JIKTEKS : Jurnal Ilmu Komputer dan Teknologi Informasi
Vol. 03, No. 02, April 2025, hlm. 54 - 63
ISSN : 2986-5417
DOI : doi.org/10.70404/jikteks.v3i02.176
Laman Jurnal: jurnal.faatuatua.com/index.php/JIKTEKS

Analisa Perbandingan Algoritma Kriptografi RC4 Dan Elgamal Dalam Pengamanan Data Teks

Martina Zendrato
Teknik Informatika, Universitas Budi Darma, Indonesia

Keywords:

Comparative Analysis,
Cryptography,
RC4,
El-Gammal,
Data Security.

ABSTRACT

Text data is a collection or text stored in digital format. This data can be text documents, blogs, text messages or other text entities. Text data usually consists of sequences of characters, words, sentences and paragraphs. Text data can be processed and analyzed using various methods and algorithms, including statistical processing, machine learning and artificial intelligence. The problem that occurs in the process of exchanging or obtaining information (text data) is if the information is spread widely due to wiretapping, theft and falsification of information so that it will cause harm to the owner of the information. To secure text data, a strong security system is needed. The text data must be processed and converted into code before being sent. One way to secure text data from crime is by using the concept of cryptography. To overcome the problem of choosing the right algorithm for securing text data, it is necessary to compare the RC4 algorithm and the Elgamal algorithm. By making a comparison, you can choose the algorithm that is most suitable for your needs in processing text data.

Kata Kunci:

Analisis Perbandingan,
Kriptografi,
RC4,
El-Gammal,
Keamanan Data.

ABSTRAK

Data teks merupakan suatu kumpulan atau teks yang disimpan dalam format digital. Data ini dapat berupa dokumen teks, blog, pesan teks atau entitas teks lainnya. Data teks biasanya terdiri dari urutan karakter, kata, kalimat dan paragraf. Data teks dapat diolah dan dianalisis menggunakan berbagai metode dan algoritma, termasuk pemrosesan statistik, pembelajaran mesin dan kecerdasan buatan Permasalahan yang terjadi pada proses pertukaran atau mendapatkan informasi (data teks) adalah apabila informasi tersebut tersebar luas karena adanya penyadapan, pencurian dan pemalsuan informasi sehingga akan menyebabkan kerugian bagi pemilik informasi. Untuk mengamankan data teks tersebut diperlukan suatu sistem keamanan yang kuat. Data teks tersebut harus diproses dan diubah dalam bentuk kode sebelum dikirimkan. Salah satu cara untuk mengamankan data teks dari tindak kejahatan tersebut dengan menggunakan konsep kriptografi. Untuk mengatasi permasalahan dalam memilih algoritma yang tepat dalam melakukan pengamanan data teks maka perlu dilakukan perbandingan antara algoritma RC4 dan algoritma Elgamal. Dengan dilakukan perbandingan dapat memilih algoritma yang paling sesuai untuk kebutuhan dalam mengamakankan data teks.

Korespondensi Penulis *):

Martin Zendrato
Universitas Budi Darma
Jl. Sisingamangaraja No. 338, Kota Medan, Indonesia

Diajukan: 03-04-2025 | Diterima: 18-04-2025 | Diterbitkan: 30-04-2025

1. PENDAHULUAN

Berkembangnya teknologi maka dibutuhkan teknik keamanan terhadap kerahasiaan data dan informasi yang dipertukarkan semakin meningkat. Kriptografi merupakan displin ilmu atau teknik untuk mengamankan sebuah komunikasi dan data dengan menggunakan teknik matematika. Kriptografi melibatkan penggunaan algoritma kriptografi untuk melakukan enkripsi dan dekripsi. Untuk mengamakan suatu pesan maka dibutuhkan suatu algoritma yang tepat agar pesan tetap terjaga keamanan serta kerahasiaannya. Untuk mendapatkan algoritma yang tepat dalam mengamankan data informasi maka dibutuhkan suatu perbandingan dari algoritma kriptografi.

Data teks merupakan suatu kumpulan atau teks yang disimpan dalam format digital. Data ini dapat berupa dokumen teks, blog, pesan teks atau entitas teks lainnya. Data teks biasanya terdiri dari urutan karakter, kata, kalimat

dan paragraf. Data teks dapat diolah dan dianalisis menggunakan berbagai metode dan algoritma, termasuk pemrosesan statistik, pembelajaran mesin dan kecerdasan buatan[1].

Permasalahan yang terjadi pada proses pertukaran atau mendapatkan informasi (data teks) adalah apabila informasi tersebut tersebar luas karena adanya penyadapan, pencurian dan pemalsuan informasi sehingga akan menyebabkan kerugian bagi pemilik informasi. Untuk mengamankan data teks tersebut diperlukan suatu sistem keamanan yang kuat. Data teks tersebut harus diproses dan diubah dalam bentuk kode sebelum dikirimkan. Salah satu

Algoritma kriptografi merupakan seni dan ilmu untuk menjaga kerahasiaan pesan dengan cara menyamarkan data atau informasi menjadi bentuk sandi yang tidak memiliki makna. Kriptografi merupakan suatu ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi, integritas data serta autentifikasi data. Semua tujuan tersebut untuk menciptakan rasa aman dan nyaman setelah memberikan data yang bersifat rahasia. Kriptografi memiliki tiga jenis algoritma yang berbeda berdasarkan kunci, yaitu algoritma kunci simetris, kunci asimetris dan fungsi hash[2]. Algoritma kriptografi untuk mengamankan data teks pada penelitian ini dengan menggunakan algoritma RC4 dan Elgamal. Algoritma RC4 adalah algoritma simetris yang berarti kunci yang sama digunakan untuk enkripsi dan dekripsi. Algoritma Elgamal adalah algoritma asimetris yang menggunakan sepasang kunci yaitu kunci publik dan kunci pribadi untuk enkripsi dan dekripsi.

