Buletin Sistem Informasi dan Teknologi Islam
Vol 3, No 3, Agustus 2022, pp. 238-247
ISSN: 2721-0901

Implementasi Algoritma Caesar Cipher Dengan Kombinasi Transposisi Diagonal Untuk Enkripsi Dekripsi Menggunakan Tabel ASCII

Andy Aprianto[a,1,*], Erick Irawadi Alwi[a,2], Herman[a,3]

[a] *Program Studi Teknik Informatika, Universitas Muslim Indonesia, Jl. Urip Sumoharjo KM.05 , Makassar dan 90231, Indonesia*
[1]andyaprianto96@gmail.com; [2]erick.alwi@umi.ac.id; [3]herman@umi.ac.id;
[*]corresponding author

INFORMASI ARTIKEL

Diterima : 05 – 04 – 2022
Direvisi : 15 – 08 – 2022
Diterbitkan : 31 – 08 – 2022

Kata Kunci:
Kriptografi
Enkripsi
Dekripsi
Caesar Cipher
Diagonal Cipher

ABSTRAK

Masalah keamanan informasi merupakan salah satu aspek penting dari sebuah sistem informasi. Keamanan informasi dimaksudkan untuk mencapai kerahasiaan, ketersediaan, dan integritas di dalam sumber daya informasi dalam suatu perusahaan. Sangat pentingnya nilai sebuah informasi menyebabkan seringkali informasi diinginkan hanya boleh diakses oleh orang-orang tertentu. Jatuhnya informasi ke tangan pihak lain dapat menimbulkan kerugian bagi pemilik informasi. Hal ini menjadi alasan kenapa diperlukan keamanan data untuk mencegah atau mengurangi peluang terjadinya kejahatan terhadap informasi atau data yang bersifat rahasia. Kriptografi adalah suatu metode yang dapat digunakan untuk mengamankan data/informasi. Salah satu cara untuk melakukan proses enkripsi agar menghasilkan *ciphertext* yang susah dipecahkan adalah konsep Super Enkripsi. Hasil penelitian ini menunjukkan bahwa mengkombinasikan metode *caesar cipher* dan *diagonal cipher* dengan menggunakan tabel ASCII dapat dilakukan untuk mengamankan data /informasi dalam bentuk file berekstensi *.docx*. Berdasarkan hasil pengujian sistem yang dilakukan pada 10 file *docx* dengan ukuran file dan jumlah karakter yang berbeda-beda, dimana ukuran file *docx* setelah dienkripsi/dekripsi lebih kecil dibandingkan sebelum dienkripsi. Proses super enkripsi ini membutuhkan waktu lebih lama karena proses enkripsi dan dekripsi menggunakan dua metode, serta jumlah karakter *plaintext* yang dimasukkan juga ikut mempengaruhi waktu proses super enkripsi.

I. Pendahuluan

Di era keterbukaan informasi dan perkembangan teknologi seperti sekarang ini, seseorang dengan mudah menyimpan, mengunggah, dan mengakses informasi baik berupa data atau apapun dengan bantuan internet. Akses internet pun kini semakin mudah, tidak hanya komputer atau laptop saja, smartphone dan gadget pun dapat mengakses internet.

Masalah dalam pengamanan data masih merupakan suatu aspek penting didalam penjagaan penyimpanan data terutama data yang tersimpan dalam bentuk digital, disebabkan karena kemajuan yang sangat pesat didalam bidang ilmu komputer dengan konsep *open-system* yang sudah banyak digunakan, sehingga dapat memudahkan seseorang untuk melakukan perusakan data terutama data yang tersimpan dalam bentuk digital tanpa harus diketahui oleh pihak penyimpan data. Untuk menjaga keamanan dan kerahasiaan data tersebut diperlukan beberapa pengamanan agar data tidak dapat dimengerti oleh pihak yang tidak berwenang, kecuali oleh penerima yang berhak. Beberapa cara untuk menangani masalah keamanan ini dengan menggunakan teknik penyandian data yang dikenal dengan ilmu kriptografi [1].

Dalam proses kriptografi ada 2 proses yaitu proses enkripsi dan dekripsi [2]. Enkripsi adalah proses diubahnya isi pesan yang dimengerti menjadi tidak bisa dimengerti lagi, karena isi sudah diubah[3]. Dekripsi adalah proses dimana pesan yang sudah diubah dikembalikan keasliannya, agar dapat dimengerti kembali seperti sedia kala [4].

Ada beberapa algoritma yang dapat digunakan untuk enkripsi dekripsi dalam mengamankan data dengan menggunakan caesar cipher dan transposisi diagonal atau disebut dengan metode subtitusi dan transposisi. Kedua metode ini merupakan kriptografi klasik, dimana kriptografi klasik ini masih bersifat lemah. Kelemahan-kelemahan kriptografi klasik terletak pada algoritmanya yang terlalu sederhana [5]. Namun Kelemahan-kelemahan kriptografi klasik ini bisa dikurangi dengan mengkombinasikan metode-metode yang ada sehingga algoritma kriptografinya menjadi lebih rumit metode seperti ini disebut super enkripsi. Super enkripsi adalah salah satu kriptografi berbasis karakter yang menggabungkan cipher subtitusi dan cipher transposisi[6]. Hal tersebut bertujuan untuk mendapatkan cipher yang lebih kuat dari pada hanya menggunakan satu cipher saja, sehingga tidak mudah untuk dipecahkan.

