ISSN 2087-2658
Prosiding Seminar Nasional Pendidikan Teknik Informatika
(SENAPATI 2012)
Singaraja – Bali, 22 September 2012

ALGORITMA ENKRIPSI CITRA DIGITAL BERBASIS CHAOS DENGAN PENGGABUNGAN TEKNIK PERMUTASI DAN TEKNIK SUBSTITUSI MENGGUNAKAN *ARNOLD CAT MAP* DAN *LOGISTIC MAP*

Oleh

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

ABSTRAK

Di dalam makalah ini dipresentasikan sebuah usulan algoritma enkripsi citra digital yang menggabungkan teknik permutasi dan substitusi. Dua buah *chaotic map* digunakan untuk masing-masing teknik yaitu *Arnold Cat Map* dan *Logistic Map*. Sebelum dienkripsi, *pixel-pixel* di dalam citra diacak dengan *Arnold Cat Map*. Selanjutnya, *pixel-pixel* tersebut diubah nilainya melalui operasi *XOR* dengan *keystream* yang dibangkitkan dari *Logistic Map*. *Pixel-pixel* dioperasikan seperti mode *cipher block chaining*. Hasil eksperimen pada citra *grayscale* dan citra berwarna menunjukkan *cipher-image* memiliki *pixel-pixel* yang terdistribusi *uniform* sehingga menyulitkan serangan dengan analisis statistik. Selain itu *pixel-pixel* yang berteatngga di dalam *cipher-image* memiliki koefisien korelasi yang rendah, yang mengindikasikan bahwa *pixel-pixel* tersebut sudah tidak memiliki hubungan linier (korelasi). Sifat sensitivitas pada *chaos* telah diperlihatkan yang mengindikasikan algoritma ini aman dari *exhaustive-key search attack*.

Kata kunci: enkripsi, citra, *chaos*, *Arnold Cat Map*, *Logistic Map*, permutasi, substitusi

1. Pendahuluan

Citra (*image*) atau gambar merupakan salah satu bentuk multimedia yang penting. Citra menyajikan informasi secara visual dan informasi yang disajikan oleh sebuah citra lebih kaya daripada yang disajikan secara tekstual. Citra digital tidak hanya disimpan di dalam storage seperti *hard disk*, *flash disk*, *CD*, *DVD*, dan perangkat memori lainnya, tetapi juga ditransmisikan melalui saluran publik seperti internet. Untuk citra yang bersifat privat atau yang bersifat rahasia, penyimpanan dan pengiriman citra perlu memperhatikan aspek

keamanan. Citra yang bersifat privat misalnya foto dokumen pribadi yang hanya boleh dilihat oleh pemilik atau orang-orang yang diberi otoritas saja. Citra yang bersifat rahasia contohnya adalah citra hasil penginderaan jarak jauh (foto satelit) yang merekam potensi kekayaan alam sebuah negara.

Selain untuk citra privat dan rahasia, aspek keamanan merupakan fitur yang penting pada citra berbayar. Hanya pelanggan yang telah membayar saja yang dapat mengakses informasi di dalam citra. Sebuah video digital pada hakekatnya disusun oleh rangkaian *frame* citra diam yang ditampilkan dalam tempo yang sangat singkat. Untuk industri multimedia seperti *Pay TV* atau *video on demand*, perlindungan terhadap siaran video memainkan peranan yang penting, sebab siaran video dipancarkan secara *broadcast* melalui saluran transmisi (yang dapat disadap) tetapi hanya pelanggan yang membayar saja yang dapat menikmati siaran TV, sedangkan pelanggan ilegal tidak dapat mengakses siaran video tersebut.

Solusi terhadap keamanan citra digital dari pengakasesan yang ilegal adalah dengan mengenkripsinya. Tujuan enkripsi citra adalah menyandikan citra (*plain-image*) sehingga tidak dapat dikenali lagi (*cipher-image*). Saat ini enkripsi citra sudah telah digunakan secara luas sebagai salah satu teknik menjaga keamanan informasi. Enkripsi merupakan salah satu teknik keamanan pesan di dalam kriptografi, termasuk pesan dalam bentuk citra. Salah satu layanan yang diberikan oleh kriptografi adalah kerahasiaan pesan (*confidentiality*), dan *confidentiality* diimplementasikan dengan enkripsi dan dekripsi pesan (Schneier, 1996).

Para peneliti sudah banyak mengembangkan algoritma kriptografi untuk enkripsi, namun sebagian besar algoritma tersebut ditujukan untuk mengenkripsi pesan dalam bentuk teks. Meskipun algoritma enkripsi konvensional seperti *DES*, *AES*, *Blowfish*, *Serpent*, *RC4*, *RSA*, *ElGamal*, *Rabin*, dapat juga mengenkripsi citra, namun mereka tidak mangkus untuk diterapkan. Hal ini disebabkan citra mempunyai karakteristik yang berbeda dengan data tekstual. Sebuah citra umumnya memiliki kapasitas data yang sangat besar, sehingga enkripsi citra memerlukan volume komputasi yang besar. Beberapa aplikasi yang mempunyai kebutuhan *real-time* seperti *teleconference*, *live video streaming*, dan lain-lain, jelas memerlukan kecepatan komputasi yang sangat tinggi sehingga algoritma konvensional jelas tidak cocok untuk mengenkripsi citra.

Selain karena alasan volume, karakteristik citra yang membedakan dengan teks adalah korelasi data antar tetangga. Data di dalam teks hanya bertetangga dengan data sebelum (*predecessor*) dan sesudahnya (*successor*), sedangkan di dalam citra *pixel-pixel*-nya bertetangga dengan *pixel-pixel* lain dalam delapan penjuru mata angin sehingga korelasinya dengan delapan *pixel* tetangganya tinggi. Oleh karena itu, setelah sebuah citra dienkripsi maka yang harus diperhatikan adalah *pixel-pixel* di dalam *cipher-image* seharusnya tidak memiliki korelasi dengan *pixel-pixel* tetangganya.

Karena alasan-alasan spesifik di atas, maka perlu dikembangkan algoritma yang khusus untuk citra digital. Penelitian tentang enkripsi citra dilakukan dengan intensif. Para peneliti telah banyak mengembangkan algoritma enkripsi citra digital. Menurut Younes (2008), kebanyakan algoritma-algoritma enkripsi citra dapat dikelompokkan menjadi dua

kelompok. Kelompok pertama adalah algoritma enkripsi selektif non-*chaos*, sedangkan kelompok kedua adalah algoritma enkripsi selektif atau non-selektif yang berbasis *chaos*. Yang dimaksud dengan algoritma selektif adalah algoritma yang mengenkripsi hanya sebagian elemen di dalam citra namun efeknya citra terenkripsi secara keseluruhan. Tujuan algoritma enkripsi selektif adalah mengurangi volume komputasi, yang konsekuensinya adalah menghemat waktu proses enkripsi. Enkripsi selektif cocok untuk aplikasi yang membutuhkan persyaratan *real-time*.

