APLIKASI ENKRIPSI CITRA MENGGUNAKAN ALGORITMA KRIPTOGRAFI *ARNOLD CAT MAP* Dan *LOGISTIC MAP*

Pahrul Irfan

Tenaga Pengajar STMIK Bumigora Mataram
JL. Ismail Marzuki Mataram - NTB
ip.5090@gmail.com

Abstrak

Data security in the process of information exchange is very important. One way to secure the image is to use cryptographic techniques. Cryptographic algorithms applied to the image is used to randomize the position of pixels using a secret key parameters, so that images can not be recognized anymore after the encryption process. In this study, researchers used the algorithm of chaos known as algorithms compact, fast and commonly used in cryptography especially those in the image file. The results showed the image that has been through an encryption process can not be recognized because the randomization process image pixel position is performed using chaos algorithm.

Keywords : chaos, image, image encryption, random pixel.

I. PENDAHULUAN

Informasi telah menjadi bagian terpenting dalam kehidupan manusia. Kemampuan untuk mengakses dan meyediakan informasi secara cepat dan akurat menjadi sangat penting bagi seluruh kalangan baik organisasi, perguruan tinggi, lembaga pemerintahan, perusahaan, sampai dengan individual (perorangan).

Perkembangan teknologi informasi telah membuat penyimpanan dan transmisi citra menjadi lebih mudah dan efisien. Persoalan yang timbul dari kemudahan ini adalah terdapatnya celah keamanan bagi pihak – pihak yang tidak bertanggung jawab untuk melakukan pencurian terhadap data, baik dalam proses transmisi atau yang tersimpan pada *harddrive*.

Meningkatnya jumlah pencurian data oleh *hacker* di seluruh penjuru dunia menjadikan kebutuhan sistem keamanan sangat penting diterapkan, sehingga keamanan data yang disimpan atau dikirimkan akan terjamin.

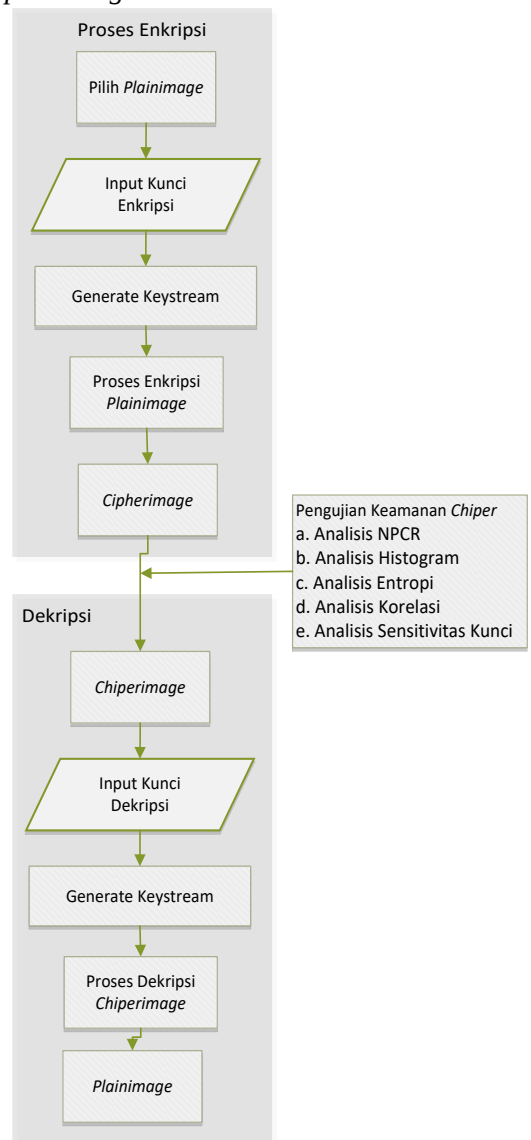
Kriptografi merupakan cara pengamanan data yang telah dikenal sejak perang dunia kedua [2]. Pada dasaranya kriptografi memiliki dua proses yaitu enkripsi dan dekripsi. Data awal disebut sebagai *plaintext* atau *plainimage*, sedangkan teknik untuk membuat data tidak dapat terbaca disebut proses *enkripsi*, kemudian dekripsi adalah teknik untuk merubah data hasil enkripsi menjadi data awal. Data yang sudah melewati tahap enkripsi disebut *cipher* [3]. Parameter yang digunakan untuk dapat melakukan proses enkripsi dan dekripsi yaitu kunci rahasia yang bisa berupa angka atau karakter khusus.

Perkembangan metode yang digunakan dalam enkripsi data sangat pesat, baik yang bersifat Simetris (enkripsi menggunakan satu kunci) atau Asimetris (enkripsi menggunakan dua kunci). Kriptografi yang bersifatsimetris antara lain DES, 3DES, IDEA, AC5, RC4, AES, Chaos. Sedangkan untuk algoritma kriptografi yang bersifat Asimetris yaitu RSA, Deffie-Hellman, DSA, ElGamal [2].

