

# PENERAPAN ALGORITMA *BERNOULLI MAP* DALAM PROGRAM APLIKASI ENKRIPSI CITRA DIGITAL

*Nadya Sofia Laura*<sup>1</sup>,  
*Edi Sukirman*<sup>2</sup>,  
*Suryadi M.T*<sup>3</sup>,

<sup>1,2</sup>Fakultas Ilmu Komputer, Universitas Gunadarma,

<sup>3</sup>Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Indonesia

<sup>1</sup>[nadyasofialaura@student.gunadarma.ac.id](mailto:nadyasofialaura@student.gunadarma.ac.id)

<sup>2</sup>[ediskm@staff.gunadarma.ac.id](mailto:ediskm@staff.gunadarma.ac.id)

<sup>3</sup>[yadi.mt@sci.ui.ac.id](mailto:yadi.mt@sci.ui.ac.id)

## Abstrak

Pesatnya perkembangan teknologi menuntut sekuritas (keamanan) terhadap kerahasiaan data atau informasi. Banyak pengguna (user) yang tidak ingin data atau informasi yang disampaikan dicuri oleh orang lain. Pengiriman dan penyimpanan data atau informasi tanpa dilakukan pengamanan akan beresiko terhadap penyadapan dan pencurian. Dengan demikian data atau informasi yang ada di dalamnya dapat mudah diketahui oleh pihak-pihak yang tidak berhak. Oleh karena itu, dikembangkanlah cabang ilmu yang mempelajari tentang cara-cara pengamanan data atau informasi yang dikenal dengan istilah kriptografi dengan cara melakukan enkripsi data. Penelitian terus dilakukan untuk meningkatkan daya tahan algoritma yang digunakan pada proses enkripsi dari serangan brute force, salah satunya dengan cara mengimplementasikan teori chaos. Salah satu algoritma yang mengimplementasikan teori chaos adalah algoritma Bernoulli Map. Algoritma ini mengimplementasikan teori chaos dengan membangkitkan deret bilangan yang bersifat acak. Hasil penelitian menunjukkan bahwa algoritma ini dapat mengenkripsi sejumlah citra grayscale maupun berwarna yang berekstensi bmp dan png. Waktu proses enkripsi dan dekripsi berbanding lurus dengan besarnya dimensi citra. Histogram citra terenkripsi memiliki distribusi seragam sehingga tahan terhadap penyerang yang akan melakukan analisis frekuensi. Algoritma ini memang memberikan keamanan yang baik dari serangan brute force.

Kata kunci: Algoritma Enkripsi, Chaos, Citra Digital, Citra BMP, Citra PNG, Algoritma Bernoulli Map.

## THE APPLICATION OF *BERNOULLI MAP* ALGORITHM IN THE ENCRPTION APPLICATION OF DIGITAL IMAGE

### Abstract

The development of technology requires securities of the data. Many users protect heir data or information from being stolen by others. The delivery and storage of data or information without any security clampdown will be result in the risk of eavesdropping and theft. The data will be easily identified by parties who are not eligible. Therefore, it is necessary to develop the branch of science that studies the securing data or information that is known as cryptography, i.e. encrypting the data. Ongoing research to improve the durability of the algorithm used in the encryption process of a brute force attack, such as by implementing the chaos theory. One algorithm that implements the algorithm of chaos theory was Bernoulli Map. This algorithm implements chaos theory to generate series of the random numbers. The results showed that this algorithm can encrypt any number of grayscale images with extension bmp and png. Time encryption and decryption process is directly proportional to the magnitude of the dimensions of the image. Encrypted image histogram has a

*uniform distribution so that the resistance against the aggressor will perform frequency analysis. This algorithm does provide good security from brute-force attacks.*

*Keywords : Encryption Algorithm, Chaos, Digital Image, Image BMP, PNG image, the algorithm Bernoulli Map.*

## PENDAHULUAN

Pesatnya perkembangan teknologi menuntut sekuritas (keamanan) terhadap kerahasiaan data atau informasi yang dikirimkan sehingga menjadi hal yang sangat penting untuk diperhatikan. Banyak pengguna (*user*) seperti depar-temen pertahanan, suatu perusahaan, atau bahkan individu-individu yang tidak ingindata atau informasi yang disampaikan dicuri oleh orang lain. Pengiriman dan penyimpanan data atau informasi tanpa dilakukan pengamanan akan beresiko terhadap penyadapan dan pencurian. Dengan demikian data atau informasi yang ada di dalamnya dapat mudah diketahui oleh pihak-pihak yang tidak berhak. Oleh karena itu, dikembangkanlah cabang ilmu yang mempelajari tentang cara-cara pengamanan data atau informasi yang dikenal dengan istilah kriptografi.

