Volume 18 No. 1 (November 2025), pp. 74-86 | ISSN 2541-1760 (online) | ISSN 1907-3984 (print)
Jurnal Akademika
http://ojs.unh.ac.id/index.php/ | DOI: https://doi.org/10.53564/6w0a4245

Analisis Ketahanan Kombinasi Teknik Kriptografi Rail Fence dan Vigenere Cipher Terhadap Serangan *Brute-Force* Menggunakan Pemrograman Python

Aulia Rachmawati [1], Yandi Anzari [2], Muhammad Damas Fatih [3], Pariyadi [4]
[1,2,3,4] Program Studi Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Jambi, Jambi, Indonesia
Email: [1]auliarachmawati@unja.ac.id , [2]yandi.anzari@unja.ac.id, [3]muhammaddamasfatih@unja.ac.id , [4]pariyadi@unja.ac.id

Article Information

Article history

Received 14 October 2025
Revised 04 November 2025
Accepted 07 November 2025
Available 29 November 2025

Keywords

Algorithms
Kriptografi
Rail Fence cipher
Vigenere cipher
Brute-Force

Corresponding Author:

Aulia Rachmawati,
Program Studi Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Jambi,
auliarachmawati@unja.ac.id

Abstract

Data is an essential asset that must be protected due to its strategic role in maintaining the continuity of information systems. Therefore, data security becomes a crucial aspect in information management. One of the approaches to ensure data security is through the application of cryptographic techniques. Two commonly used classical cryptographic algorithms are the Rail Fence Cipher and the Vigenere Cipher. However, the use of a single algorithm has weaknesses, as it remains vulnerable to brute-force attacks. This study aims to combine two cryptographic methods, namely the transposition technique (Rail Fence) and the substitution technique (Vigenere), to enhance resistance against brute-force attacks. The research employs an experimental method by implementing the combined algorithms using the Python programming language. Testing was conducted on three scenarios: encryption using Rail Fence, Vigenère, and their combination. The results show that the combined algorithm increases key space and produces more complex character distribution patterns, thereby improving security against brute-force attacks.

Keywords: *Algorithms, Brute-Force, Kriptografi, Rail Fence Cipher, Vigenere cipher*

Abstrak

Data merupakan aset penting yang harus dilindungi karena memiliki peran strategis dalam menjaga keberlangsungan suatu sistem informasi. Oleh karena itu, keamanan data menjadi aspek yang sangat krusial dalam pengelolaan informasi. Salah satu upaya untuk menjaga keamanan data adalah melalui penerapan teknik kriptografi. Dua di antara algoritma kriptografi klasik yang umum digunakan adalah Rail Fence Cipher dan Vigenere Cipher. Namun, penggunaan algoritma tunggal masih memiliki kelemahan karena rentan terhadap serangan *brute-force*. Penelitian ini bertujuan untuk mengkombinasikan dua metode kriptografi, yaitu teknik transposisi (Rail Fence) dan teknik substitusi (Vigenere), guna meningkatkan ketahanan terhadap serangan *brute-force*. Metode penelitian yang digunakan adalah metode eksperimen dengan implementasi algoritma menggunakan bahasa pemrograman Python. Pengujian dilakukan pada tiga skenario, yaitu enkripsi dengan Rail Fence, Vigenere, dan kombinasi keduanya. Hasil penelitian menunjukkan bahwa kombinasi kedua algoritma tersebut mampu memperluas ruang kunci dan menghasilkan pola distribusi karakter yang lebih kompleks sehingga meningkatkan keamanan terhadap serangan *brute-force*.

Kata Kunci: Algoritma, *Brute-Force,* Kriptografi*, Rail Fence Cipher, Vigenere cipher*

1. Pendahuluan

Perkembangan sistem informasi yang semakin pesat pada era digital mendorong pemanfaatan teknologi secara luas. Peningkatan intensitas penggunaan data dan informasi menjadikan aspek keamanan sebagai faktor krusial yang tidak dapat diabaikan, hal ini dikarenakan data merupakan aset berharga dalam suatu sistem informasi [1]. Data memiliki peran fundamental dalam mendukung keberlangsungan suatu organisasi maupun bisnis, sehingga memerlukan perlindungan yang optimal. Dalam konteks keamanan data, terdapat tiga elemen utama yang harus diperhatikan, yaitu *Confidentiality* (kerahasiaan), *Integrity* (integritas), dan *Availability* (ketersediaan) [2]. Kemudahan akses terhadap data dan informasi yang disediakan oleh perkembangan teknologi tentu menimbulkan potensi risiko, salah satunya adalah permasalahan terkait keamanan data [3]. Oleh karena itu, diperlukan suatu teknik pengamanan data yang mampu memenuhi prinsip-prinsip keamanan tersebut. Salah satu pendekatan yang umum digunakan adalah menjaga aspek kerahasiaan data melalui penerapan kriptografi [4].

