JAVA Journal of Electrical and Electronics Engineering
Volume 10, Number 2, October 2012

Penulisan Pesan Tersembunyi Pada Citra JPEG dengan Metode F5

Eko Pramunanto[1],Muhtadin[2], Yanu Perwira Adi Putra, dan Ahmad Zaini
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember (ITS)
Surabaya, Indonesia
ekopram@te.its.ac.id[1], muhtadin@ee.its.ac.id[2]

Abstract*— Komunikasi digital rentan terhadap pencurian informasi secara langsung ataupun tidak langsung. Hal ini dapat diatasi dengan beberapa cara, salah satunya dengan teknik steganografi. Steganografi dapat digunakan untuk menyembunyikan pesan teks pada sebuah citra dijital khususnya JPEG. Salah satu metode steganografi yang umum adalah F5. Metode F5 merupakan metode yang menyisipkan bit data pesan ke dalam bit koefisien DCT hasil kuantisasi yang terlebih dahulu telah dipermutasi. Metode F5 mempunyai keakuratan penyembunyian pesan teks dan keakuratan descriptory cukup baik, sehingga dapat dimanfaatkan untuk keperluan perlindungan data atau informasi rahasia. Pada penelitian ini telah dilakukan implementasi metode F5 untuk menyisipkan pesan kedalam citra JPEG, hasil implementasi kemudian dilakukan pengujian kapasitas pesan yang disisipkan terhadap banyaknya variasi warna pada file citra JPEG. Selain itu dilakukan penyisipan pesan berupa pesan ekstensi doc dan citra lain (JPEG) kedalam file citra carier. Dari hasil pengujian didapatkan bahwa semakin banyak pesan yang disisipkan kedalam file citra akan mengakibatkan banyaknya perubahan bit citra tersebut. Kualitas kompresi citra yang akan disisipi pesan sangat berpengaruh terhadap keberhasilan proses pemisahan pesan, semakin tinggi kualitas kompresi maka semakin rendah prosesntase keberhasilan pemisahan pesan dari citra pembawa.

Keywords—JPEG, Metode F5, Steganografi

I. PENDAHULUAN

Sesuatu yang mendukung dalam pertukaran eamanan dalam teknologi komunikasi merupakan informasi. Dengan semakin berkembangnya dunia komunikasi, maka semakin banyak pertukaran informasi yang terjadi dengan berbagai media komunikasi. Dalam melakukan beberapa pertukaran informasi perlu adanya perlindungan terhadap informasi yang dikirim. Ada beberapa teknik yang dapat dilakukan untuk melindungi informasi tersebut, salah satunya dengan steganografi.

Steganografi adalah suatu teknik yang mempelajari penyembunyian informasi rahasia di didalam informasi induk sehingga keberadaannya sulit untuk diketahui oleh pihak yang tidak berkepentingan.

Pada Teknik Steganografi banyak format digital yang dapat dijadikan media penyembunyian pesan, antara lain :

- Format image : bitmap (bmp), gif, pcx, jpeg, dll.
- Format audio : wav, voc, mp3, dll.
- Format lain : teks file, html, pdf, dll.

Media-media yang didukung sudah sangat familiar dalam media pertukaran informasi dijital, khususnya media JPEG. Hal ini merupakan suatu kelebihan dari teknik steganografi.

Kelebihan media gambar sebagai wadah penyembunyian pesan adalah gambar merupakan hal yang biasa dalam pertukaran informasi di dunia digital. Salah satu format gambar yang terkenal adalah JPEG. Kebanyakan orang tidak akan menyadari, bahwa didalam gambar ada suatu informasi yang tersembunyi, dan hal ini menjadi salah satu kelebihan.

Banyak metode yang dapat digunakan untuk menerapkan teknik steganografi. Salah satu metode yang ada adalah F5. Pada penelitian ini, penulis menggunakan metode F5 untuk menerapkan teknik steganografi, dimana algoritma F5 menyisipkan bit informasi ke dalam bit koefisien DCT hasil kuantisasi yang terlebih dahulu telah dipermutasi.

II. URAIAN PENELITIAN

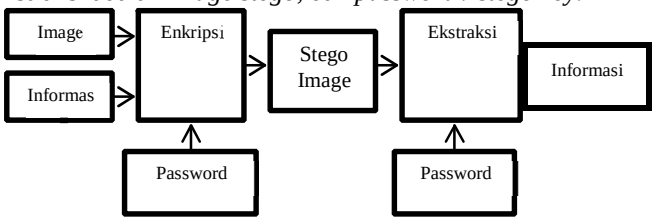
Perancangan algoritma penyisipan ataupun extraksi file pada metode F5 merupakan bersifat spesifik, yang mengartikan penyisipan atau ekstraksi hanya berlaku pada *file image.* Metode F5 merupaka metode yang mampu menjaga sifat-sifat histogram DCT dengan cukup baik dan cukup unggul dalam hal kapasitas [1]. Metode F5 berbeda dengan metode yang lainnya mengenai penyisipan bit, dimana koefisien yang disisipi bukan mengalami penimpaan bit, melainkan pengurangan nilai.

