IMAGE STEGANOGRAPHY DENGAN METODE *LEAST SIGNIFICANT BIT* (LSB)

M. Miftakul Amin

Jurusan Teknik Komputer Politeknik Negeri Sriwijaya Palembang
Jalan Srijaya Negara, Palembang 30139
Telp. 0711 – 353414 Fax. 0711 – 355918
website : http://polsri.ac.id
e-mail : mmiftakulamin@gmail.com

ABSTRACT

Security in delivering a secret message is an important factor in the spread of information in cyberspace. Protecting that message to be delivered to the party entitled to, should be made a message concealment mechanism. The purpose of this study was to hide a secret text message into digital images in true color 24 bit RGB format. The method used to insert a secret message using the LSB (Least Significant Bit) by replacing the last bit or 8th bit in each RGB color component. RGB image file types option considering that messages can be inserted capacity greater than if use a grayscale image, this is because in one pixel can be inserted 3 bits message. Tests provide results that are hidden messages into a digital image does not reduce significantly the quality of the digital image, and the message has been hidden can be extracted again, so that messages can be delivered to the recipient safely.

Keywords : LSB, RGB

ABSTRAK

Keamanan dalam menyampaikan pesan rahasia merupakan faktor penting dalam penyebaran informasi di dunia maya. Melindungi supaya pesan yang akan dikirimkan sampai kepada pihak yang berhak, perlu dibuat sebuah mekanisme penyembunyian pesan. Tujuan dari penelitian ini adalah menyembunyikan pesan berupa teks rahasia ke dalam citra digital true colour 24 bit dalam format RGB. Metode yang digunakan untuk menyisipkan pesan rahasia menggunakan metode LSB (Least Significant Bit) dengan mengganti bit terakhir atau bit ke-8 dalam setiap komponen warna RGB. Pilihan jenis file citra RGB dengan pertimbangan kapasitas pesan yang dapat disisipkan lebih besar dibandingkan jika menggunakan citra grayscale, hal ini dikarenakan dalam 1 pixel dapat disisipkan 3 buah bit pesan. Ujicoba yang dilakukan memberikan hasil bahwa pesan yang disembunyikan ke dalam citra digital tidak mengurangi kualitas citra digital secara signifikan, dan pesan yang telah disembunyikan dapat diekstrak kembali, sehingga pesan yang dikirimkan dapat sampai dengan aman kepada penerima.

KataKunci : LSB, RGB

PENDAHULUAN

Munculnya teknologi internet dan multimedia telah mendorong berbagai macam usaha untuk melindungi, mengamankan dan menyembunyikan data pada file digital dari pihak-pihak yang tidak mempunyai otoritas mengakses file-file tersebut. Salah satu usaha untuk mengamankan data diantaranya dengan menggunakan kriptografi. Berbagai macam algoritma kriptografi dapat diimplementasikan untuk mewujudkan sistem keamanan data. Selain kriptografi juga terdapat steganography sebagai alternatif untuk mengamankan data.

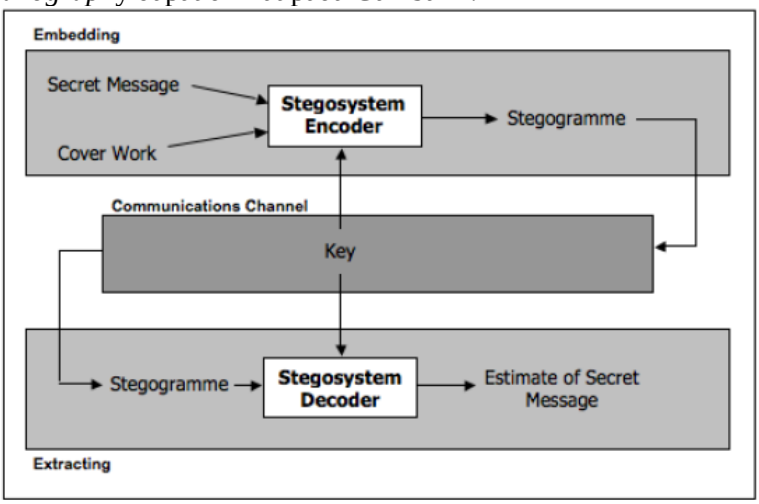
Perkembangan dunia komputer dan pendukung perangkat lainnya yang serba digital, telah membuat data-data digital semakin banyak digunakan. Terdapat sejumlah faktor yang membuat data digital (seperti audio, video, citra dan teks) semakin banyak digunakan [6], antara lain:

1. Mudah diduplikasi dan hasilnya sama dengan aslinya,
2. Mudah untuk penduplikasian dan penyimpanan,
3. Mudah disimpan untuk kemudian diolah atau diproses lebih lanjut,
4. Serta mudah didistribusikan, baik dengan media disk maupun melalui jaringan internet

Dengan tersedianya jaringan internet, memungkinkan untuk melakukan proses pertukaran data dan informasi. Dalam tukar menukar informasi, aspek keamanan memegang peranan penting, terutama jika informasi tersebut bersifat rahasia. Untuk menjaga kerahasiaan informasi dapat digunakan teknik *steganography*. Informasi yang akan dikirim disembunyikan dalam file digital (teks, image, audio, video). Kemudian data digital tersebut dikirim seperti data biasa, sehingga pihak ketiga tidak curiga bahwa di dalamnya terdapat informasi rahasia. Informasi yang disembunyikan dalam data digital tersebut dapat diekstrak kembali oleh penerima pesan. Informasi tersebut juga harus sama dengan informasi sebelum disisipkan dalam data digital, meskipun data digital tersebut telah mengalami proses manipulasi, seperti pengeditan, pemotongan atau kompresi.

