ANALISIS KEAMANAN ALGORITMA ENKRIPSI CITRA DIGITAL MENGGUNAKAN KOMBINASI DUA *CHAOS MAP* DAN PENERAPAN TEKNIK SELEKTIF

Rinaldi Munir[1]
[1]Sekolah Teknik Elektro dan Informatika (STEI), Institut Teknologi Bandung (ITB)
Jalan Ganesha 10, Bandung 40132
E-mail: rinaldi-m@stei.itb.ac.id

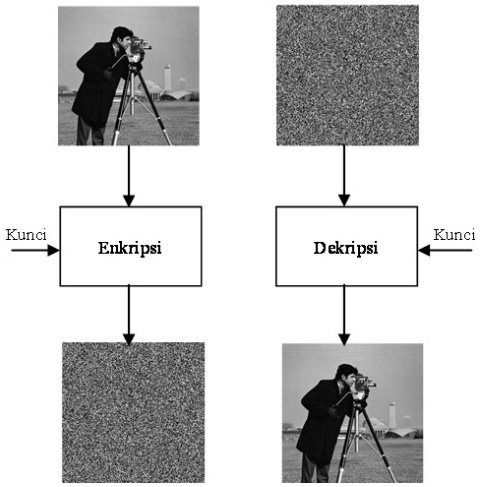
ABSTRAK

Di dalam makalah ini dipresentasikan analisis keamanan dari sebuah usulan algoritma enkripsi citra yang berbasis chaos dan penggunaan teknik enkripsi selektif untuk mengurangi volume komputasi. Algoritma enkripsi menggunakan Arnold Cat Map untuk mengacak citra, selanjutnya teknik enkripsi selektif diterapkan dengan memilih hanya empat bit MSB dari setiap pixel untuk di-XOR-kan dengan keystream yang dibangkitkan dari Logistic Map. Analisis keamanan meliputi analisis ruang kunci, analisis histogram, analisis korelasi, analisis entropi, dan analisis sensitivitas. Secara keseluruhan dapat dinyatakan bahwa analisis keamanan terhadap algoritma enkripsi citra yang ditinjau menunjukkan bahwa algoritma tersebut aman dari berbagai serangan yang bertujuan untuk menemukan kunci atau pixel-pixel di dalam plain-image

***Kata Kunci**:* Enkripsi Citra, *Chaos*, Selektif, Analisis Keamanan.

1. PENDAHULUAN

Enkripsi citra merupakan teknik untuk melindungi kerahasiaan citra dari pengaksesan ilegal. Enkripsi diperlukan karena dalam era digital sekarang ini citra digital mudah disimpan atau ditransmisikan melalui saluran publik seperti internet. Pengiriman citra melalui saluran publik rawan terhadap penyadapan, dan penyimpanan citra di dalam media *storage* rawan terhadap pengaksesan oleh pihak-pihak yang tidak memiliki otoritas. Enkripsi menyandikan citra (*plain-image*) ke bentuk visual lain yang tidak bermakna (*cipher-image*). Gambar 1 memperlihatkan diagram enkripsi-dekripsi citra digital.



Gambar 1. Diagram enkripsi-dekripsi citra digital

Mengenkripsi citra dengan algoritma kriptografi konvensional yang khusus untuk pesan teks (*DES*, *AES*, *Blowfish*, *RC4*, *RSA*, dan lain-lain) tidak mangkus. Hal ini disebabkan karena sebuah citra umumnya bervolume relatif sangat besar dibandingkan dengan data tekstual, sehingga proses komputasinya memakan waktu yang lama. Untuk kebutuhan aplikasi yang *real-time* seperti *teleconference*, *live video streaming*, dan lain-lain, jelas algoritma konvensional kurang cocok untuk mengenkripsi citra.

Selain masalah volume data, karakteristik citra yang membedakannya dengan data tekstual adakah korelasi *pixel-pixel* di dalamnya. Di dalam citra sebuah *pixel* berkorelasi erat dengan delapan *pixel* tetangganya. Proses enkripsi seharusnya membuat *pixel-pixel* yang bertetangga tidak lagi berkorelasi sehingga menyulitkan penyerang melakukan analisis statistik.