Kriptografi memiliki dua konsep utama yaitu enkripsi dan dekripsi. Enkripsi adalah proses dimana informasi atau data yang hendak dikirim diubah menjadi bentuk yang hampir tidak dikenali sebagai informasi awalnya dengan menggunakan algoritma tertentu. Enkripsi dilakukan dengan cara mengacak suatu data atau informasi menggunakan kunci rahasia sehingga data atau informasi tersebut menjadi tidak berarti karena tidak dapat diketahui. Dekripsi adalah kebalikan dari enkripsi yaitu mengubah data atau informasi dalam suatu bahasa sandi menjadi data atau informasi awal kembali.

Proses enkripsi diimplementasi-

kan dalam berbagai bentuk data atau informasi. Data dapat berupa angka, karakter, simbol, citra digital, suara atau tanda-tanda yang dapat digunakan untuk dijadikan informasi. Informasi dapat berupa hasil gabungan, hasil analisa, hasil penyimpulan dan juga hasil pengolahan sistem informasi komputerisasi. Salah satu bentuk data adalah citra digital. Citra merupakan istilah lain dari gambar, yang merupakan data atau informasi yang berbentuk visual. Suatu citra diperoleh dari penangkapan kekuatan sinar yang dipantulkan oleh objek. Citra Digital adalah representasi dari sebuah citra dua dimensi sebagai sebuah kumpulan nilai digital yang disebut elemen gambar atau piksel. Piksel adalah elemen terkecil yang menyusun citra dan mengandung nilai yang mewakili kecerahan dari sebuah warna pada sebuah titik tertentu. Pada penelitian terdahulu, telah diusulkan aplikasi enkripsi dengan menggunakan algoritma *Gingerbreadman Map* [1].

Kriptografi pada citra digital dilakukan dengan cara mengubah warna-warna pada setiap piksel. Perubahan warna yang terjadi pada setiap piksel citra digital membuat data atau informasi yang terkandung pada citra digital tidak dapat diketahui. Banyak algoritma yang dapat digunakan untuk melakukan enkripsi, salah satunya dengan mengimplementasikan teori chaos. Teori chaos merupakan cabang dari matematika yang mempelajari cara membangkitkan suatu bilangan acak yang berguna untuk proses enkripsi.

Salah satu algoritma yang mengimplementasikan teori chaos adalah algoritma Bernoulli map. Algoritma

Bernoulli map mengenkripsi citra digital dengan cara membangkitkan bilangan acak yang bertujuan mempersulit orang yang tidak berhak untuk mengetahui isi data atau informasinya.

Sebelumnya telah dilakukan percobaan untuk mengamankan data berupa teks dengan menggunakan algoritma RC-5 [2]. Pada tahap enkripsi, data

yang akan dienkripsi dikembangkan menjadi dua bagian kiri dan bagian kanan. Dilakukan penjumlahan dengan *keyword* yang telah diekspresi sebelumnya. Kemudian dilakukan operasi enkripsi dan tahap terakhir dilakukan penggabungan untuk mendapatkan data yang telah dienkripsi.

Penelitian lain dilakukan dengan menggunakan algoritma Arnold Cat map dan Logistic map. Pada tahap pertama dalam proses enkripsi adalah dengan melakukan permutasi piksel-piksel di dalam citra. Kemudian ekstrak empat bit setiap piksel dari citra. Iterasikan Logistic map untuk memperoleh nilai *keystream*. Selanjutnya menggantikan empat bit dari setiap piksel yang dienkripsi [3].

Teknik enkripsi dilakukan dengan beberapa langkah. Langkah pertama adalah memasukkan kunci dan citra asli, langkah kedua, melakukan 200 kali iterasi sehingga menghasilkan nilai decimal dari  $X_{*++}$ . Langkah ketiga melakukan pengecekan kondisi. Jika ya, maka lakukan tiga kali iterasi. Jika tidak, maka lakukan dari proses awal kembali untuk mendapatkan citra yang terenkripsi. Langkah keempat adalah melakukan pengecekan dari proses iterasi. Setelah itu lakukan operasi XOR pada setiap integer bit [4].

Bernoulli map merupakan salah satu fungsi untuk membuat bilangan acak yang digunakan dalam aplikasi kriptografi [5]. Berdasarkan pertimbangan di atas, digunakan metode yang

dapat digunakan untuk keamanan data dengan implementasi proses enkripsi dan dekripsi citra digital menggunakan metode algoritma *Bernoulli map* dalam pembuatan aplikasi enkripsi dan dekripsi citra digital.

## METODE PENELITIAN

### Analisis Masalah

Pada penelitian ini dilakukan pembuatan aplikasi untuk mengamankan data dengan mengimplementasikan algoritma *Bernoulli Map* pada proses enkripsi dan dekripsi citra digital berwarna bertipe *bitmap* (.bmp) dan *portable network graphics* (.png).

Analisis masalah aplikasi dibuat dalam perancangan aplikasi. Pertama kali aplikasi dijalankan, maka akan muncul halaman menu utama aplikasi. Pada menu utama terdiri dari 7 (tujuh) menu yaitu menu enkripsi, menu dekripsi, menu histogram, menu perbandingan citra (*compare*), menu bantuan dan menu tentang.

