

Penyembunyian Pesan pada Citra *GIF* Menggunakan Metode Adaptif

Prasetyo Andy Wicaksono (13505030)

Teknik Informatika, Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
e-mail: prasetyoandyw@gmail.com

Abstrak - Steganografi merupakan ilmu dan seni yang mempelajari teknik dan cara penyembunyian pesan rahasia ke dalam suatu media sedemikian rupa sehingga pihak ketiga tidak dapat melihat dan menyadari keberadaan pesan rahasia tersebut. Makalah ini membahas studi mengenai bagaimana steganografi pada media citra digital citra *GIF*.

Terdapat juga sebuah perangkat lunak yang dihasilkan, yang memiliki fungsi utama untuk melakukan steganografi pada citra *GIF* menggunakan metode adaptif. Perangkat lunak menggunakan kunci yang dibangkitkan oleh pembangkit bilangan acak semu, yang digunakan pada proses penyisipan dan ekstraksi pesan.

Kata kunci: steganografi, citra *GIF*, metode adaptif

1. PENDAHULUAN

Steganografi adalah ilmu dan seni penyembunyian informasi yang dapat mencegah pendeteksian terhadap informasi yang disembunyikan [1]. Steganografi dapat dilakukan pada media digital berbentuk citra, video maupun audio. Namun umumnya steganografi dilakukan pada berkas citra.

Salah satu format berkas citra digital yang digunakan adalah format gambar *GIF*. Format ini adalah salah satu format gambar yang umum digunakan karena ukurannya yang cukup kecil dan dapat menampilkan animasi sederhana. Format citra *GIF* mempunyai dua buah versi, yaitu *GIF87a* dan *GIF89a* [2].

Secara mendasar terdapat dua buah macam metode penyisipan informasi pada format *GIF*, yaitu metode non adaptif dan adaptif [3]. Metode non adaptif menyisipkan pesan rahasia ke dalam format gambar *GIF* tidak berkorelasi pada fitur gambar. Contohnya metode LSB yang memilih secara acak piksel yang ingin disisipkan oleh pesan rahasia, contoh lain adalah *EZStego* dan *GifShuffle*. Sedangkan metode adaptif mengkorelasikan modifikasi pada tiap piksel

sesuai konten gambar. Salah satu keunggulan metode adaptif adalah dapat menanggulangi perubahan warna yang drastis pada area dengan warna yang sama.

Dalam makalah ini akan juga dibahas mengenai dampak perubahan kualitas dari citra *GIF* yang dihasilkan setelah penyisipan, yang diukur secara subjektif (kasat mata), dan objektif. Metode objektif menggunakan metode *PSNR* (*Peak Signal to Noise Ratio*) yang mengukur tingkat perbedaan citra setelah disisipkan pesan dengan yang belum.

2. ANALISIS

Pada bagian ini akan dibahas mengenai proses penyisipan dan ekstraksi pesan. Terdapat juga analisis mengenai ukuran maksimum pesan.

2.1 Penyisipan Pesan

Untuk melakukan penyisipan pesan pada citra *GIF*, dibutuhkan masukan berupa citra yang akan disisipkan pesan, pesan yang akan disisipkan (*message*), dan kunci (*key*).

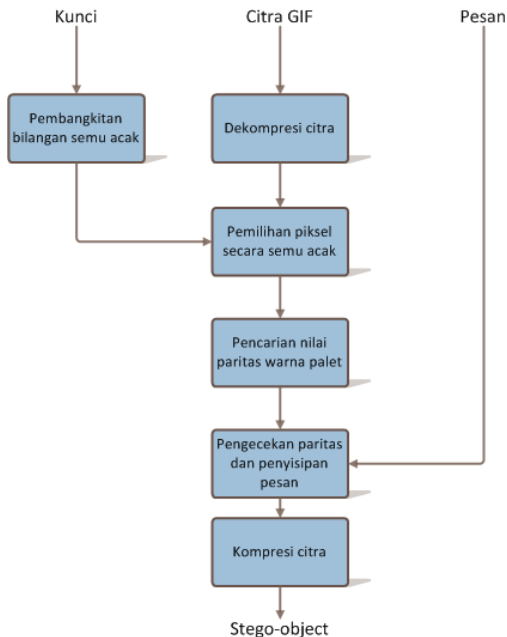
Proses penyisipan pesan, dapat dilihat skema kerjanya pada Gambar 1, terdiri dari beberapa tahapan. Pada awalnya dilakukan pembangkitan bilangan semu acak menggunakan algoritma *Linear Congruential Generator* (LCG).

