

# Penyembunyian Pesan pada Citra Terkompresi *JPEG*

## Menggunakan Metode *Spread Spectrum*

Winda Winanti (13505017)

Teknik Informatika, Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Bandung  
e-mail: [if15017@students.if.itb.ac.id](mailto:if15017@students.if.itb.ac.id)

**Abstrak** - Keamanan dalam pengiriman informasi yang bersifat rahasia merupakan salah satu faktor penting yang harus dijaga. Salah satu teknik yang dapat dipakai untuk menangani hal tersebut adalah steganografi. Steganografi merupakan ilmu dan seni yang mempelajari cara penyembunyian informasi rahasia ke dalam suatu media sedemikian sehingga manusia tidak dapat menyadari keberadaan pesan tersebut.

Pada makalah ini, dilakukan studi mengenai bagaimana steganografi pada media citra digital. Citra digital yang digunakan adalah citra terkompresi dengan format file *JPEG*, oleh karena itu dilakukan pula studi terhadap bagaimana representasi dari citra terkompresi *JPEG*. Kemudian dilakukan juga implementasi terhadap hasil studi beserta analisis yang telah dikerjakan.

Perangkat lunak yang dikembangkan bernama *JtripleS*. Kakas pembangun yang digunakan adalah Java dengan IDE *NetBeans*.

**Kata kunci:** steganografi, citra *JPEG*, *Spread Spectrum*

### 1. PENDAHULUAN

Steganografi adalah teknik menyisipkan pesan ke dalam suatu media [1], dimana pesan rahasia yang akan dikirimkan tidak diubah bentuknya, melainkan disisipkan pada sebuah media lain (*cover-object*) yang digunakan dalam kehidupan sehari-hari. Media baru yang telah disisipi pesan rahasia (*stego-object*) kemudian dikirim kepada penerima tanpa menimbulkan kecurigaan dari pihak luar, karena perbedaan dari media asli (*cover-object*) dengan media yang telah disisipi pesan rahasia (*stego-object*) tidak dapat disadari secara langsung oleh manusia. Steganografi pada masa kini dilakukan pada media digital berupa citra, audio, maupun video.

Salah satu jenis media lain yang dapat digunakan adalah file citra digital dengan format *JPEG*. Format gambar *JPEG* ini merupakan format citra yang sangat umum dipergunakan karena berkas dengan format *JPEG* telah dikompresi dengan baik sehingga memiliki ukuran yang kecil. Ukuran yang

kecil ini menjadi alasan mengapa orang banyak menggunakan berkas dengan format *JPEG* ketika melakukan pertukaran di dalam media internet.

Ada dua buah proses dalam steganografi, yaitu proses penyisipan pesan dan ekstraksi pesan. Proses penyisipan pesan membutuhkan masukan media penyisipan, pesan yang akan disisipkan dan kunci. Keluaran dari proses penyisipan ini adalah media yang telah berisi pesan. Proses ekstraksi pesan membutuhkan masukan media yang telah berisi pesan. Keluaran dari proses ekstraksi pesan adalah pesan yang telah disisipkan.

Teknik steganografi yang digunakan adalah *Spread Spectrum*. Dengan metode ini pesan dikodekan dan disebar ke setiap spektrum frekuensi yang memungkinkan [2]. Metode *Spread Spectrum* mentransmisikan sebuah sinyal pita informasi yang sempit ke dalam sebuah kanal pita lebar dengan penyebaran frekuensi. Penyebaran ini berguna untuk menambah tingkat redundansi. Besaran redundansi ditentukan oleh faktor pengali  $cr$  yang bernilai skalar. Panjang bit-bit hasil penyebaran ini menjadi  $cr$  kali panjang bit-bit awal.

Dalam makalah ini juga akan dibahas mengenai dampak perubahan kualitas dari citra yang dihasilkan setelah penyisipan, yang akan diukur secara subjektif, dan objektif. Pengukuran secara subjektif dilakukan dengan pengamatan langsung, sedangkan pengukuran secara objektif dilakukan dengan menggunakan metode *Peak Signal to Noise Ratio* (PSNR) yang mengukur tingkat perbedaan dari citra yang telah disisipkan dengan yang belum.

