Jurnal TRANSFORMATIKA
Vol.20, No.2, Juli 2023, pp. 38 -
P-ISSN: 1693-3656, E-ISSN: 2460-6731
https://journals.usm.ac.id/index.php/transformatika/

Pengamanan Citra Digital Terenkripsi Menggunakan Kombinasi Steganografi *Bit Matching* dan Kriptografi *Cipher Feedback* (CFB) dengan Pembangkit Bilangan Acak *Linear Congruential Generator* (LCG).

Dionaldo P. Laning[1*], Derwin R.Sina[2], Tiwuk Widiastuti[3], Arfan Mauko[4]
[1,2,3,4,]Ilmu Komputer, Fakultas Sains dan Teknik, Universitas Nusa Cendana
Jln AdiSucipto Penfui Kupang, NTT, e-mail: dionlaningg12@gmail.com, derwinsina@staf.undana.ac.id, tiwukwidiastuti@staf.undana.ac.id, arfanmauko@staf.undana.ac.id

ARTICLE INFO

History of the article :
Received 3 Juni 2023
Received in revised form 27 Juli 203
Accepted 27 Juli 2023
Available online 30 Juli 2023

Keywords:

Steganografi; Kriptografi; Pencocokan Bit; *Cipher Feedback*; *Linear Congruential Generator*

*** Correspondece:**

Telepon:
+62 81328381577

E-mail:
tiwukwidiastuti@staf.undana.ac.id

ABSTRACT

In this study, the design and manufacture of an encrypted digital image security system was carried out by combining bit matching steganography and cryptographic cipher feedback (CFB) using a linear congruential generator (LCG) as a key generator. The test material used 60 message images measuring 120x120 pixels with 1 cover image measuring 64x64 pixels. The results of the study were obtained: (1) the image can be hidden but there is an increase in the size of the embedding file with an average difference of 459.1167 KB (kilobytes) or an increase of 2135.65%, (2) The image can be reconstructed, the physical image of the extraction result is the same as the original image with an MSE value = 0, but there is a change in size that varies with an average difference of 3.433333 KB (kilobytes) or by 16.20%, (3) Some robust tests such as image rotation testing, brightness manipulation, addition of gaussian noise make the extracted image damaged. The image is resistant to all contrast tests. Salt and pepper noise testing does not damage the image too much, the image is still resistant to salt and pepper noise testing of 0.001 with an average MSE = 0.

INTRODUCTION

Citra atau gambar merupakan informasi visual dari suatu objek yang dapat diartikan sebagai gambar dari suatu objek[1]. Ada kalanya sebuah citra memuat suatu informasi penting atau rahasia sehingga perlu dijaga kerahasiaannya supaya tidak dimanipulasi oleh orang-orang yang tidak bertanggung jawab baik saat informasi tersebut diterima maupun saat dikirimkan. Teknik yang dipakai untuk mengatasi masalah ini adalah teknik kriptografi dimana data dienkripsi menjadi pesan acak dengan suatu kunci enkripsi sehingga data sulit dibaca jika tidak memiliki

kunci dekripsi [2] dan teknik steganografi yaitu teknik menyisipkan pesan rahasia didalam suatu media lain sebagai sehingga keberadaan pesan rahasia tidak dapat terdeteksi [3]. Teknik kriptografi mempunyai kelemahan data yang telah dienkripsi menghasilkan simbol-simbol yang aneh sehingga dapat menimbulkan kecurigaan bagi orang yang melihatnya [4] sehingga dibutuhkan sebuah media yang dapat digunakan untuk menyembunyikan pesan tersebut salah satunya dengan menggunakan teknik steganografi.Dengan menggabungkan kriptografi dan steganografi maka Keamanan pesan rahasia dapat ditingkatkan. Caranya dengan pesan mula-mula dienkripsikan dengan sebuah algoritma Kriptografi, dan hasil proses tersebut selanjutnya disembunyikan didalam media lain.

