ISSN : 2548-9704

PERANCANGAN APLIKASI KEAMANAN PESAN MENGGUNAKAN ALGORITMA ELGAMAL DENGAN MEMANFAATKAN ALGORITMA ONE TIME PAD SEBAGAI PEMBANGKIT KUNCI

Achmad Fauzi[1)], Yani Maulita[2)], Novriyenni[3)]

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

Abstrak

Pengamanan pesan diperlukan dalam rangka untuk mencegah pesan yang didistribusikan dapat dibuka oleh pihak lain yang tidak berkepentingan di mana pada akhirnya dapat mengancam kemanan dan kenyamanan dari si pengirim maupun penerima pesan tersebut. Untuk mengamankan pesan tersebut dalam dilakukan penerapan ilmu kriptografi yang bertujuan untuk mengubah pesan asli (plaintext) menjadi pesan terenkripsi (ciphertext), di mana untuk membukapesan tersebut memerlukan kunci.Algoritma One Time Paddikenal dengan nama holy grail algorithm dikarenakan algoritma kriptografi One Time Pad adalah algoritma yang sempurna yang tidak bisa dipecahkanbiarpun begitu algoritma One Time Pad memiliki kelemahan dalam menjaga kerahasiaan atau keamanan kunci sehingga harus diberikan pengamanan pada kunci agar kunci dari OTP itu selama pengiriman terjaga kerahasiaanya. Sedangkan pada algoritma asimetri atau kunci publik ada algoritma Elgamal yang juga mempunyai keamanan yang tinggi karena kompleksitas algoritmanya.Dengan disuper enkripsikannya algoritma one time pad dan ElGamal tersebutdapat meningkatkan keamanan pada pesan dan juga dapat menjaga kerahasiaan atau keamanan kunci dari one time pad selama proses pengiriman pesan dan kunci.

Kata Kunci : ElGamal, Kriptografi, One Time Pad , Pengamanan.

1. PENDAHULUAN

1.1 Latarbelakang

Keamanan merupakan masalah besar dan mengamankan data yang penting sangat penting, sehingga data tersebut tidak dapat disadap atau disalahgunakan untuk tujuan ilegal sehingga merugikan pihak lain. Untuk itulah pemerintah dan lembaga lainnya berusaha mengamankan data mereka sekuat tenaga agar tidak terjadi pembobolan. Biarpun begitu tetap aja ada pihak-pihak yang berusaha membobol itu dengan menggunakan berbagai kunci dan juga metode.Untuk menghindari hal tersebut maka data yang dikirim diubah kedalam data yang tidak dapat dibaca oleh sang pembajak kemudian data tersebut diubah kembali dalam bentuk yang bisa dibaca oleh penerimanya. Teknik dan ilmu untuk membuat data yang tidak dapat dibaca sehingga hanya orang yang berwenang yang mampu membaca data, inilah yang disebut dengan kriptografi (Goyal & Kinger, 2013). Kriptografi secara umum adalah ilmu dan seni untuk menjaga kerahasiaan pesan ketika pesan dikirim dari suatu tempat ke tempat yang lain.

Pada kriptografi terdapat proses enkripsi dan dekripsi yang bertujuan untuk mengamankan data. Enkripsi adalah mengubah pesan asli (*plaintext*) menjadi pesan dalam bentuk tersandi (*ciphertext*). Proses enkripsi akan menghasilkan data tersandi dan hanya dapat dibuka atau dibaca oleh pihak penerima yang memiliki kunci (key) sedangkan proses dekripsi adalah proses mengembalikan data tersebut menjadi bentuk asli. (Munir, 2006). Menurut kunci yang digunakan kriptografi terbagi atas dua yaitu kriptografi asimetri dan kriptografi simetri. Kriptografi simetri adalah kriptografi yang menggunakan kunci yang sama saat proses enkripsi dan dekripsi sedangkan kriptografi asimetri adalah kriptografi yang proses enkripsi dan dekripsi menggunakan dua kunci yang berbeda. Pada kriptografi simetri terdapat sebuah algoritma kriptografi klasik yang dikenal sebagai *holy grail algorithm*.