Steganography merupakan sebuah ilmu dan seni untuk menuliskan pesan tersembunyi sedemikian rupa sehingga pihak selain yang berhak menerima tidak mengetahui keberadaan pesan tersebut [1]. *Steganography* berbeda dengan kriptografi, dimana kriptografi keberadaan pesannya jelas tetapi maknanya dikaburkan.

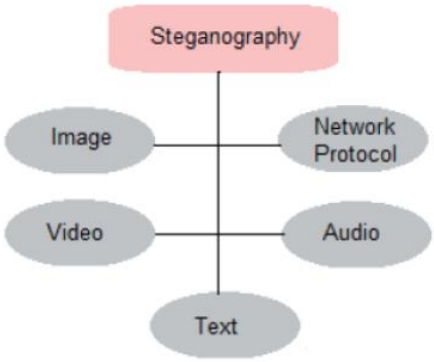
Dalam pengembangan *steganography* terdapat dua algoritma penting yaitu untuk melakukan *embedding* dan satu lagi untuk melakukan *extracting* [2]. Proses *embedding* merupakan proses untuk menyisipkan pesan rahasia (*secret message*) ke dalam *cover work* yang berupa file *image*, *video*, *audio* maupun teks sebagai media untuk menyisipkan pesan. Output dari proses *embedding* disebut sebagai *Stegograme* yang berisi *cover work* dan pesan tersembunyi. Sedangkan *extracting* adalah proses untuk memunculkan kembali pesan yang tersembunyi dari *cover work*. Keseluruhan proses dalam *steganography* dapat dilihat pada Gambar 1.



Gambar 1. Skema Proses *Steganography*

Dalam Gambar 1 dapat dilihat bahwa dalam proses *embedding* diperlukan 2 buah masukan berupa *secret message* biasanya berisi file/teks yang akan disembunyikan, dan *cover work* berupa media yang akan digunakan untuk menyisipkan *secret message*. Langkah selanjutnya adalah melewatkan 2 masukan tadi ke dalam *stegosystem encoder* yang akan melakukan proses penyisipan pesan ke dalam *buffer*/salinan dari *cover work*. Proses dalam *stegosystem encoder* biasanya melakukan upaya meminimalisasi distorsi dari *cover work*, dimana semakin rendah distorsinya semakin baik hasil outputnya untuk tidak terdeteksi. Dalam proses *stegosystem encoder* diperlukan kunci untuk melakukan operasi penyisipan pesan dan juga diperlukan pada fase *extracting*. Kunci ini merupakan sebuah ukuran keamanan yang dirancang untuk melindungi pesan rahasia. *Stegogramme* sebagai output dari proses encoder selanjutnya dikirim kepada penerima pesan melaui jalur komunikasi.

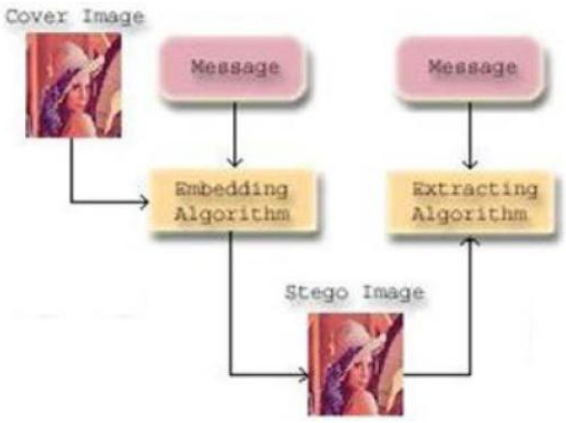
Berdasarkan jenis *cover work* yang digunakan, ada beberapa jenis steganography yang diterapkan dalam rangka menciptakan sebuah sistem keamanan. *Cover work* yang digunakan berupa sebuah citra digital (*image*) dikenal dengan istilah *image steganography* yang menggunakan intensitas pixel untuk menyembunyikan pesan rahasia. Jika yang digunakan sebagai cover work adalah protokol jaringan, seperti TCP, UDP, ICMP, IP dan jenis protokol lainnya maka dikenal dengan istilah *network protocol steganography*. *Video steganography* adalah teknik menyembunyikan pesan rahasia dalam media digital berupa video. Pada dasarnya *video* merupakan kumpulan dari sekian banyak *image* yang digunakan untuk menyisipkan pesan rahasia. *Video steganography* biasanya menggunakan beberapa jenis format *video* seperti AVI, MPEG, MP4 dan jenis format *video* lainnya. *Audio steganography* menggunakan media digital berupa file *audio*, jenis file *audio* yang digunakan adalah WAVE, MIDI, WAVE dan jenis format audio lainnya. Sedangkan jenis steganography yang paling sederhana menggunakan file teks, dengan menyisipkan beberapa pesan rahasis ke dalam sekumpulan teks, seperti dengan teknik menyisipkan setiap karakter di awal kata, di akhir kata atau kombinasi penyisipan tertentu di dalam teks. Beragam jenis *steganography* seperti diperlihatkan pada Gambar 2 [8].



