Jurnal ELTIKOM, Vol. 2, No. 2, Desember 2018, hal. 50-57
ISSN 2598-3245 (Print), ISSN 2598-3288 (Online)
Tersedia online di http://eltikom.poliban.ac.id
DOI : http://doi.org/10.31961/eltikom.v2i2.85

PENGGUNAAN ARNOLD CAT MAP DAN BETA CHAOTIC MAP PADA ENKRIPSI DATA CITRA

Weny Mistarika Rahmawati[1)], Febri Liantoni[2)]
[1, 2)]Teknik Informatika, Institut Teknologi Adhi Tama Surabaya
e-mail: wenymistarika@gmail.com[1)], febri.liantoni@ko2pi.org[2)]

ABSTRACT

The use of the image in daily life keeps increasing as information technology is constantly developing. For this reason, we need a method so that image data can be transmitted safely. One of method is by encrypting the image. Encrypted image will only make the image readable by authorized parties. Schemes used in the encryption process can be permutations. In this research using Arnold cat map to permute image encryption. However, permutation alone is not safe enough to encrypt images. Mutated images are then added with other chaos-based algorithms. Beta chaotic map is used in this study because it has more parameters than other types of maps. With larger parameters it will strengthen the results of encryption. The results of the tests carried out in this study indicate that the encryption scheme has resistance to brute force attacks and histogram analysis attacks. The original image is very different from the encrypted image as evidenced by the calculation of the NPCR value.
Keywords*: Arnold cat map, Beta chaotic map, Image encryption.*

ABSTRAK

Penggunaan citra dalam kehidupan sehari-hari mengalami peningkatan seiring berkembangnya teknologi informasi. Untuk itu diperlukan sebuah cara agar data citra dapat ditransmisikan dengan aman. Salah satunya adalah dengan melakukan enkripsi pada citra. Citra terenkripsi akan membuat citra hanya dapat dibaca oleh pihak yang berwenang saja. Skema yang digunakan pada proses enkripsi dapat berupa permutasi. Pada penelitian ini menggunakan Arnold cat map untuk melakukan permutasi pada enkripsi citra. Namun permutasi saja tidak cukup aman untuk mengenkripsi citra. Citra yang telah dipermutasi selanjutnya ditambah dengan algoritma lain berbasis chaos. Beta chaotic map digunakan dalam penelitian ini karena memiliki parameter yang lebih banyak dibandingkan dengan map jenis lain. Dengan parameter yang lebih besar maka akan memperkuat hasil enkripsi. Hasil pengujian yang dilakukan pada penelitian ini menunjukkan bahwa skema enkripsi memiliki ketahanan terhadap serangan brute force dan serangan analisis histogram. Citra asli akan memiliki bentuk yang sangat berbeda dengan citra hasil enkripsi yang dibuktikan dengan perhitungan nilai NPCR.

Kata Kunci*: Arnold cat map, Beta chaotic map, enkripsi citra*.

I. PENDAHULUAN

MENINGKATNYA kebutuhan akan data menyebabkan peningkatan kebutuhan akan keamanan dalam proses transmisinya. Hampir setiap pengiriman data dilakukan secara daring. Sangat jarang ditemui pengiriman yang dilakukan tanpa koneksi internet. Kebutuhan akan banyaknya pengiriman data ini menyebabkan mulai banyak dikembangkannya penelitian tentang bagaimana cara agar pengguna dapat mengirimkan data dengan aman, yaitu dengan melakukan enkripsi pada data yang akan ditransmisikan. Tidak dapat dipungkiri bahwa banyak pihak yang bisa dengan mudah mendapatkan data yang kita transmisikan lewat internet. Namun bukan berarti pihak tersebut dapat dengan leluasa membaca data kita apabila data kita dienkripsi dengan baik.

Data yang digunakan oleh pengguna bermacam-macam tidak hanya data teks, tapi juga dapat berupa data citra. Data citra banyak digunakan oleh berbagai bidang seperti bidang kesehatan, transportasi, militer, dll. Pada beberapa bidang tertentu sangat membutuhkan keamanan yang tinggi untuk pengiriman data. Hal ini dapat dikarenakan data dari institusi ini bersifat rahasia. Misalnya pada bidang kesehatan atau militer. Bidang-bidang ini membutuhkan keamanan yang tinggi dalam transmisi data.