ISSN : 2548-9704

Algoritma itu dikenal dengan nama One Time Pad dikarenakan algoritma kriptografi One Time Pad adalah algoritma yang sempurna yang tidak bisa dipecahkan biarpun begitu algoritma One Time Pad memiliki kelemahan dalam menjaga kerahasiaan atau keamanan kunci sehingga harus diberikan pengamanan pada kunci agar kunci dari OTP itu selama pengiriman terjaga kerahasiaanya (Stamp, 2011). Sedangkan pada algoritma asimetri atau kunci publik ada algoritma Elgamal yang juga mempunyai keamanan yang tinggi karena kompleksitas algoritmanya.

Berdasarkan alasan itu penulis mencoba melakukan penelitian dengan menggabungkan algoritma simetris yang diambil dr algoritma klasik yaitu *One Time Pad*(OTP) dengan algoritma asimetris atau algoritma kunci publik yaitu algoritma ElGamal yang dimana nantinya algoritma ElGamal akan digunakan untuk mengamankan kunci dari One Time Pad sebelum kunci dikirim ke penerima. Dengan disuper enkripsinya algoritma nantinya akan memunculkan suatu super enkripsi algoritma yang dapat meningkatkan keamanan sehingga pesan lebih sulit dipecahkan dan juga keamanan kunci OTP dapat terjaga dengan aman sehingga pihak ketiga tidak mudah menjebol pesan yang dikirim.

1.2 Rumusan Masalah

Dalam rumusan masalah diatas penulis merumuskan bagaimana proses enkripsi algoritma ElGamal dan algoritma One Time Pad dapat dibagun sehingga dapat meningkatkan keamanan pesan dan kunci.

1.3 Tujuan Penelitian

Tujuan peneliti disini adalah menganalisa kerja enkripsi algoritma ElGamal dan algoritma One Time Pad untuk proses keamanan pada teks sehingga dari super enkripsi algoritma tersebut dapat meningkatkan keamanan pada pesan dan juga dapat menjaga kerahasiaan atau keamanan kunci selama proses pengiriman

2. LANDASAN TEORI

2.1 Definisi Kriptografi

Kriptografi adalah merupakan ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data serta otentikasi. Kriptografi adalah proses penggunaan berbagai teknik dan atau ilmu dan seni untuk menjaga keamanan pesan. *Cryptographic algorithm* adalah fungsi matematika yang digunakan untuk enkripsi dan dekripsi. Terdapat dua fungsi yang saling berhubungan yaitu satu untuk enkripsi dan satu lagi untuk dekripsi. Enkripsi merupakan proses pengkodean sebuah pesan sehingga isi dari pesan tersebut tidak diketahui. Dekripsi adalah proses kebalikan dari enkripsi yaitu mentransformasi pesan yang dienkripsi kembali menjadi bentuk semula. Sebuah sistem enkripsi dan dekripsi disebut *cryptosystem*. Bentuk asli dari sebuah pesan disebut *plaintext* dan bentuk asli yang dienkripsi disebut *ciphertext*.

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)

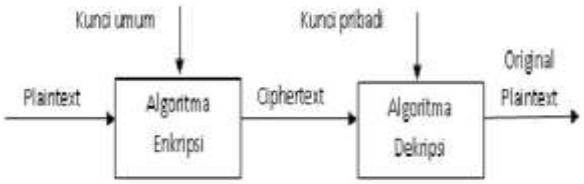
1. Enkripsi merupakan hal yang sangat penting dalam kriptografi, merupakan pengamanan data yang dikirimkan agar terjaga kerahasiannya. Pesan asli disebut *plaintext*, yang diubah menjadi kode-kode yang tidak dimengerti. Enkripsi bisa diartikan dengan cipher atau kode. Untuk mengubah teks asli ke bentuk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

ISSN : 2548-9704

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 2.1



Gambar 2.1 Konsep Proses Enkripsi dan Dekripsi
Sumber : Kriptografi, Dony Ariyus, Andi Publisher

2.3 Super Enkripsi

Super Enkripsi merupakan gabungan antara *cryptosystem* yang memakai *asymmetric cryptosystem* dan *cryptosystem* yang memakai *symmetric cryptosystem.* (Schneier, 1996). Cryptosystem adalah suatu fasilitas untuk mengkonversikan plaintext ke ciphertext dan sebaliknya, cryptosystem terdiri dari suatu algoritma seluruh kemungkinan plaintext, ciphertext, dan kunci.

Algoritma super enkripsi 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). Kriptografi super enkripsi 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.

