PERANCANGAN DAN ANALISIS STEGANOGRAFI VIDEO DENGAN MENYISIPKAN TEKS MENGGUNAKAN METODE DCT

PLANNING AND ANALYSIS VIDEO STEGANOGRAPHY BY EMBEDDING TEXT WITH DISCRETE COSINE TRANSFORM METHOD

[1]Ryan Anggara, [2]Gelar Budiman, S.T.,M.T., [3]Ledya Novamizanti S.Si.,M.T.

Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom
[1]rayenangga@gmail.com [2]gelarbudiman@telkomuniversity.ac.id [3]ledyaldn@telkomuniversity.ac.id

Abstrak

Perkembangan pada teknologi internet yang pesat, membuat tingkat privasi dalam pertukaran dan pengiriman informasi penting menjadi menurun. Sehingga tidak sedikit informasi penting tersebut bisa terbongkar. Salah satu metode untuk menyembunyikan pesan ke suatu media adalah dengan Steganografi. Dengan steganografi, pesan rahasia akan bisa disisipkan dalam suatu media dan dapat diakses atau di ekstraksi oleh pihak tertentu. Pada tugas akhir ini dirancang sebuah perangkat lunak untuk dapat menyisipkan teks ke dalam suatu video, kemudian ekstraksi ciri pada video tersebut untuk mendapatkan pesan yang sudah disisipkan, dan akan ada skenario pengujian dengan menggunakan beberapa parameter dan diberikan serangan pada video tersebut. Pada dasarnya, steganografi video mirip dengan steganografi gambar. Karena peracangan tugas akhir ini dimulai dari steganografi gambar. Sehingga pada tugas akhir ini akan dirancang juga ekstraksi video referensi berformat avi ke frame-frame dan kemudian penyisipan akan dilakukan pada frame-frame tersebut. Perangkat lunak akan dirancang dengan menggunakan Matlab. Metode yang digunakan untuk steganografi video sudah banyak. Tetapi pada tugas akhir ini, penulis akan menggunakan metode Discrete Cosine Transform (DCT). Karena DCT salah satu metode yang terbaik untuk melakukan penyisipan teks dan ekstraksi yang baik. Sehingga DCT memiliki akurasi yang baik untuk membaca teks. Dengan tugas akhir ini, penulis memiliki pendapat bahwa steganografi dengan menggunakan metode DCT bisa menyisipkan pesan dan ekstraksi pesan dengan baik ke semua frame video.

Kata kunci : Steganografi, DCT, matlab, Watermarking

1. Pendahuluan

Dewasa ini, pertumbuhan internet yang begitu cepat membuat kebutuhan akan informasi data semakin meningkat. Berlaku juga untuk informasi data yang bersifat rahasia. Untuk menyembunyikan pesan rahasia ke dalam suatu data, steganografi adalah salah satu metode untuk menyembunyikan suatu pesan ke suatu media yang sudah ada dari jaman dahulu. Steganografi terdiri dari 2 kata bahasa yunani, steganos yang berarti melindungi dan graphein yang berarti tulisan. Kata steganografi untuk beberapa kalangan akan membandingkannya dengan kriptografi. Keduanya memiliki tujuan yang hampir sama, yaitu melindungi informasi rahasia. Steganografi menyembunyikan keberadaan pesan, sedangkan Kriptografi menyembunyikan isi pesan dengan mengkonstruksi secara acak media yang diisi pesan tersebut [1].

Dengan menggunakan steganografi, metode untuk menyembunyikan suatu pesan atau teks ke dalam suatu media, misal video, sangat mungkin untuk dilakukan. Steganografi video dapat digunakan dengan metode steganografi gambar, karena video terdiri dari beberapa frame gambar . Banyak metode steganografi gambar yang bisa digunakan untuk diimplementasikan dalam video, tetapi metode yang digunakan dalam tugas akhir ini adalah teknik Discrete Cosine Transform (DCT). Video yang disisipkan dengan teknik DCT akan terkurang sedikit kualitas videonya tetapi memiliki robustness (ketahanan) yang baik dan penyembunyian pesan yang baik [2].

2. Landasan Teori

2.1 Steganografi

Steganografi secara harfiah berarti "pesan tertutup" dan melibatkan transmisi pesan rahasia melalui tampaknya file berbahaya. Tujuannya agar tidak hanya melakukan pesan tetap tersembunyi, tetapi juga bahwa pesan tersembunyi bahkan dikirim tidak terdeteksi. Steganografi mendukung menyembunyikan pesan di antara volume besar lalu lintas internet, dalam file media mana Selain dari pesan tersembunyi sulit untuk mendeteksi dengan mata manusia bahkan jika file tersebut dilihat.

