**KOMBINASI ELGAMAL DAN KNAPSACK DENGAN**

**PEMBANGKIT KUNCI *LINEAR CONGRUENTIAL***

***GENERATOR* PADA TEKS**

Risya Farisi, Nerfita Nikentari, Ferdi Chahyadi

risya.farisi@gmail.com

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Maritim Raja Ali Haji

**Abstract**

 *In securing data can be done by way of encryption or commonly called Cryptography. The purpose of using Cryptography itself is to make the data sent difficult to read by third parties who are not responsible for stealing data. In cryptographic text, a key generator is used as an automatic generating process, namely the Linear Congruential Generator in the encryption process to make it easier to process by the Elgamal and Knapsack algorithms using a combination of cryptographic algorithms, where messages will be encoded using the ElGamal algorithm, then the results (ciphertext) will be encrypted. again using the Knapsack algorithm, ElGamal Algorithm and Knapsack meet the data integrity requirements. The results of the message decryption test will be the same as the initial plaintext (the original message). And the results of the ciphertext decryption test will be the same as the key that has been used for the text encryption process.*

*Keywords: ElGamal, Linear Congruential Generator,Knapsack,Teks*

1. **Pendahuluan**

Algoritma ElGamal sudah lama digunakan sebagai dasar pada digital *signature*, di era sekarang ElGamal digunakan dan dimodifikasi sebagai proses enkripsi dan deskripsi pesan ElGamal sendiri merupakan Algoritma terbaik yang digunakan dalam kriptografi. ElGamal di temukan sekitar tahun 1985 oleh Taher ElGamal. Algoritma ini membutuhkan perhitungan logaritma diskrit yang cukup rumit pada modulo prima yang besar, jadi akan cukup sulit untuk memecahkan logaritma ini. Hal itu membuat algoritma ini bisa dibilang cukup terjamin keamanannya (Sholeh, 2018).

Dalam Penelitian (Mariana dan Purba, 2019) yang berjudul “Implementasi Algoritma Knapsack Dan *Base64* Pada Pengamanan File Teks “ menyatakan bahwa Algoritma ElGamal merupakan algoritma yang baik untuk kriptografi apabila dikombnasikan dengan Algoritma Knapsack yang juga sama-sama bagus untuk kriptografi. Knapsack merupakan algoritma kriptografi asimetris yang menggunakan kunci publik untuk mengenkripsi pesan dan menggunakan kunci pribadi untuk mendeskripsikan pesan. Keamanan algoritma ini terletak pada sulitnya memecahkan persoalan knapsack (*Knapsack Problem*). Sebelum melakukan proses enkripsi, pertama-tama *user* harus menciptakan pasangan kunci publik dan kunci privat terlebih dahulu, yang nantinya akan digunakan untuk melakukan proses enkripsi dan dekripsi pada algoritma Knapsack. Untuk melakukan proses pembentukan kunci, jalankan aplikasi sehingga akan muncul halaman utama. Lalu klik menu Pembentukan Kunci, sehingga halaman pembentukan kunci akan muncul.

Dalam kriptografi teks digunakan sebuah kunci pembangkit sebagai proses generate otomatis pada proses enkripsi agar lebih mudah untuk di proses oleh algoritma ElGamal dan Knpasack dengan menggunakan kombinasi algoritma kriptografi, dimana pesan akan disandikan menggunakan algoritma ElGamal, lalu hasilnya (*cipherteks*) akan disandikan lagi menggunakan algoritma Knapsack diharapkan dapat memperkuat keamanan dari pesan rahasia, sehingga akan lebih sulit untuk dipecahkan.

Metode kriptografi yang digunakan dalam pengamanan teks ini adalah ElGamal,Knapsack dan *Linear Congruential Generator* yang merupakan sebuah metode kriptografi paduan antara asimetris dan simetris yang paling cepat dan sederhana. Hal ini sesuai dengan sifat dari algortima simetris yang membentuk kunci enkripsi dan deskripsi hanya dengan satu kunci (*single-key algorithm*) dan algoritma simetris yang memiliki dua kunci berbeda untuk proses enkripsi dan dekripsi.

1. **Metode Penelitian**

**2.1 Kriptografi**

Kriptografi merupakan salah satu teknik yang digunakan di dalam mengamankan data atau informasi yang bersifat penting dan rahasia. Teknik kriptografi sebagai salah satu teknik keamanan memiliki banyak algoritma untuk mengimplementasikan fungsinya. Algoritma tersebut seperti *Hill Cipher, Triangle Chain Cipher, Affine Cipher, DES, GOST,* RSA, ElGamal dan lainnya. Kriptografi berasal dari bahasa Yunani yaitu “*kriptos*” dan “*graphia*”.

