STUDI MENGENAI APLIKASI STEGANOGRAFI CAMOUFLAGE

Anggya N.D. Soetarmono, S.Kom.*

ABSTRAK

Steganografi adalah ilmu dan seni menyembunyikan pesan rahasia di dalam pesan lain sehingga keberadaan pesan rahasia tersebut tidak dapat diketahui secara kasat mata. Secara umum, steganografi dapat dikatakan sebagai suatu teknik yang digunakan untuk menyimpan data di dalam data lainnya. Penggunaan steganografi dapat dilakukan untuk berbagai jenis data, seperti citra, audio, atau bahkan video. Beberapa metode dapat digunakan untuk menyimpan informasi dalam berbagai jenis data tersebut, antara lain metode LSB (*least significant byte*), *spread spectrum*, ataupun *injection*.

Saat ini telah terdapat banyak aplikasi yang diciptakan untuk memfasilitasi penggunaan steganografi, bahwa steganografi tidak hanya diperuntukkan untuk para ahli namun juga dapat digunakan secara luas oleh masyarakat awam. Salah satu contoh aplikasi steganografi yang ada saat ini adalah *Camouflage*. *Camouflage* memungkinkan pengguna komputer untuk menjaga keamanan arsip-arsip personal yang dimilikinya tetap aman dari pengganggu. *Camouflage* memungkinkan pengguna untuk menyembunyikan arsip dengan mengacaknya dan menyisipkannya ke dalam arsip lain yang dipilih. Arsip yg telah di-*camouflage* akan tetap tampak dan berlaku seperti arsip normal lainnya, dan dapat disimpan atau dikirim tanpa menimbulkan kecurigaan apapun. Penelitian ini akan membahas mengenai steganografi, pengenalan terhadap aplikasi *Camouflage* berikut dengan penggunaanya, pembahasan mengenai bentuk data yang telah dimanipulasi oleh *Camouflage*, dan pemecahan algoritma yang digunakan oleh *Camouflage* dalam menyembunyikan informasi ke dalam data yang ada.

Kata Kunci: *Camouflage, Steganografi, LSB(Least Significant Byte)*

1. PENDAHULUAN

Seperti telah dijelaskan sebelumnya, steganografi adalah ilmu dan seni untuk menyembunyikan pesan rahasia di dalam pesan lain sehingga keberadaan pesan rahasia tersebut tidak dapat diketahui. Steganografi dapat dipandang sebagai kelanjutan kriptografi dan dalam prakteknya pesan rahasia dienkripsi terlebih dahulu, kemudian cipherteks disembunyikan di dalam media lain sehingga pihak ketiga tidak menyadari keberadaannya. Pesan rahasia yang disembunyikan dapat diekstraksi kembali persis sama seperti aslinya. Terdapat beberapa teknik dalam melakukan penyembunyian data menggunakan steganografi, dan sejumlah perangkat lunak telah tersedia, antara lain yang banyak digunakan yaitu metode LSB (*least significant byte)*, *spread spectrum*, kunci publik steganografi, domain transformasi, dan *embedding/injection.*

* Staf Pengajar Program Studi S1-Sistem Informatika IKADO

1.1. Latar Belakang

Dari sejumlah aplikasi yang diciptakan untuk memfasilitasi penggunaan steganografi, salah satunya adalah *Camouflage*. *Camouflage* adalah suatu aplikasi steganografi yang memungkinkan pengguna komputer untuk menjaga keamanan dari arsip-arsip yang dimilikinya dari pihak yang tidak bertanggung jawab. Sesuai dengan bidangnya, *Camouflage* melakukan penyembunyian arsip dengan teknik steganografi yaitu menyembunyikan suatu arsip rahasia ke dalam arsip lainnya.