Kriptografi berbasis *chaos* menjadi topik penelitian yang atraktif saat ini. *Chaos* digunakan di dalam kriptografi karena tiga alasan: (1) sifat *chaos* yang sensitif terhadap kondisi awal sistem, (2) *chaos* berkelakuan acak, dan (3) nilai-nilai *chaos* tidak memiliki periode. *Review* beberapa algoritma enkripsi citra dengan menggunakan skema *chaos* dapat dibaca di dalam Sharma (2010).

Dua operasi dasar di dalam algoritma enkripsi citra adalah permutasi (atau transposisi) dan substitusi. Permutasi mengubah posisi *pixel-pixel* di dalam citra, sedangkan subsitutsi mengubah nilai *pixel*. Makalah ini menyajikan sebuah algoritma enkripsi citra berbasis *chaos* yang mengkombinasikan teknik permutasi dan substitusi. Dua buah fungsi *chaos* yang digunakan adalah *Arnold Cat Map* dan *Logistic Map*. *Arnold Cat Map* digunakan untuk mengacak susunan *pixel-pixel*, sedangkan *Logistic Map* digunakan sebagai pembangkit *keystream* yang kemudian dienkripsikan dengan *pixel-pixel* hasil permutasi. *Pixel-pixel* dienkripsikan dengan mode seperti mode *CBC* pada *block-cipher*, meskipun yang dioperasikan tidak dalam bentuk blok-blok data melainkan *pixel* per *pixel*.

2. Chaos

Teori *chaos* sudah menajdi topik penelitian yang atraktif di dalam bdiang keamanan informasi. Karakteristik yang menarik dari chaos adalah sensitivitasnya terhadap nilai awal (*initial value*). Jika nilai awal sistem *chaos* diubah sedikit saja, misalnya sebesar 10^{-10}, maka bila sistem *chaos* tersebut diiterasikan sejumlah kali, hasil iterasinya berbeda signifikan dengan sebelumnya. Sensisitivitas ini sangat berguna di dalam kriptografi karena bersesuaian dengan prinsip *diffusion* dari Shannon dalam merancang sebuah algoritma kriptografi (Schneier, 1996). Dengan prinsip *diffusion* ini maka pengubahan satu bit nilai awal *chaos* dapat menyebabkan cipherteks tetap tidak berhasil didekripsi. Dua buah fungsi *chaos* yang digunakan di dalam algoritma ini adalah *Logistic Map* dan *Arnold Cat Map*.

2.1 Logistic Map

Logistic Map adalah fungsi *chaos* satu dimensi yang telah digunakan secara luas, yang didefinsikan sebagai

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

Nilai-nlai x_i adalah bilangan riil di dalam selang (0, 1), sedangkan μ adalah parameter fungsi yang menyatakan laju pertumbuhan yang nilainya di dalam selang (0, 4]. *Logistic Map* akan bersifat *chaos* bilamana $3.5699456 \leq \mu \leq 4$ (Hongmei, 2010). Untuk memulai iterasi *Logistic Map* diperlukan nilai awal x_0. Perubahan sedikit saja pada nilai awal ini (misalnya sebesar 10^{-10}) akan menghasilkan nilai-nilai *chaos* yang berbeda secara signifikan setelah *Logistc Map* diiterasi sejumlah kali. Di dalam sistem kriptografi simetri, nilai awal *chaos*, x_0, dan parameter μ berperan sebagai kunci rahasia. Nilai-nilai acak yang dihasilkan dari persamaan (1) tidak pernah berulang kembali sehingga *Logistic Map* dikatakan tidak mempunyai periode.

2.2 Arnold Cat Map

Arnold Cat Map (*ACM*) merupakan fungsi *chaos* dwimatra dan bersifat *reversible*. Fungsi *chaos* ini ditemukan oleh Vladimir Arnold pada tahun 1960, dan kata "*cat*" muncul karena dia menggunakan citra seekor kucing dalam eksperimennya.

ACM mentransformasikan koordinat (x, y) di dalam citra yang berukuran $N \times N$ ke koordinat baru (x', y'). Persamaan iterasinya adalah

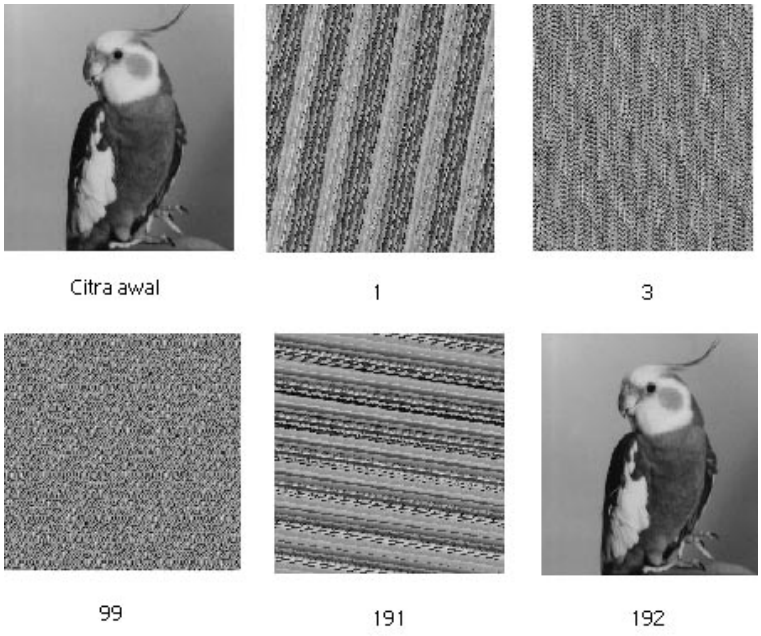
$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & b \\ c & bc+1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \bmod(N) \qquad (2)$$

yang dalam hal ini (x_i, y_i) adalah posisi *pixel* di dalam citra, (x_{i+1}, y_{i+1}) posisi *pixel* yang baru setelah iterasi ke-*i; b* dan *c* adalah *integer* positif sembarang. Determinan matriks $\begin{bmatrix} 1 & b \\ c & bc+1 \end{bmatrix}$ harus sama dengan 1 agar hasil transformasinya bersifat *area-preserving*, yaitu tetap berada di dalam area citra yang sama. *ACM* termasuk pemetaan yang bersifat satu-ke-satu karena setiap posisi *pixel* selalu ditransformasikan ke posisi lain secara unik. *ACM* diiterasikan sebanyak *m* kali dan setiap iterasi menghasilkan citra yang acak. Nilai *b*, *c*, dan jumlah iterasi *m* dapat dianggap sebagai kunci rahasia.

