Analisa PSNR Pada Teknik Steganografi Menggunakan Spread Spectrum

Aries Pratiarso, Mike Yuliana, M. Zen Samsono Hadi, Fatchul Bari H., Brahim W.
Departemen Teknik Elektro
Politeknik Elektronika Negeri Surabaya
Kampus PENS, Jalan Raya ITS Sukolilo, Surabaya 60111
Tel : (031) 594 7280; Fax : (031) 594 6114
Email : aries@eepis-its.edu, mieke@eepis-its.edu, zenhadi@eepis-its.edu

Abstrak

Penelitian ini bertujuan untuk menganalisa pengaruh derau (noise) pada gambar yang terstego pada proses steganografi. Algoritma yang digunakan adalah spread spectrum dan parity coding, algoritma ini akan diterapkan pada proses steganografi untuk aplikasi e-commerce. Pesan rahasia yang distegokan meliputi nomor dan password kartu kredit. Data terstego berupa berkas gambar baru yang telah tersisipi data penting tersebut yang kemudian dikirim kembali ke server untuk proses pengembalian. Hasil akhir yang diperoleh dari penelitian ini adalah analisa ketahanan gambar jika terkena derau (noise) pada proses steganografi.Pengujian PSNR dilakukan 2 tahap, tahap pertama pada proses pembentukan gambar stego dari gambar asli, dimana kedua metode memberikan PSNR diatas 50 dB. Tahap kedua yaitu disimulasikan pada proses pengiriman gambar stego terdapat serangan MITM (man in the middle attack) dengan memberikan titik-titik hitam pada gambar, maka hasil PSNR langsung turun dibawah 30 dB. Pada pemberian titik hitam ini dilakukan pengujian sebanyak 10 titik hitam, pada metode parity coding, tidak bisa mengembalikan data yang disisipkan sedangkan pada metode spread spectrum masih bisa.

Kata kunci: *Spread Spectrum, Parity Coding, steganografi*

1. Pendahuluan

Pada era komunikasi digital seperti sekarang ini, transaksi penjualan dan pembelian barang ataupun jasa dapat dilakukan secara *online*.Transaksi semacam ini dikenal dengan istilah *e-commerce* atau *electronic commerce*. Dalam aplikasi *e-commerce* ini tentu mengharuskan adanya sistem keamanan yang menjamin segala data yang menyangkut transaksi seperti nomor kartu kredit dan kata sandi kartu kredit saat dilakukan proses transaksi. Hal ini dikarenakan data-data tersebut bersifat sangat rahasia dan rentan terhadap adanya transaksi yang tak diinginkan jika telah tersebar.

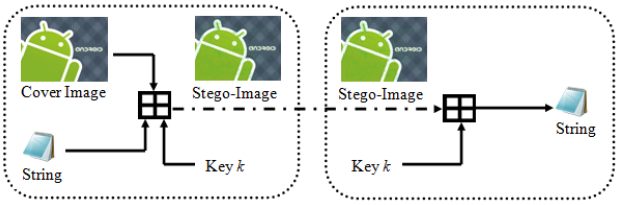
Sistem pengamanan yang digunakan di dunia pertelekomunikasian saat ini adalah kriptografi. Sistem kriptografi mempunyai salah satu kelemahan, dimana pesan rahasia akan tampak teracak sehingga dapat diketahui keberadaannya oleh pihak ketiga. Oleh karena itu diperlukan sistem steganografi yang mempunyai kelebihan yang akan menyisipkan pesan rahasia dalam sebuah *cover-image* sehingga pihak ketiga tidak akan mengetahui adanya pesan rahasia yang tersembunyi tersebut.

Terdapat berbagai metode dalam menjalankan teknik steganografi ini, antara lain: . Metode *spread spectrum* ini telah dikerjakan pada penelitian sebelumnya yang dengan menggunakan berkas audio sebagai *cover-object*-nya, yaitu oleh Riko Arlando Saragih [6]. Cara atau langkah menyisipkan pesan rahasia kedalam *cover-object* pun berbeda-beda, seperti yang telah diteliti oleh Ingemar J. Cox dan kawan-kawan menggunakan cara *watermarking*, yaitu melalui proses FFT (*Fast Fourier Transform*) dan IFFT (Inverse Fast Fourier Transform) agar pesan rahasia lebih merata penyisipannya [7] dan ada pula yang menggunakan langkah penyisipan pada LSB (*Least Significant Bit*)-nya saja, seperti yang pernah diteliti oleh M.A. Ineke Pakereng dan kawan-kawan [5]. Untuk serangan pada teknik steganografi, terdapat beberapa serangan yang dapat dijadikan referensi yaitu visual attack dan statistical attack [9].Metode steganografi ini juga diterapkan pada aplikasi m-banking [10]. Sedangkan analisa PSNR pada proses steganografi dengan metode LSB telah juga dilakukan [11].

