COSMIC JURNAL TEKNIK
VOL. 2, NO. 3, AGUSTUS 2025: 106-117
DOI : 10.55537/cosmic
e-ISSN 3063-4687

Analisis Perbandingan Efisiensi Enkripsi Dan Dekripsi Algoritma Sandi Caesar Dan Sandi Vigenere Untuk Keamanan Pesan

Rahmat Abdillah Nuh Siregar[1], Suhardi[2]
[1,2]Ilmu Komputer, Universitas Islam Negeri Sumatera Utara
E-mail: [1]rahmatabdillahnuh@gmail.com, [2]suhardi@uinsu.ac.id
Korespondensi : rahmatabdillahnuh@gmail.com

Abstrak

Tingginya kebutuhan akan perlindungan data pada sistem komunikasi digital menimbulkan permasalahan terkait efektivitas algoritma kriptografi klasik dalam menghadapi ancaman modern, sehingga diperlukan evaluasi empiris terhadap algoritma yang umum digunakan. Penelitian ini bertujuan untuk membandingkan efisiensi dan keamanan dua algoritma kriptografi klasik, yaitu Sandi Caesar dan Sandi Vigenere, guna menentukan tingkat kebermanfaatannya dalam lingkungan komputasi modern serta aplikasi dengan sumber daya terbatas. Metode yang digunakan adalah pendekatan kuantitatif dengan desain eksperimental, di mana kedua algoritma diimplementasikan menggunakan Python dan diuji menggunakan berbagai ukuran teks masukan (100–10.000 byte), kemudian dianalisis berdasarkan waktu eksekusi, penggunaan memori, serta ketahanan terhadap serangan kriptoanalisis. Hasil penelitian menunjukkan bahwa pada ukuran teks 10.000 byte, Sandi Caesar mencatat rata-rata waktu eksekusi sebesar 11,99 ms (enkripsi) dan 11,82 ms (dekripsi), sedangkan Sandi Vigenere mencapai 40,64 ms (enkripsi) dan 45,53 ms (dekripsi). Penggunaan memori pada ukuran teks yang sama juga lebih rendah pada Sandi Caesar (9,86 KB) dibandingkan dengan Sandi Vigenere (9,99 KB). Dari sisi keamanan, simulasi brute force menunjukkan bahwa Sandi Caesar menghasilkan 26 kemungkinan hasil dekripsi (rentan), sedangkan analisis frekuensi Vigenere memberikan distribusi huruf yang lebih acak meskipun metode Kasiski tidak berhasil mengidentifikasi panjang kunci yang benar. Temuan ini menegaskan adanya tradeoff antara efisiensi komputasi dan tingkat keamanan, sehingga pemilihan algoritma harus disesuaikan dengan kebutuhan aplikasi.

Kata kunci: Sandi Caesar; Sandi Vigenere; Efisiensi; Keamanan; Kriptografi

Abstract

The high need for data protection in digital communication systems raises problems related to the effectiveness of classical cryptographic algorithms in dealing with modern threats, so an empirical evaluation of commonly used algorithms is needed. This study aims to compare the efficiency and security of two classical cryptographic algorithms, namely Cipher Caesar and Cipher Vigenere, to determine their usefulness in modern computing environments as well as applications with limited resources. The method used is a quantitative approach with an experimental design, where both algorithms are implemented using Python and tested using various input text sizes (100–10,000 bytes), then analyzed based on execution time, memory usage, and resistance to cryptoanalysis attacks. The results showed that at a text size of 10,000 bytes, Sandi Caesar recorded an average execution time of 11.99 ms (encryption) and 11.82 ms (decryption), while Sandi Vigenere reached 40.64 ms (encryption) and 45.53 ms (decryption). Memory usage at the same text size is also lower in Caesar Cipher (9.86 KB) compared to Vigenere Cipher (9.99 KB). On the security side, the brute force simulation showed that Cipher Caesar generated 26 possible (vulnerable) decryption results, while Vigenere's frequency analysis provided a more random distribution of letters even though Kasiski's method did not manage to identify the correct key length. These findings confirm the tradeoff between computing efficiency and security levels, so the selection of algorithms must be tailored to the needs of the application.

Keywords: Caesar Cipher; Vigenere Cipher; Efficiency; Security; Cryptography

1. PENDAHULUAN

Komunikasi digital telah menjadi inti dari aktivitas modern, mencakup interaksi personal, bisnis, dan pemerintahan, dengan pertukaran data yang cepat dan efisien [1]. Namun, kemudahan ini

e-ISSN 3063-4687

disertai risiko terhadap kerahasiaan, integritas, dan otentisitas informasi, seperti ancaman penyadapan, peretasan, dan modifikasi tanpa izin [2]. Kriptografi, sebagai ilmu untuk menyandikan pesan, menawarkan solusi untuk melindungi data dari pihak yang tidak berwenang. Teknik klasik seperti Sandi Caesar dan Sandi Vigenere membentuk dasar pemahaman prinsip enkripsi dan dekripsi, memberikan wawasan tentang evolusi keamanan informasi [3]. Dengan meningkatnya kebutuhan akan keamanan data pada berbagai platform, memahami algoritma klasik ini tetap relevan untuk aplikasi tertentu dan pendidikan kriptografi [4][5]. Penelitian ini menyoroti Sandi Caesar dan Sandi Vigenere, dua algoritma klasik yang berbeda dalam pendekatan enkripsi. Sandi Caesar menggunakan pergeseran alfabet tetap, menjadikannya sederhana dan cepat namun rentan terhadap serangan karena sifatnya yang monoalfabetik [6][7]. Sebaliknya, Sandi Vigenere memanfaatkan kunci polialfabetik, meningkatkan kompleksitas dan keamanan tetapi membutuhkan sumber daya komputasi lebih besar. Kedua algoritma ini sering digunakan sebagai dasar pembelajaran kriptografi atau dalam sistem dengan keterbatasan perangkat keras, seperti perangkat Internet of Things (IoT) [8]. Penelitian ini berfokus pada perbandingan efisiensi komputasi dan keamanan kedua algoritma untuk mengevaluasi penerapannya dalam lingkup modern [9]. Meskipun Sandi Caesar dan Vigenere telah dipelajari secara luas, analisis sering terbatas pada aspek teoretis atau historis, dengan fokus pada kerentanan matematis seperti analisis frekuensi. Penelitian sebelumnya oleh [7] mengembangkan aplikasi berbasis Java untuk mengenkripsi teks dengan menggabungkan kedua algoritma, tetapi tidak mengukur efisiensi waktu eksekusi dan penggunaan memori secara kuantitatif. Analisis keamanan terhadap serangan seperti analisis frekuensi atau brute force juga tidak dilakukan, dan fokusnya lebih pada kombinasi algoritma daripada perbandingan performa individu. Keterbatasan ini menciptakan kebutuhan untuk penelitian yang mengevaluasi efisiensi dan keamanan secara empiris, terutama dalam lingkungan komputasi digital modern. Berbeda dari penelitian sebelumnya, penelitian ini bertujuan untuk membandingkan efisiensi dan keamanan Sandi Caesar dan Vigenere melalui pendekatan kuantitatif berbasis eksperimen [10][11][12]. Penekanan ditempatkan pada pengukuran waktu eksekusi dan penggunaan memori selama proses enkripsi dan dekripsi, serta ketahanan terhadap serangan kriptoanalisis seperti analisis frekuensi dan brute force [13]. Dengan menilai kedua algoritma secara empiris dan terpisah, penelitian ini memberikan gambaran yang lebih lengkap mengenai performa masing-masing algoritma dan menyoroti relevansi praktisnya dalam aplikasi terbatas maupun pendidikan kriptografi [14][15]. Untuk mencapai hal tersebut, penelitian ini menggunakan desain eksperimental yang dikombinasikan dengan analisis statistik untuk membandingkan performa kedua algoritma [16][17]. Efisiensi ditinjau melalui waktu eksekusi dan penggunaan memori, sedangkan keamanan dievaluasi berdasarkan ketahanan terhadap serangan kriptoanalisis. Pendekatan ini memungkinkan perbandingan yang mendalam untuk menjelaskan keunggulan dan kelemahan masing-masing algoritma, sekaligus menyediakan data empiris yang dapat mendukung pemilihan algoritma pada aplikasi dengan sumber daya terbatas, serta menjadi acuan dalam pengajaran kriptografi. Hasilnya diharapkan dapat memperkuat pemahaman tentang prinsip kriptografi klasik, memberikan dasar untuk menghargai kompleksitas algoritma modern, dan menawarkan wawasan praktis untuk aplikasi terbatas maupun Pendidikan [18][19].

