Steganografi Video H263 dengan Metode Discrete Cosine Transform

Herlinawati

Jurusan Teknik Elektro Universitas Lampung, Bandar Lampung
Jl. Prof. Sumantri Brojonegoro No.1 Bandar Lampung 35145
herlinawati.rusydi@yahoo.com

Intisari **— Steganografi merupakan teknik penyembunyian sinyal informasi dengan cara menyisipkan informasi ke dalam media tertentu sehingga orang tidak menyadari keberadaan pesan tersebut. Penelitian yang dilakukan dengan menyisipkan pesan ke dalam video H263 menggunakan metode Discrete Cosine Transform, yaitu melakukan perubahan terhadap nilai koefisien DCT pada video sesuai dengan pesan masukan serta mengimplementasikan teknik steganografi menggunakan perangkat lunak XGP Desktop dan mengukur kualitas video menggunakan nilai PSNR (Peak Signal to Noise Ratio). Hasil akhir yang diperoleh adalah telah berhasil melakukan penyisipan pesan ke dalam video H263 dan dapat mengekstraksi kembali pesan tersebut menggunakan kunci yang sama. Hasil pengukuran kualitas video yang diperoleh menunjukkan bahwa semakin banyak pesan yang disisipkan maka kualitas video akan semakin buruk.**
Kata kunci — **Steganografi, Discrete Cosine Transform, PSNR.**

Abstract — **Steganography is a technique of hiding confidential information signal by inserting message into certain media so that people are not aware of the existence of the messrage. Research conducted by inserting messages into video H263 using Discrete Cosine Transform, which made changes to the value of DCT coefficients in a video in accordance with the input message and implement steganographic techniques using software XGP Desktop and measure video quality using PSNR (Peak Signal to Noise Ratio). The final results obtained is has succeeded in inserting messages into H263 video and can extract the message using the same key. Video quality measurement results obtained show that the more message that are inserted then the video quality will be worse.**
Keywords— **Steganography, Discrete Cosine Transform, PSNR**

I. PENDAHULUAN

Perkembangan komunikasi digital membuat lalu lintas pengiriman data informasi semakin pesat. Hal inilah yang menuntut adanya sistem pengamanan terhadap data informasi agar bisa aman sampai di tujuan.

Teknik Steganografi merupakan ilmu dan seni penyembunyian informasi dengan menyisipkan suatu informasi ke dalam suatu media sehingga orang tidak menyadari keberadaan pesan tersebut [1]. Dengan menggunakan teknik ini, isi informasi yang ingin dikirimkan terlebih dahulu disembunyikan / disisipkan ke dalam suatu bentuk media umum (*cover object*) berupa gambar, teks, audio, video yang dapat kita temui sehari-hari. Hasil dari penyisipan (*stego object*) dapat dikirimkan dengan aman, karena tidak menimbulkan kecurigaan dan dokumen yang disisipkan hanya dapat diambil oleh orang yang memiliki kata kunci untuk mengaksesnya. Pemilihan jenis media disesuaikan dengan kebutuhan. Apabila pesan yang ingin disembunyikan berukuran besar, maka media yang cocok adalah video. Format video mp4 dengan codec H263 mengandung sejumlah frame dimana masing-masing frame dapat disisipi, sehingga kapasitas penyimpanan pada video menjadi besar.

Metode modifikasi *Discrete Cosine Transform* digunakan karena penurunan kualitas pada video yang dihasilkan tidak signifikan, dan pesan didalamnya tidak akan

hilang apabila dilakukan perubahan terhadap video tersebut. Perangkat lunak yang digunakan dikembangkan di lingkungan PC (*Personal Computer*) / *notebook* yang memiliki kapasitas memori yang lebih besar, sehingga dapat menyimpan dan mengolah video dengan kapasitas dan jumlah yang banyak.

II. TINJAUAN PUSTAKA

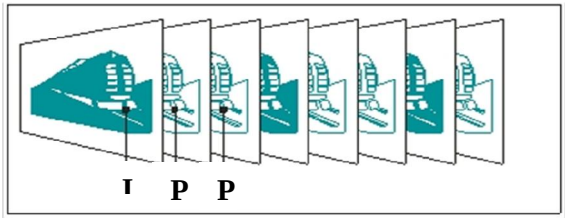
Video digital tersusun atas serangkaian *frame* yang masing-masing *frame* merupakan *image* digital. *Image* digital direpresentasikan dengan sebuah matriks. Bila X adalah matriks dua dimensi, maka X(m,n) adalah nilai intensitas yang sesuai pada baris x dan kolom y pada matriks tersebut. Titik-titik dimana *image* dicuplik (di-*sampling*) disebut *picture element (pixel)* [2]. Karakteristik video digital ditentukan oleh resolusi (dimensi *frame*), kedalaman piksel (*pixel depth*) dan *framerate.*