2.4 Pembangkit Bilangan Acak Semu

Pembangkit Bilangan Acak-Semu atau yang biasa dikenal dengan singkatan PRNG (*Pseudo-Random Number Generator*) adalah sebuah algoritma untuk menghasilkan suatu urutan bilangan yang terlihat acak, namun sebenarnya urutan tersebut tidak benar-benar acak karena urutan tersebut ditentukan oleh suatu nilai awal. Urutan bilangan yang terlihat acak ini sangat penting karena bisa dimanfaatkan untuk suatu parameter bagi percobaan atau simulasi dan juga menjadi pusat pake praktik kriptografi.

Sebuah pembangkit bilangan acak-semu bisa dimulai dengan memberikan nilai umpan. Pembangkit bilangan acak-semu ini akan selalu memberikan urutan bilangan yang sama jika diberikan nilai umpan yang sama, dengan jumlah bilangan yang dihasilkan bergantung kepada besar nilai umpan yang diukur dengan satuan bit.

Keuntungan dari penggunaan pembangkit bilangan acak-semu ini adalah efisien, algoritma ini mampu menghasilkan banyak angka dalam waktu singkat, dan tertentu, urutan yang digunakan bisa dimunculkan kembali dengan mudah jika nilai awalnya diketahui. Efisien adalah karakteristik yang sangat baik jika aplikasi kita membutuhkan banyak angka. Tertentu juga akan berguna jika kita perlu mengulang suatu urutan bilangan.

2.5 Pembangkit Bilangan Prima

Sebagian besar algoritma kunci-publik menggunakan bilangan prima sebagai salah satu nilai parameternya. Bilangan prima yang disarankan berkuran besar sehingga penggunaan tipe data bilangan bulat yang besar mutlak diperlukan. Dalam menghasilkan bilangan prima dapat digunakan berbagai metode. Agar dapat menghasilkan bilangan prima yang besar maka harus menggunakan ruang memori dan waktu.

Secara umum pembangkitan bilangan prima dapat dibagi menjadi dua, yaitu dengan membangkitkan bilangan prima dari bilangan prima terkecil dengan pengujian yang akan menghasilkan bilangan prima dengan persentase 100% atau dengan menguji bilangan acak dan kemudian menguji apakah bilangan tersebut termasuk bilangan

ISSN : 2548-9704

prima.

2.6 One Time Pad

One Time Pad termasuk dalam kriptografi klasik yang berkunci simetris. One Time Pad disebut juga sebagai algoritma yang tidak terpecahkan atau juga diketahui sebagai *holy grail algorithm*. (Horstmeyer, Judkewitz, Vellekoop, Assawawarrarit & Yamg, 2013). Algoritma *One Time Pad* mempunyai cara kerja dimana penerima pesan mempunya salinan kunci yang sama dan kunci tersebut hanya dipakai satu kali (*one time*) untuk enkripsi dan dekripsi dan setelah digunakan maka pad (kertas *blocknot*)harus segera dihancurkan agar tidak bisa dipakai lagi untuk enkripsi dan dekripsi pesan yang lain.Pengirim danpenerimaharus sama-samamemiliki satu set materi kunci yang besar dan juga acak, selamakombinasi dari semuapesan yang pernah dikirimkan.

Jadi secara teori alasan OTP tidak dapat dipecahkan jika kuncinya secara sempurna diacak, dirahasiakan dan hanya dipakai sekali saja. (Nemati & Yang, 2011) Pada algoritma OTP mempunyai panjang kunci yang sama dengan panjang *plaintext.*Sehingga tidak ada kebutuhan untuk mengulang penggunaaan kunci selama proses enkripsi.

Pada algoritma OTP mempunyai panjang kunci yang sama dengan panjang *plaintext.* Sehingga tidak ada kebutuhan untuk mengulang penggunaaan kunci selama proses enkripsi.

Adapun aturan enkripsi dan dekripsi dari *one time pad* adalah sebagai berikut:

1. Enkripsi $C_i = (P_i + K_i) \bmod 26$
2. Dekripsi $C_i=(P_i - K_i) \bmod 26$

Skema OTP tidak dapat dipecahkan karena alasan sebagai berikut : (Ariyus, 2008)

1. Barisan kunci acak + teks asli yang tidak acak = teks kode yang seluruhnya acak.
2. Mendekripsi teks kode dengan berbagai kunci berbeda dapat menghasilkan plainteks yang beragam sehingga kripnatalis tidak punya cara untuk menemukan plainteks mana yang benar.

