Available *online* at : http://bit.ly/InfoTekJar

InfoTekJar : Jurnal Nasional Informatika dan Teknologi Jaringan

ISSN (Print) 2540-7597 | ISSN (Online) 2540-7600

Penerapan Algoritma One Time Pad & Linear Congruential Generator Untuk Keamanan Pesan Teks

Rachmat Aulia [1], Ahmad Zakir [2], Muhammad Zulhafiz [1]

[1] *Prodi Teknik Informatika Fakultas Teknik dan Komputer Universitas Harapan Medan, Jl.H.M Joni No.70C, Medan, Sumatera Utara, Indonesia*
[2] *Prodi Sistem Informasi Fakultas Teknik dan Komputer Universitas Harapan Medan, Jl.H.M Joni No.70C, Medan, Sumatera Utara, Indonesia*

KEYWORDS

Kriptografi, One Time Pad, Linier Congruential Generator, OTP, LCG

CORRESPONDENCE

E-mail: jackm4t@gmail.com
suratzakir@gmail.com
muhammadzulhafiz021@gmail.com

A B S T R A C T

Keamanan informasi memiliki peran penting dalam teknologi informasi. Pengiriman pesan ada baiknya dilakukan dengan menerapkan teknik kriptografi. Hal ini dilakukan untuk meminimalkan pihak-pihak yang tidak bertanggung jawab dalam melakukan pencurian atau pendayapan pesan. Kriptografi dapat diartikan sebagai pesan berbentuk teks, yang tidak diketahui maksudnya. Dalam pelaksanaannya, pesan asli ditransformasikan ke dalam bentuk tidak beraturan, dimana ketika sampai dengan pasti ke target, pesannya dapat dikembalikan lagi ke bentuk aslinya. Kriptografi diklasifikasikan dalam tiga: simetri, asimetri, dan *hash*. *One Time Pad* merupakan jenis kriptografi simetris dimana enkripsi dan dekripsinya menggunakan kunci yang sama. Penggunaan kunci yang berbeda akan mengakibatkan hasil yang berbeda. Proses pada *One time Pad* adalah panjang pesan harus sama dengan panjang kunci. Salah satu mekanisme yang dapat membantu dalam membangkitkan kunci pada algoritma *One Time Pad* adalah menggunakan pembangkit kunci yang mampu membangkitkan kunci unik yang cukup panjang sesuai dengan panjang teks yang digunakan. Pembangkit kunci tersebut adalah *Linier Congruential Generator*. LCG adalah salah satu pembangkit bilangan acak tertua dan cukup terkenal. Kombinasi dari kedua teknik ini yaitu OTP dan LCG dapat menghasilkan enkripsi dan deskripsi pesan secara efisien, sehingga pesan aman pada saat dikirim melalui internet.

PENDAHULUAN

Perkembangan teknologi semakin hari semakin meningkat. Hal ini dapat menyebabkan semakin tingginya tingkat ancaman terhadap keamanan penyebaran data dan informasi. Keamanan merupakan masalah utama yang terdapat dalam jaringan global berkaitan dengan pengiriman informasi dari sumber menuju target begitupun sebaliknya [1]. Tidak ada yang dapat memastikan pesan yang dikirim melalui jaringan internet dalam kondisi aman. Seperti yang kita ketahui, saat ini internet sangat mudah untuk diperoleh, sehingga membuka peluang atau celah bagi orang-orang yang tidak bertanggung jawab dalam mencuri informasi dan data-data pribadi yang dimiliki perusahaan maupun individual. Data atau informasi yang menjadi target pencurian adalah berkas-berkas atau dokumen digital yang mempunyai nilai tinggi (*valuable*).

Keamanan informasi merupakan prioritas utama di era saat ini. Dokumen–dokumen ataupun catatan–catatan kecil namun penting dapat menyebabkan kerugian yang cukup besar jika tidak diberikan keamanan tambahan. Untuk mengatasi itu dibutuhkan mekanisme keamanan digital yang tergolong ke dalam bidang kriptografi. Kriptografi (*cryptography*) awalnya diadopsi dari bahasa yunani yang memiliki dua suku kata: kryptos (tersembunyi) dan graphein (tulisan), jadi apabila digabungkan menjadi "tulisan tersembunyi" [2]. Kriptografi tidak hanya mencakup teknik-teknik menyandikan informasi, tetapi juga teknik untuk membongkar sandi. Terdapat dua proses utama dalam kriptografi: enkripsi dan dekripsi. Enkripsi adalah teknik yang mengubah *plaintext* menjadi *ciphertext*. Sedangkan deskripsi sebaliknya.

