Jurnal Mahajana Informasi, Vo.4, No.2, 2019
e-ISSN:2527-8290

KOMPRESI DAN PENGAMANAN CITRA MENGGUNAKAN LOSSLESS CHAOS-BASED CRYPTO COMPRESSION SCHEME

Sunario Megawan[1)], **Irpan Adiputra Pardosi**[2)], **William Chaiyanda**[3)], **Louis Adam Rivandy**[4)], **Joshua**[5)]
[1,2,3,4,5)]Program StudiTeknik Informatika STMIK Mikroskil Indonesia Medan
Jl. Thamrin No. 112, 124, 140 Medan 20212 Medan Telp (061)-4573767
e-mail : sunario@mikroskil.ac.id ,irpan@mikroskil.ac.id, williamChai26@gmail.com, louisadamr77@gmail.com, joshuatanjayaa@gmail.com

Abstrak
Pada era digital sekarang ini, citra digital mudah disimpan atauditransmisikan melalui saluran publik sepertiinternet.Pengiriman citra melalui saluran publikrawan akses ilegal dan dapat merugikan pemilik citra digital. Selain masalah keamanan, masalah lain yang muncul adalah ukuran citra yang besar dapat memboros penggunaan kapasitas penyimpanan.Metode yang dapat digunakan untuk melakukan kompresi dan sekaligus pengamanan terhadap citra adalah metode Lossless Chaos-Based Crypto Compression Scheme. Metode ini melakukan kompresi terhadap citra dengan menggunakan metode Arithmetic Coding (AC) yang telah dimodifikasi dengan adanya tambahan penggunaan kunci untuk mengacak hasil AC, sehingga hanya orang-orang tertentu yang dapat mendekripsi dan mendekompresi citra kembali ke awal. Pengacakan dilakukan dengan menggunakan Logistic Map (LM) pseudo random bit generator untuk membangkitkan bit acak yang akan mengacak hasil kompresi dari metode AC.Hasil penelitian berupa aplikasi yang dapat digunakan untuk mencegah akses ilegal terhadap citra digital yang bersifat rahasia dengan cara melakukan proses enkripsi terhadap citra dan menghemat kapasitas ruang penyimpanan harddisk dengan cara melakukan proses kompresi terhadap citra.

Kata Kunci: Kriptografi, Citra Digital, Kompresi Data

1. PENDAHULUAN

Enkripsi citra merupakan teknik untuk melindungi kerahasiaan citra dari pengaksesan ilegal. Enkripsi diperlukan karena dalam era digital sekarang ini citra digital mudah disimpan atau ditransmisikan melalui saluran publik seperti internet. Pengiriman citra melalui saluran publik rawan terhadap penyadapan, dan penyimpanan citra di dalam media storage rawan terhadap pengaksesan oleh pihak-pihak yang tidak memiliki otoritas[7].Hal-hal seperti ini dapat dicegah dengan melakukan pengamanan terhadap citra digital melalui proses enkripsi. Masalah lain yang muncul adalah ukuran citra yang besar dapat memboros penggunaan kapasitas penyimpanan [2]. Oleh karena itu, dibutuhkan proses kompresi untuk menghemat ruang penyimpanan dan sekaligus pengamanan padacitra agar citra hanya dapat diakses oleh pihak tertentu.

Beberapa metode yang dapat digunakan untuk melakukan kompresi dan enkripsi terhadap citra digital adalah penelitian berjudul ”A Modified Encryption Algorithm for Compression of Color Image” yang menggunakan metode Data Encryption Standard (DES) untuk enkripsi dan metode kompresi Huffman untuk kompresi [5]. Penelitian lainnya dibuat oleh Kale dan Natikar yang menggunakan metode Rivest-Shamir-Adleman (RSA) dan 3-D Advanced Encryption Standard (3D-AES) untuk enkripsi dan metode Shanon Fano untuk kompresi [3]. Kemudian penelitian pada tahun 2014 berjudul "Designing an Efficient Image Encryption Then Compression System with Haar and Daubechies Wavelet" membahas mengenai enkripsi citra dengan random permutation dan kompresi citra menggunakan Discrete Wavelet Transform (DWT) [1]. Dalam studi ini, digunakan skema baru dalam melakukan kompresi lossless dan enkripsi gambar. Kompresi lossless dilakukan oleh Arithmetic Coding (AC) sementara enkripsi didasarkan pada generator bit pseudorandom berbasis chaos, sehingga metode ini dinamakan dengan Lossless Chaos-Based Crypto Compression Scheme.