Gambar 2. Media Digital yang Digunakan dalam Steganography

Dalam terminology *image steganography* dikenal istilah penting untuk mengembangkan *image steganography* seperti diperlihatkan pada Gambar 3, yaitu [8]:

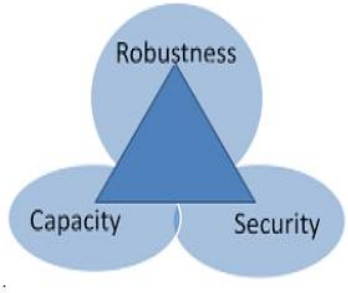
- *Cover image*, merupakan media yang digunakan untuk menyembunyikan pesan rahasia.
- *Embedded message*, merupakan pesan rahasia yang ingin disembunyikan dapat berupa teks maupun image.
- *Stego image*, cover image yang sudah berisi embedded message.
- *Stego key*, kunci berupa sebuah algoritma yang digunakan untuk melakukan penyisipan dan ekstraksi pesan rahasia dari stego image.



Gambar 3. Property Image Steganography

Beberapa *property steganography* yang perlu diperhatikan [4] yang biasa dikenal dengan segitiga *steganography*. Satu *property* dengan *property* yang lain saling bergantung, seperti dapat dilihat pada Gambar 4. ***Robustness*** dapat diartikan sebagai kemampuan pesan yang disisipkan

supaya tetap utuh terjaga, jika *stego image* mengalami perubahan karena proses edit. ***Security*** mengacu pada proses untuk melindungi supaya penyadap/orang yang tidak berhak, tidak mampu untuk mendeteksi informasi yang tersembunyi. ***Capacity*** bertujuan supaya ukuran informasi yang dapat disembunyikan relatif terhadap cover image tidak menyebabkan kualitasnya berkurang.



Gambar 4. Segitiga Steganography

Secara umum algoritma *steganography* dikelompokkan berdasarkan 2 pendekatan [3], yaitu:

1. *Spatial domain based steganography*
   Algoritma yang termasuk dalam kategori ini adalah LSB (*Least Significant Bit*) sebagai pendekatan sederhana untuk menyisipkan informasi berupa bit ke dalam bit terakhir yang terdiri dari 8 bit (1 byte) yang ada dalam *cover image*. Sebagai contoh

   Pixel:
   (10101111 11101001 10101000)
   (10100111 01011000 11101001)
   (11011000 10000111 01011001)

   Secret message:
   01000001

   Result:
   (10101110 11101001 10101000)
   (10100110 01011000 11101000)
   (11011000 10000111 01011001)

Algoritma untuk menyisipkan pesan rahasia berupa teks menggunakan LSB dapat dilakukan dengan langkah-langkah sebagai berikut:

Step 1 : baca *cover image* dan pesan rahasia yang akan disisipkan dalam *cover image*.
Step 2 : konversi *cover image* setiap pixelnya dari representasi desimal menjadi biner
Step 3 : konversi pesan rahasia menjadi biner, jika pesan berupa file teks maka setiap karakter dapat diketahui kode ASCII nya berupa bilangan desimal untuk kemudian dikonversi menjadi biner
Step 4 : hitung LSB dari setiap pixel cover image
Step 5 : ganti LSB dari cover image dengan setiap bit dari pesan rahasia satu per satu
Step 6 : tulis stego image yang merupakan hasil akhir dari proses penyisipan, biasanya dengan cara menyimpan ulang file gambar yang telah disisipi pesan rahasia

Sedangkan algoritma untuk membaca pesan rahasia langkah-langkahnya sebagai berikut:

Step 1 : baca stego image
Step 2 : hitung LSB dari setiap pixel dalam stego image
Step 3 : ambil bit-bit yang diperoleh pada step 2 untuk selanjutnya dikonversi menjadi karakter

2. *Transform domain based steganography*
   Algoritma jenis ini beroperasi dengan cara mentransformasikan *image* ke domain frekuensi. Contoh algoritma pendekatan transform domain diantaranya *Discrete Cosine Transform*, *KL Transform* dan *Wavelet Transform*.

Microsoft Visual Basic merupakan bahasa pemrograman tingkat tinggi (*High Level Languange*). Microsoft Visual Basic juga merupakan bahasa pemrograman *Object Oriented Programming* (OOP), yaitu pemrograman berorientasi pada objek. Microsoft Visual Basic memiliki beberapa versi yaitu Microsoft Visual Basic 3.0, Microsoft Visual Basic 5.0, Microsoft Visual Basic Versi 6.0, VB. Net. Dan mungkin akan berkembang lagi dengan berbagai versi dan semakin sempurna dalam penggunaannya.