Untuk mengatasi permasalahan dalam memilih algoritma yang tepat dalam melakukan pengamanan data teks maka perlu dilakukan perbandingan antara algoritma RC4 dan algoritma Elgamal. Dengan dilakukan perbandingan dapat memilih algoritma yang paling sesuai untuk kebutuhan dalam mengamakankan data teks.

Berdasarkan penelitian terdahulu yang dilakukan oleh Yuri Ariyanto, dengan judul penelitian "Algoritma RC4 dalam proteksi transmisi dan hasil query untuk ORDBMS POSTGRESOL", mengemukakan bahwa algoritma RC4 merupakan salah satu jenis stream chipper sehingga dapat memproses unit atau input data, pesan atau informasi pada satu unit. Algoritma ini tidak harus menunggu sejumlah input data, pesan atau informasi tertentu sebelum diproses atau menambahkan byte untuk mengenkripsi[3].

Penelitian yang telah dilakukan oleh Rahmad Awaludin mengemukakan bahwa terdapat jenis aliran kode pada RC4 dilakukan per karakter 1 byte untuk satu operasi. Algoritma RC4 merupakan salah satu algoritma kunci simetris yang dirancang oleh RSA Data Security Inc, dalam bentuk stream cipher[4].

Penelitian yang telah dilakukan oleh Ahir Yugo Nugroho, et.al, 2022, dengan judul penelitian "Penerapan Elgamal Guna Meningkatkan Keamanan Data Text dan Docx", mengemukakan bahwa algoritma Elgamal merupakan bagian dari kriptografi asimetris yang membentuk salah satu kunci menggunakan bilangan prima dan menitikberatkan kekuatan kuncinya pada pemecahan masalah logaritma diskrit[5].

Algoritma Elgamal terbentuk melalui tiga proses yaitu proses pembentukan kunci, proses enkripsi data dan proses dekripsi data dan hasilnya digabungkan agar menjadi suatu pesan yang utuh dan mudah dimengerti. Dalam algoritma Elgamal diperlukan bilangan prima p dan elemen primitif[6]

2. METODE PENELITIAN

2.1 Kerangka Kerja Penelitian

Kerangka kerja penelitian menunjukan tentang tahapan kegiatan yang dilakukan pada penelitian ini, yang penulis gambarkan dalam bentuk diagram.

Gambar 1. Kerangka Kerja Penelitian

Deskripsi Tahapan Kerangka Kerja Penelitian

1. Identifikasi Masalah
   Tahapan ini bertujuan untuk memahami latar belakang kebutuhan akan sistem keamanan data, terutama dalam bentuk teks. Penelitian dimulai dengan mengidentifikasi permasalahan yang muncul dalam pengamanan data digital dan pentingnya memilih algoritma kriptografi yang sesuai. Fokus utamanya adalah perlindungan data teks terhadap penyadapan atau modifikasi oleh pihak yang tidak berwenang.
2. Analisa Metode RC4 dan Elgamal
   Pada tahap ini dilakukan kajian mendalam terhadap dua algoritma kriptografi: RC4 yang merupakan algoritma stream cipher simetris, dan Elgamal yang merupakan algoritma kriptografi asimetris. Analisis mencakup aspek teknis seperti struktur algoritma, kompleksitas enkripsi dan dekripsi, efisiensi performa, serta tingkat keamanan masing-masing metode.
3. Perbandingan Metode RC4 dan Elgamal
   Setelah analisa dilakukan, tahap selanjutnya adalah membandingkan kedua algoritma berdasarkan parameter-parameter tertentu seperti kecepatan proses, konsumsi sumber daya, panjang kunci, serta kerentanan terhadap serangan. Perbandingan ini bertujuan untuk mengetahui kelebihan dan kekurangan masing-masing algoritma dalam konteks pengamanan data teks.
4. Perancangan Aplikasi

Berdasarkan hasil analisis dan perbandingan, dilakukan perancangan aplikasi yang mengimplementasikan kedua algoritma tersebut. Aplikasi ini akan digunakan sebagai media uji coba dalam melakukan enkripsi dan dekripsi data teks menggunakan RC4 dan Elgamal, sehingga memungkinkan pengamatan performa secara langsung.

