ENKRIPSI CITRA DIGITAL MENGGUNAKAN *ARNOLD'S CAT MAP* DAN *NONLINEAR CHAOTIC ALGORITHM*

Ronsen Purba[1], Arwin Halim[2], Indra Syahputra[3]

[1,2,3] STMIK MIKROSKIL

Jl. Thamrin no 112, 124, 140 Medan 20212

[1]ronsen@mikroskil.ac.id, [2]arwin@mikroskil.ac.id, [3]indrasyahpoetra91@gmail.com

Abstrak

Dalam penelitian diusulkan algoritma enkripsi citra digital menggunakan dua buah fungsi *chaos*, yakni *Arnold's Cat Map* (ACM) dan *Nonlinear Chaotic Algorithm* (NCA). Hal ini dilakukan untuk mendapatkan enkripsi citra yang lebih robust. Proses enkripsi yang dilakukan meliputi pengacakan susunan *pixel* menggunakan *Arnold's Cat Map*, pengacakan nilai RGB dan pengubahan nilai dalam citra memanfaatkan bilangan acak yang dibangkitkan menggunakan *Nonlinear Chaotic Algorithm* (NCA). Hasil pengujian menunjukkan *cipher image* memiliki distribusi intensitas *pixel* yang uniform. Selain itu *pixel-pixel* yang bertetangga memiliki koefisien korelasi yang rendah, koefisien korelasi yang rendah mengindikasikan *pixel* yang bertetangga tidak memiliki hubungan. Sifat *chaos* yang sensitif terhadap kondisi awal ditunjukkan dengan *cipher image* yang didekripsi tidak kembali ke citra semula jika kunci yang digunakan tidak sama dengan kunci yang digunakan waktu proses enkripsi.

Kata kunci : *enkripsi, citra digital, chaos, Arnold's Cat Map, Nonlinear Chaotic Algorithm*

1. Pendahuluan

Citra adalah salah satu media yang menyajikan informasi secara visual, terkadang informasi yang ada pada citra bersifat privasi dan rahasia sehingga aspek keamanannya perlu diperhatikan. Algoritma kriptografi konvensional seperti AES, DES, IDEA, RC4, dan lain sebagainya dianggap kurang cocok dalam pengamanan informasi citra. Karena data citra berbeda dengan data tekstual, data citra memiliki unsur yang spesial seperti volume data besar, redudansi tinggi dan *pixel* saling berhubungan [1]. Proses enkripsi seharusnya membuat *pixel* di dalam citra tidak lagi berhubungan sehingga menyulitkan penyerang dalam melakukan analisis statistik [2]. Penerapan teori *chaos* pada kriptografi modern telah menjadi topik penelitian dan perdebatan, alasannya terletak pada unsur intrinsik seperti sensitivitas terhadap kondisi awal dan kontrol parameter, nilai yang dihasilkan acak dan lain sebagainya [3]. Sensisitivitas pada *chaos* sesuai dengan dua sifat dasar dari *cipher* yang baik : *confusion* and *diffusion* [4]. Beberapa contoh dari *chaos* adalah *Arnold's Cat Map* (ACM), Circle Map, *Logistic Map, Tent Map, Baker's Map* dan lain sebagainya.

Enkripsi citra yang hanya mengunakan *Arnold's Cat Map* (ACM) dianggap tidak aman karena sifat priodiknya yang dapat mengembalikan citra asli melalui serangan brute force artinya nilai parameternya dapat ditemukan dengan mudah [5]. Penelitian yang dilakukan Munir [2] dan Kumari et al [6], menggunakan *Arnold's Cat Map* (ACM) untuk mengacak susunan *pixel* dan *Logistic Map* untuk mengubah nilai *pixel*. Hasil analisis menunjukkan *cipher image* tidak dapat dikenali dan nilai *pixel*-nya tidak saling berhubungan, namun kriptografi berbasis *Logistic Map* memiliki ruang kunci yang kecil dan keamanan yang lemah [7]. Oleh karena itu, Gao et al (2006)[7], merancang *Nonlinear Chaotic Algorithm* (NCA).

Berdasarkan hasil analisis yang dilakukan, enkripsi citra menggunakan *Nonlinear Chaotic Algorithm* (NCA) menunjukan *cipher image* tidak dapat dikenali, *pixel* tidak saling berhubungan, ruang kunci besar dan keamanan yang tinggi.

Penelitian ini menyajikan algoritma enkripsi citra digital berbasis *chaos* dengan menggunakan ACM dan NCA. ACM digunakan untuk mengacak susunan *pixel* dan NCA digunakan sebagai pengganti *Logistic Map*. NCA tidak hanya digunakan untuk mengubah nilai pada *plain image*, nilai acak yang dihasilkan dimanfaatkan untuk mengacak susunan RGB dari *plain image*.