Algoritma Pengamanan pesan rahasia yang digunakan adalah Algoritma kriptografi *cipher feedback* (CFB) Algoritma CFB dapat mengenkripsi pesan dengan baik sehingga mempersulit kriptanalis menganalisis karakter atau kunci. CFB mempunyai kelemahan yaitu kunci yang sama yang digunakan untuk mengenkripsi plainteks yang berbeda menghasilkan *keystream* yang sama. Oleh karena itu dibutuhkan sebuah pembangkit bilangan acak sebagai pengganti kunci yang dapat membangkitkan sebuah bilangan acak pada tiap iterasinya. Salah satu algoritma pembangkit bilangan acak yang mudah diimplementasikan dan dapat digunakan adalah algoritma *linear congruential generator* (LCG)[5].

Salah satu teknik steganografi adalah *Bit Matching* (pencocokan bit). Keuntungan dari teknik ini yaitu kapasitas pesan yang disimpan dapat lebih besar dari [6] citra dan kualitas citra tidak berubah seperti aslinya [7].

Berdasarkan latar belakang kombinasi steganografi dan kriptografi yang dilakukan dapat digunakan untuk pengamanan citra digital. Peneltian ini menggunakan steganografi berbasis pencocokan bit (*bit matching*) pada bit pertama (*most significant bit*) dan algortima kriptografi *cipher feedback* dikombinasikan dengan algortima LGC yang digunakan sebagai pembangkit kunci dalam proses enkripsi dan dekripsi.

RESEARCH METHODS

2.1. Data Penelitian

Data yang digunakan adalah citra gambar yang didapat dari internet, dimanat citra yang digunakan pada proses *embedding* adalah citra dengan format jpg, jpeg dan bmp dengan jenis citra warna RGB dan citra keabuan *grayscale* yang ditransformasikan dari citra warna RGB. Total citra yang digunakan adalah 30 citra warna RGB gambar dan 30 citra keabuan *grayscale* dan 1 citra *cover* dengan jenis RGB dengan format bmp seperti pada Gambar 1

Gambar 1. Sampel citra *cover*

DOI : 10.26623/transformatika.v21i2.6978

2.2. Metode yang digunakan

Penelitian ini menggabungkan antara steganografi dengan kriptografi. Metode kriptografi yang digunakan adalah CFB. Sedangkan steganografi menggunakan dua proses yaitu : embedding dan ekstraksi. Kombinasi steganografi dan kriptografi ini dibutuhkan 4 proses, yaitu enkripsi, pencocokan bit dan rekonstruksi dan dekripsi. Rincian proses – proses tersebut adalah sebagai berikut :

Modifikasi Algoritma CFB

Pada mode *cipher feedback* (CFB) yang asli,pada setiap blok kuncinya sama, sehingga peneliti mengusulkan menggunakan mode CFB dengan kunci yang berbeda pada setiap blok. Algoritma *linear congruential generator* (LCG) membangkitkan bilangan acak yang berguna sebagai kunci pada setiap blok.

***Linear Congruential Generator* (LCG)**

Merupakan algoritma yang paling pambangkit kunci paling poluler karena mudah dipahami dan diimplementasikan. Secara umum algoritma LCG didefinisikan dengan persamaan berikut:

$$x_{i+1} + (a * x_i + b)m \quad (1)$$

Dalam penelitian ini nilai a,b, dan m yang digunakan mengacu pada penelitan [8] dapat dilihat pada Tabel 1

Tabel 1. Parameter LGC yang digunakan dalam penelitian ini

Variabel	Keterangan
m	m = 256;
b	b = bilangan prima yang lebih besar dari ½ m .
a	$a = \frac{m}{2} + 1$;
x_0	$x_0 = \left(\sum_{n=1}^{n} int(n[i])\right) mod\ m$

Ket :

x_0 = kunci pembangkit atau umpan (*seed*)

n= panjang kata kunci

$int(n[i])$ = nilai ASCII dari karakter ke i kata kunci

m = modulus LCG (256)

Enkripsi dan Dekripsi CFB

Proses enkripsi dan dekripsi pada penelitian ini adalah :

1) Algoritma enkripsi
a) Masukan citra pesan (*plainimage*), kunci dan ukuran blok.
b) Bagi blok citra pesan sesuai dengan ukuran blok.
c) Lakukan proses pembangkit bilangan acak dengan algoritma LGC.
d) Lakukan operasi CFB dan pergeseran karakter ke kanan sebesar total pergeseran.
e) Hasil enkripsi (*cipherimage*).
2) Algoritma dekripsi
a) Masukan *cipherimage*, kunci dan ukuran blok.
b) Bagi blok *cipherimage* sesuai dengan ukuran blok.
c) Lakukan proses pembangkit bilangan acak dengan algoritma LGC
d) Lakukan operasi CFB dan pergeseran karakter ke kiri sebesar total pergeseran.
e) Hasil dekripsi (*plainimage*).