A. Group of Picture (GoP)

GoP adalah kelompok gambar yang berurutan dalam sebuah *codec video stream*. Suatu GoP berisi tipe gambar sebagai berikut:

1) *I-frame* (*Intra code frame*) , merupakan *frame* yang dikodekan secara penuh dari suatu gambar tanpa menggunakan prediksi dari *frame* lain dan *frame* ini merupakan awal dari GoP. *Frame I* mempunyai laju bit paling tinggi di antara ketiga tipe *frame*.
2) *P-frame* (*Predictive code frame*) , merupakan *frame* yang diprediksi berdasarkan *frame* referensi I atau P sebelumnya.
3) *B-frame* (*Bi-directional code frame*) , merupakan *frame* yang diprediksi berdasarkan *frame* referensi I dan P sebelumnya dan *frame* I atau P berikutnya. *Frame* B mempunyai laju bit paling rendah di antara ketiga tipe *frame*.

Video *codec* H.263 *baseline profile* hanya terdiri dari dua jenis *frame*, yaitu I-*frame* (INTRA) dan P-*frame* (INTER). Keduanya ditandai dengan *video coding type*, dimana "0" untuk I-*frame* dan "1" untuk P-*frame* [3].



Gbr. 1 *Group of Pictures* H.263

B. Kompresi Berkas Video

Ukuran berkas video sangat besar untuk memanipulasi sebuah berkas multimedia, sehingga diperlukan proses *encoding* adalah memadatkan berkas multimedia menjadi data terkompresi dan proses *decoding* adalah proses sebaliknya yang dilakukan oleh setiap pengguna yang ingin menonton video.

Dua jenis kompresi yang dapat dilakukan pada sebuah video, yaitu kompresi *intraframe* dan *interframe.*

1) Kompresi *intraframe*

Proses kompresi terfokus pada satu *frame* saja yaitu I-*frame*. Contohnya yaitu kompresi gambar berformat JPEG yang menggunakan *Discrete Cosine Transform* (DCT) dan *Inverse Discrete Cosine Transform* (IDCT) sebagai proses transformasi.

Fungsi DCT yaitu mentransformasi data dari tempat spasial (*spatial domain*) ke tempat frekuensi (*frequency domain*). DCT digunakan, terutama pada kompresi JPEG, untuk mentransformasikan blok 8x8 *pixels* yang berurutan dari gambar menjadi 64 koefisien DCT. Jika x[m,n] adalah nilai *pixel* pada sebuah blok, maka koefisien DCT dari setiap blok pada gambar dapat dihitung sebagai berikut [4]:

$$X(u,v) = \frac{C[u]C[v]}{4} \sum_{m=0}^{7} \sum_{n=0}^{7} x[m,n] \cos\frac{(2m+1)u\pi}{16} \times \frac{\cos(2n+1)v\pi}{16} \quad (1)$$

dimana : $0 \le u, v \le 7$

= , jika nilai u sama dengan 0,

dan , jika .

IDCT merupakan kebalikan dari DCT yang akan mengembalikan koefisien pada matriks frekuensi menjadi matriks *spatial* dengan persamaan sebagai berikut:

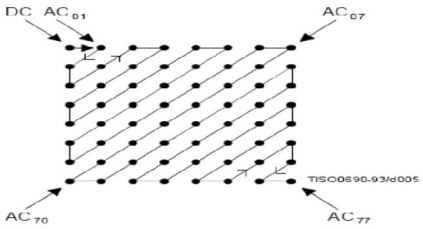
$$x[m,n] = \sum_{u=0}^{7}\sum_{v=0}^{7} \frac{C[u]C[v]}{4} X[u,v] \cos\frac{(2m+1)u\pi}{16} \times \cos\frac{(2n+1)v\pi}{16} \quad (2)$$

dimana: $0 \le m, n \le 7$.

Nilai-nilai *pixel* akan disubstitusikan ke dalam persamaan (1), sehingga diperoleh koefisien DCT. Koefisien pada posisi [0,0] merupakan koefisien DC, 63 koefisien lainnya merupakan koefisien AC. Setelah koefisien DCT diperoleh, dilakukan proses kuantisasi yaitu proses pemotongan nilai koefisien DCT dimana matriks frekuensi dibagi dengan matriks kuantisasi

$$qX = \left[\frac{X[m,n]}{q[m,n]}\right] \quad (3)$$

Matriks frekuensi setelah kuantisasi biasanya banyak memiliki nilai 0 dibagian kanan bawah. Proses kompresi dilanjutkan dengan melakukan *entropy coding* untuk menyimpan matriks dengan urutan *zig-zag*. Urutan *zig-zag* digunakan untuk mengelompokkan koefisien frekuensi rendah pada bagian atas vektor, sehingga meningkatkan efisiensi pengelompokkan komponen tidak nol.



