PRISMA 4 (2021): 540-547
PRISMA, Prosiding Seminar Nasional Matematika
https://journal.unnes.ac.id/sju/index.php/prisma/
ISSN 2613-9189

Pengkodean *Polyalphabetic* dengan Modifikasi Algoritma ElGamal-*Caesar Cipher*

Rahmawati Awwaliyah Putri[a,*], Kiswara Agung Santoso[a], Ahmad Kamsyakawuni[a]

[a] *Universitas Jember, Jl. Kalimantan 37 Jember 68121 Indonesia*

** Alamat Surel: ra.putrii97@gmail.com*

Abstrak

Kriptografi adalah ilmu atau seni untuk menjaga keamanan pesan. Algoritma ElGamal merupakan bentuk algoritma asimetris dimana algoritma ini memiliki dua kunci berbeda, yakni kunci publik dan kunci privat. Algoritma ini menghasilkan karakter yang berbeda setiap karakter pada *plaintext* di enkripsi. Artikel ini akan membahas tentang proses enkripsi dan dekripsi pada modifikasi algoritma ElGamal-*Caesar Cipher* serta membandingkan keamanan modifikasi algoritma ElGamal-*Caesar Cipher* dengan algoritma ElGamal. Pembahasan ini dimulai dari pembentukan kunci publik *b* dengan menggunakan rumusan kunci algoritma ElGamal, kemudian masuk ke dalam tahap enkripsi. Analisis keamanan dilakukan dengan menghitung bobot koefisien korelasi dari modifikasi algoritma ElGamal-*Caesar Cipher* serta algoritma ElGamal dengan menggunakan *plaintext* yang sama. Hasil analisis menunjukkan bahwa modifikasi algoritma ElGamal-*Caesar Cipher* lebih baik daripada algoritma ElGamal.

Kata kunci:

Kriptografi, algoritma ElGamal, *caesar cipher*, *polyalphabetic*.

1. Pendahuluan

Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan keamanan informasi, seperti kerahasiaan data, integritas data, juga otentikasi. Bahasa sederhananya, kriptografi adalah ilmu atau seni untuk menjaga keamanan pesan. Kriptografi terdiri dari dua proses, yaitu enkripsi dan dekripsi (Munir, 2006).

Bentuk algoritma berdasarkan kunci yang digunakan terdiri dari algoritma simetris yang mempunyai satu kunci dan asimetris yang memiliki kunci ganda. Algoritma ElGamal termasuk dalam algoritma asimetris (Mulyana, 2009). Algoritma ElGamal mempunyai kunci publik berupa tiga bilangan dan kunci rahasia berupa satu bilangan. Algoritma ini menghasilkan *ciphertext* dengan jumlah dua kali lipat dari *plaintext*. Setiap karakter yang sama pada *plaintext* akan memberikan karakter *ciphertext* berbeda setiap kali dienkripsi (Ifanto, 2009).

Beberapa artikel sebelumnya yang berhubungan dengan penelitian ini yaitu Warnilah, dkk pada tahun 2018 dengan penelitiannya yang berjudul Komparasi Algoritma Kriptograsi ElGamal dan *Caesar Cipher* untuk Enkripsi dan Dekripsi Pesan; yang kedua ada Ifanto pada tahun 2009 dengan judul Metode Enkripsi dan Dekripsi dengan Menggunakan Algoritma ElGamal; dan terakhir Massandy dan Danang Tri dengan judul penelitian Algoritma ElGamal dalam Pengamanan Pesan Rahasia pada tahun 2009.

Peneliti mencoba menggabungkan algoritma ElGamal dan *Caesar Cipher* dengan modifikasi didalamnya serta menganalisis keamanan gabungan algoritma ini. Penelitian ini bertujuan untuk membandingkan keamanan dari algoritma ElGamal dengan modifikasi algoritma ElGamal-*Caesar Cipher*.

2. Metode

To cite this article:
Putri, R. A., Santoso, K. A., & Kamsyakawuni, A. (2021). Pengkodean *Polyalphabetic* dengan Modifikasi Algoritma ElGamal-*Caesar Cipher*. *PRISMA, Prosiding Seminar Nasional Matematika 4*, 540-547

Penelitian ini menggunakan data berupa data teks. Data tersebut dapat berupa huruf, angka, dan simbol yang termasuk adalam ASCII *Printable Characters*. Data teks kemudian akan dikonversi ke dalam bentuk desimal dan biner ASCII, baik dalam proses enkripsi maupun dekripsi.

