BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan zaman membuat teknologi informasi dan komunikasi semakin maju. Proses bertukar pesan atau informasi menjadi semakin mudah dilakukan. Dalam proses bertukar pesan sangat penting menjaga keamanan pesan atau informasi agar pesan tersebut tidak dapat dimengerti oleh pihak lain maupun pihak yang tidak berwenang (Makhomah et al., 2021).

Dengan pesatnya perkembangan teknologi di dunia internet, masyarakat dapat mengumpulkan informasi, bertukar informasi dan berita dengan mudah, cepat dan bebas. Bebasnya akses informasi di internet juga dapat menimbulkan dampak negatif khususnya cybercrime seperti akses tidak sah terhadap data berita, penyalahgunaan informasi untuk keuntungan pribadi yang merugikan pengguna internet. Itulah mengapa penting untuk melindungi dan mengamankan pesan. Keberadaan pengamanan pesan bertujuan untuk melindungi pesan dari berbagai kejahatan dunia maya.

Kriptografi adalah seni atau ilmu untuk menghasilkan pesan rahasia. Pesan asli, disebut *plaintext*, disandikan menjadi pesan terenkripsi yang disebut dengan *ciphertext* melalui proses enkripsi, dan *ciphertext* diubah kembali menjadi *plaintext* melalui proses dekripsi. Kriptografi memiliki beberapa algoritma yang banyak digunakan untuk mengamankan informasi (Yusfrizal, 2019).

Salah satu algoritma kriptografi yang umum digunakan dalam keamanan adalah algoritma XOR. Algoritma XOR adalah algoritma yang sering digunakan dalam sandi yang menggunakan operasi bit demi bit dan termasuk dalam kriptografi klasik. Algoritma XOR juga merupakan algoritma sederhana yang menggunakan prinsip logika XOR. Untuk proses dimana proses enkripsi dilakukan dengan kunci XOR pada plaintext unntuk mendapatkan cipherteks. Pada proses dekripsi, ciphertext dikodekan dengan XOR dengan kunci untuk

mendapatkan teks aslinya (plaintext). Proses enkripsi dan dekripsi tidak sulit dan mudah untuk diimplementasikan (Saputro, Pujo, 2023).

Algoritma enkripsi OR atau XOR eksklusif adalah sebuah algoritma Kriptografi yang melakukan logika XOR pada setiap biner dalam teks (Sulaiman et al., 2020).

Algoritma ElGamal adalah sepasang kunci yang dihasilkan dengan memilih bilangan prima p dan dua bilangan acak g dan x, dengan syarat nilai g dan x kurang dari p, yang memenuhi persamaan (Alfiah et al., 2020)

Algoritma ElGamal ini memiliki tingkat keamanan dalam pemecahan masalah logaritma diskret pada group pergandaan bilangan prima yang besar, maka upaya untuk memecahkan pesan yang telah dienkripsi menjadi sangat sulit. Selain tingkat keamanan pada pemecahan logaritma diskret, algoritma ElGamal memiliki kelebihan dalam menghasilkan *ciphertext* (pesan yang telah tersamarkan) yang berbeda untuk *plaintext* (pesan belum disamarkan, masih dapat dibaca dengan jelas) yang sama pada proses enkripsi, tetapi ketika *ciphertext* di dekripsi akan menghasilkan *plaintext* (pesan belum disamarkan, masih dapat dibaca dengan jelas) yang sama pada proses enkripsi, tetapi ketika *ciphertext* di dekripsi akan menghasilkan *plaintext* yang sama. Proses algoritma ElGamal terdiri atas 3 proses yaitu proses pembentukan kunci, proses enkripsi dan proses dekripsi. Setiap proses dalam algoritma ini menggunakan teori bilangan terutama bilangan prima dan modulo bilangan.

Namun di sisi lain, algoritma ElGamal juga mempunyai kekurangan yaitu membutuhkan resource yang besar dan processor yang mampu melakukan perhitungan besar. Meskipun memiliki kelemahan tersebut, namun algoritma ElGamal memiliki kelebihan yang jauh lebih banyak, sehingga dalam paper ini menggunakan algoritma ElGamal dalam meningkatkan keamanan data.

