JURNAL TIMES
TECHNOLOGY INFORMATICS & COMPUTER SYSTEM

Volume XIV No 1, Juni 2025
*p*ISSN : 2337 – 3601
*e*ISSN : 2549 – 015X

Tersedia *online* di http://ejournal.stmik-time.ac.id

IMPLEMENTASI METODE SUBSTITUSI *OUT BY DIAGONALS* PADA PENGAMANAN *FILE* TEKS

Virgini Graditya Ramengkomole[1)], Eliasta Ketaren[2)], Deiby Tineke Salaki[3)]
Sistem Informasi
Universitas Sam Ratulangi
Jl. Kampus Unsrat, Bahu-Kleak, Manado 95115
email: virgin17virgin@gmail.com[1)], eliasketaren@unsrat.ac.id[2)], deibyts.mat@unsrat.ac.id[3)]

Abstrak

Perkembangan teknologi informasi telah memberikan kemudahan akses terhadap data, tetapi juga meningkatkan risiko keamanan data, seperti pencurian dan penyalahgunaan informasi sensitif. File teks, sebagai format dasar yang sering digunakan karena mudah dibaca dan dipahami manusia, yang rentan terhadap akses tidak sah jika dikirim tanpa perlindungan. Oleh karena itu, diperlukan metode enkripsi untuk melindungi isi file tersebut. Metode substitusi out by diagonals menjadi solusi yang menarik karena menggunakan pola matriks diagonal untuk menyandikan karakter sehingga pesan asli sulit ditebak tanpa kunci yang tepat. Data yang digunakan mencakup 76 karakter, termasuk huruf kapital, huruf kecil, angka, dan tanda baca. Hasil penelitian menunjukkan bahwa metode susbtitusi out by diagonals berhasil diimplementasikan dengan baik. Sistem mampu menghasilkan ciphertext yang aman dan dapat mengembalikan teks asli setelah proses dekripsi. Penelitian ini menyimpulkan bahwa implementasi metode substitusi out by diagonals berhasil diimplementasikan untuk pengamanan file teks dengan format .pdf dan .docx.

Kata Kunci: Substitusi Out By Diagonals, Enkripsi, Dekripsi, File Teks, Keamanan.

1. Pendahuluan

Perkembangan teknologi informasi dan komunikasi telah berdampak besar pada berbagai sektor, seperti pemerintahan, pendidikan, kesehatan, dan bisnis. Seiring dengan kemudahan akses terhadap data dan informasi, muncul pula risiko keamanan yang mengancam privasi.

Kasus kebocoran data dan informasi terus meningkat dan menjadi ancaman serius bagi pengguna teknologi informasi. Berdasarkan data dari BPJS kesehatan telah menghadapi sejumlah insiden kebocoran data yang signifikan. Salah satu insiden terbesar terjadi pada tahun 2021, ketika data pribadi sekitar 279 juta penduduk Indonesia bocor dan diperjualbelikan di forum online [1]. Dari kasus ini menunjukkan betapa pentingnya penerapan sistem keamanan untuk melindungi data dan informasi pengguna.

File teks merupakan format data yang paling umum atau dasar dari data yang dapat diproses dalam kriptografi [2]. File teks berisi urutan karakter seperti huruf dan angka yang dapat dibaca dan dimengerti oleh manusia, sehingga menjadikannya sebagai salah satu jenis data yang paling sering digunakan. Saat file teks dikirimkan melalui jaringan public, resiko pencurian data menjadi sangat tinggi karena isi file tersebut dapat dengan mudah dibaca dan dipahami oleh siapa saja. Untuk mencegah hal tersebut, diperlukan cara untuk melindungi isi file teks tersebut dengan menggunakan metode enkripsi. Metode enkripsi merupakan suatu teknik yang digunakan untuk mengubah informasi atau oesan aski menjadi bentuk kode rahasia, sehingga tidak dapat langsung dipahami oleh pihak tang tidak berhak. Untuk membaca kembali informasi tersebut, diperlukan proses balikan yang disebut dekripsi [3]. Metode substitusi out by diagonals memiliki keunikan dalam memanipulasi urutan karakter melalui pembacaan matriks secara diagonal. Proses ini membantu menyembunyikan pola asli teks sehingga sulit ditebak atau dibaca oleh pihak yang mencoba melakukan dekripsi tanpa izin. Perlunya metode out by diagonals dalam pengamanan data karena dalam cipher substitusi setiap huruf atau karakter pada teks asli (plaintext) digantikan dengan huruf atau karakter lain untuk membentuk teks yang disandikan dan penggantian ini dapat dilakukan pada setiap huruf atau karakter secara individual pada kelompok huruf atau karakter [4]. Teknik ini sering dianggap sebagai variasi dari cipher klasik atau metode substitusi, dimana pesan asli diubah dan menyembunyikan arti sesungguhnya. Salah satu keunggulan dari metode out by diagonals yaitu dengan penambahan karakter dalam proses penyandian dari 26 karakter menjadi 76 karakter dengan panjang kunci disesuaikan, sehingga tingkat keamanan dapat ditingkatkan.

