Penggabungan Algoritma *Chaos* dan *Rivers Shamir Adleman* (RSA) untuk Peningkatan Keamanan Citra

Pahrul Irfan
Magister Teknik Informatika
Universitas Islam Indonesia(UII)
Yogyakarta, Indonesia
ip.5090@gmail.com

YudiPrayudi
Magister Teknik Informatika
Universitas Islam Indonesia (UII)
Yogyakarta, Indonesia
prayudi@uii.ac.id

***Abstrak*—Keamanan dan kerahasiaan data atau informasi pada masa sekarang ini telah menjadi perhatian penting. Penggunaan data citra semakin luas dalam berbagai bidang. Oleh karena itu, pengamanan data citra dari akses yang tidak berhak menjadi hal yang penting. Berbagai macam teknik untuk meningkatkan keamanan data atau informasi telah dikembangkan, salah satunya yaitu dengan teknik Kriptografi atau biasa disebut teknik enkripsi / dekripsi. Kriptografi adalah ilmu untuk menjaga keamanan pesan dengan cara mengubah data atau informasi menjadi suatu bentuk yang berbeda atau tidak dapat dikenali.**

Untuk mengimbangi kecepatan komputasi yang semakin meningkat, dibutuhkan lebih dari satu algorimta enkripsi untuk meningkatkan keamanan pada citra. Salah satu caranya adalah dengan menggunakan algoritma kriptorafi ganda untuk melakukan enrkripsi dan dekripsi. Algoritma kriptografi yang sering digunakan saat ini dan terbukti kekuatannya terhusus pada citra digital adalah Algoritma dengan sistem *Chaos*. Untuk meningkatkan keamanan pada citra maka digunakan algoritma tambahan yaitu algoritma *Rivers Shamir Adleman* (RSA) yang dikenal sebagai algorima standart dalam bidang kriptografi.

Penelitian ini bertujuan untuk mengoptimalkan keamanan citra format *bitmap* dengan cara menggabungkan dua algoritma kriptografi yaitu algoritma *Chaos* dan algoritma RSA pada satu aplikasi sehingga diharapkan kekuatan dari *ciphertext* yang dihasilkan sulit untuk dipecahkan.

Keywords—chaos, cipher text, citra bitmap, enkripsi citra, RSA.

I. PENDAHULUAN

Perkembangan teknologi informasi telah membuat penyimpanan dan transmisi media digital seperti citra dan video 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 yang tersimpan dalam *harddrive* atau yang ditransmisikan.

Salah satu tipe file yang banyak digunakan dan biasanya berisi informasi penting adalah data bertipe gambar atau citra digital. Saat ini citra telah digunakan pada hampir segala bidang seperti rancangan keamanan, ilmu medis, ilmu teknikmesin, arsitekur bangunan, hasil karya seni, iklan, pendidikan dan lain sebagainya.

Citra yang disimpan atau ditransmisikan dalam bentuk *plainimage* rentan terhadap penyadapan atau pencurian, sehingga informasi penting yang terdapat dalam citra dapat diakses oleh pihak - pihak yang tidak bertanggung jawab. Salah satu contoh pentingnya pengamanan citra adalah pada citra rancangan model bangunan atau rancangan produk yang dibuat oleh suatu perusahaan. Jika citra tersebut dapat diakses oleh orang yang tidak berhak, tentunya perusahaan akan mendapat kerugian baik dari segi finansial atau yang lainnya. Oleh karena itu pengamanan terhadap citra menjadi perhatian penting untuk melindungi informasi yang terdapat di dalamnya.

Enkripsi merupakan salah satu teknik pengamanan data yang dapat memberikan layanan untuk memenuhi beberapa aspek keamanan antara lain [3]:

- *Authentication* merupakan layanan yang berhubungan dengan identifikasi kebenaran sumber pesan.
- *Nonrepudiation* mencegah penyangkalan yang dilakukan oleh salah satu pihak yang mengirimkan pesan.
- *Authority* pencegahan modifikasi pesan oleh orang yang tidak berwenang
- *Integrity* layanan untuk dapat memeriksa apakah pesan yang dikirim telah dimodifikasi atau tidak.