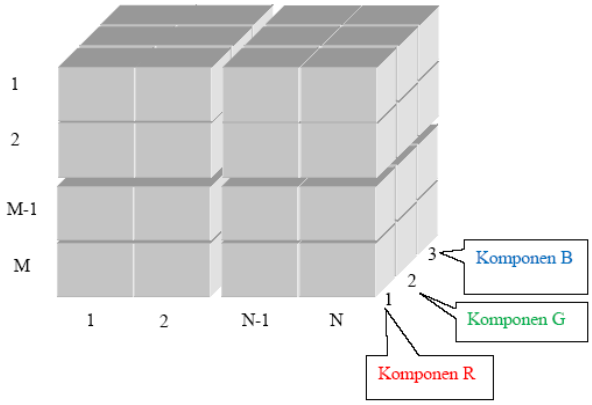
Menurut Kusrini Visual Basic adalah salah satu bahasa pemrograman komputer. Bahasa pemrograman adalah perintah-perintah yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. Visual Basic merupakan salah satu *development tool*, yaitu alat bantu untuk membuat berbagai macam program komputer, khususnya yang menggunakan sistem operasi windows [9]. Sedangkan menurut Suhata Visual Basic 6.0 merupakan salah satu bahasa pemrograman yang dapat digunakan untuk menyusun dan membuat program aplikasi pada lingkungan sistem operasi *windows*. Program aplikasi dapat berupa program database, program grafis, dan lain sebagainya. Di dalam visual basic 6.0 sudah terdapat kompenen-kompenen yang sangat membantu pembuatan program aplikasi [10].

Penelitian terkait topik *steganography* dan metode LSB pernah dilakukan oleh beberapa peneliti, diantaranya Dewi [5] yang mengembangkan perangkat lunak steganografi pada file AVI yang diberi nama AVISteg. Metode yang dikembangkan dalam penelitian ini adalah *LSB Modification*. AVISteg diimplementasikan dalam bahasa pemrograman pascal dengan kompilator Borland Delphi 7 dan beroperasi pada lingkungan sistem operasi *windows*. AVISteg ini berhasil menyisipkan data ke dalam kumpulan file BMP, tetapi tidak berhasil mengubah kembali kumpulan file BMP tersebut ke dalam file AVI. Penelitian serupa juga dilakukan oleh Wijaya [6] melakukan penelitian tentang *hidden message* menggunakan *steganography*, metode yang digunakan adalah *Dynamic Cell Spreading* yang merupakan teknik menyembunyikan dan menyisipkan data dengan bantuan *buffer memory* sebagai media penggabungan. Krisnawati [7] melakukan penelitian dengan menggunakan metode LSB dan End of File (EOF) untuk menyisipkan pesan ke dalam citra digital Grayscale.

Penelitian ini menyajikan sebuah mekanisme bagaimana proses penerapan metode LSB pada citra digital dan bagaimana proses penyisipan dan pengambilan data pada *cover image* sebagai *cover work* dapat diimplementasikan dalam perangkat lunak Visual Basic 6.0.

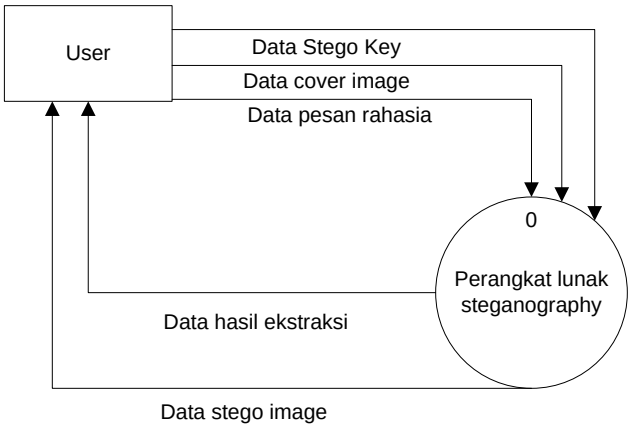
METODE PENELITIAN

Metode *Least Significant Bit* (LSB) adalah teknik penyembunyian pesan dengan cara menyisipkan pesan pada bit rendah atau bit paling kanan pada file *cover work* sebagai media untuk menyembunyikan pesan. Pada uji coba ini digunakan media citra digital *true colour* 24 bit dengan model warna RGB. Pada citra digital nantinya terdapat 3 bit yang dapat disisipi dalam 1 pixel. Hal ini dikarenakan dalam 1 pixel warna tersusun dari 3 komponen warna, yaitu Red, Green, dan Blue yang masing-masing disusun oleh 8 digit bilangan biner dari rentang nilai 0 sampai dengan 255 dalam desimal atau 00000000 sampai 11111111 dalam representasi biner. Representasi pixel citra digital 24 bit dengan model warna RGB dapat dilihat pada Gambar 5.



