Jurnal Sistem Komputer dan Informatika (JSON)
Volume 1, Nomor 3, Mei 2020
e-ISSN 2685-998X
DOI 10.30865/json.v1i3.2138

Implementasi Pengamanan Citra Digital Menggunakan Algoritma ICE

Josua Alfiandi Sinuhaji

Program Studi Teknik Informatika, STMIK Budi Darma, Medan, Indonesia
Email: josuaalfiandi96@gmail.com

Abstrak–Citra Digital saat ini sangat banyak digunakan, sehingga sangat rentan terhadap pencurian data oleh pihak-pihak yang tidak berhak. Demi menjaga keamanan citra digital dapat dilakukan dengan pemanfaatan teknik kriptografi. Teknik kriptografi dapat menyandikan citra digital dengan mengenkripsikannya ke dalam bentuk sandi-sandi yang tidak dipahami. ICE adalah sebuah block cipher yang dipublikasikan oleh Kwan pada 1997. Algoritma ini memiliki struktur yang mirip dengan DES, tetapi dengan tambahan permutasi bit yang tidak tergantung kunci dalam fungsi putarannya. Terdapat berbagai jenis varian ICE, yaitu Thin-ICE, standar ICE, dan ICE-n. Perbedaan di antara ketiganya adalah panjang kata kunci yang digunakan dan jumlah putaran. Algoritma Thin-ICE menggunakan kunci 64 bits dan 8 putaran. Standar ICE menggunakan kunci 64 bits dan 16 putaran. Algoritma ICE-n menggunakan kunci 64n bits dan 16n putaran. Penggunaan jenis algoritma dapat disesuaikan dengan kebutuhan pengguna dimana Thin-ICE memiliki tingkat keamanan terendah di antara ketiganya, sedangkan ICE-n yang tertinggi. Algoritma ini tidak menjadi subyek paten dan source code dapat digunakan dengan bebas.

Kata Kunci: Kriptografi, Citra Digital,Algoritma, ICE.

Abstract–Digital imagery is currently very widely used, so it is very vulnerable to data theft by unauthorized parties. In order to maintain the security of digital images can be done by using cryptographic techniques. Cryptographic techniques can encode digital images by encrypting them in the form of passwords that are not understood. ICE is a block cipher published by Kwan in 1997. This algorithm has a structure similar to DES, but with additional bit permutations that do not depend on the key in the rotation function. There are various types of ICE variants, namely Thin-ICE, ICE standard, and ICE-n. The difference between the three is the length of the keyword used and the number of rounds. The Thin-ICE algorithm uses 64 bits and 8 turns. The ICE standard uses 64 bits and 16 round keys. The ICE-n algorithm uses keys 64n bits and 16n turns. The use of this type of algorithm can be adjusted to the needs of users where Thin-ICE has the lowest level of security among the three, while ICE-n is the highest. This algorithm does not become the subject of a patent and the source code can be used freely.

Keywords: Cryptography, Digital Image,Algorithm, ICE

1. PENDAHULUAN

Citra digital adalah gambar dua dimensi yang bisa ditampilkan pada layar komputer sebagai himpunan/diskrit nilai digital yang disebut pixel/picture elements. Dalam tinjauan matematis, citra merupakan fungsi kontinu dari intensitas cahaya pada bidang dua dimensi. Citra digital adalah citra f(x,y) dimana dilakukan diskritisasi koordinat sampling/spasial dan diskritisasi tingkat kwantisasi (kabuan/ kecemerlangannya). Citra digital merupakan fungsi intensitas cahaya f(x,y), dimana harga x dan harga y adalah koordinat spasial. Harga fungsi tersebut di setiap titik (x,y) merupakan tingkat kecemerlangan citra pada titik tersebut.

Keamanan merupakan salah satu aspek penting dalam pengiriman data maupun komunikasi melalui jaringan. Salah satu cara untuk menjaga keamanan dan kerahasiaan suatu data maupun informasi adalah dengan teknik enkripsi dan dekripsi. Teknik ini berguna untuk membuat pesan, data, maupun informasi tidak dapat dibaca atau dimengerti oleh orang lain, kecuali untuk penerima yang berhak dan mengetahui teknik dekripsinya. Teknik enkripsi dan dekripsi dikenal dan dipelajari dalam kriptografi.