Berdasarkan permasalahan di atas, penulis bermaksud untuk mengangkat judul penelitian skripsi **"Pengamanan Data Teks Menggunakan Algoritma Kriptografi ElGamal dan XOR Dari Serangan Hacker "**.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, terdapat rumusan masalah yaitu :

1. Bagaimana proses pengamanan data dengan menggunakan Algoritma ElGamal dan XOR?
2. Bagaimana proses pengenkripsian pesan teks dengan menerapkan Algoritma ElGamal dan XOR?
3. Apakah perbedaan hasil analisa dalam pengamanan file text dengan menerapkan algoritma ElGamal dan XOR?

1.3 Batasan Masalah

Adapun batasan masalah pada skripsi ini yaitu :

1. Metode yang digunakan untuk pengamanan file text yaitu dengan menggunakan metode kriptografi Algoritma Rc4 dan ElGamal.
2. *System* yang dibuat khusus untuk mengamankan pesan teks dengan menerapkan Algoritma Rc4 dan ElGamal.
3. *System* hanya melakukan proses enkripsi dan deskripsi terhadap pesan text.

1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini yaitu :

1. Dapat megetahui manfaat dan juga implementasi dari algoritma ElGamal dan XOR dalam pengamanan file text.
2. Dapat menjaga agar data berupa pesan teks terlindungi dan juga memiliki originalitas autentikasi terhindar dari orang yang tidak bertanggung jawab.

1.5 Manfaat Penelitian

Adapun manfaat penelitian yang penulis buat yaitu :

1. Dapat mengamankan pesan teks pada gambar dengan aman.
2. Mengenalkan system yang bermanfaat bagi masyarakat.
3. Menjadi referensi bagi peneliti lain yang akan mendekati topik yang sama tetapi dengan perspektif yang berbeda.

1.6 Sistematika Penulisan

Sistematika penulisan yang dilakukan pada penelitian ini yaitu :

BAB I PENDAHULUAN

Bab ini mengulas latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, dan manfaat penelitian.

BAB II LANDASAN TEORI

Bab ini membahas teori-teori yang berhubungan dengan Algoritma ElGamal dan XOR.

BAB III METODE PENELITIAN

Bab ini membahas metodologi penelitian tentang rancangan penelitian berdasarkan analisis masalah yang digambarkan dalam bentuk diagram umum dan arsitektur umum dalam bentuk *flowchart* pada perancangan sistem berdasarkan skema Algoritma ElGamal dan XOR.

BAB IV HASIL DAN PEMBAHASAN

Bab ini membahas hasil dari penelitian berdasarkan metodologi penelitian yang telah dilakukan. Kemudian dilanjutkan dengan pembahasan dan pengujian untuk melihat apakah sistem sudah berjalan sesuai dengan perancangan atau tidak, serta menemukan kesalahan atau kekurangan pada sistem.

BAB V KESIMPULAN DAN SARAN

Bab ini membahas uraian kesimpulan dari pembahasan sampai hasil penelitian selesai dilaksanakan dan juga saran sebagai masukan untuk penelitian terkait berikutnya yang diharapkan bermanfaat sebagai pemecahan masalah tersebut.

BAB II

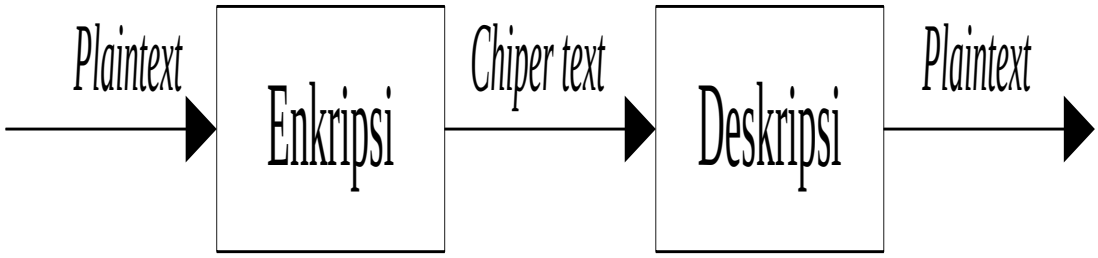
LANDASAN TEORI

2.1 Kriptografi

Kriptografi adalah studi tentang metode komunikasi yang aman antara dua belah pihak. Biasanya ada dua pihak yang saling mengirim pesan, tetapi mereka ingin menghindari kemungkinan pihak ketiga memahami isi dari pesan mereka jatuh kepada pihak yang salah. (Rubinstein_Salzedo, 2018).