Pada penelitian ini, penulis mencoba menerapkan algoritma *Chaos* menggunakan sistem *Logistic Map* untuk melakukan enkripsi pada pixel *RGB* citra, dan ditambahkan dengan algoritma *Arnold Cat Map* yang digunakan untuk melakukan pengacakan pada posisi pixel citra. Algoritma ini dipilih karena memiliki model sederhana dan cepat dalam melakukan proses enkripsi. Dari penggabungan dua algoritma diatas, diharapkan akan didapatkan *cipherimage* yang memiliki pixel teracak sempurna dan tidak dapat dikenali lagi.

II. METODOLOGI

Metode penelitian yang dilakukan dapat dilihat pada gambar 1. Berdasarkan gambar tersebut, terdapat dua proses terpisah yaitu proses enkripsi dan proses dekripsi. Proses enkripsi dimulai dengan memasukkan gambar yang akan diamankan atau dilakukan enkripsi, kemudian memasukkan kunci rahasia, barulah proses enkripsi dapat dijalankan dan akan menghasilkan *cihperimage* atau hasil enkripsi berupa gambar acak yang tidak dapat dikenali lagi. Proses dekripsi merupakan kebalikan dari proses enrkipsi. Proses ini mengembalikan *cipherimage* menjadi gambar awal atau *plainimage*.



Gambar 1. Diagram Alur Metode Penelitian

2.1 Landasan Teori

2.1.1 *Logistic Map*

Logistic map adalah sistem chaos yang paling sederhana yang berbentuk persamaan iteratif[5,6]. Algoritma ini dapat dirumuskan sebagai berikut:

$x_i + 1 = r\, x_i (1 - x_i)$ (1) (1)

Nilai x_i yaitu antara $0 \le x_i \le 1$, $i = 0, 1, 2,$ .... dan $0 \le r \le 4$. Nilai awal (*seed*) persamaan iterasi adalah x_0. Persamaan (1) bersifat deterministik sebab jika dimasukkan nilai x_0 yang sama maka dihasilkan barisan nilai *chaotik* (x_i) yang sama pula. Oleh karena itu, pembangkit bilangan acak dengan sistem chaos disebut *pseudo-random generator*. Sifat algritma *Chaos* yang paling penting adalah sensitivitasnya pada perubahan kecil nilai awal. Artinya jika tejadi perubahan nilai kunci yang digunakan, maka hasil yang didapatkan tidak akan sama.

Nilai awal *Logistic Map (x0)* di dalam algoritma kriptografi berperan sebagai kunci

rahasia. Dengan nilai awal yang tepat sama maka proses dekripsi menghasilkan *plainteks* semula. Sayangnya nilai-nilai chaos tidak dapat langsung dioperasi-modulokan dengan plainteks karena masih berbentuk bilangan riil antara 0 dan 1. Agar barisan nilai *chaotik* dapat dipakai untuk enkripsi dan dekripsi dengan *streamcipher*, maka nilai-nilai *chaos* tersebut dikonversi ke nilai *integer*[1].

Di dalam makalah ini, konversi nilai *chaos* ke *integer* dilakukan dengan menggunakan fungsi pemotongan yang diusulkan di dalam [3]. Caranya adalah dengan mengalikan nilai *chaos (x)* dengan 10 berulangkali sampai ia mencapai panjang angka (size) yang diinginkan, selanjutnya potong hasil perkalian tersebut untuk mengambil bagian integer-nya saja. Secara matematis fungsi konversi tersebut adalah: $T(x, size) = \| x*10^{count} \|, x \neq 0$

Dalam hal ini count dimulai dari 1 dan bertambah 1 hingga x 10count> $10^{size - 1}$. Hasilnya kemudian diambil bagian integer saja. Sebagai contoh, misalkan x = 0.004276501 dan size = 4, maka dimulai dari count = 1 sampai count = 6 diperoleh $0.004276501 \times 10^6 = 4276.501 > 10^3$

Dari hasil diatas ambil bagian integer-nya, hasilnya yaitu : || 4276.501 || = 4276

2.1.2 *Arnold Cat Map*

Arnold Cat Map (ACM) ditemukan oleh Vladimir Arnold pada tahun 1960[2,4]. ACM

mentransformasikan koordinat (x, y) di dalam citra yang berukuran N × N kekoordinat baru (x', y') menggunakan persamaan iterasinya sebagai berikut :

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & a \\ b & ab+1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} mod(N) \quad (2)$$

