KOMBINASI *LOGISTIC MAP* DAN *PSEUDO-RANDOM NUMBER GENERATOR* PADA PEMBANGKITAN KUNCI UNTUK ENKRIPSI CITRA DIGITAL

[1]Rini Arianty, [2]Diana Tri Susetianingtias
[1,2]Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Gunadarma
[1]Jl. Margonda Raya 100, Depok 16424, Jawa Barat
[1]rinia@staff.gunadarma.ac.id, [2]diants@staff.gunadarma.ac.id

Abstrak

Informasi berbentuk gambar yang bersifat sensitif atau rahasia, seperti data pribadi, dokumen penting yang dikirimkan melalui internet belum tentu aman dari serangan pihak luar. Kerugian yang cukup besar dapat ditimbulkan apabila data tersebut diakses dan dimanipulasi oleh orang yang tidak bertanggung jawab. Salah satu metode dalam mengamankan suatu informasi adalah kriptografi. Logistic map adalah salah satu algoritma chaos yang sering digunakan dalam kriptografi citra karena algoritma ini mampu menghasilkan deretan bilangan acak yang kompleks dengan persamaan polinomial rekursif yang sederhana. Pada penelitian ini, akan diimplementasikan algoritma chaos logistic map dan pseudo-random number generator (PRNG) dalam pengenkripsian citra. Citra input akan diubah bentuknya kedalam array lalu proses difusi dilakukan secara selektif dengan mensubstitusi 4 bit MSB setiap nilai warna citra dengan kunci logistic map. Hasil difusi tersebut akan dikonfusi dengan cara mensubstitusikan indeks arraynya dengan kunci prng sehingga didapat sebuah array baru yang teracak indeksnya. Array tersebut diubah kembali menjadi sebuah citra sehingga didapat citra terenkripsi yang aman.

***Kata Kunci**: Difusi, Konfusi, Kunci, Logistic Map, Pseudo-Random Number Generator*

Abstract

Information in the form of sensitive or confidential images, such as personal data, important documents which sent over the internet is not necessarily safe from an attack by an outside parties. Significant losses can be caused if the data is accessed and manipulated by irresponsible people. There is a method of securing information called cryptography. Logistic map is one of the chaotic algorithms that is often used in image cryptography because this algorithm is able to generate a complex random number series with simple recursive polynomial equations. In this research, a chaotic logistic map algorithm and pseudo-random number generator (PRNG) will be implemented in image encryption. The input image will be transformed into an array then the diffusion process is carried out selectively by substituting 4 MSB of each image color value with the logistic map key. The diffusion result will be confused by substituting the array index with the prng key so that a new array with randomized index is created. The array will be converted back into an image in order to obtain a secure encrypted image.

***Key Words**: Difusi, Konfusi, Kunci, Logistic Map, Pseudo-Random Number Generator*

PENDAHULUAN

Informasi yang dikirimkan melalui media internet sangat rentan terjadi serangan keamanan dari pihak luar karena resiko kemudahan akses dan manipulasi pada jaringan komputer [1] oleh pihak tidak berwenang. Informasi tersebut dapat berupa gambar (citra), dokumen yang sensitif atau rahasia, seperti data pribadi, dokumen perusahaan, dokumen negara, dan sebagainya.

Hal ini tentu saja dapat merugikan pemilik atau pihak yang bersangkutan dengan informasi tersebut. Salah satu metode dalam mengamankan suatu informasi pada data digital berbentuk citra adalah kriptografi. Kriptografi merupakan salah satu teknik untuk mengamankan sebuah pesan asli (*plaintext*) dengan cara mengacaukan struktur dan bentuknya menggunakan sebuah kunci enkripsi (*key*) sehingga pesan tersebut menjadi sulit dan tidak dapat dimengerti oleh orang yang tidak memiliki kunci untuk memecahkan pengamanan pada pesan tersebut [2]. Algoritma kriptografi yang dapat diimplementasikan untuk data digital seperti citra salah satunya adalah logistic map. Secara umum algoritma persamaan *logistic map* merupakan salah satu algoritma yang sederhana tapi efektif dalam membangkitkan bilangan acak [3] pada saat proses membangkitkan kunci dalam proses enkripsi dan dekripsi sebuah citra. Dalam kriptografi citra, terdapat dua proses yang harus dilakukan agar suatu citra dapat terenkripsi dengan baik. Proses tersebut adalah konfusi dan difusi [4]. Proses konfusi bertujuan untuk menyamarkan hubungan antara *plain image* dan *key* dan meningkatkan ambiguitas hasil enkripsi sedangkan difusi bertujuan untuk meningkatkan redudansi data sehingga hubungan antara citra asli dan citra hasil enkripsi menjadi tidak jelas dan tidak berpola.