Data citra memiliki karakteristik yang berbeda dengan data teks. Data citra memiliki ukuran yang besar dan memiliki tingkat pengulangan yang tinggi. Oleh sebab itu biasanya peneliti melakukan cara

yang berbeda pada proses enkripsi data citra dan data teks. Pada data teks, banyak algoritma enkripsi yang biasa digunakan yaitu dengan Data Encryption Standart (DES), Advanced Encryption Standart (AES), Secure Hash Algorithm(SHA), dll. Sementara pada data citra biasanya dilakukan pengacakan dengan kunci tertentu. Selain itu pada enkripsi citra kebanyakan peneliti juga menggunakan metode chaos karena karakter citra yang berbeda dengan text sehingga metode *chaos* dianggap sebagai metode yang cocok untuk melakukan enkripsi pada citra [1-2]. Banyak metode chaos yang digunakan pada proses enkripsi citra, mulai dari *Tent map, Gaussian map, Beta map,* dll [3–7].

Pada penelitian ini akan dilakukan enkripsi citra menggunakan dua metode yaitu permutasi dan *masking*. Pada proses permutasi akan menghasilkan citra yang teracak yaitu dengan menggunakan Arnold *cat map*. Hasil permutasi akan diteruskan dengan proses *masking* dengan serangkaian angka acak yang dihasilkan dengan menggunakan *Beta chaotic map. Beta chaotic map* digunakan pada penelitian ini karena kemampuannya untuk menghasilkan nilai acak dengan nilai parameter yang besar. Hasil dari penggabungan permutasi dan substitutsi ini yang nantinya akan menjadi citra terenkripsi.

II. STUDI LITERATUR

Pada bagian ini akan dijelaskan beberapa proses yang diperlukan dalam melakukan penelitian tentang enkripsi citra ini. Pada penelitian ini digunakan *MATrix LABoratory (Matlab)* sebagai *tools* pemrograman dalam penyelesaian enkripsi.

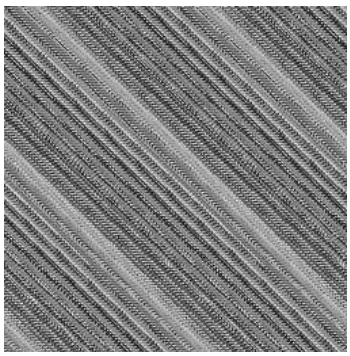
A. Arnold Cat Map

Cat map merupakan fungsi untuk mengubah koordinat *pixel* (x,y) pada citra ke koordinat lain dalam citra tersebut. Persamaan *cat map* untuk mengacak gambar ditunjukkan pada Persamaan (1). *Cat map* yang digunakan dalam penelitian ini adalah persamaan Arnold *cat map* yang telah dikembangkan.

$$\begin{bmatrix} r_{i+1} \\ s_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & g \\ h & gh+1 \end{bmatrix} \begin{bmatrix} r_i \\ s_i \end{bmatrix} mod\ N \quad (1)$$

(r_i, s_i) merupakan koordinat asli dalam citra sedangkan (r_{i+1}, s_{i+1}) merupakan koordinat baru hasil *cat map*. g dan h adalah nilai yang ditentukan yang akan menjadi kunci dari skema enkripsi ini. Untuk menggunakan cat map ini dilakukan iterasi beberapa kali agar hasilnya semakin teracak. Selain nilai g dan h, jumlah iterasi yang digunakan juga akan menjadi kunci dari enkripsi ini.

Hasil dari cat map bersifat *reversible* atau dapat dikembalikan menjadi bentuk asli dengan melakukan iterasi dengan jumlah tertentu. Untuk mengembalikan hasil pengacakan dengan Arnold *cat map* atau bisa disebut *invers* Arnold *cat map* dapat dilakukan dengan menggunakan (2) dengan keterangan sama dengan (1).

(a)  (b) 

Gambar 1 (a)Citra asli (b) citra hasil pengacakan dengan Arnold *cat map* dengan satu kali iterasi

$$\begin{bmatrix} r_i \\ s_i \end{bmatrix} = \begin{bmatrix} 1 & g \\ h & gh+1 \end{bmatrix}^{-1} \begin{bmatrix} r_{i+1} \\ s_{i+1} \end{bmatrix} mod\ N \quad (2)$$

Contoh dari penerapan *cat map* pada citra dapat dilihat pada Gambar 1. Dengan satu kali iterasi akan menghasilkan citra acak yang sulit untuk dikenali. Semakin banyak iterasi yang digunakan, maka akan semakin acak hasilnya.

