Original Article

e-ISSN: 2581-0545 - https://journal.itera.ac.id/index.php/jsat/

Modifikasi Algoritma Kriptografi Klasik dengan Implementasi *Deterministic Finite Automata* melalui Partisi Pesan Asli berdasarkan Kriteria Pesan Bagian

Received 13th November 2020
Accepted 18th December 2020
Published 20th December 2020

Open Access

DOI: 10.35472/ jsat.v4i2.346

Angga Wijaya [*a]

[a] *Program Studi Teknik Informatika Institut Teknologi Sumatera*
Jl. Terusan Ryacudu, Way Huwi, Kec. Jati Agung, Kabupaten Lampung Selatan, Lampung 35365
email : angga.wijaya@if.itera.ac.id

Abstract: Classical cryptography is study of securing a secret message (plaintext) into a hidden message (ciphertext) which in the process changes each character. The process of converting plaintext into ciphertext is called encryption, the reverse process is called decryption. There is monoalphabetic algorithm, which changes each plaintext letter paired wisely with one particular letter in the ciphertext. The weakness of this algorithm is that the encryption rules can be guessed by analysing the frequency of occurrence of letters. Meanwhile, in polyalphabetic cipher, an algorithm that allows the same letter to be encrypted into different letters. One of them is the Vigenere cipher, which uses keywords that can be repeated to add to the plaintext in the computation of integer modulo. However, this algorithm can be solved using the Kasiski method, because pattern of repeating keywords. In this research, a Deterministic Finite Automata computation model is applied to partition messages before being processed with keywords. Partitions that are formed based on the criteria of message parts ending by the letter E, as letters that often appear in English text. This can increase the security of classical cryptographic algorithms, because they cannot be attacked by frequency analysis or Kasiski methods.

Keywords: *classical cryptography, finite automata, monoalphabetic cipher, vigenere cipher*

Abstrak: Kriptografi klasik merupakan ilmu untuk mengamankan pesan rahasia (plainteks) menjadi pesan tersamarkan (cipherteks) yang dalam prosesnya dilakukan pengubahan tiap karakter. Proses mengubah plainteks menjadi cipherteks disebut enkripsi, sementara proses sebaliknya disebut dekripsi. Terdapat algoritma monoalfabetik cipher, yaitu algoritma yang mengubah setiap huruf plainteks dipasangkan secara bijektif dengan satu huruf tertentu di cipherteks. Kelemahan algoritma ini yaitu aturan enkripsi dapat diterka dengan analisis frekuensi kemunculan huruf. Sementara itu pada polialfabetik cipher, algoritma yang memungkinkan huruf yang sama akan dienkripsi menjadi huruf yang berbeda. Salah satunya Vigenere cipher, menggunakan kata kunci yang dapat diulang untuk dijumlahkan dengan plainteks dalam perhitungan modulo bilangan bulat. Namun algoritma ini masih dapat dipecahkan dengan metode Kasiski, dikarenakan pola perulangan kata kunci yang teratur secara matematis. Dalam penelitian ini diterapkan model komputasi *Deterministic Finite Automata* untuk mempartisi pesan sebelum diproses dengan kata kunci. Partisi yang dibentuk berdasarkan kriteria bagian pesan yang diakhiri oleh huruf E, sebagai huruf yang sering muncul dalam teks bahasa Inggris. Hal ini dapat meningkatkan keamanan algoritma kriptografi klasik, sebab tidak dapat diserang dengan analisis frekuensi maupun metode Kasiski.

Kata Kunci : kriptografi klasik, monoalfabetik cipher, vigenere cipher, finite automata

Pendahuluan

Pentingnya suatu pesan rahasia untuk dapat disampaikan kepada penerima yang berhak dan pada akhirnya dapat dipahami maknanya. Namun pesan ini dikirimkan secara tersamar sehingga tidak dapat diketahui dengan mudah secara langsung khususnya bagi pihak yang tidak berhak. Pesan tersebut dapat diketahui maknanya setelah melalui perhitungan matematis atau algoritma tertentu yang telah disepakati oleh pengirim dan penerima. Kriptografi yang secara terminologis artinya menulis secara rahasia, memberikan layanan akan hal itu melalui beberapa algoritma yang digunakan dalam proses pengubahan dari pesan asli (plainteks) menjadi pesan yang tersamarkan (cipherteks). Proses mengubah plainteks menjadi cipherteks disebut enkripsi, sementara proses mengembalikan cipherteks menjadi plainteks disebut dekripsi, seperti yang ditulis pada literatur [1], [2].