Gbr. 2 Urutan *zig-zag* pada *entropy coding*

2) Kompresi *interframe*

Pada kompresi ini dilakukan pencarian perubahan antar blok pada *frame* yang berbeda yang disebut dengan *motion estimation*. Nilai pergeseran lokasi antar kedua blok dinamakan dengan *motion vector*, dan nilai inilah yang disimpan. Proses mengaplikasikan *motion vector* pada *frame* untuk memperoleh *frame* berikutnya disebut dengan *motion compensation* [5][6].

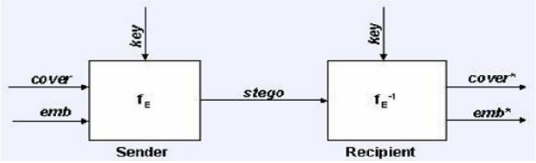
C. H. 263

Video *codec* H.263 merupakan format video yang dibuat oleh ITU-T, yaitu video terkompresi yang memiliki *bit rate* rendah sehingga cocok dipakai pada keperluan transmisi network, seperti *video conference*.

Karakteristik dari video *codec* H.263 ditentukan sesuai dengan profil yang akan menentukan fitur-fitur apa saja yang didukung video tersebut, dan *level* yang mengatur kapabilitas performansi video seperti ukuran resolusi dan *bit rate* yang digunakan.

D. Steganografi

Steganografi merupakan salah satu cara untuk menyembunyikan suatu pesan atau data rahasia di dalam data pesan lain yang tampak tidak mengandung apa-apa, kecuali bagi orang yang mengerti kuncinya [7].



Gbr. 3 Sistem Steganografi

f_E : Fungsi steganografi berupa *embedding* (memasukkan data)
f_E^{-1} : Fungsi steganografi berupa *extracting* (mengekstrak data)
cover: Tempat untuk pesan yang akan disembunyikan
emb : pesan yang akan disembunyikan
key : kunci berupa password
stego : pesan yang sudah dimasukkan ke dalam *cover message*

Gambar 3 menunjukkan sistem steganografi umum dimana di bagian pengirim pesan (*sender*), dilakukan proses *embedding* (f_E) pesan yang hendak dikirim secara rahasia (*emb*) ke dalam data *cover* sebagai tempat menyimpannya (*cover*), dengan menggunakan kunci tertentu (*key*),

sehingga dihasilkan data dengan pesan tersembunyi di dalamnya (*stego*). Di bagian penerima pesan (*recipient*), dilakukan proses *extracting* (f_E^{-1}) pada stego untuk memisahkan pesan rahasia (*emb*) dan data penyimpan (*cover*) tadi dengan menggunakan kunci yang sama seperti pada proses *embedding*. Jadi hanya orang yang mengetahui kunci yang dapat mengekstrak pesan rahasia tersebut [8].

1) Kriteria Steganografi

Kriteria utama yang digunakan dalam menilai kualitas steganografi, adalah[9]:

a. *Imperceptible*, semakin pesan rahasia tidak dapat dipersepsi semakin bagus *stego-object*
b. *Fidelity*, mutu *cover-object* tidak jauh berubah di banding sebelum dimasukan *embedded message.*
c. *Recovery*, Data yang disembunyikan harus dapat diungkap kembali.
d. *Capacity*, merupakan besar data yang bisa disisipkan dalam sebuah *cover*, dibandingkan dengan besarnya *cover* itu sendiri. Kapasitas penyimpanan ini juga sering disebut *payload*.

2) Manfaat Steganografi

Steganografi dapat digunakan untuk berbagai keperluan, diantaranya yaitu sebagai perlindungan hak cipta [7], sebagai *tag-notes* untuk citra *online*. Steganografi juga dapat digunakan untuk melakukan perawatan atas kerahasiaan informasi yang berharga, untuk menjaga data tersebut dari kemungkinan sabotase, pencuri, atau pihak yang tidak berwenang [8].

