ISSN 2548-9712

ANALISIS HYBRID CRYPTOSYSTEM ALGORITMA ALGORITMA RSA DAN TRIPLE DES

Achmad Fauzi[1)], Novriyenni [2)], Yani Maulita[3)] *Akim M.H. Pardede*[4)]

STMIK KAPUTAMA
Jl. Veteran No. 4A-9A, Binjai, Medan, Sumatera Utara
Email: fauzyrivai88@gmail.com [1)], novri_yenni@yahoo.com[2)], yassa_26@ymail.com [3)]- akimmhp@live.com[4)]

Abstrak

Keamanan data sangat dibutuhkan dalam hal berkomunikasi. Untuk menjamin keamananan data dibutuhkan teknik untuk menyadikan data dan informasi yang disebut dengan Kritografi. Penelitian ini bertujuan untuk mengalisis proses Hybrid Didalam Kritografi simetris dan Asimetris yang mengunakan Algoritma RSA dan Algoritma Triple DES. Hal ini dapat meningkatkan keamanan data sehingga data menjadi lebih terjaga kerahasiaannya. Metode yang digunakan Algoritma RSA (Riverst – Shamir- Adleman) ini merupakan algoritma asimetris menggunakan sistem bilangan prima secara Acak dalam proses keamanannya dan Algoritma Triple DES yang disebut jugan dengan algoritma simetris adalah metode OFB (Output feeback), dan sehingga ketika kedua algoritma ini digabungkan dalam proses Hybrid maka keamanan datanya semakin akurat. Analisis Hybrid Kriptosistem algoritma RSA dan algoritma Triple DES menunjukan bahwa data yang dibuat secara text akan dienkripsi menjadi chipertext dengan menggunakan kedua metode tersebut dan di deskripsikan kembali. Sehingga keamanan data nya relative aman.

Keyword: *Hybrid Kritografi, Algoritma RSA, Algoritma Triple DES*

1. PENDAHULUAN

Kritografi sudah banyak digunakan oleh peneliti untuk menelitih tugasnya baik tugas akhir maupun jurnal. Menurut penelitihan dari jurnal maupun tugas akhir, banyak Algoritma didalam Kritografi untuk mengamankan Komputer dari kejahatan yang disengaja maupun tidak, adapun algotitma yang digunakan secara klasik dan modern. Keamananan didalam kritografi dikategorikan menjadi 2 yaitu keamanan secara off-line maupun on-line.

Dalam kriptografi terdapat beberapa algoritma yang dapat menyandikan data. Dalam algoritma enkripsi hybrid yang diusulkan, Algoritma Triple DES digunakan untuk transmisi data karena efisiensi yang lebih tinggi di enkripsi blok, dan algoritma RSA digunakan untuk enkripsi kunci Triple DES karena manajemen keuntungan dalam cipher key lebih akurat.(Kakarla & Govind, 2012). Berdasarkan kunci yang digunakan untuk enkripsi dan dekripsi, kriptografi dapat dibedakan menjadi kriptografi kunci simetris *(symmetric-key cryptography)* dan kriptografi kunci asimetris *(asymmetric-key cryptography).*(Dony Arius, 2008) Masing-masing memiliki kelebihan dan kekurangannya. Kriptografi simetris memiliki kelebihan waktu proses untuk enkripsi dan dekripsi relatif cepat, hal ini disebabkan karena efisiensi yang terjadi pada pembangkit kunci namun distribusi kunci tidak aman. Sementara kriptografi asimetris justru sebaliknya, masalah keamanan pada distribusi kunci dapat diatasi namun kecepatan proses enkripsi dan dekripsi tergolong lambat. Maka ada upaya untuk menggabungkan kedua jenis kriptografi tersebut sehingga dihasilkan tingkat keamanan yang tinggi dalam proses

ISSN 2548-9712

enkripsi dan dekripsi yang dikenal dengan metode *hybrid Kriptosistem.*

RSA *(Rivest—Shamir—Adleman)* merupakan algoritma yang proses enkripsi dan dekripsinya pada konsep bilangan prima dan aritmetika modulo. Baik kunci keduanya merupakan bilangan bulat. Kunci enkripsi tidak dirahasiakan dan diberikan kepada umum sehingga disebut dengan kunci publik, namun kunci untuk dekripsi bersifat rahasia (kunci privat). Kunci privat yang dibangkitkan dari beberapa buah bilangan prima bersama-sama dengan kunci enkripsi. Untuk menemukan kunci dekripsi, seseorang harus melakukan memfaktorkan suatu bilangan non prima menjadi faktor primanya. Kenyataannya, memfaktorkan bilangan non prima menjadi faktor primanya bukanlah pekerjaan yang mudah. Belum ada algoritma yang mangkus (efisien) yang ditemukan untuk pemfaktoran itu. Semakin besar bilangan non primanya, tentu semakin sulit pula pemfaktorannya. Semakin sulit pemfaktorannya, semakin kuat pula algoritma RSA.

