Pengamanan Data Tekstual dengan Kombinasi *Vigenere Cipher* dan *Caesar Cipher*

Luh Arimas Pertiwi[a1], Ngurah Agus Sanjaya ER[a2]

[a]Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Udayana, Bali
Jln. Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, 08261, Bali, Indonesia
[1]arimaspertiwi@gmail.com
[2]agus_sanjaya@unud.ac.id

Abstract

Problems in data security is an important aspect in maintaining data storage, especially data stored in digital form. This is due to very rapid progress in the field of computer science with the open-system concept that has been widely used, so that this can make it easier for someone to destroy data, especially data stored in digital form without having to be known by the data custodian. In this case the researcher found a problem in using one algorithm, namely the Caesar cipher in data security, where there is a Brute force attack that tries all possible key combinations to crack a password. In the context of the Caesar Cipher, brute force can be used to try all possible shifts of letters and find a key that produces a plausible decrypted text. This study aims to maximize the security of textual data by combining two algorithms in it, in which the algorithm used is the Vigenere Cipher and the Caesar Cipher. The result of this research is that textual data that is secured becomes more difficult to understand for third parties who may want to manipulate data.

Keywords: *Textual Data, Vigenere Cipher, Caesar Cipher*

1. Pendahuluan

Pengamanan data adalah suatu aspek penting terutama data yang berbentuk digital, dalam hal ini data yang diamanankan lebih difokuskan pada data yang berbentuk teks. Pengamanan dapat dilakukan dengan enkripsi dan dekripsi data. Peneliti menemukan satu permasalahan dalam pengamanan data hanya dengan menggunakan 1 algoritma, yaitu *Caesar cipher*. Dikatakan bahwa *Caesar cipher* mempunyai kelemahan karena mudah diretas dengan metode brute force. Metode *brute force* yang paling sering digunakan adalah dengan menggunakan statistika frekuensi kemunculan huruf yang paling sering muncul. [4] Karena *Caesar Cipher* bekerja hanya dengan melakukan pergeseran karakter, sehingga memungkinkan untuk dipecahkan dengan menggunakan *brute force*. Karena hal ini maka peneliti mencoba meningkatkan pengamanan data tekstual dengan menggabungkan dua algoritma yaitu *Vigenere Cipher* dengan *Caesar Cipher.*

Kajian sebelumnya telah menggunakan *Caesar Cipher* pada fitur pesan teks (Samsuriah, 2023), menyatakan bahwa Proses penyandian dengan algoritma *Caesar Cipher* berhasil digunakan untuk menyembunyikan pesan dan dapat mengembalikan pesan tersebut seperti semula. Program hanya memproses karakter A hingga Z di karenakan penggunaan angka 26. Namun, penelitian ini masih sangat minim, karena peneliti mungkin belum menelusuri lebih dalam bahwa jika dengan menggunakan 1 algoritma saja apalagi algoritma *Caesar cipher*, data akan mudah diretas dengan metode *brute force*. [4]

Penelitian ini akan menggabungkan 2 algoritma yaitu *Vigenere Cipher* dengan *Caesar Cipher* dalam pengamanan data tekstual. Hal yang akan dilakukan yaitu enkripsi dan dekripsi data trekstual. Langkah – Langkah yang dilakukan saat enkripsi yaitu, pertama teks akan di proses pada vigenere cipher, dimana di tahap ini akan dimasukkan plainteks (p1) dan akan dibuatkan kunci. Setelah dihasilkan hasil enkripsi berupa cipherteks (c1) pada tahap pertama, *Cipherteks*

(c1) akan dienkripsi lagi dengan *Caesar cipher* sehingga akan menghasilkan *Cipherteks* akhir (c2). Kedua, ada tahapan dekripsi yaitu mengubah *cipherteks* menjadi *plainteks* (teks asli). Langkah yang dilakukan pada tahap dekripsi yaitu cipherteks (c2) akan diproses terlebih dahulu pada Caesar Cipher, dengan melakukan pergeseran kunci lagi maka akan mendapatkan plainteks (p2), setelah itu plainteks (p2) akan diproses lagi dalam *vigenere cipher*, sehingga mendapapatkan hasil akhir yaitu plainteks awal (p1). Dengan adanya tingkat keamanan data yang rendah pada data berupa teks maka penelitian ini diharapkan dapat memberikan prosedur pengamanan data yang lebih kuat dengan kombinasi *vigenere cipher* dan *Caesar cipher* ini.