2.1. Pembentukan Kunci

Kunci yang digunakan terdiri dari kunci publik (p, a, b) dan kunci rahasia c, dengan nilai $c \in \{0,1, \dots, p-2\}$ dan nilai b didapatkan dari persamaan (1) dan sebarang kunci $k_i \in \{0,1, \dots 32\}, i = 1,2, \dots, n$.

$$b = a^c mod\ P \tag{1}$$

2.2. Enkripsi

Proses enkripsi merupakan prosesn menyandikan pesan. Proses ini terdiri dari konversi *plaintext*, enkripsi, enkripsi biner dan terakhir enkripsi kode. Proses enkripsi menghasilkan *ciphertext* dan kode yang diteruskan kepada penerima pesan.

2.2.1. Konversi Plaintext.

Pesan yang dikirim dikonversi terlebih dahulu ke dalam desimal ASCII kemudian dilakukan pergeseran *plaintext* menggunakan *Caesar cipher* sebanyak k_i dengan persamaan (2) dimana *m* adalah pesan yang sudah diubah ke dalam karakter desimal.

$$m' = m_i + k_i\ mod\ P \tag{2}$$

2.2.2. Enkripsi

Bagian ini dilakukan dengan menghitung nilai (γ_i, δ_i) dengan persamaan (3) sehingga didapat pasangan hasil enkripsi (γ_i, δ_i), yang selanjutnya ditulis *m''*.

$$\gamma = a^{k_i} mod\ P \text{ dan } \delta = b^{k_i}.m\ mod\ P \tag{3}$$

2.2.3. Enkripsi Biner

Langkah berikutnya menggunakan teorema Euclidean. Teorema ini didefinisikan dalam persamaan (4) dengan nilai $q = 94$, nilai *r* adalah hasil operasi modulo *q*, dan nilai *t* akan digunakan sebagai kode yang dikirim ke penerima.

$$m'' = t\,.95 + r \tag{4}$$

Diperoleh nilai *r* yang panjangnya dua kali lipat dari pasangan hasil enkripsi. Nilai r_i di dan kunci k_i dikonversi ke dalam bentuk biner 7 bit, kemudian dilakukan operasi XOR dengan aturan lima digit terakhir r_1 di operasi XOR dengan lima digit terakhir k_1, seterusnya hingga k_i habis, kemudian r_i berikutnya di operasi XOR dengan pengulangan k_i. Didapatkan hasil XOR berupa angka-angka biner. Angka biner tersebut di konversi kembali ke dalam decimal ASCII kemudian diberlakukan persamaan (5) dimana c_i adalah *ciphertext* dan b_i adalah konversi biner *i* ke dalam desimal. Penggunaan modulo 94 dimaksudkan agar *ciphertext* berada dalam rentang ASCII *printable character* yang terbaca oleh komputer. Enkripsi kode dilakukan dengan mengelompokkan nilai *t* kedalam kelompok masing-masing berisi 6 deret angka secara berurutan lalu ubah kedalam biner 5 bit. Gabungkan 30 bit tersebut kemudian kelompokkan ulang dengan susunan 6 bit tiap blok sehingga dihasilkan 5 deret angka yang nantinya akan kembali dikonversi ke dalam karakter ASCII.

$$c_i = (b_i - 32)\ mod\ 95 + 32 \tag{5}$$

2.2.4. Enkripsi Kode

Proses ini dilakukan dengan mengambil nilai *t* yang didapat dari teorema Euclidean kemudian dikelompokkan per enam angka yang dikonversi ke biner 5 bit. Setelah didapat blok baru, biner tersebut dikonversi ke bentuk desimal dan kembali ke dalam karakter ASCII.

2.3. Dekripsi

Proses dekripsi merupakan kebalikan dari enkripsi, yaitu proses mengembalikan pesan yang tersandi ke bentuk pesan yang utuh. Proses ini terdiri dari dekripsi kode, dekripsi biner, dan dekripsi.

2.3.1. Dekripsi Kode

Kode yang diberikan pengirim dikelompokkan per lima angka yang dikonversi ke biner 6 bit yang disusun ulang menjadi 5 bit per blok. Setelah didapat kelompok baru, biner tersebut dikonversi ke bentuk desimal dan ubah kembali ke dalam karakter sesuai ASCII.

