Scientific Journal of Informatics, Vol. 1, No. 1, Mei 2014
ISSN 2407-7658

Kombinasi Steganografi Berbasis Bit *Matching* dan Kriptografi DES untuk Pengamanan Data

Budi Prasetiyo[1], Rahmat Gernowo[2] & Beta Noranita[3]

[1] Jurusan Ilmu Komputer, Fakltas MIPA, Universitas Negeri Semarang
[2,3] Program Magister Sistem Informasi, Universitas Diponegoro
Email: prasemath@gmail.com, rahmatgernowo@undip. ac. id & bethznice@yahoo.com

Abstrak. Pada penelitian ini dilakukan kombinasi steganografi dan kriptografi untuk pengamanan data dengan tidak mengubah kualitas media *cover*. Metode steganografi yang digunakan dengan melakukan pencocokan bit pesan pada bit MSB citra. Proses pencocokan dilakukan secara *divide and conquer*. Hasil indeks posisi bit kemudian dienkripsi menggunakan algoritma kriptografi *Data Encryption Standard* (DES). Masukkan data berupa pesan teks, citra, dan kunci. *Output* yang dihasilkan berupa chiperteks posisi bit yang dapat digunakan untuk merahasiakan data. Untuk mengetahui isi pesan semula diperlukan kunci dan citra yang sama.

Kombinasi yang dihasilkan dapat digunakan untuk pengamanan data. Kelebihan metode tersebut citra tidak mengalami perubahan kualitas dan kapasitas pesan yang disimpan dapat lebih besar dari citra. Hasil pengujian menunjukkan citra hitam putih maupun *color* dapat digunakan sebagai *cover*, kecuali citra 100% hitam dan 100% putih. Proses pencocokan pada warna citra yang bervariasi lebih cepat. Kerusakan pesan dengan penambahan *noise salt and peper* mulai terjadi pada nilai MSE 0,0067 dan *gaussian* mulai terjadi pada nilai MSE 0,00234.

Kata kunci: *Steganography; cryptography; bit matching; divide and conquer; DES.*

1. PENDAHULUAN

Seiring perkembangan zaman, kebutuhan manusia akan informasi semakin meningkat. Ditengah-tengah perkembangan teknologi informasi yang kian semarak, internet tidak lagi menjamin penyediaan informasi yang aman. Berbagai mesin-pencari (*search-engine*) terus berkembang ditambah dengan serangan virus, penyadap, *spam* maupun *hacker* yang menjamur dapat mencuri data-data bersifat rahasia [1]. Mengatasi hal tersebut berbagai cara untuk meningkatkan keamanan data terus dikembangkan, diantaranya kriptografi dan steganografi.

Steganografi adalah seni dan ilmu menyembunyikan data pada media lain sebagai *cover* (misalnya citra) sehingga terlihat samar [2]. Kriptografi adalah seni dan ilmu menjaga kerahasiaan data [3]. Pada kriptografi, data asli diubah menjadi bentuk lain yang tidak dapat dibaca. Penggabungan steganografi dan kriptografi secara bersamaan dapat meningkatkan pengamanan data [4].

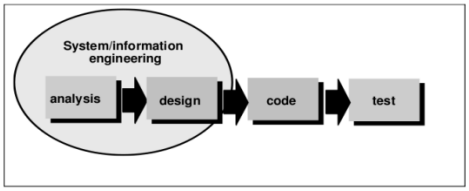
Metode penggabungan steganografi dan kriptografi banyak dikembangkan. Pada umumnya teknik yang digunakan yaitu dengan mengenkripsi pesan terlebih dahulu (kriptografi), kemudian menyisipkannya ke media *cover* (steganografi) [5]. Namun, proses penyisipan dapat berpengaruh pada kualitas media *cover* tersebut. Upaya untuk meminimalisir perubahan kualitas *cover* dapat dilakukan dengan penyisipan pada bit terakhir (*least significant bit*). Perubahan kualitas *cover* tidak tampak kasat mata [6], tetapi penyisipan pada bit terakhir mengakibatkan *cover* rentan terhadap *robust*. Ketahanan terhadap *robust* dapat dilakukan dengan pemilihan pada bit pertama (*most significant bit*), tetapi justru perubahan kualitas *cover* menjadi besar dan dapat dicurigai.

Mengembangkan cara baru penggabungan steganografi dan kriptografi tanpa mengubah media *cover*. Teknik yang dilakukan yaitu dengan mencocokan bit pesan pada *cover*, kemudian dilanjutkan proses enkripsi (kriptografi). Salah satu algoritma kriptografi yang terkenal sejak 1977 dan menjadi standar adalah *Data Encryption Standard* (DES) [7].

