Epsilon: Jurnal Matematika Murni dan Terapan
http://ppjp.ulm.ac.id/journals/index.php/epsilon
p-ISSN: 1978-4422 | e-ISSN 2656-7660
Vol.18 No.2 (Desember 2024) Hal. 149-164

PENERAPAN ALGORITMA *HILL CIPHER* DAN PERMUTASI PADA SISTEM KRIPTOGRAFI *POLYALPHABETIC*

Sindi Lukmaini, Yanuar Bhakti Wira Tama, Kartika Nugraheni

Program Studi Matematika, Jurusan Matematika dan Teknologi Informasi, Institut Teknologi Kalimantan, Jl. Soekarno Hatta KM. 15, Balikpapan 76127, Kalimantan Timur
Email: 02201020@student.itk.ac.id

ABSTRACT

Technological advances have a great influence on facilitating long-distance communication, one of which is exchanging messages. The security of a message is an important aspect of protecting sensitive information. Cryptography is a branch of science that aims to maintain the confidentiality, integrity, and authenticity of messages in communication. One of the cryptographic methods used is Hill Cipher which converts plaintext into ciphertext using a key matrix, but Hill Cipher is vulnerable to cryptanalysis attacks, especially if the key matrix is not chosen properly. Based on these problems, this research was conducted by applying the Hill Cipher algorithm with permutation as an additional layer of security. This study aims to analyze the effect of the message and key matrix on the performance of computation time in the encryption and decryption process of Hill Cipher combined with permutation, where the message characters are limited to 26 alphabetic characters and the use of Hill block length and permutation techniques set at 5 characters. The simulation results show the average performance of computation time on the encryption and decryption process is 0.000083199 seconds and 0.000113493 seconds for a message with 707 characters. The key matrix used must be invertible, and the performance of computation time on encryption and decryption with different keys is 0.000094817 seconds and 0.00025938 seconds.

Keywords: Computation, Cryptography, Hill Cipher, Permutation, Polyalphabetic Cipher

ABSTRAK

Kemajuan teknologi memiliki pengaruh yang besar untuk memudahkan berkomunikasi jarak jauh, salah satunya bertukar pesan. Keamanan suatu pesan menjadi aspek penting dalam melindungi informasi yang sensitif. Kriptografi, sebagai cabang ilmu yang bertujuan menjaga kerahasiaan, keutuhan, dan keotentikan pesan dalam komunikasi. Salah satu metode kriptografi yang digunakan adalah *Hill Cipher* yang mengubah *plaintext* menjadi *ciphertext* dengan menggunakan matriks kunci, namun *Hill Cipher* memiliki kerentanan terhadap serangan kriptanalisis, terutama jika matriks kunci tidak dipilih dengan tepat. Berdasarkan permasalahan tersebut dilakukan penelitian ini yaitu dengan menerapkan algoritma *Hill Cipher* dengan permutasi sebagai tambahan lapisan keamanan. Penelitian ini bertujuan untuk menganalisis pengaruh pesan dan matriks kunci terhadap performa waktu komputasi pada proses enkripsi dan dekripsi *Hill Cipher* yang digabungkan dengan permutasi, di mana karakter pesan terbatas pada 26 karakter alfabet dan penggunaan panjang blok *Hill* dan teknik permutasi yang ditetapkan pada 5 karakter. Hasil simulasi menunjukkan rata-rata performa waktu komputasi pada proses enkripsi dan dekripsi sebesar 0.000083199 detik dan 0.000113493 detik untuk pesan dengan 707 karakter. Matriks kunci yang digunakan harus *invertible*, dan performa waktu komputasi pada enkripsi dan dekripsi dengan kunci yang berbeda adalah 0.000094817 detik dan 0.00025938 detik.

Kata kunci: *Hill Cipher*, Komputasi, Kriptografi, Permutasi, *Polyalphabetic Cipher*

Received: 28 Juni 2024, Accepted: 16 Oktober 2024, Published: 5 November 2024

PENDAHULUAN

Perlindungan data dan informasi menjadi sangat penting dalam era digital yang semakin maju. Kemajuan teknologi memiliki pengaruh yang sangat besar terhadap aspek kehidupan manusia, salah satunya adalah internet. Dengan penggunaan internet dalam kehidupan manusia dapat memudahkan pekerjaan termasuk berkomunikasi jarak jauh. Namun jika menggunakan internet dengan tingkat keamanan relatif rendah maka pesan yang dikirim melalui jaringan mudah diketahui oleh pihak-pihak yang tidak berkepentingan. Bagi pihak pemerintahan, militer, perbankan, pendidikan dan lain-lain menggunakan internet sebagai alat untuk mengirimkan pesan rahasia, maka tingkat keamanan informasi menjadi faktor utama yang harus terpenuhi (Ramadani, 2020).

Terdapat beberapa metode yang digunakan untuk mengamankan pesan dari zaman dahulu, seperti menyembunyikan pesan ke dalam media lain agar orang lain terkecoh dengan tampilannya, ilmu ini disebut *steganography*. Selain pesan yang disembunyikan ke dalam media, ada juga ilmu pengamanan dengan menyandikan atau mengubah makna pesan tidak terbaca dengan menggunakan berbagai perhitungan, ilmu itu disebut *cryptography*. Kriptografi merupakan salah satu cabang ilmu yang bertujuan untuk menjaga kerahasiaan, keutuhan, dan keotentikan informasi dalam komunikasi yang melalui proses enkripsi dan dekripsi. Enkripsi merupakan metode untuk mengubah data asli menjadi bentuk yang tidak dapat dimengerti (sanditext) menggunakan algoritma kriptografi tertentu, sementara dekripsi adalah proses mengembalikan sanditext ke bentuk aslinya. Salah satu metode kriptografi klasik yang telah digunakan sejak lama adalah *Hill Cipher*. *Hill Cipher* adalah metode enkripsi polyalphabetic yang menggunakan matriks kunci untuk mengubah *plaintext* menjadi teks sandi. Keunggulan *Hill Cipher* terletak pada kemampuannya untuk mengatasi kelemahan metode substitusi klasik seperti *Caesar Cipher*, dengan memproses teks secara blok bukan karakter tunggal. Namun, *Hill Cipher* dapat menjadi rentan terhadap berbagai serangan kriptanalisis, terutama saat matriks kunci tidak dipilih secara tepat. Matriks kunci yang tepat ialah matriks kunci yang memiliki invers yang berarti, aljabar linear memainkan peran utama dalam enkripsi dan dekripsi suatu pesan.