3) Steganografi Pada Video dengan Metode modifikasi DCT

Penyisipan pesan ke dalam video dengan metode modifikasi DCT dilakukan dengan menyisipkan pesan ke dalam koefisien DCT yang terdapat pada I-*frame*. Pada I-*frame* terdapat dua jenis angka yang dapat menjadi tempat penyisipan, yaitu INTRA-DC dan TCOEF. Nilai INTRA-DC pasti dimiliki oleh setiap blok pada I-*frame*, namun nilai TCOEF tidak selalu ada, yaitu apabila matriks frekuensi tidak memiliki koefisien AC, atau semua koefisiennya bernilai 0.

Penyisipan pada TCOEF

Nilai TCOEF mengandung komponen LAST, RUN, dan LEVEL, yang dikodekan menggunakan VLC (*Variable Length Code*) sehingga panjangnya tidak tetap. LAST merupakan penanda apakah koefisien ini merupakan koefisien bukan 0 yang terakhir atau tidak, RUN menunjukkan berapa angka 0 yang mendahului koefisien ini, sedangkan LEVEL adalah nilai absolut dari koefisien AC. Bit terakhir dari kode VLC menunjukkan tanda positif/negatif nilai LEVEL, 0 untuk positif dan 1 untuk negatif [3].

Apabila pesan disisipkan langsung pada kode TCOEF, akan terdapat satu atau lebih komponen yang ikut berubah. Jika komponen tersebut adalah LAST atau RUN maka susunan matriks frekuensi dapat terganggu.

Perubahan nomor 1 hanya mengubah nilai LEVEL saja, sehingga tidak mengubah susunan matriks frekuensi. Perubahan pada nomor 2 mengubah nilai RUN dan LEVEL, yang menyebabkan konfigurasi nilai koefisien AC yang seharusnya bernilai 3 dan tidak ada nilai 0 di depannya, sehingga pembacaan susunan matriks frekuensi menjadi terganggu. Sedangkan pada nomor 3, nilai koefisien tidak diketahui karena tidak ada kode TCOEF 11.

Penyisipan pesan ke dalam TCOEF sebenarnya memiliki efek perubahan pada gambar yang lebih sedikit daripada INTRA-DC. Oleh karena itu, penyisipan pesan akan dilakukan pada nilai INTRA-DC pada blok I-*frame* [5].

4) Ukuran Maksimum Pesan

Media yang digunakan adalah video digital dengan format MP4 dan video *codec* yang didukung adalah H.263 *baseline profile* dan *level* 10 dengan resolusi yang didukung 128×96 *pixels* (*Sub*-QCIF) dan 174×144 *pixels* (QCIF) dan *bit rate* 64 kbps dan *frame rate* 14,98 fps. Apabila pesan akan disisipkan di dalam semua nilai LSB INTRA-DC yang terdapat pada blok, maka kapasitas maksimum pesan yang dapat disisipkan pada satu I-*frame* dapat dilihat pada tabel berikut [5][6].

Tabel 1. Kapasitas maksimum pesan pada satu I-*frames*

Resolusi	Macrob lock	Blok	Pesan Maksimum
Sub-QCIF (128×96)	48	288	36 *bytes*
QCIF (174×144)	99	594	74,25 *bytes*

5) Penilaian kualitas video steganografi

Video yang sudah disisipi dengan data rahasia tidak sama dengan *video* aslinya dan mengalami penurunan kualitas. Di sini akan diperhitungkan tingkat perubahan yang terjadi dan seberapa besar kerusakan (*error*) yang ditimbulkan. Untuk menentukan dan menilai kualitas video-*stego* dilakukan pengukuran nilai MSE dan PSNR.

a. *Mean Square Error* (MSE)

Mean square error adalah parameter yang digunakan untuk menentukan tingkat kesalahan pada citra *stego*. [9]

$$MSE = \frac{1}{M \times N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (x_0[m,n] - x_r[m,n])^2 \qquad (4)$$

dimana :

M = Panjang citra (dalam *pixel*)
N = Lebar citra (dalam *pixel*)
x_0 = Citra asli
x_r = Citra setelah rekonstruksi
$[m,n]$ = Koordinat masing-masing *pixel*

b. *Peak Signal to Noise Ratio* (PSNR)

Peak Signal to Noise Ratio adalah perbandingan antara nilai maksimum dari sinyal yang diukur dengan besarnya derau yang berpengaruh pada sinyal tersebut. PSNR menyatakan tingkat noise atas citra yang telah disisipi , dinyatakan dalam desibel (dB) [9].

$$PSNR = 10 log_{10} \frac{255^2}{MSE}$$

III. METODE PENELITIAN

A. Prosedur Penyisipan Pesan

Penyisipan pesan dilakukan menggunakan perangkat lunak XGP *Dekstop*. Diperlukan tiga buah masukan, yaitu video asli, pesan yang akan disisipkan, dan kunci dengan langkah-langkah sebagai berikut:

1) Memasukkan video asli yang akan digunakan sebagai media penyisipan pesan. Sistem akan menuliskan nama untuk video keluaran agar tidak terjadi *overwrite* terhadap video asli.
2) Memasukkan *file* pesan yang akan disisipkan. Pesan berupa *file* teks dengan format txt.
3) Mengetikkan kunci.
4) Meng-klik tombol ‘proceed’, selanjutnya sistem akan melakukan proses penyisipan pesan.