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 bersifat *simetris* antara lain DES, 3DES, IDEA, AC5, RC4, AES, Chaos. Sedangkan untuk algoritma kriptografi yang bersifat *Asimetris* yaitu RSA, Deffie-Hellman, DSA, ElGamal [3].

Salah satu algoritma enkripsi yang populer digunakan pada citra adalah enkripsi berbasis *Chaos*. Sistem *Chaos* yang digunakan bervariasi, antara lain *Logistic Map, Baker Map, Arnold Cat Map*, dan lain-lain. *Logistic Map* merupakan salah satu sistem enkripsi bersifat *Chaos* yang sederhana tetapi menghasilkan perhitungan yang kompleks, membutuhkan waktu proses yang singkat, tidak memiliki periode

perulangandan memiliki kepekaan terhadap nilai input awal [5]. Karena beberapa kelebihan algoritma *Chaos* dalam enkripsi citra, 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, sehingga diharapkan akan didapatkan *cipherimage* yang memiliki pixel teracak sempurna dan tidak dapat dikenali lagi.

Menurut Kromodimoeljo [4], untuk menghasilkan algoritma enkripsi citra yang tangguh dari serangan kriptanalis tidak cukup hanya dengan menggunakan satu algoritma enkripsi. Hal ini dikarenakan kemampuan komputasi yang semakin meningkat, sehingga dibutuhkan algoritma tambahan untuk membuat hasil enkripsi yang tahan terhadap serangan atau kriptanalis.

Rivers Shamir Adleman (RSA) merupakan sistem kriptografi asimetris terpopuler. Bisa dikatakan semua standard protokol kriptografi menggunakan algoritma RSA, termasuk SSL/TLS (untuk pengamanan http) dan SSH (*secure shell*) [4]. Keamanan algoritma RSA terletak pada sulitnya memfaktorkan bilangan yang besar menjadi bilangan faktor – faktor prima yang digunakan sebagai kunci *private* untuk membuka *chipertext* [1]. Karena kelebihan algoritma RSA yang merupakan algoritma standart untuk kriptografi baik pada enkripsi text atau tipe data lainnya, maka penulis mencoba untuk mengimplementasikan algoritma ini pada citra digital.

Di dalam penelitian ini diusulkan algoritma enkripsi ganda untuk citra digital dengan menggunakan algoritma *Chaos* dan *RSA*. Pemilihan algoritma *Chaos* karena dapat menghasilkan bilangan acak yang sensitif terhadap kondisi awal, membutuhkan waktu proses yang singkat dan tidak memiliki periode perulangan, sedangkan *RSA* dipilih karena *RSA* merupakan algoritma standar untuk protokol kirptografi. Sehingga dari penggabungan kedua algoritma tersebut diharapkan dapat meningkatkan keamanan *cipher text* dari serangan kriptanalis dan dapat menjawab tantangan perkembangan komputasi yang semakin cepat.

II. STUDI PUSTAKA

A. Chaos dan Logistic Map

Logistic map adalah sistem chaos yang paling sederhanayang berbentuk persamaan iteratif sebagai berikut:

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

Nilai $x_i$yaitu antara $0 \le x_i \le 1$, $i = 0, 1, 2, \ldots$ 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* (x_0)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*[6].

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:

$$T(x, size) = \| x * 10^{count} \|,\ x \neq 0 \qquad (2)$$

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 :

$$\| 4276.501 \| = 4276$$

B. Chaos dan Arnold Cat Map

Arnold Cat Map (ACM) ditemukan oleh Vladimir Arnold pada tahun 1960. Ketika melakukan penelitan, Arnold menggunakan sebuah gambar kucing dalam melakukan percobaan, sehingga algoritma hasil penelitian yang dilakukan dinamakan dengan *Arnold Cat Map (ACM)* [6]. ACM mentransformasikan koordinat (x, y) di dalam citra yang berukuran N × N kekoordinat baru (x', y') menggunakan persamaan iterasinya sebagai berikut :

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

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.