2. Tinjauan Pustaka

Teori *chaos* adalah studi dari sistem dinamika yang sensitif terhadap kondisi awal [8]. Dalam matematika, teori *chaos* menggambarkan kebiasaan dari suatu sistem dinamis, yang keadaannya selalu berubah seiring dengan berubahnya waktu, dan sangat sensitif terhadap kondisi awal yang diberikan [9]. Walaupun sistem *chaos* berlangsung acak tetapi sistem chaotic dapat ditentukan secara matematis, hal ini disebabkan sistem *chaos* mengikuti hukum-hukum yang berlaku di alam [10].

Penerapan teori *chaos* pada kriptografi modern telah menjadi topik penelitian dan perdebatan dalam beberapa dekade terakhir, alasannya terletak pada unsur intrinsik seperti sensitivitas terhadap kondisi awal dan kontrol parameter, nilai yang dihasilkan acak dan lain sebagainya [3]. Sensisitivitas pada *chaos* sesuai dengan dua sifat dasar dari *cipher* yang baik: *confusion* and *diffusion* [4].

2.1 *Arnold's Cat Map* (ACM)

Metode *Arnold's Cat Map* (ACM) diperkenalkan pertama kali oleh seorang ahli matematik Rusia yang bernama Vladimir I. Arnold, pada tahun 1960 yang mendemonstrasikan algoritmanya tersebut dengan menggunakan citra kucing [11]. Algoritma *Arnold's Cat Map* dapat didefenisikan sesuai Persamaan 1.

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & b \\ c & bc+1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \bmod(N) \quad (1)$$

Dimana (x,y) posisi *pixel* di dalam citra berukuran N x N dan (x_i+1, y_i+1) posisi *pixel* yang baru setelah transformasi, b dan c adalah bulat positif sembarang. Determinan matriks harus sama dengan 1 agar hasil transformasinya tetap berada di dalam area citra yang sama (area-preserving). Algoritma ini termasuk one-to-one mapping, yang berarti setiap titik dalam matriks dapat ditransformasikan ke titik lainnya. Hasil citra acak tentunya berbeda untuk tiap jumlah iterasi m dan berubah secara periodik sesuai dengan perubahan parameter b, c dan besarnya ukuran citra. Nilai b, c dan m adalah kunci rahasia dari algoritma tranformasi ACM [2]. Namun sesudah iterasi tertentu citra acak dihasilkan akan kembali ke citra semula, oleh karena itu ACM disebut memiliki periode, sehingga ACM tidak bisa dikatakan murni acak, namun dapat digolongkan sebagai *Chaos* map karena sifat-sifat acak yang dimilikinya.

Menurut Struss [5], menyatakan bahwa jumlah iterasi yang diperlukan agar kembali ke citra semula adalah kurang dari 3N untuk N adalah dimensi citra. Dikarenakan sifat periodik ACM yang dapat menghasilkan kembali citra semula, maka enkripsi dengan menggunakan ACM saja tidak aman, sebab melalui hack sederhana nilai b dan c dapat ditemukan melalui operasi *brute force*. Selain itu ACM hanya mengubah posisi *pixel* di dalam citra tetapi tidak

mengubah nilai *pixel*. Sedangkan proses deskripsi ACM kebalikan dari proses enkripsinya, sesuai Persamaan 2.

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} 1 & b \\ c & bc+1 \end{bmatrix}^{-1} \begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} \bmod(N) \quad (2)$$

2.2 *Nonlinear Chaotic Algorithm* (NCA)

Nonlinear Chaotic Algorithm merupakan algoritma pembangkit bilangan acak (PRNG) yang dirancang oleh Haojiang Gao, Yisheng Zhang, Shuyun Liang dan Dequn Li dalam jurnal yang berjudul "*A new chaotic algorithm for image encryption*" yang diterbitkan pada tahun 2006. Algoritma yang didesain dengan memanfaatkan fungsi tangen dan fungsi perpangkatan, sesuai Persamaan 3

$$x_{i+1} = \lambda * \tan(\alpha x_i) * (1 - x_i)^{\beta} \quad (3)$$

Dimana $x_i \in (0,1)$, i = 1,2,3,4…….. n dan $x_i + 1 > x_i$ ketika x_i= 1/(1+β). Sedangkan untuk parameter λ akan dijelaskan pada persamaan berikut :

$$\lambda = \mu * ctg\left(\frac{\alpha}{1+\beta}\right) * \left(1 + \frac{1}{\beta}\right)^{\beta} \quad (4)$$

$\mu = 1 - \beta^{-4} > 0$. Jadi Persamaan NCA bisa didefinisikan sebagai Berikut:

$$x_{i+1} = (1 - \beta^{-4}) * ctg\left(\frac{\alpha}{1+\beta}\right) * \left(1 + \frac{1}{\beta}\right)^{\beta} * tan(\alpha * x_i) * (1 - x_i)^{\beta} \quad (5)$$