Beberapa penelitian terkait pengamanan data dan suatu pesan informasi menggunakan *Hill Cipher* telah dilakukan oleh peneliti sebelumnya yaitu, modifikasi *Hill Chiper* dengan merepresentasikan karakter-karakter menjadi string biner (Paragas *et al*., 2019), *Hill Chiper* dikombinasikan dengan algoritma RSA (Santoso, 2021), *Hill Chiper* juga dapat dimodifikasi dengan cara kuncinya merupakan suatu matriks yang mempunyai invers/balikan matriks itu

sendiri(Lakhera *et al*., 2016), serta pengaplikasian *Hill Chiper* pada bidang kesehatan (Mohan *et al*., 2016). Begitu juga dengan Permutasi juga sudah diaplikasikan ke beberapa masalah pengalaman dari kriptografi klasik(John *et al*., 2023) sampai modern seperti pengamanan pada citra (Jolfaei *et al*., 2016), dan permasalahan Enigma (Courtois & Grajek, 2022). Dari beberapa penelitian tersebut diharapkan hasil yang sama yaitu dapat lebih mengamankan suatu pesan sehingga lebih sulit untuk dapat di pecahkan oleh pihak-pihak yang tidak diinginkan. Pada penelitian ini digunakan sistem kriptografi *polyalphabetic* dimana proses penyandian menggunakan metode *Hill Cipher* yang selanjutnya digabungkan dengan permutasi sebagai tambahan lapisan keamanan. Seperti yang diketahui permutasi melibatkan pengubahan urutan karakter atau blok data dalam plainteks sebelum proses enkripsi, sehingga dapat mempersulit upaya pihak yang tidak berwenang dalam menganalisis pola data dan menguraikan pesan terenkripsi. Penelitian ini berfokus pada penerapan aljabar linier dengan konsep operasi matriks dan modulo pada Ring Z_n menjadi pondasi dalam pembentukan matriks kunci utama yaitu kunci *Hill Cipher*. Selain itu, penelitian ini juga berfokus pada penggabungan *Hill Cipher* dengan permutasi, yang diharapkan dapat meningkatkan tingkat keamanan dan ketahanan sistem *polyalphabetic* terhadap serangan kriptanalisis.

TINJAUAN PUSTAKA

1. Kriptografi dan *Polyalphabetic*

Kriptografi merupakan salah satu bidang yang berkaitan dengan matematika atau lebih tepatnya aljabar dan teori bilangan yang tujuannya digunakan untuk menyembunyikan atau melindungi suatu informasi (Tama & Fahmi, 2023). Suatu kriptosistem terdiri dari lima-tupel $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ dengan $\mathcal{P}$ merupakan himpunan *plaintext* atau pesan awal, $\mathcal{C}$ merupakan himpunan *chipertext* atau pesan tersandi, $\mathcal{K}$ merupakan kunci yang digunakan merahasiakan pesan, $\mathcal{E}$ merupakan himpunan semua kemungkinan fungsi enkripsi yang memetakan *plaintext* ke *chipertext*, sedangkan $\mathcal{D}$ merupakan semua kemungkinan fungsi deskripsi yang memetakan balik *chipertext* ke *plaintext* (Stinson & Paterson, 2018). Misalkan untuk suatu kunci $K \in \mathcal{K}$ akan terdapat suatu fungsi enkripsi $e_K: \mathcal{P} \to \mathcal{C}$ dan suatu fungsi deskripsi $d_K: \mathcal{C} \to \mathcal{P}$, sehingga untuk setiap $x \in \mathcal{P}$ akan berlaku $d_K\big(e_K(x)\big) = x$. *Polyalphabetic Cipher* digunakan sebagai sistem kriptografi untuk mengenkripsi sekelompok karakter atau *string* dengan melibatkan penggunaan kunci (Hewage *et al*., 2022). Sesuai dengan namanya, karakter yang digunakan dalam sistem berupa huruf-huruf alfabet dengan penggunakaan karakter huruf kapital dan huruf kecil merupakan karakter yang berbeda.

2. Sandi *Hill*

Sandi *Hill* merupakan salah satu kriptosistem *polyalphabetic* yang diperkenalkan oleh Lester Hill pada tahun 1929 yang dapat dikategorikan sebagai block chipper (Forouzan, 2020). Untuk suatu *plaintext* dan *chipertext* yang didefinisikan pada gelanggang $(\mathbb{Z}_{26})^m$ dengan m merupakan bilangan bulat positif. Sandi *Hill* merupakan salah satu metode yang lebih kompleks daripada metode substitusi sederhana seperti *Caesar Cipher* (Gunawan, 2018), karena teks yang akan diproses akan dibagi menjadi blok-blok dengan ukuran m. Setiap karakter dalam satu blok akan saling mempengaruhi karakter lainnya dalam proses enkripsi dan dekripsinya, sehingga karakter yang sama tidak dipetakan menjadi karakter yang sama pula. Dalam sistem kriptografi sandi hill memanfaatkan perkalian matriks untuk enkripsi dan dekripsi suatu *chipertext* (Vishwa Nageshwar & Ravi Shankar, 2021).

Misalkan ingin dienkripsi suatu pesan dengan panjang m, dinyatakan dalam sebuah vektor $\boldsymbol{y} = (y_1, y_2, \dots, y_m)$, akan diperoleh hasil enkripsi menjadi sebuah vektor $\boldsymbol{x} = (x_1, x_2, \dots, x_m)$ dengan suatu matriks yang mempunyai invers yang disebut sebagai matriks kunci $\boldsymbol{K} = \begin{pmatrix} k_{1,1} & k_{1,2} & \cdots & k_{1,m} \\ k_{2,1} & k_{2,2} & \cdots & k_{2,m} \\ \vdots & \vdots & \vdots & \vdots \\ k_{m,1} & k_{m,2} & \cdots & k_{m,m} \end{pmatrix}$ dengan persamaan berikut:

$$\boldsymbol{y} = \boldsymbol{x} \cdot \boldsymbol{K}$$

Sedangkan untuk dekripsi pesannya menggunakan balikan dari matriks ($\boldsymbol{K^{-1}}$), yaitu

$$\boldsymbol{x} = \boldsymbol{y} \cdot \boldsymbol{K^{-1}}$$

Entri dari matriks kunci **K** dalam enkripsi dan dekripsi atas gelanggang $\mathbb{Z}_{26}$. Dalam menentukan invers matriks kunci ($\boldsymbol{K^{-1}}$) dapat menggunakan algoritma modifikasi gauss jordan, dengan mengubah perkalian entri matriks dengan invers entri pada gelanggang $\mathbb{Z}_{26}$. Dalam pemilihan matriks kunci $\boldsymbol{K}$ setidaknya determinan matriks tidak bernilai 0 di gelanggang $\mathbb{Z}_{26}$ atau tidak bernilai $0\ mod\ 26$, selain itu setiap baris dan kolomnya terdiri dari unit gelanggang $\mathbb{Z}_{26}$.

3. Permutasi

Permutasi adalah konsep matematika dasar yang digunakan dalam sistem kriptografi permutasi. Permutasi merupakan salah satu komponen fundamental kriptografi klasik, melibatkan penataan ulang karakter dalam pesan untuk mencapai kerahasiaan (John *et al*., 2023). Permutasi melibatkan pengubahan posisi elemen-elemen dalam suatu rangkaian atau blok. Dalam kriptografi, permutasi digunakan untuk mengacak atau menggantikan posisi huruf-huruf dalam pesan guna meningkatkan keamanan dan kerahasiaan pesan. Sandi Permutasi juga dikenal sebagai Sandi Transposisi didefinisikan sebagai berikut.

