Jurnal Ilmiah
ILMU KOMPUTER
Universitas Udayana

Vol. 9, No. 1, April 2016 **ISNN 1979 - 5661**

Implementasi Algortima *Selected Least Significant Bit* yang Dimodifikasi untuk Menyimpan Informasi pada Gambar

[1]Michael Senna Saputra ,[2]I Gede Wira Kusuma Jaya
[3] Ni Luh Putu Krisna Lestari, [4]Agus Muliantara
Program Studi Teknik Informatika, Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana
Email :[1]michael.senna@cs.unud.ac.id , [4]muliantara2001@yahoo.com

Abstrak

Steganografi merupakan sebuah ilmu teknik komunikasi yang "tidak terlihat". Biasanya digunakan untuk menyembunyikan rahasia di berbagai format file. Terdapat banyak teknik steganografi dimana beberapa di antaranya memiliki kelemahan dan kelebihannya masing-masing. SLSB merupakan teknik perbaikan dari LSB yang menyembunyikan datanya disalah satu warna saja, namun teknik SLSB tidak memiliki pertahanan terhadap gambar yang dibalik/flip yang menyebabkan informasi dapat menghilang

Sehingga diperlukan sebuah peningkatan pada algoritma SLSB, yang membuat SLSB memiliki metode pertahanan terhadap transformasi citra. Dilakukan teknik marking/penandaan yang bertujuan untuk mengetahui letak informasi, panjang informasi, sekaligus posisi informasi sehingga proses flip tidak akan menghilangkan data pada gambar.

Pada jurnal ini akan dijelaskan mengenai modifikasi terhadap algoritma SLSB untuk meningkatkan ketahanannya terhadap serangan yang mampu menghilangkan informasi didalamnya. Dimana Algoritma yang termodifikasi ini mampu mendeteksi informasi pada gambar berformat BMP atau PNG yang dibalik dan juga memiliki tingkat keberhasilan sebesar 99% dalam penyisipan informasi.

Kata Kunci : Steganografi, LSB, SLSB, PNG, BMP.

Abstract

Steganography is a communication method which is "invisible". It usually used to hide secret in many kind of file, there are many steganography method which has its own weakness and excess. SLSB is an improved method of LSB which hide information at one color instead of three, but SLSB doesn't have defense method against flipped image which may cause information loss.

So we need an improvement to SLSB algorithm, which should make SLSB have defend mechanism against image transformation. We implemented marking technique to know the position, length, and location of the information so that flipping process didn't delete the information on the image

In this paper, would be explained the improved method SLSB need in order to increase its defend against attack that would delete information it protect. Where the improved algorithm could detect information at flipped BMP or PNG image and has 99% of success rate in hiding information.

Keyword : Steganografi, LSB, SLSB, PNG, BMP.

1. Pendahuluan

Data merupakan sekumpulan fakta-fakta yang diperoleh dalam suatu pengamatan atau penelitian tertenu dimana data belum mengalami pengolahan, jika sudah mengalami pengolahan maka akan menjadi suatu informasi. Informasi adalah data yang telah diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat. Informasi dapat bersifat terbuka yang artinya dapat diketahui oleh orang banyak atau bersifat tertutup yang artinya tidak dapat diketahui oleh orang lain atau rahasia.

Informasi yang rahasia ini harus benar-benar terjaga, agar hanya orang-orang terentu yang dapat mengetahuinya. Ada berbagai cara untuk menyembunyikan informasi tersebut, diantaranya adalah dengan cara kriptografi. Informasi tersebut ke dalam bentuk kode-kode tertentu sehingga sulit dimengerti oleh orang lain dan hanya yang memilki kunci tertentu dapat membuka informasi tersebut. Selain kriptografi, dapat juga dengan menyembunyikan informasi tersebut ke dalam suatu gambar yang disebut dengan steganografi.