Berdasarkan penelitian dari Idrus kriptografi klasik dengan algoritma substitusi cipher dapat mengamankan pesan atau informasi. Peneliti menerapkan algoritma substitusi cipher agar mengamankan pesan pribadi dari akses yang tidak sah. Algoritma ini bekerja dengan mengganti setiap karakter plaintext dengan karakter lain dalam susunan alphabet, sehingga menghasilkan ciphertext yang tidak dapat dibaca tanpa proses dekripsi. Hasil penelitian menunjukkan bahwa algoritma substitusi cipher berhasil melakukan enkripsi dan dekripsi dengan baik,

dimana pesan yang telah dienkripsi memiliki perbedaan dengan pesan asli [5]. Berdasarkan penelitian dari Aprianto kombinasi antara algoritma caesar cipher (substitusi) dan transposisi diagonal telah diimplementasikan untuk enkripsi dan dekripsi file dokumen menggunakan tabel ASCII. Proses enkripsi dilakukan dalam dua tahap dengan substitusi karakter menggunakan caesar cipher, lalu dilanjutkan dengan transposisi karakter sesuai pola diagonal dari kiri bawah ke kanan atas. Proses dekripsi dilakukan secara terbalik, dimulai dari transposisi diagonal lalu caesar cipher. Hasil penelitian menunjukkan bahwa kombinasi kedua metode tersebut meningkatkan tingkat keamanan data dengan baik [6]. Penelitian sebelumnya yang dilakukan oleh Ketaren modifikasi algoritma vigenere cipher dapat mengamankan file teks dengan penambahan karakter dalam tabel vigenere cipher dari 26 karakter menjadi 46 karakter dengan menambahkan angka dan simbol, sehingga meningkatkan keamanan dan kekuatan enkripsi [7].

Penelitian ini bertujuan mengimplementasikan metode substitusi out by diagonals untuk pengamanan file teks format .pdf dan .docx dengan penambahan karakter dari 26 menjadi 76 karakter. Penelitian ini diharapkan dapat meningkatkan tingkat keamanan data dengan mengenkripsi file teks menggunakan metode substitusi out by diagonals serta memberikan pemahaman tentang penggunaan metode susbstitusi out by diagonals.

2. Landasan Teori

File Teks

File teks adalah satu bentuk informasi yang berisi data dalam bentuk tulisan, dan terdapat berbagai format teks seperti docx, doc, txt, dan pdf [8]. Data di dalam *file* teks bisa berupa huruf, angka, atau simbol, yang ditulis dalam format yang mudah dipahami.

Plaintext

Plaintext adalah data atau pesan informasi yang dapat dibaca secara langsung oleh manusia, sebelum dienkripsi. Isi dari *plaintext* dapat dengan mudah dibaca, dipahami, dan diakses oleh siapa saja tanpa memerlukan kunci dekripsi. Pesan asli sebelum dienkripsi disebut sebagai *plaintext* [9].

Ciphertext

Ciphertext adalah data atau pesan yang telah melalui proses enkripsi, sehingga bentuknya tidak dapat langsung dibaca atau dipahami oleh manusia tanpa proses dekripsi. Bentuk pesan yang telah disandikan disebut *ciphertext* [5].

Kriptografi

Kriptografi (*Cryptography*) berasal dari bahasa Yunani, yang terdiri dari dua kata yaitu kripto dan graphia, kripto berarti menyembunyikan dan graphia yang berarti tulisan (Miftahul, 2016). Kriptografi adalah ilmu yang mempelajari cara mengamankan pesan dengan tujuan menjaga kerahasiaan informasi yang terdapat dalam data. Kriptografi merupakan ilmu seni dalam menjaga keamanan data yang bersifat spesifik, dengan tujuan untuk mengaburkan atau menyamarkan informasi. Hal ini dilakukan dengan cara mengubah teks biasa (*plaintext*) menjadi teks yang tidak dapat dibaca (*ciphertext*) yang tidak dapat dibaca [11].

Substitusi *Out By Diagonals*

Substitusi *out by diagonal* adalah metode substitusi yang digunakan dalam kriptografi untuk menyandikan teks dengan menata karakter dalam pola diagonal melalui matriks. Pada metode ini, teks atau data yang akan dienkripsi diorganisir dalam bentuk matriks, kemudian diakses dan diganti dengan pola diagonal. Metode ini menghasilkan teks yang sulit dipahami tanpa mengetahui pola diagonal dan matriks yang digunakan, yang memberikan lapisan keamanan tambahan. Menurut Pradeka, substitusi adalah salah satu teknik dasar dalam kriptografi yang bertujuan untuk melindungi pesan dengan cara mengganti karakter asli pesan dengan karakter lain, seperti angka atau simbol [12].

Python

Python pertama kali dikembangkan oleh Guido Van Rossum pada akhir 1980-an dan dirilis pertama kali pada tahun 1991. Python merupakan bahasa pemrograman tingkat tinggi yang mudah dipelajari dan digunakan, dengan sintaksis yang jelas dan sederhana. Menurut Fauzi, python adalah bahasa pemograman yang serbaguna, dengan desain yang mengutamakan kemudahan dalam membaca dan menulis kode [13].