Proses yang terjadi di dalam setiap *iterasi ACM* adalah pergeseran (*shear*) dalam arah *y*, kemudian dalam arah *x*, dan semua hasilnya (yang mungkin berada di luar area gambar) dimodulokan dengan *N* agar tetap berada di dalam area gambar (*area preserving*)

Setelah *ACM* diiterasi sebanyak *m* kali, maka terdapat *T* sedemikian sehingga (x_T, y_T) = (x, y), yang dalam hal ini nilai *T* bergantung pada *b*, *c*, dan ukuran *N* (We-bin, 2009). Ini berarti sesudah *ACM* diiterasi sebanyak *T* kali, maka hasil iterasinya kembali ke citra semula, sehingga dikatakan *ACM* besifat *reversible* dan periodenya adalah *T*. Menurut Struss (2009), penelitian Freeman J. Dyson dan Harold Falk menemukan bahwa $T < 3N$. Gambar 1 memperlihatkan iterasi *ACM* terhadap citra 'burung'. Pada iterasi ketiga hasilnya sudah terlihat seperti citra acak, semakin banyak iterasinya citra hasil semakin acak (dalam hal ini *ACM* telah berada dalam fase *chaos*). Jika proses iterasi diteruskan maka hasilnya

kembali menuju citra semula. Pada contoh ini citra 'burung' kembali ke bentuk semula pada iterasi ke-192 sehingga dikatakan periodenya adalah $T = 192$.



Gambar 1. Iterasi ACM pada citra 'burung' dengan periode $T = 192$

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

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} 1 & b \\ c & bc+1 \end{bmatrix}^{-1} \begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} \bmod(N) \qquad (3)$$

Setelah iterasi terakhir citra hasil sama sepert citra semula. Proses dekripsipun selesai.

3. Algoritma Enkripsi/Dekripsi yang Diusulkan

Meskipun citra hasil transformasi *Arnold Cat Map* sudah teracak dan tidak bisa dikenali lagi, namun enkripsi dengan *Arnold Cat Map* saja tidak cukup aman, karena sifat periodik *ACM* dapat menghasilkan kembali citra semula. Dengan proses *hack* sederhana nilai b dan c dapat ditemukan melalui operasi *brute force* (Yu, 2006). Selain itu *ACM* hanya mengubah posisi *pixel* di dalam citra tetapi tidak mengubah nilai *pixel*. Oleh karena itu, nilai-nilai *pixel* hasil permutasi dengan *ACM* perlu diubah nilainya melalui operasi substitusi dengan *Logistic Map*.

Tanpa kehilangan generalisasi, di bawah ini diuraikan rincian algoritma enkripsi yang diusulkan untuk citra *grayscale* yang berukuran $N \times N$. Pada dasarnya algoritma enkripsi dibagi menjadi dua tahapan: tahap pengacakan dan tahap *encoding*.

3.1 Enkripsi: Tahap Pengacakan

Pada tahap ini dilakukan operasi permutasi dengan *ACM* yang bertujuan mengacak susunan *pixel-pixel* di dalam citra.

Masukan: citra *I*, *b*, *c*, dan *m*
Keluaran: citra teracak *I*'
Proses: Iterasikan *ACM* pada persamaan (1) dengan parameter *b* dan *c* terhadap citra *I* sebanyak *m* kali. Hasil iterasi terakhir adalah citra *I*'

Pseudo-code algoritmanya adalah sebagai berikut:

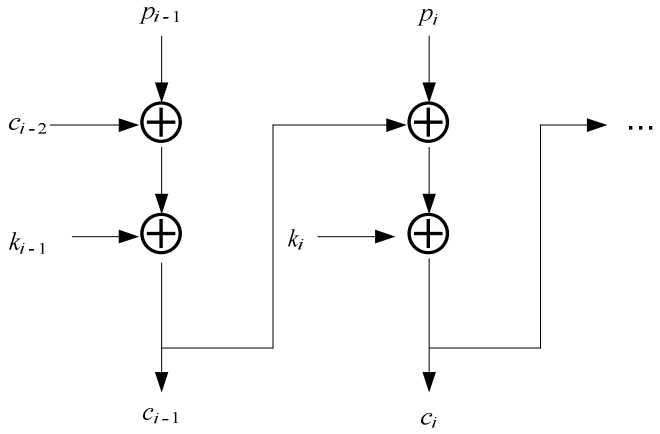
```
procedure Pengacakan(input I: image; b, c, m : integer; output I' : image)
Deklarasi
   k : integer
Algoritma
   for k ← 1 to m do
     I ← ACM1(I,b,c)  {transformasi setiap pixel citra I dengan ACM yang
                       memiliki parameter b, dan c}
   end
   I' ← I
```

3.2 Enkripsi: Tahap *Encoding*

Pada tahap ini dilakukan pengubahan nilai-nilai *pixel* dari citra hasil pengacakan dengan operasi substitusi.

Masukan: citra *I'*, x_0, dan μ
Keluaran: citra terenkripsi (*cipher-image*) *C*
Proses: *Pixel-pixel* citra *I'* = (p_1, p_2, …, $p_{N\times N}$) di-XOR-kan dengan *keystream* (*integer*) (yang dibangkitkan dari *Logistic Map* dengan nilai awal x_0 dan parameter μ). Skema enkripsinya adalah dengan mode yang diadopsi dari *cipher block chaining* (Schneier, 1996) seperti yang ditunjukkan pada Gambar 2. *Pixel-pixel* hasil enkripsi adalah $C = (c_1, c_2, \ldots, c_{N\times N})$. Proses enkripsi ini dapat dirumuskan sebagai

$$c_i = (p_i \oplus c_{i-1}) \oplus k_i \qquad (3)$$



Gambar 2. Skema enkripsi

Untuk enkripsi *pixel* pertama diperlukan c_0 yang dalam hal ini c_0 adalah *initialization vector* atau *IV* (pada algoritma ini *IV* = 0). *IV* tidak perlu rahasia tetapi harus sama nilainya pada proses dekripsi.