Kriptografi merupakan disiplin ilmu yang bertujuan untuk melindungi informasi dengan cara mengacak pesan sehingga tidak dapat dipahami oleh pihak yang tidak berwenang [5]. Secara umum, terdapat dua jenis algoritma klasik yang populer digunakan, yaitu algoritma dengan teknik transposisi dan algoritma dengan teknik substitusi [6]. Teknik transposisi bekerja dengan cara menyandikan pesan melalui pengubahan posisi atau urutan karakter dalam *plaintext* (teks asli) tanpa mengganti karakter itu sendiri [7]. Pada teknik ini, huruf-huruf *plaintext* tetap dipertahankan, tetapi disusun kembali mengikuti pola tertentu sehingga menghasilkan *ciphertext* (teks tersandi) yang sulit dipahami tanpa mengetahui aturan transposisi yang digunakan. Salah satu contoh dari teknik ini adalah Rail Fence Cipher. Sebaliknya, teknik substitusi bekerja dengan cara mengganti setiap karakter, huruf, atau kelompok simbol dalam *plaintext* dengan karakter lain sesuai aturan tertentu [8]. Prinsip utama teknik ini adalah pemetaan satu-ke-satu antara elemen *plaintext* ke dalam *ciphertext*, sehingga pesan menjadi sulit dipahami tanpa mengetahui kunci substitusi yang digunakan. Salah satu contoh metode substitusi yang terkenal adalah Vigenere Cipher.

Meskipun kedua teknik ini memberikan dasar yang baik dalam penyandian pesan, algoritma kriptografi klasik masih relatif rentan terhadap serangan kriptanalisis, khususnya serangan *brute-force*. Serangan *brute-force* merupakan metode yang mencoba memecahkan skema enkripsi dengan cara melakukan pengujian sistematis terhadap seluruh kemungkinan kunci hingga kunci yang benar ditemukan [9]. Penelitian [10] menyebutkan bahwa Rail Fence Cipher memiliki kelemahan terhadap serangan *brute-force* karena tidak melakukan perubahan terhadap karakter *plaintext*, melainkan hanya memodifikasi susunannya. Hal ini mengakibatkan pola frekuensi karakter *plaintext* tetap terlihat dalam *ciphertext*, sehingga mempermudah proses analisis kriptografi. Oleh karena itu, diperlukan modifikasi dengan memadukan beberapa metode dari algoritma klasik lain, untuk meningkatkan keamanan dan menguatkan hasil enkripsi agar lebih sulit ditembus menggunakan teknik *brute-force* [11].

Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk mengusulkan kombinasi teknik kriptografi Rail Fence Cipher dan Vigenere Cipher serta menganalisis tingkat ketahanan kombinasi tersebut terhadap serangan *brute-force*. Kebaruan (*novelty*) dari penelitian ini terletak pada pendekatan hibridisasi algoritma kriptografi klasik yang menggabungkan metode

transposisi (Rail Fence) dengan metode substitusi (Vigenere). Kombinasi ini diharapkan dapat meminimalisasi kelemahan masing-masing teknik ketika digunakan secara terpisah, sehingga mampu menghasilkan *ciphertext* dengan pola distribusi karakter yang lebih kompleks dan meningkatkan ketahanan terhadap serangan *brute-force*.

2. Kajian Terdahulu

Penerapan Rail Fence bergantung pada sebuah kunci berupa jumlah baris yang digunakan untuk membagi teks asli [12]. Pola kriptografi Rail Fence diilustrasikan pada Gambar 1.

Gambar 1. Pola Rail Fence dengan 3 *rails* [13]

Pada gambar di atas, teknik enkripsi model Rail Fence plaintext disusun kebawah secara diagonal sampai bawah sesuai dengan *rails* yang telah ditentukan dan kemudian disusun keatas dengan pola yang sama sampai ke *rails* paling atas. Pola tersebut diulang sampai semua karakter pada plaintext masuk ke dalam *rails*. Setelah menerapkan pola tersebut, ciphertext dihasilkan dengan cara membaca karakter dari kiri ke kanan dari setiap baris *rails* dimulai dari *rails* paling atas [12].

Di lain sisi, Proses enkripsi Vigenere dilakukan dengan menggunakan sebuah kata kunci (*key*) untuk menentukan pergeseran huruf pada teks asli (*plaintext*). Setiap huruf dalam teks dienkripsi berdasarkan huruf yang sesuai pada kunci, sehingga menghasilkan teks sandi (*ciphertext*) yang tampak acak [14]. Teknik kriptografi Vigenere diilustrasikan pada gambar 2.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Gambar 2. Tabel Pemetaan Enkripsi Vigenere Cipher [14]

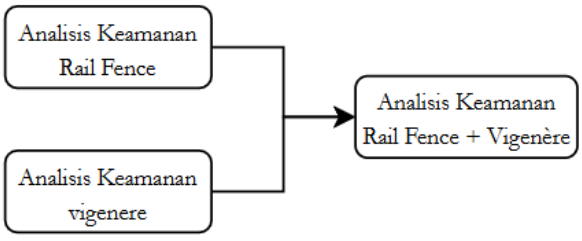
Pada gambar di atas, Vigenere mengenkripsi plainteks pada pesan dengan cara menggeser huruf pada pesan tersebut sejauh nilai kunci pada deret alphabet. Vigenere cipher adalah salah satu algoritma kriptografi klasik yang menggunakan metod substitusi abjadmajemuk. Substitusi abjad-majemuk mengenkripsi setiap huruf yang ada menggunakan kunci yang berbeda. Namun, penggunaan satu algoritma memiliki kelemahan dalam hal keamanan karena struktur dan

polanya yang sederhana. Pola enkripsi yang mudah diprediksi membuat algoritma ini rentan terhadap serangan *brute-force* [10].