Pencocokan Bit

Pencocokan bit dilakukan secara *divide* and *conquer* [9]. Masukan berupa citra pesan hasil enkripsi (*cipherimage*) dan citra *cover* (*coverimage*). Proses – proses pencocokan bit meliputi :

1. Mengubah nilai piksel citra pesan dan citra *cover* menjadi citra biner
2. Mengambil nilai MSB citra *cover*
3. Mencocokkan bit citra pesan pada bit citra *cover*. Apabila bit pesan ada pada bit citra *cover*, maka selanjutnya menyimpan posisi indeks bit yang terdiri dari posisi indeks bit awal (*start*) dan posisi indeks bit akhir (*end*). Jika bit citra pesan tidak terdapat pada bit citra *cover*, dilanjutkan proses 4)
4. Pesan dibagi menjadi dua bagian sama panjang kiri (L[i]) dan kanan(R[i]) yang akan digunakan sebagai masukan pada proses selanjutnya
5. Mengulangi langkah yang sama pada langkah nomor 3). Apabila semua bit citra pesan ada pada citra *cover*, maka pencocokan selesai dan dilanjutkan proses 6). Jika tidak, ulangi langkah 4).
6. Simpan semua indeks bit yang didapat dari hasil pencocokan.
7. Hasil Keluaran berupa vektor yang memuat susunan indeks posisi bit.

Misalkan diketahui bit piksel citra pesan dan bit citra *cover* sebagai berikut:
Pesan (P) : P = 00101100; C = 010100011
karena P tidak terdapat pada C maka pecah P menjadi 2 bagian sama Panjang L1 dan R1 dimana

1. L[1] = 0010 tidak terdapat pada citra *cover* maka membagi L[1] menjadi dua bagian yaitu :
   L[2] = 00 diperoleh kesamaan pada lokasi index ke 4 hingga 5, yaitu 0101**<u>00</u>**011
   R[2] = 10 diperoleh kesamaan pada lokasi index ke1 hingga 2, yaitu 0**<u>10</u>**100011
2. R[1] = 0101 tidak terdapat pada citra *cover* maka membagi R[1] menjadi dua bagian yaitu :
3. L[2] = 11 diperoleh kesamaan pada lokasi index ke 7 hingga 8, yaitu 0101000**<u>11</u>**
4. R[2] = 00 diperoleh kesamaan pada lokasi index ke 4 hingga 5, yaitu 0101**<u>00</u>**011
5. Jika semua posisi bit sudah ditemukan, maka proses pencocokan selesai dan dilanjutkan langkah 4.
6. Semua solusi yang diperoleh pada langkah pertama piksel pertama yaitu “4 5 1 2”, dan “7 8 4 5” digabungkan. Jadi solusi totalnya adalah 4 5 1 2 7 8 4 5.

Rekonstruksi

Tujuan dari proses rekonstruksi adalah mengembalikan pesan menjadi bentuk semula.Masukan berupa vektor indeks lokasi bit dan citra *cover*. Proses rekonstruksi dilakukan dengan cara mengambil susunan bit citra berdasarkan vektor indeks lokasi bit. Langkah-langkah yang dilakukan pada proses rekonstruksi adalah:

1. Mengkonversi nilai piksel citra *cover* dalam bentuk biner dan mengambil bit MSB citra.
2. Membaca setiap dua indeks isi vektor. Indeks pertama merupakan posisi swal bit (*start*) dan indeks kedua merupakan posisi akhir bit (*end*),
3. Mengambil nilai bit citra berdasarkan langkah 2),
4. Mengulangi proses 2) dan 3) sampai posisi indeks terakhir.
5. Susunan bit yang terbentuk akan menghasilkan keluaran berupa susunan bit

