MAKALAH SEMINAR TUGAS AKHIR

APLIKASI STEGANOGRAFI BERBASIS GUI DENGAN METODE PENGGANTIAN LSB

Didit Praditya*, Agung Budi Prasetijo**, R. Rizal Isnanto**

Abstrak **- Steganografi merupakan bagian dari teknik penyembunyian informasi. Berbeda dengan kriptografi, dimana pesan yang disandikan masih dapat dilihat, steganografi meniadakan keberadaan suatu pesan. Salah satu media yang digunakan untuk menyimpan pesan rahasia adalah citra komputer berformat raster yang merupakan kumpulan larik dari titik-titik yang disebut piksel. Pada citra *truecolor*, tiap-tiap piksel tersusun oleh tiga elemen warna, yaitu merah, hijau, dan biru atau elemen RGB (*red*, *green*, dan *blue*). Dengan mengganti satu atau beberapa bit LSB elemen RGB tersebut, dapat diperoleh ruang pada citra untuk menyimpan pesan rahasia.**

Pada Tugas Akhir ini dibuat aplikasi steganografi dengan metode penggantian LSB elemen-elemen RGB pada citra *truecolor*. Aplikasi dibuat dengan menggunakan *toolkit* Qt dan bahasa C++. Aplikasi berbasis GUI dan berjalan pada sistem operasi Linux.

Berdasarkan penggunaan aplikasi, didapatkan hasil bahwa lamanya waktu yang diperlukan untuk melakukan proses steganografi dengan metode penggantian LSB elemen-elemen RGB adalah berbanding lurus dengan besarnya ukuran berkas pesan rahasia. Perubahan piksel pada citra juga telah dapat diamati pada steganografi level 2, tetapi tidak pada citra hitam. Sedangkan untuk citra dengan warna piksel yang seragam, perubahan piksel lebih mudah diamati dibandingkan dengan citra dengan warna piksel yang tidak seragam. Perubahan piksel pada citra dengan warna piksel seragam juga lebih mudah diamati pada citra berwana putih dibandingkan dengan citra berwarna hitam.

Kata-kunci: steganografi, citra, citra *truecolor*, piksel, RGB, LSB

I. PENDAHULUAN

1.1 Latar Belakang

Steganografi merupakan salah satu cabang ilmu yang membahas tentang penyembunyian informasi di dalam informasi lainnya. Steganografi berasal dari bahasa Yunani yang berarti "tulisan yang tertutup atau tersembunyi". *Stegos*, yang artinya tertutup atau tersembunyi, dan *graphos*, yang artinya tulisan. Bersama-sama dengan kriptografi, steganografi merupakan bagian dari bidang ilmu yang membahas tentang metode-metode atau teknik-teknik tentang penyembunyian informasi (*information hiding*).

Berbeda dengan metode kriptografi, dimana pesan yang disandikan (*cipher*) masih dapat dilihat, metode steganografi menghilangkan keberadaan suatu pesan informasi sehingga pesan informasi tersebut menjadi suatu pesan yang tak terlihat (*invisible*). Pesan rahasia ini dapat dikirimkan oleh pengirim ke penerima melalui pembawa pesan (c*arrier*), tanpa diketahui orang lain bahwa terdapat pesan rahasia di dalam pembawa tersebut. Dalam steganografi berbasis komputer, media pembawa dapat berupa citra, audio, video, dan sebagainya.

1.2 Tujuan

Dalam Tugas Akhir ini akan dibuat suatu perangkat lunak steganografi yang mengimplementasikan suatu metode penyembunyian informasi melalui penyisipan bit-bit LSB pada RGB yang menggunakan citra digital komputer sebagai media penyembunyian pesan. Aplikasi yang dibuat berbasis GUI dan dibuat dengan menggunakan aplikasi Qt Open Source Edition sebagai *toolkit* untuk perancangan GUI dan bahasa C++ sebagai bahasa pemrogramannya.

1.3 Batasan Masalah

Berikut ini adalah batasan-batasan masalah pada Tugas Akhir yang dibuat.

