BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 Kriptografi

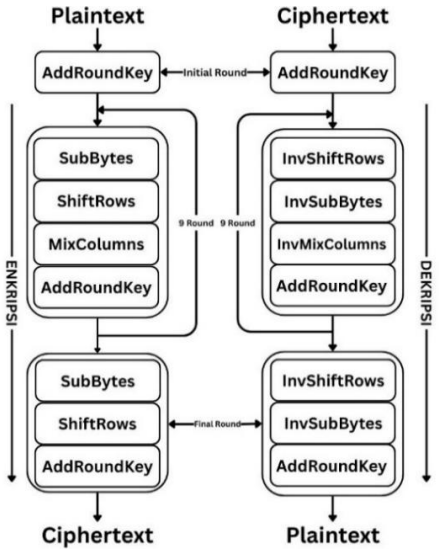
Kriptografi adalah ilmu dan seni mengamankan informasi dengan cara mengenkripsi data sehingga hanya pihak yang berwenang dengan kunci rahasia yang dapat mengembalikan plaintext. Tujuan utamanya adalah kerahasiaan (*confidentiality*), integritas, dan otentikasi data (Mohammad Shah dkk., 2023).

Pada kriptografi simetris, pengirim dan penerima memakai kunci yang sama untuk enkripsi dan dekripsi; sedangkan pada kriptografi asimetris, digunakan sepasang kunci (publik/privat). *Plaintext* (teks asli) diproses oleh algoritma enkripsi menjadi *ciphertext* (teks tersandi) dengan bantuan kunci enkripsi. Tujuan akhir adalah memastikan bahwa tanpa kunci yang benar, informasi tidak dapat dipahami. Shannon memperkenalkan konsep *perfect secrecy*, yaitu kondisi ideal di mana *ciphertext* tidak mengurangi ketidakpastian atas *plaintext* (dengan kata lain (P(M|C)=P(M))) (Zolfaghari dkk., 2022), untuk mendekati keamanan semacam ini, sistem kriptografi modern menerapkan fungsi-fungsi matematika kompleks dan penggunaan kunci yang cukup panjang.

2.1.2 *Advanced Encryption Standard* (AES-128)

Advanced Encryption Standard (AES) adalah algoritma enkripsi simetris berbasis *block cipher* yang ditetapkan sebagai standar oleh National Institute of Standards and Technology (NIST) pada tahun 2001 melalui publikasi FIPS 197.

AES mendukung tiga panjang kunci yaitu 128 bit, 192 bit, dan 256 bit, dengan ukuran blok yang tetap 128 bit. Pada varian AES-128, algoritma beroperasi menggunakan kunci sepanjang 128 bit dan menjalankan sepuluh *round* enkripsi. Data diorganisasikan dalam bentuk matriks state berukuran 4×4 byte, di mana setiap sel berisi satu byte (8 bit). Setiap *round* enkripsi kecuali *round* terakhir terdiri atas empat transformasi yang diaplikasikan secara berurutan (NIST, 2023). SubBytes menerapkan substitusi non-linear pada setiap byte menggunakan S-Box untuk menyediakan confusion. ShiftRows menggeser baris-baris dalam matriks state secara siklus untuk mencegah kolom diproses secara independen. *MixColumns* mengalikan setiap kolom state dengan matriks polinomial tetap dalam *Galois Field* GF(2^8), memberikan *diffusion* antar byte dalam satu kolom (Ganesh dkk., 2025) Namun transformasi ini merupakan komponen paling berat secara komputasi karena melibatkan operasi perkalian dalam *Galois Field* (Gamido dkk., 2018). Proses Enkripsi dan Dekripsi AES-128 dapat dilihat pada gambar 2.1.