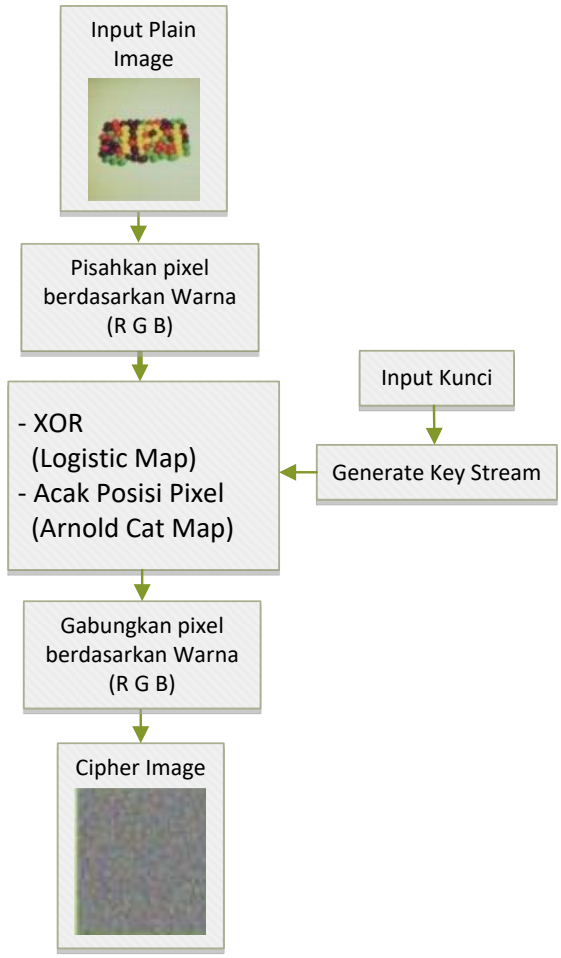
Penggunaan *modulo* dengan nilai N pada operasi ACM dimaksudkan agar nilai posisi pixel yang dilakukan pengacakan tetap pada area gambar yang ada. Karena itu, maka algoritma ACM pada dasarnya hanya dapat digunakan pada gambar dengan panjang dan lebar yang sama[5].

Seperti umumnya fungsi *chaos* yang bersifat *deterministik*, citra yang sudah teracak oleh ACM dapat direkonstruksi menjadi citra semula dengan menggunakan kunci yang sama (*a*, *b*, dan *m*). Persamaan iterasinya adalah

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} 1 & a \\ b & ab+1 \end{bmatrix}^{-1} \begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} mod(N) \quad (3)$$

2.2 Proses Enkripsi

Proses enkripsi menggunakan alogiritma *chaos* dapat dilihat pada gambar 3.



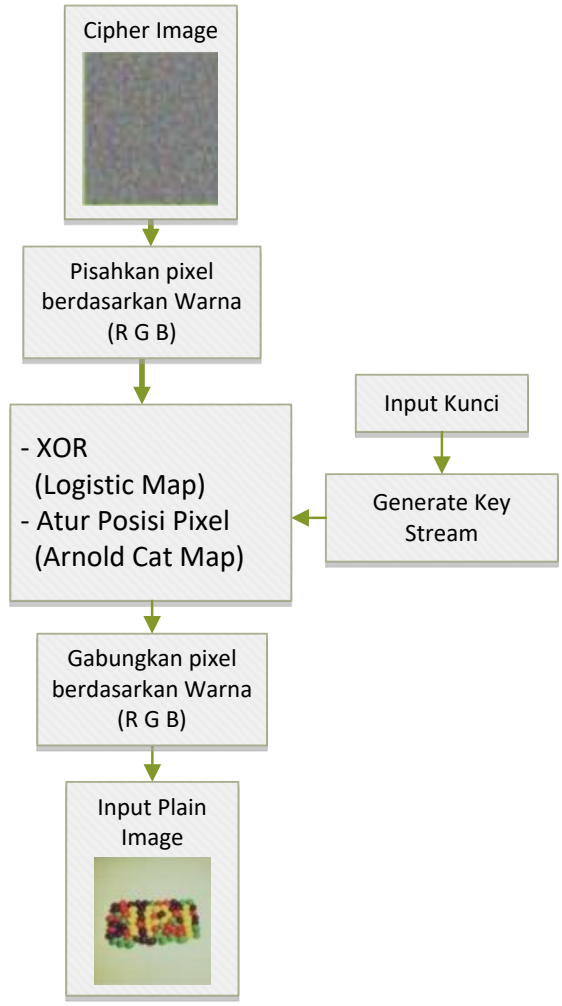
Gambar 2. Proses Enkripsi

Algoritma enkripsi ganda yang diusulkan adalah sebagai berikut:

1) Pilih citra yang akan dienkripsi (*Plain image*).
2) Input nilai awal yang merupakan variable x_0.
3) Bangkitkan kunci enkripsi menggunakan persamaan (1).
4) Lakukan proses enkripsi dengan menggunakan skema XOR untuk masing – masing komponen warna citra dengan kunci yang telah dibuat sebelumnya.
5) Lakukan pengacakan pixel menggunakan persamaan (2)
6) Hasil enkripsi dari algoritma yang pertama yaitu berupa *chiperimage*.

Hasil ahir dari seluruh proses ini adalah berupa *cipherimage.*

2.3 Proses Dekripsi



Gambar 3. Proses Deskripsi

Algoritma dekripsi citra adalah sebagai berikut :

1) Pilih *ciphertext.*
2) Selanjutnaya masukkan kunci dekripsi *Chaos* yang telah digunakan sebelumnya untuk proses enkripsi.
3) Langkah terahir adalah melakukan operasi XOR pada nilai pixel menggunakan persamaan (1) dan proses pengembalian posisi pixel ke posisi awal menggunakan persamaan (3) kemudian dihasilkan *plainimage* atau gambar semula.