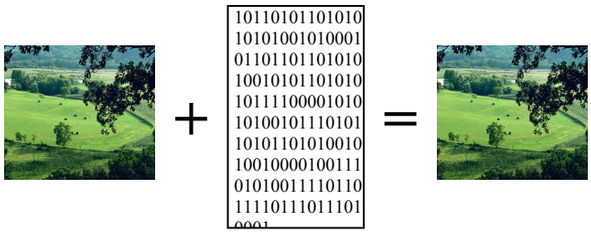
1. Aplikasi dibuat menggunakan bahasa pemrograman berorientasi objek C++ dan Qt Open Source Edition untuk perancangan GUI.
2. Citra yang dimasukkan merupakan citra *truecolor* yang didukung oleh Qt dengan kedalaman 24 bit atau 32 bit yang digunakan sebagai citra pembawa atau citra stego. Citra ini akan disimpan dalam format kompresi tak berugi dengan format PNG (*Portable Network Graphics*).
3. Penyembunyian pesan dilakukan dengan menggunakan metode penggantian atau penyisipan satu atau beberapa bit LSB pada ketiga elemen-elemen warna RGB.
4. Aplikasi berjalan pada Sistem Operasi Linux.

* Mahasiswa Jurusan Teknik Elektro Universitas Diponegoro
** Staf Pengajar Jurusan Teknik Elektro Universitas Diponegoro

II. LANDASAN TEORI

2.1 Definisi Steganografi

Dalam pengertian secara umum, steganografi merupakan berbagai metode tentang penyembunyian suatu pesan, menyembunyikan data di dalam data lainnya, metode untuk menulis pesan tersembunyi, dan lain-lain. Namun, seiring perkembangan zaman, steganografi juga digunakan dalam bidang komputer dan informasi. Terdapat beragam definisi tentang steganografi dalam istilah bidang komputer dan informasi. Salah satu pengertian steganografi yang sesuai dengan istilah dalam bidang komputer dan informasi dan sesuai dengan aplikasi yang dibuat dalam Tugas Akhir ini adalah menurut webopedia.com. Menurut webopedia.com[12], steganografi adalah seni dan ilmu pengetahuan tentang penyembunyian informasi dengan memasukkan pesan ke pesan lainnya, yang nampak tidak merugikan pesan tersebut. Steganografi bekerja dengan mengganti bit-bit data yang tidak berguna atau yang tidak terpakai dalam berkas-berkas komputer umumnya (seperti grafik, suara, teks, HTML, atau bahkan disket) dengan bit-bit yang berbeda, yaitu informasi yang tak terlihat. Informasi yang tersembunyi ini dapat berupa teks biasa (*plain text*), teks tersandi (*chiper text*), atau bahkan suatu citra.



Gambar 2.1 Steganografi dengan menggunakan citra digital sebagai pembawa (*carrier*).

2.2 Metode Steganografi

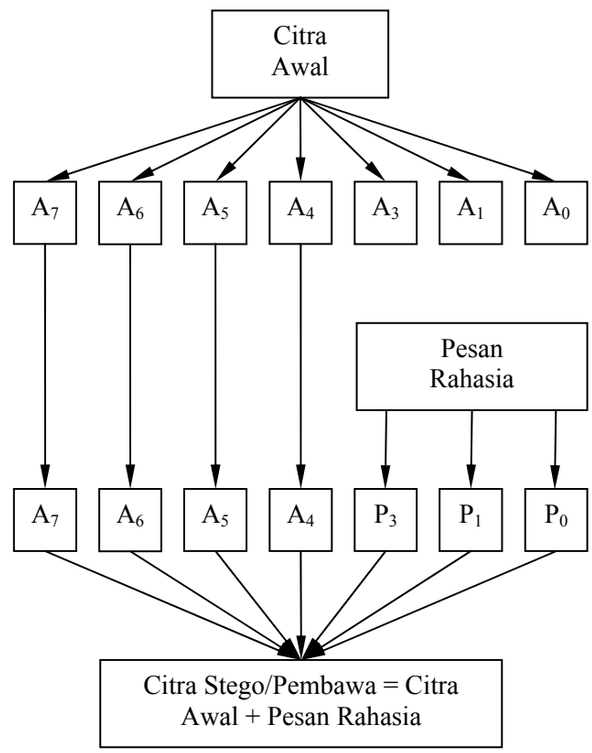
Dalam steganografi berbasis komputer, salah satu metode yang umum digunakan adalah menyembunyikan pesan atau informasi rahasia dengan menyisipkannya ke dalam berkas komputer yang berperan sebagai pembawa. Pesan rahasia yang disembunyikan terlebih dahulu dikonversi ke dalam suatu aliran bit informasi. Kemudian bit-bit informasi ini disisipkan ke dalam berkas pembawa. Metode ini memanfaatkan bit-bit yang kurang atau tidak diperlukan dari berkas (*file*) pembawa dan menggantinya dengan bit-bit dari pesan informasi.