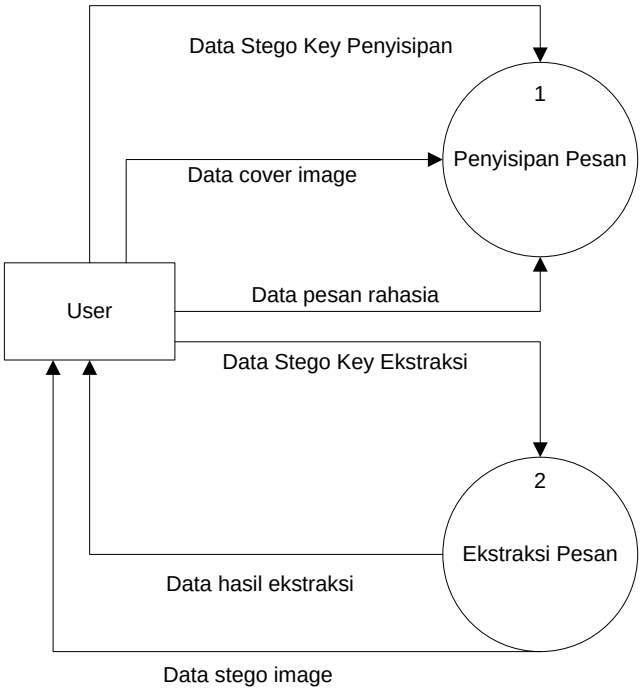
Gambar 5. Model Citra Warna RGB

Model fungsional perangkat lunak memberikan gambaran umum mengenai proses-proses yang terjadi dalam perangkat lunak tanpa memberikan detail mengenai bagaimana proses-proses tersebut diimplementasikan. Aliran informasi, dekripsi proses dan deskripsi data yang termasuk dalam kebutuhan fungsional digambarkan dalam diagram *context* pada Gambar 6 yang memperlihatkan interaksi antara user sebagai pengguna dan perangkat lunak *image steganography*. Entitas user memberi masukan berupa *cover image* yang akan disisipkan pesan, memasukkan pesan rahasia dan kunci yang digunakan untuk menyisipkan pesan rahasia ke dalam *cover image*. Perangkat lunak memberikan umpan balik kepada user berupa data image yang telah disisipi pesan dan hasil ekstraksi pesan.



Gambar 6. Context Diagram Perangkat Lunak *Image Steganography*

Dari *context* diagram selanjutnya digambarkan dua proses utama yang terdapat dalam perangkat lunak yang dikembangkan, dituangkan dalam diagram arus data level 0 seperti dapat dilihat pada Gambar 7. Dalam diagram arus data level 0 terdapat 2 proses utama, yaitu proses penyisipan pesan dan ekstraksi pesan. Pada proses penyisipan pesan diperlukan data *stego key* sebagai kunci dalam menyisipkan pesan, *cover image* sebagai media penyembunyian pesan serta pesan yang akan disisipkan sebagai data rahasia. Sedangkan pada proses ekstraksi data, untuk memunculkan kembali pesan yang telah disisipkan diperlukan kunci untuk mengekstrak pesan dari *stego image*. Sebagai hasil dari proses ekstraksi pesan, maka pesan yang disembunyikan akan diberikan kepada user.



Gambar 7. DFD Level 0 Perangkat Lunak

Langkah terpenting dalam pengembangan perangkat lunak *image steganography* menggunakan Visual Basic 6.0 adalah dengan mengambil setiap komponen warna RGB yang ada dalam setiap pixel di citra digital. Untuk mendapatkan komponen warna tersebut gunakan segmen kode program seperti berikut.

1) Membuat tipe data sendiri (user defined data type) berupa record untuk menyimpan komponen warna RGB dari citra digital, dengan segmen kode program berikut.

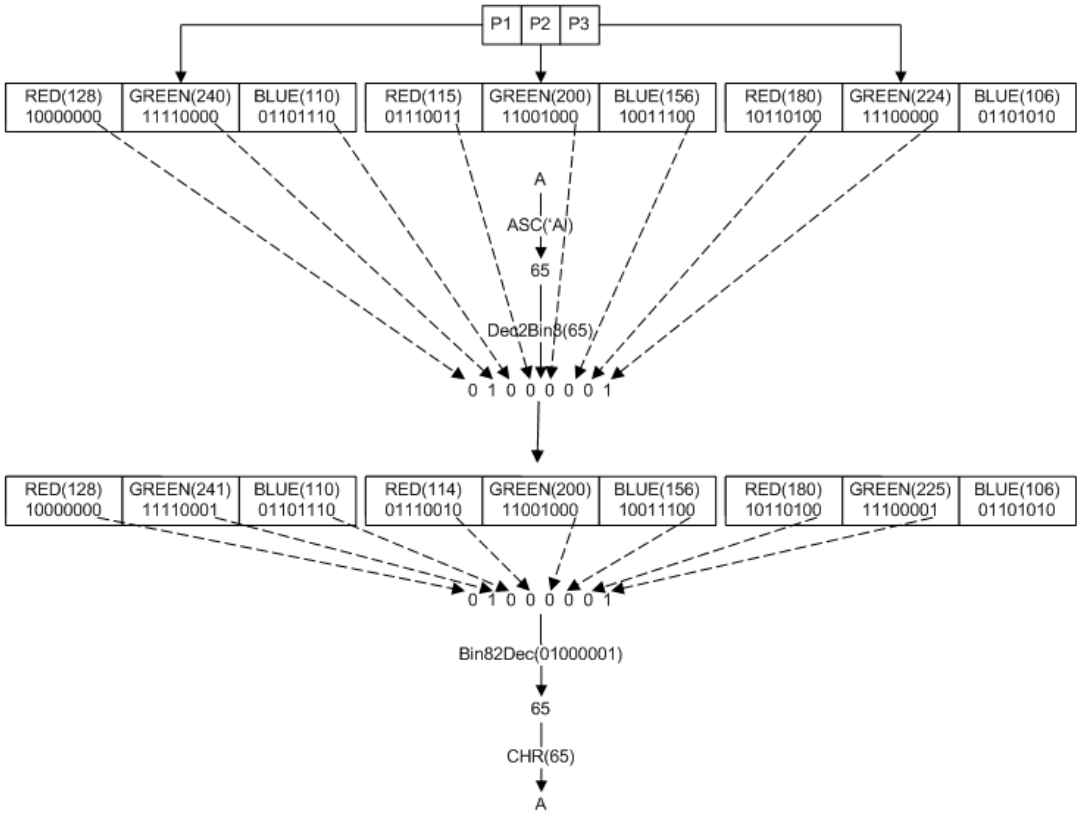
```
Public Type tRGB24
R As Byte
G As Byte
B As Byte
End Type
Public vImage(0 To 237, 0 To 237) As tRGB24
```