2. METODE PENELITIAN

Penelitian ini menggunakan metode kuantitatif dengan desain eksperimental untuk membandingkan efisiensi komputasi dan keamanan Sandi Caesar dan Sandi Vigenere. Untuk melakukan perbandingan ini secara empiris, kedua algoritma diimplementasikan dari awal menggunakan bahasa pemrograman Python di lingkungan Google Colab. Pendekatan ini memastikan bahwa perbandingan didasarkan pada performa aktual implementasi algoritma, bukan tinjauan dari literatur lain. Variabel independen dalam penelitian ini adalah jenis algoritma kriptografi yang digunakan, yaitu Sandi Caesar dan Sandi Vigenere. Variabel dependen yang diukur untuk perbandingan meliputi Waktu Eksekusi, Penggunaan Memori, dan Ketahanan

e-ISSN 3063-4687

Terhadap Serangan Kriptoanalisis [20][21].

Prosedur pengujian perbandingan dilakukan sebagai berikut:

1. Implementasi Algoritma

   Kedua algoritma (Caesar dengan pergeseran tetap shift=3 dan Vigenere) diimplementasikan dalam Python. Fungsi terpisah dibuat untuk proses enkripsi dan dekripsi masing-masing algoritma.
2. Pembuatan Data Uji

   Teks masukan acak dihasilkan menggunakan fungsi generate_random_text dengan variasi ukuran dari 100 byte, 1000 byte, hingga 10 KB. Untuk Sandi Vigenere, kunci acak dengan panjang bervariasi (5, 10, 15, dan 20 karakter) dihasilkan menggunakan fungsi generate_random_key.
3. Pengukuran Efisiensi

   Untuk setiap ukuran teks dan jenis algoritma (serta variasi panjang kunci untuk Vigenere), waktu eksekusi dan penggunaan memori diukur selama proses enkripsi dan dekripsi. Pengukuran ini dilakukan menggunakan modul time dan tracemalloc di Python. Setiap pengukuran diulang sebanyak 10 kali (iterasi) untuk mendapatkan nilai rata-rata dan standar deviasi guna mengurangi bias dan memastikan keandalan hasil.
4. Evaluasi Keamanan

   Analisis frekuensi dilakukan pada teks terenkripsi dengan ukuran terbesar (10 KB) untuk kedua algoritma menggunakan fungsi frequency_analysis dan bertujuan melihat seberapa merata distribusi huruf pada ciphertext, yang merupakan indikator ketahanan terhadap serangan analisis frekuensi, serangan brute force (Caesar) disimulasikan pada teks terenkripsi Caesar dengan ukuran terbesar untuk menunjukkan kemudahan menemukan kunci (pergeseran) dengan mencoba semua kemungkinan pergeseran (0-25), dan analisis Kasiski (Vigenere) dilakukan pada teks terenkripsi Vigenere dengan ukuran terbesar untuk mencoba mengestimasi panjang kunci, sebagai demonstrasi salah satu teknik kriptoanalisis terhadap Sandi Vigenere.
5. Analisis Data

   Data waktu eksekusi dan penggunaan memori yang terkumpul diorganisasi dalam DataFrame pandas. Analisis statistik deskriptif (mean, standard deviation, min, max, quartile) dilakukan untuk merangkum performa kedua algoritma. Uji t-test independen juga dilakukan untuk membandingkan rata-rata waktu eksekusi kedua algoritma secara statistik. Hasil analisis frekuensi, Kasiski, dan simulasi brute force disajikan untuk mendukung evaluasi keamanan.

COSMIC JURNAL TEKNIK
VOL. 2, NO. 3, AGUSTUS 2025: 106-117
DOI : 10.55537/cosmic
e-ISSN 3063-4687

3. HASIL DAN PEMBAHASAN

3.1 Hasil

1. Waktu Eksekusi

Ringkasan Statistik Waktu Eksekusi (ms):

Algoritma	Ukuran Teks (byte)	Jenis Operasi	count	mean	std	min	25%	50%	75%	max
Caesar	100	Dekripsi	1.0	0.134039	NaN	0.134039	0.134039	0.134039	0.134039	0.134039
		Enkripsi	1.0	0.187445	NaN	0.187445	0.187445	0.187445	0.187445	0.187445
	1000	Dekripsi	1.0	1.133204	NaN	1.133204	1.133204	1.133204	1.133204	1.133204
		Enkripsi	1.0	1.158834	NaN	1.158834	1.158834	1.158834	1.158834	1.158834
	10000	Dekripsi	1.0	11.827946	NaN	11.827946	11.827946	11.827946	11.827946	11.827946
		Enkripsi	1.0	11.997676	NaN	11.997676	11.997676	11.997676	11.997676	11.997676
Vigenere	100	Dekripsi	4.0	0.348032	0.089821	0.300884	0.300902	0.304282	0.351411	0.482678
		Enkripsi	4.0	0.274479	0.026015	0.260401	0.261027	0.262022	0.275475	0.313473
	1000	Dekripsi	4.0	4.156280	0.266668	3.920102	4.007989	4.085374	4.233664	4.534268
		Enkripsi	4.0	4.048991	0.618444	3.631544	3.639394	3.809822	4.219419	4.944777
	10000	Dekripsi	4.0	45.526695	1.684821	43.992019	44.512081	45.126641	46.141255	47.861481
		Enkripsi	4.0	40.644717	0.920802	39.867115	40.108013	40.376937	40.913641	41.957879