Pada steganografi, ada dua proses umum, yaitu proses penyisipan (embedding) pesan rahasia dan proses ektraksi ciri (extracting) video untuk mendapatkan pesan rahasia tersebut.

2.2 Discrete Cosinus Transform

Discrete Cosine Transform (DCT) adalah salah satu teknik steganografi pada video. DCT juga adalah salah satu transformasi populer pada multimedia coding. Saat ini, tidak sedikit video yang menjadi media penyisipan suatu pesan atau watermarking pada domain DCT. Karena DCT dapat mengurangi kompeksitas pada pengkodean, meningkatkan robustness pada pesan yang tersisipkan, dan mempercepat proses penyisipan.

DCT digunakan untuk metransformasikan blok 8x8 piksel yang berurutan dari image menjadi 64 koefisien DCT yang terdiri dari satu koefisien DC dan 63 koefisien AC. DCT merupakan fungsi yang linear dan dapat dibalikkan kembali (invertible). Hal ini tentunya menjadi penting terutama dalam proses mengembalikan pesan setelah di-embed ke dalam cover. Proses DCT ini juga dilakukan pada kompresi JPEG.

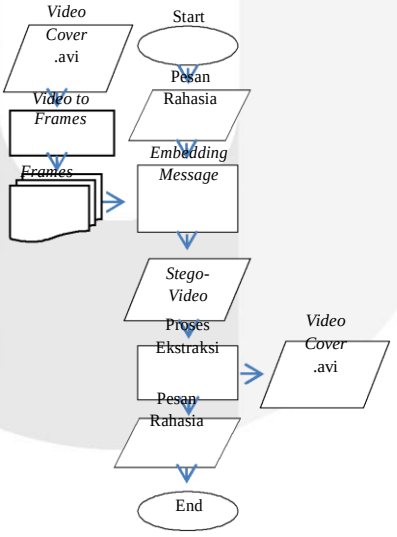
DCT sering digunakan untuk pengolahan image dan sinyal, terutama untuk kompresi dengan format lossy compression. Modifikasi terhadap DCT akan mempengaruhi semua pixel pada image.

Salah satu proses DCT bisa digabungkan dengan metode lain, contoh Least Significant Bit (LSB). Keuntungan yang dapat diambil adalah LSB hasil kuantisasi koefisien DCT dapat digunakan untuk menyembunyikan informasi seperti yang terjadi pada algoritma di atas. Pada dasarnya image yang dikompresi dengan lossy compression akan menimbulkan kecurigaan karena perubahan LSB akan terlihat jelas. Pada metode ini hal tersebut tidak akan terjadi karena metode ini terjadi di domain frekuensi di dalam image, bukan pada domain spasial, sehingga tidak akan ada perubahan yang terlihat pada cover image.

3. Perancangan

3.1 Desain Model Sistem

Model sistem pada tugas akhir ini akan dibagi menjadi dua bagian, sistem penyisipan dan sistem ekstraksi ciri. Dan berikut secara umum desain model sistem yang akan digunakan dalam tugas akhir ini.

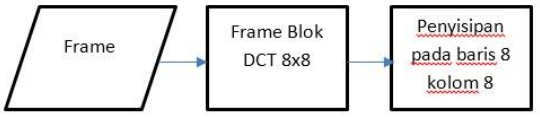


Pertama dimulai dari menyiapkan video berformat avi untuk media penyisipan dan pesan rahasia yang akan disisipkan. Kemudian video tersebut akan di ekstraksikan menjadi frame-frame. Pesan akan disisipkan ke semua frame. Pesan yang akan disisipkan ke frame tersebut akan disisipkan dengan menggunakan metode DCT. Setelah proses penyisipan, frame-frame tersebut akan di bentuk kembali menjadi *stego-video*. *Stego-video* adalah video yang sudah disisipkan oleh suatu pesan.

Stego-video tersebut akan diekstraksikan kembali menjadi frame-frame dan dilakukan proses ekstraksi ciri. Pada proses ekstraksi ciri, pesan yang sudah ada di frame-frame tersebut akan di ekstraksi. Sehingga di program Matlab, akan memunculkan pesan yang telah disisipkan.

Kemudian akan diuji sesuai skenario dengan parameter yang sudah ditentukan.