Karena setiap jenis data mempunyai karateristik yang unik, maka diperlukan algoritma enkripsi yang khusus untuk data tersebut. Sebuah algoritma enkripsi khusus citra digital telah diusulkan [1]. Algoritma tersebut berbasis *chaos* dan menggabungkan pendekatan selektif.

Chaos menjadi topik yang atraktif di dalam kriptografi karena tiga alasan: (1) sensitivitas terhadap kondisi awal, (2) berkelakuan acak, dan (3) tidak memiliki periode berulang. Penerapan *chaos* di dalam kriptografi dapat menghasilkan efek *diffusion* seperti yang dinyatakan oleh Shanon [2]. *Chaos* di dalam algoritma kriptografi umumnya digunakan sebagai pembangkit bilangan acak. Bilangan-bilangan acak itu digunakan sebagai *keystream* (dengan operasi XOR sederhana) atau

untuk mengacak susunan *pixel* di dalam citra. Barisan bilangan acak dibangkitkan dengan sebuah fungsi *chaos* (*map*). Xiang [3] menggunakan *Tent Map* sebagai pembangkit kunci enkripsi, Struss [4] dan Yu [5] menggunakan *Arnold Cat Map* untuk mengacak *pixel-pixel*. Hal yang sama juga dilakukan oleh Jolfaei [6] tetapi menggunakan *Henon Map* untuk permutasi *pixel-pixel* sebelum dienkripsi dengan *stream cipher*, sedangkan Fu [7] mengkolaborasikan *Chebysev Map* sebagai pembangkit *keystream*.

Adapun pendekatan selektif artinya hanya mengenkripsi sebagian elemen di dalam citra namun efeknya keseluruhan citra terenkripsi. Tujuan algoritma enkripsi selektif adalah mereduksi volume komputasi selama proses enkripsi dan dekripsi sehingga cocok diterapkan untuk kebutuhan aplikasi yang *real-time*.

Algoritma yang diusulkan di dalam [1] menggabungkan penggunaan dua buah fungsi *chaos* yaitu *Arnold Cat Map* dan *Logistic Map*. *Arnold Cat Map* (*ACM*) digunakan untuk mengacak susunan *pixel-pixel*, sedangkan *Logistic Map* digunakan sebagai pembangkit *keystream*. Untuk menghemat volume komputasi selama proses enkripsi/dekripsi, teknik enkripsi selektif yang diusulkan di dalam [3] diterapkan dengan hanya meng-XOR-kan *keystream* dengan bit-bit *MSB* yang berperan menentukan persepsi visual terhadap obyek di dalam citra.

Di dalam makalah ini dipresentasikan analisis keamanan algoritma enkripsi citra yang diusulkan di dalam [1]. Analisis keamanan meliputi analisis ruang kunci, analisis histogram, analisis korelasi, analisis entropi, dan analisis sensitivitas.

2. USULAN ALGORITMA

Algoritma enkripsi citra yang ditinjau di makalah ini dapat digunakan untuk mengenkripsi citra *grayscale* maupun untuk citra berwarna. Secara garis besar algoritma enkripsi terdiri dari dua bagian. Pertama: pengacakan *pixel-pixel* citra dengan *ACM*. Kedua: enkripsi *stream cipher*, yaitu operasi XOR antara 4-bit *MSB* dari setiap *pixel* dengan 4-bit *keystream*.

Algoritma dekripsi merupakan kebalikan dari enkripsi, dimulai dengan langkah kedua terlebih dahulu kemudian langkah pertama.

2.1 Pembangkitan *Keystream*

Keystream dibangkitkan dengan *Logistic Map* yang memiliki persamaan

$$x_{i+1} = r\, x_i\, (1 - x_i) \qquad (1)$$

Nilai awal *chaos*, x_0, dan konstanta r berperan sebagai parameter rahasia *Logistic Map*.

Bit-bit *MSB* yang dipilih dari setiap *pixel* di-XOR-kan dengan *keystream* yang panjangnya empat bit. Empat-bit *keystream* k_i diperoleh dengan teknik sebagai berikut: nilai *chaos* x_i diambil bagian desimalnya (setelah tanda koma) seukuran panjang angka (*size*) yang diinginkan kemduian diubah menjadi *integer*. Empat bit terakhir dari representasi biner *integer* itulah yang dijadikan sebagai k_i.