Sebagai contoh, pengguna dapat membuat suatu arsip gambar yang tampak seperti arsip normal lainnya namun sebetulnya terdiri dari suatu arsip enkripsi yang tersembunyi. Untuk keamanan tambahan, *Camouflage* juga menyediakan *password* dalam penyembunyian arsip. *Password* tersebut kemudian akan dibutuhkan saat akan mengekstraksi arsip yang bersangkutan menjadi arsip normal yang telah di-*uncamouflage*. Seperti aplikasi lainnya, *Camouflage* juga memiliki suatu teknik tertentu dalam melakukan penyisipan data ke dalam data lainnya.

1.2. Perumusan Masalah

Perumusan masalah dalam penelitian ini dapat dijabarkan sebagai berikut:

1. Mengerti apa itu steganografi dan metode-metode yang digunakan dalam steganografi.
2. Mengerti dan memahami tentang *software camouflage*.
3. Membahas salah satu metode yang digunakan oleh *software camouflage,* yaitu metode LSB (*least significant byte*) dan metode enkripsi Caesar chiper.
4. Manfaat dari *software camouflage* beserta kekurangan dari *software* tersebut.

1.3. Tujuan dan Manfaat Penelitian

Tujuan dari penelitian ini adalah untuk memberikan pandangan bahwa teknik steganografi memiliki tingkat keamanan yang cukup tinggi dibandingkan dengan teknik kriptografi. Lalu bagaimana steganografi dapat diterapkan dalam *file* multimedia ataupun *file-file* yang lain. Kemudian memaparkan cara kerja *software camouflage* dan algoritma yang digunakan.

Adapun manfaat dari penelitian ini adalah:

- Mengetahui apa itu steganografi lebih mendalam khususnya dengan menggunakan *software camouflage*.
- *Software Camouflage* digunakan sebagai pengamanan data pada jaringan komputer.

2. TINJAUAN PUSTAKA

Dalam penelitian ini dijabarkan mengenai tinjauan pustaka yang menjadi dasar dilakukannya penelitian ini. Dimana seluruh tinjauan pustaka mencakup semua hal terkait dengan penelitian yang dilakukan.

2.1. Pengertian Steganografi

Steganografi adalah suatu teknik untuk menyembunyikan informasi yang bersifat pribadi dengan sesuatu yang hasilnya akan tampak seperti informasi normal lainnya. Steganografi biasanya sering disalahkaprahkan dengan kriptografi karenanya keduanya sama-sama bertujuan untuk melindungi informasi yang berharga. Perbedaan yang mendasar antara keduanya yaitu steganografi berhubungan dengan informasi tersembunyi sehingga tampak seperti tidak ada informasi

tersembunyi sama sekali. Jika seseorang mengamati objek yang menyimpan informasi tersembunyi tersebut, ia tidak akan menyangka bahwa terdapat pesan rahasia dalam objek tersebut, dan karenanya ia tidak akan berusaha memecahkan informasi (dekripsi) dari objek tersebut

Kata steganografi berasal dari bahasa Yunani, yaitu dari kata *SteganOs* (tersembunyi) dan *Graptos* (tulisan). Steganografi di dunia modern biasanya mengacu pada informasi atau suatu arsip yang telah disembunyikan ke dalam suatu arsip citra *digital,* audio, atau video. Satu hal esensial yang menjadi kelebihan steganografi adalah kemampuannya untuk menipu persepsi manusia, manusia tidak memiliki insting untuk mencurigai adanya arsip-arsip yang memiliki informasi yang tersembunyi di dalamnya, terutama bila arsip tersebut tampak seperti arsip normal lainnya. Namun begitu terbentuk pula suatu teknik yang dikenal dengan *steganalysis,* yaitu suatu teknik yang digunakan untuk mendeteksi penggunaan steganografi pada suatu arsip.