2.4 Analisis Keamanan

Beberapa metode yang digunakan untuk melakukan analisa pada algoritma Chaos yang digunakan antara lain :

a. *Number of Pixel Change Rate* (*NPCR*) *Number of Pixel Change Rate* adalah perbandingan posisi pixel *gray* antara *plainimage* dengan *cipherimage*. Tujuan pengujian ini yaitu untuk menjamin bahwa pada setiap titik matriks terdapat perubahan elemen warna [7]. NPCR dirumuskan dengan:

$$NPCR = \frac{\sum_{i,j} D(i,j)}{W\ x\ H} x\ 100\% \quad (4)$$

Dengan *persyaratan* sebagai berikut :

$$D(i,j) = \begin{cases} 0, & jika\ C_1(i,j) = C_2(i,j) \\ 1, & jika\ C_1(i,j) \neq C_2(i,j) \end{cases}$$

Keterangan :

D : variable untuk menghitung banyaknya perbedaan pixel
C1 : Pixel *plainimage*
C2 : Pixel *cipherimage*
W : lebar citra
H : tinggi citra

Nilai *D(i,j)* adalah banyaknya perbedaan pixel yang dikaliakan dengan nilai 100% setelah itu dibagi dengan lebar dan tinggi dari citra sampel.

b. Analisis Sensitivitas Kunci

Sensitivitas kunci merupakan hal yang sangat penting dalam system kriptografi. Pengujian sensitivitas kunci bertujuan untuk melihat hasil dekripsi yang dilakukan menggunakan kunci yang berbeda. Jika gambar asli dapat didekripsi atau dapat terlihat mirip menggunakan kunci yang salah, maka algoritma enkripsi yang diterapkan tidak dapat digunakan.

c. Analisis Histogram

Teknik analisis histogram digunakan untuk melihat kesesuaian distribusi warna antara *plainimage* dengan *cipherimage*. Jika histogram *cipherimage* memiliki keragaman distribusi dan memiliki perbedaan yang signifikan dengan *plainimage*, maka dapat dikatakan *cipherimage* tidak memberikan petunjuk untuk melakukan *statistical attack* pada *cipherimage* yang dihasilkan.

d. Entropi

Teori informasi merupakan teori matematis dalam komunikasi data yang dikemukakan oleh Shannom pada tahun 1949 [6]. Nilai entropi ideal jika sebuah informasi dienkripsi dan dalam kondisi teracak adalah 7,99902 (~8). Dengan demikian sistem yang dirancang aman dari serangan entropi. Namun jika nilai entropi lebih kecil dari 8, dapat dikatakan sistem enkripsi masih dapat ditebak [1]. Entropi dari pesan dapat dihitung dengan rumus [8] :

$$H_e = -\sum_{k=0}^{G-1} P(k) \log_2\big(P(k)\big) \quad (5)$$

Keterangan:

He : Nilai entropi
G : Pesan atau nilai keabuan dari citra (0..255)
P(k) : Peluang kemunculan simbol ke-*k*

e. Korelasi

Analisis korelasi dan entropi digunakan untuk mengetahui nilai kualitas citra hasil enkripsi. Semakin rendah korelasi antar pixel dan semakin tinggi entropinya, maka sistem enkripsi dikatakan aman [6]. Rumus perhitungan korelasi yaitu [8] :

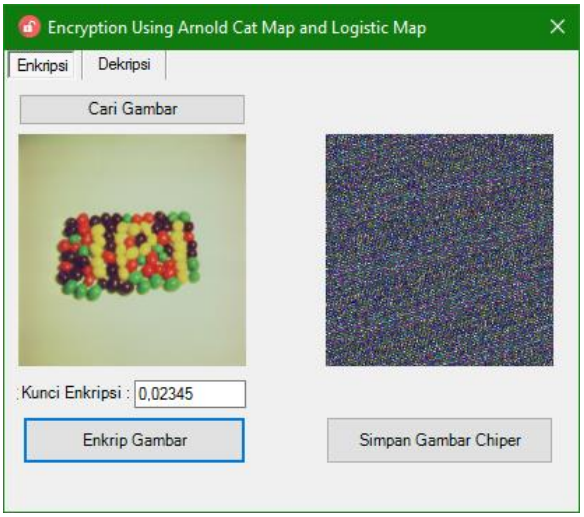
$$r = \frac{n\sum(xy) - \sum x \sum y}{\sqrt{[n\sum(x^2) - (\sum(x)^2)][n\sum(y^2) - (\sum(y)^2)]}} \quad (6)$$

Keterangan:

r : Nilai korelasi
n : Jumlah data
∑xy : Jumlah perkalian xy
∑x : Jumlah data x
∑y : Jumlah data y
∑(x2) : Jumlah kuadrat data x
∑(y2) : Jumlah kuadrat data y