Beberapa penelitian terkait telah dilakukan untuk meningkatkan keamanan data melalui kombinasi algoritma. Penelitian [13] mengevaluasi algoritma Rail Fence dan Vigenere dengan menggunakan metode evaluasi uji internal algortima menggunakan *avalanche effect*. Hasil penelitian menunjukkan kombinasi keduanya dapat mendeteksi perubahan kecil pada input sehingga meningkatkan keamanan data dari sisi *integrity*. Penelitian [15] mengevaluasi Rail Fence dan steganografi LSB untuk melindungi file gambar. Hasil penelitian menunjukkan kombinasi keduanya mampu mengamankan data yang tersembunyi pada gambar dan proses enkripsi dan deskripsinya relatif stabil. Penelitian [16] mengevaluasi Rail Fence dan Vernam Cipher pada perangkat IoT, menunjukkan efisiensi memori untuk proses enkripsi dan deskripsi. Penelitian [17] mengevaluasi *One Time Pad Cipher* dan transformasi Rail Fence untuk komunikasi teks, sementara penelitian [14] mengkombinasikan Vigenere dengan Caesar untuk mengamankan pesan teks. Di lain sisi, penelitian [18] menggabungkan Vigenere dan RSA guna meningkatkan keamanan sekaligus mempercepat enkripsi. Lalu penelitian [19] mengkombinasikan Vigenere dengan polybus untuk meningkatkan keamanan data. Namun, berdasarkan hasil literatur penelitian terdahulu, penelitian masih berfokus pada pengamanan data, padahal pengujian ketahanan teknik kriptografi baik internal maupun eksternal juga diperlukan. Oleh karena itu, penelitian ini memfokuskan untuk menggunakan kombinasi teknik kriptografi rail fence dan Vigenere untuk mengamankan data dan menganalisis ketahanan kombinasi teknik kriptografi tersebut menggunakan uji eksternal seperti *brute-force*.

3. Metodologi Penelitian

Penelitian ini menggunakan metode eksperimen. Metode eksperimen adalah metode yang digunakan untuk mencari pengaruh perlakuan tertentu terhadap dampaknya dalam kondisi yang terkendalikan [9]. Penelitian ini menerapkan kerangka kerja untuk menganalisis ketahanan kombinasi teknik kriptografi Rail Fence dan Vigenere cipher terhadap serangan brute-force. Environtment untuk melakukan penelitian yaitu menggunakan python. Python merupakan bahasa pemrograman tingkat tinggi yang mudah dipelajari, bersifat fleksibel, dan digunakan luas dalam berbagai bidang seperti pengembangan web, kecerdasan buatan, dan keamanan siber [20]. Kerangka kerja penelitian dapat dilihat pada Gambar 3.



Gambar 3. Kerangka Kerja Penelitian

Kerangka kerja penelitian ini dibagi menjadi tiga tahap, yang pertama melakukan analisis keamanan pada metode Rail Fence, tahap kedua analisis keamanan Vigenere, Terakhir analisis

keamanan kombinasi Rail Fence + Vigenere menggunakan teknik serangan *brute-force*. Masing-masing tahap dijelaskan pada paragraph berikut:

3.1. Analisis Keamanan Rail Fence

Pada tahap ini pesan teks akan dienkripsi menggunakan Rail Fence, dengan 3 *Rails*. Contoh pesan yang akan dienkripsi "HARIMAU". Cara kerja enkripsi Rail Fence sebagai berikut:

Plaintext: "HARIMAU"

Rails: 3

Rail 1	**H**				**M**		
Rail 2		**A**		**I**		**A**	
Rail 3			**R**				**U**

Ciphertext:

Rail 1	**H** →				**M** →		
Rail 2	→	**A**		**I**		**A** →	
Rail 3	→		**R**				**U**

Dari proses di atas didapatkan *ciphertext* untuk pesan "Harimau" yaitu "HMAIARU". Hal ini dikarenakan pembacaan teks sesuai arah panah berwarna biru. Hasil ciphertext ini nantinya akan di uji menggunakan *brute-force*. Uji serangan *brute-force* pada penelitian ini akan dilakukan menggunakan bahasa pemrograman python.