Pada kriptografi klasik biasa, kebanyakan plaintext hanya berisi alfabet saja [7]. Bila kita masukkan angka atau karakter lain, maka teks tidak bisa dienkripsi. Untuk mengakomodasi hal ini, maka kita akan menggunakan standar ASCII untuk merepresentasikan karakter-karakter standar yang ada. Untuk karakter standar ASCII, ada 255 karakter yang ada[8]. Dengan pemanfaatan karakter-karakter ASCII ini, kriptografi klasik masih bisa digunakan di zaman modern seperti sekarang. Keterbatasan penggunaan hanya pada alfabet bisa dieliminasi dengan cara ini. Metode kriptografi klasik caesar cipher akan menjadi sulit untuk ditembus bila kemungkinannya menjadi 255. Karakter ASCII ini sudah menjadi standar dalam penggunaan teknologi yang umum[9]. Semua pengguna teknologi rata-rata mengetahui standar ASCII sehingga proses enkripsi-dekripsi bisa dilakukan dengan baik.

Penelitian-penelitian yang terkait dengan kriptografi sudah banyak dilakukan dengan menerapkan metode kriptografi. Dalam penelitian [10] menyatakan bahwa dalam menjaga kerahasiaan suatu pesan dapat digunakan kombinasi algoritma *Vigenere Cipher* dan *Caesar Cipher* agar keamanan isi pesan terproteksi lebih kuat dan aman sehingga apabila pesan yang dikirimkan dibajak ataupun disadap oleh orang yang tidak bertanggung jawab maka si pembajak kesulitan untuk mengetahui isi pesannya. Pada penelitian ini masih terdapat kekurangan dimana jumlah karakter yang dienkripsi masih sangat terbatas yaitu hanya 25 karakter saja, penelitian yang dilakukan tidak dapat mengenkripsi nomor,simbol baca dan tidak dapat membedakan huruf kapital dan huruf kecil.

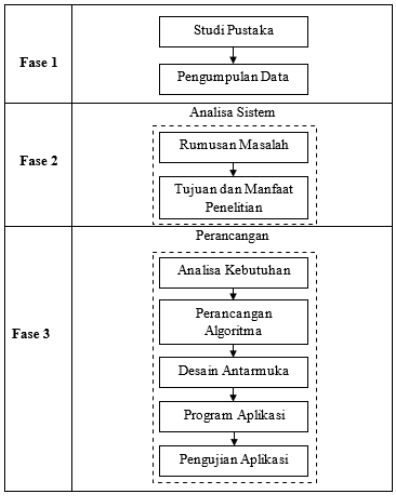
Berdasarkan permasalahan diatas peneliti berinisiatif untuk mengimplementasikan Algoritma *Caesar Cipher* Dengan Kombinasi Transposisi Diagonal Untuk Enkripsi Dekripsi Menggunakan Tabel ASCII. Salah satu cara untuk melakukan proses enkripsi agar menghasilkan ciphertext yang susah dipecahkan adalah konsep super enkripsi.

Hasil dari penelitian ini berupa sebuah aplikasi yang mana aplikasi tersebut mampu melakukan enkripsi dan dekripsi file dokumen berekstensi *.docx*. Metode yang digunakan tidak hanya dapat mengenkripsi file dokumen berekstensi *.docx* tapi juga bisa diterapkan pada file dokumen lainnya seperti file dokumen berekstensi *pdf, txt,* dan *xls,* tapi pada penelitian ini penulis membatasi file dokumen yang akan di enkripsi-dekripsi yaitu hanya fokus pada file dokumen berekstensi *.docx* saja. File dokumen asli yang berisi informasi penting akan dienkripsi menjadi kalimat tidak berarti dengan penerapan algoritma *caesar cipher* dan *diagonal cipher*. Kemudian ketika isi dari file dokumen asli ingin diketahui maka kita menggunakan teknik dekripsi file dokumen dengan penerapan algoritma *caesar cipher* dan *diagonal cipher* untuk memecah kode-kode tersebut menjadi teks yang berisi informasi sebelumnya (asli).

II. Metode

A. Tahapan penelitian

Tahapan penelitian terdiri dari beberapa fase seperti pada gambar 1.