Gambar 2.1 menunjukkan steganografi dengan pembawa citra digital. Pembawa pesan yang mengandung pesan rahasia dapat dikirimkan atau ditransmisikan oleh pengirim ke penerima, tanpa seorangpun yang mengetahui bahwa terdapat pesan rahasia di dalam pembawa tersebut.

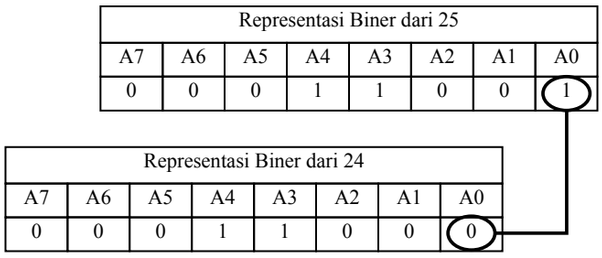
2.3 Steganografi Berbasis Citra

Citra digital terdiri atas larik titik-titik yang disebut piksel. Banyaknya piksel suatu citra berdasarkan pada panjang dan lebar atau dimensi citra tersebut. Citra dengan dimensi 800 x 600 akan mempunyai jumlah piksel sebanyak 480000 piksel. Pada citra *truecolor*, tiap-tiap piksel terdiri dari tiga buah elemen warna yang menyusun warna tertentu bagi piksel tersebut. Ketiga elemen warna tersebut merupakan tiga warna primer dan sering disebut triplet RGB atau RGB saja, yang terdiri dari warna merah (*red*), hijau (*green*), dan biru (*blue*). Pada citra dengan kedalaman 24 bit, masing-masing elemen RGB tersusun atas 8 bit (3 x 8 bit = 24 bit). Dengan demikian, satu elemen biru mempunyai variasi sebanyak 256 warna

Dalam metode penyisipan LSB seperti yang ditunjukkan oleh Gambar 2.2, steganografi dilakukan dengan cara mengganti satu atau beberapa bit-bit RGB piksel citra awal (A_0 – A_3) dengan bit-bit dari pesan rahasia (P_0 – P_3) yang akan disembunyikan. Berkas citra pembawa yang telah disisipi pesan rahasia, atau disebut juga berkas/citra stego, dapat dikirimkan ke penerima. Penerima dapat mengekstraksi pesan tersebut atau melakukan aksi desteganografi dari citra pembawa sehingga mendapatkan bit-bit pesan tersembunyi dari citra pembawa.



Gambar 2.2 Steganografi berbasis citra digital dengan metode penggantian atau penyisipan LSB.



Gambar 2.3 Penggantian LSB pada data 8 bit.

2.4 Metode Penggantian/Penyisipan LSB Pada RGB

Jika satu bit dari elemen biru triplet RGB dimodifikasi, maka perubahan warna yang terjadi pada piksel tersebut tidak akan dapat ditangkap oleh mata manusia biasa. Misalkan elemen biru mempunyai data 8 bit, yaitu $00011001_2 = 25_{10}$, jika mengganti 1 bit LSB dengan nilai 1 bit informasi, maka nilai tersebut dapat tetap atau berubah dengan perubahan yang kecil. Jika berubah, maka nilai data tersebut menjadi $00011000_2 = 24_{10}$, yang nilai perubahannya sangat kecil bagi warna suatu piksel. Gambar 2.3 menunjukkan hal tersebut. Dengan memanfaatkan satu atau beberapa bit dari elemen warna ini, bit-bit tersebut dapat digunakan sebagai tempat untuk menyimpan suatu pesan informasi.