Berbeda dengan teknik kriptografi yang dengan mudah terdeteksi keberadaanya (walaupun sulit untuk dimengerti), steganografi meyamarkan keberadaan informasi (data) yang ingin disampaikan ke dalam media penyamar, misalnya media yang berbentuk berkas

multimedia. Kelebihan dari steganografi adalah pesan yang dikirim tidak menarik perhatian. Ada beberapa metode yang dapat digunakan dalam steganografi, yakni LSB (Least Significant Bit) dan SLSB (Selected Least Significant Bit).

LSB adalah bit-bit yang jika diubah tidak akan berpengaruh secara nyata terhadap kombinasi warna yang dihasilkan oleh ketiga komponen warna RGB. Bit-bit LSB ini terdapat pada 4 bit akhir dalam 1 byte(8 bit). Metode ini paling sering digunakan dalam melakukan steganografi. Namun terdapat kelemahan dalam metode ini, yakni ukuran gambar yang dihaslikan relatif besar. Sehingga diperbaharuilah metode LSB menjadi SLSB.

SLSB bekerja dengan bit paling signifikan dari salah satu komponen warna piksel dalam gambar dan mengubah mereka sesuai dengan bit pesan untuk menyembunyikan. Sisa bit dalam komponen warna piksel yang dipilih juga diubah agar mendapatkan warna terdekat yang asli dalam skala warna [3]. Sehingga gambar yang dihasilkan relatif lebih kecil dari metode yang menggunakan LSB.

2. Landasan Teori

Landasan teori merupakan bagian yang penting dalam memahami dasar teori dan sebagai acuan dalam menyelesaikan permasalahan. Karena itu pada bagian ini akan dijelaskan teori-teori yang digunakan oleh penulis, sehingga permasalahan yang diangkat akan menjadi lebih jelas dan mudah dipahami.

2.1 Steganografi

Kata steganografi berasal dari Yunan yang berarti tertutup atau tulisan tersembunyi. Steganografi sudah dikenal sejak 440 SM. Herodotus menyebutkan dua contoh steganografi di dalam "*Histories of Herodotus*"[2]. Demeratus mengirimkan peringatan akan serangan Yunan yang selanjutnya dengan menuliskan pesan tersebut di atas sebuah papan kayu dan melapisinya dengan lilin. Papan lilin sangat umum digunakan sebagai permuakaan tulis yang dapat digunakan kembali, terkadang digunakan untuk menulis cepat.

Steganografi adalah teknik penyembunyian data rahasia ke dalam sebuah media sehingga data yang disembunyikan sulit dikenali oleh indera penglihatan manusia. Steganografi membutuhkan dua properti yaitu media penampung dan data rahasia yang akan disembunyikan. Steganografi digital menggunakan media digital sebagai media penampung, misal gambar, suara, teks dan video.

Data rahasia yang disembunyikan juga dapat berupa gambar, suara, teks atau video. Penggunaan steganografi antara lain bertujuan untuk menyamarkan eksistensi atau keberadaan data rahasia, sehingga sulit dideteksi dan dilindungi hak cipta suatu produk. Steganografi dapat dipandang sebagai kelanjutan kriptografi. Jika pada kriptografi data yang telah disandikan (ciphertext) tetap tersedia, maka dengan steganografi ciphertext tersebut dapat disembunyikan sehingga pihak ketiga tidak mengetahui keberadaannya.

Secara garis besar, teknik penyembunyian data dengan steganografi adalah dengan cara menyisipkan sepotong demi sepotong informasi asli pada sebuah media, sehingga informasi tersebut tampak kalah dominan dengan media pelindungnya.

Dalam data digital, teknik-teknik yang sering digunakan dalam steganografi modern ada empat jenis metode, yaitu :

1. *Least Significant Bit Insertion*
2. *Pixel Mapping Technique*
3. *Masking and Filtering*
4. *Algorithms Compression and Transformation*

2.2 *Least Significant Bit (LSB)*

Citra digital dapat dipandang sebagai kumpulan piksel dengan masing-masing piksel memiliki nilai tertentu yang dinyatakan dalam bilangan biner. Pada setiap *byte* dari piksel citra, terdapat bit yang paling kecil bobotnya (*Least Significant Bit* atau LSB) [1]. File bitmap 24 bit maka setiap piksel (titik) pada gambar tersebut terdiri dari susunan tiga warna merah, hijau dan biru (RGB) yang masing-masing disusun oleh bilangan 8 bit (*byte*) dari 0 sampai 255 atau dengan format biner 00000000 sampai 11111111.

