

Implementasi Algoritma Serpent untuk Mengamankan Citra Digital

Sri Dewi, Syafmi Giffari Sipayung, Fatimah

Prodi Teknik Informatika, STMIK Budi Darma, Medan, Indonesia
Jalan Sisingamangaraja No. 338, Medan, Indonesia

Abstrak

Saat ini citra digital sangat rentan terhadap pembajakan yang dilakukan oleh oknum yang tidak bertanggung jawab kemudian di salahkan. Hal tersebut akan sangat merugikan banyak pihak terutama pembuat dan pemilik citra digital. Agar pembajakan terhadap citra digital tidak, maka dilakukanlah sebuah pengamanan berdasarkan teknik kriptografi. Salah satu algoritma kriptografi yang dapat digunakan adalah algoritma serpent yang termasuk salah satu pengembangan teknik kriptografi modern. Algoritma serpent juga pernah menjadi salah satu kandidat dari blok cipher AES. Penelitian ini menguraikan bagaimana pengamanan citra digital berjenis RGB dengan format JPG. Citra tersebut akan disandikan berdasarkan algoritma serpent. Algoritma Serpent mempunyai cara kerja yang efektif sehingga dapat diimplementasikan untuk mengamankan citra digital sehingga terhindar dari penyalahgunaan oleh orang lain.

Kata Kunci : Keamanan, Citra Digital, Kriptografi, Algoritma Serpent, RGB

1. PENDAHULUAN

Negara Indonesia saat ini merupakan salah satu negara yang menerapkan Revolusi Industri 4.0, yang mana segala sesuatunya berkaitan dengan teknologi termasuk dalam pemanfaatan citra digital, literasi teknologi menunjukkan kemampuan untuk memanfaatkan teknologi digital guna mengolah data dan informasi. Citra Digital merupakan suatu matriks yang bersifat dua dimensi yang didalamnya terdiri dari elemen-elemen yang menyatakan tingkat keabuan dari elemen suatu gambar[1]. Hingga saat ini citra digital sangat berperan penting dalam menyampaikan informasi visual karena citra digital juga merupakan salah satu komponen multimedia.

Masalah keamanan pada sebuah citra digital biasanya dilakukan dengan tiga teknik yaitu teknik kriptografi, teknik Watermarking dan Teknik Steganografi. Kriptografi merupakan ilmu yang mempelajari teknik-teknik matematika dan berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, serta otentikasi[1]. Teknik kriptografi dalam proses distribusi informasi berlangsung dengan dua arah. Berbeda dengan Teknik Steganografi dan Watermarking yang menggunakan teknik pengamanan data dengan menyembunyikan pesan atau informasi pada sebuah gambar atau objek tertentu sehingga tidak terlihat perubahan objek penyembunyi secara signifikan bila dilihat secara kasat mata. Hingga saat ini sangat banyak algoritma kriptografi yang dapat dimanfaatkan untuk mengamankan data termasuk citra digital, salah satunya adalah algoritma serpent [1][2]. Citra digital juga memiliki banyak jenis sesuai dengan kebutuhan, salah satunya adalah citra warna atau RGB.

Penelitian ini menguraikan bagaimana mengamankan citra digital berjenis RGB dan berformat JPG berdasarkan algoritma serpent. Proses tersebut diawali dengan transformasi warna setiap pixel citra digital menjadi nilai menggunakan aplikasi matlab. Nilai-nilai *pixel* citra yang didapatkan akan dienkripsikan berdasarkan algoritma serpent sehingga dihasilkan cipher *image*. *Cipher image* yang dihasilkan tidak lagi memperlihatkan pola-pola citra asli sehingga dapat menyulitkan pihak ketiga untuk memahami gambar asli.