Meskipun OTP merupakan suatu algoritma yang sempurna dan aman, tetapi dalam praktik modern jarang digunakan karena disebabkan oleh panjang kunci = panjang pesan, sehingga timbul masalah dalam menjaga kerahasiaan kunci selama proses pendistribusian kunci (Stamp, 2011).

2.7 El Gamal

Algoritma ElGamal ditemukan pada tahun 1985 oleh ilmuwan Mesir yaitu Taher ElGamal. Algoritma ElGamal merupakan algoritma berdasarkan konsep kunci publik. Algoritma ini pada umumnya digunakan untuk digital signature, namun kemudian dimodifikasi sehingga bisa digunakan untuk enkripsi dan dekripsi. Algoritma kriptografi kunci publik ElGamal merupakan algoritma blok chipper yaitu algoritma yang melakukan proses enkripsi pada blok-blok plainteks yang kemudian menghasilkan blok-blok chipertext, yang nantinya blok-blok chipertext tersebut akan didekripsi kembali dan hasilnya kemudian digabungkan menjadi plainteks semula. Keamanan algoritma ElGamal terletak pada kesulitan perhitungan logaritma diskrit pada modulo prima yang besar, sehingga upaya untuk menyelesaikan masalah logaritma ini menjadi sulit untuk dipecahkan. Algoritma ini memiliki kelebihan yaitu pembangkitan kunci yang menggunakan logaritma diskrit dan metode enkripsi dekripsi yang menggunakan proses komputasi yang besar sehingga hasil enkripsinya berukuran dua kali dari ukuran semula. Kekurangan algoritma ini adalah membutuhkan resource yang besar karena chipertext yang dihasilkan dua kali panjang plaintext serta membutuhkan processor yang mampu untuk melakukan komputasi yang besar untuk perhitungan logaritma perpangkatan besar. (Madhur, Yadav, dan Vijay, 2012)

Satu karakter yang direpresentasikan dengan menggunakan bilangan bulat ASCII akan menghasilkan kode dalam bentuk blok yang terdiri atas dua nilai (a, b). *(Rashmi Singh, Shiv Kumar, 2012)*

a) Ambil sebuah karakter dalam pesan yang akan dienkripsi dan transformasi karakter tersebut ke dalam kode ASCII sehingga diperoleh bilangan bulat m. Plainteks tersebut disusun menjadi blok-blok m1, m2, …, sedemikian hingga setiap blok merepresentasikan nilai di dalam rentang 0 (nol) sampai p-1.

ISSN : 2548-9704

b) Memilih bilangan acak k, yang dalam hal ini 0 < k < p-1, sedemikian hingga k relative prima dengan p-1.

c) Hitung nilai a dan b dengan persamaan berikut :

$a = g^k$ $(mod\ p)$ **….……(4)**

$b = y^k$ $m\ (mod\ p)$ **……..(5)**

d) Diperoleh cipherteks untuk karakter m tersebut dalam blok (a,b)

e) Melakukan proses di atas untuk seluruh karakter dalam pesan termasuk karakter spasi.

Dekripsi dari cipherteks ke plainteks menggunakan kunci rahasia a yang disimpan kerahasiaanya oleh penerima pesan.
Teorema :
Diberikan (p,g, y) sebagai kunci public dan x sebagai kunci rahasia pada algoritma ElGamal. Jika diberikan cipherteks (a, b), maka

$$m = b/a^x \bmod p \ldots\ldots\ldots (4)$$

dengan M adalah plainteks.
Di mana nilai

$$(ax)^{-1} = r^{-a} = rp^{-1-a} \bmod p. \ldots (5)$$

a. Ambil sebuah blok cipherteks dari pesan yang telah dienkripsikan pengirim.

b. Dengan menggunakan a yang dirahasiakan oleh penerima, hitung nilai plainteks dengan menggunakan "persamaan (4)" dan "persamaan (5)".

Secara garis besar algoritma el-gamal mempunya langkah-langkah pembentukan kunci sebagai berikut :

a. Bilangan prima, p (bersifat public atau tidak rahasia)
b. Bilangan acak, g (dimana g < p dan bersifat public atau tidak rahasia)
c. Bilangan acak, x (dimana x < p dan bersifat private atau rahasia)
d. Bilangan acak, k (dimana k < p dan bersifat private atau rahasia)
e. m merupakan plainteks dan bersifat private/rahasia
f. a dan b merupakan pasangan chiperteks hasil enkripsi bersifat private atau tidak rahasia.