Perubahan yang dihasilkan terlalu kecil, sehingga sulit dikenali oleh mata manusia. Kekurangan dari teknik ini adalah karena teknik ini menggunakan setiap piksel dalam sebuah citra, format kompresi yang menjaga keutuhan data seperti bmp atau gif harus digunakan sebagai citra dan juga menyebabkan terjadinya perubahan warna yang ditampilkan pada citra. Apabila format kompresi yang tidak menjaga keutuhan data digunakan, beberapa informasi tersembunyi dapat hilang. Akan lebih baik jika menggunakan *image grayscale* karena perubahan warna akan lebih sulit dideteksi oleh mata manusia.

Misalnya suatu citra 24-bit (R=8-bit, G=8-bit, B=8-bit) digunakan sebagai wadah untuk menyimpan data berukuran 100 bit, jika masing-masing komponen warnanya (RGB) digunakan satu piksel untuk menyimpan informasi rahasia tersebut, maka setiap pikselnya disimpan 3 bit informasi, sehingga setidaknya dibutuhkan citra wadah berukuran 34 piksel atau setara 34 x 3 x 8 = 816 bit (8 kali-lipat). Jadi suatu citra 24-bit jika digunakan untuk menyimpan informasi rahasia hanya mampu menampung informasi maksimum berukuran 1/8 dari ukuran citra penampung tersebut.

2.3 *Selected Least Significant Bit (SLSB)*

Selected Least Significant Bit (SLSB) adalah salah satu algoritma dalam metode steganografi. SLSB meningkatkan kinerja metode LSB dalam menyembunyikan informasi hanya pada salah satu dari tiga warna pada setiap piksel dari gambar cover[3]. Algoritma *Selected Least Significant Bit* (SLSB) ini menyaring gambar cover dengan menggunakan *filter default* dan menyembunyikan informasi di area-area yang lebih baik. *Filtering* diterapkan pada bit yang paling signifikan dari setiap *piksel*, meninggalkan yang kurang signifikan untuk menyembunyikan informasi. *Filtering* memastikan untuk memilih area dari gambar yang memiliki dampak paling sedikit dengan masuknya informasi, dimana mempengaruhi tingkat kesulitan dalam mendeteksi keberadaan pesan yang tersembunyi.

Pengambilan informasi dipastikan karena bit yang digunakan untuk *filtering* tidak berubah, melainkan melakukan *filtering* kembali untuk memilih bit yang sama dalam proses penyembunyian. Sehingga metode ini merupakan metode yang sangat efisien untuk menyembunyikan informasi.

Algoritma SLSB memiliki cara kerja dengan mengambil *bit* yang paling signifikan dari salah satu komponen warna piksel sebuah gambar dan perubahan suatu gambar disesuaikan dengan *bit* pesan untuk menyembunyikannya. Sisa dari *bit* dalam komponen warna piksel yang dipilih juga diubah agar mendapatkan warna yang terdekat dengan yang asli dalam skala warna. Metode ini merupakan sebuah metode baru yang telah dibandingkan dengan sebuah kasus yang menyerupai dan bekerja pada domain spasial dan perbedaan besar yaitu fakta bahwa bit LSB dari setiap komponen warna *piksel* tidak digunakan untuk menyimpan pesan pada gambar, melainkan berasal dari komponen warna piksel yang dipilih.

2.3 Modifikasi Algoritma SLSB

Pada algortima SLSB merupakan algoritma perbaikan dari algortima LSB, dimana hanya salah satu *channel* warna yang dipilh untuk menyimpan informasi. Bit yang digunakan adalah bit terakhir dari biner nilai *channel* warna yang dipilih. Dalam algortima ini dihasilkan ukuran gambar yang lebih kecil daripada gambar hasil penyisipan informasi menggunakan algortima LSB.