Pada zaman sekarang ini, menjaga kerahasiaan informasi merupakan hal yang sangat penting. Setiap orang memiliki privasi masing-masing yang tidak ingin diketahui oleh orang lain, penyimpanan dokumen serta gambar-gambar penting adalah kewajiban yang mesti dilakukan. Penyalahgunaan data-data rahasia pribadi tersebut oleh pihak tertentu tentunya bisa saja menimbulkan kerugian yang sangat besar. Gambar yang bersifat rahasia tersebut perlu dibuatkan sistem penyimpanan dan pengunciannya agar tidak terbaca atau diubah oleh orang- orang yang tidak bertanggung jawab, baik saat data tersebut tersimpan sebagai file di dalam handphone maupun saat data tersebut dikirim melalui email. Untuk menyimpan data tersebut agar benar-benar aman, tentunya dilakukan sistem pengamanan yang baik, yang bebas dari jangkauan orang-orang yang tidak berhak, baik bebas dari jangkauan secara fisik maupun secara sistem. Maka dari itu demi menghindari masalah tersebut peneliti memberikan suatu solusi untuk pengamanan citra digital berkarakteristik jpeg dengan menerapkan metode algoritma ICE.

ICE adalah sebuah block cipher yang dipublikasikan oleh Kwan pada 1997. Algoritma ini memiliki struktur yang mirip dengan DES, tetapi dengan tambahan permutasi bit yang tidak tergantung kunci dalam fungsi putarannya. Terdapat berbagai jenis varian ICE, yaitu Thin-ICE, standar ICE, dan ICE-n. Perbedaan di antara ketiganya adalah panjang kata kunci yang digunakan dan jumlah putaran. Algoritma Thin-ICE menggunakan kunci 64 bits dan 8 putaran. Standar ICE menggunakan kunci 64 bits dan 16 putaran. Algoritma ICE-n menggunakan kunci 64n bits dan 16n putaran. Penggunaan jenis algoritma dapat disesuaikan dengan kebutuhan pengguna dimana Thin-ICE memiliki tingkat keamanan terendah di antara ketiganya, sedangkan ICE-n yang tertinggi. Algoritma ini tidak menjadi subyek paten dan source code dapat digunakan dengan bebas.

Submitted: **27/04/2020**; Accepted: **16/05/2020**; Published: **20/05/2020**

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 1, Nomor 3, Mei 2020
e-ISSN 2685-998X
DOI 10.30865/json.v1i3.2138

Penulis menerapkan algoritma ICE berdasarkan penelitian sebelumnya yang dilakukan oleh Ricky gilbert Fernando dengan judul "perbandingan algoritma kriptografi DES dengan ICE" menghasilkan bahwa algoritma ICE lebih cepat melakukan enkripsi dibandingkan DES dalam menyediakan tingkat keamanan yang sama[1]. Md Naziri, Siti Zarina & Meng Rong, Lim & Ismail, R.C. & Idris, Norina. (2015) "Development of Tiny Encryption Chip Design using Information Concealment Engine (ICE) Algorithm" menghasilkan bahwa matlab telah memberikan peluang besar dalam memahami algoritma ICE[2]. Maka dari itu peneliti memberikan kesimpulan dari kedua penelitan sebelumnya bahwa algoritma ICE bisa digunakan untuk mengamankan citra digital dengan enkripsi dan dekripsi yang cepat dan memiliki keamanan yang tinggi.

2. METODE PENELITIAN

2.1 Keamanan

Masalah keamanan merupakan salah satu aspek terpenting pada sebuah sistem informasi. Masalah keamanan sering kali kurang mendapatkan perhatian dari para perancang dan pengelola sistem informasi serta berada di urutan setelah tampilan, atau bahkan diurutan terakhir dalam daftar yang dianggap penting. Apabila menggangu performa sistem, sering kali masalah keamanan tidak begitu dipedulikan bahkan ditiadakan[3].