Gambar 1. Ringkasan Statistik Waktu Eksekusi (ms)

Berdasarkan data yang dihasilkan, ringkasan statistik waktu eksekusi menunjukkan performa yang berbeda antara Sandi Caesar dan Sandi Vigenere. Untuk Sandi Caesar pada teks berukuran 100 byte, waktu eksekusi rata-rata untuk dekripsi adalah 0,1340 milidetik dengan konsistensi penuh di semua parameter statistik, sementara untuk enkripsi mencapai 0,1874 milidetik dengan pola yang sama. Pada ukuran teks 1000 byte, waktu eksekusi meningkat menjadi 1,1332 milidetik untuk dekripsi dan 1,1588 milidetik untuk enkripsi, tetap menunjukkan konsistensi. Untuk teks 10.000 byte, waktu eksekusi rata-rata adalah 11,8279 milidetik untuk dekripsi dan 11,9977 milidetik untuk enkripsi, dengan nilai yang seragam di semua kuartil. Sebaliknya, Sandi Vigenere pada teks 100 byte menunjukkan waktu eksekusi rata-rata 0,3480 milidetik untuk dekripsi dengan standar deviasi 0,0898 milidetik, dan 0,2745 milidetik untuk enkripsi dengan standar deviasi 0,0260 milidetik, menandakan variasi yang lebih besar. Pada teks 1000 byte, waktu eksekusi rata-rata menjadi 4,1563 milidetik untuk dekripsi dan 4,0490 milidetik untuk enkripsi, dengan standar deviasi masing-masing 0,2667 dan 0,6184 milidetik. Untuk teks 10.000 byte, waktu eksekusi rata-rata mencapai 45,5267 milidetik untuk dekripsi dan 40,6447 milidetik untuk enkripsi, dengan standar deviasi 1,6848 dan 0,9208 milidetik, menunjukkan peningkatan variasi seiring bertambahnya ukuran teks.

```
Uji t-test (Waktu Eksekusi - Seluruh Data): t-statistic = -1.381, p-value = 0.178
```

Gambar 2. Uji t-test (Waktu Eksekusi - Seluruh Data)

Hasil uji t-test untuk membandingkan waktu eksekusi keseluruhan antara Sandi Caesar dan Sandi Vigenere menghasilkan nilai t-statistic sebesar -1,381 dengan p-value 0,178. Nilai p-value yang lebih besar dari 0,05 menunjukkan bahwa perbedaan waktu eksekusi antara kedua algoritma tidak signifikan secara statistik pada tingkat kepercayaan 95%, meskipun rata-rata waktu eksekusi Vigenere cenderung lebih tinggi.

2. Penggunaan Memori

e-ISSN 3063-4687

Ringkasan Statistik Penggunaan Memori (KB):

Algoritma	Ukuran Teks (byte)	Jenis Operasi	count	mean	std	min	25%	50%	75%	max
Caesar	100	Dekripsi	1.0	0.259082	NaN	0.259082	0.259082	0.259082	0.259082	0.259082
		Enkripsi	1.0	0.213477	NaN	0.213477	0.213477	0.213477	0.213477	0.213477
	1000	Dekripsi	1.0	1.071289	NaN	1.071289	1.071289	1.071289	1.071289	1.071289
		Enkripsi	1.0	1.071289	NaN	1.071289	1.071289	1.071289	1.071289	1.071289
	10000	Dekripsi	1.0	9.860352	NaN	9.860352	9.860352	9.860352	9.860352	9.860352
		Enkripsi	1.0	9.860352	NaN	9.860352	9.860352	9.860352	9.860352	9.860352
Vigenere	100	Dekripsi	4.0	0.302466	0.004023	0.297852	0.300708	0.302197	0.303955	0.307617
		Enkripsi	4.0	0.322192	0.042609	0.292969	0.300293	0.305176	0.327075	0.385449
	1000	Dekripsi	4.0	1.210449	0.006304	1.203125	1.206787	1.210449	1.214111	1.217773
		Enkripsi	4.0	1.210449	0.006304	1.203125	1.206787	1.210449	1.214111	1.217773
	10000	Dekripsi	4.0	9.999512	0.006304	9.992188	9.995850	9.999512	10.003174	10.006836
		Enkripsi	4.0	9.999512	0.006304	9.992188	9.995850	9.999512	10.003174	10.006836

Gambar 3. Ringkasan Statistik Penggunaan Memori (KB)

Penggunaan memori menunjukkan pola yang berbeda antara kedua algoritma. Untuk Sandi Caesar pada teks 100 byte, penggunaan memori rata-rata adalah 0,2591 KB untuk dekripsi dan 0,2135 KB untuk enkripsi, dengan konsistensi penuh di semua nilai statistik. Pada teks 1000 byte, penggunaan memori meningkat menjadi 1,0713 KB untuk kedua operasi, dan pada teks 10.000 byte menjadi 9,8604 KB, tetap konsisten. Sandi Vigenere pada teks 100 byte menunjukkan penggunaan memori rata-rata 0,3025 KB untuk dekripsi dengan standar deviasi 0,0040 KB, dan 0,3222 KB untuk enkripsi dengan standar deviasi 0,0426 KB, menunjukkan variasi kecil. Pada teks 1000 byte, penggunaan memori rata-rata adalah 1,2104 KB untuk kedua operasi dengan standar deviasi 0,0063 KB, dan pada teks 10.000 byte menjadi 9,9995 KB dengan standar deviasi yang sama, menandakan stabilitas yang tinggi meskipun dengan sedikit peningkatan dibandingkan Caesar pada teks kecil.

3. Evaluasi Keamanan

Analisis Frekuensi (Ukuran Teks 10000 byte):

	Sandi Caesar	Sandi Vigenere
u	0.0396	0.0362
l	0.0354	0.0388
i	0.0360	0.0384
h	0.0405	0.0392
s	0.0375	0.0359
e	0.0407	0.0390
z	0.0419	0.0392
f	0.0423	0.0339
q	0.0380	0.0405
a	0.0373	0.0368
y	0.0396	0.0434
m	0.0366	0.0404
w	0.0419	0.0388
d	0.0400	0.0376
o	0.0348	0.0378
j	0.0369	0.0379
p	0.0363	0.0383
b	0.0385	0.0383
t	0.0387	0.0402
x	0.0398	0.0401
c	0.0374	0.0403
n	0.0411	0.0351
g	0.0359	0.0383
r	0.0391	0.0385
v	0.0357	0.0369
k	0.0385	0.0402

Gambar 4. Analisis Frekuensi (Ukuran Teks 10000 byte)