3.2. Analisis Keamanan Vigenere

Pada tahap ini pesan teks akan dienkripsi menggunakan Vigenere, dengan 3 kunci. Contoh pesan yang akan dienkripsi "KUCING". Cara kerja enkripsi Vigenere sebagai berikut:

Plaintext: "KUCING"

Kunci: MBI

Ciphertext:

Tabel 1. Proses Enkripsi menggunakan Vigenere

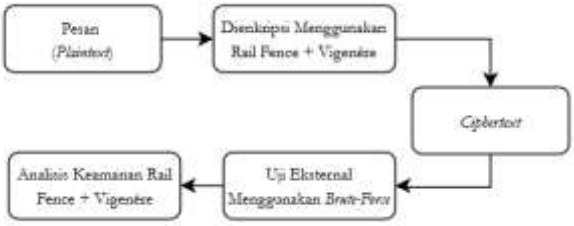
Plaintext	P (angka)	Kunci	K (angka)	Penjumlahan P+K	(P+K) mod 26	*Cipher*
K	10	M	12	10 + 12 = 22	22	W
U	20	B	1	20 + 1 = 21	21	V
C	2	I	8	2 + 8 = 10	10	K
I	8	M	12	8 + 12 = 20	20	U
N	13	B	1	13 + 1 = 14	14	O
G	6	I	8	6 + 8 = 14	14	O

Dari tabel di atas didapatkan *ciphertext* untuk pesan "Kucing" yaitu "WVKUOO". Hal ini dikarenakan setiap huruf pada *plaintext* dienkripsi dengan cara menambahkan nilai huruf kunci sesuai tabel Vigenere secara modulo 26, sehingga setiap huruf pada pesan asli mengalami

pergeseran berbeda-beda berdasarkan nilai huruf pada kuncinya. Hasil *ciphertext* ini nantinya akan di uji menggunakan *brute-force*.

3.3 Evaluasi Keamanan Rail Fence + Vigenere

Pada tahap ini pesan akan di enkripsi menggunakan kombinasi algoritma rail fence dan menggunakan algortima Vigenere. Ilustrasi tahap ini ditunjukkan pada gambar 4.



Gambar 4. Evaluasi Keamanan Rail Fence + Vigenere

Pada gambar di atas, enkripsi tidak dilakukan secara terpisah, melainkan dalam satu proses. Hasil *ciphertext* dari kombinasi teknik kemudian akan di evaluasi dengan uji eksternal menggunakan serangan *brute-force*. Hasil uji kemudian dianalisis. Uji serangan *brute-force* menggunakan bahasa pemrograman python.

4. Hasil dan Pembahasan

Dalam penelitian ini, peneliti menganalis penggunaan kombinasi teknik kriptografi Rail Fence dengan Vigenere untuk pengamanan data. Selain itu, penelitian ini juga menguji ketahanan kombinasi teknik kriptografi yang digunakan dengan uji serangan *brute-force*. Pada bagian ini peneliti akan menyajikan perbandingan hasil uji serangan *brute-force* terhadap teknik kriptogari Rail Fence, hasil uji serangan *brute-force* terhadap kombinasi Vigenere dan hasil uji serangan *brute-force* terhadap kombinasi Rail Fence + Vigenere. Pengujian ini dilakukan menggunakan laptop Asus X505Z dengan spesifikasi RAM 12GB HDD 1TB. Selain itu, pengujian dilakukan menggunakan bahasa pemrograman python versi 3.9.0 dengan editor Jupyter notebook.1.1.1. library python yang digunakan pada pengujian yaitu library *time* yang berfungsi untuk mencatat waktu secara *real-time* pada saat proses berjalan.

4.1. Uji Serangan *brute-force* terhadap Rail Fence

Pada tahap ini *plaintext* yang digunakan adalah "**Do not Spill the tea**". Dalam melakukan enkripsi menggunakan Rail Fence, *rails* yang digunakan sebanyak 3 *rails*. Proses enkripsi menggunakan bahasa pemrograman python. Adapun hasil enkripsi untuk *plaintext* di atas menggunakan Rail Fence ditunjukkan pada gambar 5.

```
--- Hasil Enkripsi ---
Plaintext (preview 200 chars):
Do not Spill the tea

Ciphertext:
Dop  ontSilteta  lhe

Waktu enkripsi: 0.018 ms
```

Gambar 5. Hasil Enkripsi menggunakan Rail Fence

Dari hasil di atas, *ciphertext* untuk pesan "Do not Spill the tea" adalah **Dop ontSilteta lhe**. Adapun waktu enkripsi yang digunakan dalam melakukan enkripsi pada metode **Rail Fence**, yaitu **0.018 *millisecond***. Hasil *ciphertext* ini kemudian di uji menggunakan uji eksternal menggunakan *brute-force*. Dalam melakukan uji serangan *brute-force* penelitian ini menggunakan bahasa pemrograman python. Hasil uji serangan *brute-force* dapat dilihat pada gambar 6.

```
Jalankan brute-force? (y/N):  y
Masukkan max rails untuk brute-force:  10000
Masukkan known plaintext (opsional, kosong untuk skip):

Menjalankan brute-force ... (akan berhenti bila plaintext ditemukan)

Brute-force selesai. Total waktu: 0.000 s (0.1 ms).
Plaintext ditemukan pada rails = 3.
Waktu sampai ditemukan: 0.000063 s (0.1 ms).

Ringkasan per-rail (urut sesuai percobaan):
 rails= 2  time=  0.039 ms  'Dlotpe t ao n tlShie'
 rails= 3  time=  0.019 ms [MATCH]  'Do not Spill the tea'

Selesai.
```