2.2 Citra Digital

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra terbagi 2 yaitu ada citra yang bersifat analog dan ada citra yang bersifat digital. Citra analog adalah citra yang bersifat kontinu seperti gambar pada monitor televisi, foto sinar X, hasil CT Scan dll. Sedangkan pada citra digital adalah citra yang dapat diolah oleh komputer. Sebuah citra digital dapat mewakili oleh sebuah matriks yang terdiri dari M kolom N baris, dimana perpotongan antara kolom dan baris disebut piksel (piksel = picture element), yaitu elemen terkecil dari sebuah citra. Piksel mempunyai dua parameter, yaitu koordinat dan intensitas atau warna. Nilai yang terdapat pada koordinat (x,y) adalah f(x,y), yaitu besar intensitas atau warna dari piksel di titik itu.Oleh sebab itu, sebuah citra digital dapat ditulis dalam bentuk matriks berikut[1].

2.3 Kriptografi

Kriptografi (cryptography) berasal dari bahasa yunani, yaitu dari kata cypto dan graphia yang berarti penulisan rahasia. Kriptografi adalah ilmu ataupun seni yang mempelajari bagaimana membuat suatu pesan yang dikirim oleh pengirim dapat disampaikan kepada penerima dengan aman. Kriptografi juga merupakan studi terhadap teknik matematis yang terkait dengan aspek keamanan suatu sistem informasi seperti kerahasiaan, integritas data, autentikasi dan ketiadaan penyangkalan[4].

2.4 Algoritma Information Concealment Engine (ICE)

ICE atau Information Concealment Engine adalah blok dari cryptosystem symmetric yang dirancang pada tahun 1995, dan pengarangnya adalah Mathew Kwan. ICE dirancang untuk menggantikan DES dalam aplikasi yang ada dan sebagai alternatif yang memungkinkan untuk DES di masa depan. Dengan tes benchmark dari penulis ICE lebih cepat dalam enkripsi dan dekripsi daripada DES, dan tidak seperti algoritma DES, algoritma ICE dapat digunakan secara bebas tanpa biaya dan dapat berguna di semua aplikasi di mana keamanan dan privasi diperlukan [1].

3. HASIL DAN PEMBAHASAN

Dalam hal ini data yang diamankan adalah sebuah citra dapat mewakili banyak informasi yang bersifat rahasia atau penting apalagi jika citra tersebut dari departemen tertentu yang sangat dilindungi ataupun dijaga kerahasiaannya, di karenakan citra tersebut mempunyai nilai yang berharga. Lemahnya pengetahuan tentang keamanan data gambar memudahkan para manipulative merekayasa image seperti manipulation, editing, duplicating, sehingga perlu dilakukan pengamanan agar para manipulative sulit untuk memanipulasi citra tersebut. Salah satu solusi yang dapat digunakan untuk menangani permasalahan tersebut adalah dengan teknik kriptografi, dengan menggunakan algoritma ICE.

Alasan dilakukannya algoritma ICE adalah untuk meningkatkan aspek keamanan suatu informasi khususnya citra. Agar citra yang dikirim aman dari orang yang tidak bertanggung jawab, citra tersebut harus disandikan menggunakan algoritma ICE. Nama lain untuk citra disebut juga data gambar yang tersimpan berupa jpeg. Bentuk gambar yang tersandi disebut cipher gambar atau kriptogram. Cipher gambar harus dapat ditransformasikan kembali menjadi citra semula agar gambar yang diterima bisa dilihat. Algoritma kriptografi yang digunakan disebut juga cipher gambar yaitu suatu bentuk aturan untuk enciphering dan deciphering, atau fungsi matematika yang digunakan untuk proses enkripsi dan dekripsi.

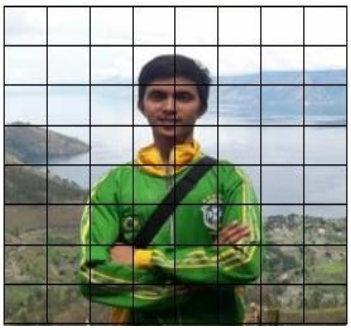
Penelitian ini menguraikan proses-proses untuk algoritma ICE tersebut agar didapatkan cipher gambar yang sulit dipecahkan. Proses yang dilakukan adalah citra asli (plain gambar) terlebih dahulu akan dienkripsi

Submitted: **27/04/2020**; Accepted: **16/05/2020**; Published: **20/05/2020**

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 1, Nomor 3, Mei 2020
e-ISSN 2685-998X
DOI 10.30865/json.v1i3.2138

berdasarkan algoritma ICE sehingga menghasilkan cipher gambar, kemudian cipher gambar didekripsi kembali berdasarkan algoritma ICE sehingga akan menghasilkan Plain gambar.

