SISTEM KRIPTOGRAFI SUBSTITUSI-PERMUTASI ATAS ALJABAR MAX-PLUS

Daniel Suleman[1*], Nurwan[2] , Armayani Arsal[3] , Nisky Imansyah Yahya[4]
[1]*Program Studi Matematika, Universitas Negeri Gorontalo, Gorontalo, Indonesia*
[2]*Program Studi Matematika, Universitas Negeri Gorontalo, Gorontalo, Indonesia*

**Penulis korespondensi: danilsuleman84@gmail.com*

ABSTRAK

Keamanan data menjadi isu penting dalam era digital, sehingga dibutuhkan sistem kriptografi yang kuat namun efisien. Penelitian ini membahas perancangan dan evaluasi sistem kriptografi substitusi-permutasi berbasis aljabar Max-Plus sebagai alternatif pendekatan kriptografi modern yang efisien dan tahan terhadap serangan. Aljabar Max-Plus merupakan struktur semiring tropis dengan operasi utama maksimum ($\oplus$) dan penjumlahan ($\otimes$), yang memungkinkan pembentukan algoritma enkripsi non-linier. Proses enkripsi dilakukan melalui konversi plaintext menjadi blok numerik dan pengolahan menggunakan operasi substitusi-permutasi berbasis matriks kunci dalam ruang Max-Plus, sedangkan dekripsi menggunakan invers dari matriks tersebut. Evaluasi keamanan mencakup analisis frekuensi dan simulasi brute force, sementara efisiensi komputasi diukur dari waktu eksekusi dan penggunaan memori. Hasil penelitian menunjukkan bahwa sistem kriptografi berbasis Max-Plus menghasilkan ciphertext yang acak, sulit ditebak, dan tahan terhadap analisis sederhana, dengan efisiensi komputasi yang meningkat secara linier terhadap panjang plaintext. Dengan demikian, sistem substitusi-permutasi atas aljabar Max-Plus berpotensi menjadi pendekatan alternatif yang aman dan efisien dalam pengembangan kriptografi modern.

Kata kunci: kriptografi, aljabar Max-Plus, substitusi-permutasi, keamanan data, efisiensi komputasi

1 PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi yang pesat telah menjadikan data digital sebagai aset berharga dalam kehidupan modern. Laporan dari Social & Meltwater (2024) menunjukkan bahwa lebih dari 65% populasi dunia telah terkoneksi ke internet, dengan pertumbuhan yang terus meningkat setiap tahunnya. Namun, kemudahan akses informasi ini juga memunculkan tantangan serius dalam keamanan data, seperti meningkatnya serangan siber berupa *phishing*, peretasan sistem, *ransomware*, dan pencurian data pribadi. Kriptografi menjadi salah satu solusi utama untuk menjaga kerahasiaan, integritas, dan keaslian informasi melalui proses enkripsi dan dekripsi (Iwan Setiadi et al., 2025; Ridho & Romli, 2024). Algoritma kriptografi modern, seperti ElGamal yang berbasis pada permasalahan logaritma diskret (ElGamal, 1985), telah terbukti kuat dan banyak digunakan. Namun, eksplorasi pendekatan baru tetap diperlukan untuk mengantisipasi perkembangan teknologi komputasi, seperti komputasi kuantum, serta membuka jalur penelitian alternatif dalam matematika terapan.

Salah satu pendekatan yang kini dikembangkan adalah kriptografi berbasis struktur semiring, khususnya aljabar max-plus. Aljabar max-plus merupakan struktur aljabar pada himpunan $\mathbb{R} \cup \{-\infty\}$ dengan operasi "penjumlahan" $\oplus$ yang didefinisikan sebagai $\max(a, b)$ dan "perkalian" $\otimes$ sebagai $a + b$, sehingga termasuk dalam kategori semiring tropis atau idempoten (Amalia, 2023). Struktur ini telah banyak digunakan untuk memodelkan sistem diskrit seperti

penjadwalan, sistem produksi, dan jaringan transportasi karena memungkinkan penyederhanaan dinamika nonlinier (Kurniawan & Suparwanto, 2020). Baru-baru ini, aljabar max-plus mulai dilirik dalam bidang kriptografi. Kavut & Tutdere (2020) menunjukkan bahwa kriptografi dengan semiring dan aljabar tropis memiliki keunikan dalam operasi yang dapat memberikan lapisan keamanan tambahan. Muanalifah et al. (2023) mengembangkan algoritma enkripsi berbasis ElGamal dalam semiring tropis menggunakan matriks diagonal sebagai kunci, menunjukkan potensi ketahanan terhadap serangan berbasis logaritma diskret klasik. Chen et al. (2023) lebih lanjut menunjukkan bahwa permasalahan dalam aljabar tropis bersifat NP-sukar, yang mengindikasikan potensi tinggi untuk aplikasi kriptografi.

Keunggulan kriptografi berbasis max-plus terletak pada keunikan operasinya, kesederhanaan komputasi, dan potensi ketahanan terhadap algoritma kuantum. Struktur max-plus tidak memiliki invers multiplikatif dalam pengertian biasa, sehingga sulit dilakukan pembalikan proses enkripsi tanpa informasi kunci yang tepat (Kavut & Tutdere, 2020). Sistem substitusi-permutasi, yang merupakan dasar dari banyak *cipher* modern seperti AES (Stallings, 2017), bertujuan meningkatkan keamanan melalui prinsip *confusion* dan *diffusion* yang diperkenalkan Shannon (1949). *Confusion* mengaburkan hubungan antara *plaintext*, *ciphertext*, dan kunci, sedangkan *diffusion* menyebarkan struktur statistik *plaintext* ke seluruh *ciphertext*. Dengan mengintegrasikan prinsip ini ke dalam struktur aljabar max-plus, diharapkan dapat tercipta sistem kriptografi baru yang lebih tangguh terhadap berbagai serangan. Penelitian terkait menunjukkan bahwa permasalahan dalam aljabar tropis, seperti faktorisasi polinomial tropis, bersifat NP-sukar sehingga dapat meningkatkan keamanan sistem kriptografi (Chen et al., 2023).