Namun, dalam algortima yang diterangkan dalam jurnal yang ada [3]. SLSB didesain untuk menjadi kebal terhadap penyerangan statistikal ataupun dengan membandingkan histogram, namun tidak dijelaskan mengenai perubahan pada gambar setelah disembunyikannya informasi, terutama untuk proses balik/flip, hal tersebut bisa mengakibatka hilangnya informasi apabila gambar yang disisipi informasi menggunakan SLSB di balik/flip. Untuk itu, dilakukan beberapa modifikasi pada algortima SLSB yang telah ada.

Untuk tahap penyisipan informasi pada gambar, masih meggunakan algortima yang ada pada SLSB yakni nilai *channel* warna terkecil yang digunakan. Jika akan menyisipkan gambar, akan ada satu buah piksel penanda yang menyatakan bahwa pada gambar tersebut terdapat informasi rahasia sehingga dietahuilah bahwa pada gambar tersebut terdapat informasi atau tidak.

3. Perancangan

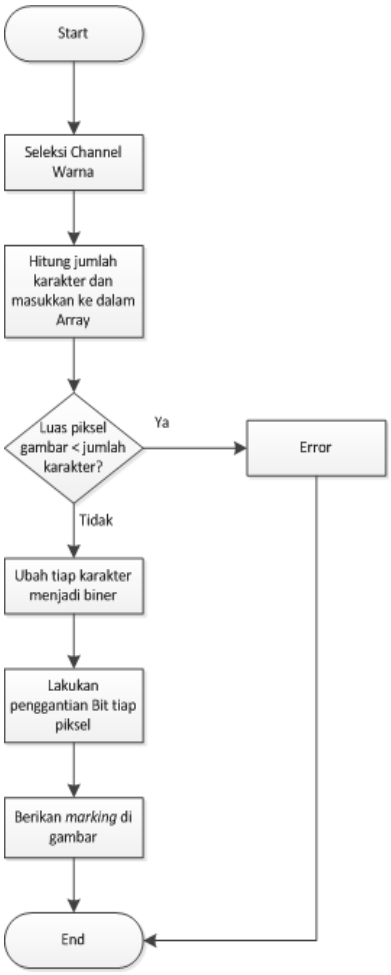
Percancangan program dilakukan dengan membagi fungsi utama program menjadi dua bagian, yakni fungsi penyisipan informasi pada gambar (*hide*) dan fungsi pengambilan informasi dari gambar (*extract*). Algoritma diterapkan dalam program dengan menggunakan bahasa pemrograman C# disertai antarmuka yang memedai bagi pengguna (*user interface*).

3.1 Flowchart Penyisipan (Hide)

Proses awal dilakukan dalam penyisipan informasi pada gambar adalah melakukan seleksi *channel* warn. Jika ditemukan jumlah warna terkecil, maka warna tersebutlah sebagai tempat penyimpanan informasi.

Kemudian informasi yang akan disisipkan jika melebihi luas piksel gambar tidak diijinkan dan jika tidak maka proses dilanjutkan

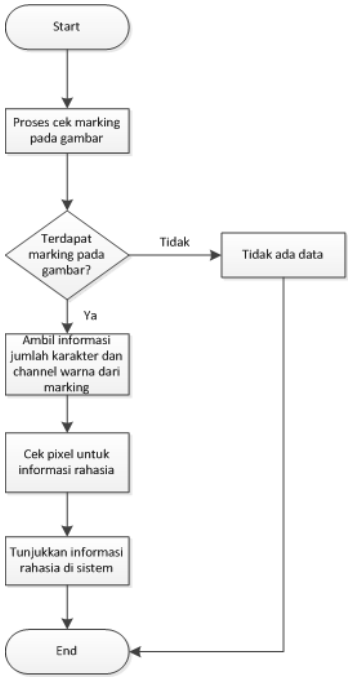
mengubah setipa karkater ke bentuk biner dan nilai biner tersebut akan menggantikan nilai bit pada *channel* warna yang telah dipilih mulai dari yang terkecil (dari kanan).Langkah terarkhir adalah memberikan piksel penanda (*marking*) pada gambar sebagai bentuk bahwa pada gambar tersebut terdapat informasi.