Setelah itu dilakukan proses dekompresi citra, dan setelah itu dilakukan pemilihan piksel secara semu acak, sesuai dengan bilangan semu acak yang telah dibangkitkan. Setelah itu dilakukan pencarian nilai paritas warna palet dari piksel-piksel yang telah dipilih. Setelah nilai paritas dari warna palet sebuah citra *bitmap* ditentukan, proses penyisipan pesan berlanjut pada tahap pengecekan paritas. Piksel yang telah dipilih secara semu acak menjadi tempat penyisipan pesan akan dicek nilai paritasnya dengan bit pesan. Jika perbandingan menghasilkan persamaan bit, maka piksel tersebut tidak dimodifikasi dan pengecekan berlanjut kepada piksel

selanjutnya. Jika terjadi perbedaan bit, maka piksel tersebut akan dimodifikasi warnanya dengan cara mencari warna tetangga terdekat menggunakan persamaan jarak:

$$d = \sqrt{(R_2 - R_1)^2 + (G_2 - G_1)^2 + (B_2 - B_1)^2}$$

yang memiliki nilai paritas yang berbeda.



Gambar 1 - Skema Penyisipan Pesan

Warna tetangga terdekat mempunyai nilai d yang terkecil dibanding warna awal dan jika pada awalnya piksel tersebut mempunyai paritas warna palet 1, maka dicari warna tetangga terdekat yang mempunyai nilai paritas 0. Lalu warna tetangga terdekat tersebut akan menggantikan warna awal pada piksel tersebut.

Pada proses penyisipan pesan perlu diketahui pula ukuran pesan yang disisipkan. Citra *GIF* memiliki fitur *comment extension* yang merupakan suatu blok pada *data stream* pada berkas citra *GIF* yang bertujuan untuk memperbolehkan sistem pengkode citra *GIF* untuk menambahkan *metadata* ataupun informasi tambahan mengenai berkas citra *GIF* tersebut.

Informasi yang ditambahkan ke dalam *comment extension* adalah ukuran pesan dan nama berkas yang berisi pesan yang disisipkan. Hal ini diperlukan untuk memberi informasi akhir pesan kepada sistem pengestraksian pesan. Informasi akhir pesan akan menjadi batas akhir pembacaan pesan saat proses ekstraksi. Pembacaan pesan akan dilakukan

maksimal sebanyak ukuran pesan. Nama berkas pesan yang disisipkan disimpan pada *comment extension* untuk keperluan pengestraksian pesan.

Setelah proses penyisipan pesan selesai, dilakukan proses kompresi citra untuk mengembalikannya ke dalam bentuk citra *GIF*.

Dari hasil analisis didapatkan bahwa setiap piksel pada citra *GIF* dapat disisipkan oleh satu buah bit pesan, sehingga maksimum jumlah ukuran pesan yang dapat disisipkan pada citra *GIF* menggunakan metode adaptif adalah sesuai dengan persamaan berikut.

$$p = \frac{m \times n}{8}$$

Dimana,

- p : ukuran pesan
- m : ukuran panjang citra *GIF* dalam piksel
- n : ukuran lebar citra *GIF* dalam piksel

2.2 Ekstraksi Pesan

Untuk ekstraksi pesan dari citra *GIF* menggunakan metode adaptif membutuhkan masukan berupa *stego-object*, dan kunci (*key*).



Gambar 2 - Skema Ekstraksi Pesan

Proses ekstraksi pesan, yang skema kerjanya dapat dilihat pada Gambar 2, diawali dengan pembangkitan bilangan semu acak menggunakan algoritma *LCG* dan kunci yang sama dengan kunci pada proses

penyisipan. Setelah itu dilakukan dekomposisi citra *GIF*.

Setelah citra didekomposisi, dilakukan pembacaan informasi pada *comment extension* yang merupakan fitur pada citra *GIF* yang memperbolehkan penggunaanya menambahkan informasi dalam bentuk ASCII ke dalam berkas. Setelah informasi dibaca, dilakukan penelusuran piksel sesuai dengan bilangan semu acak yang dibangkitkan. Terakhir dilakukan pencarian nilai paritas dari piksel tersebut dan pengambilan pesan dengan cara menggabungkan nilai paritas dari piksel yang disisipkan pesan.

2.3 Pembangkitan Bilangan Semu Acak

Bilangan semu acak dibutuhkan pada proses penyisipan untuk menjadi acuan piksel mana saja yang akan disisipkan oleh bit pesan. Untuk memenuhi kebutuhan tersebut bilangan semu acak dibangkitkan dengan menggunakan algoritma *Linear Congruential Generator* (LCG), yang persamaannya sebagai berikut.

$$x_n = (ax_{n-1} + b) \text{ mod } m$$

Pada kasus ini nilai X_0 yang digunakan adalah masukan pengguna dan nilai modulus (m) sama dengan ukuran berkas citra *GIF* yang akan disisipkan pesan, agar tidak terjadi pengulangan bilangan semu acak yang dibangkitkan.