Berdasarkan potensi tersebut, penelitian ini bertujuan untuk menganalisis cara kerja sistem kriptografi substitusi-permutasi berbasis aljabar max-plus dalam proses enkripsi dan dekripsi data, mengevaluasi tingkat keamanannya terhadap serangan kriptanalisis seperti *brute-force* dan analisis frekuensi, serta mengukur efisiensi komputasinya dalam hal waktu eksekusi dan penggunaan sumber daya. Penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan metode kriptografi berbasis aljabar max-plus, menyediakan dasar teoritis dan eksperimental bagi penelitian lebih lanjut, serta menawarkan pendekatan baru dalam desain sistem enkripsi yang lebih efisien dan aman.

2 METODE

2.1 Waktu dan Tempat Penelitian

Penelitian ini dilaksanakan pada periode Februari hingga Oktober 2025 di Laboratorium Komputasi Matematika Jurusan Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Gorontalo. Penelitian ini menggunakan perangkat keras berupa laptop dengan spesifikasi minimal prosesor Intel Core i3, RAM 8 GB, dan sistem operasi Windows 10. Perangkat lunak yang digunakan adalah Python 3.12.0 dengan pustaka NumPy untuk operasi matriks, Matplotlib untuk visualisasi data, dan Memory Profiler untuk pengukuran efisiensi sumber daya komputasi.

2.2 Desain Penelitian

Penelitian ini merupakan penelitian studi literatur (*library research*) yang dikombinasikan dengan simulasi komputasi. Studi literatur dilakukan dengan melakukan telaah terhadap berbagai sumber seperti jurnal ilmiah, artikel, dan referensi terkait konsep kriptografi, metode substitusi-permutasi, serta penerapan aljabar max-plus dalam sistem keamanan informasi. Simulasi komputasi dilakukan untuk mengimplementasikan dan mengevaluasi sistem kriptografi yang

dirancang. Sistem kriptografi dibangun sepenuhnya berdasarkan operasi dalam aljabar max-plus, yaitu operasi maksimum ($\oplus$) sebagai pengganti penjumlahan dan operasi penjumlahan (+) sebagai pengganti perkalian. Proses enkripsi dilakukan melalui dua tahap utama: substitusi menggunakan matriks kunci diagonal berukuran 3×3 yang bersifat invertibel dalam ruang aljabar max-plus, dan permutasi menggunakan matriks permutasi max-plus yang memiliki satu elemen bernilai nol pada setiap baris dan kolom dengan sisanya bernilai minus tak hingga ($-\infty$).

2.3 Prosedur Penelitian

2.3.1 Konversi Plaintext

Plaintext dikonversi ke bentuk numerik menggunakan kode ASCII standar, di mana setiap karakter dipetakan menjadi bilangan bulat dalam rentang 0-127. Data numerik kemudian dibagi menjadi blok-blok vektor berukuran 3×1. Jika panjang *plaintext* tidak habis dibagi tiga, dilakukan *padding* dengan menambahkan karakter 'X' (ASCII 88) hingga panjang data menjadi kelipatan tiga.

2.3.2 Proses Enkripsi

Proses enkripsi terdiri dari dua tahap transformasi. Tahap pertama adalah substitusi, di mana setiap blok vektor *plaintext* P dikalikan dengan matriks kunci substitusi diagonal K menggunakan operasi max-plus:

$$C = K \otimes P$$

dengan elemen hasil dihitung sebagai:

$$(C)_i = \max_{1 \le j \le 3} \left(K_{ij} + P_j \right).$$

Matriks kunci K yang digunakan adalah matriks diagonal 3×3 dengan elemen diagonal bernilai real dan elemen non-diagonal bernilai $-\infty$. Tahap kedua adalah permutasi menggunakan matriks permutasi P yang mengacak posisi elemen dalam vektor hasil substitusi:

$$C' = P \otimes C.$$

Matriks permutasi dirancang untuk melakukan rotasi elemen dengan struktur yang menjamin keberadaan invers.

2.3.3 Proses Dekripsi

Dekripsi dilakukan dengan membalik urutan operasi enkripsi. Pertama, dilakukan invers permutasi menggunakan matriks P^{-1}:

$$C = P^{-1} \otimes C'.$$

Kemudian dilakukan invers substitusi menggunakan matriks K^{-1}:

$$P = K^{-1} \otimes C.$$

Matriks K^{-1} diperoleh dengan mengubah tanda elemen-elemen diagonal matriks K, karena dalam aljabar max-plus invers penjumlahan direpresentasikan sebagai pengurangan biasa. Hasil akhir dikonversi kembali dari kode ASCII menjadi karakter untuk memperoleh *plaintext* asli.

2.4 Evaluasi Sistem

2.4.1 Evaluasi Keamanan

Evaluasi keamanan dilakukan melalui dua pendekatan utama, yaitu analisis frekuensi dan simulasi serangan brute force. Analisis frekuensi adalah teknik kriptanalisis klasik yang mengamati distribusi kemunculan simbol dalam ciphertext untuk mendeteksi pola statistika yang

dapat dieksploitasi guna memulihkan plaintext; meskipun asal-usul teoritisnya tercatat sejak Shannon, teknik ini tetap relevan dan terus dimodifikasi dalam studi modern misalnya sebagai bagian dari serangan terhadap sistem deduplikasi terenkripsi, metode metaheuristik untuk memecahkan sandi substitusi, dan evaluasi kebocoran frekuensi pada layanan terenkripsi (Li et al., 2022). Dalam penelitian ini, analisis frekuensi dilakukan untuk mengukur distribusi karakter dalam ciphertext dan membandingkannya dengan distribusi acak menggunakan visualisasi grafik batang.

