IMPLEMENTASI ALGORITMA KRIPTOGRAFI *CAESAR CIPHER* DAN *RAIL FENCE CIPHER* UNTUK KEAMANAN DATA TEKS MENGGUNAKAN PYTHON

Dewi Purnamasari
Universitas Ivet
dewi.poernamasari.09@gmail.com

ABSTRAK

Keamanan dan kerahasiaan dalam berkomunikasi menjadi suatu kebutuhan agar informasi yang dikirim dan diterima tidak disalahgunakan oleh pihak-pihak yang kurang bertanggung jawab. Kriptografi tidak hanya menyediakan alat untuk keamanan informasi tetapi juga merupakan teknik yang berguna untuk kemananan dan kerahasiaan informasi. Pada penelitian ini bertujuan mengimplementasikan algoritma *Caesar Cipher* dan *Rail Fence Cipher* untuk mengetahui proses penyandian dan juga untuk mengetahui efektifitas waktu enkripsi dan deskripsi dari kedua teknik kriptografi tersebut. Waktu proses kedua metode tersebut menunjukkan bahwa waktu deskripsi lebih lama dibandingkan proses enkripsi. Waktu rata- rata eksekusi algoritma *Caesar Cipher* lebih cepat dibandingkan dengan *Rail Fence Cipher*. Waktu rata-rata enkripsi *Caesar Cipher* adalah 0.0000802 detik dan waktu deskripsi adalah 0.000131 detik meskipun demikian *Rail Fence* lebih aman karena proses teks yang dikirim bentuk *zig zag*, maka *ciphertext* yang diterima lebih aman dibandingkan dengan *Caesar Cipher*. Pada Caesar Cipher lebih cepat karena proses penyandian huruf digeser sebanyak kunci yang diinginkan.
Kata kunci: *Caesar Cipher, Rail Fence Cipher, Key, ziq zaq, Plaintext, Chipertext*

ABSTRACT

Security and confidentiality in communicating become a necessity so that the information sent and received is not misused by irresponsible parties. Cryptography not only provides a tool for information security but is also a useful technique for the security and confidentiality of information. This study aims to implement the Caesar Cipher and Rail Fence Cipher algorithms to determine the encoding process and also to determine the effectiveness of the encryption time and description of the two cryptographic techniques. The processing time of the two methods shows that the description time is longer than the encryption process. The average execution time of the Caesar Cipher algorithm is faster than the Rail Fence Cipher. The average encryption time is 0.0000802 second and the description time is 0.000131 second, however Rail Fence is more secure because the text process is sent in a zig zag form, so the received ciphertext is more secure than Caesar Cipher. Caesar Cipher is faster because the letter encoding process is shifted as many keys as desired.
Keywords: *Caesar Cipher, Rail Fence Cipher, Key, ziq zaq, Plaintext, Chipertext*

PENDAHULUAN

Kerahasiaan suatu data atau keamanan sebuah informasi merupakan aspek penting dari berbagai informasi yang diterima maupun dikirim oleh pengguna. Perkembangan teknologi yang sangat cepat memberikan kemudahan bagi pengguna untuk memperoleh data atau informasi dengan sangat mudah. Apabila data tidak dilindungi maka orang lain dapat dengan

mudah mengambil data atau informasi yang dimiliki seseorang. Kriptografi adalah sebuah metode untuk melindungi data atau informasi menggunakan sandi, dimana sandi tersebut hanya dimengerti oleh orang yang berhak menerima data atau informasi tersebut [1]. Banyak oknum yang tidak bertanggung jawab (kriptanalis) kemudian mencari celah keamanan dalam jaringan untuk mengacaukan sirkulasi atau peredaran data. Hal tersebut yang kemudian mendorong perlu digunakan teknologi Kriptografi. Tujuan kriptografi adalah melindungi data atau informasi dari ancaman baik yang disengaja maupun tidak disengaja dari oknum yang tidak bertanggung jawab.

Ada banyak sekali jenis kriptografi yang dikenal saat ini salah satunya adalah kriptografi klasik. Kriptografi klasik memiliki metode diantaranya *Caesar Cipher* dan *Rail Fence Cipher*. Kedua metode ini dapat dimanfaatkan dalam membangun sistem keamanan kriptografi yang baik.