Pada penelitian ini akan dilakukan kombinasi steganografi dan kriptografi tanpa mengubah media *cover*. Metode steganografi yang digunakan berbasis pencocokan bit (*bit matching*) pada bit pertama (*most significant bit*) dan metode kriptografi yang digunakan yaitu algoritma DES.

2. METODOLOGI

Metode yang digunakan yaitu penggabungan steganografi dengan kriptografi. Algoritma kriptografi yang digunakan adalah DES. Terdapat 2 proses didalam steganografi, yaitu *embedding* dan ekstraksi. Pada penelitian ini dibangun suatu perangkat lunak stego-kripto dengan model *waterfall*. Metode *waterfall* ditunjukkan pada Gambar 1.



Gambar 1 Metode *waterfall* [8].

Metode *waterfall* membagi menjadi 4 tahap yang saling terkait dan mempengaruhi. Empat tahap tersebut yaitu analisa kebutuhan (*analysis*), desain (*design*), pengkodean (*code*) dan pengujian (*test*) [8]. Kombinasi kriptografi dan steganografi ini dibutuhkan 4 proses, yaitu pencocokan bit, enkripsi, dekripsi dan rekonstruksi yang secara rinci diuraikan sebagai berikut:

2.1 Pencocokan Bit

Pada penelitian ini metode pencocokan dilakukan secara *divide and conquer* [9]. Masukan pada proses ini adalah pesan dan citra.

Langkah-langkah yang dilakukan pada pencocokan bit adalah:
a) Mengkonversi pesan dan citra dalam bentuk biner
b) Mengambil nilai MSB citra
c) Melakukan pencocokan pesan pada MSB citra. Jika bit pesan terdapat pada MSB citra, maka dilanjutkan dengan menyimpan posisi indeks bit. Penyimpanan indeks terdiri dari posisi indeks bit awal (*start*) dan posisi indeks bit akhir (*end*). Jika proses pencocokan tidak terjadi, dilanjutkan proses d)
d) Membagi pesan menjadi dua bagian sama panjang kiri (L[i]) dan kanan (R[i])
e) Mengulangi langkah yang sama seperti pada nomor b), dengan L[i] dan R[i] sebagai masukan. Jika semua bit pesan terdapat pada citra, maka pencocokan selesai dan dilanjutkan proses f). Jika tidak, mengulangi langkah c) dengan L[i] dan R[i] sebagai masukkan hingga proses ke-i.
f) Menyimpan semua indeks bit hasil pencocokan
g) Keluaran berupa vektor yang memuat susunan indeks posisi bit.
   Sebagai contoh, misalkan diketahui bit pesan dan bit citra sebagai berikut:
   Pesan (P) : 10110111
   Citra(C) : 100100011010110101010011

Karena P tidak terdapat pada C maka P dipecah menjadi dua bagian kiri (L) dan kanan (R), yaitu:

1. L[1]: 1011 yang terletak pada posisi indeks "11, 14", yaitu: 10010001101011010101010011.
2. R[1]: 0111, tidak terdapat pada citra, maka membagi R[1] menjadi dua bagian yaitu:
   a. L[2]: 01, terletak pada posisi indeks ke"3 4"
   b. R[2]: 11, terletak pada posisi indeks ke"8 9"
3. Karena semua posisi bit sudah ditemukan, maka proses pencocokan selesai dan dilanjutkan langkah 4.
4. Menggabungkan semua solusi langkah 1, langkah 2a dan 2b. Diperoleh posisi indeks bit keseluruhan "11 14 3 4 8 9".

2.2 Enkripsi

Vektor posisi bit yang diperoleh pada sub Bab 3. 1 kemudian di enkripsi. Proses enkripsi dilakukan dengan algoritma DES.

2.3 Dekripsi

Masukan pada proses ini adalah *chiperteks* dan kunci. Dekripsi terhadap *cipherteks* merupakan kebalikan dari proses enkripsi. DES menggunakan algoritma yang sama untuk proses enkripsi dan dekripsi. Pada proses dekripsi urutan kunci yang digunakan adalah kebalikanya yaitu K_{16}, K_{15}, …,K_1. Untuk tiap putaran 16, 15, ..., 1, keluaran pada setiap putaran *deciphering* adalah:

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

2.4 Rekonstruksi

Rekonstruksi bertujuan untuk mengembalikan pesan menjadi bentuk semula. Masukan pada tahap ini terdiri dari vektor indeks lokasi bit dan citra. Proses yang dilakukan yaitu dengan mengambil susunan bit citra berdasar vektor indeks lokasi bit. Hasil keluaran proses tersebut berupa susunan bit pesan.