3.2 Proses Penyisipan



Frame yang digunakan pada pengujian ini memiliki size 512x512. Frame tersebut tiap 8x8 akan di blok dan dijadikan domain dct. Jumlah pesan yang bisa disisipkan adalah 4096 tempat. Hasil tersebut didapat dari pembagian 512x512 (video) dengan 8x8 (blok dct). Jika 512x512 didapat 262144 dan 8x8 didapat 64, maka tempat pesan yang bisa disisipkan pada 1 frame tersebut adalah 4096 tempat atau pada setiap 4096 bit. Pada DCT, penyisipan selalu pada baris 8 dan kolom 8. Maka setelah frame tersebut di blok dct 8x8, maka pesan yang digunakan bernilai 32x32 akan disisipkan di baris 8 dan kolom 8.

3.Pembahasan

3.1. Pengujian Penyisipan

Penyisipan dilakukan setelah mendapatkan video *cover* dan pesan yang ingin disisipkan. Video tersebut di read dengan menggunakan perintah 'VideoReader', karena 'VideoReader' adalah salah satu toolbox terbaru untuk membaca video, bisa membaca semua video yang mendukung software Matlab, dan jika dibandingkan dengan 'aviread', 'VideoReader' tidak perlu menambahkan frame dengan perintah 'addframe', dan bisa melakukan kompresi pada video jika dibutuhkan. Pada tugas akhir ini, saya menamakan program penyisipan ini dengan nama 'VidEmbed.m'.

Proses penyisipan untuk test.avi memerlukan waktu dengan perkiraan 5 menit, dan untuk test2.avi 10 menit, karena frame yang berbeda. Semakin banyak frame, maka waktu untuk penyisipan semakin lama.

Tabel 4.1 Size Video *Cover* dan Video *Embedded*

Video Cover	***Size Awal***	***Video Embedded***	***Size Video Embedded***
test.avi	660 KB	result.avi	188 MB
test2.avi	956 KB	result2.avi	282 MB

Berdasarkan tabel 4.1, size video yang sudah disisipkan oleh pesan sangatlah besar. Karena saat pembuatan video kembali dari frame, tidak menggunakan kompresi apapun. Video ini tidak menggunakan kompresi agar pesan yang telah disisipkan tidak rusak.

Tabel 4.2 Size Video *Embedded* yang telah di kompresi

Video Cover	***Size Awal***	***Video Embedded Terkompresi***	***Size Video Embedded***
test.avi	660 KB	result3.avi	3.35 MB
test2.avi	956 KB	result4.avi	7.77 MB

Dan berdasarkan tabel 4.2, saya juga mencoba video yang telah disisipkan di kompresi dengan format avi dan menggunakan profile 'Motion JPEG AVI'. Size video masih jauh diatas size awal, tetapi kualitas video yang dihasilkan hampir sama.

3.2 Pengujian Ekstraksi Pesan

Pada pengujian ekstraksi pesan, dilakukan setelah video sudah disisipkan pesan. Kedua video tersebut akan diekstrak isi pesannya dan akan dilihat apakah pesan tersebut masih utuh atau tidak.

Tabel 4.3 Keterangan Ekstraksi Pesan Rahasia Pada Kedua Video

Pesan Rahasia	*result.avi*	*result2.avi*
pesan1.bmp	Utuh dan Terbaca	Utuh dan Terbaca
pesan2.bmp	Rusak	Rusak
pesan3.bmp	Utuh dan Terbaca	Utuh dan Terbaca
pesan4.bmp	Utuh dan Terbaca	Utuh dan Terbaca
pesan5.bmp	Utuh dan Terbaca	Utuh dan Terbaca

Berdasarkan tabel 4.3, video yang digunakan adalah video yang belum di kompresi. Karena pesan rahasia disisipkan di semua frame pada video, maka pada saat ekstraksi, pesan pada semua frame diekstraksi dalam 1 folder. Kemudian semua pesan terbaca kecuali pesan2.bmp, karena melebihi batas yang telah disisipkan.

Tabel 4.4 Keterangan Ekstraksi Pesan Rahasia Pada Kedua Video Kompresi

Pesan Rahasia	*result3.avi*	*result4.avi*
pesan1.bmp	Rusak	Rusak
pesan2.bmp	Rusak	Rusak
pesan3.bmp	Rusak	Rusak
pesan4.bmp	Rusak	Rusak
pesan5.bmp	Rusak	Rusak

Berdasarkan tabel 4.4, video yang diuji untuk diekstraksi pesannya adalah video yang sudah dikompresi. Semua pesan rusak dan tidak bisa terbaca, karena kualitas pada masing-masing video tersebut sudah menurun dan akan mempengaruhi cell pada video tersebut.