Semua penelitian terdahulu menggunakan metode enkripsi yang terpisah dengan metode kompresi. Kedua tahapan dijalankan pada waktu yang berbeda (enkripsi dulu baru kompresi, atau kompresi dulu baru enkripsi), sehingga waktu proses yang dibutuhkan menjadi relatif lebih lama. Metode yang diusulkan pada aplikasi yang dikembangan penelitian ini, melakukan proses enkripsi dan sekaligus kompresi terhadap citra yang bersifat rahasia. Kedua tahapan dijalankan sekaligus sehingga dapat menghemat waktu. Proses kompresi dan sekaligus enkripsi dilakukan dengan menggunakan metode *Arithmetic Coding* (*AC*) yang telah dimodifikasi dengan adanya tambahan penggunaan kunci untuk mengacak hasil *AC*, sehingga hanya orang-orang tertentu yang dapat mendekripsi dan mendekompresi citra kembali ke awal[6].

2. TINJAUAN PUSTAKA

Proses kompresi data dan proses enkripsi dilakukan dengan menggunakan kombinasi dari metode *Arithmetic Coding* (*AC*) dan *chaos-based pseudorandom bit generator*. Proses enkripsi dan kompresi dilakukan dengan pemberian nilai awal (inisialisasi) pada variabel yang digunakan pada proses kompresi dan dekompresi, dapat dilihat pada algoritma 1: [6]

Algoritma 1 (*Initialize_AC*)

```
1:  y1 ← y0
2:  y2 ← y0
3:  y3 ← y0
4:  Top_value ← 2^N – 1      // N = 30 bits
5:  First_qtr ← Top_value / 4 + 1
6:  Half ← 2 * First_qtr
7:  Third_qtr ← 3 * First_qtr
8:  low ← 0
9:  high ← Top_value
10: S ← [0, …, 255]
11: Freq ← [f0, …, f255]
12: cum_freq ← [0, Freq[S[1]], Freq[S[1]] + Freq[S[2]], …, Σ_{i=0}^{n} Freq[S[i]]
13: R ← Σ_{i=0}^{n} Freq[S[i]]
```

Selanjutnya dilanjutkan dengan proses enkripsi dan kompresi data menggunakan algoritma 2:

Algoritma 2 (*Compression_Encryption*)

```
1:  Initialize_AC()
2:  bits_to_follow ← 0
3:  for i ← 1 to n do
4:     y2 ← 4 * y2 * (1 – y2)
5:     if y2 < 0.5 then                    // enkripsi
6:        y2 ← 4 * y2 * (1 – y2)
7:        j ← ⌊y2 * (n-1)⌋ + 2             // enkripsi
8:        S ← [S[i], S[i+1], …, S[n], S[1], …, S[i-1]]
9:        cum_freq ← [0, Freq[S[1]], Freq[S[1]] + Freq[S[2]], …, Σ Freq[S[i]]
10:    end if
11:    j ← position_of_symbol(S, xi)
12:    range ← high – low + 1
13:    high ← low + range * cum_freq[j]/R
14:    low ← low + range * cum_freq[j-1]/R
15:    for (; ;) do
16:       if high < Half then
17:          y2 = 4 * y2 * (1-y2)
18:          output_bit (0 ⊕ (y2 < 0.5))   // enkripsi
19:          low ← 2 * low
20:          high ← 2 * high + 1
21:       else if low ≥ Half then
22:          y2 = 4 * y2 * (1-y2)
23:          output_bit (1 ⊕ (y2 < 0.5))   // enkripsi
24:          low ← 2 * (low – Half)
25:          high ← 2 * (high – Half) + 1
26:       else if (low ≥ First_qtr) and (high < Third_qtr) then
27:          bits_to_follow ← bits_to_follow + 1
28:          low ← 2 * (low – First_qtr)
29:          high ← 2 * (high – First_qtr) + 1
30:       else
31:          break;
32:       end if
33:    end for
34: end for
35: bits_to_follow ← bits_to_follow + 1
36: if low < First_qtr then
37:    y3 = 4 * y3 * (1-y3)
38:    output_bit (0 ⊕ (y3 < 0.5))        // enkripsi
39: else
40:    y3 = 4 * y3 * (1-y3)
41:    output_bit (1 ⊕ (y3 < 0.5))        // enkripsi
42: end if
```

