JITET (Jurnal Informatika dan Teknik Elektro Terapan)
Vol. 12 No. 3, pISSN: 2303-0577 eISSN: 2830-7062 http://dx.doi.org/10.23960/jitet.v12i3.4794

PENERAPAN ALGORITMA KRIPTOGRAFI ELGAMAL PADA APLIKASI PENGAMANAN PESAN BERBASIS WEBSITE

Siti Nurhasanah Nugraha[1*]

[1]Sistem Informasi/Universitas Nusa Mandiri; Jl. Raya Jatiwaringin No.2, RT.8/RW.13, Cipinang Melayu, Kec. Makasar, Kota Jakarta Timur; (021) 28534471

Received: 9 Juli 2024
Accepted: 31 Juli 2024
Published: 7 Agustus 2024

Keywords:
elgamal; keamanan; kriptografi; pesan; *website*.

Corespondent Email:
siti.nhg@nusamandiri.ac.id

Abstrak. Seiring dengan perkembangan teknologi, keamanan dalam pertukaran data dan informasi menjadi sangat penting. Saat ini, keamanan dalam proses pertukaran informasi masih belum optimal, sehingga data atau informasi yang dipertukarkan sangat rentan terhadap kebocoran, penyadapan, pencurian, dan pemalsuan yang menimbulkan kerugian bagi pemilik data atau informasi tersebut. Salah satu cara untuk mengatasi permasalahan ini adalah dengan menerapkan sistem kriptografi, yaitu enkripsi dan dekripsi pesan pada proses pertukaran data atau informasi. Pada penelitian ini, algoritma ElGamal diimplementasikan pada sebuah aplikasi sederhana berbasis web yang digunakan untuk pengamanan pesan teks dengan tujuan untuk menjaga kerahasiaan pesan yang dipertukarkan. Dari penelitian yang telah dilakukan, dapat disimpulkan bahwa algoritma ElGamal yang diterapkan pada aplikasi pengamanan pesan berbasis web berjalan dengan baik dan optimal. Pemilihan algoritma ini didasarkan pada keamanannya yang terletak pada kesulitan menghitung logaritma diskrit, sehingga sulit dipecahkan oleh kriptanalis. Hasil dari penelitian ini adalah terciptanya sebuah aplikasi berbasis web yang dapat melakukan proses enkripsi pesan asli menjadi pesan yang tidak dapat dibaca, kemudian mengembalikannya ke bentuk semula melalui proses dekripsi sehingga dapat dibaca kembali oleh pengguna. Aplikasi ini dapat diterapkan untuk menciptakan lingkungan pertukaran informasi yang aman dan terpercaya.

Abstract. Along with the development of technology, security in the exchange of data and information is very important. Currently, security in the information exchange process is still not optimal, so that the data or information exchanged is very vulnerable to leakage, eavesdropping, theft, and forgery which causes losses to the owner of the data or information. One way to overcome this problem is to apply a cryptography system, namely encryption and decryption of messages in the process of exchanging data or information. In this research, the ElGamal algorithm is implemented in a simple web-based application that is used to secure text messages with the aim of maintaining the confidentiality of the messages exchanged. From the research that has been done, it can be concluded that the ElGamal algorithm applied to web-based message security applications runs well and optimally. The selection of this algorithm is based on its security which lies in the difficulty of calculating discrete logarithms, making it difficult to solve by cryptanalysts. The result of this research is the creation of a web-based application that can encrypt the original message into an unreadable message, then restore it to its original form through the decryption process so that it can

be read again by the user. This application can be applied to create a secure and trusted information exchange environment.

1. PENDAHULUAN

Pesatnya kemajuan teknologi informasi telah menjadikan informasi sebagai kebutuhan dasar bagi setiap individu, karena dapat membantu mereka untuk terus berkembang. Pemanfaatan teknologi saat ini membuat setiap individu saling bertukar informasi dengan mudah dan cepat. Namun tidak sedikit proses pertukaran informasi antar individu atau entitas dapat menimbulkan resiko terjadinya kejahatan, seperti pencurian, penyadapan bahkan sampai pemalsuan informasi [1]. Sehingga keaslian dan kerahasiaan pesan (data atau informasi) yang dipertukarkan tersebut tidak dapat terjamin lagi.

Keamanan adalah kondisi yang menunjukkan berada dalam keadaan bebas dari ancaman. Keamanan pesan (data atau informasi) dapat diartikan sebagai perlindungan data atau informasi dalam sebuah sistem dari akses yang tidak sah, perubahan, atau perusakan terhadap data atau informasi tersebut [2]. Keamanan dalam pengiriman pesan (data atau informasi) menjadi sangat penting, terutama jika data tersebut mengandung informasi yang sangat penting dan rahasia. Namun, seringkali hal ini menjadi prioritas terakhir dalam daftar hal-hal yang dianggap penting.

Salah satu cara untuk mengamankan pesan (data atau informasi) dari tindak kejahatan adalah menggunakan konsep kriptografi. Kriptografi adalah ilmu yang mempelajari cara menjaga keamanan data atau pesan selama pengiriman, agar tetap terlindungi dari gangguan pihak ketiga. Dengan kata lain, kriptografi adalah ilmu dan seni untuk menjaga pesan-pesan tetap aman dan kerahasiaannya tetap terjaga [3]. Algoritma kriptografi bekerja dengan cara menyamarkan data atau informasi menjadi bentuk kode yang tidak bermakna sehingga tidak mudah terbaca. Enkripsi (*encryption*) adalah proses mengubah teks biasa menjadi teks terenkripsi, yaitu sebuah metode untuk mengkodekan data sehingga informasinya tetap aman dan tidak dapat dibaca tanpa melalui proses dekripsi [4]. Sedangkan dekripsi adalah kebalikan dari proses enkripsi, yaitu proses mengubah data yang sudah dienkripsi menjadi data aslinya sehingga dapat dibaca.

Algoritma ElGamal adalah salah satu algoritma dalam kriptografi yang tergolong sebagai algoritma asimetris karena menggunakan kunci yang berbeda untuk enkripsi dan dekripsi [5]. Algoritma ElGamal digunakan dalam kriptografi untuk melindungi pesan (informasi atau data) karena algoritma ini memerlukan pembentukan kunci yang melibatkan bilangan prima dan penyelesaiannya menggunakan logaritma diskrit, yang sulit dipecahkan. Sehingga keamanan algoritma ini terletak pada sulitnya menghitung logaritma diskrit.