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

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} 1 & a \\ b & ab+1 \end{bmatrix}^{-1} \begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} mod(N) \qquad (4)$$

C. Rivers Shamir Adleman (RSA)

Dari sekian banyak algoritma kriptografi kunci-publik yang pernah dibuat, algoritma yang paling popular adalah algoritma RSA. Algoritma ini melakukan pemfaktoran bilangan yang sangat besar. Oleh karena alasan tersebut RSA dianggap aman.

Untuk pembangkitan pasangan kunci RSA, digunakan algoritma sebagai berikut [8] :

1) Pilih dua buah bilangan prima sembarang yang besar, *p* dan *q*. Nilai *p* dan *q* harus dirahasiakan.
2) Hitung nilai *n* dari rumus, $n = p \times q$. Besaran *n* tidak perlu dirahasiakan.
3) Hitung nilai *n* menggunakan teorema *euler* dengan rumus,$n = (p - 1)(q - 1)$
4) Pilih sebuah bilangan bulat sebagai kunci publik (e), yang relatif prima terhadap n. e relatif prima terhadap n artinya faktor pembagi terbesar keduanya adalah 1, secara matematis disebut *gcd (e,n) = 1*. Untuk mencarinya dapat digunakan algoritma *Euclid*
5) Hitung kunci privat yang disebut d sedemikian hingga agar $(d * e) \bmod n = 1$. Untuk mencari nilai *d* yang sesuai dapat juga digunakan algoritma *Extended Euclid*.Maka hasil dari algoritma tersebut diperoleh :

   kunci publik = (*e,n*)

   kunci privat = (*d,n*)

Nilai dari n bersifat publik, dan diperlukan pada perhitungan enkripsi/dekripsi. Fungsi enkripsi dan dekripsinya dijabarkan dalam fungsi berikut :

$$C = M^e \bmod n \text{ (fungsi enkripsi)} \quad (5)$$

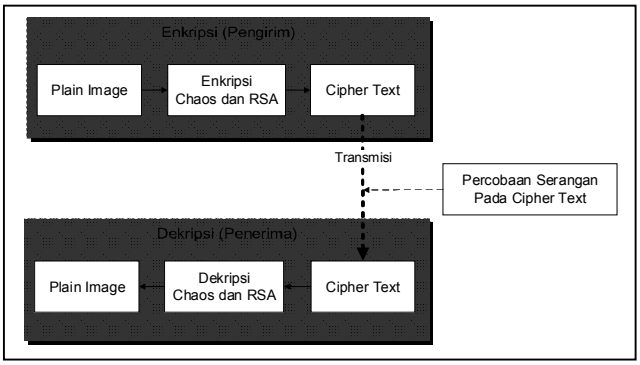
$$M = C^d \bmod n \text{ (fungsi dekripsi)} \quad (6)$$

Dimana :

C = *Cipherteks*
d =Kunci *privat*
M = Message / *Plainteks*
n = Modulo pembagi
e = Kunci publik