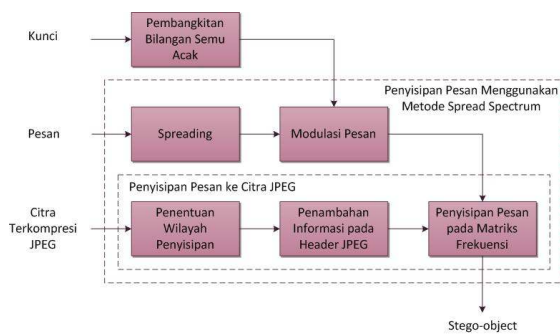
### 2. ANALISIS

Pada bagian ini akan dibahas mengenai proses penyisipan pesan, proses ekstraksi pesan, serta ukuran maksimum pesan yang dapat disisipkan. Terdapat juga analisis mengenai pembangkitan bilangan semu acak dan pengukuran kualitas dari citra.

#### 2.1 Penyisipan Pesan

Sistem untuk menyisipkan pesan pada citra terkompresi *JPEG* membutuhkan masukan berupa citra terkompresi *JPEG*, pesan yang ingin

disisipkan dan kunci yang akan digunakan untuk proses modulasi pesan. Skema penyisipan pesan dapat dilihat pada Gambar 1.



**Gambar 1** Skema penyisipan pesan

Proses penyisipan pesan menggunakan metode *Spread Spectrum* ini terdiri dari tiga proses, yaitu *spreading*, modulasi, dan penyisipan pesan ke citra *JPEG*. Pada awalnya dilakukan proses *spreading*.

Setelah itu dilakukan proses modulasi. Proses ini merupakan proses pengacakan pesan yang telah disebar dengan bilangan *pseudonoise* yang telah dibangkitkan menggunakan algoritma LCG. Panjang dari bilangan *pseudonoise* ini disesuaikan dengan panjang dari pesan. Jika panjang pesan lebih kecil dari panjang bilangan *pseudonoise*, bilangan *pseudonoise* tersebut akan dipotong sesuai dengan ukuran pesan. Sebaliknya, jika panjang pesan lebih besar dari panjang bilangan *pseudonoise*, maka bilangan tersebut akan diulang sampai panjangnya sama dengan panjang pesan. Proses modulasi tersebut dilakukan dengan menggunakan fungsi XOR (*Exclusive OR*). Nilai yang dihasilkan dari proses modulasi inilah yang kemudian akan disisipkan ke dalam berkas *JPEG*.

Proses selanjutnya adalah penyisipan pesan ke dalam citra *JPEG*. Penyisipan pesan ke dalam citra terkompresi *JPEG* terdiri dari tiga proses, yaitu penentuan wilayah penyisipan, penambahan informasi pesan pada *header JPEG*, dan penyisipan pesan pada matriks frekuensi.

Pada berkas *JPEG* terdapat matriks frekuensi berukuran 8 x 8 yang dapat disisipkan oleh pesan. Untuk mendapatkan hasil yang efektif, penyisipan hanya dapat dilakukan pada *low* dan *mid-frequency* yang terdapat pada matriks frekuensi tersebut.

Untuk menentukan *low* dan *mid-frequency* yang akan memberikan hasil yang optimal, dipilih dua buah nilai sekuensial dari hasil pemindaian matriks secara *zig-zag*,  $o_1$  dan  $o_2$  dimana  $1 < o_1 < o_2 < 64$ . Untuk setiap pasangan nilai  $o_1$  dan  $o_2$  dihitung nilai *peak signal noise ratio* (PSNR) dengan menggunakan perhitungan sebagai berikut

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|I(i, j) - K(i, j)\|^2$$

Dimana:

- $MSE$  : nilai *mean square error* dari citra
- $m$  : panjang citra dalam piksel
- $n$  : lebar citra dalam piksel
- $i, j$  : koodinat masing-masing piksel
- $I$  : nilai bit citra pada koordinat  $i, j$
- $K$  : nilai derajat keabuan pada koordinat  $i, j$

$$PSNR = 10 \cdot \log_{10} \left( \frac{MAX_I^2}{MSE} \right)$$

Dimana:

- $PSNR$  : nilai PSNR citra
- $MAX_I$  : nilai maksimum piksel
- $MSE$  : nilai MSE

Kemudian pasangan nilai  $o_1$  dan  $o_2$  yang menghasilkan nilai PSNR paling besarlah yang akan dipilih untuk melakukan penyisipan pesan.

