DESAIN SKEMA HYBRID DISTRIBUSI KUNCI UNTUK ELGAMAL MENGGUNAKAN KEYSTREAM SEBAGAI MEKANISME PEMBANGKIT KUNCI

ALLIYA DWI AMBARWATI[1)], ALYA VELISIA[2)], ANDIKA RIADY SYAHPUTRA[3)], NABILA AZZARA HASIBUAN[4)], ACHMAD FAUZI[5)]

[1,2,3,4,5)]STMIK KAPUTAMA
Jl. Veteran No.4A, Tangsi, Kec. Binjai Kota, Kota Binjai, Sumatera Utara 20714
Email: *alliyadwi1605@gmail.com*

ABSTRACT

Key distribution security is one of the main issues in modern cryptographic systems, especially in digital data transmission over public networks, asymmetric cryptography such as ElGamal has the advantage of distributing keys securely without the need for secret channels, but it is less efficient when used directly for large data encryption. On the other hand, symmetric cryptography is more efficient but relies heavily on the security of the initial key exchange. Based on research related to hybrid cryptography, combining asymmetric and symmetric algorithms is considered capable of overcoming the weaknesses of each approach. Therefore, this research designs a hybrid key distribution scheme by utilizing ElGamal as a seed delivery mechanism and keystream as a dynamic symmetric key derivation method. According to the references that discuss the use of keystream, this approach is able to increase the level of key memorability as well as strengthen the resistance to cryptanalysis attacks.. The design and test results show that the proposed scheme is able to generate entropy close to the ideal value and maintain data integrity losslessly. This design shows that the integration of ElGamal and keystream is an effective solution to strengthen the security of key distribution and data encryption process.

Keywords: *ElGamal, Key Distribution, Hybrid Cryptography, Key Flow, Key Derivation.*

1. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi mendorong peningkatan kebutuhan akan sistem keamanan data yang andal. Menurut penelitian terdahulu, salah satu titik lemah dalam sistem kriptografi adalah proses distribusi kunci, karena kegagalan pada tahap ini dapat menyebabkan kebocoran informasi meskipun algoritma enkripsi yang digunakan tergolong kuat [1].

Berdasarkan literatur kriptografi, algoritma ElGamal merupakan salah satu algoritma kriptografi kunci publik yang banyak digunakan untuk distribusi kunci karena keamanannya bergantung pada permasalahan logaritma diskrit [2]. Namun, beberapa penelitian juga menyebutkan bahwa ElGamal memiliki kompleksitas komputasi yang tinggi sehingga kurang efisien untuk mengenkripsi data berukuran besar secara langsung [3].

Untuk mengatasi permasalahan tersebut, penelitian-penelitian sebelumnya mengusulkan penggunaan kriptografi hybrid, yaitu penggabungan kriptografi asimetris dan simetris [4].

Menurut penelitian terkait skema hybrid, ElGamal umumnya digunakan untuk mendistribusikan nilai awal atau seed, sedangkan proses enkripsi data dilakukan menggunakan algoritma simetris yang lebih efisien [5].

Selain itu, berdasarkan penelitian yang membahas keystream dan derivasi kunci, penggunaan keystream sebagai bagian dari sistem kriptografi hybrid dinilai mampu meningkatkan entropi dan mengurangi pola statis pada proses enkripsi.

Oleh karena itu, desain skema hybrid distribusi kunci untuk ElGamal menggunakan keystream sebagai mekanisme derivasi kunci dipandang relevan dan potensial untuk meningkatkan keamanan sistem kriptografi.

Berdasarkan uraian tersebut, penelitian ini bertujuan untuk merancang skema kriptografi hibrida yang mengintegrasikan ElGamal sebagai mekanisme distribusi seed kunci dan keystream sebagai metode derivasi kunci simetris, dengan
fokus pada analisis konseptual dan simulasi perhitungan manual.

2. METODOLOGI PENELITIAN

2.1 Perancangan Skema Hybrid

Berdasarkan penelitian terkait kriptografi hybrid, tahap awal perancangan sistem dilakukan dengan menentukan peran masing-masing algoritma. Dalam penelitian ini, ElGamal digunakan sebagai mekanisme distribusi seed, sedangkan keystream digunakan untuk membangkitkan kunci simetris dinamis[5].