Misalkan m merupakan bilangan bulat positif. Misalkan $\mathcal{P} = \mathcal{C} = (\mathbb{Z}_{26})^{\mathrm{m}}$ dan $\mathcal{K}$ terdiri dari semua permutasi dari $\{1, 2, \dots, \mathrm{m}\}$. Untuk kunci π, didefinisikan

$$e_\pi(x_1, x_2, \dots, x_m) = \left(x_{\pi(1)}, x_{\pi(2)}, \dots, x_{\pi(m)}\right)$$

$$d_\pi(y_1, y_2, \dots, y_m) = \left(y_{\pi^{-1}(1)}, y_{\pi^{-1}(2)}, \dots, y_{\pi^{-1}(m)}\right)$$

Dimana π^{-1} adalah invers permutasi untuk π

4. Operasi Matriks

Operasi matriks yang akan dijelaskan pada tinjauan pusataka ini adalah modifikasi Gauss Jordan atas gelanggang $\mathbb{Z}_{\mathrm{n}}$ dan determinan matriks. Algoritma dalam menentukan invers matriks atas gelanggang $\mathbb{Z}_{\mathrm{n}}$ dapat dilihat di Gambar 1.

Algorithm 1 Modifikasi Algoritma Gauss Jordan untuk menentukan invers matriks atas gelanggang $\mathbb{Z}_n$

```
for i = 1, 2, ..., r do
    if A_{i,i} = 0 then
        for j = i + 1, ..., r do
            if A_{j,i} not unit in Z/nZ then
                Swap A_{j,·} and A_{i,·}
                Break
            end if
        end for
    end if
    invers ← invers A_{i,i} in Z/nZ
    for k = 1, 2, ..., 2r do
        A_{i,k} ← invers A_{i,k} in Z/nZ
    end for
    for l = i + 1, ..., r do
        p ← A_{l,i}*invers
        for m = i + 1, ..., 2r do
            A_{l,m} ← (A_{l,m} − p * A_{i,m}) mod n
        end for
        A_{l,i} ← 0
    end for
end for
for i = r, r − 1, ..., 1 do
    for j = 1, ..., i − 1 do
        p ← A_{j,i}
        for k = 1, 2, ..., 2r do
            A_{j,k} ← (A_{j,k} − p * A_{i,k}) mod n
        end for
    end for
end for
Ainv ← A_{{r+1,r+2,...,2r},{r+1,r+2,...,2r}}
```

Gambar 1. Modifikasi Algoritma Gauss Jordan untuk menentukan invers matriks atas gelanggang $\mathbb{Z}_{26}$

Perbedaan yang cukup terlihat dari algoritma Gauss Jordan atas bilangan real adalah untuk memperoleh nilai 1 pada diagonal utama, dikalikan dengan invers dari entri yang merupakan unit. Sedangkan untuk determinan matriks dapat digunakan metode kofaktor dengan terlebih dahulu mendefinisikan matriks minor dari suatu matriks $\boldsymbol{A}$ berukuran $n \times n$. Matriks minor $\boldsymbol{A}$ baris i kolom j yang dinotasikan sebagai $\boldsymbol{A}_{ij}$ dapat diperoleh dengan cara menghapus baris i kolom j sehingga ukuran matriks $\boldsymbol{A}_{ij}$ adalah $(n-1) \times (n-1)$. Dengan demikian, dapat didefinisikan matriks kofaktor $\boldsymbol{C}_{ij}$ sebagai berikut:

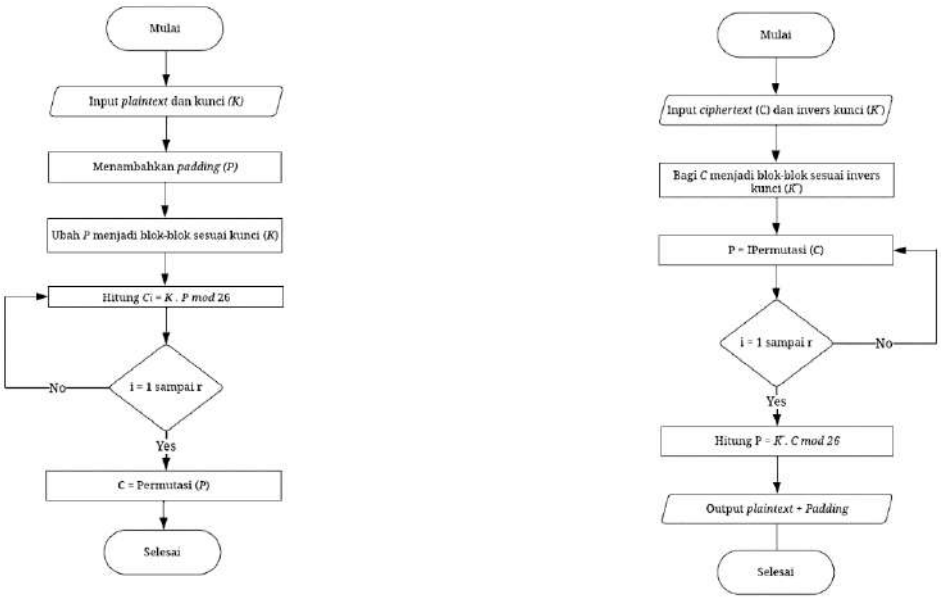
$$\boldsymbol{C}_{ij} = (-1)^{i+j}\boldsymbol{A}_{ij}$$

Determinan matriks $\boldsymbol{A}$ dapat didefinisikan dengan kofaktor matriks sebagai berikut:

$$\det(\boldsymbol{A}) = \sum_{i=1}^{n} a_{ij}\boldsymbol{C}_{ij}$$

METODE PENELITIAN

Metode yang dilakukan dalam pada penelitian ini yaitu dengan melakukan penerapan algoritma *Hill Cipher* dan permutasi dalam proses enkripsi dan dekripsi pada sistem kriptografi *polyalphabetic Cipher* yang dapat dilihat pada Gambar 1



(a) **(b)**

Gambar 2. Diagram Alir Proses Enkripsi (a) dan Dekripsi (b)

Adapun tahapan dari Gambar 2 yaitu:

- Input *plaintext* dan kunci enkripsi *Hill Cipher* dengan ukuran $m \times m$, di mana m sesuai dengan jumlah karakter dalam setiap blok *plaintext* dan memastikan matriks kunci memiliki invers agar proses dekripsi dapat dilakukan.
- Selanjutnya proses enkripsi, *plaintext* dibagi menjadi blok-blok huruf atau karakter dan memastikan setiap blok memiliki jumlah karakter yang sesuai dengan ukuran matriks kunci *Hill Cipher* serta menambahkan *padding* pada blok akhir.

- Menerapkan *Hill Cipher* pada setiap blok karakter dengan melibatkan operasi matriks untuk mengalikan matriks kunci dengan vektor kolom dari karakter-karakter dalam blok.
- Menerapkan teknik permutasi pada setiap blok karakter. Permutasi akan mengubah urutan karakter dalam blok berdasarkan kunci permutasi yang telah ditentukan.
- Menggabungkan semua blok *ciphertext* menjadi satu pesan *ciphertext* yang lengkap.
- Selanjutnya proses dekripsi, dimana tahapannya merupakan kebalikan dari enkripsi, pada proses ini kunci yang digunakan adalah invers dari kunci tersebut, baik kunci pada *Hill Cipher* maupun permutasi, sehingga diperoleh *plaintext* dengan *padding*.