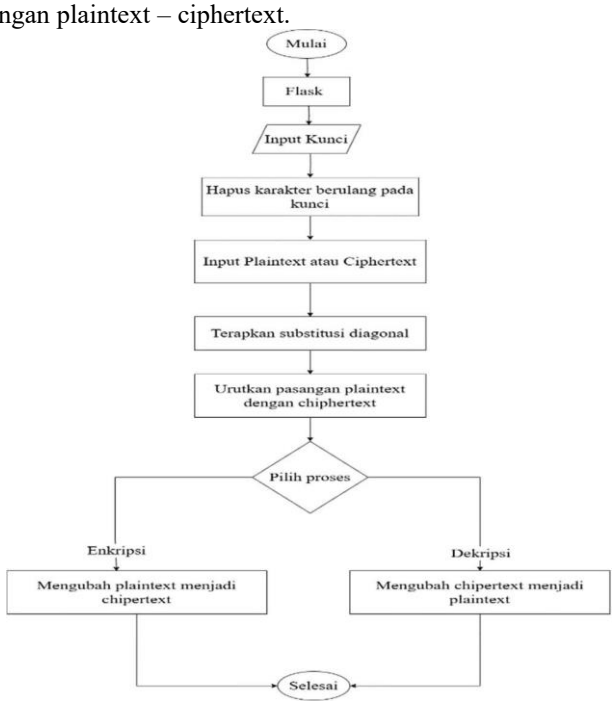
Flask

Flask adalah sebuah *framework* web ringan (*micro-framework*) yang dikembangkan menggunakan bahasa pemrograman python. Flask pertama kali dibuat oleh Armin Ronacher pada tahun 2010. Flask tidak menyertakan komponen-komponen umum seperti *form validation, database* atau lainnya. *Framework* Flask dirancang untuk mendukung ekstensi yang memungkinkan pengembang menambahkan berbagai fitur atau layanan yang tersedia sesuai kebutuhan aplikasi yang akan dibuat [14].

3. Metode Penelitian

Flowchart

Pada Gambar 1. merupakan proses kerja sistem enkripsi dan dekripsi menggunakan metode substitusi out by diagonals. Proses dimulai dengan inisialisasi aplikasi Flask sebagai platform untuk menjalankan aplikasi web, kemudian pengguna diminta untuk memasukan kunci (key) yang digunakan dalam proses enkripsi atau dekripsi. Kunci ini selanjutnya dihilangkan karakter-karakter yang berulang untuk memastikan bahwa setiap karakter dalam kunci bersifat unik. Pengguna memasukan teks (plaintext) untuk enkripsi (ciphertext) untuk dekripsi. Sistem menerapkan substitusi diagonal, dimana karakter dalam teks diubah berdasarkan pola diagonal sesuai kunci. Kemudian mengurutkan pasangan plaintext – ciphertext.



Gambar 1. Flowchart Enkripsi dan Dekripsi

Implementasi Metode Substitusi Out By Diagonals

Penerapan algoritma substitusi out by diagonals untuk menyandikan file teks, .pdf dan .docx. Metode ini memiliki beberapa tahapan untuk mengimplementasikan pola ini pada file teks.

Berikut adalah tahapanya:
Menetapkan kunci berupa kata yang berisi huruf, angka dan tanda baca, hapus karakter berulang pada kunci, masukan karakter yang belum ada pada kunci, terapkan pola substitusi diagonal untuk ciphertext, urutkan pasangan plaintext dan ciphertext, lakukan enkripsi atau dekripsi.

Berikut adalah tabel karakter Plaintext dan Ciphertext

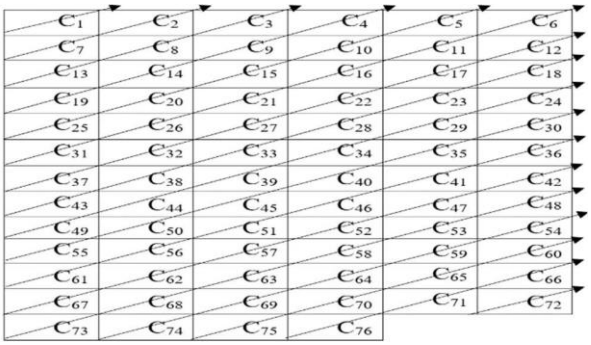
Tabel 1. Karakter Plaintext dan Ciphertext

Plaintext	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
Ciphertext	C_1	C_7	C_2	C_{13}	C_8	C_3	C_{19}	C_{14}	C_9	C_4	C_{25}	C_{20}	C_{15}	C_{10}	C_5	C_{31}	C_{26}