Karena nilai-nilai acak yang dibangkitkan dari *Logistic Map* berupa bilangan riil, nilai tersebut harus ditransformasikan menjadi *integer*. Transformasi yang sederhana adalah dengan mengambil bagian desimal dari bilangan riil, membuang angka nol yang tidak signifikan, lalu mengekstrak *t* digit *integer*. Sebagai contoh, misalkan x_i = 0.003176501 dan t = 4, maka diambil bagian desimalnya yaitu 003176501, buang dua buah nol yang tidak signifikan di depannya, lalu ekstrak sebanyak 4 digit yaitu 3176. Inilah *keytream* yang akan di-XOR-kan dengan *pixel* ke-*i*. Karena nilai-nilai *pixel* berada di dalam rentang *integer* [0, 255], maka sebelum di-XOR-kan *keystream* di-modulus-kan dengan 256. Jadi, pada contoh ini k_i = 3176 mod 256 = 104.

Pseudo-code algoritmanya adalah sebagai berikut:

```
procedure Encoding(input I': image; x0, μ : integer; output C : image)
Deklarasi
   p, c, k, i, j : integer
   x : real
Algoritma
   x ← x0
   c ← 0   { IV }
   for i ← 1 to N do
     for j ← 1 to N do
       p ← I'(i,j)          { ambil pixel p di dalam citra I'}
       x ← μ * x *  (1 – x)   { Logistic Map }
```

```
    k ← transformasikan x menjadi integer
    c ← (p ⊕ c) ⊕ k
    C(i,j) ← c          { Simpan c ke dalam C(i,j) }
  end
end
```

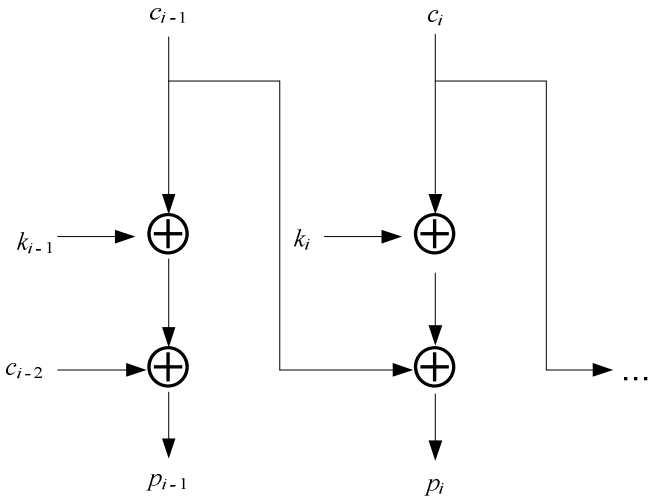
Algoritma dekripsi berkebalikan dengan algoritam enkripsi. Urutannya adalah tahap *decoding* terlebih dahulu, kemudian tahap balik-pengacakan. Penjelasannya adalah seperti di dalam upa-bab di bawah ini.

3.3 Dekripsi: Tahap *Decoding*

Pada tahap ini dilakukan pengubahan nilai-nilai *pixel* dari *cipher-image* dengan operasi substitusi. Algoritmanya adalah sebagai berikut:

Masukan: citra C, x_0, dan μ
Keluaran: citra hasil *decoding*, P
Proses: *Pixel-pixel* citra $C = (c_1, c_2, \ldots, c_{N\times N})$ di-XOR-kan dengan *keystream* (*integer*) (yang dibangkitkan dari *Logistic Map* dengan nilai awal x_0 dan parameter μ). Skema dekripsinya adalah dengan mode yang ditunjukkan pada Gambar 3. *Pixel-pixel* hasil dekripsi adalah $P = (p_1, p_2, \ldots, p_{N\times N})$. Proses dekripsi ini dapat dirumuskan sebagai

$$p_i = (c_i \oplus k_1) \oplus c_{i-1} \qquad (4)$$



Gambar 3. Skema dekripsi

Pseudo-code algoritmanya adalah sebagai berikut:

```
procedure Decoding(input C : image; x0, μ : integer; output P : image)
Deklarasi
   p, c, k, i, j : integer
   x : real
Algoritma
   x ← x0
   cPrev ← 0    { IV }
   for i ← 1 to N do
     for j ← 1 to N do
       c ← C(i,j)          { ambil pixel c di dalam citra C}
       x ← μ * x *  (1 – x)   { Logistic Map }
       k ← transformasikan x menjadi integer
       p ← (c ⊕ k) ⊕ cPrev
       P(i,j) ← p          { Simpan p ke dalam P(i,j)}
       cPrev ← c
     end
   end
```

3.4 Dekripsi: Tahap Balik-Acak

Pada tahap ini dilakukan operasi permutasi dengan *invers ACM* yang bertujuan mengembalikan susunan *pixel-pixel* menjadi susunan semula

Masukan: citra *P*, *b*, *c*, dan *m*
Keluaran: citra semula *I*
Proses: Iterasikan *ACM* pada persamaan (2) dengan parameter *b* dan *c* terhadap citra *P* sebanyak *m* kali. Hasil iterasi terakhir adalah citra semula, *I*.

Pseudo-code algoritmanya adalah sebagai berikut:

```
procedure Balik-Acak(input P: image; b, c, m : integer; output I : image)
Deklarasi
   k : integer
Algoritma
   for k ← 1 to m do
     P ← ACM2(P,b,c)  {transformasi setiap pixel citra I dengan invers ACM yang
                       memiliki parameter b, dan c}
   end
   I ← P
```

3.5 Enkripsi dan Dekripsi pada Citra Berwarna

Pixel-pixel pada citra berwarna disusun oleh tiga buah kanal warna, yaitu red (*R*), green (*G*), dan blue (*B*). Oleh karena itu, tahap pengacakan dan tahap encoding dilakukan secara terpisah untuk masing-masing kanal. Jadi proses enkripsi/dekripsinya tiga kali lebih lama daripada citra *grayscale*.