Kriptografi mempunyai variasi teknik dan metode, salah satunya adalah metode *On Time Pad* (OTP). *On Time Pad* adalah teknik enkripsi yang menggunakan pasangan *plaintext* dengan sebuah kunci rahasia yang diperoleh secara acak. Kemudian setiap bit dari *plaintext* dienkripsi dengan mengkombinasikan dengan bit tambahan yang diperoleh dari kunci acak menggunakan penjumlahan modulo. Keamanan dari *One Time Pad* sendiri bergantung pada kunci acak yang digunakan. Semakin tinggi jumlah kunci acak yang digunakan dan tidak menggunakannya kembali, maka keamanan *chipertext* dari *One Time Pad* akan semakin tinggi.

Penerapan kunci acak pada *One Time Pad* haruslah dapat digunakan kembali ketika ingin melakukan teknik dekripsi. Oleh karena itu diperlukan suatu pembangkit bilangan acak yang

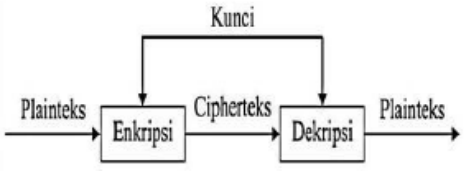
https://doi.org/10.30743/infotekjar.v4i1.1590

dapat digunakan kembali, salah satunya adalah metode Linier Congruential. *Linear Congruential Method* (LCM) adalah bilangan yang tidak dapat diprediksi kemunculannya. Tidak ada komputasi yang benar-benar menghasilkan deret bilangan acak secara sempurna [3]. *LCM* merupakan metode pembangkit bilangan acak semu yang menghasilkan bilangan acak yang memiliki periode dan sangat ditentukan oleh parameter input sehingga bilangan yang dibangkitkan tidak sepenuhnya acak.

METODE

Kriptografi

Untuk mengubah pesan asli ke dalam bentuk yang tidak beraturan diperlukan suatu tahapan. Dalam ilmu kriptografi, tahapan untuk menghasilkan pesan yang terenkripsi dapat diterapkan sesuai ilustrasi berikut:



Gambar 1. Dasar Kriptografi [4]

Teknik enkripsi dapat dijalankan dengan memasukkan pesan asli dan kunci, selanjutnya menghasilkan luaran *ciphertext* (pesan terenkripsi). *Ciphertext* dapat dikembalikan dalam bentuk aslinya (*plaintext*) dengan memasukkan kembali kunci berserta *ciphertext*-nya.

One Time Pad (OTP)

Algoritma OTP mempunyai panjang kunci yang sama dengan panjang *plaintext*, sehingga tidak ada kebutuhan untuk mengulang penggunaaan kunci selama proses enkripsi. Adapun aturan enkripsi dan dekripsi dari *one time pad* adalah sebagai berikut:

1. Enkripsi C = (Pi + Ki) mod 26
2. Dekripsi C=(Pi - Ki) mod 26

Note: P_i adalah *plaintext*

K_i adalah kunci.

Contoh: Bila diketahui Plainteks: "WANGIJERUK" dengan kunci: "*RKBIRKBIRK*". Jika kita asumsikan A = 0, B = 1 ,….., Z = 25, dengan menggunakan rumus enkripsi => "C = (Pi + Ki) mod 26", maka diperolehlah hasil *ciphertext "*NKOOZTFZKU" yang didapat dari perhitung sebagai berikut:

Tabel 1. Contoh Proses Enkripsi

Rumus	Nilai	Hasil	Char
(W + R) mod 26	= (22 + 17) mod 26 =	13	N
(A + K) mod 26	= (0 + 10) mod 26 =	1 0	K
(N + B) mod 26	= (13+ 1) mod 26 =	14	O
(G + I) mod 26	= (6 + 8) mod 26 =	1 4	O
(I+ R) mod 26	= (8 + 17) mod 26 =	2 5	Z
(J+ K) mod 26	= (9 + 10) mod 26 =	1 9	T
(E+ B) mod 26	= (4 + 1) mod 26 =	5	F
(R+ I) mod 26	= (17 + 8) mod 26 =	2 5	Z
(U+ R) mod 26	= (20 + 17) mod 26 =	1 1	K
(K+ I) mod 26	= (10 + 10) mod 26 =	2 0	U