Gambar 1. Tahapan Penelitian

Berikut merupakan penjelasan dari tahapan penelitian yang sudah dikemukakan:

1) Studi pustaka, dalam skripsi ini penulis ambil dari berbagai referensi.
2) Pengumpulan data, dalam skripsi ini penulis mengumpulkan data yang berhubungan dengan teknik yang digunakan.
3) Analisa sistem, masalah yang diangkat dalam skripsi adalah bagaimana melakukan proses enkripsi dan dekripsi dengan kombinasi algoritma *caesar cipher* dan transposisi diagonal.
4) Analisa kebutuhan, untuk membuat sistem ini penulis membutuhkan beberapa perangkat keras dan perangkat lunak yang digunakan.
5) Metode, metode algoritma yang penulis gunakan dalam penulisan tugas akhir adalah dengan menggunakan teknik kombinasi algoritma *caesar cipher* dan transposisi diagonal menggunakan tabel ASCII.
6) Desain sistem, penulis memulai proses mendesain sistem dengan menggunakan UML agar terlihat alur proses enkripsi dan dekripsi dengan teknik kombinasi algoritma *caesar cipher* dan transposisi diagonal.
7) Pembuatan sistem, penulis membuat sistem dengan menggunakan neatbeans.
8) Program aplikasi berfungsi untuk menguji teknik kombinasi *caesar cipher* dan transposisi diagonal menggunakan tabel ASCII.
9) Pengujian Aplikasi, penulis mengimplementasikan sistem dengan menjalankan program aplikasi yang dibuat dan menguji kebenaran hasil yang diperoleh pada saat pengujian dengan menggunakan program aplikasi tersebut.

B. Teknik Pengumpulan Data

Teknik pengumpulan bertujuan untuk mengumpulkan berbagai referensi yang dapat digunakan untuk membuat program aplikasi sesuai dengan topik Implementasi Algoritma *Caesar Cipher* Dengan Kombinasi Transposisi Diagonal Untuk Enkripsi Dekripsi Menggunakan Tabel ASCII. Teknik dapat dilakukan dengan beberapa cara, antara lain:

1. Studi Literatur
   Studi literatur adalah mencari teori-teori pendukung yang relevan agar proses dari pembuatan program aplikasi berjalan dengan baik dan benar.
2. Observasi
   Observasi merupakan teknik yang digunakan untuk mengumpulkan data dengan melakukan pengamatan terhadap fenomena yang terjadi pada teknik *caesar cipher* dan *diagonal cipher*.

C. Analisa Sistem

Sistem yang dirancang diharapkan dapat melakukan proses enkripsi dan dekripsi dengan menggunakan teknik kombinasi *caesar cipher* dengan *diagonal cipher*. Pada sistem terdapat *input*, proses dan *output*. Ketiga bagian tersebut harus terhubung secara baik dan benar. Pada penggunaan enkripsi tersebut, sistem akan mengenkripsi karakter yang ada pada file *.docx* sehingga berubah menjadi *ciphertext*. Pesan *ciphertex*t tidak akan dapat dipahami oleh orang yang membaca. Teknik kombinasi *caesar cipher* dengan *diagonal cipher* akan diterapkan pada saat pengguna mulai melakukan proses enkripsi atau dekripsi pada file yang diinputkan. File tersebut akan melalui 2 kali proses enkripsi begitupun dengan proses dekripsi sehingga akan menghasilkan *ciphertext* yang kuat dibandingkan dengan hanya menggunakan satu algoritma saja.

D. Kombinasi *Caesar Cipher* dengan Transposisi Diagonal

Caesar Cipher dan Transposisi Diagonal merupkan kriptografi klasik yang apabila di implementasikan sendiri-sendiri akan menghasilkan *ciphertext* yang lemah tetapi apabila di kombinasikan dengan teknik kriptografi klasik yang ada akan menghasilkan *ciphertext* yang rumit sehingga seorang kripnatalis tidak dapat mengetahui isi pesan yang di kirim. Dalam penelitian ini penulis mengkombinasikan algoritma *Caesar Cipher* dengan Transposisi Diagonal dalam mengamankan file ekstensi *docx*.

Pada penelitian ini dilakukan modifikasi, modifikasi yang dilakukan adalah tidak melakukan pembuangan spasi pada *plaintext* dan modulus yang digunakan adalah modulus 255 karena jumlah kode ASCII yang digunakan adalah 255 yang terdiri dari karakter dan simbol. Berikut adalah rumus modifikasi enkripsi *caesar cipher* yang diterapkan :

$$E(x) = ((x - spasi + k) \bmod 255) + spasi \quad (1)$$

dimana x adalah karakter yang akan diganti dan k adalah nilai kunci. Spasi adalah nomor ASCII dari karakter spasi.

Misal *plaintext* yang akan dienkripsi adalah “Halo, Apa Kabar?~” (tidak termasuk tanda kutip) dengan kunci pergeseran 3. Beberapa hasil enkripsinya adalah :

a : ((97-32+3) mod 255)+32 = 100 =
d , : ((44-32+3) mod 255)+32 = 47 = /
spasi : ((32-32+3) mod 255)+ 32 = 35 = #

~ : ((126-32)+3) mod 255)+32 = 34 = ”

Dengan demikian ciphertext hasil penerapan modifikasi *caesar cipher* menjadi Kdor/#Dsd#NdeduB"

Pada transposisi diagonal mula-mula hitung P = banyaknya *plaintext*

Bentuk matriks *n* x *n* dengan $n = \sqrt{C} + 1$ (2)