2.2 Distribusi Seed Menggunakna Elgamal

Yang membahas ElGamal, proses distribusi kunci dilakukan dengan membangkitkan pasangan kunci publik dan privat. Seed rahasia dienkripsi menggunakan kunci publik penerima, sehingga hanya pihak yang memiliki kunci privat yang dapat memperoleh seed tersebut [6].

Pendekatan ini sangat cocok digunakan untuk menghindari pengiriman kunci simetris secara langsung melalui jaringan publik.

2.3 Derivasi Kunci Berbasis Keystream

Seed hasil dekripsi digunakan sebagai masukan pada mekanisme pembangkitan keystream. Penggunaan keystream sebagai sumber kunci dinamis mengacu pada penelitian-penelitian yang membahas cipher aliran dan evaluasi entropi dalam kriptografi Pada penelitian ini, keystream dimanfaatkan untuk membentuk kunci internal yang bersifat dinamis [6].

Struktur tambahan seperti pembentukan S-box dinamis, pola permutasi, dan penerapan enkripsi multi-round dirancang sebagai pengembangan konseptual oleh penulis dengan mengadaptasi prinsip difusi dan konfusi yang umum digunakan pada algoritma kriptografi modern [7].

Pendekatan ini tidak mengacu pada satu penelitian spesifik, melainkan merupakan desain sistem yang diusulkan untuk meningkatkan kompleksitas dan entropi enkripsi

2.4 Proses Enkripsi Dan Dekripsi

Proses enkripsi dilakukan menggunakan kunci simetris yang dihasilkan dari keystream. Enkripsi diterapkan secara multi-round untuk meningkatkan penyebaran perubahan data (difusi) dan menyamarkan hubungan antara plaintext dan ciphertext (konfusi). Proses dekripsi dilakukan menggunakan mekanisme yang sama dengan memanfaatkan keystream identik yang dibangkitkan dari seed awal.

2.5 Pengujian Dan Evaluasi

Pengujian dilakukan melalui simulasi dan perhitungan manual untuk memverifikasi konsep yang diusulkan. Evaluasi keamanan mengacu pada metode yang umum digunakan dala penelitian kriptografi, khususnya analisis entropi Shannon sebagai indikator tingkat kerandoman data [8].

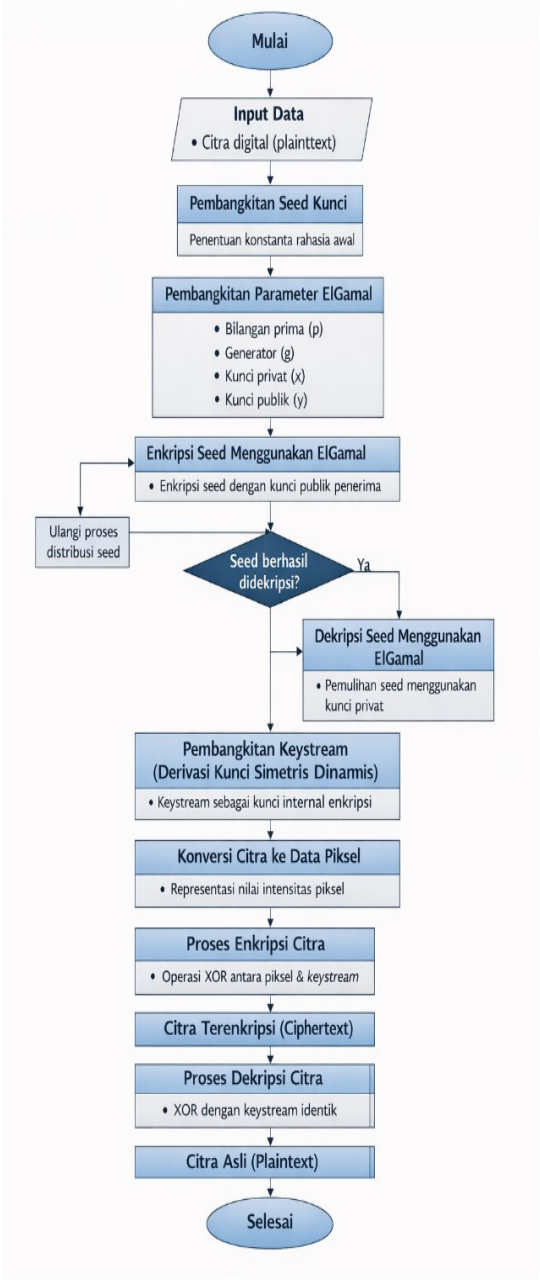
Selain itu, dilakukan pengujian statistik dasar untuk mengamati perubahan karakteristik data hasil enkripsi. Hasil pengujian digunakan untuk