2.3.2. Dekripsi Biner

Langkah berikutnya dilakukan operasi XOR dengan k_i. Konversi kembali hasil operasi XOR biner ke desimal ASCII kemudian di masukan ke dalam persamaan (4) untuk mencari nilai *m"* dimana kode t_i sudah diberikan oleh pengirim. Nilai *m"* yang dihasilkan kemudian di urutkan dan dipasangkan secara berurutan. Pasangan berurutan *m"* selanjutnya ditulis (γ_i, δ_i).

2.3.3. Dekripsi

Proses ini dilakukan dengan menggunakan persamaan (6) dimana *c* merupakan kunci privat yang hanya diketahui penerima. Hasil dari proses ini adalah *plaintext* bayangan dan langsung dilakukan pergeseran kembali dengan kunci k_i sehingga pesan yang sebenernya bisa tersampaikan.

$$m = \delta.(\gamma^{p-1-c} \bmod p) \tag{6}$$

2.4. Bobot Koefisien Korelasi

Bobot Koefisien Korelasi digunakan untuk mengetahui hubungan tiap karakter antara *plaintext* dan *ciphertext* yang dihasilkan. Pehitungan ini diadaptasi dari pengukuran panjang teks dan juga gabungan operasi XOR (Idrus *et al*, 2008). Setiap karakter pada *ciphertext* akan diberi bobot poin 1-3. Poin 3 diberikan jika satu karakter pada *ciphertext* tidak ada pada *plaintext*; poin 2 diberikan jika terdapat satu karakter *ciphertext* yang sama dengnn *plaintext* namun posisinya berbeda; sedangkan poin terendah yaitu 1 diberikan jika satu karakter *ciphertext* dan posisinya sama dengan *plaintext*. Poin ini merupakan nilai B pada persamaan (2.6), sedangkan *n* adalah jumlah karakter keseluruhan pada *plaintext*. Hasil akhirnya jika bernilai mendekati 0, maka algoritma yang digunakan sangat baik dan hubungan antara *plaintext* dan *ciphertext* tidak bersinggungan; jika hasil akhir bernilai mendekati 1, maka algoritma yang digunakan tidak baik atau hubungan antara *plaintext* dan *ciphertext* bersinggungan.

$$BKK = \frac{n}{\sum_{i=1}^{n} B_i} \tag{7}$$

3. Hasil dan Pembahasan

3.1 Pembentukan Kunci

Proses ini dilakukan oleh penerima pesan untuk mendapatkan kunci publik *b*. Kunci ini nantinya akan diberikan kepada pembuat pesan untuk digunakan pada proses enkripsi. Kunci publik *b* didapatkan dari persamaan (1) sehingga didapat nilai kunci *b* = 105. Kunci *b* digunakan hanya untuk proses enkripsi, sedangkan kunci *c* hanya digunakan untuk proses dekripsi.

3.2 Enkripsi Modifikasi Algoritma ElGamal dan Caesar Cipher

Proses enkripsi dilakukan dengan meng-input *plaintext*, kunci publik dan kunci *k*. *plaintext* kemudian di enkripsi sehingga menghasilkan *ciphertext* acak yang dua kali lebih panjang daripada *plaintext*.

Tabel 1. *Plaintext, ciphertext* dan kunci yang digunakan

Plaintext		: sekolah
Ciphertext		: 7%A6,37=7MA5,4
Kunci publik	**P**	: 131

	a	: 9
Kunci privat	**c**	: 71
Kunci k		: kami

3.2.1 Konversi Plaintext

Proses ini bertujuan mengubah *plaintext* menjadi *plaintext* bayangan dengan menggunakan persamaan (2). Sebelumnya, *plaintext* dan kunci *k* diubah ke bentuk desimal sesuai ASCII.

Tabel 2. Konversi *plaintext* ke bentuk desimal sesuai ASCII

s	= 115	**e**	= 101	**k**	= 107	**o**	= 111
l	= 108	**a**	= 97	**h**	= 104		

Tabel 3. Konversi kunci *k* ke bentuk desimal sesuai ASCII

K	= 107	**a**	= 97	**m**	= 109	**i**	= 105

Proses pergeseran *plaintext* menggunakan persamaan (2):

$s + k = 115 + 107 \, mod \, 131 = 91$ $\quad$ $l + k = 108 + 107 \, mod \, 131 = 84$

$e + a = 101 + 97 \, mod \, 131 = 67$ $\quad$ $a + a = 97 + 97 \, mod \, 131 = 63$

$k + m = 107 + 109 \, mod \, 131 = 85$ $\quad$ $h + m = 104 + 109 \, mod \, 131 = 82$

$o + i = 111 + 105 \, mod \, 131 = 85$

Plaintext bayangan (m') yang didapat adalah 91 67 85 85 84 63 82

3.2.2 Enkripsi

Tahap ini menggunakan enkripsi dari algoritma ElGamal. Proses ini dapat dilihat pada Tabel 4.