Menurut Iqbal & Krawec, (2020), Kriptografi adalah sebuah seni yang berfokus pada menyembunyikan dan mengirimkan pesan secara diam-diam. Banyak sandi yang digunakan sepanjang sejarah, banyak diantaranya sekarang dianggap tidak aman menurut standar modern. Nyatanya, baru pada pertengahan abad ke-20 kriptografi berubah dari seni menjadi sebuah sains.

Kriptografi telah digunakan selama ribuan tahun untuk membantu menyediakan rahasia komunikasi antara pihak-pihak yang saling percaya. Dalam bentuknya yang paling dasar, dua orang, sering dilambangkan sebagai Alice dan Bob, telah menyepakati kunci rahasia tertentu. Di lain waktu, Alice mungkin ingin mengirim pesan rahasia ke Bob (atau Bob mungkin ingin mengirim pesan ke Alice). Kunci digunakan untuk mengubah pesan asli (yang biasanya kita sebut dengan plaintext) menjadi bentuk acak yang tidak dapat dipahami kepada siapa saja yang tidak memiliki kunci. Proses ini disebut enkripsi, dan pesan yang diacak disebut ciphertext. Ketika Bob menerima ciphertext, dia dapat menggunakan kunci untuk mengubah ciphertext kembali menjadi plaintext atau teks asli, ini disebut dengan proses dekripsi (Stinson & Paterson, 2019).



Gambar 2.1 Skema Enkripsi dan Dekripsi

Komponen algoritma kriptografi :

1. *Input* : *plaintext*, pesan (data) yang akan dikirim (berisi data/informasi dalam Bahasa aslinya). *Plaintext* digunakan selama proses enkripsi biasanya dalam bentuk *text.*
2. *Output* : *ciphertext* adalah hasil enkripsi dari *plaintext* dalam bentuk yang tidak dapat dimengerti dan tidak dapat dikenali sebagai pesan data atau informasi.
3. Encryption : proses untuk mengubah plainteks menjadi *ciphertext*
4. *Decryption* : proses mengembalikan ciphertext kedalam *ciphertext.*
5. *Key* : proses enkripsi dan dekripsi membutuhkan kunci keduanya bisa *public key*

2.1.1 Tujuan Kriptografi

Adapun tujuan kriptografi adalah sebagai berikut :

1. *Confidentiality* (Kerahasiaan) Adalah layanan yang ditujukan untuk menjaga agar pesan tidak dapat dibaca.
2. *Authentiation* (Otentikasi) Adalah identifikasi, baik mengidentifikasi kebenaran pihak-pihak yang berkomunikasi (user authentication atau entity authentication). Maupun mengidentifikasi kebenaran sumber pesan.
3. *Data Integrity* (Integritas Data) Adalah layanan yang menjamin bahwa pesan masih asli atau belum pernah dimanipulasi selama pengiriman.
4. *Non - repudiation* (Tanpa Penyangkalan) Adalah layanan untuk mencegah entitas yang berkomunikasi melakukan penyangkalan yaitu pengiriman atau penerima pesan menyangkal telah menerima pesan.

2.2 ElGamal

El-Gamal adalah sistem enkripsi kunci asimetris yang ditemukan Taher El-Gamal pada tahun 1985. Algoritma ini merepresentasikan metode alternatif untuk cipher kunci publik RSA. Perbedaan utama antara algoritma El Gamal dan RSA adalah bahwa keamanan RSA bergantung pada kesulitan faktorisasi bilangan prima besar, sementara El-Gamal bergantung pada kesulitan dalam menghitung modulus logaritmik diskrit dari bilangan prima besar. Masalah logaritma diskrit adalah masalah sulit dalam matematika karena itu penting terutama pada konjungtur untuk mendapatkan semua solusi yang mungkin. Jadi sistem crypto ini hampir rusak tidak tersedia atau membutuhkan waktu lama. Terutama Keunggulan teknologi El Gamal adalah pesan teks yang sama menghasilkan pesan teks rahasia yang berbeda setiap saat jika dienkripsi.(Yousif et al., 2020).