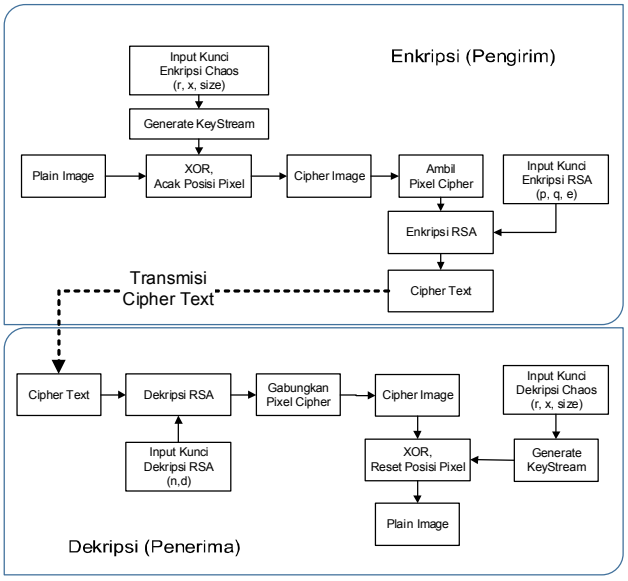
III. METODOLOGI

Rancangan yang akan digunakan dalam penelitian ini digambarkan pada Gambar 1



Gambar 1 Rancangan Skema Enkripsi Ganda

Skema di atas diuraikan kembali pada gambar 3.2



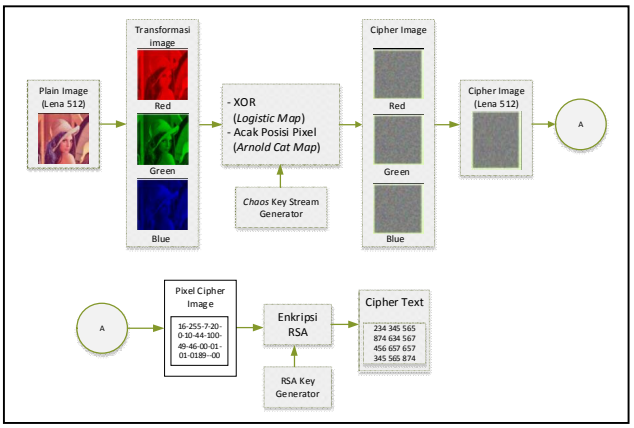
Gambar 2 Rancangan Enkripsi dan Deksirpsi

A. *Rancangan Algorima 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 (3)
6) Hasil enkripsi dari algoritma yang pertama yaitu berupa *chiperimage.*
7) Selanjutnya pada *cipherimage* dilakukan ekstraksi untuk mengambil seluruh bit pixel yang ada untuk kemudian dilakukan enkripsi tahap kedua algortima RSA.
8) Untuk algotima RSA langkah pertama yang dilakukan adalah memasukkan tiga buah nilai variable yaitu *p, q* dan *e* yang merupakan pembentuk kunci untuk melakukan enkripsi.
9) Kemudian dilanjutkan dengan proses enkripsi pada seluruh bit pixel citra yang telah diekstraksi sebelumnya menggunakan persamaan (5).
10) Hasil ahir dari seluruh proses ini adalah berupa *ciphertext*dari nilai pixel.

Proses enkripsi digambarkan pada Gambar 3.3



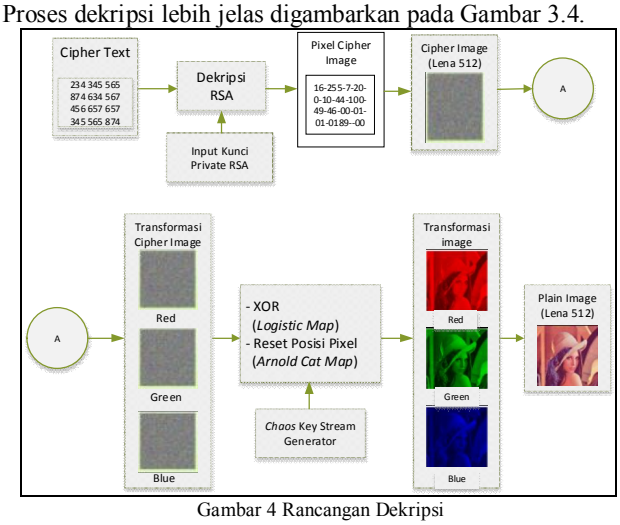
Gambar 3 Rancangan enkripsi

B. *Rancangan Algorima Dekripsi*

Algoritma dekripsi citra adalah sebagai berikut:

1) Pilih *ciphertext.*
2) Masukkan kunci *private (n, d)* untuk melakukan operasi dekripsi tahap pertama yang menggunakan algoritma RSA menggunakan persamaan (6).
3) Dari langkah kedua didapatkan nilai pixel citra yang tidak terenkripsi yang kemudian akan disatukan kembali menjadi *cipherimage*.
4) Selanjutnaya masukkan kunci dekripsi *Chaos*yang telah digunakan sebelumnya untuk proses enkripsi.
5) Langkah terahiradalah melakukan operasi XOR dan proses pengembalian posisi pixel ke posisi awal menggunakan persamaan (4) kemudian dihasilkan *plainimage* atau gambar semula.