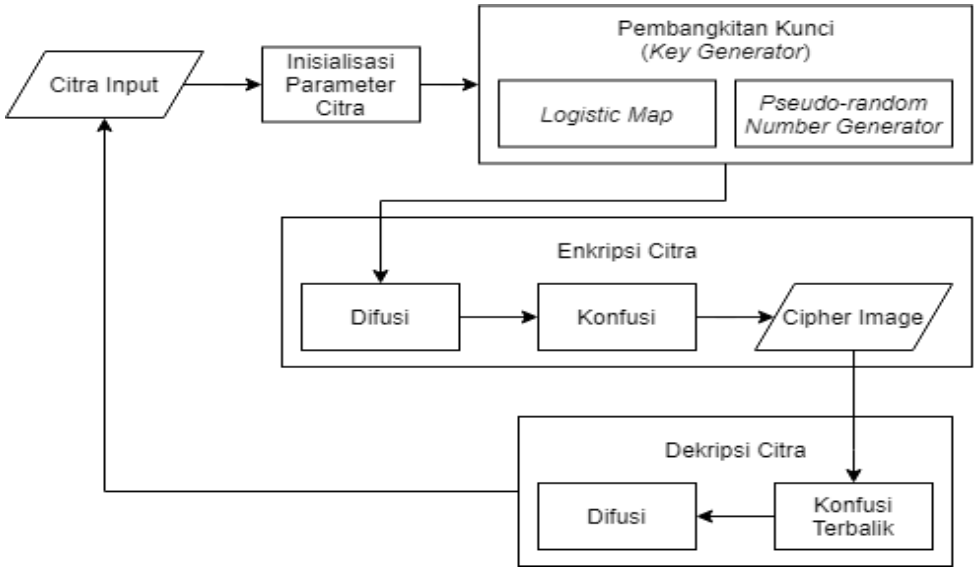
Penelitian terkait kriptografi citra menggunakan algoritma *chaos logistic map* sudah dilakukan oleh beberapa peneliti terdahulu. Pada kriptografi modern, digunakan teori *chaos* untuk membangkitkan bilangan secara acak. Teori *chaos* ini sangat sensitif pada nilai awal (*initial condition*) [5], sehingga dapat digunakan sebagai pembangkit kunci acak dalam proses pembangkitan kunci (*key generation*). Semakin acak bilangan yang dihasilkan, semakin baik pula tingkat keamanan [5] dari suatu ciphertekS.

Penelitian [6] menggunakan algoritma *chaos Arnold's cat map* untuk melakukan proses konfusi dan *2D logistic map* untuk proses difusi dalam melakukan kriptografi citra. Penelitian [7] juga menggunakan algoritma *chaos Arnold's cat map* pada proses konfusi dan algoritma *logistic map* pada proses difusi yang dimodifikasi yang disebut *Efficient Diffusion*. Penelitian [8] melakukan kriptografi citra dengan mendifusi citra menggunakan *logistic map* lalu hasil difusi tersebut ditransformasikan menggunakan algoritma *discrete fractional angular transform* (DFAT) sehingga didapat citra yang terenkripsi. Penelitian [9] menggunakan *logistic map* dengan parameter yang bervariasi untuk melakukan proses konfusi kemudian dilanjutkan dengan proses difusi menggunakan algoritma *dynamical encryption*. Penelitian [10] menggabungkan algoritma *logistic map* dengan *tent map* dan *sine map* yang disebut dengan *tent delay-sine cascade with logistic map* (TDSCL) yang dapat melakukan proses konfusi dan difusi secara bersamaan dalam melakukan kriptografi citra.