R	S	T	U	V	W	X	Y	Z	a	b	c	d	e	f	g	h	i	j
C_{21}	C_{16}	C_{11}	C_6	C_{37}	C_{32}	C_{27}	C_{22}	C_{17}	C_{12}	C_{43}	C_{38}	C_{33}	C_{28}	C_{23}	C_{18}	C_{49}	C_{44}	C_{39}
k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	0	1	2
C_{34}	C_{29}	C_{24}	C_{55}	C_{50}	C_{45}	C_{40}	C_{35}	C_{30}	C_{61}	C_{56}	C_{51}	C_{46}	C_{41}	C_{36}	C_{67}	C_{62}	C_{57}	C_{52}
3	4	5	6	7	8	9	.	,	?	:	(	)	@	!	#	/	+	=
C_{47}	C_{42}	C_{73}	C_{68}	C_{63}	C_{58}	C_{53}	C_{48}	C_{74}	C_{69}	C_{64}	C_{59}	C_{54}	C_{75}	C_{70}	C_{65}	C_{60}	C_{76}	C_{61}
*	&																	
C_{66}	C_{72}																	

Tabel ini merupakan Penerapan Metode Substitusi *Out By Diagonals*

Tabel 2. Penerapan Pola Substitusi *Out By Diagonals*

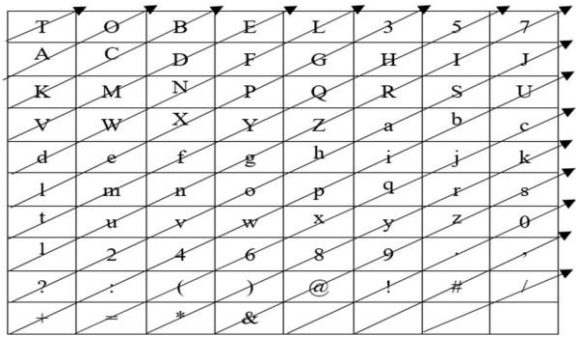


Proses *Enkripsi* dan *Dekripsi*

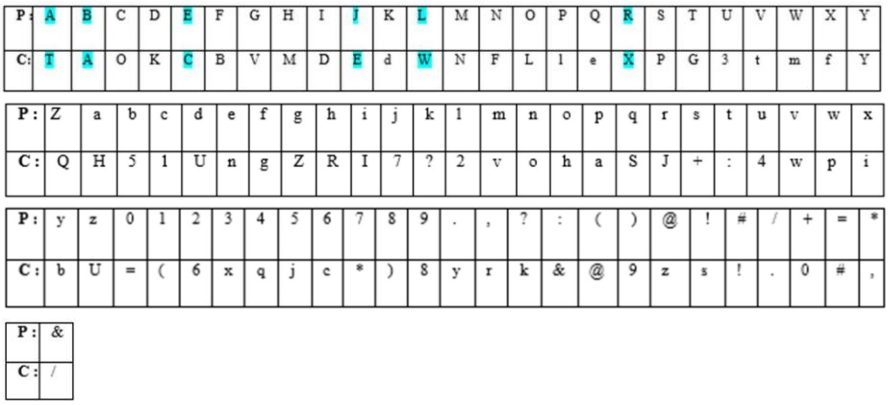
TeksProses *Enkripsi* Teks
Plaintext :
BELAJARKunci
: TOBELO357
Ciphertext: ACWTETX

Tahap 1: Membentuk matriks dengan pola Horizontal-UP dari kunci

Tabel 3. Karakter *Plaintext* dan *Ciphertext*



Tahap 2: Membentuk pasangan *plaintext* dengan *ciphertext* dari kunci pada tahap 1.



Tahap 3: Memasangkan karakter dari *plaintext* awal dengan *ciphertext* yang dilihat pada tabel tahap 2

Plaintext:	B	E	L	A	J	A	R
Ciphertext:	A	C	W	T	E	T	X

Proses *Dekripsi* Teks
Ciphertext:
ACWTETXKunci:
TOBELO357
Plaintext: BELAJAR

Tahap 1: Membentuk pasangan *plaintext* dengan *ciphertext* dari kunci

P:	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
C:	T	A	O	K	C	B	V	M	D	E	d	W	N	F	L	l	e	X	P	G	3	t	m	f	Y

P:	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
C:	Q	H	5	1	U	n	g	Z	R	I	7	?	2	v	o	h	a	S	J	+	:	4	w	p	i

P:	y	z	0	1	2	3	4	5	6	7	8	9	.	,	?	:	(	)	@	!	#	/	+	=	*
C:	b	U	=	(	6	x	q	j	c	*	)	8	y	r	k	&	@	9	z	s	!	.	0	#	,

P:	&
C:	/

Tahap 2: Memasangkan karakter dari *ciphertext* dengan *plaintext* yang dilihat pada tabel tahap 1

Ciphertext:	A	C	W	T	E	T	X
Plaintext:	B	E	L	A	J	A	R

Algoritma Substitusi Out By Diagonals Dalam Codingan

Berikut adalah langkah-langkah algoritma yang diimplementasikan dalam sistem

Pemrosesan Kunci

Langkah 1: Hapus Karakter Duplikat Dari Kunci

```
20  def remove_duplicates(key):
21      seen = set()
22      unique_key = []
23      for char in key:
24          if char not in seen:
25              seen.add(char)
26              unique_key.append(char)
27      return ''.join(unique_key)
```