Pada menu enkripsi, pengguna diminta untuk memasukkan nilai  $X_n$  dan  $r$  sebagai kunci enkripsi. Menu ini digunakan untuk mengubah gambar asli menjadi gambar acak yang tidak dapat terlihat lagi. Citra yang dapat dienkripsi hanya citra yang memiliki ekstensi .bmp atau .png. Citra dapat dipilih pada folder kerja matlab dengan menekan tombol pilih gambar. Setelah citra dipilih dan kunci ditentukan, langkah selanjutnya adalah pengguna menekan tombol enkripsi supaya citra tersebut diproses. Waktu proses enkripsi akan tampil pada bagian bawah nilai  $X_n$  dan  $r$  dalam satuan detik. Citra yang sudah berhasil dienkripsi dapat disimpan dengan menggunakan tombol simpan gambar.

Pada menu dekripsi, pengguna diminta untuk memasukkan nilai  $X_n$  dan  $r$  kembali. Nilai  $X_n$  dan  $r$  yang dimasukkan dalam menu ini harus sama dengan nilai  $X_n$  dan yang dimasukkan dalam menu enkripsi.

Menu ini digunakan untuk mengubah gambar acak hasil enkripsi menjadi gambar asli seperti sebelum dienkripsi. Sama seperti ada menu enkripsi, pada menu dekripsi citra yang dapat diproses hanya citra dengan ekstensi .bmp atau .png dapat dipilih pada folder kerja matlab dengan menekan tombol pilih gambar. Setelah citra dipilih dan kunci ditentukan, langkah selanjutnya adalah pengguna menekan tombol dekripsi supaya citra tersebut diproses. Waktu proses dekripsi akan tampil pada bagian bawah nilai  $X_n$  dan  $r$  dalam satuan detik. Citra yang sudah berhasil didekripsi dapat disimpan dengan menggunakan tombol simpan gambar.

Pada menu histogram, akan masuk ke dalam submenu histogram terlebih dahulu. Submenu histogram terdiri dari 3 (tiga) menu yaitu menu citra *grayscale*, menu citra berwarna atau RGB dan menu kembali. Menu histogram citra *greyscale* dapat dilakukan dengan cara menekan tombol pilih gambar dimana citra yang dipilih adalah citra yang akan diketahui gambar histogramnya. Setelah citra sudah dipilih, maka tekan tombol cek histogram untuk menampilkan gambar dari histogram citra yang telah dipilih tadi. Penggunaan pada menu histogram citra berwarna atau RGB sama seperti penggunaan pada menu histogram citra *greyscale*. Perbedaannya adalah pada menu histogram citra berwarna atau RGB akan menampilkan 3 (tiga) histogram dari tiap-

tiap lapisan warna yaitu lapisan warna merah (*red*), lapisan warna hijau (*green*) dan lapisan warna biru (*blue*).

Pada menu perbandingan citra (*compare*), akan masuk ke dalam submenu perbandingan citra terlebih dahulu. Submenu perbandingan citra terdiri dari 3 (tiga) menu yaitu menu citra *grayscale*, menu citra berwarna atau RGB dan menu kembali. Perbandingan citra *greyscale* dilakukan dengan cara menekan tombol pilih gambar yang berada dibawah tulisan citra asli untuk memasukkan gambar dari citra asli, lalu menekan tombol pilih gambar yang berada dibawah tulisan citra enkripsi untuk memasukkan citra acak yang tidak dapat terlihat dengan baik dan menekan tombol pilih gambar yang berada dibawah tulisan citra dekripsi untuk memasukkan gambar dari citra acak yang telah berhasil dikembalikan ke citra semula. Setelah semua citra telah dipilih, tekan tombol bandingkan citra untuk melakukan proses perbandingan citra. Setelah itu akan muncul histogram dari tiap-tiap citra beserta informasi lainnya yang terdiri dari nama *file*, ukuran *file*, lebar citra (*width*) dalam piksel, tinggi citra (*height*) dalam piksel, format *file*, kedalaman citra dalam bit, dan tipe warna. Pada menu perbandingan citra berwarna atau RGB sama seperti penggunaan pada menu perbandingan citra *greyscale*. Perbedaannya adalah pada menu perbandingan citra berwarna atau RGB masing-masing citra akan menampilkan 3 (tiga) histogram dari tiap-tiap lapisan warna yaitu lapisan warna merah (*red*), lapisan warna hijau (*green*) dan lapisan warna biru (*blue*).

Pada menu *Peak Signal to Noise Ratio* (PSNR) juga sama seperti menu histogram dan menu perbandingan citra. Menu ini akan masuk dahulu ke dalam submenu *Peak Signal to Noise Ratio* (PSNR).