Tulis *chipertext* ke dalam matriks bujur sangkar

$$M_{n \times n} = \begin{bmatrix} C_1 & C_2 & C_3 \\ C_4 & C_5 & C_6 \\ C_7 & C_8 & C_n \end{bmatrix} \quad (3)$$

Huruf *chipertext* adalah elemen matriks M yang dibaca menurut diagonal kiri bawah ke kanan atas.

$$M_{n \times n} = \begin{bmatrix} C_1 & C_2 & C_3 \\ C_4 & C_5 & C_6 \\ C_7 & C_8 & C_n \end{bmatrix} \quad (4)$$

C = $C_1, C_4, C_2, C_7, C_5, C_3, C_8, C_6, \ldots\ldots C_n$

Proses dekripsi memasukan *ciphertext*, tentukan kunci, kunci yang digunakan harus sama dengan kunci enkripsi.

Hitung C = banyaknya chiperteks

Bentuk matriks *n* x *n* dengan $n = \sqrt{C} + 1$ (5)

Tulis c*iphertext* ke dalam matriks dengan pola diagonal

$$M_{n \times n} = \begin{bmatrix} C_1 & C_3 & C_6 \\ C_2 & C_5 & C_8 \\ C_4 & C_7 & C_n \end{bmatrix} \quad (6)$$

Huruf *ciphertext* adalah elemen matriks yang dibaca dari kiri ke kanan

$$M_{n \times n} = \begin{bmatrix} C_1 & C_3 & C_6 \\ C_2 & C_5 & C_8 \\ C_4 & C_7 & C_n \end{bmatrix} \quad (7)$$

Sehingga di peroleh *ciphertext* = $C_1, C_3, C_6, C_2, C_5, C_8, C_4, C_7, \ldots C_n$

Geser chiperteks dengan pergeseran karakter sebelumnya sesuai dengan kunci yang telah ditentukan dan menggunakan persamaan:

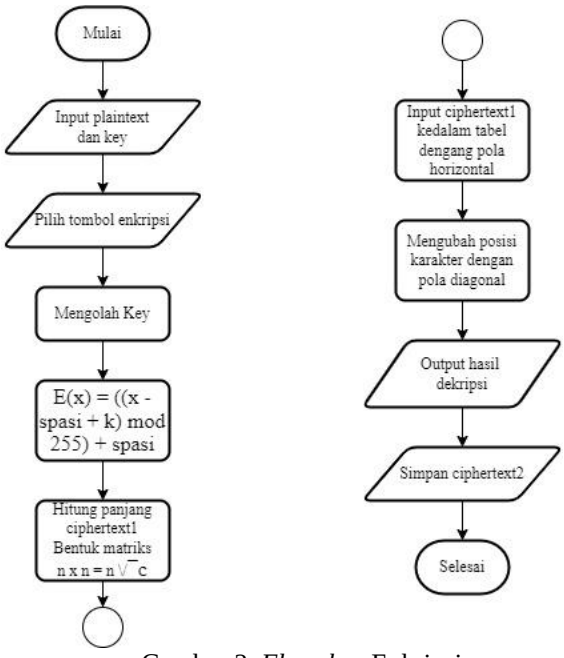
D(x) = ((255 + x – spasi - kunci) mod 255) + spasi (8)

Setelah melalui 2 kali proses dekripsi akan di peroleh kembali *plaintext* asli.

E. Rancangan Model Diagram

1. *Flowchar* Enkripsi

Pada bagian ini akan dijelaskan *flowchar* proses enkripsi yang menjelaskan alur dari proses enkripsi dengan teknik kombinasi *caesar cipher* dan *diagonal cipher*.



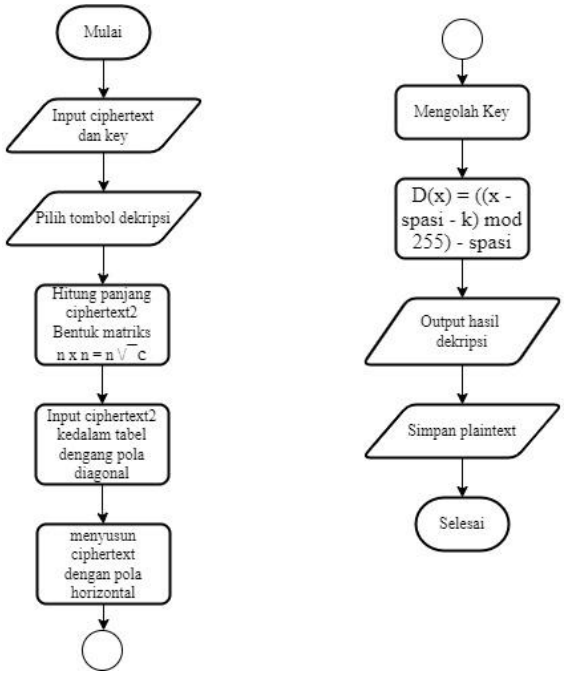
Gambar 2. *Flowchar* Enkripsi

Gambar 2 adalah *flowchar* enkripsi dimana proses enkripsi pengguna harus memasukan file dokumen terlebih dahulu sebagai *plaintext*, atau menginputkan karakter secara langsung. Tekan tombol enkripsi setelah