Setelah wilayah penyisipan didapatkan, selanjutnya dilakukan penambahan informasi pada *header* dari berkas *JPEG* yang dijadikan media penyisipan pesan. Informasi yang ditambahkan adalah nama berkas yang akan disisipkan, ukuran berkas yang akan disisipkan, wilayah penyisipan pesan, dan faktor bilangan pengali yang digunakan. Untuk keamanan, informasi tersebut dienkripsi sebelum ditambahkan pada *header* dari berkas *JPEG*. Informasi-informasi tersebut ditambahkan dengan menambahkan *tag XML* baru yang akan digunakan sebagai media penyimpanan informasi pada *header* berkas *JPEG*.

Setelah proses penambahan informasi selesai dilakukan, selanjutnya dilakukan tahap terakhir dalam penyisipan pesan pada citra terkompresi *JPEG*. Tahap terakhir tersebut adalah penyisipan pesan pada matriks frekuensi. Pesan yang akan disisipkan dalam tahap ini adalah hasil dari proses modulasi yang telah dilakukan sebelumnya.

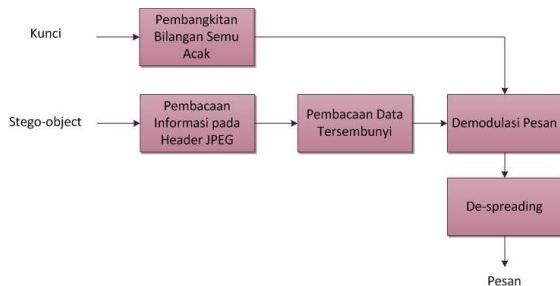
Penyisipan pesan pada matriks frekuensi dilakukan dengan cara menyisipkan bit pesan pada bit terakhir dari nilai yang terdapat di matriks frekuensi. Penyisipan tidak dapat dilakukan pada nilai 0, 1, dan 255 karena perubahan pada ketiga nilai tersebut dapat mengakibatkan berubahnya susunan matriks frekuensi. Penyisipan juga harus dilakukan sesuai dengan cara pembacaan matriks frekuensi, yaitu *zig-zag*.

Hal lain yang perlu diperhatikan dalam menyisipkan pesan pada matriks frekuensi adalah pembagian penyisipan yang merata pada seluruh matriks frekuensi yang terdapat pada berkas *JPEG*. Untuk itu penyisipan akan dilakukan secara selang-seling berdasarkan jumlah matriks frekuensi yang ada pada berkas *JPEG* tersebut.

## 2.2 Ekstraksi Pesan

Sistem untuk mengekstraksi pesan pada citra terkompresi *JPEG* membutuhkan masukan berupa citra terkompresi *JPEG* yang telah disisipkan pesan dan kunci yang akan digunakan untuk proses demodulasi pesan. Skema penyisipan pesan dapat dilihat pada Gambar 2.

Proses ekstraksi pesan menggunakan metode *Spread Spectrum* ini terdiri dari tiga proses, yaitu pengambilan pesan dari matriks frekuensi, demodulasi, dan de-*spreading*.



Gambar 2 Skema penyisipan pesan

Proses pengambilan pesan dari matriks frekuensi diawali dengan pembacaan informasi yang terdapat pada *header JPEG* ini dilakukan terhadap isi dari *tag XML* yang telah didefinisikan khusus sebelumnya. Adapun informasi yang didapatkan dari pembacaan ini adalah nama berkas yang disisipkan, ukuran berkas yang disisipkan, wilayah penyisipan pesan, dan faktor bilangan pengali yang digunakan.

Kemudian dilakukan pembacaan data tersembunyi dilakukan berdasarkan informasi wilayah penyisipan pesan yang didapatkan dari tahap sebelumnya, yaitu pembacaan informasi pada *header JPEG*. Pembacaan akan dilakukan secara berselang-seling pada matriks frekuensi yang terdapat pada citra dan berlangsung sampai data yang dibaca besarnya sama dengan informasi ukuran berkas yang disisipkan.

Setelah data tersembunyi berhasil dikumpulkan, dilakukan proses demodulasi terhadap data tersebut. Proses demodulasi ini melibatkan bilangan acak yang dibangkitkan dari kunci masukan menggunakan algoritma LCG. Adapun proses pembangkitan bilangan acak yang dilakukan pada tahap ekstraksi pesan sama seperti proses pembangkitan bilangan acak pada tahap penyisipan pesan.