Walaupun proses permutasi menggunakan Arnold *cat map* telah menghasilkan citra yang acak, namun histogram hasil *cat map* masih sama dengan histogram citra asli. Histogram yang sama ini akan rentan dengan serangan analisa histogram. Penyerang masih akan mampu mengira-ngira citra asli. Sehingga penggunaan permutasi dengan Arnold *cat map* saja dianggap kurang mampu menjaga kerahasiaan dari citra asli.

B. Beta Chaotic Map

Chaotic map merupakan algoritma yang banyak digunakan pada enkripsi karena memiliki sifat yang sulit untuk ditebak walaupun sangat sederhana. *Chaotic map* akan menghasilkan angka pseudorandom yang nantinya akan digunakan untuk proses enkripsi. Hasil *pseudorandom* yang dihasilkan oleh *chaotic map* tergantung dari parameter yang diinputkan. Ada beberapa *chaotic map* yang dapat digunakan diantaranya adalah *Tent map, Circle map, Gaussian map, Beta map* dll. Dari beberapa jenis *chaotic map* tersebut, *Beta map* merupakan jenis *chaotic map* yang memiliki parameter masukan yang besar[5]. Dengan parameter yang besar akan menghasailkan kunci yang besar pula sehingga tingkat keamanan juga lebih besar. Untuk menghasilkan *pseudorandom* dari Beta *chaotic map* dapat menggunakan (3).

$$x_{n+1} = k \times Beta(x_n; x_1, x_2, p, q) \tag{3}$$

Dimana
$p = b_1 + c_1 \times a$
$q = b_2 + c_2 \times a$

b_1, c_1, b_2, c_2 merupakan nilai konstan yang dipilih sebegai kunci. k merupakan parameter yang mengalikan *chaotic map* untuk mengontrol amplitudo Beta *map*, sedangkan a merupakan parameter percabangan. x_n merupakan hasil Beta *chaotic map* pada index ke n.

Fungsi Beta yang digunakan mengacu pada fungsi Beta yang biasa digunakan pada teori statistika dan *probabilistic*[8-9]. Dengan keterangan yang sama dengan (3), fungsi Beta dapat didefinisikan seperti persamaan (4).

$$Beta(x; p, q, x_1, x_2) = \begin{cases} \left(\frac{(x - x_1)}{(x_c - x_1)}\right)^p \left(\frac{(x_2 - x)}{(x_2 - x_c)}\right)^q & x \in]x_1, x_2[\\ 0 & else \end{cases} \tag{4}$$

Beta *map* akan menghasilkan nilai random sejumlah n berupa *sequence*. *Sequence* ini nantinya akan digunakan untuk mengacak *sequence* lain yang memiliki panjang yang sama dengan proses *masking*. Oleh sebab itu, Beta *chaotic map* akan di-*generate* sepanjang input yang akan diacak.

Dari (3) dan (4) dapat dilihat bahwa parameter yang dibutuhkan untuk menggunakan *Beta chaotic map* adalah nilai $a, x_0, x_1, x_2, k, b_1, b_2, c_1, c_2$. Parameter-parameter tersebut akan sangat memengaruhi hasil *sequence* yang dihasilkan.

C. Masking

Masking adalah proses untuk mengubah suatu nilai atau sekumpulan nilai dengan melibatkan nilai atau sekumpulan nilai lain yang panjangnya sama dengan sebuah persamaan. Persamaan yang digunakan pada proses *masking* ditunjukkan dengan (5) yang diadopsi dari [10] yang telah dimodifikasi. Masukan dari masking adalah *sequence* dari hasil dari permutasi dan sequence random hasil dari Beta *chaotic map*. Hasil dari *masking* adalah *sequence* baru yang panjangnya sama dengan dua masukannya, yaitu masukan yang me-*mask* dan masukan yang di-*mask*.

$$c_i = (r_i + m_i + c_{i-1} + r_{i+1}) mod 256 \tag{5}$$

Dengan keterangan c_i adalah hasil *masking* nilai ke-i. r_i adalah nilai ke-i yang di-*mask* atau nilai asli, m_i adalah nilai ke-i yang me-*mask* dalam hal ini adalah hasil dari *Beta chaotic map*, sedangkan *mod 256* supaya nilai yang dihasilkan tetap pada range 0 sampai 255.