Pada penelitian ini, akan di-implementasikan algoritma *chaos logistic map* dan *pseudo-random number generator* (PRNG) dalam mengenkripsi citra *grayscale* dan citra berwarna. Proses konfusi dilakukan dengan menggunakan algoritma PRNG sedangkan proses difusi dilakukan dengan menggunakan kunci yang dihasilkan oleh algoritma *logistic map* berbentuk biner sepanjang 4 bit. Proses difusi dilakukan dengan cara mensubstitusi XOR kunci *logistic map* dengan *Most Significant Bits* (MSB) setiap warna citra RGB (*Red, Green, Blue*). Hasil penelitian ini diharapkan dapat meningkatkan keamanan pada citra digital khususnya sebelumnya dikirimkan melalui jaringan komputer. Pemanfaatan pembangkitan kunci secara chaos (kacau) diharapkan dapat menghindarkan kriptoanalis untuk menebak kunci yang dibangkitkan baik pada proses enkripsi maupun dekripsi citra.

METODE PENELITIAN

Penelitian ini terdiri dari beberapa tahapan proses dalam melakukan kriptografi pada citra digital. Gambaran umum diagram alir program pada penelitian ini dapat dilihat pada Gambar 1. Langkah pertama dilakukan untuk mendapatkan atribut citra seperti lebar citra, tinggi citra, dan warna RGB dari citra input. Atribut-atribut tersebut disimpan ke dalam variabelnya masing-masing lalu dilanjutkan dengan pembangkitan *keystream* atau kunci enkripsi menggunakan algoritma *logistic map* untuk proses difusi dan *pseudo-random number generator* untuk proses konfusi.



Gambar 1. Gambaran Umum Tahapan Metode Penelitian

https://doi.org/10.35760/tr.2020.v25i3.3120

Citra Input

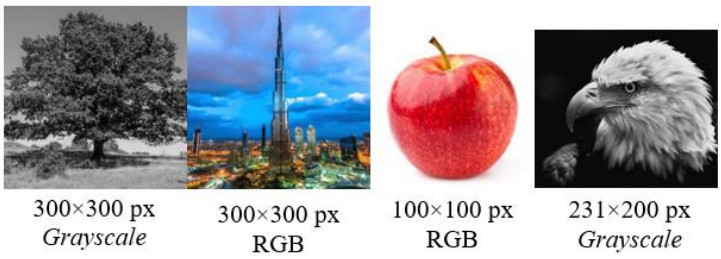
Citra input berupa *file* citra *grayscale* maupun citra RGB yang berekstensi *.jpg*, *.png*, atau *.bmp*.. Penelitian ini menggunakan sampel dengan ukuran maksimal 300x300 piksel, akan tetapi dapat juga digunakan citra digital dengan ukuran lebar dan tinggi yang berbeda (misalnya: 231×200 piksel) seperti dapat dilihat pada Gambar 2.

Inisialisasi Parameter Citra

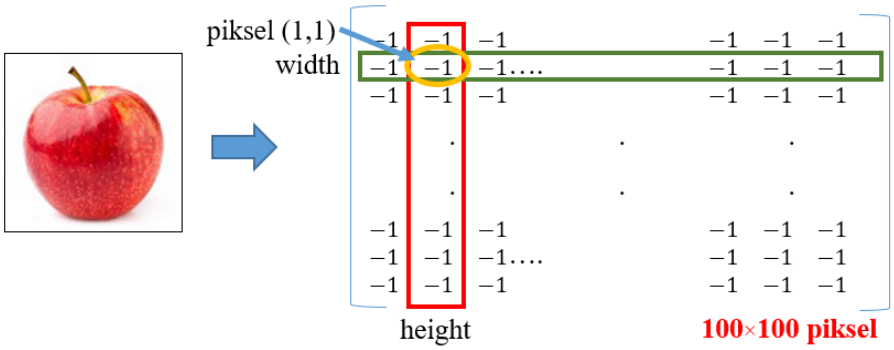
Inisialisasi parameter citra dilakukan untuk untuk membentuk matriks dengan atribut pada citra input seperti lebar, tinggi serta komponen warna RGB setiap pikselnya yang diperlukan dalam proses enkripsi dan dekripsi. Bitmap dan Color merupakan *library-library* Java yang menyediakan ber-bagai fungsi yang dapat digunakan untuk mengambil dan memanipulasi atribut dari suatu citra. Sebagai contoh, deskripsi inisialisasi parameter pada *input* citra apel dapat dilihat pada Gambar 3.

Citra input akan dikonversikan ke dalam bentuk matriks. Citra sampel yang digunakan memiliki ukuran 100×100 piksel seperti terlihat pada Gambar 3. Matriks dari citra tersebut memiliki sejumlah angka yang menunjukkan nilai warna setiap pikselnya. Kemudian angka "-1" yang dilingkari warna kuning akan dipecah oleh fungsi pada kelas *color* sehingga menghasilkan warna dasar merah, hijau, dan biru yang kemudian akan dikonversikan ke bilangan biner 8-bit.