Awal mulanya akan dilakukan penampungan seluruh koefisien DCT dalam satu *variable array*,kemudian dilakukan operasi *permutative straddling* terhadap seluruh koefisien. Setelah tahap penyisipan selesai, maka koefisien DCT yang telah mengalami perubahan akan dilakukan proses *Huffman Code.*

Pada bagian ekstraksi dilakukan penggalian informasi gambar yang diawali dengan proses Huffman code sehingga akan didapatkan koefisien DCT. Koefisien yang didapat, akan diproses kembali dengan permutative straddling. Koefisien yang diperoleh, kemudian dilakukan pengecekan satu demi satu sehingga akan mendapatkan susunan bit data tersembunyi

Perancangan metodologi pada penelitianini, digambarkan pada gambar 1, terdapat dua blok besar proses, antara lain Enkripsi dan Ekstraksi. Proses enkripsi merupakan proses penyisipan informasi kedalam suatu *image* / citra. Input dari proses enkripsi terdiri dari *image* / citra, informasi dan *password / stego key*. Proses ekstraksi merupakan proses untuk mendapatkan informasi apa yang tersembunyi didalam *image*

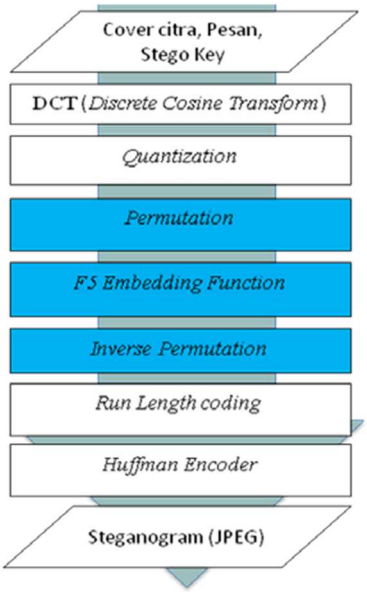
yang telah mengalami proses enkripsi. *Input* dari proses ekstraksi adalah *image stego*, dan *password / stego key*.



Gambar 1. Metodologi

A. *Tahap Enkripsi*

Terdapat beberapa tahap dalam proses penyisian file pada media image jpeg. Secara rinci dapat dilihat pada blok diagram gambar 2.



Gambar 2. Enkripsi

Enkripsi data merupakan tahap yang memanfaatkan proses kompresi dari JPEG. Proses kompresi akan berhenti sementara pada saat proses kuantisasi selesai. Dari proses tersebut akan didapatkan seluruh koefisien DCT yang telah melalui tahap kuantisasi dan siap untuk tahap enkripsi dengan metode F5.

Input dari enkripsi data adalah *image*, pesan dan *password*. *Image* sebagai *input*, yang akan dilakukan proses DCT untuk mendapatkan nilai koefisien DCT nya. Tetapi sebelum proses DCT, akan dilakukan konversi gambar dari RGB ke YUV dan pembagian blok piksel dengan ukuran 8x8.

Proses kuantisasi adalah pembagian antara setiap koefisien DCT dengan koefisien dari tabel kuantisasi, dan kemudian dilakukan pembulatan. Setelah tahap kuantiasis selesai, maka akan memasuki tahap inti dari proses enkripsi data.

Enkripsi data menggunakan metode F5, pada intinya memanfaatkan nilai nilai koefisien yang telah dihasilkan dari proses kuantisasi. Sebelum memasuki tahap penyisipan, urutan koefisien akan dilakukan pengacakan dengan metode *permutative straddling* dengan *password* sebagai *input*. *permutative straddling* adalah proses mendapatkan bilangan acak yang dihasilkan oleh *randomgenerator*. Bilangan yang dihasilkan akan digunakan sebagai permutasi yang berfungsi untuk mengacak urutan koefisien.