Tabel 1. Penelitian Terkait

No	Penulis	Judul	Kesimpulan
1	Arinten Dewi Hidayat dan Irawan Afrianto	Sistem Kriptografi Citra Digital Pada Jaringan Intranet Menggunakan Metode Kombinasi Chaos Map dan Teknik Selektif[3]	Berdasarkan penelitian terdahulu dapat disimpulkan bahwa pengiriman citra di intranet menunjukkan citra yang telah terenkripsi tidak dapat dilihat aslinya seperti halnya yang belum terenkripsi
2	Tan Samuel Permana, Christy Atika Sari, Eko Hari Rachmawanto, dkk	Implementasi Pengamanan Citra Digital Berbasis Metode Kriptografi Vernam Cipher[4]	Berdasarkan penelitian terdahulu dapat disimpulkan bahwa pemanfaatan kunci yang acak pada proses enkripsi dan dekripsi penyerang mengalami kerumitan untuk mengetahui dan memahami citra asli.
3	Bayu Rizki R, R. Rumani M, Muhammad Nasrun	Analisis Perbandingan Antara Algoritma Kriptografi Serpent dan AES pada Implementasi Enkripsi SMS di Perangkat Android[5]	Berdasarkan penelitian terdahulu dapat disimpulkan bahwa proses enkripsi dan dekripsi berdasarkan algoritma serpent akan lebih cepat bila diimplementasikan pada data dengan panjang pesan 16 dan 300 karakter.

2. METODOLOGI PENELITIAN

2.1 Kriptografi

Kriptografi merupakan ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, serta otentikasi[1]. Terdapat 2 jenis algoritma kriptografi berbasis kunci yaitu

kunci simetris dan kunci publik. Kriptografi sebenarnya adalah suatu metode yang sering kali digunakan untuk melindungi berbagai macam data yang prosesnya disebut dengan enkripsi[2]. Proses enkripsi, dekripsi dan kunci merupakan fungsi pokok dari teknik kriptografi agar tujuan keamanan terhadap data tercapai[6].

2.2 Algoritma Serpent

Algoritma serpent merupakan algoritma yang dapat digunakan untuk melakukan proses enkripsi data sehingga data aslinya hanya dapat dibaca oleh individu yang memiliki kunci enkripsi yang telah dibuat[7]. Algoritma ini menggunakan teknik *cipher* blok dengan *key* simetris. Serpent memiliki ukuran blok 128, 192, dan 256 bit, teknik *cipher* yang dilakukan berdasarkan struktur jaringan substitusi permutasi dengan jumlah *round* 32 kali pada 4 blok dengan ukuran masing-masing 32 bit. Setiap *round* akan dilakukan penerapan 1 dari 8 SBOX berukuran 4 x 4 bit sebanyak 32 kali secara paralel serta terdiri dari 32 putaran[5],[7], yaitu :

1. Kunci di satukan dalam setiap putaran yang mana pada setiap putarannya terdiri dari 128-bit *subkey* di *exclusive OR*-kan dengan data B_i secara *live*.
2. *S-Box* merupakan campuran dari 128-bit *input* dan kunci yang termasuk empat *word* 32-bit. SBOX yang diterapkan sebagai urutan operasi karena berada diperangkat keras yang diaplikasikan untuk setiap empat *word* dan hasilnya *shift* kiri pada putaran akhir, transformasi linear ini diganti dengan kunci tambahan pencampuran $B_{32} = S_7 (B_{31} K_{31}) K_{32}$. Jadi pada setiap tahap, $IP(B_i) = ^A B_i$ dan $IP(K_i) = ^A K_i$.

Transformasi linear yang digunakan L pada $Y_i = (y_0, y_1, y_2, y_3)$ ditentukan sebagai berikut :

$$\begin{aligned} X_0, X_1, X_2, X_3 &:= S_i(B_i \oplus K_i) \\ X_0 &:= X_0 \lll 13 \\ X_1 &:= X_1 \oplus X_0 \oplus X_2 \\ X_2 &:= X_2 \oplus X_1 \oplus (X_0 \lll 3) \\ X_1 &:= X_1 \lll 1 \\ X_2 &:= X_2 \lll 7 \\ X_0 &:= X_0 \oplus X_1 \oplus X_2 \\ X_2 &:= X_2 \oplus X_3 \oplus (X_1 \lll 7) \\ X_0 &:= X_0 \lll 5 \\ X_2 &:= X_2 \lll 22 \\ B_{i+1} &:= X_0, X_1, X_2, X_3 \end{aligned}$$

Dimana simbol \lll dinotasikan sebagai rotasi kiri dan \ll didenotasikan untuk shift kiri.