(a)

```
Jalankan brute-force? (y/N):  y
Masukkan max rails untuk brute-force:  100000
Masukkan known plaintext (opsional, kosong untuk skip):

Menjalankan brute-force ... (akan berhenti bila plaintext ditemukan)

Brute-force selesai. Total waktu: 0.000 s (0.1 ms).
Plaintext ditemukan pada rails = 3.
Waktu sampai ditemukan: 0.000060 s (0.1 ms).

Ringkasan per-rail (urut sesuai percobaan):
 rails= 2  time=  0.037 ms  'Dlotpe t ao n tlShie'
 rails= 3  time=  0.019 ms [MATCH]  'Do not Spill the tea'

Selesai.
```

(b)

Gambar 6. Hasil Uji serangan *brute-force* pada *ciphertext* Rail Fence (a) 10.000 percobaan (b) 100.000 percobaan

Berdasarkan hasil pada gambar di atas, proses *brute-force* dilakukan 10.000 percobaan dan 100.000 percobaan terhadap *ciphertext* yang dienkripsi menggunakan metode **Rail Fence** memerlukan waktu sekitar **0,1 *milisecond*** untuk melakukan dekripsi. Hal tersebut disebabkan karena jumlah *rails* pada algoritma Rail Fence hanya memiliki panjang satu huruf ('1', '2', '3'), sehingga ruang kunci (*key space*) yang harus diuji relatif kecil. Akibatnya, proses penemuan kunci melalui serangan *brute-force* dapat dilakukan dengan lebih cepat.

4.2. Uji Serangan *brute-force* terhadap Vigenere

Pada tahap ini *plaintext* yang digunakan adalah "**Do not Spill the tea**". Dalam melakukan enkripsi menggunakan Vigenere, *key* yang digunakan sebanyak 3 *key* yaitu "**gxt**". Proses enkripsi menggunakan bahasa pemrograman python. Adapun hasil enkripsi untuk *plaintext* di atas menggunakan Vigenere ditunjukkan pada gambar 7.

```
=== HASIL ENKRIPSI VIGENERE ===
Plaintext   : Do not Spill the tea
Key         : gxt
Ciphertext  : Jl guq Lvfer qak qxg
Waktu proses (enkripsi): 0.0000000000 detik
```

Gambar 7. Hasil Enkripsi Menggunakan Vigenere

Dari hasil di atas, *ciphertext* untuk pesan "Do not Spill the tea" adalah **Jl guq Lvfer qak qxg**. Adapun waktu enkripsi yang digunakan dalam melakukan enkripsi pada metode **Vigenere**, yaitu **0.000 *millisecond***. Hasil *ciphertext* ini kemudian di uji menggunakan uji eksternal menggunakan *brute-force*. Dalam melakukan uji serangan *brute-force* penelitian ini menggunakan bahasa pemrograman python. Hasil uji serangan *brute-force* dapat dilihat pada gambar 8.

```
Jalankan brute-force untuk menemukan key? (y/n):  y
Masukkan panjang kunci yang ingin dicoba (L, contoh 3):  3
Masukkan timeout dalam detik (kosongkan untuk tanpa timeout):
Masukkan max_attempts (kosongkan untuk tanpa limit):  10000

Menjalankan brute-force (L=3) ...

=== HASIL BRUTE-FORCE ===
Catatan : found_exact_match
Found    : True
Key      : GXT
Attempts : 4,674
Time     : 0.057908 detik
Throughput (candidates/sec): 80714.52

Recovered plaintext : Do not Spill the tea
Recovered stats -> 5 kata, 16 huruf
```

(a)

```
Jalankan brute-force untuk menemukan key? (y/n):  y
Masukkan panjang kunci yang ingin dicoba (L, contoh 3):  3
Masukkan timeout dalam detik (kosongkan untuk tanpa timeout):
Masukkan max_attempts (kosongkan untuk tanpa limit):  100000

Menjalankan brute-force (L=3) ...

=== HASIL BRUTE-FORCE ===
Catatan : found_exact_match
Found    : True
Key      : GXT
Attempts : 4,674
Time     : 0.061759 detik
Throughput (candidates/sec): 75681.52

Recovered plaintext : Do not Spill the tea
Recovered stats -> 5 kata, 16 huruf
```

(b)

Gambar 8. Hasil Uji serangan *brute-force* pada *ciphertext* Vigenere (a) 10.000 percobaan (b) 100.000 percobaan