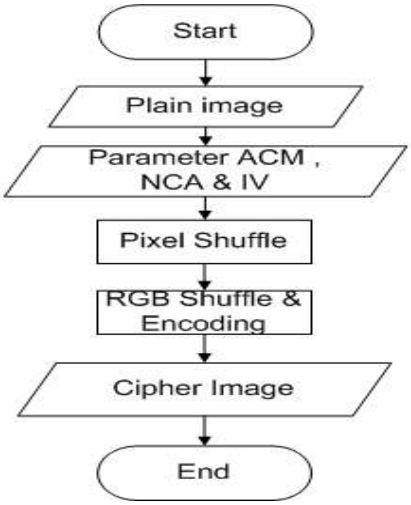
Dimana $X_i \in (0, 1)$, $\alpha \in (0,1.4]$, $\beta \in [5,43]$, atau $X_i \in (0, 1)$, $\alpha \in (1.4,1.5]$, $\beta \in [9,38]$, atau $X_i \in (0, 1)$, $\alpha \in (1.5,1.57]$, $\beta \in [3,15]$.

Berdasarkan eksperimen yang dilakukan, pada proses enkripsi dan dekripsi citra menggunakan NCA menunjukkan keuntungan yakni ruang kunci yang lebih besar, kemananan yang tinggi, dan efisien. Hal ini sangat cocok untuk mengenkripsi citra secara *realtime* dan aplikasi transmisi lainnya. Selain itu mengubah kondisi awal dan parameter bisa memiliki efek yang nyata pada citra yang terenkripsi sehingga tidak bisa kembali ke citra semula [7].

3. Analisis Proses Enkripsi dan Dekripsi

3.1 Proses Enkripsi

Proses enkripsi meliputi: *pixel shuffle*, RGB *shuffle* dan e*ncoding*. *Pixel shuffle* menggunakan Algoritma ACM pada persamaan (1), untuk mengacak susunan *pixel* pada *plain image*, sedangkan RGB *shuffle* dan *encoding* digunakan untuk mengacak nilai RGB pada *plain image* serta mengubah nilai RGB dengan peng *XOR*-an yang menggunakan skema *Chaining Block Cipher* (CBC). Proses ini memanfaatkan bilangan acak yang dibangkitkan menggunakan NCA seperti pada Persamaan 5. Gambar 1 di bawah ini menunjukkan Diagram Alir proses enkripsi.



Gambar 1. Diagram Alir Proses Enkripsi

1. *Pixel Shuffle*.

Pada tahap ini dilakukan operasi permutasi menggunakan ACM untuk mengacak susunan *pixel*. Berikut ini contoh proses *pixel shuffle* menggunakan *plain image* berukuran 3x3

	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

Ket : Posisi *pixel plain image*.

Parameter *Arnold's Cat Map* yang diinput *m=2, b=2*, dan *c=4*, kemudian dilakukan dengan tranformasi ACM dan menghasilkan Iterasi pertama m=1.

- $\begin{bmatrix}1 & 2\\4 & 9\end{bmatrix} * \begin{bmatrix}0\\0\end{bmatrix} = \begin{bmatrix}0\\0\end{bmatrix} \quad mod\ 3 = \begin{bmatrix}0\\0\end{bmatrix}$
- $\begin{bmatrix}1 & 2\\4 & 9\end{bmatrix} * \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}2\\9\end{bmatrix} \quad mod\ 3 = \begin{bmatrix}2\\0\end{bmatrix}$
- $\begin{bmatrix}1 & 2\\4 & 9\end{bmatrix} * \begin{bmatrix}0\\2\end{bmatrix} = \begin{bmatrix}4\\18\end{bmatrix} \quad mod\ 3 = \begin{bmatrix}1\\0\end{bmatrix}$
- $\begin{bmatrix}1 & 2\\4 & 9\end{bmatrix} * \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}1\\4\end{bmatrix} \quad mod\ 3 = \begin{bmatrix}1\\1\end{bmatrix}$
- $\begin{bmatrix}1 & 2\\4 & 9\end{bmatrix} * \begin{bmatrix}1\\1\end{bmatrix} = \begin{bmatrix}3\\13\end{bmatrix} \quad mod\ 3 = \begin{bmatrix}0\\1\end{bmatrix}$
- $\begin{bmatrix}1 & 2\\4 & 9\end{bmatrix} * \begin{bmatrix}1\\2\end{bmatrix} = \begin{bmatrix}5\\22\end{bmatrix} \quad mod\ 3 = \begin{bmatrix}2\\1\end{bmatrix}$
- $\begin{bmatrix}1 & 2\\4 & 9\end{bmatrix} * \begin{bmatrix}2\\0\end{bmatrix} = \begin{bmatrix}2\\8\end{bmatrix} \quad mod\ 3 = \begin{bmatrix}2\\2\end{bmatrix}$
- $\begin{bmatrix}1 & 2\\4 & 9\end{bmatrix} * \begin{bmatrix}2\\1\end{bmatrix} = \begin{bmatrix}4\\17\end{bmatrix} \quad mod\ 3 = \begin{bmatrix}1\\2\end{bmatrix}$
- $\begin{bmatrix}1 & 2\\4 & 9\end{bmatrix} * \begin{bmatrix}2\\2\end{bmatrix} = \begin{bmatrix}6\\26\end{bmatrix} \quad mod\ 3 = \begin{bmatrix}0\\2\end{bmatrix}$