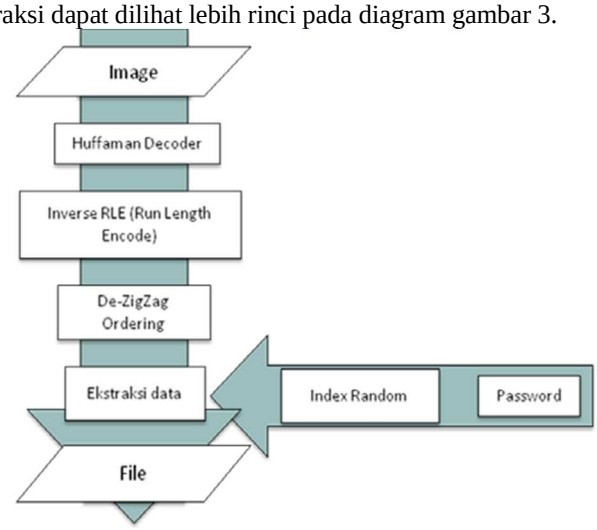
Informasi penting yang harus diambil dari data yang disisipkan adalah ukuran dari data yang akan disisipkan, terdiri dari 3 *byte* data yang memungkinkan penyisipan sampai ukuran 16.777.215 *byte*.

Tahap berikutnya adalah proses penyisipan file dengan metode F5. Semua koefisien DCT akan dicek satu persatu sesuai dengan urutan yang telah diacak oleh permutasi. Koefisien yang bernilai 0 dan nilai DC tidak dilakukan pemrosesan penyisipan. Proses penyisipan akan lebih rinci dijelaskan pada sub bab *matriks encoding*.

Inverse permutasi atau pengembalian urutan permutasi ke urutan awal, merupakan tahap setelah proses penyisipan selesai. Koefisien yang telah mengalami perubahan akan dilanjutkan ke proses kompresi JPEG selanjutnya, dengan melakukan RLC dan *Huffman code*

B. *Tahap Dekripsi*

Proses dekripsi merupakan proses untuk menggali informasi yang tersembunyi didalam imagesteganografi. Tahapan proses ekstraksi dapat dilihat lebih rinci pada diagram gambar 3.



Gambar 3. Dekripsi

Proses dekripsi memerlukan inputimage JPEG yang telah terenkripsi dan password. Tahap pertama yang dilakukan adalah mendapatkan koefisien DCT yang berasal dari image JPEG. Koefisien DCT ini diperoleh ketika proses zigzag ordering selesai.

Nilai-nilai koefisien DCT yang didapatkan akan menjadi input dari tahap ekstraksi. Tahap selanjutnya adalah mendapatkan urutan acak koefisien DCT yang sesuai dengan urutan acak pada saat proses penyisipan, untuk meperoleh urutan yang benear menggunakan permutative straddling. Urutan acak yang dihasilkan permutative straddling berdasarkan pada password yang dimasukkan, apabila password yang dimasukkan berbeda, maka tidak akan mendapatkan urutan yang sebenarnya

C. Permutative Straddling

F5 menggunakan mekanisme *straddling* yang berfungsi mengacak letak semua koefisien DCT terlebih dahulu dengan menggunakan permutasi [11]. F5 kemudian menyisipkan data steganografi menurut urutan permutasi tersebut dan mengirimkannya kepada tahap *Huffman Code* dalam urutan sebenarnya sesudah penyisipan selesai, urutan permutasi itu sendiri didapat dari *password* yang dimasukkan oleh *user*. Dengan *password* yang benar, *user* lain akan mendapatkan urutan permutasinya dan dapat membaca data yang tersembunyi dengan benar.

Banyak Metode untuk menghasilkan bilangan random yang berfungsi untuk menggenerate bilangan acak yang digunakan untuk pengacakan index, antara lain yang digunakan adalah *Blum Blum Shub* (BBS). *Blum Blum Shub* (BBS) adalah sebuah *pseudorandomnumber generator* yang dibuat pada tahun 1986 oleh Lenore Blum, Manuel Blum, dan Michael Shub. BBS mengambil bentuk sebagai berikut:

$$X_{n+1} = (X_n)^2 mod\ M \quad \text{.....(1)}$$

di mana :

X_{n+1} = bilangan acak ke-n+1 dari deretnya
X_n = bilangan acak sebelumnya
M = modulus

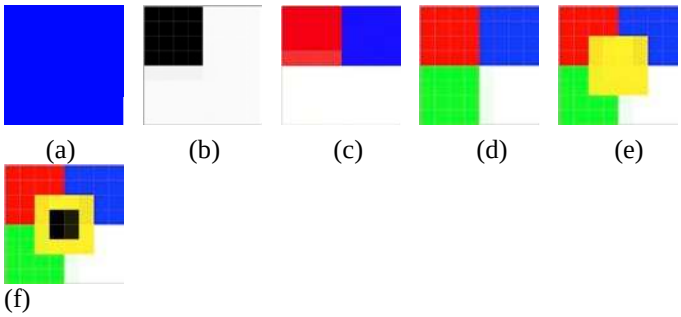
D. Matriks Encoding

Matrix encoding adalahpenghitungan kode Hamming yang sesuai (1, (2k-1), k) dengan menghitug ukuran blok pesan k dari panjang pesan dan jumlah koefisien-koefisien non DC yang tidak nol [12]. Kode Hamming (1, 2k-1, k) merupaka pengkodean pesan rahasia k-bit dari m kata pesan kedalam nbit kata kode a dengan n=2k-1. Kode Hamming dapat merecover dari single bit yang error dalam kode.