2. Metode Penelitian

2.1 Gambaran Umum System

System ini menggunakan dua buah algoritma untuk pengamanan data teks. Yang pertama yaitu Algoritma *Vegenere Cipher*, sedangkan yang kedua yaitu Algoritma *Caesar Cipher.*

- Pertama *plainteks* akan diproses melalui *Vigenere Cipher*, lalu dilakukan pemilihan kata kunci. Kata kunci ini dapat terdiri dari alfabet dan dapat berulang atau memiliki panjang yang sama dengan pesan yang akan dienkripsi.
- Selanjutnya, *plainteks* dan kata kunci akan dikonversi menjadi angka sesuai dengan posisi huruf dalam alfabet pada gambar 2.1.
- Setelah terbentuk *cipherteks* (c1), maka akan dilanjutkan enkripsi kedua dengan Algoritma *Caesar Cipher*.
- Enkripsi kedua (c1) yang dilakukan pada Algoritma *Caesar Cipher* ini dilakukan dengan pengacakan pergeseran kunci yang ditetapkan.
- Setelah melakukan kedua tahap tersebut, maka akan terbentuk cipherteks akhir (c2)
- Sekarang untuk dekripsi *cipherteks*, dilakukan dengan kebalikan semua proses enkripsi tadi
- Pertama, lakukan dekripsi dengan algoritma *cipherteks* dengan penggeseran kata menggunakan kunci yang digunakan, sehingga mendapatkan *plainteks* (p2)
- Plainteks (p2) selanjutnya akan diproses dengan algoritma *Vigenere cipher* Kembali dengan menggunakan kata kunci sesuai dengan yang digunakan pada tahap dekripsi.
- Sehingga setelah melalui semua tahap dekripsi diatas, akan dihasilkan plainteks awal (teks asli).

Gambar 1. Konversi huruf alfabet

2.2 Rumus yang digunakan

a. Vigenere Cipher

Sandi *Vigenère* adalah metode menyandikan teks alfabet dengan menggunakan deretan sandi Caesar berdasarkan huruf-huruf pada kata kunci. Sandi *Vigenère* merupakan bentuk sederhana dari sandi substitusi polialfabetik. Kelebihan sandi ini dibanding sandi Caesar dan sandi monoalfabetik lainnya adalah sandi ini tidak begitu rentan terhadap metode pemecahan sandi yang disebut analisis frekuensi. [5]

Rumus enkripsi dan dekripsi vegenere cipher:

Enkripsi: $\boldsymbol{Ci = (Pi + Ki)\ mod\ 26}$
Dekripsi: $\boldsymbol{Pi = (Ci - Ki)\ mod\ 26}$

Keterangan:

Ci = nilai desimal karakter ciphertext ke-i
Pi = nilai desimal karakter plaintext ke-i
Ki = nilai desimal karakter kunci ke-i

3. Caesar Cipher

Dalam kriptografi, sandi Caesar, atau sandi geser, kode Caesar atau Geseran Caesar adalah salah satu teknik enkripsi paling sederhana dan paling terkenal. Sandi ini termasuk sandi substitusi dimana setiap huruf pada teks terang (plaintext) digantikan oleh huruf lain yang memiliki selisih posisi tertentu dalam alfabet. [3]

Rumus enkripsi dan dekripsi caesar cipher:

Enkripsi: $\boldsymbol{Ci = (Pi + Key)\ mod\ 26}$
Dekripsi: $\boldsymbol{Pi = (Ci - Key)\ mod\ 26}$

Keterangan:

Ci = ciphertext ke-i
Pi = plaintext ke-i
Key = kunci

Contoh:

Pi = UJIAN PROPOSAL
Ki = SNATIA
Key = 3

Jawab:

Perhitungan manual dengan table konversi pada gambar 2.1

Enkripsi:

Tabel 1. Enkripsi

Pi	U	J	I	A	N	P	R	O	P	O	S	A	L
	20	9	8	0	13	15	17	14	15	14	18	0	11
Ki	S	N	A	T	I	A	S	N	A	T	I	A	S
	18	13	0	19	8	0	18	13	0	19	8	0	18
+	12	22	8	19	21	15	9	1	15	7	0	0	3
Ci	M	W	I	T	V	P	J	B	P	H	A	A	D
Key =3 (Ci2)	P	Z	L	W	Y	S	M	E	S	K	D	D	G

Jadi *cipherteks* yang didapat dari hasil enkripsi yaitu PZLWYSMESKDDG

Dekripsi:

Tabel 2. Dekripsi

Ci2	P	Z	L	W	Y	S	M	E	S	K	D	D	G

Ci	M	W	I	T	V	P	J	B	P	H	A	A	D
	12	22	8	19	21	15	9	1	15	7	0	0	3

Ki	S	N	A	T	I	A	S	N	A	T	I	A	S
	18	13	0	19	8	0	18	13	0	19	8	0	18
Pi	U	J	I	A	N	P	R	O	P	O	S	A	L
	20	9	8	0	13	15	17	14	15	14	18	0	11

Jadi *plainteks* yang didapat dari hasil dekripsi yaitu Ujian Proposal

Perhitungan manual dengan rumus:

Pi = UJIAN PROPOSAL
Ki = SNATIA

Enkripsi dengan vigenere cipher:

- Ci = (Pi + Ki) mod 26
  = (U + S) mod 26
  = (20 + 18) mod 26
  = 38 mod 26
  = 12 => M
- Ci = (Pi + Ki) mod 26
  = (J + N) mod 26
  = (9 + 13) mod 26
  = 22 mod 26
  = 22 => W
- Ci = (Pi + Ki) mod 26
  = (I + A) mod 26
  = (8 + 0) mod 26
  = 8 mod 26
  = 8 =>I
- Ci = (Pi + Ki) mod 26
  = (A + T) mod 26
  = (0 + 19) mod 26
  = 19 mod 26
  = 19 => T
- Ci = (Pi + Ki) mod 26
  = (N + I) mod 26
  = (13 + 8) mod 26
  = 21 mod 26
  = 21 =>V
- Ci = (Pi + Ki) mod 26
  = (P + A) mod 26
  = (15 + 0) mod 26
  = 15 mod 26
  = 15 =>P
- Ci = (Pi + Ki) mod 26
  = (R + S) mod 26
  = (17 + 18) mod 26
  = 35 mod 26
  = 9 => J
- Ci = (Pi + Ki) mod 26
  = (O + N) mod 26
  = (14 + 13) mod 26
  = 27 mod 26
  = 1 =>B
- Ci = (Pi + Ki) mod 26
  = (P + A) mod 26
  = (15 + 0) mod 26
  = 15 mod 26
  = 15 =>P
- Ci = (Pi + Ki) mod 26
  = (O + T) mod 26
  = (14 + 19) mod 26
  = 33 mod 26
  = 7 =>H
- Ci = (Pi + Ki) mod 26
  = (S + I) mod 26
  = (18 + 8) mod 26
  = 26 mod 26
  = 0 => A
- Ci = (Pi + Ki) mod 26
  = (A+ A) mod 26
  = (0 + 0) mod 26
  = 0 mod 26
  = 0 => A
- Ci = (Pi + Ki) mod 26
  = (L + S) mod 26
  = (11 + 18) mod 26
  = 29 mod 26
  = 3 => D

Hasil *cipherteks* dengan algoritma *vigenere cipher*: MWITVPJBPHAAD

Enkripsi dengan caesar cipher:

- Ci = (Pi + Key) mod 26
  = (M + 3) mod 26
  = (12+ 3) mod 26
  = 15 mod 26
  = 15 => P
- Ci = (Pi + Key) mod 26
  = (W + 3) mod 26
  = (22 + 3) mod 26
  = 25 mod 26
  = 25 => Z
- Ci = (Pi + Key) mod 26
  = (I + 3) mod 26
  = (8 + 3) mod 26
  = 11 mod 26
  = 11 => L
- Ci = (Pi + Key) mod 26
  = (T + 3) mod 26
  = (19 + 3) mod 26
  = 22 mod 26
  = 22 => W
- Ci = (Pi + Key) mod 26
  = (V + 3) mod 26
  = (21 + 3) mod 26
  = 24 mod 26
  = 24 => Y
- Ci = (Pi + Key) mod 26
  = (P + 3) mod 26
  = (15 + 3) mod 26
  = 18 mod 26
  = 18 => S
- Ci = (Pi + Key) mod 26
  = (J + 3) mod 26
  = (9 + 3) mod 26

= 12 mod 26
= 12 => M
- Ci = (Pi + Key) mod 26
= (B + 3) mod 26
= (1 + 3) mod 26
= 4 mod 26
= 4 => E
- Ci = (Pi + Key) mod 26
= (P + 3) mod 26
= (15 + 3) mod 26
= 18 mod 26
= 18 => S
- Ci = (Pi + Key) mod 26
= (H + 3) mod 26
= (7 + 3) mod 26
= 10 mod 26
= 10 => K
- Ci = (Pi + Key) mod 26
= (A + 3) mod 26
= (0 + 3) mod 26
= 3 mod 26
= 3 => D
- Ci = (Pi + Key) mod 26
= (A + 3) mod 26
= (0 + 3) mod 26
= 3 mod 26
= 3 => D
- Ci = (Pi + Key) mod 26
= (D + 3) mod 26
= (3 + 3) mod 26
= 6 mod 26
= 6 => G

Jadi hasil *cipherteks* akhir yaitu PZLWYSMESKDDG

Dekripsi dengan caesar cipher:

- Pi = (Ci - Key) mod 26
= (P - 3) mod 26
= (15 - 3) mod 26
= 12 mod 26
= 12 => M
- Pi = (Ci - Key) mod 26
= (Z - 3) mod 26
= (25 - 3) mod 26
= 22 mod 26
= 22 => W
- Pi = (Ci - Key) mod 26
= (L - 3) mod 26
= (11 - 3) mod 26
= 8 mod 26
= 8 => I
- Pi = (Ci - Key) mod 26
= (W - 3) mod 26
= (22 - 3) mod 26
= 19 mod 26
= 19 => T
- Pi = (Ci - Key) mod 26
= (Y - 3) mod 26
= (24 - 3) mod 26
= 21 mod 26
= 21 => V
- Pi = (Ci - Key) mod 26
= (S - 3) mod 26
= (18 - 3) mod 26
= 15 mod 26
= 15 => P
- Pi = (Ci - Key) mod 26
= (M - 3) mod 26
= (12 - 3) mod 26
= 9 mod 26
= 9 => J
- Pi = (Ci - Key) mod 26
= (E - 3) mod 26
= (4 - 3) mod 26
= 1 mod 26
= 1 => B
- Pi = (Ci - Key) mod 26
= (S - 3) mod 26
= (18 - 3) mod 26
= 15 mod 26
= 15 => P
- Pi = (Ci - Key) mod 26
= (K - 3) mod 26
= (10 - 3) mod 26
= 7 mod 26
= 7 => H
- Pi = (Ci - Key) mod 26
= (D - 3) mod 26
= (3 - 3) mod 26
= 0 mod 26
= 0 => A
- Pi = (Ci - Key) mod 26
= (D - 3) mod 26
= (3 - 3) mod 26
= 0 mod 26
= 0 => A
- Pi = (Ci - Key) mod 26
= (G - 3) mod 26
= (6 - 3) mod 26
= 3 mod 26
= 3 => D

Hasil plainteks dengan algoritma *caesar cipher*: MWITVPJBPHAAD

Dekripsi dengan vigenere cipher:

- Pi = (Ci - Ki) mod 26
= (M - S) mod 26
= (12 - 18) mod 26
= 20 mod 26
= 20 => U
- Pi = (Ci - Ki) mod 26
= (W - N) mod 26
= (22 - 13) mod 26
= 9 mod 26
= 9 => J
- Pi = (Ci - Ki) mod 26
= (I – A) mod 26
= (8 - 0) mod 26
= 8 mod 26
= 8 => I
- Pi = (Ci - Ki) mod 26
= (T - T) mod 26
= (19 - 19) mod 26
= 0 mod 26
= 0 => A
- Pi = (Ci - Ki) mod 26
= (V - I) mod 26
= (21 - 8) mod 26
= 13 mod 26
= 13 => N
- Pi = (Ci - Ki) mod 26
= (P - A) mod 26
= (15 - 0) mod 26
= 15 mod 26
= 15 => P
- Pi = (Ci - Ki) mod 26
= (J - S) mod 26
= (9 - 18) mod 26
= 17 mod 26
= 17 => R
- Pi = (Ci - Ki) mod 26
= (B - N) mod 26
= (1 - 13) mod 26
= 14 mod 26
= 14 => O
- Pi = (Ci - Ki) mod 26
= (P - A) mod 26

= (15 - 0) mod 26
= 15 mod 26
= 15 => P

- Pi = (Ci - Ki) mod 26
  = (H - T) mod 26
  = (7 - 19) mod 26
  = 14 mod 26
  = 14 => O
- Pi = (Ci - Ki) mod 26
  = (A - I) mod 26
  = (0 - 8) mod 26
  = 18 mod 26
  = 18 => S
- Pi = (Ci - Ki) mod 26
  = (A - A) mod 26
  = (0 - 0) mod 26
  = 0 mod 26
  = 0 => A
- Pi = (Ci - Ki) mod 26
  = (D - S) mod 26
  = (3 - 18) mod 26
  = 11 mod 26
  = 11 => L

Jadi hasil dekripsi menjadi *plainteks* asli yaitu Ujian Proposal

2.3 Perancangan System

Perancangan system dijelaskan dengan flowchart. Secara garis besar jalannya system dibagi menjadi 2 proses yaitu proses enkripsi dan proses dekripsi, yang dijelaskan pada gambar 2 untuk enkripsi dan gambar 3 untuk dekripsi.

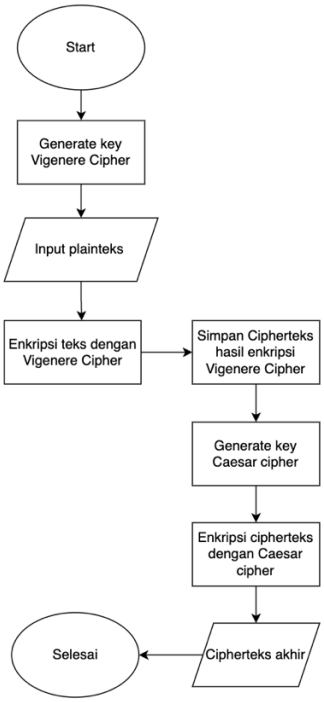
2.4 Pengujian System

Pengujian sistem dimaksudkan untuk mengetahui seberapa jauh tingkat keberhasilan dalam mengamankan file yang berupa teks serta untuk menguji apakah sistem yang telah dibuat berjalan tanpa mengalami *erro*r. Pengujian kali ini tidak serta mengujikan aplikasi dikarenakan aplikasi untuk sistem ini masih dalam tahap perancangan.

4. Hasil dan Pembahasan

4.1 Perancangan System

a. Enkripsi

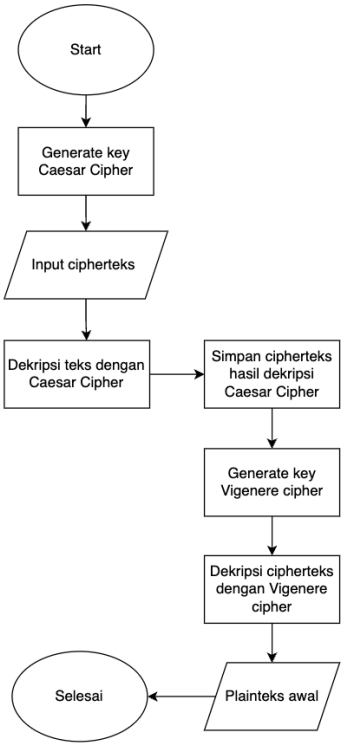


Gambar 2. Flowchart Enkripsi *Plainteks*

Proses enkripsi dimulai dengan memasukkan plainteks dan juga kunci *vigenere cipher*. Kunci yang dimasukkan berupa huruf alfabet yang nantinya akan diproses bersamaan

dengan plainteks. Setelah itu dilanjutkan dengan pembangkitan kunci *Caesar cipher*. Setelah semua proses tadi dilakukan maka akan terbentuk cipherteks akhir.