Submenu *Peak Signal to Noise Ratio* (PSNR) terdiri dari 3 (tiga) menu yaitu menu citra *grayscale*, menu citra berwarna atau RGB dan menu kembali. Menu *Peak Signal to Noise Ratio* (PSNR) citra *grayscale* dapat dilakukan dengan cara menekan tombol pilih gambar 1 dimana citra yang dipilih adalah citra asli dan menekan tombol pilih gambar 2 untuk memasukkan citra acak yang telah berhasil dikembalikan ke citra asli. Setelah citra sudah dipilih, maka tekan tombol bandingkan citra untuk menampilkan gambar dengan perhitungan *Mean Square Error* (MSE) dari citra yang telah dipilih beserta informasi lainnya yang terdiri dari nama *file*, ukuran *file*, lebar citra (*width*) dalam piksel, tinggi citra (*height*) dalam piksel, format *file*, kedalaman citra dalam bit, tipe warna, *Mean Square Error* (MSE), *Peak Signal to Noise Ratio* (PSNR) dan Tingkat kesamaan citra. Penggunaan pada menu *Peak Signal to Noise Ratio* (PSNR) citra berwarna atau RGB sama seperti penggunaan pada menu *Peak Signal to Noise Ratio* (PSNR) citra *grayscale*.

Pada menu bantuan digunakan untuk menampilkan bantuan dalam penggunaan aplikasi, sedangkan menu tentang berisi judul aplikasi dan biodata pembuat aplikasi.

### Tahap Pembangkitan Keystream

Pada tahap awal ini, pembangkitan *keystream* dilakukan dengan menggunakan algoritma Bernoulli Map seperti pada persamaan (1) kunci yang diminta akan digunakan sebagai parameter. Kunci nya yaitu nilai  $X_n$  dan  $r$ .

Bernoulli map merupakan salah satu fungsi untuk membuat bilangan acak yang digunakan dalam aplikasi

kriptografi . Fungsinya dinyatakan sebagai : [5]

$$X_{n+1} = r \times X_n \text{ mod } 1 \quad (1)$$

dengan :

1.  $X_n$  mengambil nilai dari rentang  $0, 1, r \in (0, 1)$ .
2. Variabel  $r$  mengambil nilai dari 0 sampai  $\infty, \in (0, \infty)$ .
3. Nilai awal  $X_n = 0, 1$ .
4. Iterasi pengulangan = 8000 untuk  $r$  bertambah 0.001.

Algoritma ini membangkitkan deret bilangan real, agar deret bilangan ini dapat digunakan sebagai *keystream*, maka bilangan-bilangan tersebut harus dikonversikan menjadi bilangan integer dengan rentang 0 sampai 255.

Proses tersebut dilakukan dengan cara mengabsolutkan barisan bilangan yang dihasilkan dari persamaan (1), lalu masing-masing dari bilangan tersebut dikalikan dengan 10000 dan dibulatkan kebawah (*floor*) untuk menghasilkan bilangan *integer*. Secara matematis fungsi konversi *integer* tersebut dituliskan pada persamaan (2) sebagai berikut :

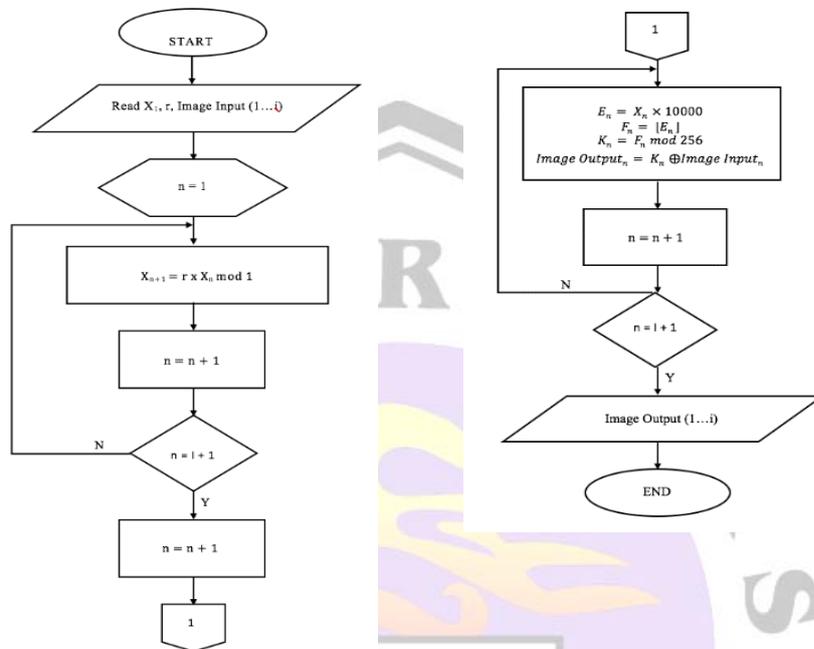
$$E_n = X_n \times 10000 \quad (2)$$

$$F_n = \lfloor E_n \rfloor \quad (3)$$

Tahap selanjutnya adalah melakukan pembulatan kebawah, sehingga menghasilkan bilangan *integer* seperti yang dapat dilihat pada persamaan (3). Setelah didapatkan barisan bilangan integer, deret tersebut dipetakan pada rentang antara 0 sampai 255. Secara matematis fungsi pemetaan dituliskan pada persamaan (4) sebagai berikut :