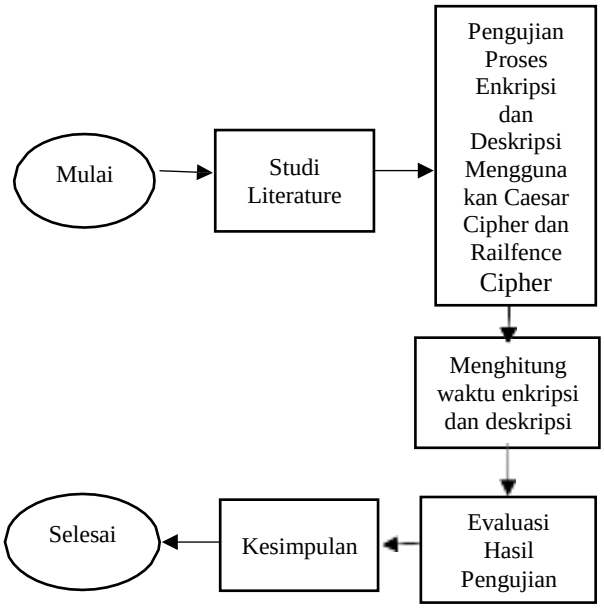
Beberapa penelitian terdahulu yang relevan dengan yang dilakukan oleh penulis antara lain seperti algoritma *Caesar Cipher* menggunakan Matlab yang diterapkan bertujuan untuk mengetahui proses enkripsi dan deskripsi [2]. Enas Ismael Imran dan Farah Abdulameerabdulkareem pada penelitiannya tentang peningkataan *Caesar Cipher* untuk keamanan lebih baik menunjukkan bahwa *Caesar Cipher* menjadi salah satu algoritma untuk mengenkripsi yang paling sederhana dan banyak teknik yang dapat digunakan untuk memperkuat bahkan melebihi apa yang dicapai algoritma *Caesar Cipher* [3]. Menurut penelitian Kusumaningtyas membahas tentang Analisa Algoritma *Ciphers Transposition*: Studi Litearure. Hasil penelitiannya menunjukkan bahwa Algoritma *Rail Fence Cipher* mempunyai kelebihan dalam kerumitan cipherteks yang dihasilkan [4]. Yang membedakan penelitian penulis dengan penelitian sebelumnya adalah penelitian sebelumnya tidak menggunakan efektivitas waktu untuk mengetahui kinerja kriptografi *Caesar Cipher* dan *Rail Fence Cipher* dan penelitian penulis menggunakan Bahasa Python.

Dari permasalahan dan penelitian sebelumnya dapat diambil rumusan masalah Bagaimana mengimplementasikan *Caesar cipher* dan *Rail Fence Cipher* untuk kemananan data teks.

Penelitian ini bertujuan untuk mengetahui implementasi dan efektivitas waktu enkripsi dan diskripsi untuk metode *Caesar Cipher* dan *Rail Fence Cipher* yang digunakan untuk keamanan data teks.

METODE PENELITIAN

Penelitian yang dilakukan merupakan penelitian yang bersifat studi literatur dan metode eksperimen *Caesar Cipher* dan *Railfence Cipher*. Adapun alur penelitian yang digunakan dapat dilihat pada Gambar 1.



Gambar 1. Alur Penelitian

Tahapan penelitian yang dilakukan adalah:

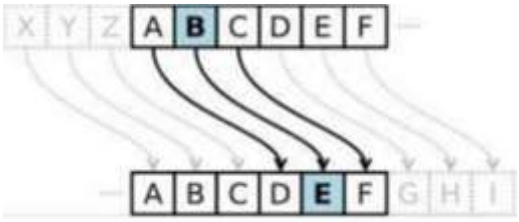
1. Studi Literature
   Pada tahap ini dilakukan peninjauan terhadap buku, artikel jurnal maupun

hasil penelitian terdahulu sebagai referensi yang diperlukan dalam melakukan penelitian. Ini dilakukan untuk memperoleh informasi terkait dengan operasi *Caesar Cipher* dan *Railfence Cipher* serta pemrograman Python yang digunakan untuk pemrograman menguji enkripsi dan deskripsi.