III. HASIL dan PEMBAHASAN

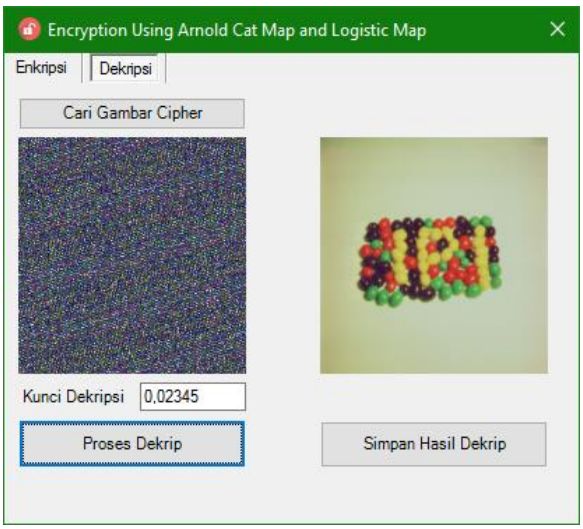
3.1 Interface Proses Enkripsi



Gambar 4. interface Proses Enkripsi

Gambar 4 diatas merupakan form yang digunakan untuk melakukan proses enkripsi gambar. Proses enkripsi dimuali dengan mencari gambar yang akan dienkripsi dengan menekan tombol cari gambar, kemudian memasukkan kunci yang akan digunakan untuk melakukan proses enkripsi, selanjutnya dilakukan proses enkripsi dengan menekan tombol enkrip gambar. Jika proses enrkipsi telah selesai, maka *chipperimage* akan muncul pada kotak sebelah kanan. *Chipper* ini nanti bisa disimpan dengan melakukan klik pada tombol simpan gambar.

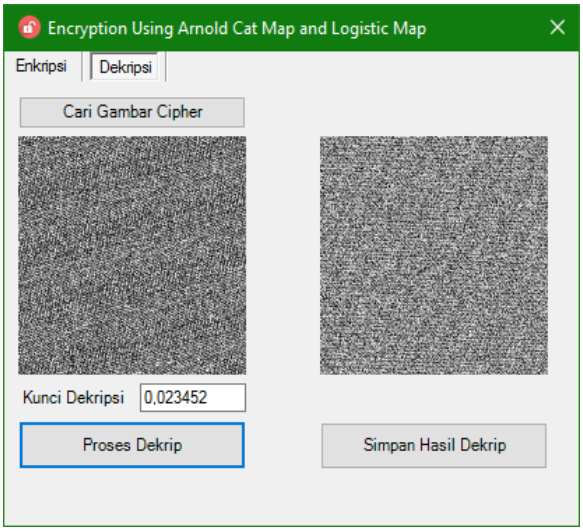
3.2 Interface Proses Dekripsi



Gambar 5. Interface Proses Dekripsi

Gambar 5 diatas merupakan form yang digunakan untuk melakukan proses dekripsi gambar. Proses dekripsi dimuali dengan mencari gambar *chipper* yang akan dilakukan proses dekripsi dengan menekan tombol cari gambar chipper, kemudian memasukkan kunci yang akan digunakan untuk melakukan proses dekripsi, selanjutnya dilakukan proses dekripsi dengan menekan tombol proses dekrip. Jika proses dekripsi telah selesai, maka gambar asli hasil dekripsi akan muncul pada kotak sebelah kanan. Gambar ini bisa disimpan dengan melakukan klik pada tombol simpan hasil dekrip.

Jika terjadi kesalahan kunci dekripsi, maka gambar tidak akan dapat dikembalikan. Hal ini dapat dilihat pada gambar 6 dibawah ini.



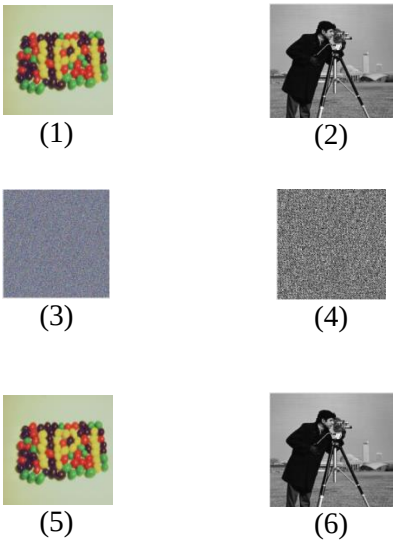
Gambar 6. Proses dekripsi menggunakan kunci berbeda

3.3 Hasil Enkripsi dan Dekripsi

Proses enkripsi dan dekripsi dimulai dengan pembangkitan kunci menggunakan persamaan (1). Dalam penelitian ini Sebagai pembangkit kunci (*keystream generator)* adalah pembangkit bilangan acak menggunakan metode *Logistic Map*. Dalam hal ini, setiap pixel gambar akan di-XOR-kan *keystream* yang dibangkitkan menggunakan persamaan (1), kemudian hasil XOR nilai pixel ini diacak posisinya menggunakan metode *Arnold Cat Map* atau persamaan (2) sehingga menghasilkan gambar yang memiliki nilai pixel dan posisi pixel yang telah berubah sempurna. Hal serupa juga digunakan pada proses dekripsi, perbedaannya terletak pada proses pengembalian posisi acak pixel yang

menggunakan persamaan (3). Untuk citra berwarna XOR nilai pixel dilakukan tiga kali, masing-masing untuk kanal *red* (*R*), *green* (*G*), dan *blue* (*B*).