$$K_n = F_n \text{ mod } 256 \quad (4)$$

## Flowchart Proses Enkripsi dan Dekripsi



Gambar 1. Flowchart Proses Enkripsi dan Dekripsi

Pada Gambar 1 ditunjukkan proses pada tahap enkripsi dan deskripsi. Tahap enkripsi merupakan tahap citra semula atau *plain image* ( $P_n$ ) diubah menjadi citra terenkripsi atau *chiper image* ( $C_n$ ) dengan cara meng-XOR kan *pixel-pixel plain image* ( $P_n$ ) terhadap *keystream* ( $K_n$ ) yang telah dibangkitkan. Secara matematis fungsi enkripsi ditunjukkan oleh persamaan (5) sebagai berikut:

$$C_n = P_n \oplus K_n \quad (5)$$

Keterangan :

- : *Cipher image* (citra terenkripsi)
- : *Plain image* (citra semula)
- : *Keystream*

Tahap dekripsi memiliki proses yang sama dengan tahap enkripsi, hanya saja citra masukannya merupakan citra hasil enkripsi atau *chiper image* ( $C_n$ ). Untuk mendekripsikan kembali citra semula dari hasil *chiper image*

$$P_n = C_n \oplus K_n$$

$$= (P_n \oplus K_n) \oplus K_n$$

$$= P_n \oplus (K_n \oplus K_n)$$

$$= P_n \oplus 0$$

$$= P_n$$

( $C_n$ ), maka dilakukan operasi XOR dari *pixel-pixel chiper image* ( $C_n$ ) terhadap

keystream ( $K_n$ ) yang telah dibangkitkan. Secara matematis fungsi enkripsi ini dapat dituliskan sebagai berikut:

Keterangan :

## HASIL DAN PEMBAHASAN

Implementasi aplikasi dilakukan pada komputer denganspesifikasiperangkat keras: processor Intel® Core™i5- 4278U CPU @ 2.6GHz, kapasitas RAM 8 GB, system type 64-bit OS.Hal utama yang perlu dilakukan untuk melakukan implementasi aplikasi adalah dengan menginstall Bahasa pemrograman Matlab pada komputer.

Pada bagian ini akan diuraikan hasil uji coba dari proses enkripsi dan dekripsi citraantaralainuntukmelihatkesamaancitra semula dengan citra hasil dekripsi, menghitung waktu proses enkripsi dan dekripsi, melihat pengaruh komposisi dan keragaman citra terhadap waktu proses enkripsi dan dekripsi, melihat sensitivitas kunci, dan melihat ruang kunci dan waktu yang dibutuhkan untuk memecahkan kunci dari serangan *brute force*.

Pada tahap uji coba, proses enkripsi dan dekripsi menggunakan sejumlah citra. Data citra yang digunakan dalam penelitian ini dapat

Tabel 1.Data Citra Yang Digunakan Pada Uji Coba

Data Uji ke-	Tampilan Citra	Nama Citra	Ukuran Citra (pixel)	Ukuran File (byte)	Jenis Citra
1		abu 1.bmp	320 × 240	93 KB	Grayscale
2		abu 2.bmp	400 × 300	124 KB	Grayscale
3		abu 3.bmp	800 × 600	209 KB	Grayscale
4		abu 4.bmp	1024 × 768	276 KB	Grayscale
5		abu 5.bmp	1440 × 1080	427 KB	Grayscale
6		matahari 1.bmp	180 × 240	217 KB	Truecolor
7		matahari 2.bmp	360 × 480	789 KB	Truecolor
8		matahari 3.bmp	450 × 600	1.2 MB	Truecolor
9		matahari 4.bmp	576 × 768	1.9 MB	Truecolor
10		matahari 5.bmp	810 × 1080	3.6 MB	Truecolor
11		gs 1.png	240 × 240	95 KB	Grayscale
12		gs 2.png	480 × 480	170 KB	Grayscale
13		gs 3.png	600 × 600	218 KB	Grayscale
14		gs 4.png	768 × 768	293 KB	Grayscale
15		gs 5.png	1080 × 1080	459 KB	Grayscale
16		sun 1.png	247 × 240	196 KB	Truecolor
17		sun 2.png	493 × 480	372 KB	Truecolor
18		sun 3.png	616 × 600	539 KB	Truecolor
19		sun 4.png	789 × 768	763 KB	Truecolor
20		sun 5.png	1109 × 1080	1.2 MB	Truecolor

$G$  : Cipher image (citra terenkripsi)

