Terbit *online* pada laman: https://journal.fkpt.org/index.php/BIT

BULLETIN OF INFORMATION TECHNOLOGY (BIT)

ISSN (Media Online) 2722-0524

IMPLEMENTASI ALGORITMA TRIANGLE CHAIN CIPHER DAN GOST PADA PENGAMANAN CITRA DIGITAL

Vivi Irda Anggraini
Program Studi Teknik Informatika, Universitas Budi Darma, Sumatera Utara, Indonesia
Email: viviirda28@gmail.com

INFORMASI ARTIKEL

Sejarah Artikel:
Diterima Redaksi : 10 Oktober 2021
Revisi Akhir : 1 November 2021
Diterima : 20 November 2021
Diterbitkan *Online* : 28 November 2021

KATA KUNCI

Kata Kunci: *Citra Digital, Kriptografi, Triangle Chain Cipher, GOST*

KORESPONDENSI

E-mail: viviirda28@gmail.com

A B S T R A C T

Keamanan citra digital penting untuk dijaga agar informasi tersebut tetap utuh dan terjamin. 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. Algoritma Triangle Chain Cipher dan GOST adalah salah satu algoritma yang dapat diandalkan dalam mewujudkan teknik kriptografi. Kombinasi kedua algoritma ini akan menghasilkan tingkat keamanan yang lebih tinggi terhadap pengamanan citra digital karena dapat menyandikannya ke bentuk sandi dengan proses yang cukup rumit sehingga akan mempersulit kriptanalis untuk mengaksesnya. Penelitian ini akan menguraikan proses enkripsi berdasarkan triangle chain cipher kemudian di enkripsi lagi berdasarkan GOST. Hal ini dilakukan sebagai upaya untuk meminimalisir tindakan-tindakan penyalahgunaan citra digital .

Abstract

The security of digital images is important to maintain so that the information remains intact and guaranteed. The security of digital images can be done by utilizing cryptographic techniques. Cryptographic techniques can encode digital images by encrypting them in the form of passwords that are not understood. The Triangle Chain Cipher and GOST Algorithms are one of the algorithms that can be relied upon in realizing cryptographic techniques. The combination of these two algorithms will result in a higher level of security against digital image security because it can encode it into a password with a fairly complex process that will make it difficult for cryptanalysts to access it. This research will describe the encryption process based on the triangle chain cipher and then encrypted again based on GOST. This is done as an effort to minimize acts of abuse of digital images.

1. PENDAHULUAN

Citra digital merupakan gambaran atau kemiripan dari suatu objek yang dapat diolah komputer. Umumnya citra memiliki data yang cukup besar sehingga membutuhkan daya komputasi yang besar pula. Citra digital memiliki berbagai informasi yang dapat menjadi penting apabila didalamnya terdapat informasi berharga. Foto pribadi dan dokumen penting lainnya merupakan salah satu citra digital yang bersifat rahasia yang apabila diakses pihak yang tidak bertanggung jawab tentunya akan berbahaya dan merugikan pemiliknya [1]. Salah satu teknik yang dapat digunakan untuk mengamankan suatu citra digital adalah menggunakan teknik kriptografi.

Kriptografi merupakan ilmu yang mempelajari tentang teknik matematika yang berhubungan dengan kerahasiaan, keamanan data, dan integritas data. Pentingnya kriptografi untuk keamanan data berkaitan dengan penggunaan komputer. Menjaga kerahasiaan, integritas data, otentikasi, dan nirpenyangkalan merupakan tujuan kriptografi [2].

Metode kriptografi yang akan digunakan untuk mengamankan ctra digital adalah algoritma Triangle Chain Cipher dan GOST, dimana proses algoritma GOST menggunakan 64 bit block cipher dan 256 bit kunci, sampai saat ini

kelemahan GOST dikarenakan key schedule yang sederhana [3], sehingga penulis mengkombinasikan dengan algoritma triangle chain cipher agar mengoptimalkan keamanan citra digital. Proses enkripsi dan dekripsi pada algoritma triangle chain cipher dilakukan secara berantai berdasarkan perkalian nilai kunci dan factor pengali yang dilakukan secara ganda sehingga hasil yang didapat jauh berbeda dengan pesan asli.