5. Pengujian
   Pada tahap ini, aplikasi yang telah dirancang akan diuji dengan menggunakan beberapa skenario dan dataset teks. Tujuan pengujian adalah untuk mengukur kecepatan, keakuratan, dan efisiensi dari proses enkripsi serta dekripsi. Selain itu, dilakukan juga pengujian terhadap kemungkinan kerentanan keamanan atau error selama proses.
6. Dokumentasi
   Tahapan akhir adalah dokumentasi seluruh proses penelitian, mulai dari identifikasi masalah, analisis algoritma, implementasi aplikasi, hingga hasil pengujian. Dokumentasi ini akan menjadi bukti ilmiah serta referensi bagi penelitian lanjutan, sekaligus memberikan kesimpulan terhadap efektivitas RC4 dan Elgamal dalam pengamanan data teks.

2.2 Algoritma RC4

Konsep RC4 adalah jenis aliran kode yang berarti operasi enkripsi dilakukan per karakter 1 *byte* untuk satu operasi. Algoritma kriptografi Rivest Code 4 (RC4) merupakan salah satu algoritma kunci simetris yang dirancang oleh RSADSI (RSA *Data Security Inc*) dalam bentuk stream cipher. RC4 menggunakan panjang kunci dari 1 hingga 256 *byte* yang digunakan untuk menginisialisasi tabel 256 *byte*. Algoritma RC4 menggunakan dua *Substitusi Box* (S-Box), yaitu array dengan panjang 256 yang berisi dari 0 sampai 255, dan S-Box kedua, berisi permutasi yang dimana memiliki fungsi dari kunci dengan panjang variabel. Cara kerja algoritma RC4 adalah dengan menginisialisasi S-Box pertama, S[0], S[1], … , S[255], dengan angka 0 sampai 255. Langkah-langkah penyelesaian algoritma RC4 antara lain [11] :

a. Proses inisialisasi S-Box (*Array* S)
   For i = 0 ; i < 256 ; i++; ..... (1)
   S[i] = i
b. Proses inisialisasi S-Box (*Array* K)
   j = 0 ..... (2)
   for i = 0 ; I < 256 ; i++;
   j = (j + S[i] + (this -> key[i % strlen (this ->1key)])) mod1256
   x = S[i]
   S[i] = S[j]
   S[j] = x
   Kemudian pada lakukan pengacakan S-Box
c. Proses pengacakan S-Box
   i = 0; j = 0; res = 0; ..... (3)
   for y = 0;y < strlen (this -> str) ; y++;
   Setelah itu, buat pseudo *random byte* pada tabel 4 dengan langkah sebagai berikut:
   Pseudo1Random1Byte
   1= (i + 1) mod 256;
   J = (j + S[i]) mod 256;
   x = S[i];
   S[i] = S[j];
   S[j] = x;
   res = this ->str[y] ^ chr(S[(S[i] + S[j]) mod 256]);
d. Proses permutasi elemen S [0, 1,……, 255], supaya elemen dalam S menjadi acak
   j ← 0 ..... (4)
   for i ← 0 *to* 255 *do*
   j ← (*j* + S[*i*] + K[*i mod Lenght* (K)] *mod* 256
   swap (S[*i*], S[*j*])
   end.

   Setelah sub-proses *KSA* selesai maka, sub-proses kedua adalah *PRGA.* Sub- proses *PRGA* membangkitkan kunci-alir dengan cara mengambil nilai S[*i*] dan S[*j*], kemudian mempertukarkannya lalu menjumlahkannya dalam modulus 256. Kunci-alir tersebut kemudian di-XOR-kan dengan sebuah karakter *plainteks*, adapaun algortimanya sebagai berikut :
   i ← *0, j* ← *0* ..... (5)
   for i ← 0 *to Lenght*(P)-1 *do //* P adalah *plaintext*
   i ← *i* + 1
   j ← (*j* + S[*i*]) *mod* 256
   swap (S[*i*], S[*j*])
   k= S [S[*i*] + S[*j*]] mod 256
   C = P xor *k*
   end.

2.3 Algoritma 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 *chipper text*, yang nantinya blok-blok *chipper text* tersebut akan didekripsi kembali dan hasilnya kemudian digabungkan menjadi plainteks semula [5].

Algoritma Elgamal memiliki beberapa tahapan dalam melakukan pengamanan data yang bersifat rahasia antara lain [12] :