Algoritma Triple DES berasal dari algoritma DES, karena itu kelemahan yang terdapat pada algoritma DES dapat ditemukan juga pada algoritma Triple DES meskipun ada sebagian kelemahan pada DES yang sudah diatasi di dalam Triple DES. Triple DES menggunakan kunci berukuran tiga kali lipat dari kunci DES yang berukuran 56 bits, karena itu untuk mencari kunci tersebut pada Triple DES diperlukan waktu yang lebih lama daripada DES. Kunci pada Triple DES dapat dipecahkan dengan menggunakan Differential Attack. Prinsip daripada serangan tersebut adalah dengan membandingkan pasangan plaintext dan pasangan ciphertext untuk menemukan kunci yang digunakan.

Berdasarkan uraian di atas, maka penulis tertarik untuk melakukan penelitian dengan menganalisis Algoritma RSA dan Algoritma Triple DES dengan metode *hybrid kriptosistem*. Algoritma RSA merupakan contoh kriptografi Kunci Publik dan Algoritma Triple DES, merupakan kriptografi Simetris dan asimetri yang menggunakan enkripsi simetris dan asimetri.

2. LANDASAN TEORI

2.1 Definiasi Kriptografi

Kriptografi adalah sebuah teknik dalam mengamankan dan mengirimkan data dalam bentuk yang hanya diketahui oleh pihak yang membukanya, sehingga dapat mengamankan informasi-informasi penting baik yang tersimpan dalam media penyimpanan maupun yang ditransmisikan melalui jaringan komunikasi.(Ariyus, 2008). Kriptografi adalah proses penggunaan berbagai teknik dan atau ilmu dan seni untuk menjaga keamanan pesan. Kriptografi adalah ilmu mengenai teknik enkripsi dimana data diacak menggunakan suatu kunci enkripsi menjadi sesuatu yang sulit dibaca oleh seseorang yang tidak memiliki kunci deskripsi. Deskripsi menggunakan kunci dekripsi mendapatkan data asli. Proses enkripsi dilakukan menggunakan suatu algoritma dengan beberapa parameter. Biasanya algoritma tidak dirahasiakan, bahkan enkripsi yang mengandalkan kerahasiaan algoritma dianggap sesuatu yang tidak baik. Rahasia terletak dibeberapa parameter yang digunakan, jadi kunci ditentukan oleh parameter. Parameter yang menentukan kunci dekripsi itulah yang harus dirahasiakan (parameter menjadi ekuivalen dengan kunci). Dalam kritografi klasik, teknik enkrips yang digunakan adalah enkrispsi simetris dimana kunci deskripsi sama dengan kunci enkrisi. (Kromodimoeljo, 2010)

2.2 Algoritma Kriptografi

Algoritma kriptografi merupakan langkah-langkah logis bagaimana menyembunyikan pesan dari orang-orang yang tidak berhak atas pesan tersebut. Algoritma kriptografi terdiri dari tiga fungsi dasar yaitu : (Ariyus, 2008)

1. Enkripsi merupakan hal yang sangat penting dalam kriptografi, merupakan pengamanan data yang dikirimkan agar terjaga kerahasiannya. Pesan asli disebut *plaintext*, yang diubah menjadi kode-kode yang tidak dimengerti. Enkripsi bisa diartikan dengan cipher atau kode. Untuk mengubah teks asli ke bentuk teks kode digunakan algoritma yang dapat mengkodekan data.
2. Dekripsi merupakan kebalikan dari enkripsi. Pesan yang telah dienkripsi

ISSN 2548-9712

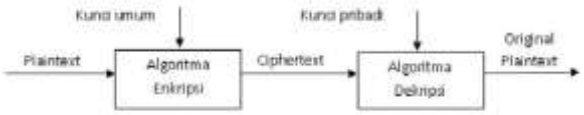
dikembalikan ke bentuk asalnya (teks asli/*plaintext*) disebut dengan dekripsi.

3. Kunci yang dipakai untuk melakukan enkripsi dan dekripsi. Kunci terbagi menjadi dua bagian yaitu kunci rahasia (*private key*) dan kunci umum (*public key*)

Biasanya algoritma kriptografi dapat dinotasikan sebagai berikut :

Plaintext(M)
Ciphertext(C)
Enkripsi (fungsi E)
Dekripsi (fungsi D)

Kriptografi itu sendiri terdiri dari dua proses utama yakni proses enkripsi dan proses dekripsi. Seperti yang telah dijelaskan di atas, proses enkripsi mengubah *plaintext* menjadi *ciphertext* (dengan menggunakan kunci tertentu) sehingga isi informasi pada pesan tersebut sukar dimengerti. Adapun alur dari proses enkripsi dan dekripsi pada kriptografi dapat dilihat pada gambar 1



Gambar 1 Konsep Proses Enkripsi dan Dekripsi
Sumber : Kriptografi, Dony Ariyus, Andi Publisher

2.6. Hibrid Kriptosystem

Hybrid kriptosistem merupakan gabungan antara *kriptosistem* yang memakai *asymmetric kriptosistem* dan *kriptosistem* yang memakai *symmetric kriptosistem.* (Schneier, 1996)

Kriptosistem adalah suatu fasilitas untuk mengkonversikan plaintext ke ciphertext dan sebaliknya, *kriptosistem* terdiri dari suatu algoritma seluruh kemungkinan plaintext, ciphertext, dan kunci.

