# Pengamanan Informasi Menggunakan Steganografi dengan Metode PVD (Pixel Value Differencing) dan Enkripsi Pesan Rahasia dengan Pembangkit Bilangan Acak LCG (Linear Congruential Generator) pada PT Primatama Duta Antaran

**Fitri Rachmawati [1)], Utomo Budiyanto [2)], Rusdah[3)]**
[1,2,3)]*Fakultas Teknologi Informasi, Universitas Budi Luhur*
*Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260*
*Telp. (021) 5853753, Fax. (021) 5866369*
*E-mail :* fitrirachmawati19@gmail.com[1)], utomo.budiyanto@budiluhur.ac.id[2)], rusdah@budiluhur.ac.id[3)]

**Abstract-Steganography is a technique of hiding information, data or secret message into a media such as image, video or audio. PT Primatama Duta Antaran is a company that provides warehouse rental services and delivery of goods since 2006. The exchange of informations, data or messages between employees and customers is usually done via email. Those informations stored in company data center are still in the unprotected form, so they can easily be read by anyone.Hence, an application of data security is needed to protect all secret information, data or message from unauthorized parties. Pixel Value Difference nguji method is one of the Steganography methods. It uses the difference between one pixel score and the other. The difference will be used to insert a hidden message into other media. In order to increase the security level of information, data or message that will be inserted, Steganography can be combined with encryption. One of the encryption method used is Linear Congruential Generator. The result of the study showed that the larger difference score between two pixels, the large capacity of the inserted message. The size between the Steganography and the original image is the same. The use of password on Steganography application will increase the security level of information, data or message.**
**Keywords; Steganografi, Pixel Value Differencing (PVD), Linear Congruential Generator (LCG)**

## 1. Pendahuluan

Keamanan dan kerahasiaan merupakan aspek penting yang dibutuhkan dalam proses pertukaran pesan melalui media jaringan *internet*. Melalui jaringan *internet*, memungkinkan setiap orang untuk saling bertukar data, informasi, atau pesan kepada orang lain tanpa batasan jarak dan waktu. Namun dengan berkembangnya teknologi informasi dan komunikasi, makin berkembang pula kejahatan teknologi dengan berbagai teknik interupsi (penyelaan), penyadapan, modifikasi (mengubah data atau program), maupun fabrikasi (menyisipkan objek-objek palsu). Tanpa adanya jaminan keamanan, orang lain dapat dengan mudah mendapatkan pesan atau informasi yang dikirimkan melalui jaringan *internet*.

Pengamanan informasi menggunakan teknik steganografi banyak digunakan oleh peneliti sebelumnya [1] [4] [5] [8] dengan keunggulan pada kapasitas penyimpanan [8], kemampuan untuk menyisipkan file text pada citra berformat JPEG, BMP, PNG [1], kualitas gambar sesudah proses relatif sama [4], diaplikasikan dengan baik pada citra 24 bit [5]. Teknik kriptografi menggunakan *Linear Congruential Generator*(LCG) digunakan oleh [2] yang menghasilkan data di dalamnya terjamin kerahasiaannya.

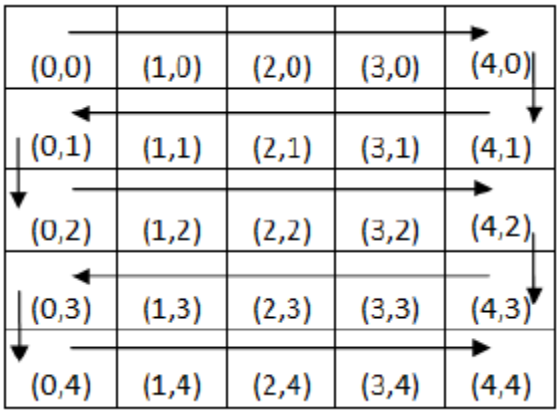
Teknik kriptografi dan steganografi dapat digunakan untuk memberi perlindungan keamanan pada pesan rahasia. Perbedaan pada kedua teknik tersebut yakni pada kriptografi adalah pesan rahasia diacak dengan sebuah algoritma enkripsi (pengacakan) terlebih dahulu sebelum pesan dikirim. Akibatnya pesan berubah menjadi kode-kode yang tidak dimengerti oleh pihak-pihak yang tidak berhak menerimanya. Adapun pada steganografi, kerahasiaan pesan terhadap pihak yang tidak berhak dilakukan dengan cara menyembunyikannya pada sebuah media yang disebut *cover*. Jika sebuah *filecover* telah disisipi pesan rahasia dan siap untuk dikirim maka *file* tersebut disebut dengan *stego image*.