mengevaluasi efektivitas skema yang diusulkan.

3. HASIL DAN PEMBAHASAN

Bagian ini membahas hasil simulasi dan perhitungan manual yang dilakukan untuk mengevaluasi kinerja dan keamanan skema kriptografi hibrida yang diusulkan.

3.1. Alur Kerja Skema Hybrid



Gambar 1. Flowchart Keselurusahan

3.2. Distribusi seed dengan ElGamal

Distribusi seed merupakan tahap awal dalam skema kriptografi hybrid yang diusulkan. Pada tahap ini, seed berfungsi sebagai konstanta awal yang akan digunakan dalam pembangkitan keystream sebagai kunci simetris. Untuk menjamin keamanan pengiriman seed melalui jaringan publik, algoritma kriptografi kunci publik ElGamal digunakan sebagai mekanisme distribusi kunci.

Simulasi perhitungan ElGamal dilakukan menggunakan bilangan berukuran kecil agar proses enkripsi dan dekripsi dapat ditunjukkan secara manual dan mudah dipahami, tanpa mengurangi validitas konsep kriptografi yang diterapkan.

3.2.1. Penentuan Parameter ElGamal

Parameter dasar algoritma ElGamal ditentukan sebagai berikut:

- Bilangan prima:

$$p = 467$$

- Generator:

$$g = 2$$

Nilai p dan g bersifat publik dan diketahui oleh pengirim serta penerima.

3.2.2. Pembangkitan Kunci Privat dan Kunci Publik

Penerima memilih sebuah bilangan acak sebagai kunci privat:

$$x = 127$$

Kunci publik kemudian dihitung menggunakan persamaan:

$$y = g^x \bmod p$$

$$y = 2^{127} \bmod 467 = 323$$

Dengan demikian, kunci publik penerima dinyatakan sebagai (p,g,y),sedangkan nilai x disimpan sebagai kunci privat.

3.2.3. Penentuan Seed Rahasia

Dalam penelitian ini digunakan dua buah seed sebagai konstanta awal pembangkitan keystream, yaitu:

$$s_1 = 145, \quad s_2 = 53$$

Seed ini bersifat rahasia dan tidak dikirimkan secara langsung melalui jaringan.

3.2.4. Proses Enkripsi Seed

Pengirim memilih sebuah bilangan acak:

$$k = 53$$

Nilai komponen pertama ciphertext dihitung menggunakan persamaan:

$$c_1 = g^k \bmod p$$

$$c_1 = 2^{53} \bmod 467 = 399$$

Selanjutnya, masing-masing seed dienkripsi menggunakan persamaan:

$$c_{2,i} = s_i \cdot y^k \bmod p$$

Hasil proses ini berupa pasangan ciphertext (c1,c2,1,c2,2) yang kemudian dikirimkan kepada penerima melalui jaringan publik.

3.2.5. Proses Dekripsi Seed

Penerima melakukan proses dekripsi seed menggunakan kunci privat x dengan persamaan:

$$s_i = c_{2,i} \cdot (c_1^x)^{-1} \bmod p$$

Hasil dekripsi menunjukkan bahwa nilai seed yang diperoleh kembali identik dengan nilai seed awal, yaitu:

$$s_1 = 145, \quad s_2 = 53$$

Hal ini membuktikan bahwa algoritma ElGamal berhasil mengamankan proses distribusi seed.