Untuk proses pengembalian hasil *masking (unmasking)*, harus diketahui salah satu masukan dari *masking*. Apabila ingin mencari *sequence* yang di-*mask*, maka hasil *masking* dan *sequence* yang me-*mask* harus diketahui. Begitu pula apabila yang dicari adalah *sequence* yang me-*mask*, maka hasil *masking* dan *sequence* yang di-*mask* harus diketahui. Misalnya dari (5) yang ingin didapatkan kembali adalah yang di-*mask* r, maka persamaan yang dibutuhkan ditunjukkan pada (6).

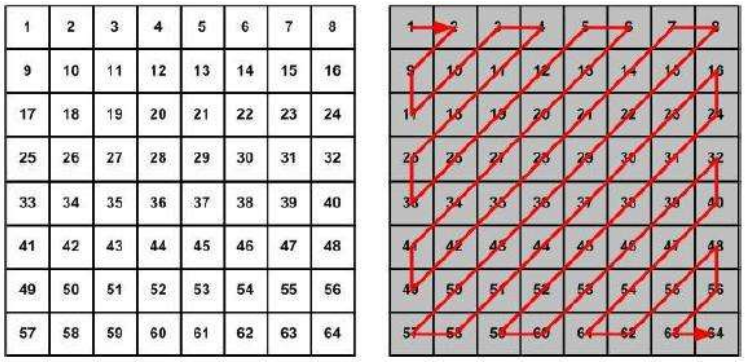
$$r_i = (c_i - m_i - c_{i-1} - r_{i+1}) mod 256 \qquad (6)$$

Di mana r_i adalah *sequence* hasil *unmask* pada nilai ke-i. c_i adalah hasil masking nilai ke-i. m_i adalah nilai ke-i yang me-*mask*.

D. Zigzag Scanning

Zigzag *Scanning* merupakan cara untuk membaca sebuah matriks dengan cara zigzag. Matriks yang dibaca biasanya berbentuk persegi atau berukuran n x n. Matriks dibaca secara zigzag dari kiri atas ke kanan bawah seperti yang ditunjukkan pada Gambar 2 [11]. Zigzag *scanning* dilakukan pada matriks hasil kuantisasi dengan tujuan agar koefisien frekuensi rendah terkelompokkan diawal *sequence*.

Hasil dari zigzag *scanning* adalah sebuah *sequence* yang berisi nilai-nilai matriks yang telah diurutkan sesuai posisinya secara zigzag. Pada proses dekompresi, *sequence* harus dikembalikan menjadi matriks semula. *Sequence* yang telah terbentuk juga harus disusun sesuai uratan zigzag yang telah dilakukan.



Gambar 2 Zigzag Scanning

III. METODE PENELITIAN

Metode yang digunakan pada penelitian ini terdiri dari proses enkripsi dan dekripsi. Untuk proses enkripsi secara garis besar dapat dilihat pada Gambar 2. Dari Gambar 2 dapat dilihat bahwa masukan dari proses enkripsi ini adalah sebuah citra biasa. Keluarannya adalah *cipher image* yang berupa gambar teracak. Sementara kunci-kunci yang digunakan pada proses enkripsi ditunjukkan pada kotak putus-putus.

Penjelasan dari metode enkripsi yang digunakan pada penelitian ini adalah sebagai berikut:

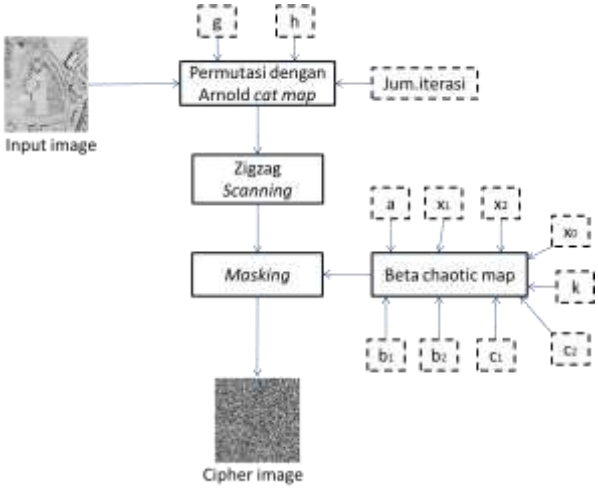
1. Nilai pada citra mula-mula diacak atau dipermutasi menggunakan Arnold *cat map*. Arnold *cat map* sendiri memiliki 3 parameter yang akan dijadikan kunci dari enkripsi yaitu nilai g, h, dan $jum.iterasi$.
2. Hasil dari citra yang telah diacak menggunakan Arnold *cat map* selanjutnya dijadikan sebuah sequence panjang menggunakan zigzag *scanning*.
3. Mengenerate sequence random menggunakan Beta chaotic map menggunakan 9 parameter yang dijadikan kunci yaitu $a, x_0, x_1, x_2, k, b_1, b_2, c_1, c_2$. *Sequence* yang digenerate sebanyak hasil dari langkah 2.
4. Hasil dari langkah 2 dan langkah 3 di*masking* menggunakan (5).
5. Hasil masking yang masih berupa *sequence* panjang disusun kembali menjadi ukuran citra awal dan akan menghasilkan *cipher image* atau citra terenkripsi.