Proses yang dilakukan adalah citra asli (*plain* gambar) terlebih dahulu akan dienkripsi berdasarkan algoritma *ICE* sehingga menghasilkan *cipher* gambar , kemudian *cipher* gambar pertama didekripsi kembali berdasarkan algoritma *ICE* sehingga akan menghasilkan *plain* gambar. Sebagai contoh citra 24 bit berformat *jpeg* dengan resolusi 8x8 piksel, memperlihatkan matriks pada nilai-nilai piksel setiap baris dan kolom adalah sebagai berikut:



Gambar 1. Citra Sampel

Pada sampel contoh kasus ini dicoba dilakukan enkripsi dan dekripsi terhadap citra dengan ukuran 1x8 :

R	232	221	209	189	162	141	126	123
G	247	234	219	193	163	140	124	121
B	216	206	195	176	155	136	127	126

Nilai matriks tersebut diatas menjadi *plain* gambar pada proses enkripsi ICE :
Asumsikan kunci :
Kunci : **JOSHUA**
Kunci Algoritma ICE kedalam urutan bilangan HEX dan BINER pada ASCII :

CHAR	Dec	BINER
J	74	01001010
O	79	01001111
S	83	01010011
H	72	01001000
U	85	01010101
A	65	01000001

Kelompokkan setiap bit menjadi 3 blok :

23 = 11011111
56 = 00111000
11 = 01101111
26 = 11100010 — Blok 1
13 = 00001011
13 = 01110001
26 = 11001110
18 = 10110100

14 = 10001100
2 = 00000010 — Blok 2
94 = 01011110
25 = 11010111

19 = 10010101
17 = 10101100 — Blok 2
24 = 11110100
18 = 10110010

15 = 10011011
63 = 00111111
89 = 01011001
43 = 00101011 — Blok 3
12 = 01110000
20 = 00010100
30 =11100110
89 = 01011001

Blok *cipher*-asli dipermutasi dengan matrik permutasi awal (*initial permutation* atau IP). Bisa ditulis *x0 = IP (x) = L0 R0*, dimana *L*0 terdiri dari 32 bit pertama dari x0 dan 32 bit terakhir dari *R*0.

IP(x) : 11011111 00111000 01101111 11100010 00001011 01110001 11001110 10110100
Pecah bit pada IP(x) menjadi 2 bagian yaitu:

L_0 : 11011111 00111000 01101111 11100010
R_0: 00001011 01110001 11001110 10110100
Lakukan pergeseran kiri (Left Shift) pada L0 dan R0, sebanyak 1 atau 2 kali berdasarkan kali putaran yang ada pada tabel putaran sebagai berikut :

Submitted: **27/04/2020**; Accepted: **16/05/2020**; Published: **20/05/2020**

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 1, Nomor 3, Mei 2020
e-ISSN 2685-998X
DOI 10.30865/json.v1i3.2138

Tabel 1. *Left Shift*

Putaran, *i*	Jumlah Pergeseran Bit
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

Untuk putaran ke 1 sampai dengan 16 dilakukan pegeseran 1 bit ke kiri
Berikut hasil outputnya:
L0 : 11011111 00111000 01101111 11100010
R0 : 00001011 01110001 11001110 10110100
Digeser 1 bit ke kiri
L1 : 011011111 00111000 01101111 1110001
R1 : 000001011 01110001 11001110 1011010
Digeser 2 bit ke kiri

Hasil putaran terakhir digabungkan kembali menjadi L_iR_i dan di XOR kan dengan kunci yang telah di tentukan :
Berikut hasil prosesnya:
Proses iterasi 1 adalah melakukan XOR antara kunci $L_{16}R_{16}$ dan K1
Iterasi - 1
E(R(1)) = 11111001 11000011 0111111 10001011 00010110 11100011 10011100 11010000
K1 = 01011001 01010101 01001100 01001001 01000001 01001110 01010100 01001001
---⊕
A1 = 10100000 10010110 00110011 11000011 01010111 10101101 11001000 10011001