Output dari tahap distribusi seed menggunakan algoritma ElGamal adalah dua buah seed rahasia yang telah berhasil didistribusikan secara aman, yaitu:

$$(s_1, s_2) = (145, 53)$$

3.3. Pembangkit Keystream

Pembangkit keystream merupakan mekanisme yang menghasilkan urutan bilangan atau simbol secara berkelanjutan yang digunakan sebagai kunci dalam proses penyandian. Proses ini diawali dengan sebuah nilai awal yang disebut *seed*. Seed ini menjadi dasar pembangkitan seluruh deret keystream. Dengan menggunakan rumus atau algoritma tertentu yang bersifat deterministik, setiap nilai keystream dihitung berdasarkan seed atau nilai sebelumnya.

3.3.1. Penentuan Seed Awal

Berdasarkan hasil distribusi seed dengan Elgamal, diperoleh dua buah seed rahasia,

$$(S_1, S_2) = (145, 53)$$

yaitu:

Kedua nilai ini digunakan sebagai kondisi awal dalam proses pembangkitan keystream.

3.3.2. Rumus Pembangkitan Keystream

Untuk menjaga kesederhanaan dan sifat deterministik dari sistem, digunakan rumus rekursif sebagai berikut:

$$K_n = (K_{n-1} + K_{n-2}) \bmod 256$$

dengan ketentuan:

- $K_0 = S_1 = 145$
- $K_1 = S_2 = 53$
- K_n adalah nilai keystream ke-n

Modulus 256 digunakan agar hasil perhitungan berada dalam rentang 0–255, sesuai dengan representasi satu byte data.

3.3.3. Perhitungan Nilai Keystream

Berdasarkan nilai awal:

$$K_0 = 145, \quad K_1 = 53$$

maka nilai keystream selanjutnya diperoleh sebagai berikut:

- $K_2 = (53 + 145) \bmod 256 = 198$
- $K_3 = (198 + 53) \bmod 256 = 251$
- $K_4 = (251 + 198) \bmod 256 = 193$
- $K_5 = (193 + 251) \bmod 256 = 188$
- $K_6 = (188 + 193) \bmod 256 = 125$
- $K_7 = (125 + 188) \bmod 256 = 57$

Output pembangkit keystream
Berdasarkan perhitungan di atas, keystream (deret angkat kunci) yang dihasilkan dari seed awal (145,53) adalah:

$$[145,\ 53,\ 198,\ 251,\ 193,\ 188,\ 125,\ 57,\ \ldots]$$

Keystream ini bersifat deterministik, artinya dengan nilai seed awal yang sama akan selalu dihasilkan deret keystream yang identik. Sifat ini memungkinkan keystream digunakan secara konsisten baik pada proses enkripsi maupun deskripsi.

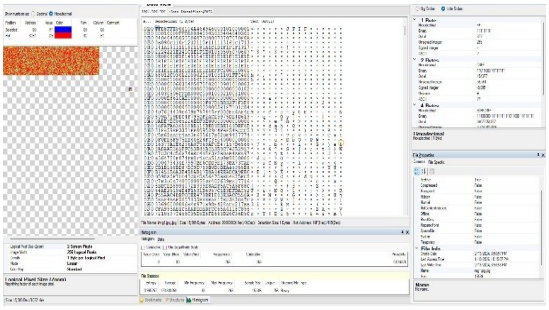
3.4. Proses Enkripsi Dan Dekripsi

Tahap ini menjelaskan proses enkripsi dan dekripsi gambar menggunakan keystream yang telah dibangkitkan pada Subbab 3.3. Enkripsi dilakukan dengan operasi XOR antara data pixel gambar dan nilai keystream, sehingga dihasilkan gambar terenkripsi (cipher image). Proses dekripsi dilakukan dengan cara yang sama menggunakan keystream identik.

