ISSN : 1693 - 1173

Modifikasi *Least Significant Bit* dalam Steganografi

Wawan Laksito YS [1)]

Abstrak

Algoritma Least Significant Bit (LSB) merupakan teknik yang umum digunakan dalam penyisipan pesan Steganografi. Untuk memperkuat penyembuanyian pesan dilakukan modifikasi teknik LSB. Modifikasi dilakukan dengan teknik penggunaan pixel gambar yang tidak berurutan (random pixel),penyisipan pada LSB bit ke-1 dan LSB bit ke-2 secara bergantian dan acak (random lsb),mengubah seluruh lsb pada gambar (modify all).

I. Pendahuluan

Steganografi merupakan teknik yang digunakan untuk menyisipkan data teks pada media gambar dan menguraikannya. Pada proses penyisipan pesan (*embedding text*), teks disisipkan dengan metode *LSB(Least Significant Bit)* sehingga menghasilkan gambar stego (gambar yang mengandung pesan tersembunyi). Gambar stego inilah yang merupakan hasil dari pengolahan data pada program aplikasi. Dengan gambar inilah diharapkan data teks menjadi aman.

LSB merupakan salah satu teknik yang paling umum digunakan dalam steganografi karena kesederhanaan dalam implementasinya dan teknik ini memberikan pengaruh terkecil pada cover image karena hanya mempengaruhi 1 atau 2 bit pada LSB. Walaupun demikian, membuat program semacam ini bukanlah suatu hal mudah, mengingat dan menimbang adanya banyak faktor teknis dan keamanan yang harus diperhatikan.

Untuk memperkuat keamanan penyisipan pesan dilakukan modifikasi dari teknik LSB, dimana modifikasi dilakukan dengan mengacak posisi lsb dan bit data pada cover pesan.

II. Tujuan

Menentukan algoritma pengacakan bit *cover media* dan LSB dalam proses penyandian data teks dengan steganografi.

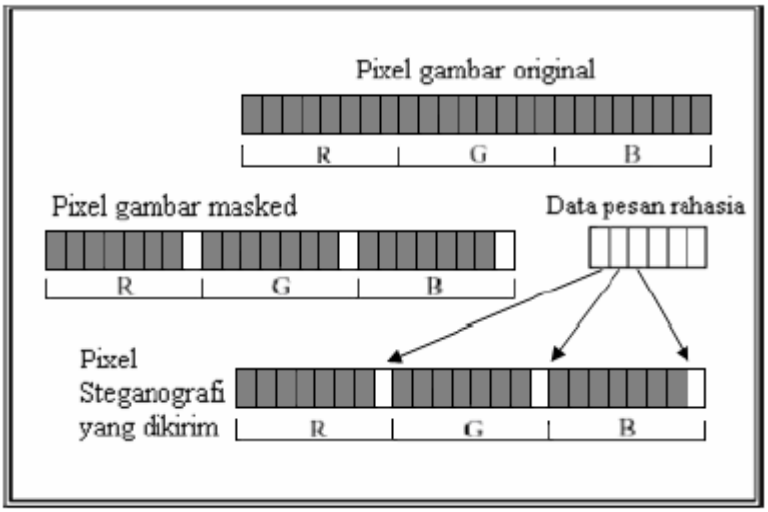
[1)] Staf Pengajar STMIK Sinar Nusantara Surakarta

III. Penyelesaian

1.1 Teknik *Least Significant Bit*

Metode yang dipakai dalam proses penyisipan bit-bit pesan ke dalam bit-bit gambar adalah dengan menggunakan teknik LSB *Insertion* atau Penyisipan pada LSB. LSB (*Least Significant Bit*) adalah bit yang mempunyai nilai paling rendah, atau bit yang berada pada posisi paling kanan. Penyisipan LSB dilakukan dengan memodifikasi bit terakhir dalam satu byte data. Bit yang diganti adalah LSB karena perubahan pada LSB hanya menyebabkan perubahan nilai *byte* satu lebih tinggi atau satu lebih rendah. Misalkan data yang diubah adalah warna hijau, maka perubahan pada LSB hanya menyebabkan sedikit perubahan yang tidak dapat dideteksi oleh mata manusia. Data yang akan disisipkan adalah data teks atau karakter. Bit bit pesan disisipkan ke dalam LSB (*least significant bit*) gambar.