Pada penelitian ini dibuat sebuah aplikasi *e-commerce* berbasis Android yang menggunakan sistem pengamanan steganografi dengan metode *spread spectrum* dan akan dibandingkan dengan metode *parity coding*serta dianalisa pada bagian PSNR-nya. Aplikasi ini akan menyisipkan (*embedding*) pesan penting menyangkut pertransaksian pada sebuah *cover-image* berupa berkas gambar. Sehingga hasilnya adalah terciptanya sebuah aplikasi *e-commerce* berbasis Android dimana memiki sistem keamanan yang handal.

2. Metode Spread Spectrum

Steganografi membutuhkan dua properti, yaitu pesan dan media penampung. Media penampung yang umumnya digunakan sekarang dapat berupa suara, gambar, atau video .Sedangkan pesan yang disembunyikan dapat berupa tulisan, gambar, atau pesan lainnya. Penjelasan singkat mengenai teknik dasar steganografi dapat dijelaskan sebagai berikut:



Gambar 1 Model sistem steganografi [4]

Dalam sistem steganograpfi *spread spectrum* ini, memiliki beberapa kelebihannya, anatara lain [3] :

1. penyembunyian sinyal,
2. komunikasi yang aman,
3. proteksi terhadap perusakan yang disengaja,
4. kecil kemungkinan untuk terdeteksi dan

Selain pada berkas gambar, steganografi *spread spectrum* ini sebelumnya juga pernah diteliti dan diuji pada berkas audio dengan hasil yang cukup memuaskan [7]. Steganografi *spread spectrum* ini juga dapat lebih diamankan dengan metode penyisipan *watermarking.* Penyisipan tersebut lebih sulit, karena menggunakan fungsi IFFT dan FFT dalam proses didalamnya [8]. Metode penyisipan lainnya juga dapat dilakukan dengan cara*Second LSB*.

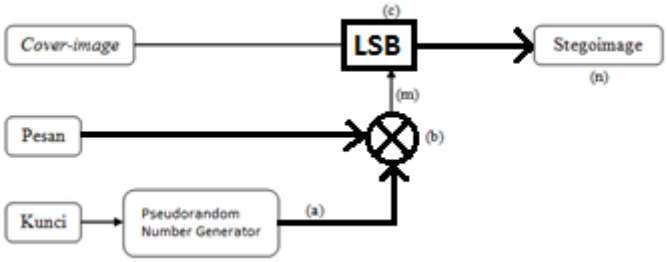
Penjelasan mengenai sistem kerja steganografi menggunakan metode *spread spectrum* ini dibagi menjadi dua bagian, bagian pertama adalah *embedding* (penyisipan) data informasi terhadap *cover-image* dan bagian kedua adalah *extraction* (pengambilan) terhadap *stego-image* yang digunakan. Didalam dua bagian tersebut, terdapat tahapan pembangkitan PRN (*PseudoRandom Number*) menggunakan metode LCG (*Linear Congruential Generator*) untuk membuat angka-angka semi random.

2.1 Proses penyisipan informasi ke *cover-image* pada steganografi *spread spectrum*

Steganografi *Spread Spectrum* bekerja dengan cara menyimpan sebuah pesan sebagai derau semu dalam sebuah gambar. Pada level daya derau rendah, degradasi sebuah *stego-image* tidak akan dapat dideteksi oleh mata manusia. Apabila level daya derau ditingkatkan, maka derau tersebut akan muncul seperti bercak-bercak atau ‘butiran salju’[5]. Langkah-langkah utama dalam proses penyisipan ini adalah sebagai berikut:

(a) Membangkitkan deretan *pseudorandom number* dengan menggunakan sebuah kunci,
(b) Bersama dengan pesan yg telah ter-*spreaded*, dilakukan proses XOR dengan deretan *pseudorandom number* yang telah dibangkitkan sebelumnya, menghasilkan suatu derau/*noise* (m),
(c) Menambahkan *noise* tersebut kedalam *cover-image* menggunakan metode LSB sehingga berubah menjadi *stego-image* (n).