Langkah-langkah yang dilakukan pada proses rekonstruksi adalah:

a) Mengkonversi citra dalam bentuk biner dan mengambil bit MSB citra.
b) Membaca setiap dua indeks isi vektor. Indeks pertama merupakan posisi awal bit (*start*) dan indeks kedua merupakan posisi akhir bit (*end*),
c) Mengambil nilai bit citra berdasarkan langkah b),
d) Mengulangi proses b) dan c) sampai posisi indeks terakhir.
e) Susunan bit yang terbentuk akan menghasilkan keluaran berupa susunan bit pesan.

Sebagai contoh, misalkan diketahui vektor dan citra sebagai berikut:
Vektor : 11 14 3 4 8 9
Citra : 100100011010110101010011

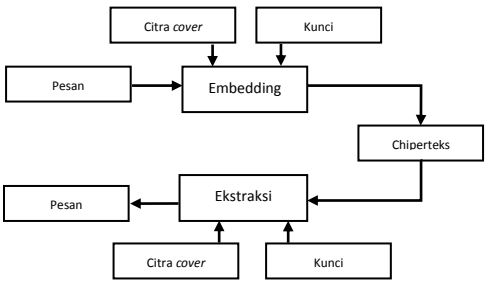
Langkah ekstraksi yaitu dengan mengambil nilai bit citra berdasarkan lokasi vektor. Menyusun semua nilai bit hasil pencocokan dari vektor. Pada kasus tersebut diperoleh kecocokan yaitu:
a. Vektor 11 14, menghasilkan 1011
b. Vektor 3 4, menghasilkan 01
c. Vektor 8 9, menghasilkan 11

Semua hasil diatas digabung, sehingga menghasilkan *output* 10110111.

2.5 Kombinasi Steganografi dan Kriptografi

2.5.1 Gambaran Umum

Kombinasi steganografi dan kriptografi pada penelitian ini terdiri dari 2 proses utama, yaitu proses *embedding* dan ekstraksi yang secara umum ditunjukkan pada Gambar 2.

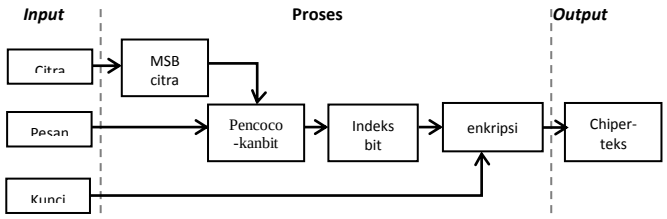


Gambar 2 Gambaran umum kombinasi steganografi dan kriptografi pada penelitian ini.

Proses *embedding* pada Gambar 3 terdiri dari pencocokan bit dan enkripsi, hasilnya *chiperteks*. Proses ekstraksi pada Gambar 5 terdiri dari dekripsi dan rekonstruksi, hasilnya berupa pesan.

2.5.2 Proses Embedding

Proses*embedding* pada Gambar 3 bertujuan untuk menghasilkan indeks posisi bit. Masukkan proses *embedding* berupa pesan, citra dan kunci.

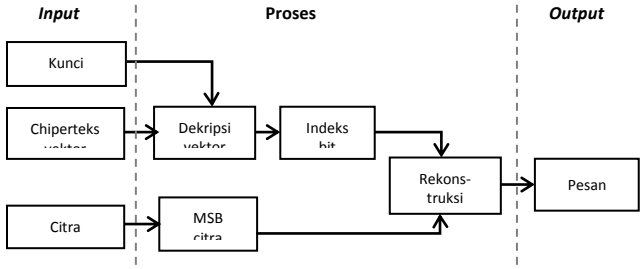


Gambar 3 Proses *embedding*.

Langkah-langkah *embedding* adalah sebagai berikut:
a. Memasukkan *input* berupa citra, pesan dan kunci.
b. Mengkonversi pesan dan citra dalam bentuk biner.
c. Mencocokan bit pesan dengan bit MSB citra. Posisi bit yang sama disimpan dalam vektor indeks bit.
d. Mengenkripsi vektor indeks bit dengan algoritma DES.
e. Hasil keluaran berupa *chiperteks*. *Chiperteks* tersebut memuat vektor indeks bit yang telah terenkripsi.
f. Selesai.

2.5.3 Proses Ekstraksi

Proses ekstraksi pada Gambar 4 bertujuan untuk mengembalikan pesan ke bentuk semula sehingga dapat diketahui isinya. Masukkan proses ekstraksi berupa *chiperteks* vektor, kunci dan citra.



Gambar 4 Proses ekstraksi.