2) Menggunakan perulangan untuk memindai setiap pixel yang ada dalam citra digital, untuk mendapatkan komponen warna RGB, dengan segmen kode program berikut.

```
Dim scanX, scanY, TempCol As Long
Dim cRed, cGreen, cBlue As Integer

Picture1.ScaleMode = vbPixels
Picture1.AutoRedraw = True

For scanY = 0 To Picture1.ScaleHeight - 1
   For scanX = 0 To Picture1.ScaleWidth - 1

      TempCol = Picture1.Point(scanX, scanY)
      cRed = TempCol And &HFF
      cGreen = (TempCol \ &H100) And &HFF
      cBlue = (TempCol \ &H10000) And &HFF

      'menempatkan nilai RGB dalam vImage
      vImage(scanX, scanY).R = cRed
      vImage(scanX, scanY).G = cGreen
      vImage(scanX, scanY).B = cBlue
      TempCol = RGB(cRed, cGreen, cBlue)

   Next scanX
Next scanY
```

Ada beberapa tahapan dalam proses menyisipkan pesan rahasia ke dalam citra digital true colour 24 bit. Tahapan tersebut dapat dilihat pada Gambar 8. Misalnya akan menyisipkan sebuah huruf A dalam citra true colour 24 bit. Huruf A tersebut terlebih dahulu harus dikonversi menjadi kode ascii yang menghasilkan bilangan desimal 65. Di dalam Visual Basic 6.0 untuk mendapatkan nilai ascii menggunakan fungsi ASC(). Selanjutnya nilai desimal 65 dikonversi menjadi bilangan biner 8 bit menggunakan fungsi buatan sendiri (*user defined function*) bernama Dec2Bin8(). Setelah diperoleh 8 bit hasil konversi, selanjutnya masing-masing bit akan disipkan untuk menggantikan bit ke-8 dari setiap pixel. Karena 1 pixel terdiri dari komponen warna RGB, maka dalam 1 pixel dapat menampung sebanyak 3 bit. Sehingga untuk menyisipkan karakter A, diperlukan 3 pixel walaupun pada pixel ke-3 hanya memerlukan 2 komponen warna R dan G saja.

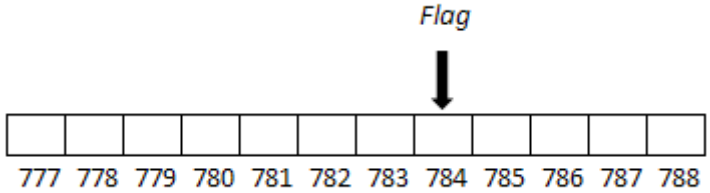
Dapat dilihat bahwa dari proses penyisipan menggunakan metode *Least Significat Bit* (LSB) hanya 3 komponen warna saja yang berubah yaitu P1(G) dari 240 →241, P2(R) dari 115→114 dan P3(G) dari 224→225. Kenaikan ataupun penurunan intensitas warna hanya selisih 1 nilai saja, sehingga mata manusia biasa tidak mampu membedakan dengan warna image sebelum disisipi pesan.



Gambar 8. Tahapan Penyisipan Pesan

Seperti yang dilihat pada Gambar 8, untuk melakukan proses eksraksi pesan berupa karakter A diperlukan fungsi buatan sendiri Bin82Dec( ) yang berguna untuk mengkonversi dari bilangan biner 8 bit menjadi bilangan desimal, untuk selanjutnya bilangan desimal tersebut dikonversi kembali menjadi karakter huruf A menggunakan fungsi CHR( ).

Dalam proses penyisipan dan ekstraksi pesan digunakan sebuah *flag* sebagai kunci untuk menandai sampai *bit* ke berapa pesan selesai disisipkan. Hal ini disebabkan karena *cover image* memiliki ruang *bit* yang lebih besar, sehingga perlu dibatasi sampai *bit* ke berapa proses penyisipan selesai dilakukan. Sebagai ilustrasi *flag* ini dapat dilihat pada Gambar 9. Jika seandainya terjadi ruang yang disediakan oleh *cover image* lebih kecil dari pesan yang akan disisipkan, maka yang terjadi adalah *overflow*, yaitu proses penyisipan dihentikan karena melebihi kapasitas media sisip.



Gambar 9. Ilustrasi *Flag* dalam Proses Penyisipan dan Ekstraksi

Seperti yang dilihat pada Gambar 9 proses ekstraksi pesan hanya akan mengambil bit sampai *flag* 784, *bit* ke 785 dan seterusya tidak akan diproses. Selanjutnya setiap *bit* akan dikonversi menjadi bilangan desimal, selanjutnya diubah menjadi karakter. Dengan adanya *flag* sebagai kunci ini tidak akan melakukan proses memindai seluruh *bit* yang ada dalam *stego image* melainkan sebatas *flag* yang diberikan.