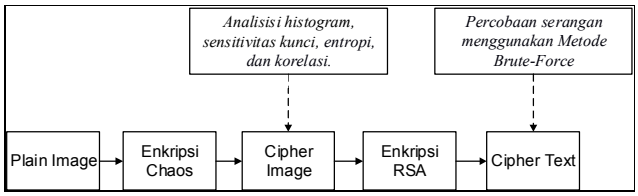
Proses dekripsi lebih jelas digambarkan pada Gambar 3.4.



Gambar 4 Rancangan Dekripsi

C. *Rancangan Analisa dan Pengujian Algoritma*

Pengujian algoritma enkripsi ganda digambarkan pada skema pengujian dibawah :



Gambar 5 Rancangan Analisa dan Pengujian Algoritma

Pengujian Algoritma enkripsi dibagi menjadi dua tahap secara paralel yaitu:

1) Pengujian Enkripsi Berbasis *Chaos*
   Pada pengujian enkripsi berbasis *Chaos* dilakukan dengan menganalisa beberapa parameter yang menjadi acuan untuk mengetahui keamanan enkripsi citra. Parameter yang digunakan antara lain Analisis Korelasi Analisa Entropi, Analisa Histogram dan analisis sensitivitas kunci.
2) Pengujian Enkripsi Berbasis *Chaos* dan *RSA*
   Hasil ahir dari algoritma yang akan digunakan adalah berupa *cihpertex* yang kemungkinan serangan yang paling efektif digunakan yaitu serangan *brute-force*. Percobaan dilakukan dengan melihat lebar kunci yang ada setelah terjadi proses enkripsi dua tahap. Jika kunci yang digunakan cupuk lebar, maka tentunya citra yang terenkripsi dapat dikatakan aman.

IV. HASIL DAN PEMBAHASAN

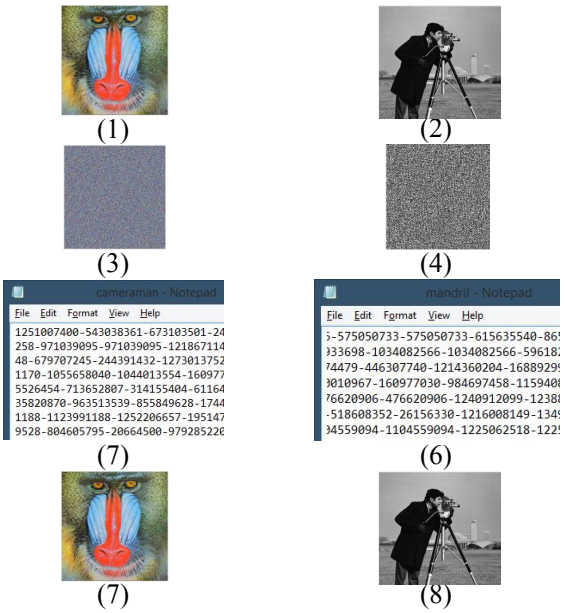
Percobaan pada penelitian ini menggunakan perangkat Visual Studio 2012. Citra uji yang digunakan dipilih dari *grayscale* dan citra berwarna. Dua buah citra uji yang dipakai adalah 'mandril' dengan dimensi 512x512pixel dan ukuran file 768 KB yang ditunjukkan pada gambar 5(1) dan citra 'cameraman' dengan dimensi 256x256 pixel dan ukuran 37 KB yangditunjukkan pada gambar 5(2). Keduanya merupakan citra standard yang biasa digunakan dalam penelitian tentang pengolahan citra. Kunci yang dipakai di dalam percobaan adalah: x_0 = 0,05678912, a = 78, b = 91, m = 2, p =11, q= 17 dan e = 7 .