Proses penyembunyian pesan dilakukan dengan menyisipkan 1 bit pesan pada LSB (bit pertama) secara langsung untuk setiap posisi yang bersesuaian.



Seperti diketahui untuk file bitmap 24 bit maka setiap *pixel* (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`. Dengan demikian pada setiap *pixel* file bitmap 24 bit dapat disisipkan 3 bit data.

Contohnya huruf UB dapat kita sisipkan dalam 6 pixel.
Misalnya data gambar original adalah sebagai berikut:

```
(00100111 11101001 11001000)
(00100111 11001000 11101001)
(11001000 00100111 11101001)
(00100111 11101001 11001000)
(00100111 11001000 11101001)
(00100111 11101001 11001000)
```

Sedangkan representasi biner huruf UB adalah 01010101 01000010.
Dengan menyisipkan-nya pada data pixel diatas maka akan dihasilkan:

```
(00100110 11101001 11001000)
(00100111 11001000 11101001)
(11001000 00100111 11101000)
(00100111 11101000 11001000)
(00100110 11001000 11101001)
(00100110 11101001 11001000)
```

Untuk memperkuat teknik penyembunyian data, dapat ditambahkan beberapa metode yang merupakan modifikasi dari metode yang telah dijelaskan di atas.
Modifikasi yang dilakukan antara lain :

- penggunaan pixel gambar yang tidak berurutan (*random pixel*),
- penyisipan pada LSB bit ke-1 dan LSB bit ke-2 secara bergantian dan acak (*random lsb*),
- mengubah seluruh *lsb* pada gambar (*modify all*).

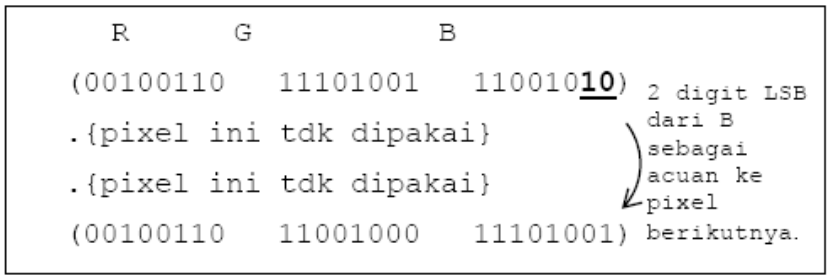
1.2 Random Pixel

Random pixel artinya bit-bit data rahasia tidak digunakan mengganti pixel yang berurutan, namun dipilih susunan pixel secara acak. Misalnya jika terdapat 20 pixel dan 6 bit data yang akan disembunyikan, maka *pixel* yang diganti bit *LSB*-nya dipilih secara acak, misalkan pixel nomor 1, 3, 6, 7, 10, 14. Parameter yang digunakan sebagai acuan untuk menuju pixel berikutnya adalah 2 bit *lsb* dari komponen warna *blue* dari pixel yang sedang ditempati. Penggunaan metode ini akan mengurangi daya tampung gambar, karena tidak semua pixel dipakai untuk

menyembunyikan pesan. Prosedur penyembunyian bit pesan secara *random pixel* adalah sebagai berikut:

a. Tentukan f(x,y). f(x,y) adalah pixel dengan posisi koordinat x,y.
b. Dapatkan nilai byte dari komponen warna Blue (B) dari f(x,y).
c. Dapatkan 2 lsb dari B dengan n = B AND (11)2
d. Tambahkan nilai n ke x. f'(x,y) = f'(x+n, y)

Ilustrasi Random Pixel seperti terlihat pada gambar berikut :