Hasil dari proses demodulasi tersebut akan mengalami proses de-*spreading*. Proses de-*spreading* ini adalah proses yang dilakukan untuk mendapatkan bit-bit dari pesan tersembunyi. Proses de-*spreading* ini bekerja menggunakan faktor besaran pengali yang dimasukkan oleh pengguna pada proses penyisipan pesan.

## 2.3 Ukuran Maksimum Pesan yang Dapat Disisipkan

Pesan disisipkan kedalam LSB dari nilai yang terdapat pada matriks frekuensi. Ukuran pesan yang disisipkan bergantung pada 4 hal, yaitu banyaknya matriks frekuensi yang terdapat pada citra terkompresi *JPEG* yang menjadi media penyisipan pesan, banyaknya matriks yang merupakan *header* dari citra terkompresi *JPEG*, faktor besaran pengali *cr* yang dipilih, dan besar wilayah penyisipan pesan pada sebuah matriks frekuensi. Masing-masing berkas *JPEG* memiliki jumlah matriks frekuensi yang berbeda sesuai ukuran dari berkas *JPEG* tersebut, sedangkan faktor besaran pengali *cr* dapat diubah-ubah sesuai keinginan pengirim pesan.

Dengan memperhitungkan bahwa nilai 0, 255, 1 pada matriks frekuensi tidak dapat disisipkan pesan, akan didapatkan perhitungan sebagai berikut

$$8 \times p \times cr = ((m - h) \times w) - i$$

Dimana

- p* : ukuran pesan yang dapat disisipkan
- cr* : faktor besaran pengali
- m* : jumlah matriks frekuensi pada berkas *JPEG*
- h* : jumlah matriks yang merupakan *header* dari berkas *JPEG*
- w* : besar wilayah penyisipan pada tiap matriks frekuensi
- i* : jumlah nilai 0, -1, 1 yang terdapat pada wilayah penyisipan

## 2.4 Pembangkitan Bilangan Semu Acak

Pada steganografi, pembangkitan bilangan acak dapat digunakan untuk menentukan kunci penyisipan dan ekstraksi data dari berkas media. Komputer mampu menghasilkan bilangan semu acak (*pseudorandom*). Deret bilangan *pseudorandom* adalah deret bilangan bilangan yang kelihatan acak dengan kemungkinan pengulangan yang sangat kecil atau periode pengulangan yang sangat besar.

Salah satu algoritma pembangkitan bilangan pseudorandom adalah *Linear Congruential Generator* (LCG). Algoritma LCG ini diciptakan oleh D.H. Lehmer pada tahun 1951. Deret bilangan bulat dalam LCG dirumuskan sebagai berikut:

$$Z_i = (aZ_{i-1} + c) \text{ mod } m$$

Dalam hal ini:

- $Z_i$  : bilangan bulat ke- *i*
- a* : bilangan pengali
- c* : bilangan penambah
- m* : modulus
- $Z_0$  : nilai awal berupa bilangan bulat positif

Untuk memulai bilangan acak ini dibutuhkan sebuah bilangan bulat  $Z_0$ , yang dijadikan sebagai nilai awal (bibit pembangkitan). Bilangan acak pertama yang dihasilkan selanjutnya menjadi bibit pembangkitan bilangan bulat acak selanjutnya. Jumlah bilangan acak yang tidak sama satu sama lain (unik) adalah sebanyak  $m$ . Semakin besar nilai  $m$ , semakin kecil kemungkinan akan dihasilkan nilai yang sama.

## 2.5 Pengukuran Kualitas Citra

Penilaian kualitas citra terkompresi *JPEG* dilakukan secara subjektif dan objektif. Penilaian subjektif dilakukan dengan cara melihat citra secara kasat mata. Penilaian objektif dilakukan dengan cara menghitung nilai PSNR (*Peak Signal to Noise Ratio*). Nilai PSNR dalam satuan desibel (dB) dihitung berdasarkan persamaan:

$$PSNR = 10 \cdot \log_{10} \left( \frac{MAX_I^2}{MSE} \right)$$

Dimana:

$PSNR$  : nilai PSNR citra  
 $MAX_I$  : nilai maksimum piksel  
 $MSE$  : nilai MSE

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|I(i,j) - K(i,j)\|^2$$

Dimana:

$MSE$  : nilai *mean square error* dari citra  
 $m$  : panjang citra dalam piksel  
 $n$  : lebar citra dalam piksel  
 $i, j$  : koordinat masing-masing piksel  
 $I$  : nilai bit citra pada koordinat  $i, j$   
 $K$  : nilai derajat keabuan pada koordinat  $i, j$

Nilai PSNR yang wajar pada perbandingan dua berkas citra adalah 30-50 dB.