Diagram alur proses penyisipan informasi tersebut ditampilkan pada Gambar 2 berikut :



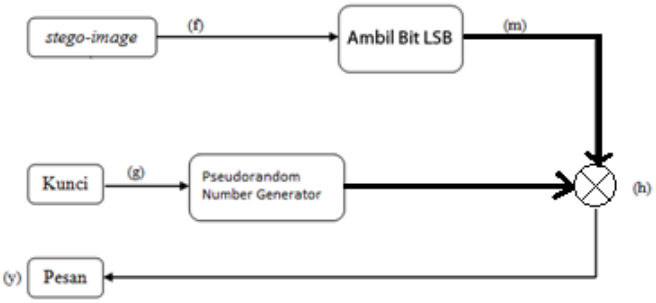
Gambar 2 Diagram alur proses penyisipan (*embedding*)

2.2 Proses pengambilan informasi dari *stego-image* pada steganografi *spread spectrum*

Gambar 3 menampilkan proses pengambilan informasi. Dalam hal ini, steganografi *spread spectrum* tidak membutuhkan berkas gambar aslinya untuk mengembalikan informasi tersembunyi didalamnya. Deretan perintah digunakan untuk mengambil LSB dari *stego-image*, menghasilkan sebuah deretan bit-bit yang mirip dengan *noise*.Semakin baik perintah ini bekerja, maka semakin sedikit kesalahan dalam informasi hasil pengambilan tersebut. Proses kebalikamn (pengambilan/*extraction*), mengambil dan mengembalikan informasi aslinya, tentu sangat mirip dengan proses penyisipannya:

a) Memproses *stego-image*, diambil bit LSB dari semua komponen warna, sehingga menghasilkan deretan bit-bit LSB yang mirip derau/*noise* (x),
b) Membangkitkan *pseudorandom number* yang sama, tentu dengan kunci yang sama dari proses penyisipan.
c) Meng-XOR-kan antara *noise* yang telah diambil dari langkah (f) tersebut dengan *pseudorandom number* yang telah dibangkitkan dari langkah (g) hingga menghasilkan pesan aslinya kembali (y) melalui proses *despreading*.

Berikut ini adalah diagram alur dari proses pengambilan (*extraction*) pada steganografi *spread spectrum* :



Gambar 3 Diagram alur proses pengambilan (*extraction*)

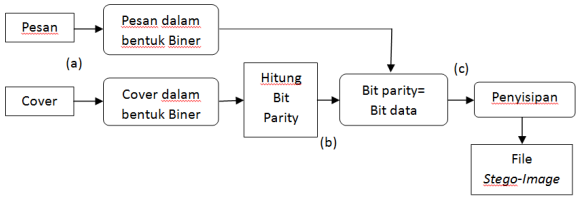
3. Metode Parity Coding

Teknik steganografi menggunakan metode *parity coding* adalah proses penghitungan bit-bit dengan

kondisi *even parity*. Hasil penghitungan dicek, apabila bit 1 berjumlah ganjil, maka nilai *parity* bitnya bernilai 1. Jika bit 1 berjumlah genap, maka nilai parity bitnya bernilai 0. Penjelasan mengenai sistem kerja steganografi menggunakan metode *parity coding* ini dibagi menjadi dua bagian, pada bagian pengirim yakni proses *embedding* (penyisipan) dan bagian penerima yakni *extraction* (pengambilan).

Data rahasia yang disisipkan pada metode *parity coding* merupakan teks dalam kode ASCII dan kombinasi bitnya disisipkan dalam *cover-image*. Diagram blok proses penyisipan data pada metode *parity coding* diperlihatkan pada Gambar 4. Secara umum cara kerja penyisipan data rahasia dari metode *parity coding* adalah menjumlahkan bit milik *cover-image* dan mencocokan dengan pesan rahasia, apabila tidak sama maka dilakukan perubahan bit LSB milik *cover-image*. Berikut merupakan prosedur kerja dari metode *parity coding*:

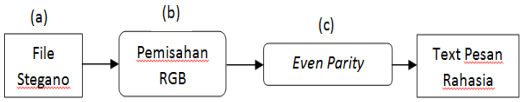
a) File pesan dan *cover-image* diubah dalam bentuk biner.
b) File *cover-image* dipilah dan dihitung sesuai RGB-nya secara *even parity*, agar dapat di gabungkan dengan file pesan.
c) Jika bit hasil penjumlahan *parity* tidak sama dengan satu bit dari pesan maka perlu diubah nilai LSB dari RGB (jika 1 diganti dengan 0, begitu pula sebaliknya), proses penyisipan dapat dilakukan karena bit keduanya telah sama, dan menghasilkan file *stego-image*.