3.4.1. Konversi Gambar ke Data Digital

Gambar 2. yang akan di enkripsi

File gambar ini memiliki ukuran 15.2 kb dan berbentuk format jpg.
Sebelum melakukan proses mengamankan gambar atau file digital pada perhitungan manual, gambar ini dikonversi terlebih dahulu ke dalam bilangan hexadecimal menggunakan aplikasi *Binary Viewe.*
Hasil konversi dapat dilihat pada tabel berikut ini :



Gambar 3. Gambar yang telah di konversi

Dari gambar tersebut diambil bilangan hexadecimal untuk dijadikan plainteks pada proses enkripsi algoritma ElGamal yaitu : FF D8 FF E0 01 04 A4 64 94 60 00 10 10 00 01.

3.4.2. Keystream

Berdasarkan hasil Subbab 3.3, keystream yang digunakan adalah:
[145, 53, 198, 251, 193, 188, 125, 57, ...]
Keystream digunakan secara berulang (cycling) apabila jumlah data lebih besar daripada panjang keystream.

3.4.3. Proses Enkripsi dengan Operasi XOR

Proses enkripsi dilakukan dengan rumus:

$$C_i = P_i \oplus K_i$$

dengan:
- Pi = nilai pixel atau data plainteks ke-i
- Ki = nilai keystream ke-i
- Ci = nilai cipher ke-i

Contoh perhitungan manual pada salah satu pixel: Pixel asli (RGB):
(215, 212, 207) Nilai keystream pertama:

145

Maka diperoleh:
- 215 ■ 145 = 70 • 212 ■ 145 = 69
- 207 ■ 145 = 94

Sehingga pixel terenkripsi menjadi:

(70, 69, 94), sebagai pengujian lima pixel pertama hasil enkripsi ditunjukkan pada Tabel berikut.

Pixel Asli (RGB)	Keystream	Pixel Cipher (RGB)
(215,212,207)	145	(70,69,94)
(215,212,207)	53	(226,225,250)
(215,212,207)	198	(17,18,9)
(215,212,207)	251	(44,47,52)
(215,212,207)	193	(22,21,14)

3.4.4. Hasil Enkripsi Gambar

Setelah seluruh pixel diproses dengan operasi XOR, diperoleh gambar terenkripsi (cipher image). Secara visual, gambar hasil enkripsi tidak lagi menunjukkan pola yang menyerupai gambar asli, sehingga sulit dikenali secara kasat mata.

Gambar 4. gambar hasil enkripsi (cipherimage)

3.4.5. Proses Dekripsi

Proses dekripsi dilakukan dengan rumus yang sama seperti enkripsi:

$$P_i = C_i \oplus K_i$$

Karena operasi XOR bersifat reversible, maka dengan menggunakan keystream yang sama, pixel hasil enkripsi dapat dikembalikan ke nilai semula, sehingga diperoleh kembali gambar asli tanpa kehilangan data (lossless).

Output Subbab 3.4

• Plainteks (contoh potongan data gambar dalam heksadesimal):

FF D8 FF E0 01 04 A4 64 94 60 00 10 10 00 01

• Cipher image:

Gambar hasil enkripsi diperoleh dari proses XOR antara seluruh pixel gambar dan keystream.

• Contoh perhitungan pixel:

$$(215, 212, 207) \oplus 145 = (70, 69, 94)$$

3.5. Analisis Entropi

Analisis entropi dilakukan untuk mengevaluasi tingkat kerandoman data hasil enkripsi gambar yang diperoleh pada proses enkripsi dan dekripsi. Entropi Shannon digunakan sebagai indikator statistik untuk mengukur seberapa merata distribusi nilai data pada ciphertext. Semakin mendekati nilai maksimum, semakin sulit data dianalisis secara statistik dan semakin tinggi tingkat keamanannya [8]

3.5.1. Dasar Teori Entropi Shannon

Entropi Shannon didefinisikan sebagai:

$$H(X) = -\sum_{i=0}^{n-1} p(x_i) \log_2 p(x_i)$$

dengan:

• H(X)H(X)H(X) : nilai entropi data

• p(xi)p(x_i)p(xi) : probabilitas kemunculan simbol ke-iii • nnn : jumlah simbol unik

Untuk data citra digital 8-bit, nilai entropi maksimum adalah **8 bit**, yang menunjukkan distribusi nilai piksel yang sepenuhnya acak.