Lakukan hal yang sama sampai proses iterasi ke-16

Setiap Vektor A_i disubstitusikan kedelapan buah S-Box(*Substitution Box*), tetapi pada proses ini hanya di butuhkan 6 bit untuk setiap SBox, maka 2 bit terakhir akan diabaikan, 2 bit pertama baris dan empat bit kolom.
Contoh :
Biner : 10011000
Baris: 10 | 0 1 1 0 | 00
Kolom
Berikut cara pembacaan contoh di atas :
Dua bit pertama berupa biner yaitu 10 dirubah kedesimal yaitu 2 yang akan dijadikan alamat baris, selanjutnya empat bit biner berikutnya yaitu 0110 dirubah kedesimal yaitu 6 yang akan dijadikan alamat kolom dan dua bit berikutnya akan diabaikan maka dapat dilihat pada table *S-Boxes* dibawah ini :

Tabel 2. Pendefenisian *S-Boxes* dari Algoritma ICE Column

S1 :

Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S2 :

Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	15	4	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

Submitted: **27/04/2020**; Accepted: **16/05/2020**; Published: **20/05/2020**

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 1, Nomor 3, Mei 2020
e-ISSN 2685-998X
DOI 10.30865/json.v1i3.2138

S3:

Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1	13	7	0	9	3	4	6	10	2	81	5	14	12	11	15	1
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S4 :

Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S5:

Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S6 :

Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
2	9	14	15	5	2	8	12	2	7	0	4	10	1	13	11	6
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S7 :

Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S8:

Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	1	15	13	8	10	3	7	4	12	5	5	11	0	14	9	2
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
3	2	1	14	7	4	10	81	3	15	12	9	0	3	5	6	11

Berdasarkan hasil dari *S-box* diatas dikonversikan ke biner dengan jumlah 8 bit sebagai berikut:
2=00000010 4=00000100 13=00001101 1=00000001 11=00001011 12=00001100 4= 00000100 15=00001111

Tabel 3. Hasil Pengambilan Nilai *S-box*

S-BOX	HASIL	
	DECIMAL	BINER
1	2	00000010
2	4	00000100
3	13	00001101
4	1	00000001
5	11	00001011
6	12	00001100
7	4	00000100
8	15	00001111

Kemudian biner hasil nilai *S-box* digabungkan menjadi L0R0 berikut pembentukkannya :
L0R0 : 00000010 00000100 00001101 00000001 00001011 00001100 00000100 00001111
Setelah 64 bit nilai biner digabungkan maka nilai biner di bagi menjadi dua bagian yaitu 32 bit L0 dan 32 bit R0, berikut pembentukkannya :
Kemudian hasil nilai *S-box dibagi* menjadi dua bagian yaitu 32 bit L0 dan 32 bit R0, berikut pembentukkannya :
L0= 00000010 00000100 00001101 00000001
R0= 00001011 00001100 00000100 00001111
Dikarenakan *ICE* mempunyai 16 putaran maka dibutuhkan kunci internal sebanyak 16 buah. Kunci Internal dibangkitkan sebelum proses enkripsi atau bersamaan dengan proses enkripsi berikut :

Submitted: **27/04/2020**; Accepted: **16/05/2020**; Published: **20/05/2020**

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 1, Nomor 3, Mei 2020
e-ISSN 2685-998X
DOI 10.30865/json.v1i3.2138

Tabel 4. PC-1

PC-1																
O	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
I	57	49	41	33	25	17	9	1	58	50	42	34	26	18	10	2
O	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
I	59	51	43	35	27	19	11	3	60	52	44	36	63	55	47	16
O	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
I	31	23	15	7	62	54	47	38	30	22	14	6	61	53	45	3
O	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
I	29	21	13	5	28	20	12	4	3	23	2	12	4	24	22	16

Berdasarkan matriks permutasi di atas maka telah ditentukan tabel shift menurut jumlah bit yang akan diproses yaitu 64 bit maka proses pergeseran bit menjadi 16 kali pergeseran :

Tabel 5. Jumlah Pergeseran Bit Pada Setiap Putaran

O	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
I	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	2

Langkah selanjutnya melakukan pergeseran dari kanan ke kiri sesuai tabel sebelum nya dengan memecah bit menjadi 2 bagian:
L0=00000010 00000100 00001101 00000001
R0=00001011 00001100 00000100 00001111
L1=100000010 00000100 00001101 0000000
R1=100001011 00001100 00000100 0000111
Sampai
R16=00000001 00000000 00001000 00000011

