

Pengamanan Pesan Teks Menggunakan Affine Cipher dan Algoritma Goldbach Code

Rafika Zahrotul Fauziah¹, Muhammad Khudzaifah², Erna Herawati³

^{1,2,3}Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim
Malang, Indonesia

Email: ¹rafikazahrotul503@gmail.com, ²khudzaifah@uin-malang.ac.id, ³faridatul_mahya@uin-malang.ac.id

Abstrak

Perkembangan informasi saat ini memiliki dampak negatif juga positif. Untuk menanggulangi adanya dampak negatif, diperlukan adanya kriptografi. Hal ini bertujuan agar keamanan pesan teks terjaga. Affine Cipher merupakan salah satu algoritma kriptografi simetris yang menggunakan (m) sebagai kunci multiplikatif dan (b) sebagai jumlah pergeseran pada saat proses enkripsi. Pada saat proses dekripsi menggunakan balikan dari kunci multiplikatif (m^{-1}). Goldbach Code diasumsikan bahwa setiap bilangan bulat genap yang lebih besar dari empat merupakan penjumlahan dari dua bilangan prima. Penelitian ini memiliki dua tahapan, yaitu: proses enkripsi dan dekripsi. Pada penelitian ini, proses yang akan dilakukan bertujuan untuk mengamankan pesan teks yang diawali dengan mengenkripsikannya menggunakan Affine Cipher dan setelahnya akan dikompresi dengan menggunakan Algoritma Goldbach Code untuk menghasilkan cipherteks. Kemudian untuk mengembalikan pesan akan didekompresi dengan menggunakan Algoritma Goldbach Code dan dekripsi dengan menggunakan Affine Cipher. Dengan menggabungkan Affine Cipher dan Algoritma Goldbach code hasil dari proses pengamanan pesan teks akan lebih aman dikarenakan cipherteks yang dihasilkan memiliki panjang bit yang berbeda.

Kata kunci: *Affine Cipher; Goldbach Code; Algoritma Kompresi; Kriptografi*

Text Message Security Using Affine Cipher and Goldbach Code Algorithm

Abstract

The development of information today has both negative and positive impacts. To overcome the negative impact, cryptography is needed. This aims to maintain the security of text messages. Affine Cipher is a symmetric cryptography algorithm that uses (m) as the multiplicative key and (b) as the number of shifts during the encryption process. The decryption process uses the reciprocal of the multiplicative key (m^{-1}). Goldbach Code assumes that every even integer greater than four is the sum of two prime numbers. This research has two stages, namely: encryption and decryption process. In this research, the process that will be carried out aims to secure text messages that begin by encrypting them using Affine Cipher and afterwards will be compressed using the Goldbach Code Algorithm to produce ciphertext. Then to restore the message, it will be decompressed using the Goldbach Code Algorithm and decrypted using Affine Cipher. By combining Affine Cipher and Goldbach Code Algorithm, the result of the text message security process will be more secure because the resulting ciphertext has a different bit length.

Keywords: *Affine Cipher; Goldbach Code; Compression Algorithm; Cryptography*

1. PENDAHULUAN

Perkembangan teknologi informasi saat ini berdampak besar pada semua aspek kehidupan masyarakat, termasuk komunikasi. Dengan adanya teknologi internet dapat membantu manusia dalam berkomunikasi dengan cepat dan mudah. Dengan kemudahan ini, kebutuhan terhadap teknologi informasi dan komunikasi merupakan hal yang sangatlah penting. Pertukaran informasi dapat menimbulkan dampak positif maupun negatif. Sehingga solusi yang dapat diambil untuk mengatasi permasalahan tersebut adalah dengan kriptografi. Kriptografi sendiri bertujuan untuk menyembunyikan suatu informasi dari individu yang tidak berwenang dengan perubahan pesan asli menjadi pesan yang

tidak bermakna. Enkripsi dan dekripsi adalah dua proses dari kriptografi. Enkripsi adalah proses memodifikasi informasi yang sebelumnya dapat dibaca dengan mudah kemudian diubah dengan menggunakan suatu algoritma sehingga tidak dapat dibaca oleh siapapun. Sementara dekripsi adalah teknik membuat data yang tidak dapat dibaca menjadi bentuk yang dapat dibaca, dekripsi adalah kebalikan dari enkripsi