Kata *kriptos* untuk mendeskripsikan pada sesuatu yang disembunyikan atau di samarkan. Sedangkan kata *graphia* lebih ke arah tulisan yang berbentuk. Secara keseluruhan kriptografi berarti bentuk tulisan yang bisa di samarkan sebagai rahasia atau Kriptografi adalah suatu ilmu dari sebuah teknik enkripsi “naskah asli” (*plaintext*) diacak menjadi enkripsi “naskah acak yang sulit dibaca” (*ciphertext*) oleh seseorang pihak ketiga. Kriptografi sejak jaman dahulu sudah di gunakan sebagai pengaman rahasia tulisan dengan menggeser karakter suatu nilai (Lombu dkk, 2018).

**2.1 Proses Pembentukan Kunci LCG**

*Linear Congruential Generator* (LCG) adalah algoritma yang sering diimplementasikan pada beberapa bahasa pemrograman untuk membangkitkan bilangan acak. Kunci pembangkit adalah X0 yang disebut seed (*secret seed*). Bilangan acak yang dihasilkan oleh *Linear Congruential Generator* ini akan digunakan untuk menentukan posisi dari bilangan 1 sampai n pada setiap baris. Cara penentuannya adalah sebagai berikut (Yunarti, 2018) :

1. Menentukan nilai dari setiap variable dimana nilai modulus (m) selalu nilai paling besar atau nilai jumlah data dari keseluruhan di tambah 1.
2. Melakukan random nilai
3. Pengeksekusian nilai random oleh rumus
4. Hasil random

**2.2 Proses Enkripsi ElGamal**

Adapun langkah – langkah Enkripsi algoritma ElGamal dilakukan sebagai berikut (Afif, 2018) :

1. Pilih karakter plaintext dan konversikan ke dalam bentuk desimal berdasarkan tabel ASCII yang disebut dengan m.
2. Tentukan angka acak x dan g yang merupakan bilangan prima kemudian Bangkitkan angka acak x dengan syarat 1 ≤ x ≤ p-2.
3. Hitung nilai a dan b menggunakan persamaan.

 a = gx mod p

 b = yx m mod p

1. Hasil perhitungan akan mendapatkan blok *ciphertext* dari karakter m dalam blok (a,b).
2. Lakukan langkah 1 dan 3 hingga karakter *plaintext* terakhir.
3. Gabungkan setiap blok (a,b) yang didapat untuk mendapatkan seluruh ciphertext yang utuh.

**2.3 Proses Dekripsi ElGamal**

Proses dekripsi menggunakan kunci pribadi x dan p untuk mendekripsi a dan b menjadi plaintext m dengan persamaan berikut (Afif, 2018) :

 (ax)-1 = ap-1-x mod p

 m = b\*ax mod p

Sehingga plaintext dapat ditemukan kembali dari pasangan *ciphertext* a dan b. Adapun langkah – langkah deskripsi ElGamal dilakukan sebagai berikut :

1. Ambil blok *ciphertext* hasil enkripsi oleh pengirim Dengan menggunakan bilangan x, yang menjadi bagian dari kunci rahasia,
2. Maka lakukan pengoperasian terhadap blok *ciphertext* dengan persamaan diatas.
3. Hasil dekripsi akan dalam bentuk desimal, konversikan desimal kembali ke dalam karakter berdasarkan tabel ASCII.
4. Ulangi langkah 1 dan 3 hingga blok *ciphertext* terakhir.
5. Gabungkan setiap karakter yang dihasilkan untuk mendapatkan *plaintext*

**2.4****Proses Enkripsi Knapsack**

Setelah Kunci dibangkitkan pada *supercreasing* knapsack kemudian proses enkripsi dilakukan Berikut ini adalah langkah – langkah enkripsi Algoritma Knapsack dengan tahapan sebagai berikut (Munir, 2006) :

1. Enkripsi dilakukan dengan dengan menggunakan algoritma Knapsack.
2. Mula-mula *plaintext* dipecah menjadi blok bit yang panjangnya sama dengan kardinalitas barisan kunci publik.
3. Kalikan setiap bit di dalam blok dengan elemen yang berkoresponden di dalam kunci publik.

**2.5 Proses Dekripsi Knapsack**

Berikut ini adalah langkah – langkah dekripsi Algoritma Knapsack Proses enkripsi dilakukan dengan tahapan sebagai berikut (Wina, 2019) :

1. Dekripsi dilakukan dengan menggunakan kunci privat.
2. Awalnya penerima pesan menghitung n–1 , yaitu balikan n modulo m, sedemikian sehingga

 n . n–1 ≡ 1 (mod m). Kekongruenan ini dapat dihitung dengan persamaan sebagai berikut :

n .n–1 ≡ 1 (mod m)

n .n–1 = 1 + km

n–1 = (1 + km)/n dengan k sembarang bilangan bulat.