Algoritma hibrid adalah algoritma yang memanfaatkan dua tingkatan kunci yaitu kunci rahasia (simetris) – yang disebut juga *session key* (kunci sesi) – untuk enkripsi data dan pasangan kunci rahasia – kunci public untuk pemberian tanda tangan digital serta melidungi kunci simetriss. (Ariyus, 2008)

Pada sistem hibrid ini enkripsi/dekripsi pesan menggunakan kriptografi kunci simetris, sedangkan kunci simetris dienkripsi/dekripsi dengan menggunakan kunci publik. Kunci simetris (yang disebut juga kunci sesi) dibangkitkan oleh salah satu pihak dan mengenkripsi pesan dengan kunci tersebut. Selanjutnya kunci sesi dienkripsikan dengan kunci publik penerima lalu dikirim bersama-sama dengan pesan yang sudah dienkripsi. Penerima mula-mula mendekripsikan kunci sesi dengan kunci privatnya, lalu mendekripsikan pasan dengan kunci sesi tersebut.

Kriptografi hibrid sering dipakai karena memanfaatkan keunggulan kecepatan pemrosesan data oleh algoritma simetris dan kemudahan transfer kunci menggunakan algoritma asimetris. Hal ini mengakibatkan peningkatan kecepatan tanpa mengurangi kenyamanan serta keamanan.

2.3 Manajemen Kunci

Aspek manajemen kunci sangat penting dalam aplikasi kriptografi. Manajemen kunci yang tidak baik dapat berakibat fatal. Proses pembuatan kunci sangat penting dan sebaiknya proses ini benar acak. Sumber acak (entropi) dapat diambil dari proses fisika acak seperti proses radio-aktif. Sumber acak dapat juga diambil dari berbagai kejadian (*events*) yang muncul secara acak. Aplikasi kriptografi dapat menggunakan *random number generator* yang disediakan *operating system* untuk pembuatan kunci, akan tetapi sebaiknya ini dilakukan hanya jika *random number generator* yang disediakan cukup acak. **(**Kromodimoeljo, 2010)

Distribusi kunci secara aman juga penting untuk keperluan pengamanan komunikasi. Sebagai contoh, untuk komunikasi yang diamankan dengan en- kripsi simetris, tentunya kedua mitra dalam komunikasi harus menggunakan kunci yang sama. Kunci ini dapat dibuat oleh satu pihak dan dikirim secara aman ke mitra komunikasi. Pengiriman kunci dapat dilakukan *out-of-band* yaitu menggunakan jalur khusus diluar jalur normal komunikasi, atau dilakukan *in-band* melalui jalur normal menggunakan sarana *public key*

ISSN 2548-9712

cryptography. Alternatif dari pengiriman kunci adalah *key agreement*, dimana kedua mitra berpartisipasi membuat kunci tanpa dapat diketahui oleh pihak ketiga. *Key agreement* juga menggunakan sarana *public key cryptography*.

Penyimpanan kunci jelas sangat penting untuk pengamanan sistem enkripsi secara menyeluruh. Kunci yang disimpan secara sembrono akan mudah untuk \dicuri" oleh pihak yang tidak diinginkan. Solusi untuk penyimpanan kunci beraneka ragam, mulai dari penggunaan hardware khusus dimana semua proses kriptografi dilakukan didalam hardware khusus dan kunci enkripsi disimpan dan tidak dapat keluar dari hardware, sampai dengan penyimpanan dalam file yang dienkripsi menggunakan password atau *passphrase*. Karena praktis, metode terakhir sangat populer, yang berarti pengamanan password menjadi penting.

Pengamanan password juga mempunyai beberapa masalah, dari masalah manusia seperti menulis password di secarik kertas yang ditempelkan ke meja kerja, sampai dengan masalah sistem seperti program yang menyimpan pass- word dalam bentuk teks.

Pada dasarnya masalah akses terhadap sesuatu yang penting seperti kunci enkripsi menjadi masalah *authentication* dan tren saat ini mengarah pada *mul- tiple factor authentication*. Kebenaran identititas seseorang atau sesuatu dinilai dari gabungan berbagai atribut yang cukup unik seperti sidik jari, pengetahuan password, dan kepemilikan sesuatu yang unik lainnya.