2.4 Pengukuran Kualitas Citra

Pengukuran kualitas citra *GIF* yang telah disisipkan pesan dilakukan secara subjektif dan objektif. Pengukuran subjektif dilakukan dengan cara melihat secara visual perbedaan bentuk dan warna dari citra yang telah disisipkan dengan yang belum. Pengukuran secara objektif dilakukan dengan cara menghitung nilai PSNR (*Peak Signal to Noise Ratio*). Nilai PSNR dalam satuan desibel (dB) dihitung sesuai persamaan:

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

Dimana nilai MSE (*Mean Square Error*) didapat dari persamaan:

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

Persamaan MSE membutuhkan dua buah masukan citra, lalu dicari nilainya. Setelah itu dihitung nilai PSNR. Nilai PSNR yang wajar pada perbandingan dua buah berkas citra adalah di atas 30 dB.

3. HASIL DAN PEMBAHASAN


Berdasarkan hasil analisis, perangkat lunak yang memiliki fungsi untuk menyisipkan dan ekstraksi pesan telah berhasil dikembangkan. Proses selanjutnya adalah melakukan pengujian untuk memeriksa apakah perangkat lunak yang dikembangkan mempunyai kinerja yang baik atau tidak.

Perangkat lunak yang dikembangkan diberi nama *SteGIF*. Untuk menguji kebenaran perangkat lunak ini, dilakukan pengujian dengan melakukan penyisipan dan ekstraksi pesan pada citra *GIF*. Terdapat tiga buah kasus uji yang dilakukan, yaitu menguji kebenaran proses penyisipan dan ekstraksi, menguji kebenaran penggunaan kunci, dan menguji perubahan kualitas citra setelah disisipkan dengan yang belum disisipkan, menggunakan cara subjektif dan objektif. Sehingga setelah pengujian dapat disimpulkan tingkat kebenaran implementasi dari perangkat lunak yang dikembangkan.

3.1 Pengujian Kebenaran Proses Penyisipan dan Ekstraksi (Kasus Uji 1)

Pengujian ini dilakukan untuk menguji kebenaran proses penyisipan pesan dan ekstraksi pesan. Cara yang dilakukan adalah menyisipkan pesan dengan sebuah kunci, lalu mengekstraksinya dengan kunci yang sama. Pengujian akan berhasil apabila pesan berhasil disisipkan, dan dapat diekstraksi kembali dengan benar. Isi berkas penyisipan dapat dilihat pada Tabel 1, dan isi berkas hasil ekstraksi dapat dilihat pada Tabel 2.



Tabel 1 - Isi berkas penyisipan

Jenis berkas	Isi Berkas
Berkas yang disisipkan	

Jenis berkas	Isi Berkas
Media penyisipan	

Kunci yang digunakan berbentuk alfanumerik dengan nilai 'kunci'. Kunci ini dimasukkan pada proses penyisipan dan ekstraksi.

Tabel 2 - Isi berkas ekstraksi

Jenis berkas	Isi Berkas
Berkas yang disisipkan	
Media penyisipan	

Dari hasil pengujian, terbukti bahwa SteGIF 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. Namun waktu yang dibutuhkan pada beberapa tes cukup lama, dan sebanding dengan besarnya ukuran pesan. Hal ini dikarenakan pembacaan citra dan pesan dilakukan secara bit per bit, tidak secara langsung membaca berkas secara utuh.

3.2 Pengujian Proses Penggunaan Kunci (Kasus Uji 2)

Pengujian ini dilakukan untuk menguji kebenaran dari penggunaan kunci, yaitu dengan cara memasukkan kunci yang salah pada saat ekstraksi. Pengujian akan berhasil apabila pesan yang

diekstraksi memiliki isi yang tidak sama dengan pesan yang asli.

Dengan menggunakan berkas penyisipan yang sama dengan kasus uji 1, dilakukan proses ekstraksi menggunakan kunci yang salah. Pada kasus uji ini digunakan kunci alfanumerik bernilai 'stego'. Hasil pengujian dapat dilihat pada Tabel 3.

Dari hasil pengujian, terbukti bahwa SteGIF dapat melakukan aspek penggunaan kunci dengan baik. Proses ekstraksi dengan kunci yang salah dapat ditangani, yaitu dengan tetap menghasilkan pesan yang berbeda dengan pesan yang asli, namun karena menggunakan kunci yang berbeda, penghitungan semu acak untuk mendapatkan piksel mana yang akan diambil nilai paritasnya menghasilkan nilai yang berbeda, sehingga isi pesan yang terekstraksi juga berbeda.