Gambar 1 Flowchart Penyisipan

3.2 Flowchart Pengambilan (Extract)

Untuk mengambil informasi yang terdapat dalam gambar dilakukan pemeriksaan terhadap piksel penanda, jika ditemukan maka pada gambar tersebut terdapat informasi dan jika tidak maka dapat dipastikan bahwa pada gambar tersebut tidak terdapat informasi. Pengambilan informasi berdasar jumlah karakter yang ada dan *channel* warna dari piksel penanda. Kemudian dilakukan pengambilan nilai dari *channel* warna penyimpan informasi lalu menampilkannya ke program.



Gambar 2 Flowchart Pengambilan

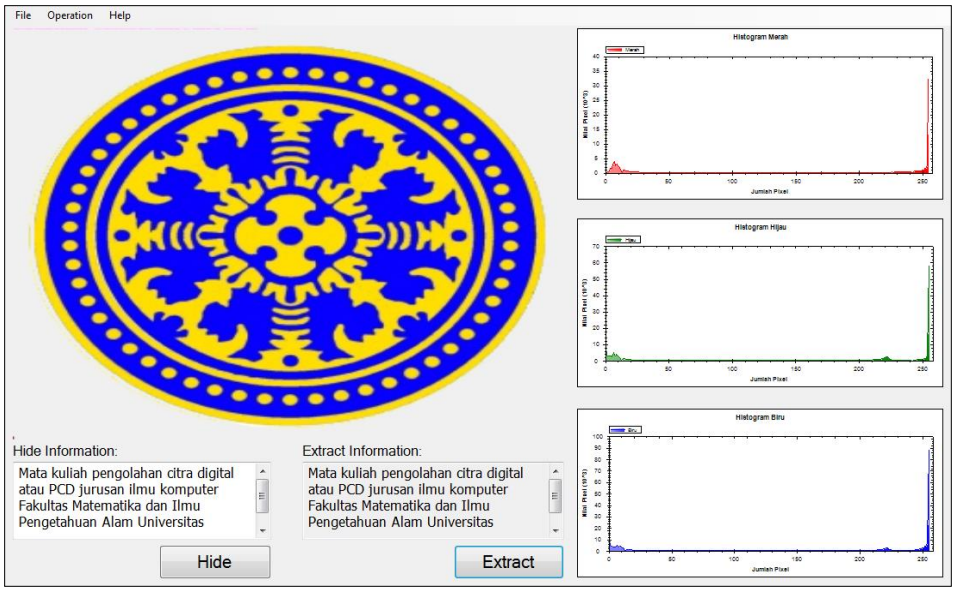
4. Implementasi

Berdasar pada teori algoritma SLSB yang ada kami melakukan implementasi berbentuk program dengan menggunakan bahasa pemrograman C#, langkah-langkah kerja program yang telah dibuat adalah sebagai berikut

1. Pilih jenis gambar yang diinginkan, BMP atau PNG
2. Lakukan pemilihan *channel* warna yang digunakan untuk penyisipan data, dimana dari tiga *channel* warna yang ada yakni merah, biru dan hijau. Dipilih nilai total warna keseluruhan gambar paling kecil
3. Ubah pesan yang ingin disisipkan menjadi nilai biner
4. Lakukan penggantian atau *replace* data di gambar dengan data biner di langkah 2, nilai digit terakhir dari nilai biner *channel* warna yang telah ditentukan

5. Berikan tanda atau *marking* yang menunjukkan bahwa di gambar memiliki pesan rahasia
6. Langkah terakhir adalah simpan gambar dengan ekstensi BMP atau PNG