## 3. HASIL DAN PEMBAHASAN

Berdasarkan hasil analisis, perangkat lunak yang memiliki fungsi menyisipkan pesan dan ekstraksi pesan pada citra terkompresi *JPEG* menggunakan metode *Spread Spectrum* telah berhasil dikembangkan dengan baik. Selanjutnya, dilakukan pengujian untuk memeriksa kebenaran perangkat lunak serta menguji kinerja dari perangkat lunak tersebut.

Untuk menguji kebenaran perangkat lunak dilakukan tiga buah kasus uji, yaitu menguji kebenaran proses penyisipan dan ekstraksi, menguji proses penggunaan kunci, dan menguji pengaruh dari penggunaan faktor bilangan pengali. Sedangkan untuk menguji kinerja perangkat lunak

dilakukan pengujian terhadap dampak dari proses penyisipan pada citra.

### 3.1 Menguji Kebenaran Proses Penyisipan dan Ekstraksi (Kasus Uji 1)

Pengujian ini dilakukan dengan cara menyisipkan pesan ke dalam citra terkompresi *JPEG*, kemudian mengekstraksinya kembali. Citra yang menjadi media adalah citra pada Gambar 3 (*stone.jpg*). *File* pesan masukannya adalah sebuah *file* teks dan *file* gambar, yang isinya ditunjukkan pada Tabel 1. Kunci yang digunakan adalah 11, faktor bilangan pengali yang digunakan adalah 1 dan citra keluaran diberi nama baru yang sesuai dengan *file* pesan.



Gambar 3 Citra media penyisipan

Tabel 1 Isi *file* penyisipan

Jenis <i>file</i>	Isi <i>file</i>
<i>File</i> teks	Joint Photographic Experts Group (JPEG) adalah format gambar yang banyak digunakan untuk menyimpan gambar-gambar dengan ukuran lebih kecil.
<i>File</i> gambar	

Setelah proses penyisipan selesai, dilakukan proses ekstraksi dari masing-masing citra yang telah disisipkan pesan. Kunci dan faktor bilangan pengali yang digunakan sama dengan proses penyisipan, yaitu 11 dan 1, sehingga diharapkan pesan yang dihasilkan dari proses ekstraksi memiliki isi yang sama dengan pesan yang digunakan pada proses penyisipan. Isi dari *file* hasil ekstraksi dapat dilihat pada Tabel 2.

Tabel 2 Isi *file* ekstraksi

Jenis <i>file</i>	Isi <i>file</i>
<i>File</i> teks	Joint Photographic Experts Group (JPEG) adalah format gambar yang banyak digunakan untuk menyimpan gambar-gambar dengan ukuran lebih kecil.
<i>File</i> gambar	

Dari hasil pengujian terbukti bahwa perangkat lunak yang dibuat sudah berhasil menjalankan proses penyisipan dan ekstraksi dengan benar. Semua pesan yang menjadi masukan telah berhasil disisipkan, dan kemudian dapat diekstraksi dengan baik. Pesan yang diekstraksi sama dengan pesan yang asli, dan kunci yang digunakan pada proses penyisipan dan ekstraksi juga sama.

### 3.2 Menguji Kebenaran Proses Penyisipan dan Ekstraksi (Kasus Uji 2)

Pengujian ini dilakukan dengan cara melakukan ekstraksi dari citra hasil penyisipan pada kasus uji 1, namun dengan kunci yang berbeda, yaitu 22. Isi dari file hasil ekstraksi dapat dilihat pada Tabel 3.

Tabel 3 Isi file penyisipan

Jenis file	Isi file
File teks	뽁< 穎祛司<뽁鍊嚮궁·뽁 釘 嚮誤 諫員 闡覽 用 枙張瘖 민 ·肝 兪 俞 砌 辟 梵 彤 日 嚮 嚮
File gambar	Gambar tidak dapat dimunculkan

Dari hasil pengujian, terbukti bahwa perangkat lunak yang dibuat dapat melakukan aspek penggunaan kunci dengan baik. Proses ekstraksi dengan kunci yang salah dapat ditangani, yaitu dengan menghasilkan pesan yang berbeda dengan pesan yang asli.