4. Eksperimen dan Pembahasan Hasil

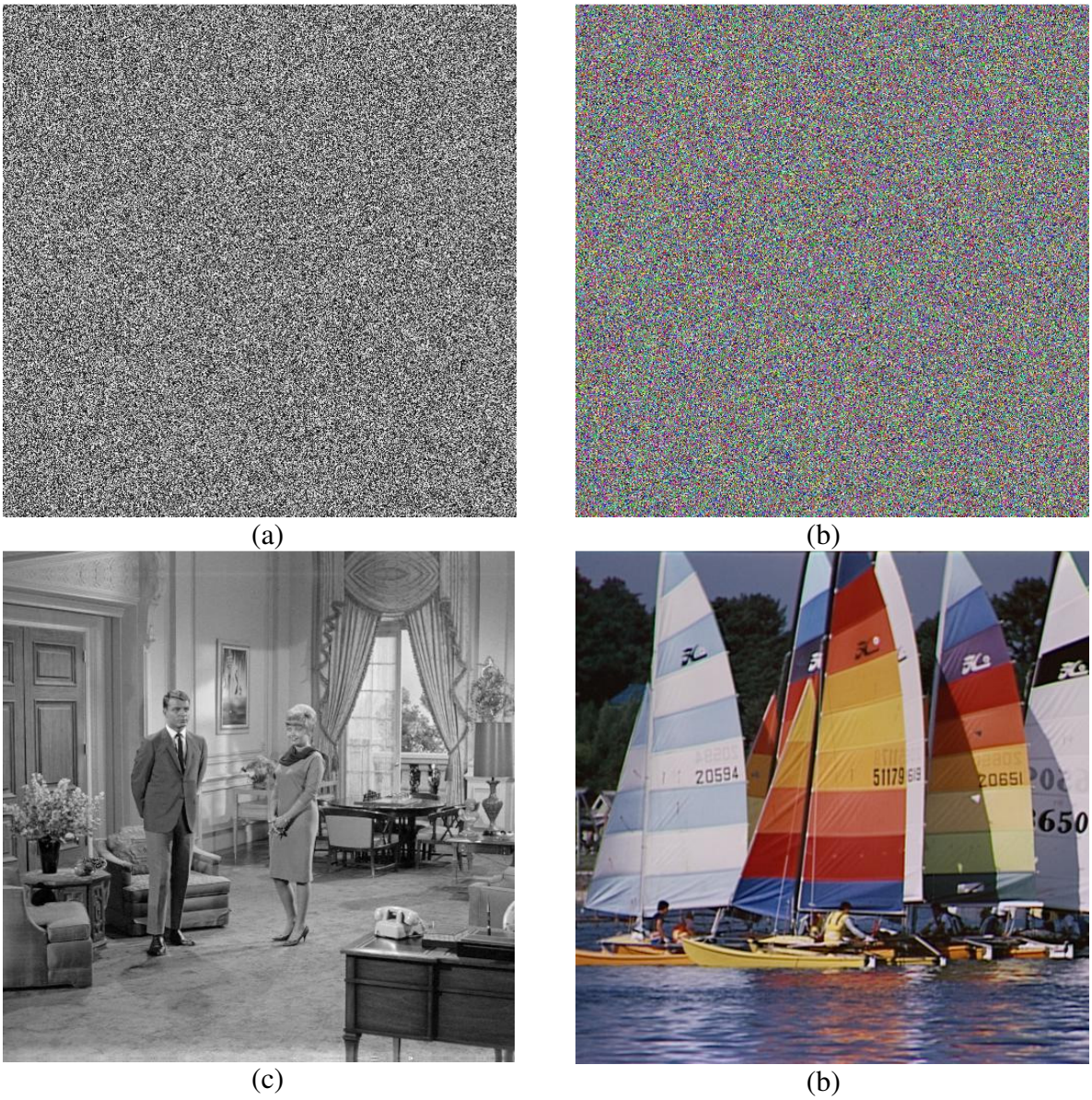
Algoritma enkripsi/dekripsi yang telah didekripsikan di atas disimulasikan dengan dengan menggunakan kakas *Matlab*. Eksperimen dilakukan pada dua buah citra uji, masing-masing citra *grayscale* dan citra berwarna. Kedua buah citra tersebut merupakan citra uji standard di dalam bidang pengolahan citra, yaitu citra 'couple' (512 × 512) dan citra 'yacht' (512 × 512), seperti ditunjukkan pada Gambar 4(a) dan 4(b). Parameter kunci yang dipakai di dalam eksperimen adalah: $b = 32$, $c = 41$, $r = 3.9728$, $x_0 = 0.3$, dan $m = 5$.

(a) (b)

Gambar 4. Dua buah citra uji: (a) 'couple' (*grayscale*) (b) 'yacht' (berwarna)

4.1 Hasil Enkripsi dan Dekripsi

Citra hasil enkripsi (*cipher-image*) masing-masing diperlihatkan pada Gambar 5(a) dan 5(b). Citra hasil enkripsi terlihat sudah tidak dapat dikenali lagi dan tampak seperti citra acak. Dekripsi terhadap *cipher-image* dengan parameter kunci yang sama menghasilkan kembali tepat seperti citra semula (Gambar 5(c) dan 5(d)).



(a) (b)

(c) (b)

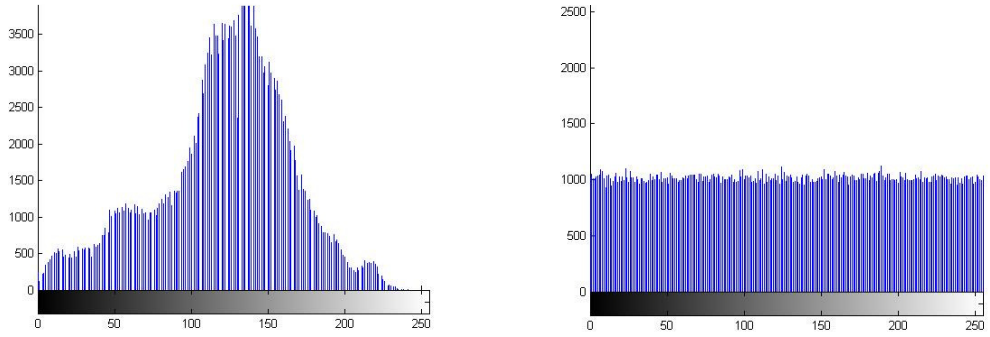
Gambar 5. (a) *cipher-image* dari ‘couple’; (b) *cipher-image* dari ‘yacht’; (c) hasil dekripsi citra ‘couple’; (d) hasil dekripsi citra ‘yacht’

4.2 Analisis Histogram

Histogram merupakan salah satu fitur citra yang penting, sebab sebuah histogram memperlihatkan distribusi intensitas *pixel-pixel* di dalam citra tersebut. Dalam melakukan serangan dengan teknika analisis statistik, penyerang menggunakan histogram untuk

menganalisis frekuensi kemunculan intensitas *pixel* untuk mendeduksi kunci atau *pixel-pixel* di dalam *plain-image*. Agar serangan dengan analisis statistik tidak dimungkinkan, maka di dalam enkripsi citra penting untuk menghasilkan histogram *cipher-image* yang tidak memiliki kemiripan secara statistik dengan histogram *plain-image*. Oleh karena itu, *pixel-pixel* di dalam *cipher-image* seharusnya memiliki distribusi yang (relatif) *uniform* atau ditunjukkan dengan histogram yang terlihat datar (*flat*).