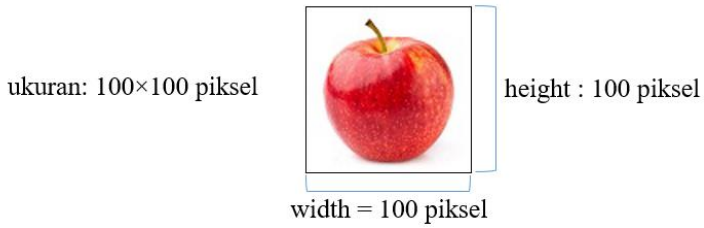
Inisialisasi input parameter yang digunakan pada penelitian ini antara lain :



Gambar 2. Contoh Citra Input



Gambar 3. Deskripsi Inisialisasi Parameter Citra Apel



Gambar 4. Penentuan Parameter Lebar dan Tinggi Citra

1. Parameter Lebar dan Tinggi Citra

Lebar dan tinggi citra input merupakan sebuah bilangan bulat yang dapat diakses dengan menggunakan fungsi getWidth() dan getHeight() yang disediakan oleh kelas *Bitmap*. Lebar dan tinggi citra akan digunakan sebagai parameter penentu dalam pembangkitan kunci enkripsi. Gambar 4 merupakan contoh citra input yang digunakan sebagai sampel berukuran 100×100 piksel.

2. Parameter Komponen RGB Citra

Komponen warna RGB setiap piksel pada citra diperlukan agar dapat dimanipulasi sehingga citra dapat terenkripsi. Dalam melakukan pengambilan komponen warna RGB citra digunakan fungsi getPixel(x,y) untuk mengambil nilai warna dari suatu piksel pada citra . Nilai masing-masing warna dapat dipecah ke dalam menggunakan fungsi yang disediakan oleh kelas *Color*. Fungsi tersebut akan mengambil nilai masing-masing warna sehingga dapat disimpan kedalam sebuah variabel. Sebagai contoh, nilai warna merah pada gambar 4 adalah 176. Angka-angka tersebut akan dikonversikan kedalam bentuk biner sepanjang 8-bit seperti berikut.

176 = 1 0 1 1 0 0 0 0 $_2$

4-bit terdepan yaitu 1 0 1 1 merupakan MSB dari bilangan tersebut. 4-bit MSB tersebut akan di-XOR-kan dengan kunci enkripsi lalu hasilnya akan menggantikan 4-bit MSB citra asli sehingga didapat nilai warna baru.

Pembangkitan Kunci Enkripsi Citra

Proses pembangkitan kunci enkripsi dilakukan pada proses difusi dan konfusi pada citra menggunakan *logistic map* digunakan untuk membangkitkan kunci proses difusi sedangkan *pseudo-random number generator* untuk membangkitkan kunci proses konfusi dengan penjelasan sebagai berikut :

1. Logistic Map

Dalam penelitian ini, *logistic map* digunakan sebagai pembangkit kunci enkripsi untuk proses difusi. Parameter nilai awal x_0 dan konstanta λ yang digunakan berupa bilangan desimal bertipe *Double*. Berikut adalah prosedur pembangkitan kunci enkripsi menggunakan algoritma *logistic map*:

1) Nilai awal (*x0*) *logistic map* didapatkan menggunakan persamaan [3] :

$$\frac{lebar\,citra + tinggi\,citra}{lebar\,citra \times tinggi\,citra} \quad (1)$$

https://doi.org/10.35760/tr.2020.v25i3.3120

Persamaan 1 akan selalu menghasilkan sebuah nilai desimal yang berkisar diantara angka 0 dan 1 selama lebar dan tinggi citra lebih besar dari 2 piksel. Contoh sampel yang digunakan berukuran 100×100 piksel, maka lebarnya adalah 100 dan tingginya 100 sehingga nilai awal (*x0*) yang dihasilkan oleh persamaan 1 adalah 0.02.