Interasi 1

	0	1	2
0	1	5	9
1	3	4	8
2	2	6	7

Interasi 2

	0	1	2
0	1	5	9
1	3	4	8
2	2	6	7

2. RGB *Shuffle* dan *Encoding*.

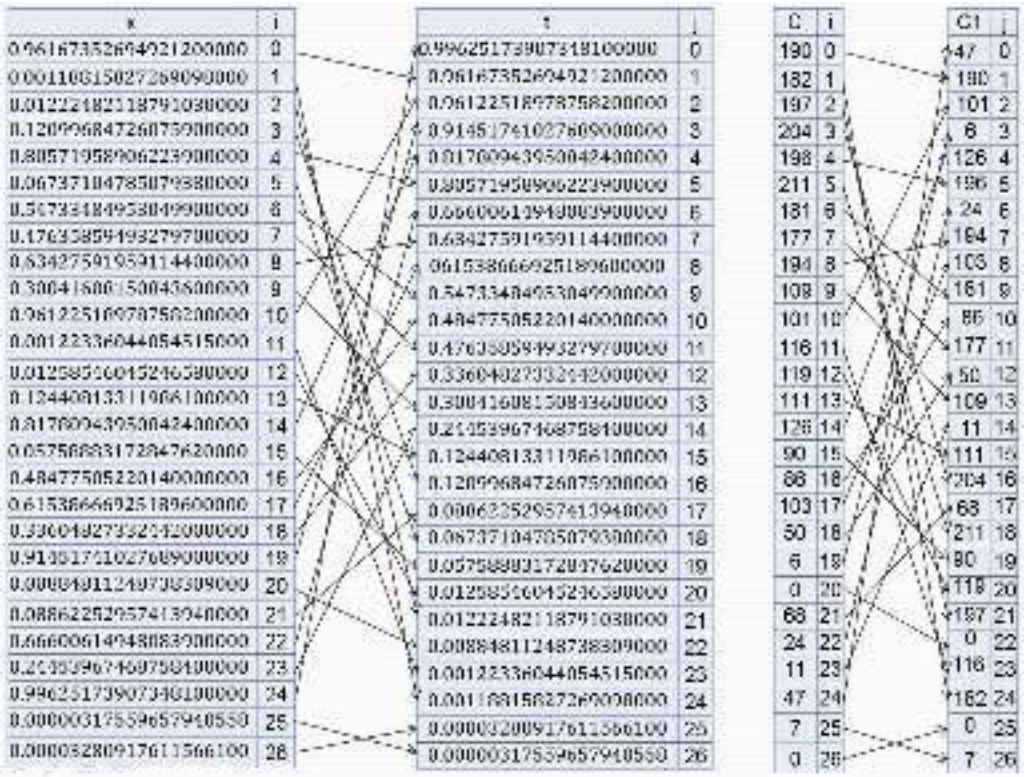
Proses ini meliputi pembangkitan deretan bilangan acak sebanyak *M* x *N* x *3*, kemudian deretan bilangan acak diduplikasi ke dalam variabel *t* dan diurutkan secara

menurun. Selanjutnya bilangan acak tersebut dikonversi dari bilangan riil ke bilangan bulat menggunakan Persamaan 6 [12] :

$$T(x, size) = \left\| x * 10^{count} \right\|, x \neq 0 \quad (6)$$

Dimana *count di*mulai dari 1 dan bertambah 1 sampai $x * 10^{count} > 10^{size-1}$. Hasilnya kemudian diambil bagian bulatnya saja (dilambangkan dengan pasangan garis ganda pada persamaan 6). Sebagai contoh, misalkan $x = 0.0005467854$ dan $size = 4$, maka dimulai dari $count = 1$ sampai $count = 7$ diperoleh $0.0005467854 * 10^7 = 5467.854 > 10^3$. Kemudian ambil bagian bulat dari $||5467.854||=5467$, kemudian nilai yang dikonversi selanjutnya dimodulokan dengan 256 dan ditampung kedalam *array K*. Cara yang sama dilakukan dengan nilai acak yang lainnya. Kemudian nilai RGB *plain image* ditampung ke dalam 1 array, Selanjutnya nilai acak yang dibangkitkan dibandingkan dengan nilai yang telah diurutkan. Sesudah itu nilai RGB *plain image* diacak sesuai dengan indeks perbandingan. Gambar 2. berikut menunjukkan ilustrasi proses perbandingan dan pengacakan nilai RGB pada *plain image*. Selanjutnya masuk pada tahap *encoding* yang diambil berdasarkan skema *encoding* yang dilakukan oleh Munir (2012)[2], yaitu skema subsitusi yang diadopsi dari *Cipher block chaining* dengan melakukan operasi subsitusi *XOR*. Persamaan 7 menunjukkan skema subsitusi *Cipher block chaining*.