$P_n$  : Plain image (citra semula)

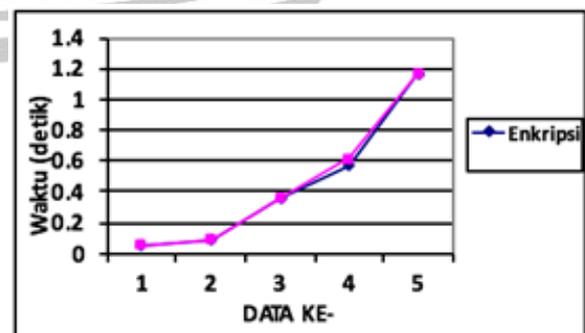
$K_n$  : Keystream

dilihat pada Tabel 4.1.

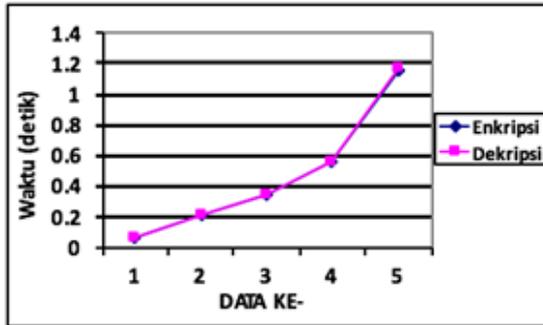
Setelah pengujian dilakukan pada sejumlah citra, baik citra *gray-scale* maupun citra berwarna (RGB) dengan format .bmp dan.png,semuacitradapatdienkripsidengan baik menjadi citra hasil enkripsi (*chiper image*).Citra tersebut dapat didekripsi kembali menjadi citra semula (*plain image*) dengan memasukkan kunci yang tepat.

## Analisis Waktu Enkripsi dan Dekripsi

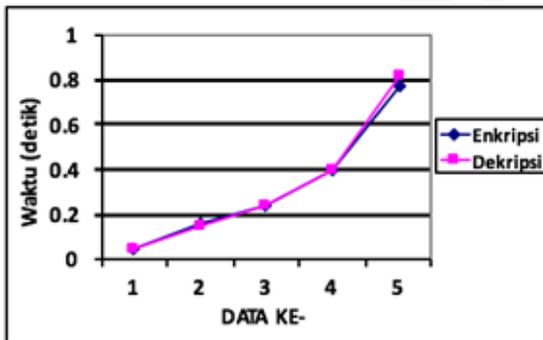
Berikut adalah tabel perbandingan antara waktu proses rata-rata enkripsi dan dekripsi untuk tiap data uji citra dengan nilai kunci yang sama untuk setiap citra, nilai  $X_n = 0.1$  dan  $r = 8,1$ . Pengambilan nilai waktu proses rata-rata dilakukan setelah melakukan lima kali uji coba untuk masing-masingcitra. Berikut adalah data hasil uji coba pertama untuk citra *grayscale* dengan format .bmp beserta tampilan grafik dari data hasil uji coba tersebut.



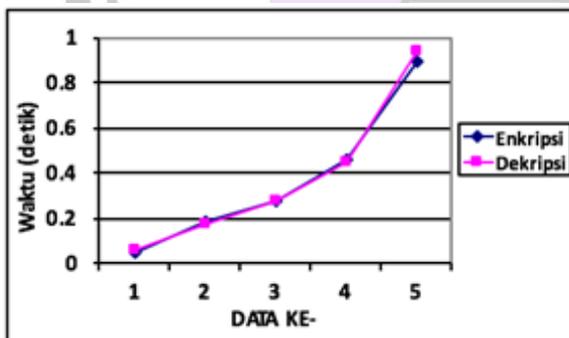
Gambar 2.Grafik Rata-Rata Waktu Enkripsi dan Dekripsi Data Uji Citra Grayscale Format BMP



Gambar 3. Grafik Rata-Rata Waktu Enkripsi dan Dekripsi Data Uji Citra Berwarna (RGB) Format BMP



Gambar 4. Grafik Rata-rata Waktu Enkripsi dan Dekripsi Data Uji Citra Grayscale Format PNG



Gambar 5. Grafik Rata-rata Waktu Enkripsi dan Dekripsi Data Uji Citra Berwarna (RGB) Format PNG

Pada Gambar 2, Gambar 3, Gambar 4 dan Gambar 5 ditunjukkan rata-rata waktu enkripsi dan deskripsi untuk masing masing citra grayscale dan citra berwarna, dalam format .bmp dan .png. Untuk uji coba pada citra berwarna (RGB) dengan format .png, memiliki hasil yang sama seperti 3 (tiga) percobaan yang sudah dilakukan sebelumnya, yaitu waktu proses

meningkat seiring dengan bertambahnya ukuran citra (*pixel*). Kesimpulan dari analisis waktu yaitu bahwa semakin besar ukuran citra (*pixel*) pada suatu citra, baik itu citra *grayscale* maupun citra berwarna (RGB) dalam berbagai format (.bmp dan .png), waktu yang dibutuhkan untuk melakukan proses enkripsi dan dekripsi semakin lama. Jadi sudah terbukti bahwa perbedaan ukuran citra sangat berpengaruh dalam waktu proses.