2. TINJAUAN PUSTAKA

Kriptografi ada 3 macam yaitu, kriptografi simetris, kriptografi asimetris, dan *hybrid*. Algoritma yang dikenal sebagai kriptografi simetris menggunakan kunci yang sama untuk enkripsi dan dekripsi. Dalam kriptografi simetris, pengirim dan penerima harus berbagi kunci yang sama sebelum

mengirim dan menerima pesan agar penerima dapat mendekripsi pesan yang diterimanya. Caesar Cipher, Vigenere Cipher, Affine Cipher merupakan contoh dari kriptografi simetris. Sedangkan dalam kriptografi asimetris, kunci yang berbeda digunakan untuk enkripsi dan dekripsi. Teknik kriptografi ini antara lain RSA, Elgamal, Diffie-Hellman, dan lain-lain. Penggabungan dari kriptografi simetris dan asimetris dikenal sebagai kriptografi *hybrid*.

Affine cipher merupakan sandi substitusi yang mana setiap huruf dalam alfabet diubah menjadi numerik yang dienkripsi dengan persamaan aritmatika sederhana dan diubah kembali menjadi huruf tersebut. Affine cipher dapat dengan mudah membuat sistem terasa aman dengan mengalikan setiap nilai teks biasa dengan angka berbeda dan kemudian menambahkan dengan nilai pergeseran. Metode affine cipher merupakan perluasan dari metode Caesar cipher, yang mana metodenya adalah dengan mengalikan plainteks (P) dengan kunci multiplikatif (m) dan menambah dengan sebuah pergeseran (b). Sehingga P menghasilkan cipherteks C yang dinyatakan dengan fungsi kongruen. Pada penelitian kali ini proses enkripsi pada Affine Cipher akan termodifikasi modulonya sehingga cipherteks yang akan dihasilkan merupakan ASCII *printable*. Sehingga untuk perhitungan enkripsi dan dekripsinya sebagai berikut:

$$C \equiv (m(P - 32) + b) \pmod{95} + 32$$

$$P \equiv (m^{-1}(C - 32 - b) \pmod{95}) + 32$$

Sedangkan goldbach code adalah algoritma yang diasumsikan menggunakan teori golbach conjecture. Teori golbach conjecture itu sendiri yaitu semua bilangan genap positif yang lebih besar dari 2 merupakan penjumlahan dari dua bilangan prima. Goldbach code sendiri memiliki tiga code, yang perama dapat dinamakan "G0" yang mengkodekan bilangan bulat positif n yang diubah menjadi bilangan bulat positif genap dengan $2(n+3)$ dan dituliskan pasangan penjumlahan bilangan prima dalam keadaan terbalik.

3. METODOLOGI

Penelitian ini memiliki 2 tahapan, yaitu:

3.1. Proses enkripsi :

- a. Membentuk kombinasi Algoritma enkripsi dan kompresi.
- b. Melakukan simulasi enkripsi dan kompresi.
 - 1) Menentukan pesan (plainteks).
 - 2) Mengkonversi plainteks menjadi bilangan desimal.
 - 3) Menentukan m dan b .
 - 4) Melakukan perhitungan dengan menggunakan Affine Cipher yang telah dimodifikasi.

$$C \equiv (m(P - 32) + b) \pmod{95} + 32$$

- 5) Mengkonversi hasil cipherteks dari Affine dengan menggunakan Algoritma Goldbach Code.
- 6) Mengubah hasil dari Algoritma Goldbach Code ke bentuk ASCII.
- 7) Mendapatkan pesan acak (cipherteks)

3.2. Proses dekripsi :

- a. Membentuk algoritma dekripsi dan dekompresi.
- b. Melakukan simulasi dekripsi dan dekompresi.
 - 1) Dekompresi pesan acak (cipherteks) dengan menggunakan Algoritma Goldbach Code menjadi bilangan biner.
 - 2) Mengubah bilangan biner dengan menggunakan Algoritma Goldbach Code menjadi cipherteks biasa.
 - 3) Menentukan nilai dari m^{-1} .
 - 4) Mendekripsi dengan menggunakan Affine Cipher yang telah dimodifikasi dengan nilai b yang sama.

Proses $P \equiv (m^{-1}(C - 32 - b) \pmod{95}) + 32$
Mendapatkan pesan asli (plainteks).