Gambar 4. Diagram alur proses penyisipan (*embedding*)

Untuk mengekstraksi isi dari pesan ada dilakukan proses yang berkebalikan dengan proses penyisipan, yaitu melakukan ekstraksi dengan *parity coding*, setelah itu medekripsikan kembali agar bisa mendapatkan isi yang sebenarnya. Berikut merupakan proses dari metode *parity coding*:

a) Memilih file Stegano.
b) File Stegano dipilah sesuai dengan susunan RGB.
c) Setelah pemisahan RGB, dilakukan proses penghitungan ulang secara *even parity* pada setiap bit RGB, dan dari hasil *parity* tersebut merupakan nilai dari pesan rahasia.

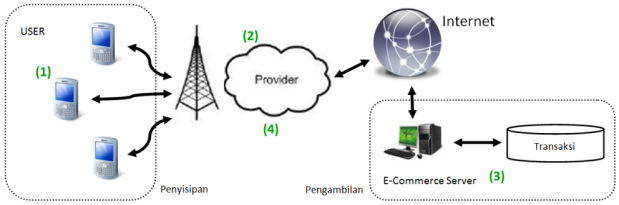


Gambar 5 Diagram alur proses pengambilan (*extraction*)

4. Bagan Sistem

Pada Penelitian ini dibuat sebuah sistem keseluruhan yang berisi integrasi dari keseluruhan sistem. Berikut penjelasan dari alur kerja dari sistem pada Gambar 6:

(1) Dari sisi *user*, terdapat aplikasi Android yang berperan menampilkan daftar produk, menangani pendaftaran anggota toko dan memproses steganografi untuk pengamanan data kartu kreditnya (nomor dan password).
(2) Data keseluruhan yang berupa total pembelanjaan dan *stego-image* dikirim ke *server* melalui *provider* dari masing-masing pengguna.
(3) Setelah data tersebut sampai di *server*, berkas *stego-image* diproses untuk mengeluarkan infomasi rahasia berupa nomor dan password kartu kredit. Lalu data detail kartu kredit dan total pembelanjaan disimpan dalam database untuk arsip.
(4) Setelah berhasil, konfirmasi keberhasilan pembelanjaan dikirimkan kembali ke *user*.



Gambar 6 Skema integrasi sistem keseluruhan

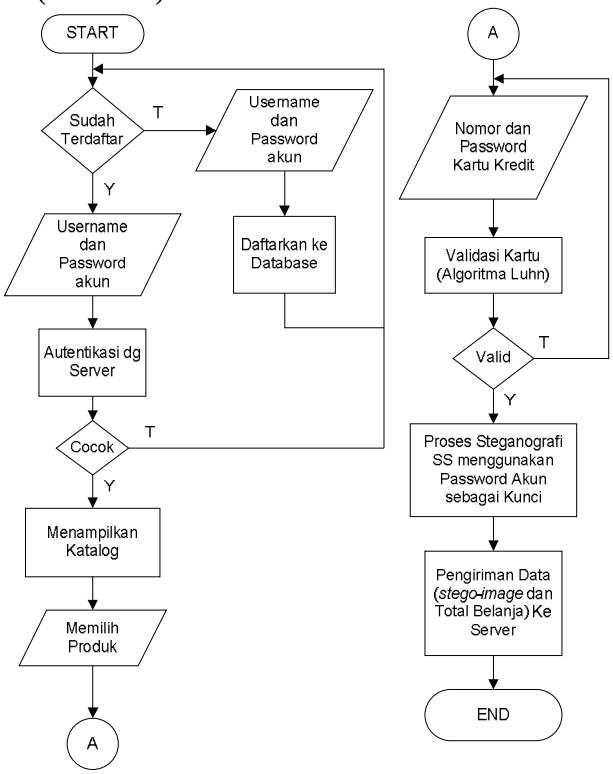
Sistem kerja aplikasi *e-commerce* ini perlu dirancang sedemikian rupa guna menghindari kebocoran data penting yang digunakan saat transaksi terjadi, dimana dalam penelitian ini data penting tersebut merupakan nomor kartu kredit/debit dan password milik pengguna.Gambar 6 menunjukkan bagaimana langkah pengimplementasian *e-commerce* dalam penelitian ini. Pada awalnya, aplikasi *Androi*d meminta detail akun *user* berupa *username* dan *password* yang telah teregistrasi sebelumnya untuk proses autentikasi, namun apabila *user* belum melakukan pendaftaran/registrasi di *e*-commerce ini maka *user* dapat memilih opsi "Daftarkan Diri" sehingga Aplikasi *Androi*dakan menampilkan *form* guna pendaftaran diri *user* tersebut.