*PixelValueDifferencing* (PVD). Pada Metode PVD ini menggunakan selisih nilai piksel yang satu dengan nilai piksel yang lain, dimana hasil selisih kedua piksel tersebut nantinya akan digunakan untuk menyisipkan pesan pada media lain yang ingin disembunyikan. Setelah pesan sudah disisipkan pada media penampung pesan, maka nilai piksel-piksel tersebut akan berubah. Nilainya akan berubah sesuai dengan rumus perhitungan PVD. Untuk menambah tingkat keamanan informasi, pesan maupun data yang akan disisipkan ke dalam citra, steganografi dapat dikombinasikan dengan enkripsi, sehingga informasi yang disisipkan ke dalam citra tidak mudah dibaca oleh orang lain. Metode enkripsi pesan yang digunakan adalah Pembangkit Bilangan Acak berupa LCG (*Linear Congruential Generator* ) dengan nilai ASCII *password* sebagai parameternya. *LinearCongruentialGenerator* atau jika diubah ke dalam Bahasa Indonesia menjadi Pembangkit Bilangan Acak Kongruen-Lanjar merupakan pembangkit bilangan acak yang sederhana, mudah dimengerti teorinya, dan juga mudah untuk diimplementasikan. Keunggulan Kongruen-Lanjar terletak pada kecepatannya dan hanya membutuhkan sedikit operasi *bit*.

## 2. Landasan Teori

### 2.1 Metode PVD *(Pixel Value Differencing)*

*Pixel Value Differencing* (PVD) merupakan salah satu metode yang dapat digunakan dalam steganografi. Metode ini beroperasi pada ranah spasial (spatial domain) dari citra. Penelitian awal algoritma PVD dilakukan oleh Wu et al., 2001, dengan mencari selisih nilai dua piksel terdekat [6]. Selisih digunakan untuk menentukan jumlah data yang dapat disisipkan berdasarkan jangkauan tabel yang dipilih. Berdasarkan analisis terhadap sistem penglihatan manusia yang menyatakan bahwa, mata manusia tidak sensitif terhadap perubahan pada pixel yang memiliki kekontrasan tinggi melainkan sensitif terhadap perubahan pada pixel yang memiliki kekontrasan rendah. Melalui sifat tersebut maka lebih banyak bit data rahasia yang dapat disisipkan pada pixel yang memiliki nilai kekontrasan tinggi, dan sedikit bit yang dapat disisipkan pada pixel dengan kekontrasan rendah. Hal tersebut yang menjadi dasar pemikiran metode Pixel Value Differencing (PVD) pada steganografi. Algoritma PVD adalah dengan sistem arah pencarian selisih dua piksel terdekat seperti pada Gambar 1 (Wu & Tsai, 2003):




Gambar 1: Pola Proses Penyisipan File Cover

Proses penyisipan pada metode ini dilakukan dengan cara membandingkan dua pixel yang bertetangga $P_{(i)}$ dan $P_{(i+1)}$ dengan menggunakan rumus:

$$d_i = |P_{(i)} - P_{(i+1)}|$$

Hasil dari perbandingan tersebut digunakan untuk mengetahui berapa banyak *bit* yang dapat disisipkan ke dalam dua *pixel* yang dibandingkan. Metode ini menggunakan skema Wu dan Tsai untuk mengetahui *range* dari perbandingan *pixel* sebelumnya. Skema Wu dan Tsai yang digunakan yaitu R = { [0-7], [8-15], [16-31], [32-63], [64-127], [128-255] } terlihat pada tabel 1.

**Tabel 1: Range Table**

| *Range* | 0-7 | 8-15 | 16-31 | 32-63 | 64-127 | 128-255 |
|---|---|---|---|---|---|---|
| *Hiding Capacity* | 3 | 3 | 4 | 5 | 6 | 7 |