2. METODOLOGI PENELITIAN

2.1 Aplikasi Algoritma Crochemore Perrin

Algoritma Crochemore-Perrin yang sering juga disebut dengan algoritma Two Way Algorithm, atau Algoritma Dua Arah dipublikasikan Maxime Crochemore dan Dominique Perrin pada tahun 1991. Algoritma ini memfaktorkan pattern menjadi dua bagian patternkanan dan patternkiri. Fase pencocokan pada algoritma ini tediri dari dua bagian, pertama pencocokan karakter patternkanan dari kiri kekanan lalu pencocokan karakter patternkiri dari kanan kekiri.

2.2 String Matching

String Matching adalah proses pencarian semua kemunculan query yang selanjutnya disebut pattern ke dalam string yang lebih panjang (teks). Pattern dilambangkan dengan x=x[0..m-1] dan panjangnya adalah m. Teks dilambangkan dengan y=y[0..n-1] dan panjangnya adalah n.

2.3 Aplikasi

Aplikasi itu sebenernya adalah program, tetapi berbeda dari titi pandang. Program adalah istilah yang biasa dipakai oleh pembuat program, sedangkan aplikasi adalah istilah dari sisi pemakai (user). Program adalah yang ditulis oleh pemogram, sedangkan aplikasi adalah hasil terjemahan program, berupa kode yang di pahami oleh mesin [6]..

3. ANALISA DAN PEMBAHASAN

3.1 Analisa

Citra matematis adalah citra berbentuk abstrak yang di dalamnya terdapat fungsi kontinu dan fungsi diskrit. Citra berfungsi diskrit atau citra digital inilah yang dapat diolah komputer. Sebuah matriks yang terdiri dari kolom dan baris dapat mewakili sebuah citra digital yang dimana perpotongan antara kolom dan baris disebut piksel [9].

3.2 Analisa

Penerapan pengkombinasiaan algoritma *triangle chain cipher* dan GOST diimplementasi pada citra *grayscale* berukuran 2x4 yang telah dipilih sebagai contoh. Langkah-langkah yang dilakukan pada penerapan kombinasi kedua algoritma ini mengikuti aturan yang ditetapkan pada masing-masing algoritma. Sebelum proses enkripsi dilakukan, terlebih dahulu ubah citra warna menjadi *grayscale* menggunakan *software photoshop*, selanjutnya ambil nilai piksel pada setiap elemen piksel menggunakan *software* matlab. Kemudian lakukan pembacaan nilai kunci dan pembentukan tabel faktor pengali. Selanjutnya enkripsi citra digital rahasia (*plainimage*) berdasakan algoritma *triangle chain cipher*, kemudian binerkan *cipherimage* hasil dari pengamanan algoritma *triangle chain cipher* dan enkripsi kembali menggunakan algoritma GOST.

Proses enkripsi pada algoritma *triangle chain cipher* dilakukan secara berantai berdasarkan perkalian nilai kunci dan faktor pengali yang dilakukan melalui dua tahapan yaitu segitiga pertama dan segitiga kedua. Sedangkan proses enkripsi pada algoritma GOST dilakukan perblok. Proses enkripsi dan dekripsi dilakukan sebanyak 32 putaran yang menggunakan 64 bit blok cipher dan 256 bit kunci, serta menggunakan 8 buah tabel S-BOX

Contoh Kasus

Berikut ini diuraikan penerapan algoritma *triangle chain cipher* dan algoritma GOST dalam mengamankan sebuah citra *grayscale* ber*ekstensi* .jpg dengan resolusi 186 x 138.

Gambar 3.1 *Plainimage* Resolusi 138 x 184

Berdasarkan *plainimage* diatas, maka akan diambil delapan piksel sebagai contoh perhitungan manual. Delapan piksel tersebut akan diambil nilai desimal pada setiap piksel.