HASIL DAN PEMBAHASAN

Dilakukan pendefinisian karakter yang digunakan dalam proses enkripsi dan dekripsi. Kemudian dilakukan proses kombinasi algoritma kriptografi yang digunakan. Setelah itu dilakukan analisis untuk mengetahui tingkat keamanan dari modifikasi kriptografi.

1. Rancangan Algoritma

Rancangan algoritma pada penelitian ini menggunakan suatu pesan dimana karakter yang digunakan pada *plaintext* akan diubah menjadi karakter huruf kapital dengan mengabaikan nomor, tanda baca dan spasi terlebih dahulu, kemudian *plaintext* yang sudah berupa huruf kapital tanpa karakter lain selain huruf diubah menjadi angka-angka anggota $\mathbb{Z}_{26}$, sehingga sebanyak $n = 26$.

Proses enkripsi dan dekripsi dilakukan dengan menggunakan dua algoritma yaitu *Hill Cipher* dan *Permutation Cipher* menggunakan *padding.* Penambahan *padding* dilakukan baik saat jumlah karakter pada *plaintext* tidak sesuai maupun sudah sesuai dengan ukuran matriks kunci utama. Matriks kunci utama disini merujuk pada matriks kunci *Hill Cipher* dengan ukuran 5×5 karena ukuran yang tidak terlalu kecil untuk melihat proses algoritma dan tidak terlalu besar agar komputasi tidak terlalu lama. Rancangan proses enkripsi dan dekripsi ini dapat dilihat pada Gambar 2.

2. Enkripsi Menggunakan Algoritma *Hill Cipher* dan *Permutation Cipher*

Proses enkripsi dilakukan dengan rancangan yang telah dibuat, dimana pengirim melakukan proses enkripsi menggunakan algoritma *Hill Cipher* dan *Permutation Cipher* yang selanjutnya akan menghasilkan suatu *ciphertext* untuk penerima. Pada proses enkripsi baik pengirim maupun penerima sudah menyepakati kunci yang akan digunakan. Kunci yang digunakan pada proses enkripsi menggunakan algoritma *Hill Cipher* yaitu kunci simetris. Akan dicoba

terlebih dahulu suatu matriks kunci yang mempunyai invers atas gelanggang $\mathbb{Z}_{26}$ sebagai berikut,

$$K = \begin{bmatrix} 6 & 24 & 1 & 13 & 16 \\ 10 & 20 & 17 & 15 & 7 \\ 23 & 18 & 17 & 0 & 4 \\ 10 & 14 & 24 & 5 & 3 \\ 21 & 13 & 2 & 8 & 24 \end{bmatrix}$$

dimana perlu dilakukan pengecekan pada matriks kunci K bahwa matriks tersebut mempunyai $K^{-1}\, mod\, 26$ dengan menentukan $\det K \neq 0$ dengan menggunakan metode kofaktor.

$$|K| = \begin{vmatrix} 6 & 24 & 1 & 13 & 16 \\ 10 & 20 & 17 & 15 & 7 \\ 23 & 18 & 17 & 0 & 4 \\ 10 & 14 & 24 & 5 & 3 \\ 21 & 13 & 2 & 8 & 24 \end{vmatrix}$$

$$|K| = a_{31}C_{31} - a_{32}C_{32} + a_{33}C_{33} - a_{34}C_{34} + a_{35}C_{35}\, mod\, 26$$

$$= 23 \times \begin{vmatrix} 24 & 1 & 13 & 16 \\ 20 & 17 & 15 & 7 \\ 14 & 24 & 5 & 3 \\ 13 & 2 & 8 & 24 \end{vmatrix} - 18 \times \begin{vmatrix} 6 & 1 & 13 & 16 \\ 10 & 17 & 15 & 7 \\ 10 & 24 & 5 & 3 \\ 21 & 2 & 8 & 24 \end{vmatrix}$$

$$+ 17 \times \begin{vmatrix} 6 & 24 & 13 & 16 \\ 10 & 20 & 15 & 7 \\ 10 & 14 & 5 & 3 \\ 21 & 13 & 8 & 24 \end{vmatrix} - 0$$

$$\times \begin{vmatrix} 6 & 24 & 1 & 16 \\ 10 & 20 & 17 & 7 \\ 10 & 14 & 24 & 3 \\ 21 & 13 & 2 & 24 \end{vmatrix} + 4 \times \begin{vmatrix} 6 & 24 & 1 & 13 \\ 10 & 20 & 17 & 15 \\ 10 & 14 & 24 & 5 \\ 21 & 13 & 2 & 8 \end{vmatrix}\, mod\, 26$$

$$= 23 \times (-53827) - 18 \times (39185) + 17 \times (33150) - 0$$

$$+ 4 \times (57680)\, mod\, 26$$

$$= -1238021 - 705330 + 563550 + 230720 \bmod 26 = -1149081 \bmod 26$$

karena karakter terbatas pada modulo 26, maka

$$= -1149081\, mod\, 26 = 15 \bmod 26$$

diperoleh determinan $K = 15 \neq 0$, yang artinya terbukti bahwa K memiliki invers dan dapat digunakan sebagai kunci *Hill Cipher*. Selanjutnya untuk memastikan matriks kunci K tersebut mempunyai $K^{-1}\, mod\, 26$ dilakukan perhitungan dengan menggunakan modifikasi eliminasi Gauss Jordan seperti pada Gambar 1. dan dilanjutkan dengan memeriksa $K.K^{-1} = I$ sebagaimana yang dilakukan pada langkah berikut:

$$\left[\begin{array}{ccccc|ccccc} 6 & 24 & 1 & 13 & 16 & 1 & 0 & 0 & 0 & 0 \\ 10 & 20 & 17 & 15 & 7 & 0 & 1 & 0 & 0 & 0 \\ 23 & 18 & 17 & 0 & 4 & 0 & 0 & 1 & 0 & 0 \\ 10 & 14 & 24 & 5 & 3 & 0 & 0 & 0 & 1 & 0 \\ 21 & 13 & 2 & 8 & 24 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \xrightarrow{b_1 \rightleftharpoons b_3} \left[\begin{array}{ccccc|ccccc} 23 & 18 & 17 & 0 & 4 & 0 & 0 & 1 & 0 & 0 \\ 10 & 20 & 17 & 15 & 7 & 0 & 1 & 0 & 0 & 0 \\ 6 & 24 & 1 & 13 & 16 & 1 & 0 & 0 & 0 & 0 \\ 10 & 14 & 24 & 5 & 3 & 0 & 0 & 0 & 1 & 0 \\ 21 & 13 & 2 & 8 & 24 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