Langkah terakhir adalah menggabungkan R_{16} dengan L_{16} Sehingga Input :
$R_{16}L_{16}$ = 00000111 00000101 00001110 00001001 00000001 00000000 00001000 00000011
Menghasilkan Output:
Cipher (dalam biner) = 10011000=**152** 00011110=**30** 10000001=**129** 01110010=**114** 10001101=**141** 00000010=**3** 11010111=**215** 00010110=**22**
Akhirnya, keempat *sub-block*tersebut digabungkan kembali sehingga menghasilkan nilai RGB yang sudah disandi rahasiakan atau disebut dengan *cipher.*

Matriks Cipher Nilai RGB :

R	152	114	215
G	30	141	22
B	129	3	

Untuk mencari cipher blok 2 dan 3 dilakukan dengan cara yang sama seperti proses blok 1.
Hasil akhir blok 2 :

R	97	90	101
G	1	111	76
B	33	95	

Hasil akhir blok 3 :

R	75	15	110
G	16	30	20
B	30	8	

Akhirnya ketiga *sub-block* tersebut digabungkan kembali sehingga menghasilkan nilai RGB *Cipher* gambar dari dekripsi *ICE* :

R	152	114	215	1	111	76	30	8
G	30	141	22	33	95	75	15	110
B	129	3	97	90	101	16	30	20

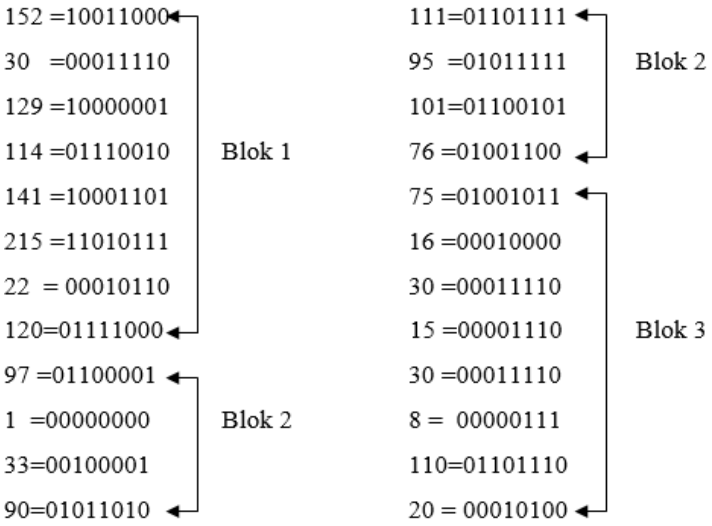
Proses Dekripsi Cipher gambar
Pada nilai RGB lakukan perulangan dari biner *cipher* ke Initial *Permutation* (IP) menggunakan tabel IP berikut :
Nilai ASCII Cipher gambar adalah :

R	152	114	215	1	111	76	30	8
G	30	141	22	33	95	75	15	110
B	129	3	97	90	101	16	30	20

Kelompokkan setiap bit menjadi 3 blok :

Submitted: **27/04/2020**; Accepted: **16/05/2020**; Published: **20/05/2020**

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 1, Nomor 3, Mei 2020
e-ISSN 2685-998X
DOI 10.30865/json.v1i3.2138



Blok pertama :

R	152	114	215
G	30	141	22
B	129	3	

Blok pertama *cipher gambar* dipermutasi dengan matrik permutasi awal (*initial permutation* atau IP). Bisa ditulis *x0 = IP (x) = L0 R0*, dimana *L*0 terdiri dari 32 bit pertama dari x0 dan 32 bit terakhir dari *R*0.
IP(x) : 01000011 00001010 00101110 00110010 00000010 00011011 01010100 01111000
Pecah bit pada IP(x) menjadi 2 bagian yaitu:
L_0 : 10011000 00011110 10000001 01110010
R_0: 10001101 00000010 11010111 00010110

Lakukan pergeseran kanan (*right Shift*) pada L0 dan R0, sebanyak 1 atau 2 kali berdasarkan kali putaran yang ada pada tabel putaran sebagai berikut:

Tabel 6. *Left Shift*

Putaran, *i*	Jumlah Pergeseran Bit
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

Untuk putaran ke 1 sampai dengan 16 dilakukan pegeseran 1 bit ke kanan
Berikut hasil outputnya:
L_0 : 10011000 00011110 10000001 01110010
R_0: 10001101 00000010 11010111 00010110
Digeser 1 bit ke kanan
L1 : 0011000 00011110 10000001 011100101
R1 : 0001101 00000010 11010111 000101101
Digeser 2 bit ke kanan
L_2 : 11000000 11110100 00001011 10010100
R_2 : 01101 00000010 11010111 00010110100