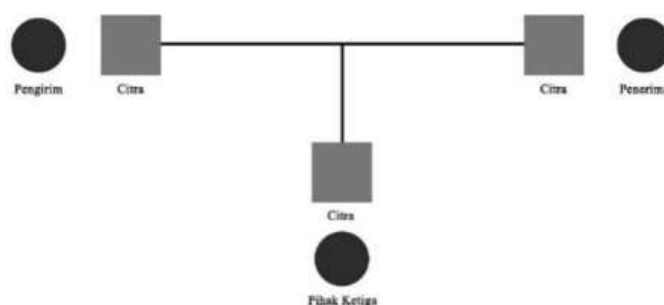
2.3 Citra Digital

Citra (*Image*) secara literal merupakan sebuah larik (array) yang berisi nilai-nilai real maupun kompleks yang dipresensasikan dengan deretan bit tertentu, suatu citra dapat di defenisikan sebagai $f(x,y)$ berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial, dan amplitudo f di titik (x,y) dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut. Apabila nilai x, y dan nilai amplitudo f secara keseluruhan berhingga (*finite*) dan bernilai diskrit maka dapat di katakan bahwa citra tersebut adalah citra digital[1],[4]. Citra digital juga memiliki beberapa jenis citra yang salah satunya adalah citra RGB, yang mana citra RGB terdiri dari 3 warna yaitu *red*, *Green*, dan *Blue*[8]. Begitu juga dengan format nya, format citra digital juga memiliki beberapa macam yaitu salah satunya format jpg[9].

3. ANALISA DAN PEMBAHASAN

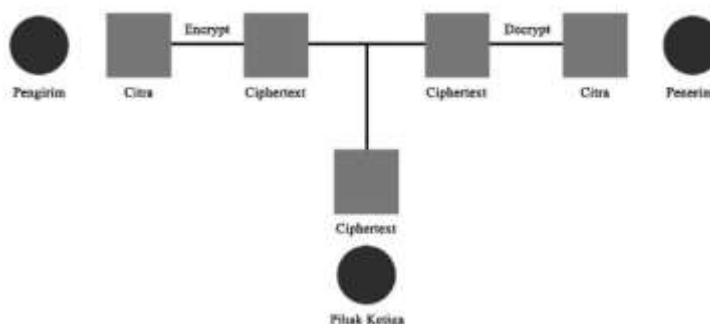
3.1 Analisa

Permasalahan dalam penelitian ini adalah bagaimana mengimplementasikan algoritma serpent untuk mengamankan citra digital berjenis RGB (*True Color*) dengan ekstensi atau format jpg. Pengamanan ini dilakukan karena mudahnya pengaksesan pada citra digital oleh semua orang, sehingga rentan terhadap manipulasi *value* ataupun informasi dari citra digital. Kerentanan yang dimaksud adalah dalam pendistribusian atau dalam perjalanan dari pengirim kepada penerima, ilustrasi dapat dilihat pada pada gambar 1.



Gambar 1. Ilustrasi Pengiriman Tanpa Enkripsi

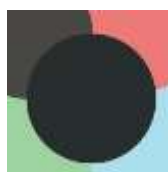
Pengenkripsian atau pengamanan dilakukan karena data tersebut bersifat sangat rahasia, sehingga harus tahan terhadap pembajakan oleh pihak ketiga. Algoritma serpent merupakan salah satu metode pengamanan kriptografi dalam menyelesaikan masalah pembajakan tersebut, hal tersebut dapat digambarkan pada gambar 2.