Dari setiap langkah yang telah ada, kami membuat program dengan antarmuka yang memadai sehingga dapat dengan mudah melakukan penyisisipan data atau informasi pada gambar yang ditentukan



Gambar 3 Program Steganografi

Dari program tersebut (Gambar 3), berikut penjelasan masing bagian pada program yang telah dibuat untuk melakukan proses penyimpanan informasi pada gambar

1. *File* terdiri dari
   - *Open* digunakan untuk membuka gambar yang akan digunakan
   - *Save* digunakan untuk menyompan gambar yang digunakan
   - *Exit* keluar dari program yang digunakan
2. *Operation* terdiri dari
   - *Flip Horizontal* digunakan untuk melakukan *flip* secara horizontal terhadap gambar yang digunakan
   - *Flip Vertical* digunakan untuk melakukan *flip* secara vertikal terhadap gambar yang digunakan
3. *Help* hanya terdiri dari *About* yang digunakan untuk melihat anggota yang telah mengerjakan program tersebut
4. *Picture Box* digunakan untuk meletakan gambar yang akan digunakan
5. *TextBox Hide Infromation* digunakan untuk menampilkan informasi yang akan dimasukan ke dalam gambar
6. *TextBox Extract Information* digunakan untuk menampilkan informasi rahasia yang tersimpan dalam gambar
7. *Button Hide* digunakan untuk memasukan informasi rahasia pada gambar
8. *Button Extract* digunakan untuk mengambil informasi rahasis dari gambar
9. *ZedGraphControl Red Channel* digunakan untuk menampilakan *channel* merah
10. *ZedGraphControl Green Channel* digunakan untuk menampilakan *channel* hijau
11. *ZedGraphControl Blue Channel* digunakan untuk menampilakan *channel* biru

Jika membuka program tersebut pertama kali, maka bagian *TextBox* dan *Button* tidak akan aktif sehingga tidak dapat digunakan. Bagian tersebut akan aktif atau dapat digunakan jika pengguna telah memilih gambar dan secara otomatis histogram warna gambar tersebut akan tampil pada *ZedGrapghControl* pada masing-masing *channel* warna. Untuk *TetxBox Extract Information* bersifat *read only* sehingga hanya dapat dilihat atau dibaca saja. Jika proses penyisipan sukses maka akan tampil pemberitahuan bahwa proses sukse dan jika tidak ada informasi pada gambar maka jika menekan *Button Extract* akan muncul tampilan bahwa tidak informasi pada gambar.

5. Analisis Hasil

Hasil dari implementasi pada program yang pertama yang kami gunakan dengan menggunakan dua buah gambar, dimana satu buah gambar dengan menggunakan ekstensi BMP (*Bitmap*) dan satu buah gambar dengan ekstensi PNG (*Portable Network Graphic*) dengan jumlah karakter yang disisipkan adalah 133 karakter.

(a) Gambar percobaan oleh Roque

(b) Gambar percobaan Roque menggunakan SLSB yang dimodifkasi

(c) Gambar png berwama

(d) Gambar PNG berwama uji coba SLSB yang dimodifikasi

Dalam percobaan yang dilakukan (gambar a,b,c dan d), dilakukan percobaan menggunakan gambar yang digunakan pada penelitian oleh Roque (a) dan juga pada gambar PNG lainnya yang berwarna (c), Jika secara sekilas dilihat, maka gambar (b) dan (d) akan terlihat sama tapi jika dilihat dengan baik maka akan terlihat perbedaanya walapun sangat tipis terjadi perubahan warna pada deretan piksel kiri atas.