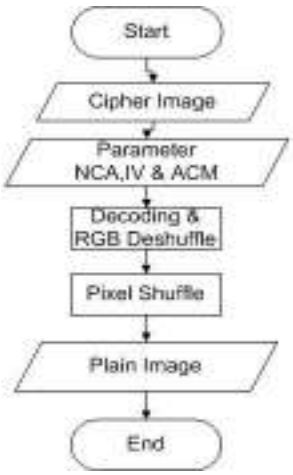
$$C_i = (P_i \oplus C_{i-1}) \oplus K_i \quad (7)$$



Gambar 2. Ilustrasi RGB *shuffle*.

3.2 Proses Dekripsi

Proses dekripsi adalah proses kebalikan dari enkripsi dimana proses yang dilakukan terlebih dahulu adalah proses *decoding* dan RGB *deshuffle* kemudian *pixel shuffle*. Diagram *activity* proses dekripsi diperlihatkan pada Gambar 3.

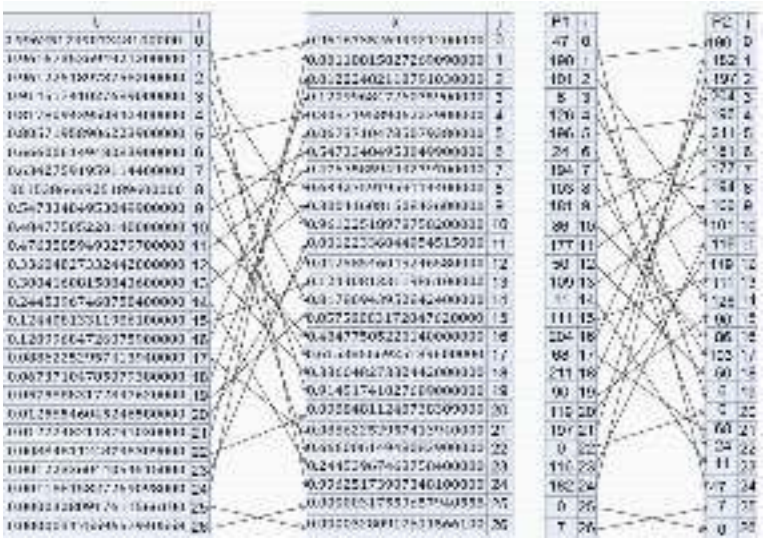


Gambar 3. Diagram Alir Proses Dekripsi

Dimana proses *decoding* meggunakan Persamaan 8.

$$\mathbf{P_i = (C_i \oplus K_i) \oplus C_{i-1}} \quad (8)$$

Sedangkan Proses RGB *deshuffle* sama seperti proses RGB *shuffle* pada proses enkripsi perbedaannya hanya terletak pada proses perbandingan nilai acak, dimana proses perbandingan yang dilakukan terhadap nilai yang sudah diurutkan dengan nilai acak yang belum di urutkan, Gambar 4 menunjukan ilustrasi dari proses RGB *deshuffle*. Untuk proses *pixel deshuffle* menggunakan invers dari Persamaan 1 yang menghitung kebalikan transformasi *Arnold's Cat Map* yang ditunjukan pada Persamaan 2. Setelah tahap *pixel deshufle* dilakukan, maka proses dekripsi telah selesai sehingga citra dapat kembali ke citra awal.



Gambar 4. Ilustrasi RGB *deshuffle*

4. Hasil dan Pembahasan

4.1 Hasil

Berikut hasil pengujian yang dilakukan terhadap *plain image* (Gambar 5) berukuran 250 x 250 *pixel*, yang disimulasikan menggunakan bahasa pemrograman VB.net. Parameter kunci yang digunakan di dalam pengujian ini adalah m=51, b=212, c= 313, x_0= 0.3101, α=0.01, β= 6.2, iv= 16.

Gambar 5. Citra Sample

Citra hasil enkripsi (*cipher image*) ditunjukkan pada Gambar 6(a) dan dapat dilihat bahwa citra tersebut tidak dapat dikenali, menggunakan kunci yang sama citra hasil deksripsi terhadap *cipher image* yang ditunjukkan Gambar 6(b) dan dapat dilihat bahwa citra yang dihasilkan kembali ke citra semula.

(a)

(b)

Gambar 6 (a) *Cipher Image*; (b) Citra hasil dekripsi.