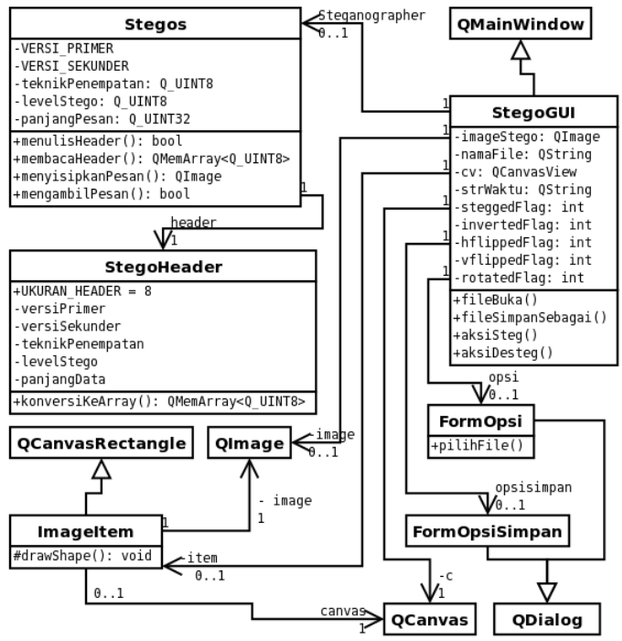
III. PERANCANGAN PERANGKAT LUNAK

3.1 Diagram Kelas

Diagram kelas ini merupakan diagram kelas yang dipandang dari sisi perangkat lunak. Diagram kelas mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. Diagram kelas juga menunjukkan properti dan operasi sebuah kelas dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut[6]. Diagram kelas yang menggambarkan struktur antar kelas pada aplikasi ditunjukkan pada Gambar 3.1.

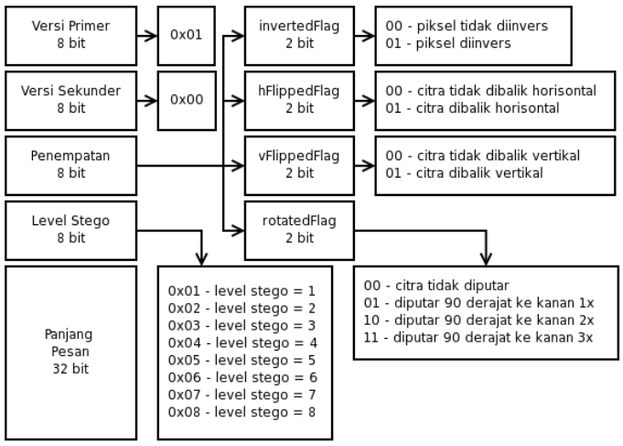
3.2 Header

Header yang disisipkan ke dalam citra dalam proses steganografi oleh aplikasi akan digunakan sebagai informasi bagaimana berkas atau pesan rahasia disimpan ke dalam citra. Header ini juga digunakan untuk proses desteganografi atau pengambilan pesan.

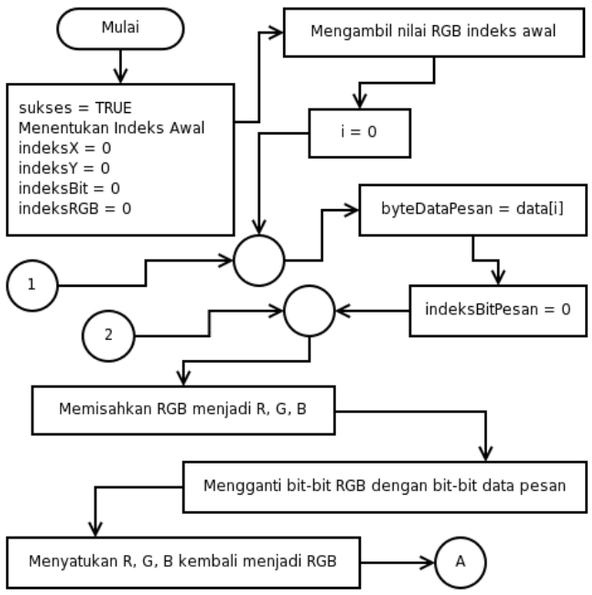


Gambar 3.1 Diagram kelas.

Header yang digunakan terdiri atas 64 bit dan terbagi menjadi 5 bagian data yang bertipe `Q_UINT8` (tipe data *unsigned integer* 8 bit pada Qt) dan `Q_UINT32` (tipe data *unsigned integer* 32 bit pada Qt), yaitu: informasi tentang versi primer (8 bit), versi sekunder (8 bit), penempatan (8 bit), level stego (8 bit), dan panjang berkas/pesan (32 bit). Informasi tentang versi (primer dan sekunder) digunakan untuk memastikan bahwa citra stego yang akan diambil pesan yang terkandung di dalam citra tersebut adalah citra stego yang juga disispi pesan oleh aplikasi dengan versi yang sama. Informasi penempatan menyimpan hasil dari pemodifikasian atau pengeditan citra sebelum disisipi pesan yang berupa pemanji (*flags*) (yaitu: `invertedFlag`, `hFlippedFlag`, `vFlippedFlag`, dan `rotatedFlag`). Data level stego menyimpan informasi tentang level stego yang digunakan dalam proses penyisipan pesan (steganografi), yang berkisar antara 1 sampai 8. Bagian header panjang berkas/pesan menyimpan informasi tentang besarnya panjang pesan yang disisipkan ke dalam citra stego. Format header ditunjukkan pada Gambar 3.2.