Proses Pembentukan kunci Algoritma ElGamal Proses pembentukan kunci merupakan proses penentuan suatu bilangan yang kemudian akan digunakan sebagai kunci pada proses enkripsi dan dekripsi pesan. Kunci untuk enkripsi dibangkitkan dari nilai p, g, y sedangkan kunci untuk dekripsi terdiri dari nilai x, p. Masing-masing nilai mempunyai persyaratan yang harus dipenuhi.

Langkah-langkah dalam pembuatan kunci adalah sebagai berikut :

1. Pilih sembarang bilangan prima p, dengan syarat p > 255.
2. Pilih bilangan acak g dengan syarat g < p.
3. Pilih bilangan acak x dengan syarat $1 \leq x \leq p - 2$.
4. Hitung $y = g^{\wedge x} \bmod p$.

Kunci public nya adalah y, g, p sedangkan kunci private nya adalah x. nilai y, g, dan p tidak dirahasiakan sedangkan nilai x harus dirahasiakan karena merupakan kunci privat untuk mendekripsi plainteks. (Rashmi Singh, Shiv Kumar, 2012).

3. METODOLOGI PENELITIAN

Subyek penelitian ini adalah Memanfaatkan algoritma kriptografi Elgamal dan One Time Pad (OTP) dalam penyandian 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

4.1 Proses Enkripsi dan Dekripsi Algoritma One Time Pad

Pada tahap awal enkripsi, pesan yang dibutuhkan adalah berupa teks yang akan dienkripsi, dengan menggunakan kunci acak dimana panjang kunci harus sama dengan *plaintext*. Langkah-langkah proses enkripsinya :

Dimasukkan plainteks adalah : FASILKOM dengan kunci yang dibangkitkan secara acak yaitu XBYHMDKO.Misalkan nilai setiap huruf atau karakter yaitu sebagai berikut :

Tabel 3.1 Nilai Karakter

A	B	C	D	E	F	G	H	I	J	K
0	1	2	3	4	5	6	7	8	9	10

Sampai huruf Z = 25

ISSN : 2548-9704

Hitung proses enkripsi dengan menggunakan rumus *ci = (pi + ki)* mod 26.

(F + X) mod 26 = (5 + 23) mod 26 = 28 mod 26 = 2 = C
(A + B) mod 26 = (0 + 1) mod 26 = 1 mod 26 = 1 = B
(S + Y) mod 26 = (18 + 24)mod 26 = 42 mod 26 = 16 = Q
(I + H) mod 26 = (8 + 7) mod 26 = 15 mod 26 = 15 = P
(L + M) mod 26 = (11 + 12) mod 26 = 23 mod 26 = 23 = X
(K + D) mod 26 = (10 + 3) mod 26 = 13 mod 26 = 13 = N
(O + K) mod 26 = (14 + 10) mod 26 = 24 mod 26 = 24 = Y
(M + O) mod 26 =(12 + 14) mod 26 = 26 mod 26 = 0 = A

Berdasarkan perhitungan diatas diperolehlah *ciphertext* CBQPXNYA. Ciphertext yang didapat inilah yang akan dikirim ke penerima. Kemudian kunci dari OTP yaitu "XBYHMDKO" terlebih dahulu akan diamakan menggunakan ElGamal sebelum dikirim.

4.2 Proses Pembangkitan Kunci Elgamal

Pada proses ini dilakukan pembangkitan kunci publik dan privat dari elgamal yang nantinya akan digunakan untuk proses pengamanan kunci dari OTP "XBYHMDKO".

Berikut urutan langkah-langkah proses pembangkitan kunci :

1. Memilih sebuah bilangan acak prima yang diberi simbol p
   p = 127
2. Menentukan akar primitvie α modulo p
   nilai α modulo p = 13
3. memilih bilangan acak a dengan syarat $2 \leq a \leq p - 1$
   a = 17
4. Hitung nilai $\beta = \alpha^a \bmod p$
   $\beta = 13^{17} \bmod 127 = 44$

Hasil pembangkitan kunci adalah
kunci publik adalah triple (44, 13, 127)
kunci private adalah pasangan (17, 127)

Dua pasangan kunci publik dan kunci privat yang sudah ditentukan ini yang akan digunakan pada enkripsi untuk mengamankan kunci OTP.