Algoritma kriptografi klasik dalam prosesnya mengenkripsi atau mengubah plainteks menjadi cipherteks secara karakter per karakter atau huruf per huruf. Salah satu contoh algoritma kriptografi klasik adalah algoritma Caesar cipher. Dimana algoritma ini melakukan pergeseran ke kanan sejumlah nilai k tertentu sebagai kuncinya. Apabila pergeserannya melebihi huruf Z maka akan kembali lagi ke huruf A (dalam hal ini untuk alfabet 26 huruf**)** [2]**,** [3]**.** Algoritma ini termasuk monoalfabetik cipher, dimana setiap huruf alfabet selalu dienkripsi menjadi huruf cipherteks yang sama. Hal ini menjadi kelemahan dari cipher ini karena kuncinya atau pesan aslinya dapat dideteksi dengan mudah oleh pihak asing yang tidak berhak. Untuk Caesar Cipher kunci yang mungkin ada 26 kunci, sehingga pihak penyerang hanya perlu mencoba 26 kemungkinan saja untuk mengetahui pesan aslinya. Sementara untuk monoalfabetik cipher secara umum, kemungkinan kunci yang ada sebanyak 26! (26 faktorial). Meskipun kemungkinannya sangat banyak, namun algoritma ini dapat dipecahkan dengan menggunakan analisis frekuensi kemunculan huruf. Dalam teks berbahasa Inggris frekuensi kemunculan huruf yang sering muncul adalah huruf E, T, A, O, I, N, S, H, R, D, L, U, dan seterusnya. Hal ini akan membuat huruf cipherteks yang bersesuaian juga akan muncul dengan urutan frekuensi kemunculan yang sama. Sehingga dengan mengetahui bahasa yang digunakan dalam plainteks, pihak lain dapat menerka aturan enkripsi tiap tiap hurufnya [4].

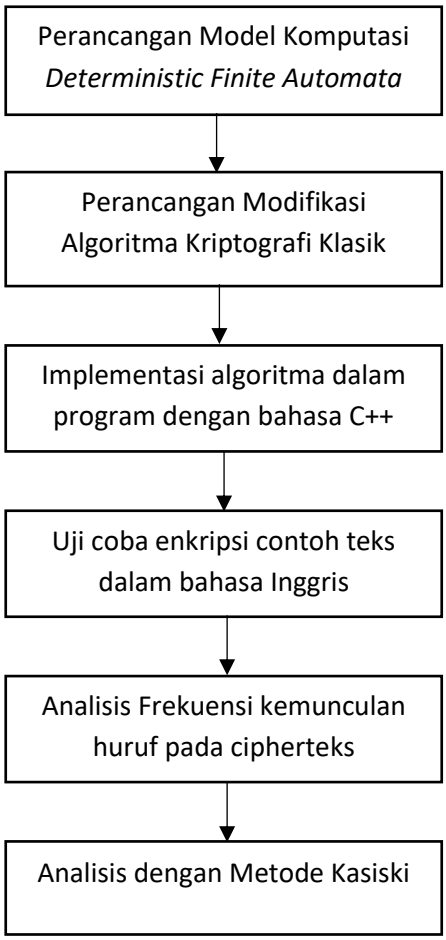
Contoh lain algoritma kriptografi klasik yaitu Vigenere Cipher. Proses enkripsi dilakukan dengan menjumlahkan setiap huruf pada plainteks dengan huruf pada kata kunci sesuai posisi/urutannya. Kunci yang digunakan adalah suatu kata atau rangkaian huruf. Apabila panjang kunci lebih pendek dari plainteks maka kata kunci akan diulangi terus. Sehingga huruf yang sama pada plainteks akan berkemungkinan dienkripsi ke huruf yang berbeda pada cipherteks. Dengan algoritma ini tentu memiliki tingkat keamanan yang lebih tinggi dari pada algoritma monoalfabetik cipher [5]. Namun terdapat metode Kasiski, yang merupakan metode untuk menerka panjang kunci dengan cara melihat bagian cipherteks berulang yang terpisah sejauh jarak tertentu di dalam cipherteks [6]. Bagian cipherteks yang berulang ini diduga merupakan hasil enkripsi bagian plainteks dengan kunci yang sama pada posisinya. Penentuan panjang kunci dilakukan dengan menemukan pasangan – pasangan bagian cipherteks yang berulang dan jarak berulangnya. Dari nilai – nilai jarak tersebut dihitung Faktor Persekutuan Terbesarnya yang akan menjadi kemungkinan panjang kunci. Pada penelitian yang dilakukan oleh Muslim Ramli, digunakan kunci dengan panjang bilangan prima yang membuat metode Kasiski menjadi lebih sulit dilakukan [6]**.** Namun masih terdapat kemungkinan adanya cara lain untuk memodifikasi Vigenere Cipher menjadi lebih aman dari serangan dengan metode Kasiski.