Gambar 2.1 Diagram Proses AES

2.1.3 Block cipher dan Cipher Block Chaining

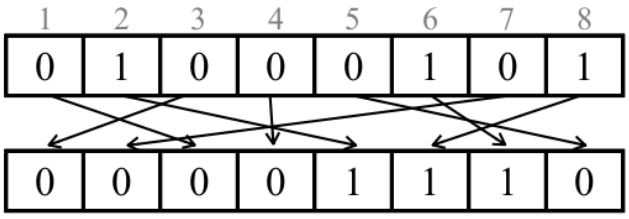
Block cipher adalah algoritma kriptografi simetris yang mengenkripsi data dalam blok berukuran tetap, misalnya 128 bit pada AES. Pada data yang lebih besar, blok-blok tersebut harus diproses dengan mode operasi tertentu agar hasil enkripsi tidak mereplikasi pola plaintext yang sama, seperti yang terjadi pada mode Electronic Codebook (ECB) yang kurang aman karena setiap blok plaintext identik menghasilkan *ciphertext* identik. Berdasarkan hal tersebut, penggunaan mode operasi seperti Cipher Block Chaining (CBC) sangat umum diterapkan dalam implementasi AES untuk meningkatkan kerahasiaan data dengan menggabungkan hasil enkripsi blok sebelumnya dalam proses enkripsi blok saat ini (Fahlevvi dkk., 2025).

Pada mode CBC, setiap blok plaintext pertama kali di-XOR dengan *ciphertext* blok sebelumnya sebelum dienkripsi, dan untuk blok pertama digunakan Initialization Vector (IV) yang acak agar hasilnya tetap berbeda meskipun plaintext yang dienkripsi sama. Dengan cara ini, *ciphertext* yang dihasilkan untuk setiap blok sangat bergantung pada seluruh rangkaian plaintext sebelumnya, sehingga pola plaintext tidak mudah terdeteksi dalam *ciphertext* dan difusi enkripsi secara praktis meningkat dibandingkan ECB.

2.1.4 *Bit permutation*

Bit permutation adalah operasi yang mengubah posisi bit-bit dalam sebuah byte tanpa mengubah nilai bit itu sendiri. Jika sebuah byte direpresentasikan sebagai urutan delapan bit, maka tabel permutasi BP mendefinisikan bahwa bit output ke-i diambil dari posisi input tertentu sesuai tabel tersebut. Bit permutasi

memiliki relevansi langsung dalam kriptografi karena merupakan implementasi dari properti diffusion, yaitu operasi permutasi menyebarkan pengaruh setiap bit input ke posisi berbeda di output, sehingga perubahan satu bit dapat berdampak pada banyak bit output setelah beberapa *round* (Gamido dkk., 2018).



Gambar 2.2. Proses *Bit permutation* (sumber: researchgate)

Gamido (2018) membuktikan bahwa bit permutasi dapat digunakan sebagai pengganti transformasi *MixColumns* dalam AES karena mampu memberikan diffusion yang memadai dengan beban komputasi yang jauh lebih rendah, mengingat operasinya hanya berupa pertukaran posisi tanpa melibatkan aritmetika *Galois Field*. Penelitian ini dikembangkan lebih lanjut oleh Baladhay dan De Los Reyes (2024) yang mengimplementasikan bit permutasi melalui *precomputed lookup table* berisi 256 entri, memungkinkan operasi dilakukan dalam waktu O(1) per byte menggunakan akses tabel langsung. Penelitian ini memodifikasi prinsip bit permutasi dengan membuat tabel permutasinya bersifat dinamis dan bergantung pada kunci enkripsi, mengikuti prinsip yang dikemukakan oleh Salih dkk. (2020) bahwa permutasi yang bergantung pada kunci dapat meningkatkan ketergantungan kunci dalam proses enkripsi serta menghasilkan perubahan pola permutasi yang dinamis, sehingga berpengaruh pada difusi dan kerumitan *ciphertext*.

2.1.5 *Logistic chaotic map*

Logistic map adalah sistem dinamik diskrit yang diformulasikan dengan persamaan rekursif sederhana:

$$x_{n+1} = r \times x_{\mathrm{n}} \times (1 - x_{\mathrm{n}}) \quad (1)$$

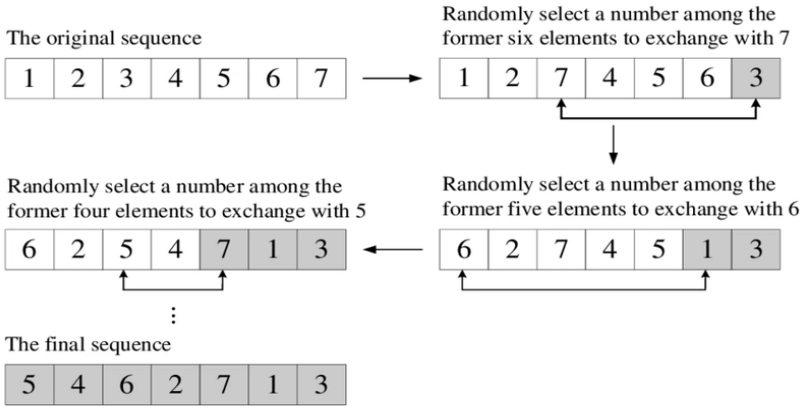
$x_n \in (0, 1)$ adalah nilai state pada iterasi ke-n dan r adalah parameter kontrol yang menentukan perilaku dinamis sistem. Pada nilai r > 3,57, sistem memasuki rezim chaotic di mana output tidak dapat diprediksi meskipun persamaannya sepenuhnya deterministik (Wang & Liu, 2022).