4. PEMBAHASAN

Pada penelitian ini, proses yang akan dilakukan bertujuan untuk mengamankan pesan teks yang diawali dengan menenkripsikannya menggunakan Affine Cipher dan setelahnya akan dikompresi dengan menggunakan Algoritma Goldbach Code pada cipherteks yang dihasilkan saat proses enkripsi sehingga hasil dari cipherteks tersebut berbeda panjang bitnya. Kemudian didekompresi dengan menggunakan Algoritma Goldbach Code dan dekripsi dengan menggunakan Affine Cipher. Sehingga proses yang akan terjadi antara lain:

4.1. Simulasi Enkripsi Affine Cipher

Proses enkripsi yang akan dilakukan sebagai berikut:

1. Menentukan plainteks atau pesan asli. Pada penelitian kali ini, peneliti memilih plainteks "FORGIVE AND FORGET".
2. Mengonversi plainteks ke dalam bentuk desimal.

Tabel 1 Konversi Karakter Plainteks menjadi Kode ASCII (Desimal)

Plainteks	Kode ASCII (desimal)
F	70
O	79
R	82
G	71
I	73
V	86
E	69
Space	32

Plainteks	Kode ASCII (desimal)
A	65
N	78
D	68
Space	32
F	70
O	79
R	82
G	71
E	69
T	84

- Menentukan kunci multiplikatif (m) dan pergeseran (b). Peneliti memilih $m = 71$ dikarenakan $GCD(71,95) = 1$ dan $b = 30$.
- Melakukan perhitungan dengan menggunakan Affine Cipher dengan rumus yang digunakan mengalami modifikasi berupa adanya penambahan dan juga pengurangan 32. Hal ini dilakukan dengan tujuan untuk membuat karakter yang akan muncul berada pada rentang 32-126 sesuai dengan kode ASCII printable. Dan menggunakan modulo 95 untuk menyesuaikan dengan jumlah karakter ASCII printable.

$$C \equiv (m(P - 32) + b)(mod\ 95) + 32$$

Tabel 2 Enkripsi Menggunakan Affine Cipher

Indeks Plainteks	Enkripsi Affine Cipher	Indeks Cipherteks	Cipher teks
70	$(71(70 - 32) + 30)(mod\ 95) + 32$ $= (71(38) + 30)(mod\ 95) + 32$ $= (2698 + 30)(mod\ 95) + 32$ $= 2728(mod\ 95) + 32$ $= 68 + 32$ $= 100$	100	d
79	$(71(79 - 32) + 30)(mod\ 95) + 32$	74	J
82	$(71(82 - 32) + 30)(mod\ 95) + 32$	97	a
71	$(71(71 - 32) + 30)(mod\ 95) + 32$	76	L
73	$(71(73 - 32) + 30)(mod\ 95) + 32$	123	{
86	$(71(86 - 32) + 30)(mod\ 95) + 32$	96	,
69	$(71(69 - 32) + 30)(mod\ 95) + 32$	124	
32	$(71(32 - 32) + 30)(mod\ 95) + 32$	62	>
65	$(71(65 - 32) + 30)(mod\ 95) + 32$	125	}
78	$(71(78 - 32) + 30)(mod\ 95) + 32$	98	b
68	$(71(68 - 32) + 30)(mod\ 95) + 32$	53	S

Indeks Plainteks	Enkripsi Affine Cipher	Indeks Cipherteks	Cipher teks
32	$(71(32 - 32) + 30)(mod\ 95) + 32$	62	>
70	$(71(70 - 32) + 30)(mod\ 95) + 32$	100	d
79	$(71(79 - 32) + 30)(mod\ 95) + 32$	74	J
82	$(71(82 - 32) + 30)(mod\ 95) + 32$	97	a
71	$(71(71 - 32) + 30)(mod\ 95) + 32$	76	L
69	$(71(69 - 32) + 30)(mod\ 95) + 32$	124	
84	$(71(84 - 32) + 30)(mod\ 95) + 32$	49	1

Cipherteks yang dihasilkan pada proses enkripsi menggunakan Affine Cipher yaitu "dJaL{>}bS>dJaL|1"

4.2. Simulasi Kompresi Algoritma Goldbach Code

Setelah hasil cipherteks telah diketahui maka akan dilakukan proses kompresi dengan menggunakan Algoritma Goldbach Code. Sehingga proses yang dilakukan sebagai berikut:

- Melakukan kompresi menggunakan Algoritma Goldbach Code.