Selanjutnya, simulasi serangan brute force dilakukan sebagai metode kriptanalisis dengan mencoba seluruh kemungkinan kunci secara sistematis hingga diperoleh plaintext yang sesuai (Humaira & Rafiqa, 2019). Pada penelitian ini, serangan brute force dilakukan dengan menguji 1000 kombinasi acak matriks permutasi dan kunci substitusi. Ruang kunci dihitung berdasarkan jumlah permutasi blok (3! = 6) dan rentang nilai elemen diagonal matriks kunci. Keberhasilan serangan dinilai berdasarkan kemampuan sistem dalam memulihkan plaintext asli.

2.4.2 Evaluasi Efisiensi Komputasi

Efisiensi komputasi diukur melalui dua parameter. Pertama, waktu eksekusi diukur dengan mengenkripsi *plaintext* dengan panjang bervariasi (3 hingga 57 karakter dengan kenaikan 3 karakter) sebanyak 1000 kali untuk setiap panjang menggunakan modul time.perf counter dalam Python. Rata-rata dan standar deviasi waktu eksekusi dihitung dan divisualisasikan dalam grafik. Kedua, penggunaan memori diukur menggunakan pustaka memory_profiler untuk mencatat penggunaan memori puncak (*peak memory usage*) selama proses enkripsi pada berbagai panjang *plaintext*. Hasil pengukuran dianalisis untuk mengevaluasi skalabilitas algoritma terhadap peningkatan ukuran data.

3 HASIL DAN PEMBAHASAN

3.1 Rancangan Sistem Kriptografi Substitusi-Permutasi Berbasis Aljabar Max-Plus

3.1.1 Konversi Plaintext dan Pembentukan Blok

Sistem kriptografi yang dirancang menggunakan operasi dalam aljabar max-plus, yaitu struktur aljabar pada himpunan $\mathbb{R}_{\max} = \mathbb{R} \cup \{-\infty\}$dengan dua operasi biner, yaitu operasi maksimum $a \oplus b = \max(a, b)$dan operasi penjumlahan $a \otimes b = a + b$. Plaintext P terlebih dahulu dikonversi ke bentuk numerik menggunakan fungsi ASCII, sehingga setiap karakter c_i dipetakan menjadi nilai integer $p_i = \text{ASCII}(c_i)$, dengan $i = 1,2, \dots, n$, di mana n adalah panjang plaintext.

Untuk memfasilitasi operasi matriks, data numerik dibagi menjadi blok-blok vektor berukuran 3×1. Jika $n \ mod \ 3 \neq 0$, maka dilakukan *padding* dengan menambahkan karakter ‘X’ (ASCII = 88) sebanyak $3 - (n \ mod \ 3)$ karakter. Setelah proses *padding*, plaintext memiliki panjang baru $n' = n + (3 - n \ mod \ 3)$ dan dibagi menjadi $m = \frac{n'}{3}$ blok. Blok ke-k didefinisikan sebagai

$$P^{(k)} = \begin{matrix} p_{3k-2} \\ [p_{3k-1}], \\ p_{3k} \end{matrix} \qquad k = 1,2, \dots, m.$$

Sebagai contoh, plaintext “MAHASISWA MATEMATIKA” memiliki $n = 20$ karakter. Setelah dilakukan *padding* dengan satu karakter ‘X’, diperoleh $n' = 21$ dan $m = 7$ blok. Hasil konversi ASCII menghasilkan vektor

$$P = [77,65,72,65,83,73,83,87,65,32,77,65,84,69,77,65,84,73,75,65,88].$$

Pembagian blok menghasilkan $P^{(1)} = [77,65,72]^T$, $P^{(2)} = [65,83,73]^T$, dan seterusnya hingga $P^{(7)} = [75,65,88]^T$.

3.1.2 Proses Substitusi

Tahap substitusi dilakukan dengan mengalikan setiap blok plaintext dengan matriks kunci substitusi Kmenggunakan operasi perkalian matriks–vektor dalam aljabar max-plus. Matriks kunci Kmerupakan matriks diagonal berukuran 3×3yang didefinisikan sebagai

$$K = \begin{bmatrix} k_1 & -\infty & -\infty \\ -\infty & k_2 & -\infty \\ -\infty & -\infty & k_3 \end{bmatrix},$$

dengan $k_1 = 2$, $k_2 = 3$, dan $k_3 = 1$. Operasi substitusi untuk blok ke-k didefinisikan sebagai

$$C^{(k)} = K \otimes P^{(k)}.$$

Elemen ke-i dari vektor hasil dihitung menggunakan operasi max-plus sebagai

$$(C^{(k)})_i = \bigoplus_{j=1}^{3} (K_{ij} \otimes P_j^{(k)}) = \max_{j=1,2,3} (K_{ij} + P_j^{(k)}).$$

Karena matriks Kberbentuk diagonal, maka perhitungan tersebut menyederhana menjadi

$$(C^{(k)})_i = K_{ii} + P_i^{(k)} = k_i + p_{3k-3+i}.$$

Sebagai ilustrasi, untuk blok pertama $P^{(1)} = [77, 65, 72]^T$, diperoleh hasil substitusi:

$$(C^{(1)})_1 = \max(2 + 77, -\infty + 65, -\infty + 72) = 79,$$
$$(C^{(1)})_2 = \max(-\infty + 77, 3 + 65, -\infty + 72) = 68,$$
$$(C^{(1)})_3 = \max(-\infty + 77, -\infty + 65, 1 + 72) = 73.$$