Gambar 2. Hapus Karakter Duplikat Dari Kunci

Langkah 2: Hasilkan Kunci Final (76 Karakter)

```python
def generate_final_key(key):
    all_chars = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789.,?:()@!#/+=*&"
    final_key = list(key)
    for char in all_chars:
        if char not in final_key:
            final_key.append(char)
    return ''.join(final_key[:76])
```

Gambar 3. Hasilkan Kunci Final

Pembentukan Matriks Karakter

Langkah 3: Buat Matriks Dari Kunci Final

```python
def create_matrix(final_key, unique_key_length):
    cols = unique_key_length
    rows = math.ceil(76 / cols)
    matrix = [[' ' for _ in range(cols)] for _ in range(rows)]
    index = 0
    for i in range(rows):
        for j in range(cols):
            if index < len(final_key):
                matrix[i][j] = final_key[index]
                index += 1
    return matrix, rows, cols
```

Gambar 4. Buat Matriks Dari Kunci Final

Pembacaan Matriks Secara Diagonal (Out By Diagonals)

Langkah 4: Baca Matriks Dari Kiri Bawah Ke Kanan Atas

```python
def read_diagonally(matrix, rows, cols):
    ciphertext = []
    for diag in range(rows + cols - 1):
        i = min(diag, rows - 1)
        j = diag - i
        while i >= 0 and j < cols:
            if matrix[i][j] != ' ':
                ciphertext.append(matrix[i][j])
            i -= 1
            j += 1
    return ''.join(ciphertext)[:76]
```

Gambar 5. Baca Matriks Dari Kiri Bawah Ke Kanan Atas

Pembuatan Maaping Enkripsi/Dekripsi

Langkah 5: Buat Mapping

```python
def create_mapping(plaintext_chars, ciphertext_chars):
    mapping = {}
    for plain_char, cipher_char in zip(plaintext_chars, ciphertext_chars):
        mapping[plain_char] = cipher_char
    return mapping

def encrypt_text(plaintext, mapping):
    ciphertext = ""
    for char in plaintext:
        if char in mapping:
            ciphertext += mapping[char]
        else:
            ciphertext += char
    return ciphertext

def decrypt_text(ciphertext, reverse_mapping):
    plaintext = ""
    for char in ciphertext:
        if char in reverse_mapping:
            plaintext += reverse_mapping[char]
        else:
            plaintext += char
    return plaintext
```

Gambar 6. Buat Mapping

Langkah 6: Untuk Dekripsi, Balik Mapping

```python
reverse_mapping = {v: k for k, v in mapping.items()}
```

Gambar 7. Balik Mapping

Pseudocode

Algoritma Utama: Substitusi Out By

DiagonalsInput:
- mode (enkripsi/dekripsi)
- teks/file
- kunci

Langkah-langkah :

Validasi Kunci
- jika kunci kosong > error
- jika ada karakter tidak diizinkan > error

Hapus Duplikat dari kunci
unique_key ← ""
for setiap karakter c di kunci:
jika c belum ada di
unique_key:
tambahkan c ke unique_key

Hasilkan Kunci Lengkap 76 Karakter
all_chars ←
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789.,?:()@!#/+=*&"
final_key ← unique_key
for setiap karakter c di all_chars:
jika c tidak ada di final_key:
tambahkan c ke final_key
final_key ← ambil 76 karakter
pertama

Buat Matriks
cols ←
panjang(unique_key)
rows ← ceil(76 / cols)

```
        matrix ← matriks kosong ukuran rows
        × colsindex ← 0
    for i dari 0 hingga rows-1:
      for j dari 0 hingga cols-1:
            jika index < 76:
              matrix[i][j] ←
              final_key[index]index ←
              index + 1
```

Baca Matriks Secara Diagonal (Out by Diagonals)

```
      ciphertext_chars ← ""
      for diag dari 0 hingga (rows + cols - 2):
        i ← min(diag,
        rows - 1)j ← diag -
        i
        while i ≥ 0 dan j < cols:
          jika matrix[i][j] ≠ ' ':
            ciphertext_chars ← ciphertext_chars +
          matrix[i][j]i ← i - 1
          j ← j + 1
      ciphertext_chars ← ambil 76 karakter pertama
```

Buat Mapping Enkripsi

```
      plaintext_chars ←
    "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789.,?:()@!#/+=*&"
      mapping ← {}
      for k dari 0 hingga 75:
mapping[plaintext_chars[k]] ← ciphertext_chars[k]
```

Proses Enkripsi atau Dekripsi

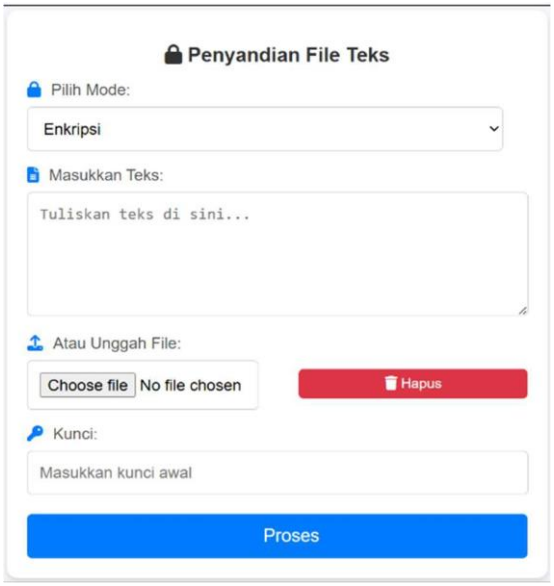
```
      jika mode = "encrypt":
        hasil ← ""
        for setiap karakter c di teks:
          jika c ada di mapping:
            hasil ← hasil +
          mapping[c]else:
            hasil ← hasil
    + cjika mode =
      "decrypt":
        reverse_mapping ← {v: k untuk setiap (k,v) di
        mapping}hasil ← ""
        for setiap karakter c di teks:
          jika c ada di reverse_mapping:
            hasil ← hasil +
          reverse_mapping[c]else:
            hasil ← hasil + c

Kembalikan hasil
```