Misalkan sebuah vektor dan citra sebagai berikut:
Vektor : 4 5 1 2 7 8 4 5
Citra : 010100011
Selanjutnya dilakukan ekstraksi dengan mengambil nilai bit citra sesuai lokasi vektor kemudian semua nilai bit hasil pencocokan dari vektor kemudian semua nilai bit hasil pencocokan dari vector disusun. Hasil pencocokkan ini adalah :

1. Vektor 4 5, menghasilkan 00
2. Vektor 1 2, menghasilkan 10
3. Vektor 7 8, menghasilkan 11
4. Vektor 4 5, menghasilkan 00

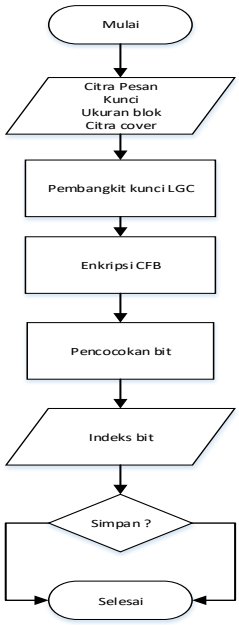
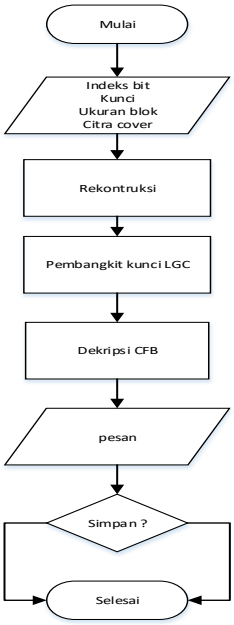
DOI : 10.26623/transformatika.v21i2.6978

Semua hasil diatas digabung, sehingga menghasilkan output 00101100 (dikonversi menjadi 44 dalam desimal). Nilai ini akan digunakan sebagai nilai piksel citra.

Kombinasi Steganografi dan kriptografi

Kombinasi steganografi dan kriptografi terdiri dari dua proses utama yaitu proses *embedding* dan proses ektraksi. Penelitian ini menggunakan metode steganografi yang dilakukan dengan mencocokan bit pesan pada *cover* [10]. *Flowchart* proses *embedding* dan ekstraksi dapat dilihat pada Tabel 1

Tabel 1. Flowchart proses *embedding* dan ekstraksi

Flowchart embedding	***Flowchart* ekstraksi**
	

Proses *embedding*

Tujuan dari proses embedding adalah menghasilkan indeks posisi bit. Masukkan berupa citra pesan, citra *cover*, kunci dan ukuran blok. Tahapan proses *embedding* :

1. Input citra pesan, citra *cover*, kunci dan ukuran blok
2. Enkripsikan citra pesan dengan algoritma CFB
3. konversikan citra pesan (*cipherimage*) dan citra *cover* dalam bentuk biner.
4. Ambil nilai bit MSB citra *cover*
5. cocokkan setiap bit citra pesan dengan bit citra *cover*. Posisi bit yang sama disimpan dalam vektor indeks bit
6. Hasil keluaran berupa vektor urutan indeks bit
7. Selesai.

Proses ekstraksi

Bertujuan untuk mengembalikan citra pesan kebentuk semula. Masukkan berupa vektor urutan indeks (indeks bit), kunci, ukuran blok dan citra *cover*. Langkah-langkah proses ekstraksi adalah sebagai berikut:

1. Input kunci,vektor indeks bit, dan citra *cover*.
2. Rekonstruksi citra pesan dengan mencocokan bit citra *cover* berdasar vektor indeks bit
3. Dekripsi citra pesan dengan algoritma CFB