Gambar 2. Ilustrasi Pengiriman Setelah Enkripsi

3.2 Pembahasan

Penelitian ini menggunakan algoritma serpent yang berkonsep *symmetric cryptosystem*. *Symmetric cryptosystem* sangat mengedepankan kerahasiaan kunci untuk enkripsi dan dekripsi. Pengenkripsian terhadap citra digital dengan jenis RGB dan format jpg menggunakan gambar sampel dengan nama Sampel.jpg dapat dilakukan dengan melakukan langkah-langkah sebagai berikut:



Gambar 3. Sampel.jpg

1. Memecah bagian RGB menjadi 3 bagian yaitu R,G, dan B. *Tools* yang digunakan untuk memisahkan setiap bagian tersebut adalah matlab, pada pengambilan nilai awal adalah dengan hanya menggunakan nilai *pixel* dari Red saja, untuk langkah mengenkripsi Green dan Blue dilakukan dengan cara yang sama yaitu menggunakan *script* berikut ini.

Tabel 2. Nilai Pixel Red Sampel.jpg

71	70	69	68	67	68	70	70	69	73	67	66	71	63	56	118
70	67	67	71	72	70	67	67	62	70	73	75	76	70	64	116
69	67	68	71	71	68	68	70	74	75	75	71	67	61	51	82
68	71	71	68	66	67	71	75	81	70	61	52	49	47	37	45
67	72	71	66	66	72	72	66	57	47	42	36	40	44	40	39
68	70	68	67	72	75	64	47	41	41	45	41	45	45	48	46
70	67	68	71	72	64	49	38	43	47	52	46	50	43	46	45
70	67	70	75	66	47	38	39	44	46	47	42	52	46	48	46
73	63	76	73	47	41	43	49	46	46	46	46	46	46	46	46
71	67	74	63	43	43	44	46	46	46	46	46	46	46	46	46
69	72	69	50	39	46	45	44	46	46	46	46	46	46	46	46
71	75	62	43	39	46	46	45	46	46	46	46	46	46	46	46
73	72	54	41	42	46	46	49	46	46	46	46	46	46	46	46
72	67	48	42	45	44	46	49	46	46	46	46	46	46	46	46
66	62	45	43	44	44	48	46	46	46	46	46	46	46	46	46
61	59	44	42	42	44	50	42	46	46	46	46	46	46	46	46

```
Image = imread (sample.jpg);
R = Image (:,:,3);
G = Image (:,:,2);
B = Image (:,:,1);
asci = uint8(R)
asci = uint8(G)
asci = uint8(B)
```