BULLETIN OF INFORMATION TECHNOLOGY (BIT)
Volume 2, No. 3, Nopember 2021, pp 119- 128
ISSN 2722-0524 (media online)

Gambar 3.2 *Plainimage* Contoh Sebanyak 8 Piksel

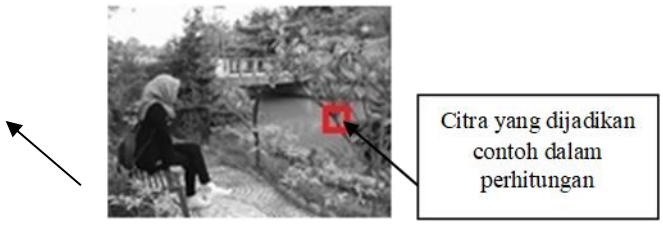
Nilai piksel dari delapan piksel *plainimage* contoh diatas di ambil dengan menggunakan *software* matlab, sehingga diperoleh:

Tabel 3.1 Piksel Citra Contoh

Piksel	
50	126
99	103
150	53
146	99

Berdasarkan tabel 3.1 diatas, maka dapat dilihat nilai desimal *plainimage* adalah 50, 126, 99, 103, 150, 53, 146, 99

1. Proses enkripsi berdasarkan algoritma *triangle chain cipher* dan GOST



Gambar 3.3 Contoh Citra *Grayscale*

a. Proses enkripsi algoritma *triangle chain cipher*
Proses enkripsi algoritma *triangle chain cipher* dilakukan dengan dua tahap yaitu enkripsi segitiga pertama dan enkripsi segitiga kedua yang menghasilkan cipher akhir. Berikut ini uraian proses enkripsi :
1) Matrix enkripsi segitiga pertama
Plainimage = 50, 126, 99, 103, 150, 53, 146, 99
N = 8
K = 3
R = 1,2,3,4,5,6,7,8
Untuk baris pertama (i = 1), maka :

$M_{1,1}$ = (P[1] + 3 * R[1] mod 256
= (50 + 3* (1)) mod 256
= (50 + 3) mod 256
= 53 mod 256
= 53

$M_{1,2}$ = (P[2] + 3 * R[1] mod 256
= (126 + 3 * (1)) mod 256
= (126 + 3) mod 256
= 129 mod 256
= 129

$M_{1,3}$ = (P[3] + 3 * R[1] mod 256
= (99 + 3 * (1)) mod 256
= (99 + 3) mod 256
= 102 mod 256
= 102

$M_{1,4}$ = (P[4] + 3 * R[1] mod 256
= (103 + 3 * (1)) mod 256
= (103 + 3) mod 256
= 106 mod 256
= 106

Volume 2, No. 3, Nopember 2021, pp 119- 128
ISSN 2722-0524 (media online)

$M_{1,5}$ = (P[5] + 3 * R[1] mod 256
= (150 + 3 * (1)) mod 256
= (150 + 3) mod 256
= 153 mod 256
= 153

$M_{1,6}$ = (P[6] + 3 * R[1] mod 256
= (53 + 3 * (1)) mod 256
= (53 + 3) mod 256
= 56 mod 256
= 56

$M_{1,7}$ = (P[7] + 3 * R[1] mod 256
= (146 + 3 * (1)) mod 256
= (146 + 3) mod 256
= 149 mod 256
= 149

$M_{1,8}$ = (P[8] + 3 * R[1] mod 256
= (99 + 3 * (1)) mod 256
= (99 + 3) mod 256
= 102 mod 256
= 102

Hasil dari enkripsi baris pertama adalah :

50, 126, 99, 103, 150, 53, 146, 99 → i = 0
53, 129, 102, 106, 153, 56, 149, 102 → i = 1

Hasil enkripsi dari baris pertama (i = 1) akan digunakan sebagai *plainimage* pada proses kedua (i = 2), dimana j ≥ i, sehingga : i = 2, j = 2

$M_{2,2}$ = (P[2] + 3 * R[2] mod 256
= (129 + 3 * (2)) mod 256
= (129 + 6) mod 256
= 135 mod 256
= 135

$M_{2,3}$ = (P[3] + 3 * R[2] mod 256
= (102 + 3 * (2)) mod 256
= (102 + 6) mod 256
= 108 mod 256
= 108

$M_{2,4}$ = (P[4] + 3 * R[2] mod 256
= (106 + 3 * (2)) mod 256
= (106 + 6) mod 256
= 112 mod 256
= 112

$M_{2,5}$ = (P[5] + 3 * R[2] mod 256
= (153 + 3 * (2)) mod 256
= (153 + 6) mod 256
= 159 mod 256
= 159