2) Konstanta λ yang digunakan dalam penelitian ini adalah 4, karena akan menghasilkan kunci yang benar-benar acak saat konstanta λ bernilai 4 [3] sesuai dengan teori *logistic map*. Penggunaan nilai awal (*x0*) = 0.02 dan konstanta λ = 4, 5 angka pertama yang dihasilkan oleh algoritma *logistic map* [3] secara berurutan adalah 0.0784, 0.28901376, 0.821939226, 0.585420539, dan 0.970813326. Terlihat bahwa angka-angka tersebut terlihat acak [3].
3) Setiap angka tersebut akan diubah ke dalam tipe data *String* lalu di*substring* sehingga menyisakan 4 angka yang berada di belakang koma sehingga menjadi: 0784, 2890, 8219, 5854, dan 9708.
4) Angka-angka tersebut akan diubah menjadi *BinaryString* sepanjang 8-bit sehingga didapat 10000, 1001010, 11011, 11011110, dan 11101100.
5) Setiap *BinaryString* akan di*substring* sehingga menyisakan 4-bit terdepan sampai didapat kunci berikut: 1000, 1001, 1101, 1101, dan 1110. *Padding bits* akan dilakukan untuk mengisi bit-bit yang kosong dengan cara menambahkan karakter "0" didepan bilangan tersebut apabila panjangnya tidak sampai 4-bit agar setiap kunci memiliki panjang 4-bit.
6) Kunci yang dihasilkan tersebut disimpan kedalam sebuah array Integer sehingga didapat deretan kunci *logistic map* sepanjang 4-bit.

Ilustrasi sederhana pembuatan kunci *logistic map* pada angka pertama yang dihasilkan oleh algoritma tersebut dari bentuk angka desimal hingga menjadi bilangan biner berukuran 4-bit dapat dilihat pada sederetan angka berikut:

Angka ke-1 ⇨	*substring* ⇨	*BinaryString* ⇨	*substring*
0.0784	0784	10000	1000

2. Pseudo-random Number Generator

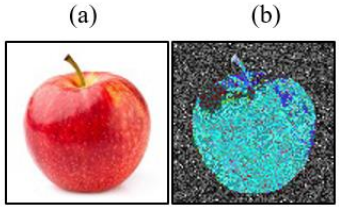
Pseudo-random number generator (PRNG) digunakan pada proses konfusi sebagai pembangkit bilangan acak semu dengan menggunakan *seed* yang ditentukan dari lebar citra. Berikut adalah tahapan yang dilakukan dalam membangkitkan kunci enkripsi untuk proses konfusi.

1. Mendeklarasikan dua buah array satu dimensi sepanjang total piksel citra dikurangi 1 karena indeks array di-mulai dari angka 0.
2. Berikutnya membuat fungsi *random* menggunakan *seed* yang ditentukan berdasarkan lebar citra. Citra sampel yang digunakan memiliki resolusi 100×100 piksel, maka *seed* yang di-gunakan untuk citra tersebut adalah 100.
3. Penggunaan *seed* pada *random number generator* bertujuan agar fungsi random ini dapat menghasilkan deretan bilangan acak yang sama setiap kali dieksekusi. Lebar citra sebagai penentu *seed* bertujuan agar deretan bilangan acak yang dihasilkan akan berbeda jika lebar citra berbeda walaupun total resolusinya sama.
4. Array pertama akan diisi oleh indeks array citra yaitu angka-angka dari 0 sampai total piksel citra dikurangi 1 lalu setiap elemennya diacak dengan menggunakan *pseudo-random number generator* yang dibuat menggunakan fungsi *random*.
5. Array kedua akan menyimpan hasil pengacakan indeks array pertama tersebut. Array inilah yang akan digunakan sebagai kunci PRNG pada proses konfusi.

Enkripsi Citra

Dalam melakukan proses enkripsi, proses difusi dieksekusi terlebih dahulu untuk mengubah komposisi nilai warna setiap piksel pada citra lalu dilanjutkan dengan proses konfusi. Proses konfusi adalah proses untuk mengacak posisi dari setiap piksel yang ada pada citra sehingga dapat menghasilkan citra baru yang terenkripsi.

1. **Difusi.** Proses difusi berfungsi untuk mengubah atau memodifikasi komponen nilai warna pada setiap piksel citra sehingga hubungan antar piksel citra menjadi tersamarkan seperti dapat dilihat pada gambar 5.