2. Mengambil nilai desimal pixelnya pada setiap bagian R, G, dan B.
3. Pengenkripsian dilakukan pada masing-masing R, G, B. Enkripsi menggunakan algoritma serpent dengan *cipher* blok 128 bit, yang artinya mengenkripsi 16 karakter dalam sekali putaran. Setiap satu *pixel* dalam citra dihitung 8 bit atau satu karakter, sehingga akan mengenkripsi 16 *pixel* pada algoritma tersebut. *Script* yang digunakan adalah bahasa pemrograman visual basic.

```
Private Sub blokEncrypt(ByRef in_blk() As blokData, ByRef out_blk() As blokData)
Dim g, e, c, a, b, d, f, h As blokData
```

```

#If BLOCK_REVERSE Then
    a = bswap(in_blk(3))
    b = bswap(in_blk(2))
    c = bswap(in_blk(1))
    d = bswap(in_blk(0))
#Else
    MCopy (in_blk(0), a)
    MCopy (in_blk(1), b)
    MCopy (in_blk(2), c)
    MCopy (in_blk(3), d)
#End If

k_xor(0, a, b, c, d) : sb0(a, b, c, d, e, f, g, h) : rot(e, f, g, h)
k_xor(1, e, f, g, h) : sb1(e, f, g, h, a, b, c, d) : rot(a, b, c, d)
k_xor(2, a, b, c, d) : sb2(a, b, c, d, e, f, g, h) : rot(e, f, g, h)
k_xor(3, e, f, g, h) : sb3(e, f, g, h, a, b, c, d) : rot(a, b, c, d)
k_xor(4, a, b, c, d) : sb4(a, b, c, d, e, f, g, h) : rot(e, f, g, h)
k_xor(5, e, f, g, h) : sb5(e, f, g, h, a, b, c, d) : rot(a, b, c, d)
k_xor(6, a, b, c, d) : sb6(a, b, c, d, e, f, g, h) : rot(e, f, g, h)
k_xor(7, e, f, g, h) : sb7(e, f, g, h, a, b, c, d) : rot(a, b, c, d)
k_xor(8, a, b, c, d) : sb0(a, b, c, d, e, f, g, h) : rot(e, f, g, h)
k_xor(9, e, f, g, h) : sb1(e, f, g, h, a, b, c, d) : rot(a, b, c, d)
k_xor(10, a, b, c, d) : sb2(a, b, c, d, e, f, g, h) : rot(e, f, g, h)
k_xor(11, e, f, g, h) : sb3(e, f, g, h, a, b, c, d) : rot(a, b, c, d)
k_xor(12, a, b, c, d) : sb4(a, b, c, d, e, f, g, h) : rot(e, f, g, h)
k_xor(13, e, f, g, h) : sb5(e, f, g, h, a, b, c, d) : rot(a, b, c, d)
k_xor(14, a, b, c, d) : sb6(a, b, c, d, e, f, g, h) : rot(e, f, g, h)
k_xor(15, e, f, g, h) : sb7(e, f, g, h, a, b, c, d) : rot(a, b, c, d)
k_xor(16, a, b, c, d) : sb0(a, b, c, d, e, f, g, h) : rot(e, f, g, h)
k_xor(17, e, f, g, h) : sb1(e, f, g, h, a, b, c, d) : rot(a, b, c, d)
k_xor(18, a, b, c, d) : sb2(a, b, c, d, e, f, g, h) : rot(e, f, g, h)
k_xor(19, e, f, g, h) : sb3(e, f, g, h, a, b, c, d) : rot(a, b, c, d)
k_xor(20, a, b, c, d) : sb4(a, b, c, d, e, f, g, h) : rot(e, f, g, h)
k_xor(21, e, f, g, h) : sb5(e, f, g, h, a, b, c, d) : rot(a, b, c, d)
k_xor(22, a, b, c, d) : sb6(a, b, c, d, e, f, g, h) : rot(e, f, g, h)
k_xor(23, e, f, g, h) : sb7(e, f, g, h, a, b, c, d) : rot(a, b, c, d)
k_xor(24, a, b, c, d) : sb0(a, b, c, d, e, f, g, h) : rot(e, f, g, h)
k_xor(25, e, f, g, h) : sb1(e, f, g, h, a, b, c, d) : rot(a, b, c, d)
k_xor(26, a, b, c, d) : sb2(a, b, c, d, e, f, g, h) : rot(e, f, g, h)
k_xor(27, e, f, g, h) : sb3(e, f, g, h, a, b, c, d) : rot(a, b, c, d)
k_xor(28, a, b, c, d) : sb4(a, b, c, d, e, f, g, h) : rot(e, f, g, h)
k_xor(29, e, f, g, h) : sb5(e, f, g, h, a, b, c, d) : rot(a, b, c, d)
k_xor(30, a, b, c, d) : sb6(a, b, c, d, e, f, g, h) : rot(e, f, g, h)
k_xor(31, e, f, g, h) : sb7(e, f, g, h, a, b, c, d) : k_xor(32, a, b, c, d)
#If BLOCK_REVERSE Then
    out_blk(3) = bswap(a)
    out_blk(2) = bswap(b)
    out_blk(1) = bswap(c)
    out_blk(0) = bswap(d)
#Else
    MCopy (a, out_blk(0))
    MCopy (b, out_blk(1))
    MCopy (c, out_blk(2))
    MCopy (d, out_blk(3))
#End If
End Sub

```

Dengan menggunakan *script* diatas maka menghasilkan nilai-nilai *pixel* baru dari warna *Red*, yaitu:

Tabel 3. Hasil Enkripsi.jpg

119	118	119	121	123	124	123	127	127	128	131	130	128	129	133	
120	120	120	122	123	122	120	117	122	122	126	130	131	129	130	134
121	121	121	121	121	120	117	115	117	118	122	128	130	130	132	136
124	122	119	117	116	116	117	117	115	115	118	124	126	128	132	137

122	121	118	116	115	116	117	119	117	115	116	119	122	125	131	138
118	119	120	119	118	117	117	117	119	116	115	116	119	122	130	137
119	120	120	119	117	116	116	117	118	115	114	116	118	122	129	137
123	122	120	116	114	114	117	120	116	114	114	116	119	122	129	136
121	119	118	118	120	120	118	116	117	119	120	117	115	120	132	143
121	118	115	113	114	115	115	115	119	120	121	118	118	123	134	143
131	127	121	117	116	118	120	122	116	117	118	117	119	125	134	141
135	131	124	118	116	116	118	119	115	116	118	120	124	130	137	142
136	133	128	122	118	116	116	116	118	120	123	126	131	137	143	146
138	137	134	129	125	121	120	120	119	121	125	129	133	137	141	144
135	136	135	132	128	125	124	125	124	128	132	135	136	138	140	142
142	143	144	142	139	137	137	138	136	140	144	146	145	145	146	147

Setelah mendapatkan hasil dari 3 nilai RGB dan disatukan kembali dengan menggunakan matlab maka akan menghasilkan citra yang telah terenkripsi yaitu:



Gambar 4. Hasil Enkripsi Sampel.jpg

Proses dekripsi *cipher image*, dilakukanlah langkah-langkah berikut ini:

1. Setelah Setiap bagian pixel di enkripsi maka menuliskan kembali matriks R,G, dan B kemudian menyatukan kembali RGB sehingga akan menghasilkan citra yang telah terenkripsi. Untuk mendekripsi kembali citra maka dilakukan kembali pemisahan RGB dan dekripsi setiap pixelnya dengan menggunakan *script* berikut:

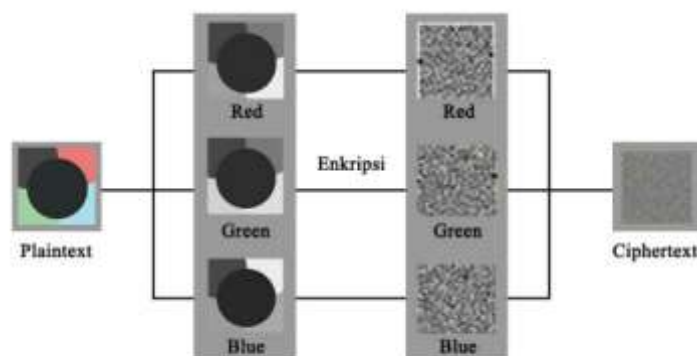
```
Private Sub blokDecrypt(ByRef in_blk() As blokData, ByRef out_blk() As blokData)
    Dim g, e, c, a, b, d, f, h As blokData
    #If BLOCK_REVERSE Then
        a = bswap(in_blk(3))
        b = bswap(in_blk(2))
        c = bswap(in_blk(1))
        d = bswap(in_blk(0))
    #Else
        MCopy (in_blk(0), a)
        MCopy (in_blk(1), b)
        MCopy (in_blk(2), c)
        MCopy (in_blk(3), d)
    #End If
    k_xor(32, a, b, c, d) : ib7(a, b, c, d, e, f, g, h) : k_xor(31, e, f, g, h)
    irot(e, f, g, h) : ib6(e, f, g, h, a, b, c, d) : k_xor(30, a, b, c, d)
    irot(a, b, c, d) : ib5(a, b, c, d, e, f, g, h) : k_xor(29, e, f, g, h)
    irot(e, f, g, h) : ib4(e, f, g, h, a, b, c, d) : k_xor(28, a, b, c, d)
    irot(a, b, c, d) : ib3(a, b, c, d, e, f, g, h) : k_xor(27, e, f, g, h)
    irot(e, f, g, h) : ib2(e, f, g, h, a, b, c, d) : k_xor(26, a, b, c, d)
    irot(a, b, c, d) : ib1(a, b, c, d, e, f, g, h) : k_xor(25, e, f, g, h)
    irot(e, f, g, h) : ib0(e, f, g, h, a, b, c, d) : k_xor(24, a, b, c, d)
    irot(a, b, c, d) : ib7(a, b, c, d, e, f, g, h) : k_xor(23, e, f, g, h)
    irot(e, f, g, h) : ib6(e, f, g, h, a, b, c, d) : k_xor(22, a, b, c, d)
    irot(a, b, c, d) : ib5(a, b, c, d, e, f, g, h) : k_xor(21, e, f, g, h)
    irot(e, f, g, h) : ib4(e, f, g, h, a, b, c, d) : k_xor(20, a, b, c, d)
    irot(a, b, c, d) : ib3(a, b, c, d, e, f, g, h) : k_xor(19, e, f, g, h)
    irot(e, f, g, h) : ib2(e, f, g, h, a, b, c, d) : k_xor(18, a, b, c, d)
    irot(a, b, c, d) : ib1(a, b, c, d, e, f, g, h) : k_xor(17, e, f, g, h)
    irot(e, f, g, h) : ib0(e, f, g, h, a, b, c, d) : k_xor(16, a, b, c, d)
    irot(a, b, c, d) : ib7(a, b, c, d, e, f, g, h) : k_xor(15, e, f, g, h)
    irot(e, f, g, h) : ib6(e, f, g, h, a, b, c, d) : k_xor(14, a, b, c, d)
    irot(a, b, c, d) : ib5(a, b, c, d, e, f, g, h) : k_xor(13, e, f, g, h)
    irot(e, f, g, h) : ib4(e, f, g, h, a, b, c, d) : k_xor(12, a, b, c, d)
    irot(a, b, c, d) : ib3(a, b, c, d, e, f, g, h) : k_xor(11, e, f, g, h)
    irot(e, f, g, h) : ib2(e, f, g, h, a, b, c, d) : k_xor(10, a, b, c, d)
    irot(a, b, c, d) : ib1(a, b, c, d, e, f, g, h) : k_xor(9, e, f, g, h)
```

```

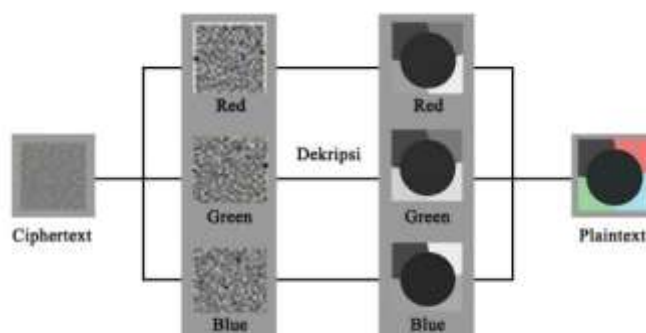
irot(e, f, g, h) : ib0(e, f, g, h, a, b, c, d) : k_xor(8, a, b, c, d)
irot(a, b, c, d) : ib7(a, b, c, d, e, f, g, h) : k_xor(7, e, f, g, h)
irot(e, f, g, h) : ib6(e, f, g, h, a, b, c, d) : k_xor(6, a, b, c, d)
irot(a, b, c, d) : ib5(a, b, c, d, e, f, g, h) : k_xor(5, e, f, g, h)
irot(e, f, g, h) : ib4(e, f, g, h, a, b, c, d) : k_xor(4, a, b, c, d)
irot(a, b, c, d) : ib3(a, b, c, d, e, f, g, h) : k_xor(3, e, f, g, h)
irot(e, f, g, h) : ib2(e, f, g, h, a, b, c, d) : k_xor(2, a, b, c, d)
irot(a, b, c, d) : ib1(a, b, c, d, e, f, g, h) : k_xor(1, e, f, g, h)
irot(e, f, g, h) : ib0(e, f, g, h, a, b, c, d) : k_xor(0, a, b, c, d)
#If BLOCK_REVERSE Then
  out_blk(3) = bswap(a)
  out_blk(2) = bswap(b)
  out_blk(1) = bswap(c)
  out_blk(0) = bswap(d)
#Else
  MCopy (a, out_blk(0))
  MCopy (b, out_blk(1))
  MCopy (c, out_blk(2))
  MCopy (d, out_blk(3))
#End If
End Sub

```

Ilustrasi dari penjelasan diatas dapat dilihat pada gambar 5 dan gambar 6.