1. Kalikan dengan n–1 mod m, lalu nyatakan hasil kalinya sebagai penjumlahan elemen-elemen kunci privat untuk memperoleh *plaintext* dengan menggunakan algoritma pencarian menjadi *Superincreasing* Knapsack.

**2.7 Bahan atau Materi Penelitian**

Bahan atau materi yang digunakan dalam penelitian ini adalah data primer berupa pengujian kata pada aplikasi.

**2.8 Kerangka Pikir Penelitian**

Adapun Kerangka Pikir Penelitian yang dilakukan pada implementasi Sistem keamanan ini adalah sebagai berikut :



 Gambar 1. Kerangka pikir penelitian

Kerangka pikir penelitian diawali dengan penelitian dimulai dengan studi literatur dengan mencari referensi informasi yang diperoleh dari jurnal - jurnal ilmiah, dan hasil- hasil penelitian mahasiswa dalam berbagai bentuk misalnya skripsi, tesis, laporan praktikum, dan sebagainya. Selanjutnya mengidentifikasi masalah, mencari masalah di mana suatu objek tertentu sebagai suatu masalah. Selanjutnya focus penelitian dalam melakukan penelitian Kemudian membangun aplikasi. Setelah aplikasi selesai maka akan dilakukan pengujian, jika aplikasi sesuai dengan yang diharapkan maka peneliti dapat menarik kesimpulan jika tidak maka kembali ke proses pembangunan aplikasi dan selesai.

**2.9 Flowchart Enkripsi**

Mulai

Pembangkit kunci

Bangkitkan kunci dengan *Linear Congruential generator*

Tidak

Kunci bilangan prima?

Ya

Kunci , Plaintext

Kunci , Ciphertext

Ciphertext

Selesai

Enkripsi dengan Knapsack

Enkripsi dengan ElGamal

Gambar 2. Flowchart Enkripsi

Pada gambar 2 Sistem melakukan proses input nilai pembangkit kunci terlebih dahulu, kemudian sistem akan memproses nilai kunci dengan *Linear Congruential Generator* apabila kunci bukan bilangan prima maka proses akan kembali dengan pembangkitan nilai lain dengan *Linear Congruential Generator* jika nilai kunci adalah bilangan prima maka hasil *output* akan mengeluarkan kunci. Kemudian input *plaintext* yang akan digunakan pada sistem , sistem akan memproses *plaintext* dengan enkripsi ElGamal setelah itu akan keluar outputan berupa kunci ElGamal dan *output* *Ciphertext* yang akan dienkripsikan lagi dengan Knapsack, hasilnya adalah berupa *ciphertext final* dari proses dalam bentuk akan setelah itu enkripsi selesai.

**2.10 Flowchart Dekripsi**

Pada gambar 3 Sistem mendapatkan hasil outputan kunci Knapsack dan *ciphertext* final dari hasil enkripsi sebelumnya oleh ElGamal dan Knapsack, hasil outputan tersebut langsung dieksekusi oleh sistem, dimana *ciphertext* dan kunci apakah sudah sesuai dengan kunci privat yang dibentuk sebelumnya jika tidak maka sistem akan mengulang proses dengan dekripsi Knapsack, kemudian hasil dekripsi akan mengeluarkan *output* dari kunci ElGamal apakah kunci sesuai jika kunci sesuai maka sistem akan melakukan proses enkripsi dengan Knapsack dan dikonversikan kembali menjadi *plaintext* dan selesai.

Mulai

Kunci Knapsack dan Ciphertext

Tidak Kunci

Sesuai

Ya

Deskripsi dengan Knapsack

Kunci ElGamal

Tidak

Kunci Sesuai

Ya

Enkripsi dengan Knapsack

Plaintext

Selesai

Gambar 3. Flowchart Dekripsi

1. **Hasil dan Pembahasan**

**3.1** **Hasil Penelitian**

Data yang akan digunakan pada penelitian adalah beberapa data berjenis teks

**3.2 Pengujian dan hasil Enkripsi dan Dekripsi**

Pengujian dilakukan untuk mengetahui apakah teks dapat terdekripsi dan terenkripsi dengan baik dan sempurna