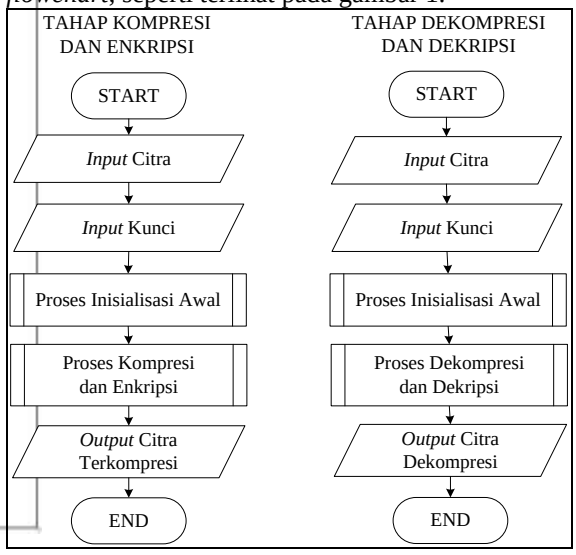
Untuk melakukan proses deskripsi dan dekompresi dapat menjalankan algoritma 3:

Algoritma 3 (*Decompression_Decryption*)

```
1:
2:  Initialize_AC()
3:  value ← 0
4:  for i ← 1 to N do
5:     y2 ← 4 * y2 * (1 – y2)
6:     value ← 2 * value + input_bit() ⊕ (y2 < 0.5)   // dekripsi
7:  end for
8:  for (; ;) do
9:     y2 ← 4 * y2 * (1 – y2)
10:    if y2 < 0.5 then                    // dekripsi
          y2 ← 4 * y2 * (1 – y2)
11:       j ← ⌊y2 * (n-1)⌋ + 2             // dekripsi
12:       S ← [S[i], S[i+1], …, S[n], S[1], …, S[i-1]]
13:       cum_freq ← [0, Freq[S[1]], Freq[S[1]] + Freq[S[2]], …, Σ Freq[S[i]]
14:    end if
15:    range ← high – low + 1
16:    cum ← ((value – low + 1) * R – 1) / range
17:    j ← 1
18:    while cum_freq[j] ≤ cum do
19:       j ← j + 1
20:    end while
21:    high ← low + range * cum_freq[j]/R
22:    low ← low + range * cum_freq[j-1]/R
23:    for (; ;) do
24:       if high < Half then
25:          value ← 2 * value
26:          low ← 2 * low
27:          high ← 2 * high + 1
28:       else if low ≥ Half then
29:          value ← 2 * (value – Half)
30:          low ← 2 * (low – Half)
31:          high ← 2 * (high – Half) + 1
32:       else if (low ≥ First_qtr) and (high < Third_qtr) then
33:          value ← 2 * (value – First_qtr)
34:          low ← 2 * (low – First_qtr)
35:          high ← 2 * (high – First_qtr) + 1
36:       else
37:          break;
38:       end if
39:       y2 ← 4 * y2 * (1 – y2)
40:       value ← value + input_bit() ⊕ (y2 < 0.5)      // dekripsi
41:    end for
42:    symbol ← S[i]
```

3. METODE PENELITIAN

Proses kompresi dan enkripsi, serta proses dekompresi dan dekripsi dengan menggunakan menggunakan *Lossless Chaos-Based Crypto Compression Scheme*, atau dalam hal ini adalah kombinasi antara metode *Arithmetic Coding (AC)* dan*Logistic Map* (*LM*) *pseudo random bit generator* dapat digambarkan dalam bentuk *flowchart*, seperti terlihat pada gambar 1.



Gambar 1. Proses Pengamanan dan Kompresi Citra Digital

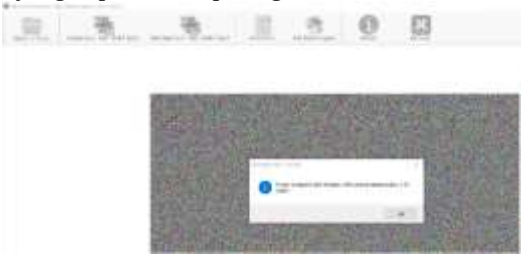
4. HASIL DAN PEMBAHASAN

Hasil penelitian ini menghasilkan aplikasi kompresi dan pengamanan citra digital dengan menggunakan *Lossless Chaos-Based Crypto Compression Scheme*. Tampilan *aplikasi* dapat dilihat pada gambar 2.Untuk memulai proses kompresi dan enkripsi citra, pengguna dapat menekan tombol "Kompresi dan Enkripsi" pada *toolbar* di atas *form* utama dari aplikasi. *Forminput* kunci enkripsiakan tampil seperti terlihat pada gambar 3. Untuk menghasilkan nilai kunci secara acak, pengguna dapat menekan tombol "Acak" pada *form input* kunci enkripsi. Nilai kunci *generator* bit acak adalah nilai *y1*, *y2* dan *y3* yang diisi dengan nilai antara 0 dan 1. Contoh nilai kunci yang digunakan seperti terlihat pada gambar 3.