Penyisipan pesan berhasil jika XGP *Dekstop* dapat menyisipkan pesan ke dalam video dan mampu menangani penyisipan pesan hingga 3 LSB INTRA-DC. Jika ukuran pesan melebihi 3 LSB maka pesan yang akan disisipkan hanya sebagian yaitu tidak lebih dari penggunaan 3 LSB.

B. Prosedur Ekstraksi Pesan

Ekstraksi pesan dilakukan menggunakan perangkat lunak XGP *Dekstop*. Diperlukan dua buah masukan, yaitu video-*stego*, dan kunci. Langkah-langkah yang dilakukan sebagai berikut:

1) Memasukkan video-*stego* (video yang telah disisipi pesan).

2) Mengetikkan kunci yang sama dengan kunci yang digunakan saat penyisipan pesan.
3) Mengklik tombol 'proceed', selanjutnya sistem akan melakukan proses ekstraksi pesan. Pesan hasil ekstraksi akan disimpan secara otomatis oleh sistem dengan nama *file* yang baru sehingga tidak akan terjadi *overwrite* terhadap *file* pesan asli.
4) Melakukan langkah a sampai c, dengan mengetikkan kunci yang berbeda dengan kunci yang digunakan saat penyisipan pesan.

Proses berhasil dan diterima jika XGP *Dekstop* dapat mengekstraksi pesan yang telah disisipkan. Jika proses ekstraksi menggunakan kunci yang sama dengan saat penyisipan, maka isi *file* pesan hasil ekstraksi harus sama dengan *file* asli. Proses ekstraksi dengan menggunakan kunci yang berbeda harus menghasilkan *file* dengan isi yang berbeda dengan pesan yang asli untuk *file* pesan dengan format .txt.

C. *Prosedur Pengukuran Kualitas Video Secara Obyektif*

Dibutuhkan dua buah masukan yaitu *frame* video asli dan *frame* video-*stego* yang bersesuaian. Kedua video tersebut diekstrak terlebih dahulu, menjadi *frame* (gambar) berformat jpeg menggunakan perangkat lunak *Total Video Converter*. Kemudian menghitung nilai MSE dari masing-masing *frame* pada video asli dengan *frame* pada video-*stego* yang bersesuaian. Langkah-langkah yang dilakukan sebagai berikut:

1) Memasukkan *frame* video asli.
2) Memasukkan *frame* video-*stego*.
3) Memilih kategori resolusi *frame* (QCIF atau Sub-QCIF) sesuai dengan ukuran resolusi *frame* tersebut.
4) Mengklik tombol 'MSE' sehingga akan ditampilkan nilai MSE dari kedua *frame* tersebut.
5) Menghitung nilai rata-rata MSE yang diperoleh.
6) Menghitung nilai PSNR dengan menggunakan nilai rata-rata MSE yang diperoleh.

IV. PEMBAHASAN

A. *Hasil Implementasi Steganografi*

1) Hasil Penyisipan Pesan ke dalam Video

Terdapat dua video yang akan disisipkan pesan yaitu video Satu.mp4 (sub-QCIF) dan video Dua.mp4 (QCIF). Pesan masukannya adalah file(1).txt dan file(2).txt yang akan disisipkan secara bergantian ke dalam video asli. Kunci yang digunakan adalah *string* aimh dan video keluaran diberi nama baru sesuai dengan *file* pesan. Hasil penyisipan pesan ditunjukkan pada Tabel 2.

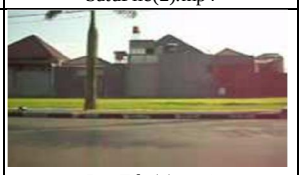
Tabel 2. Hasil penyisipan pesan ke dalam video

Masukan Video Asli	Masukan Pesan	Masukan Kunci	Keluaran Video Stego
Satu.mp4	file(1).txt	aimh	SatuFile(1).mp4
	file(2).txt	aimh	SatuFile(2).mp4
Dua.mp4	file(1).txt	aimh	DuaFile(1).mp4
	file(2).txt	aimh	DuaFile(2).mp4

Beberapa perbandingan *frame* dari video asli dan video hasil penyisipan (video *stego*) dapat dilihat pada Tabel 3.