Submitted: **27/04/2020**; Accepted: **16/05/2020**; Published: **20/05/2020**

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 1, Nomor 3, Mei 2020
e-ISSN 2685-998X
DOI 10.30865/json.v1i3.2138

Hasil putaran terakhir digabungkan kembali menjadi L_iR_i dan di XOR kan dengan kunci yang telah di tentukan :

Berikut hasil outputnya:

Iterasi - 1 = 01010011 00000011 11010000 00101110 11010001 10100000 01011010 11100010

K1 = 10001100 00001010 10011101 01110010 10001101 00011000 11001010 00010101

----------⊕

A1 = 11011111 00001001 01001101 01011100 01011100 10111000 10010000 11110111

Lakukan hal yang sama sampai proses iterasi ke-16

Lanjutkan dengan menggunakan kunci yang kedua (K2) :

Iterasi - 1

E(R(1) = 10001100 00001010 10011111 01110010 10001101 00011000 11001010 00010101

K2 = 10000110 00001010 10000001 01111010 10010000 00011011 11011011 00010011

----------⊕

A2 = 11011111 01011111 11001101 00110011 11011001 01010101 10001111 01011011

Lakukan hal yang sama sampai proses iterasi ke-16

Lanjutkan dengan menggunakan kunci yang ketiga (K3) :

Iterasi - 1

E(R(1) = 10001100 00001010 10011111 01110010 10001101 00011000 11001010 00010101

K3 = 10011000 00011110 100000001 01110010 10001101 00000010 11010111 00010110

----------⊕

A3 = 11000001 01001011 11001101 00111011 11001100 01001100 10000011 01011110

Lakukan hal yang sama sampai proses iterasi ke-16

Setiap Vektor A_i disubstitusikan kedelapan buah S-Box(*Substitution Box*), Tetapi pada proses ini hanya di butuhkan 6 bit untuk setiap sbox, maka 2 bit terakhir akan diabaikan.

Tabel 7. Pendefenisian *S-Boxes* dari Algoritma ICE Column

S1 :

Row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S2 :

row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	15	4	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S3:

row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1	13	7	0	9	3	4	6	10	2	81	5	14	12	11	15	1
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S4 :

row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S5:

row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S6 :

row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
2	9	14	15	5	2	8	12	2	7	0	4	10	1	13	11	6
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

Submitted: **27/04/2020**; Accepted: **16/05/2020**; Published: **20/05/2020**

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 1, Nomor 3, Mei 2020
e-ISSN 2685-998X
DOI 10.30865/json.v1i3.2138

S7 :

row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S8:

row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	1	15	13	8	10	3	7	4	12	5	5	11	0	14	9	2
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
3	2	1	14	7	4	10	81	3	15	12	9	0	3	5	6	11

Berdasarkan S-box diatas diperoleh hasil sebagai berikut:
7=00000111 5=00000101 14=00001110 9=00001001 1=00000001 0=00000000 8= 00001000 3=00000011

Tabel 8. Hasil Pengambilan Nilai *S-box*

S-BOX	HASIL	
	DECIMAL	BINER
1	7	00000111
2	5	00000101
3	14	00001110
4	9	00001001
5	1	00000001
6	0	00000000
7	8	00001000
8	3	00000011

Kemudian biner hasil nilai *S-box* digabungkan menjadi L0R0 berikut pembentukkannya :
L0R0 : 00000111 00000101 00001110 00001001 00000001 00000000 00001000 00000011
Setelah 64 bit nilai biner digabungkan maka nilai biner di bagi menjadi dua bagian yaitu 32 bit L0 dan 32 bit R0, berikut pembentukkannya :
L0= 00000111 00000101 00001110 00001001
R0= 00000001 00000000 00001000 00000011
Langkah selanjutnya adalah menentukan matriks permutasi untuk menentukan jumlah pergeseran bit yang akan diproses, berikut tabel permutasinya :