2. Proses Enkripsi dan Deskripsi menggunakan dua algoritma *Caesar Cipher* dan *Railfence Cipher* dengan menggunakan K=3 dan K=4.

Caesar Cipher

Metode penyandian dalam kriptografi klasik yang paling terkenal adalah *Caesar Cipher*. Teknik kriptografi ini merupakan kriptografi yang paling sedrhana dan banyak digunakan diaman setiap huruf pada plainteksnya digantikan dengan huruf lain dengan pergeseran sebanyak nilai kunci [5-6]. Dalam penelitian ini nilai kunci dari *Caesar Cipher* adalah 3 dan 4. Untuk proses pergeseran karakter di *Caesar Cipher* dapat dilihat pada Gambar 2.



Gambar 2. Proses Pergeseran dalam *Caesar*

Contoh:
Plaintext : Universitas Ivet
Kunci 3
Ciphertext : qlyhuvlwdvcLyhw

Rail Fence Cipher

Merupakan salah satu algoritma *cipher transposisi* yang mengacak urutan huruf-huruf pesan. Algoritma ini melibatkan penulisan *plainteks* ke bawah secara berturut turut yang memiliki baris atas dan baris bawah. Sedangkan *ciphertext* nya diperoleh dengan membaca hutruf berdasarkan baris.

Algoritma *Rail Fence Cipher* menyusun plainteks secara *ziq-zag* dengan turun kebawah dan naik keatas sesuai ukuran kolom dan baris yang ditentukan oleh *key* [7].

Contoh:

*Plaintex*t : Universitas Ivet
Kunci 3
Ciphertext: UetInvria vtisse

U				e				t				v		
	n		v		r		i		a		I		e	
		i				s				s				t

Gambar 3. *Zig Za*g Kriptografi *Rail Fence*

3. Menghitung waktu enkripsi dan deskripsi dengan Python adalah dengan mengurangkan waktu berhenti proses dikurangi dengan waktu mulai proses.
4. Evaluasi hasil pengujian dengan mengimplementasikan dan membandingkan hasil kebenaran metode *Caesar Cipher* dan *Rail Fence Cipher* dan juga membanding waktu proses enkripsi dan deskripsi.

HASIL DAN PEMBAHASAN

Pada proses ini akan dilakukan pengujian dengan 2 sampel plainteks Universitas Ivet dan Universitas Ivet Unisvet dengan kunci yang berbeda K1=3 dan K2=4, tujuannya adalah untuk membandingkan efektivitas waktu yang digunakan masing masing teknik kriptografi tersebut. Hasil pengujian enkripsi dan deskripsi ini dapat dilihat pada Tabel 1 dan Tabel 2 berikut ini:

Tabel 1. Enkripsi dan Deskripsi *Caesar Cipher*

PLAIN TEXT	K	ENKRIPSI	*TIME* (detik)	DES KRIPSI	*TIME* (detik)
Universitas Ivet	3	XqlyhuvlwdvcLyhw	0.0000632	Universitas Ivet	0.0000859
Universitas Ivet	4	YrmzivwmxewdMzix	0.0001263	Universitas Ivet	0.0002615
Universitas Ivet Unisvet	3	XqlyhuvlwdvcLyhwcXqlvyhw	0.0000625	Universitas Ivet Unisvet	0.0000812
Universitas Ivet Unisvet	4	YrmzivmxewdMzixdYrmwzix	0.0000689	Universitas Ivet Unisvet	0.0000964

Tabel 1 pada *Plaintext* Universitas Ivet dengan Key 3 dan Key 4 menggunakan algoritma *Caesar Cipher* menunjukkan bahwa waktu enkripsi lebih cepat dibanding deskripsi. Proses enkripsi menunjukkan bahwa semakin kunci ditambah dari K=3 ke K=4 maka waktu eksekusi enkripsi menjadi semakin lama. Begitu juga sama halnya di waktu eksekusi deskripsi. Jika panjang karakter *plaintext* ditambah dari Universitas Ivet menjadi Universitas Ivet Unisvet dengan K=3 waktu 0.0000632 detik menjadi 0.0000625 detik, maka waktu eksekusi nya di enkripsi maupun deskripsi menjadi lebih cepat. Waktu rata-rata enkripsi di Caesar Cipher adalah 0.0000802 detik sedangkan deskripsi adalah 0.000131 detik. Hal ini menunjukkan bahwa waktu enkripsi lebih cepat dari deskripsi.