Skema ini digunakan untuk mengetahui terdapat di range mana selisih dari dua pixel tersebut, jika telah diketahui dimana letak range nya, maka kita dapat mengetahui batas bawah ($L_i$) dan batas atas ($U_i$). Setelah mengetahui batas bawah dan batas atas range-nya, kita dapat menghitung lebar (width) dari optimum range ($W_i$) dengan persamaan rumus:

$$W_i = U_i - L_i + 1$$

Langkah selanjutnya yaitu mencari jumlah *bit* pesan ($t_i$) yang dapat disisipkan melalui rumus:

$$t_i = [log_2(W_i)]$$

Penyisipan pesan dapat dilakukan dengan mengambil sebanyak $t_i$ bit dari pesan yang akan disisipkan. Setalah mengetahui berapa banyak bit pesan yang akan disisipkan, ubahlah bit-bit pesan yang akan disisipkan tersebut ke dalam decimal ($b$).Selanjutnya dihitung nilai difference value yang baru untuk penyisipan ke dalam citra menggunakan rumus:

$$d'_i = b + L_i$$

Untuk menentukan nilai piksel baru yang telah disisipkan pesan, ada beberapa aturan yang harus dipenuhi yaitu:

$$(P'_{(i)}, P'_{(i+1)}) = \begin{cases} \left(P_{(i)} + \left[\frac{m}{2}\right], P_{(i+1)} - \left[\frac{m}{2}\right]\right), jika\ P_{(i)} \geq P_{(i+1)}\ dan\ d'_i > d_i \\ \left(P_{(i)} - \left[\frac{m}{2}\right], P_{(i+1)} + \left[\frac{m}{2}\right]\right), jika\ P_{(i)} < P_{(i+1)}\ dan\ d'_i > d_i \\ \left(P_{(i)} - \left[\frac{m}{2}\right], P_{(i+1)} + \left[\frac{m}{2}\right]\right), jika\ P_{(i)} \geq P_{(i+1)}\ dan\ d'_i \leq d_i \\ \left(P_{(i)} + \left[\frac{m}{2}\right], P_{(i+1)} - \left[\frac{m}{2}\right]\right), jika\ P_{(i)} < P_{(i+1)}\ dan\ d'_i \leq d_i \end{cases}$$

Dimana $m$ didapat dari selisih $d'_i$ dengan $d_i$ menggunakan rumus:

$$m = d'_i - d_i$$

Proses-proses tersebut dilakukan terus hingga *bit* pesan tersisipi semuanya ke dalam citra.

Proses ekstraksi pesan dari citra stego menggunakan metode PVD dimulai dengan mengurutkan semua piksel pada gambar yang telah disisipkan pesan, sesuai cara pengambilan pesannya. Kemudian dihitung selisih nilai difference value baru ($d_i$). Nilai difference value baru tersebut digunakan untuk mengetahui nilai continuous ranges (R) yang sudah didefinisikan menggunakan skema Wu dan Tsai. Dengan demikian didapatkan pesan yang telah disisipkan ($b'$).

Berdasarkan informasi tersebut dapat diketahui seberapa panjang data rahasia yang disisipkan pada kedua pixel, sehingga pesan rahasia yang telah disisipkan didapatkan kembali. Proses algoritma PVD untuk ekstraksi pesan dapat dijelaskan secara singkat sebagai berikut:

1) Hitunglah selisih nilai $d_i$ dengan rumus:
$$d_i = |P'_{(i)} - P'_{(i+1)}|$$
Dimana $P'_{(i)}$ dan $P'_{(i+1)}$ adalah piksel yang telah disisipkan pesan.
2) Kemudian cari nilai $U_i$ dan $L_i$ yang merupakan nilai batas atas dan nilai batas bawah pada range table dengan menggunakan nilai $d_i$ yang telah diperoleh.
3) Setelah mengetahui batas bawah dan batas atas range-nya, kita dapat menghitung lebar (width) dari optimum range ($W_i$) dengan rumus:
$$W_i = U_i - L_i + 1$$
4) Lalu kita dapat mengetahui pesan yang telah disisipkan dengan rumus:
$$b' = d_i - L_i$$
Jika stego-image tidak berubah maka $b' = b$. Kemudian ubah ke biner.
5) Setelah itu gunakan perhitungan $t_i$ dengan rumus dibawah ini untuk mengetahui seberapa panjang bit pesan yang diekstraksi.
$$t_i = |log_2(W_i)|$$
6) Lakukan sampai semua piksel telah berhasil di proses.

## 2.2 LCG (*Linear Congruential Generator*)

Semua deretan bilangan acak yang dibangkitkan dari rumus matematika, serumit apapun, dianggap sebagai deret acak semu, karena dapat diulang pembangkitannya. Sementara itu, banyak produk *software* yang dinyatakan sebagai produk yang aman karena menggunakan bilangan acak semacam OTP (*One Time Programmable*). Namun karena OTP ini dibangkitkan dari bilangan acak semu, maka keamanan yang diperoleh juga semu. Salah satu pembangkit bilangan acak yang sering diimplementasikan adalah *Linear Congruential Generator* (LCG).