$M_{2,6}$ = (P[6] + 3 * R[2] mod 256
= (56 + 3 * (2)) mod 256
= (56 + 6) mod 256
= 62 mod 256
= 62

$M_{2,7}$ = (P[7] + 3 * R[2] mod 256
= (149 + 3 * (2)) mod 256
= (149 + 6) mod 256
= 155 mod 256
= 155

$M_{2,8}$ = (P[8] + 3 * R[2] mod 256
= (102 + 3 * (2)) mod 256
= (102 + 6) mod 256
= 108 mod 256
= 108

Hasil dari enkripsi baris kedua adalah :

50, 126, 99, 103, 150, 53, 146, 99 → i = 0

BULLETIN OF INFORMATION TECHNOLOGY (BIT)
Volume 2, No. 3, Nopember 2021, pp 119- 128
ISSN 2722-0524 (media online)

53, 129, 102, 106, 153, 56, 149, 102 → i = 1
135, 108, 112, 159, 62, 155, 108 → i = 2

Baris selanjutnya dapat dicari dengan cara yang sama seperti diatas, sehingga di dapatkan *cipher* sebagai berikut :

Tabel 3.2 Hasil Enkripsi Segitiga Pertama

$M_{i\backslash j}$	1	2	3	4	5	6	7	8
0	50	126	99	103	150	53	146	99
1	53	129	102	106	153	56	149	102
2		135	108	112	159	62	155	108
3			117	121	168	71	164	117
4				133	180	83	176	129
5					195	98	191	144
6						116	209	162
7							230	183
8								207

Hasil enkripsi segitiga pertama :

Sesuai dengan formula M_{ij} huruf pertama dari masing-masing baris sebanyak satu karakter pada nilai j = (N + i) – N akan menjadi *chipherimage* pada proses enkripsi segitiga pertama :

53, 129, 102, 106, 153, 56, 149, 102 hasil dari baris pertama
i = 1 dan j = (8 + 1) – 8 = 1

135, 108, 112, 159, 62, 155, 108 hasil dari baris kedua
i = 2 dan j = (8 + 2) – 8 = 2

117, 121, 168, 71, 164, 117 hasil dari baris ketiga
i = 3 dan j = (8 + 3) – 8 = 3

113, 180, 83, 176, 129 hasil dari baris keempat
i = 4 dan j = (8 + 4) – 8 = 4

195, 98, 191, 144 hasil dari baris kelima
i = 5 dan j = (8 + 5) – 8 = 5

116, 209, 162 hasil dari baris keenam
i = 6 dan j = (8 + 6) – 8 = 6

230, 183 hasil dari baris ketujuh
i = 7 dan j = (8 + 7) – 8 = 7

207 hasil dari baris kedelapan
i = 8 dan j = (8 + 8) – 8 = 8

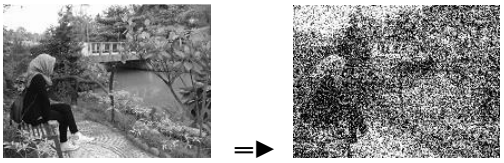
Sehingga *cipherimage* yang di hasilkan adalah :
53, 135, 117, 113, 195, 116, 230, 207

Tabel 3.3 *Cipherimage* Enkripsi Segitiga Pertama

Piksel		⟹	Piksel	
50	126		53	135
99	103		117	113
150	53		195	116
146	99		230	207
Citra Asli			*Cipherimage*	

enkripsi segitiga pertama

BULLETIN OF INFORMATION TECHNOLOGY (BIT)
Volume 2, No. 3, Nopember 2021, pp 119- 128
ISSN 2722-0524 (media online)



Gambar 3.4 *Cipherimage* Enkripsi Segitiga Pertama
Citra Asli *Cipherimage*
enkripsi segitiga pertama