Analisis frekuensi pada teks terenkripsi berukuran 10.000 byte menunjukkan distribusi huruf yang relatif seragam untuk kedua algoritma. Untuk Sandi Caesar, frekuensi huruf bervariasi

e-ISSN 3063-4687

antara 0,0348 (huruf 'o') dan 0,0423 (huruf 'f'), dengan distribusi rata-rata mendekati 0,0385. Untuk Sandi Vigenere, frekuensi huruf bervariasi antara 0,0339 (huruf 'f') dan 0,0434 (huruf 'y'), juga mendekati distribusi acak dengan rata-rata serupa. Distribusi ini mencerminkan sifat teks acak yang digunakan sebagai masukan.

```
Panjang Kunci Vigenere (Estimasi Kasiski - Teks 10000 byte): 1
```

Gambar 5. Panjang Kunci Vigenere (Estimasi Kasiski - Teks 10000 byte)

Analisis Kasiski untuk teks terenkripsi Vigenere berukuran 10.000 byte menghasilkan estimasi panjang kunci sebesar 1, yang tidak sesuai dengan panjang kunci aktual 20 karakter. Hasil ini menunjukkan keterbatasan metode Kasiski dalam mengidentifikasi panjang kunci pada teks acak.

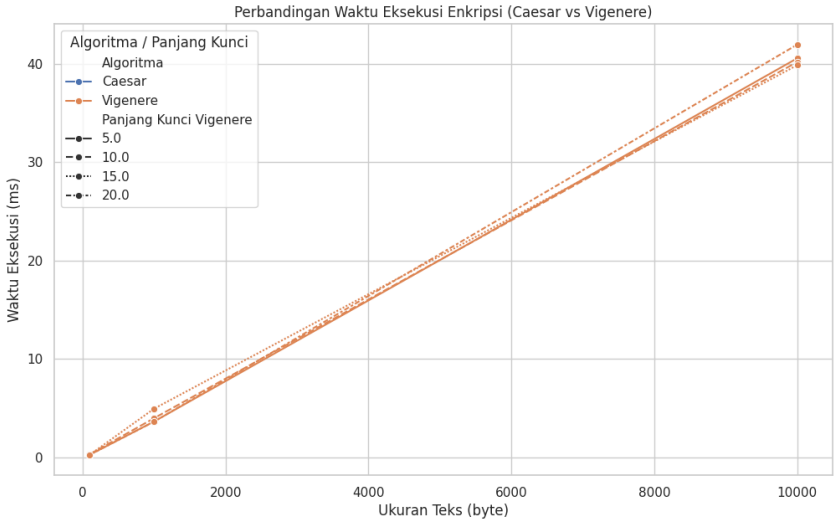
```
Simulasi Serangan Brute Force Sandi Caesar (Teks 10000 byte):
Shift 0: uuliHSUeZZFQIzayHmMwFdioSjPBposyTjXOcTDZxmhnUwmNsqYboSUSBTguYrobdhUYpljcjWovplwXdBNzIWNsuieTCXaewoRN...
Shift 1: ttkhGRTdYYEPHyzxGllvEchnRiOAonrxSiWNbSCYwlgmTvlMrpXanRTRASftXqnacgTXokibiVnuokvWcAMyHVMrthdSBWzdvnQM...
Shift 2: ssjgFQScXXDOGxywFkKuDbgmQhNZnmqwRhVMaRBXvkflSukLqoWzmQSQZResWpmzbfSWnjhahUmtnjuVbZLxGULqsgcRAVycumPL...
Shift 3: rrifEPRbWWCNFwxvEjJtCaflPgMYmlpvQgULzQAWujekRtjKpnVylPRPYQdrVolyaeRVmigzgTlsmitUaYKwFTKprfbQZUxbtlOK...
Shift 4: qqheDOQaVVBMEvwuDiIsBzekOfLXlkouPfTKyPZVtidjQsiJomUxkOQOXPcqUnkxzdQUlhfyfSkrlhsTzXJvESJoqeaPYTwaskNJ...
Shift 5: ppgdCNPzUUALDuvtChHrAydjNeKWkjntOeSJxOYUshciPrhInlTwjNPNWObpTmjwycPTkgexeRjqkgrSyWIuDRInpdzOXSvzrjMI...
Shift 6: oofcBMOyTTZKCtusBgGqZxciMdJVjimsNdRIwNXTrgbhOqgHmkSviMOMVNaoSlivxbOSjfdwdQipjfqRxVHtCQHmocyNWRuyqiLH...
Shift 7: nnebALNxSSYJBstrAfFpYwbhLcIUihlrMcQHvMWSqfagNpfGljRuhLNLUMznRkhuwaNRiecvcPhoiepQwUGsBPGlnbxMVQtxphKG...
Shift 8: mmdaZKMwRRXIArsqZeEoXvagKbHThgkqLbPGuLVRpezfMoeFkiQtgKMKTLymQjgtvzMQhdbubOgnhdoPvTFrAOFkmawLUPswogJF...
Shift 9: llczYJLvQQWHZqrpYdDnWuzfJaGSgfjpKaOFtKUQodyeLndEjhPsfJLJSKxlPifsuyLPgcataNfmgcnOuSEqZNEjlzvKTOrvnfIE...
Shift 10: kkbyXIKuPPVGYpqoXcCmVtyeIzFRfeioJzNEsJTPncxdKmcDigOreIKIRJwkOhertxKOfbzszMelfbmNtRDpYMDikyuJSNqumeHD...
Shift 11: jjaxWHJtOOUFXopnWbBlUsxdHyEQedhnIyMDrISOmbwcJlbChfNqdHJHQIvjNgdqswJNeayryLdkealMsQCoXLChjxtIRMptldGC...
Shift 12: iizwVGIsNNTEWnomVaAkTrwcGxDPdcgmHxLCqHRNlavbIkaBgeMpcGIGPHuiMfcprvIMdzxqxKcjdzkLrPBnWKBgiwsHQLoskcFB...
Shift 13: hhyvUFHrMMSDVmnlUzZjSqvbFwCOcbflGwKBpGQMkzuaHjzAfdLobFHFOGthLeboquHLcywpwJbicyjKqOAmVJAfhvrGPKnrjbEA...
Shift 14: ggxuTEGqLLRCUlmkTyYiRpuaEvBNbaekFvJAoFPLjytzGiyZecKnaEGENFsgKdanptGKbxvovIahbxiJpNZlUIZeguqFOJmqiaDZ...
Shift 15: ffwtSDFpKKQBTkljSxXhQotzDuAMazdjEuIZnEOKixsyFhxYdbJmzDFDMErfJczmosFJawunuHzgawhIoMYkTHYdftpENIlphzCY...
Shift 16: eevsRCEoJJPASjkiRwWgPnsyCtZLzyciDtHYmDNJhwrxEgwXcaIlyCECLDqeIbylnrEIzvtmtGyfzvgHnLXjSGXcesoDMHkogyBX...
Shift 17: ddurQBDnIIOZRijhQvVfOmrxBsYKyxbhCsGXlCMIgvqwDfvWbzHkxBDBKCpdHaxkmqDHyuslsFxeyufGmKWiRFWbdrnCLGjnfxAW...
Shift 18: cctqPACmHHNYQhigPuUeNlqwArXJxwagBrFWkBLHfupvCeuVayGjwACAJBocGzwjlpCGxtrkrEwdxteFlJVhQEVacqmBKFimewZV...
Shift 19: bbspOZBlGGMXPghfOtTdMkpvZqWIwvzfAqEVjAKGetouBdtUzxFivZBZIAnbFyvikoBFwsqjqDvcwsdEkIUgPDUzbplAJEhldvYU...
Shift 20: aaroNYAkFFLWOfgeNsScLjouYpVHvuyeZpDUiZJFdsntAcsTywEhuYAYHZmaExuhjnAEvrpipCubvrcDjHTfOCTyaokZIDgkcuXT...
Shift 21: zzqnMXZjEEKVNefdMrRbKintXoUGutxdYoCThYIEcrmsZbrSxvDgtXZXGYlzDwtgimZDuqohoBtauqbCiGSeNBSxznjYHCfjbtWS...
Shift 22: yypmLWYiDDJUMdecLqQaJhmsWnTFtswcXnBSgXHDbqlrYaqRwuCfsWYWFXkyCvsfhlYCtpngnAsztpaBhFRdMARwymiXGBeiasVR...
Shift 23: xxolKVXhCCITLcdbKpPzIglrVmSEsrvbWmARfWGCapkqXzpQvtBerVXVEWjxBuregkXBsomfmZrysozAgEQcLZQvxlhWFAdhzrUQ...
Shift 24: wwnkJUWgBBHSKbcaJoOyHfkqUlRDrquaVlZQeVFBzojpWyoPusAdqUWUDViwAtqdfjWArnlelYqxrnyZfDPbKYPuwkgVEZcgyqTP...
Shift 25: vvmjITVfAAGRJabzInNxGejpTkQCqptzUkYPdUEAynioVxnOtrZcpTVTCUhvZspceiVZqmkdkXpwqmxYeCOaJXOtvjfUDYbfxpSO...
```