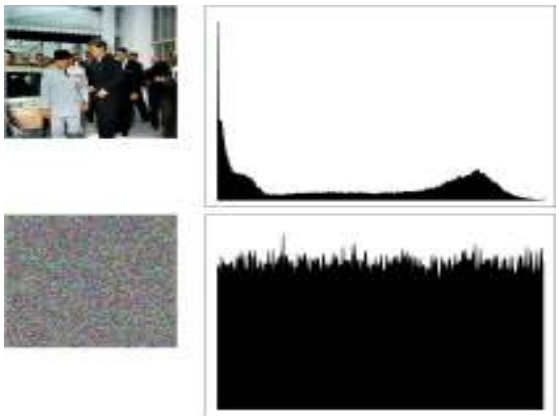
4.2 Pembahasan

Terhadap sistem yang dibuat dilakukan pengujian terhadap 3 hal, yakni (1) analisis histogram untuk melihat perbedaan histrogram citra asli dengan citra hasil enkripsi, (2) analisis korelasi untuk melihat hilangnya hubungan antara citra asli dengan citra hasil enkripsi dan (3) analisis sensitivitas kunci untuk mengetahui tingkat sensitivitas perubahan kunci.

4.2.1 Analisis Histogram

Analisis histogram dilakukan untuk mengetahui informasi dari penyebaran nilai *pixel*, distribusi nilai *pixel* pada gambar biasanya berkonsentrasi pada sebagian ruang nilai *pixel*, Enkripsi yang baik menyebabkan nilai *pixel* menyebar disepanjang ruang nilai *pixel*, selain itu histogram dari *cipher image* harus berbeda dari *plain image*. Jika histogram pada *cipher image* dan *plain image* memiliki kemiripan, maka penyerang dapat melakukan analisis statistik untuk mendapatkan beberapa informasi [13]. Menurut Munir (2012)[2], dalam melakukan serangan dengan teknik analisis statistik, penyerang menggunakan histogram untuk menganalisis frekuensi kemunculan intensitas *pixel* untuk mendeduksi kunci atau *pixel-pixel* didalam *plain image*. Enkripsi citra seharusnya menghasilkan histogram *cipher image* yang tidak memiliki kesamaan secara statistik dengan histogram *plain image* dan distribusi nilai pada *pixel* di dalam *cipher image* seharusnya memiliki distribusi yang (relatif) uniform yang ditunjukan dengan histogram yang terlihat datar, sehingga serangan dengan analisis statistik kurang efisien dilakukan.

Gambar 7 berikut ini memperlihatkan perbedaan histogram antara *plain image* dengan *cipher image*, dimana histogram yang dihasilkan oleh *cipher image* terlihat datar atau berdistribusi uniform sedangkan histogram yang dihasilkan *plain image* lebih berfokus pada beberapa ruang nilai *pixel*.



Gambar 7. Histogram pengujian citra sample

4.2.2 Analisis Korelasi

Analisis korelasi digunakan untuk menentukan hubungan antara dua variabel untuk menentukan kualitas enkripsi dari kriptosistem [14]. Enkripsi citra dikatakan bagus, jika algoritma enkripsi yang digunakan mengaburkan hubungan dari *plain image*, dan *cipher image* yang dihasilkan betul-betul acak dan tidak memiliki korelasi [15]. Nilai koefisien korelasi tidak dapat melebihi 1 dalam harga mutlak. Nilai koefisien korelasi +1 menyatakan hubungan linier (korelasi) sempurna yang menaik, nilai koefisien korelasi -1 menyatakan hubungan linier (korelasi) sempurna yang menurun, sedangkan antara -1 dan +1 menyatakan derajat ketergantungan linier antara dua peubah. Nilai koefisien yang dekat dengan -1 atau +1 menyatakan hubungan linier yang kuat antara x dan y, sedangkan nilai koefisien yang dekat dengan 0 menyatakan hubungan linier yang lemah. [2]

Untuk mengevaluasi korelasi dari 2 *pixel* yang saling berdekatan pada *plain image* dan *cipher image*, digunakan persamaan sebagai berikut [13]

$$r_{xy} = \frac{\mathrm{cov}(x, y)}{\sqrt{D(x)D(y)}} \tag{9}$$

Dimana:

$$\mathrm{cov}(x, y) = \frac{1}{n}\sum_{i=1}^{n}[x_i - E(x)][y_i - E(y)] \quad \text{(kovariansi)} \tag{10}$$