Tanpa kehilangan generalisasi, berikut ini dijelaskan langkah-langkah di dalam algoritma enkripsi untuk citra *grayscale*.

2.2 Enkripsi

Input: citra awal P (*plain-image*) berukuran $N \times N$, p, q, m (jumlah iterasi *ACM*), r, x_0
Output: citra terenkripsi C (*cipher-image*)

Langkah 1) Lakukan permutasi, yaitu mengacak *pixel-pixel* di dalam citra P dengan mengiterasikan *ACM* sejumlah m kali. Persamaan *ACM* adalah

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & p \\ q & pq+1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \bmod(N) \qquad (2)$$

Parameter *ACM*, yaitu p dan q, dan jumlah iterasi m, berperan sebagai kunci rahasia.

Langkah 2) Ekstraksi 4-bit *MSB* setiap *pixel* dari citra hasil langkah 1 di atas, nyatakan setiap 4-bit tersebut sebagai p_i (i = 1,2, .. n). Catatan: $n = N \times N$.

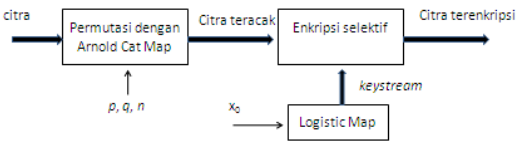
Langkah 3) Iterasikan *Logistic Map* untuk memperoleh nilai-nilai *keystream* sesuai dengan paparan di dalam 2.1.

Langkah 4) Enkripsi p_i dengan k_i menggunakan persamaan:

$$c_i = p_i \oplus k_i \qquad (3)$$

Langkah 5) c_1, c_2, …, c_n selanjutnya menggantikan 4-bit *MSB* dari setiap *pixel* yang dienkripsi. Hasil enkripsi terhadap seluruh *pixel* adalah citra terenkripsi (*cipher-image*), C.

Gambar 2 memperlihatkan diagram proses enkripsi citra digital.



Gambar 2. Diagram enkripsi

2.3 Dekripsi

Input: citra terenkripsi C (*cipher-image*), p, q, m (jumlah iterasi *ACM*), r, x_0
Output: citra semula P (*plain image*)

Langkah 1) Ekstraksi 4-bit *MSB* setiap *pixel* dari *cipher-image* *C*, nyatakan setiap 4-bit tersebut sebagai c_i (i = 1,2, .. n). Catatan: $n = N \times N$.

Langkah 2) Iterasikan *Logistic Map* untuk memperoleh nilai-nilai *keystream* sesuai dengan paparan di dalam 2.1.

Langkah 3) Dekripsi c_i dengan k_i menggunakan persamaan:

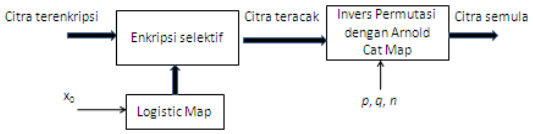
$$p_i = c_i \oplus k_i \qquad (4)$$

Langkah 4) p_1, p_2, …, p_n selanjutnya menggantikan 4-bit *MSB* dari setiap *pixel* yang didekripsi.

Langkah 5) Lakukan *inverse permutation*, yaitu menyusun kembali *pixel-pixel* citra hasil dari langkah 4 dengan persamaan *invers ACM* sebagai berikut:

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} 1 & b \\ c & bc+1 \end{bmatrix}^{-1} \begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} \bmod(N) \qquad (5)$$

Hasil *inverse permutation* ini adalah citra semula (*plain-image*), *P*. Gambar 3 memperlihatkan diagram proses dekripsi citra digital.