2.4 Algoritma RSA (**R**ivest—**S**hamir—**A**dleman).

RSA merupakan algoritma kriptografi asimetris. Ditemukan pertama kali pada tahun 1977 oleh Ron Rivest, Adi Shamir, dan Leonard Adleman. Nama RSA sendiri diambil dari inisial nama Ron Rivest, Adi Shamir, dan Leonard Adleman. Nama RSA sendiri diambil dari inisial nama depan ketiga penemunya tersebut. Sebagai algoritma kunci publik, RSA mempunyai dua kunci, yaitu kunci publik dan kunci pribadi. Kunci publik boleh diketahui oleh siapa saja, dan digunakan untuk proses enkripsi.

Sedangkan kunci pribadi hanya pihak - pihak tertentu saja yang boleh mengetahuinya, dan digunakan untuk proses dekripsi. Algoritma RSA masih digunakan hingga pada saat ini seperti yang diuraikan M. Zaki Riyanto dan Ardhi Ardhian: Keamanan sandi RSA terletak pada sulitnya memfaktorkan bilangan yang besar. Sampai saat ini RSA masih dipercaya dan digunakan secara luas di internet. (Ivan Wibowo, Budi Susanto & Junius Karel T, 2009).

Untuk membangkitan pasangan kunci RSA, digunakan algoritma sebagai berikut: (Parsi K, 2012)

1. Dipilih dua buah bilangan prima sembarang yang besar, p dan q. Nilai p dan q harus dirahasiakan.
2. Dihitung n = p x q. Besaran n tidak perlu dirahasiakan.
3. Dihitung fungsi Euler's totient $\phi(n) = (p - 1)(q - 1)$
4. Dipilih sebuah bilangan bulat sebagai kunci publik, disebut namanya e, yang relatif prima terhadap $\phi(n)$. e relatif prima terhadap $\phi(n)$ artinya faktor pembagi terbesar keduanya adalah 1, secara matematis disebut gcd (e, $\phi(n)$) = 1.
5. Dihitung diterminasi d dengan rumus $d = e^{-1} \pmod{\phi(n)}$, d adalah multiplikasi invers dari e mod $\phi(n)$
6. d sebagai komponen kunci private sehingga e * d mod (n) = 1
7. Kunci pablik mengandung modulo n dan eksponen e, sehingga (e,n)
8. Kunci privat mengandung modulo n dan eksponen d, sehingga (d,n)

Keamanan algoritma RSA terletak pada tingkat kesulitan dalam memfaktorkan bi- langan non prima menjadi faktor primanya, yang dalam hal ini n = p x q. Jika n berhasil difaktorkan menjadi p dan q, maka m = (p – 1)(q – 1) dapat dihitung. Dan karena kunci enkripsi e telah diumumkan (tidak dirahasi- akan), maka kunci dekripsi d dapat dihitung melalui persamaan (d x e) mod n = 1. Se-lama belum ditemukan cara untuk memfak-torkan bilangan besar menjadi faktor-faktor primanya, maka selama itu pula keamanan algoritma RSA terjamin.

ISSN 2548-9712

Penemu algoritma RSA menyarankan nilai p dan q panjangnya lebih dari 100 digit. Dengan demikian hasil kali n = p x q akan berukuran lebih dari 200 digit. Dengan asumsi bahwa algoritma pemfaktoran yang digunakan adalah algoritma yang tercepat saat ini dan komputer yang dipakai mem-punyai kecepatan 1 milidetik, menurut Rivest dan kawan-kawan, usaha untuk men-cari faktor bilangan 200 digit membutuhkan waktu komputasi selama 4 milyar tahun.

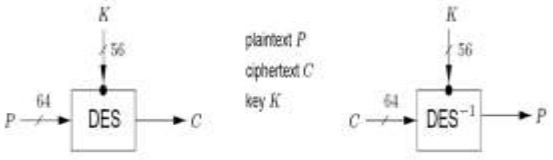
Adapun Proses Enkripsi dan dekripsi RSA antara lain sebagai berikut :

1. Proses Enkripsi :
   a. Nyatakan pesan menjadi blok-blok plainteks: m_1, m_2, m_3, … (harus dipenuhi persyaratan bahwa nilai m_i harus terletak dalam himpunan nilai 0, 1, 2, …, n – 1 untuk menjamin hasil perhitungan tidak berada di luar himpunan)
   b. Hitung blok cipherteks c_i untuk blok plainteks p_i dengan persamaan $c_i = m_i^e$ **mod** n dalam hal ini, e adalah kunci publik.
2. Proses Dekripsi :

2.5 Proses dekripsi dilakukan dengan menggunakan persamaan $m_i = c_i^d$ **mod** n, yang dalam hal ini, d adalah kunci privat.