Penelitian tentang penerapan algoritma kriptografi dalam mengamankan pesan (data atau informasi) telah banyak dilakukan. Penelitian oleh [6] penerapan kriptografi digunakan untuk pengamanan data digital dengan menerapkan algoritma RSA (*Rivest Shamir Adleman*). Penggunaan algoritma RSA telah berhasil digunakan untuk mengamankan data pesan text karena algoritma ini memakai 2 buah kunci yaitu kunci publik (*public key*) dan kunci privat (*private key*) sehingga akan lebih dapat mengamankan data tersebut. Pada penelitian oleh [7] kriptografi digunakan untuk memodifikasi deretan simbol dan huruf pada *smartphone* dan laptop menggunakan metode *Caesar chiper*. Pada penelitian ini dari simbol yang telah diacak pada proses dekripsi menghasilkan huruf-huruf yang tidak mudah dibaca sehingga membuat proses enkripsi menjadi lebih sulit bagi pihak yang tidak memiliki akses atau hak dalam pesan tersebut.

Selain itu penelitian oleh [8] algoritma kriptografi ElGamal digunakan untuk pengamanan identitas pelapor pada Pusat Pelayanan Terpadu Pemberdayaan Perempuan dan Anak (P2TP2A) Kota Denpasar. Dari proses autentikasi, otorisasi, enkripsi dan pengujian menggunakan *Avalanche Effect* menunjukkan hasil rata-rata sebesar 58,2 % dimana hasil tersebut dikategorikan baik. Berdasarkan hasil tersebut algoritma ElGamal telah berhasil dalam menjaga kerahasiaan identitas pelapor sehingga tingkat keamanan

aplikasi pelaporan tindak kekerasan dalam rumah tangga ini dinyatakan aman dari serangan kriptanalis.

Algoritma kriptografi ElGamal juga telah digunakan pada penelitian [9] untuk optimalisasi keamanan data teks. Untuk menjaga kerahasiaan data teks pada penelitian ini menggunakan kombinasi antara algoritma kriptografi ElGamal dan *Vigenere Cipher*. Hasil penelitian ini menunjukkan bahwa proses enkripsi dengan kombinasi dari kedua algoritma ini berjalan dengan baik dan optimal. Pesan teks memiliki tingkat keamanan yang tinggi karena menggunakan banyak kunci dan pergeseran alfabet yang kompleks, sehingga sulit untuk dipecahkan oleh kriptanalis.

Sedangkan pada penelitian [10] algoritma ElGamal digabungkan dengan algoritma XOR digunakan untuk pengamanan data teks pada aplikasi berbasis desktop. Algoritma XOR yang merupakan algoritma sederhana yang menggunakan prinsip logika, sementara algoritma ElGamal mengandalkan kekuatan kuncinya dalam memecahkan masalah logaritma diskrit. Penggabungan kedua algoritma ini berhasil diterapkan pada aplikasi ini, sehingga pesan teks memiliki keamanan yang berlapis karena memiliki banyak kunci.

Dalam penelitian ini akan mengimplementasikan Algoritma Kriptografi ElGamal dalam sistem keamanan pesan berbasis *website* untuk meningkatkan keaslian dan kerahasiaan informasi yang dikirimkan. Dengan memanfaatkan kekuatan algoritma ElGamal dalam enkripsi dan dekripsi menggunakan kunci asimetris, penelitian ini bertujuan untuk mengembangkan metode pengamanan pesan yang lebih efektif sehingga informasi yang dikirimkan tetap terjaga kerahasiaannya dan meminimalisir risiko terhadap serangan kriptanalis. Hasil dari penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam menciptakan lingkungan pertukaran informasi yang aman dan terpercaya.

2. TINJAUAN PUSTAKA

2.1. Enkripsi dan Dekripsi

Enkripsi adalah sebuah proses untuk mengamankan data dengan menyembunyikannya atau mengubah data menjadi bentuk yang tidak dapat dibaca atau dimengerti [11]. Enkripsi telah dimanfaatkan untuk melindungi komunikasi di berbagai negara, namun hanya organisasi tertentu dan individu dengan kebutuhan yang sangat mendesak akan kerahasiaan yang menggunakannya.

Sedangkan dekripsi adalah kebalikan dari enkripsi. Dekripsi diartikan sebagai proses mengubah data yang telah dienkripsi menjadi data aslinya sehingga dapat dibaca atau dimengerti kembali [11]. Pesan yang akan dienkripsi disebut *plaintext* dan dilambangkan sebagai P, proses enkripsi dilambangkan sebagai E, proses dekripsi dilambangkan sebagai D, dan pesan yang telah dienkripsi disebut *ciphertext* dan dilambangkan sebagai C.

2.2. Kriptografi

Kriptografi awalnya didefinisikan sebagai ilmu yang mempelajari cara menyembunyikan pesan. Namun, dalam pengertian modern kriptografi adalah ilmu yang didasarkan pada teknik matematika untuk menangani keamanan informasi, termasuk kerahasiaan, keutuhan data dan otentikasi entitas [12]. Dengan demikian, pengertian kriptografi secara modern tidak hanya berkaitan dengan penyembunyian pesan, tetapi juga melibatkan sekumpulan Teknik untuk menyediakan keamanan informasi.

2.3. Algoritma ElGamal

Algoritma ElGamal diciptakan pada tahun 1985 oleh ilmuwan asal Mesir, Taher ElGamal. Algoritma ini didasarkan pada konsep kunci publik dan awalnya digunakan untuk tanda tangan digital, namun kemudian dimodifikasi sehingga bisa digunakan untuk enkripsi dan dekripsi. Algoritma kunci publik ELGamal merupakan algoritma blok cipher yang mengenkripsi blok-blok *plaintext* menjadi blok-blok *ciphertext*, yang kemudian akan didekripsi kembali dan digabungkan menjadi *plaintext* semula [13].

Keamanan algoritma ElGamal terletak pada kesulitan perhitungan logaritma diskrit dengan modulo prima yang besar, sehingga sulit untuk dipecahkan [5]. Algoritma ini memiliki keunggulan dalam pembangkitan kunci yang menggunakan logaritma diskrit dan metode enkripsi-dekripsi yang memiliki komputasi besar, sehingga hasil enkripsinya berukuran dua kali lipat dari ukuran semula.

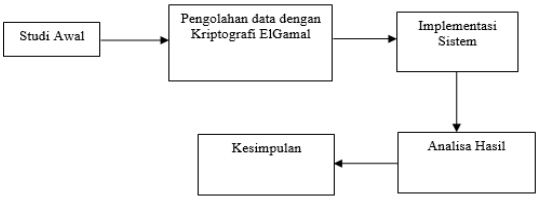
Namun, kekurangan algoritma ini adalah membutuhkan sumber daya yang besar karena ukuran *ciphertext* dua kali lebih panjang daripada *plaintext*, serta memerlukan prosesor yang mampu melakukan komputasi besar untuk perhitungan logaritma eksponensial [13]. Proses dekripsi algoritma ini juga membutuhkan waktu yang lebih lama karena kompleksitas yang tinggi, memerlukan dua kali komputasi dibandingkan dengan ukuran *ciphertext* yang lebih besar daripada *plaintext* [14].