Lalu apabila telah berhasil melewati proses autentikasi pada database *server*, *user*akan ditampilkan katalog produk-produk yang dapat dibeli. Kemudian *user* tersebut akan memilih barang-barang yang akan dibeli dengan variasi jenis produk dapat lebih dari satu. Pemilihan produk ini akan menghasilkan total pembelanjaan berdasarkan dari pemilihan sebelumnya, kemudian total pembelanjaan ini ditampilkan kembali guna konfirmasi kepada pengguna

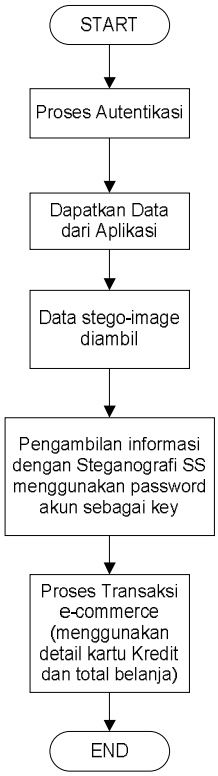
Apabila daftar pembelian telah dikonfirmasi oleh *user*, maka selanjutnya adalah memasukkan detail kartu kredit/debit (nomor kartu kredit/debit beserta *password* milik *user*) yang digunakan sebagai alat transaksi di *e-commerce* ini. Kemudian dilakukan

proses simulasi autentikasi kartu kredit/debit untuk memeriksa keabsahan dari kartu kredit/debit tersebut. Selanjutnya data-data tersebut dilakukan proses steganografi menggunakan teknik *spread spectrum* dengan password akun sebagai kuncinya, sehingga saat dilakukan proses pengiriman data-data penting tersebut menuju *server* menjadi aman.

Pada sisi lain, *server* hanya menunggu *request* data dari *client* saat transaksi dijalankan. Apabila *server* telah mendapatkan data transaksi dari *user*, kemudian data *stego-image*akan diproses tersendiri menggunakan steganografi *spread spectrum* dengan password akunsebagai kuncinya. Apabila telah valid, maka proses transaksi dapat segera dilakukan dengan penggabungan data detail pembelian dari*user* tadi. Penggunaan token tidak dilakukan karena penggunaan kunci disini bersifat simetris dan tetap (tidak acak).



Gambar 7 Flowchart implementasi disisi aplikasi *e-commerce* berbasis Android



Gambar 8 Flowchart implementasi di sisi *server* penyedia *e-commerce*

5. Hasil

Hasil dari aplikasi berbasis *e-commerce* adalah sebagai berikut :

Gambar 9 Aplikasi e-commerce pada *handphone*

Pada gambar di atas, terlihat 2 data yang tampil pada *handphone* jenis *android* yaitu nomor kartu kredit dan password yang akan dienkripsi datanya dengan teknik steganografi.

Pengujian pada sistem ini akan difokuskan pada perhitungan *Peak Signal to Noise Ratio* (PSNR) dan *Mean Square Error* (MSE). Dimana PSNR digunakan untuk mengetahui perbedaan nilai puncak sinyal dengan *noise*, *noise* yang dimaksud adalah gambar terstego (*stego-image*)sedangkan sinyal yang dimaksud adalah gambar asli atau (*cover-image*).Sedangkan MSE digunakan untuk mengetahui

perbedaan *error* antara gambar *stego-image* dengan *cover-image*.

Tabel 1 Daftar *cover-image*

Nama Berkas *Cover Image*	Tipe Gambar	Ukuran Berkas (Bytes)
cross	BMP	1.194
fb	JPG	700
help	PNG	3.336

Gambar 10 Cover-*image* (cross, fb, help)

Sedangkan teks rahasia yang akan di-*embed*-kan adalah 16 karakter nomor kartu kredit dan passwordnya.