Tabel 4. Proses enkripsi menggunakan algoritma ElGamal

No	m'_i	k_i	$\gamma = a^{k_i} \, mod \, P$ $\gamma = 9^{k_i} \, mod \, 131$	$\delta = b^{k_i} . m'_i \, mod \, P$ $\delta = 105^{k_i} . m'_i \, mod \, 131$
1	91	107	123	4
2	67	97	44	31
3	85	109	7	18
4	85	105	84	115
5	84	107	123	44
6	63	97	44	123
7	82	109	7	116

Pasangan (γ_i, δ_i) yang didapat, yakni (123,4)(44,31)(7,18)(84,115)(123,44)(44,123)(7,116).

3.2.3 Enkripsi Biner

Hasil yang didapat dari enkripsi elgamal kemudian di urutkan (m'') untuk dimasukkan ke dalam Persamaan (4). Proses tersebut dapat dilihat pada Tabel 5 berikut.

Tabel 5. Penggunaan teorema Euclidean

No	m''	t	r
1	123	1	28
2	4	0	4
3	44	0	44
4	31	0	31
5	7	0	7
6	18	0	18

7	84	0	84
8	115	1	20
9	123	1	28
10	44	0	44
11	44	0	44
12	123	1	28
13	7	0	7
14	116	1	21

Hasil dari Tabel 5 didapat nilai *t* yang akan di enkripsi kembali untuk digunakan sebagai kode dan nilai *r* yang akan dioperasi XOR untuk mendapatkan *ciphertext*. Operasi XOR dilakukan dengan mengubah *r* dan kunci *k* terlebih dahulu ke bentuk biner 7 bit. Biner 7 bit *r* kemudian dioperasi XOR dengan 5 bit terakhir dari biner kunci *k*. Berikut perhitungan operasi XOR dengan aturan pengulangan pada kunci *k* jika jumlah *plaintext* lebih banyak daripada kunci *k*.

1. 0 1 1 1 1 0 0 ⊕ 0 0 0 1 0 1 1 = 0 1 1 0 1 1 1

2. 0 1 0 0 1 0 0 ⊕ 0 0 0 0 0 0 1 = 0 1 0 0 1 0 1

⋮

13. 0 1 0 0 1 1 1 ⊕ 0 0 0 1 0 1 1 = 0 1 0 1 1 0 0

14. 0 1 1 0 1 0 1 ⊕ 0 0 0 0 0 0 1 = 0 1 1 0 1 0 0

Dari proses enkripsi terakhir didapatkan deretan biner kemudian diubah menjadi karakter sesuai ASCII sehingga didapatkan *ciphertext* : 7%A6,3y=7MA5,4

3.2.4 Enkripsi Kode

Nilai *t* yang didapat dari Tabel 5 selanjutnya akan di enkripsi untuk dijadikan kode unik kepada penerima. Nilai *t* = 1 0 0 0 0 0 0 1 1 0 0 1 0 1 0 0 0 0. Proses enkripsi dimulai dari pengelompokan nilai *t* setiap 6 angka kemudian konversi ke bentuk biner 5 bit. Setelah didapat blok biner baru, konversi biner tersebut ke dalam desimal kemudian dikonversi kembali kedalam karakter ASCII.

Tabel 6. Hasil konversi perubahan blok biner

Blok biner baru	Konversi ke desimal	+ 32	Konversi ke bentuk karakter
000010 000000 000000 000000 000000	4 8 0 32 33	34 32 32 32 32	“(sp)(sp)(sp)(sp)
000000 000100 001000 000000 000001	2 0 0 32 0	32 36 40 32 33	(sp)$((sp)!
000000 000100 000000 000000 000000	4 8 8 1 1	32 36 32 32 32	(sp)$(sp)(sp)(sp)

Keterangan: (sp) adalah karakter spasi.

Kode yang didapat adalah “(sp)(sp)(sp)(sp) (sp)$((sp)! (sp)$(sp)(sp)(sp).

3.3 Dekripsi Modifikasi Algoritma ElGamal dan Caesar Cipher

Proses dekripsi dilakukan dengan meng-*input* kan *ciphertext*, kode, kunci *k* dan kunci privat. Tahapan yang dilakukan adalah sebagai berikut.