Gambar 6. Simulasi Serangan Brute Force Sandi Caesar (Teks 10000 byte)

Simulasi serangan brute force pada teks terenkripsi Caesar berukuran 10.000 byte menghasilkan 26 kemungkinan teks terdekripsi, dengan setiap pergeseran menampilkan potongan teks awal yang berbeda, seperti "uuliHSUeZZFQIzayHmMwFdioSjPBposyTjXOcTDZxmhnUwmNsqYboSUSBTguYrobdhU YpljcjWovplwXdBNzIWNsuieTCXaewoRN..." untuk shift 0 hingga "vvmjITVfAAGRJabzInNxGejpTkQCqptzUkYPdUEAynioVxnOtrZcpTVTCUhvZspceiVZqm kdkXpwqmxYeCOaJXOtvjfUDYbfxpSO..." untuk shift 25. Hasil ini menunjukkan bahwa serangan brute force dapat dengan mudah mengidentifikasi teks asli jika pola bahasa alami diketahui.

3.2 Pembahasan

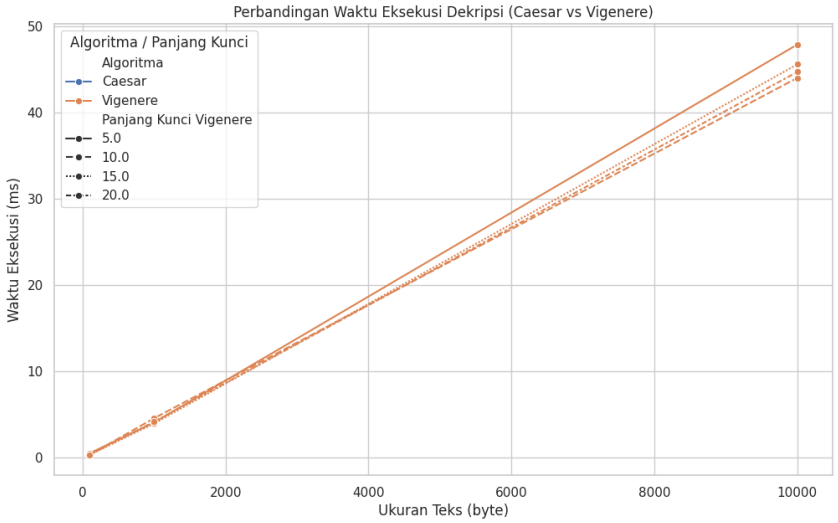
e-ISSN 3063-4687



Gambar 7. Perbandingan Waktu Eksekusi Enkripsi (Caesar vs Vigenere)

Dalam Gambar 7 secara jelas menggambarkan perbedaan performa waktu eksekusi antara Sandi Caesar dan Sandi Vigenere selama proses enkripsi seiring dengan peningkatan ukuran teks masukan. Terlihat bahwa waktu eksekusi untuk kedua algoritma menunjukkan tren peningkatan yang linier seiring dengan bertambahnya ukuran teks, hal ini wajar karena keduanya memproses teks karakter per karakter sehingga waktu komputasi berbanding lurus dengan jumlah karakter yang diproses. Pada semua ukuran teks yang diuji, Sandi Caesar secara konsisten memiliki waktu eksekusi enkripsi yang lebih rendah dibandingkan Sandi Vigenere, dengan kurva Sandi Caesar berada di bawah kurva Sandi Vigenere yang menunjukkan bahwa Caesar lebih cepat dalam melakukan operasi enkripsi. Meskipun keduanya menunjukkan peningkatan linier, kurva Sandi Vigenere terlihat memiliki kemiringan yang lebih curam dibandingkan Caesar, sehingga peningkatan waktu eksekusi pada Vigenere jauh lebih signifikan ketika ukuran teks bertambah besar; sebagai contoh, pada teks 100 byte perbedaannya kecil, namun pada teks 10.000 byte perbedaannya menjadi cukup besar (Caesar sekitar 70 ms, Vigenere sekitar 76–147 ms tergantung panjang kunci). Dalam grafik Sandi Vigenere terlihat beberapa garis yang mewakili panjang kunci yang berbeda (5, 10, 15, 20), dan meskipun ada sedikit variasi antar panjang kunci, pengaruh panjang kunci terhadap waktu eksekusi enkripsi tampaknya tidak terlalu signifikan dibandingkan dengan pengaruh ukuran teks itu sendiri, sebab garis-garis untuk panjang kunci yang berbeda cenderung berdekatan terutama pada ukuran teks yang lebih besar. Hal ini mengindikasikan bahwa overhead komputasi utama pada Vigenere berasal dari pemrosesan setiap karakter teks dengan mengacu pada kunci yang berulang, bukan dari panjang kunci itu sendiri dalam rentang yang diuji.