Logistic map telah banyak digunakan dalam kriptografi karena memiliki properti yang diperlukan untuk pembangkitan kunci yang aman. Abu-Faraj dkk menggunakan 2D *logistic map* sebagai pembangkit kunci dalam sistem enkripsi gambar dan membuktikan bahwa *logistic map* dengan parameter r mendekati 4,0 menghasilkan distribusi output yang seragam dan sensitivitas tinggi terhadap kondisi awal, dua properti esensial untuk pembangkitan kunci kriptografi, Perubahan sekecil 10^{-15} pada nilai awal x_0 menghasilkan urutan yang divergen total setelah beberapa iterasi, yang dikenal sebagai butterfly effect (Abu-Faraj dkk., 2023). Penelitian sebelumnya memanfaatkan hasil hash SHA-256 sebagai bagian dari proses pembentukan parameter awal pada sistem *Chaos* yang digunakan dalam skema enkripsi. Pendekatan tersebut memungkinkan setiap kunci menghasilkan kondisi awal yang berbeda sehingga urutan *Chaos* yang dibangkitkan juga berbeda (Fan & Zhang, 2024). Prinsip yang sama digunakan dalam penelitian ini, di mana hasil SHA-256 kunci enkripsi dinormalisasi menjadi nilai awal $x_0$untuk membangkitkan deret *Logistic chaotic map* yang digunakan pada proses

pembentukan tabel permutasi dinamis. Nilai r=3.99 digunakan pada penelitian ini karena berada pada wilayah *Chaos* Logistic Map sehingga mampu menghasilkan deret yang memiliki sensitivitas tinggi terhadap kondisi awal dan karakteristik pseudo-acak yang diperlukan dalam proses pembangkitan permutasi. Penggunaan nilai r=3.99 sebagai kondisi chaotic juga digunakan dalam berbagai penelitian enkripsi berbasis Logistic Map (Khaitan dkk., 2019).

2.1.6 *Fisher-Yates shuffle*

Fisher–Yates Shuffle merupakan algoritma permutasi yang digunakan untuk menghasilkan urutan acak dengan distribusi yang seragam melalui proses pertukaran elemen secara bertahap. Algoritma ini bekerja dengan memilih satu indeks acak pada setiap iterasi kemudian menukarkan elemen pada posisi tersebut dengan elemen yang sedang diproses hingga seluruh elemen selesai diacak. Karena setiap elemen hanya diproses satu kali, Fisher–Yates Shuffle memiliki kompleksitas waktu O(n) dan kompleksitas ruang O(1), sehingga banyak digunakan pada aplikasi yang membutuhkan pembangkitan permutasi secara efisien.



Gambar 2.3. Proses *Fisher-Yates shuffle* (sumber: springer.com)

Sumber keacakan pada Fisher–Yates tidak harus berasal dari pseudo-random number generator konvensional, tetapi dapat dibangkitkan menggunakan sistem *Chaos* dalam bidang kriptografi. Salah satunya adalah 2D Logistic ICMIC Cascade Map (2D-LICM), yang digunakan untuk menghasilkan sequence pada proses Fisher–Yates shuffling sehingga permutasi yang dihasilkan menjadi bergantung pada parameter *Chaos* dan kunci yang digunakan (Naim & Ali Pacha, 2023). Pendekatan tersebut menunjukkan bahwa *chaotic map* dapat dimanfaatkan sebagai sumber keacakan pada *Fisher–Yates Shuffle* untuk meningkatkan kompleksitas dan sensitivitas terhadap kunci dalam proses enkripsi.