III. HASIL PERCOBAAN

A. Analisa Kemampuan Image Untuk Penyimpanan

Pada pengujian A, bertujuan untuk menganalisa apakah keberagaman warna berpengaruh terhadap kapasitas. Analisa ini menggunakan image dengan berbagai dimensi piksel. Pembeda dari masing masing image adalah banyaknya jumlah keaneragaman warna yang digunakan.



(a) (b) (c) (d) (e) (f)

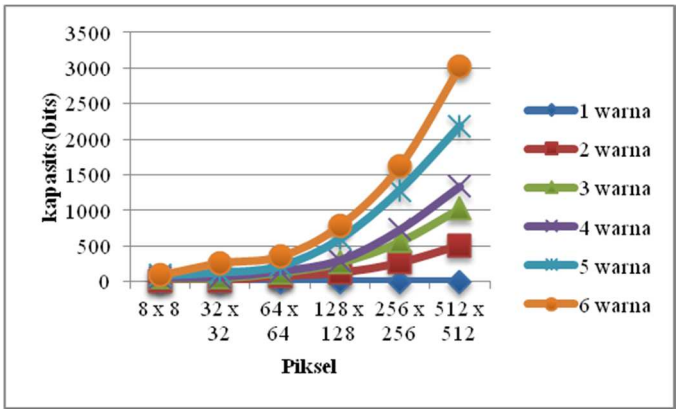
Gambar 4 Percobaan kemampuan image penyisipan
(a) satu warna; (b) dua warna; (c) tiga warna; (d) empat warna; (e) lima warna; (f) enam warna

Image yang memiliki beberapa variasi warna dengan berbagai ukuran pixel. Ukuran Image yang diuji antara lain 8 x 8 pixel, 32 x 32 pixel, 64 x 64 pixel, 128 x 128 pixel, 256 x 256 pixel, 512 x 512 pixel. File yang disisipkan sebesar 1 Byte.

Pada Tabel 1 ditampilkan hasil kemampuan image dalam kemampuan kapasitas. Semakin besar image maka semakin besar pula kapasitas yang dihasilkan, demikian juga dengan keberagaman warna semakin banyak, maka kapasitas yang dihasilkan juga semakin meningkat.

Tabel 1 Perbandingan ukurang piksel dan banyak warna

Ukuran Piksel	Gambar pada Pengujian (bits)					
	a	b	c	d	e	f
8 x 8	0	14	57	58	86	91
32 x 32	0	19	54	64	129	265
64 x 64	0	83	130	144	217	357
128 x 128	0	127	270	298	598	790
256 x 256	0	266	564	728	1285	1625
512 x 512	0	506	1030	1340	2183	3019



Gambar 5 Grafik perbandingan ukuran piksel dan banyak warna

Pada Tabel 1 menunjukkan perbandingan ukuran piksel dan keberagaman warna. Hasil yang ditunjukkan adalah kapasitas maksimal yang diperoleh untuk media penyisipan. Tabel tersebut menampilkan data dalam bentuk nilai besar bits. Pada Gambar 5 ditampilkan grafik peningkatan terhadap ukuran piksel dan jumlah warna, dapat disimpulkan kapasitas akan semakin naik secara non linier, kenaikan hampir dikatakan dua kali lipat terhadap penambahan setiap warna.

B. Analisa Kapasitas Image

Pada analisa kapasitas, terdapat 2 *image* yang digunakan. *Image* tersebut memiliki ukuran luas piksel yang sama tetapi dengan intensitas warna yang berbeda. 2 *image* tersebut ditunjukkan pada Gambar 6.

(a)

(b)

Gambar 6 Image uji coba steganografi (a) Logo ITS; (b) Rumput berembun

Gambar 6 (a) dan (b) merupakan gambar yang berukuran 640x346 pixel. Kedua gambar akan diuji kemampuannya dalam hal kapasitas dengan berbagai file sisipan yang bervariasi besar kapasitasnya.