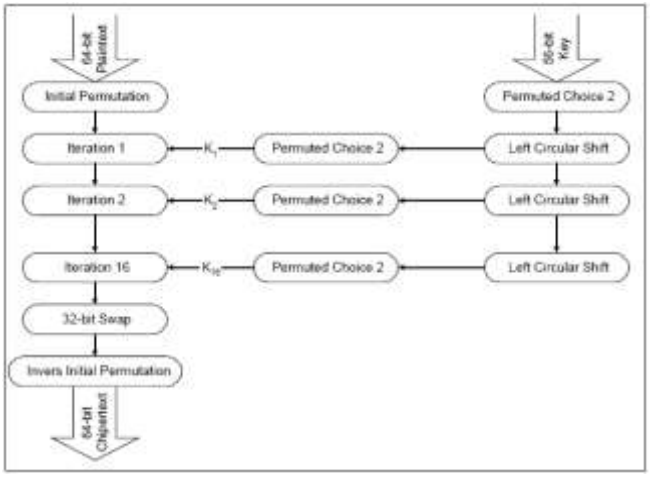
2.6 Algoritma DES

Data Encrytion Standar (DES) merupakan teknik atau algoritma kriptografi penyandian pesan yang menggunakan teknik blok simetris, Algoritma DES juga memerlukan sebuah kunci yang panjang bloknya 64 bit di setiap blok DES digunakan untuk mengamankan data pada perangkat lunak dan keras negara tersebut. Dan berikut desain *input-output* algoritma DES.



Gambar 2 DES input-output
Sumber : (ZHONG PuCha & BAO WanSu,2010)

Dapat dilihat bahwa ada dua input untuk fungsi enkripsi, yaitu *plaintext* dengan panjang 64-bit dan kunci dengan panjang 56-bit. Seperti pada gambar 3



Gambar 3 Tahapan proses enkripsi DES
Sumber : Munir, 2008

Untuk mengenkripsi data dengan menggunakan algoritma DES, dimulai dengan membagi bit dari teks tersebut kedalam blok-blok dengan ukuran blok sebesar 64-bit, yang kemudian disebut blok *plaintext*. Ukuran efektif dari kunci rahasia (*secret key*) *K* adalah *k* = 56 bit, masukan kunci (*input key*) *K* dispesifikasikan sebagai 64-bit kunci (*key*), dan 8-bit (bit 8, 16,…,64) digunakan sebagai *parity bit*. *Parity bit* tersebut akan mereduksi ukuran efektif *key* dari 64-bit menjadi 56-bit. Proses enkripsi dimulai dengan 16 iterasi blok *ciphertext* (disebut juga *round*) dengan menggunakan *initial permutation* (*IP*) dan diakhiri dengan *invers initial permutation* (IP^{-1}). (ZHONG PuCha & BAO WanSu, 2010)

Algoritma DES akan digunakan untuk enkripsi dan dekripsi dan mempunyai cara kerja yang hampir sama, hanya perbedaannya pada dekripsi pengiriman *subkey* yang terbalik. Untuk proses Dekripsi yang berdasarkan algoritma enkripsi dengan menggunakan kunci yang sama tetapi tetapi dengan membalikkan *key schedule*, yang dimulai dari *K*16,*K*15,…,*K*1. Efek dari *IP*-1

ISSN 2548-9712

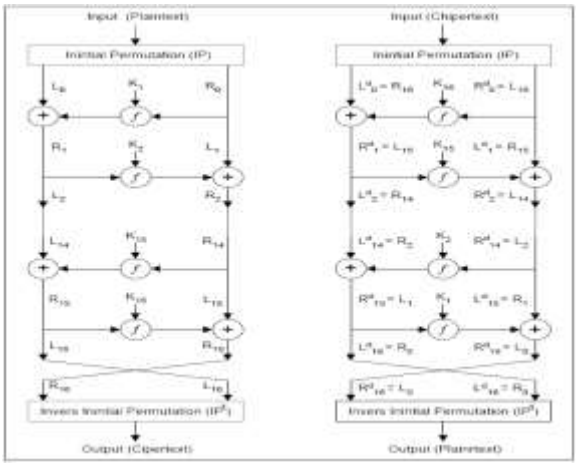
dibatalkan oleh *IP* pada proses dekripsi, membebaskan (*R*16, *L*16); dengan menggunakan *round-1* sebagai input. Operasi pada bagian kiri (*L*) adalah:

*L*0 *f*(*R*0, *Ki*) Maka : *R*16 ⊕⊕ *f*(*L*16, *K*16)

Setelah : *L*16 = *R*15 dan *R*16 = *L*15 ⊕ *f*(*R*15, *K*16) Ini sama dengan :

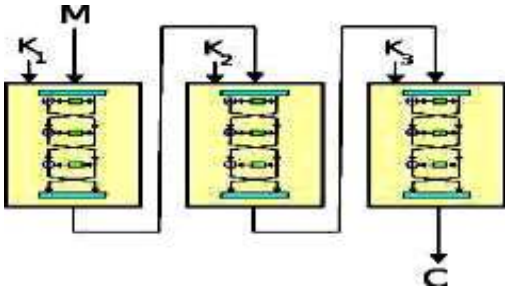
$L15\ f(R\oplus_{15}, K_{16}) \oplus f(R_{15}, K_{16}) = L15$