$$\xrightarrow{17b_1}\left[\begin{array}{ccccc|ccccc}1&20&3&0&16&0&0&17&0&0\\10&20&17&15&7&0&1&0&0&0\\6&24&1&13&16&1&0&0&0&0\\10&14&24&5&3&0&0&0&1&0\\21&13&2&8&24&0&0&0&0&1\end{array}\right]\xrightarrow[b_4-10b_1]{\substack{b_2-10b_1\\b_3-6b_1\\b_5-21b_1}}\left[\begin{array}{ccccc|ccccc}1&20&3&0&16&0&0&17&0&0\\0&2&13&15&3&0&1&12&0&0\\0&8&9&13&24&1&0&2&0&0\\0&22&20&5&25&0&0&12&1&0\\0&9&17&8&0&0&0&7&0&1\end{array}\right]$$

$$\xrightarrow{b_2\rightleftharpoons b_5}\left[\begin{array}{ccccc|ccccc}1&20&3&0&16&0&0&17&0&0\\0&9&17&8&0&0&0&7&0&1\\0&8&9&13&24&1&0&2&0&0\\0&22&20&5&25&0&0&12&1&0\\0&2&13&15&3&0&1&12&0&0\end{array}\right]\xrightarrow{3b_2}\left[\begin{array}{ccccc|ccccc}1&20&3&0&16&0&0&17&0&0\\0&1&25&24&0&0&0&21&0&3\\0&8&9&13&24&1&0&2&0&0\\0&22&20&5&25&0&0&12&1&0\\0&2&13&15&3&0&1&12&0&0\end{array}\right]$$

$$\xrightarrow[b_5-2b_2]{\substack{b_3-8b_2\\b_4-22b_2}}\left[\begin{array}{ccccc|ccccc}1&20&3&0&16&0&0&17&0&0\\0&1&25&24&0&0&0&21&0&3\\0&0&17&3&24&1&0&16&0&2\\0&0&16&23&25&0&0&18&1&12\\0&0&15&19&3&0&1&22&0&20\end{array}\right]\xrightarrow{23b_3}\left[\begin{array}{ccccc|ccccc}1&20&3&0&16&0&0&17&0&0\\0&1&25&24&0&0&0&21&0&3\\0&0&1&17&6&23&0&4&0&20\\0&0&16&23&25&0&0&18&1&12\\0&0&15&19&3&0&1&22&0&20\end{array}\right]$$

$$\xrightarrow[b_5-15b_3]{b_4-16b_3}\left[\begin{array}{ccccc|ccccc}1&20&3&0&16&0&0&17&0&0\\0&1&25&24&0&0&0&21&0&3\\0&0&1&17&6&23&0&4&0&20\\0&0&0&11&7&22&0&6&1&4\\0&0&0&24&17&19&1&14&0&6\end{array}\right]\xrightarrow{19b_4}\left[\begin{array}{ccccc|ccccc}1&20&3&0&16&0&0&17&0&0\\0&1&25&24&0&0&0&21&0&3\\0&0&1&17&6&23&0&4&0&20\\0&0&0&1&3&2&0&10&19&24\\0&0&0&24&17&19&1&14&0&6\end{array}\right]$$

$$\xrightarrow{b_5-24b_4}\left[\begin{array}{ccccc|ccccc}1&20&3&0&16&0&0&17&0&0\\0&1&25&24&0&0&0&21&0&3\\0&0&1&17&6&23&0&4&0&20\\0&0&0&1&3&2&0&10&19&24\\0&0&0&0&23&23&1&8&12&2\end{array}\right]\xrightarrow{17b_5}\left[\begin{array}{ccccc|ccccc}1&20&3&0&16&0&0&17&0&0\\0&1&25&24&0&0&0&21&0&3\\0&0&1&17&6&23&0&4&0&20\\0&0&0&1&3&2&0&10&19&24\\0&0&0&0&1&1&17&6&22&8\end{array}\right]$$

$$\xrightarrow[b_4-3b_5]{\substack{b_1-16b_5\\b_3-6b_5}}\left[\begin{array}{ccccc|ccccc}1&20&3&0&0&10&14&25&12&2\\0&1&25&24&0&0&0&21&0&3\\0&0&1&17&0&17&2&20&24&24\\0&0&0&1&0&25&1&18&5&0\\0&0&0&0&1&1&17&6&22&8\end{array}\right]\xrightarrow[b_3-17b_4]{b_2-24b_4}\left[\begin{array}{ccccc|ccccc}1&20&3&0&0&10&14&25&12&2\\0&1&25&0&0&24&2&5&10&3\\0&0&1&0&0&8&11&0&17&24\\0&0&0&1&0&25&1&18&5&0\\0&0&0&0&1&1&17&6&22&8\end{array}\right]$$

$$\xrightarrow[b_2-25b_3]{b_1-3b_3}\left[\begin{array}{ccccc|ccccc}1&20&0&0&0&12&7&25&13&8\\0&1&0&0&0&6&13&5&1&1\\0&0&1&0&0&8&11&0&17&24\\0&0&0&1&0&25&1&18&5&0\\0&0&0&0&1&1&17&6&22&8\end{array}\right]\xrightarrow{b_1-20b_2}\left[\begin{array}{ccccc|ccccc}1&0&0&0&0&22&7&3&19&14\\0&1&0&0&0&6&13&5&1&1\\0&0&1&0&0&8&11&0&17&24\\0&0&0&1&0&25&1&18&5&0\\0&0&0&0&1&1&17&6&22&8\end{array}\right]$$

Sehingga diperoleh invers matriks kunci K atau K^{-1} yaitu sebagai berikut:

$$K^{-1}=\begin{bmatrix}22&7&3&19&14\\6&13&5&1&1\\8&11&0&17&24\\25&1&18&5&0\\1&17&6&22&8\end{bmatrix}$$

Selanjutnya dilakukan pengecekan apakah $K.K^{-1}\,mod\,26 = I\,mod\,26$ sebagai berikut:

$$K.K^{-1}mod\,26=\begin{bmatrix}6&24&1&13&16\\10&20&17&15&7\\23&18&17&0&4\\10&14&24&5&3\\21&13&2&8&24\end{bmatrix}.\begin{bmatrix}22&7&3&19&14\\6&13&5&1&1\\8&11&0&17&24\\25&1&18&5&0\\1&17&6&22&8\end{bmatrix}mod\,26$$

$$= \begin{bmatrix} 625 & 650 & 468 & 572 & 260 \\ 858 & 651 & 442 & 728 & 624 \\ 754 & 650 & 183 & 832 & 780 \\ 624 & 572 & 208 & 703 & 754 \\ 780 & 754 & 416 & 1014 & 547 \end{bmatrix} mod\ 26 \equiv \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} mod\ 26$$

terbukti bahwa hasil perkalian matriks kunci K dan matriks kunci K^{-1} menghasilkan suatu matriks identitas. Dengan K^{-1} yang sudah sesuai ini dapat dilakukan langkah dekripsi selanjutnya pada algoritma *Hill Cipher*.