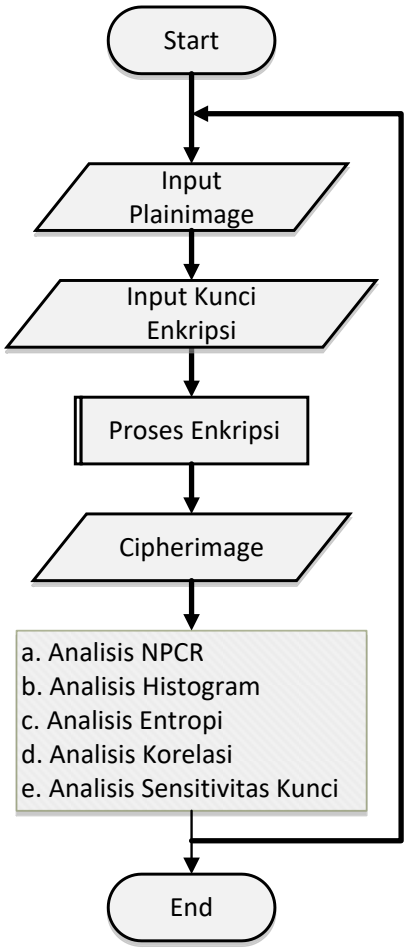
Citra hasil enkripsi menggunakan gabungan dua algoritma yang digunakan menghasilkan *cipherimage* yang dapat dilihat pada Gambar 7(3)dan 7(4). Citra hasil enkripsi terlihat sudah tidak dapat dikenali lagi.



Gambar 7. (1) *Plainimage* ‘A; (2) *Plainimage* ‘B; (3) *Cipherimage* ‘A; (4) *Cipherimage* ‘B; (5) hasil dekripsi ‘A; (6) hasil dekripsi ‘B’

3.4 Pengujian Keamanan *Chipherimage*

Analisis keamanan *chipherimager* dapat digunakan untuk mengetahui tingkat keamanan *chipherimage* terhadap beberapa percobaan yang mungkin akan dilakukan kriptanalis untuk dapat membuka citra asli. *Flowchart* pengujian keamanan dapat dilihat pada gambar 7 dibawah ini.



Gambar 8. Flowchart pengujian keamanan

Hasil dari analisis keamanan pada *chiperimage* hasil enkripsi ganda yang telah diterapkan adalah sebagai berikut :

a. *Number of Pixel Change Rate* (*NPCR*)

Number of Pixel Change Rate merupakan perbandingan posisi pixel antara *plainimage* dengan *cipherimage*. Tujuan pengujian ini yaitu untuk menjamin bahwa pada setiap titik matriks terdapat perubahan elemen warna [7]. Pengujian NPCR menggunakan persamaan 4 dengan hasil sebagai berikut :

Tabel 1. Hasil Analisis NPCR

No	**Nama File**	Ukuran	**NPCR (100%)**
1	Airplane	512x512	99,76
2	Jelly	256x256	99,74
3	Cameraman	256x256	99,69
4	Sailboat	512x512	99,44

Dari Tabel 1 terlihat rata – rata yang didapatkan untuk nilai pengujian NPCR hampir mencapai 100% atau mendekati sempurna. Hal

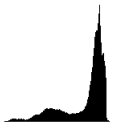
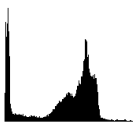
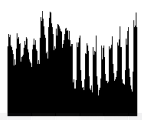
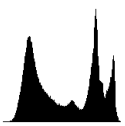
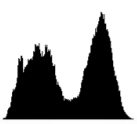
ini mengindikasikan terjadi perubahan pada matriks pixel gambar secara merata atau pada keseluruhan pixel gambar sehingga dapat diambil kesimpulan bahwa tidak terdapat kesamaan matriks atau posisi pixel antara *cipherimage* dengan *plainimage*.

b. Analisis Histogram

Histogram pada gambar menunjukkan penyebaran nilai warna yang terdapat dalam sebuah gambar. Analisis histogram digunakan untuk melihat perubahan histogram warna antara *plainimage* dengan *chipherimage*.

Jika histogram *cipherimage* memiliki keragaman distribusi dan memiliki perbedaan yang signifikan dengan *plainimage*, maka dapat dikatakan *cipherimage* yang dihasilkan tidak memberikan petunjuk bagi kriptanalis dalam mencari kesamaan nilai warna pada gambar atau biasa disebut dengan teknik *statistical attack*. Hasil analisis histogram ditunjukkan pada tabel 2 dibawah ini.

Tabel 2. Hasil Analisis histogram

No	*Plainimage*	Histogram *plainimage*	Histogram *cipherimage*
1			
2			
3			
4			

Dari tabel 2 diatas, jelas terlihat perubahan histogram yang signifikan antara *plainimage* dengan *chiperimage*. Sehingga dapat diambil kesimpulan bahwa terdapat perubahan nilai warna yang menyeluruh pada gambar asli.

c. Entropi

Nilai entropi ideal jika sebuah informasi dienkripsi dan dalam kondisi teracak sempurna adalah 7,99902 (~8). Dengan demikian sistem yang dirancang aman dari serangan entropi. Namun jika nilai entropi lebih kecil dari 8, dapat dikatakan sistem enkripsi masih dapat ditebak [1].