Karena bilangan yang digunakan adalah bilangan prima, sehingga sangat sulit bahkan tidak mungkin untuk mendapatkan kunci privat dari kunci publik yang diketahui, meskipun serangan dilakukan dengan menggunakan sumber daya komputer yang sangat besar [15]. Algoritma ELGamal memiliki dua jenis kunci, yaitu kunci publik yang dapat diakses oleh umum dan kunci privat yang bersifat rahasia.

3. METODE PENELITIAN

3.1. Tahapan Penelitian

Tahapan penelitian yang dilakukan pada penelitian ini ditampilkan pada Gambar 1.



Gambar 1. Tahapan Penelitian

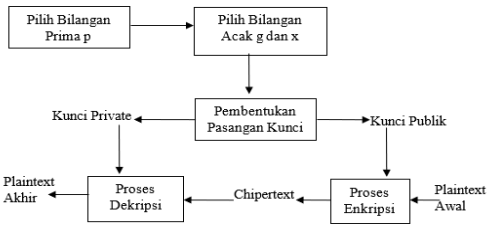
Tahap pertama yang dilakukan pada penelitian ini yaitu identifikasi permasalahan yang akan dibahas yaitu dengan mempelajari permasalahan yang terjadi. Dari hasil identifikasi yang dilakukan ditemukan bahwa masalah yang sering terjadi pada pengiriman pesan (data atau informasi) adalah minimnya keamanan pada saat proses pengiriman pesan tersebut. Selanjutnya dilakukan studi literatur untuk mencari literatur-literatur yang berkaitan dengan permasalahan yang akan dibahas. Selanjutnya adalah menganalisis masalah menggunakan teknik kriptografi dengan algoritma ElGamal untuk mencapai tujuan yang diinginkan.

Selanjutnya dilakukan implementasi sistem untuk menerapkan algoritma ElGamal tersebut ke dalam sistem informasi berbasis *website* yang akan digunakan untuk keamanan pesan. Pada tahap ini sistem yang akan dibangun menggunakan bahasa pemrograman PHP. Selanjutnya, dilakukan analisis berdasarkan pengolahan data dan implementasi dari sistem yang telah dikembangkan untuk memperoleh hasil yang dapat digunakan dalam proses enkripsi dan deskripsi.

Langkah-langkah analisis yang dilakukan meliputi perhitungan algoritma secara manual untuk proses enkripsi dan dekripsi dalam pengamanan pesan, melakukan perhitungan algoritma enkripsi dan dekripsi menggunakan sistem yang telah dikembangkan, serta membandingkan hasil perhitungan algoritma secara manual dengan perhitungan menggunakan sistem yang telah dikembangkan untuk memastikan kesesuaian hasil. Selanjutnya dilakukan penarikan kesimpulan dari seluruh tahapan penelitian yang telah dilakukan.

3.2. Tahapan Algoritma ElGamal

Langkah-langkah penyelesaian algoritma kriptografi ElGamal ditampilkan pada Gambar 2.



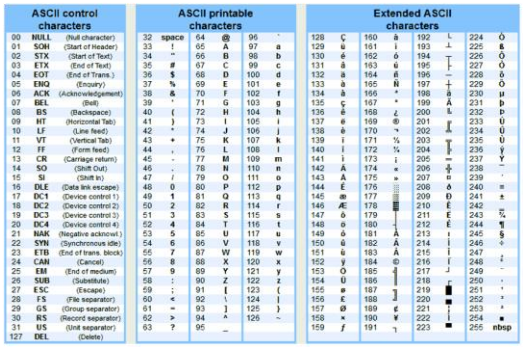
Sumber: [16]

Gambar 2. Tahapan Penyelesaian Algoritma ElGamal

a. Pilih sembarang bilangan prima p
b. Pilih 2 bilangan acak, g dan x dimana $g < p$ dan $1 \leq x \leq p-2$
c. Pembentukan pasangan kunci, yang terdiri dari kunci publik dan kunci privat
d. *User* memasukkan *plaintext* yang akan dienkripsi
e. Proses enkripsi menggunakan kunci public
f. Hasil proses enkripsi berupa *chipertext*
g. Untuk memperoleh *plaintext* kembali, dilakukan dekripsi terhadap *plaintext* menggunakan kunci privat.

Pada penelitian ini karakter teks (*plaintext*) akan dikonversi menjadi angka sebelum proses

enkripsi dengan menggunakan nilai ASCII. Setelah dikonversi menjadi angka, nilai-nilai tersebut yang akan dienkripsi menggunakan algoritma ElGamal, sehingga algoritma tersebut bekerja pada angka-angka bukan pada karakter langsung. Angka yang telah dienkripsi kemudian menjadi bagian dari *ciphertext*. Selanjutnya *ciphertext* yang berupa angka tersebut didekripsi menggunakan kunci privat. Hasil dari proses dekripsi ini berupa angka yang kemudian nantinya akan dikonversi kembali menjadi karakter menggunakan nilai ASCII sehingga menghasilkan kembali *plaintext* atau teks asli yang telah dienkripsi. Nilai ASCII yang digunakan pada penelitian ini dapat dilihat pada Gambar 3.