Frame video yang telah disisipkan berupa pesan file(1).txt dan file(2).txt masih terlihat mirip dengan *frame* video asli, baik video Satu.mp4 maupun video Dua.mp4, karena ukuran pesan yang disisipkan masih dapat ditampung pada 1 LSB INTRA-DC.

Tabel 3. Perbandingan *frame* video asli dengan *frame* video *stego*

Frame Video Asli	Frame Video Stego
file pesan: file(1).txt (42 *bytes*)	file pesan: file(1).txt (42 *bytes*)
	
Satu.mp4	SatuFile(1).mp4
file pesan: file(1).txt (77 *bytes*)	file pesan: file(1).txt (77 *bytes*)
	
Dua.mp4	DuaFile(1).mp4
file pesan: file(2).txt (77 *bytes*)	file pesan: file(2).txt (77 *bytes*)
	
Satu.mp4	SatuFile(2).mp4
	
Dua.mp4	DuaFile(2).mp4

2) Hasil ekstraksi pesan

Video *stego* diekstraksi untuk dapat menghasilkan pesan yang telah disisipkan ke dalam video tersebut. Nama *file* pesan hasil ekstraksi akan ditambahkan dengan penomoran secara otomatis sesuai dengan urutan proses ekstraksi agar *file* hasil ekstraksi tidak *overwrite* terhadap *file* asli, namun hal ini tidak akan mempengaruhi isi pesan. Perbandingan pesan asli dan pesan hasil ekstraksi dengan menggunakan kunci yang sama seperti pada saat penyisipan yaitu *string* tien dapat dilihat pada Tabel 4.

Pada tabel 4 dapat dilihat bahwa telah berhasil dilakukan proses ekstraksi pesan yang telah disisipkan kedalam sebuah video. Hasil ekstraksi sama dengan pesan asli yang berhasil disisipkan.

Tabel 4. Perbandingan pesan asli dan pesan hasil ekstraksi dengan menggunakan kunci yang benar

Nama video stego	Pesan asli	Pesan hasil ekstraksi
SatuFile(1).mp4	file(1).txt dengan isi: jurusan teknik elektro universitas lampung	file(1)_1.txt dengan isi: jurusan teknik elektro universitas lampung
SatuFile(2).mp4	file(2).txt dengan isi: jangan pernah menunda pekerjaan	file(2)_1.txt dengan isi: jangan pernah menunda pekerjaan
DuaFile(1).mp4	file(1).txt: jurusan teknik elektro universitas lampung	file(1)_2.txt: jurusan teknik elektro universitas lampung
DuaFile(2).mp4	file(2).txt dengan isi: jangan pernah menunda pekerjaan	file(2)_2.txt dengan isi: jangan pernah menunda pekerjaan

Pengujian selanjutnya yaitu melakukan ekstraksi pesan dengan menggunakan kunci yang berbeda dari kunci yang digunakan pada saat proses penyisipan. Dalam hal ini kunci yang digunakan yaitu *string* miah. Data pesan hasil ekstraksi ditunjukkan pada tabel 5.

Dari hasil pengujian terbukti bahwa proses ekstraksi dengan menggunakan kunci yang salah atau berbeda dari kunci saat proses penyisipan akan menghasilkan pesan dengan isi yang berbeda dari pesan asli, karena kunci yang digunakan berfungsi sebagai *seed* yaitu suatu angka yang dibutuhkan untuk algoritma pembangkitan bilangan acak yang akan mengatur letak dan urutan pesan. Kunci yang berbeda akan menghasilkan letak dan urutan pesan yang berbeda. Dengan demikian hanya yang mengetahui kuncinya saja yang dapat memperoleh pesan sesuai dengan aslinya, hal ini sesuai dengan pengertian dan tujuan steganografi yaitu merupakan salah satu cara untuk menyembunyikan suatu pesan atau data rahasia di dalam data pesan lain yang tampak tidak mengandung apa-apa, kecuali bagi orang yang mengerti kuncinya.