1. *Generate key*
   Algoritma ElGamal memerlukan sepasang kunci yang dibangkitkan dengan memilih bilangan prima p dan dua buah bilangan acak (random) g dan x, dengan syarat bahwa nilai g dan x lebih kecil dari p yang memenuhi persamaan.
   $Y = g^{\wedge} x \bmod p$ ..... (6)
   Dari persamaan tersebut nilai y, g, dan p merupakan pasangan kunci publik, sedangkan x, p merupakan pasangan kunci pribadi. Besaran-besaran digunakan dalam algoritma kriptografi Elgamal adalah :
   a. P adalah bilangan prima dengan syarat P > 255
   b. G adalah bilangan acak, dengan syarat G < P
   c. X adalah bilangan acak, dengan syarat X < P
   d. Y adalah hasil perhitungan dari Y = G ^ x mod P
   e. Lalu dihasilkan *public key* = y g p, dan *private key* = p x
2. Proses Enkripsi
   Proses enkripsi merupakan proses mengubah pesan asli (plaintext) menjadi pesan rahasia (chipertext). Pada proses ini digunakan *public key* (p,g,y). Untuk proses enkripsi maka digunakan persamaan sebagai berikut.
   $a = g^{\wedge} k \bmod p$ ..... (7)
   a. Potong plaintext menjadi blok-blok m1, m2. . . .
   b. Ubah nilai blok pesan ke dalam nilai ASCII
   c. Pilih bilangan acak k, dengan syarat $1 \leq k \leq p\text{-}2$. Nilai k digunakan untuk menghitung nilai a dan b.
   d. Setiap blok m dienkripsi dengan rumus sebagai berikut:
      Nilai a (Gamma) = g ^ k mod p
      Nilai b (Delta) = y ^ k.m mod p
   e. Susun ciphertext dengan urutan
      a1,b1,a2,b2,……,aN,bN
3. Proses Dekripsi
   Proses Dekripsi Merupakan proses mengubah pesan rahasia (ciphertext) menjadi pesan asli (*plaintext*). Proses dekripsi menggunakan kunci pribadi x dan p untuk mendekripsi a dan b menjadi *plaintext* (m) dengan persamaan:
   $m = b*a^{\wedge} (p\text{-}1\text{-}x) \bmod p$ ..... (8)

3. HASIL DAN ANALISIS

3.1 Analisa Masalah

Pengamanan data teks dengan menggunakan kriptografi memerlukansebuah kunci. Teks yang akan dikirim terlebih dahulu disandikan dengan kunci yang tidak dapat diketahui orang lain dan hanya yang bersangkutan dapat mengetahui kunci pesan teks tersebut. Dalam penelitian ini, akan membandingkan dua algoritma kriptografi yaitu dengan menggunaan algoritma RC4 dan algoritma Elgamal. Pada pengimplementasiannya, sistem yang dibangun pada penelitian ini bertujuan untuk mengamankan data teks. Cara kerja sistem bermula dari *plaintext* akan dienkripsi menggunakan algoritma RC4 dan Elgamal sehingga menghasilkan *ciphertext* atau pesan yang sudah diamankan sehingga tidak dapat dibaca lagi. Sedangkan untuk mengembalikan pesan *text* yang telah di enkripsi, maka tahapan yang harus dilakukan dimulai dengan melakukan dekripsi terlebih dahulu dengan *plaintext* yang dihasilkan supaya nantinya akan menghasilkan sebuah *plaintext* atau pesan yang dapat dibaca dan dipahami maknanya.

Perbandingan antara algoritma RC4 dan algoritma Elgamal dilakukan untuk mengetahui algoritma yang mana cocok (tepat) dalam hal melakukan pengamanan data teks Sampel merupakan sebagian dari populasi yang karakteristiknya hendak diteliti. Untuk sampel data yang digunakan sebagai objek penelitian ini adalah sandi akun gmail penulis sendiri.

Tabel 1. Sampel Data

Plainteks	**S**	**e**	**l**	**a**	**m**	**a**	**T**	**spasi**	**P**	**A**	**G**	**I**
Binary	83	101	108	97	109	97	116	32	80	65	71	73

3.2 Penerapan Algoritma RC4

Setelah melakukan proses pengurutan bilangan heksadesimal berdasarkan frekuensi kemunculannya dan mendapatkan nilai biner yang sesuai, langkah berikutnya melibatkan penerapan proses kompresi melalui algoritma Lempel Ziv Welch. Detail mengenai langkah-langkah dalam kompresi berkas video dapatdilihat pada table 4. dibawah ini Untuk proses melakukan pengamanan data teks menggunakan algoritma RC4,

Langkah pertama yang harus dilakukan adalah menentukan ekspansi kunci dengan mode 12 *byte*. S Box dengan panjang 12 *byte,* dengan S[0] = 0, S[1] = 1, S[2] = 2, S[3] = 3, S[4] = 4, …… S[11] = 11, Sehingga array S menjadi = 0 1 2 3 4 5 6 7 8 9 10 11. Inisialisasi 12 *byte* kunci array, sehingga array K berisi 0 3 6 7 9 2 8 1 5 10 11 13 dan dilakukan proses enkripsi dengan *plaintext* : Selamat PAGI.

Inisialisasi i dan j dengan 0 kemudian dilakukan KSA agar tercipta *state-array* yang acak. Penjelasan iterasi lebih lanjut adalah sebagai berikut.

Iterasi 1
i = 0
j = (0+S[0]+K[0 mod 12]) mod 12
= (0+0+2) mod 12 = 2
Swap = (S[0], S[2])
Iterasi 2
i = 1
j = (2+S[1]+K[1 mod 12]) mod 12
= (2+1+3) mod 12 = 6
Swap = (S[1], S[6])
Iterasi 3
i = 2
j = (6+S[2]+K[2 mod 12]) mod 12
= (6+6+6) mod 12 = 6
Swap = (S[2], S[6])
Iterasi 4
i = 3
j = (6+S[3]+K[3 mod 12]) mod 12
= (6+3+7) mod 12 = 4
Swap = (S[3], S[4])
Iterasi 5
i = 4
j = (4+S[4]+K[4 mod 12]) mod 12
= (4+4+9) mod 12 = 5
Swap = (S[4], S[5])
Iterasi 6
i = 5
j = (5+S[5]+K[5 mod 12]) mod 12
= (5+5+2) mod 12 = 0
Swap = (S[5], S[0])
Iterasi 7
i = 6
j = (0+S[6]+K[2 mod 12]) mod 12
= (0+6+8) mod 12 = 2
Swap = (S[6], S[2])
Iterasi 8
i = 7
j = (2+S[7]+K[7 mod 12]) mod 12
= (2+7+1) mod 12 =8
Swap = (S[7], S[8])
Iterasi 9
i = 8
j = (10+S[8]+K[8 mod 12]) mod 12
= (10+8+5) mod 12 = 9
Swap = (S[8], S[11])
Iterasi 10
i = 9
j = (11+S[9]+K[9 mod 12]) mod 12
= (11+9+10) mod 12 = 6
Swap = (S[9], S[6])
Iterasi 11