memasukan kunci dan sistem akan melakukan proses enkripsi dimana *plaintext* akan melalui dua kali proses enkripsi. Proses enkripsi pertama menggunakan teknik *caesar cipher* dimana karakter akan di geser sebanyak kunci sehingga menghasilkan *ciphertext*. Setelah *ciphertext* diperoleh masuk ke proses enkripsi kedua menggunakan teknik *diagonal cipher* sistem akan menghitung jumlah *plaintext* untuk menentukan jumlah baris dan kolom kemudian melakukan transposisi karakter pada *ciphertext* hasil enkripsi *caesar cipher*. Setelah proses enkripsi kedua selesai akan ditampilkan *ciphertext* dan lokasi penyimpanan.

2. *Flowchar* Dekripsi

Pada bagian ini akan dijelaskan *flowchar* proses dekripsi yang menjelaskan alur dari proses dekripsi dengan teknik kombinasi *caesar cipher* dan *diagonal cipher*.



Gambar 3. *Flowchar* Dekripsi

Gambar 3 adalah *flowchar* dekripsi dimana proses dekripsi pengguna harus memasukan file dokumen yang sudah di enkripsi dan kunci, kunci yang digunakan harus sama pada saat melukukan enkripsi . Tekan tombol dekripsi dan sistem akan melakukan proses dekripsi dimana *ciphertext* akan melalui dua kali proses dekripsi. Proses dekripsi pertama menggunakan teknik *diagonal cipher* sistem akan menghitung jumlah *ciphertext* untuk menentukan jumlah baris dan kolom kemudian melakukan transposisi karakter. Setelah proses dekripsi pertam selesai lanjut ke proses dekripsi kedua menggunakan teknik *caesar cipher* dimana karakter akan digeser sebanyak kunci sehingga menghasilkan *plaintext*.Setelah proses dekripsi kedua selesai akan ditampilkan *plaintext* dan lokasi penyimpanan.

III. Hasil dan Pembahasan

A. Uji Coba Sistem

Uji coba sistem adalah tahapan pengujian dengan menjalankan program aplikasi yang dibuat dan menguji kebenaran hasil yang diperoleh pada saat pengujian dengan menggunakan program aplikasi tersebut. Pengujian juga dilakukan untuk melihat kecepatan proses enkripsi, dekripsi dan perubahan ukuran file setelah dienkripsi/dekripsi. Maka dari itu penulis akan menguji program dengan 10 file *docx* dengan jumlah karakter atau panjang *plaintext* yang berbeda. Untuk melakukan proses enkripsi dan dekripsi file *docx* yang akan di uji coba dapat dilihat pada tabel 1.

Tabel 1. Pengujian file *docx*

No.	Ukuran File	Jumlah Karakter	Waktu Enkripsi	Waktu Dekripsi	Ukuran File Hasil Enkripsi	Ukuran File Hasil Dekripsi
1	12 KB	19	0.031 Detik	0.015 Detik	1.37 KB	1.36 KB
2	12 KB	28	0.078 Detik	0.031 Detik	1.46 KB	1.44 KB
3	13 KB	466	0.203 Detik	0.172 Detik	1.70 KB	1.61 KB
4	14 KB	651	0.393 Detik	0.296 Detik	1.82 KB	1.71 KB
5	15 KB	1.777	1.317 Detik	1.296 Detik	2.53 KB	2.11 KB
6	15 KB	1.943	1.323 Detik	1.328 Detik	2.63 KB	2.19 KB
7	16 KB	2.220	1.641 Detik	2.187 Detik	2.85 KB	2.35 KB
8	19 KB	2.523	2.343 Detik	2.093 Detik	3.21 KB	2.30 KB
9	21 KB	5.928	8.013 Detik	7.951 Detik	5.09 KB	3.44 KB
10	24 KB	6.426	9.279 Detik	9.326 Detik	5.49 KB	3.78 KB

Berdasakan tabel diatas hasil percobaan enkripsi dan dekripsi 10 jenis file *docx* dengan ukuran file yang berbeda-beda dan jumlah karakter yang berbeda, dapat disimpulkan bahwa lambat cepatnya proses enkripsi dan dekripsi itu dipengaruhi oleh panjang *plaintext / ciphertext* yang diinputkan, dimana semakin panjang *plaintext / ciphertext* yang digunakan maka semakin lama waktu yang dibutuhkan untuk proses enkripsi dan dekripsi. Ukuran file *docx* setelah dienkripsi / dekripsi mengalami perubahan dimana ukuran file *docx* setelah dienripsi / dekripsi lebih kecil dibandingkan file *docx* sebelum dienkripsi / dekripsi, ini disebabkan karena *style* pada file *docx* yang diatur pada *Software Microsoft Word* ketika di masukkan kedalam sistem yang dibuat tidak terdeteksi. File *docx* setalah dilakukan enkripsi menghasilkan kalimat yang tidak berarti atau disebut dengan *ciphertext*, untuk mengetahui informasi yang ada pada *ciphertext* dilakukan proses dekripsi untuk mengembalikan ke *plaintext* asli. Setelah dilakukan proses dekripsi pada *ciphertext*, hasil yang di peroleh adalah *plaintext* aslinya tanpa mengurangi informasi yang ada di dalamnya ini membuktikan bahwa sistem yang di buat sudah berjalan dengan benar.