Kemudian untuk mendekripsikannya lakukan lagi perhitungan diatas dengan menggunakan "*Pi =(Ci - Ki)* mod 26", dengan menggunakan hasil chipertext dari Enkripsi => "NKOOZTFZKU" dan kunci = "*RKBIRKBIRK*" , maka dapatlah hasil dari proses dekripsi = "WANGIJERUK" [5]

Tabel 2. Contoh Proses Dekripsi

Rumus	Nilai	Hasil	Char
(N - R) mod 26	= (13 - 17) mod 26 =	22	W
(K- K) mod 26	= (10 - 10) mod 26 =	0	A
(O - B) mod 26	= (14 - 1) mod 26 =	13	N
(O - I) mod 26	= (14 - 8) mod 26 =	6	G
(Z - R) mod 26	= (25 - 17) mod 26 =	8	I
(T - K) mod 26	= (19 - 10) mod 26 =	9	J
(F- B) mod 26	= (5 - 1) mod 26 =	4	E
(Z- I) mod 26	= (25 - 8) mod 26 =	1 7	R
(K- R) mod 26	= (11 - 17) mod 26 =	2 0	U
(U - I) mod 26	= (20 - 10) mod 26 =	1 0	K

Note : Penjabaran pengurangan baris ke-1 (tabel 2) => ((N-R) mod 26) = (13-17=(-4)), jika minus maka hasil pengurangan ditambah dengan nilai mod agar dekripsinya berhasil, (-4)+26=22.

Linier Congruential Generator (LCG)

Bilangan acak adalah bilangan yang tidak dapat diprediksi kemunculannya. Tidak ada komputasi yang benar-benar menghasilkan deret bilangan acak secara sempurna. Banyak algoritma atau metode yang dapat digunakan untuk membangkitkan bilangan acak salah satunya adalah pembangkit bilangan acak *Linear Congruential Generators* [3]. LCG adalah algoritma yang sering diimplementasikan pada beberapa bahasa pemrograman untuk membangkitkan bilangan acak. LCG didefinisikan dalam relasi rekurens:

$X_n = (aX_n\text{-i} + b) \bmod m$

Contoh:

Diketahui => a = 5 , b = 3 , m = 26 , & X_n-i = 8

Untuk mengetahui hasilnya dapat dilihat seperti berikut:

Tabel 3. Contoh Proses Pembangkit Kunci

Rumus	Nilai	Hasil	Char
$(aX_n\text{-i} + b) \bmod m$	= (5.8+3) mod 26 =	17	R
$(aX_n\text{-i} + b) \bmod m$	= (5.17+3) mod 26 =	10	K
$(aX_n\text{-i} + b) \bmod m$	= (5.10+3) mod 26 =	1	B
$(aX_n\text{-i} + b) \bmod m$	= (5.1+3) mod 26 =	8	I
$(aX_n\text{-i} + b) \bmod m$	= (5.8+3) mod 26 =	17	R
$(aX_n\text{-i} + b) \bmod m$	= (5.17+3) mod 26 =	10	K
$(aX_n\text{-i} + b) \bmod m$	= (5.10+3) mod 26 =	1	B
$(aX_n\text{-i} + b) \bmod m$	= (5.1+3) mod 26 =	8	I
$(aX_n\text{-i} + b) \bmod m$	= (5.8+3) mod 26 =	17	R
$(aX_n\text{-i} + b) \bmod m$	= (5.17+3) mod 26 =	10	K

X_n = bilangan acak ke-n dari deretnya X_{n-1}= bilangan acak sebelumnya a = factor pengali b = penambah *(increment)* m = modulus (a , b , dan m semuanya konstans).