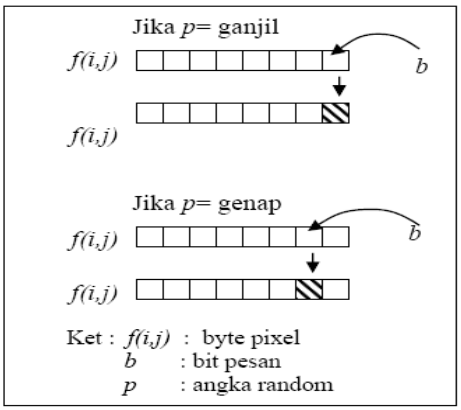


1.3 *Random LSB*

Random lsb artinya bit LSB yang dipakai untuk menampung bit pesan tidak selalu LSB bit pertama, tetapi juga memakai LSB bit kedua. Penerapan metode ini dilakukan bersamaan dengan pembangkitan angka random dari sebuah fungsi random generator sebagai acuan untuk penyisipan. Prosedur penyembunyian pesan menggunakan bit LSB yang berbeda (*random lsb*) adalah sebagai berikut:

a. Bangkitkan *p (pseudorandom number).*
b. Lakukan proses penyembunyian dengan cara menyisipkan 1 bit pesan dengan aturan seperti berikut :
   - jika *p* adalah bilangan ganjil, sisipkan 1 bit pesan *b* pada LSB (bit pertama).
   - jika *p* adalah bilangan genap, sisipkan 1 bit pesan *b* pada LSB (bit kedua) .

   Ilustrasi random lsb seperti terlihat pada gambar berikut :



Untuk membangkitkan angka-angka random, digunakan sebuah fungsi pembangkit bilangan acak dengan rumus :

$$X_n=(7X_{n-1}+11)\text{Mod}17.$$

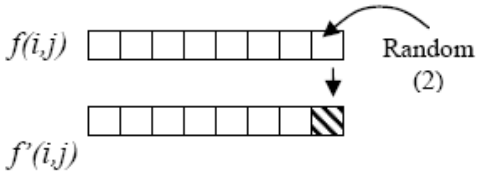
Pembangkit semacam ini disebut *Linier congruential generators* (LCG).

1.4 Mengubah Seluruh LSB (*modify all)*

Modify all atau mengubah seluruh *lsb* gambar yang bertujuan untuk menyamarkan posisi pesan yang disembunyikan dalam gambar. Dengan mengubah seluruh *lsb* dalam gambar akan menimbulkan kesan bahwa seolah-olah seluruh pixel dalam gambar memuat bit-bit pesan, padahal hanya tempat tempat tertentu saja yang disisipi pesan. Prosedur untuk mengubah seluruh lsb dalam gambar adalah sebagai berikut :

a. Tentukan *f(i,j), byte pixel* pada gambar.
b. Ubah 1 bit LSB dengan nilai 0 atau 1 secara random.

Ilustrasi *modify all* terlihat pada gambar berikut :



1.5 Penguraian Pesan

Penguraian pesan dalam gambar berarti mengambil bit-bit karakter yang tersebar dalam pixel gambar. Prosedur untuk mendapatkan kembali bit-bit karakter dalam gambar adalah sebagai berikut :