Gambar 5. Hasil Difusi Proses Enkripsi

https://doi.org/10.35760/tr.2020.v25i3.3120

Proses difusi dilakukan dengan cara sebagai berikut :

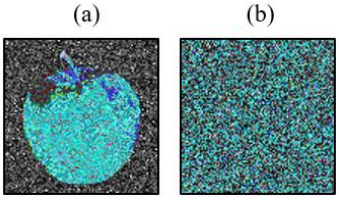
1) Mengambil nilai warna merah, hijau dan biru pada piksel (x,y) lalu mengubahnya menjadi tipe data *String* agar dapat diambil 8-bit *BinaryString* setiap komponen warna pada citra.
2) Melakukan *padding bits* apabila panjangnya tidak sampai 8-bit lalu setiap *BinaryString* akan disimpan kedalam sebuah array Integer.
3) Proses difusi dilakukan secara selektif dengan mensubstitusi XOR kunci *logistic map* dengan 4-bit MSB (4-bit terdepan dari bilangan biner) setiap nilai warna.
4) Hasil substitusi XOR tersebut akan menggantikan 4-bit MSB komponen citra asli sehingga didapatlah nilai warna baru. Nilai warna merah, hijau, dan biru yang telah didifusi akan digabungkan menjadi sebuah nilai warna RGB lalu disimpan kedalam sebuah array baru.
5) Langkah 1-4 akan diulang sampai setiap piksel selesai diproses sehingga didapat hasil difusi berupa array yang berisi nilai warna baru.

2. Konfusi

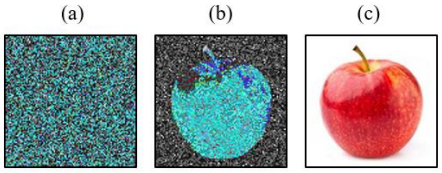
Proses konfusi bertujuan untuk mengacak posisi piksel citra sehingga citra menjadi tidak dapat dikenali. Hasil dari proses difusi yang dilakukan sebelumnya akan dikonfusi menggunakan kunci PRNG. Proses konfusi dilakukan dengan cara mensubstitusi indeks citra dengan kunci PRNG sehingga didapat sebuah array baru yang merupakan array citra yang sudah terenkripsi. *Pseudocode* dibawah ini adalah perulangan yang dilakukan untuk menukar posisi setiap elemen citra sesuai dengan indeks yang telah diacak menggunakan PRNG. Citra hasil konfusi yang diperoleh dapat dilihat pada Gambar 6.

Dekripsi Citra

Proses dekripsi bertujuan agar citra terenkrip dapat dikembalikan ke bentuk semula sehingga bisa dimengerti. Proses dekripsi dilakukan dengan cara melakukan konfusi terbalik terlebih dahulu kemudian dilanjutkan dengan proses difusi sehingga citra asli didapat kembali.



Gambar 6. Hasil Konfusi Citra: (a) Hasil Difusi; (b) Hasil Konfusi



Gambar 7. Hasil Difusi Proses Dekripsi: (a) Citra Terenkripsi; (b) Hasil Konfusi Terbalik; dan (c) Hasil Dekripsi

1. Konfusi Terbalik

Konfusi terbalik dilakukan untuk mengembalikan posisi indeks setiap elemen citra ke posisi semula sebelum dikonfusi. Setiap piksel citra terenkripsi akan disimpan kedalam sebuah array. Array baru akan dibuat untuk menampung hasil konfusi terbalik yang dilakukan.