Tabel 2 Perbandingan kemampuan kapasitas

Nama *File*	Ukuran File Text (*Byte*)	Nama *Image*	Ukuran File *Image* (*Byte*)	Setelah Penyisipan (*Byte*)	Hasil Ekstraksi File (*Byte*)
text1.txt	316	Logo ITS	50306	27359	316
text2.txt	532	Logo ITS	50306	27197	532
text3.txt	1291	Logo ITS	50306	26558	1291
text4.txt	3022	Logo ITS	50306	25317	25317
text1.txt	316	Rumput	63245	32339	316
text2.txt	532	Rumput	63245	32087	532
text3.txt	1291	Rumput	63245	31281	1291
text4.txt	3022	Rumput	63245	29463	3022

Dari tabel 2 terlihat walaupun memilki ukuran piksel yang sama yakni 640 x 346 pixel, tidak menjamin kemampuan dalam hal besar kapasitas memiliki kemampuan yang sama. Pada Gambar 6 (a) yang disisipkan file text4.txt tidak mampu menyimpan file secara keseluruhan. Sedangkan pada Gambar 6 (b) mampu menyimpan file text4.txt dengan sempurna.

C. Analisa MSE dan PSNR

Analisa berikutnya akan diuji dengan membandingkan nilai maksimum dari sinyal yang diukur dengan besarnya derau yang berpengaruh pada sinyal tersebut, pembandingan ini dikenal dengan PSNR. Nilai PSNR diukur dalam satuan *decibel*. PSNR dapat dihitung dengan persamaan 4.1.

$$PSNR = 10 \log \left[\frac{255^2}{MSE}\right] \quad \text{.....(2)}$$

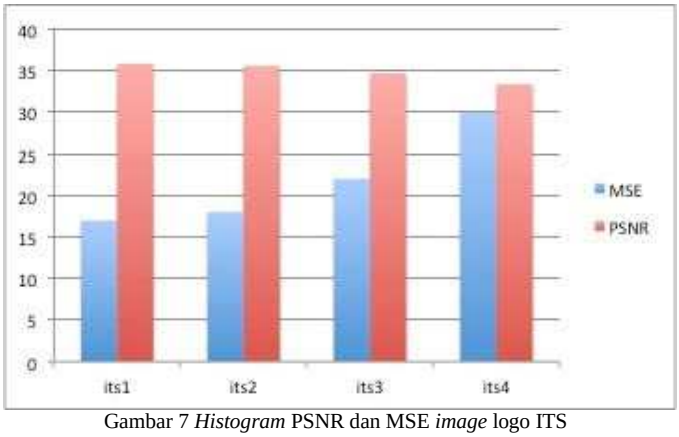
$$MSE = \sqrt{\left(\frac{1}{N_1 N_2} \sum_{j=1}^{N_1} \sum_{i=1}^{N_2} (P_{ij} - Q_{ij})^2\right)} \quad \text{.....(3)}$$

Keterangan :
N_1 dan N_2 = Dimensi Piksel
P dan Q = Gambar 1 dan Gambar 2

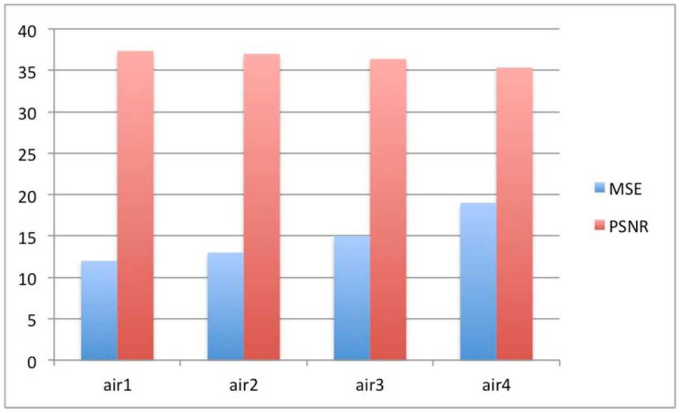
Tabel 3 Tabel hasil MSE dan PSNR

Nama *File*	Ukuran File Text (*Byte*)	Nama *Image*	Ukuran File *Image* (*Byte*)	MSE	PSNR
text1.txt	316	Logo ITS	50306	17	35,8263143
text2.txt	532	Logo ITS	50306	18	35,5780785
text3.txt	1291	Logo ITS	50306	22	34,7065769
text4.txt	3022	Logo ITS	50306	30	33,3595910
text1.txt	316	Rumput	63245	12	37,3389911
text2.txt	532	Rumput	63245	13	36,9913698
text3.txt	1291	Rumput	63245	15	36,3698910
text4.txt	3022	Rumput	63245	19	35,3432675

Dalam bentuk histogram akan lebih jelas perbedaannya dalam penurunan kualitas gambar, pada Gambar 7 dan Gambar 8.