B. Pembahasan

Pembahasan berfungsi untuk menguji hasil yang dikeluarkan oleh program aplikasi apakah sudah sesuai dengan perhitungan sebenarnya. Dengan ini, diperlukan perhitungan manual yang akan membuktikan kebenaran dari teknik *caesar cipher* dan *diagonal cipher*. Proses yang dilakukan terdiri dari dua proses, yaitu proses enkripsi dan proses dekripsi. Penjelasan dan perhitungan berikut ini adalah langkah lengkap proses enkripsi dan dekripsi pada Teknik *caesar cipher* dan *diagonal cipher*.

1. Pembahasan Implementasi Enkripsi

a. Proses Enkripsi *Caesar Cipher*

Pada penelitian ini, proses enkripsi tahap pertama adalah enkripsi menggunakan *caesar cipher*. Berikut *plaintext* yang akan dienkripsi adalah sebagai berikut.

***Plaintext* : Belajar Kriptografi**
***Key* : Caesar123**

Mengolah *key* dengan menghilangkan karakter yang sama. *Key* yang diinputkan adalah "Caesar123", maka hasil pengolahan *key* dapat dilihat berikut ini.

***Key* : Caesar123**
Hasil *Key* : 'C', 'a', 'e', 's', 'r', '1', '2', '3'

Selanjutnya akan dienkripsi dengan metode *caesar cipher*, *plaintext* "Belajar Kriptografi" dengan pergeseran 8.

Karakter [B]Diubah Menjadi Karakter [J]
Karakter [e]Diubah Menjadi Karakter [m]
Karakter [l]Diubah Menjadi Karakter [t]
Karakter [a]Diubah Menjadi Karakter [i]
Karakter [j]Diubah Menjadi Karakter [r]
Karakter [a]Diubah Menjadi Karakter [i]
Karakter [r]Diubah Menjadi Karakter [z]
Karakter [ ]Diubah Menjadi Karakter [( ]
Karakter [K]Diubah Menjadi Karakter [S]
Karakter [r]Diubah Menjadi Karakter [z]
Karakter [i]Diubah Menjadi Karakter [q]
Karakter [p]Diubah Menjadi Karakter [x]
Karakter [t]Diubah Menjadi Karakter [|]

Karakter [o]Diubah Menjadi Karakter [w]
Karakter [g]Diubah Menjadi Karakter [o]
Karakter [r]Diubah Menjadi Karakter [z]
Karakter [a]Diubah Menjadi Karakter [i]
Karakter [f]Diubah Menjadi Karakter [n]
Karakter [i]Diubah Menjadi Karakter [q]
Hasil dari proses enkripsi adalah *ciphertext* yaitu sebagai berikut.
Ciphertext : Jmtiriz(Szqx|wozinq

b. Proses Enkripsi *Diagonal Cipher*

Proses enkripsi *diagonal cipher* dilakukan setelah *plaintext* telah dienkripsi menggunakan *caesar cipher*. Proses enkripsi menggunakan *diagonal cipher* dilakukan terhadap hasil enkripsi dari *caesar cipher* yang selanjutnya disebut sebagai *ciphertext1*.

***Ciphertext1* : Jmtiriz(Szqx|wozinq**

Menghitunng panjang karakter dari *ciphertext1* (teks yang akan dienkripsi).

Ciphertext1 : Jmtiriz(Szqx|wozinq
Panjang Karakter : 19 karakter

Bentuk matriks *n* x *n* dengan $n = \sqrt{C}$

$n = \sqrt{C}$
$n = \sqrt{19}$
$n = 4$

karena 4 x 4 = 16, sementara panjang karakter 19 yang artinya jumlah kolom dan barisnya kurang maka dilakukan penambahan baris dan kolom sehingga *n* = 4 + 1 jadi *n* = 5.

Ukuran Tabel Diagonal Cipher = 5 x 5

Membuat tabel dengan jumlah baris 5 dan kolom 5, kemudian masukkan ciphertext secara horizontal dimulai dari baris pertama sebelah kiri. Jika terdapat baris atau kolom yang kosong maka akan di isi dengan simbol "¯"

J	m	t	i	r
i	z	(	S	z
q	x	|	w	o
z	i	n	q	¯
¯	¯	¯	¯	¯

Selanjutnya mengubah susunan abjad dengan pola diagonal, maka :