Langkah-langkah proses ekstraksi adalah sebagai berikut:
a. Memasukkan *input* berupa kunci, *chiperteks* vektor dan citra.
b. Mendekripsi vektor dengan kunci, hasil dekripsi berupa *plainteks* indeks bit.
c. Melakukan rekonstruksi pesan dengan mencocokan bit MSB citra berdasar vektor indeks bit.
d. Hasil *output* berupa pesan.
e. Selesai.

3. HASIL DAN PEMBAHASAN

Hasil penelitian diimplementasikan kedalam bentuk program aplikasi yang dibangun menggunakan bahasa pemrograman matlab R2009a kemudian dilakukan pengujian untuk pengamanan data.

3.1 Pengujian Citra

1) Proses *Embedding*

Langkah-langkah untuk melakukan proses *embedding* adalah sebagai berikut:

a. Memilih file pesan yang akan di-*embed*. Pada kasus ini dipilih sebuah file transfer. txt dengan isi:

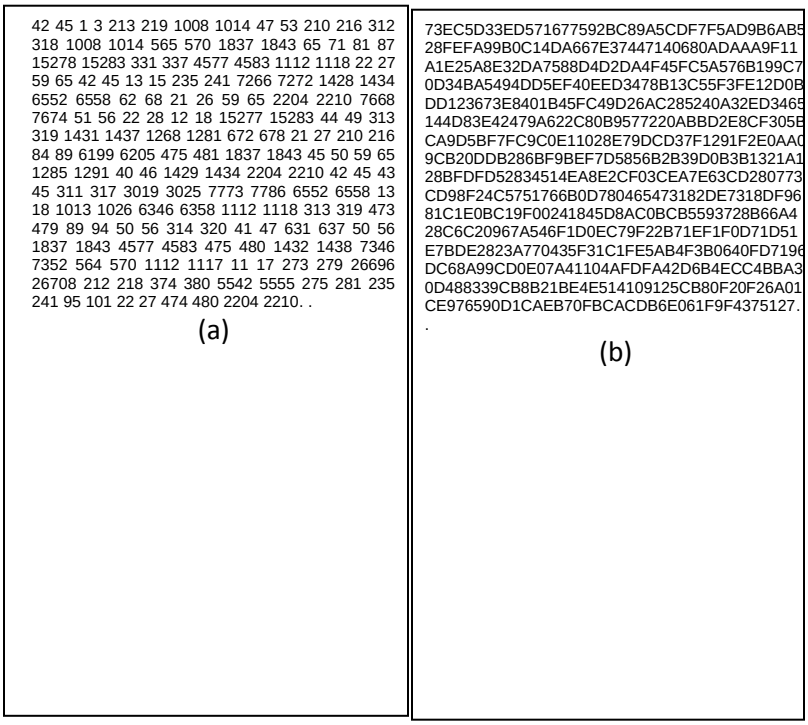
Transfer uang 50 juta via ATM.
Nomor rekening: 0123456
Password PIN: 9x8d7g
a. n. Budi Prasetiyo

b. Memilih citra sebagai *cover* file, misalnya Baboon.bmp yang terlihat pada Gambar 5.

Gambar 5 Citra *cover*.

c. Mengetik *password* sebagai kunci untuk enkripsi DES, misalkan kunci: 1234567.

d. Setelah proses*Embedding* dihasilkan keluaran berupa indeks bit terlihat pada Gambar 6a dan *chiperteks indeks* bit terlihat pada Gambar 6b.



Gambar 6 (a) Indeks bit, (b) Indeks bit terenkripsi.

2) Proses Ekstraksi

Pada pengujian proses ekstraksi, penulis akan mengembalikan pesan dari hasil vektor yang telah terenkripsi dengan melakukan ekstraksi file tersebut. Sedangkan untuk waktu ekseskusi pada citra hitam puih dapat dilihat pada Tabel 1 dan waktu ekseskusi pada citra *color* dapat dilihat pada Tabel 2.

Langkah-langkah untuk melakukan proses ekstraksi adalah sebagai berikut:

a. Memilih vektor file, vektor_Baboon. txt
b. Memilih citra sebagai *cover* file, yaitu Baboon.bmp yang dapat dilihat seperti pada Gambar 5.
c. Menginputkan kunci. Kunci harus sama seperti saat melakukan *embedding*, yaitu "1234567".
d. Melakukan ekstraksi.

Setelah proses ekstrak pesan berhasil kembali seperti semula, dengan keluaran:

Transfer uang 50 juta via ATM.
Nomor rekening: 0123456
Password PIN: 9x8d7g
a. n. Budi Prasetiyo

Tabel 1 Waktu ekseskusi pada citra hitam putih.