2.2. Sejarah Steganografi

Seperti kriptografi, penggunaan steganografi sebetulnya telah digunakan berabad-abad yang lalu bahkan sebelum istilah steganografi itu sendiri muncul. Berikut adalah contoh penggunaan steganografi di masa lalu:

1. Selama terjadinya Perang Dunia ke-2, tinta yang tidak tampak *(invisible ink)* telah digunakan untuk menulis informasi pada lembaran kertas sehingga saat kertas tersebut jatuh di tangan pihak lain hanya akan tampak seperti lembaran kertas kosong biasa. Cairan seperti air kencing *(urine),* susu, vinegar, dan jus buah digunakan sebagai media penulisan sebab bila salah satu elemen tersebut dipanaskan, tulisan akan menggelap dan tampak melalui mata manusia.
2. Pada sejarah Yunani kuno, masyarakatnya biasa menggunakan seorang pembawa pesan sebagai perantara pengiriman pesan. Pengirim pesan tersebut akan dicukur rambutnya, untuk kemudian dituliskan suatu pesan pada kepalanya yang sudah botak. Setelah pesan dituliskan, pembawa pesan harus menunggu hingga rambutnya tumbuh kembali sebelum dapat mengirimkan pesan kepada pihak penerima. Pihak penerima kemudian akan mencukur rambut pembawa pesan tersebut untuk melihat pesan yang tersembunyi.
3. Metode lain yang digunakan oleh masyarakat Yunani kuno adalah dengan menggunakan lilin sebagai media penyembunyi pesan mereka. Pesan dituliskan pada suatu lembaran, dan lembaran tersebut akan ditutup dengan lilin untuk menyembunyikan pesan yang telah tertulis. Pihak penerima kemudian akan menghilangkan lilin dari lembaran tersebut untuk melihat pesan yang disampaikan oleh pihak pengirim.

2.3. Kegunaan Steganografi

Seperti perangkat keamanan lainnya, steganografi dapat digunakan untuk herbagai macam alasan, beberapa diantaranya untuk alasan yang baik, namun dapat juga untuk alasan yang tidak baik. Untuk tujuan legitimasi dapat digunakan pengamanan seperti citra dengan *watermarking* dengan alasan untuk perlindungan *copyright. Digital watermark* (yang juga dikenal dengan *fingerprinting,* yang dikhususkan untuk hal-hal menyangkut *copyright)* sangat mirip dengan steganografi karena menggunakan metode penyembunyian dalam arisp, yang muncul sebagai bagian asli dari arsip tersebut dan tidak mudah dideteksi oleh kebanyakan orang.

Steganografi juga dapat digunakan sebagai cara untuk membuat pengganti suatu nilai *hash* satu arah (yaitu pengguna mengambil suatu masukan panjang variabel dan membuat sebuah keluaran panjang statis dengan tipe string untuk melakukan verifikasi bahwa tidak ada perubahan yang dibuat pada variabel masukan yang asli). Selain itu juga, steganografi dapat digunakan sebagai *tag- notes* untuk citra *online.*

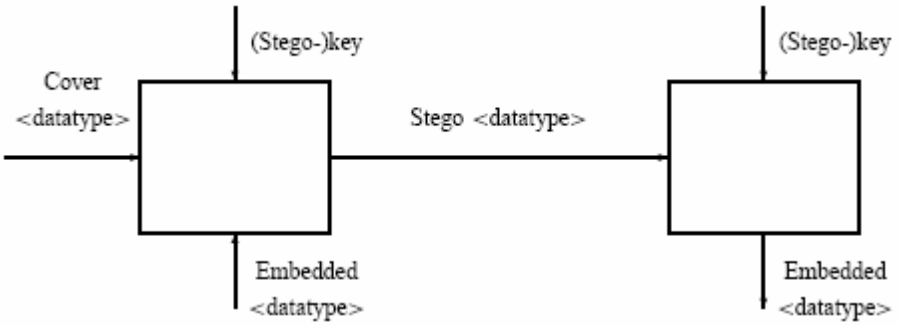
Terakhir, steganografi juga dapat digunakan untuk melakukan perawatan atas kerahasiaan informasi yang berharga, untuk menjaga data tersebut dari kemungkinan sabotasi, pencuri, atau dari pihak yang tidak berwenang. Sayangnya, steganografi juga dapat digunakan untuk alasan yang ilegal. Sebagai contoh, jika seseorang telah mencuri data, mereka dapat menyembunyikan arsip curian tersebut ke dalam arsip lain dan mengirimkannya keluar tanpa menimbulkan kecurigaan siapapun karena tampak seperti *email* atau arsip normal. Selain itu, seseorang dengan hobi menyimpan pornografi, atau lebih parah lagi, menyimpannya dalam *hard disk*, mereka dapat menyembunyikan hobi buruk mereka tersebut melalui steganografi. Begitu pula dengan masalah terorisme, steganografi dapat digunakan oleh para teroris untuk menyamarkan komunikasi mereka dari pihak luar.