Tabel 3 - Isi berkas ekstraksi menggunakan kunci yang berbeda

Jenis berkas	Isi Berkas
Berkas yang disisipkan	<pre> y_ ybp0-mWnũy>ũiyi-β°y_ yũyyŨ» Bæip̄yoŨm̄i~þ-βþyyip̄ou.ÿ_δiBũyãÿ ã¶øÿziŷ½iŷi+iußÿ□ÿ_ßóÿ□}ú×ú ũÿÿyũÿe}iÿδ_ÿÿ+ũs_+ -iũÿY+im̄pi úÿBÿú□ÿyũũÿ¶iÿÿiŨiũ~ÿ-βã8ũ÷ã_ þi@ÿ_δiB {ÿóÿ÷·Ũÿ½÷ũ½_þi÷}ú÷ÿ ÿ÷iB÷¾4ÿ_□=ßÿũÿÿyũÿe]ÿÿ;ÿÿÿÿÿ Ũ_ó[ÿ@i÷β÷ÿÿÿÿÿÿÿ_ßéÿÿ {úÿÿÿÿÿ □ßiũ~ÿÿßũ¾4ÿÿ~ÿÿi¾4oÿÿÿÿÿÿÿÿÿ Ũÿÿ_ũ½□ÿÿéß_ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ </pre>
Media penyisipan	

3.3 Pengujian Perubahan Kualitas Citra GIF

Pengujian ini dilakukan untuk menguji kualitas dari citra hasil penyisipan, yaitu dengan membandingkannya dengan citra yang asli. Perbandingan ini memakai masukan dari citra GIF 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:

1. Pada cara subjektif, citra dianggap mirip.
2. Pada cara objektif, hasil penghitungan PSNR di atas 30 dB.

Pada cara subjektif, semua citra hasil penyisipan dianggap mirip dengan citra asli. Sehingga SteGIF dapat menyisipkan pesan tanpa menimbulkan kecurigaan secara subjektif, atau secara kasat mata.

Pada cara objektif, dilakukan penghitungan nilai PSNR antara berkas yang telah disisipkan dengan yang belum, dan menghasilkan nilai PSNR: 54,640614262. Sehingga hal ini membuktikan bahwa proses penyisipan dengan menggunakan metode adaptif tidak mengubah kualitas struktur citra secara signifikan.

Dari semua hasil pengujian yang telah dilakukan, terlihat bahwa perangkat lunak SteGIF dapat menjalankan semua fungsi dan spesifikasi dengan benar. Penurunan kualitas citra yang dihasilkan juga tidak terlalu signifikan dan secara kasat mata tidak terlihat secara signifikan perbedaan antara citra *GIF* awal dengan yang sudah disisipi pesan. Penurunan kualitas citra yang dihasilkan juga bergantung pada besarnya ukuran pesan yang disisipkan.

4. KESIMPULAN

Berikut beberapa kesimpulan yang dapat diambil dari kegiatan analisis ini:

1. Telah berhasil dikembangkan perangkat lunak yang dapat melakukan steganografi pada citra *GIF*. Kebutuhan fungsional dari perangkat lunak, seperti proses penyisipan dan ekstraksi pesan, serta penggunaan kunci sudah dapat dilakukan dengan benar.
2. Metode adaptif sebagai metode penyisipan pesan sudah dapat dilakukan dengan benar, yaitu menyisipkan pesan berdasarkan nilai paritas dari nilai piksel yang terdapat pada citra *GIF*. Keberhasilan ini terletak pada proses pengambilan nilai paritas dan penentuan warna tetangga, sehingga perubahan yang dilakukan dengan mengganti warna dengan warna tetangga terdekat yang memiliki nilai paritas berbeda.
3. Kualitas citra *GIF* yang dihasilkan bergantung pada besarnya ukuran pesan. Berdasarkan pengamatan yang dilakukan saat pengujian, citra *GIF* yang disisipkan dengan ukuran pesan yang lebih besar akan mengalami perubahan kualitas yang lebih besar. Maksimum ukuran pesan dalam satuan *byte* yang dapat disisipkan

menggunakan metode adaptif yang telah diimplementasikan adalah sesuai persamaan berikut.

$$p = \frac{m \times n}{8}$$

Dimana,

p : ukuran pesan

m : ukuran panjang citra *GIF* dalam piksel

n : ukuran lebar citra *GIF* dalam piksel

DAFTAR REFERENSI

- [1] Johnson, Neil F. 1998. *Exploring Steganography: Seeing the Unseen*. IEEE.
- [2] *Graphics Interchange Format™*. 1989. Ohio: CompuServe Incorporated.
- [3] Fridrich, Jiri. 1998. *A New Steganographic Method for Palette-Based Images*. Center for Intelligent Systems, SUNY Binghamton.