Sementara proses dekripsi yaitu untuk mengembalikan *cipher image* menjadi image awal yang dapat

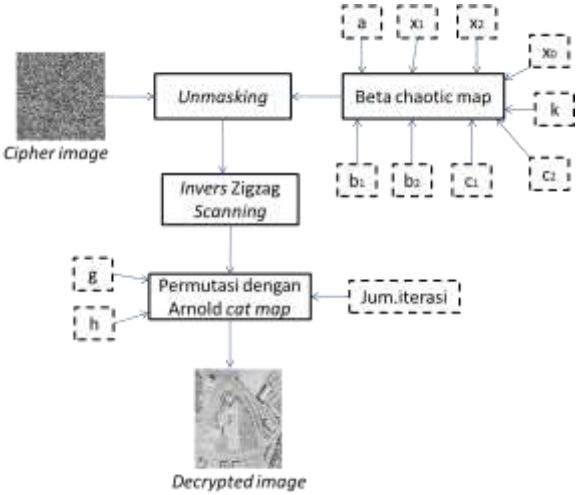
dibaca. Secara garis besar proses dekripsi dapat dilihat pada Gambar 3. Pada dasarnya proses dekripsi yang digunakan merupakan kebalikan dari proses enkripsi.

Penjelasan dari proses dekripsi yang digunakan dijelaskan sebagai berikut:

1. Cipher image disusun menjadi sebuah sequence panjang.
2. Mengenerate *sequence* random menggunakan Beta *chaotic map* menggunakan 9 parameter yang dijadikan kunci yaitu $a, x_0, x_1, x_2, k, b_1, b_2, c_1, c_2$. Kunci yang digunakan harus sama dengan kunci yang digunakan saat melakukan enkripsi. Sequence yang di-*generate* sebanyak panjang hasil dari langkah 1.
3. Hasil dari langkah 1 dan langkah 2 dilakukan *unmasking* atau kebalikan dari *masking* menggunakan (6).
4. Hasil dari langkah 3 disusun menjadi ukuran citra awal dengan cara *invers* zigzag *scanning*.
5. Hasil dari langkah 4 yang sudah berukuran sesuai citra asli dipermutasi menggunakan *invers* Arnold *cat map* sehingga posisi akan kembali seperti semula. Hasilnya berupa citra awal yang dapat dibaca oleh pengguna.



Gambar 3 Proses Enkripsi



Gambar 4 Proses Dekripsi

IV. HASIL DAN PEMBAHASAN

Enkripsi pada citra kebanyakan dilakukan dengan melakukan analisa statistik, dimana penyerang akan mencari hubungan antara citra asli dan citra hasil enkripsi. Tidak hanya dari gambar asli, tapi penyerang juga dapat melihat hubungan antara histogram yang dihasilkan. Pada bagian ini akan dijelaskan mengenai hasil dari penelitian yang telah dilakukan. Yaitu bagaimana menunjukkan bahwa metode yang digunakan tahan terhadap beberapa serangan. Mulai dari serangan brute force, serangan analisa histogram, dan serangan analisa statistik. Pada bagian ini juga membandingkan dengan hasil penelitian sebelumnya, dimana menunjukkan bahwa hasil dari penelitian ini lebih tahan terhadap beberapa serangan yang dapat dilakukan. Pembahasan untuk masing-masing analisa hasil dari proses enkripsi dibagi menjadi beberapa pengujian sebagai berikut:

A. Analisa panjang kunci

Sebuah algoritma enkripsi dikatakan baik ketika memiliki kunci yang panjang sehingga tahan terhadap serangan *brute force*. Serangan *brute force* sendiri adalah serangan dengan mencoba berbagai kemungkinan kunci. Semakin panjang kunci maka akan semakin tahan terhadap serangan *brute force*. Panjang kunci yang digunakan pada algoritma enkripsi ini adalah sepanjang 768 bit yang dibagi menjadi 12 kunci. Panjang kunci tersebut lebih panjang daripada [5] sehingga dapat disimpulkan bahwa penggunaan metode ini lebih tahan terhadap serangan *brute force*.