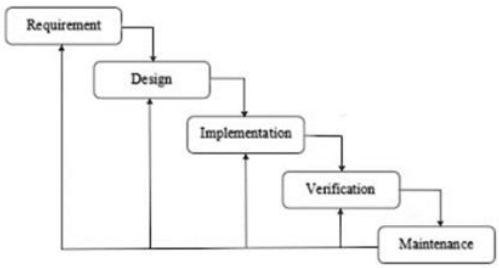
ASCII control characters			ASCII printable characters						Extended ASCII characters							
00	NULL	(Null character)	32	space	64	@	96	`	128	Ç	160	á	192	└	224	Ó
01	SOH	(Start of Header)	33	!	65	A	97	a	129	ü	161	í	193	┴	225	ß
02	STX	(Start of Text)	34	"	66	B	98	b	130	é	162	ó	194	┬	226	Ô
03	ETX	(End of Text)	35	#	67	C	99	c	131	â	163	ú	195	├	227	Ò
04	EOT	(End of Trans.)	36	$	68	D	100	d	132	ä	164	ñ	196	─	228	õ
05	ENQ	(Enquiry)	37	%	69	E	101	e	133	à	165	Ñ	197	┼	229	Õ
06	ACK	(Acknowledgement)	38	&	70	F	102	f	134	å	166	ª	198	ã	230	µ
07	BEL	(Bell)	39	'	71	G	103	g	135	ç	167	º	199	Ã	231	þ
08	BS	(Backspace)	40	(	72	H	104	h	136	ê	168	¿	200	╚	232	Þ
09	HT	(Horizontal Tab)	41	)	73	I	105	i	137	ë	169	®	201	╔	233	Ú
10	LF	(Line feed)	42	*	74	J	106	j	138	è	170	¬	202	╩	234	Û
11	VT	(Vertical Tab)	43	+	75	K	107	k	139	ï	171	½	203	╦	235	Ù
12	FF	(Form feed)	44	,	76	L	108	l	140	î	172	¼	204	╠	236	ý
13	CR	(Carriage return)	45	-	77	M	109	m	141	ì	173	¡	205	═	237	Ý
14	SO	(Shift Out)	46	.	78	N	110	n	142	Ä	174	«	206	╬	238	¯
15	SI	(Shift In)	47	/	79	O	111	o	143	Å	175	»	207	¤	239	´
16	DLE	(Data link escape)	48	0	80	P	112	p	144	É	176	░	208	ð	240	≡
17	DC1	(Device control 1)	49	1	81	Q	113	q	145	æ	177	▒	209	Ð	241	±
18	DC2	(Device control 2)	50	2	82	R	114	r	146	Æ	178	▓	210	Ê	242	‗
19	DC3	(Device control 3)	51	3	83	S	115	s	147	ô	179	│	211	Ë	243	¾
20	DC4	(Device control 4)	52	4	84	T	116	t	148	ö	180	┤	212	È	244	¶
21	NAK	(Negative acknowl.)	53	5	85	U	117	u	149	ò	181	Á	213	ı	245	§
22	SYN	(Synchronous idle)	54	6	86	V	118	v	150	û	182	Â	214	Í	246	÷
23	ETB	(End of trans. block)	55	7	87	W	119	w	151	ù	183	À	215	Î	247	¸
24	CAN	(Cancel)	56	8	88	X	120	x	152	ÿ	184	©	216	Ï	248	°
25	EM	(End of medium)	57	9	89	Y	121	y	153	Ö	185	╣	217	┘	249	¨
26	SUB	(Substitute)	58	:	90	Z	122	z	154	Ü	186	║	218	┌	250	·
27	ESC	(Escape)	59	;	91	[	123	{	155	ø	187	╗	219	█	251	¹
28	FS	(File separator)	60	<	92	\	124	|	156	£	188	╝	220	▄	252	³
29	GS	(Group separator)	61	=	93	]	125	}	157	Ø	189	¢	221	¦	253	²
30	RS	(Record separator)	62	>	94	^	126	~	158	×	190	¥	222	▌	254	■
31	US	(Unit separator)	63	?	95	_			159	ƒ	191	┐	223	▀	255	nbsp
127	DEL	(Delete)														

Sumber: [17]

Gambar 3. Nilai ASCII

3.3. Metode Pengembangan Perangkat Lunak

Metode yang digunakan pada pengembangan perangkat lunak untuk implementasi algoritma ElGamal pada sistem pengamanan pesan menggunakan metode *waterfall*. Tahapan-tahapan dalam metode *waterfall* ditampilkan pada Gambar 4.



Sumber: [18]

Gambar 4. Tahapan Metode *Waterfall*

Tahap pertama dalam metode *waterfall* adalah analisis kebutuhan untuk pengembangan aplikasi. Berdasarkan hasil analisis, dibuat desain antarmuka aplikasi, yang kemudian diubah ke dalam bahasa pemrograman melalui proses *coding*. Setelah itu, dilakukan pengujian untuk memastikan *software* bebas dari *error* dan sesuai dengan keinginan. Sistem yang telah dibuat kemudian diimplementasikan, dalam hal ini diimplementasikan pada sistem pengamanan pesan berbasis teks. Perangkat lunak mungkin mengalami perubahan setelah dikirim ke pengguna karena kesalahan yang tidak terdeteksi saat pengujian atau kebutuhan untuk beradaptasi dengan lingkungan baru. Tahap pemeliharaan dapat mengulang proses pengembangan untuk perubahan perangkat lunak yang sudah ada, tetapi tidak untuk membuat perangkat lunak baru.

4. HASIL DAN PEMBAHASAN

Analisa kebutuhan pada pembuatan aplikasi pengamanan pesan berbasis *website* terdiri dari kebutuhan pengguna, kebutuhan sistem dan kebutuhan perhitungan algoritma ElGamal. Pada analisa kebutuhan pengguna, pengguna dapat melakukan enkripsi dan dekripsi melalui menu yang tersedia di aplikasi. Hanya pengguna yang memiliki hak akses dengan kunci privat yang diberikan oleh pengguna yang melakukan enkripsi yang dapat melakukan dekripsi. Setiap pengguna dapat melakukan beberapa kali enkripsi dan dekripsi.

Selanjutnya pada Analisa kebutuhan sistem, sistem ini mampu melakukan enkripsi pesan yang diinputkan oleh pengguna, mengubah *plaintext* menjadi *ciphertext* melalui proses enkripsi, serta mengembalikan *ciphertext* ke *plaintext* melalui proses dekripsi. Selain itu, sistem dapat melakukan perhitungan enkripsi dan dekripsi menggunakan algoritma ElGamal. Hasil akhir dari proses enkripsi ditampilkan sebagai *ciphertext* yang tidak dapat terbaca, sedangkan hasil akhir dari proses dekripsi ditampilkan sebagai *plaintext*.

Sebagai contoh, pesan rahasia yang akan dikirimkan adalah “SELAMAT PAGI” yang akan dienkripsi dan didekripsi menggunakan algoritma ElGamal. Langkah-langkah yang dilakukan sebagai berikut:

a. Pembentukan Kunci

Pembangkitan pasangan kunci dilakukan dengan memilih sembarang bilangan untuk p, g dan x. Pada penelitian ini nilai p = 787, g = 185

dan x = 32. Kemudian nilai tersebut digunakan untuk menghitung y, dengan rumus seperti pada rumus (1).

$$y = g^x \ mod \ p \quad (1)$$

$$y = 185^{32} \ mod \ 787$$

$$y = 754$$

Sehingga, kunci publik untuk pesan tersebut adalah y = 754, g = 185 dan p = 787. Sedangkan, kunci privat nya adalah x = 32.

b. Enkripsi Pesan

Sebelum proses enkripsi, karakter pesan yang akan dikirim dikonversi ke dalam angka-angka berdasarkan nilai pada kode ASCII. Adapun nilai ASCII untuk pesan "Selamat PAGI" adalah 83 101 108 97 109 97 116 32 80 65 71 73. Kemudian nilai ASCII tersebut dimasukkan ke dalam blok-blok nilai m secara berurutan dengan perhitungan [19] sebagai berikut:

1) Pilih bilangan acak k di mana $1 \leq k \leq p-2$
2) Hitung $a = g^k \ mod \ p$ (2)
3) Hitung $b = m \ x \ y^k \ mod \ p$ (3)

Hasil proses enkripsi dari kata "Selamat PAGI" ditampilkan pada Tabel 1.