Tabel 1 Persentase Penympanan Informasi pada Gambar

Jenis Gambar	Jumlah Gambar	Gambar Berhasil	Persentase Keberhasilan
BMP	100	99	99 %
PNG	100	98	98 %

Kemudian dilakukan kembali pengujian dengan menggunakan gambar dengan ektensi BMP dan PNG. Masing-masing gambar berjumlah 100 buah dengan berbagai ukuran besar gambar, dimensi dan beragam jenis warna. Gambar berupa hewan, tumbuhan, kota, pemandangan dan lain sebagainya. Informasi yang dimasukan menggunakan informasi dengan menggunakan informasi kalimat:

Mata kuliah pengolahan citra digital atau PCD jurusan ilmu komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Udayana

Pengujian dilakukan mulai dari penyisipan gambar hingga pengambilan informasi pada gambar.Setelah dilakukan pengujian tersebut (Tabel 2), nilai persentase keberhasilan sangat tinggi yakni di atas 90 persen atau hampir 100 persen untuk setiap ekstensi gambar. Selanjutnya dilakukan pengujian jumlah karakter yang mampu ditampung pada setiap ekstensi gambar. Untuk itu digunakan satu jenis gambar sama yang diubah ekstensinya ke BMP dan PNG dengan dimensi 1920 piksel x 1200 piksel.

Tabel 2 Karakter Maksimum Disimpan dalam Gambar

Jumlah Karakter	BMP	PNG
133	Berhasil	Berhasil
200	Berhasil	Berhasil
300	Berhasil	Berhasil
400	Berhasil	Berhasil
800	Berhasil	Berhasil
1600	Berhasil	Berhasil
3200	**Gagal**	**Gagal**
1920	**Gagal**	**Gagal**
1800	Berhasil	Berhasil
1900	Berhasil	Berhasil
1919	**Berhasil**	**Berhasil**

Piksel gambar	Jumlah I*	Hasil BMP	Hasil PNG	Jumlah II**	Hasil BMP	Hasil PNG	Jumlah III***	Hasil BMP	Hasil PNG
240	239	Berhasil	Berhasil	240	Gagal	Gagal	241	Gagal	Gagal
400	399	Berhasil	Berhasil	400	Gagal	Gagal	401	Gagal	Gagal
800	799	Berhasil	Berhasil	800	Gagal	Gagal	801	Gagal	Gagal
1280	1279	Berhasil	Berhasil	1280	Gagal	Gagal	1281	Gagal	Gagal
1600	1599	Berhasil	Berhasil	1600	Gagal	Gagal	1601	Gagal	Gagal

*) Jumlah I : jumlah karakter ialah nilai piksel gambar dikurangi 1
**) Jumlah II : jumlah karakter sesuai dengan nilai piksel gambar
***) Jumlah III : jumlah karakter ialah nilai piksel gambar ditambah 1

Tabel 3 Karakter Maksimum Disimpan pada Gambar Menurut Panjang Pikselnya

Perhatikan hasil percobaan Tabel 2, jumlah karakter 3200 merupakan jumlah yang melebihi panjang piksel dari gambar yang digunakan (melebihi nilai 1920) sedangkan jumlah karakater 1920 merupakan jumlah yang sama dengan nilai panjang piksel gambar yang digunakan yakni 1920 piksel. Hasil penyisipan berhasil, namun saat pengambilan informasi mengalami kegagalan jadi dapat diartikan kedua proses tersebut gagal.

Untuk jumlah karakter kurang dari 1920 mengalami kesuksesan penyisisapan dan pengambilan informasi. Khusus untuk baris tabel berwarna hijau dengan jumlah karakter 1919 merupakan nilai kurang satu (-1) dari panjang piksel gambar yang digunakan yakni sebesar 1920. Untuk itu, kami melakukan kembali percobaan untuk mencari jumlah karakter maksimum yang dapat ditampung dalam gambar dengan program yang telah dibuat.