No	Citra	Embedding (detik)			Ekstraksi (detik)		
		Matching	Enkripsi	Total	Dekripsi	Rekonstruksi	Total
1.	Block.bmp	1,327	24,265	25,82	25,044	0,397	25,86
2.	Gradasi.bmp	0,374	31,53	32,06	33,386	0,321	34,37
	Rata-rata	0,850	27,897	28,940	29,215	0,359	30,115

Tabel 2 Waktu ekseskusi pada citra *color*.

No	Citra	Embed (detik)			Ekstrak (detik)		
		Matching	Enkripsi	Total	Dekripsi	Rekonstruksi	Total
1.	Lenna.bmp	0,165	12,451	12,71	12,621	0,184	13
2.	Pepper.bmp	0,157	10,152	10,40	11,872	0,183	12,24
3.	Jet.bmp	0,138	11,828	12,06	12,131	0,154	12,46
4.	Baboon.bmp	0,162	10,121	10,37	10,143	0,138	10,45
5.	Foto.bmp	0,199	15,108	15,41	15,318	0,160	15,72
	Rata-rata	0,164	11,932	12,19	12,417	0,164	12,774

Berdasar pengujian menunjukkan bahwa pada citra hitam putih diperoleh rata-rata proses *embedding* 28,94 dtk dengan lama waktu pencocokan bit 0,850 dtk dan enkripsi 27,897 dtk. Rata-rata proses ekstraksi yaitu 30,115 dtk dengan lama dekripsi 29,215 dtk dan rekonstruksi pesan 0,359 dtk. Sedangkan pada citra warna, rata-rata proses *embedding* adalah 12,19 dtk dengan lama waktu untuk pencocokan bit 0,164 dan enkripsi 11,932. Pada proses ekstraksi membutuhkan waktu rata-rata 12,774 dtk dengan lama waktu untuk dekripsi 12,417 dtk dan rekonstruksi pesan 0,1638 dtk.

3.2 Hasil Pengujian dengan Berbagai Ukuran Resolusi

Pengujian dengan berbagai ukuran citra mulai 512px, 256px, 128px dan 64px. Berdasar pengujian pada Tabel 3 menunjukkan bahwa semakin besar resolusi citra maka proses pencockan bit akan semakin lama. Rata-rata waktu pencocokan bit yang paling singkat yaitu “Baboon” (0,590 dtk), sedangkan pencocokan bit paling lama yaitu “Block” (2,022 dtk). Baboon memiliki variasi warna yang paling banyak, sedangkan “Block” hanya memiliki 2 variasi warna (hitam dan putih).

Tabel 3 Hasil pengujian dengan berbagai ukuran citra.

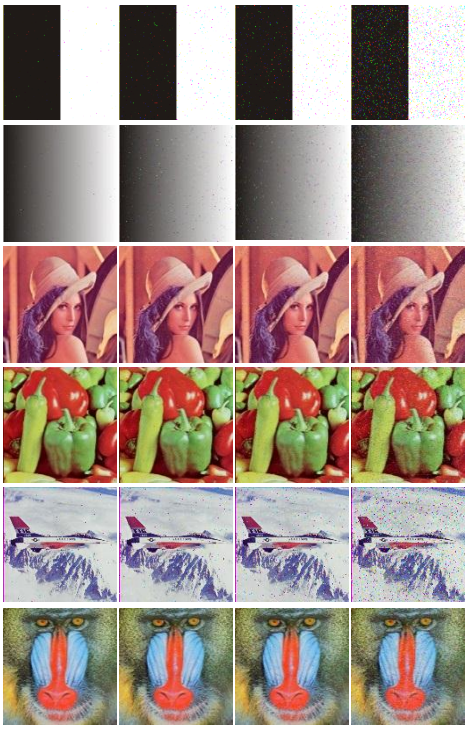
Citra	Resolusi (px)	Proses Embedding (dtk)			Proses Ekstraksi (dtk)		
		Matching	Enkripsi	Total	Dekripsi	Rekons-truksi	Total
"Block"	512 x 512	5,576	6,576	7,576	8,576	9,576	10,576
	256 x 256	1,706	30,165	32,133	29,587	0,409	30,703
	128 x 128	0,548	31,673	32,375	27,225	0,338	28,096
	64 x 64	0,256	24,033	24,417	23,873	0,320	24,594
	Rata-rata	2,022	23,112	24,125	22,315	2,661	23,492
"Gradation"	512 x 512	5,542	24,986	31,086	24,723	0,696	25,909
	256 x 256	1,726	31,963	33,967	29,819	0,414	30,961
	128 x 128	0,541	27,441	28,149	27,388	0,337	28,216
	64 x 64	0,256	24,033	24,417	23,873	0,320	24,594
	Rata-rata	2,016	27,106	29,405	26,451	0,442	27,420
"Lena"	512 x 512	1,866	9,681	11,860	11,786	0,179	12,197
	256 x 256	0,560	10,919	11,691	10,627	0,297	11,081
	128 x 128	0,193	11,026	11,340	10,308	0,213	10,715
	64 x 64	0,094	10,033	10,204	10,089	0,184	10,418
	Rata-rata	0,678	10,415	11,274	10,703	0,218	11,103
"Pepper"	512 x 512	1,861	9,637	11,988	9,460	0,568	10,170
	256 x 256	0,646	10,097	10,922	10,545	0,275	10,997
	128 x 128	0,239	10,214	10,587	10,265	0,237	10,998
	64 x 64	0,095	9,427	9,639	9,547	0,136	9,824
	Rata-rata	0,710	9,844	10,784	9,954	0,304	10,497
	512 x 512	2,038	12,186	14,713	10,564	0,181	10,900
"Jet"	256 x 256	0,567	11,341	12,095	11,099	0,267	11,574
	128 x 128	0,221	10,714	11,039	10,704	0,183	11,503
	64 x 64	0,089	10,734	10,902	10,575	0,179	10,946
	Rata-rata	0,729	11,244	12,187	10,736	0,203	11,231
	512 x 512	1,636	7,375	9,561	7,337	0,568	8,011
"Baboon"	256 x 256	0,468	7,067	7,715	7,057	0,247	7,421
	128 x 128	0,165	7,453	7,669	7,223	0,166	7,526
	64 x 64	0,090	8,645	8,625	8,358	0,163	8,641
	Rata-rata	0,590	7,635	8,393	7,494	0,286	7,900
	512 x 512	2,444	14,795	17,751	14,742	0,563	15,541
"Foto"	256 x 256	0,684	15,591	16,495	14,901	0,273	15,443
	128 x 128	0,278	15,497	15,874	15,161	0,223	15,620
	64 x 64	0,113	14,128	14,352	13,852	0,192	14,270