Tabel 1. Hasil Enkripsi Pesan

Karakter (m)	ASCII	K (acak)	$a=g^k$ mod p	$b=m \ x \ y^k$ mod 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	109	299	643	18	(643, 18)
a	97	66	279	197	(279, 197)
t	116	457	485	261	(485, 261)
	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, 759)

Berdasarkan Tabel 1 dapat dilihat bahwa setelah mendapatkan nilai a dengan perhitungan seperti pada rumus (2) dan nilai b dengan perhitungan seperti pada rumus (3) dihasilkan pola (a1 b1, a2 b2, a3 b3, a4 b4, a5 b5, a6 b6, a7 b7, a8 b8, a9 b9, a10 b10, a11 b11, a12 b12). Sehingga pola tersebut membentuk *chipertext* yang merupakan pesan enkripsi yang akan dikirim.

c. Dekripsi Pesan

Untuk proses dekripsi pesan *ciphertext* membutuhkan nilai *ciphertext* dari proses enkripsi dan kunci privat x. Selanjutnya *ciphertext* tersebut di dekripsi dengan perhitungan [19] sebagai berikut:

1) Gunakan kunci privat x untuk menghitung $(a^x)^{-1} = a^{p-1-x} \ mod \ p$ (4)
2) Hitung $m = b \ (a^x)^{-1} \ mod \ p$ (5)

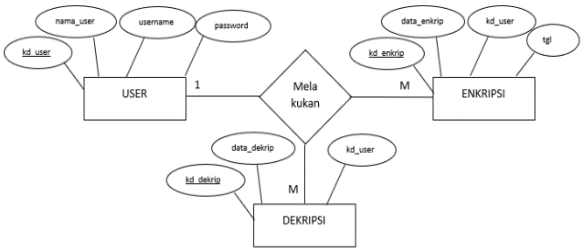
Hasil proses dekripsi yang telah dilakukan sesuai dengan perhitungan tersebut dapat dilihat pada Tabel 2.

Tabel 2. Hasil Dekripsi Pesan

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	
(253,208)	182	80	P
(320,237)	422	65	A
(546,759)	138	71	G
(750,759)	554	73	I

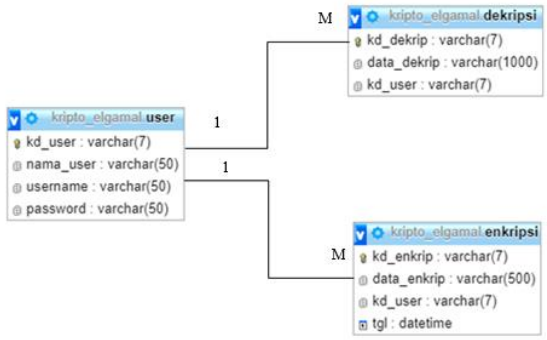
Berdasarkan Tabel 2 dapat diketahui bahwa pesan *ciphertext* telah berhasil di dekripsi menjadi pesan asli dengan perhitungan algoritma ElGamal.

Tahapan selanjutnya dalam pembuatan sistem yaitu tahap desain yang meliputi pembuatan *Entity Relationship Diagram* (ERD) dan *Logical Record Structure* (LRS). ERD yang dirancang untuk aplikasi sistem pengamanan pesan berbasis *website* dapat dilihat pada Gambar 5.



Gambar 5. *Entity Relationship Diagram*

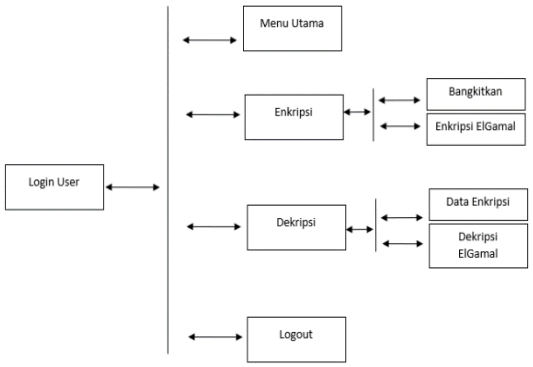
Pada Gambar 5 dapat diketahui bahwa ERD tersebut menggambarkan relasi antara tiga entitas yaitu '*user*', 'dekripsi' dan 'enkripsi' yang dihubungkan melalui relasi 'melakukan'. Diagram ini menunjukkan bahwa seorang '*user*' dapat melakukan banyak proses 'enkripsi' dan 'dekripsi' terhadap data yang berbeda-beda. Sedangkan LRS yang dirancang untuk aplikasi sistem pengamanan pesan berbasis *website* dapat dilihat pada Gambar 6.



Gambar 6. *Logical Record Structure*

Berdasarkan LRS pada Gambar 6, menggambarkan hubungan antara tiga tabel dalam sebuah *database* '*kripto_elgamal*' yang berkaitan dengan proses enkripsi dan dekripsi oleh pengguna pada sistem ini. Secara keseluruhan, LRS ini menggambarkan bagaimana data pengguna, enkripsi dan dekripsi disimpan dan dihubungkan dalam sebuah *database*. Relasi antar tabel ini yaitu '*one to many*' yang artinya, satu 'user' dapat memiliki banyak data enkripsi dan dekripsi.

Selain itu, struktur navigasi pada penelitian ini juga dirancang untuk mempermudah pengguna dalam memahami alur dan fungsi yang tersedia dalam aplikasi. Struktur navigasi pada penelitian ini dapat dilihat pada Gambar 7.



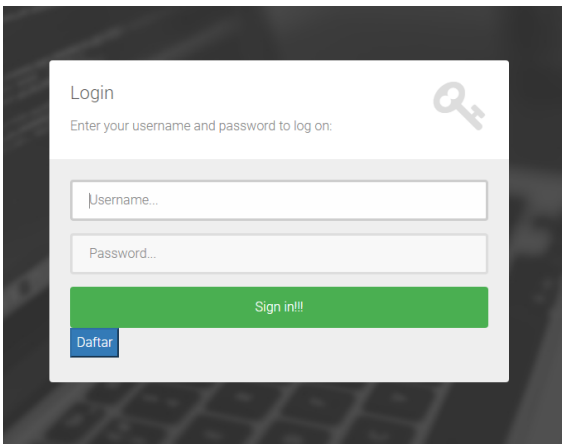
Gambar 7. Struktur Navigasi

Struktur navigasi pada Gambar 7 menjelaskan alur penggunaan aplikasi, dimana pengguna memulai login menggunakan akun masing-masing dan kemudian akan diarahkan ke halaman 'menu utama'. Pada halaman 'menu utama', pengguna dapat memilih fitur 'enkripsi', 'dekripsi' atau 'logout'. Di dalam fitur 'enkripsi' dan 'dekripsi' terdapat opsi untuk menjalankan operasi spesifik terkait enkripsi dan dekripsi data dengan menggunakan algoritma ElGamal.