Berdasarkan hasil yang ditunjukkan pada gambar di atas, proses **brute-force** terhadap *ciphertext* yang dienkripsi dengan algoritma **Vigenere** memerlukan waktu sekitar **0.058**

milisecond untuk 10.000 percobaan dan **0,062 *milisecond*** untuk 100.000 percobaan dalam menyelesaikan dekripsi. Waktu tersebut menunjukkan bahwa kompleksitas pencarian kunci pada kasus ini relatif rendah, sehingga proses dekripsi dapat dilakukan dengan sangat cepat. Walaupun demikian, proses penemuan kunci melalui serangan *brute-force* pada Vigenere sedikit lebih lama dibandingkan dengan metode Rail Fence. Hal ini dikarenakan proses enkripsi dan dekripsi pada algoritma Vigenere melibatkan operasi substitusi berdasarkan nilai kunci huruf demi huruf, sedangkan pada metode Rail Fence hanya melibatkan proses transposisi posisi karakter tanpa perhitungan numerik tambahan.

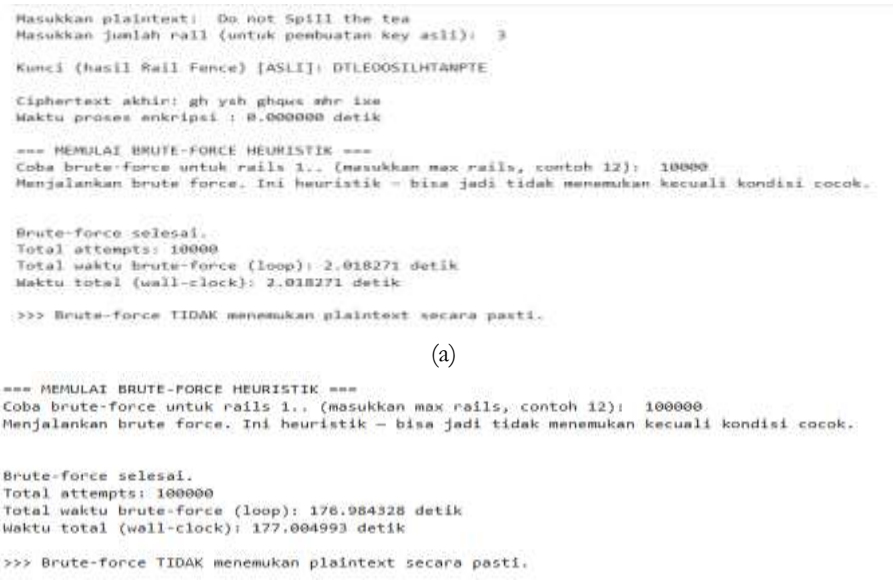
4.3. Uji Serangan *brute-force* terhadap Rail Fence + Vigenere

Pada tahap ini *plaintext* yang digunakan adalah "**Do not Spill the tea**". Dalam melakukan enkripsi menggunakan kombinasi teknik kriptografi Rail Fence + Vigenere, *rails* untuk Rail Fence menggunakan 3 *rails* sedangkan *key* yang digunakan untuk enkripsi Vigenere merupakan hasil dari enkripsi Rail Fence. Proses enkripsi menggunakan bahasa pemrograman python. Adapun hasil enkripsi untuk *plaintext* di atas menggunakan kombinasi Rail Fence + Vigenere ditunjukkan pada gambar 9.

```
Masukkan plaintext:  Do not Spill the tea
Masukkan jumlah rail:  3
Kunci (hasil Rail Fence): DTLEOOSILHTANPTE
Ciphertext akhir: gh ysh ghqws mhr ixe
Waktu proses enkripsi : 0.000000 detik
```

Gambar 9. Hasil Uji serangan *brute-force* pada *ciphertext* Vigenere

Dari hasil di atas, *ciphertext* untuk pesan "Do not Spill the tea" adalah **gh ysh ghqws mhr ixe**. Adapun waktu enkripsi yang digunakan dalam melakukan enkripsi pada metode kombinasi **Rail Fence + Vigenere**, yaitu **0.000 *millisecond***. Dari gambar di atas, kunci yang digunakan untuk enkripsi vigenere diambil dari hasil enkripsi Rail Fence. Hasil akhir *ciphertext* dari kombinasi ini kemudian di uji menggunakan uji eksternal menggunakan *brute-force*. Dalam melakukan uji serangan *brute-force* penelitian ini menggunakan bahasa pemrograman python. Hasil uji serangan *brute-force* dapat dilihat pada gambar 10.