4. Selesai.

3. HASIL

Sistem pengaman citra digital dibangun berbasis desktop. Gambar 2 memperlihatkan menu utama. Menu utama memuat tombol *embedding*, ekstraksi dan pengujian yang ketika diklik masing-masing akan menampilkan tampilan menu *embedding* ekstraksi dan pengujian. Menu *embedding* dapat dilihat pada Gambar 3 berfungsi untuk menyembunyikan citra pesan yang telah terenkripsi. Menu ekstraksi dapat dilihat pada Gambar 4 digunakan untuk mengembalikan citra pesan ke bentuk semula. Menu pengujian dapat dilihat pada Gambar 5 digunakan untuk menghitung nilai MSE dan PSNR citra. Pada proses *embedding* citra pesan dienkripsikan terlebih dahulu baru kemudian dilakukan proses pencocokan bit citra hasil enkripsi dengan citra *cover*. Hasil dari proses *embedding* berupa vektor urutan posisi disimpan dalam file dengan format .txt yang nantinya digunakan sebagai masukan pada proses ekstraksi untuk mengembalikan citra pesan ke bentuk semula. Contoh hasil *embedding* dapat dilihat pada Gambar 6.

Figure 2. Menu Utama

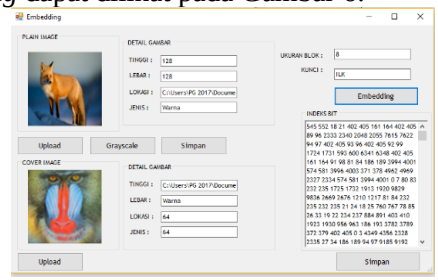


Figure 3. Citra yang terenkripsi

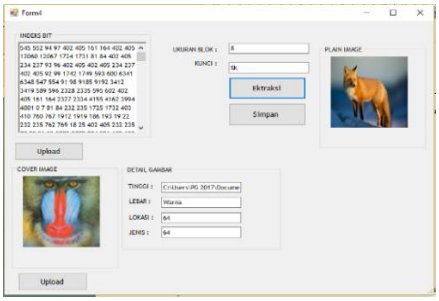


Figure 4. Menu ekstraksi

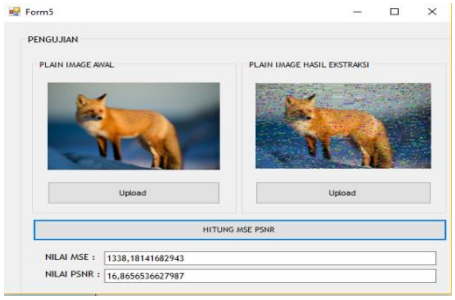
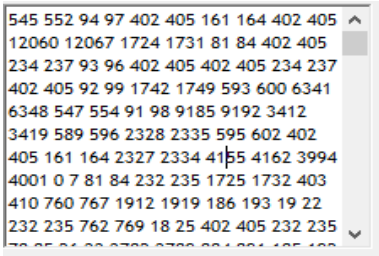


Figure 5. Menu pengujian



Gambar 6. Indeks bit hasil *embedding*

RESULTS AND DISCUSSION

Pengujian dilakukan pada 60 citra pesan yang terdiri dari 30 gambar RGB dan 30 gambar *grayscale* berukuran 128 x 128 dengan ekstensi jpg, jpeg, bmp dan 1 citra *cover* berukuran 64 x 64 dengan ekstensi bmp. Pengujian yang dilakukan berupa pengujian *fidelity*, pengujian

recorvery, dan pengujian *robustness* (manipulasi citra *cover*) yang terdiri dari pengujian rotasi gambar, manipulasi *contrast*, manipulasi *brightness*, pengujian penambahan *noise gaussian* dan *noise salt and peper*. Hasil pengujian disajikan dalam tabel dibawah.

Tabel 2. Hasil pengujian *fidelity*

No	Citra Pesan	*Size*(kb)	*Embedding*		
			Hasil *Embedding*	*Size*(kb)	Selisih (kb)
1	coba1.jpg	19	indeksbit1.txt	481	462
2	coba2.jpg	25	indeksbit2.txt	480	455
...	...	...	...	...	...
59	coba1.bmp	19	indeksbit59.txt	480	461
60	coba1.bmp	23	indeksbit60.txt	481	458
Rata-rata		21,5		480,6667	459,1167

Tabel 2 memperlihatkan hasil pengujian *fidelity* yang telah dilakukan. Aplikasi yang dibangun dapat menyembunyikan citra namun ukuran (*size*) file hasil *embedding* menjadi lebih besar dari ukuran citra dengan rata-rata selisih ukuran sebesar 459,1167 kb atau terjadi peningkatan sebesar 2135,6%.