4. Hasil Penelitian

Implementasi Sistem

Dalam penelitian ini, sistem *enkripsi* dan *dekripsi* menggunakan metode substitusi *out by diagonals* telah berhasil diimplementasikan untuk mengamankan *file* teks, Pdf dan Docx. Proses dimulai dengan identifikasi masalah yang berkaitan dengan keamanan data, di mana *file* teks yang berisi informasi sensitif rentan terhadap akses tidak sah. Rancangan sistem dilakukan dengan merancang algoritma substitusi *out by diagonals* dan diimplementasikan menggunakan aplikasi berbasis Flask.

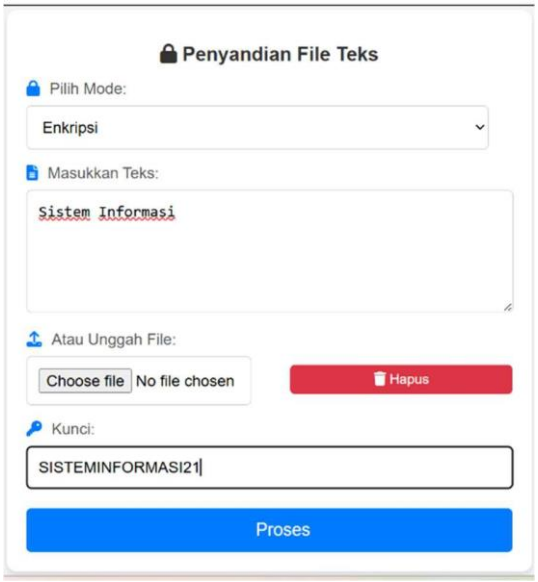
Tampilan Halaman Awal Proses Penyandian *File* Teks



Gambar 8. Tampilan Halaman Awal Penyandian Teks

Proses *Enkripsi* Teks Melalui Pengetikan

Pada Gambar 9. terlihat tampilan proses *enkripsi* teks melalui pengetikan, di mana pengguna memasukan teks yang ingin di *enkripsi* atau dilindungi "Sistem Informasi" dan memasukan kunci "SISTEMINFORMASI21" untuk proses penyandian teks dan kemudian dilanjutkan pada proses penyandian.



Gambar 9. Tampilan Proses *Enkripsi* Teks Melalui Pengetikan

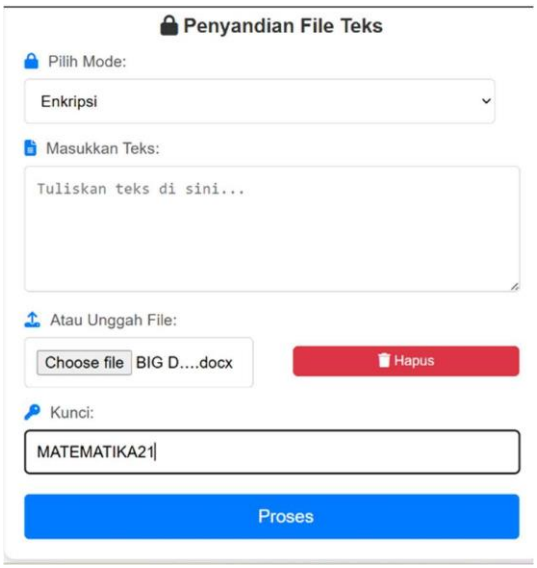
Hasil *Enkripsi* Teks Melalui Pengetikan

Gambar 10. Menampilkan hasil *enkripsi* teks melalui pengetikan yang sudah diproses oleh System dariteks asli "Sistem Informasi" menjadi teks yang sudah di *enkripsi* atau disandikan

Gambar 10. Tampilan Proses *Enkripsi* Teks Melalui

Proses *Enkripsi* Melalui Unggah *File* Docx

Gambar 11. Menampilkan proses *enkripsi* melalui unggah *file* docx, di mana pengguna mengunggah *file* yang ingin di *enkripsi* atau dilindungi dan memasukan kunci "MATEMATIKA21" untuk proses penyandian teks dan kemudian dilanjutkan pada proses penyandian.