i = 10
j = (6+S[10]+K[10 mod 12]) mod 12
= (6+10+11) mod 12 = 3
Swap = (S[10], S[3])
Iterasi 12
i = 11
j = (3+S[11]+K[11 mod 12]) mod 12
= (3+11+13) mod 12 = 3
Swap = (S[11], S[3])

Setelah menemukan kunci untuk tiap karakter, maka selanjutnya dilakukan operasi XOR antara karakter pada *plaintext* dengan kunci yang telah dihasilkan. Berikut ini merupakan proses XOR dari *plaintext* dengan *key* yang didapatkan.

Tabel 2. Proses XOR dari Plaintext Dengan Key

Plainteks	Binary	Key	Ciphertext
S	01010011	00110010	01110011
E	01100101	00110110	01110111
L	01101100	00110110	01111110
A	01100001	00110100	01110101
M	01101101	00110101	01111101
A	01100001	00110000	01110001
T	01110100	00110010	01110110
Spasi	00100000	00111000	00111000
P	01010000	00111001	01111001
A	01000001	00110110	01110111
G	01000111	00110011	01110111
I	01001001	00110011	01111011

3.3 Penerapan Algoritma Elgamal

Proses User A ingin mengirimkan data teks berbentuk pesan "Selamat PAGI", kepada user B melalui user C. User A tidak ingin data tersebut tersebut diketahui oleh user C, maka user A mengenkripsi data teks tersebut untuk sampai kepada user B, selanjutnya user A memberikan data teks tersebut dan kunci pribadi (*private key*) untuk proses dekripsi kepada user B melalui user C. Untuk penyelesaian manual enkripsi dan dekripsi menggunakan algoritma Elgamal, antara lain :

a. Pembentukan Kunci
User A melakukan proses untuk membangkitkan pasangan kunci dengan memilih bilangan :
p = 787 g = 185 x = 32
Selanjutnya p, g, x dapat dipergunakan untuk menghitung y :
$y = g^x \bmod p$
$y = 185^{32} \bmod 787$
y = 754
Maka kunci publik yang dimiliki user A adalah y =754 g =185 p = 787 dan kunci *private* yang akan dikirimkan ke user B untuk proses dekripsi adalah x = 32 p = 787

b. Proses Enkripsi
Proses enkripsi yang pertama harus dilakukan adalah melakukan konversi data teks yang berbentuk pesan tersebut ke dalam bilangan desimal dengan menggunakan nilai ASCII.
Selanjutnya nilai ASCII tersebut dimasukkan ke dalam blok-blok nilai M secara berurutan, dengan perhitungan sebagai berikut.

Plainteks	S	e	l	a	m	a	t	spasi	P	A	G	I
Binary	83	101	108	97	109	97	116	32	80	65	71	73

1. $M_1 = 83$
Generate k = 673, $1 \le k \le p-2$
$a = g^k \bmod p$ $b = y^k * m \bmod p$
$a = 185^{673} \bmod 787$ $b = 754^{673} * 83 \bmod 787$
a = 347 b = 531
2. $M_2 = 101$
Generate k = 220, $1 \le k \le p-2$
$a = g^k \bmod p$ $b = y^k * m \bmod p$
$a = 185^{220} \bmod 787$ $b = 754^{220} * 101 \bmod 787$
a = 267 b = 225
3. $M_3 = 108$

Generate k = 497, $1 \leq k \leq p-2$

$a = g^k \bmod p$	$b = y^k * m \bmod p$
$a = 185^{497} \bmod 787$	$b = 754^{497} * 108 \bmod 787$
$a = 447$	$b = 506$

4. $M_4 = 97$

Generate k = 535, $1 \leq k \leq p-2$

$a = g^k \bmod p$	$b = y^k * m \bmod p$
$a = 185^{535} \bmod 787$	$b = 754^{535} * 97 \bmod 787$
$a = 147$	$b = 728$

5. $M_5 = 109$

Generate k = 229, $1 \leq k \leq p-2$

$a = g^k \bmod p$	$b = y^k * m \bmod p$
$a = 185^{229} \bmod 787$	$b = 754^{229} * 109 \bmod 787$
$a = 643$	$b = 18$

6. $M_6 = 97$

Generate k = 66, $1 \leq k \leq p-2$

$a = g^k \bmod p$	$b = y^k * m \bmod p$
$a = 185^{66} \bmod 787$	$b = 754^{66} * 97 \bmod 787$
$a = 279$	$b = 197$