*Linear Congruential Generator* (LCG) mewakili salah satu algoritma *pseudo random number generator* yang tertua dan paling popular. Teori dari algoritma ini mudah dipahami dan dapat diimplementasikan secara cepat. Keuntungan dari LCG adalah operasinya yang sangat cepat.LCG dapat didefinisikan dengan rumusan berikut:

$$X_n = (aX_{n-1} + b) mod\ m$$

Dimana:

$X_n$ = bilangan acak ke-n dari deretnya
$X_{n-1}$ = bilangan acak sebelumnya
$a$ = faktor pengali
$b$ = *increment*
$m$ = modulus

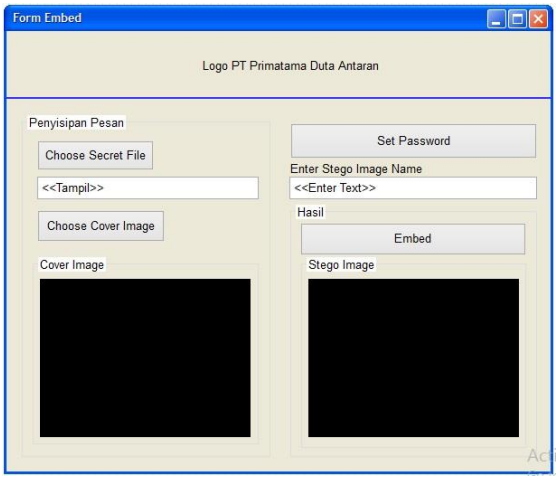
Periode dari LCG umumnya adalah sebesar nilai $m$. Masalah pada LCG adalah *lower-order bit* yang digenerasi mempunyai periode yang lebih pendek dari deretan secara keseluruhan jika M di-set menjadi pangkat 2. Tanpa desain yang benar, dengan m yang sangat besar, bisa jadi periode bilangan acak yang dihasilkan tidak akan maksimal, bahkan mungkin jauh lebih pendek daripada periode maksimalnya. Menurut para ahli kriptografi, LCG tidak dapat digunakan untuk kriptografi karena urutan keluarnya dapat diprediksi. Bahkan mereka juga dapat membongkar urutan bilangan acak yang dibangkitkan oleh pembangkit kuadratik. Namun ada beberapa parameter yang menjadikan algoritma LCG lebih baik dan lebih kuat. Salah satunya adalah penggunaan nilai ASCII *password* sebagai penentu, sehingga bilangan randomnya menjadi lebih sulit diprediksi.

## 3. RANCANGAN SISTEM DAN APLIKASI

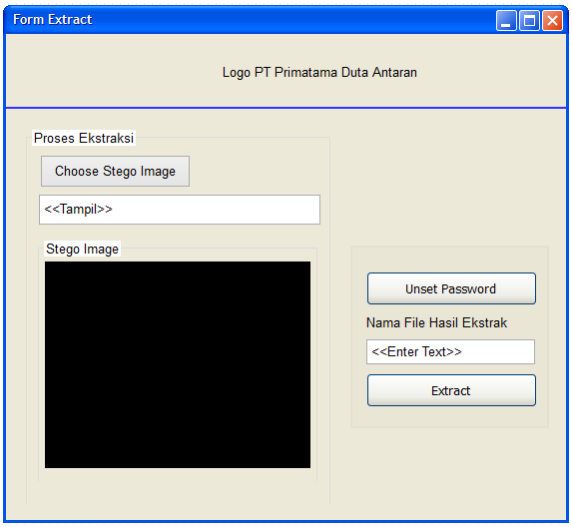
Rancangan sistem yang akan dibangun pada aplikasi ini adalah sebagai berikut:

1) Proses pengamanan data dilakukan menggunakan aplikasi berbasis desktop.
2) Keamanan dan konsistensi isi data harus terjamin.
3) Aplikasi mampu mengacak data, mengacak kata sandi, dan juga menyisipkan data dan kata sandi ke dalam bit-bit gambar, dan aplikasi ini mampu membuka data tanpa adanya perubahan pada isi data tersebut.