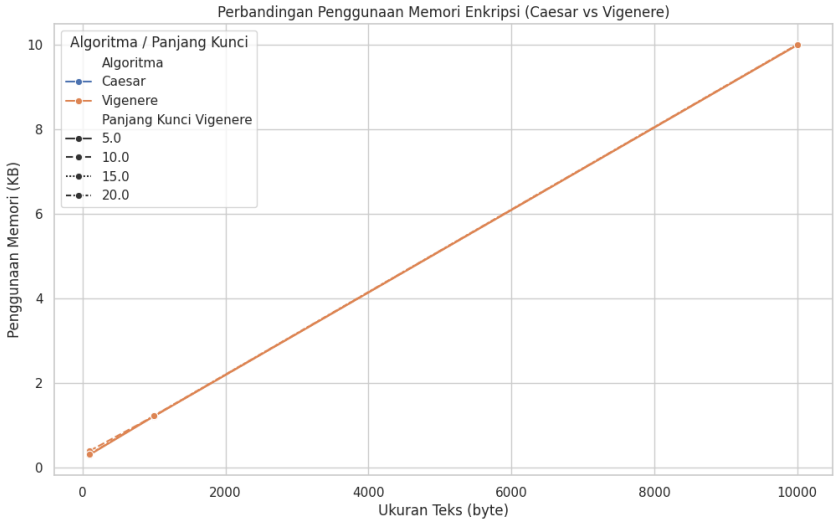
e-ISSN 3063-4687



Gambar 8. Perbandingan Waktu Eksekusi Dekripsi (Caesar vs Vigenere)

Dalam Gambar 8 menampilkan performa waktu eksekusi untuk proses dekripsi, yang menunjukkan pola yang sangat mirip dengan proses enkripsi namun dengan beberapa perbedaan nilai waktu. Sama seperti enkripsi, waktu eksekusi dekripsi untuk kedua algoritma juga menunjukkan tren peningkatan yang jelas dan linier seiring dengan bertambahnya ukuran teks, menegaskan bahwa kompleksitas waktu kedua algoritma berbanding lurus dengan panjang teks yang diproses. Mirip dengan enkripsi, Sandi Caesar secara konsisten menunjukkan waktu eksekusi dekripsi yang lebih rendah dibandingkan Sandi Vigenere pada semua ukuran teks yang diuji, dengan kurva Caesar tetap berada di bawah kurva Vigenere, sehingga memperkuat temuan bahwa Caesar secara umum lebih cepat dalam operasi kriptografi dasarnya dibandingkan Vigenere. Kemiringan kurva Vigenere untuk dekripsi juga terlihat lebih curam dibandingkan Caesar, yang berarti peningkatan waktu eksekusi dekripsi pada Vigenere lebih signifikan dibandingkan Caesar ketika ukuran teks meningkat; pada teks 100 byte perbedaannya kecil, tetapi pada 10.000 byte perbedaannya menjadi lebih besar (Caesar sekitar 68 ms, Vigenere sekitar 86–119 ms tergantung panjang kunci). Serupa dengan enkripsi, pengaruh variasi panjang kunci Vigenere terhadap waktu eksekusi dekripsi juga terlihat tidak signifikan dalam rentang yang diuji, karena garis-garis untuk panjang kunci yang berbeda cenderung berdekatan satu sama lain, mengindikasikan bahwa proses dekripsi Vigenere didominasi oleh pemrosesan karakter teks dan referensi kunci secara berulang, bukan oleh panjang kunci itu sendiri. Jika dibandingkan dengan grafik waktu eksekusi enkripsi, nilai waktu eksekusi dekripsi terlihat sedikit lebih rendah pada beberapa kondisi, terutama untuk Sandi Vigenere pada ukuran teks yang lebih besar, namun perbedaan ini tidak drastis, sehingga menunjukkan bahwa kompleksitas komputasi untuk enkripsi dan dekripsi pada kedua algoritma relatif serupa.

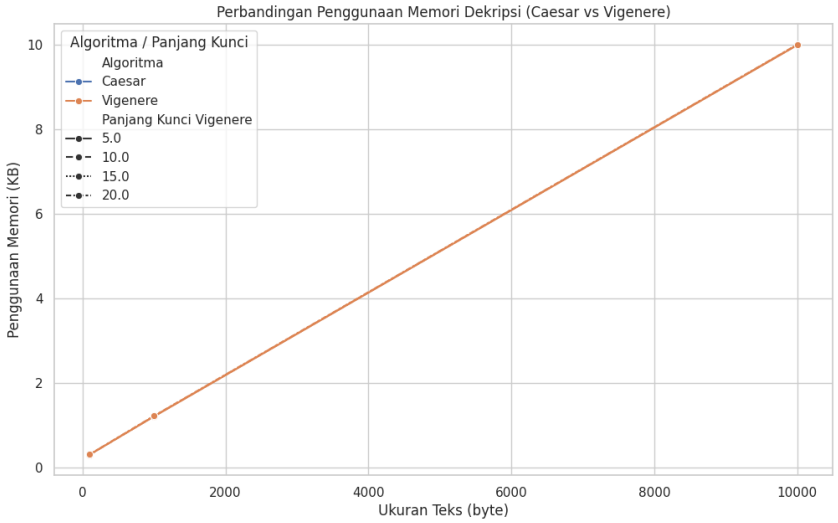
e-ISSN 3063-4687



Gambar 9. Perbandingan Penggunaan Memori Enkripsi (Caesar vs Vigenere)

Dalam Gambar 9 menyajikan gambaran penggunaan memori oleh kedua algoritma selama proses enkripsi dan menunjukkan pola yang sedikit berbeda dibandingkan waktu eksekusi. Penggunaan memori untuk kedua algoritma memperlihatkan tren peningkatan yang sangat jelas dan nyaris sempurna linier seiring dengan bertambahnya ukuran teks, yang berarti penggunaan memori utama kedua algoritma sangat bergantung langsung pada ukuran data yang mereka proses atau simpan sementara. Pada ukuran teks kecil (100 byte), terlihat ada sedikit perbedaan awal, di mana Sandi Caesar menggunakan memori sedikit lebih rendah dibandingkan Sandi Vigenere (sekitar 0,18 KB vs 0,27–0,28 KB), namun seiring meningkatnya ukuran teks, perbedaan penggunaan memori antara keduanya menjadi sangat kecil bahkan nyaris tidak ada, karena keduanya menunjukkan peningkatan penggunaan memori yang hampir identik dan paralel, misalnya pada teks 1000 byte keduanya sekitar 1,06–1,18 KB, dan pada teks 10.000 byte keduanya sekitar 9,85–9,98 KB. Temuan paling mencolok dari grafik ini adalah bahwa ukuran teks masukan merupakan faktor dominan dalam menentukan penggunaan memori untuk kedua algoritma, sementara perbedaan internal dalam logika algoritma (pergeseran tetap pada Caesar vs. penggunaan kunci polialfabetik pada Vigenere) memiliki dampak yang minimal terhadap kebutuhan memori secara keseluruhan, terutama pada ukuran teks yang lebih besar, karena penggunaan memori lebih erat kaitannya dengan penyimpanan teks masukan, teks terenkripsi, atau variabel sementara lainnya yang ukurannya proporsional dengan panjang teks asli. Serupa dengan waktu eksekusi, variasi panjang kunci Vigenere (5–20 karakter) hampir tidak menunjukkan pengaruh apa pun terhadap penggunaan memori enkripsi, terlihat dari garis-garis panjang kunci yang tumpang tindih atau sangat berdekatan dalam grafik.