Tabel 3. Hasil pengujian *recorvery*

No	Citra Pesan	*Size*(kb)	Ekstraksi				
			Hasil ekstraksi	*Size*(kb)	Selisih(kb)	MSE	PSNR
1	coba1.jpg	19	citrahasil1,jpg	36	17	0	∞
2	coba2.jpg	25	citrahasil2.jpg	36	11	0	∞
...	...	...	...	...	...	...	...
59	coba59.bmp	19	citrahasil59.bmp	18	-1	0	∞
60	coba60.bmp	23	citrahasil60.bmp	23	0	0	∞
Rata-rata		21,5		24,98333	3,433333	0	∞

Keterangan:
- : citra hasil mengalami penambahan ukuran (*size*).
- : citra hasil mengalami pengurangan ukuran (*size*).

Tabel 3 memperlihatkan hasil pengujian *recorvery*. Citra pesan dapat dikembalikan seperti semula. Fisik citra hasil ektraksi sama seperti citra aslinya dengan rata-rata MSE = 0, namun ukuran citra yang dihasilkan bervariasi dengan rata-rata selisih 3,433333 atau terjadi peningkatan sebesar 16.20%.

Tabel 4. Hasil pengujian rotasi gambar

No	Citra Pesan	180°	90° ke kanan	90° ke kiri
1				
2				
...	...	...	...	...
59				
60				

Tabel 5. Nilai MSE dan PSNR pengujian rotasi gambar

No	180°		90° Kekanan		90° kekiri	
	MSE	PSNR	MSE	PSNR	MSE	PSNR
1	8962,97	9	8971,38	9	8905	9
2	11091,38	8	11058,35	8	11190,18	8
...	...	...	...	...	...	...
59	11606,54	7	11489,79	8	11554,52	8
60	8575,36	9	8426,41	9	8723,94	9
Rata-rata	12180,01	7,423729	11959,44	7,5	11956,95	7,566667

Tabel 4 dan 5 memperlihatkan hasil pengujian rotasi diuji dengan merotasi citra *cover* sebesar 180°, 90° ke kanan dan 90° ke kiri. Hasil pegujian diperoleh citra hasil ekstraksi menjadi rusak. Hal ini disebabkan karena proses pencocokan bit dilakukan pada nilai MSB citra *cover* sedangkan merotasi citra *cover* mengakibatkan posisi indeks bit berubah.

Tabel 6. Citra hasil pengujian manipulasi *contrast*

No	Citra Pesan	Citra Hasil Ekstraksi					
		-50%	-25%	-10%	+10%	+25%	+50%
1							
2							
...	...	...	...	...	...	...	...	...
59							
60							

Tabel 7. Nilai MSE dan PSNR pengujian manipulasi *contrast*

No	Nilai *brightness*					
	-50%		-25%		-10%	
	MSE	PSNR	MSE	PSNR	MSE	PSNR
1	0	∞	0	∞	0	∞
2	0	∞	0	∞	0	∞
...	0	∞	0	∞	0	∞
59	0	∞	0	∞	0	∞
60	0	∞	0	∞	0	∞
Rata-rata	0	∞	0	∞	0	∞

No	Nilai *brightness*					
	+10%		+25%		+50%	
	MSE	PSNR	MSE	PSNR	MSE	PSNR
1	0	∞	0	∞	0	∞
2	0	∞	0	∞	0	∞
...	0	∞	0	∞	0	∞

DOI : 10.26623/transformatika.v21i2.6978

29	0	∞	0	∞	0	∞
30	0	∞	0	∞	0	∞
Rata-rata	0	∞	0	∞	0	∞

Tabel 6 dan 7 memperlihatkan hasil pengujian manipulasi *contrast* diuji dengan mengubah *contrast* pada citra *cover* dengan besaran -50%, -25%, -10%, +10%, +25%, +50%. Citra tahan terhadap pemberian *contrast*, citra masih dapat dikenali dengan rata-rata nilai MSE = 0. Penyebabnya adalah pemberian *contrast* tidak mempengaruhi nilai bit citra *cover*. Proses pencocokan bit berdasarkan indeks posisi bit yang tepat sehingga citra masih dapat direkontruksi seperti semula.