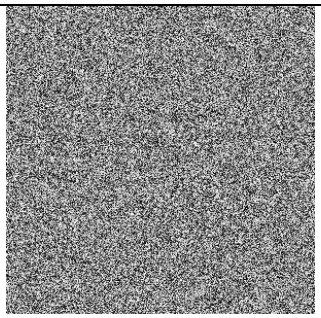
2. Difusi

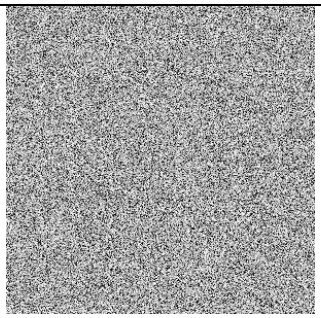
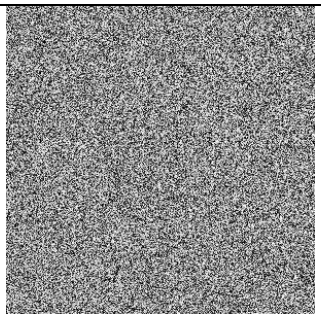
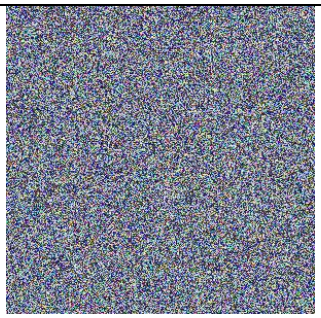
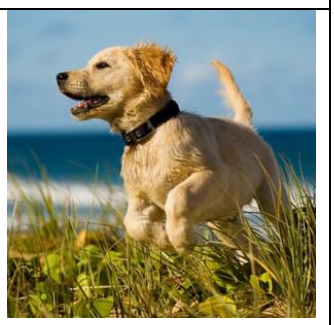
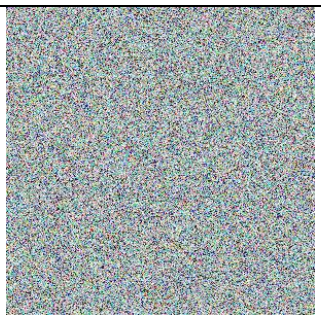
Langkah proses difusi yang dilakukan pada proses dekripsi sama seperti yang dilakukan pada proses enkripsi. Setiap elemen dari array yang didapat dari hasil konfusi terbalik akan disubstitusi XOR lagi dengan menggunakan kunci *logistic map* sehingga didapat kembali nilai warna citra asli. Perbandingan hasil proses dekripsi terhadap citra apel yang sudah dienkripsi sebelumnya terdapat pada Gambar 7.

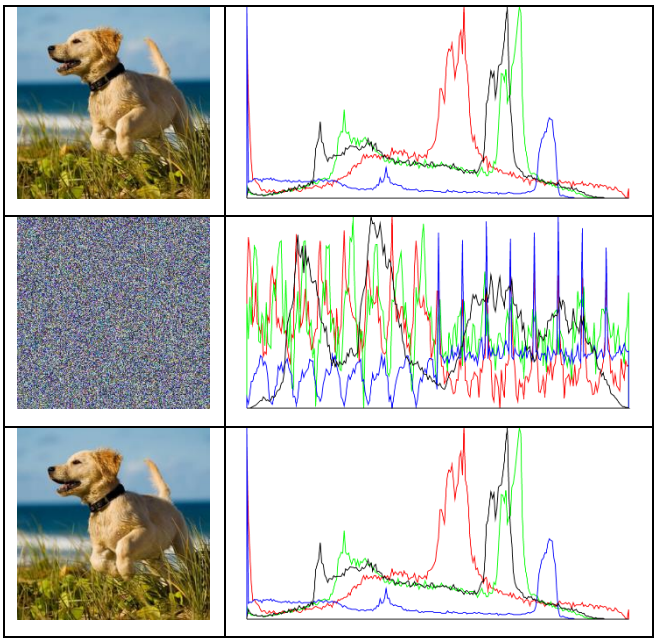
HASIL DAN PEMBAHASAN

Pada tahap uji coba, setiap sampel citra input akan dienkripsi kemudian hasil enkripsi tersebut akan didekrips. Sebagai contoh, percobaan dilakukan dengan menggunakan 5 sampel citra seperti dapat dilihat pada Tabel 1.

Tabel 1. Hasil Uji Coba Percobaan Pertama

No	Citra Asli	Hasil Enkripsi	Hasil Dekripsi
1	 Resolusi: 300×300px Ukuran: 123 KB	 Resolusi: 300×300px Ukuran: 280 KB	 Resolusi: 300×300px Ukuran: 135 KB

No	Citra Asli	Hasil Enkripsi	Hasil Dekripsi
2	 Resolusi: 300×300px Ukuran: 149 KB	 Resolusi: 300×300px Ukuran: 260 KB	 Resolusi: 300×300px Ukuran: 166 KB
3	 Resolusi: 300×300px Ukuran: 141 KB	 Resolusi: 300×300px Ukuran: 279 KB	 Resolusi: 300×300px Ukuran: 160 KB
4	 Resolusi: 300×300px Ukuran: 146 KB	 Resolusi: 300×300px Ukuran: 306 KB	 Resolusi: 300×300px Ukuran: 160 KB
5	 Resolusi: 300×300px Ukuran: 211 KB	 Resolusi: 300×300px Ukuran: 298 KB	 Resolusi: 300×300px Ukuran: 227 KB



Gambar 8. Contoh Evaluasi Hasil Histogram

Berdasarkan Tabel 1, citra hasil enkripsi menjadi tidak dapat dikenali lagi sedangkan citra hasil dekripsi terlihat sama dengan citra asli. Resolusi citra tidak berubah setelah proses enkripsi dan dekripsi dilakukan sedangkan ukuran citra berubah-ubah pada proses enkripsi dan dekripsi.