2.1.7 Difusi *Ciphertext*

Difusi *ciphertext* merupakan salah satu prinsip fundamental dalam kriptografi yang bertujuan menyebarkan pengaruh perubahan kecil pada plaintext atau kunci ke seluruh bagian *ciphertext* sehingga pola asli data sulit dikenali. Pada algoritma block cipher modern, difusi dicapai melalui kombinasi operasi substitusi dan permutasi yang dilakukan secara berulang pada setiap *round* enkripsi. Semakin baik proses difusi yang dihasilkan, semakin sulit kriptanalis menemukan hubungan statistik antara plaintext dan *ciphertext*. Salah satu penelitian menjelaskan bahwa mekanisme difusi yang kuat mampu meningkatkan resistansi algoritma terhadap serangan statistik maupun diferensial karena perubahan kecil pada input akan menyebar ke banyak bit output *ciphertext* (Li dkk., 2024).

Kualitas difusi *ciphertext* umumnya dievaluasi menggunakan beberapa parameter pengujian, yaitu *Avalanche effect*, *Strict Avalanche Criterion* (SAC), entropi Shannon, dan *Key sensitivity*. *Avalanche effect* dan SAC digunakan untuk

mengukur sejauh mana perubahan satu bit pada plaintext atau kunci dapat mengubah sekitar 50% bit pada *ciphertext*. Selain itu, entropi Shannon juga digunakan sebagai indikator kualitas difusi karena menunjukkan tingkat keacakan distribusi byte pada *ciphertext*. Abu-Faraj dkk. (2023) menyatakan bahwa *ciphertext* dengan nilai entropi mendekati 8 bit menunjukkan distribusi data yang semakin acak sehingga pola statistik plaintext berhasil disamarkan dengan baik. Sementara itu, *Key sensitivity* digunakan untuk mengukur sensitivitas algoritma terhadap perubahan kecil pada kunci enkripsi. Elghandour dkk. (2022) menjelaskan bahwa perubahan satu bit pada kunci yang menghasilkan *ciphertext* berbeda secara signifikan menunjukkan bahwa proses difusi dan ketergantungan terhadap kunci berjalan dengan baik. Oleh karena itu, penelitian ini menggunakan *Avalanche effect*, entropi Shannon, dan *Key sensitivity* untuk mengevaluasi kualitas difusi *ciphertext* pada algoritma AES-128 termodifikasi yang diusulkan.

2.2 Parameter Pengujian

2.2.1 Kecepatan Enkripsi

Kecepatan enkripsi merupakan parameter yang digunakan untuk mengukur waktu yang dibutuhkan algoritma dalam mengubah plaintext menjadi *ciphertext*. Dalam penelitian ini, kecepatan enkripsi menjadi parameter utama karena modifikasi AES-128 yang diusulkan memang diarahkan untuk meningkatkan efisiensi proses enkripsi melalui pengurangan jumlah *round* dan penggantian fungsi *MixColumns* dengan *dynamic bit permutation*. AES sendiri merupakan block cipher yang memproses data dalam blok 128 bit, sehingga setiap pengurangan beban komputasi pada proses *round* dapat berpengaruh terhadap waktu pemrosesan data

secara keseluruhan (NIST, 2023). Parameter ini digunakan untuk membuktikan apakah algoritma yang diusulkan benar-benar mampu meningkatkan efisiensi enkripsi dibandingkan AES-128 standar maupun penelitian baseline.

2.2.2 Kecepatan Dekripsi

Kecepatan dekripsi merupakan parameter yang digunakan untuk mengukur waktu yang dibutuhkan algoritma dalam mengembalikan *ciphertext* menjadi plaintext. Parameter ini penting karena algoritma kriptografi simetris tidak hanya dituntut cepat pada proses enkripsi, tetapi juga harus efisien pada proses dekripsi (Gamido, 2020). Dalam konteks penelitian ini, proses dekripsi melibatkan operasi kebalikan dari *dynamic bit permutation*, yaitu penggunaan tabel invers dari BP1, BP2, BP3, dan BP4. Oleh karena itu, pengujian kecepatan dekripsi diperlukan untuk memastikan bahwa mekanisme permutasi dinamis tidak menambah beban komputasi yang berlebihan pada proses pengembalian data. Kecepatan dekripsi dihitung berdasarkan selisih waktu antara awal dan akhir proses dekripsi.