Kunci pembangki adalah X_0 yang disebut *seed (secret seed)*. Dalam hal ini X_0 bersifat rahasia. LCG mempunyai periode tidak lebih besar dari m, dan pada kebanyakan kasus periodenya kurang dari itu. Periode penuh (m-1) kepunyaan LCG dapat terjadi jika memenuhi syarat sebagai berikut: [6]

1. b relative prima terhadap m
2. a-1 dapat dibagi dengan semua faktor prima dari m
3. a-1 adalah kelipatan 4 jika m adalah kelipatan 4
4. m >maks (a, b, X_0)
5. a > 0 , b > 0

Analisis Algoritma

Analisis merupakan aktivitas yang terjadi untuk menjelaskan langkah-langkah berkaitan dengan enkripsi dan dekripsi menggunakan algoritma *One Time Pad* dan penerapan metode *Linier Congruential Generator* untuk membangkitkan suatu kunci. Bagian ini terdiri dari *key generator*, enkripsi dan dekripsi.

https://doi.org/10.30743/infotekjar.v4i1.1590

Key Generator

Analisis pembangkitan kunci pada One Time Pad menggunakan nilai indeks karakter yang dioperasikan dengan kunci dan modulus dari jumlah karakter yang digunakan. Pesan Teks yang akan dienkripsi menggunakan karakter huruf besar, huruf kecil, angka, dan tanda baca yang dapat dilihat pada tabel berikut:

Tabel 4. Daftar Karakter Enkripsi & Deskripsi [6]

Karakter	Kode	Karakter	Kode
Space	0	P	48
!	1	Q	49
"	2	R	50
#	3	S	51
$	4	T	52
%	5	U	53
&	6	V	54
'	7	W	55
(	8	X	56
)	9	Y	57
*	10	Z	58
+	11	[	59
,	12	\	60
-	13	]	61
.	14	^	62
/	15	_	63
0	16	`	64
1	17	a	65
2	18	b	66
3	19	c	67
4	20	d	68
5	21	e	69
6	22	f	70
7	23	g	71
8	24	h	72
9	25	i	73
:	26	j	74
;	27	k	75
<	28	l	76
=	29	m	77
>	30	n	78
?	31	o	79
@	32	p	80
A	33	q	81
B	34	r	82
C	35	s	83
D	36	t	84
E	37	u	85
F	38	v	86
G	39	w	87
H	40	x	88
I	41	y	89
J	42	z	90
K	43	{	91
L	44	|	92
M	45	}	93
N	46	~	94
O	47		

Berikutnya, proses pembangkitan kunci OTP menggunakan LCG, dapat dilakukan dengan cara menyepakati parameter yang digunakan untuk mengirim dan menerima pesan teks. Contoh pesan yang akan dikirimkan adalah "Saya Rachmat".

Nilai parameter M (modulus) yang digunakan adalah 95. Karena berdasarkan tabel 4 di atas, kode dimulai dari nilai nol (0).
Berikut adalah nilai-nilai awal parameter yang digunakan untuk menghasilkan nilai LCG-nya:

Formula => X_n = (aX_(n-1) + b) mod M , dimana
X_0 = 34
a = 12
b = 22
Proses nilai LCG untuk X_1:
X_n=1 = (12 * 34) + 22 mod 95
Hasilnya adalah 50
Selanjutnya, lakukan proses nilai LCG-nya s.d 18 kali proses, yang hasil keseluruhannya seperti berikut:

Tabel 5. Nilai Acak LCG

X(i)	LCG(i)
1	50
2	52
3	76
4	79
5	20
6	72
7	31
8	14
9	0

X(i)	LCG(i)
10	22
11	1
12	34
13	50
14	52
15	76
16	79
17	20
18	72

Enkripsi

Kegiatan yang mentransformasikan pesan asli menjadi pesan acak (tidak beraturan) sehingga pesan tidak dimengerti atau dipahami maknanya [8]. Metode OTP mengenkripsi pesan menggunakan key generator yang terdapat pada metode LCG yang telah dijabarkan sebelumnya. Adapun *plaintext* yang digunakan pada proses enkripsi adalah "Saya Rachmat".
Tahapan enkripsi selangkapnya dijelaskan sebagai berikut:

1. Konversikan pesan teks yang akan diubah ke nilai karakter, seperti tampilan berikut:

Tabel 6. Konversi *Plaintext* ke Nilai Karakter

S	**a**	**y**	**a**	**(spasi)**	**R**	**a**	**c**	**h**	**m**
51	65	89	65	0	50	65	67	72	77
a	**t**								
65	84								

2. Enkripsikan pesan teks menggunakan hasil dari nilai acak yang berasal dari metode LCG
   Formula => C = (Pi + Ki) mod 95
   Karena *plaintext* ("Saya Rachmat") yang akan dienkripsi berjumlah dua belas (12) karakter, maka nilai acak LCG yang digunakan cukup sampai dua belas juga