7. $M_7 = 116$

Generate k = 457, $1 \leq k \leq p-2$

$a = g^k \bmod p$	$b = y^k * m \bmod p$
$a = 185^{457} \bmod 787$	$b = 754^{457} * 116 \bmod 787$
$a = 485$	$b = 261$

8. $M_8 = 32$

Generate k =530, $1 \leq k \leq p-2$

$a = g^k \bmod p$	$b = y^k * m \bmod p$
$a = 185^{530} \bmod 787$	$b = 754^{530} * 32 \bmod 787$
$a = 381$	$b = 711$

9. $M_9 = 80$

Generate k = 780, $1 \leq k \leq p-2$

$a = g^k \bmod p$	$b = y^k * m \bmod p$
$a = 185^{780} \bmod 787$	$b = 754^{780} * 80 \bmod 787$
$a = 253$	$b = 208$

10. $M_{10} = 65$

Generate k =761, $1 \leq k \leq p-2$

$a = g^k \bmod p$	$b = y^k * m \bmod p$
$a = 185^{761} \bmod 787$	$b = 754^{761} * 65 \bmod 787$
$a = 320$	$b = 237$

11. $M_{11} = 71$

Generate k =597, $1 \leq k \leq p-2$

$a = g^k \bmod p$	$b = y^k * m \bmod p$
$a = 185^{597} \bmod 787$	$b = 754^{597} * 71 \bmod 787$
$a = 546$	$b = 759$

12. $M_{12} = 73$

Generate k = 635, $1 \leq k \leq p-2$

$a = g^k \bmod p$	$b = y^k * m \bmod p$
$a = 185^{635} \bmod 787$	$b = 754^{635} * 73 \bmod 787$
$a = 750$	$b = 709$

Setelah mendapatkan nilai a dan b, maka disusun dengan pola a1, b1, a2, b2, a3, b3, a4, b4, a5, b5, a6, b6, a7, b7, a8, b8, sehingga akan membentuk *ciphertetxt* 347 531, 267 225, 447 506, 147 728, 643 18, 279 197, 485 261, 381 711, 253 208, 320 237, 546 759, 750 709.

Tabel 3. Hasil Enkripsi Elgamal

Karakter (M)	ASCII	Kunci (k) Acak	$a = g^k \bmod p$	$b = y^k * M \bmod p$	Cipher (a, b)
S	83	673	347	531	[347,531]
E	101	220	267	225	[267,225]
L	108	497	447	506	[447,506]
A	97	535	147	728	[147,728]
M	108	299	643	18	[643,18]

A	97	66	279	197	[279,197]
T	116	457	485	261	[485,261]
Spasi	32	530	381	711	[381,711]
P	80	780	253	208	[253,208]
A	65	761	320	237	[320,237]
G	71	597	546	759	[546,759]
I	73	635	750	709	[750,709]

c. Proses Dekripsi

User B ingin mendeskripsikan data teks dari user A dengan menggunakan rumus :

$M_i = b_i * (a^x)^{-1} \bmod p$ dan kunci private x = 32 dan p = 787

1. Cipher (a, b) = (347, 531)

$(a^x)^{-1} = a^{p-1-x} \bmod p$	$m = b * (a^x)^{-1} \bmod p$
$(a^x)^{-1} = 347^{787-1-32} \bmod 787$	$m = 531 * 593 \bmod 787$
$(a^x)^{-1} = 347^{754} \bmod 787$	$m = 159075 \bmod 257$
$(a^x)^{-1} = 593$	$m = 83$

2. Cipher (a, b) = (267, 225)

$(a^x)^{-1} = a^{p-1-x} \bmod p$	$m = b * (a^x)^{-1} \bmod p$
$(a^x)^{-1} = 267^{787-1-32} \bmod 787$	$m = 225 * 707 \bmod 787$
$(a^x)^{-1} = 267^{754} \bmod 787$	$m = 8114 \bmod 257$
$(a^x)^{-1} = 707$	$m = 101$

3. Cipher (a, b) = (447, 506)

$(a^x)^{-1} = a^{p-1-x} \bmod p$	$m = b * (a^x)^{-1} \bmod p$
$(a^x)^{-1} = 447^{787-1-32} \bmod 787$	$m = 506 * 725 \bmod 787$
$(a^x)^{-1} = 447^{754} \bmod 787$	$m = 366850 \bmod 257$
$(a^x)^{-1} = 725$	$m = 108$

4. Cipher (a, b) = (147, 728)

$(a^x)^{-1} = a^{p-1-x} \bmod p$	$m = b * (a^x)^{-1} \bmod p$
$(a^x)^{-1} = 147^{787-1-32} \bmod 787$	$m = 728 * 732 \bmod 787$
$(a^x)^{-1} = 147^{754} \bmod 787$	$m = 532896 \bmod 257$
$(a^x)^{-1} = 732$	$m = 97$

5. Cipher (a, b) = (643, 18)

$(a^x)^{-1} = a^{p-1-x} \bmod p$	$m = b * (a^x)^{-1} \bmod p$
$(a^x)^{-1} = 643^{787-1-32} \bmod 787$	$m = 18 * 487 \bmod 787$
$(a^x)^{-1} = 643^{754} \bmod 787$	$m = 8766 \bmod 257$
$(a^x)^{-1} = 487$	$m = 109$