b. Dekripsi



Gambar 3. Flowchart Dekripsi Cipherteks

Proses dekripsi dimulai dengan menginputkan kunci *Caesar cipher* dengan membangkitkan kuncinya. Setelah itu dilanjutkan dengan pembangkitan kunci *Vigenere Cipher*, maka setelah melalui proses tadi akan dihasilkan plainteks awal.

4.2 Pengujian Sistem

```
Masukkan teks: UJIAN PROPOSAL
Apa yang ingin Anda lakukan? (enkripsi/dekripsi): enkripsi
Masukkan kunci Vigenere: SNATIA
Masukkan pergeseran Caesar: 3
Teks terenkripsi: PZLWY SMESKDDG
```

Gambar 4. Output Hasil Enkripsi

Diberikan salah satu contoh output hasil enkripsi teks seperti pada gambar 4. Dimasukkan plainteks UJIAN PROPOSAL dengan kunci vigenere SNATIA dan kunci pergeseran *Caesar cipher* sebanyak 3 kali maka akan menghasilkan Cipherteks PZLWY SMESKDDG

```
Masukkan teks: PZLWY SMESKDDG
Apa yang ingin Anda lakukan? (enkripsi/dekripsi): dekripsi
Masukkan kunci Vigenere: SNATIA
Masukkan pergeseran Caesar: 3
Teks terdekripsi: UJIAN PROPOSAL
```

Gambar 5. Output Hasil Dekripsi

Selanjutkan diberikan contoh output hasih dekripsi *cipherteks* seperti pada gambar 5. Diinputkan *Chiperteks* PZLWY SMESKDDG dengan kunci vigenere dan ciperteks yang sama saat enkripsi tadi, maka akan menghasilkan plainteks awal.

5. Kesimpulan

Dari penelitian yang telah dilakukan, kombinasi algoritma *Vigenere Cipher* dengan *Caesar Cipher* dapat diimplementasikan dalam enkripsi dan dekripsi teks. Tahap enkripsi dijalankan dengan algoritma vigenere terlebih dahulu, lalu dilanjutkan dengan algoritma Caesar cipher. Sedangkan tahap dekripsi dijalankan dengan algoritma *Caesar cipher* dahulu lalu dilanjutkan dengan algoritma vigenere cipher. Dengan penggabungan kedua algoritma ini membuktikan bahwa keamanan data akan semakin kuat terjaga, karena tidak hanya menggunakan 1 algoritma saja untuk melindungi data.

Daftar Pustaka

[1] Alasi, T. S., & Fitriani, P. (2022). Peningkatan Keamanan untuk Password menggunakan Algoritma Vigenere Cipher. *Jurnal Mantik Penusa*, *6*(1), 1-10.
[2] Gunawan, I. (2018). Kombinasi algoritma Caesar cipher dan algoritma RSA untuk pengamanan file dokumen dan pesan teks. *InfoTekJar: Jurnal Nasional Informatika dan Teknologi Jaringan*, *2*(2), 124-129.
[3] Kriptografi untuk Keamanan Data. (2018). (n.p.): Deepublish
[4] Samsuriah, Ida. (2023). Penerapan Kriptografi Caesar Cipher Pada Fitur Pesan Teks. Nusantara Hasana Journal : Volume 2 No. 9 (Februari 2023), Page: 254-259 E-ISSN : 2798-1428
[5] *Wikipedia.com*. (2022, 11). Diambil kembali dari https://id.wikipedia.org/wiki/Sandi_Vigen%C3%A8re (2022, November).