Jadi hasil dekripsi pada *round-1* adalah (R_{15}, L_{15}). Penggunaan *key schedule* pada dekripsi DES, *subkey* $K_1,..,K_{16}$ dibuat berdasarkan algoritma. Setelah K_{16} telah dibuat, nilai asli dari register 28-bit *C* dan *D* dimasukkan kembali (masing-masing di rotasi 28-bit). Untuk membuat *subkey* $K_1,\ldots,K_{16}$, caranya adalah dengan menukarkan rotasi *left-shift* dengan *right-shift* dan merubah nilai *shift v*1 = 0.Untuk melihat algoritma yang sama dalam urutan terbalik dapat dilihat pada Gambar 6 [*STALL95*]. Pada gambar tersebut dapat dilihat bahwa nilai berikutnya dari proses dekripsi adalah sama dengan nilai yang sama dari proses enkripsi dengan dua bagian dari nilai yang ditukarkan (*swap*). Dengan cara lain, dianggap bagian enkripsi yang ke-*i* dinotasikan dengan *Li* || *Ri* (*Li* dikonkantinasi dengan *Ri*). Kemudian untuk input (16-*i*) yang sama dari taraf dekripsi adalah *Ri* || *Li*.



Gambar. 4 Proses Enkripsi dan Dekripsi Algoritma DES
Sumber : Munir, 2008

Tahap pertama, plainteks yang diinputkan dioperasikan dengan kunci eksternal pertama (K1) dan melakukan proses enkripsi dengan menggunakan algoritma DES. Sehingga menghasilkan pra-cipherteks pertama. Tahap kedua, pra-cipherteks pertama yang dihasilkan pada tahap pertama, kemudian dioperasikan dengan kunci eksternal kedua (K2) dan melakukan proses enkripsi atau proses dekripsi (tergantung cara pengenkripsian yang digunakan) dengan menggunakan algoritma DES. Sehingga menghasilkan pra-cipherteks kedua. Tahap terakhir, pra-cipherteks kedua yang dihasilkan pada tahap kedua, dioperasikan dengan kunci eksternal ketiga (K3) dan melakukan proses enkripsi dengan menggunakan algoritma DES, sehingga menghasilkan cipherteks (C).



Gambar. 5 Algoritma Triple DE S(NIST, 2004)
Sumber : Munir, 2008

Keterangan pada gambar :

1. Dilakukan tiga kali DES dengan cara seri; hasil enkripsi DES1 menjadi masukan DES2 dan hasil enkripsi DES2 menjadi masukan DES3.
2. DES1, DES2, dan DES3 memiliki tiga kunci dengan kemungkinan:
   a. Ketiga kunci berlainan sama sekali.
   b. Kunci 1 dan kunci 2 berlainan; kunci 1 = kunci 3
   c. Ketiga kunci sama
3. Hasil enkripsi DES3 merupakan cipherteks yang dikirimkan.

3. METODOLOGI PENELITIAN

Subyek penelitian ini adalah menghibritkan algoritma triple DES dengan

ISSN 2548-9712

algoritma RSA untuk mengamankan pesan maupun sebuah data, sehingga nantinya dari proses keamanan akan menghasilkan algoritma yang baru yang mempunyai tingkat kesulitan pengamanan data yang tinggi dan cepat dalam proses enkripsi maupun dekripsi.

Adapun metodologi yang digunakan pada penyusunan penelitian diatas antara lain adalah : Studi pustaka, pengumpulan jurnal ilmiah, pengumpulan ebook dan uji coba progam.

4. ANALISA DAN PERANCANGAN

Bab ini membahas hasil penelitian yang dilakukan yaitu menganalisis hasil enkripsi dan deskripsi pesan, wkatu yang dibutuhkan untuk proses enkripsi dan deskripsi serta besarnya file setelah dienkripsi menggunakan algoritma RSA dan algoritma Triple Des dalam Hybrid Kritografi.

4.1 Proses Enkripsi pesan Teks – pengiriman menggunakan RSA

1. Misalkan plainteksnya adalah " FASILKOM" selanjutnya akan dilakukan proses enkripsi dengan algoritma RSA.

2. Setelah mendapatkan hasil Chiperteks Dari RSA. Proses enkripsi kunci RSA dilakukan mengkil enkripsi makan secara otomatis akan dibangkitkan bilangan secara acak pada algoritma RSA. Pengambilan bilangan prima P = 41 Q = 97 kemudian N= 3977, □□n) =3840, e = 79 dan d= 2479

   Hasil dari proses pembentukan kunci tersebut diperoleh :

   Kunci Publik : e = 79 dan N = 3977

   Kunci Private : d = 2479 dan N= 203

   Proses Enkrisi algoritma RSA dimana Desimalnya adalah :

   75 65 101 81 87 85 93 89

   75^79 mod 3977 = 2855
   65 ^ 79 mod 3977= 146
   101 ^ 79 mod 3977= 146
   81 ^ 79 mod 3977 = 2423
   87 ^ 79 mod 3977 = 3030
   85 ^ 79 mod 3977 = 2432
   93^ 79 mod 3977 = 3258
   89 ^ 79 mod 3977 = 1028