Citra	Resolusi (px)	Proses Embedding (dtk)			Proses Ekstraksi (dtk)		
		Matching	Enkripsi	Total	Dekripsi	Rekons-truksi	Total
	Rata-rata	0,880	15,003	16,118	14,664	0,313	15,219

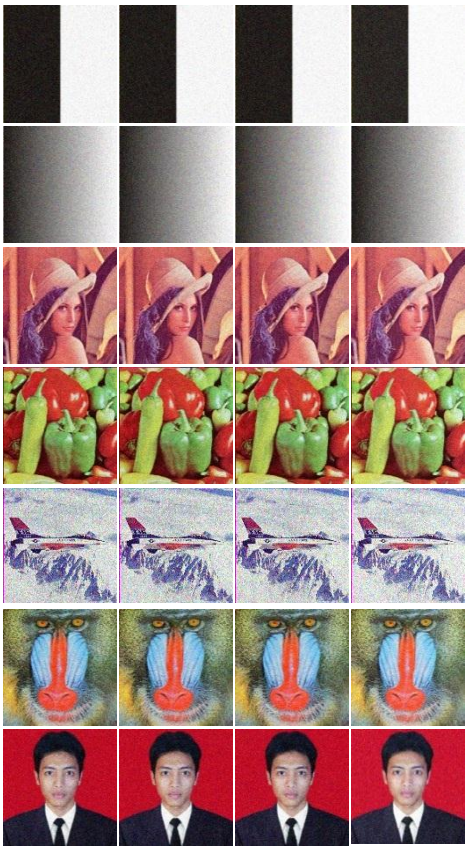
Secara umum dapat ditarik kesimpulan bahwa rata-rata proses *embedding* pada citra hitam putih 2 kali lebih lama dibanding citra warna. Proses pencocokan bit pada citra dengan variasi warna banyak lebih singkat, daripada citra yang memiliki variasi warna sedikit. Hal ini disebabkan pada citra variasi warna banyak memungkinkan banyak peluang terjadinya kesamaan susunan bit antara bit pesan dan bit citra, sehingga memakan waktu lebih singkat.

3.3 Pengujian dengan Pemberian Noise

Citra yang digunakan pada proses *embedding* selanjutnya diberi *noise' salt and pepper'* yang dapat dilihat pada Gambar 7dan *gaussian* dapat dilihat pada Gambar 8. Citra diberi *noise* '*salt and pepper'* dengan standar deviasi *d*= 0,001; 0,005; 0,01; 0,05. Pemberian citra dengan *noise 'gaussian'* menggunakan *mean* noldan standar deviasi *d*=0,001; 0,005; 0,01; 0,05. Citra yang telah diberi *noise* kemudian diuji pada proses ekstraksi untuk mengembalikan pesan.