Gambar 3.2 Format header yang digunakan pada aplikasi.



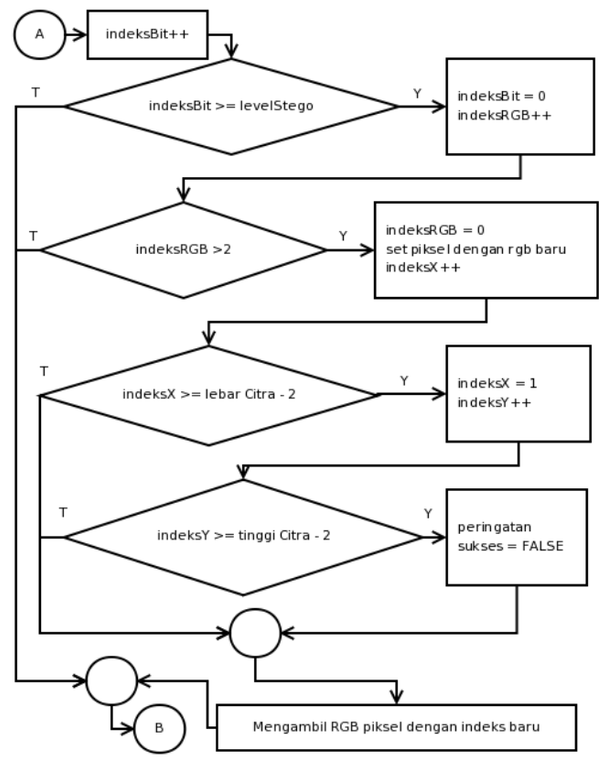
Gambar 3.3 Diagram alir operasi `menulisData` pada kelas `Stegos` bagian pertama.

3.3 Aksi Steganografi

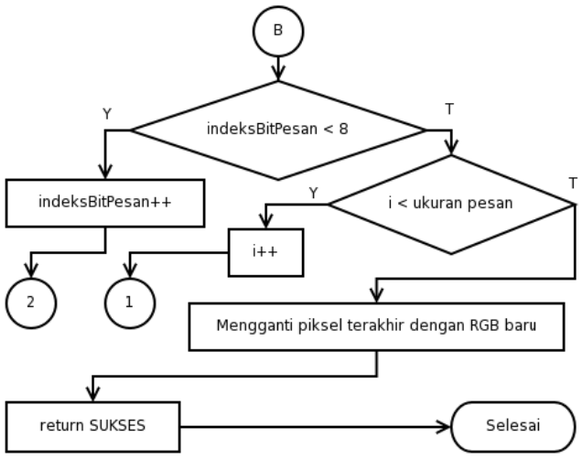
Operasi penyisipan aliran pesan yang telah dikonversi menjadi aliran bit ke dalam citra atau operasi penulisan data ke dalam citra dilakukan oleh kelas Stegos. Diagram alir dari operasi menulisData pada kelas Stegos ditunjukkan oleh Gambar 3.3, 3.4 dan 3.5.

Pada bagian pertama diagram alir yang ditunjukkan oleh Gambar 3.3, aksi dimulai dengan terlebih dahulu menentukan indeks-indeks awal, kemudian nilai piksel awal (nilai RGB) citra diambil. Data pesan rahasia yang merupakan suatu larik kemudian diberikan ke `byteDataPesan`. Kemudian, untuk sebanyak ukuran larik byte pesan rahasia, RGB dipisahkan menjadi R, G, dan B. Lalu, bit LSB pada R, G, dan B, diganti dengan bit-bit dari data pesan. Setelah diganti, R, G, dan B disatukan kembali menjadi RGB.

Selanjutnya, seperti yang ditunjukkan pada Gambar 3.4, `indeksBit` diperiksa, jika lebih besar atau sama dengan level stego, maka `indeksBit` direset dan `indeksRGB` dinaikkan satu. Kemudian `indeksRGB` diperiksa, jika `indeksRGB` lebih besar dari dua, maka `indeksRGB` direset, nilai RGB piksel sebelumnya diganti dengan nilai RGB yang baru, dan `indeksX` dinaikkan untuk piksel selanjutnya. Ini dilakukan sampai semua bit LSB piksel disisipi bit pesan. Jika `indeksX` telah mencapai lebar citra, maka pemrosesan piksel dilanjutkan pada baris berikutnya pada citra dengan menaikkan `indeksY` dan mengembalikan `indeksX` ke 1. Jika pemrosesan mencapai panjang citra (`indeksY` = panjang citra), maka terjadi kesalahan, karena sebelumnya program telah mengambil nilai `indeksX` yang sudah maksimal (`indeksX` = lebar citra) dan program telah menghitung ukuran pesan sehingga jika ukuran pesan terlalu besar maka proses penyisipan tidak akan dilakukan. Jika penggantian telah dilakukan, maka program mengambil piksel dengan indeks selanjutnya.