Tabel 3. Hasil Analisis *Entropy*

No	Nama citra	Type	Nilai Entropi
1	Airplane	Grayscale	7,943
2	Jelly	Color	7,991
3	Cameraman	Grayscal	7,923
4	Sailboat	Color	7,987

Tabel 3 memperlihatkan rata-rata nilai entropi adalah 7,932. Berdasarkan hasil penelitian Jolfae dan Mirghadri (2011) menyatakan bahwa, jika sebuah informasi dienkripsi dan dalam kondisi teracak sempurna, maka nilai entropi yang ideal adalah ≈ 8. Berdasarkan teori tersebut maka algoritma enkripsi yang dirancang ini aman dari serangan entropi atau sulit ditebak oleh kriptanalis karena nilai informasi yang terdapat didalam gambar telah berada dalam keadaan teracak sempurna.

d. Korelasi

Tabel 4. Analisis Korelasi

No	Nama citra	Type	Nilai Korelasi
1	Airplane	Grayscale	0,0023
2	Jelly	Color	0,0026
3	Cameraman	Grayscal	0,0031
4	Sailboat	Color	0,0055

Dari Tabel 4 terlihat bahwa nilai rata – rata korelasi antara *plainimage* dengan *cipherimage* adalah 0,0033. Hasil tersebut menunjukkan bahwa sistem enkripsi yang diusulkan sesuai dengan teori *perfect secrecy* yang dikemukakan oleh Shannon, yaitu semakin rendah korelasi antar piksel maka sistem enkripsi dapat dikatakan aman [6].

e. Analisis Sensitivitas Kunci

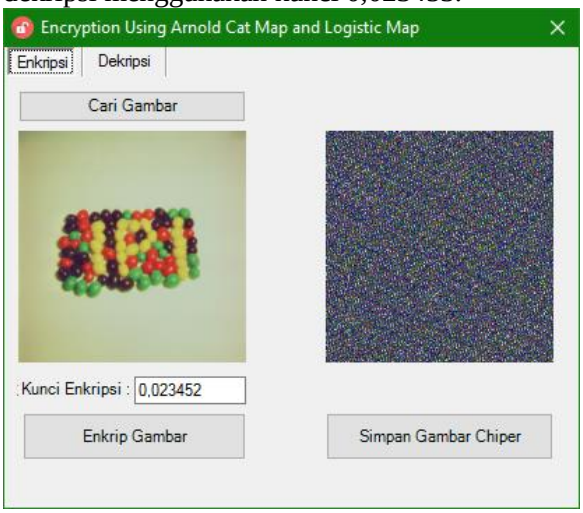
Sensitivitas kunci merupakan hal yang sangat penting dalam system kriptografi. Pengujian sensitivitas kunci bertujuan untuk melihat hasil dekripsi yang dilakukan menggunakan kunci yang berbeda.

Teori ini dapat diuji dengan cara melakukan dekripsi menggunakan kunci berbeda dengan kunci enkripsi. Kunci awal atau x_0 yang digunakan untuk enkripsi yaitu 0,023452, kemudian dilakukan perubahan

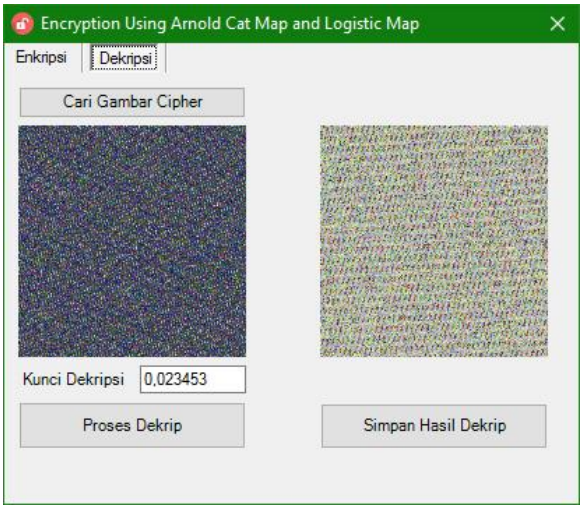
dilakukan sebagai berikut :

1) Penambahan 0,000001 sehingga x_0 = 0, 023453
2) Penambahan 0,00001 sehingga x_0 = 0, 023462

Percobaan dilakukan dengan mencoba mendekripsi *cipherimage* menggunakan kunci yang telah dirubah. Gambar 9 dibawah ini merupakan percobaan enkripsi pada file jelly.bmp menggunakan kunci 0,023452, dan pada gambar 10 merupakan percobaan dekripsi menggunakan kunci 0,023453.



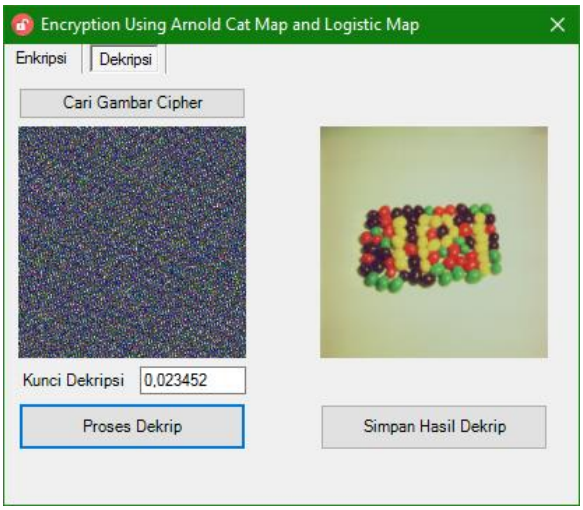
Gambar 9. Proses Enkripsi menggunakan kunci 0,023452



Gambar 10. Proses dekripsi menggunakan kunci 0,023453

Pada gambar 10 diatas, terlihat bahwa algoritma yang digunakan untuk melakukan enkripsi gambar sangat sensitif terhadap perubahan kunci yang digunakan. Hal ini tentunya semakin meningkatkan keamanan dari gambar hasil enkripsi atau *chipperimage* dikarenakan akan sulit untuk melakukan percobaan menggunakan algoritma *bruteforce*.