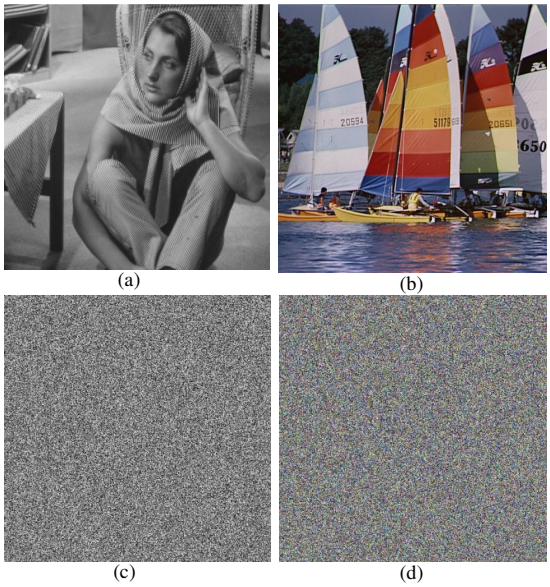


Gambar 3. Diagram dekripsi

Algoritma enkripsi/dekripsi di atas dapat dirampatkan untuk citra berwarna, yang dalam hal ini setiap *pixel* memiliki komponen *red* (*R*), *green* (*G*), dan *blue* (*B*). Prosesnya enkripsinya dilakukan tiga kali, masing-masing untuk kanal *R*, *G*, dan *B*. Jadi, dari setiap kanal warna diambil 4-bit *MSB* kemudian dioperasikan dengan algoritma di atas secara terpisah untuk masing-masing kanal. Pengacakan *pixel-pixel* dengan *ACM* juga dilakukan masing-masing untuk setiap kanal warna.

3. EKSPERIMEN

Eksperimen dilakukan dengan menggunakan kakas Matlab. Dua buah citra uji yang digunakan adalah sebuah citra *grayscale* dan sebuah citra berwarna. Kedua buah citra tersebut adalah citra 'barbara' (512 × 512) dan citra 'yacht' (512 × 512), seperti ditunjukkan pada Gambar 4(a) dan 4(b). Parameter kunci yang dipakai di dalam eksperimen adalah: p = 27, q = 89, r = 3.98, x_0 = 0.6, dan m = 5. Citra hasil enkripsi (*cipher-image*) masing-masing dapat dilihat pada Gambar 4(c) dan 4(d). Citra hasil enkripsi terlihat sudah tidak dapat dikenali lagi dan tampak seperti citra acak. Dekripsi terhadap *cipher-image* menghasilkan kembali tepat seperti citra 4(a) dan 4(b) semula.



Gambar 4. (a) dan (b) *plain-images*, (c) dan (d) *cipher-images*

4. ANALISIS KEAMANAN

Pada bagian ini didiskusikan analisis keamanan terhadap algoritma di atas. Analisis keamanan meliputi analisis ruang kunci, analisis histogram, analisis korelasi, analisis entropi, dan sensitivitas.

4.1 Analisis Ruang Kunci

Serangan *brute-force* mencoba semua kumungkinan kunci untuk melakukan dekripsi. Agar serangan brute-force tidak efektif, maka ruang kunci harus dibuat cukup besar. Ruang kunci menyatakan jumlah total kunci yang berbeda yang dapat digunakan untuk melakukan enkripsi/dekripsi [7]. Parameter kunci rahasia yang digunakan di dalam algoritma enkripsi ini lebih dari satu buah, yaitu p, q, m, x_0, dan r. Tiga parameter pertama, p, q, dan m adalah *integer* positif. Matlab mendukung maksimum *unsigned integer* hingga 32 bit, sehingga nilai pilihan nilai *integer* yang mungkin adalah sekitar $2^{32} = 4.3 \times 10^9$. Untuk nilai awal *Logistic Map* (x_0), presisi komputasi untuk *double-pecision* 64-bit menurut standard *floating-point IEEE* adalah 10^{-15} [7], sehingga jumlah kemungkinan nilai x_0 adalah 10^{15}. Dengan demikian, ruang kunci seluruhnya adalah

$$H(p, q, m, x_0, r) \approx (4.3 \times 10^9) \times (4.3 \times 10^9) \times (10^{15}) \times (10^{15}) \approx 18.49 \times 10^{48}$$

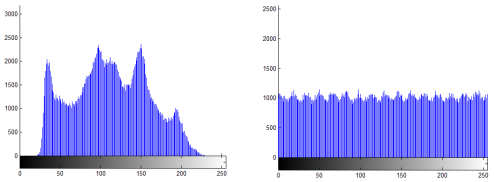
Ukuran ruang kunci ini cukup besar sehingga algoritma dapat bertahan terhadap serangan *brute-force attack.*

4.2 Analisis Histogram

Di dalam bidang pengolahan citra histogram memperlihatkan distribusi nilai *pixel* di dalam sebuah citra. Histogram digunakan penyerang (*attacker*) untuk melakukan kriptanalisis dengan memanfaatkan frekuensi kemunculan *pixel* di dalam histogram. Penyerang berharap nilai *pixel* yang sering muncul di dalam *plain-image* berkorelasi dengan nilai *pixel* yang sering muncul di dalam *cipher-image*. Dengan menganalisis frekuensi kemunculan nilai *pixel*, penyerang mendeduksi kunci atau *pixel-pixel* di dalam *plain-image*.