Pada perancangan aplikasi steganografi ini diusulkan terdapat dua menu yaitu menu *Embed* seperti pada gambar 2, menu *Extract* terlihat pada gambar 3. Untuk melakukan penyisipan pesan atau *file* informasi ke dalam gambar, *user* dapat memilih menu *Embed*. Pada menu ini memilik empat buah masukan, yaitu gambar atau *cover*, pesan atau *file* yang akan disisipkan, *password* dan nama *stego image* yang akan di simpan. Penggunaan *password* dimaksudkan agar pesan tidak dapat diekstraksi jika *user* tidak mengetahui *password*-nya. Pada proses pertama penyisipan yang akan dilakukan adalah membaca gambar dan mengubahnya menjadi bentuk data *byte*, begitu juga dengan pesan akan diubah menjadi bentuk data *byte*. Sedangkan untuk *password* akan dilakukan pembangkit bilangan acak (*pseudo random number generator*) berupa LCG sehingga menghasilkan sebuah *keystream*. Lalu setelah itu pesan akan di enkripsi dengan men-XOR kan tiap *byte* pesan tersebut dengan *byte-byte* pada *keystream*. Setelah itu maka akan menghasilkan pesan yang sudah terenkripsi. Maka pesan yang terenkripsi inilah yang akan disisipkan ke dalam gambar, dan akan menghasilkan keluaran berupa gambar yang telah tersisipi pesan tersebut. Untuk melakukan ekstraksi pesan, *user* dapat memilih menu *Extract*. Pada submenu ini memiliki tiga buah masukan, yaitu gambar atau *cover* yang telah tersisipi pesan, *password* dan nama untuk *file* hasil ekstrak. Sama seperti sebelumnya, untuk melakukan ekstraksi ini, gambar akan dibaca dan dipisahkan antara data *byte* pesan dengan data *byte* gambar. Kemudian jika pesan dalam bentuk data yang terenkripsi, maka *password* akan membangkitkan sebuah *keystream* kemudian akan dilakukan dekripsi dengan cara men-XOR-kan nya dengan pesan yang terenkripsi tersebut yang akan diubah menjadi bentuk *file*.




**Gambar 2: Rancangan Layar Form Embed**




Gambar 3: Rancangan Layar Form Extract

## 4. HASIL DAN PEMBAHASAN

Aplikasi diuji coba menggunakan *cover* 24 bit berformat bmp dengan kriteria nilai warna kurang dari nilai warna maksimal (255). Data atau *file* yang dapat disembunyikan berupa data dengan tipe *file* *.doc atau *.docx, *xls, *.pdf, *.txt.

### 4.1. Pengujian *Embed* Pesan

Pengujian *embed* dilakukan untuk mengetahui apakah gambar yang telah dipilih *user* dapat disisipkan pesan atau tidak. Tabel 2 menunjukan pengujian *embed* pesan:

Tabel 2: Pengujian Embed Pesan

| No. | Citra Asli | *File* yang disisipkan | Status |
|---|---|---|---|
| 1. | Jellyfish<br>Bitmap image<br>Date modified: 1/1/2016 4:56 PM<br>Dimensions: 1920 x 1200<br>Size: 6.59 MB | Payroll_Salary_DetailReport PR...<br>Microsoft Excel 97-2003 Worksheet<br>Date modified: 1/2/2016 2:16 PM<br>Authors: user<br>Tags: Add a tag<br>Size: 111 KB | Berhasil |
| 2. | City<br>Bitmap image<br>Date modified: 10/29/2012 12:34 PM<br>Dimensions: 3000 x 2000<br>Size: 17.1 MB | DATA CUSTOMER PDA<br>PDF Document<br>Date modified: 1/5/2016 11:45 PM<br>Size: 31.0 KB | Berhasil |
| 3. | Snake<br>Bitmap image<br>Date modified: 1/1/2016 5:02 PM<br>Dimensions: 2560 x 1600<br>Size: 11.7 MB | MASTER MOU 2013 PT. PRIMATAMA<br>Microsoft Word Document<br>Title: PERJANIIAN KERJASAMA SERV...<br>Authors: User<br>Size: 31.5 KB | Berhasil |
| 4. | Yellow_Blossom<br>Bitmap image<br>Date modified: 1/1/2016 5:20 PM<br>Dimensions: 1280 x 1024<br>Size: 3.75 MB | Tata Tertib Perusahaan<br>Text Document<br>Date modified: 1/6/2016 1:36 AM<br>Size: 1.11 KB | Berhasil |
| 5. | Strawberry<br>Bitmap image<br>Date modified: 12/13/2015 10:36 PM<br>Dimensions: 529 x 509<br>Size: 789 KB | FORMAT SLIP GAJI<br>Microsoft Excel 97-2003 Worksheet<br>Date modified: 1/6/2016 12:33 AM<br>Authors: user1<br>Tags: Add a tag<br>Size: 25.5 KB | Gagal |
| 6. | Cupcakes<br>Bitmap image<br>Date modified: 12/13/2015 11:03 PM<br>Dimensions: 300 x 335<br>Size: 294 KB | MASTER MOU 2013 PT. PRIMATAMA<br>Microsoft Word Document<br>Title: PERJANIIAN KERJASAMA SERV...<br>Authors: User<br>Size: 31.5 KB | Gagal |
| 7. | Dolphin<br>Bitmap image<br>Date modified: 12/13/2015 10:54 PM<br>Dimensions: 363 x 480<br>Size: 511 KB | Payroll_Salary_DetailReport PR...<br>Microsoft Excel 97-2003 Worksheet<br>Date modified: 1/2/2016 2:16 PM<br>Authors: user<br>Tags: Add a tag<br>Size: 111 KB | Gagal |

