PESAN RAHASIA.	18	90367 103932 90367 9382 107821 18477 71206 84814 103932 9382 4317 84814 73507 84814 71206 90367 84814 21217	108
ELGAMAL					
No.	Pembentukan Kunci (Bit)	PESAN	JUMLAH PESAN (Karakter)	ENKRIPSI	
				CHIPERTEXT	JUMLAH CHIPERTEXT (Karakter)
1	9	INI PESAN RAHASIA.	18	104 83 111 394 372 201 338 321 252 1 226 290 155 428 306 466 386 124 202 127 384 407 457 160 329 433 347 167 142 266 250 197 404 42 162 213	140
2	9	INI PESAN RAHASIA.	18	454 13 356 149 260 143 260 43 304 273 443 134 324 51 412 217 222 417 58 455 387 1 354 162 307 270 222 108 276 114 132 63 340 213 474 359	137
3	9	INI PESAN RAHASIA.	18	32 98 138 234 81 213 55 401 206 403 193 3 355 37 136 405 80 439 96 140 224 409 6 472 261 404 99 310 466 171 93 191 125 317 117 273	131
4	9	INI PESAN RAHASIA.	18	206 242 149 231 431 158 326 256 113 190 64 147 227 460 115 114 47 25 284 163 252 13 225 402 320 210 269 251 201 154 458 31 281 112 61 142	138
5	9	INI PESAN RAHASIA.	18	229 282 12 474 463 391 425 82 315 92 198 338 157 107 280 316 100 152 156 295 345 470 76 440 92 151 164 45 310 346 322 4 117 417 469 476	136

6. KESIMPULAN

Berdasarkan pengujian dan analisis yang telah dibahas dan dilaksanakan, maka dapat disimpulkan beberapa hal sebagai berikut :

1. Algoritma RSA lebih unggul dalam kecepatan waktu proses enkripsi dan dekripsi yang jauh lebih cepat, serta jumlah karakter chipertext yang dihasilkan lebih sedikit.
2. Penggunaan CPU dan Memory pada kedua Algoritma relatif berimbang. Penggunaan CPU rata-rata sebesar 25% dan penggunaan Memory rata-rata sebesar 33%. Hal tersebut juga membuktikan penggunaan CPU dan Memory

tidak terpengaruh dengan jumlah pesan dan besaran kunci yang digunakan.

3. Algoritma El-Gamal memiliki keunggulan pada tingkat keamanan yang lebih tinggi. Hal ini terlihat dari chipertext yang dihasilkan akan selalu berbeda meskipun menggunakan kunci dan pesan yang sama. Meskipun hal tersebut membuat kecepatan waktu proses enkripsi dan dekripsi menjadi lebih lambat dan jumlah karakter chipertext yang dihasilkan lebih banyak.

DAFTAR PUSTAKA :

[1] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. "Chapter 8 Public Key Encryption". Handbook of Applied Cryptography.

[2] J. Slagell, Adam. 2001.A Simple, Portable And Expandable Cryptographic Application Program Interface.

[3] Johan Håstad, transcribed by Johan Linde. 2006. "Lecture 7: ElGamal and Discrete Logarithms". Foundations of Cryptography.

[4] Bailey, Tammy. 2004. "Lecture 16: Implementing RSA Encryption in Java". Principles Of Computer Science. http://db.cs.duke.edu/courses/cps001/ summer04/lectures/Lecture16.pdf..

Indra, Nikolaus. 2011. Analisis dan Perbandingan Kecepatan Algoritma RSA dan Algoritma ElGamal. Makalah IF3058 Kriptografi – Sem. II Tahun 2010/2011. Institut Teknologi Bandung.

[5] Chandra, Wiko. 2012. Perbandingan Algoritma Kunci Public RSA dan ElGamal. Makalah IF3058 Kriptografi – Sem. II Tahun 2011/2012. Institut Teknologi Bandung.

Mulya, Megah. 2013. Perbandingan Kecepatan Algoritma Kriptografi Asimetris. Journal of Research in Computer Science and Applications – Vol. I, No. 2,

[6] Januari2013. Universitas Sriwijaya.

Do My Best: How to get java cpu usage (jvm instance). http://knight76.blogspot.co.il/2009/05/how-to-get-java-cpu-usage-jvm-instance.html

Waktu akses : 12 Agustus 2014 pukul 21.15.

[7] Vogel, Lars. 2014. Java Performance - Memory and Runtime Analysis – Tutorial. http://www.vogella.com/tutorials/Java Performance/article.html

Waktu akses : 12 Agustus 2014 pukul 21.30