Gambar 11. Tampilan Proses *Enkripsi* Melalui Unggah *File*

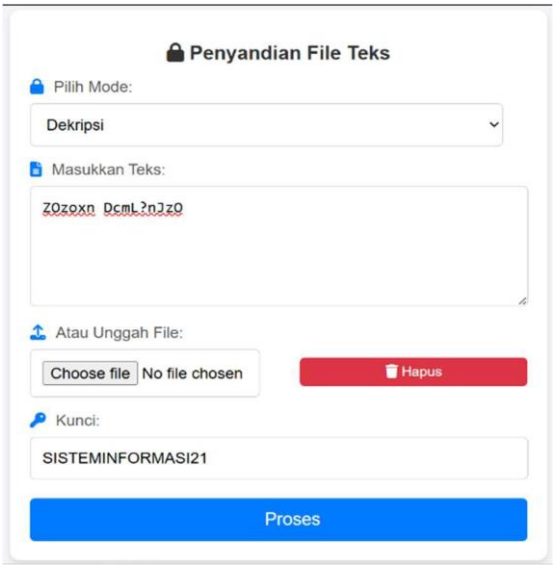
Hasil *Enkripsi* Melalui Unggah *File* Docx

Gambar 12. terlihat hasil *enkripsi* melalui unggah *file* docx yang sudah diproses oleh sistem dengan metode substitusi *out by diagonals* menjadi teks yang sudah dienkripsi atau disandikan.

Gambar 12. Tampilan Proses *Enkripsi* Melalui Unggah *File*

Proses *Dekripsi* Teks Melalui Pengetikan

Gambar 13. terlihat proses *dekripsi* teks melalui pengetikan, di mana pengguna memasukan teks yang ingin di *dekripsi* atau mengembalikan teks asli "ZOzoxn DcmL?nJzO" dan memasukan kunci "SISTEMINFORMASI21" untuk proses mengembalikan teks asli dan kemudian dilanjutkan pada proses.



Gambar 13. Tampilan Proses *Dekripsi* Teks Melalui Pengetikan

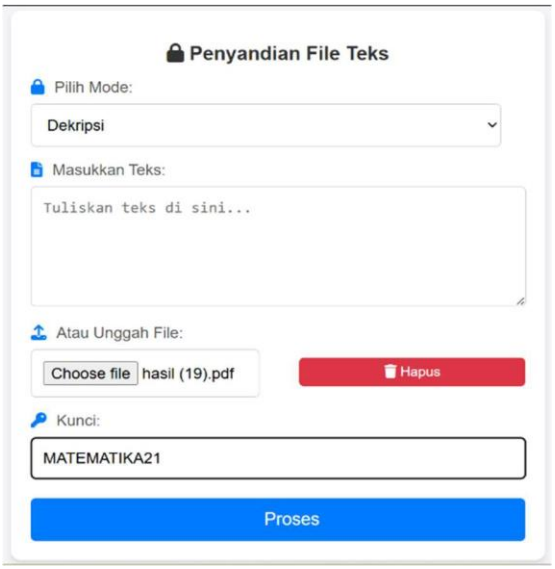
Hasil *Dekripsi* Teks Melalui Pengetikan

Gambar 14. menampilkan hasil *dekripsi* teks melalui pengetikan yang sudah diproses oleh sistem dengan metode substitusi *out by diagonals* menjadi teks asli.

Gambar 14. Tampilan Hasil Proses *Dekripsi* Teks Melalui Pengetikan

Proses *Dekripsi* Teks Melalui Unggah *File* Docx

Gambar 15. menampilkan proses dekripsi melalui unggah *file* docx, di mana pengguna mengunggah *file* yang ingin di *dekripsi* atau mengembalikan teks asli dan memasukan kunci "MATEMATIKA21" untuk proses mengembalikan teks asli dan kemudian dilanjutkan pada proses.



Gambar 15. Tampilan Proses *Dekripsi* Melalui Unggah *File* Docx

Hasil *Dekripsi* Teks Melalui Unggah *File* Docx

Gambar 16. menampilkan proses *dekripsi* melalui unggah *file* docx yang sudah diproses oleh sistem

dengan metode substitusi *out by diagonals* menjadi teks asli.

Gambar 16. Tampilan Hasil Proses *Dekripsi* Melalui Unggah *File* Docx

Pengujian Fungsi

Tabel 4. Pengujian Fungsi Dari *File* Pdf dan Docx

Teks Asli (Nama *File*)	Kunci	Hasil Enkripsi	Hasil Dekripsi
BIG DATA. DOCX	MATEMATIKA21 (Benar)	Berhasil	Berhasil
Tugas BHS INDO. PDF	Manado17 (Benar)	Berhasil	Berhasil
Parameter .DOCX	Jakarta%$ (Salah)	Tidak Berhasil	Tidak Berhasil
Dinamika .PDF	<jogja>82 (Salah)	Tidak Berhasil	Tidak Berhasil

Hasil pengujian menunjukkan bahwa semua fungsi bekerja sesuai dengan metode substitusi *out by diagonals*, menghasilkan *ciphertext* yang sesuai dari *plaintext* dan dapat mengembalikan teks asli yang sudah dienkripsi.