Agar penyerang tidak dapat menggunakan histogram untuk melakukan analisis frekuensi, maka histogram *plain-image* dan histogram *cipher-image* seharusnya berbeda secara signifikan atau secara statistik tidak memiliki kemiripan. Oleh karena itu, histogram *cipher-image* seharusnya datar (*flat*) atau secara statistik memiliki distribusi (relatif) *uniform*. Distribusi yang (relatif) *uniform* pada *cipher-image* adalah sebuah indikasi bahwa algoritma enkripsi citra memiliki tingkat keamanan yang bagus [6].

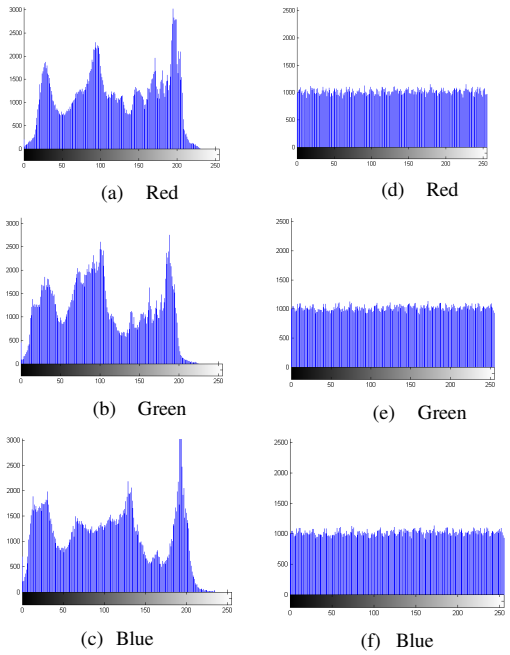
Gambar 5(a) memperlihatkan histogram citra 'kapal' sebelum dienkripsi, dan Gambar 5(b) adalah histogram *cipher-image*-nya. Histogram *cipher-image* terlihat datar dan berbeda secara signifikan dengan histogram *plain-image*.



(a) (b)

Gambar 5. (a) Histogram citra 'barbara' (*plain-image*) dan (b) histogram *cipher-image*.

Gambar 6(a) sampai 6(c) memperlihatkan histogram citra 'yacht' (*plain-image*) untuk setiap kanal warna *RGB* dan Gambar 6(d) sampai 6(f) adalah histogram masing-masang kanal warna pada *cipher-image*. Sama seperti citra 'barbara', histogram *cipher-image* pada setiap kanal *RGB* juga terlihat *flat* atau terdistribusi *uniform*.



(a) Red (d) Red

(b) Green (e) Green

(c) Blue (f) Blue

Gambar 6. (a)-(c) Histogram citra 'yacht' (*plain-image*) untuk masing-masing kanal *RGB*; dan (d)-(f) histogram *cipher-image* untuk setiap kanal.

4.3 Analisis Korelasi

Korelasi adalah ukuran yang menyatakan kekuatan hubungan linier antara dua peubah acak. Korelasi dari dua buah peubah acak diskrit yang masing-masing beranggotakan *n* elemen dinyatakan dengan koefisien korelasi yang dihitung dengan rumus sebagai berikut [8]:

$$r_{xy} = \frac{\text{cov}(x, y)}{\sqrt{D(x)D(y)}} \qquad (6)$$

yang dalam hal ini "cov" adalah kovariansi dan "D" adalah standard deviasi:

$$\text{cov}(x, y) = \frac{1}{n}\sum_{i=1}^{n}[x_i - E(x)][y_i - E(y)] \qquad (7)$$

$$D(x) = \frac{1}{n}\sum_{i=1}^{n}[x_i - E(x)]^2 \qquad (8)$$

$$E(x) = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \text{(rata-rata)} \qquad (9)$$

Di dalam *natural-image*, *pixel-pixel* yang bertetangga memiliki hubungan linier yang kuat. Ini ditandai oleh koefisien korelasinya yang tinggi

(mendekati +1 atau -1). Di dalam citra acak, korelasi antar *pixel* bertetangga tidak ada atau koefisien korelasinya nol. Enkripsi citra bertujuan membuat korelasi *pixel-pixel* yang bertetangga di dalam *cipher-image* menjadi lemah atau dengan kata lain membuat koefisien korelasinya mendekati nol.