2.2.3 *Avalanche effect*

Algoritma yang tidak menunjukkan *Avalanche effect* yang memadai memiliki randomisasi yang lemah, sehingga kriptanalis dapat membuat prediksi tentang input hanya berdasarkan analisis output (Mohamed dkk., 2022).

Strict Avalanche Criterion (SAC) merupakan formalisasi matematis dari *Avalanche effect* yang terpenuhi jika, setiap kali satu bit input dibalik (complemented), setiap bit output berubah dengan probabilitas tepat 50% (Mohamed dkk., 2022). Nilai AE yang memenuhi SAC adalah AE ≥ 50%. *Avalanche effect* merupakan salah satu

faktor terpenting dalam menilai kekuatan algoritma kriptografi karena membuktikan bahwa output bersifat acak terlepas dari input, sehingga algoritma mampu menyembunyikan seluruh informasi berguna dari data asli (Mohamed dkk., 2022). Penelitian Baladhay dan De Los Reyes (2024) mampu menghasilkan *Avalanche effect* rata-rata 52,92% yang melampaui ambang batas SAC 50% dan menjadi target pembanding dalam penelitian ini.

2.2.4 *Throughput*

Throughput merupakan parameter yang digunakan untuk mengukur banyaknya data yang dapat diproses oleh algoritma dalam satuan waktu tertentu. Berbeda dengan kecepatan enkripsi dan dekripsi yang hanya menunjukkan waktu proses, *Throughput* memberikan gambaran efisiensi algoritma terhadap ukuran data yang diproses. Semakin tinggi nilai *Throughput* , semakin besar jumlah data yang dapat dienkripsi atau didekripsi dalam waktu yang sama. Jika hanya menggunakan waktu enkripsi dan dekripsi, hasil pengujian dapat dipengaruhi oleh ukuran file. *Throughput* membantu menormalkan hasil pengujian dengan melihat kemampuan algoritma memproses data per satuan waktu. Parameter ini juga digunakan dalam penelitian Baladhay dan De Los Reyes (2024), yang menunjukkan peningkatan *Throughput* enkripsi sebesar 31,12% dan *Throughput* dekripsi sebesar 25,50% dibandingkan AES standar.

2.2.5 Entropi Shannon

Entropi Shannon adalah ukuran kuantitatif dari ketidakpastian atau keacakan dalam sebuah sumber informasi yang digunakan untuk mengukur

keacakan distribusi byte dalam *ciphertext.* Penelitian terbaru oleh Abu-Faraj dkk. (2023) mengonfirmasi bahwa algoritma enkripsi berbasis *logistic chaotic map* secara konsisten menghasilkan *ciphertext* dengan entropi Shannon mendekati 8,0, membuktikan efektivitas pendekatan chaotic dalam menghasilkan randomisasi yang tinggi. Pada data 8-bit, nilai entropi maksimum adalah 8,0 bit yang tercapai ketika setiap nilai byte (0–255) muncul dengan probabilitas sama yaitu 1/256. Entropi Shannon yang mendekati nilai maksimum pada *ciphertext* mengindikasikan bahwa algoritma berhasil mengaburkan pola statistik dari *plaintext* asli, sebuah manifestasi dari properti confusion yang baik. Sebaliknya, *ciphertext* dengan nilai entropi yang jauh di bawah 8,0 mengindikasikan adanya distribusi byte yang tidak seragam, yang berpotensi memberikan informasi parsial kepada penyerang (Zolfaghari dkk., 2022).

2.2.6 *Key sensitivity*

Key sensitivity merupakan parameter yang digunakan untuk mengukur sensitivitas algoritma terhadap perubahan kecil pada kunci enkripsi. Algoritma kriptografi yang baik harus memiliki sensitivitas kunci yang tinggi, artinya perubahan satu bit pada kunci harus menghasilkan *ciphertext* yang sangat berbeda meskipun plaintext yang digunakan sama. Parameter ini penting karena difusi algoritma simetris sangat bergantung pada kerahasiaan dan pengaruh kunci terhadap proses enkripsi (Elghandour dkk., 2022). Pengujian *Key sensitivity* perlu dilakukan karena penelitian ini mengusulkan *dynamic key-dependent bit permutation*, yaitu tabel permutasi yang dibangkitkan berdasarkan kunci enkripsi. Jika perubahan kecil pada kunci tidak menghasilkan perubahan *ciphertext* yang