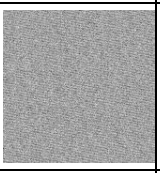
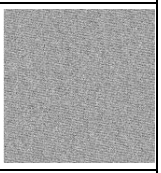
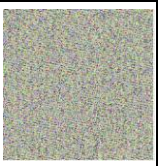
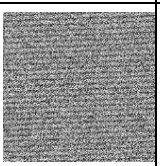
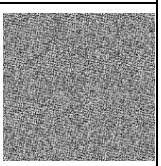
Proses dekripsi hanya dapat dilakukan dengan kunci yang tepat atau persis sama dengan kunci yang digunakan untuk melakukan enrkripsi. Hal ini dapat dilihat pada gambar 11 dibawah ini.



Gambar 11. Proses dekripsi menggunakan kunci 0,023452

Percobaan pada file lainnya dapat dilihat Tabel 5 dengan perubahan kunci dekripsi yang digunakan.

Tabel 5 Pengaruh perubahan x_0 terhadap hasil dekripsi

No	Citra Asli	Hasil Dekripsi	
		a	b
1			
2			
3			
4			

Dari Tabel 5 terbukti bahwa algoritma *Chaos* yang digunakan memiliki sensitifitas yang tinggi terhadap kunci dekripsi yang digunakan. Artinya, jika terjadi perubahan pada kunci yang digunakan untuk melakukan dekripsi *cipherimage*, walaupun hanya sebesar 0,000001, maka *plainimage* atau gambar asli tidak akan didapatkan.

IV. KESIMPULAN dan SARAN

Penelitian dan percobaan yang telah dilakukan menghasilkan kesimpulan sebagai berikut:

1) Proses enkripsi dan dekripsi dapat dilakukan dengan menggunakan dua metode enrkipsi berbeda yaitu menggunakan metode *Arnold Cat Map* dan *Logistic Map*.
2) *Chipperimage* yang dihasilkan dari algoritma gambungan akan lebih sulit dipecahkan oleh kriptanalis karena harus membuka dua algoritma enkripsi.
3) *Chipherimage* yang dihasilkan terbukti aman dari serangan kriptanalis. Hal ini dilihat dari hasil analisis keamanan yang telah dilakukan.

Saran yang dianjurkan penulis untuk pengembangan penelitian ini yaitu :

1) Diharapkan tambahkan fungsi Send pada Halaman form enkripsi agar mempermudah dalam proses pengiriman hasil enkripsi melalui email.
2) Pengembangan aplikasi agar dapat digunakan pada sistem operasi lain seperti android.
3) Penelitian selanjutnya diharapkan dapat melakukan enkripsi pada semua ukuran dan tipe gambar.

V. UCAPAN TERIMA KASIH

Peneliti mengucapkan terima kasih kepada semua pihak yang telah membantu sehingga penelitian ini dapat selesai tepat waktu.

VI. REFERENSI

[1] Jolfaei A, Mirghadri A. (2011). Image Encryption Using Chaos and Block Cipher. Computer and Information Science. 4:1.

[2] Kurniawan, Yusuf. 2004. Kriptografi Keamanan Internet dan Jaringan Komunikasi. Bandung : Informatika.

[3] Munir, R. 2006. Kriptografi. Penerbit Informatika. Bandung

[4] Munir, R. 2011. Enkripsi Selektif Citra Digital dengan Stream Cipher Berbasiskan pada Fungsi Chaotik Logistic Map. Prosiding Seminar Nasional dan Expo Teknik Elektro 2011, ISSN : 2088-9984

[5] Stallings, William. (2004). Cryptography and Network Security : Principles and Practice. Prentice-Hall, New Jersey

[6] Stinson, R, D. 2002. Cryptography Theory and Practice 2nd Edition. CRC Press Inc. Boca Raton, London

[7] Huang, Mao-Yu., Huang, Yueh-Min., Wang, Ming-Shi. 2010. Image Encryption Algorithm Based on Chaotic Map. Computer Symposium (ICS) International. IEEE Xplore, 154-158.

[8] Younes, M A B , Jantan A. (2008). "Image Encryption Using Block-Based Transformation Algorithm". IAENG International Journal of Computer Science. 35:1

[9] C. Fu, J. Chen, H. Zou, W. Meng, Y. Zhan, Y. Yu. 2012. A Chaos-based Digital Image Encryption Scheme with an improved Diffusion Strategy. Journal Optic Express 2363, Vol. 20. No.3.