6. Cipher (a, b) = (279, 197)

$(a^x)^{-1} = a^{p-1-x} \bmod p$	$m = b * (a^x)^{-1} \bmod p$
$(a^x)^{-1} = 279^{787-1-32} \bmod 787$	$m = 197 * 388 \bmod 787$
$(a^x)^{-1} = 279^{754} \bmod 787$	$m = 76436 \bmod 257$
$(a^x)^{-1} = 388$	$m = 97$

7. Cipher (a, b) = (485, 261)

$(a^x)^{-1} = a^{p-1-x} \bmod p$	$m = b * (a^x)^{-1} \bmod p$
$(a^x)^{-1} = 485^{787-1-32} \bmod 787$	$m = 261 * 700 \bmod 787$
$(a^x)^{-1} = 485^{754} \bmod 787$	$m = 182700 \bmod 257$
$(a^x)^{-1} = 700$	$m = 116$

8. Cipher (a, b) = (381, 711)

$(a^x)^{-1} = a^{p-1-x} \bmod p$	$m = b * (a^x)^{-1} \bmod p$
$(a^x)^{-1} = 381^{787-1-32} \bmod 787$	$m = 711 * 41 \bmod 787$
$(a^x)^{-1} = 381^{754} \bmod 787$	$m = 29151 \bmod 257$
$(a^x)^{-1} = 41$	$m = 31$

9. Cipher (a, b) = (253, 208)

$(a^x)^{-1} = a^{p-1-x} \bmod p$	$m = b * (a^x)^{-1} \bmod p$
$(a^x)^{-1} = 253^{787-1-32} \bmod 787$	$m = 208 * 182 \bmod 787$
$(a^x)^{-1} = 253^{754} \bmod 787$	$m = 37856 \bmod 257$
$(a^x)^{-1} = 182$	$m = 80$

10. Cipher (a, b) = (320, 237)

$(a^x)^{-1} = a^{p-1-x} \bmod p$	$m = b * (a^x)^{-1} \bmod p$
$(a^x)^{-1} = 320^{787-1-32} \bmod 787$	$m = 237 * 422 \bmod 787$
$(a^x)^{-1} = 320^{754} \bmod 787$	$m = 100014 \bmod 257$
$(a^x)^{-1} = 422$	$m = 65$

11. Cipher (a, b) = (546, 759)

$(a^x)^{-1} = a^{p-1-x} \bmod p$	$m = b * (a^x)^{-1} \bmod p$

$(a^x)^{-1} = 546^{787-1-32} \bmod 787$ m = 759 * 138 mod 787
$(a^x)^{-1} = 546^{754} \bmod 787$ m = 104742 mod 257
$(a^x)^{-1} = 138$ m = 71

12. Cipher (a, b) = (750, 709)

$(a^x)^{-1} = a^{p-1-x} \bmod p$ m = b * $(a^x)^{-1}$ mod p
$(a^x)^{-1} = 750^{787-1-32} \bmod 787$ m = 709 * 554 mod 787
$(a^x)^{-1} = 750^{754} \bmod 787$ m = 392786 mod 257
$(a^x)^{-1} = 554$ m = 73

Tabel 4. Hasil Deksripsi Elgamal

Cipher (a,b)	$(a^x)^{-1} = a^{p-1-x}$ mod p	M = b*$(a^x)^{-1}$ mod p	Karakter (m)
[347,531]	593	83	S
[267,225]	707	101	E
[447,506]	725	108	L
[147,728]	732	97	A
[643,18]	487	109	M
[279,197]	388	97	A
[485,261]	700	116	T
[381,711]	41	32	Spasi
[253,208]	182	80	P
[320,237]	422	65	A
[546,759]	138	71	G
[750,709]	554	73	I

3.4 Perbandingan Metode

Berdasarkan perbandingan waktu enkripsi dan dekripsi antara algoritma RSA dan Elgamal dengan melakukan pengujian maka didapatkan hasil sebagai berikut.

Tabel 5. Perbandingan Waktu Enkripsi dan Deksripsi Antara Algoritma RSA dan Elgamal

Ukuran Data (Bit)	Tingkat Keamanan	Enkripsi (Detik)		Dekripsi (Detik)		Total Waktu	
		Elgamal	**RSA**	**Elgamal**	**RSA**	**Elgamal**	**RSA**
64 Bit	80	2.1685	0.1366	5.9099	5.5372	8.0874	5.6738
	112	9.9855	0.1635	6.9333	20.4108	16.9188	20.5743
	128	15.0882	0.1672	7.3584	46.4782	22.4466	46.6454
	144	20.2308	0.1385	8.4785	77.7642	28.7093	77.9027
256 Bit	80	7.9240	0.5596	22.8851	19.3177	30.8091	19.8772
	112	39.7008	0.5815	26.3331	102.0337	66.0339	102.6153
	128	58.4386	0.5611	27.4060	209.6086	85.8446	210.1697
	144	77.5034	0.5718	32.1522	311.0649	109.6556	311.6368

Hasil perbandingan di atas maka dapat diketahui bahwa pada saat proses enkripsi algoritma RSA memiliki waktu enkripsi yanglebih unggul jika dibandingkan dengan algoritma Elgamal, sementara pada proses dekripsi algoritma Elgamal memiliki waktu proses yang lebih unggul dibandingkan algoritma RSA. Pada total waktu enkripsi dan dekripsi algoritma Elgamal memiliki waktu proses yang lebih unggul, disamping perbandingan waktu tersebut, dapat dilihat pada tingkat keamanan yang sama algoritma Elgamal memiliki ukuran besar kunci yang lebih kecil dibandingkan algoritma RSA.