Dalam penelitian ini, waktu enkripsi/dekripsi yang dibutuhkan tidak diukur karena banyaknya faktor yang menyebabkan perubahan waktu proses seperti optimasi pemrograman, perangkat keras yang digunakan,dansebagainya.

Pengujian kemaanan dan analisis hasil enkripsi dilakukan dalam dua tahapan yaitu pada masing – masing algoritma enrkipsi secara paralel. Hal ini dimaksudkan untuk melihat tahapan yang harus dilalui oleh seorang kriptanalis jika ingin memecahkan algoritma yang digunakan. Untuk analisis hasil enkripsi dari algoritma *chaos* dilakukan dengan melakukan analisa pada histgram *cipherimage*, sensitifitas kunci dari algoritma, dan analisa statistik menggunkan parameter korelasi dan entropi. Sedangkan pada tahapan selanjutnya, analisis keamanan pada *ciphertext* dengan cara melihat ruang kunci yang tersedia jika dilakukan penyerangan menggunakan metode *brute-force.*

A. Hasil Enkripsi dan Dekripsi

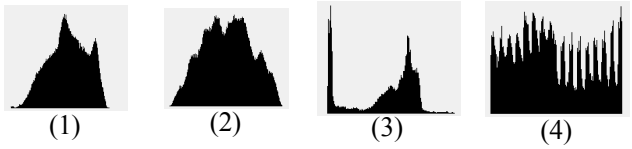
Hasil enkripsi dari alogitma yang digunakan berupa *ciphertext*, tetapi untuk memperjelas proses yang terjadi, maka penulis menampilkan hasil enkripsi pada masing – masing algoritma. Citra hasil enkripsi menggunakan algoritma *chaos* yaitu berupa *cipherimage* diperlihatkan pada Gambar 5(2)dan 5(3). Citra hasil enkripsi terlihat sudah tidak dapat dikenali lagi. Sedangkan hasil ahir yaitu berupa *ciphertext*diperlihatkan pada gambar 5(5) dan 5(6). Untuk hasil dekripsi diperlihatkan pada gambar 5(7) dan 5(8).



Gambar 5. (1)*Plainimage* 'mandril'; (2)*Plainimage* 'cameraman'; (3)*Cipherimage* 'mandril'; (4)*Cipherimage* 'cameraman'; (5)Ciphertext 'mandril'; (6)*Ciphertext* 'cameraman'; (7) hasil dekripsi 'mandril';(8) hasil dekripsi 'cameraman'

B. AnalisisHistogram

Analisis histgram dilakukan untuk melihat kekuatan dari *cipherimage* yang dihasilkan dari serangan analisis statistik. Serangan ini menggunakan histogram untuk menganalisa kemunculan *pixel* dalam sebuah citra, dengan cara mendedukasi kunci atau pixel dalam citra [6]. Untuk menghindari serangan semacam ini, maka histogram *cipherimage*harus berbeda atau tidak memiliki kemiripan secara statistik dengan histogram *plainimage*. Pada gambar 6(1) diperlihatkan histgoram dari citra 'mandril' sedangkan gambar 6(2) memperlihatkan histgram dari *cipherimage*dari citra 'mandril'. Gambar 6(3) merupakan histgram *plainimage* dari gambar 'cameraman' dan gambar 6(4) merupakan histogram dari *cipherimage* yang dihasilkan.



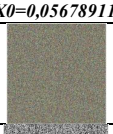
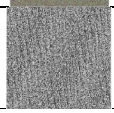
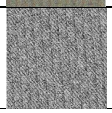
Gambar 6. (1)Histogram *plainimage* 'mandril'; (2)Histogram *cipherimage* 'mandril'; (3)Histogram *plainimage* 'cameraman'; (4)Histogram*cipherimage* 'cameraman';

Dari Gambar 6, terlihat bahwa histogram *cipher-image* yang berbeda signifikan dengan histogram *plain-image*. Hal ini tentunya menyulitkan kriptanalisuntuk melakukan serangan dengan menggunakan analisis frekuensi kemunculan nilai-nilai *pixel*, sebab secara statistik keduanya tidak memiliki hubungan.