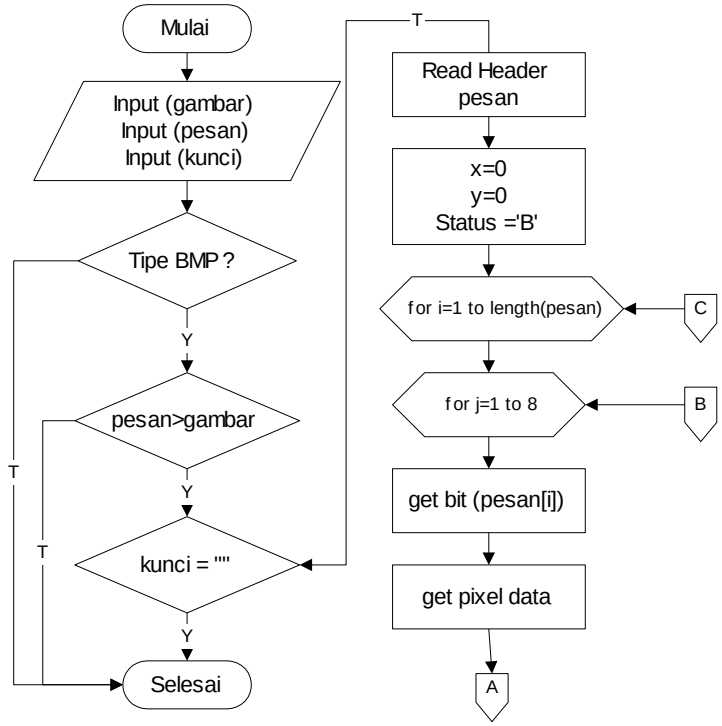
a. Tentukan *f(x,y)*

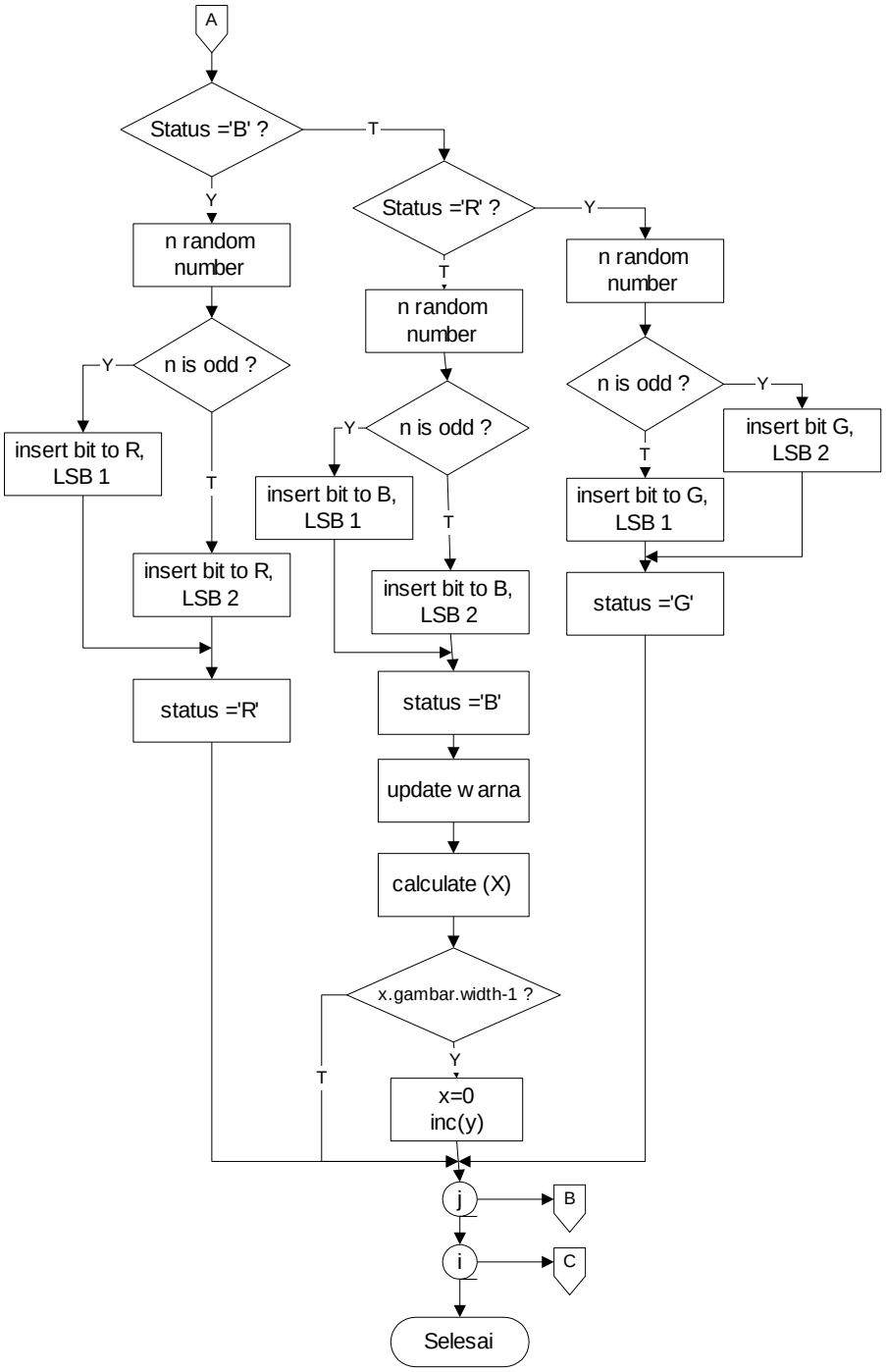
b. Untuk masing-masing komponen warna *R,G,B* dari *f(x,y),* bangkitkan angka random, jika ganjil maka dapatkan *bit* terakhir atau LSB dari masing-masing komponen warna dengan *bit* = [*R,G,B*] *AND 1,* jika genap maka dapatkan *bit* terakhir atau LSB dari masing-masing warna dengan bit = [R,G,B] *AND 2*.