Gambar 3.4 Diagram alir operasi `menulisData` pada kelas `Stegos` bagian kedua.



Gambar 3.5 Diagram alir operasi `menulisData` pada kelas `Stegos` bagian ketiga.

Kemudian dalam diagram alir pada Gambar 3.5, program memeriksa `indeksBitPesan`, jika bernilai kurang dari 8 (banyaknya bit tiap bagian pesan dalam satu larik), berarti belum semua data larik pesan dimasukkan ke dalam LSB, maka program akan menaikkan `indeksBitPesan` dan kembali memasukkan data larik pesan ke dalam piksel. Jika `indeksBitPesan` sudah bernilai 8, maka program memeriksa indeks larik pesan atau ukuran pesan. Jika ukuran pesan belum maksimal, maka larik pesan selanjutnya akan diproses, sedangkan jika ukuran pesan sudah maksimal maka program akan mengganti piksel terakhir yang dengan nilai RGB yang baru. Sebelum operasi selesai, program memberikan nilai kembali TRUE setelah menyisipkan pesan jika tidak ada kesalahan yang terjadi.

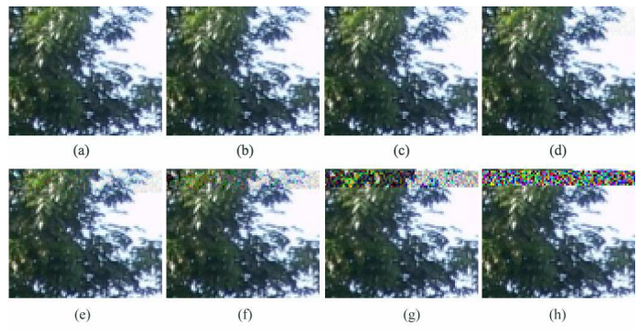
IV. PENGUJIAN PERANGKAT LUNAK

4.1 Uji Skenario Dalam Menyisipkan Pesan

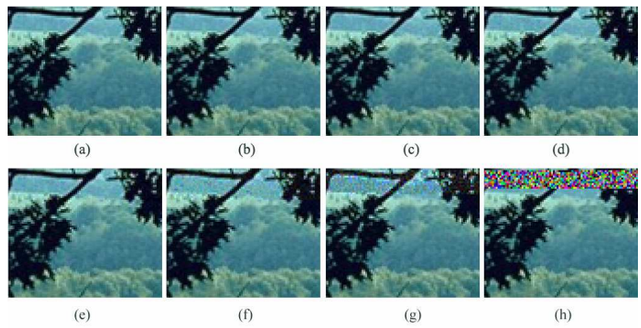
Setelah menjalankan aplikasi, aksi menyisipkan pesan dapat dilakukan. Aplikasi steganografi yang ditampilkan merupakan objek dari kelas `StegoGUI` yang berasal dari kelas `QMainWindow`. Gambar 4.1 menunjukkan tampilan menu dari objek GUI aplikasi.

Gambar 4.1 Tampilan menu GUI aplikasi.

Hasil-hasil pengujian yang berupa perbandingan level stego citra stego 1, citra stego 2, citra hitam, dan citra putih dengan level stego 1 sampai dengan level stego 8 ditunjukkan pada Gambar 4.2, Gambar 4.3, Gambar 4.4, dan Gambar 4.5.