4.3 Proses Enkripsi Elgamal

Setelah penentuan bilangan prima p = 127 dan elemen primitif α = 13, maka terbentuklah hasil kunci publik (127, 13, 44), langkah selanjutnya adalah melakukan enkripsi pesan kedalam bentuk *chipertext.* Adapun urutan proses enkripsi pesan tersebut adalah :

1. Masukkan teks yang akan dienkripsi yaitu berupa kunci dari OTP yang akan diamanankan.
   Plaintext = XBYHMDKO
2. Pesan dipotong menjadi blok-blok karakter dan dikonversi ke dalam kode ASCII.

Tabel 3.2 Konversi Blok Karakter ke dalam kode ASCII

I	Karakter	Plainteks m_i (ASCII)
1	X	88
2	B	66
3	Y	89
4	H	72
5	M	77
6	D	68
7	K	75
8	O	79

3. Langkah selanjutnya, menentukan bilangan acak k ∈{1, 2, ……, 126}, kemudian dilakukan perhitungan a = α^k mod p dan b = β^k.m mod p.

Tabel 3.3 Perhitungan Enkripsi *plaintext*

I	m_i	K	α^k mod p	b = β^k.m mod p
1	88	71	69	8
2	66	107	49	12
3	89	47	84	101
4	72	67	113	104
5	77	103	60	39
6	68	89	98	122
7	75	79	62	51
8	79	29	41	18

4. Kemudian hasil perhitungan disusun dengan pola (a_1, b_1,a_2, b_2, ….., a_i, b_i) maka didapatlah *ciphertext*nya sebagai berikut :
   69 8 49 12 84 101 113 104 60 39 98 122 62 51 41 18
   Kunci OTP yang sudah diamankan inilah yang nantinya akan dikirim ke penerima.

4.4 Proses Dekripsi

Pada proses ini penerima melakukan dekripsi terhadap kunci OTP yang telah diamankan oleh Elgamal terlebih dahulu

ISSN : 2548-9704

kemudian baru digunakan untuk membuka pesan yang dienkripsi oleh OTP.

4.5 Proses Dekripsi Elgamal

Setelah melakukan proses enkripsi maka untuk membuka kembali *plaintext* yang sudah dienkrip dilakukan proses dekripsi agar kunci OTP dapat dibaca oleh penerima dan digunakan untuk mendekripsi pesan OTP. Adapaun urutan proses dekripsi adalah sebagai berikut :

1. Masukkan *ciphertext* yang akan didekripsi
   Ciphertext = 69 8 49 12 84 101 113 104 60 39 98 122 62 51 41 18
2. Langkah selanjutnya dilakukan perhitungan $m_i = b_i \cdot a_i^{p-1-x} \bmod p$

Tabel 3.4 Perhitungan dekripsi *ciphertext*

i	a_i	b_i	$m_i = b_i \cdot a_i^{p-1-x} \bmod p$	Karakter
1	69	8	88	X
2	49	12	66	B
3	84	101	89	Y
4	113	104	72	H
5	60	39	77	M
6	98	122	68	D
7	62	51	75	K
8	41	18	79	O

3. Didapatlah kembali kunci OTP XBYHMDKO yang kemudian digunakan untuk mendekripsi pesan oleh OTP.

4.6 Proses dekripsi One Time Pad

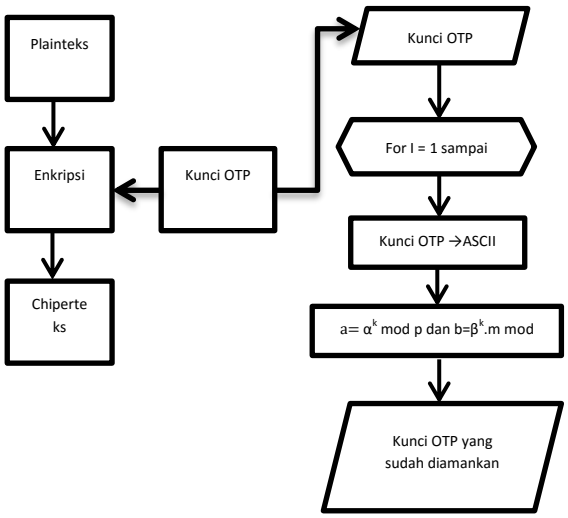
Pada proses ini penerima melakukan dekripsi atas ciphertext yang dikirim dengan menggunakan kunci OTP yang telah didekripsin untuk mendekripsikannya *ciphertext* pada One-time pad digunakanlah rumus di = (c*i* – *ki*) mod 26. cara yang dilakukan adalah sebagai berikut :