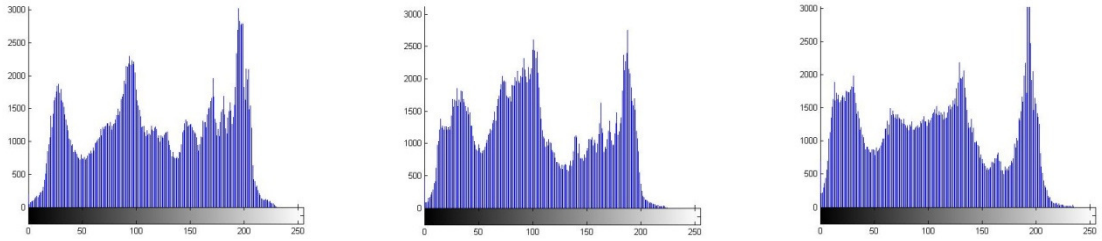
Gambar 6(a) memperlihatkan histogram citra 'couple' dan Gambar 6(b) adalah histogram *cipher-image*-nya. Histogram *cipher-image* terlihat datar dan berbeda dengan histogram *plain-image*.



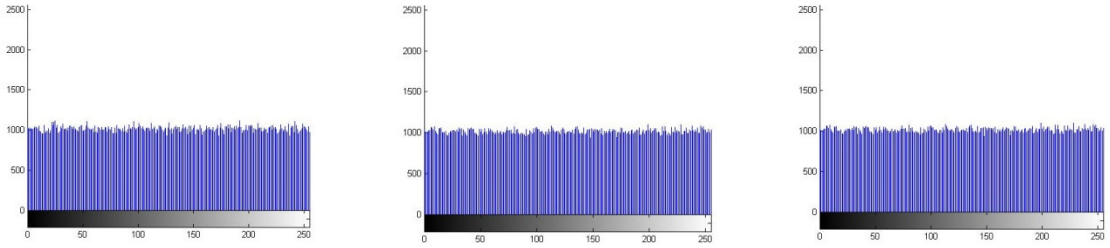
(a) (b)

Gambar 6. (a) Histogram citra 'couple' (*plain-image*) dan (b) histogram *cipher-image*.

Gambar 7(a) sampai 7(c) adalah histogram citra 'yacht' (*plain-image*) untuk setiap kanal warna *RGB*, sedangkan Gambar 7(d) sampai 7(f) adalah histogram masing-masang kanal warna pada *cipher-image*. Seperti pada citra 'couple', histogram setiap kanal *RGB* pada *cipher-image* juga berbentuk *flat* atau terdistribusi *uniform*.



(a) (b) (c)



(d) (e) (f)

Gambar 7. (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 besaran statistik yang menyatakan kekuatan hubungan linier antara dua peubah acak (Xiang, 2007). Koefisien korelasi (r_{xy}) dari dua buah peubah acak diskrit yang masing-masing beranggotakan *n* elemen dihitung dengan rumus berikut (Hongmei, 2010):

$$r_{xy} = \frac{\text{cov}(x, y)}{\sqrt{D(x)D(y)}} \tag{5}$$

yang dalam hal ini

$$\text{cov}(x, y) = \frac{1}{n}\sum_{i=1}^{n}[x_i - E(x)][y_i - E(y)] \quad \text{(kovariansi)} \tag{6}$$

$$D(x) = \frac{1}{n}\sum_{i=1}^{n}[x_i - E(x)]^2 \quad \text{(standard deviasi)} \tag{7}$$

$$E(x) = \frac{1}{n}\sum_{i=1}^{n} x_i \quad \text{(rata-rata)} \tag{8}$$

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.

Pada kebanyakan citra *plain-image*, koefisien korelasi antara *pixel-pixel* bertetangga biasanya tinggi (mendekati +1 atau -1). Tujuan enkripsi citra adalah menghilangkan korelasi antara *pixel-pixel* tersebut atau membuat koefisien korelasinya mendekati nol.

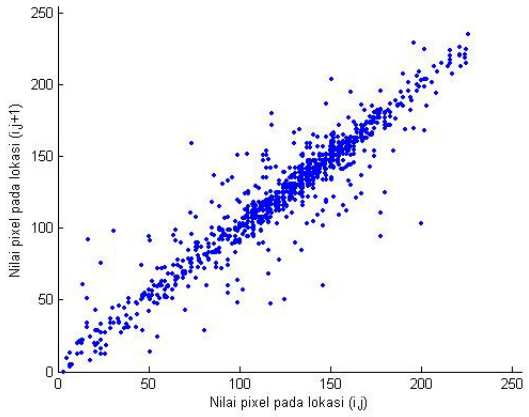
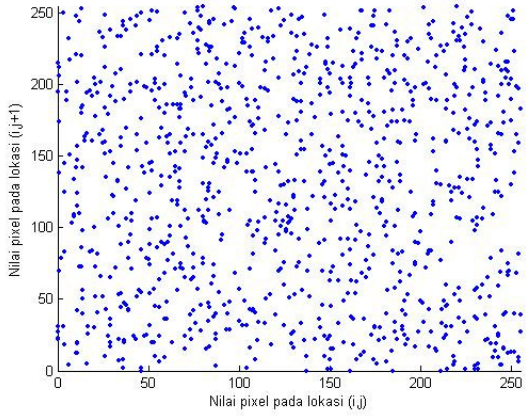
Untuk menyelidiki korelasi pada *plain-image* dan *cipher-image*, maka dihitung koefisien korelasi antara dua *pixel* yang bertetangga secara horizontal [$f(i,j)$ dan $f(i, j+1)$], dua *pixel* yang bertetangga secara vertkal [$f(i,j)$ dan $f(i+1, j)$], dan dua *pixel* yang bertetangga secara diagonal $f(i,j)$ dan $f(i+1, j+1)$], baik pada *plain-image* maupun pada *cipher-image*.