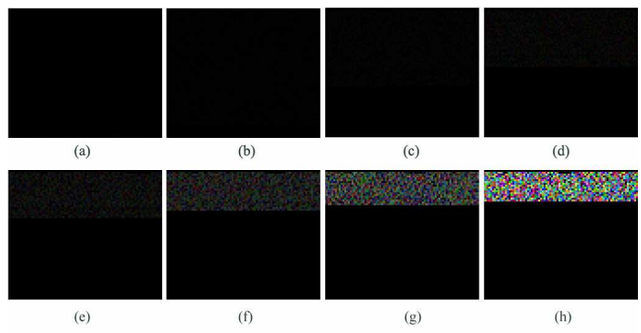


Gambar 4.2 Perbandingan citra stego 1. (a) level 1 (b) level 2 (c) level 3 (d) level 4 (e) level 5 (f) level 6 (g) level 7 (h) level 8

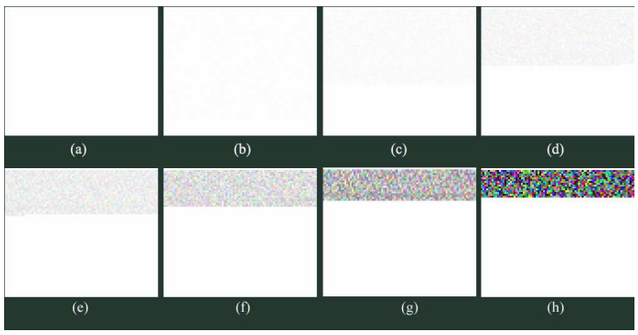


Gambar 4.3 Perbandingan citra stego 2. (a) level 1 (b) level 2 (c) level 3 (d) level 4 (e) level 5 (f) level 6 (g) level 7 (h) level 8

Berdasarkan hasil-hasil pengujian tersebut, didapatkan hasil sebagai berikut. Pengamatan yang dilakukan pada layar monitor perangkat pengujian menunjukkan perubahan piksel pada citra putih untuk level stego 2 dan semakin jelas untuk level stego yang lebih besar. Pada citra hitam perubahan piksel dapat diamati pada level stego 5 dan semakin jelas untuk level stego yang lebih besar. Sedangkan pada citra 1 perubahan piksel dapat diamati pada level stego 2 dan pada citra 2 juga pada level stego 2.



Gambar 4.4 Perbandingan citra hitam. (a) level 1 (b) level 2 (c) level 3 (d) level 4 (e) level 5 (f) level 6 (g) level 7 (h) level 8



Gambar 4.5 Perbandingan citra putih. (a) level 1 (b) level 2 (c) level 3 (d) level 4 (e) level 5 (f) level 6 (g) level 7 (h) level 8

Berdasarkan pengamatan juga dapat diketahui bahwa citra dengan warna piksel yang seragam (*uniform*) yaitu pada citra putih dan citra hitam, perubahan piksel lebih mudah dilihat daripada citra 1 atau citra 2. Perubahan piksel juga lebih mudah diamati pada citra putih yang lebih terang daripada citra hitam yang lebih gelap.

Pada pengujian dalam menyisipkan pesan juga didapatkan hasil berupa lamanya waktu yang diperlukan untuk proses penyisipan. Waktu yang dihasilkan adalah berdasarkan dari perangkat keras dengan spesifikasi: prosesor AMD Sempron™ 2200+ dengan kecepatan 1,50 GHz, dan mempunyai RAM 512 MB yang menjalankan sistem operasi Linux Mandriva 2005. Hubungan antara ukuran pesan dan waktu yang diperlukan untuk melakukan proses steganografi ditunjukkan pada Tabel 4.1. Sedangkan, hubungan level stego dan waktu ditunjukkan pada Tabel 4.2.

Berdasarkan Tabel 4.1, dapat diketahui bahwa semakin besar ukuran pesan, maka semakin lama waktu yang diperlukan. Sedangkan berdasarkan Tabel 4.2, dapat diketahui bahwa untuk level stego yang berbeda, waktu yang diperlukan untuk melakukan steganografi tidak menunjukkan perbedaan waktu yang sangat besar.

TABEL 4.1 HUBUNGAN ANTARA UKURAN PESAN DAN WAKTU YANG DIPERLUKAN UNTUK MELAKUKAN PROSES STEGANOGRAFI.

Ukuran Pesan (Byte)	Level Stego	Waktu (ms)
509	1	794
1024	1	1834
9869	1	16798
18349	1	29906
37455	1	57547
136454	1	218941
286927	1	450084

TABEL 4.2 HUBUNGAN ANTARA LEVEL STEGO DAN WAKTU YANG DIPERLUKAN UNTUK MELAKUKAN PROSES STEGANOGRAFI.