Dengan demikian diperoleh

$$C^{(1)} = [79, 68, 73]^T.$$

Proses yang sama diterapkan pada seluruh blok lainnya sehingga menghasilkan hasil substitusi sebagai berikut :

$$C = [79,68,73,67,86,74,85,90,66,34,80,66,86,72,78,67,87,74,77,68,89]$$

3.1.3 Proses Permutasi

Tahap permutasi bertujuan untuk mengacak posisi elemen dalam setiap blok hasil substitusi dengan menggunakan matriks permutasi P berukuran 3×3 yang didefinisikan sebagai

$$P = \begin{bmatrix} -\infty & 0 & -\infty \\ -\infty & -\infty & 0 \\ 0 & -\infty & -\infty \end{bmatrix}$$

Matriks ini merepresentasikan operasi rotasi kanan pada elemen vektor, yaitu elemen pertama berpindah ke posisi terakhir, elemen kedua berpindah ke posisi pertama, dan elemen ketiga berpindah ke posisi kedua. Operasi permutasi untuk blok ke-k didefinisikan sebagai

$$C'^{(k)} = P \otimes C^{(k)},$$

dengan elemen ke-i dihitung menggunakan operasi max-plus sebagai

$$(C'^{(k)})_i = \max_{j=1,2,3}(P_{ij} + C_j^{(k)}).$$

Untuk blok pertama $C^{(1)} = [79, 68, 73]^T$, hasil permutasi diperoleh sebagai berikut:

$$(C'^{(1)})_1 = \max(-\infty + 79, 0 + 68, -\infty + 73) = 68,$$
$$(C'^{(1)})_2 = \max(-\infty + 79, -\infty + 68, 0 + 73) = 73,$$
$$(C'^{(1)})_3 = \max(0 + 79, -\infty + 68, -\infty + 73) = 79.$$

Dengan demikian diperoleh

$$C'(1) = [68{,}73{,}79]T.$$

Ciphertext akhir diperoleh dengan menggabungkan seluruh blok hasil permutasi sebagai

$$C' = [(C'^{(1)})^T, (C'^{(2)})^T, \ldots, (C'^{(7)})^T]^T,$$

sehingga dihasilkan vektor *ciphertext*:

$$C' = [68, 73, 79, 86, 74, 67, 90, 66, 85, 80, 66, 34, 72, 78, 86, 87, 74, 67, 68, 89, 77].$$

3.1.4 Proses Dekripsi

Proses dekripsi dilakukan dengan membalik urutan operasi enkripsi secara matematis. Tahap pertama adalah melakukan invers permutasi menggunakan matriks P^{-1} yang didefinisikan sebagai

$$P^{-1} = \begin{bmatrix} -\infty & -\infty & 0 \\ 0 & -\infty & -\infty \\ -\infty & 0 & -\infty \end{bmatrix},$$

yang merupakan matriks rotasi kiri. Operasi invers permutasi untuk blok ke-k didefinisikan sebagai

$$C^{(k)} = P^{-1} \otimes C'^{(k)},$$

yang berfungsi untuk mengembalikan vektor ke posisi sebelum permutasi. Tahap kedua adalah melakukan invers substitusi menggunakan matriks

$$K^{-1} = \begin{bmatrix} -k_1 & -\infty & -\infty \\ -\infty & -k_2 & -\infty \\ -\infty & -\infty & -k_3 \end{bmatrix},$$

dengan $-\mathrm{k}_1 = -2$, $-\mathrm{k}_2 = -3$, dan $-\mathrm{k}_3 = -1$. Operasi invers substitusi untuk blok ke-k dinyatakan sebagai

$$P^{(k)} = K^{-1} \otimes C^{(k)},$$

yang bertujuan untuk mengembalikan blok ciphertext menjadi blok plaintext semula. Untuk ciphertext pertama $\mathrm{C}'^{(1)} = [68, 73, 79]^{\mathrm{T}}$, hasil invers permutasi adalah $\mathrm{C}^{(1)} = [79, 68, 73]^{\mathrm{T}}$, dan invers substitusi menghasilkan

$$(P^{(1)})_i = K_{ii}^{-1} + C_i^{(1)} = -k_i + C_i^{(1)}.$$

Dengan demikian diperoleh

$$P^{(1)} = [79 - 2, 68 - 3, 73 - 1]^T = [77, 65, 72]^T,$$

yang identik dengan blok plaintext asli. Proses ini diterapkan pada seluruh blok, kemudian hasil akhirnya dikonversi kembali dari nilai ASCII ke karakter, sehingga diperoleh plaintext "MAHASISWA MATEMATIKA", yang membuktikan bahwa sistem kriptografi yang dirancang bersifat reversibel.

3.2 Evaluasi Keamanan Sistem

3.1.5 Analisis Frekuensi

Analisis frekuensi dilakukan untuk mengevaluasi distribusi karakter dalam *ciphertext* dan mendeteksi kemungkinan adanya pola statistik yang dapat dieksploitasi. Tabel 1 menunjukkan distribusi frekuensi kemunculan setiap nilai numerik dalam ciphertext hasil enkripsi *plaintext* "MAHASISWA MATEMATIKA".