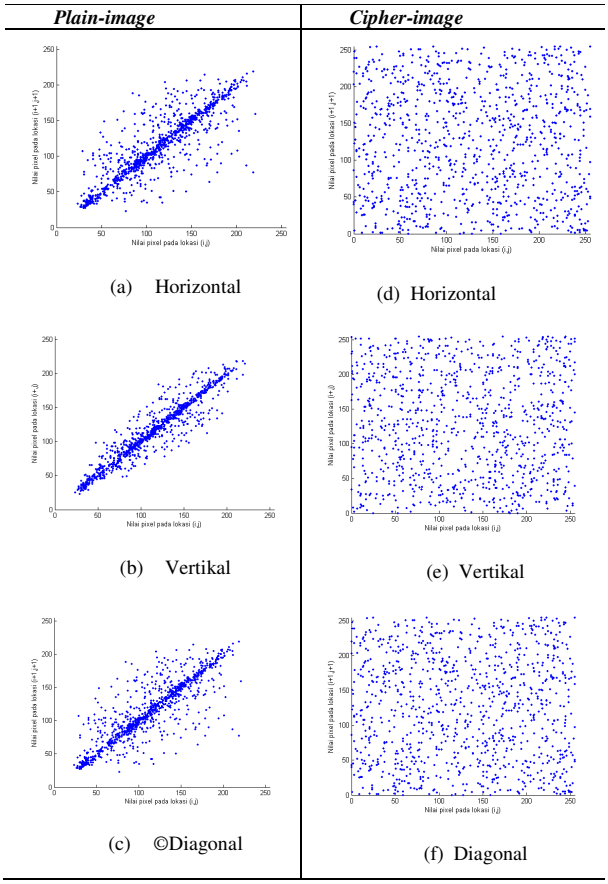
Untuk mengetahui korelasi *pixel-pixel* di dalam *plain-image* maupun *cipher-image*, maka dihitung koefisien korelasi antara dua *pixel* bertetangga secara horizontal [$f(i,j)$ dan $f(i, j+1)$], dua *pixel* bertetangga secara vertkal [$f(i,j)$ dan $f(i+1, j)$], dan dua *pixel* bertetangga secara diagonal $f(i,j)$ dan $f(i+1, j+1)$]. Secara acak dipilih 1000 pasang *pixel* bertetangga pada setiap arah (vertikal, horizontal, dan diagonal), masing-masing pada citra *plain-image* dan *cipher-image*. Tanpa kehilangan generalisasi, analisis korelasi dilakukan pada citra *grayscale* saja. Koefisien korelasi untuk citra 'barbara' dihitung dengan persamaan (6), yang dalam hal ini x dan y adalah nilai keabuan dari dua *pixel* bertetangga. Hasil perhitungan korelasi diperlihatkan pada Tabel 1.

Tabel 1. Perbandingan koefisien korelasi antara dua *pixel* bertetangga

Koefisien korelasi	Horizontal	Vertikal	Diagonal
Plain-image	0.8834	0.9487	0.8620
Cipher-image	0.0379	-0.0137	-0.0020

Dari Tabel 1 dapat dilihat bahwa koefisien korelasi pada *pixel-pixel* bertetangga pada setiap arah di dalam *plain-image* nilainya mendekati 1, yang mengindikasikan korelasi yang kuat diantara *pixel-pixel* tersebut. Sebaliknya pada *cipher-image* koefisien korelasinya mendekati nol, yang mengindikasikan *pixel-pixel* yang bertetangga tidak lagi berkorelasi.