c. Susun kembali LSB pada setiap elemen *lsb* hingga menjadi pesan dengan masing-masing 8-bit untuk tiap karakter.

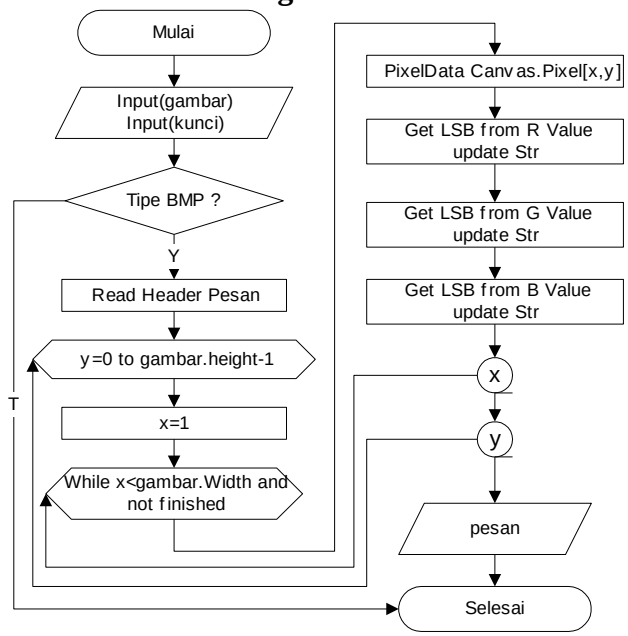
Flowchart Penyisipan dan penguraian pesan seperti diperlihatkan pada gambar berikut :

Flowchart Penyisipan Pesan




A
Status ='B' ?
T
Y
Status ='R' ?
Y
T
n random number
n random number
n random number
n is odd ?
Y
n is odd ?
Y
n is odd ?
Y
insert bit to R, LSB 1
T
insert bit to B, LSB 1
T
insert bit G, LSB 2
T
insert bit to G, LSB 1
insert bit to R, LSB 2
insert bit to B, LSB 2
status ='G'
status ='R'
status ='B'
update w arna
calculate (X)
x.gambar.width-1 ?
T
Y
x=0
inc(y)
j
B
i
C
Selesai

Flowchart Pengambilan Pesan



IV. Kesimpulan

Proses penyisipan dilakukan dengan cara mengganti bit-bit gambar dengan bit-bit pesan. Dalam proses penguraian pesan dalam gambar, pertama adalah membaca *Header Messag*e yang terdapat pada pixel[0,0] untuk mendapatkan informasi proses pembacaan pesan. Kemudian dilakukan penelusuran per pixel untuk mendapatkan bit LSB dari masing-masing komponen warna R,G,B, mulai dari pixel pertama sampai ditemukan penanda akhir pesan, yang diindikasikan dengan fungsi `finished`.

Pustaka

1. Gonzalez, Rafael C. , Woods, Richard E.. **Digital Image Processing Second Editio**n, Prentice Hall. 2002
2. Johnson, Neil F. , **Steganography**. http://www.jjtc.com/stegdoc/SEC201.htm.
3. Jorn Daub EDV, **File Formats Collection BMP**, http://www.daubnet.com/ formats/ BMP.html
4. Wohlgemuth , Sven, **Steganography and Watermarking**, http://www.informatik.unifreiburg.de/~softech/teaching/ws01/itsec