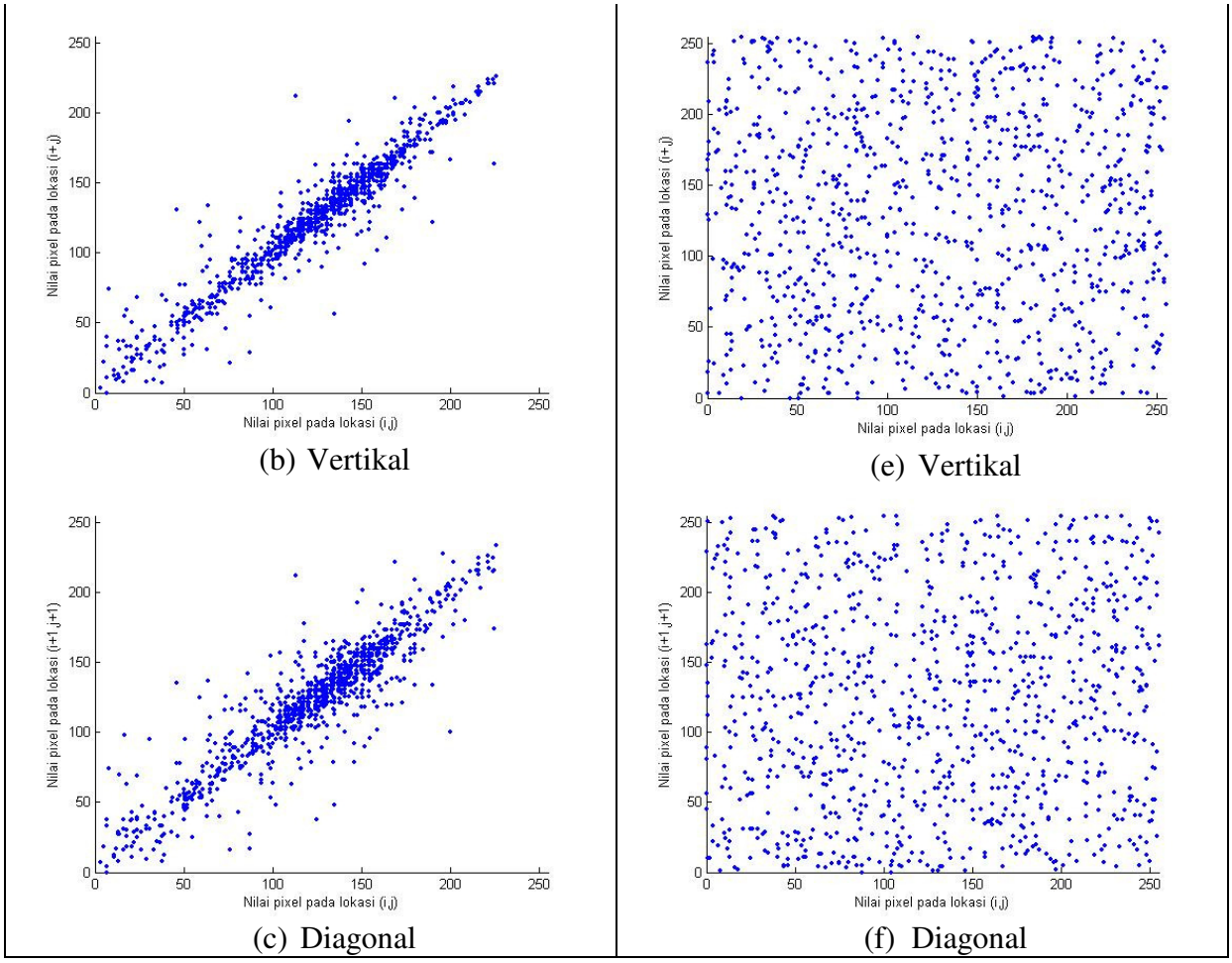
Di dalam eksperimen ini dipilih secara acak 1000 pasang *pixel* bertetangga pada setiap arah (vertikal, horizontal, dan diagonal) dari citra 'couple' beserta citra hasil enkripsinya. Koefisien korelasi dihitung dengan persamaan (5), yang dalam hal ini x dan y adalah nilai keabuan dari dua *pixel* bertetangga. Hasil eksperimen diperlihatkan pada Tabel 1.

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

Koefisien korelasi	Horizontal	Vertikal	Diagonal
Plain-image	0.9364	0.9422	0.9046
Cipher-image	-0.0142	0.0244	0.0471

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

Plain-image	***Cipher-image***
	
(a) Horizontal	(d) Horizontal



Gambar 8. Distribusi korelasi *pixel-pixel* bertetangga pada *plain-image* dan *cipher-image*

Untuk memperlihatkan situasi yang lebih jelas mengenai korelasi, Gambar 8 memperlihatkan distribusi korelasi *pixel-pixel* yang bertetangga pada masing-masing *plain-image* (kolom kiri) dan *cipher-image* (kolom kanan). Pada *plain-image*, *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.

4.5 Analisis Sensitivitas

Untuk mengetahui sensitivitas *chaos* terhadap perubahan kecil nilai awal, maka dilakukan

eksperimen dengan mengubah nilai awal *logistic map* (x_0) sebesar Δ menjadi $x_0 + \Delta$. Selanjutnya *cipher-image* didekripsi dengan $x_0 + \Delta$. Misalkan $\Delta = 10^{-10}$ sehingga nilai awal *logistic map* menjadi 0.3000000001. Hasil dekripsi terhadap *cipher-image* dengan nilai awal 0.3000000001 diperlihatkan pada Gambar 9(a), yang ternyata tetap seperti citra acak (tidak kembali menjadi citra semula). Eksperimen ini menunjukkan bahwa *chaos* menenuhi prinsip *diffusion* dari Shanon (Schneier, 1996), sehingga serangan *brute force* akan gagal karena perubahan satu bit saja pada kunci menyebabkan hasil dekripsi tetap salah.