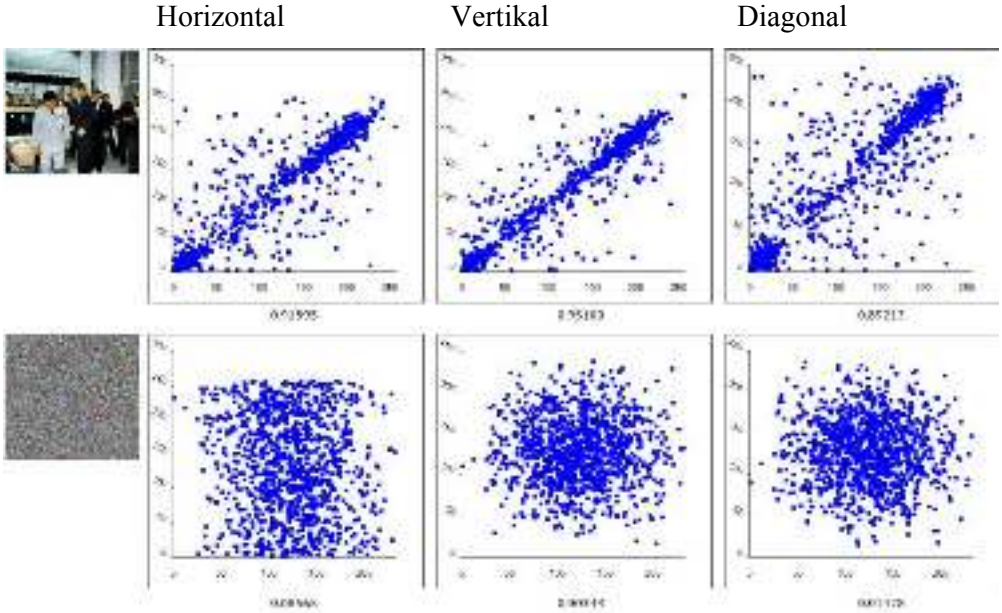
$$D(x) = \frac{1}{n}\sum_{i=1}^{n}[x_i - E(x)]^2 \quad \text{(standard deviasi)} \tag{11}$$

$$E(x) = \frac{1}{n}\sum_{i}^{n} x_i \quad \text{(rata-rata)} \tag{12}$$

Sedangkan untuk x dan y adalah nilai keabuan (gray scale value) dari dua *pixel* pada koordinat yang sama pada *plain image* dan *cipher image* [16]. Dengan persamaan di atas maka dihitung koefisien korelasi antara dua *pixel* yang bertetangga secara horizontal [f(x,y) dan f(x+1,y)], vertikal [f(x,y) dan f(x,y+1)], dan diagonal [f(x,y) dan f(x+1,y+1)], pada *plain image* maupun *cipher image* [2]. Gambar 8 menunjukkan korelasi antar *pixel* bertetangga terhadap *plain image* dan *cipher image*. Dapat dilihat pada Tabel 1 bahwa distribusi korelasi pada *pixel* yang bertetangga pada *plain image* di setiap arah nilai-nilainya berada disekitar garis diagonal 45°, yang mengindikasikan korelasi yang kuat dengan nilai koefisien korelasi yang mendekati angka 1. Sebaliknya, pada *cipher image* nilai-nilai *pixel* menyebar yang mengindikasikan *pixel* di dalamnya tidak lagi berkorelasi yang ditunjukkan dengan koefisien korelasi yang mendekati angka 0.

Tabel 1 Koefisien Korelasi Citra *sample*

Koefisien Korelasi	Horizontal	Vertikal	Diagonal
Plain image	0.91594	0.95099	0.89212
Cipher image	0.00922	0.00675	0.02698



Gambar 8. Analisis korelasi citra sampel.

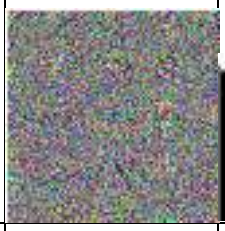
4.2.3 Analisis Sensitivitas Kunci

Sensitivitas kunci merupakan indeks yang sangat penting dalam system kriptografi, perubahan kecil pada kunci mengakibatkan hasil enkripsi yang berbeda. Dibandingkan dengan metode (algoritma) konvensional, sistem kriptografi berbasis *chaos* memiliki tingkat sensitivas kunci yang sangat tinggi terhadap perubahan kondisi awal [17].

Pengujian ini dilakukan terhadap nilai awal (x_0) untuk mengetahui perubahan kondisi yang terjadi pada *cipher image*. Tabel 2 menunjukkan pengujian yang dilakukan terhadap perubahan nilai awal (x_0), dengan kondisi sebagai berikut:

1. Percobaan 1 nilai x_0 + 0.0000001.
2. Percobaan 2 nilai x_0 + 0.000000001.
3. Percobaan 3 nilai x_0 + 0.00000000001.

Tabel 2 Perubahan nilai awal x_0

No	Parameter	Citra *output*	*Histogram*	K. Korelasi
1	- m = 51 - b = 212 - c = 313 - x_0 = 0.3101001 - α = 0.01 - β = 6.2 - iv = 16			- Hor = 0.00260 - Ver = -0.05272 - Dia = 0.00388
2	- m = 51 - b = 212 - c = 313 - x_0 = 0.310100001 - α = 0.01 - β = 6.2 - iv = 16			- Hor = -0.03284 - Ver = 0.01339 - Dia = -0.01339
3	- m = 51 - b = 212 - c = 313 - x_0 =0.31010000001 - α = 0.01 - β = 6.2 - iv = 16			- Hor = 0.02738 - Ver = 0.03775 - Dia = 0.01526