Gambar 10. Hasil Uji serangan *brute-force* pada *ciphertext* kombinasi Rail Fence + Vigenere (a) 10.000 percobaan, (b) 100.000 percobaan

Berdasarkan hasil pengujian yang ditunjukkan pada gambar di atas, proses *brute-force* dilakukan sebanyak 10.000 percobaan dengan waktu **2,02 *milisecond*** dan 100.000 percobaan dengan waktu **177 *milisecond*** terhadap ciphertext yang dihasilkan dari algoritma kombinasi **Rail Fence + Vigenere**. Proses pencarian kunci masih tergolong cepat dari sisi komputasi. Namun demikian, hasil pengujian menunjukkan bahwa *brute-force* tidak berhasil menemukan *plaintext* asli, meskipun seluruh ruang pencarian kunci hingga batas tertentu telah diuji. Hal ini mengindikasikan bahwa penggabungan dua algoritma klasik, yaitu transposisi (Rail Fence) dan substitusi (Vigenere), berhasil meningkatkan tingkat kerumitan struktur *ciphertext* yang dihasilkan. Kombinasi tersebut menghasilkan ruang kunci yang lebih luas dan pola enkripsi yang lebih sulit dianalisis secara langsung oleh serangan *brute-force* konvensional. Dengan demikian, peluang penyerang untuk menebak kunci secara acak tanpa informasi tambahan menjadi sangat kecil. Meskipun waktu komputasi *brute-force* relatif singkat, hal ini tidak berbanding lurus dengan keberhasilan dekripsi. Kecepatan proses tidak menjamin efektivitas serangan, karena algoritma kombinasi ini memiliki mekanisme pengacakan karakter yang berlapis. Kombinasi Rail Fence + Vigenere terbukti lebih tahan terhadap serangan *brute-force* sederhana.

Berdasarkan hasil enkripsi Rail Fence, Vigenere, dan Kombinasi Rail Fence + Vigenere di atas, adapun analisis perbandingan hasil uji serangan *brute-force* terhadap ketiganya dapat dilihat pada tabel 2.

Tabel 2. Analisis Perbandingan hasil uji serangan *brute-force* Rail Fence, Vigenere, dan Rail Fence + Vigenere

No.	Metode Kriptografi	Waktu yang dibutuhkan untuk enkripsi	Waktu yang dibutuhkan Serangan *brute-*	Waktu yang dibutuhkan Serangan	Analisis Keamanan Metode Kriptografi

			***force* = 10.000 percobaan**	***brute-force* = 100.000 percobaan**	
1	Rail Fence	0.018 *milisecond*	0,1 *milisecond*	0,1 *milisecond*	Jumlah *rails* pada algoritma Rail Fence hanya memiliki panjang satu huruf ('1', '2', '3'), sehingga ruang kunci (*key space*) yang harus diuji relatif kecil. Akibatnya, proses penemuan kunci melalui serangan *brute-force* dapat dilakukan dengan lebih cepat.
2	Vigenere	0.000 *milisecond*	0,057 *milisecond*	0.062 *milisecond*	Proses penemuan kunci melalui serangan *brute-force* pada Vigenere sedikit lebih lama dibandingkan dengan metode Rail Fence. Hal ini dikarenakan proses enkripsi dan dekripsi pada algoritma Vigenere melibatkan operasi substitusi berdasarkan nilai kunci huruf demi huruf, sedangkan pada metode Rail Fence hanya melibatkan proses transposisi posisi karakter tanpa perhitungan numerik tambahan.
3	Rail Fence + Vigenere	0.000 *milisecond*	2,02 *milisecond*	177 *milisecond*	Proses brute-force menjadi tidak efektif karena

	ruang kunci (key space) yang harus diuji semakin besar akibat penggabungan dua tahapan enkripsi, yaitu pembangkitan kunci melalui Rail Fence dan proses substitusi pada Vigenere.

5. Kesimpulan

Algoritma kombinasi Rail Fence + Vigenere memiliki tingkat keamanan yang lebih tinggi dibandingkan penggunaan algoritma tunggal. Hal ini dibuktikan melalui uji *brute-force* sebanyak 10.000 percobaan hingga 100.000 percobaan yang **tidak berhasil** menemukan *plaintext* asli. Kombinasi kedua algoritma tersebut menghasilkan *ciphertext* dengan pola yang lebih kompleks dan ruang kunci yang lebih luas, sehingga menyulitkan proses penebakan kunci secara acak. Selain itu, kombinasi Rail Fence + Vigenere efektif dalam meningkatkan resistansi terhadap serangan *brute-force* sederhana. Namun, untuk penerapan pada sistem keamanan modern, kombinasi ini masih perlu dikembangkan lebih lanjut agar mampu menghadapi bentuk serangan kriptanalisis yang lebih kompleks, seperti *frequency analysis* atau *known-plaintext attack*.

6. Pernyataan Penulis

Penulis menyatakan bahwa tidak ada konflik kepentingan terkait publikasi artikel ini. Penulis menyatakan bahwa data dan makalah bebas dari plagiarisme serta penulis bertanggung jawab secara penuh atas keaslian artikel.

Daftar Pustaka

[1] A. Renaldy *et al.*, "Peran Sistem Informasi dan Teknologi Informasi Terhadap Peningkatan Keamanan Informasi Perusahaan," *J. Ilmu Multidisplin*, vol. 2, no. 1, pp. 15–22, 2023, doi: 10.38035/jim.v2i1.212.

[2] R. Vansuri *et al.*, "Peran CIA (Confidentiality, Integrity, Availability) Terhadap Manajemen Keamanan Informasi," *J. Ilmu Multidisplin*, vol. 2, no. 1, pp. 106–113, 2023, doi: 10.38035/jim.v2i1.234.