Tabel 9. PC-1

	(PC-1)															
O	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
I	57	49	41	33	25	17	9	1	58	50	42	34	26	18	10	2
O	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
I	59	51	43	35	27	19	11	3	60	52	44	36	63	55	47	39
O	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
I	31	23	15	7	62	54	47	38	30	22	14	6	61	53	45	3
O	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
I	29	21	13	5	28	20	12	4	3	23	2	12	4	24	22	16

Berdasarkan matriks permutasi di atas maka telah ditentukan tabel shift menurut jumlah bit yang akan diproses yaitu 64 bit maka proses pergeseran bit menjadi 16 kali pergeseran

Tabel 10. Jumlah Pergeseran Bit Pada Setiap Putaran

O	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
I	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	2

Langkah selanjutnya melakukan pergeseran kiri ke kanan sesuai tabel sebelum nya dengan memecah bit menjadi 2 bagian:
L0=00000111 00000101 00001110 00001001
R0=00000001 00000000 00001000 00000011
Langkah terakhir adalah menggabungkan R_{16} dengan L_{16} Sehingga Input :
$R_{16}L_{16}$ = 00000111 00000101 00001110 00001001 00000001 00000000 00001000 00000011

Submitted: **27/04/2020**; Accepted: **16/05/2020**; Published: **20/05/2020**

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 1, Nomor 3, Mei 2020
e-ISSN 2685-998X
DOI 10.30865/json.v1i3.2138

Menghasilkan Output:
Cipher (dalam biner) = 11011111=**223** 11011101=**221** 11010001=**209** 11110111 =**247** 11101010=**234** 10001000 =**136** 1111111 =**127** 1111110=**126**
Akhirnya, keempat *sub-block* tersebut digabungkan kembali sehingga menghasilkan nilai RGB yang sudah disandi rahasiakan atau disebut dengan *cipher*.
Matriks Cipher Nilai RGB :

R	223	221	209
G	247	234	219
B	216	206	

Untuk mencari cipher blok 2 dan 3 dilakukan dengan cara yang sama seperti proses blok 1
Hasil akhir blok 2 :

R	189	162	141
G	193	163	140
B	195	176	155

Hasil akhir blok 3:

R		126	123
G	124	121	
B	136	127	126

Akhirnya ketiga *sub-block* tersebut digabungkan kembali sehingga menghasilkan nilai RGB *Cipher gambar* dari dekripsi *ICE* :
Maka *Plain* gambar dari citra berwarna yang memiliki resolusi 1 x 8 sebagai berikut:

R	232	221	209	189	162	141	126	123
G	247	234	219	193	163	140	124	121
B	216	206	195	176	155	136	127	126

4. KESIMPULAN

Berdasarkan hasil studi literatur, analisis, perancangan, implementasi, dan pengujian aplikasi ini, maka kesimpulan yang didapat adalah sebagai berikut :

1. Dalam mengimplementasikan pengamanan citra berbasis eclipse setelah melakukan proses enkripsi menghasilkan chiper pada citra dan setelah dilakukan proses dekripsi kembali menghasilkan plain citra atau citra awal.
2. Penerapan algoritma ICE sebagai teknik pengamanan pada citra dengan menggunakan kunci algoritma ICE-16 yang menjalankan 16n putaran dan memiliki 16n sub-kunci dan panjang kunci 64n di mana n adalah angka alam yang lebih besar dari satu.
3. Perancangan aplikasi pengamanan citra menggunakan bahasa pemograman android pada sofware Eclipse.

REFERENCES

[1] B. Van Rompay, L. R. Knudsen, and V. Rijmen, “Differential Cryptanalysis of the ICE Encryption Algorithm,” pp. 270–283, 1998.
[2] C. Meyer, M. Matyas, and R. Adler, “Perbandingan algoritma kriptografi des dengan ice,” 1997.
[3] Y. Aprianto, “CITRA DIGITAL MENGGUNAKAN ALGORITMA RIJNDAEL,” pp. 1–11.
[4] P. Hasil, P. Skripsi, J. T. Elektro, F. Teknik, U. Brawijaya, and P. Studi, “IMPLEMENTASI ALGORITMA KRIPTOGRAFI ELGAMAL UNTUK FILE CITRA 2 DIMENSI.”
[5] D. Pemrograman and V. Basic, “Modul 1 , Pengenalan Visual Basic Memilih jenis Project.”

Submitted: **27/04/2020**; Accepted: **16/05/2020**; Published: **20/05/2020**