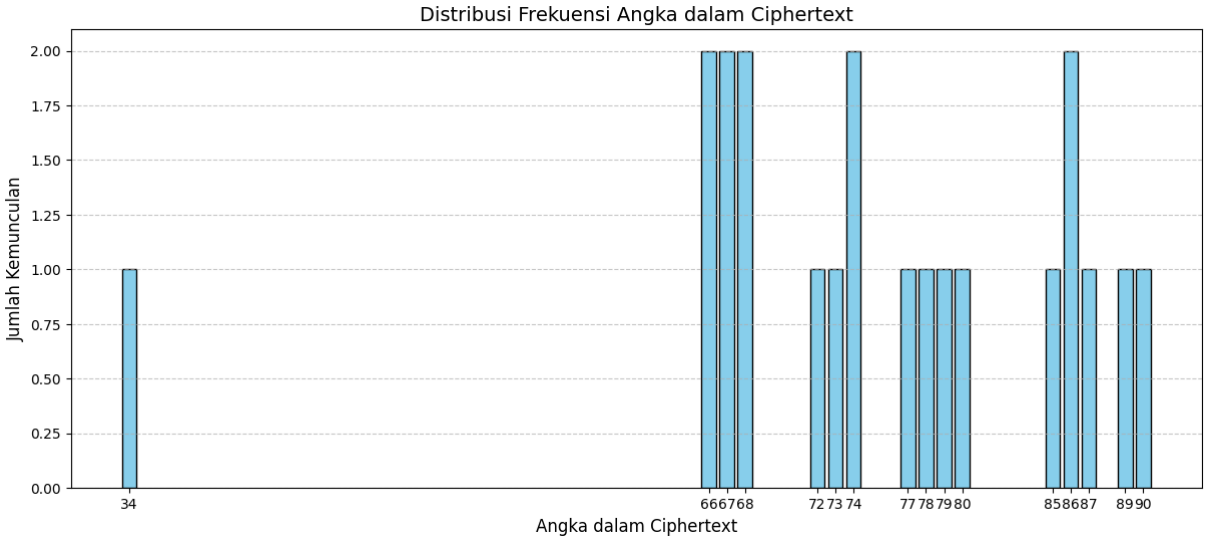
Tabel 1. Distribusi Frekuensi Karakter dalam Ciphertext

Simbol (ASCII)	Jumlah kemunculan
34	1
66	2
67	2
68	2
72	1
73	1
74	2
77	1
78	1
79	1
80	1
85	1
86	2

87	1
89	1
90	1

Berdasarkan Tabel 1, terlihat bahwa tidak terdapat dominasi nilai numerik tertentu dalam ciphertext. Sebagian besar nilai hanya muncul satu atau dua kali, dan tidak membentuk pola statistik yang mencolok. Distribusi ini mengindikasikan bahwa ciphertext memiliki karakteristik keacakan yang tinggi dan tidak mencerminkan distribusi frekuensi bahasa alami yang biasanya memiliki karakter dominan tertentu.

Visualisasi distribusi frekuensi ditampilkan dalam Gambar 1 untuk memperjelas pola penyebaran nilai dalam ciphertext. Grafik menunjukkan bahwa mayoritas nilai memiliki frekuensi kemunculan yang relatif seragam tanpa adanya puncak frekuensi yang signifikan.



Gambar 1. Grafik Hasil Analisis Frekuensi Ciphertext

Gambar 1 menunjukkan bahwa distribusi nilai numerik dalam ciphertext relatif seragam dengan frekuensi kemunculan setiap nilai berada dalam kisaran 1-2 kali. Ketidakhadiran nilai dengan frekuensi yang jauh lebih tinggi artinya tidak terdapat "puncak" dominan mendukung kesimpulan bahwa algoritma enkripsi berhasil menyamarkan struktur statistik dari plaintext. Kondisi tersebut mengindikasikan bahwa metode klasik kriptanalisis berbasis distribusi frekuensi tidak dapat secara efektif mengekstrak pola atau struktur dari plaintext asli. Dengan demikian, sistem enkripsi menunjukkan ketahanan terhadap serangan statistik tradisional, terutama bila mekanisme enkripsi melibatkan randomisasi atau penyamaran frekuensi (*frequency smoothing*) untuk menanggulangi kebocoran frekuensi (Li et al., 2022).

3.1.6 Serangan Brute Force

Evaluasi ketahanan terhadap serangan brute force dilakukan dengan mensimulasikan upaya dekripsi menggunakan kombinasi acak matriks permutasi dan kunci substitusi. Simulasi dilakukan sebanyak 1000 kali dengan mencoba berbagai kombinasi matriks permutasi (6 kemungkinan rotasi untuk blok 3 elemen) dan kunci substitusi diagonal dengan rentang nilai [−999, 999] untuk setiap

elemen diagonal. Tabel 2 menampilkan lima contoh hasil simulasi yang tidak berhasil memulihkan plaintext asli.

Tabel 2. Contoh Hasil Simulasi Brute Force yang Tidak Valid

No	Permutasi	Kunci Substitusi	Ciphertext hasil dekripsi
1	[1, 0, 2]	[-104, 804, 514]	[177, -736, -435, 178, -718, -447, 170, -714, -429, 170, -724, -480, 182, -73, -428, 178,-717, -447, 193, -736, -437]
2	[1, 2, 0]	[20, 829, -222]	[59, −761, 295, 47, −743, 296, 65, −739, 288, 14, −749, 288, 66, −757, 300, 47, −742, 296, 57, −761, 311]
3	[1, 2, 0]	[-63, 321, -408]	[142, −253, 481, 130, −235, 482, 148, −231, 474, 97, −241, 474, 149, −249, 486, 130, −234, 482, 140, −253, 497]
4	[1, 0, 2]	[396, -994, -103]	[−323, 1062, 182, −322, 1080, 170, −330, 1084, 188, −330, 1074, 137, −318, 1066, 189, −322, 1081, 170, −307, 1062, 180]
5	[2, 1, 0]	[-960, -237, 208]	[1039, 310, −140, 1027, 311, −122, 1045, 303, −118, 994, 303, −128, 1046, 315, −136, 1027, 311, −121, 1037, 326, −140]

Meskipun simulasi hanya menguji 1000 kombinasi kunci secara acak, tidak satu pun yang berhasil mengembalikan plainteks. Namun, argumen utama ketahanan terhadap brute force berasal dari ukuran ruang kunci yang mencapai sekitar $4{,}78 \times 10^{10}$ kombinasi, sehingga pencarian ekshaustif menjadi tidak realistis secara komputasional. (Katz & Lindell, 2020). Perhitungan teoritis menunjukkan bahwa total kombinasi brute force adalah hasil perkalian antara jumlah permutasi ($3! = 6$) dengan kemungkinan nilai kunci substitusi ($1999^3 \approx 7{,}97 \times 10^9$), menghasilkan sekitar $4{,}78 \times 10^{10}$ kombinasi. Lebih lanjut, karena sistem tidak membatasi nilai elemen diagonal hanya pada bilangan bulat dalam rentang terbatas, ruang kunci sesungguhnya dapat diperluas hingga tak terbatas jika menggunakan bilangan real. Kompleksitas ini menjadikan metode brute force sebagai pendekatan yang tidak realistis secara komputasional (*computationally infeasible*) untuk membobol sistem ini (Alagic et al., 2022; Stinson & Paterson, 2018).