Tabel 8. Citra hasil pengujian manipulasi *brightness*

No	**Citra Pesan**	**Citra Hasil Ekstraksi**					
		-50%	**-25%**	**-10%**	**+10%**	**+25%**	**+50%**
1							
2							
...	...	...	...	...	...	...	...
59							
60							

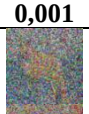
Tabel 9. Nilai MSE dan PSNR pengujian manipulasi*brightness*

No	**Nilai *brightness***					
	-50%		**-25%**		**-10%**	
	MSE	**PSNR**	**MSE**	**PSNR**	**MSE**	**PSNR**
1	8847,6	9	8503,25	9	6693,3	10
2	11142,37	8	10426,06	8	7573,92	9
...	...	...	...	...	...	...
59	11622,33	7	10888,21	8	7710,29	9
60	8591,06	9	8196,4	9	6507,15	10
Rata-rata	12069,62	7,416667	11214,63	7,8	7966,992	9,216667

No	**Nilai *brightness***					
	+10%		**+25%**		**+50%**	
	MSE	**PSNR**	**MSE**	**PSNR**	**MSE**	**PSNR**
1	4283,88	12	6715,23	10	7989,33	9
2	4726,6	11	7885,34	9	9654,75	8
...	...	...	...	...	...	...
29	4852,1	11	8173,19	9	10167,98	8
30	4269,81	12	6737,42	10	7885,49	9
Rata-rata	4879,605	11,26667	8390,055	8,933333	10306,53	8,083333

Tabel 8 dan 9 memperlihatkan hasil pengujian manipulasi *brightness* diuji dengan mengubah *brightness* pada citra *cover* dengan besaran -50%, -25%, -10%, +10%, +25%, +50%. Pemberian *brightness* mengakibatkan citra hasil ekstraksi menjadi rusak. Hal ini disebabkan pemberian *brightness* mempengaruhi nilai bit citra *cover*. Semakin besar atau kecil nilai yang diberikan membuat citra hasil ekstraksi semakin sulit dikenali. Citra masih sedikit dapat dikenali pada pemberian *brightness* +10% dengan rata-rata MSE = 4879,605 dan PSNR sebesar 11,26667db.

Tabel 10. Citra hasil pengujian penambahan*noise gaussian* (*mean* 0)

No	Citra Pesan	Citra hasil ekstraksi			
		0,001	0,01	0,005	0,05
1					
2					
…	…	…	…	…	…
59					
60					

Tabel 11. Nilai MSE dan PSNR pengujian penambahan *noise gaussian*

No	***Noise gaussian* (*mean*=0)**							
	0,001		**0,01**		**0,005**		**0,05**	
	MSE	**PSNR**	**MSE**	**PSNR**	**MSE**	**PSNR**	**MSE**	**PSNR**
1	7158,43	10	6170,17	10	7437,56	9	6589,18	10
2	8507,93	9	7185,52	10	8786,98	9	7723,86	9
…	…	…	…	…	…	…	…	…
59	8654,85	9	7339,67	9	8970,35	9	8001,96	9
60	6925,16	10	6085,13	10	7186,14	10	6454,81	10
Rata-rata	9000,852	8,75	7657,699	9,316667	9442,332	8,483333	8220,836	9,05

Tabel 10 dan 11 memperlihatkan hasil pengujian *noise gaussian* diuji dengan menambahkan *noise gaussian* pada citra dengan *mean* = 0 dan standar deviasi *d* = 0,001%; +0,005%; +0,01%; +0,05%. Pemberian *noise gaussian* mengakibatkan citra hasil ekstraksi menjadi rusak sehingga menghasilkan nilai MSE dan PSNR yang buruk. Citra masih sedikit dapat dikenali pada pemberian *noise* 0.01 dengan rata-rata MSE = 7657,699 dan PSNR = 9,316667 db.