Gambar 5. Ilustrasi Enkripsi




Gambar 6. Ilustrasi Dekripsi

3.2 Pengujian

Pengujian yang dilakukan dengan menyesuaikan langkah-langkah yang telah ditentukan, akan didapatkan hasil yang didasari oleh seberapa cepat enkripsi dengan menguji ukuran yang berbeda-beda. Hasil dapat dilihat pada tabel 1.

Tabel 4. Hasil Pengujian

No	Gambar	Ukuran	Waktu Enkripsi	Waktu Dekripsi
1	 Sampel1.jpg	32x32	1,2455413 detik	1,19451212 detik

2



144x144

3,524485 detik

3,25421542 detik

Sampel2.jpg

4. KESIMPULAN

Berdasarkan hasil pengujian yang telah dilakukan, maka didapatkan kesimpulan bahwa algoritma serpent dapat mengenkripsi citra digital dengan jenis RGB dan format jpg, sehingga lebih meningkatkan keamanan dari citra itu sendiri dan menekan kemungkinan diambil oleh pihak ketiga. Enkripsi ini sangat bergantung pada ukuran dari citra itu sendiri, semakin besar ukuran citra maka semakin lama pula waktu yang dibutuhkan untuk enkripsi. Tetapi hal yang menarik adalah waktu dekripsi memakan waktu lebih sedikit dibanding enkripsi, yaitu berbeda 0,1 – 0,2 detik pendekripsian.

REFERENCES

- [1] R. Munir, *Pengolahan Citra Digital*. Informatika Bandung, 2006.
- [2] D. Br Tarigan, K. Kunci, and K. Hill Cipher, "Implementasi Algoritma Kriptografi Hill Cipher Dalam Penyandian Data Gambar," no. 0911610, pp. 76–81, 2014.
- [3] A. D. Hidayat and I. Afrianto, "Sistem Kriptografi Citra Digital Pada Jaringan Intranet Menggunakan Metode Kombinasi Chaos Map dan Teknik Selektif," *ULTIMATICS*, vol. 9, no. 1, pp. 59–66, 2017.
- [4] T. Samuel Permana, C. Atika Sari, E. H. Rachmawanto, D. R. I. M. Setiadi, and E. R. Subhiyanto, "Implementasi Pengamanan Citra Digital Berbasis Metode Kriptografi Vernam Cipher," *Techno.COM*, vol. 16, no. 4, pp. 337–347, 2017.
- [5] B. Rizki R, R. Rumani M, and M. Nasrun, "Analisis Perbandingan Antara Algoritma Kriptografi Serpent dan AES pada Implementasi Enkripsi SMS di Perangkat Android," in *e-Proceeding of Engineering*, 2015, pp. 3511–3517.
- [6] T. Zebua, "Encoding the Record Database of Computer Based Test Exam Based on Spritz Algorithm," *Lontar Komput.*, vol. 9, no. 1, pp. 52–62, 2018.
- [7] A. D. Rambe and L. T. Sianturi, "PERANCANGAN APLIKASI KEAMANAN FILE CITRA DENGAN," vol. 12, no 1. September, pp. 322–327, 2017.
- [8] D. Putra, *Pengolahan Citra Digital*. Yogyakarta: Andi, 2010.
- [9] C. Iswahyudi, E. Setyaningsih, and N. Widyastuti, "Pengamanan Kunci Enkripsi Citra Pada Algoritma Super Enkripsi Menggunakan Metode End of File," *Pros. Semin. Nas. Apl. Sains Teknol. Periode III Yogyakarta*, no. November, p. B 278-B 284, 2012.