Pada pengujian *embed* pesan nomor 5 dan nomor 6 tidak berhasil (gagal) disebabkan karena ketika proses perhitungan nilai warna baru, nilai warna sebelumnya ditambah dan hasilnya melebihi nilai maksimal warna (255). Kemudian pada pengujian *embed* pesan nomor 7 juga tidak berhasil atau gagal, hal tersebut dikarenakan total bit pesan yang dapat disisipkan kurang dari penjang pesan x 8 bit.

### 4.2Pengujian *Extract* Pesan

Setelah dilakukan pengujian *embed* pesan, selanjutnya dilakukan pengujian *extract*. Pengujian ini dilakukan untuk pengujian *embed* pesan sebelumnya yang telah berhasil disembunyikan pesan. Jika tidak terdapat pesan yang disembunyikan maka pada saat melakukan ekstraksi akan muncul pesan yang mengatakan bahwa tidak terdapat pesan dalam gambar. Gambar yang telah disembunyikan pesan hanya dalam bentuk format *.bmp*. Untuk ukuran gambar sebelum dan sesudah disisipi pesan, untuk format selain *.bmp* akan berubah menjadi besar. Hal ini terjadi karena terjadi konversi antara format lain ke format *.bmp*. Sedangkan untuk citra asli dengan format *.bmp* tidak terjadi perubahan pada ukurannya. Tabel 3 menunjukan pengujian *extract* pesan:

**Tabel 3: Pengujian Extract Pesan**

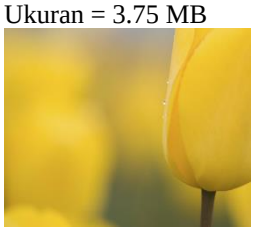
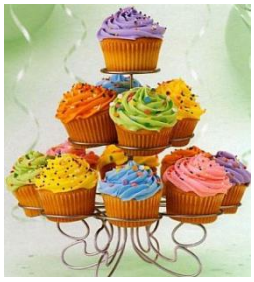
| No. | *Stego Image* | Hasil Ekstrak *File* | Status |
|---|---|---|---|
| 1. | Jellyfish Stego<br>Bitmap image<br>Date modified: 1/3/2016 4:25 PM<br>Dimensions: 1920 x 1200<br>Size: 6.59 MB | Dokumen 1<br>Microsoft Excel 97-2003 Worksheet<br>Date modified: 1/3/2016 4:26 PM<br>Authors: user<br>Tags: Add a tag<br>Size: 111 KB | Berhasil |
| 2. | City Stego<br>Bitmap image<br>Date modified: 1/3/2016 4:28 PM<br>Dimensions: 3000 x 2000<br>Size: 17.1 MB | Dokumen 2<br>PDF Document<br>Date modified: 1/6/2016 1:33 AM<br>Size: 31.0 KB | Berhasil |
| 3. | Snake Stego<br>Bitmap image<br>Date modified: 1/3/2016 4:29 PM<br>Dimensions: 2560 x 1600<br>Size: 11.7 MB | Dokumen 3<br>Microsoft Word Document<br>Title: PERJANIIAN KERJASAMA SERVI...<br>Authors: User<br>Size: 31.5 KB | Berhasil |

| | | | |
|---|---|---|---|
| 4. | Yellow Blossom Stego<br>Bitmap image<br>Date modified: 1/3/2016 4:38 PM<br>Dimensions: 1280 x 1024<br>Size: 3.75 MB | Dokumen 4<br>Text Document<br>Date modified: 1/6/2016 1:37 AM<br>Size: 1.11 KB | Berhasil |
| 5. | Strawberry Stego<br>Bitmap image<br>Date modified: 1/3/2016 4:40 PM<br>Dimensions: 529 x 509<br>Size: 789 KB | Dokumen 5<br>File<br>Date modified: 1/6/2016 1:39 AM<br>Size: 0 bytes | Gagal |
| 6. | Cupcakes Stego<br>Bitmap image<br>Date modified: 1/3/2016 6:45 PM<br>Dimensions: 300 x 335<br>Size: 294 KB | Dokumen 6<br>File<br>Date modified: 1/6/2016 1:40 AM<br>Size: 0 bytes | Gagal |