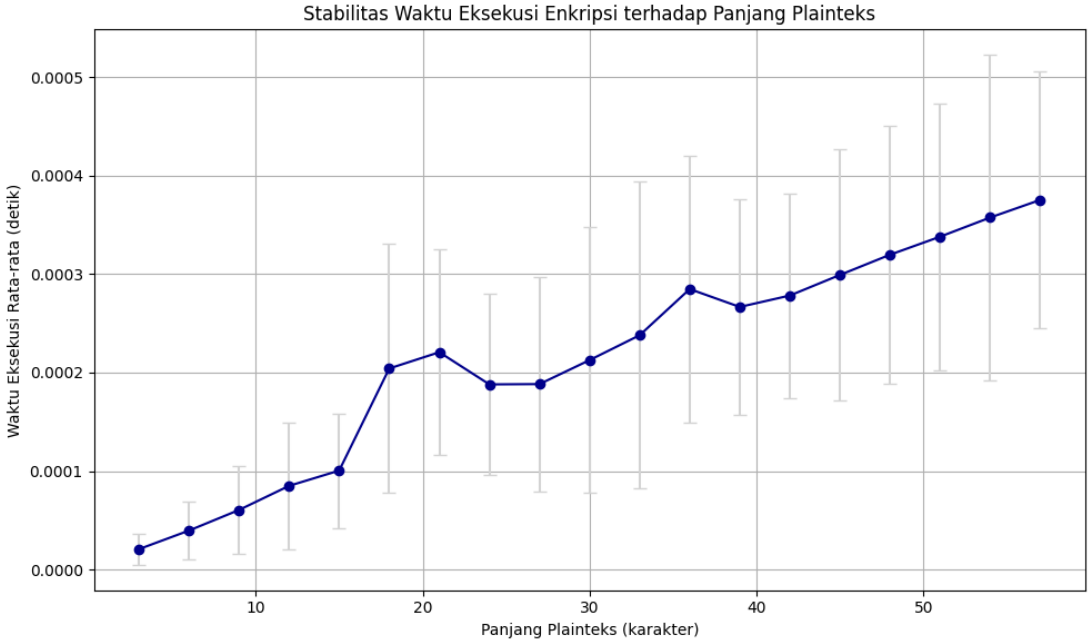
3.3 Evaluasi Efisiensi Komputasi

3.1.7 Waktu Eksekusi

Pengukuran waktu eksekusi dilakukan dengan mengenkripsi plaintext dengan panjang bervariasi dari 3 hingga 57 karakter dengan kenaikan 3 karakter setiap pengujian. Setiap proses enkripsi diulang sebanyak 1000 kali untuk memperoleh nilai rata-rata yang stabil dan akurat. Gambar 2 menunjukkan hubungan antara panjang plaintext dan rata-rata waktu eksekusi proses enkripsi.

Gambar 2 menunjukkan bahwa waktu eksekusi meningkat secara bertahap seiring dengan bertambahnya panjang plaintext, mengikuti kompleksitas linier O(n) dari algoritma yang digunakan (Banu et al., 2025). Meskipun terdapat fluktuasi minor pada beberapa titik data, tren keseluruhan menunjukkan pertumbuhan yang konsisten dan proporsional. Fluktuasi yang terjadi disebabkan oleh faktor non-deterministik dalam lingkungan eksekusi Python seperti beban CPU, manajemen memori, dan aktivitas garbage collector (Castro et al., 2024; Ruys et al., 2025). Secara

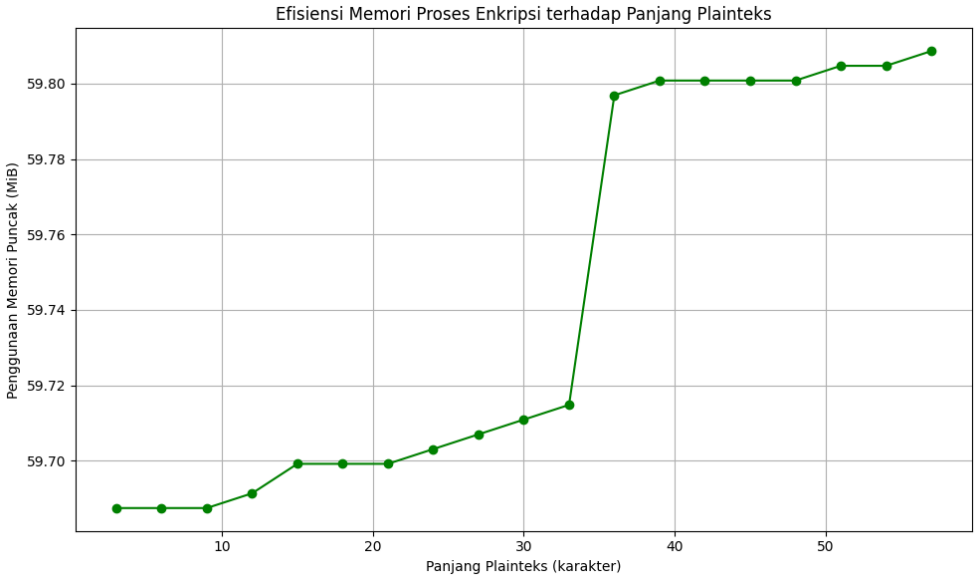
umum, hasil ini menunjukkan bahwa sistem memiliki efisiensi yang baik dalam menangani peningkatan jumlah data dan dapat diandalkan untuk aplikasi dengan variasi ukuran plaintext.