Dengan mengunakan lima buah gambar sama yang berbeda ektensi (Tabel 3), maka dihasilkan data seperti tabel di atas. Jumlah karakter yang melebihi atau sama dengan panjang piksel mengalami kegagalan sedangkan jumlah karakater kurang dari panjang piksel gambar yang digunakan mengalami keberhasilan. Jadi, jumlah karakter maksimum yang dapat ditampung adalah

$$maxln = pix - 1$$

Dimana maxln adalah jumlah informasi maksimum yang bisa ditampung sebuah gambar, dan pix merupakan nilai panjang piksel gambar. Setelah berhasil menyimpan informasi dalam gambar, maka selanjutnya dilakukan test transformasi citra, test ini membandingkan algoritma SLSB menggunakan aplikasi buatan Cédric Bonhomme [4] dengan aplikasi algoritma SLSB yang telah dimodifikasi untuk melihat peningkatan ketahanan yang berhasil dicapai. Hasil dari perbandingan bisa dilihat di Tabel 4, pada transformasi citra, algoritam SLSB asli tidak mampu mempertahankan informasi pada gambar yang menyebabkan hilangnnya informasi sehingga pada seluruh test transformasi citra, algoritma SLSB gagal.

Sedangkan untuk SLSB modifikasi mampu mempertahankan informasi untuk proses flipping baik untuk flip horizontal ataupun flip vertical untuk seluruh gambar (dengan catatan proses penyisipan data berhasil dilakukan). Namun, untuk transformasi citra berupa *rotation*, informasi pada gambar tidak ditemukan karena piksel penanda (*marking*) tidak ditemukan. Untuk transformasi citra berupa *scaling*, informasi ditemukan dalam kondisi rusak, hal ini terjadi karena piksel penyimpan informasi mengalami perubahan posisi atau jumlah piksel dalam deretan piksel tersebut bertambah.

Transformasi Citra	SLSB	SLSB modifikasi
Flip Horizontal	X	✔
Flip Vertical	X	✔
Rotation	X	X
Scaling	X	X

Tabel 4 Transformasi Citra pada Gambar

6. Kesimpulan

Berdasar pada penelitian yang telah dilakukan, maka dapat diambil kesimpulan

1. Algortima SLSB yang telah dimodifikasi dan diterapakan dalam bentuk program dapat melakukanpenyisipan informasi ke gambar dan pengambilan informasi dari gambar dengan baik untuk gambar dengan ekstensi BMP dan PNG.
2. SLSB modifkasi mampu menyimpan data sebanyak Nilai Panjang Piksel Gambar–1.
3. Ukuran gambar sebelum dan sesudah penyimpanan tidak berubah drastis bahkan beberapa gambar tidak terdapat perubabahan ukuran, sehingga program ini sangat baik digunakan.
4. SLSB modifikasi mampu membuat gambar yang dibalik/flip tetap dapat menyimpan informasi dengan baik karena

adanya fase marking dalam prosesnya yang membuat data tetap dapat terdeteksi meskipun di balik/flip berulang kali.

7. Saran

Berikut merupakan saran untuk peningkatan pada penelitian selanjutnya mengenai SLSB modifikasi. Pertama adalah jumlah jenis karakter yang terbatas (hanya 64 karakter), mulai dari angka 0 sampai 9, alfabet kecil dan kapital serta karakter spasi dan tanda seru (!) sehingga perlu dikembangkan pengkompressan karakter ke biner atau kamus untuk kata-kata yang belum ada. Jumlah karakter yang disimpan maksimum adalah jumlah panjang piksel gambar kurang satu piksel, semakin panjang gambar semakin besar daya tampung informasi. Maka perlu dilakukan pengembangan cara penyimpanan data.

Selain itu SLSB mdofikasi hanya mampu mempertahankan informasi terhadap transformasi citra berupa flipping sehingga perlu dilakukan penilitian lanjutan terhadap transformasi citra berupa rotasi dan scalling.

8. Referensi

[1] Arnia, Fitri. (2009). Implementasi Steganografi Dengan Metode Least Significant Bit.

[2] Susanti, I. (2007). Penerapan Steganografi Gambar Pada Least Significant Bit (LSB) Dengan Pengunaan Prng (Pseudo Random Number Generator).

[3] Roque, J. J., & Minguet, J. M. (2009). SLSB: Improving the Steganographic Algorithm LSB. In WOSIS (pp. 57-66).

[4] Bonhomme , Cédric. (2013) Stéganô (Version 0.4) [Computer program]. Available at https://bitbucket.org/ cedricbonhomme/stegano