Tabel 3 Proses Kompresi Cipherteks

Cipherteks	Frek	N	2(N+3)	Prima	Code
d	2	1	8	$3 \cdot 1 + 5$ $\cdot 1$	11
J	2	2	10	$3 \cdot 1 + 5$ $\cdot 0 + 7$ $\cdot 1$	101
a	2	3	12	$3 \cdot 0 + 5$ $\cdot 1 + 7$ $\cdot 1$	011
L	2	4	14	$3 \cdot 1 + 5$ $\cdot 0 + 7$ $\cdot 0 + 11$ $\cdot 1$	1001
	2	5	16	$3 \cdot 0 + 5$ $\cdot 1 + 7$ $\cdot 0 + 11$ $\cdot 1$	0101
>	2	6	18	$3 \cdot 0 + 5$ $\cdot 0 + 7$ $\cdot 1 + 11$ $\cdot 1$	0011
{	1	7	20	$3 \cdot 0 + 5$ $\cdot 0 + 7$ $\cdot 1 + 11$ $\cdot 0 + 13$ $\cdot 1$	00101
,	1	8	22	$3 \cdot 0 + 5$ $\cdot 1 + 7$ $\cdot 0 + 11$ $\cdot 0 + 13$ $\cdot 0 + 17$ $\cdot 1$	010001

Cipherteks	Frek	N	2(N+3)	Prima	Code
}	1	9	24	$3 \cdot 0 + 5$ $\cdot 0 + 7$ $\cdot 0 + 11$ $\cdot 1 + 13$ $\cdot 1$	00011
b	1	10	26	$3 \cdot 0 + 5$ $\cdot 0 + 7$ $\cdot 1 + 11$ $\cdot 0 + 13$ $\cdot 0 + 17$ $\cdot 0 + 19$ $\cdot 1$	001000 1
s	1	11	28	$3 \cdot 0 + 5$ $\cdot 0 + 7$ $\cdot 0 + 11$ $\cdot 1 + 13$ $\cdot 0 + 17$ $\cdot 1$	000101
l	1	12	30	$3 \cdot 0 + 5$ $\cdot 0 + 7$ $\cdot 0 + 11$ $\cdot 0 + 13$ $\cdot 1 + 17$ $\cdot 1$	000011

1. Berdasarkan pada tabel diatas, cipherteks “dJaL{>}bS>dJaL|1” berubah menjadi “11 101 011 1001 00101 010001 0101 0011 00011 0010001 000101 0011 11 101 011 1001 0101 000011”

2. Mengubah hasil dari Algoritma Goldbach Code menjadi bentuk ASCII

Dari bit-bit yang dihasilkan akan dipecah menjadi 8-bit dan akan diubah menjadi karakter ASCII sehingga akan menghasilkan bentuk yang berbeda seperti yang terdapat pada tabel 4:

Tabel 4 Hasil Kompresi dengan Algoritma Goldbach Code

Biner	Desimal	Karakter
11101011	235	Û
10010010	146	Æ
10100010	162	ó
10100110	166	a
00110010	50	2
00100010	34	"
10011111	159	f
01011100	92	\
10101000	168	ı
011	3	ETX

Sehingga cipherteks yang dihasilkan dengan menggunakan Affine Cipher yaitu “dJaL{>}bS>dJaL|1” dengan panjang bit sebanyak 144 bit dan setelah dilakukan kompresi dengan menggunakan Algoritma Goldbach Code menjadi “ÛÆó2" fıETX” dengan panjang 80 bit.

4.3. Simulasi Dekompresi Algoritma Goldbach Code

Langkah-langkah yang dilakukan sebagai berikut:

1. Mendekomposisi cipherteks dengan menggunakan Algoritma Goldbach Code menjadi sebuah bilangan biner.

Cipherteks : ÛÆó2" fıETX dengan menggunakan tabel ASCII maka didapatkan hasil bilangan biner seperti tabel 5:

Tabel 5 Mengubah Cipherteks Menjadi Bilangan Biner

Karakter	Desimal	Biner
Û	235	11101011
Æ	146	10010010
ó	162	10100010
a	166	10100110
2	50	00110010
"	34	00100010
f	159	10011111
\	92	01011100
ı	168	10101000
ETX	3	011

2. Sehingga bilangan biner yang diperoleh adalah “11101011 10010010 10100010 10100110 00110010 00100010 10011111 01011100 10101000 011”

3. Dari bilangan biner yang dihasilkan dapat kita pisah dengan berhenti untuk “1” yang kedua sehingga menghasilkan: “11 101 011 1001 00101 010001 0101 0011 00011 0010001 000101 0011 11 101 011 1001 0101 000011” dengan melihat pada Tabel 3 bilangan biner akan menjadi “dJaL{>}bS>dJaL|1”.