2. Matriks enkripsi segitiga kedua
   Langkah-langkah yang dilakukan pada enkripsi segitiga kedua hampir sama dengan enkripsi segitiga pertama. Pada proses ini menggunakan faktor pengali dan kunci yang digunakan pada enkripsi segitiga pertama, dan *cipherimage* hasil enkripsi dari segitiga pertama menjadi *plainimage* untuk proses enkripsi segitiga kedua, yaitu :
   Plainimage = 53, 135, 117, 133, 195, 116, 230, 207
   N = 8
   K = 3
   R = 1,2,3,4,5,6,7,8
   Untuk baris pertama (i = 1), maka :
   $M_{1,1}$ = (P[1] + 3 * R[1] mod 256
   = (53 + 3* (1)) mod 256
   = (53 + 3) mod 256
   = 56 mod 256
   = 56
   $M_{1,2}$ = (P[2] + 3 * R[1] mod 256
   = (135 + 3 * (1)) mod 256
   = (135 + 3) mod 256
   = 138 mod 256
   = 138
   $M_{1,3}$ = (P[3] + 3 * R[1] mod 256
   = (117 + 3 * (1)) mod 256
   = (117 + 3) mod 256
   = 120 mod 256
   = 120
   $M_{1,4}$ = (P[4] + 3 * R[1] mod 256
   = (133 + 3 * (1)) mod 256
   = (133 + 3) mod 256
   = 136 mod 256
   = 136
   $M_{1,5}$ = (P[5] + 3 * R[1] mod 256
   = (195 + 3 * (1)) mod 256
   = (195 + 3) mod 256
   = 198 mod 256
   = 198
   $M_{1,6}$ = (P[6] + 3 * R[1] mod 256
   = (116 + 3 * (1)) mod 256
   = (116 + 3) mod 256
   = 119 mod 256
   = 119
   $M_{1,7}$ = (P[7] + 3 * R[1] mod 256
   = (230 + 3 * (1)) mod 256
   = (230 + 3) mod 256
   = 233 mod 256
   = 233
   $M_{1,8}$ = (P[8] + 3 * R[1] mod 256
   = (207 + 3 * (1)) mod 256
   = (207 + 3) mod 256
   = 210 mod 256
   = 210

Hasil dari enkripsi baris pertama adalah :
53, 135, 117, 113, 195, 116, 230, 207 → i = 0
56, 138, 120, 136, 198, 119, 233, **210** → i = 1

BULLETIN OF INFORMATION TECHNOLOGY (BIT)
Volume 2, No. 3, Nopember 2021, pp 119- 128
ISSN 2722-0524 (media online)

Hasil enkripsi dari baris pertama (i = 1) akan digunakan sebagai *plainimage* pada proses enkripsi baris kedua, dimana nilai $j \leq (N + 1) - i$,
sehingga : $i = 2;\ j \leq (8 + 1) - 2 \rightarrow j \leq 7$

$M_{2,1}$ = (P[1] + 3 * R[2] mod 256
= (56 + 3 * (2)) mod 256
= (56 + 6) mod 256
= 62 mod 256
= 62

$M_{2,2}$ = (P[2] + 3 * R[2] mod 256
= (138 + 3 * (2)) mod 256
= (138 + 6) mod 256
= 144 mod 256
= 144

$M_{2,3}$ = (P[3] + 3 * R[2] mod 256
= (120 + 3 * (2)) mod 256
= (120 + 6) mod 256
= 126 mod 256
= 126

$M_{2,4}$ = (P[4] + 3 * R[2] mod 256
= (136 + 3 * (2)) mod 256
= (136 + 6) mod 256
= 142 mod 256
= 142

$M_{2,5}$ = (P[5] + 3 * R[2] mod 256
= (198 + 3 * (2)) mod 256
= (198 + 6) mod 256
= 204 mod 256
= 204

$M_{2,6}$ = (P[6] + 3 * R[2] mod 256
= (119 + 3 * (2)) mod 256
= (119 + 6) mod 256
= 125 mod 256
= 125

$M_{2,7}$ = (P[7] + 3 * R[2] mod 256
= (233 + 3 * (2)) mod 256
= (233 + 6) mod 256
= 239 mod 256
= 239

Hasil dari enkripsi baris kedua adalah :

53, 135, 117, 113, 195, 116, 230, 207 → i = 0
56, 138, 120, 136, 198, 119, 233, **210** → i = 1
62, 144, 126, 142, 204, 125, **239** → i = 2

Baris selanjutnya dapat dicari dengan cara yang sama seperti diatas, sehingga di dapatkan *cipher* sebagai berikut :