Oleh karena adanya kelemahan monoalfabetik cipher dan Vigenere Cipher ini, penulis ingin meneliti bagaimana jika dalam proses enkripsi, pesan dipartisi terlebih dahulu ke dalam beberapa bagian pesan yang ukurannya tidak teratur secara matematis. Cara mempartisi pesan menggunakan konsep yang ada di *finite automata*. Dimana seperti yang telah diketahui bahwa finite automata merupakan model komputasi sederhana yang salah satu manfaatnya untuk merepresentasikan mesin yang mempu menguji apakah suatu string valid atau tidak berdasarkan definisi bahasa yang diberikan. *Finite automata* dapat disajikan berupa kondisi/state dalam notasi lingkaran yang terhubung melalui transisi berarah. Dimana setiap transisi membawa suatu simbol inputan untuk diproses pada kondisi selanjutnya [7]. Dalam penelitian ini, penulis menggunakan kriteria partisi pesan asli ke dalam bagian – bagian pesan yang diakhiri oleh huruf E, dengan alasan huruf E adalah huruf yang paling sering muncul dalam teks bahasa Inggris. Nantinya setiap bagian pesan asli akan dienkripsi oleh satu huruf yang ada di kata kunci layaknya Vigenere Cipher. Dengan demikian ukuran tingkat keamanan menjadi lebih tinggi dari monoalfabetik cipher karena tidak dapat dideteksi dengan analisis frekuensi kemunculan huruf, dan juga lebih aman dari Vigenere Cipher karena ukuran kata kunci lebih sulit untuk diterka dan panjangnya bervariasi.

Metode

Metode yang digunakan dalam penelitian ini adalah pemodelan komputasi sederhana dan simulasi menggunakan program bahasa C++. Perancangan algoritma kriptografi dilengkapi dengan model komputasi sederhana yaitu *deterministic finite automata*.

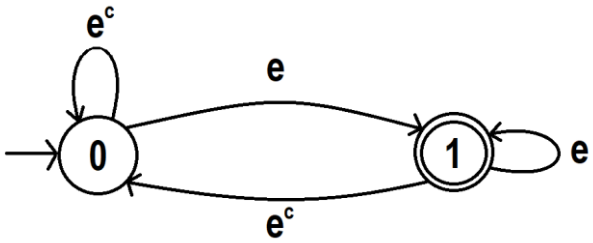
Alur penelitian dapat dilihat pada Gambar 1.



Gambar 1. Diagram Alur Penelitian

Hasil dan Pembahasan

Berikut merupakan hasil dan pembahasan dalam penelitian ini. Pertama akan diperlihatkan hasil rancangan model komputasi sederhana berupa diagram transisi *deterministic finite automata* yang dapat dilihat pada Gambar 2.