Kegagalan pada pengujian *extract* nomor 5 dan 6 terjadi karena ketika proses perhitungan nilai warna baru, nilai warna sebelumnya ditambah dan hasilnya melebihi nilai maksimal warna (255).

### 4.3 Perbandingan Citra Asli dengan *Stego Image*

Dengan dilakukan pengujian *embed* pesan dengan *extract* pesan, maka dapat dilakukan perbandingan citra dan *size* antara citra asli dengan *stego image*.Tabel 4 menampilkan perbandingan citra asli dengan *stego image*:

| No. | Citra Asli | *Stego Image* |
|---|---|---|
| 1. | Jellyfish.bmp<br>Ukuran = 6.59 MB<br> | Jellyfish Stego.bmp<br>Ukuran = 6.59 MB<br> |
| 2. | City.bmp<br>Ukuran = 17.1 MB<br> | City Stego.bmp<br>Ukuran = 17.1 MB<br> |
| 3. | Snake.bmp<br>Ukuran = 11.7 MB | Snake Stego.bmp<br>Ukuran = 11.7 MB |
| | | |
| 4. | Yellow_Blossom.bmp<br>Ukuran = 3.75 MB<br>  | Yellow Blossom Stego.bmp<br>Ukuran = 3.75 MB<br> |
| 5. | Strawberry.bmp<br>Ukuran = 789 KB<br> | Strawberry Stego.bmp<br>Ukuran = 789 KB<br> |
| 6. | Cupcakes.bmp<br>Ukuran = 294 KB<br>  | Cupcakes Stego.bmp<br>Ukuran = 294 KB<br> |

Pengujian juga dilakukan untuk mengetahui sejauh mana *user* dapat memodifikasi *stego image* tanpa merusak informasi yang *user* sembunyikan. Adapun pengujian-pengujian yang telah dilakukan adalah sebagai berikut:

1) Mengganti nama *stego image:* Saat mengganti nama *stego image* dan kemudian melakukan ekstraksi maka informasi yang disisipkan akan tetap dapat di ekstrak. Hal ini dimungkinkan karena letak informasi mengenai nama *file*tidak bersinggungan dengan bit-bit yang kita gunakan dalam proses penyisipan informasi.
2) Mengkompres *stego image*: Saat mengirim *stego image* (misalnya lewat *email*) terkadang merasa perlu untuk mengkompres *stego image* untuk memudahkan dan mempersingkat waktu pengiriman. Berdasarkan pengujian yang dilakukan, hasil setelah proses ekstraksi

dilakukan adalah informasi yang disisipkan menjadi 0 KB. Hal ini dimungkinkan ketika dikompresi, tipe *file stego image* yang semula adalah *.bmp* menjadi *.jpg* atau *.jpeg* dan ukurannya menjadi sangat jauh berbeda. Faktor-faktor tersebutlah yang mungkin menyebabkan informasi yang berada didalam *stego image* tersebut menjadi terpotong atau hilang.

Me-*resize stego image*: Saat me-*resize stego image* kemudian mengekstraknya, informasi yang disisipkan menjadi gagal di ekstrak, *password* yang juga berada di dalam *stego image* tersebut menjadi tidak diketahui. Hal ini dimungkinkan karena ukurannya berubah jauh meskipun tipe *file* tetap sama yaitu *.bmp*, dan dimungkinkan juga terpotongnya atau hilangnya *password* yang berada di dalamnya sehingga informasi yang berada didalamnya tidak dapat diungkapkan kembali.