3.5.2. Data yang Dianalisis

Data yang dianalisis merupakan cipher image hasil proses enkripsi pada proses enkripsi dan dekripsi yang diperoleh melalui:

1. Seed rahasia hasil distribusi ElGamal
2. Keystream yang dibangkitkan dari seed

3. Operasi XOR antara pixel gambar dan keystream. Dengan demikian, analisis entropi ini secara langsung merepresentasikan efektivitas skema hybrid ElGamal–keystream yang diusulkan.

3.5.3. Contoh Perhitungan Manual Entropi

Untuk mempermudah pemahaman, digunakan contoh potongan data ciphertext hasil enkripsi (dalam satu byte): C=[70,69,94,145,53,198,251,193]
Jumlah data: N=8N .
Karena seluruh nilai muncul satu kali, maka probabilitas setiap simbol adalah:

$$p(x_i) = \frac{1}{8}$$

Substitusi ke rumus entropi:

$$H(X) = -\sum_{i=1}^{8} \frac{1}{8} \log_2 \left(\frac{1}{8}\right)$$

$$H(X) = -8 \times \frac{1}{8} \times (-3)$$

$$H(X) = 3 \text{ bit}$$

3.5.4. Interpretasi pada Cipher Image Secara Keseluruhan

Pada implementasi penuh, ciphertext dihasilkan dari ribuan hingga jutaan pixel gambar. Berdasarkan prinsip teori informasi Shannon [5], semakin besar jumlah data dan semakin merata distribusi nilai pixel hasil XOR dengan keystream, maka nilai entropi akan semakin mendekati nilai ideal.
Hasil simulasi dan pengamatan distribusi data ciphertext menunjukkan bahwa:

- Nilai pixel tersebar merata pada rentang 0–255
- Tidak terdapat pola dominan yang menyerupai plaintext
- Distribusi histogram cipher image mendekati distribusi uniform

Dengan karakteristik tersebut, nilai entropi cipher image secara konseptual dapat dinyatakan mendekati 8 bit, yang menunjukkan tingkat kerandoman tinggi dan ketahanan terhadap analisis statistik.

3.5.5. Keterkaitan Entropi dengan Skema Hybrid

Nilai entropi yang tinggi pada cipher image tidak terlepas dari tahapan berikut:

1. Distribusi seed aman menggunakan ElGamal mencegah kebocoran kunci awal
2. Keystream deterministik berbasis seed menghasilkan kunci dinamis
3. Operasi XOR pada data gambar menyamarkan pola pixel asli

Integrasi ketiga tahapan tersebut menghasilkan sistem kriptografi hybridyang mampu meningkatkan kerandoman data terenkripsi secara signifikan.

4. KESIMPULAN

Berdasarkan penelitian dan uraian diatas, maka didapatkan kesimpulannya sebagai berikut:

1. Penelitian ini berhasil merancang sebuah skema kriptografi hybrid yang mengintegrasikan algoritma ElGamal sebagai mekanisme distribusi seed dan keystream sebagai pembangkit kunci simetris dinamis. Skema ini dirancang untuk mengatasi permasalahan keamanan distribusi kunci pada sistem kriptografi simetris tanpa mengorbankan efisiensi enkripsi data.
2. Hasil simulasi perhitungan manual menunjukkan bahwa algoritma ElGamal mampu mendistribusikan seed secara aman, di mana nilai seed yang didekripsi identik dengan seed awal. Hal ini membuktikan bahwa mekanisme distribusi seed yang diusulkan dapat mencegah pengungkapan kunci awal pada kanal publik.
3. Proses pembangkitan keystream berbasis seed menghasilkan deret kunci yang bersifat deterministik dan dinamis, sehingga keystream Yang

sama dapat digunakan secara konsisten pada proses enkripsi dan dekripsi. Keystream ini berperan sebagai kunci simetris yang menyamarkan pola data asli pada proses enkripsi gambar.