Untuk melihat lebih jelas korelasi antara *pixel-pixel* bertetangga, maka Gambar 7 memperlihatkan distribusi korelasi *pixel-pixel* yang bertetangga. Kolom sebelah kiri adalah distribusi korelasi pada *plain-image* dan kolom kanan adalah distribusi korelasi pada *cipher-image*. Pada *plain-image* dapat dilihat bahwa *pixel-pixel* yang bertetangga nilai-nilainya berada di sekitar garis diagonal 45°, yang mengindikasikan korelasi yang kuat antara *pixel-pixel* tersebut. Sebaliknya, pada *cipher-image* nilai-nilai *pixel* tersebar merata di seluruh area bidang datar, yang mengindikasikan *pixel-pixe*l di dalamnya tidak lagi berkorelasi.



Gambar 7. Distribusi korelasi *pixel-pixel* bertetangga pada *plain-image* dan *cipher-image* dari citra 'barbara'

4.4 Analisis Entropi

Di dalam teori informasi, entropi menyatakan derajat ketidakpastian di dalam sistem. Entropi pesan m dihitung dengan persamaan [6]:

$$H(m) = \sum_{i=0}^{2M-1} P(m_i) \log_2 \frac{1}{P(m_i)} \qquad (10)$$

yang dalam hal ini $P(m_i)$ menyatakan peluang simbol m_i di di dalam pesan dan entropi dinyatakan dalam satuan bit. Pesan acak seharusnya memiliki entropi yang ideal sama dengan 8, sedangkan pada pesan yang kurang acak nilai entropinya kurang dari delapan. Jika entropi kurang dari delapan, maka terdapat derajat mampu-prediksi (*predictability*) yang merupakan ancaman bagi keamanan [6].

Pada kasus enkripsi citra, *cipher-image* yang dihasilkan adalah citra acak, maka entropinya seharusnya ideal 8. Karena ada 256 derajat keabuan di dalam citra ($m_0 = 0$, $m_1 = 1$, …, $m_{255} = 255$) dan setiap derajat keabuan dicatat peluangnya (dihitung

dari histogramnya), maka untuk *cipher-image* pada Gambar 4(c) nilai entropinya adalah

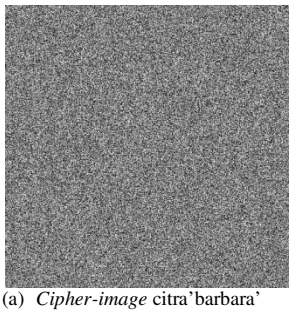
$$H(m) = \sum_{i=0}^{255} P(m_i) \log_2 \frac{1}{P(m_i)} = 7.9983$$

Nlai entropi ini sangat dekat dengan 8 yang berarti algoritma enkripsi aman dari serangan entropi (*entropy attack*) untuk memprediksi informasi di dalam citra.

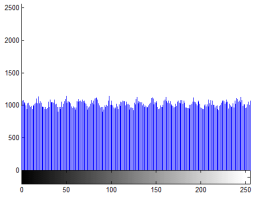
4.5 Analisis Sensitivitas

Parameter nilai awal fungsi *chaos* berperan sebagai (salah satu) kunci rahasia. Sifat *chaos* adalah sensitif terhadap perubahan kecil nilai awal. Sensitif berarti jika nilai kunci diubah sedikit saja maka hasil dekripsi terhadap *cipher-image* menghasilkan *cipher-image* lain yang berbeda (gagal mengembalikan *cipher-image* menjadi *plain-image* semula).

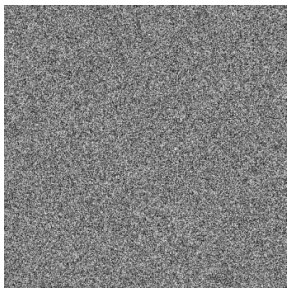
Di dalam algoritma enkripsi citra yang diusulkan, *Logistic Map* digunakan untuk membangkitkan 4-bit *keystream* dari nilai-nilai *chaos* yang kemudian di-XOR-kan dengan 4-bit dari *pixel*. Perubahan kecil nilai awal *chaos* membuat nilai-nilai acak yang dihasilkan dari *Logistic Map* berbeda signifikan setelah fungsi *chaos* diiterasi sejumlah kali. Akibatnya, *keystream* yang dibangkitkan juga berbeda signifikan, dan sebagai hasilnya operasi XOR memberikan citra yang berbeda signifikan pula.