signifikan, maka mekanisme permutasi dinamis belum menunjukkan ketergantungan kunci yang kuat. Sebaliknya, apabila perubahan satu bit pada kunci menghasilkan perubahan besar pada *ciphertext*, maka mekanisme *logistic chaotic map* dan *Fisher-Yates shuffle* dapat dikatakan berfungsi dalam membangkitkan pola permutasi yang sensitif terhadap kunci. Pada algoritma berbasis *chaotic map*, sensitivitas terhadap kunci menjadi karakteristik penting karena perubahan kecil pada nilai awal *Chaos* seharusnya menghasilkan urutan yang berbeda secara signifikan. Oleh karena itu, parameter ini digunakan untuk membuktikan bahwa tabel permutasi yang dihasilkan benar-benar bersifat dinamis, *key-dependent*, dan tidak sama untuk setiap kunci.

2.2.7 *Frequency test*

Frequency test merupakan pengujian statistik yang digunakan untuk menilai keseimbangan kemunculan bit 0 dan bit 1 dalam suatu deret biner. Dalam penelitian ini, uji tersebut digunakan untuk melihat apakah *ciphertext* yang dihasilkan memiliki distribusi bit yang mendekati seimbang, karena kondisi tersebut menunjukkan bahwa hasil enkripsi tidak memiliki bias statistik yang mudah dikenali. Pada pengujian berbasis NIST, Frequency (Monobit) Test menilai apakah jumlah bit 0 dan 1 hampir sama sebagai indikator awal randomisasi deret bit (Ramar dkk., 2025).

2.2.8 *Runs test*

Runs test merupakan pengujian statistik yang digunakan untuk menilai pola pergantian bit dalam suatu deret biner. Uji ini mengamati jumlah run, yaitu

rangkaian bit bernilai sama yang muncul secara berurutan, untuk mengetahui apakah pergantian antarbit berlangsung secara wajar seperti pada data acak. Dalam konteks penelitian ini, *Runs test* digunakan untuk mengevaluasi apakah *ciphertext* masih menunjukkan keteraturan pola atau telah menyebar secara acak, sehingga mendukung penilaian terhadap kualitas difusi hasil enkripsi (Almaraz Luengo dkk., 2023).

2.2.9 *Chi-Square test*

Chi-Square test merupakan pengujian statistik yang digunakan untuk mengukur kesesuaian antara distribusi frekuensi data hasil pengamatan dan distribusi yang diharapkan. Pada evaluasi *ciphertext*, uji ini dipakai untuk menilai apakah distribusi byte hasil enkripsi mendekati distribusi seragam, sehingga pola statistik plaintext tidak lagi terlihat pada *ciphertext*. Dalam penelitian ini, *Chi-Square test* digunakan untuk mendukung penilaian bahwa hasil enkripsi memiliki tingkat keseragaman yang baik dan tidak menunjukkan dominasi nilai tertentu yang dapat dimanfaatkan dalam analisis statistik (Ali P dkk., 2024).

2.3 State of The Art

Berdasarkan rumusan masalah dan tujuan penelitian yang telah dibuat, maka dilakukan penyusunan literature review dari penelitian sebelumnya yang berkaitan dengan modifikasi AES. Beberapa penelitian sebelumnya yang dilakukan dapat dilihat pada tabel 2.1.

Tabel 2.1. State of The Art

NO	Penulis	Judul	Algoritma/Metode	Parameter yang Diukur	Hasil/Kesimpulan
1	(Gamido, 2020)	Implementation of a *Bit permutation-Based Advanced Encryption Standard* for Securing Text and Image Files	*MixColumns* diganti *Bit permutation* statis.	*Avalanche effect*, Entropi, Koefisien Korelasi, NPCR, UACI, Kecepatan Enkripsi	Enkripsi teks 18,47% lebih cepat, dekripsi 18,77% lebih cepat. *Avalanche effect* meningkat. AE dan entropi memenuhi standar.
2	(Salih dkk., 2020)	Dual *Dynamic* XOR Table and *MixColumns* Transformation on AES	*MixColumns* dibuat dinamis via 3D Chebyshev *Chaotic map*, XOR table	*Avalanche effect*, Resistansi Kriptanalisis	Komponen dinamis berbasis *chaotic map* meningkatkan resistansi kriptanalisis, beban