Tabel 12. Citra hasil pengujian penambahan *noise salt and pepper*

No	Citra Pesan	Citra hasil ekstraksi			
		0,001	0,01	0,005	0,05
1					
2					

DOI : 10.26623/transformatika.v21i2.6978

…	…	…	…	…	…
59					
60					

Tabel 13. Nilai MSE dan PSNR pengujian penambahan *noise salt and pepper*

No	***Noise salt and pepper***							
	0,001		**0,01**		**0,005**		**0,05**	
	MSE	**PSNR**	**MSE**	**PSNR**	**MSE**	**PSNR**	**MSE**	**PSNR**
1	0	∞	187,03	25	46,19	31	1370,55	17
2	0	∞	185,11	25	42,96	32	1427,17	17
…	…	…	…	…	…	…	…	…
59	0	∞	182,15	26	44,62	32	1330,12	17
60	0	∞	194,97	25	44,38	32	1361,67	17
Rata-rata	0	∞	183,9518	25,5	44,3755	31,91433	1392,69	16,95

Tabel 12 dan 13 memperlihatkan hasil pengujian *noise salt and pepper* diuji dengan menambahkan *noise salt and peper* pada citra *cover* dengan standar deviasi *d* = 0,001; +0,005; +0,01; +0,05. Pemberian *noise salt and peper* tidak terlalu merusak citra. Citra masih tahan terhadap pemberian *noise* 0,001 dengan rata-rata MSE = 0 sedangkan gambar mulai mengalami kerusakan pada pemberian *noise* 0,005, 0,01, dan 0,05.

5. KESIMPULAN

Berdasarkan pengujian yang dilakukan sistem pengaman dibangun dengan kombinasi steganografi dan kriptografi yang dipakai dapat digunakan untuk pengamanan citra digital. Citra pesan dapat disembunyikan dan dapat dikembalikan kebentuk semula namun ukuran (*size*) file hasil *embedding* lebih besar dari ukuran citra pesan dan ukuran (*size*) citra hasil ekstraksi yang dihasilkan bervariasi. Citra tahan terhadap pengujian *robust* seperti pengujian manipulasi *contrast* dan penambahan *noise salt and pepper* 0,001 dengan rata-rata MSE = 0, tetapi menghasilkan nilai MSE dan PSNR yang buruk pada pengujian rotasi gambar, pengujian manipulasi *brightness*, dan penambahan *noise gaussian* sehingga mengakibatkan citra hasil ekstraksi menjadi rusak.

6. DAFTAR PUSTAKA

[1] R. Munir, *Pengolahan Citra Digital dengan Pendekatan Algoritmik*. Bandung: Informatika, 2004.

[2] Sentot Kromodimoedjo, *Teori Dan Aplikasi Kriptografi*. SPK IT Consulting, 2009.

[3] R. Munir, *Kriptografi*, 2 ed. Informatika Bandung, 2019.

[4] A. Sonbay, "Penyandian Data Teks Menggunakan Algoritma Cipher Feedback Dan Chaotic Skew Tent Map.," vol. 5, 2017.

[5] K. Wibowo, "Pengujian Tingkat Keacakan dan Kecepatan Beberapa Algoritma Pembangkit Bilangan Acak Semu dengan Perbandingan Standar Deviasi dan Perbandingan Waktu Pembangkitan," 2012.

[6] R. Supardi, "Implementasi metode bit matching untuk keamanan pesan teks menggunakan visual basic. Net," vol. 1, Desember 2017.

[7] B. Prasetyo, "Kombinasi Steganografi Berbasis Bit Matching dan Kriptografi DES untuk Pengamanan Data," vol. 1, 2014.
[8] J. Fora, "Enkripsi Citra Digital Grayscale 8 Bit Dan Rgb 24 Bit Menggunakan Algoritma Output Feedback (OFB) Dengan Pembangkit Bilangan Acak Combined Linear Congruential Generator (ClGC)." 2015.
[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Introduction to Algorithms, Third Edition".
[10] K. Challita, and H. Farhat, "Combining Steganography and Cryptography: New Directions," 2011.