Tabel 7. Enkripsi *Plaintext*

X(i)	LCG(i)
1	6
2	22
3	70
4	49
5	20
6	27

X(i)	LCG(i)
7	1
8	81
9	72
10	4
11	66
12	23

3. Konversi nilai enkripsi *plaintext* menjadi karakter *ciphertext*

Tabel 8. Konversi Nilai Enkripsi *Plaintext* ke *Ciphertext*

6	**22**	**70**	**49**	**20**	**27**	**1**	**81**	**72**	**4**
&	**6**	**f**	**Q**	**4**	**;**	**!**	**q**	**h**	**$**
66	**23**								
b	**7**								

Hasil *ciphertext*-nya adalah:

& 6 f Q 4 ; ! q h $ b 7

Dekripsi

Teknik yang mengembalikan *ciphertext* ke dalam bentuk aslinya (*plaintext*) [9]. Algoritma OTP menggunakan key generator yang dihasilkan dari metode LCG untuk melakukan proses dekripsi. Kunci yang digunakan adalah kunci yang dihasilkan pada saat mengerjakan proses enkripsi.

Tahapan dekripsi selangkapnya dijelaskan sebagai berikut:

1. Kembalikan karakter *ciphertext* ke dalam nilai hasil enkripsi *plaintext*

Tabel 9. Ubah *Ciphertext* Menjadi Nilai Enkripsi *Plaintext*

&	6	f	Q	4	;	!	q	h	$
6	22	70	49	20	27	1	81	72	4
b	7								
66	23								

2. Lakukan proses dekripsi menggunakan nilai acak yang berasal dari metode LCG
   Formula => P = (Ci - Ki) mod 95

Tabel 10. Dekripsi *Ciphertext*

P(i)	Ci	Ki	(Ci – Ki) mod 95
1	6	50	51
2	22	52	65
3	70	76	89
4	49	79	65
5	20	20	0
6	27	72	50
7	1	31	65
8	81	14	67
9	72	0	72
10	4	22	77
11	66	1	65
12	23	34	84

3. Konversikan nilai hasil dekripsi ke *plaintext*

Tabel 11. Konversi Hasil Dekripsi ke *Plaintext*

51	65	89	65	0	50	65	67	72	77
S	a	y	a	space	R	a	c	h	m
65	84								
a	t								

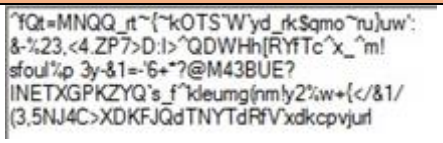
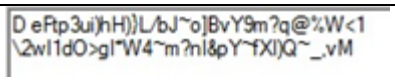
Kesimpulan

Pengujian algoritma *On Time Pad* dan *Linier Congruential Generator* untuk mengamankan pesan teks, tidak hanya dilakukan pada contoh kasus yang telah dijabarkan di atas. Namun, percobaan telah dilakukan pada beberapan *plaintext* dengan pembangkit kunci yang berbeda juga, seperti yang tampak pada tampilan berikut:

Tabel 12. *Plaintext testing*

No	Pesan Teks	LCG
1	17 Agutus 11945 adalah hari keerdekaan indonesia, dimana para pejuang dan pahlawan dulu merebut dan mempertahankan indonesia dari para penjajah yang ingin menguasai indosenia	1945
2	gempa di aceh pukul 07.35, telah memakan korban sebanyak 35 orang	60
3	Insiden yang terjadi di Masjid Al Aqsa, Palestina pada Jumat 21 Juli 2017 telah menimbulkan keprihatinan banyak pihak, tak terkecuali Pemerintah Indonesia.! Karena itu, Duta Besar Republik Indonesia untuk Mesir, Helmy Fauzi mendorong Liga @rab untuk menyatukan langkah menyikapi aksi brutal *Israel*.(Helmy Fauzi) melalui nota diplomatiknya kepada Sekjen Liga Arab menyatakan, Pemerintah Indonesia mendukung semua upaya positif yang dilakukan Liga Arab untuk menghentikan kesewenang-wenangan Israel terhadap warga muslim Palestina yang hendak melakukan ibadah di <Masjid> [Al-Aqsa].	189
4	Sistem ekonomi makro suatu negara dapat disimulasikan sebagai model persamaan linear variabel keadaan waktu diskret : x(k + 1) = Ax(k) + Bu(k) dan y(k) = Cx(k) + Du(k).Dimana variabel keadaan (state variable) x(k) pada tahun ke k adalah : belanja konsumtif dan investasi bisnis swasta. Masukan (input) u(k) adalah : pajak dan belanja negara, sedangkan keluaran (output) y(k) adalah : pendapatan nasional.	212