### Analisis Sensitivitas Kunci

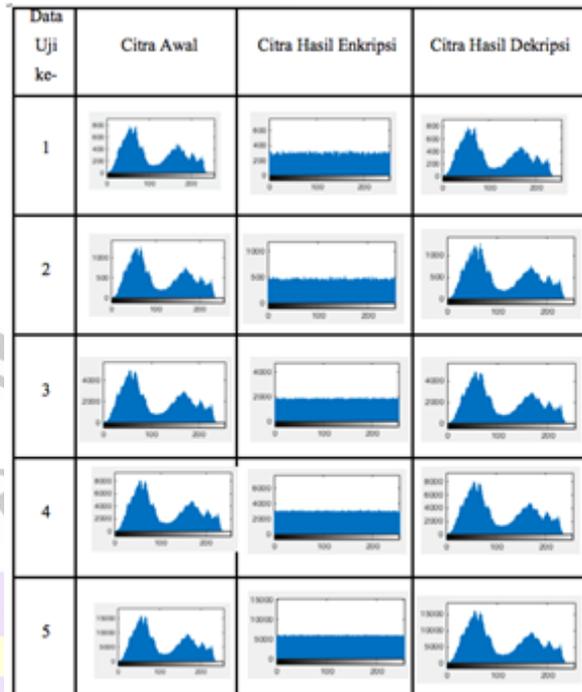
Pengujian dilakukan dengan membandingkan citra hasil dekripsi pada sejumlah citra uji coba yang telah dienkripsi dengan nilai kunci  $X_n$  sebesar 0,1 dan  $r$  sebesar 8,1 dengan perubahan yang sangat kecil dari nilai pada salah satu kunci. Hasil pengujian dapat dilihat pada Tabel 2 berikut:

Tabel 2. Data Hasil Pengujian Sensitivitas

Nama File	Citra Hasil	Citra Hasil	Citra Hasil
	Enkripsi dengan Kunci $X_n = 0,1$ dan Kunci $r = 8,1$	Dekripsi dengan Kunci $X_n = 0,1$ dan Kunci $r = 8,1 + 10^{-15}$	Dekripsi dengan Kunci $X_n = 0,1$ dan Kunci $r = 8,1 + 10^{-16}$
abu 1.bmp			
matahari 1.bmp			
gs 1.png			
sun 1.png			

Nama File	Citra Hasil Enkripsi dengan Kunci $X_n = 0,1$ dan Kunci $r = 8,1$	Citra Hasil Dekripsi dengan Kunci $X_n = 0,1+10^{-16}$ dan Kunci $r = 8,1 + 10^{-12}$	Citra Hasil Dekripsi dengan Kunci $X_n = 0,1+10^{-17}$ dan Kunci $r = 8,1 + 10^{-16}$
abu 1.bmp			
matahari 1.bmp			
gs 1.png			

Tabel 3. Hasil Histogram Citra Uji Coba



### Algoritma

BernoulliMapmemilikinilai sensitivitas kunci yang cukup kecil yakni sampai  $10^{&#x207E}$  pada kunci  $r$  dan  $10^{&#x207E}$  pada kunci  $X_n$ . Sensitivitas kunci tersebut dapat diketahui dengan dilakukan sebuah percobaan dengan mengubah nilai kunci dari algoritma Bernoulli Map dengan menambahkan nilai sebesar  $10^{&#x207E}$  pada kunci  $r$  dan  $10^{&#x207E}$  pada kunci  $X_n$  dan ternyata menghasilkan citra yang sama seperti citra hasil enkripsi. Sedangkan ketika nilai kunciditambahkan menjadi  $10^{&#x207E}$  dan kunci  $X_n$ ditambahkan menjadi $10^{&#x207E}$ , citra tersebut berhasil didekripsi. Hal ini membuktikan bahwa algoritma Bernoulli Map cukup baik untuk menghentikan serangan *brute force* dikarenakan perubahan satu bit pada kunci akan menyebabkan sebuah hasil yang berbeda.

### Analisis Kesamaan Citra

Untuk mengetahui kesamaan citra, maka perlu dilakukan pengujian dengan membandingkan histogram dari tiap citra serta membandingkan perhitungan selisih nilai *Mean Square Error* (MSE) dan *Peak Signal Noise Ratio* (PSNR).