Gambar 7 *Histogram* PSNR dan MSE *image* logo ITS



Gambar 8 Histogram PSNR dan MSE image rumput berembun

Dari analisa PSNR dan MSE dapat dilihat, semakin banyak data yang disisipkan, maka akan semakin memperburuk kualitas citra. Penurunan kualitas citra pada percobaan PSNR masi dalam ambang batas baik. Katagori baik disini untuk perubahan PSNR mulai dari 30db – 40db. Sedangkan hasil dari percobaan menghasilkan nilai diatas 30db.

Untuk MSE, semakin kecil nilai yang dihasilkan maka dikatakan sedikit perbedaan. Terlihat dari histogram pada Gambar 7 dan Gambar 8 perubahan histogram MSE dikatakan baik diantara range 20 db kebawah. Terlihat hasil yang kurang baik pada gambar Logo ITS, logo its sisipan text3.txt dan logo its sisipan text4.txt terlihat hasil telah melampaui batas.

*D. Pengujian Image Kompleks Terhadap Penyisipan Data Berekstensi *.TXT*

Pengujian berikutnya akan dilakukan terhadap beberapa image kompleks dengan beragam variasi warna. File yang disisipkan hanya satu file typetext dengan besar file 449 byte. Berikut ditampilkan hasil pengujiannya pada Tabel 4.

Tabel 4 Pengujian *image* kompleks

Gambar	Besar Data (byte)	Dimensi piksel	Besar sisipan text(byte)	besar data hasil stegano (byte)	Nilai MSE	Nilai PSNR
1	103.826	1024 × 767	449	55.258	3	43,3595
12	117.481	1024 × 767	449	63.875	11	37,7168
11	145.807	1024 × 767	449	82.020	7	39,6798
6	156.992	1024 × 767	449	73.593	8	39,0999
5	161.067	1024 × 767	449	82.831	4	42,1102
10	186.576	1024 × 767	449	115.914	18	35,578
8	188.427	1024 × 767	449	96.803	13	36,9913
9	192.598	1024 × 767	449	114.458	8	39,0999
13	227.416	1024 × 767	449	127.157	15	36,3698
4	291.049	1024 × 767	449	151.714	20	35,1205
3	322.109	1024 × 767	449	186.414	30	33,3595
2	381.755	1024 × 767	449	212.692	29	33,5068
7	406.959	1024 × 767	449	240.395	45	31,5986

Tabel 4 menunjukkan mulai dari data image yang berukuran kecil sampai besar tidak selalu menjamin sebuah kapasitas penyimpanan data tersembunyi semakin besar, hal ini tergantung dari intensitas warna yang ada. Semua data dikatakan berhasil disisipkan karena semua imagehost mampu dalam hal kapasitas.

E. Pengujian Image Kompleks Terhadap Penyisipan Data Image

Pengujian berikutnya akan dilakukan pengujian terhadap beberapa *image* yang komplek dengan beragam variasi warna. File yang disisipkan hanya satu *file typeimage* JPEG dengan besar *file* 19595 *byte* dan ukuran piksel 320 x 173, *image* yang disisipkan ditunjukkan pada Gambar 4.1. Berikut ditampilkan hasil pengujiannya pada Tabel 6.

Gambar 9 Image sisipan

Tabel 6 Pengujian penyisipan dengan *image*

Nama	Besar Data (byte)	Dimensi piksel	Besar sisipan gambar (byte)	Dimensi piksel sisipan	besar data hasil stegano (byte)	keberhasilan ekstrak data
1	103.82	1024 × 767	19.595	320 × 173	46.281	gagal
12	117.48	1024 × 767	19.595	320 × 184	53.548	gagal
11	145.80	1024 × 767	19.595	320 × 183	69.513	gagal
6	156.99	1024 × 767	19.595	320 × 178	61.776	gagal
5	161.06	1024 × 767	19.595	320 × 177	71.567	gagal
10	186.57	1024 × 767	19.595	320 × 182	99.110	gagal
8	188.42	1024 × 767	19.595	320 × 180	82.443	gagal
9	192.59	1024 × 767	19.595	320 × 181	96.360	gagal
13	227.41	1024 × 767	19.595	320 × 185	107.91	gagal

4	291.04	1024 × 767	19.595	320 × 176	130.93	gagal
3	322.10	1024 × 767	19.595	320 × 175	165.87	berhasil
2	381.75	1024 × 767	19.595	320 × 174	193.26	berhasil
7	406.95	1024 × 767	19.595	320 × 179	224.08	berhasil

Tabel 6 menghasilkan ada beberapa image host yang tidak mampu untuk disisipi secara sempurna, dikarenakan kapasitas memang kurang atau tidak mencukupi. Pada Tabel 7 ditunjukkan hasil ekstraksi data image.