Gambar 2. Tampilan Citra pada *Form* Utama

Gambar 3. Nilai Kunci Enkripsi yang Digunakan

Setelah nilai kunci dimasukkan, pengguna menekan tombol "OK" pada *form input* kunci untuk memulai proses kompresi dan enkripsi terhadap citra. Hasil proses kompresi dan enkripsi citra berupa citra biner yang dapat dilihat pada gambar 4 berikut.



Gambar 4. Hasil Proses Kompresi dan Enkripsi

Pengujian dilakukan pada aplikasi yang dihasilkan untuk mengetahui keberhasilan proses enkripsi dan kompresi citra digital dengan menggunakan 10 citra digital. Hasil pengujian dapat dilihat pada tabel 1 dan tabel 2.

Tabel 1. Hasil Pengujian Proses Kompresi

Citra	Proses	Waktu Proses	Ukuran Citra Sebelum Proses	Ukuran Citra Setelah Proses	Rasio Kompresi (%)
01.bmp	Kompresi dan Enkripsi	7.48	852 x 480 (1,226,934 *byte*)	852 x 367 (938,106 *byte*)	76.46
02.bmp	Kompresi dan Enkripsi	1.84	300 x 354 (318,654 *byte*)	300 x 277 (249,354 *byte*)	78.25
03.bmp	Kompresi dan Enkripsi	6.25	650 x 533 (1,040,470 *byte*)	650 x 418 (815,990 *byte*)	78.43
04.bmp	Kompresi dan Enkripsi	9.44	756 x 603 (1,367,658 *byte*)	756 x 475 (1,077,354 *byte*)	78.8
05.bmp	Kompresi dan Enkripsi	5.73	702 x 465 (980,274 *byte*)	702 x 327 (689,370 *byte*)	70.32
06.bmp	Kompresi dan Enkripsi	12.14	886 x 621 (1,651,914 *byte*)	886 x 493 (1,311,434 *byte*)	79.39
07.bmp	Kompresi dan Enkripsi	14.19	975 x 758 (2,219,478 *byte*)	975 x 394 (1,153,686 *byte*)	51.98
08.bmp	Kompresi dan Enkripsi	7.67	846 x 468 (1,188,774*byte*)	846 x 359 (911,914 *byte*)	76.71
09.bmp	Kompresi dan Enkripsi	13.81	914 x 759 (2,082,750 *byte*)	914 x 395 (1,083,934 *byte*)	52.04
10.bmp	Kompresi dan Enkripsi	16.19	855 x 862 (2,213,670 *byte*)	855 x 655 (1,682,094 *byte*)	75.99