Data Uji ke-	Citra Awal	Citra Hasil Enkripsi	Citra Hasil Dekripsi
9			
10			
11			
12			
13			

Data Uji ke-	Citra Awal	Citra Hasil Enkripsi	Citra Hasil Dekripsi
18			
19			
20			

Data Uji ke-	Citra Awal	Citra Hasil Enkripsi	Citra Hasil Dekripsi
14			
15			
16			
17			

Pada Tabel 3 ditunjukkan histogram dari masing-masing citra dalam format .bmp dan .png, dengan tipe warna *grayscale* maupun berwarna (RGB). Dari tabel tersebut terbukti bahwa hasil dekripsi citra sama dengan citra asli yaitu citra sebelum dilakukan proses enkripsi. Pada histogram citra hasil enkripsi memiliki persebaran piksel yang seragam atau distribusi *uniform*. Histogram citra hasil enkripsi relatif datar sehingga tahan terhadap penyerang yang akan melakukan analisis frekuensi. Oleh karena itu, histogram citra awal dan histogram citra hasil enkripsi seharusnya tidak memiliki kesamaan secara statistik. Distribusi *uniform* pada citra hasil enkripsi merupakan sebuah indikasi bahwa algoritma enkripsi citra memiliki kualitas baik.

## KESIMPULAN DAN SARAN

Program aplikasi enkripsi citra digital memiliki fasilitas menu enkripsi,

dekripsi, histogram, *compare*, *Peak Signal to Noise Ratio* (PSNR), bantuan dan tentang aplikasi. Aplikasi ini dapat melakukan enkripsi dan dekripsi untuk citra *grayscale* maupun RGB (berwarna) dengan format .bmp (bitmap) dan .png (portable network graphics) saja yang dapat di proses dengan baik.

Tidak ada perubahan ukuran file dan dimensi dari citra berwarna pada citra semulad an citra hasil dekripsi. Proses enkripsi pun berjalan dengan relatif cepat, hanya membutuhkan waktu beberapa detik saja. Lamanya waktu proses enkripsi dan dekripsi berbanding lurus dengan besarnya ukuran piksel citra. Semakin besar ukuran piksel suatu citra maka semakin lama waktu proses enkripsi dan dekripsi. Waktu proses enkripsi dan dekripsi citra digital mempunyai waktu yang relatif sama. Citra semula atau citra sebelum enkripsi dan citra hasil dekripsi tidak memiliki perbedaan. Hal ini dibuktikan dengan tingkat kesamaan citra sebesar 100%, hasil perhitungan *Mean Square Error* (MSE) sebesar 0 dB dan *Peak Signal to Noise Ratio* (PSNR) memiliki nilai *Infinity* (Inf) dB. Algoritma Bernoulli Map dapat memberikan keamanan yang baik dari serangan *brute force* yang memiliki tingkat sensitivitas kunci  $X^n$  sebesar  $10^k$  (dan kunci  $r$  sebesar  $10^k$ ). Pada histogram citra hasil enkripsi memiliki distribusi *uniform* atau seragam persebaran pikselnya. Hal ini akan membuat pihak ketiga semakin kesulitan untuk dapat membuka file citra hasil enkripsi tersebut.

Aplikasi Enkripsi dan Dekripsi Citra Digital dengan menerapkan algoritma Bernoulli Map masih dapat dikembangkan lagi, salah satu contohnya yaitu pengembangan untuk menambahkan format citra digital yang

umum digunakan seperti .jpeg dan .gif serta format citra digital lainnya. Saat ini aplikasi hanya digunakan untuk citra digital format .bmp (bitmap) dan .png (*portable network graphics*) saja. Selain itu diharapkan pada aplikasi enkripsi dan dekripsi citra digital ini dapat dikembangkan menjadi aplikasi yang dapat diterapkan dalam perangkat *mobile*.

## DAFTAR PUSTAKA

- [1] Suryadi MT dan Tony Gunawan. 2014. "Aplikasi Enkripsi Citra Digital Menggunakan Algoritma Gingerbreadman Map". KOMMIT. ISSN : 2302-3740, pp. 370 – 375.
- [2] Kurniawan, Mike Yulianadan M.Zen Samsono Hadi. 2012. "Analisa Dan Implementasi Sistem Keamanan Data Dengan Menggunakan Metode Enkripsi Algoritma RC-5". Proyek Akhir PENS-ITS.
- [3] Rinaldi Munir. 2012. "Algoritma Enkripsi Citra Digital dengan Kombinasi Dua Chaos Map dan Penerapan Teknik Selektif terhadap Bit-Bit MSB". Seminar Nasional Aplikasi Teknologi Informasi. ISSN: 1907-5022.
- [4] Suryadi MT, Eva Nurpeti, Dhian Widya. 2014. "Performance of Chaos- Based Encryption Algorithm for Digital Image". TELEKOMNIKA Vol 12 No 3.
- [5] Ahmed, Hossam Eldin H. dan Ayma n H. Abd El-aziem. 2014. "Image Encryption Using Development of Chaotic Logistic Map Based on Feedback Stream Cipher". Recent Advances In Telecommunications, Informatics And Educational Technologies. ISBN: 978-1-61804-262-0, pp. 274-283.