Nama Berkas	Ukuran Pesan (Byte)	Level Stego	Waktu (ms)
GPL.TXT	18349	1	27769
		2	21829
		3	21785
		4	29988
		5	22853
		6	21300
		7	21945
		8	22366
	18349	1	26955
		2	24741
		3	25866
		4	24328
		5	23057
		6	22775
		7	22778
		8	26063

V. PENUTUP

5.1 Kesimpulan

1. Lamanya waktu yang diperlukan untuk proses steganografi dengan metode penggantian LSB pada RGB berbanding lurus dengan besarnya ukuran berkas pesan yang disisipkan.
2. Berdasarkan pengamatan, perubahan piksel pada citra sudah dapat diamati untuk steganografi level 2, yaitu menggunakan 2 bit pada LSB tiap-tiap elemen RGB untuk menyimpan pesan rahasia. Namun hal ini tidak ditemukan pada citra hitam
3. Berdasarkan pengamatan, citra dengan warna piksel yang seragam, perubahan piksel lebih mudah diamati daripada citra dengan warna piksel yang berbeda-beda. Pada citra dengan warna piksel yang seragam, perubahan piksel juga lebih mudah diamati pada citra putih daripada citra hitam.

5.2 Saran

1. Terdapat metode-metode steganografi lain, selain metode penggantian LSB RGB, maka dari itu aplikasi-aplikasi steganografi dengan metode-metode yang lain juga dapat dikembangkan.
2. Dapat dikembangkan juga aplikasi steganografi dengan menggunakan media lain selain citra sebagai *carrier* seperti: audio, video, *datastream*, dan sebagainya
3. Metode steganografi dapat digabungkan dengan metode-metode kriptografi. Sehingga selain didapatkan pesan rahasia yang tersembunyi, juga pesan rahasia yang tersandi.

4. Steganografi dengan metode penggantian LSB RGB dapat diimplementasikan oleh bahasa pemrograman lain yang ber-*platform* independen. Sehingga dapat diaplikasikan pada telepon genggam, dan sebagainya.

DAFTAR PUSTAKA

[1] Bailey, K., K. Curran, and J. Condell, *An Evaluation of Automated Stegodetection Methods In Images*, http://www.irishscientist.ie/Steganography.htm, Oktober 2005

[2] Baldwin, R. G., *Steganography 101 using Java*, http://www.developer.com/java/other/article.php/10936_3530866_1, Oktober 2005.

[3] Braun D., J. Sivils, A. Shapiro, and J. Versteegh, *Unified Modeling Language (UML) Tutorial*, http://pigseye.kennesaw.edu/~dbraun/csis4650/A&D/UML_tutorial/index.htm, Kennesaw State University, Spring 2001, Oktober 2005.

[4] Cohoon, J. P. and J. W. Davidson, *C++ Program Design: An Introduction to Programming and Object-Oriented Design*, 3rd ed, McGraw-Hill, NY, 2002.

[5] Corbett, S., *StegPng*, http://www.scorbett.ca - StegPng.htm, Oktober 2005.

[6] Eckel, B., *Thinking in C++,* 2nd ed. Volume 1, Prentice Hall, New Jersey, 2000.

[7] Fowler, M., *UML DISTILLED, 3th Ed*., *Panduan Singkat Bahasa Pemodelan Objek Standar*, Ed. I., Penerbit ANDI, Yogyakarta, 2005.

[8] Johnson, N. F., *Steganography*, www.jjtc.com/stegdoc/stegdoc.html, George Mason University Information System and Software Engineering, 1995-2003.

[9] Lavigne, D, *Hiding Secrets with Steganography*, http://www.onlamp.com/Lavigne/ONLamp_com Hiding Secrets with Steganography.htm, ONLamp.com, Oktober 2005.

[10] Lea, D., *From Design to Implementation*, http://g.oswego.edu/dl/oosdw3/ch25.html, Agustus 2006.

[11] Roelofs, G., *PNG: The Definitive Guide*, Second Edition (HTML Version), Published by Greg Roelofs, roelofs@pobox.com, 2002-2003.

[12] ---, *Qt Reference Documentation (Open Source Edition)*, Trolltech, 2004.

[13] ---, *Steganography*, http://en.wikipedia.org/wiki/Steganography.htm, Wikipedia, April 2006.

[14] ---, *Steganography*, http://www.webopedia.com/TERM/S/What is steganography - A Word Definition From the Webopedia Computer Dictionary.htm, Webopedia, Oktober 2005.

DIDIT PRADITYA
(L2F305300)
Mahasiswa Jurusan Teknik Elektro, Fakultas Teknik Universitas Diponegoro Semarang, dengan Konsentrasi Informatika dan Komputer.

Menyetujui dan Mengesahkan

Pembimbing II

R. Rizal Isnanto, S.T., M.M., M.T.
NIP. 132 288 515
Tanggal