Nilai Baris : 0 & Kolom : 0 = J
Nilai Baris : 1 & Kolom : 0 = i
Nilai Baris : 0 & Kolom : 1 = m
Nilai Baris : 2 & Kolom : 0 = q
Nilai Baris : 1 & Kolom : 1 = z
Nilai Baris : 0 & Kolom : 2 = t
Nilai Baris : 3 & Kolom : 0 = z
Nilai Baris : 2 & Kolom : 1 = x
Nilai Baris : 1 & Kolom : 2 = (
Nilai Baris : 0 & Kolom : 3 = i
Nilai Baris : 4 & Kolom : 0 = ¯
Nilai Baris : 3 & Kolom : 1 = i
Nilai Baris : 2 & Kolom : 2 = |
Nilai Baris : 1 & Kolom : 3 = S
Nilai Baris : 0 & Kolom : 4 = r
Nilai Baris : 4 & Kolom : 1 = ¯
Nilai Baris : 3 & Kolom : 2 = n
Nilai Baris : 2 & Kolom : 3 = w
Nilai Baris : 1 & Kolom : 4 = z
Nilai Baris : 4 & Kolom : 2 = ¯
Nilai Baris : 3 & Kolom : 3 = q
Nilai Baris : 2 & Kolom : 4 = o
Nilai Baris : 4 & Kolom : 3 = ¯
Nilai Baris : 3 & Kolom : 4 = ¯

Nilai Baris : 4 & Kolom : 4 = ¯
Sehingga *ciphertext* yang terbentuk adalah **Jimqztzx(i¯i|Sr¯nwz¯qo¯¯¯¯**.

2. Pembahasan Implementasi Dekripsi
a. Proses Dekripsi *Diagonal Cipher*

Pada penelitian ini, proses dekripsi tahap pertama adalah dekripsi menggunakan *diagonal cipher* . Berikut *plaintext* yang akan di dekripsi adalah sebagai berikut.

Ciphertext : Jimqztzx(i¯i|Sr¯nwz¯qo¯¯¯¯
Panjang Karakter : 25 karakter

Bentuk matriks *n* x *n* dengan $n = \sqrt{C}$
$n = \sqrt{C}$
$n = \sqrt{25}$
$n = 5$

Membuat tabel dengan jumlah baris 5 dan kolom 5
Tulis *ciphertext* kedalam tabel secara *diagonal* dimulai dari :
Nilai Baris : 0 & Kolom : 0=J
Nilai Baris : 1 & Kolom : 0=i
Nilai Baris : 0 & Kolom : 1=m
Nilai Baris : 2 & Kolom : 0=q
Nilai Baris : 1 & Kolom : 1=z
Nilai Baris : 0 & Kolom : 2=t
Nilai Baris : 3 & Kolom : 0=z
Nilai Baris : 2 & Kolom : 1=x
Nilai Baris : 1 & Kolom : 2=(
Nilai Baris : 0 & Kolom : 3=i
Nilai Baris : 4 & Kolom : 0=¯
Nilai Baris : 3 & Kolom : 1=i
Nilai Baris : 2 & Kolom : 2=|
Nilai Baris : 1 & Kolom : 3=S
Nilai Baris : 0 & Kolom : 4=r
Nilai Baris : 4 & Kolom : 1=¯
Nilai Baris : 3 & Kolom : 2=n
Nilai Baris : 2 & Kolom : 3=w
Nilai Baris : 1 & Kolom : 4=z
Nilai Baris : 4 & Kolom : 2=¯
Nilai Baris : 3 & Kolom : 3=q
Nilai Baris : 2 & Kolom : 4=o
Nilai Baris : 4 & Kolom : 3=¯
Nilai Baris : 3 & Kolom : 4=¯
Nilai Baris : 4 & Kolom : 4=¯

J	m	t	i	r
i	z	(	S	z
q	x	|	w	o
z	i	n	q	¯
¯	¯	¯	¯	¯

Untuk memporel hasil dekripsi *diagonal cipher* dibaca secara *horizontal* di mulai dari sudut atas kiri baris pertama. Berikut hasil dekripsi dari proses tersebut adalah **Jmtiriz(Szqx|wozinq¯¯¯¯¯¯**.

b. Proses Dekripsi *Caesar Cipher*

Pada penelitian ini, proses dekripsi tahap kedua adalah dekripsi menggunakan *caesar cipher.* Berikut *plaintext* yang akan di dekripsi adalah sebagai berikut.

***Ciphertext* : Jmtiriz(Szqx|wozinq**
***Key* : Caesar123**

Mengolah *key* dengan menghilangkan karakter yang sama. *Key* yang diinputkan adalah “Caesar123”, maka hasil pengolahan *key* dapat dilihat berikut ini.