Tabel 7 Hasil ekstraksi *image*

Nama Gambar	Besar Data (byte)	Dimensi piksel	Hasil Ekstrak	Besar File (byte)
1	103.826	1024 × 767		4.990
12	117.481	1024 × 767		6.966
11	145.807	1024 × 767		8.947
6	156.992	1024 × 767		7.795
5	161.067	1024 × 767		9.065
10	186.576	1024 × 767		13.628
8	188.427	1024 × 767		11.017
9	192.598	1024 × 767		13.501
13	227.416	1024 × 767		15.144
4	291.049	1024 × 767		18.442
3	322.109	1024 × 767		19.595
2	381.755	1024 × 767		19.595
7	406.959	1024 × 767		19.595

Pada Tabel 8 ditunjukkan tingkat keberhasilan yang di hasilkan, semakin tinggi *imagehost* akan diikuti peningkatan kemampuan kapasitas.

Tabel 8 Persentasi Keberhasilan

Nama	Besar File Image Host (Byte)	Besar File Ekstrak (Byte)	Persentasi Keberhasilan (%)
1	103.826	4990	25,46568002
12	117.481	6966	35,54988517
11	145.807	8947	45,65960704
6	156.992	7795	39,78055626
5	161.067	9065	46,26180148
10	186.576	13628	69,54835417
8	188.427	11017	56,22352641
9	192.598	13501	68,90022965
13	227.416	15144	77,28502169
4	291.049	18442	94,11584588
3	322.109	19595	100
2	381.755	19595	100
7	406.959	19595	100

F. Pengujian Kualitas Kompresi JPEG terhadap Kemampuan Kapasitas Penyisipan

Pengujian berikutnya akan dilakukan pengujian terhadap satu *image* dengan besar *file image* 63,245 *byte* berukuran 640 x 346 piksel. *Image* tersebut ditunjukkan pada Gambar 10 (a) dan untuk *image* sisipannya adalah pada Gambar 10 (b). Proses penyisipan akan dilakukan dengan perubahan kualitas kompresi JPEG. Perubahan kualitas dilakukan pada saat proses kuantisasi. Penentuan tingkat kualitas kompresi berkisar antara 10-100, dimana angka 10 menunjukkan bahwa kualitas citra

yang rendah dengan kompresi yang tinggi, sedangkan angka 100 menunjukkan bahwa kualitas yang dimiliki citra bernilai tinggi namun tingkat kompresi rendah. Proses penentuan kualitas ini dengan melakukan perubahan terhadap tabel kuantisasi. Hasil dari perubahan kualitas kompresi terhadap proses embedding data akan ditunjukkan pada tabel 4.9.

(a)

(b)

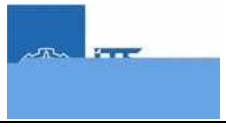
Gambar 10 Image Steganografi (a) Image Host; (b) Image Sisipan

Tabel 9 Kompresi dengan perbedaan kualitas dan dienkripsi

Kualitas	Besar Data Image (byte)	Dimensi Piksel	Hasil Steganografi	Besar Data Stegano (Byte)
10	63.245	640×346		7.464
20	63.245	640×347		10.516
30	63.245	640×348		13.172
40	63.245	640×349		15.432
50	63.245	640×350		17.525
60	63.245	640×351		20.044
70	63.245	640×352		23.358
80	63.245	640×353		29.341
90	63.245	640×354		46.866
100	63.245	640×355		106.481

Pada Tabel 9 menunjukkan hasil kompresi juga mempengaruhi kualitas *image* menjadi lebih buruk, dan besar data steganografi menjadi lebih kecil. Data steganografi tersebut akan diekstrak dan ditampilkan pada Tabel 10.

Tabel 10 Ekstraksi terhadap kualitas

Quality	Besar Data Image Stego (byte)	Hasil Ekstrak	Besar data Hasil Ekstrak (Byte)
10	7.464	Tidak dapat ditampilkan	504
20	10.516	Tidak dapat ditampilkan	1.008
30	13.172	Tidak dapat ditampilkan	1.426
40	15.432	Tidak dapat ditampilkan	1.774
50	17.525	Tidak dapat ditampilkan	2.092
60	20.044	Tidak dapat ditampilkan	2.469
70	23.358		2.958
80	29.341		3.761
90	46.866		6.146
100	106.481		10.829

Terlihat pada Tabel 10, semakin rendah kualiats kompresi dan kualitas gambar semakin baik, hal ini akan diikuti dengan kemampuan *image* dalam kemampuan kapasitas penyimpanan untuk informasi tersembunyi. Kualitas kompresi sangat berpengaruh terhadap kapasitas yang dihasilkan. Pada Tabel 11 ditunjukkan tingkat keberhasilan hasil ekstraksi.