Setelah dilakukan perubahan nilai awal (x_0) , dapat dilihat bahwa masing-masing citra hasil terlihat seperti citra acak yang tidak dikenali dengan histogram berdistribusi uniform yang menunjukan intensitas *pixel* tetap menyebar di seluruh ruang nilai *pixel*. Sementara nilai koefisien korelasi berbeda-beda menunjukkan sensitivitas akibat perubahan nilai x_0. Jadi, perubahan yang dilakukan tidak membuat *cipher image* kembali ke *plain image* dan membuat citra hasil tampak seperti citra acak dengan distribusi intensitas *pixel* yang menyebar dan korelasi yang lemah antara *pixel* bertetangga

5. Kesimpulan

Melalui proses analisis dan pengujian yang dilakukan didapat kesimpulan sebagai berikut:

1. Analisis *histogram* memperlihatkan perbedaan antara *plain image* dan *cipher image*, dimana intesitas *pixel* pada *cipher image* menyebar di seluruh ruang nilai *pixel*, sehingga serangan menggunakan analisis statistik tidak dimungkinkan
2. *Pixel* dalam *cipher image* tidak lagi memiliki korelasi dengan *pixel* tetangganya yang diperlihatkan dengan disribusi *pixel* yang menyebar dengan nilai koefisien korelasi yang mendekati 0
3. Sifat *chaos* yang sensitif terhadap perubahan nilai awal yang sangat kecil yang membuat algoritma ini sulit untuk dijebol

Referensi

[1] Ranjan, Rajiv., dan Pal, Arup Kumar., 2012, Encryption of *Image* Using Chaotic Maps, International Conference on Recent Trends in Engineering & Technology.

[2] Munir, Rinaldi., 2012, Algoritma Enkripsi Citra Digital Berbasis *Chaos* dengan Penggabungan Teknik Permutasi dan Teknik Subsitusi Menggunakan *Arnold's Cat Map* dan *Logistic Map,* Prosiding Seminar Nasional Pendidikan Teknik Informatika.

[3] Ye, Ruisong., dan Ma, Yuanlin., 2013, A Secure and Robust *Image* Encryption Scheme Based on Mixture of Multiple Generalized Bernoulli Shift Maps and Arnold Maps, I.J. Computer network and Information Security.

[4] Pisarchik, Alexander N., dan Zanin, Massimiliano, 2010 Chaotic Map Cryptography and Security, Nova Science Publishers, Inc.

[5] Struss, Katherine., 2009, A Chaotic *Image* Encryption, Mathematics Senior Seminar.

[6] Kumari, S.Vani., dan Neelima, G., 2013, An Efficient *Image* Cryptographic Technique by Applying Chaotic *Logistic Map* and Arnold Cat Map, International journal of Advanced Research in Computer Science and Software Engineering.

[7] Gao, Haojiang., Zhang, Yisheng., Liang, Shuyun., dan Li, Dequn., 2006, A new chaotic algorithm for *image* encryption, *chaos*, solitions and Fractal 29 ScienceDirect.

[8] Wood, A, C., 2011, *Chaos*-Based Symmetric Key Cryptosystems, Departement of Computer Science at the Rochester Institute of Technology.

[9] Susanto, Alvin., 2008, Penerapan Teori *Chaos* di dalam Kriptografi, Informatika Institut Teknologi Bandung.

[10] Kusmarni, Yani., 2008, Teori *Chaos* Sebuah Keteraturan dalam Keacakan, Universitas Pendidikan Indonesia Bandung.

[11] Peterson, Gabriel., 1997, *Arnold's Cat Map*, Math45-linear algebra.

[12] James, Lampton., 2002, *Chaos* Cryptography: Protecting data Using *Chaos,* Missisippi School for Mathematics and Science.

[13] Huang., Xiaoling., Ye, Guodong., dan Wong, Kwok-Wo., 2013, Chaotic *Image* Encryption Algorithm Based on Circulant Operation. Hindawi Publising Corporation Abstract and Applied Analysis.

[14] Elashry, ibrahim F., Farag, Allah O S., Abbas, Alaa M., El-Rabaie, S., Abd El-Samie, Fathi E., 2009, Homomorphic *image* encryption, Journal of Electronic Imaging.

[15] Kamali, S.H., Shakerian, R., Hedayati, M., dan Rahmani, M., 2010, A new modified version of advanced encryption standard based algorithm for *image* encryption, in Electronics and Information Engineering

[16] Ahmad, Jawad., dan Ahmed, Fawed, 2012, Efficiency Analysis and Security Evaluation of *Image* Encryption Schemes, International Journal of Video and *Image* Processing and Network Security

[17] Cui, Jianjiang., Li, Siyuan., dan Xue., Dingyu, 2013, Novel Color *Image* Cryptosystem Using Chaotic Cat and Chebyshev Map, International Journal of Computer Science Issues.