Tabel **1.** Hasil Pengujian

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nilai pembangkit kunci | *Plaintext* | *ciphertext* | Hasil |  Keterangan |
|  Tidak |  Error |
| a = 106 | b = 1283 | m = 6075 | KAMPUS | KABJFEIG |  |  |  |
| a = 211 | b = 1663 | m = 7875 | UMRAH | DAH.OBM. |  |  |  |
| a = 421 | b = 2273 | m = 7875 | TEKNIK | TEKNIK |  |  |  |
|  |  |  |  |  |  |  |  |
| Tabel 1. Hasil Pengujian (Lanjutan) |
| a = 430 | b = 2351 | m = 11979 | INFORMATIKA | INFORMATIKA. |  |  |  |
| a = 936 | b = 1399 | m = 6655 | ELEKTRO | ELEKTRO. |  |  |  |
| a =1366 | b = 1283 | m = 6075 | DATAMINING | DAGRMIAZAX |  |  | Nilai a > b |
| a = 171 | b = 11213 | m = 53125 | DATABASE | DATABASE |  |  |  |
| a =859 | b = 2531 | m = 11979 | PEMROGRAMAN | PEMROGRAMAN. |  |  |  |
| a =419 | b = 6173 | m = 29282 | MATRIKS | MATRIKS. |  |  |  |
| a =967 | b = 3041 | m = 14406 | BERORIENTASI | BERORIENTASI |  |  |  |
| a= 141 | b = 28411 | m = 134456 | PEMROGRAMANBERORIENTASI OBJEK | PEMROGRAMANBERORIENTASI OBJEK |  |  |  |

1. **Kesimpulan**

Dari pengujian dan analisa yang telah dilakukan sebelumnya, terdapat beberapa kesimpulan, yaitu :

* 1. Dengan adanya proses pengamanan teks menggunakan teknik kombinasi, enkripsi teks menggunakan algoritma ElGamal dapat melindungi teks dan enkripsi kunci pesan menggunakan algoritma knapsack juga dapat mengamankan distribusi kunci pesan.
	2. Algoritma ElGamal dan knapsack memenuhi syarat keutuhan data. Hasil dari pengujian dekripsi pesan akan sama dengan plainteks awal (pesan semula). Dan hasil dari pengujian dekripsi ciphertext akan sama dengan kunci yang telah digunakan untuk proses enkripsi text.
	3. Terdapat Error pada nilai yang tidak sesuai dengan syarat sehingga muncul error apabila nilai a > b, nilai tidak akar primitive dan perulangan nilai kecil pada proses enkripsi.
1. **Daftar Pustaka**

Afif, Badawi. 2019. InfoTekJar : Evaluasi Pengaruh Modifikasi Three Pass Protocol Terhadap Transmisi Kunci Enkripsi. vol.1.

Aulia, R., Zakir, A., & Zulhafiz, M. 2019. InfoTekJar : Jurnal Nasional Informatika dan Teknologi Jaringan Penerapan Algoritma One Time Pad & Linear Congruential Generator Untuk Keamanan Pesan Teks. vol.1.

Bellare, M., Rogaway, P., Barroso, A., Bell, K., Bimpikis, K., Boldyreva, A., Wong, J. 2015. Introduction to Modern Cryptography. hlm.1–283.

Cobit, I. M. 2015. Jurnal Informasi vol.VII No.2 , hlm.33–47.

Hellman, A. A. M. 2017. Perancangan Aplikasi Keamanan Data Teks. vol.16, hlm.302– 305.

Hernandes, A., Hartini, H., & Sartika, D. 2019. Steganografi Citra Menggunakan Metode Least Significant Bit (LSB) dan Linear Congruential Generator (LCG). JATISI (Jurnal Teknik Informatika Dan Sistem Informasi), vol.5, no.2, hlm.137–146.

Lombu, D., Tarihoran, S. D., & Gulo, I. 2018. Kombinasi Mode Cipher Block Chaining Dengan Algoritma Triangle Chain Cipher Pada Penyandian Login Website. J-SAKTI (Jurnal Sains Komputer Dan Informatika), vol.2, no.1, hlm.1.

Munir, Rinaldi. 2004. Teori Bilangan (number teory). Penerbit Informatika : Bandung

Munir, Rinaldi. 2006. Kriptografi. Penerbit Informatika : Bandung

Nugroho, A. D., Bandung, I. T., & Bandung, J. G. 2015. Penggabungan Algoritma Kriptografi Simetris dan Kriptografi Asimetris untuk Pengamanan Pesan.

P., Mariana, W., Purba, B., 2019. Implementasi Algoritma Knapsack Dan Base64 Pada Pengamanan File Teks. , hlm.291–302.

Ping, Y., Wang, B., Tian, S., Zhou, J., & Ma, H. 2019. PKCHD: Towards a probabilistic knapsack public-key cryptosystem with high density. Information (Switzerland), vol.10, no.2, hlm.1–27.