Tabel 3.4 Hasil Enkripsi Segitiga Kedua

$M_{i,j}$	1	2	3	4	5	6	7	8
0	53	135	117	133	195	116	230	207
1	56	138	120	136	198	119	233	210
2	62	144	126	142	204	125	239	
3	71	153	135	151	213	134		
4	83	165	147	163	225			
5	98	180	162	178				
6	116	198	180					
7	137	219						
8	161							

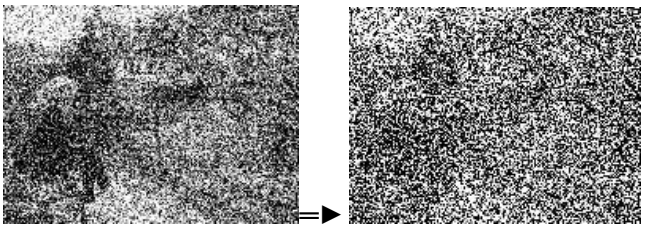
BULLETIN OF INFORMATION TECHNOLOGY (BIT)
Volume 2, No. 3, Nopember 2021, pp 119- 128
ISSN 2722-0524 (media online)

Hasil enkripsi segitiga kedua :
Sesuai dengan formula M_{ij} krakter terakhir dari masing-masing baris sebanyak satu karakter pada nilai j = (N + 1) – i akan menjadi *chipherimage* pada proses enkripsi segitiga kedua :
56, 138, 120, 136, 198, 119, 233, **210** hasil dari baris pertama
i = 1 dan j = (8 + 1) – 1 = 8
62, 144, 126, 142, 204, 125, **239** hasil dari baris kedua
i = 2 dan j = (8 + 1) – 2 = 7
71, 153, 135, 151, 213, **134** hasil dari baris ketiga
i = 3 dan j = (8 + 1) – 3 = 6
83, 165, 147, 163, **225** hasil dari baris keempat
i = 4 dan j = (8 + 1) – 4 = 5
98, 180, 162, **178** hasil dari baris kelima
i = 5 dan j = (8 + 1) – 5 = 4
116, 198, **180** hasil dari baris keenam
i = 6 dan j = (8 + 1) – 6 = 3
137, **219** hasil dari baris ketujuh
i = 7 dan j = (8 + 1) – 7 = 2
161 hasil dari baris kedelapan
i = 8 dan j = (8 + 1) – 8 = 1

Sehingga *cipherimage* yang di hasilkan adalah :
161, 219, 180, 178, 225, 134, 239, 210

Tabel 3.5 *Cipherimage* Enkripsi Segitiga Kedua

Piksel		⟹	Piksel	
53	135		161	219
117	113		180	178
195	116		225	134
230	207		239	210
Cipherimage enkripsi segitiga pertama			*Cipherimage* enkripsi segitiga kedua	



Gambar 3.5 *Cipherimage* Enkripsi Segitiga Kedua
Cipherimage enkripsi segitiga pertama *Cipherimage* enkripsi segitiga kedua

4. IMPLEMENTASI

4.1. Hasil

Implementasi adalah proses penggunaan aplikasi dan menguji aplikasi yang telah dibuat sesuai dengan rancangan. Adapun kebutuhan sistem dalam pembahasan ini dibagi menjadi 2 bagian yaitu, kebutuhan perangkat keras dan kebutuhan perangkat lunak.

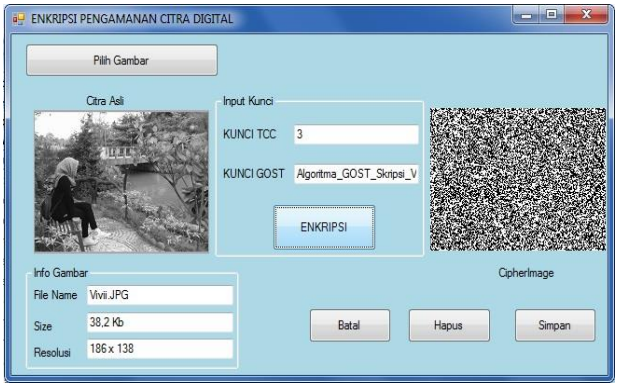
Tampilan Program

Tampilan program aplikasi terdiri dari tampilan menu utama, tampilan *form* enkripsi, tampilan *form* dekripsi dan tampilan *form info.*