4.4 Ruang Kunci

Ruang kunci menyatakan jumlah total kunci yang berbeda yang dapat digunakan unruk enkripsi/dekripsi (Fu, 2012). Ruang kunci seharusnya cukup besar agar serangan *brute-force attack* menjadi tidak efisien dilakukan. Parameter kunci rahasia yang digunakan di dalam algoritma enkripsi lebih dari satu buah, yaitu b, c, m, x_0, dan r. Tiga parameter pertama, b, c, dan m adalah *integer* positif. Matlab mendukung maksimum *unsigned integer* hingga 32 bit, sehingga nilai pilihan *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} (Fu, 2012), sehingga jumlah kemungkinan nilai x_0 adalah 10^{15}. Dengan demikian, ruang kunci seluruhnya adalah

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

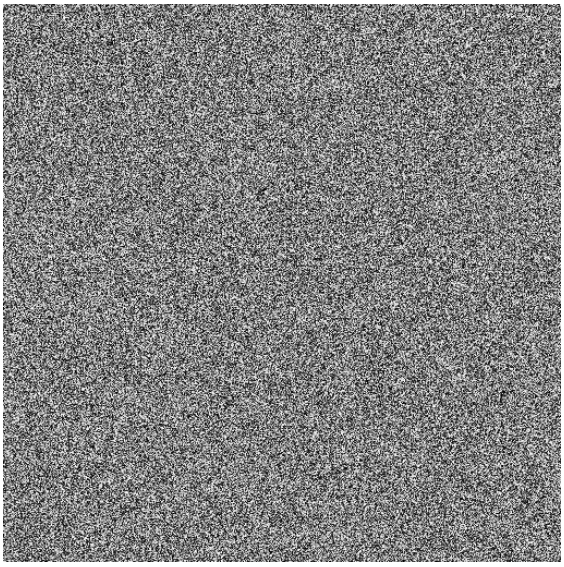
yang cukup besar bertahan terhadap serangan *brute-force attack.*

5. Kesimpulan

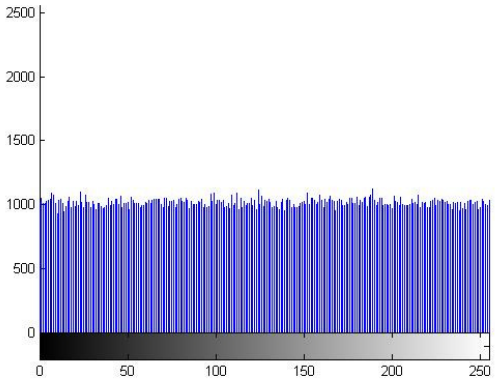
Di dalam makalah telah dipresentasikan sebuah algoritma enkripsi citra digital berbasis *chaos* yang menggabungkan teknik permutasi dan teknik substitusi. Teknik permutasi menggunakan *Arnold Cat Map*, dan teknik substitusi menggunakan *Logistic Map*.

Melalui simulasi eksperimen, algoritma ini dapat mengenkripsi sembarang citra (baik citra grayscale maupun citra berwarna) dengan baik. Citra hasil enkripsi (*cipher-image*) terlihat seperti citra acak dan sudah tidak dapat dikenali lagi. Analisis histogram memperlihatkan *pixel-pixel* di dalam *cipher-image* mempunyai distribusi *uniform,* sedangkan analisis korelasi menunjukkan bahwa *pixel-pixel* di dalam *cipher-image* tidak berkorelasi, sehingga membuat serangan dengan analisis statistik menjadi sulit.

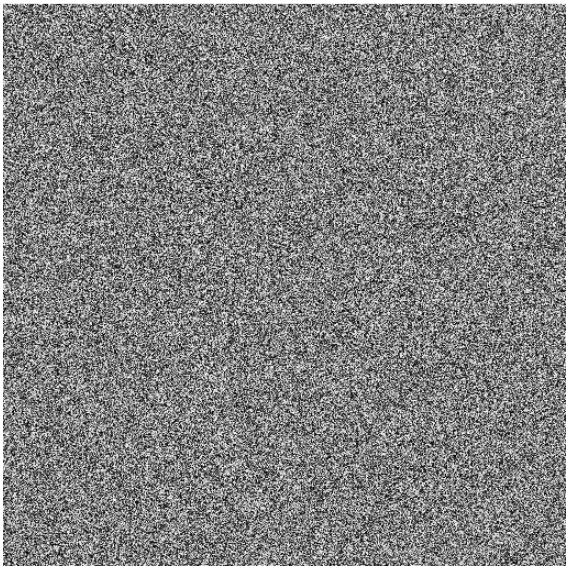
Sifat *chaos* yang sensitif terhadap perubahan kecil nilai awal telah ditunjukkan dengan melakukan perubahan kecil pada kunci yang menyebabkan citra tidak berhasil diedekripsi, sehingga algoritma ini aman dari serangan *exhaustive-key search attack.*



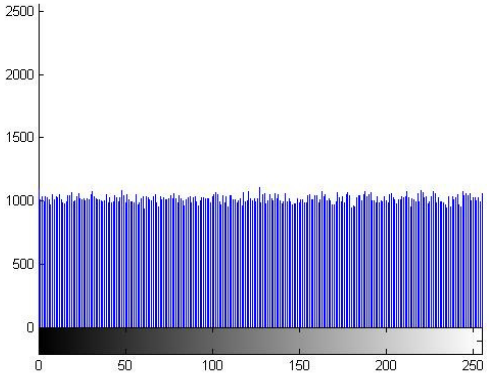
(a) *Cipher-image* citra 'couple'



(b) Histogram dari citra (a)



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



(d) Histogram dari citra (c)

Gambar 9. Hasil eksperimen dekripsi dengan pengubahan x_0 sebesar $\Delta = 10^{-10}$.

ACKNOWLEDGMENT

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

PUSTAKA

1. Sharma, M., Kowar, M.K. (2010), *Image Encryption Technique Using Chaotic Schemes: A Review*, International Journal of Engineering, Science, and Technology Vol 2 (6) 2010.
2. Schneier, B. (1996), *Applied Cryptography 2nd Edition*, Wiley & Sons.
3. Hongmei, T., Liying, H., Yu, H., Xia, W., (2010), *An Improved Compound Image Encryption Scheme*, Proceeding of 2010 International Conference on Computer and Communication Technologies in Agriculture Engineering.
4. Wei-bin, C., Xin, Z. (2009): *Image Encryption Algorithm Based on Henon Chaotic System*, Proceeding of International Conference on Image Analysis and Signal Processing (IASP 2009).
5. Struss, K. (2009), A Chaotic Image Encryption, *Mathematics Senior Seminar*, 4901, University of Minnesota, Morris.
6. Yu, X., Zhang, J., Ren, H., Xu, G., dan Luo, X. (2006), Chaotic Scrambling Algorithm Based on S-DES, *Journal of Physics: Conference Series 48*, 349-353.
7. Xiang, T, Wong, K., dan Liao, X. (2007), Selective Image Encryption Using a spatiotemporal Chaotic System, *Chaos Volume 17*.