3.3 Pengujian Video Diberikan Serangan

Pada pengujian ini akan dibagi dalam 4 bagian, saat video di *rotate*, diberikan *noise Gaussian*, *noise Salt* and *Pepper*, dan *noise Speckle*. Kedua video sebelum diuji di konversi menjadi matriks, karena video yang pada dasarnya adalah cell, harus diubah menjadi matriks terlebih dahulu dengan menggunakan perintah 'cell2mat' agar bisa diuji. Perintah 'im*noise*' dan 'im*rotate*' hanya bisa dilakukan jika isi yang akan diuji tersebut adalah matriks, dan jumlah matriks harus sama.

Hasil pengujian dalam bentuk tabel yang ada di lampiran dan grafik.

3.3.1 *Noise Gaussian*

Pada pengujian *noise Gaussian*, kedua video yang telah disisipkan akan diberikan *noise* dengan varian mulai dari 0.01 hingga 0.09 dan dengan Mean 0.

Pada kedua video yang diberikan serangan noise Gaussian, nilai PSNR tidak ada perbedaan secara signifikan. Tetapi kualitas yang masih bisa diterima adalah saat kedua video dengan noise Gaussian varian 0.01 yang memiliki nilai PSNR diatas 20 dB, sedangkan untuk varian 0.02 hingga 0.09 kualitas video dan isi pesan rahasia menurun.

Untuk pengujian pada parameter BER, pesan yang telah diberikan noise Gaussian memiliki BER yang cukup besar, karena noise Gaussian memberikan kerusakan yang lebih pada gambar. BER terendah adalah pada varian 0.01 dengan nilai BER 0.1508 dan BER tertinggi adalah pada varian 0.09 dengan 0.201.

3.3.2 *Noise Salt & Pepper*

Sama seperti *noise Gaussian*, kedua video yang telah disisipkan akan diberikan *noise* dengan varian mulai dari 0.01 hingga 0.09.

Untuk PSNR pada kedua video yang sudah disisipkan pesan dan diberikan noise Salt & Pepper, hampir sama dengan Gaussian, tidak ada perbedaan secara signifikan. Video dengan kualitas masih bisa diterima adalah saat noise dengan varian 0.01 dan 0.02, karena kedua varian tersebut menghasilkan nilai PSNR diatas 20dB.

BER pada kedua video tidak teralu tinggi, terutama saat varian Salt and Pepper tinggi. Seperti saat varian 0.01, nilai BER hanya 0.0049 dan saat varian 0.09 nilai BER hanya 0.0459.

3.3.3 *Noise Speckle*

Sama seperti *noise Gaussian* dan *noise Salt* and *Pepper*, kedua video yang telah disisipkan akan diberikan *noise* dengan varian mulai dari 0.01 hingga 0.09.

Nilai PSNR pada video yang diberikan noise Speckle lebih besar dari noise Gaussian dan noise Salt and Pepper. Video dengan kualitas yang bisa diterima adalah pada saat varian 0.01 hingga 0.04, karena nilai PSNR diatas 20 dB. Tetapi untuk video test2.avi yang telah disisipkan pesan, karena pada semua varian memiliki nilai PSNR diatas 20, maka kualitasnya masih bisa diterima.

Untuk nilai BER pada pesan yang diberikan noise speckle, BER stabil di antara 0.004 dan 0.008. BER terendah adalah saat varian 0.07 dengan 0.004 dan BER tertinggi saat varian 0.04 dengan 0.0077.

3.3.4 Rotate

Pada pengujian dengan video yang diputar ini, video akan diputar sebesar 180°. Dihasilkan PSNR pada video test.avi yang disispkan pesan sebesar 13.744. Dan pada video test2.avi memiliki nilai PSNR 14.0419.

3.3.5 Mean Opinion Score (MOS)

Mean Opinion Score dilakukan dengan 30 koresponden. Masing-masing koresponden akan menonton empat video, 2 video yang belum diproses dan 2 video yang telah diproses. Hasil lengkap MOS akan dilampirkan di bagian lampiran.

Setelah MOS dilakukan, semua nilai akan dijumlahkan pada masing-masing video dan akan di cari rata-ratanya. Rata-rata tersebut akan menjadi nilai ukur kualitas pada video tersebut dengan subjektif.

Pada video test.avi yang disisipkan pesan memiliki rata-rata MOS 3.57, sehingga video ini masih memiliki kualitas yang baik dan tidak memiliki perubahan yang banyak.

Pada video test2.avi yang disisipkan pesan memiliki rata-rata MOS 4.23. Secara subjektif, video ini memiliki kualitas yang lebih baik dari video test.avi, dan hampir sama dengan video aslinya.