HASIL dan ANALISIS

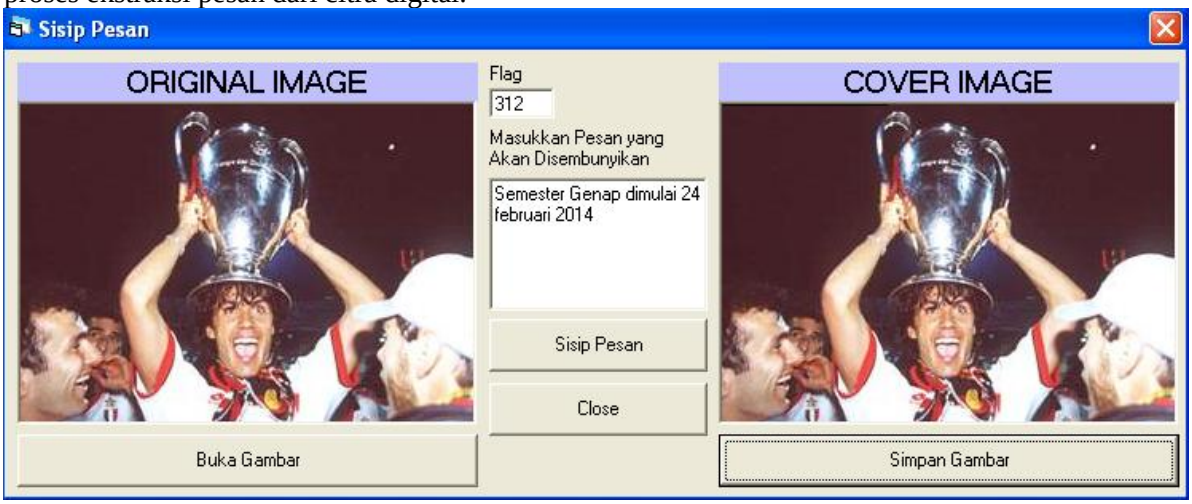
Penelitian ini menghasilkan sebuah perangkat lunak untuk menyisipkan pesan teks sebagai sebuah pesan tersembunyi ke dalam sebuah citra digital. Perangkat lunak ini dibangun

menggunakan bahasa pemrograman visual basic 6.0 yang memiliki beberapa dukungan untuk pemrograman citra digital. Untuk menampung citra pada saat proses penyembunyian dan pembacaan pesan digunakan kontrol *picture box*. Tabel 1 memberikan beberapa fungsi penting yang digunakan dalam mengembangkan perangkat lunak.

Tabel 1. Daftar Fungsi dalam Pengembangan *Image Steganography*

No.	Nama Fungsi	Jenis Fungsi	Keterangan
1.	Dec2Bin8( )	User Defined Function	Digunakan untuk mengubah bilangan desimal menjadi bilangan biner 8 bit
2.	Bin82Dec( )	User Defined Function	Mengubah bilangan biner 8 bit menjadi bilangan desimal
3.	PesanKeBiner ( )	User Defined Function	Mengubah sederetan karakter pesan rahasia menjadi deretan bilangan biner 8 bit
4.	replaceBitKe8 ( )	User Defined Function	Digunakan untuk mengganti bit ke-8 setiap komponen warna dalam RGB, pada proses penyisipan pesan
5.	ambilBitKe8 ( )	User Defined Function	Digunakan untuk mengambil bit ke-8 setiap komponen warna dalam RGB pada proses ekstraksi pesan
6.	CHR( )	Built in function	Digunakan untuk mendapatkan karakter dari kode ascii
7.	ASC( )	Built in function	Digunakan untuk mendapatkan nilai ascii dari sebuah karakter

Dalam implementasinya untuk menyisipkan pesan, cukup dilakukan dengan memilih sebuah citra digital sebagai *cover image*, kemudian menuliskan sebuah pesan yang akan disisipkan ke dalam citra digital. Selanjutnya citra digital yang telah disisipi pesan disimpan ulang sebagai *stego image*. Proses penyisipan pesan dapat dilihat pada Gambar 10. Pada saat proses penyisipan citra selesai dilakukan, akan terbentuk sebuah *flag* yang berfungsi untuk menandai sampai bit ke berapa dalam citra digital disisipi bit pesan rahasia. Penanda bit ini digunakan untuk melakukan proses ekstraksi pesan dari citra digital.



Gambar 10. Menyisipkan Citra Digital.

Proses pembacaan pesan cukup dilakukan dengan memilih sebuah citra digital yang telah disisipi pesan rahasia. Selanjutnya dengan memasukkan flag user cukup mengklik tombol Baca Pesan untuk menampilkan pesan yang tersembunyi dalam file citra digital. Proses pembacaan pesan dapat dilihat pada Gambar 11. Pada gambar terlihat bahwa *stego image* yang sudah berisi pesan rahasia kualitas gambarnya tidak jauh berbeda dengan *cover image* yang belum disisipi pesan rahasia.