Gambar 2 Diagram Transisi *Deterministic Finite Automata*

Pada Gambar 2, diagram memiliki 2 kondisi/state, yaitu 0 dan 1. State 0 merupakan state inisial dan state 1 merupakan final state. Simbol e dan e^c merupakan simbol yang diproses atau simbol yang ditransisikan. Dimana e menyatakan huruf E/e pada plainteks. Huruf ini dipilih karena frekuensi kemunculannya yang paling tinggi dalam teks bahasa Inggris. Sementara e^c menyatakan huruf – huruf lain dalam alfabet selain huruf E/e.

Berdasarkan diagram tersebut dapat dijelaskan bahwa setiap bagian pesan (*sub-plaintext*) dilakukan pengecekan string dari awal huruf. Pembagian setiap blok pesan dilakukan mulai dari awal karakter hingga pada karakter E/e. Proses ini diulang hingga seluruh karakter pada pesan terbagi kedalam blok-blok Setiap berhenti di state 1 (final state), maka pesan akan dipotong pada bagian tersebut. Dengan kata lain setiap bagian pesan selalu diakhiri dengan huruf E, kecuali bagian pesan paling akhir dari plainteks keseluruhan, memungkinkan tidak diakhiri huruf E.

Contoh:
Plainteks : REPRESENTATIVES
Partisi : RE PRE SE NTATIVE S

Pada contoh di atas terdapat 5 blok pesan, merupakan pesan bagian (*sub-plaintext*) yang diakhiri oleh huruf E/e (kecuali blok yang terakhir).

Published by: Lembaga Penelitian, Pengabdian Masyarakat, dan Penjaminan Mutu
Institut Teknologi Sumatera, Lampung Selatan, Indonesia

Setelah mendapatkan hasil perancangan diagram finite automata, selanjutnya dilakukan perancangan kode program dalam bahasa C++.

Berikut ini adalah alur bekerja program yang digunakan untuk implementasi algoritma yang telah dirancang.

1. Langkah awal, program membuka file eksternal "plainteks.txt". Isi file merupakan plainteks yang telah diproses sebelumnya.
2. Program mendeklarasikan array 2 dimensi yang menyatakan fungsi transisi dalam model *deterministic finite automata* yang telah dirancang.
3. Selanjutnya program meminta pengguna memasukan kata kunci yang akan digunakan untuk enkripsi. Kata kunci yang digunakan berupa satu kata dari rangkaian huruf alfabet kapital.
4. Program melakukan proses enkripsi sesuai algoritma yang telah dirancang. Peninjauan dilakukan huruf demi huruf. Mulai dari huruf pertama di plainteks akan menjadi inputan pada kondisi/state di *deterministic finite automata*. Lalu enkripsi huruf tersebut dengan huruf pertama pada kata kunci. Jika kondisi tujuan adalah final state maka lakukan pergeseran indeks kata lunci ke kanan yang akan dipakai untuk enkripsi huruf plainteks selanjutnya. Indeks huruf kata kunci akan berulang ke paling kiri setelah mencapai indeks paling kanan. Proses ini dilakukan seterusnya hingga seluruh huruf pada plainteks diproses.
5. Hasil enkripsi berupa cipherteks dituliskan dalam file eksternal "cipherteks.txt."

Contoh :

Plainteks : WEARETHEWORLD

Kata kunci : CRYPTO

Cipherteks : YGRIVRFCLDGAS

Perhatikan bahwa huruf – huruf pada plainteks dienkripsi dengan huruf pada kata kunci seperti pada Tabel 1.

Tabel 1. Contoh enkripsi pesan singkat

Plainteks	W	E	A	R	E	T	H	E	W	O	R	L	D
Kunci	C	C	R	R	R	Y	Y	Y	P	P	P	P	P
Cipherteks	Y	G	R	I	V	R	F	C	L	D	G	A	S

Setiap pesan bagian (*sub-plaintext*) yang diakhiri oleh huruf E/e, akan dienkripsi dengan huruf kunci yang berbeda sesuai kata kunci yang digunakan. Dengan cara di atas, setiap huruf tidak selalu akan dienkripsi menjadi huruf cipherteks yang sama, dan panjang kata kunci akan sulit dideteksi.