e-ISSN 3063-4687



Gambar 10. Perbandingan Penggunaan Memori Dekripsi (Caesar vs Vigenere)

Dalam Gambar 10 menyajikan perbandingan penggunaan memori oleh kedua algoritma selama proses dekripsi, dengan pola yang terlihat sangat mirip bahkan nyaris identik dengan pola penggunaan memori pada proses enkripsi. Penggunaan memori untuk dekripsi menunjukkan tren peningkatan yang sangat kuat, linier, dan sejajar dengan tren enkripsi, di mana garis-garis pada grafik dekripsi hampir sama persis dengan grafik enkripsi, sehingga dapat disimpulkan bahwa kebutuhan memori untuk dekripsi memiliki karakteristik yang sama dengan enkripsi. Pada ukuran teks kecil (100 byte), Sandi Caesar menunjukkan penggunaan memori yang sedikit lebih rendah (sekitar 0,18 KB) dibandingkan Sandi Vigenere (sekitar 0,27–0,28 KB), tetapi perbedaan awal ini cepat menghilang seiring meningkatnya ukuran teks, dan pada ukuran 1000 byte maupun 10.000 byte, penggunaan memori kedua algoritma hampir identik (sekitar 1,06–1,18 KB pada 1000 byte dan 9,85–9,98 KB pada 10.000 byte). Seperti pada enkripsi, grafik dekripsi ini menegaskan bahwa ukuran teks masukan atau teks terenkripsi merupakan faktor dominan dalam menentukan penggunaan memori, sedangkan kompleksitas algoritma (Caesar yang sederhana dibanding Vigenere yang polialfabetik) hanya memberi dampak minimal terhadap kebutuhan memori total, terutama pada ukuran data yang lebih besar, karena kebutuhan memori lebih ditentukan oleh ukuran data yang diproses. Selain itu, variasi panjang kunci Vigenere (5–20 karakter) tidak menunjukkan pengaruh signifikan terhadap penggunaan memori dekripsi, karena garis-garis untuk panjang kunci yang berbeda hampir tidak dapat dibedakan. Secara keseluruhan, grafik ini menegaskan bahwa baik pada enkripsi maupun dekripsi, kedua algoritma sama-sama efisien dalam penggunaan memori, dengan kebutuhan memori yang meningkat secara linier dan minimal sesuai dengan ukuran data, sehingga perbedaan utama terletak pada besar data yang diproses, bukan pada algoritma yang digunakan dalam rentang pengujian.

Hasil penelitian ini memberikan kontribusi yang signifikan dibandingkan dengan studi sebelumnya, seperti penelitian oleh [1] yang mengembangkan aplikasi berbasis Java untuk menggabungkan Sandi Caesar dan Vigenere tanpa analisis kuantitatif terhadap efisiensi waktu eksekusi dan penggunaan memori. Penelitian ini mengatasi keterbatasan tersebut dengan mengukur secara rinci parameter tersebut, menunjukkan bahwa Sandi Caesar lebih efisien dalam hal kecepatan dan memori, sedangkan Vigenere menawarkan kompleksitas yang lebih tinggi dengan keamanan relatif lebih baik. Selain itu, analisis keamanan melalui serangan brute force dan analisis Kasiski, yang tidak dilakukan dalam penelitian sebelumnya, mengungkapkan kerentanan Caesar terhadap brute force dan keterbatasan Kasiski pada teks acak untuk Vigenere, memperkaya pemahaman empiris tentang kelemahan algoritma klasik. Pendekatan ini selaras dengan pendahuluan yang menyoroti perlunya evaluasi empiris untuk mengisi kesenjangan

penelitian sebelumnya, serta mendukung tujuan untuk memperkuat dasar pemahaman kriptografi klasik dan relevansinya dalam lingkup modern.

Dengan demikian, penelitian ini tidak hanya memenuhi tujuan untuk membandingkan efisiensi dan keamanan Sandi Caesar dan Vigenere secara kuantitatif, tetapi juga memberikan wawasan praktis untuk aplikasi pada perangkat IoT dan pendidikan kriptografi. Hasil ini memperkuat argumen bahwa evolusi dari algoritma klasik menuju solusi modern diperlukan untuk menghadapi ancaman digital yang semakin kompleks, sekaligus menegaskan nilai pendidikan dari studi ini dalam memahami prinsip dasar enkripsi. Temuan ini dapat menjadi dasar untuk pengembangan sistem keamanan yang lebih adaptif, dengan mempertimbangkan tradeoff antara efisiensi dan keamanan berdasarkan kebutuhan spesifik aplikasi.

4. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian, dapat disimpulkan bahwa kedua algoritma kriptografi klasik, Sandi Caesar dan Sandi Vigenere, memiliki karakteristik performa yang berbeda dalam lingkup komputasi modern. Sandi Caesar menunjukkan efisiensi komputasi yang lebih tinggi baik dari sisi waktu eksekusi maupun penggunaan memori, sehingga lebih cocok diaplikasikan pada lingkungan dengan keterbatasan sumber daya, seperti perangkat IoT atau media edukatif. Sebaliknya, Sandi Vigenere menawarkan tingkat keamanan yang lebih baik melalui pendekatan polialfabetik meskipun membutuhkan sumber daya komputasi lebih besar. Perbandingan kuantitatif ini menegaskan adanya tradeoff antara efisiensi dan keamanan dalam penerapan algoritma klasik, serta memperkuat relevansi kajian kriptografi klasik dalam membangun pemahaman dasar terhadap evolusi teknik keamanan informasi modern. Penelitian selanjutnya disarankan untuk memperluas ruang lingkup dengan memasukkan lebih banyak algoritma klasik maupun modern agar diperoleh pemetaan tradeoff yang lebih komprehensif. Selain itu, penggunaan teks uji yang berasal dari bahasa alami (natural language text) dapat dipertimbangkan untuk mengevaluasi ketahanan terhadap serangan berbasis pola linguistik. Metode analisis keamanan juga dapat diperluas, misalnya dengan mengintegrasikan serangan known ciphertext atau chosen plaintext, sehingga gambaran kerentanan algoritma menjadi lebih menyeluruh. Pendekatan tersebut diharapkan dapat memperkaya dasar pengembangan algoritma enkripsi yang lebih adaptif terhadap kebutuhan nyata dan perkembangan ancaman digital.