B. Analisa Histogram

Analisa histogram dilakukan dengan membandingkan histogram citra asli dan histogram yang citra yang telah dienkripsi. Hasil dari analisa histogram dapat dilihat pada Gambar 5. Histogram citra yang telah terenkripsi memiliki bentuk yang sama sekali berbeda dengan histogram citra asli. Selain itu

bentuk dari histogram citra terenkripsi sangat datar dan seragam sehingga susah ditebak bagaimana kemungkinan citra asli. Hal ini menandakan bahwa hasil enkripsi yang dihasilkan tahan terhadap serangan analisa histogram.

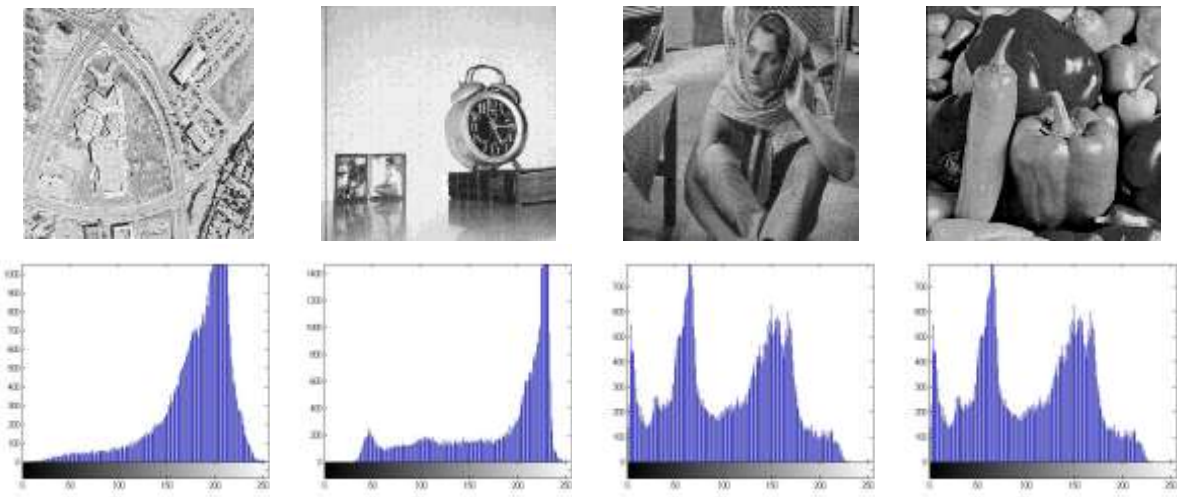
C. Analisa Number of Pixel Change Rate (NPCR)

NPCR digunakan untuk menghitung jumlah pixel yang berbeda pada dua citra. Pada kasus ini yang akan dihitung adalah jumlah pixel yang berbeda antara citra asli dan citra hasil enkripsi. Semakin besar nilai NPCR maka semakin berbeda citra asli dan citra hasil enkripsi. Untuk menghitung NPCR citra yang memiliki panjang *M* dan lebar *N* dapat dilihat pada (7). Bila ingin membandingkan citra asli dan citra hasil enkripsi maka C_1 merupakan pixel citra asli pada posisi (i, j) sedangkan C_2 merupakan pixel citra terenkripsi pada posisi (i, j). Hasil dari NPCR metode ini dan dengan dibandingkan dengan hasil NPCR penelitian sebelumnya [5], [12] dapat dilihat pada tabel 1. Dari tabel 1 dapat dilihat bahwa NPCR pada data yang diujikan adalah mendekati 100%.. Berarti hampir semua nilai pixel citra asli berbeda dari nilai pixel citra terenkripsi. Selain itu nilai NPCR yang dihasilkan dengan metode ini lebih tinggi yang berarti lebih baik dari penelitian sebelumnya karena lebih tahan terhadap serangan diferensial.