Dalam percobaan ini diberikan contoh enkripsi dengan pesan yaitu Mahasiswa ITK-2020 dengan kunci matriks K. Kemudian dilakukan proses konversi *plaintext*, pesan yang sebelumnya Mahasiswa ITK-2020 menjadi MAHASISWAITKDHL dimana spasi, tanda baca, dan angka diabaikan dan terjadi penambahan *padding* yaitu DHL. Pada proses enkripsi dengan algoritma *Hill Cipher* diperoleh *ciphertext* yaitu $DBZEWGMEAOAZKCX$. Setelah diperoleh *ciphertext* pada algoritma ini dilakukan proses enkripsi kedua menggunakan *Permutation Cipher* dengan matriks kunci permutasi (π) yang sudah ditentukan yaitu:

$$K_{\pi(x)} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Sehingga diperoleh *ciphertext* akhir yaitu $EWBDZAOMGECXZAK$.

3. Dekripsi Menggunakan Algoritma *Permutation Cipher* dan *Hill Cipher*

Setelah mendapatkan *ciphertext* dari pengirim, untuk mengetahui isi pesan dari pengirim, penerima melakukan proses dekripsi dengan rancangan yang telah dibuat menggunakan *Permutation Cipher* dan algoritma *Hill Cipher* yang selanjutnya akan menghasilkan suatu *plaintext*. Pada proses dekripsi, kunci yang digunakan pada proses dekripsi merupakan invers dari kunci yang telah disepakati antara pengirim dan penerima dan tahapan pada proses dekripsi merupakan kebalikan dari proses enkripsi. Sehingga proses dekripsi dimulai dari permutasi kembali terlebih dahulu dan dilanjutkan dengan *Hill Cipher*.

Kunci permutasi yang digunakan pada proses dekripsi adalah invers dari $K_{\pi(x)}$, maka diperoleh:

$$K_{\pi^{-1}(x)} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

dengan demikian $ciphertext$ yang semula $EWBDZAOMGECXZAK$ dibagi menjadi blok-blok yang terdiri dari lima huruf, yaitu sebagai berikut:

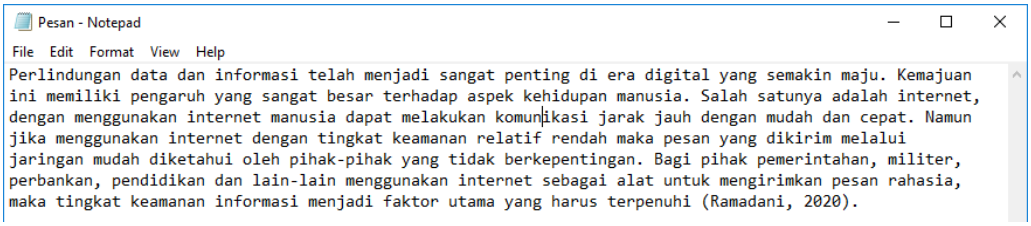
$$EWBDZ \mid AOMGE \mid CXZAK$$

blok-blok tersebut selanjutnya dituliskan dalam bentuk matriks dan dikalikan dengan $K_{\pi^{-1}(x)}$, sehingga *ciphertext* menjadi $DBZEWGMEAOAZKCX$. Selanjutnya dilakukan proses dekripsi menggunakan algoritma *Hill Cipher*.

Sebelumnya pada proses pembentukan kunci sudah didapatkan invers dari matrik kunci K *Hill Cipher*. Dekripsi suatu *ciphertext* dengan menggunakan algoritma *Hill Cipher* dapat dilakukan dengan menerapkan persamaan $P = K^{-1}.C \ mod \ N$, dimana pada proses dekripsi sebelumnya dilakukan dengan menggunakan *Permutation Cipher* terdapat 3 blok *ciphertext*, sehingga diperoleh *plaintext* dengan *padding* yang sudah di tambahkan yaitu MAHASISWAITKDHL.

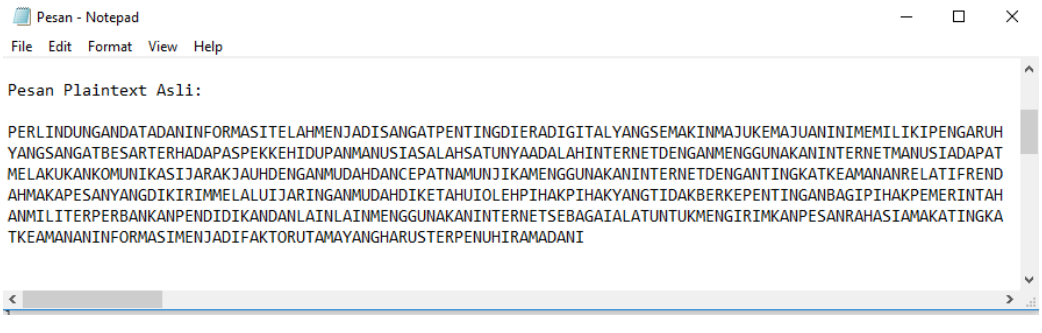
4. Simulasi Enkripsi dan Dekripsi

Proses enkripsi dan dekripsi akan disimulasikan dengan menggunakan pesan yang berbeda serta menggunakan beberapa kunci yang berbeda, dimana simulasi proses enkripsi dan dekripsi dilakukan dengan menggunakan bahasa pemrograman Python. Pada simulasi dengan menggunakan pesan yang berbeda, menggunakan suatu pesan yang memiliki karakter sebanyak 707 karakter termasuk huruf kecil dan kapital, spasi, nomor, dan juga tanda baca.



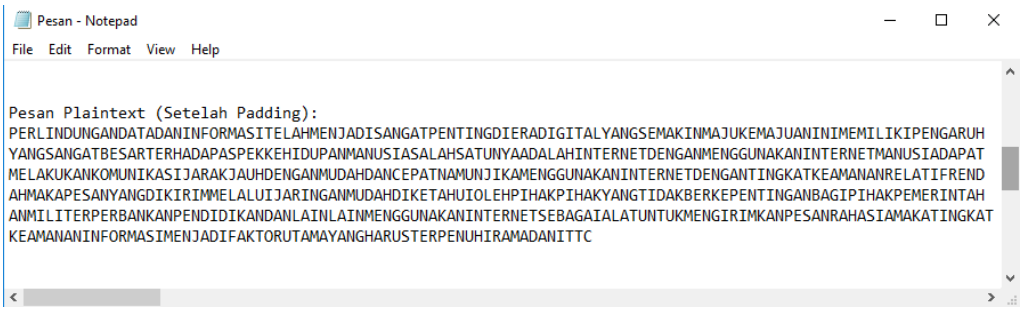
Gambar 3. Pesan Rahasia

Pesan rahasia pada Gambar 3, selanjutnya diubah menjadi *plaintext* berupa teks huruf kapital tanpa spasi, nomor atau karakter apapun kecuali alfabet sebagai berikut:



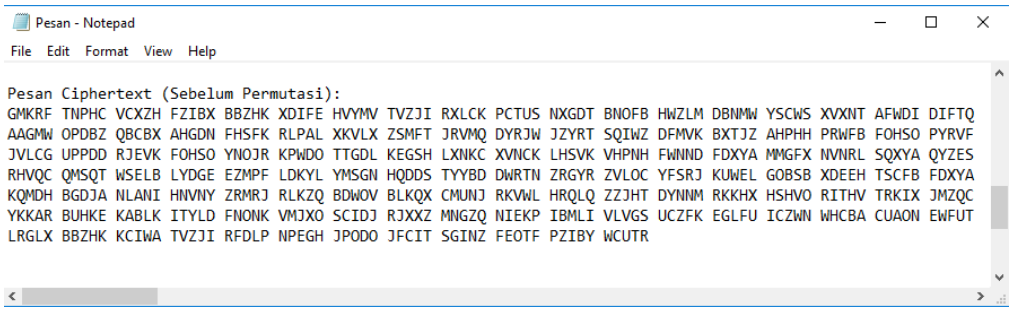
Gambar 4. *Plaintext* Asli

Plaintext asli pada Gambar 4 kemudian ditambahkan *padding* di akhir pesan sesuai dengan rancangan yang telah dibuat sebelumnya, sehingga teks menjadi:



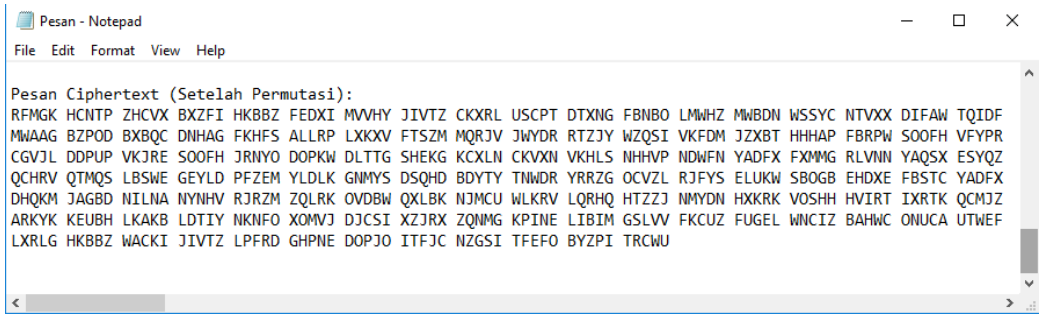
Gambar 5. *Plaintext* dengan *Padding*

Plaintext yang sudah ditambahkan *padding* ini selanjutnya dienkripsi mengggunakan algoritma *Hill Cipher* terlebih dahulu, dan diperoleh hasil sebagai berikut:



Gambar 6. *Ciphertext* Hill Cipher

Ciphertext yang diperoleh pada Gambar 6 merupakan *ciphertext* yang belum dipermutasi, sehingga selanjutnya dilakukan permutasi dari *ciphertext* yang telah diperoleh sebelumnya pada *Hill Cipher* dan diperoleh hasil sebagai berikut:



Gambar 7. *Ciphertext* Setelah Permutasi

Ciphertext yang telah diperoleh dari keseluruhan tahap enkripsi kemudian didekripsi dengan memasukkan input yaitu *ciphertext* dan kunci yang sama pada proses enkripsi dimana kunci di invers terlebih dahulu, sehingga didapatkan

kembali *plaintext* seperti pada Gambar 5. Terlihat bahwa *plaintext* yang dihasilkan dari rancangan ini berhasil mengembalikan *ciphertext* yang sebelumnya sulit untuk dibaca dan dipahami, namun hasil *plaintext* yang dihasilkan disini bukan berisi *plaintext* yang sebenarnya, melainkan *plaintext* yang disertai dengan *padding* yang berguna apabila terdapat pihak yang tidak terkait akan terkecoh dengan *padding* yang sudah di tambahkan, dimana hanya pengirim dan penerima sebenarnya yang mengetahui isi *plaintext* yang sebenarnya. Selanjutnya, dilakukan pengambilan waktu komputasi terhadap performa program pada proses enkripsi dan dekripsi sebanyak lima kali percobaan menggunakan *file.txt* yang berisi teks Undang-undang Dasar yang berbeda karakternya, yaitu sebagai berikut.

Tabel 1. Waktu Komputasi Enkripsi dan Dekripsi.

No.	*Plaintext*	Rata-rata waktu enkripsi (milidetik)	Rata-rata waktu dekripsi (milidetik)
1.	teksnormal.txt	1.1476536	9.7639848
2.	tanpaspasi.txt	0.9310666	8.0445124
3.	tanpatandabaca.txt	1.1324636	6.7689578
4.	hurufkecil.txt	0.766846	8.1908408
5.	hurufkapital.text	0.7155142	9.9049714

Dalam perhitungan waktu digunakan Python pada *Google Collaboratory* pada computer dengan RAM ~16GB dan Prosesor Intel® Core™ i5-1240P (16 CPUs) generasi 12. Diperoleh hasil rata-rata performa waktu komputasi pada Tabel 1 saat proses enkripsi dan dekripsi dengan pesan berupa teks normal berturut-turut adalah 1.1476536 milidetik dan 9.7639848 milidetik. Rata-rata performa waktu komputasi pesan tanpa spasi saat proses enkripsi dan dekripsi berturut-turut adalah 0.9310666 milidetik dan 8.0445124 milidetik. Rata-rata performa waktu komputasi pesan tanpa tanda baca saat proses enkripsi dan dekripsi berturut-turut adalah 1.1324636 milidetik dan 6.7689578 milidetik. Rata-rata performa waktu komputasi pesan dengan huruf kecil saat proses enkripsi dan dekripsi berturut-turut adalah 0.766846 milidetik dan 8.1908408 milidetik. Rata-rata performa waktu komputasi pesan dengan huruf kapital saat proses enkripsi dan dekripsi berturut-turut adalah 0.7155142 milidetik dan 9.9049714 milidetik. Terlihat pada rata-rata waktu komputasi terkecil terjadi pada saat proses enkripsi pesan dengan huruf kapital. Dan rata-rata waktu komputasi terbesar pada saat proses enkripsi pesan dengan pesan yang berisi teks normal yang terdapat tanda baca, huruf kapital dan huruf kecil, serta spasi.

Simulasi ini akan menggunakan beberapa kunci yang berbeda dari percobaan sebelumnya baik pada kunci *Hill Cipher* maupun pada Permutasi dengan menggunakan pesan Matematika. Selanjutnya dilakukan pengambilan waktu1 komputasi performa program pada proses enkripsi dan dekripsi sebanyak lima kali percobaan pada tiap kunci yang digunakan secara berturut-turut yaitu sebagai berikut.

Tabel 2. Simulasi dengan Kunci Berbeda.