Gambar 2. Grafik Rata-rata Waktu Eksekusi Enkripsi Terhadap Panjang Plaintext

3.1.8 Penggunaan Memori

Evaluasi efisiensi sumber daya dilakukan dengan mengukur penggunaan memori puncak (peak memory usage) selama proses enkripsi menggunakan pustaka memory_profiler dalam Python. Pengujian dilakukan terhadap plaintext dengan panjang 3 hingga 57 karakter. Gambar 3 menampilkan grafik penggunaan memori puncak terhadap panjang plaintext.



Gambar 3. Penggunaan memori puncak terhadap panjang plaintext

Gambar 3 menunjukkan bahwa penggunaan memori relatif stabil pada rentang 59,28–59,32 MiB untuk plaintext dengan panjang 3 hingga 36 karakter. Ketika panjang plaintext mencapai 39 karakter, terjadi kenaikan ke sekitar 59,43 MiB yang kemudian menetap stabil untuk panjang berikutnya. Konsistensi penggunaan memori ini terjadi karena proses enkripsi dilakukan per blok (3 karakter per blok) tanpa menyimpan keseluruhan hasil atau memuat seluruh plaintext sekaligus ke dalam memori. Dengan demikian, kompleksitas ruang (space complexity) dari algoritma ini tidak meningkat secara signifikan meskipun panjang plaintext diperbesar, menunjukkan bahwa algoritma memiliki efisiensi tinggi terhadap sumber daya memori dan sangat cocok untuk diimplementasikan dalam sistem dengan keterbatasan memori seperti perangkat tertanam (embedded system) atau Internet of Things (IoT).

4 KESIMPULAN

Penelitian ini berhasil merancang dan mengevaluasi sistem kriptografi substitusi-permutasi berbasis aljabar max-plus sebagai pendekatan alternatif dalam pengamanan informasi. Sistem yang dikembangkan memanfaatkan operasi maksimum dan penjumlahan dalam kerangka aljabar max-plus untuk melakukan transformasi plaintext melalui dua tahap utama: substitusi menggunakan matriks kunci diagonal berukuran 3×3 dan permutasi menggunakan matriks rotasi. Proses enkripsi dan dekripsi yang dirancang terbukti bersifat reversibel, di mana plaintext dapat dipulihkan secara sempurna melalui operasi invers matriks substitusi dan permutasi dalam ruang max-plus.

Evaluasi keamanan menunjukkan bahwa sistem memiliki ketahanan yang tinggi terhadap serangan kriptanalisis klasik. Analisis frekuensi terhadap ciphertext menghasilkan distribusi karakter yang acak dan merata tanpa dominasi nilai tertentu, mengindikasikan bahwa struktur statistik plaintext berhasil disamarkan secara efektif. Simulasi serangan brute force dengan 1000 kombinasi acak matriks permutasi dan kunci substitusi menunjukkan tingkat kegagalan 100%, membuktikan bahwa ruang kunci yang sangat besar (mencapai $4{,}78 \times 10^{10}$ kombinasi bahkan dengan asumsi konservatif) menjadikan pendekatan exhaustive search tidak praktis secara komputasional. Hal ini mengonfirmasi bahwa sistem substitusi-permutasi berbasis aljabar max-plus mampu memberikan tingkat keamanan yang memadai untuk melindungi informasi dari upaya dekripsi tanpa kunci yang sah.

Dari perspektif efisiensi komputasi, sistem menunjukkan performa yang baik dengan waktu eksekusi yang meningkat secara linier terhadap panjang plaintext dan penggunaan memori yang stabil. Kompleksitas waktu yang linier mengindikasikan skalabilitas algoritma yang baik untuk menangani data berukuran bervariasi, sementara penggunaan memori yang konsisten (sekitar 59,28-59,43 MiB) menunjukkan efisiensi dalam pengelolaan sumber daya. Karakteristik ini menjadikan sistem berpotensi untuk diimplementasikan pada perangkat dengan keterbatasan sumber daya seperti sistem tertanam atau perangkat Internet of Things.

Berdasarkan temuan penelitian ini, beberapa pengembangan lebih lanjut dapat dilakukan untuk meningkatkan kapabilitas sistem. Pertama, eksplorasi ukuran matriks yang lebih besar (misalnya 4×4 atau 5×5) dapat meningkatkan kompleksitas dan ruang kunci sistem secara signifikan. Kedua, integrasi sistem max-plus dengan teknik kriptografi modern lainnya seperti chaos-based cryptography atau elliptic curve cryptography dapat menghasilkan skema hybrid yang lebih tangguh. Ketiga, pengujian pada berbagai jenis data (gambar, audio, atau video) perlu dilakukan untuk mengevaluasi generalisasi sistem terhadap format data yang berbeda. Keempat, analisis keamanan lanjutan menggunakan metode differential cryptanalysis dan linier cryptanalysis perlu dilakukan untuk mengidentifikasi potensi kerentanan yang lebih spesifik. Penelitian ini memberikan kontribusi awal yang menjanjikan dalam pengembangan kriptografi

berbasis aljabar max-plus, dan diharapkan dapat menjadi dasar bagi penelitian selanjutnya dalam ranah keamanan informasi dengan pendekatan matematika non-konvensional.