Algoritma ElGamal adalah sepasang kunci yang dihasilkan dengan memilih bilangan prima p dan dua bilangan acak g dan x, dengan syarat nilai g dan x kurang dari p, yang memenuhi persamaan (Alfiah et al., 2020).

ElGamal dapat digunakan untuk tanda tangan digital dan enkripsi, keamanannya bergantung pada kesulitan menghitung logaritma diskrit dalam bidang yang terbatas.

Untuk menghasilkan pasangan kunci, pertama pilih bilangan prima, p, dan dua bilangan acak, g dan x, sehingga g dan x keduanya lebih kecil dari p, lalu hitung $= g^X \, mod \, p$.

Kunci publiknya adalah y, g dan p. Baik g dan p dapat dibagikan oleh sekelompok pengguna. Kunci pribadinya adalah X (Jorgensen, 2003).

2.2.1 Proses Pembentukan Kunci 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 $1 = x = p - 2$.
4. Hitung $y = g^p mod\ p$.

Kunci *public* adalah y, g, p sedangkan kunci *private* adalah x. Nilai $y, g,$ dan p tidak dirahasiakan sedangkan nilai x harus dirahasiakan karena merupakan kunci *private* untuk mendekripsi *plaintext.*

2.2.2 Contoh Perhitungan ElGamal

Berikut contoh perhitungan manual proses pembentukan kunci algoritma ElGamal, proses enkripsi, dan dekripsi

A. Pemilihan Parameter :

Pembentukan kunci *public* dan kunci rahasia dilakukan oleh penerima pesan yaitu Alice. Adapun Langkah-langkah yang dilakukan Alice dalam pembentukan kunci yaitu :

- Menentukan bilangan prima p. Disarankan ambil nilai p yang besar. Maka diambil nilai $p = 107$.
- Kemudian Alice memilih angka 53 untuk nilai q. Maka $q = 53$.
- Selanjutnya Alice mencari elemen primitive a dari $Z_p *$.
- Alice membuat table perhitungan untuk beberapa nilai a untuk mengecek apakah nilai tersebut termasuk elemen primitive atau tidak.

Tabel 2.1 Perhitungan $a^2\ mod\ p$ dan $a^a\ mod\ p$

a	2	3	4	5	6
$a^2\ mod\ p$	4	9	16	25	36
$a^q\ mod\ p$	106	1	1	106	106

- Dari bebrapa nilai tersebut Alice mendapatkan beberapa nilai a dari Z_p* yaitu 2, 5, dan 6. Alice memilih nilai a yang dipakai sebagi elemen primitif adalah $a = 2$.
- Kemudian Alice menentukan kunci rahasia a (dimana $a \in \{0, 1, \dots, \dots, p-2\}$) dengan nilai 63 sehingga sekarang Alice mempunyai nilai (p, a, a) = (107, 2, 63).
- Dengan nilai a yang sudah diketahui, Alice mencari nilai β.

  $\beta = a^a \ mod \ p$

  $\beta = 2^{63} \ mod \ 107$

  $\beta = 46.$
- Alice mendapatkan kunci *public* (p, a, β) = (107, 2, 46) dan kunci rahasia $a = 63$. Kunci *public* tersebut diberitahukan ke Bob untuk mengenkripsi pesan yang akan dikirim Bob ke Alice dan kunci rahasia dijaga keamanannya oleh Alice.

B. Enkripsi Pesan Oleh Pengirim

Bob menerima kunci publik dari Alice (p, α, β) = (107, 2 , 46). Dengan kunci publik tersebu, Bob mengenkripsi pesan "**SELAMAT PAGI**" untuk dikirimkan ke Alice. Langkah-langkah yang dilakukan Bob dalam mengenkripsi pesan tersebut :

a. Pertama Bob mengonversi pesan tersebut dalam kode ASCII.