Key **: Caesar123**
Hasil *Key* **: 'C', 'a', 'e', 's', 'r', '1', '2', '3'**
Dekripsi dengan metode *caesar cipher*, *plaintext* "**Jmtiriz(Szqx|wozinq**". Berikut proses pergeseran dengan kunci 8:
Karakter [J]Diubah Menjadi Karakter [B]
Karakter [m]Diubah Menjadi Karakter [e]
Karakter [t]Diubah Menjadi Karakter [l]
Karakter [i]Diubah Menjadi Karakter [a]
Karakter [r]Diubah Menjadi Karakter [j]
Karakter [i]Diubah Menjadi Karakter [a]
Karakter [z]Diubah Menjadi Karakter [r]
Karakter [( ]Diubah Menjadi Karakter [ ]
Karakter [S]Diubah Menjadi Karakter [k]
Karakter [z]Diubah Menjadi Karakter [r]
Karakter [q]Diubah Menjadi Karakter [i]
Karakter [x]Diubah Menjadi Karakter [p]
Karakter [|]Diubah Menjadi Karakter [t]
Karakter [w]Diubah Menjadi Karakter [o]
Karakter [o]Diubah Menjadi Karakter [g]
Karakter [z]Diubah Menjadi Karakter [r]
Karakter [i]Diubah Menjadi Karakter [a]
Karakter [n]Diubah Menjadi Karakter [f]
Karakter [q]Diubah Menjadi Karakter [i]
Hasil dari dekripsi *ciphertext* dengan metode *caesar cipher* diperoleh *plaintext* sebagai berikut.
Plaintext **: Belajar Kriptografi**

IV. Kesimpulan dan saran

Waktu yang dibutuhkan dalam melakukan proses super enkripsi dengan menggunakan dua metode membutuhkan waktu yang lebih lama karena proses enkripsi maupun dekripsi dilakukan sebanyak 2 kali dibandingkan dengan menggunakan satu metode. Durasi dalam melakukan enkripsi/deksipsi di pengruhi oleh panjang *plaintext* yang di inputkan, semakin panjang *plaintext* yang di inputkan maka semakin lama pula waktu yang dibutuhkan. Ukuran file *docx* setelah dienkripsi / dekripsi lebih kecil dibandingkan file *docx* sebelum dienkripsi.

Pada penelitian ini file .*docx* yang akan dienkripsi ketika di masukkan ke dalam sistem, *style* teks yang ada pada file docx akan berubah sehingga tampilan setalah di lakukan enkripsi dan dekripsi akan berbeda namun tidak mengurangi informasi yang ada pada file .*docx* tersebut. Pada Penelitian selanjutnya dapat menerapkan suatu metode yang baru agar *style* file .*docx* ketika dimasukkan kedalam sistem tidak mengalami perubahan. Penelitian selanjutnya, bisa mengimplementasikan proses super enkripsi dalam melakukan pengamanan file seperti gambar atau audio.

Daftar Pustaka

[1] I. Gunawan, "Kombinasi Algoritma Caesar Cipher dan Algoritma RSA untuk pengamanan File Dokumen dan Pesan Teks," *InfoTekJar (Jurnal Nas. Inform. dan Teknol. Jaringan)*, vol. 2, no. 2, pp. 124–129, 2018, doi: 10.30743/infotekjar.v2i2.266.
[2] Y. Wiharto and A. Irawan, "Enkripsi Data Menggunakan Advanced Encryption Standard 256," *J. KILAT*, vol. 7, no. 2, pp. 91–99, 2018.
[3] I. M. Yusup and I. Purnamasari, "Implementasi Algoritma Caesar Cipher Dan Steganografi Least Significant Bit Untuk File Dokumen," *JuTISI*, vol. 6, no. 3, pp. 434–441, 2020.
[4] Y. B. Rio Irawan, Ilhamsyah, "Dekripsi Pesan Singkat Menggunakan Algoritma Knapsack Berbasis Android," *J. Coding Sist. Komput. Untan*, vol. 29, no. 6, pp. 58–59, 2015.
[5] B. S. Hasugian, "Peranan Kriptografi Sebagai Keamanan Sistem Informasi Pada Usaha Kecil Dan Menengah," *J. War. Ed.*, 2017.
[6] I. N. Diana, "Algoritma Affine Cipher dan Modifikasi Affine Cipher , serta Kombinasinya dengan Cipher Transposisi Grup Simetri untuk Mengamankan Pesan Teks," *KUBIK J. Publ. Ilm. Mat.*, vol. 7, no. 1, 2022.
[7] R. Latifah, S. N. Ambo, and S. I. Kurnia, "Modifikasi Algoritma Caesar Chiper Dan Rail Fence Untuk Peningkatan Keamanan Teks Alfanumerik Dan Karakter Khusus," *Semin. Nas. Sains dan Teknol. 2017*, 2017.
[8] N. P. S. Winarno and T. A. Cahyanto, "Penggunaan Karakter Kontrol ASCII Untuk Integrasi Data Pada Hasil Enkripsi Algoritma Caesar Cipher," *Informatics J.*, vol. 6, no. 3, pp. 197–204, 2021.

[9] Supiyandi, Hermansyah, and K. A. P. Sembiring, "Implementasi dan Penggunaan Algoritma Base64 dalam Pengamanan File Video," *J. MEDIA Inform. BUDIDARMA*, vol. 4, no. April, pp. 340–346, 2020, doi: 10.30865/mib.v4i2.2042.

[10] V. C. Hardita and E. W. Sholeha, "Penerapan Kombinasi Metode Vigenere Cipher, Caesar Cipher Dan Simbol Baca Dalam Mengamankan Pesan," *J. SAINTEKOM*, vol. 11, no. 1, pp. 34–43, 2021, doi: 10.33020/saintekom.v11i1.202.