Selanjutnya dari desain yang telah dibuat kemudian diubah ke dalam bahasa pemrograman melalui proses *coding* menggunakan bahasa pemrograman PHP. Sehingga menghasilkan sebuah implementasi dari sistem pengamanan pesan berbasis *website*.

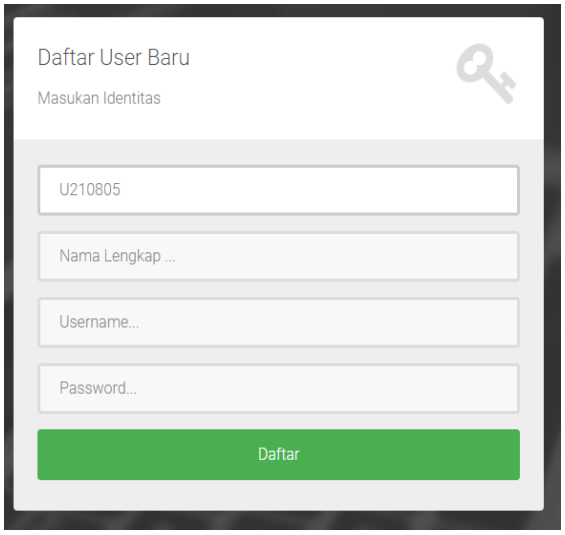
a. Halaman *Login* dan Daftar

User harus melakukan *login* terlebih dahulu untuk dapat mengakses aplikasi ini. Halaman *login* pada aplikasi ini dapat dilihat pada Gambar 8.



Gambar 8. Halaman *Login*

Untuk masuk ke dalam aplikasi diperlukan *username* dan *password* dari akun yang dimiliki masing-masing *user*. Jika *user* belum memiliki akun, maka *user* dapat melakukan daftar terlebih dahulu dengan mengklik tombol "Daftar". Tampilan halaman daftar *user* dapat dilihat pada Gambar 9.



Gambar 9. Halaman Daftar *User*

Pada halaman daftar ini *user* dapat mengisi data yang dibutuhkan untuk mendaftar akun, mulai dari nama lengkap, username dan password yang akan digunakan untuk masuk ke dalam sistem. Jika semua data sudah terisi maka *user* dapat mengklik tombol "Daftar" dan akan diarahkan ke halaman '*Login*'.

b. Halaman Menu Utama

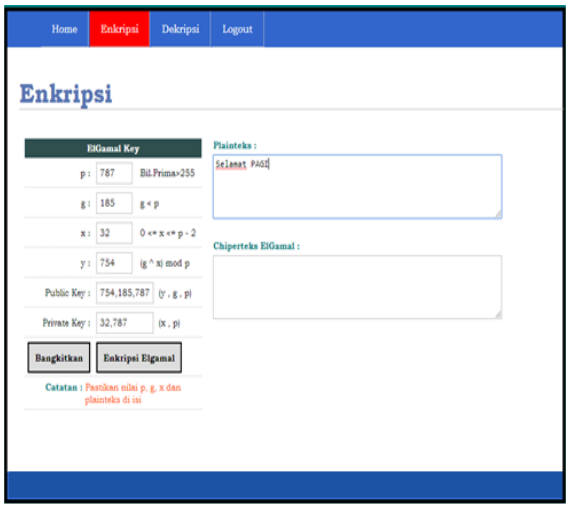
Ketika *user* sudah berhasil *login* maka *user* akan diarahkan ke tampilan halaman Menu Utama. Tampilan halaman menu utama dapat dilihat pada Gambar 10.

Gambar 10. Halaman MenuUtama

Pada halaman ini menampilkan sedikit informasi yang berkaitan dengan kriptografi. Pada menu utama terdapat 4 menu diantaranya, "Home" yang menampilkan halaman menu utama. Menu "Enkripsi" yang digunakan untuk proses enkripsi pesan, menu "Dekripsi" untuk proses dekripsi pesan dan menu "*Logout*" untuk keluar dari sistem.

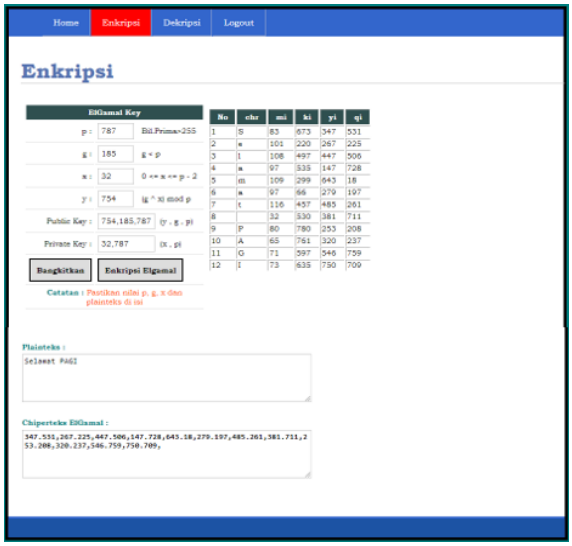
c. Halaman Enkripsi

Halaman ini merupakan halaman yang menampilkan proses enkripsi. Pada halaman ini, pengguna perlu menginputkan bilangan dan pesan *plaintext* yang diperlukan untuk proses enkripsi. Tampilan dari halaman enkripsi ini dapat dilihat pada Gambar 11.



Gambar 11. Halaman Enkripsi

Pada Gambar 11 dapat dilihat bahwa terdapat kolom "p", "g", "x" dan "y" yang merupakan parameter-parameter bilangan yang digunakan dalam algoritma ElGamal. Selain itu pada kolom "*Plaintext*" *user* dapat menginputkan kata atau informasi yang akan di enkripsi. Setelah bagian-bagian tersebut terisi, *user* dapat menekan tombol "Bangkitkan" untuk menghasilkan kunci publik dan kunci privat untuk proses enkripsi dan dekripsi. Selanjutnya *user* dapat menekan tombol "Enkripsi Elgamal" untuk melakukan proses enkripsi pada *plaintext* yang telah diinputkan menggunakan kunci yang dihasilkan. Hasil dari proses enkripsi ini sistem akan menampilkan output seperti yang ditampilkan pada Gambar 12.



Gambar 12. Tampilan Hasil Proses Enkripsi

Gambar 12 menampilkan tabel yang berisi masing-masing karakter dari *plaintext* yang sudah di enkripsi menjadi *ciphertext* dan akan otomatis terisi pada kolom "*Ciphertext ElGamal*".

d. Halaman Dekripsi

Halaman dekripsi membantu *user* untuk melakukan proses dekripsi. Pada halaman ini *user* juga dapat melihat *ciphertext* yang telah dienkripsi sebelumnya. Tampilan halaman dekripsi ditampilkan pada Gambar 13.