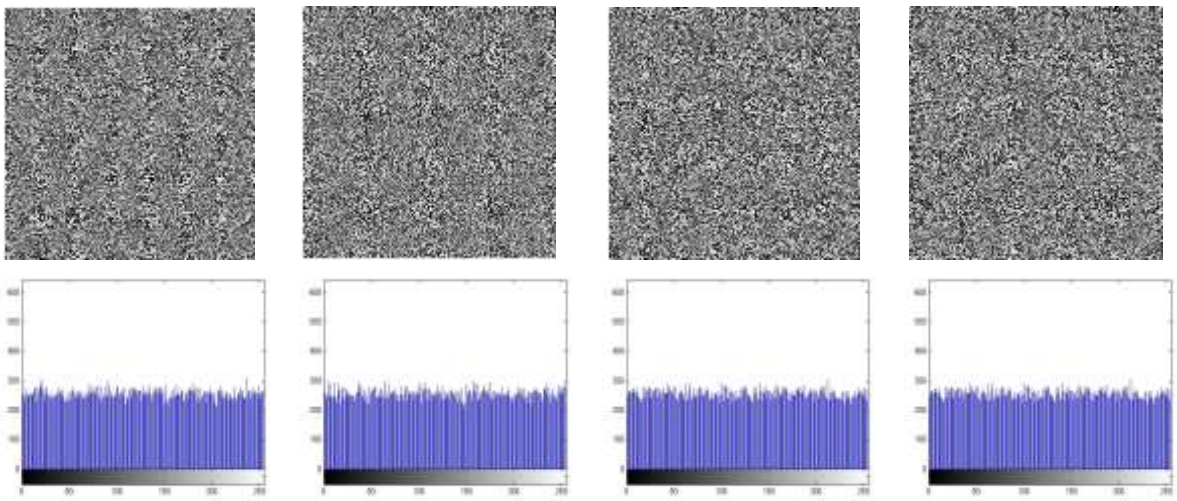
$$NPCR = \sum_{i=1}^{M}\sum_{j=1}^{N} D(i, j) \times \frac{100\%}{M \times N} \tag{7}$$

Dimana

$$D(i, j) = \begin{cases} 0 & if \quad C_1(i, j) = C_2(i, j) \\ 1 & if \quad C_1(i, j) \neq C_2(i, j) \end{cases}$$



Gambar 5 Citra Asli dan histogramnya



Gambar 6 Citra terenkripsi dan histogramnya

D. Analisa Peak Signal-to-Noise Ratio (PSNR)

Peak Signal-to-Noise Ratio (PSNR) merupakan perhitungan untuk mengevaluasi perbedaan citra hasil enkripsi terhadap citra awal. Perhitungan PSNR didasarkan pada perhitungan *Mean Square Error* (MSE). Hasil dari perhitungan MSE dan PSNR dapat dilihat pada tabel 2. Perhitungan PSNR ditunjukkan pada (8) sedangkan perhitungan MSE ditunjukkan pada (9).

$$PSNR = 10 \times log_{10} \frac{255^2}{MSE}(\text{dB}) \tag{8}$$

$$MSE = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} \quad (I_{ij} - I'_{ij})^2 \tag{9}$$

W merupakan lebar citra, sedangkan H adalah panjang citra. I_{ij} merupakan nilai *pixel* pada citra awal dengan posisi (i, j) dan I'_{ij} merupakan nilai *pixel* pada citra terenkripsi dengan posisi (i, j). Satuan PSNR yang dihasilkan adalah dB.

TABEL I
HASIL PERHITUNGAN NPCR DIBANDINGKAN DENGAN PENELITIAN SEBELUMNYA

Nama Citra	Metode yang Di-ajukan	Ref [5]	Ref[12]
Cameraman	99,6490	99,6124	99,6205
Barb	99,6170	99,6215	99,6092
Peppers	99,6201	99,6040	99,6319
Moon surface	99,5941	99,6276	99,6139
Clock	99,6048	99,5727	99,6102
Chemical plant	99,6231	99,6200	99,6121
Lena	99,5621	99,6253	99,6228
Boat	99,6262	99,6227	99,6102
Man	99,6414	99,6189	99,6070
Couple	99,6017	99,6185	99,6399