4.4. Simulasi Dekripsi Affine Cipher

Dari hasil dekomposisi tersebut cipherteks dikompresi menjadi bilangan desimal sehingga menghasilkan seperti pada tabel 6:

Tabel 6 Mengubah Cipherteks menjadi Desimal

Cipherteks	Desimal
d	100
J	74
a	97
L	76
{	123
`	96
	124
>	62
}	125
b	98
S	83
>	62
d	100
J	74
a	97
L	76
	124
l	49

Sebelum melakukan dekripsi menggunakan Affine Cipher, yang harus dilakukan yaitu mencari nilai dari m^{-1} dari $m = 71$. Sehingga

$$91 \cdot 71x \equiv 91 \cdot 1 \pmod{95}$$

$$x \equiv 91 \pmod{95}$$

Sehingga didapatkan $71 \cdot 91 \equiv 1 \pmod{95}$ maka $m^{-1} = 91$.

Dengan menggunakan nilai $m^{-1} = 91$ dan $b = 30$ dapat didekripsi dengan menggunakan Affine Cipher. Sehingga

$$P \equiv (91(C - 32 - 30) \pmod{95}) + 32$$

Tabel 7 Dekripsi dengan Menggunakan Affine Cipher

Indek Cipher teks	Enkripsi Affine Cipher	Indeks Plain teks	Plain teks
100	$(91(100 - 32 - 30) \pmod{95}) + 32$ $= (91(38) \pmod{95}) + 32$ $= (3458 \pmod{95}) + 32$ $= 38 + 32$ $= 70$	70	F
74	$(91(74 - 32 - 30) \pmod{95}) + 32$	79	O
97	$(91(97 - 32 - 30) \pmod{95}) + 32$	82	R
76	$(91(76 - 32 - 30) \pmod{95}) + 32$	71	G
123	$(91(123 - 32 - 30) \pmod{95}) + 32$	73	I
96	$(91(96 - 32 - 30) \pmod{95}) + 32$	86	V
124	$(91(124 - 32 - 30) \pmod{95}) + 32$	69	E
62	$(91(62 - 32 - 30) \pmod{95}) + 32$	32	Space
125	$(91(125 - 32 - 30) \pmod{95}) + 32$	65	A
98	$(91(98 - 32 - 30) \pmod{95}) + 32$	78	N
53	$(91(53 - 32 - 30) \pmod{95}) + 32$	68	D
62	$(91(62 - 32 - 30) \pmod{95}) + 32$	32	Space
100	$(91(100 - 32 - 30) \pmod{95}) + 32$	70	F
74	$(91(74 - 32 - 30) \pmod{95}) + 32$	79	O
97	$(91(97 - 32 - 30) \pmod{95}) + 32$	82	R
76	$(91(76 - 32 - 30) \pmod{95}) + 32$	71	G
124	$(91(124 - 32 - 30) \pmod{95}) + 32$	69	E
49	$(91(49 - 32 - 30) \pmod{95}) + 32$	84	T

Sehingga hasil dari dekompresi dan dekripsi pesan menggunakan Algoritma Goldbach Code dan Affine Cipher yaitu **"FORGIVE AND FORGET"**.

5. KESIMPULAN

Berdasarkan pembahasan di atas, dapat ditarik kesimpulan bahwa proses enkripsi dengan menggunakan Affine Cipher dan kompresi pesan menggunakan Algoritma Goldbach Code diawali dengan menentukan nilai dari kunci multiplikatif (m) = 71 dan pergeseran (b) = 30. Kemudian memasukkan indeks plainteks dan melakukan perhitungan menggunakan Affine Cipher yang telah dimodifikasi $C \equiv (m(P - 32) + b) \pmod{95} + 32$. Setelah hasil enkripsi menggunakan Affine Cipher

diketahui, cipherteks akan dikompresi dengan menggunakan Algoritma Goldbach Code. Cipherteks tersebut awalnya akan dikompresi dengan goldbach code sehingga menghasilkan bilangan biner. Selanjutnya bilangan biner akan dikelompokkan menjadi 8 bit sehingga dari bilangan biner yang berjumlah 8 bit tersebut dapat dikonversi menjadi karakter ASCII. Sehingga cipherteks yang panjang bit awalnya berjumlah 144 bit akan dikompresi menjadi 80 bit.