Gambar 13. Tampilan Halaman Dekripsi

Untuk proses dekripsi pesan, *user* harus menginputkan kunci privat pada kolom "ElGamal Key" yang telah diberikan oleh *user* yang melakukan enkripsi. Selanjutnya pengguna memilih pesan yang akan di dekripsi dari data *ciphertext* yang terdapat pada tabel. Setelah *user* memilih pesan yang akan didekripsi dan menginputkan kunci privat, *user* harus menekan tombol "Dekripsi ElGamal" untuk sistem dapat memproses dekripsi pesan. Hasil dari proses dekripsi ini akan mengembalikan pesan *ciphertext* menjadi pesan *plaintext* yang dapat dibaca. Hal ini dapat dilihat pada kolom "*Ciphertext* ElGamal" dan "*Plaintext* ElGamal".

Selanjutnya perlu dilakukan pengujian terhadap sistem yang telah dibuat untuk mengetahui apakah program menjalankan proses sesuai dengan apa yang diharapkan atau tidak. Pengujian dilakukan dengan menggunakan *black box testing* yang pengujiannya fokus pada proses masukan dan keluaran program.

a. Pengujian *Form Login*

Pengujian pada *form* ini dilakukan untuk mengetahui apakah sistem menjalankan proses *login* sesuai dengan logika program yang diberikan. Hasil pengujian pada *form login* dapat dilihat pada Tabel 3.

Tabel 3. Hasil Pengujian *Form Login*

No	Skenario pengujian	*Test case*	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	*Username* dan *Password* tidak diisi kemudian klik tombol "*Sign In*"	*Username* : (kosong) *Password* : (kosong) Akses : (kosong)	Sistem akan menolak akses dan tombol "*Sign In!!!*" tidak dapat diklik	Sesuai Harapan	*Valid*
2	Memasukan kondisi salah satu terisi dan satu lagi tidak diisi kemudian klik tombol "*Sign In*"	*Username* : (admin) *Password* : (kosong)	Sistem akan menolak akses dan tombol "*Sign In!!!*" tidak dapat diklik	Sesuai Harapan	*Valid*
3	Memasukan *Username* dan *Password* yang salah kemudian klik tombol "*Sign In*"	*Username* : (salah) *Password* : (salah)	Sistem akan menolak akses dan menampilkan notifikasi "Gagal masuk, *username* atau kata sandi anda salah!"	Sesuai Harapan	*Valid*
4	Memasukan kondisi salah satu benar	*Username* : (benar)	Sistem akan menolak akses dan	Sesuai Harapan	*Valid*

No	Skenario pengujian	*Test case*	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
	dan satu salah kemudian klik tombol "*Sign In*"	*Password* : (salah)	menampilkan notifikasi "Gagal masuk, *username* atau kata sandi anda salah!"		
5	Memasukan *Username* dan *Password* dengan data yang benar kemudian klik tombol "*Sign In*"	*Username* : admin (benar) *Password* : admin (benar) Akses : admin (benar)	Sistem menerima akses login kemudian langsung menampilkan "*login* berhasil" dan masuk ke halaman beranda admin	Sesuai Harapan	*Valid*

Hasil pengujian yang telah dilakukan seperti pada Tabel 3 menunjukkan bahwa sistem menjalankan proses *login* sesuai dengan yang diharapkan. Hal ini dapat dilihat pada kolom Kesimpulan pada Tabel 3 menampilkan bahwa setiap proses pengujian hasilnya "*Valid*" sesuai harapan.

b. Pengujian *form* enkripsi

Pengujian pada bagian ini dilakukan untuk mengetahui apakah sistem menjalankan proses enkripsi sesuai dengan logika program yang diberikan. Hasil pengujian ini dapat dilihat pada Tabel 4.

Tabel 4. Hasil Pengujian *Form* Enkripsi

No	Skenario pengujian	*Test case*	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	Bilangan prima < 255 kemudian klik "Bangkitkan"	p : diisi dengan bilangan < 255 g : diisi dengan bilangan < p x : diisi dengan bilangan < p - 2	Sistem akan menampilkan hasil *ciphertext* berupa angka 0000000	Sesuai Harapan	*Valid*
2	Salah satu bilangan tidak terisi kemudian klik "Bangkitkan"	p : diisi dengan bilangan < 255 g : kosong x : diisi dengan bilangan < p - 2	Sistem akan menampilkan notifikasi "g harus diisi"	Sesuai Harapan	*Valid*
3	Salah satu bilangan tidak terisi kemudian klik "Bangkitkan"	p : kosong g : diisi x : diisi dengan bilangan < p – 2	Sistem akan menampilkan notifikasi "p harus diisi"	Sesuai Harapan	*Valid*
4	Salah satu bilangan tidak terisi kemudian klik "Bangkitkan"	p : diisi dengan bilangan < 255 g : diisi x : kosong	Sistem akan menampilkan notifikasi "g harus diisi"	Sesuai Harapan	*Valid*
5	Bilangan g diisi dengan bilangan > p	p : diisi dengan bilangan prima > 255 g : diisi dengan bilangan > p x : diisi dengan bilangan < p - 2	Sistem akan menampilkan notifikasi "g harus lebih kecil dari p"	Sesuai Harapan	*Valid*

Hasil pengujian yang telah dilakukan seperti pada Tabel 4 menunjukkan bahwa pada *form* ini sistem menjalankan proses sesuai dengan yang diharapkan. Hal ini dapat dilihat pada kolom Kesimpulan pada Tabel 4 menampilkan bahwa setiap proses pengujian hasilnya "*Valid*" sesuai harapan.

c. Hasil Pengujian *Form* Dekripsi

Pada bagian ini dilakukan pengujian untuk mengetahui apakah sistem menjalankan proses dekripsi sesuai dengan logika program yang diberikan. Hasil pengujian dapat dilihat pada Tabel 5.

Tabel 5. Hasil Pengujian *Form* Dekripsi

No	Skenario pengujian	*Test case*	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
1	Kunci tidak sesuai dengan kunci untuk enkripsi	Kunci diisi dengan bilangan yang salah	Sistem akan menampilkan hasil dekripsi berupa *plaintext* yang tidak sesuai dengan *plaintext* awal	Sesuai Harapan	*Valid*

No	Skenario pengujian	*Test case*	Hasil yang diharapkan	Hasil pengujian	Kesimpulan
2	Kunci sesuai dengan kunci untuk enkripsi tetapi pesan *Cipherte xt* tidak diisi	Kunci : diisi dengan bilangan yang benar *Cipherte xt* : Kosong	Sistem akan menampilkan pesan "Chiper teks masih kosong.."	Sesuai Harapan	*Valid*

Hasil pengujian pada Tabel 5 menunjukkan bahwa pada *form* dekripsi sistem menjalankan proses sesuai dengan yang diharapkan. Hal ini dapat dilihat pada kolom Kesimpulan pada Tabel 5 menampilkan bahwa setiap proses pengujian hasilnya "*Valid*" sesuai harapan.