Tabel 5. Perbandingan pesan asli dengan pesan hasil ekstraksi menggunakan kunci yang salah

Nama video stego	Pesan asli	Pesan hasil ekstraksi
SatuFile(1).mp4	file(1).txt dengan isi:jurusan teknik elektro universitas lampung	file(1)_3.txt dengan isi: %ıh>ž¯ãe€ıEï?·n ŽèëÍ@NUtU_.€Í²ı Cpoe:IÂÅ€
SatuFile(2).mp4	file(2).txt dengan isi: jangan pernah menunda pekerjaan	file(2)_3.txt dengan isi: ‹z”Ã-G)Ë"Z´Ãè+¿o B„ãgCX%„<«òO` 4äDÍHÖ…oı\˜”t‘ A2ÂóGŠÞ´ÏÖH|º — Aìò˜í'!ø«
DuaFile(1).mp4	file(1).txt: jurusan teknik elektro universitas lampung	file(1)_4.txt: %ıh>ž¯ãe€ıEï?·n ŽèëÍ@NUtU_.€Í²ı Cpoe:IÂÅ€
DuaFile(2).mp4	file(2).txt dengan isi: jangan pernah menunda pekerjaan	file(2)_4.txt dengan isi:‹z”Ã-¶G)Ë"Z´Ãè+¿o B„ãgCX%„<«òO` 4äDÍHÖ…oı\˜”t‘ A2ÂóGŠÞ´ÏÖH|º — Aìò˜í'!ø«

Dari hasil ekstraksi dapat disimpulkan bahwa XGP *Dekstop* berhasil melakukan proses ekstraksi dengan baik sesuai dengan yang diharapkan dan dapat diterima.

Tabel 6. Hasil ekstraksi pesan dari video *stego* dengan mengunakan kunci yang benar

Masukan video stego (.mp4)	Kunci	Keluaran	Kesimpulan
SatuFile(1)	aimh	file(1)_1.txt,dengan isi sama dengan file(1).txt	Diterima
SatuFile(2)	aimh	file(2)_1.txt,dengan isi sama dengan file(2).txt	Diterima
DuaFile(1)	aimh	file(1)_2.txt,dengan isi sama dengan file(1).txt	Diterima
DuaFile(2)	aimh	file(2)_2.txt,dengan isi sama dengan file(2).txt	Diterima

Tabel 7. Hasil ekstraksi pesan dari video *stego* dengan menggunakan kunci yang salah

Masukan video stego (.mp4)	Kunci	Keluaran	Kesimpulan
SatuFile(1)	miah	file(1)_1.txt,dengan isi berbeda dari file(1).txt	Diterima
SatuFile(2)	miah	file(2)_1.txt,dengan isi berbeda dari file(2).txt	Diterima
DuaFile(1)	miah	file(1)_2.txt,dengan isi berbeda dari file(1).txt	Diterima
DuaFile(2)	miah	file(2)_2.txt,dengan isi berbeda dari file(2).txt	Diterima

B. Hasil Pengukuran Kualitas Video Secara Obyektif

Pengukuran dilakukan dengan menghitung nilai MSE setiap *frame* yang bersesuaian dan mencari nilai rata-rata MSE pada satu video. Kemudian menghitung nilai PSNR dengan menggunakan nilai MSE rata-rata. Berikut ini adalah data hasil pengukuran kualitas video secara obyektif.

Tabel 8. Hasil pengukuran kualitas video secara obyektif

Masukan video asli	Masukan video stego	Nilai Ratarata MSE	PSNR
Satu.mp4	SatuFile(1).mp4	0.000416	82.25369 dB
Satu.mp4	SatuFile(2).mp4	0.001545	76.24152 dB
Dua.mp4	DuaFile(1).mp4	0.000130	86,99137 dB
Dua.mp4	DuaFile(2).mp4	0.000170	85,82631 dB

Berdasarkan tabel 8 dapat dilihat bahwa semakin besar pesan yang disisipkan maka nilai PSNR semakin kecil yang berarti bahwa kualitas video akan semakin buruk. Secara umum nilai PSNR video Dua.mp4 (resolusi 176x144 *pixels*) lebih tinggi dibandingkan dengan video Satu.mp4 (resolusi 128x96 *pixels*). Video dengan resolusi yang lebih besar memiliki *block* yang lebih banyak, sehingga terdapat lebih banyak INTRA-DC sebagai tempat penyisipan pesan. Jika video

dengan resolusi lebih besar dapat menampung seluruh pesan pada 1 LSB INTRA-DC di semua I-*frame*, maka kemungkinan video dengan resolusi kecil akan menampung penyisipan pesan hingga 2 LSB INTRA-DC. Hal ini mengakibatkan banyaknya perubahan pada *frame* yang bersesuaian antara video asli dengan video-*stego*, sehingga semakin berkurangnya kualitas video tersebut. Nilai PSNR yang dihasilkan dipengaruhi oleh banyaknya *frame* karena nilai PSNR diperoleh dari nilai MSE yang menghitung perubahan yang terjadi pada *frame*.