Tabel 2.2 Konversi karakter ke kode ASCII

i	**Karakter**	***Plaintext*** M_i	**ASCII**
1	**S**	M_1	83
2	**E**	M_2	69
3	**L**	M_3	76
4	**A**	M_4	65
5	**M**	M_5	77
6	**A**	M_6	65

7	**T**	M_7	84
8	**<spasi>**	M_8	32
9	**P**	M_9	80
10	**A**	M_{10}	65
11	**G**	M_{11}	71
12	**I**	M_{12}	73

b. Kemudian Bob menentukan bilangan acak k (k $\in\{0,1,...,106\}$) yang dijaga kerahasiaannya untuk setiap plainteks M dan mengenkripsi plainteks tersebut dengan menghitung nilai r dan t.

Tabel 2.3 Enkripsi *Plaintext* ke *Ciphertext*

$\boldsymbol{i}$	$\boldsymbol{M_i}$	$\boldsymbol{k_i}$	$r = 2^k \ (mod\ 107)$	$t = 46^k\ M_i \ (mod\ 107)$
1	83	57	91	21
2	69	43	7	78
3	76	65	77	82
4	65	88	89	66
5	77	34	9	98
6	65	46	56	93
7	84	47	5	4
8	32	76	85	22

9	80	87	98	83
10	65	69	55	23
11	71	41	82	11
12	73	35	18	23

c. Berdasarkan tabel tersebut, Bob memperoleh cipherteks (ri, ti), i = 1, 2, …, 12 sebagai berikut: (91, 21) (7, 78) (77, 82) (89, 66) (9, 98) (56, 93) (5, 4) (85, 22) (98, 83) (55, 23) (82, 11) (18, 23).

d. Kemudian *ciphertext* tersebut akan dikirimkan ke Alice.

C. Dekripsi Pesan Oleh Penerima

Alice memperoleh pesan yang telah disamarkan dari Bob. Karena Alice yang memegang kunci rahasia $(a = 63)$ dari enkripsi tersebut, Alice dapat mendekripsi pesan tersebut agar dapat dibaca. Alice melakukan perhitungan untuk tiap blok *ciphetext.*

Tabel 2.4 Dekripsi *Ciphertext* ke *Plaintext*

$\boldsymbol{i}$	$\boldsymbol{r}$	$\boldsymbol{t}$	$\boldsymbol{r^{43}\ mod\ 107}$	$\boldsymbol{M_i = t\ r^{43}\ mod\ 107}$	**Karakter** $\boldsymbol{M_i}$
1	91	21	60	83	S
2	7	78	5	69	E
3	77	82	74	76	L
4	89	66	48	65	A
5	9	98	39	77	M

6	56	93	3	65	A
7	5	4	21	84	T
8	85	22	89	32	
9	98	83	68	80	P
10	55	23	54	65	A
11	82	11	94	71	G
12	18	23	59	73	I

Berdasarkan tabel 2.4, Alice memperoleh *plaintext* dari mendekripsi *ciphertext* yang diberikan Bob.

2.3. Xor

Teknik XOR melakukan enkripsi dan dekripsi terhadap sebuah informasi dengan menggunakan kunci tunggal dan operasi bit XOR (Sidik et al., 2019). Tabel logika dari operasi XOR adalah sebagai berikut :

Tabel 2.5 Tabel Logika Operasi XOR

A	B	A ⊕B
0	0	0
0	1	1
1	0	1
1	1	0

2.3.1 Proses Enkripsi XOR

Proses enkripsi atau dekripsi di awali dengan merubah setiap nilai plaintext ke biner Formula untuk melakukan proses enkripsi dan dekripsi adalah :Enkripsi : $C_i = P_i \, XOR \, K_i$Deskripsi
: $P_i = C_i \, XOR \, k_o$

2.3.2 Contoh Perhitungan Enkripsi XOR

Untuk melakukan enkripsi XOR dapat kita lakukan sebagai berikut :