Pada tahap evaluasi, digunakan plainteks berukuran 99999 huruf. Plainteks diperoleh dari teks cerita fiksi yang ada di website http://textfiles.com/stories/ dengan judul "The Heart of Darkness" by Joseph Conradfile. Kunci yang digunakan dalam enkripsi adalah kata INSTITUTTEKNOLOGISUMATERA yang memiliki panjang 25 karakter.

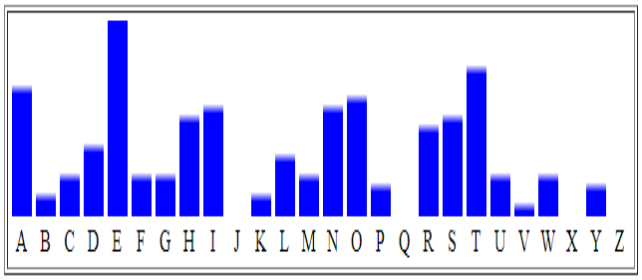
Evaluasi dilakukan dengan dua pengujian berdasarkan serangan yang mungkin dilakukan, yaitu analisis frekuensi dan metode Kasiski.

Analisis frekuensi menggunakan tools yang dapat diakses pada website https://www.mtholyoke.edu/. Teks yang sudah diproses sedemikian sehingga hanya berupa huruf kapital saja, diinputkan pada tools ini, kemudian dilakukan perhitungan frekuensi kemunculan tiap huruf. Hasilnya diperoleh seperti pada Tabel 2.

Tabel 2. Daftar Frekuensi Kemunculan Huruf Plainteks

E	12592	12,59%		**M**	2510	2,51%
T	9514	9,51%		**F**	2381	2,38%
A	8203	8,20%		**G**	2251	2,25%
O	7624	7,62%		**C**	2222	2,22%
I	6885	6,89%		**Y**	1908	1,91%
N	6848	6,85%		**P**	1672	1,67%
S	6232	6,23%		**B**	1499	1,50%
H	6127	6,13%		**K**	1016	1,02%
R	5421	5,42%		**V**	928	0,93%
D	4300	4,30%		**X**	155	0,16%
L	4064	4,06%		**Z**	107	0,11%
U	2790	2,79%		**J**	98	0,10%
W	2569	2,57%		**Q**	83	0,08%

Pada tabel 2 terlihat presentase kemunculan tiap huruf pada plainteks. Urutan kemunculan dari yang tertinggi yaitu huruf E, dilanjutkan huruf T, A, O, I dan seterusnya. Secara statistik dapat dilihat pada Gambar 3.



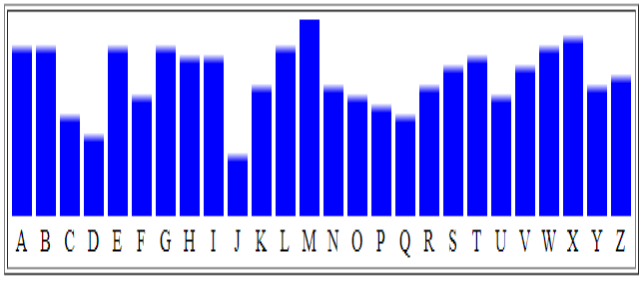
Gambar 3 Statistik frekuensi huruf pada plainteks

Selanjutnya plainteks tersebut dienkripsi sehingga didapatkan cipherteks menggunakan program yang telah dirancang. Pada hasil cipherteks dilakukan analisa frekuensi kemunculan huruf dengan tools serupa, sehingga didapatkan hasil sebagai berikut.

Pada Tabel 3 terlihat presentase kemunculan tiap huruf pada cipherteks. Urutan kemunculan dari yang tertinggi yaitu huruf M, dilanjutkan huruf A, X, G, B dan seterusnya. Secara statistik dapat dilihat pada Gambar 4.