Evaluasi terhadap histogram dari citra asli juga dilakukan, dimana hasil enkripsi dan hasil dekripsi setiap sampel. Sebagai contoh pada sampel ke 4, Histogram ditampilkan agar dapat dibandingkan dengan histogram lainnya. Rincian histogram untuk setiap citra asli dapat dilihat pada Gambar 8.

Seperti dapat dilihat pada Gambar 8, terdapat perbedaan yang signifikan antara histogram citra asli dengan histogram citra hasil enkripsi. Hal ini menandakan bahwa citra asli sudah berhasil tersamarkan. Histogram citra hasil dekripsi yang sama dengan histogram citra asli menandakan bahwa proses dekripsi mampu mengubah kembali *cipher image* menjadi *plain image* tanpa ada intensitas warna yang berubah pada citra.

SIMPULAN DAN SARAN

Berdasarkan hasil uji coba kriptografi citra menggunakan algoritma *logistic map* pada penelitian ini dapat ditarik beberapa kesimpulan. *Pseudo-random number generator* dengan menggunakan *seed* tertentu dapat menghasilkan deretan bilangan acak yang sama setiap kali dieksekusi. Proses enkripsi yang dilakukan mampu menghasilkan *cipher image* yang berbeda dengan citra asli serta memiliki histogram berbeda juga. Proses

dekripsi telah dapat berfungsi dengan benar karena dapat menghasilkan kembali citra dengan histogram yang sama dengan *plain image*.

Pada penelitian selanjutnya dapat dikembangkan dengan menentukan variabel enkripsi yang lebih bervariasi menggunakan metode lain sangat disarankan pada penelitian lebih lanjut agar *cipher image* yang dihasilkan dapat dibandingkan dengan hasil yang didapat pada penelitian ini. Proses difusi dan konfusi yang lebih mendalam dengan menggunakan metode atau algoritma lain sangat disarankan untuk peneliti lain yang tertarik untuk melakukan penelitian mengenai topik yang dibahas, agar dapat menjadi referensi atau perbandingan dalam pembuatan algoritma kriptografi yang lebih aman.

DAFTAR PUSTAKA

[1] N. P. Smart, *Cryptography Made Simple*, New York: Springer International Publishing, 2016.

[2] H. Delfs dan H. Knebl , *Introduction to Cryptography (Principles and Applications)*, Berlin: Springer-Verlag, 2015

[3] A.G. Radwan, " On some generalized discrete logistic maps," *Journal of Advanced Research*, vol.4, no. 2, hal. 163–171, 2013.

[4] M. Berezowski dan M. Lawnik, "Identification of fast-changing signals by means of adaptive chaotic transformations," *Nonlinear Analysis: Modelling and Control*, vol. 19, no. 2, hal. 172–177, 2014.

[5] R. Munir, Diktat Kuliah IF5054 "Kriptografi," Departemen Teknik Informatika Institut Teknologi Bandung, 2006

[6] R. Sujarani, dan D. Manivannan, (). A Secure Image Cryptosystem Using 2D Arnold Cat Map and Logistic Map. International Journal of Pharmacy & Technology, vol. 8, no. 4, hal. 25173-25182, 2016.

[7] A. Bhagat, A. Surve, S. Kalgutkar, dan A. Waghmare, " Chaos Based Image Encryption and Decryption," *International Research Journal of Engineering and Technology (IRJET)*, vol. 3, hal. 1440–1445, 2016.

[8] J. Yu, Y. Li, X. Xie, dan N. Zhou, " Image encryption algorithm by using the logistic map and discrete fractional angular transform," *Optica Applicata*, vol. XLVII, no. 1, hal. 141–155, 2017.

[9] L. Liu, dan S. Miao, " A new image encryption algorithm based on logistic chaotic map with varying parameter," *SpringerPlus*, vol. 5, no. 289, hal. 1–12, 2016.

[10] G. Zhang, W. Ding, dan L. Li, " Image Encryption Algorithm Based on Tent Delay-Sine Cascade with Logistic Map," *Symmetry*, vol. 12, no. 3, hal. 1 – 14, 2020.