   Maka Hasil enkrisi algoritma RSA 2855, 146, 146, 2423, 3030, 2423, 3258, 1028

4.2 Proses Enkripsi pesan Algoritma Triple DES

Proses selanjutnya hasil cipherteks pada algoritma RSA akan dirubah kedalam bentuk bilangan biner sepanjang 64 bit dan masing-masing akan dibagi menjadi dua bagian yaitu 32 bit per bagian dan akan melakukan proses enkripsi dengan algoritma triple des sebagai berikut :

Langkah-Langkah dalam perhitungan proses enkripsi pesan teks menggunakan Kritografi Triple Des adalah Sebagai Berikut :

1. Plainteks : 2855, 146, 146, 2423, 3030, 2423, 3258, 1028 dan Rubahlah menjadi biner.

Rubahlah angka Plainteks 2855, 146, 146, 2423, 3030, 2423, 3258, 1028 Menjadi Biner

0	1	0	0	0	1	1	0
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	1
0	1	0	0	1	0	0	1
0	1	0	0	1	1	0	0
0	1	0	0	1	0	1	1
0	1	0	0	1	1	1	1
0	1	0	0	1	1	0	1

2. Lalu Dapatlah Panjang 64 Bits
   0100011001000001010100110100100
   10011000100101101001111010011101
3. Lalu plainteks dengan panjang 64 bits dibagi menjadi 32 bits
   P:10001100100000101010011010010010100110001001011010011110100 1101
4. Lalu masukan Kunci, Boleh Dibuat dengan menggunakan abjad ataupun bilangan hexa. Adapun kunci Triple Des:

   K **75**
   O **79**
   M **77**
   P **80**

ISSN 2548-9712

	U	**85**
	T	**84**
	E	**69**
	R	**82**

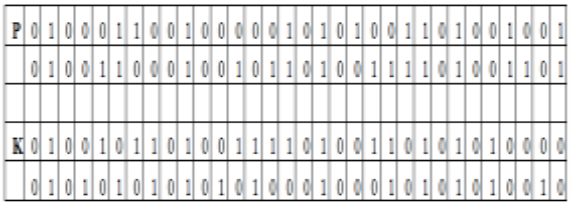
5. Lalu ubahlah Kunci menjadi biner

5	0	0	1	0	1	1
	0	0	1	1		
	0	0	1	1		
	0	0	0	0		
	0	0	0	1		
	0	0	0	1		
	0	0	0	1		
	0	0	0	0		

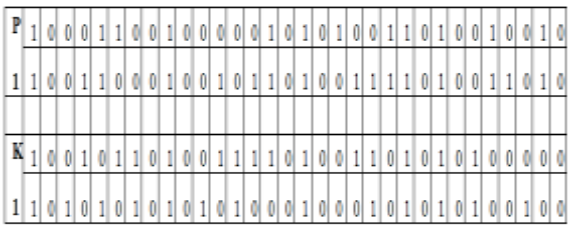
6. Panjang Kunci Dibuat menjadi 64 Bits
   0100101101001111010011010100000
   1010101010101000100010101010010

7. Kunci 64 bits dibagi menjadi 32 bits
   K:
   0100101101001111010011010100000
   0101010101010100010001010101010010

9. Maka kita Melakukan Metode OFB dari kiri ke kanan



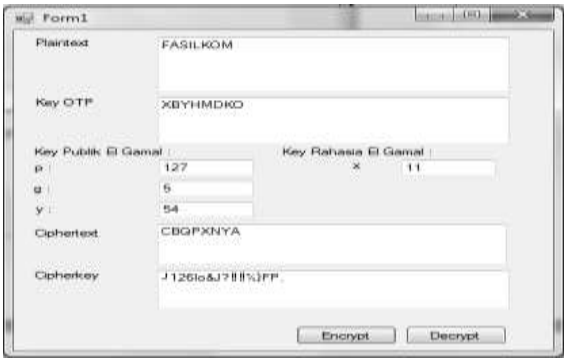
Puteran 1



Setelah selesai putaran 1 selanjutnya akan dipermutasikan sampai putaran ke 16 setelah itu akan dioperasikan dengan menggunakan operasi XOR sehingga mendapatkan cipherteks. Dan dengan cara yang sama proses dekripsinya.

3. HASIL DAN PEMBAHASAN

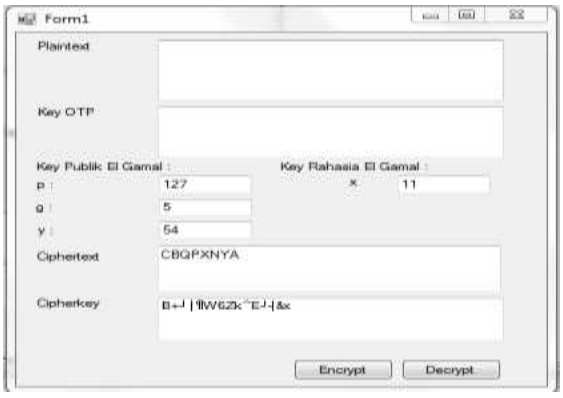
Ada pun hasil proses enkripsi dari algoritma OTP dan algoritma ElGamal sehingga didapatlah hasil *ciphertext* dan *cipherkey*nya. Untuk lebih jelasnya dapat dilihat pada gambar dibawah ini.