Tabel 2. Hasil Pengujian Nilai Kunci dengan Selisih Interval Kunci yang Kecil

Nilai Kunci Enkripsi-1	Perubahan Nilai Kunci	Nilai Kunci Enkripsi-2	MSE
y1 = 0.17745455 *y2* = 0.87978795 *y3* = 0.62917225	Nilai *y1* ditambah 0.00000001	*y1* = 0.17745456 *y2* = 0.87978795 *y3* = 0.62917225	10871.68607
y1 = 0.17745455 *y2* = 0.87978795 *y3* = 0.62917225	Nilai *y2* ditambah 0.00000001	*y1* = 0.17745455 *y2* = 0.87978796 *y3* = 0.62917225	10901.53442
y1 = 0.17745455 *y2* = 0.87978795 *y3* = 0.62917225	Nilai *y3* ditambah 0.00000001	*y1* = 0.17745455 *y2* = 0.87978795 *y3* = 0.62917226	10911.75641
y1 = 0.17745455 *y2* = 0.87978795 *y3* = 0.62917225	Nilai *y1* dan *y2* ditambah 0.00000001	*y1* = 0.17745456 *y2* = 0.87978796 *y3* = 0.62917225	10911.7564
y1 = 0.17745455 *y2* = 0.87978795 *y3* = 0.62917225	Nilai *y2* dan *y3* ditambah 0.00000001	*y1* = 0.17745455 *y2* = 0.87978796 *y3* = 0.62917226	10928.90275
y1 = 0.17745455 *y2* = 0.87978795 *y3* = 0.62917225	Nilai *y1* dan *y3* ditambah 0.00000001	*y1* = 0.17745456 *y2* = 0.87978795 *y3* = 0.62917226	10924.82318
y1 = 0.17745455 *y2* = 0.87978795 *y3* = 0.62917225	Nilai *y1*, *y2* dan *y3* ditambah 0.00000001	*y1* = 0.17745456 *y2* = 0.87978796 *y3* = 0.62917226	10941.19561
y1 = 0.17745455 *y2* = 0.87978795 *y3* = 0.62917225	Nilai *y1* dikurangi 0.00000001	*y1* = 0.17745454 *y2* = 0.87978795 *y3* = 0.62917225	10866.93454
y1 = 0.17745455 *y2* = 0.87978795 *y3* = 0.62917225	Nilai *y2* dikurangi 0.00000001	*y1* = 0.17745455 *y2* = 0.87978794 *y3* = 0.62917225	10898.22034
y1 = 0.17745455 *y2* = 0.87978795 *y3* = 0.62917225	Nilai *y3* dikurangi 0.00000001	*y1* = 0.17745455 *y2* = 0.87978795 *y3* = 0.62917224	10924.20942
y1 = 0.17745455 *y2* = 0.87978795 *y3* = 0.62917225	Nilai *y1* dan *y2* dikurangi 0.00000001	*y1* = 0.17745454 *y2* = 0.87978794 *y3* = 0.62917225	10866.93457
y1 = 0.17745455 *y2* = 0.87978795 *y3* = 0.62917225	Nilai *y2* dan *y3* dikurangi 0.00000001	*y1* = 0.17745455 *y2* = 0.87978794 *y3* = 0.62917224	10987.22313
y1 = 0.17745455 *y2* = 0.87978795 *y3* = 0.62917225	Nilai *y1* dan *y3* dikurangi 0.00000001	*y1* = 0.17745454 *y2* = 0.87978795 *y3* = 0.62917224	10938.23101
y1 = 0.17745455 *y2* = 0.87978795 *y3* = 0.62917225	Nilai *y1*, *y2* dan *y3* dikurangi 0.00000001	*y1* = 0.17745454 *y2* = 0.87978794 *y3* = 0.62917224	10971.94275

5. KESIMPULAN

Berdasarkan pengujian yang telah dilakukan, maka dapat disimpulkan beberapa hal sebagai berikut :

1. Aplikasi dapat digunakan untuk mencegah akses ilegal terhadap citra digital yang bersifat rahasia dengan cara melakukan proses enkripsi terhadap citra. Citra yang telah di kompresi dan dienkripsi, berhasil dikembalikan ke citra semula dengan nilai kunci yang sama, sebaliknya citra hasil kompresi dan enkripsi tidak dapat dikembalikan ke citra awal dengan nilai kunci yang berbeda meskipun dengan perbedaan nilai yang sangat kecil, sehingga dapat disimpulkan bahwa nilai kunci yang digunakan dalam proses pengamanan harus sama.
2. Aplikasi dapat digunakan untuk menghemat kapasitas ruang penyimpanan *harddisk* dengan cara melakukan proses kompresi terhadap citra yang diketahui dari hasil pengujian dengan rasio 51.98% sampai 79.39%.

DAFTAR PUSTAKA

[1]Aujla, H.K., dan Sharma, R., 2014, *Designing an Efficient Image Encryption-Then-Compression System with Haar and Daubechies Wavelet*, Punjab Technical University, India.

[2] Hardianto, R.H., 2017, *Implementasi dan Analisis Kompresi Hybrid pada Citra Medis Digital Hasil Rontgen Kanker Payudara*, Telkom University, Bandung.

[3]Kale dan Natikar, 2014, *Secured Mobile Messaging for Android Application using 3D-AES, PGP and Steganography*, Vishwabharati Academy, India.

[4] Madhu, B., et.al. 2016, *An Overview of Image Security Techniques*, Visvesvaraya Technological University, India.

[5] Mariselvi, C., dan Kumar, A., 2014, *A Modified Encryption Algorithm for Compression of Color Image*, ManonmaniamSundaranar University, Ayikudy.

[6]Masmoudi, A., dan Puech, W., 2014, *Loseless Chaos-Based Crypto-Compression Schme for Image Protection*, University of Sfax, Tunisia.

[7] Rinaldi Munir, 2012, *Analisis Keamanan Algoritma Enkripsi Citra Digital Menggunakan Kombinasi Dua Chaos Map dan Penerapan Teknik Selektif*, JUTI: Jurnal Ilmiah Teknologi Informasi, Surabaya.