C. Analisis sensitivitas kunci

Sensitivitas kunci merupakan hal yang sangat penting dalam system kriptografi. Pada enkripsi berbasis chaos perubahan kecil pada kunci mengakibatkan hasil dekripsi yang berbeda jauh dari gambar asli. Pengujian pengaruh perubahan kunci dimaksudkan untuk melihat hasil dekripsi jika menggunakan kunci yang salah. Pengujian ini bertujuan untuk melihat kekuatan *cipherimage* jika dilakukan percobaan dekripsi menggunakan kunci yang berbeda. Tabel 4.1 menunjukkan pengujian yang dilakukan terhadap perubahan nilai awal (x_0). Nilai $x_0$awal yang digunakan adalah 0,05678912

TABLE I. PENGARUH PERUBAHAN X_0 TERHADAP HASIL DEKRIPSI

Citra Asli	Hasil Dekripsi		
	Perbuahan 0,00000001 X0=0,05678911	***Perbuahan 0,0000001 X0=0,05678922***	***Perbuahan 0,000001 X0=0,05678022***
			
			

D. Analisis Statistik

Untuk melakukan analisis statistik digunakan dua parameter pengujian yaitu analisis korelasi dan analisis nilai entropi.Penghitungan korelasi dan entropi dilakukanuntuk menilai kualitas citra hasil enkripsi.Semakin rendah korelasi antar pixel dansemakin tinggi entropinya, maka sistemenkripsi dapat dikatakan aman dari serangan entropi (younes 2008). Sedangkan untuk statistik dari ukuran *ciphertext* yang dihasilkan, maka disertakan hasil perubahan ukuran *plainimage* setelah dilakukan enkripsi.

Analisis korelasi bertujuan untuk melihat adanya keterkaitan antara dua objek, dalam hal ini antara *plainimage* dengan *cipherimage*. Nilai korelasi berada dalam rentan 0 sampai 1, jika bernilai 1 maka kedua objek memiliki kesamaan, sedangkan jika nilai korelasi mendekati 0, maka tidak ada kesamaan antara kedua objek [9]. Untuk menghitung korelasi digunakan rumus [10] :

$$r = \frac{n\sum(xy) - \sum x \sum y}{\sqrt{[n\sum(x^2) - (\sum(x)^2)][n\sum(y^2) - (\sum(y)^2)]}} \quad (7)$$

Dimana :

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

Nilai entropi ideal adalah 7,99902 (≈8).Dengan demikian sistem enkripsi yangdirancang aman dari serangan entropi

[2].Entropi dari pesan dapat dihitung menggunakan rumus [10]:

$$H_e = -\sum_{k=0}^{G-1} P(k) \log_2(P(k)) \quad (8)$$

Dimana :

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

TABLE II. HASIL UJI STATISTIK

Nama File	Ukuran *Plainimage*	Ukuran *Ciphertext*	Nilai Statistik	
			Nilai Korelasi	*Nilai Entropi*
Mandril	768 KB	1.237 KB	0,091281	7,7706
Cameraman	37 KB	73 KB	0,004590	7,9771

Pada tabel II, nilai korelasi antara *plainimage* dengan *cipherimage* bernilai 0,091281 dan 0,004590. Karena nilai korelasinya mendekati nol maka dapat diketahui bahwa tidak terdapat keterhubungan antara *plainimage* dan *cipherimage*. Sedangkan nilai entropi adalah 7,7706 dan 7,9771. Terlihat disini nilai *entropi* mendekati nilai ideal seperti yang dikemukakan oleh Jolfae dan Mirghadri [2] sehingga algoritma yang digunakan dapat dikatakan aman dari serangan *entropi*.