Gambar 7 Citra dengan *noise 'salt and pepper'*, (dari kiri ke kanan, *d*= 0,001; 0,005; 0,01; 0,05).



Gambar 8 Citra dengan *noise 'Gaussian'* mean nol, (dari kiri ke kanan, d= 0,001; 0,005; 0,01; 0,05).

Tabel 4 Hasil pengujian dengan *Noise.*

Citra	Salt & pepper (d)				Gaussian (mean=0)			
	∂	MSE	PSNR	Pesan	∂	MSE	PSNR	Pesan
"Lenna"	0,001	0,00030	34,987	baik	0,10%	0,00307	52,3101	Rusak
	0,005	0,00140	28,435	baik	0,50%	0,00304	52,3080	Rusak
	0,01	0,00310	25,014	terbaca rusak	1%	0,00302	52,2815	Rusak
	0,05	0,01400	18,282	terbaca rusak	5%	0,00345	51,4609	Rusak
"Pepper"	0,001	0,00010	67,295	baik	0,10%	0,00328	52,4225	Rusak
	0,005	0,00483	50,094	terbaca rusak	0,50%	0,00329	52,4045	Rusak
	0,01	0,00094	57,320	terbaca rusak	1%	0,00333	52,3485	Rusak
	0,05	0,00483	50,094	terbaca rusak	5%	0,00404	51,3140	Rusak
"Baboon"	0,001	0,00013	67,773	baik	0,10%	0,00320	52,2707	Rusak
	0,005	0,00049	60,369	terbaca rusak	0,50%	0,00318	52,2517	Rusak
	0,01	0,00091	57,660	terbaca rusak	1%	0,00320	52,2316	Rusak
	0,05	0,00494	50,410	terbaca rusak	5%	0,00380	51,3988	Rusak
"Jet"	0,001	0,00014	61,398	baik	0,10%	0,00322	48,2885	Rusak
	0,005	0,00056	55,936	baik	0,50%	0,00319	48,3709	Rusak
	0,01	0,00123	52,807	baik	1%	0,00321	48,2902	Rusak
	0,05	0,00515	46,077	terbaca rusak	5%	0,00379	47,6589	Rusak
"Foto"	0,001	0,00010	66,863	baik	0,10%	0,00296	53,0124	Rusak
	0,005	0,00067	59,135	terbaca rusak	0,50%	0,00296	52,9414	Rusak
	0,01	0,00116	56,401	terbaca rusak	1%	0,00305	52,8322	Rusak
	0,05	0,00601	49,286	terbaca rusak	5%	0,00386	51,5197	Rusak
"Grad"	0,001	0,00009	67,871	baik	0,10%	0,00310	52,4209	Rusak
	0,005	0,00058	59,557	baik	0,50%	0,00309	52,4483	Rusak
	0,01	0,00108	56,889	baik	1%	0,00311	52,4256	Rusak
	0,05	0,00538	50,026	rusak	5%	0,00371	51,6761	Rusak
"Block"	0,001	0,00012	65,675	baik	0,10%	0,00220	54,0692	terbaca rusak
	0,005	0,00069	58,685	baik	0,50%	0,00221	54,0967	terbaca rusak
	0,01	0,00163	55,358	baik	1%	0,00216	54,1347	Rusak
	0,05	0,00745	48,624	baik	5%	0,00234	53,6779	terbaca rusak

Hasil rekonstruksi pesan pada citra hitam putih dengan penambahan *noise salt & pepper* tetap baik. Pesan dapat dibaca, kecuali 1 yang mengalami kerusakan. Pada citra warna sebagian besar mengalami kerusakan, kecuali citra "Jet" dan "Lenna" yang mengalami sedikit kerusakan dengan nilai MSE paling tinggi 0,014. Kerusakan pada citra warna terjadi mulai MSE 0,0067 pada citra "Foto" yang notabene memiliki variasi warna cukup sederhana. Penambahan *noiseGausian* membuat sebagian besar isi pesan rusak. Kerusakan mulai terjadi pada nilai MSE 0,00234. Hal ini disebabkan penambahan *noise* mempengaruhi nilai bit citra, sedangkan pencocokan bit mengambil pada indeks posisi bit yang tepat. Sehingga hasil rekonstruksi pesan akan menghasilkan pesan yang berubah pula.