5.1 Pengujian PSNR antara *cover-image* dengan *stego-image*

Pada penelitian ini dilakukan perhitungan PSNR dan MSE antara gambar awal (*cover-image*) dengan gambar terstego (*stego-image*) pada berbagai jenis gambar yang berbeda. Parameter yang digunakan antara lain: dimensi gambar adalah 19 x 19, jumlah *spreading* adalah 4 kali danjenis gambar yang diuji adalah BMP, PNG dan JPG. Sehingga pengujian ini mendapatkan korelasi antara jenis gambar berbeda dengan MSE dan PSNR yang dihasilkan oleh *stego-image*-nya.

Hasil pengujian dengan metode *spread spectrum* :

Tabel 2 Hasil pengujian PSNR dan MSE metode *spread spectrum*

Cover Img	Jenis Gbr	Output Gbr	MSE	PSNR (dB)
cross	BMP	PNG	0.468	51.446
fb	JPG		0.493	51.216
help	PNG		0.479	51.326

Berdasarkan ketiga data tersebut, dapat diamati bahwa perbedaan antar nilai MSE dan PSNR dari ketiga jenis gambar berbeda tersebut tidak terlalu signifikan, sehingga dapat dikatakan bahwa perbedaan jenis *cover-image* yang ada tidak terlalu berpengaruh terhadap perbedaan kualitas *stego-image*-nya.

Tabel 3 Hasil pengujian PSNR dan MSE metode *parity coding*

Cover Img	Jenis Gbr	Output Gbr	MSE	PSNR (dB)
cross	BMP	PNG	0,113	57,603
fb	JPG		0,084	58,902
help	PNG		0,1	58,121

Dari hasil di atas terlihat bahwa PSNR yang dihasilkan oleh *parity coding* lebih baik daripada *spread spectrum*, hal ini terjadi karena *parity coding* hanya mempengaruhi satu bit dibelakang saja dan ditempatkan pada piksel paling atas sedangkan pada metode *spread spectrum* berpengaruh pada seluruh piksel pada gambar.

5.2 PSNR dan MSE antara *stego-image* di sisi *client* dengan *stego-image* di sisi *server* yang telah dimodifikasi oleh MITM

Perhitungan PSNR dan MSE kali ini antara *stego-image* di sisi *client* yang akan dikirim dengan *stego-image* yang telah dimodifikasi oleh MITM di sisi *server*. Gambar terstego dari MITM tersebut telah dijelaskan sebelumnya pada subbab 4.15 sedangkan parameter yang digunakan untuk gambar terstegonya antara lain: dimensi gambar 76 x 76, jenis gambar terstego PNG dan jumlah *spreading* bervariasi (4 kali, 8 kali dan 16 kali). Berikut adalah hasil pengujian yang telah dilakukan:

Berikut hasil pengujian dengan metode *spread spectrum*:

Tabel 4 Hasil pengujian PSNR dan MSE *client* dengan MITM dengan metode *spread spectrum*

	Tipe Gambar	Ns	MSE	PSNR (dB)	Kbs
cross 76_ss stego	PNG	4 kali	225.118	25.599	Gagal
		8 kali	257.523	24.927	Berhasil
		16 kali	310.803	23.653	Berhasil

NB:
Ns : jumlah spreading
Kbs : keberhasilan sistem membaca teks yang tersembunyi

Berdasarkan data diatas dengan serangan jumlah 50 titik hitam yang sama, menunjukkan bahwa :

(1) serangan titik hitam sebanyak 50 titik mengakibatkan nilai MSE meningkat sampai angka ratusan sehingga nilai PSNR-nya pun kecil,
(2) semakin banyak jumlah *spreading* yang diterapkan pada gambar, walaupun dengan posisi titik yang acak mengakibatkan nilai PSNR-nya semakin menurun.

Kedua hal ini disebabkan karena apabila jumlah *spreading*-nya semakin besar, maka *pixel* yang tersisipi pesan rahasia juga semakin banyak, dengan demikian perubahan yang terjadi akibat penambahan titik hitam pun juga semakin besar, sehingga nilai PSNR-nya pun semakin menurun.Akan tetapi walaupun nilai PSNR menurun, metode ini masih berhasil membaca hasil teks yang tersembunyi pada jumlah *spreading*8x dan 16x. Pada jumlah *spreading* 4x ini sebenarnya bisa membaca teksnya tetapi ada beberapa karakter yang hilang.