Gambar 6 Tampilan Hasil Enkripsi

Untuk dekripsinya terlebih dahulu kolom padaplaintext dan key OTP dikosongkan atau dihapus seperti yang terlihat pada gambar dibawah ini.

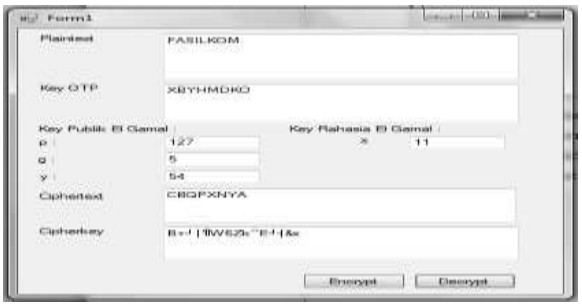
Ada pun hasil implementasi sebelum proses dekripsi dari algoritma OTP dan algoritma ElGamal sehingga didapatlah hasil *ciphertext* dan *cipherkey*nya. Untuk lebih jelasnya dapat dilihat pada gambar dibawah ini



Gambar 7 Tampilan sebelum di dekripsi

Setelah itu tekan tombol decrypt untuk melakukan proses dekripsi pesan OTP dan kunci OTP sehingga didpatlah kembalih hasil plaintext dan kunci OTPnya. Untuk lebih jelasnya dapat dilihat pada gambar dibawah ini.

ISSN 2548-9712



Gambar 8 Tampilan setelah dilakukan proses dekripsi pesan dan kunci

4. KESIMPULAN DAN SARAN

Apabila kita menggunakan suatu algoritma yang sudah ada bahkan yang sudah memiliki source code nya, maka dengan mudah pesan tersebut akan mudah dibobol dengan menggunakan algoritma yang sudah ada. Dari beberapa percobaan yang dilakukan dari menggabungkan algoritma One Time Pad dan ElGamal yang ada ini, dapat diambil kesimpulan :

1. Merupakan suatu Super Enkripsi algoritma yang baru.
2. Dapat diimplementasikan pada keamanan pesan.
3. Dapat menutupi kelemahan One Time Pad yaitu dimana panjang kunci sama dengan panjang pesan sehingga saat melakukan pengiriman pada dua saluran komunikasi kunci OTP memerlukan perlindungan. Kelemahan ini sudah ditutupi dengan algoritma ElGamal yang melakukan enkripsi terhadap kunci OTP sehingga keamanan kunci dari OTP terjaga begitu juga dengan pesannya. Dengan melakukan enkripsi pada kunci OTP mempunyai dua keuntungan yaitu kerahasian kunci terjaga dan juga dapat mengecoh pembobol karena mereka bisa saja berpikir bahwa kunci hasil dari enkripsi ElGamal ini kunci OTP yang asli.

Saran untuk perbaikan penelitian ini agar lebih baik yaitu:

1. Untuk pengembangannya dapat dilakukan pengacakan kunci dari OTP secara random atau acak.
2. Untuk bilangan prima pada kunci ElGamal dapat dikembangkan dengan melakukan pengujian menggunakan algoritma untuk mengetest keprimaan suatu angka.

DAFTAR PUSTAKA

[1] Ariyus, Dony. 2008. *Pengantar Ilmu Kriptografi :Teori, Analisis dan Implementasi*, Penerbit Andi:Yogyakarta.

[2] Ariyus, Dony. 2006. Computer Security. Penerbit Andi:Yogyakarta.

[3] Fauziah Yuli, 2008. *Pengamanan Pesan Dalam Editor Teks Mengunakan Hybrid Cryptosystem. SemnasIF*

[4] Kromodimoeljo Sentot, 2010, *Teori & Ap likasi Kriptografi*, SPK IT Consulting

[5] Munir, Rinaldi. 2006. *Kriptografi.* Penerbit Informatika:Bandung.

[6] Madhur, Kapil.,Yadav, Singh, Jitendra.& Vijay, Ashish, 2012. *Modified Elgamal over RSA Digital Signature Algorithm (MERDSA). International Journal of Advanced Research in Computer Science and Software Engeneering(1*): 2277-128X

[7] Mollin Richard, 2007 *An Introduction to Cryptography*, Taylor & Francis Group

[8] Munir Rinaldi, 2006, *Kriptografi.* Penerbit informatika, Bandung

[9] Sadikin Rifki, 2012, *Kriptografi untuk keamanan jaringan*, CV Andi Offset,Yogyakarta

[10] Schneier, Bruce., 1996, *Applied Cryptography : Protocols, Algorithms, and Source Code in C*, 2nd Edition John Wiley & Sons Inc.