3.3.1 Dekripsi Kode

Kode “(sp)(sp)(sp)(sp) (sp)$((sp)! (sp)$(sp)(sp)(sp) akan didekripsi terlebih dahulu. Kode tersebut dibagi dalam beberapa blok lalu di konversi ke dalam desimal kemudian dikurangi 32 dan di konversi kembali ke bentuk biner 6 bit. Hasil biner tersebut diubah kedalam diubah dalam bentuk desimal kemudian dikurangi lagi dengan 32. Setelah itu, dikelompokkan ulang 6 karakter per blok kemudian ubah ke dalam biner 5 bit.

Tabel 7. Konversi biner

Pengelompokan ulang biner	*t*
00001 00000 00000 00000 00000 00000	1 0 0 0 0 0
00000 00001 00001 00000 00000 00001	0 1 1 0 0 1
00000 00001 00000 00000 00000 00000	0 1 0 0 0 0

Didapatkan nilai *t*: 1 0 0 0 0 0 0 1 1 0 0 1 0 1 yang akan digunakan dalam dekripsi biner.

3.3.2 Dekripsi Biner

Proses dekripsi biner dilakukan dengan menggunakan *ciphertext* 7%A6,3y=7MA5,4 kunci *k* dan nilai *t* yang sudah di dekripsi sebelumnya.

Tabel 8. Konversi *ciphertext* ke bentuk desimal

7	= 55	%	= 37	A	= 65	6	= 54
,	= 44	3	= 51	y	= 121	=	= 61
7	= 55	M	= 77	A	= 65	5	= 53
,	= 44	4	= 52				

Plaintext dan kunci *k* kemudian diubah ke dalam biner 7 bit untuk dioperasi XOR dengan aturan 5 bit kunci *k* yang digunakan. Berikut operasi XOR *ciphertext* dengan kunci *k*:

1. 0 1 1 0 1 1 1
   0 0 0 1 0 1 1 ⊕
   0 1 1 1 1 0 0

2. 0 1 0 0 1 0 1
   0 0 0 0 0 0 1 ⊕
   0 1 0 0 1 0 0

⋮

13. 0 1 0 1 1 0 0
    0 0 0 1 0 1 1 ⊕
    0 1 0 0 1 1 1

14. 0 1 1 0 1 0 0
    0 0 0 0 0 0 1 ⊕
    0 1 1 0 1 0 1

Dari hasil operasi XOR tersebut, didapatkan *r* : 28 4 44 31 7 18 84 20 28 44 44 28 7 21. Nilai *r* ini akan dimasukkan ke dalam teorema Euclidean. Berdasarkan persamaan 4.1, proses untuk mendapatkan m'' ditunjukkan dalam Tabel 9.

Tabel 9. Proses dalam teorema Euclidean

No	*t*	*r*	m''
1	1	28	123
2	0	4	4
3	0	44	44
4	0	31	31
5	0	7	7
6	0	18	18
7	0	84	84
8	1	20	115
9	1	28	123
10	0	44	44
11	0	44	44
12	1	28	123
13	0	7	7
14	1	21	116

Didapat pasangan (γ_i, δ_i) : (123,4)(44,31)(7,18)(84,115)(123,44)(44,123)(7,116).

3.3.3 Dekripsi

Tahap ini dilakukan menggunakan persamaan (6) yang merupakan persamaan asli milik algoritma ElGamal. Proses pada tahap ini dapat dilihat pada Tabel 10.

Tabel 10. Proses dekripsi ElGamal

No	γ	δ	$m' = \delta\ (\gamma^{P-1-c} mod\ P)$ $= \delta\ (\gamma^{59} mod\ 131)$
1	123	4	91
2	44	31	67
3	7	18	85
4	84	115	85
5	123	44	84
6	44	123	63
7	7	116	82

Setelah itu mengonversi m'' dengan menggunakan persamaan (2). Berikut proses konversi *plaintext*..

$91 - 107\ mod\ 131 = 115$ $\quad$ $84 - 107\ mod\ 131 = 108$

$67 - 97\ mod\ 131 = 101$ $\quad$ $63 - 97\ mod\ 131 = 97$

$85 - 109\ mod\ 131 = 107$ $\quad$ $82 - 109\ mod\ 131 = 104$

$85 - 105\ mod\ 131 = 111$

Gabungkan hasil prosesnya sehingga didapat *plaintext "*sekolah".