Berikut hasil perhitungan dengan metode *parity coding*:

Tabel 5. Hasil pengujian PSNR dan MSE *client* dengan MITM

Stego-Image	**Nt**	**MSE**	**PSNR (dB)**	**Kbs**
cross_paritystego	5	413,223	22,644	Gagal
	10	954,305	18,729	Gagal
	15	1468,24	16,810	Gagal

NB:
Nt: Jumlah titik hitam yang ditambahkan pada bagian atas gambar
Kbs : keberhasilan sistem membaca teks yang tersembunyi

Berdasarkan pada Tabel di atas dapat dianalisa bahwa penambahan titik hitam yang diberikan oleh pihak MITM, menyebabkan nilai MSE yang semakin

tinggi, dan berbanding terbalik dengan nilai PSNR-nya yang semakin turun. Maka apabila terjadi serangan pada gambar *stego-image*akan berakibat pada penurunan kualitas gambar yang nantinya dikirimkan menuju ke *server*.

Disamping itu, teks yang tersembunyi juga gagal untuk dibaca walaupun jumlah titik yang ditambahkan hanya 5, dibandingkan dengan metode spread spectrum yang ditambahkan titik sampai 50 titik akan tetapi masih bisa membaca teks yang tersembunyi.

6. Kesimpulan

Kesimpulan yang dapat diambil pada pengujian adalah sebagai berikut :

1. Pada pengukuran PSNR dari *cover image* menjadi *stego image*, baik metode *spread spectrum* maupun *parity coding*menghasilkan PSNR di atas 50dB, dengan *parity coding* lebih baik.
2. Pada pengukuran PSNR yang disimulasikan ditambahkan titik hitam oleh MITM (*man in the middle attack*), kedua metode menghasilkan nilai PSNR dibawah 30dB, akan tetapi *spread spectrum* memiliki kelebihan yaitu masih sanggup membaca teks yang tersembunyi dibandingkan dengan *parity coding* yang gagal membaca.

Referensi

[1] Canggih Satriatama, "*Implementasi Dan Analisa Teknik Steganografi Multi-Carrier Pada Berkas Multimedia*", Tugas Akhir PENS –ITS, Surabaya : 2011.

[2] Munir, Rinaldi. "*Diktat Kuliah IF5054 Kriptografi*", Kuliah Umum ITB, Bandung : 2006.

[3] Yus Gias Vembrina, "*Spread Spectrum Steganography*", Tugas Akhir ITB, Bandung : 2006.

[4] Stefan Katzenbeisser and Fabien A. P. Petitcolas. "*Information Hiding Techniques for Steganography and Digital Watermarking*", Artech House, Boston, London : 2000.

[5] Marvel L.M., C.G. Boncelet Jr., dan C.T. Retter. "*Spread Spectrum Image Steganography*", IEEE Transactions on Image Processing, : 1999.

[6] M.A. Ineke Pakereng, Yos Richard Beeh, dan Sonny Endrawan, "*Perbandingan Steganografi Metode Spread Spectrum dan Least Significant Bit (LSB) Antara Waktu Proses dan Ukuran File Gambar*", Tugas Akhir UK Duta Wacana, Yogyakarta: 2010.

[7] Riko Arlando Saragih, "*Metode parity Coding Versus Metode Spread Spectrum pada Audio Steganography*", Tugas Akhir UK Maranatha, Bandung: 2006.

[8] Ingemar J. Cox, Joe Kilian, Tom Leighton dan Talal Shamoon, "*Secure Spread Spectrum Watermarking for Multimedia*", IEEE Trans. On Image Processng, 6, 12, 1663-1687, 1997.

[9] Andreas Westfeld, Andreas Pfitzmann, "*Attack on Steganographic Systems*", Lecture Notes in Computer Science, 2000.

[10] Shirali-Shahreza, Mohammad, "*Improving Mobile Banking Security Using Steganography*", IEEE Conference Publications, April 2007.

[11] Ajit Singh, Upasana Jauhari, "*A Symmetric Steganography with Secret Sharing and PSNR Analysis for Image Steganography*", International Journal of Scientific & Engineering Research Volume 3, June 2012.