### 3.3 Menguji Pengaruh dari Penggunaan Faktor Bilangan Pengali (Kasus Uji 3)

Pengujian ini dilakukan untuk menguji pengaruh dari penggunaan faktor bilangan pengali, yaitu dengan cara memasukkan faktor bilangan pengali yang berbeda dengan masukan pesan dan gambar yang sama dengan kasus uji 1. Pengujian akan berhasil apabila terdapat perbedaan ukuran maksimum pesan yang dapat disisipkan. Hasil dari pengujian ini dapat dilihat pada Tabel 4.

Tabel 4 Hasil pengujian kasus uji 3

Jenis file	Faktor bilangan pengali	Keluaran citra
File teks	1	stone_jpeg1.jpg
	2	stone_jpeg2.jpg
	3	stone_jpeg3.jpg
	4	Pesan tidak dapat disisipkan
File gambar	1	stone_facebook1.jpg
	2	Pesan tidak dapat disisipkan

Dari hasil pengujian, terbukti bahwa perangkat lunak yang dibangun penggunaan faktor bilangan

pengali berpengaruh terhadap ukuran maksimum pesan yang dapat disisipkan. Semakin besar nilai dari faktor bilangan pengali, semakin kecil ukuran pesan yang dapat disisipkan. Hal ini dikarenakan pada proses penyisipan pesan dengan metode *Spread Spectrum* pesan akan digandakan sebanyak faktor bilangan pengali.

### 3.4 Menguji Dampak Penyisipan Pesan pada Citra (Kasus Uji 4)

Pengujian ini dilakukan untuk menguji kualitas dari citra hasil penyisipan, yaitu dengan membandingkannya dengan citra yang asli. Perbandingan ini memakai masukan dari citra terkompresi *JPEG* yang telah melewati proses penyisipan pada Kasus uji 1, dan menggunakan dua cara perbandingan, yaitu subjektif dan objektif. Pengujian akan berhasil apabila dari masing-masing cara, didapatkan hasil seperti berikut:

- Pada cara subjektif, citra dianggap mirip.
- Pada cara objektif, nilai *Peak Signal Noise Ratio* (PSNR) yang didapatkan berada diantara 30 dan 50. Hasil perhitungan nilai PSNR dapat dilihat pada Tabel 6.

Tabel 5 Hasil perhitungan nilai PSNR

Jenis file pesan	Nilai PSNR (dB)
File teks	37.40943307
File gambar	42.02490002

Pada cara subjektif, semua citra hasil penyisipan dianggap mirip dengan citra yang asli. Pada cara objektif, yaitu dari Tabel 5 dapat dilihat bahwa semua perbandingan memperoleh nilai *Peak Signal Noise Ratio* (PSNR) diantara 30 dan 50. Hal ini membuktikan bahwa proses penyisipan dengan metode *Spread Spectrum* ini tidak mengubah kualitas struktur citra secara signifikan.

## 4. KESIMPULAN

Beberapa kesimpulan yang dapat diambil adalah sebagai berikut:

1. Telah berhasil dikembangkan perangkat lunak yang dapat melakukan steganografi pada citra terkompresi *JPEG*. Kebutuhan fungsional dari perangkat lunak, seperti proses penyisipan dan ekstraksi pesan, serta penggunaan kunci sudah dapat dilakukan dengan benar.
2. Metode *Spread Spectrum* sebagai metode penyisipan pesan sudah dapat dilakukan dengan benar, yaitu melakukan proses *spreading* terhadap pesan, modulasi pesan dengan kunci, dan penyisipan pesan dalam matriks frekuensi yang terdapat pada citra terkompresi *JPEG*.

3. Kualitas citra terkompresi *JPEG* yang dihasilkan bergantung dari besarnya ukuran pesan. Berdasarkan pengamatan yang dilakukan saat pengujian, citra *JPEG* yang disisipkan lebih banyak akan mengalami perubahan yang lebih besar.

#### DAFTAR REFERENSI

- [1] Cole, Eric. 2003. *Hiding in Plainsight : Steganography and the Art of Covert Communication*. Wiley Publishing, Inc.
- [2] Torrieri, Don. 2005. *Principles of Spread Spectrum Communications System*. Springer.