BULLETIN OF INFORMATION TECHNOLOGY (BIT)
Volume 2, No. 3, Nopember 2021, pp 119- 128
ISSN 2722-0524 (media online)

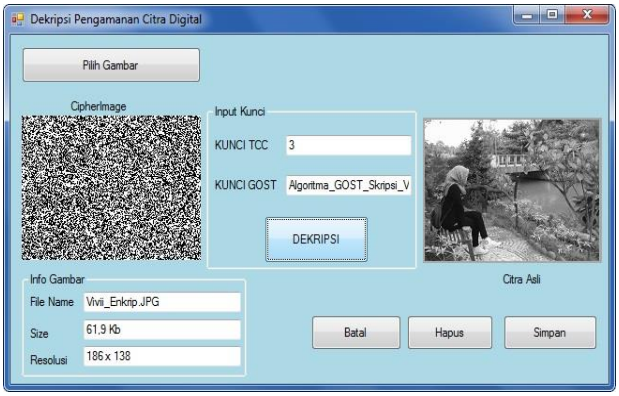
Gambar 4.1 Tampilan Menu Utama Aplikasi

1. Proses enkripsi *citra digital*



Gambar 4.2 Tampilan *Form* Enkripsi

2. Proses dekripsi *citra digital*



Gambar 4.3 Tampilan *Form* Dekripsi

5. KESIMPULAN

Berdasarkan hasil penerapan algoritma Triangle Chain Cipher dan GOST pada pengamanan citra digital yang telah dilakukan, penulis dapat menarik kesimpulan sebagai berikut:

1. Penerapan kombinasi algoritma triangle chain cipher dan algoritma GOST untuk meningkatkan keamanan citra digital dan meningkatkan kerahasiaan citra dimana citra tersebut disampaikan melalui citra lain sehingga tidak disalah gunakan oleh orang tidak bertanggung jawab.
2. Prosedur pengkombinasian algoritma triangle chain cipher dan algoritma GOST dilakukan secara berurut, dimana algoritma triangle chain cipher digunakan untuk menyandikan citra digital pertama kali, sedangkan algoritma GOST digunakan untuk menyandikan kembali citra digital.
3. Perancangan aplikasi yang dibangun ini sangat membantu dan mempermudah proses penyandian citra digital serta proses pengembalian citra digital kebentuk asli. Aplikasi yang dibangun berjalan normal pada sistem operasi windows 7 serta bahasa pemrograman yang digunakan dalam bahasa pemrograman visual basic 2008.

REFERENCES

[1] D. Ratnasari *et al.*, "Enkripsi Citra Digital Menggunakan Kombinasi Algoritme Hill Cipher Dan Chaos Map Dengan Penerapan Teknik Selektif Pada Bit Msb," *J. Teknol. TECHNOSCIENTIA*, vol. 10, no. 1, hal. 109–117, 2017.
[2] E. Setyaningsih, *Kriptografi & Implementasi Menggunakan MATLAB*. yogyakarta: Andi, 2015.
[3] J. Siregar, "Implementasi Keamanan Data Teks Dengan Algoritma Gost Dan Rot13," *J. INFOTEK*, vol. 1, no. 2, hal. 68–73, 2016.
[4] R. Syahputra, "Penerapan Mode Operasi Cipher Block Chaining Dan Metode Lsb-1 Dalam Pengamanan Data Teks," *J. Pelita Inform.*, vol. 16, no. Juli, hal. 318–321, 2017.
[5] R. Munir, *KRIPTOGRAFI*, Oktober 20. Bandung: Informatika, 2006.
[6] R. K. Hondro, "Analisis Dan Perancangan Sistem Yang Menerapkan Algoritma Triangle Chain Cipher (TCC) Untuk Enkripsi Record Tabel Database," *J. Teknol. Inf. Dan Komun.*, vol. 3, no. 2, 2014.
[7] R. Munir, *Algoritma & Pemrograman Dalam Bahasa Pascal dan C*. Bandung: Informatika, 2011.
[8] T. Zebua, "Analisa Dan Implementasi Algoritma Triangle Chain Pada Penyandian Record Database," *Pelita Inform. Budi Darma*, vol. 3, no. 2, 2013.
[9] E. Sutoyo, T Mulyanto, *Teori Pengolahan Citra Digital*. Yogyakarta: Andi, 2009.