UCAPAN TERIMAKASIH

Ucapan terima kasih penulis sampaikan kepada Bapak Nurwan, S.Pd., M.Si., Ibu Armayani Arsal, S.Si., M.Si., dan Bapak Nisky Imansyah Yahya, S.Pd., M.Si. selaku dosen pembimbing sekaligus rekan penulis atas bimbingan, arahan, dan masukan ilmiah yang sangat berharga dalam penyusunan artikel ini. Penulis berterima kasih kepada rekan-rekan sejawat yang telah memberikan saran, masukan, dan bantuan teknis terkait implementasi algoritma serta analisis data. Apresiasi yang tulus juga disampaikan kepada keluarga dan semua pihak yang tidak dapat disebutkan satu per satu atas dukungan moral dan doa yang senantiasa diberikan sehingga penelitian dan penulisan artikel ini dapat terselesaikan dengan baik.

DAFTAR PUSTAKA

Alagic, G., Apon, D., Cooper, D., Dang, Q., Dang, T., Kelsey, J., Lichtinger, J., Liu, Y.-K., Miller, C., Moody, D., Peralta, R., Perlner, R., Robinson, A., & Smith-Tone, D. (2022). *Status report on the third round of the NIST Post-Quantum Cryptography Standardization process*. https://doi.org/10.6028/NIST.IR.8413

Amalia, R. (2023). *Ekspansi CSR dalam Aljabar Max-Plus untuk Menentukan Keperiodikan Pangkat Matriks Irreducible*. https://eprints.walisongo.ac.id/id/eprint/23425/1/1908046051_RIZKY%20AMALIA_LENGKAP%20TA%20-%20rizky%20amalia.pdf

Banu, Y., Rath, B. K., & Gountia, D. (2025). Analyzing cryptographic algorithm efficiency with in graph-based encryption models. *Frontiers in Computer Science*, *7*. https://doi.org/10.3389/fcomp.2025.1630222

Castro, O., Bruneau, P., Sottet, J.-S., & Torregrossa, D. (2024). Landscape of High-Performance Python to Develop Data Science and Machine Learning Applications. *ACM Computing Surveys*, *56*(3), 1–30. https://doi.org/10.1145/3617588

Chen, J., Grigoriev, D., & Shpilrain, V. (2023). *Tropical cryptography III: digital signatures*. https://eprint.iacr.org/2023/1475

ElGamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, *31*(4), 469–472.

Humaira, & Rafiqa. (2019). Kriptanalisis dengan metode brute force pada graphics processing unit. *Jurnal Nasional Teknik Elektro*, *8*(2), 150–158. https://openlibrary.telkomuniversity.ac.id/pustaka/files/105457/jurnal_eproc/kriptanalisis-dengan-metode-brute-force-pada-graphics-processing-unit.pdf

Iwan Setiadi, Santi Widianti, & I Putu Prachanda Kayuan. (2025). Implementasi Kriptografi Pengamanan Data Soal Ujian di Lingkungan Perguruan Tinggi Menggunakan Algoritma

AES-256 dan SHA-256. *Jurnal Penelitian Rumpun Ilmu Teknik*, *4*(3), 65–90. https://doi.org/10.55606/juprit.v3i4.4569

Katz, J., & Lindell, Y. (2020). *Introduction to Modern Cryptography* (3rd ed.). CRC Press. https://yehudalindell.com/introduction-to-modern-cryptography-3rd-edition/

Kavut, S., & Tutdere, S. (2020). Highly nonlinear (vectorial) Boolean functions that are symmetric under some permutations. *Advances in Mathematics of Communications*, *14*(1), 127–136. https://doi.org/10.3934/amc.2020010

Kurniawan, A., & Suparwanto, A. (2020). Max-Plus Linear Equation System and Its Application on Railway Network System. *Jurnal Matematika Thales*, *2*(1). https://doi.org/10.22146/jmt.55316

Li, J., Wei, G., Liang, J., Ren, Y., Lee, P. P. C., & Zhang, X. (2022). Revisiting Frequency Analysis against Encrypted Deduplication via Statistical Distribution. *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, 290–299. https://doi.org/10.1109/INFOCOM48880.2022.9796897

Muanalifah, A., Isnawati, A. R., Artes Jr., R., & Nurwan, N. (2023). The tropical version of El Gamal Encryption. *Journal of Natural Sciences and Mathematics Research*, *9*(2), 151–157. https://doi.org/10.21580/jnsmr.v9i2.18704

Ridho, A., & Romli, M. A. (2024). SISTEM PENGAMANAN DOKUMEN MENGGUNAKAN ALGORITMA KRIPTOGRAFI ADVANCED ENCRYPTION STANDARD (AES-256). *Jurnal Informatika Teknologi Dan Sains (Jinteks)*, *6*(4), 1044–1052. https://doi.org/10.51401/jinteks.v6i4.4887

Ruys, W., Lee, H., You, B., Talati, S., Park, J., Almgren-Bell, J., Yan, Y., Fernando, M., Erez, M., Gligoric, M., Burtscher, M., Rossbach, C. J., Pingali, K., & Biros, G. (2025). Performance Characterization of Python Runtimes for Multi-device Task Parallel Programming. *International Journal of Parallel Programming*, *53*(2), 16. https://doi.org/10.1007/s10766-025-00788-1

Shannon, C. E. (1949). Communication Theory of Secrecy Systems*. *Bell System Technical Journal*, *28*(4), 656–715. https://doi.org/10.1002/j.1538-7305.1949.tb00928.x

Social, W. A., & Meltwater. (2024). *Digital 2024 Global Overview Report*. https://wearesocial.com/wp-content/uploads/2024/02/Digital_2024_Global.pdf

Stallings, W. (2017). *Cryptography and Network Security: Principles and Practice* (7th ed.). Pearson Education.

Stinson, D. R., & Paterson, M. B. (2018). *Cryptography: Theory and Practice* (4th ed.). Chapman and Hall/CRC. https://cs.uwaterloo.ca/~dstinson/CTAP4.html