5. Kesimpulan

Berdasarkan hasil penelitian dan pembahasan yang telah dilakukan, dapat disimpulkan bahwa metode substitusi out by diagonals berhasil diimplementasikan untuk mengamankan file teks dalam format .pdf dan .docx. Proses enkripsi menggunakan pola diagonal dalam matriks karakter menghasilkan ciphertext yang sulit didekripsi tanpa kunci yang sesuai, serta mampu merekonstruksi teks asli secara baik melalui proses dekripsi.

Untuk pengembangan lebih lanjut, dapat dilakukan penambahan jumlah karakter dalam proses enkripsi yang akan memperluas cakupam sistem sehingga mampu mendukung berbagai jenis data dengan format yang lebih kompleks dan sistem juga dapat dikembangkan dengan mengintegrasikan algortima kriptografi modern seperti AES (Advanced Encryption Standard) atau RSA (Rivest-Shamir- Adleman) yang terbukti handal dalam menjaga keamanan informasi digital.

6. References

[1] A. H. Satria Nusantara, I. Kahirul Umam, and M. Lubis, "Jaminan Informasi dan Keamanan yang Lebih Baik: Studi Kasus BPJS Kesehatan," *Nuansa Inform.*, vol. 18, no. 2, pp. 120–127, 2024, doi: 10.25134/ilkom.v18i2.202.

[2] A. Z. F. Rangkuti and H. Fahmi, "Implementasi Kriptografi Untuk Keamanan File Text Dengan Menggunakan Metode MD5," *J. Nas. Komputasi dan Teknol. Inf.*, vol. 3, no. 2, pp. 170–175, 2020, doi: 10.32672/jnkti.v3i2.2384.

[3] Taufan Maynard Prananda Sancaka1 and Veronica Lusiana, "Penerapan Metode Playfair Cipher Dalam Aplikasi Enkripsi- Dekripsi File Teks," *Elkom J. Elektron. dan Komput.*, vol. 15, no. 2, pp. 260–270, 2022, doi: 10.51903/elkom.v15i2.937.

[4] N. Siregar and A. Usman, "Penerapan Algoritma Kriptografi Hybrid Subtitusi dan Transposisi Spiral Dalam Mengamankan Data Teks," vol. 6341, no. April, pp. 81–90, 2021.

[5] H. Idrus et al., "[99-106]+Peningkatan+Pengamanan+Pesan+Pribadi+Menggunakan+Kriptografi+Klasik+Berbasis+Algoritma +Subtitusi+Chiper," vol. 01, no. September, pp. 99–106, 2023.

[6] A. Aprianto, E. I. Alwi, and H. Herman, "Implementasi Algoritma Caesar Cipher Dengan Kombinasi Transposisi Diagonal Untuk Enkripsi Dekripsi Menggunakan Tabel ASCII," Bul. Sist. Inf. dan Teknol. Islam, vol. 3, no. 3, pp. 238–247, 2022, doi: 10.33096/busiti.v3i3.1142.

[7] E. Ketaren et al., "Modifikasi Algoritma Vigenere Cipher Untuk Pengamanan File," vol. XIII, no. 2, pp. 319–325, 2024.

[8] L. Oktasari, “Perbandingan Algoritma RC4 + dan RC4a dalam Pengamanan File Teks,” Bull. Comput. Sci.
Res., vol. 1, no. 3, pp. 83–92, 2021, [Online]. Available: https://hostjournals.com/bulletincsr
[9] M. Liesdiani, “Sistem Kriptografi pada Citra Digital Menggunakan Metode Substitusi dan Permutasi,” Pros. SNATIKA, vol. 4, pp. 24–31, 2017, [Online]. Available: https://jurnal.stiki.ac.id/SNATIKA/article/view/140
[10] M. A. M., “Komunikasi Berbasis Teks,” J. Pseudocode, vol. III, no. September, pp. 129–136, 2016.
[11] S. P. Ananda and S. Lukman, “Analisa Metode Kriptografi Modern Advance Encryption Standard (AES) 128 Bit dalam Mengenkripsi dan Mendekripsi File Dokumen Digital,” J. Ilm. Komputasi, vol. 21, no. 3, pp. 333–344, 2022, doi: 10.32409/jikstik.21.3.2973.
[12] D. Pradeka, “Implementasi Aplikasi Kriptografi Berbasis Android Menggunakan Metode Subtitusi Dan Permutasi,” In Search, vol. 18, no. 1, pp. 161–168, 2019, doi: 10.37278/insearch.v18i1.148.
[13] H. A. Fauzi, K. A. Putra, and A. Tri, “Analisis Perbandingan Performa Web Service Menggunakan Bahasa Pemrograman Python , Php , dan Perl pada Client Berbasis Android,” J. Pengemb. Teknol. Inf. dan Ilmu Komput., vol. 2, no. 1, pp. 237–245, 2018, [Online]. Available: https://j-ptiik.ub.ac.id/index.php/j- ptiik/article/view/781
[14] Y. T. Bota and N. Setiyawati, “Pengembangan Sistem Informasi Perantara Bisnis Menggunakan Framework Flask,” J. Inf. Technol. Ampera, vol. 3, no. 2, pp. 79–93, 2022, doi: 10.51519/journalita.volume3.isssue2.year2022.page79-93.