NO	Penulis	Judul	Algoritma/Metode	Parameter yang Diukur	Hasil/Kesimpulan
			dinamis, tetapi *MixColumns* tetap dipertahankan	Diferensial dan Linear	komputasi *Galois Field* tidak berkurang karena *MixColumns* tetap ada.
3	(Baladhay & De Los Reyes, 2024)	AES-128 reduced-*round* permutation by replacing *MixColumns*	*MixColumns* diganti *Bit permutation* (tabel statis) + pengurangan *round* 10 menjadi 6.	*Avalanche effect*, Kecepatan Enkripsi, Kecepatan Dekripsi, *Throughput*	Enkripsi 76,76% lebih cepat, dekripsi 55,46% lebih cepat, AE 52,92% (melampaui SAC 50%).
4	(Zinabu & Asferaw, 2022)	Enhanced Efficiency of *Advanced Encryption Standard* (EE-AES) Algorithm	*MixColumns* diganti Bitwise Reverse Transposition.	Kecepatan Enkripsi, Kecepatan Dekripsi, *Throughput*	Enkripsi 128,953% lebih cepat, dekripsi 115,4% lebih cepat, *Throughput* meningkat 140,8%.
5	(Alemami dkk., 2023)	Advanced approach for encryption using *Advanced*	kunci dinamis via Sine *Chaotic map*, S-Box dinamis,	*Avalanche effect*, Histogram, Kecepatan	AE mencapai 72,656%. kunci lebih kompleks dan dinamis, namun

NO	Penulis	Judul	Algoritma/Metode	Parameter yang Diukur	Hasil/Kesimpulan
		Encryption Standard with *chaotic map*	operasi XOR tambahan, tetapi *MixColumns* tetap dipertahankan.	Enkripsi/Dekripsi, *Throughput*	kompleksitas meningkat signifikan dan *MixColumns* tidak digantikan
6	Agil Zulfa (Hasil penelitian)	MODIFIKASI ALGORITMA AES-128 DENGAN *REDUCED ROUND* DAN *DYNAMIC BIT PERMUTATION* BERBASIS *LOGISTIC CHAOTIC MAP*	Penggantian *MixColumns* dengan *Bit permutation* dinamis via *Logistic chaotic map* + *Fisher-Yates shuffle*, tabel dibangkitkan dari SHA-256(key).	Kecepatan Enkripsi, Kecepatan Dekripsi, *Throughput , Avalanche effect*, Entropi Shannon, dan *Key sensitivity*.	-

2.4 Matriks Penelitian

Tabel 2.2 merupakan matriks penelitian yang memuat perbandingan ruang lingkup, metode, dan karakteristik penelitian terdahulu yang relevan dengan modifikasi algoritma AES. Selain itu, matriks ini dapat memberikan informasi tentang perbedaan penelitian yang dilakukan dan penelitian terdahulu.

Tabel 2.2. Matriks Penelitian

No.	Penulis	Judul	Algoritma	Ruang Lingkup							
				Penerapan Algoritma							
				MixColumns	Added XOR	*Round* Reduction	3D *Chaotic map*	1D *Chaotic map*	*Bit permutation* Statis	*Bit permutation* Dinamis	Bitwise Reverse Transposition
1	(Gamido, 2020)	Implementation of a *Bit permutation-Based Advanced Encryption Standard* for Securing Text and Image Files	*MixColumns* diganti *Bit permutation* statis.						✓		

No.	Penulis	Judul	Algoritma	Ruang Lingkup							
				Penerapan Algoritma							
				MixColumns	Added XOR	*Round* Reduction	3D *Chaotic map*	1D *Chaotic map*	*Bit permutation* Statis	*Bit permutation* Dinamis	Bitwise Reverse Transposition
2	(Salih dkk., 2020)	Dual *Dynamic* XOR Table and *MixColumns* Transformation on AES	*MixColumns* dibuat dinamis via 3D Chebyshev *Chaotic map*, XOR table dinamis, tetapi *MixColumns* tetap dipertahankan	✓	✓		✓				
3	(Baladhay & De Los Reyes, 2024)	AES-128 reduced-*round* permutation by replacing *MixColumns*	*MixColumns* diganti *Bit permutation* (tabel statis) + pengurangan *round* 10 menjadi 6.			✓			✓		