[3] K. Gita Segara and M. Irwan Padli Nasution, "Perkembangan Teknologi Informasi di Indonesia: Tantangan dan Peluang," *J. Sains Student Res.*, vol. 3, no. 1, pp. 21–33, 2025, [Online]. Available: https://doi.org/10.61722/jssr.v3i1.3128

[4] Y. Pratama and T. Sutabri, "Analisis Kriptografi Algoritma Blowfish pada Keamanan Data menggunakan Dart," *J. Inform. Terpadu*, vol. 9, no. 2, pp. 126–135, 2023, doi: 10.54914/jit.v9i2.975.

[5] N. Wulan, H. Harahap, and Y. S. Siregar, "Watermarking Citra Digital DFT Dan Kriptografi Algoritma RSA Pada Sistem Berbasis Web," *Pros. Semin. Nas. Teknol. Inov. dan Kolaborasi Disiplin Ilmu*, vol. 1, no. 1, pp. 244–255, 2024.

[6] I. Riandi, A. Fadlil, and F. Auliya Tsani, "Pengamanan Citra Digital Berbasis Kriptografi Menggunakan Algoritma Vigenere Cipher," *J. Inform. Sunan Kalijaga*, vol. 7, no. 1, pp. 33–45, 2022.

[7] A. P. U. Siahaan, "Pengamanan Pesan Teks Dengan Teknik Transposisi Karakter," *Escaf*, 2023, [Online]. Available: https://semnas.univbinainsan.ac.id/index.php/escaf/article/view/456%0Ahttps://semnas.univbinainsan.ac.id/index.php/escaf/article/download/456/282

[8] N. P. E. Merliana, "Pemanfaatan Teknologi Kriptografi dalam mengatasi kejahatan Cyber," *Satya Dharma J. Ilmu Huk.*, vol. 3, no. 2, pp. 23–40, 2020, [Online]. Available: https://ejournal.iahntp.ac.id/index.php/satya-dharma/article/view/678

[9] D. A. P. Putri and A. Rachmawati, "Honeypot cowrie implementation to protect ssh protocol in ubuntu server with visualisation using kippo-graph," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 8, no. 6, pp. 3200–3207, 2019, doi: 10.30534/ijatcse/2019/86862019.

[10] A. Fauzi and S. Syahputra, "A Combination Of A Rail Fence Cipher And Merkle Hellman Algorithm For Digital Image Security," vol. 2, no. 3.

[11] A. Syarif, "Modifikasi Caesar Cipher dengan Permutasi, Transposisi, Binary, Gerbang Logika, ASCII Dan HEXA," *J. Teknol. Inf.*, vol. 4, no. 2, pp. 234–240, 2020, doi: 10.36294/jurti.v4i2.1350.

[12] Nurdayati dkk, "MODIFIKASI RAIL FENCE TRANSPOSITION CIPHER DENGAN CHESS BOARD PATTERN," 2021.

[13] N. Syah and E. Ardhianto, "Meningkatkan Keamanan Data menggunakan Super Enkripsi Kombinasi Rail Fence dan Vigenere Autokey," *J. Ilm. Komputasi*, vol. 23, no. 3, pp. 293–300, 2024, doi: 10.32409/jikstik.23.3.3612.

[14] V. M. Hidayah, D. I. Mulyana, and Y. Bachtiar, "Algoritma Caesar Cipher atau Vigenere Cipher pada Pengenkripsian Pesan Teks," vol. 05, no. 03, pp. 8563–8573, 2023.

[15] D. Rachmawati, M. A. Budiman, and A. Yusuf, "Combination of Rail Fence Cipher Algorithm and Least Significant Bit Technique to Secure the Image File," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 851, no. 1, 2020, doi: 10.1088/1757-899X/851/1/012069.

[16] A. A. Bitar and D. V. Sujatha, "Merging Vernam Cipher stream and Rail ?Fence Algorithms and How Effective They are on Internet of Things Devices," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, vol. 3307, pp. 686–691, 2021, doi: 10.32628/cseit2173149.

[17] F. Adji S, "IMPLEMENTASI ALGORITMA ONE TIME PAD CIPHER DAN TRANSFORMASI RAIL FENCE CIPHER PADA PESAN TEKS," 2020.

[18] Jamaludin and Romindo, "Implementation of Combination Vigenere Cipher and RSA in Hybrid Cryptosystem for Text Security," *Int. J. Inf. Syst. Technol. Akreditasi*, vol. 4, no. 1, pp. 471–481, 2020.

[19] S. Vatshayan, R. A. Haidri, and J. K. Verma, "Design of Hybrid Cryptography System based on Vigenère Cipher and Polybius Cipher," *2020 Int. Conf. Comput. Perform. Eval. ComPE 2020*, pp. 848–852, 2020, doi: 10.1109/ComPE49325.2020.9199997.

[20] M. H. Maulana, "Python Bahasa Pemograman Yang Ramah Bagi Pemula," *JISCO (Journal Inf. Syst. Comput.*, vol. 2, pp. 73–78, 2024.