Tabel 2. Enkripsi dan Deskripsi *Rail Fence Cipher*

PLAIN TEXT	K	ENKRIPSI	*TIME* (detik)	DES KRIPSI	*TIME* (detik)
Universitas Ivet	3	UetInvria vtisse	0.0000750	Universitas Ivet	0.000130
Universitas Ivet	4	UsInri Uvietsevat	0.0000630	Universitas Ivet	0.000143
Universitas Ivet Unisvet	3	XUetI snvria vtUivtissne	0.0001640	Universitas Ivet Unisvet	0.000283
Universitas Ivet Unisvet	4	UsInnri vUitietse sevatv	0.0000794	Universitas Ivet Unisvet	0.0000975

Tabel 2 dengan Kriptografi *Rail Fence* menunjukkan bahwa waktu eksekusi enkripsi Proses enkripsi menunjukkan bahwa semakin kunci ditambah dari K=3 ke K=4 maka waktu eksekusi enkripsi menjadi semakin cepat berkebalikan dengan *Caesar Cipher*. Waktu rata-rata enkripsi di *Rail Fence Cipher* adalah 0.000087 detik sedangkan deskripsi adalah 0.0006535 detik. Hal ini menunjukkan bahwa waktu enkripsi dan deskripsi *Caesar Cipher* lebih cepat dari deskripsi.
Berikut adalah *source code* kriptografi *Caesar Cipher* dapat dilihat dibawah ini:

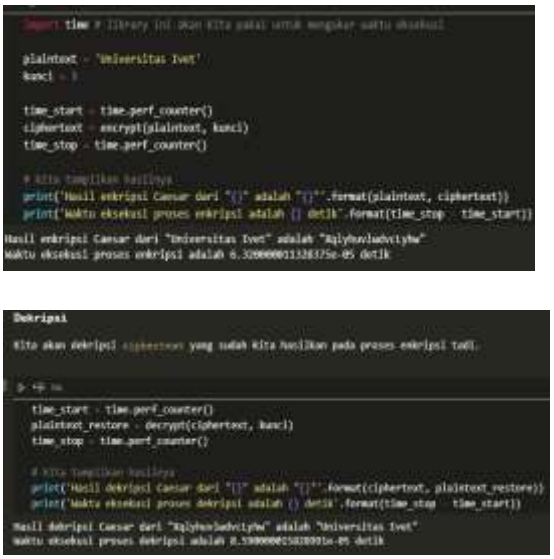
```
def encDec(alphabet, key, text, isEncrypt):
    ans = ""
    if not isinstance(key, int):
        raise Exception("Key must be integer")

    for char in text:
        try:
            alphIndex = alphabet.index(char)
        except ValueError:
            wrchar = char.encode('utf-8')
            raise Exception("Can't find char '" + wrchar + "' of text in alphabet!")
        alphIndex = (alphIndex + isEncrypt * key) % len(alphabet)
        ans += alphabet[alphIndex]
    return ans
```

Pada pemrograman Python yang digunakan penulis menggunakan 4 parameter: *alphabet, key, text, IsEncrypt*. *Alphabet* adalah *alphabet* yang dikenali oleh *Cipher* ini. *Key* adalah kata kunci. *Text* adalah *plaintext/ciphertext* tergantung dari konteksnya.

Encrypt adalah fungsi ini hanya memanggil fungsi__*encDec* dengan isi parameter *isEncrypt* adalah 1.Nilaiparameter *alphabet*, secara default ,di isi dengan "abcdefghijklmnopqrstuvwxyzABC DEFGHIJKLMNOPQRSTUVWXYZ".
" untuk memudahkan pemanggilan fungsi, sehingga user tidak perlu mengisinya (kecuali jika memang menginginkan proses enkripsi hanya menggunakan alfabet tertentu saja).