## 5 KESIMPULAN DAN SARAN

Adapun kesimpulan yang diperoleh setelah melewati tahap perancangan, pembuatan, serangkaian uji coba dan analisa program aplikasi steganografi ini, maka dapat dibuat suatu kesimpulan antara lain:

a. Program aplikasi yang dibuat berhasil mengimplementasikan metode PVD sebagai proses *embedding* dan *extraction* nya dan pembangkit bilangan acak LCG sebagai enkripsi dan dekripsi pesan dan *password* nya.

b. Semakin besar selisih kedua piksel ( dan ), maka semakin besar kapasitas pesan yang ditampung.

c. Gambar dengan kriteria tertentu yang dijadikan media penampung *file* rahasia susah dibedakan dengan gambar aslinya.

d. Ukuran gambar asli dengan *stego image* tetap sama (tidak berubah).

e. Dengan menggunakan *password* pada proses penyisipan dan ekstraksi maka keamanan data atau pesan rahasia yang disisipkan pada gambar dapat terjaga dengan lebih aman.

f. Pada saat proses perhitungan nilai warna baru di proses *embed*, jika nilai warna sebelumnya ketika ditambah hasilnya melebihi warna maksimal (255), maka proses penyisipan pesan gagal.

g. Gambar yang mengalami perubahan (misalnya cropping, resize, compress), isi pesannya akan hancur.

h. Sulit menemukan kriteria gambar yang bisa dijadikan media penampung pesan.

## SARAN

Dengan prototype ini, beberapa saran yang perlu dipertimbangkan guna pengembangan aplikasi lebih lanjut, antara lain:

a. Melakukan implementasi metode PVD dan LCG berbasis *mobile device* atau web.
b. Mengembangkan metode PVD dengan memodifikasi pola pengambilan *pixel* nya, seperti pola spiral, diagonal atau heksagonal.
c. Mengkombinasikan metode PVD dengan metode enkripsi pesan atau pembangkit bilangan acak selain LCG.

## DAFTAR PUSTAKA


[1] Betananda, Yohanes. 2013. *Perancangan Program Aplikasi Steganografi Citra menggunakan Metode Pixel Value Differencing Diagonal*. *Skripsi*. Tidak dipublikasikan. Jakarta : Bina Nusantara.

[2] Bilqis. 2012. *Analisa dan Perancangan Aplikasi Pesan Rahasia Menggunakan Algoritma One Time Pad (OTP) dengan Pembangkit Bilangan Acak Linear Congruential Generator (LCG)*. *Skripsi*. Tidak dipublikasikan. Medan : Universitas Sumatera Utara.

[3] Butar, Klementina Butar. 2015. *Implementasi Steganografi Metode EOF pada PT Star Three Sixty dengan Pesan Terenkripsi menggunakan Vigenere Cipher*. *Skripsi*. Tidak dipublikasikan. Jakarta : Universitas Budi Luhur.

[4] Kusumawati, Vivi. 2013. *Perancangan Program Aplikasi Steganografi Citra menggunakan Metode Pixel Value Differencing Spiral*. *Skripsi*. Tidak dipublikasikan. Jakarta : Bina Nusantara.

[5] Mahardika, Raditya. 2010. *Steganografi pada Citra Digital dengan Menggunakan Metode Tri-Way Pixel Value Differencing*. *Skripsi*. Tidak dipublikasikan. Bandung : Institut Teknologi Telkom.

[6] Mandal, J.K & Das, Debashis. 2012. *Colour Image Steganography Based on Pixel Value Differencing in Spatial Domain*. International Journal of Information Sciensces and Techniques, Vol. 2:83-93.

[7] Minartiningtyas, Brigida Arie. 2013. *Linear Congruential Generator (LCG) (online)*, diakses 18 Oktober 2015.<http://informatika.web.id/linear-congruential-generator-lcg.htm>.

[8] Putra, M. Ramadhan Dharma. 2014. *Aplikasi Steganografi untuk Keamanan Basis Data dengan Metode Pixel Value Differencing (PVD) dan Algoritma*

*Rijndael*. *Skripsi*. Tidak dipublikasikan. Bandung : Universitas Komputer Indonesia, Bandung.

[9] Rojali. 2009. *Perbaikan dan Evaluasi Kinerja Algoritma Pixel Value Differencing (PVD). Skripsi*. Tidak dipublikasikan. Bogor : Institut Pertanian Bogor.

[10] Wu, Da-Chun & Tsai, Wen-Hsiang. 2003. *A Steganographic Method for Images by Pixel-Value Differencing*. Pattern Recognition Letters, Vol. 24:1613-1626.

[11] Wulandari, Ni Putu. 2015. *Implementasi Steganografi dengan Metode BPCS (Bit Plane Complexity Segmentation) untuk Menyembunyikan Pesan pada Citra Digital*. *Skripsi.* Tidak dipublikasikan. Jakarta : Universitas Budi Luhur.