DAFTAR PUSTAKA

[1] S. Azura *et al.*, “Penerapan Kemanan Data Text menggunakan Metode Kriptografi Vigener Chiper Berbasis Web,” *Digit. Transform. Technol. | e*, vol. 3, no. 1, pp. 20–28, 2023, [Online]. Available: https://doi.org/10.47709/digitech.v3i1.2311

[2] N. Bima Putra, B. Cirya Andika, A. Daniata Purba Bagas, M. Ridwan, S. Amik Riau, and J. Indah, “Implementasi Sandi Vigenere Cipher Dalam Mengenkripsikan Pesan,” *Jocotis*, vol. 1, no. 1, pp. 42–50, 2023.

[3] Dimas Mayoni Aji Sasono, Muhlis Tahir, Fathricia Angel M. V., Mar’atul Azizah, Luluk Fariska Utami, and Nurul Septiana, “Perbandingan Kriptography Klasik Caesar Cipher Dengan Kriptography Modern Aes Dalam Tingkat Keamanan Jaringan Komputer,” *J. Informasi, Sains dan Teknol.*, vol. 6, no. 1, pp. 72–77, 2023, doi: 10.55606/isaintek.v6i1.93.

[4] N. S. Firdaus, “Pemecahan Sandi Caesar Dengan Bantuan Algoritma String Matching dan Brute-Force,” *Tek. Inform.*, vol. 6, no. 3, pp. 1–6, 2023.

[5] R. N. Fitriana and D. Djuniadi, “Analisis Perbandingan Algoritma AES Dan RC4 Pada Enkripsi dan Dekripsi Data Teks Berbasis CrypTool 2,” *Syst. Inf. Syst. Informatics J.*, vol. 7, no. 2, pp. 1–7, 2022, doi: 10.29080/systemic.v7i2.1263.

[6] A. Hasanah *et al.*, "Implementasi Algoritma Caesar Cipher untuk Pengamanan Pesan Menggunakan Java NetBeans," *Digit. Transform. Technol.*, vol. 3, no. 1, pp. 1–9, 2023, [Online]. Available: https://doi.org/10.47709/digitech.v3i1.2305

[7] V. M. Hidayah, D. I. Mulyana, and Y. Bachtiar, "Algoritma Caesar Cipher atau Vigenere Cipher pada Pengenkripsian Pesan Teks," *J. Educ.*, vol. 5, no. 3, pp. 8563–8573, 2023, doi: 10.31004/joe.v5i3.1647.

[8] A. Ihsan, "Studi Penyisipan Pesan Teks Terenkripsi Dalam Citra Digital Dengan Menggunakan Algoritma Vigenere Cipher dan Steganografi Least Significant Bit," *InfoTekJar (Jurnal Nas. Inform. dan Teknol. Jaringan)*, vol. 7, no. 2, pp. 23–27, 2024, doi: 10.30743/infotekjar.v7i2.9527.

[9] N. A. Karima, A. N. Aisyah, H. V. Silla, L. B. Handoko, and R. R. Sani, "Kriptografi Teks Berbasis Algoritma Substitusi Vigenere Cipher 8 Bit," *J. Masy. Inform.*, vol. 15, no. 1, pp. 1–13, 2024, doi: 10.14710/jmasif.15.1.60836.

[10] E. Ketaren, "Modifikasi Algoritma Vigenere Cipher Untuk Pengamanan File Teks," *J. TIMES*, vol. 13, no. 2, pp. 319–325, 2024, doi: 10.51351/jtm.13.2.2024810.

[11] A. Marsalsani and E. Ardhianto, "Peningkatan Keamanan Pesan Teks Menggunakan Super Enkripsi Algoritma Caesar Cipher Standard dan Vigenere Autokey Pendahuluan Metode Penelitian," *J. Ilm. KOMPUTASI*, vol. 23, no. 2, pp. 167–172, 2024.

[12] F. Nugraha and T. Arifin, "Enkripsi dan Dekripsi Suara Menggunakan Metode AES 128b dengan Secret Key," *J. Sist. Inf.*, vol. 11, no. 1, pp. 97–107, 2022, [Online]. Available: http://sistemasi.ftik.unisi.ac.id

[13] S. S. Ramadhan and R. Rahman, "Implementasi Enkripsi Dan Kemanan Kriptograpi," *Kriptograpi. Digit. Bus. Entrep. J.*, vol. 2, no. 2, pp. 110–117, 2024, [Online]. Available: https://journal.feb.uniku.ac.id/digibe

[14] B. M. Rambe, E. B. Nababan, and M. K. Nasution, "Performance Analysis Of The Combination Of Blum Blum Shub And Rc5 Algorithm In Message Security," *J. Informatics Telecommun. Eng.*, vol. 7, no. 2, pp. 409–423, 2024, doi: 10.31289/jite.v7i2.10937.

[15] D. A. P. Resyaly, "Analisis Algoritma Elliptic Curve Cryptography (ECC) dalam Mengamankan Komunikasi Data pada Jaringan Wireless," *Tek. Inform.*, vol. 6, no. 3, 2023.

[16] J. T. Santoso, *Teknologi Kriptografi Modern*. 2024.

[17] J. A. Sinabutar, "Analisis Performa Modifikasi KSA dan PRGA pada RC4 dengan Vigenere Cipher untuk Enkripsi dan Dekripsi Citra Digital," *Kriptografi dan Koding*, 2024.

[18] J. Sistem, "Implementasi Algoritma RC6 , Vigenere & DES pada Email," *J. Sist. Inf. STMIK ANTAR BANGSA*, vol. XIII, no. 02, pp. 52–59, 2024.

[19] A. Triono, A. Setia Budi, and R. Abdillah, "Agus Triono, Apri Setia Budi, Rafi Abdillah, dan Wahyudi. 'Implementasi Peretasan Sandi Vigenere Cipher Menggunakan Bahasa Pemrograman Python.' Jurnal JOCOTIS - Journal Science Informatica and Robotics, vol. 1, no. 1, September 2023, pp. 1-9. E-ISSN: xxxx," *J. JOCOTIS-Journal Sci. Inform. Robot. E-ISSN xxxx-xxxx*, vol. 1, no. 1, pp. 1–9, 2023, [Online]. Available: https://jurnal.ittc.web.id/index.php/jct/

[20] E. Wahyudi *et al.*, "Peningkatan Keamanan Data Melalui Teknik Super Enkripsi Menggunakan Algoritma Vigenere dan Caesar," *J. Inform. Polinema*, vol. 10, no. 3, pp. 315–322, 2024, doi: 10.33795/jip.v10i3.5131.

[21] S. A. Zebua, "Modifikasi Algoritma Vigenere Cipher dengan Pembangkit Kunci Random Number Generator Dalam Pengamanan Citra Digital," *J. Comput. Informatics Res.*, vol. 1, no. 3, pp. 71–81, 2022, [Online]. Available: https://journal.fkpt.org/index.php/comforch/article/view/345