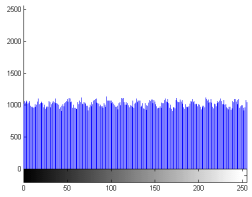
(a) *Cipher-image* citra'barbara'



(b) Histogram dari citra (a)



(c) Citra 'barbara' hasil dekripsi (x_0 ditambah sebesar $\Delta = 10^{-10}$)



(d) Histogram dari citra (c)

Gambar 8. Hasil eksperimen dekripsi dengan pengubahan x_0 sebesar $\Delta = 10^{-10}$.

Pada eksperimen ini nilai awal *logistic map* diubah sebesar Δ sehingga menjadi $x_0 + \Delta$, kemudian citra didekripsi dengan kunci $x_0 + \Delta$ tersebut. Misalkan $\Delta = 10^{-10}$ sehingga nilai awal *logistic map* menjadi 0.60000000001. Gambar 9 memperlihatkan hasil dekripsi terhadap *cipher-image* dari citra 'barbara'. Hasilnya adalah *cipher–image* lain yang ternyata tetap teracak (tidak kembali menjadi citra semula). Penyerang yang melakukan *exhaustive key search attack* untuk menemukan kunci akan frustasi karena perubahan sangat kecil pada kunci menyebabkan hasil dekripsi tetap salah.

5. KESIMPULAN

Di dalam makalah telah disajikan analisis keamanan algoritma enkripsi citra digital yang menggabungkan pengunaan dua buah chaos map (*Arnold Cat Map* dan *Logistic Map*) dan teknik enkripsi selektif. Analisis keamanan meliputi analisis ruang kunci, analisis histogram, analisis korelasi, analisis entropi, dan analisis sensitivitas

Analisis ruang kunci menunjukkan bahwa jumlah kemugkinan kunci sangat besar sehingga algoritma aman dari serangan *brute-force attack*. Analisis histogram memperlihatkan bahwa histogram *cipher-image* berbentuk datar atau terdistribusi *uniform*, sehingga akgoritma aman dari serangan analisis frekuensi. Analisis korelasi memperlihatkan *pixel-pixel* di dalam *cipher-image* tidak berkorelasi satu lain (memiliki koefisien korelasi yang mendekati nol), sehingga algoritma aman dari serangan analisis statistik untuk menemukan kunci atau plain-image. Analisis entropi memperlihatkan algoritma memiliki entripi yang mendekati nilai entropi ideal (8), sehingga algoritma aman dari kebocoran informasi, Analisis sensitivitas menunjukkan bahwa perubahan nilai awal *chaos* memperlihatkan bahwa algoritma ini aman dari *exhaustive-key search attack*.

Secara keseluruhan algoritma enkripsi citra yang dibahas aman dari serangan untuk menemukan kunci atau *pixel-pixel* di dalam *plain-image*.

6. ACKNOWLEDGMENT

Penelitian yang dipublikasikan di dalam makalah ini sepenuhnya didukung oleh dana **Riset dan Inovasi KK 2012** (Program Riset ITB 2012).

7. DAFTAR PUSTAKA

[1] Rinaldi Munir. 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 Yogyakarta, 2012.*

[2] B. Schneier. 1996. *Applied Cryptography 2nd Edition*. Wiley & Sons.

[3] T. Xiang, K. Wong, and X. Liao. 2007. Selective Image Encryption Using a Spatiotemporal Chaotic System. *Chaos Volume 17*.

[4] K. Struss. 2009. A Chaotic Image Encryption, *Mathematics Senior Seminar*, 4901, University of Minnesota, Morris.

[5] X. Yu, J. Zhang, H. Ren, G. Xu, and X. Luo. 2006. Chaotic Scrambling Algorithm Based on S-DES. *Journal of Physics: Conference Series 48*, 349-353

[6] A. Jolfaei, A. Mirghadri. 2010. An Image Encryption Approach Using Chaos and Stream Cipher. *Journal of Theoretical and Applied Information Technology*.

[7] 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*.

[8] T. Hongmei, H. Liying, and W. Xi. 2010. "An Improved Compound Image Encryption Scheme". *Proceeding of 2010 International Conference on Computer and Communication Technologies in Agriculture Engineering*, 2010

.

.