TABEL II
HASIL MSE DAN PSNR METODE YANG DIAJUKAN

Nama Citra	MSE	PSNR(dB)
Cameraman	3743,0	12,3986
Barb	2752,4	13,7338
Peppers	2830,0	13,6129
Moon surface	3073,3	13,2548
Clock	10523	7,9093
Chemical plant	2287,6	14,5370
Lena	3567,4	12,6073
Boat	4572,0	11,5297
Man	2788,7	13,6767
Couple	3183,1	13,1023

V. KESIMPULAN

Pada penelitian ini dilakukan enkripsi pada data citra dengan skema yang menggabungkan Arnold *cat map* dan Beta *chaotic map.* Arnold *cat map* digunakan untuk melakukan permutasi pada citra. Sementara Beta chaotic map di*generate* menghasilkan nilai random yang di-*mask* dari hasil permutasi. Skema yang diajukan ini membutuhkan kunci yang berukuran besar sehingga akan tahan terhadap serangan *brute force*. Citra terenkripsi yang dihasilkan sangat berbeda dengan citra asli yang dapat dilihat dari perhitungan NPCR dan PSNR. Begitu juga histogram yang dihasilkan oleh citra terenkripsi sangat berbeda dengan histogram citra asli. Histogram citra terenkripsi yang datar dan seragam menunjukkan bahwa hasil dari skema enkripsi ini tahan terhadap serangan analisis histogram.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Institut Teknologi Adhi Tama yang telah memberikan fasilitas dalam rangka terselesaikannya penelitian ini. Selain itu peneliti juga mengucapkan terima kasih

kepada RISTEKDIKTI yang telah memberikan kesempatan dan bantuan finansial sehingga penelitian ini dapat berjalan dengan baik.

DAFTAR PUSTAKA

[1] E. Yavuz, R. Yazıcı, M. C. Kasapbaşı, and E. Yamaç, "A chaos-based image encryption algorithm with simple logical functions," *Comput. Electr. Eng.*, vol. 54, pp. 471–483, Aug. 2016.
[2] Y. Zhou, L. Bao, and C. L. P. Chen, "A new 1D chaotic system for image encryption," *Signal Processing*, vol. 97, pp. 172–182, Apr. 2014.
[3] C. Li, G. Luo, K. Qin, and C. Li, "An image encryption scheme based on chaotic tent map," *Nonlinear Dyn.*, vol. 87, no. 1, pp. 127–133, Jan. 2017.
[4] J. Khan, J. Ahmad, and S. O. Hwang, "An efficient image encryption scheme based on: Henon map, skew tent map and S-Box," in *2015 6th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO)*, 2015, pp. 1–6.
[5] R. Zahmoul, R. Ejbali, and M. Zaied, "Image encryption based on new Beta chaotic maps," *Opt. Lasers Eng.*, vol. 96, pp. 39–49, Sep. 2017.
[6] Z. Hua, Y. Zhou, C.-M. Pun, and C. L. P. Chen, "2D Sine Logistic modulation map for image encryption," *Inf. Sci. (Ny).*, vol. 297, pp. 80–94, Mar. 2015.
[7] A. Sahay and C. Pradhan, "Gauss iterated map based RGB image encryption approach," in *2017 International Conference on Communication and Signal Processing (ICCSP)*, 2017, pp. 0015–0018.
[8] C. Ben Amar, M. Zaied, and A. Alimi, "Beta wavelets. Synthesis and application to lossy image compression," *Adv. Eng. Softw.*, vol. 36, no. 7, pp. 459–474, Jul. 2005.
[9] R. Kumar, A. Kumar, and R. K. Pandey, "Electrocardiogram Signal Compression Using Beta Wavelets," *J. Math. Model. Algorithms*, vol. 11, no. 3, pp. 235–248, Sep. 2012.
[10] C.-H. Yuen and K.-W. Wong, "A chaos-based joint image compression and encryption scheme using DCT and SHA-1," *Appl. Soft Comput.*, vol. 11, no. 8, pp. 5092–5098, Dec. 2011.
[11] "File:Zigzag scanning.jpg - Wikimedia Commons." [Online]. Available: https://commons.wikimedia.org/wiki/File:Zigzag_scanning.jpg. [Accessed: 07-Sep-2018].
[12] A. Belazi, A. A. Abd El-Latif, and S. Belghith, "A novel image encryption scheme based on substitution-permutation network and chaos," *Signal Processing*, vol. 128, pp. 155–170, Nov. 2016.