Diketahui *plaintext* yang akan dienkripsi adalah “SELAMAT PAGI”, dengan *key* : “HALLO”. Adapun Langkah yang akan dilakukan untuk mengenkripsi kalimat “SELAMAT PAGI” adalah sebagai berikut :

- Ubah *plaintext* dan *key* menjadi biner.
- *Plaintext* “SELAMAT PAGI” jika dikonversi ke biner maka akan menjadi :

S = 83 : 01010011
E = 69 : 01000101
L = 76 : 01001100
A = 65 : 01000001
M = 77 : 01001101
A = 65 : 01000001
T = 84 : 01010100
<spasi> = 32 : 00100000
P = 80: 01010000
A : 65 : 01000001
G : 71 : 01000111
I : 73 : 01001001

- *Key* “HALLO” jika dikonversi ke biner maka akan menjadi :

H = 72 : 01001000
A = 65 : 01000001
L = 76 : 01001100
L = 76 : 01001100
O = 79 :0100111

- Kemudian ulangi kunci “HALLO” hingga mencapai Panjang *plaintext* “SELAMAT PAGI. Maka kunci yang diulang yaitu : “HALLOHALLOHALLOHA”

- Sekarang lakukan operasi XOR antara setiap bit dalam teks dan kunci yang sesuai :

S ⊕ H = 01010011 ⊕ 01001000 = 00011011

E ⊕ A = 01000101 ⊕ 01000001 = 00000100

L ⊕ L = 01001100 ⊕ 01001100 = 00000000

A ⊕ L = 01000001 ⊕ 01001100 = 00001101

M ⊕ O = 01001101 ⊕ 01001111 = 00000010

A ⊕ H = 01000001 ⊕ 01001000 = 00001001

T ⊕ A = 01010100 ⊕ 01000001 = 00010101

<spasi> ⊕ L = 00100000 ⊕ 01001100 = 01101100

P ⊕ L = 01010000 ⊕ 01001100 = 00011100

A ⊕ O = 01000001 ⊕ 01001111 = 00001110

G ⊕ H = 01000111 ⊕ 01001000 = 00001111

I ⊕ A = 01001001 ⊕ 01000001 = 00001000

- Maka *ciphertext* dari Kata "SELAMAT PAGI" dengan *key* "HALLO" adalah : 00011011, 00000100, 00000000, 00001101, 00000010, 00001001, 00010101, 01101100, 00011100, 00001110,0001111, 00001000.

2.3.3 Contoh Perhitungan Proses Dekripsi XOR

Untuk melakukan dekripsi XOR dapat kita lakukan sebagai berikut :

Diketahui *ciphertext* yang akan didekripsi adalah "00011011, 00000100, 00000000, 00001101, 00000010, 00001001, 00010101, 01101100, 00011100, 00001110,0001111, 00001000", dengan key : "HALLO". Adapun Langkah yang akan dilakukan untuk mendekripsi kalimat "00011011, 00000100, 00000000, 00001101, 00000010, 00001001, 00010101, 01101100, 00011100, 00001110,0001111, 00001000" adalah sebagai berikut:

00011011 ⊕ 01001000 = 01010011 = 83 = **S**

00000100 ⊕ 01000001 = 01000101 = 69 = **E**

00000000 ⊕ 01001100 = 01001100 = 76 = **L**

00001101 ⊕ 01001100 = 01000001 = 65 = **A**

00000010 ⊕ 01001111 = 01001101 = 77 = **M**

00000010 ⊕ 01001000 = 01000001 = 65 = **A**

00010101 ⊕ 01000001 = 01010100 = 84 = **T**

01101100 ⊕ 01001100 = 00100000 = 32 = **<spasi>**

00011100 ⊕ 01001100 = 01010000 = 80 = **P**

00001110 ⊕ 01001111 = 01000001 = 65 = **A**

00001111 ⊕ 01001000 = 01000111 = 71 = **G**

00001000 ⊕ 01000001 = 01001001 = 73 = **I**

- Maka *ciphertext* dari : 00011011, 00000100, 00000000, 00001101, 00000010, 00001001, 00010101, 01101100, 00011100, 00001110,0001111, 00001000 adalah : **SELAMAT PAGI.**