Fungsi *decrypt* adalah fungsi ini hanya memanggil fungsi__encDec dengan isi parameter *isEncrypt* adalah -1. Jadi bisa dikatakan bahwa *decrypt adalah kebalikan* dari *encrypt* . Program Python dapat dilihat pada Gambar 4 mengenai pengujian enkripsi dan deskripsi , dan juga untuk menguji waktu eksekusi dari enkripsi dan diskripsi.



Gambar 4. Pengujian Enkripsi dan Deskripsi *Caesar Cipher* Menggunakan *Python*

Pada Gambar 4 menunjukkan contoh pengujian *Caesar Cipher* dengan Bahasa pemrogramana Python digunakan untuk mengetahui waktu ekeskusi proses dan hasil enkripsi dan deskripsi. Waktu eksekusi didapatkan dari *time stop* dikurangkan dengan *time start*. Maksudnya adalah waktu berhenti proses dikurangi waktu ketika proses dimulai dimana waktu mulai memasukkan *plaintext* dan kunci.

Pertama dengan memasukkan *plaintext* Universitas Ivet, kunci=3.maka hasil enkripsi (*ciphertext*) adalah XqlyhuvlwdvcLyhw dengan waktu eksekusi 8.69 e-05 detik. *Ciphertext* adalah pesan yang tidak bisa dibaca supaya pihak kriptanalis tidak mengetahui pesan yang dikirim. *Plaintext* adalah pesan yang bisa dibaca. Pesan ini adalah pesan yang dikirim. Dengan *ciphertext* yang dihasilkan XqlyhuvlwdvcLyhw membuat kriptanalis tidak mengetahu pesan tersebut. Sedangkan proses deskripsi adalah mengubah *ciph*ertext XqlyhuvlwdvcLyhw menjadi *plaintext*. Sisi penerima mengetahui bahwa pesan yang dikirim adalah *Plaintext* Universitas Ivet dengan waktu eksekusi 0.0000632 detik. Proses enkripsi dan deskripsi *Caesar Cipher* dengan K=3 adalah sama hanya selisih sedikit sama sama di waktu 0.0000859 detik.

Berikut adalah *source code* kriptografi *Rail Fence* dapat dilihat dibawah ini:

```
def encDec(key, text, isEncrypt):
    ans = ""
    if not isinstance(key, int):
        raise Exception("Key must b
e integer")

    if isEncrypt == 1:
        # implementasi algoritma en
kripsi
        r = list(range(key))
        p = cycle(r + r[-2:0:-1])
        ans = ''.join(sorted(text,
key=lambda i: next(p)))
    elif isEncrypt == -1:
        # implementasi algoritma de
kripsi
        r = list(range(key))
        p = cycle(r + r[-2:0:-1])
```

```
        idx = sorted(range(len(text
)), key=lambda i: next(p))
        res = [''] * len(text)
        for i, c in zip(idx, text):
            res[i] = c
        ans = ''.join(res)

    return ans
```

Pada *Rail Fence Cipher* mempunyai 3 fungsi: yang pertama adalah implementasi algoritma umum dari *cipher Rail Fence* (nama fungsi:_encDec), fungsi ini akan dipanggil oleh fungsi enkripsi dan dekripsi, kedua adalah fungsi enkripsi *(encrypt)* merupakan fungsi yang dipanggil jika user ingin mengenkripsi suatu *plaintext*, ketiga adalah fungsi dekripsi *(decrypt):* merupakan fungsi yang dipanggil jika user ingin mendekripsi suatu *ciphertext.*

_encDec adalah fungsi yang memiliki tiga parameter: *key*: ini adalah kata kunci yang akan dipakai untuk enkripsi/dekripsi, text: *plaintext/ciphertext* tergantung konteks kegiatannya, *isEncrypt*: ini untuk membedakan antara kegiatan enkripsi dengan dekripsi. Bernilai 1 jika enkripsi dan -1 jika dekripsi.