2.4. Metode Steganografi

Terdapat banyak metode-metode yang sering digunakan dalam melakukan penyembunyian data ke dalam data lainnya. Berikut adalah penjelasan mengenai beberapa metode-metode yang banyak digunakan dalam steganografi.

2.4.1. Metode *Embedding*

Steganografi menyimpan pesan rahasia dalam suatu arsip yang biasanya diparameterisasi oleh suatu kunci-stego. dan pendeteksian atau pembacaan atas informasi tersembunyi tersebut dapat dilihat pada gambar di bawah ini. **Metode** *embedding* ini juga biasa disebut dengan metode *injection* karena pesan rahasia "disuntikkan" langsung pada arsip lainnya dengan sedikit pengacakan atau enkripsi.



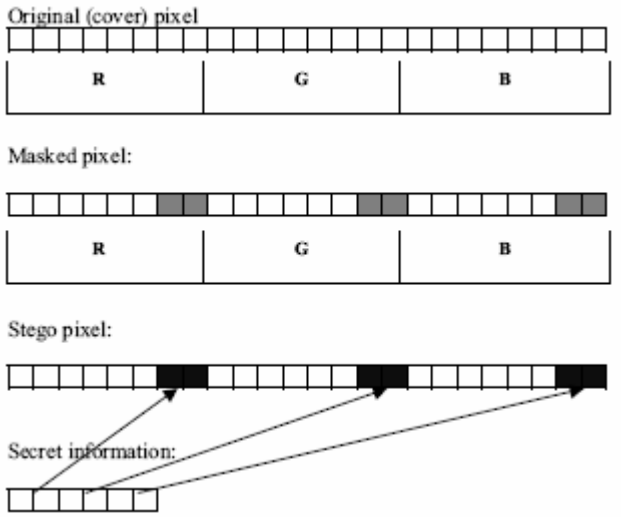
Gambar Proses steganografi dengan metode *embedding*

2.4.2. Metode *Least Significant Bit (LSB)*

Biasanya arsip 24-bit atau 8-bit digunakan untuk menyimpan citra *digital*. Representasi warna dari pixel-pixel dapat diperoleh dari warna-warna primer yaitu merah, hijau, dan biru. Citra 24-bit menggunakan 3 bytes untuk masing-masing pixel, dimana setiap warna primer direpresentasikan dengan ukuran 1 byte. Penggunaan citra 24-bit memungkinkan setiap pixel direpresentasikan dengan nilai warna sebanyak 16.777.216 macam. Dua bit dari saluran warna ini dapat digunakan untuk menyembunyikan data, yang akan mengubah jenis warna untuk pixelnya menjadi 64-

warna, namun hal ini akan mengakibatkan sedikit perbedaan yang dapat dideteksi secara kasat mata oleh manusia. Metode sederhana ini disebut dengan *Least Significant Bit* (LSB).

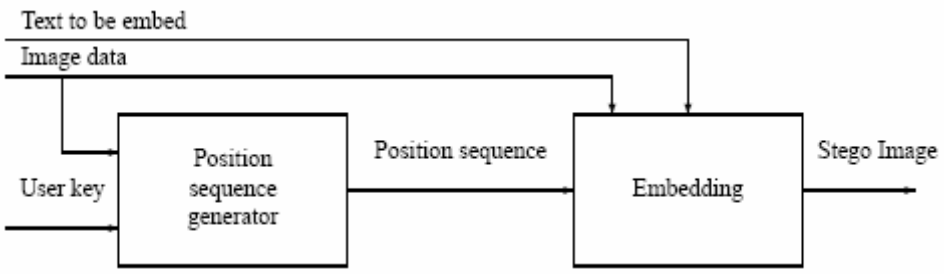
Dengan penggunaan metode ini, dimungkinkan adanya penambahan sejumlah besar informasi tanpa adanya degradasi tampilan dari citra itu sendiri. Gambar di bawah ini menunjukkan proses kerja LSB.