Tabel 11 Persentasi keberhasilan

Quality	Besar File Stegano (Byte)	Besar File Ekstrak (Byte)	Persentasi Keberhasilan (%)
10	7.464	504	2,572084715
20	10.516	1008	5,144169431
30	13.172	1426	7,277366675
40	15.432	1774	9,053329931
50	17.525	2092	10,67619291
60	20.044	2469	12,6001531
70	23.358	2958	15,09568768
80	29.341	3761	19,19367186
90	46.866	6146	31,36514417
100	106.481	10829	55,26409798

Tabel 11 menunjukkan tingkat kualitas kompresi data sangat berpengaruh terhadap prosentasi keberhasilan ekstraksi data. semakin sempurna kompresi data atau dengan kualitas 10 maka prosentasi keberhasilan hanya 2,5 % atau sangat minim. Sedangkan dengan kualitas kompresi 100, tingkat keberhasilan hanya 55 %. Tingkat keberhasilan ini juga dipengaruhi dari kemampuan *image* dalam menghasilkan kapasitas steganografi misalnya terhadap keberagaman warna dan besar piksel.

Keberhasilan hanya 2,5 % atau sangat minim. Sedangkan dengan kualitas kompresi 100 tingkat keberhasilan hanya 55 %. Tingkat keberhasilan ini juga dipengaruhi dari kemampuan image dalam menghasilkan kapasitas steganografi misalnya terhadap keberagaman warna dan besar piksel.

IV. KESIMPULAN DAN SARAN

A. *Kesimpulan*

Dari data yang diperoleh dalam penelitian ini dapat ditarik suatu kesimpulan bahwa :

1) Image yang memiliki hanya satu warna, tidak akan bisa digunakan sebagai media steganografi, dikarenakan kapasitas yang dihasilkan kurang, minimal lebih dari 1 warna dalam image agar bisa digunakan sebagai media steganografi.

2) Semakin banyak warna dan semakin besar piksel pada image akan berpengaruh terhadap peningkatan kapasitas secara nonlinier.

3) Menghitung berapa besar perbedaan yang terjadi antara dua image asli dan image steganografi, dilakukan perhitungan MSE dan PSNR. Semakin banyak file yang disisipkan maka perbedaan akan semakin besar dengan ditunjukkan hasil dari perhitungan MSE dan PSNR. Nilai PSNR akan berbanding terbalik dengan kapasitas yang berhasil disisipkan. Nilai minimal PSNR dikatakan bagus adalah 35 Db.

4) Semua jenis data dapat disisipkan kedalam image karena penyisipan menggambil nilai bit yang akan disisipkan kedalam image.

5) Kualitas kompresi JPEG sangat berpengaruh terhadap prosentasi keberhasilan ekstraksi data. semakin sempurna kompresi data atau dengan kualitas 10 maka prosentasi [5].tutorials.setvideo.com/JPEG_Basics.html,

B. *Saran*

Penelitian selanjutnya dapat dilakukan uji coba terhadap keamanan data dengan cara memadukan berbagai teknik proses PRNG (*Pesudo Random Generator*) di dalam proses *permutation straddling*

DAFTAR PUSTAKA

[1] *Neil F. Johnson, "Steganography. Technical Report", November 1995.*

[2] Neil F. Johnson, Zoran Duric, Sushil Jajodia, "Information Hiding: Steganography and Watermarking - Attacks and Countermeasures", Kluwer Academic Press, Norwrll, MA, New York, The Hague, London, 2000.

[3] David Kahn, "*The Codebreakers"*, The Macmillan Company. New York, NY 1967.

[4] John Loomis, "*JPEG Tutorial*", 30 july 1997.

[5] Andrew B. Watson, "Image Compression Using the Discrete Cosine Transform", Mathematica Journal, 4(1), 1994, p. 81-88, NASA Ames Research Center.

[6] Mr.S. V. Viraktamath, Dr. Girish V. Attimarad, "Impact of Quantization Matrix on the Performance of JPEG", International Journal of Future Generation Communication and Networking, Vol. 4, No. 3, September, 2011.

[7] Gregory K. Wallace, "The JPEG Still Picture Compression Standard", Multimedia Engineering, Digital Equipment Corporation, Maynard, Massachusetts, December 1991.

[8] Steven Pigeon, "Huffman Coding, Chapter 1", Universit´e de Montr´eal.

[9] Sariel Har-Peled, "Huffman Coding, chapter 25", December 6, 2007.

[10] Andreas Westfeld, "F5—A Steganographic Algorithm - High Capacity Despite Better Steganalysis"

[11] Jessica Fridrich, Miroslav Goljan, Dorin Hogea, "Steganalysis of JPEG Images: Breaking the F5 Algorithm"