5. KESIMPULAN

Dari hasil penelitian yang telah dilakukan dapat ditarik kesimpulan, diantaranya:

a. Pengamanan pesan dapat dilakukan dengan proses enkripsi dan dekripsi untuk menjaga kerahasiaan pesan tersebut.
b. Proses enkripsi dan dekripsi yang telah dilakukan menggunakan algoritma ElGamal telah berjalan dengan baik dan optimal.
c. Dengan penggunaan algoritma ElGamal ini membuat pesan yang dikirim memiliki keamanan yang berlapis karena memiliki kunci privat sehingga tidak mudah diketahui oleh orang lain.
d. Implementasi algoritma ElGamal pada sistem berbasis *website* juga telah berjalan dengan optimal.
e. Proses enkripsi yang dilakukan pada program ini berjalan sangat baik dan pada proses dekripsi, sistem dapat mengembalikan pesan ke bentuk semula dengan benar tanpa mengubah pesan aslinya.
f. Kesimpulan dapat berupa paragraf, namun sebaiknya berbentuk poin-poin dengan menggunakan *numbering* atau *bullet*. (*The conclusion can be in the form of a paragraph, but it should be in bullet points using numbering or bullets.*)

UCAPAN TERIMA KASIH

-

DAFTAR PUSTAKA

[1] M. Maxrizal and S. Irawadi, "Analisis Sistem Kriptografi ElGamal Untuk Membentuk Sistem Kunci Publik Berbasis Grup Non-Komutatif," *J. Mat. Integr.*, vol. 16, no. 2, pp. 117–125, 2020.
[2] A. S. Sinaga, *Kemanan Komputer*. Solok: INSAN CENDEKIA MANDIRI PUBLISHER, 2020.
[3] S. Vivi Wahdini, D. Hartama, I. Okta Kirana, Poningsih, and Sumarno, "Pengamanan Data Pelanggan dan Penjualan Menggunakan Implementasi Algoritma Kriptografi," *J. Informatics Manag. Inf. Technol.*, vol. 1, no. 3, pp. 101–107, 2021.
[4] K. YUSUF, "Penerapan Algoritma Md5 Sebagai Pengaman Akun Pada Aplikasi Web Emusrenbang Kota Binjai," *JTIK (Jurnal Tek. Inform. Kaputama)*, vol. 4, no. 1, pp. 29–34, 2020.
[5] R. Solin and P. Ramadhani, "Modifikasi Pembangkit Kunci Algoritma Elgamal Dengan Menerapkan Algoritma Freivalds," *KOMIK (Konferensi Nas. …*, vol. 4, no. 1, pp. 351–356, 2020.
[6] A. T. F. Alhamdi and R. F. Siahaan, "Penerapan Kriptografi Dalam Pengamanan Pesan Text Berbasis Android Dengan Menggunakan Metode Rijndael," *J. Mahajana Inf.*, vol. 6, no. 2, pp. 69–74, 2021.
[7] P. G. Pamungkas and A. H. Muhammad, "Modifikasi Algoritma Kriptografi Caesar Chiper pada Deretan Simbol dan Huruf di Smarphone dan Laptop," *J. Inf. Technol.*, vol. 2, no. 1, pp. 1–5, 2022.
[8] N. K. A. S. Anggreni, L. Linawati, and N. P. Sastra, "Sistem Pengamanan Anonym dengan Menggunakan Algoritma Kriptografi ElGamal," *Maj. Ilm. Teknol. Elektro*, vol. 18, no. 2, 2019.
[9] B. V. Indriyono *et al.*, "Optimalisasi Keamanan Data Teks Menggunakan Kombinasi Algoritma Kriptografi ElGamal Dan Vigenere Cipher," in *Prosiding SEMNAS INOTEK (Seminar Nasional Inovasi Teknologi)*, 2023, vol. 7, pp. 18–26.
[10] K. Khairani and M. Z. Siambaton, "Pengamanan Data Teks Menggunakan Algoritma Kriptografi Elgamal dan XOR dari Serangan Hacker," *sudo J. Tek. Inform.*, vol. 2, no. 4, pp. 176–187, 2023.
[11] E. Nirmala, "Penerapan Steganografi File Gambar Menggunakan Metode Least Significant Bit (LSB) dan Algoritma Kriptografi Advanced Encryption Standard (AES) Berbasis Android," *J. Inform. Univ. Pamulang*, vol. 5, no. 1, pp. 36–47, 2020.
[12] R. Fauzi, "Implementasi Algoritma Kriptografi Elgamal Untuk Pesan Rahasia Berbasis Web Di Markas Pmi Kota Tangerang," *50 |Jurnal Ilmu Komput. JIK*, vol. VI, no. 03, pp. 50–54, 2023.
[13] S. F. Yousif, A. J. Abboud, and H. Y. Radhi, "Robust Image Encryption With Scanning Technology, the El-Gamal Algorithm and Chaos

Theory," *IEEE Access*, vol. 8, pp. 155184–155209, 2020.
[14] R. R. Judhieputra and I. N. Anisa, *Kriptografi: Penerapan dalam Keamanan Transaksi Komersial*. Indonesia Emas Group, 2024.
[15] D. Anggraini and S. Juanita, "Aplikasi E-Arsip Pengamanan Pesan Elektronik Berbasis Web dengan Mengimplementasikan Algoritma Kriptografi RSA dan Elgamal pada Klinik Dr. H. Hartono," *J. Ilm.*, vol. 6, no. 3, p. 122, 2020.
[16] M. A. Hidayat, M. Sihombing, and A. Sihombing, "Teknik Algoritma Elgamal Dan Steganografi First of File (Fof) Untuk Penyisipan Pesan Dalam Citra," *J. Tek. Inform. Kaputama*, vol. 6, no. 1, pp. 191–200, 2022.
[17] D. R. Saragi, J. M. Gultom, J. A. Tampubolon, and I. Gunawan, "Pengamanan Data File Teks (Word) Menggunakan Algoritma RC4," *J. Sist. Komput. dan Inform.*, vol. 1, no. 2, p. 114, 2020.
[18] M. Alfiyana, "Perancangan Website Untuk Media Pembelajaran Bahasa Jepang Dengan Tema Penggunaan Kata Keterangan Tingkat Dan Kuantitas," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 2, 2024.
[19] R. Munir, "Algoritma Elgamal," *Sekolah Teknik Elektro dan Informatika*, 2023. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2022-2023/20-Algoritma-Elgamal-2023.pdf. [Accessed: 20-Jun-2024].