(C - X) mod 26 = (2 - 23) mod 26 = -22 mod 26 = 5 = F
(B - B) mod 26 = (1 - 1) mod 26 = 0 mod 26 = 0 = A
(Q - Y) mod 26 = (16 - 24) mod 26 = -8 mod 26 = 18 = S
(P - H) mod 26 = (15 - 7) mod 26 = 8 mod 26 = 8 = I
(X - M) mod 26 = (23 - 12) mod 26 = 11 mod 26 = 11 = L
(N - D) mod 26 = (13 - 3) mod 26 = 10 mod 26 = 10 = K
(Y - K) mod 26 = (24 - 10) mod 26 = 14 mod 26 = 14 = O
(A - O) mod 26 =(0 - 14) mod 26 = -14 mod 26 = 12 = M

Jika terdapat nilai minus atau negatif dalam perhitungan maka ditambahkan 26 untuk membuat angkanya menjadi positif. Maka dari perhitungan diatas penerima mendapatkan kembali *plaintext* yang dikirim yaitu“FASILKOM”

4.7 Proses Rancangan yang berjalan

Tahapan ini dimulai dengan terlebih dahulu mengamankan pesan dengan cara setiap karakter dari pesan dienkrip dengan kunci yang ada, kemudian kunci dari OTP (*One Time Pad*) diamankan lagi menggunakan Elgamal agar keamanan kunci saat didistribusi tidak bisa dijebol dan juga untuk mengecoh pihak ketiga. Tahapan dalam mengamankan kunci dimulai dengan pembentukan bilangan prima *p*, menentukan akar primitif α dan menentukan bilangan bulat x, proses selanjutnya menghitung $\beta = \alpha^x \bmod p$ dan kemudian akan didapatkan kunci publik (p, α, β) dan kunci *private*(x, p). setelah proses pembentukan kunci Elgamal, dilanjutkan ke proses enkripsi pesan dimana pesan di Elgamal adalah kunci dari OTP yang akan diamankan. Adapun alur dari proses enkripsi pesan dengan OTP dan enkripsi kunci OTP yang akan diamankan dapat dilihat pada gambar 3.1.



Gambar 4.1. *Flowchart* proses enkripsi pesan – kunci

ISSN : 2548-9704

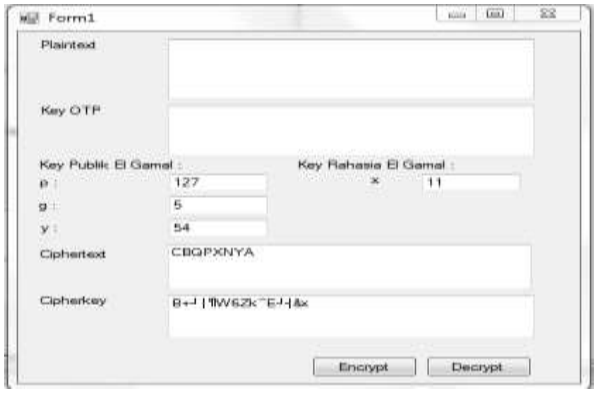
5. HASIL DAN PEMBAHASAN

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 5.1 Tampilan Hasil Enkripsi

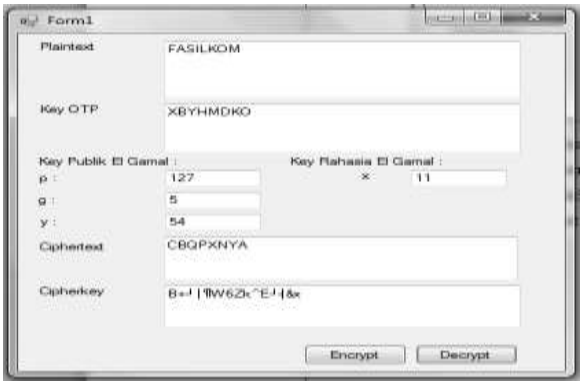
Untuk dekripsinya terlebih dahulu kolom padaplaintext dan key OTP dikosongkan atau dihapus seperti yang terlihat pada gambar dibawah ini.

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 5.2 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.



Gambar 5.3 Tampilan setelah dilakukan proses dekripsi pesan dan kunci

6. 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

ISSN : 2548-9704

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 & Aplikasi 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