4. SIMPULAN

Proses steganografi pada penelitian ini meliputi pencocokan bit dan rekonstruksi, sedangkan proses kriptografimeliputi enkripsi dan dekripsi. Kombinasi steganografi dan kriptografi pada penelitian ini dapat digunakan untuk pengamanan data. Masukan data pesan, citra dan kunci. Hasilnya adalah *chipterteks*. Untuk mengetahui isi pesan dibutuhkan kunci dan citra yang sama.

Citra *grayscale* maupun citra warna dapat digunakan sebagai media *cover*. Kecuali citra dengan warna 100% hitam atau 100% putih, karena citra tersebut terdiri dari susunan bit yang *homogen*. Semua nilai bit pada citra 100% bernilai 0 (nol) dan citra 100% hitam bernilai 1 (satu), padahal susunan bit pesan bervariasi antara 0 dan 1, sehingga pencocokan bit tidak akan menemukan hasil.

Penambahan *noise* pada citra mengakibatkan sebagian isi pesan berubah, dengan tingkat perubahan yang bervariasi. Pada citra hitam putih tidak terjadi perubahan yang berarti, namun pada citra warna terjadi banyak perubahan isi pesan. Kerusakan terjadi pada penambahan *salt and peper* mulai nilai MSE 0,0067 dan pada *gaussian* mulai terjadi kerusakan pada MSE 0,00234.

Proses pencocokan bit dengan variasi wana yang banyak lebih singkat dibanding citra dengan variasi warna yang lebih sedikit. Kelebihan metode ini diantaranya kualitas citra tidak dirubah. Dari segi pengamanan, seandainya vektor indeks bit tidak di enkripsi sebenarnya sudah cukup untuk mengamankan data. Karena untuk mengembalikan indeks menjadi pesan dibutuhkan citra yang tepat, jika tidak maka hasilnya tidak akan terbaca. Peneliti lain bisa melakukan enkripsi pada citra terlebih dahulu sebelum dilakukan pencocokan bit. Operasi pada citra juga dapat dilakukan agar citra 100% hitam maupun 100% putih dapat dijadikan sebagai *cover*. Disamping itu peneliti lain bisa memodifikasi *output* chiperteks menjadi *stego image*.

REFERENSI

[1] Kautzar, M. G., *Studi Kriptografi Mengenai Triple DES dan AES*, ITB, Bandung, 2007.
[2] Provos, N. & Honeyman, P., *Hide and Seek: An Introduction to Steganography*, IEEE Security & Privacy Vol. 1(3), 32-44, 2003.
[3] Schneier, B., *Applied Cryptography 2nd Edition*, Wiley & Sons. Inc., New York, 1996.
[4] Krenn, R., *Steganography and Steganalysis*, Whitepaper, 2004.

[5] Raphael., Sundaram, A. J. & Sundaram, V., *Cryptography and Steganography – A Survey*, *International Journal Comp*. Tech. Applied Vol. 2 (3), 626-630, 2011.
[6] Chan, C. K. & L. M. Cheng., *Hiding Data in Images by Simple LSB Substitution*, Pattern Recognition Vol. 37(3), 469–474, 2004.
[7] Challita, K. & Farhat, H., *Combining Steganography and Cryptography: New Directions*, International Journal on New Computer Architectures and Their Applications (IJNCAA) 1(1), 199-208, 2011.
[8] Pressman, R. S., *Software Engineering*: *A Practitioner's Approach, 6th Edition*, The McGraw-Hill Companies, Inc, Singapore, 2001.
[9] Cormen, T. H., Leiserson C. E., Rivest R. L. & Stein D., *Introduction to Algorithms*, Third Edition, The MIT Press, England, 2009.
[10] Kekre, H. B., Archana A. & Pallavi N. H., *Comparison between the basic LSB Replacement Technique and Increased Capacity of Information Hiding in LSB's Method for Images*, International Journal of Computer Applications (0975 – 8887) Vol . 45 (1), 33-38, 2012.
[11] Menezes A. J., Oorschot, P. C. & Vanstone, S. A., *Handbook of Applied Cryptography*, CRC Press, Boca Raton, New York, 1996.
[12] Munir, R., *Pengantar Kriptografi*, ITB, Bandung, 2006.
[13] Narayana, S. & Prasad, G., *Two New Approaches for Secured Image Steganography Using Cryptographic Techniques and Type Conversions*, Signal & Image Processing: An International Journal (SIPIJ) Vol. 1(2), 60-73, 2010.
[14] Seth, D., Ramanathan, L. & Pandey, A., *Security Enhancement: Combining Cryptography and Steganography*, International Journal of Computer Applications (0975 – 8887) Vol. 9 (11), 3-6, 2010.
[15] Sharp, T., *An implementation of Key-based Digital Signal Steganography*,Proc. Information Hiding Workshop Vol. 2137, Springer LNCS, 13–26, 2001.