4. KESIMPULAN

Hasil dari penelitian dan pembahasan dari pengamanan *text chat*, maka dapat dibuat kesimpulan antara lain :

1. Algoritma RC4 dan Elgamal dalam proses enkripsi serta dekripsi telah mampu melakukan pengamanan data teks sesuai dengan prosedur yang ditentukan.
2. Proses perbandingan algoritma RC4 dan Elgamal berdasarkan waktu enkripsi dan dekripsi menyimpulkan bahwa algoritma Elgamal memiliki total waktu yang lebih baik dari algoritma RC4, sehingga dapat dinyatakan bahwa algoritma Elgamal lebih cocok dalam melakukan pengamanan data teks.
3. Proses pengujian dengan menggunakan aplikasi yang dirancang menghasilkan proses enkripsi dan dekripsi tidak sesuai dengan kaidah perbandingan algoritma RC4 dan Elgamal.

UCAPAN TERIMA KASIH
Ucapan terima kasih dapat diperuntukkan kepada pihak-pihak yang telah membantu dalam penelitian, instansi yang menjadi objek penelitian, bisa juga kepada pihak yang membantu dalam publikasi artikel.

REFERENSI

[1] Afnibar, 2020, "Pemanfaatan Whatsapp Sebagai Media Komunikasi Antara Dosen dan Mahasiswa Dalam Menunjang Kegiatan Belajar (Studi Terhadap Mahasiswa UIN Imam Bonjol Padang), Jurnal Komunikasi dan Penyiaran Islam, Vol. 11, No. 1
[2] Alif Khamsyar, 2022, " Aplikasi Enkripsi Gambar Menggunakan Metode (Rivest Shamir Adleman) RSA", Jurnal Sintaks Logika, Vol. 2, No. 3.
[3] Yuri Ariyanto, 2020"Algoritma RC4 dalam proteksi transmisi dan hasil query untuk ORDBMS POSTGRESOL", Jurnal Informatika, Vol. 10, No. 1
[4] Rahmad Awaludin, *et.al*, 2022, "Implementasi Algoritma RC4 Untuk Keamanan File Berbasis Web Pada SDIT Ar Rahman", Jurnal SENAFTI,
[5] Ahir Yugo Nugroho, *et.al*, 2022, "Penerapan Elgamal Guna Meningkatkan Keamanan Data Text dan Docx", Jurnal IT, Vol. 10, No. 1
[6] Fahcriyan Rizki Ibrahim, *et.al*, 2019, "Implementasi Algoritma Elgamal Dalam Proses Enkripsi dan Dekripsi Untuk Pengamanan Citra Digital Pada Perangkat Mobile", Seminar TEKNO, Vol. 4
[7] Wahyudi, *et.al*, 2022, "Implementasi Algoritma Kriptografi Rivest Shamir Adleman Untuk Mengamankan Data Ijazah Pada SMK Swasta Prama Artha Kab. Simalungun", Jurnal JIKI, Vol. 2, No. 1
[8] Hilda Dwi, *et.al*, 2023, "Implementasi Kriptografi Advanced Encryption Standard 128 Bit Dalam Pengamanan Data Keuangan Kas", Jurnal Jukomtek, Vol. 01, No. 02
[9] Julieta Adhelia Pratiwi, 2022, " Penggunaan QR Code Berbasis Kriptografi Menggunakan Algoritma *Elliptic Curve Cryptography"*, JINACS, Vol. 3, No. 4
[10] SB. Sinaga, 2018, "Pengamanan Pesan Komunikasi Menggunakan Algoritma RSA, Rabbin Miller dan Fungsi SHA-1 Serta Penanganan Man In The Middle Attack Dengan Interlock Protocol", Jurnal Teknik Informatika Unika St. Thomas, Vol. 03, No. 01, ISSN : 2548-1916
[11] Adiguna Ahlul, *et.al*, 2019, "Metode Algoritma RC4 (Rivest Code 4) Untuk Pengamanan Database Transaksi Pada Glory Digital Sablon, Jurnal EXPLORE, Vol. 13, No. 1
[12] M. Taufiq Tamam, *et.al*, 2017, "Penerapan Algoritma Kriptorgafi Elgamal Untuk Pengamanan File Citra", Jurnal EECCIS, Vol. IV, No. 1
[13] Sri Melvani Hari, *et.al*, 2018, ""Implementasi Kriptografi Hibrid dengan algoritma Elgamal dan Algoritma One Time Pad dalam Pengamanan File Audio" jurnal TECHSI, Vol. 10, No. 2