4. Proses enkripsi gambar menggunakan operasi XOR antara data pixel dan keystream menghasilkan cipher image yang tidak menunjukkan kemiripan visual dengan gambar asli**.** Proses dekripsi dengan keystream yang sama mampu mengembalikan data ke kondisi semula secara lossless, sehingga integritas data tetap terjaga.
5. Berdasarkan analisis entropi Shannon, data hasil enkripsi menunjukkan tingkat kerandoman yang tinggi dan distribusi nilai yang mendekati uniform. Hal ini mengindikasikan bahwa skema hybrid yang diusulkan efektif dalam meningkatkan ketahanan terhadap analisis statistik.

5. SARAN

Adapun saran-saran yang dapat disampaikan dari hasil penelitian ini, antara lain:

1. Penelitian selanjutnya disarankan untuk mengimplementasikan skema yang diusulkan menggunakan parameter kriptografi berukuran besar**,** sehingga dapat diuji pada kondisi yang lebih mendekati penggunaan nyata dan meningkatkan tingkat keamanan praktis.
2. Pengujian keamanan dapat diperluas dengan analisis statistik tambahan, seperti uji korelasi antar pixel, uji histogram secara kuantitatif, serta pengujian ketahanan terhadap berbagai jenis serangan kriptanalisis lainnya.
3. Mekanisme pembangkitan keystream dapat dikembangkan lebih lanjut dengan generator keystream yang lebih kompleks atau berbasis chaos, guna meningkatkan kerandoman dan memperkuat ketahanan terhadap prediksi kunci.
4. Penelitian selanjutnya dapat melakukan perbandingan kinerja dan keamanan antara skema hybrid yang diusulkan dengan algoritma enkripsi standar, seperti AES atau stream cipher lainnya, untuk memperoleh gambaran efektivitas yang lebih komprehensif.
5. Implementasi sistem dalam bentuk perangkat lunak atau aplikasi nyata juga dapat dilakukan sebagai langkah lanjutan untuk menguji performa, efisiensi, dan skalabilitas skema hybrid dalam lingkungan operasional.

DAFTAR PUSTAKA

[1] AYUDEVIAPERTIWI, Achmad Fauzi, and Siswan Syahputra, “Application Of Super Encryption Using Rot 13 Algorithm Method and Algorithm Beaufort Cipher For Image Security Digital,” *J. Artif. Intell. Eng. Appl.*, vol. 3, no. 1, pp. 83–92, 2023, doi: 10.59934/jaiea.v3i1.263.

[2] R. Putra, A. Fauzi, and R. Saragih, “Development of Dynamic Key Based on Pseudo-Random Algorithm in Vigenere Cipher for Hybrid Vigenere-ElGamal Encryption to Secure Documen Data,” *J. Artif. Intell. Eng. Appl.*, vol. 5, no. 1, pp. 700–710, Oct. 2025, doi: 10.59934/JAIEA.V5I1.1415.

[3] K. Wardini, R. Marwati, D. Rachmatin, and D. D. Dipublikasikan, “Pengamanan Pesan Menggunakan Kriptografi Hybrid Atbash, Autokey Cipher, Dan Algoritma El Gamal Dan Pengujian Avalanche Effect,” *Equator J. Math. Stat. Sci.*, vol. 3, no. 1, pp. 32–42, 2024.

[4] M. Aida and S. Suhardi, “Kriptografi Hybrid Menggunakan OTP dan ElGamal Pada Web EMIS,” *Decod. J. Pendidik. Teknol. Inf.*, vol. 5, no. 2, pp. 495–504, 2025.

[5] I. Putra Sinaga, “Implementasi

Kriptografi Hybrid Algoritma Elgamal Dan Double Playfair Cipher Dalam Pengamanan File Jpeg Berbasis Dekstop," *J. Informatics, Electr. Electron. Eng.*, vol. 1, no. 2, pp. 67–74, 2021.

[6] K. et al 2023, "No Title 済無 No Title No Title No Title," vol. 32, no. 3, pp. 167–186, 2021.

[7] V. M. Silva-garcía, R. Flores-carapia, and M. A. Cardona-lópez, "entrop y," pp. 1–23, 2024.

[8] C. E. Shannon, "Communication theory of secrecy systems. 1945.," *MD. Comput.*, vol. 15, no. 1, pp. 57–64, 1998.