No.	Kunci *Hill Cipher*	Kunci *Permutation Chiper*	Hasil Enkripsi	Hasil Deskripsi
1.	$\begin{bmatrix} 3 & 1 & 4 & 7 & 5 \\ 2 & 9 & 6 & 8 & 10 \\ 11 & 13 & 15 & 12 & 14 \\ 16 & 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 & 25 \end{bmatrix}$	(3,1,4,5,2)	JSSTE TRHKN IQGZS	MATEMATIKAVYBQZ
2.	$\begin{bmatrix} 17 & 17 & 0 & 4 & 10 \\ 14 & 24 & 5 & 3 & 24 \\ 18 & 17 & 20 & 7 & 21 \\ 13 & 2 & 8 & 24 & 1 \\ 16 & 23 & 18 & 1 & 14 \end{bmatrix}$	(2,3,5,1,4)	RSECA GHTZE KKDGM	MATEMATIKATHQKH
3.	$\begin{bmatrix} 15 & 7 & 12 & 6 & 4 \\ 18 & 9 & 10 & 12 & 21 \\ 23 & 4 & 14 & 15 & 16 \\ 7 & 8 & 9 & 10 & 11 \\ 3 & 1 & 4 & 5 & 2 \end{bmatrix}$	(4,3,2,5,1)	LOEAM MAHXD BTTNH	MATEMATIKAJGNAD
4.	$\begin{bmatrix} 15 & 1 & 14 & 9 & 8 \\ 10 & 2 & 12 & 5 & 3 \\ 7 & 11 & 4 & 13 & 16 \\ 6 & 18 & 25 & 21 & 23 \\ 19 & 20 & 17 & 22 & 24 \end{bmatrix}$	(5,3,1,2,4)	RZSCE MXFPR MGUHY	MATEMATIKAUHDFJ
5.	$\begin{bmatrix} 9 & 2 & 8 & 12 & 3 \\ 4 & 7 & 6 & 15 & 10 \\ 5 & 1 & 11 & 14 & 25 \\ 13 & 16 & 17 & 18 & 19 \\ 20 & 21 & 22 & 23 & 24 \end{bmatrix}$	(2,4,1,5,3)	GYEFJ HTCQQ ZGNZN	MATEMATIKAGLDQT

Berdasarkan hasil simulasi yang dilakukan sebanyak 5 kali, terlihat bahwa hasil dari proses enkripsi dan dekripsi terdiri dari 15 karakter yang selanjutnya dibagi menjadi blok-blok yang ukuranya sesuai dengan matriks kunci yang digunakan. Hasil dari proses enkripsi tersebut terbagi menjadi 3 blok yang setiap bloknya terdiri dari 5 karakter. Terlihat juga bahwa matriks kunci yang digunakan harus *invertible* (memiliki invers) dimana matriks dengan ukuran $m \times m$ memiliki setidaknya satu unit dari Ring $\mathbb{Z}_n$ pada tiap kolom atau baris, pada matriks yang digunakan juga tidak ada baris maupun kolom yang tidak terdapat suatu unit. Dalam simulasi ini unit yang dimaksud dalam Ring $\mathbb{Z}_{26}$ dan menggunakan ukuran matriks 5×5 berdasarkan batasan masalah yaitu berfokus pada penggunaan panjang blok *Hill* dan teknik permutasi yang ditetapkan pada 5 karakter.

KESIMPULAN

Berdasarkan hasil penelitian yang sudah dilaksanakan, dengan menggunakan matriks berukuran 5×5 dapat digunakan dalam mengamankan pesan teks dengan

menggunakan sandi hill atas gelanggang $\mathbb{Z}_{26}$. Hasil rata-rata waktu komputasi untuk enkripsi yaitu sebesar 1.1476536 milidetik, sedangkan pada proses dekripsi yaitu 9.7639848 milidetik. Dalam penelitian selanjutnya dapat digunakan matriks yang lebih besar atau lebih kecil kemudian dapat dibandingkan dengan waktu komputasi untuk enkripsi dan dekripsi.

UCAPAN TERIMA KASIH

Penulis ucapkan terima kasih yang sebesar-besarnya kepada Lembaga Penelitian dan Pengabdian Kepada Masyarakat (LPPM) Institut Teknologi Kalimantan (ITK) yang telah mendanai penelitian ini.

REFERENSI

Courtois, N. T., & Grajek, M. (2022). On Latin Squares, Invariant Differentials, Random Permutations and Historical Enigma Rotors. *Cryptologia*. Vol. 46(5), 387–421. https://doi.org/10.1080/01611194.2021.1920070

Forouzan, B. A. (2020). *Introduction to Cryptography and Network Security* (1st ed.). McGraw-Hill.

Gunawan, I. (2018). Kombinasi Algoritma Caesar Cipher dan Algoritma RSA untuk pengamanan File Dokumen dan Pesan Teks. *InfoTekJar (Jurnal Nasional Informatika dan Teknologi Jaringan)*. Vol. 2(2), 124–129. https://doi.org/10.30743/infotekjar.v2i2.266

Hewage, C., Jayal, A., Jenkins, G., & Brown, R. J. (2022). A Learned Polyalphabetic Decryption Cipher. *SNE Simulation Notes Europe*. Vol. 28(4), 141–148.

John, M. N., Ozioma, O., Otobong. G., U., Nwala, B. O., & Ngozi, O. P. (2023). Cryptographic Encryption Based on Rail-Fence Permutation Cipher. *GPH - International Journal of Mathematics*. Vol. 6(11), 1–6.

Jolfaei, A., Wu, X.-W., & Muthukkumarasamy, V. (2016). On the Security of Permutation-Only Image Encryption Schemes. *IEEE Transactions on Information Forensics and Security*. Vol. 11(2), 235–246. https://doi.org/10.1109/TIFS.2015.2489178

Lakhera, M., Rauthan, M. M. S., & Agarwal, A. (2016). Securing Biometric Template Using Double Hill Cipher with Self-invertible Key and Random Permutation of Pixels Locations. *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*. 814–817. https://doi.org/10.1109/NGCT.2016.7877522

Mohan, M., Kavithadevi, M. K., & Jeevan Prakash, V. (2016). Improved Classical Cipher for Healthcare Applications. *Procedia Computer Science*. Vol. 93, 742–750. https://doi.org/10.1016/j.procs.2016.07.285

Paragas, J. R., Sison, A. M., & Medina, R. P. (2019). Hill Cipher Modification: A Simplified Approach. *2019 IEEE 11th International Conference on Communication Software and Networks (ICCSN)*. 821–825. https://doi.org/10.1109/ICCSN.2019.8905360

Ramadani, S. (2020). Hybird Cryptosystem Algoritma *Hill Cipher* dan Algoritma Elgamal pada Keamanan Citra. *Methomika Jurnal Manajemen Informatika dan Komputerisasi Akuntansi*. Vol. 4(1), 1–9. https://doi.org/10.46880/jmika.Vol4No1.pp1-9

Santoso, Y. S. (2021). Message Security Using a Combination of Hill Cipher and RSA Algorithms. *Jurnal Matematika dan Ilmu Pengetahuan Alam LLDikti Wilayah 1 (JUMPA)*. Vol. 1(1), 20–28. https://doi.org/10.54076/jumpa.v1i1.38

Stinson, D. R., & Paterson, M. B. (2018). *Cryptography*. Chapman and Hall/CRC. https://doi.org/10.1201/9781315282497

Tama, Y. B. W., & Fahmi, M. F. (2023). Sistem Kriptografi Klasik dengan Memanfaatkan Orde dari Grup Titik pada Kurva Eliptik Bentuk Montgomery. *Euler: Jurnal Ilmiah Matematika, Sains da n Teknologi*. Vol. 11(2), 361–371. https://doi.org/10.37905/euler.v11i2.23009

Vishwa Nageshwar, K., & Ravi Shankar, N. (2021). *Cryptanalysis of Modification in Hill Cipher for Cryptographic Application*. 659–666. https://doi.org/10.1007/978-981-15-5243-4_62