Sebagai contoh, Video Dua.mp4 terdiri dari 150 *frame* sedangkan video Satu.mp4 terdiri dari 91 *frame*. Misalkan sebuah pesan yang disisipkan akan menyebabkan perubahan hingga 91 *frame*. Jika pesan tersebut disisipkan pada video Satu.mp4 maka seluruh *frame*-nya akan mengalami perubahan, namun jika pesan tersebut disisipkan pada video Dua.mp4 maka masih tersisa 59 *frame* yang tidak mengalami perubahan. Dengan demikian video Dua.mp4 memiliki nilai MSE yang lebih kecil dibandingkan video Satu.mp4. Semakin kecil nilai MSE maka nilai PSNR semakin besar.

V. SIMPULAN

1) Implementasi steganografi dengan metode DCT *Modification* telah berhasil dilakukan dengan menggunakan perangkat lunak XGP *Dekstop*.
2) Video Satu.mp4 dengan ukuran 39.819 *bytes* dapat menyisipkan pesan maksimal sebesar 640 *bytes*, sedangkan video Dua.mp4 dengan ukuran 192.579 *bytes* dapat menyisipkan pesan maksimal sebesar 2.304 *bytes*.
3) Proses ekstraksi pesan dengan menggunakan kunci yang sama dengan saat penyisipan akan menghasilkan pesan yang sama dengan pesan asli yang disisipkan, sedangkan ekstraksi dengan menggunakan kunci yang berbeda dari kunci yang digunakan saat proses penyisipan akan menghasilkan pesan yang salah..
4) Kualitas video yang dihasilkan bergantung dari besarnya ukuran pesan, semakin besar pesan yang disisipkan maka kualitas video akan semakin buruk.
5) Penyisipan pesan pada 3 LSB INTRA-DC sudah menunjukkan penurunan kualitas yang cukup signifikan.

REFERENSI

[1] Cachin, Christian. 2005. *Digital Steganography*. 21 Oktober 2009. http://www.zurich.ibm.com/~cca/papers/encyc.pdf.

[2] Masaleno, Andino. 2006. Pengantar Steganografi. 1 Agustus 2009. http://ismailzone.com/download/cryptography/andino-steganografi.pdf

[3] Hariyanto, Paul Gunawan. 2008. Studi dan Implementasi Steganografi pada Video Digital di *Mobile Phone* dengan Metode DCT *Modification*.Teknik Informatika ITB. Bandung. 58 hlm.

[4] Bovik, Al (editor). *Handbook of Image and Video Processing*. Academic Press. Canada. 891 hlm.

[5] Ghanbari, Mohammed. 2003. *Standard Codecs: Image Compression to Advanced Video Coding.* Institution of Electrical Engineers. 407 hlm.

[6] Tarigan, Jhon Kalvin. 2008. *Implementasi Steganografi pada Video AVI yang tidak Terkompresi (Full Frames) Menggunakan Metode SSB-4*. IT Telkom. Bandung. 4 Oktober 2009.

[7] International Telecommunication Union. 2005. *ITU-T Recommendation H.263: "Video coding for low bit rate communication"*.

[8] Anonim. 2009. Keamanan Multimedia. 4 Oktober 2009. http://elista.akprind.ac.id/staff/catur/Sistem%20Multimedia/12-Keamanan%20Multimedia.pdf

[9] Rizaldy, Muhammad Ray. *Teknik Penyembunyian Pesan pada Berkas Video*.7 Oktober 2009.

http://www.informatika.org/~rinaldi/Kriptografi/2008-2009/Makalah1/MakalahIF30581-2009-a037.pdf

[10] Nazaret, Ralph. Video Steganography. 2 November 2009. http://www.keepandshare.com/doc/view.php?id=595718&dn=y

[11] Pratama, Ekky. 2008. *Pengkodean Video H.264/AVC*. Teknik Informatika ITB. Bandung.

[12] Texas Instruments. 2001. *H.263 Standard – Overview and TMS320C6000DSP Information.* White Paper. 15 Februari 2010. http://dspvillage.ti.com/pdfs/spra018.pdf.

[13] Wu, H.R. dan K.R. Rao. 2006. *Digital Video Image Quality and Perceptual Coding.* Crc Press & tayylor Francis Group.

[14] Xu, Changyong dan Ping, Xijian dan Zhang, Tao. 2006. *Steganography in Compressed Video Stream. IEEE International Conference on Innovative Computing, Information and Control (ICICIC'06) Journal*. 4 hlm