Tabel 3. Daftar Frekuensi Kemunculan Huruf Cipherteks

M	5674	5,67%		**R**	3670	3,67%
A	4788	4,79%		**Z**	3652	3,65%
X	4783	4,78%		**O**	3599	3,60%
G	4764	4,76%		**N**	3555	3,56%
B	4749	4,75%		**F**	3549	3,55%
W	4647	4,65%		**K**	3532	3,53%
H	4447	4,45%		**Y**	3461	3,46%
T	4347	4,35%		**U**	3206	3,21%
E	4318	4,32%		**C**	2933	2,93%
L	4300	4,30%		**P**	2877	2,88%
S	4273	4,27%		**Q**	2638	2,64%
I	4236	4,24%		**D**	2047	2,05%
V	4195	4,20%		**J**	1759	1,76%



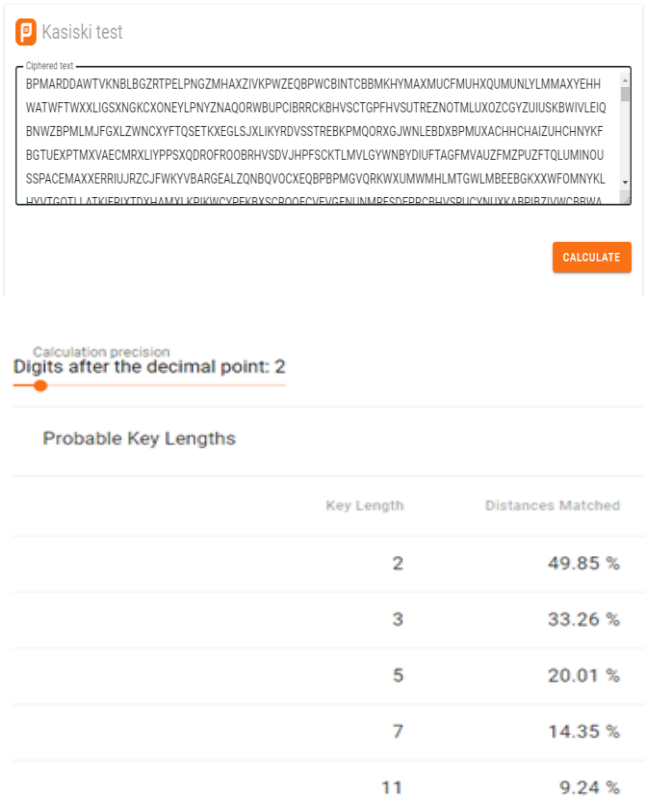
Gambar 4 Statistik frekuensi huruf pada cipherteks

Perhatikan bahwa sebaran presentase kemunculan huruf pada plainteks dan cipherteks memiliki proporsi yang berbeda. Pada plainteks kemunculan huruf lebih cenderung didominasi oleh beberapa huruf tertentu. Namun pada cipherteks, kemunculan huruf hampir merata di beberapa huruf. Sehingga analisis frekuensi tidak dapat digunakan dengan tepat untuk menerka aturan enkripsi.

Published by: Lembaga Penelitian, Pengabdian Masyarakat, dan Penjaminan Mutu Institut Teknologi Sumatera, Lampung Selatan, Indonesia

Selanjutnya dilakukan pengujian dengan metode Kasiski pada cipherteks menggunakan tools yang ada secara online di website https://planetcalc.com.

Cipherteks diinputkan kemudian dianalisa dengan tools ini, sehingga hasilnya diperoleh sebagai berikut (Gambar 5).



Gambar 5 Hasil analisa panjang kunci dengan metode Kasiski

Berdasarkan gambar di atas, hasil analisa panjang kunci dengan metode Kasiski diperoleh bahwa kemungkinan panjang kunci yang diterka yaitu sepanjang 2 karakter, dengan presentase 49,85%. Dilanjutkan dengan 3, 5, 7, dan 11 karakter. Padahal panjang kunci yang digunakan adalah 25 karakter. Sehingga metode Kasiski sulit digunakan untuk menerka panjang kunci enkripsi dengan algoritma yang digunakan dalam penelitian ini.