PENUTUP

Kesimpulan dari penelitian ini masih sebatas proses penerapan enkripsi dan deskripsi terhadap *plaintext* yang digunakan, dengan adanya keterbatasan dalam pengujian yang lebih efeketif. Penggunaan algoritma *Caesar Cipher* masih sangat berguna terhadap eksperimen yang lebih praktis terhadap informasi yang memadai. Waktu eksekusi *Caesar Cipher* lebih cepat dibanding *Rail Fence Cipher*. Tingkat keamanan lebih aman *Rail Fence Cipher*. Saran untuk penelitian selanjutnya adalah perlu adanya kombinasi teknik kriptografi klasik dengan modern dan pemakaian kunci yang berlapis supaya data tidak dapat dicuri oleh kriptanalis.

DAFTAR PUSTAKA

[1] Saputra, M. 2021 Penerapan Kombinasi Algoritma Caesar Cipher pada Block Acak dan Cipher Transposisi dalam Mengamankan Pesan. Journal Of Information Technology.1 (1), 22-28.

[2]J.SinghandS.S.Yadav.2014.Implementation of Caesar Cipher and Chaotic Neural Network by using MATLAB Simulator. International Journal of Recent development in Engineering and Technology.2 (6), 2347-6435.

[3] P. E. Ismael Imran and P. F. abdulameerabdulkareem. 2014. Enhancement Caesar Cipher for Better Security IOSR J.Comput.Eng.16 (3), 01-05.

[4] J. A. Kusumaningtyas.2018. Analisa Algoritma Ciphers Transposition: Study Literature. Multimatrix.1 (1), 1-12.

[5] Y. Dwi Putri, R. Rosihan, and S. Lutfi, 2019. Penerapan Kriptografi Caesar Cipher Pada Fitur Chatting Sistem Informasi Freelance. *JIKO (Jurnal Inform. dan Komputer)*.2 (2). 97-94.

[6] S. Y. Wulandari. 2020.Cryptography: A Combination of Caesar and Affine Cipher to Conceal the Message. *PROC. INTERNAT. CONF. SCI. ENGIN*.3 (1), 741-744.

[7] Nardiati. D.G. 2019. Kombinasi Algoritma Kriptografi Transposisi rail Fence Cipher dan Route Cipher. *Prosiding Seminar Nasional Teknologi Informatika*. Politeknik Negeri Medan: Teknik Informatika.

[8] R.Feraldi. 2021.Kombinasi algoritma Kriptografi Caesar Cipher dan Permutation Cipher Untuk Pesan Teks Menggunakan Python. Riau Journal of Computer Science.7(01).76-86.

.

[9] J. A. Dar and S. Sharma. 2014. Implementation of One Time Pad Cipher with Rail Fence and Simple Columnar Transposition Cipher, for Achieving Data Security. *Int. J. Sci. Res*.3 (11), 2415-2421

[10] Mesran. 2020.Peningkatan Keamanan Kriptografi Caesar Cipher dengan Menerapakn Algoritma Kompres"Stout Codes". *Jurnal Rekayasa Sistem dan teknologi Informasi*.4 (6), 1209-1215.

[11] I. Gunawan. 2018.Cryptography: Kombinasi Algoritma Caesar Cipher dan Algoritma RSA untuk Pengamanan File Dokumen dan Pesan. *InfoTekJar (Jurnal Nas. Inform. dan Teknol. Jaringan*.2 (2), 124-129.

[12] Rusmala.2019.Implementasi Rail Fence Chiper dan Row Transposition Cipher Pada Mata Kuliah Kriptografi. Jurnal Ilmiah d'Computare . Vol (9), 8-14.

[13] S. Y. Wulandari. 2020.Cryptography: A Combination of Caesar and Affine Cipher to Conceal the Message. *PROC. INTERNAT. CONF. SCI. ENGIN*.3 (1), 741-744

[14]R.Febrianingsih.2019.Implementasi Kriptografi Berbasis Cipher Untuk Keamanan Data. Jurnal Informasi dan Komputer. 7(2).81-86.

[15]D.Ratna. 2018. Implementasi Algoritma Rail Fence Chiper dalam Keamanan Data Gambar 2 Dimensi.Jurnal Pelita Informatika.17 (3), 267-271.

.