No.	Penulis	Judul	Algoritma	Ruang Lingkup							
				Penerapan Algoritma							
				MixColumns	Added XOR	*Round* Reduction	3D *Chaotic map*	1D *Chaotic map*	*Bit permutation* Statis	*Bit permutation* Dinamis	Bitwise Reverse Transposition
4	(Zinabu & Asferaw, 2022)	Enhanced Efficiency of *Advanced Encryption Standard* (EE-AES) Algorithm	*MixColumns* diganti Bitwise Reverse Transposition.								✓
5	(Alemami dkk., 2023)	Advanced approach for encryption using *Advanced Encryption Standard* with *chaotic map*	kunci dinamis via Sine *Chaotic map*, S-Box dinamis, operasi XOR tambahan, tetapi *MixColumns* tetap dipertahankan.	✓	✓			✓			

No.	Penulis	Judul	Algoritma	Ruang Lingkup							
				Penerapan Algoritma							
				MixColumns	Added XOR	*Round* Reduction	3D *Chaotic map*	1D *Chaotic map*	*Bit permutation* Statis	*Bit permutation* Dinamis	Bitwise Reverse Transposition
6	Agil Zulfa (Hasil penelitian)	MODIFIKASI ALGORITMA AES-128 DENGAN *REDUCED ROUND* DAN *DYNAMIC BIT PERMUTATION* BERBASIS *LOGISTIC CHAOTIC MAP*	Penggantian *MixColumns* dengan *Bit permutation* dinamis via *Logistic chaotic map* + *Fisher-Yates shuffle*, tabel dibangkitkan dari SHA-256(key).			✓		✓		✓	

Berdasarkan kajian terhadap lima penelitian terdahulu yang tersaji pada tabel 2.2, terdapat dua arah pendekatan utama dalam modifikasi algoritma AES yang telah dilakukan oleh para peneliti sebelumnya. Pendekatan pertama berfokus pada penggantian fungsi

MixColumns dengan mekanisme yang lebih ringan secara komputasi. Gamido (2020) mengganti *MixColumns* dengan *bit permutation* berbasis precomputed lookup table, Zinabu dan Asferaw (2022) menggantinya dengan bitwise reverse transposition, dan Baladhay serta De Los Reyes (2024) mengembangkan pendekatan serupa dengan tambahan pengurangan jumlah *round* dari sepuluh menjadi enam.

Ketiga penelitian ini berhasil meningkatkan kecepatan enkripsi secara signifikan, namun memiliki kelemahan yang sama yaitu tabel permutasi yang digunakan bersifat statis dan tidak bergantung pada kunci enkripsi, sehingga rentan terhadap eksploitasi sebagaimana diidentifikasi oleh Salih dkk. (2020). Pendekatan kedua berfokus pada penggunaan *chaotic map* untuk membuat komponen AES bersifat dinamis dan bergantung pada kunci. Salih dkk. (2020) membuat *MixColumns* dinamis menggunakan 3D Chebyshev *chaotic map*, dan Alemami dkk. (2023) menggunakan Sine *chaotic map* untuk membangkitkan S-Box dan kunci secara dinamis. Meskipun kedua penelitian ini berhasil meningkatkan resistansi terhadap kriptanalisis, keduanya tetap mempertahankan *MixColumns* sehingga beban komputasi akibat operasi *Galois Field* tidak berkurang.

Dari pemetaan dua arah pendekatan tersebut, dapat diidentifikasi bahwa penelitian-penelitian yang berhasil ditinjau umumnya hanya berfokus pada salah satu pendekatan, yaitu penggantian *MixColumns* atau penggunaan *chaotic map*, sehingga belum ditemukan penelitian yang secara simultan mengintegrasikan kedua pendekatan tersebut dalam satu algoritma AES. Penelitian yang mengganti

MixColumns tidak menggunakan *chaotic map*, dan penelitian yang menggunakan *chaotic map* tidak mengganti *MixColumns*. Celah inilah yang menjadi landasan penelitian ini, yaitu mengusulkan modifikasi AES-128 dengan menggantikan *MixColumns* menggunakan *dynamic key-dependent bit permutation* berbasis *logistic chaotic map*, sehingga kelebihan kecepatan dari pendekatan Baladhay dan De Los Reyes (2024) dapat dipertahankan sekaligus kelemahan tabel permutasi statis dapat diatasi melalui mekanisme dinamisme berbasis *chaotic map* sebagaimana dibuktikan relevansinya oleh Salih dkk. (2020) dan Alemami dkk. (2023).