Kesimpulan

Berdasarkan penelitian dan pengujian yang telah dilakukan, dapat disimpulkan bahwa :

1. Algoritma kriptografi Caesar cipher dan secara umum algoritma kriptografi monoalfabetik cipher tidak optimal digunakan untuk mengamankan pesan teks dikarenakan untuk Caesar cipher kemungkinan kunci yang sedikit dan secara umum untuk monoalfabetik cipher dapat dengan mudah dipecahkan melalui analisis frekuensi kemunculan huruf.
2. Algoritma kriptografi Vigenere cipher juga tidak optimal digunakan untuk mengamankan pesan teks dikarenakan panjang kunci dapat diterka melalui metode Kasiski.
3. Model komputasi *deterministic finite automata* dapat digunakan untuk mempartisi plainteks dengan kriteria string yang diakhiri oleh huruf E sebagai huruf yang sering muncul dalam teks bahasa Inggris, sehingga setiap string bagian dari plainteks akan dienkripsi dengan huruf pada kata kunci dengan pola yang tidak teratur secara matematis.
4. Hasil pengujian analisis frekuensi pada enkripsi kriptografi klasik dengan model komputasi ini yaitu tidak terlihat hubungan antara sebaran kemunculan huruf di plainteks dengan kemunculan huruf di cipherteks, sehingga kriptanalis tidak dapat menerka aturan enkripsi yang digunakan. Begitu juga metode Kasiski tidak dapat digunakan untuk menerka panjang kunci, dikarenakan penggunaan huruf kunci tidak teratur secara matematis pada setiap huruf.

Conflicts of interest

Penelitian ini bersifat teoritis yang merupakan ide pengembangan dari penelitian – penelitian yang sudah ada sebelumnya. Sehingga tidak diangkat ataupun dilibatkan dengan konflik atau permasalahan nyata yang ada di sekitar.

Penutup

Algoritma kriptografi klasik memang sudah jarang digunakan karena pada masa ini penggunaan informasi berbentuk digital sehingga representasi pesan asli dinyatakan dengan kode bit biner. Hal ini merupakan

yang menjadi ciri algoritma kriptografi modern. Namun untuk ide algoritma dan perhitungannya diadopsi dari algoritma kriptografi klasik. Sehingga dari penelitian ini yang merupakan modifikasi algoritma kriptografi klasik, nantinya dapat diterapkan pada algoritma kriptografi modern.

Daftar Referensi

[1] R. Munir, *Kriptografi*, 2nd ed. Bandung: Informatika Bandung, 2019.

[2] M. M. Amin, "Implementasi Kriptografi Klasik Pada Komunikasi Berbasis Teks," *Pseudocode*, vol. 3, no. 2, pp. 129–136, 2017, doi: 10.33369/pseudocode.3.2.129-136.

[3] T. Limbong, "Pengujian Kriptografi Klasik Caesar Chipper Menggunakan Matlab," *Semin. Nas. Inov. dan Teknol. Inf. Sept.*, no. September 2015, pp. 77–80, 2015.

[4] Y. Permanasari, "Kriptografi Klasik Monoalphabetic," *Matematika*, vol. 16, no. 1, pp. 7–10, 2017, doi: 10.29313/jmtm.v16i1.2543.

[5] A. Anggie, "P Rogram S Tudi D Oktor," *Pemodelan Arsit. Sist. Inf. Perizinan Menggunakan Kerangka Kerja Togaf Adm*, vol. 4, no. 1, p. (halaman 2), 2018.

[6] M. Zarlis, "Implementasi Algoritma Vigenere Subtitusi dengan Shift Indeks Prima," *Univ. Sumatera Utara*, pp. 149–154, 2017.

[7] A. A. Sharipbay, Z. S. Saukhanova, G. B. Shakhmetova, and N. S. Saukhanov, "Application of finite automata in cryptography," *ACM Int. Conf. Proceeding Ser.*, pp. 3–5, 2019, doi: 10.1145/3330431.3330452.

Published by: Lembaga Penelitian, Pengabdian Masyarakat, dan Penjaminan Mutu Institut Teknologi Sumatera, Lampung Selatan, Indonesia