Perubahan ukuran dari *plainimage* ke *ciphertext* terlihat sangat signifikan.Ukuran *ciphertext* yang dihasilkan hampir dua kali lipat dari ukuran *plainimage*. Hal ini bergantung pada panjang kunci yang digunakan pada algoritma RSA sehingga semakin panjang kunci yang digunakan, semakin besar ukuran file *cipher* yang dihasilkan.

E. Analisis ruang kunci

Parameter kunci yang digunakan pada algoritma yang digunakan yaitu *a, b, m, x_0,r, n dan e*. Kunci *a, b* dan *m* merupakan nilai kunci yang digunakan pada algoritma *Arnold Cat Map* yang digunakan untuk melakukan pengacakan nilai pixel. Sedangkan nilai x_0 dan *r* digunakan pada algoritma *Logistic Map* yang digunakan untuk membangkitkan nilai *chaos* dan kunci n dan e merupakan kunci dekripsi untuk algoritma *RSA*. Nilai *a, b* dan *m* merupakan nilai *integer* positif sehingga kemungkinan panjang kunci yang digunakan adalah $2^{32} = 4.3 \times 10^9$ begitu pula dengan kunci *n* dan *e* menggunakan integer positif. Sedangkan untuk kunci x_0 dan *r* bernilai *double-precision* atau menggunakan *standart floating-point IEEE* yaitu 10^{15}[7]. Sehingga kemungkinan ruang kunci yang ada adalah:

$H(p, q, m, x_0, r, n, e) \approx (4.3 \times 10^9)^5 \times (10^{15})^2 \approx 21.5 \times 10^{75}$

V. KESIMPULAN

Penelitian dan percobaan yang telah dilakukan menghasilkan kesimpulan sebagai berikut:

1) Proses enkripsi dan dekripsi dapat dilakukan dengan menggunakan dua algoritma enrkipsi berbeda yaitu menggunakan algoritma kriptografi *simetris(Chaos)* dan *asimetris (RSA)*.

2) Penggabungan dua algoritma kriptografi akan menghasilkan *ciphertext* yang sulit untuk dipecahkan, karena kriptanalis harus menggunakan dua tahap pencarian untuk memecahkan *ciphertext*.

3) Dari hasil analisis serangan yang dilakukan pada kedua algoritma enrkispsi yang digunakan. Faktor keamanan pada algoritma enkripsi pertama ditunjukkan oleh hasil analisis statistik yang menunjukkan bahwa rata-rata nilai entropi yang mendekati sempurna yaitu 7,849 dan nilai korelasi antara *plainimage* dengan *cipherimage* rata-rata bernilai 0,0479355 atau mendekati 0 sehingga diketahui tidak ada keterkaitan antara *cipherimage* dengan *plainimage*. Hasil pengujian juga membuktikan sensitivitas dari perubahan nilai awal *chaos (X_0)* yang memberikan keamanan dari serangan *exhaustive attack* sedangkan histogram yang dihasilkan dari *cipherimage* berbeda jauh dari histogram *plainimage* sehingga aman dari serangan yang bersifat statistik. Hasil analisa panjang kunci juga menunjukkan besarnya rentan kunci yang digunakan sehingga akan menyulitkan dalam melakukan serangan *brute-force*.

REFERENCES

[1] Awad, A. dan Saadane, A. 2010. "New Chaotic Permutation Methods for Image Encryption." IAENG International Journal of Computer Science.

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

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

[4] Kromodimoeljo, Sentot. 2010. Teori dan Aplikasi Kriptografi. Jakarta : SPK IT.

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

[6] 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.

[7] Munir, R. 2012. "Algoritma Enkripsi Citra dengan Kombinasi Dua Chaos Map dan Penerapan Teknik Selektif Terhadap Bit-bit MSB." Prosiding Seminar Nasional dan Aplikasi Teknologi Informasi (SNATI). Universitas Islam Indonesia.

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

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

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