Tabel 13. Hasil Enkripsi *Plaintext Testing*

No	Ciphertext
1	
2	
3	
4	

Berdasarkan hasil pengujian *plaintext* di atas, didapatlah beberapa informasi:

1. Tingkat keamanan pesan dapat dikatakan cukup karena menggunakan dua metode: pertama, mengenkripsi pesan teksnya, dan kedua, menghasilkan pembangkit kunci enkripsinya untuk mengamankan pesan teks sebelum dikirim ke target.
2. Pesan yang telah terenkripsi tidak dapat dikembalikan ke dalam bentuk aslinya, jika pembangkit kuncinya salah atau tidak tepat.
3. Proses enkripsi mengkonsumsi waktu yang lama, jika pesan yang dimasukkan mengkonsumsi memori lebih dari 7.5 KB.

https://doi.org/10.30743/infotekjar.v4i1.1590

4. Proses enkripsi berjalan lumayan lama, jika spesifikasi komputer yang digunakan prosesornya dibawah core I dan RAM-nya kecil.
5. Karakter teks yang digunakan tidak seperti yang terdapat dalam keyboard, namun dapat dikatakan cukup banyak karena melibatkan karakter "A – Z" (besar dan kecil), interger (0-9) dan beberapa karakter tertentu seperti yang tampak pada tabel 4 di atas.
6. Penerapan kedua metode ini OTP dan LCG untuk keamanan pesan teks hanya sebatas desktop saja. Kedepannya diharapan kedua metode ini, dapat diterapkan pada *mobile* dan web.

REFERENCES

[1] R. Aulia, "Pemanfaatan Website Sebagai Sarana Managing Data Dalam Suatu Organisasi (Studi Kasus: Pertemuan Ilmiah Nasional (Pin) Perhimpunan Dokter Spesialis Saraf Indonesia (Perdossi) 2013 Medan)," *InfoTekJar (Jurnal Nasional Informatika dan Teknologi Jaringan)*, vol. 1, no. 1, pp. 1–6, 2016.

[2] S. Sitinjak, Y. Fauziah, and Juwairiah, "C-78 Aplikasi Kriptografi File Menggunakan Algoritma Blowfish," *SemnasIF (Seminar Nasional Informatika)*, UPN "Veteran" Yogyakarta, 2010.

[3] M. Manssen, M. Weigel, and A. K. Hartmann, "Random number generators for massively parallel simulations on GPU," *Eur. Phys. J. Spec. Top.*, vol. 210, no. 1, pp. 53–71, 2012.

[4] R. Aulia, A. Zakir, and D. A. Purwanto, "Penerapan Kombinasi Algoritma Base64 Dan Rot47 Untuk Enkripsi Database Pasien Rumah Sakit Jiwa Prof. Dr. Muhammad Ildrem," *InfoTekJar (Jurnal Nasional Informatika dan Teknologi Jaringan)*, vol. 2, no. 2, pp. 146–151, 2018.

[5] M. Stamp, *Information Security Principles and Practice*, 2nd ed. Canada: Wiley, 2011.

[6] M. K. Harahap and R. Rina, "Kombinasi Kriptografi RSA dengan Linear Congruential Generator," *SinkrOn (Jurnal & Penelitian Teknik Informatika)*, vol. 3, no. 1, p. 267, 2018.

[7] "Ascii Table and Description." [Online]. Available: http://www.asciitable.com/. [Accessed: 10-Aug-2019].

[8] R. Aulia, A. Sembiring, A. Zakir, and B. A. U. Siregar, "PENYANDIAN TEXTS CHAT VIA INTERNET DENGAN ALGORITMA VIGENERE CIPHER," *JSIK (Jurnal Sistem Informasi Kaputama)*, vol. 3, no. 2, pp. 28–34, 2019.

[9] I. Gunawan, "Kombinasi Algoritma Caesar Cipher dan Algoritma RSA untuk pengamanan File Dokumen dan Pesan Teks," *InfoTekJar (Jurnal Nasional Informatika dan Teknologi Jaringan)*, vol. 2, no. 2, pp. 124–129, 2018.