3.4 Analisis

Video yang disisipkan pesan rahasia menggunakan DCT ini akan rusak pesannya jika video tersebut di kompresi atau di beri noise, karena pada saat di kompresi atau di beri noise, akan mengubah seluruh cell pada frame di video tersebut.

Saat diberi noise, kualitas video yang masih diterima adalah saat nilai PSNR diatas 20dB [6]. Maka video yang masih dengan kualitas baik adalah saat diberikan *speckle*, karena saat diberikan noise *speckle*, dalam beberapa varian masih bertahan diatas 20dB. Sedangkan untuk noise *Gaussian* dan noise *Salt and Pepper* memiliki PSNR yang kurang baik, karena *Gaussian* akan memberikan sinyal acak pada video tersebut sehingga video akan terlihat blur, dan *Salt and Pepper* mengubah pixel warna pada video. Kedua noise tersebut lebih mengubah tatanan cell pada video tersebut.

Sedangkan dari nilai BER, pada pesan yang diberikan noise *Speckle* lebih baik. Nilai BER pesan yang diberikan noise *Speckle* masih dibawah 1 persen. Nilai BER yang baik berarti bit error pada pesan tersebut tidak banyak. Sehingga jika dibandingkan dengan noise *Gaussian* yang memiliki nilai BER hingga 20 persen dan noise *Salt and Pepper* yang memiliki nilai BER hingga 5 persen, pesan pada video dengan noise *Speckle* lebih baik.

Untuk analisis dengan steganografinya, saat video diberikan serangan, sesuai dengan BER yang di dapat, pesan akan rusak. Karena pesan akan bisa dibaca jika BER bernilai 0. Sedangkan pada pengujian, BER yang didapat diatas 0, maka pesan tidak bisa dibaca.

Pengujian pada video yang di putar, memiliki PSNR dibawah 20dB. Sehingga pesan di video tersebut akan rusak dan tidak terbaca.

Sesuai dengan nilai dan hasil MOS, analisis dengan steganografinya, kualitas pada video yang tersisip tidak akan banyak berubah. Sehingga pada saat di ekstraksi pesan, koresponden tersebut bisa mengetahui pesan tersebut dengan baik.

4. Kesimpulan

Berdasarkan hasil penelitian dan analisis data steganografi video dengan menyisipkan teks dan menggunakan metode Discrete Cosinus Transform, maka dapat ditarik kesimpulan sebagai berikut.

1. Metode DCT memungkinkan untuk menyisipkan teks tanpa harus dengan metode tambahan (LSB). Dengan membuat teks tersebut menjadi gambar.
2. Pada matlab, jika video yang sudah disisipkan tidak dikompresi akan menghasilkan size yang sangat besar.
3. Jika video yang telah disisipkan di kompresi atau diberikan serangan, maka akan mengakibatkan kerusakan pada pesan rahasia.
4. Sesuai analisis pada skenario, *noise* akan sangat mengurangi kualitas pada video yang akan membuat kualitas pesan menurun juga.
5. Untuk *noise*, hasil PSNR pada *Speckle* paling baik dengan nilai paling tinggi (30.0915), sedangkan untuk hasil BER *Speckle* paling baik (0.004). *Gaussian* lebih merusak video dan pesan pada video tersebut dengan PSNR (12.3985) dan BER (0.201).
6. Video yang di rotate tidak memiliki kualitas yang baik karena sesuai dengan analisis, memiliki PSNR dan BER yang buruk.
7. Sesuai dengan hasil penilaian subjektif MOS, video yang telah disisipkan pesan rahasia masih memiliki kualitas video yang baik.

Daftar Pustaka:

[1] Yadaf, Pooja, Nischol Mishra, dan Sanjeev Sharma. “A Secure Video Steganography with Encryption Based on LSB Technique.” 2013.
[2] Jianfei, Li, dan Sui Aina. “A Digital Video Watermarking Algorithm Based on DCT Domain.” 2012.
[3] Jianfeng, Lu, Yang Zhenhua, Yang Fan, dan Li li. “A MPEG2 Video Watermarking Algorithm Based on DCT Domain .” 2011.
[4] Ayyapan, Sonal, dan Aparna J R. “Comparison of Digital Watermarking Techniques.” 2014.
[5] Piarsa, I Nyoman. “Steganografi Pada Citra JPEG dengan Metode Sequential dan Spreading.” 2011.
[6] Thomos, Nikolaos, dan Strintzis, Michael G. “Optimized Transmission of JPEG2000 Streams Over Wireless Channels.” 2006
[7] http://www.radio-electronics.com/info/rf-technology-design/ber/bit-error-rate-tutorial definition.php
[8] http://www.mathworks.com/help/comm/ref/biterr.html