3.4 Bobot Koefisien Korelasi

$$MCC = \frac{n}{\sum_{i=1}^{n} B_i}$$

$$MCC = \frac{7}{3+3+3+3+3+3+3+3+3+3+3+3+3+3}$$

$$MCC = \frac{7}{42} = 0.167$$

Bobot koefisien korelasi pada penelitian ini untuk algoritma ElGamal-*Caesar cipher* adalah 0.167, dan untuk algoritma ElGamal sendiri adalah 0,175. Karakter *ciphertext* pada hasil algoritma ElGamal terdapat dua karakter yang sama dengan *plaintext* awal namun berbeda posisi, sehingga poinnya berjumlah dua. Percobaan lain yang dilakukan lebih banyak menghasilkan nilai yang sama pada algoritma ElGamal-*Caesar cipher* dan algoritma ElGamal, sehingga bisa dikatakan algoritma ElGamal-*caesar cipher* lebih baik daripada algoritma ElGamal itu sendiri.

4. Simpulan

Berdasarkan penelitian yang telah dilakukan, diperoleh kesimpulan sebagai berikut. Proses modifikasi algoritma ElGamal dan *Caesar Cipher* dimulai dari pembentukan kunci untuk mendapatkan kunci publik, dilanjutkan dengan konversi *plaintext* untuk mendapatkan *plaintext* bayangan. Proses enkripsi terbagi menjadi tiga, yaitu enkripsi menggunakan algoritma Elgamal, kemudian enkripsi biner untuk mendapatkan *ciphertext* dan enkripsi kode untuk mendapatkan kode. Proses dekripsi juga terbagi menjadi tiga, yaitu dekripsi kode terlebih dahulu, kemudian dekripsi *ciphertext* menggunakan biner dan dekripsi menggunakan algoritma ElGamal. Langkah terakhir mengkonversi *plaintext* bayangan dari hasil algoritma ElGamal untuk mendapatkan *plaintext* asli. Bobot koefisien korelasi pada penelitian ini untuk algoritma ElGamal-*Caesar cipher* adalah 0.167, dan untuk algoritma ElGamal sendiri adalah 0,175. Karakter *ciphertext* pada hasil algoritma ElGamal terdapat dua karakter yang sama dengan *plaintext* awal namun berbeda posisi, sehingga poinnya berjumlah dua. Percobaan lain yang dilakukan lebih banyak menghasilkan nilai yang sama pada

algoritma ElGamal-*Caesar cipher* dan algoritma ElGamal, sehingga bisa dikatakan algoritma ElGamal-*caesar cipher* lebih baik daripada algoritma ElGamal itu sendiri.

Daftar Pustaka

Ariyus, D. (2006). *Kriptografi*. Yogyakarta: CV. Andi *Offset*.

Flourensia, Saptu Rahayu. (2005). *Suplemen Bahan Ajar Mata Kuliah proteksi dan Teknik Keamanan Sistem Informasi*, Jakarta*:* Tugas Kriptografi Fakultas Ilmu Komputer Universitas Indonesia.

Idrus, S. Z. S., Aljunid, S. A., Asi, S. M., Sudin, S., Ahmad. R. B. (2008). Performance Analysis of Encryption Algorithms' Text Length Size on Web Browsers. *International Journal of Science and Network Security (IJCSNS)*, *8*(1), 20-25.

Ifanto, M. (2009). *Metode Enkripsi dan Dekripsi dengan Menggunakan Algoritma Elgamal.* Makalah Struktur Diskrit Program Studi Informatika: Institut Teknologi Bandung.

Massandy, Danang Tri. (2009). *Algoritma Elgamal Dalam Pengamanan Pesan Rahasia.* Makalah Struktur Diskrit Program Studi Teknik Informatika: Institut Teknologi Bandung.

Mulyana, D. (2008). *Ilmu Komunikasi; Suatu Pengantar*. Bandung: Remaja Rosdakarya.

Munir, R. (2004). *Teori Bilangan (Number Theory*). Departemen Teknik Infomatika: Institut Teknologi Bandung.

Munir, R. (2006). *Diktat Kuliah IF5054 Kriptografi*. Jakarta: Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika.

Suhardi. (2016). Aplikasi Kriptografi Data Sederhana dengan Metode Exclusive-OR (XOR). *Jurnal Teknovasi*, *3*(2), 23-31.

Warnilah, Ai Ilah, Nugraha, Siti Nurhasanah. (2018). Komparasi Algoritma Kriptografi Elgamal dan Caesar Cipher untuk Enkripsi dan Dekripsi Pesan. *Indonesian Journal on Computer and Information Technology (IJCIT)*, *3*(2), 243-252.