Gambar 11. Proses Pembacaan Pesan

Untuk menentukan kualitas citra digunakan metode *peak signal to noise ratio* (PSNR) sebagai pembanding kualitas citra hasil rekonstruksi (*stego image*) dengan citra asli (*cover image*). Istilah *peak signal-to-noise ratio* (PSNR) adalah sebuah istilah dalam bidang teknik yang menyatakan perbandingan antara kekuatan sinyal maksimum yang mungkin dari suatu sinyal digital dengan kekuatan derau yang mempengaruhi kebenaran sinyal tersebut. Oleh karena banyak sinyal memiliki *dynamic range* yang luas, maka PSNR biasanya diekspresikan dalam skala *logarithmic decibel* [11]. Formula untuk menghitung PSNR adalah sebagai berikut:

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

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

PSNR didefinisikan melalui *signal-to-noise ratio* (SNR). SNR digunakan untuk mengukur tingkat kualitas sinyal. Nilai ini dihitung berdasarkan perbandingan antara sinyal dengan nilai derau. Kualitas sinyal berbanding lurus dengan dengan nilai SNR. Semakin besar nilai SNR semakin baik kualitas sinyal yang dihasilkan. Tabel 1 memperllihatkan hasil perhitungan nilai PSNR yang direpresentasikan dalam skala desibel (dB).

Tabel 1. Nilai MSE dan PSNR dengan Pesan dan Output Citra Digital yang Berbeda

Cover Image	Teks Pesan	Stego Image	MSE(dB)	PSNR(dB)
winner.bmp (165 kb)	Semester Genap dimulai 24 februari 2014	winner-semester.bmp	1.3967	46.68
winner.bmp (165 kb)	Kita ketemu di hari valentine	winner-valentine.jpg	0.86916	48.74
winner.bmp (165 kb)	Bulan Februari	winner-februari.png	0.4955	51.18

Seperti hasil perhitungan pada Tabel 1 menunjukkan bahwa penyisipan sebuah teks pesan dengan ukuran yang berbeda-beda akan menghasilkan nilai MSE dan PSNR yang berbeda pula. Semakin besar ukuran file pesan maka nilai MSE akan semakin besar dan nilai PSNR semakin kecil, begitu pula sebaliknya semakin kecil ukuran file pesan maka nilai MSE semakin kecil dan nilai PSNR akan semakin besar. Jika nilai PSNR tersebut kecil maka dapat dikatakan kualitas citra semakin buruk itu artinya kualitas citra secara fisik buruk pula. Sedangkan apabila nilai PSNR besar maka kualitas citra tetap bagus, yang artinya kerusakan pada citra relatif sedikit.

KESIMPULAN

Dari penelitian yang telah dilakukan dapat disimpulkan beberapa hal sebagai berikut:

1. Steganografi merupakan teknik yang sangat efisien dan kuat yang memungkinkan untuk mengirimkan pesan secara aman dan tersembunyi.
2. Metode LSB yang diterapkan pada proses penyembunyian pesan tidak mempengaruhi kualitas dari *cover image* secara signifikan.
3. Perangkat lunak Visual Basic 6.0 dapat digunakan untuk menerapkan *image steganography* karena memiliki dukungan berupa objek-objek dan fungsi-fungsi *builtin* yang siap digunakan.

DAFTAR RUJUKAN

[1] Khandekar, S. A.; Dixit, M. R. 2012. Steganography for Text Messages Using Image. *IOSR Journal of Electronics and Communication Engineering (IOSRJECE)ISSN: 2278-2834 Volume 2, Issue 3 (July-Aug 2012), PP 01-04*.

[2] Bateman, P. 2008. *Image Steganography and Stageanalysis*. M. Sc. Thesis University of Surrey. United Kingdom.

[3] Kaur. 2012. A Steganography Implementation based on LSB & DCT. *International Journal for Science and Emerging Technologies with Latest Trends 4(1): 35-41 (2012) ISSN No. (Online):2250-3641*

[4] Ramaiya, M.; Hemrajani, N.; Saxena, A.K. 2013. Secured Steganography Approach Using AES. *International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR) ISSN 2249-6831 Vol. 3, Issue 3, Aug 2013, 185-192*

[5] Dewi, A.K. 2007. *Studi dan Implementasi Penyembunyian Data di Dalam File Video Digital dengan Metode Least Significant Bit Modification, Tugas Akhir Program Sarjana*. Bandung: Teknik Informatika ITB

[6] Wijaya, E. S.; Prayudi, Y. 2004. Konsep Hidden Message Menggunakan Teknik Steganography Dynamic Cell Spreading. *Jurnal Media Informatika, Vol. 2 No. 1, Juni 2004, 23-38, ISSN:0854-4743.* Yogyakarta: Universitas Islam Indonesia

[7] Krisnawati. 2008. Metode Least Significant Bit (LSB) dan End Of File (EOF) Untuk Menyisipkan Citra Ke dalam Citra Grayscale. *Seminar Nasional Informatika (semnasIF 2008).* Yogyakarta: Universitas Pembangunan Nasional

[8] Hussain, M. 2013. A Survey of Image Steganography Techniques. *International Journal of Advanced Science and Technology, Vol. 54, May, 2013*

[9] Kusrini. 2007. *Strategi Perancangan dan Pengelolaan Basis Data*. Yogyakarta: STMIK Amikom

[10] Suhata. 2005. *VB sebagai Pusat Kendali Peralatan Elektronik*. Jakarta: PT.Elex Media Komputindo.

[11] Alfatwa, F. D. 2005. *Watermarking Pada Citra Digital Menggunakan Discrete Wavelet Transform*. Bandung: Institut Teknologi Bandung.