Proses dekripsi menggunakan Affine Cipher dan dekompresi menggunakan Algoritma Goldbach Code. Proses ini akan diawali dengan dekompresi cipherteks dengan menggunakan Algoritma Goldbach Code. Pada proses dekompresi ini awalnya akan mengubah cipherteks menjadi bilangan biner. Kemudian bilangan biner akan dipisah sesuai dengan jumlah bit sebelumnya. Setelah dilakukan pemisahan akan dilakukan dekompresi dengan melihat tabel kompresi goldbach code. Kemudian menentukan balikan dari kunci multiplikatif (m^{-1}) = 91. Selanjutnya memasukkan indeks cipherteks dan melakukan perhitungan menggunakan Affine Cipher $P \equiv (91(C - 32 - 30) \pmod{95}) + 32$ sehingga akan menghasilkan plainteks.

DAFTAR PUSTAKA

- Ariyus, D. (2006). *KRIPTOGRAFI KEAMANAN DATA DAN KOMUNIKASI*.
- Ariyus, D. (2008). *Pengantar Ilmu Kriptografi Teori Analisis dan Implementasi*.
- Aznadillah, S., Matematika, P. S., Sains, F., Teknologi, D. A. N., Islam, U., Maulana, N., & Ibrahim, M. (2022). *Modifikasi Affine Cipher Menggunakan Algoritma Blum-Blum Shub Dalam Modifikasi Affine Cipher Menggunakan Algoritma Blum-Blum Shub Dalam*. 1–74.
- Dwi, B. J., Joko Priono, M., Suhendri, A., & Darwis, D. (2018). Implementasi Kombinasi Affine Cipher Dan One-Time Pad Dalam. *Jurnal Informatika*, 18(2), 124–129.
- Kester, Q.-A. (2021). A cryptosystem based on Vigenère cipher with varying key (virtual) View project. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 1(10), 15–17. <https://www.researchgate.net/publication/235618077>
- Khairani, T., Agung, K., & Kamsyakawuni, A. (2021). Pengkodean Monoalphabetic Menggunakan Affine Cipher dengan Kunci Diffie-Hellman. *Prisma (Prosiding Seminar Nasional Matematika)*, 4, 553–559. <https://journal.unnes.ac.id/sju/index.php/prisma> a/ ISSN
- Konheim, A. G. (1986). Computer Security and cryptography. In *IEEE Communications Magazine* (Vol. 23, Issue 7). <https://doi.org/10.1109/MCOM.1985.1092611>

- Maulana, P. A. (2019). Proses Enkripsi & Dekripsi Pada Polinomial Dengan Menggunakan Metode Affine Cipher. *Etheses Uin*, 28.
- Nasution, S. D., Ginting, G. L., Syahrizal, M., & Rahim, R. (2017). Data Security Using Vigenere Cipher and Goldbach Codes Algorithm. *International Journal of Engineering Research & Technology (IJERT)*, 6(01), 360–363.
- Prastya, R., Pardede, A. M. H., & Fauzi, A. (2022). Teknik Pembangkit Kunci Algoritma RSA Menggunakan Algoritma Diffie Hellman pada Keamanan Citra. *KAKIFIKOM (Kumpulan Artikel Karya Ilmiah Fakultas Ilmu Komputer)*, 04(01), 16–22. <https://doi.org/10.54367/kakifikom.v4i1.1872>
- Prihandono, B., & Kusumastuti, N. (2013). Kriptografi Klasik Dengan Metode Modifikasi Affine Cipher Yang Diperkuat dengan vigenere Cipher. *Buletin Ilmiah Mat. Stat. Dan Terapannya (Bimaster)*, 02(2), 87–92.
- Priyono, P. (2016). Penerapan Algoritma Caesar Cipher Dan Algoritma Vigenere Cipher Dalam Pengamanan Pesan Teks. *Jurnal Riset Komputer (JURIKOM)*, 3, Nomor:(Algoritma Caesar Cipher), 351–356.
- Rachmawati, D., & Candra, A. (2015). Implementasi Kombinasi Caesar dan Affine Cipher untuk Keamanan Data Teks. *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, 1(2). <https://doi.org/10.26418/jp.v1i2.12587>
- Studi, P., Informatika, T., Ilmu, F., Universitas, K., & Bengkulu, D. (2014). *Aplikasi Kriptografi Pesan Menggunakan Algoritma*. 10(2), 120–128.
- T. Arroyo, J. C. (2020). Caesar Cipher with Goldbach Code Compression for Efficient Cryptography. *International Journal of Emerging Trends in Engineering Research*, 8(7), 2999–3002. <https://doi.org/10.30534/ijeter/2020/19872020>