Gambar Proses Steganografi dengan Metode Least Significant Bit (LSB)

Beberapa versi dari metode LSB telah bermunculan. Kini sangat memungkinkan untuk menggunakan menggunakan pembangkit nomer acak yang diinisialisasi dengan kunci-stego dan mengkombinasikan keluarannya dengan data masukan, dan kemudian menyembunyikannya dalam suatu arsip citra.

Kehadiran seorang pengawas tidak cukup untuk meyakinkan keberhasilan penambahan sebuah pesan di lokasi tertentu (pada rentetan bit tertentu), karena pengawas itu sendiri sangat mungkin mengubah letak dari pesan rahasia tersebut, bahkan walaupun ia tidak mengetahui lokasi dari pesan rahasia tersebut atau ia tidak dapat membacanya karena telah dienkripsi. Karena itulah penggunaan kuncistego menjadi penting karena keamanan atas suatu system proteksi tidak dapat didasarkan pada kerahasiaan dari algoritmanya itu sendiri, namun karena adanya keberadaan dari suatu kunci rahasia. Gambar di bawah ini menunjukkan proses tersebut.



Gambar Proses Penggunaan Kunci Stego pada Steganografi

Metode LSB pada umumnya beroperasi pada citra *bitmap*. Data yang disembunyikan tidak dapat dikategorikan sebagai *watermark* karena bahkan jika terjadi perubahan kecil pada citra tersebut (pemotongan, kompresi, atau degradasi warna) maka informasi tersembunyi tersebut akan hilang walaupun perubahan yang terjadi selama proses *embedding* adalah tidak terlihat.

2.4.3. Metode Enkripsi Data

Pada dasarnya metode enkripsi ini dapat dikelompokkan kedalam dua macam *chiper* yaitu :

1. Chiper Subtitusi
   Dalam *chiper* subtitusi setiap *plaintext* akan diganti dengan satu unit *chipertext.* Satu "unit" disini bisa berarti satu huruf, pasangan huruf atau kelompok lebih dari dua huruf. Contoh *chiper* subtitusi adalah *caesar chiper* yang tiap hurufnya akan disubtitusikan dengan huruf ketiga berikutnya dari susunan alfabet yang sama. Dalam hal ini *key* yang digunakan adalah jumlah penggeseran huruf yaitu 3 huruf dari depan akan di geser kebelakang.
2. Chiper Transposisi
   Pada *chiper* transposisi huruf-huuruf didalam *plaintext* tetap sama, hanya saja urutannya yang diubah. Dengan kata lain algoritma ini melakukan *transpose* terhadap rangkaian karakter didalam *text.*

2.4.4. Metode Domain Transformasi

Algoritma ekstraksi tujuan dapat dibagi menjadi 2 grup, yaitu teknik domain ruang/waktu dan domain transformasi. Untuk kasus domain ruang diterapkan pada materi citra, dan untuk domain waktu diterapkan pada materi audio. Metode domain transformasi dioperasikan pada *Discrete Cosine Transform, Fourier* atau domain transformasi dari sinyal. Algoritma *Patchwork* (yang dikembangkan di MIT) memilih sejumlah pasang pixel acak, dan meningkatkan *brightness* dari pixel yang terang, dan menurunkan *britghtness* tersebut untuk pixel yang gelap. Algoritma ini menunjukkan ketahanan tinggi atas semua modifikasi citra nirgeometrik. Jika dianggap penting untuk menyediakan proteksi atas serangan penyaringan, maka kapasitas penyimpanan informasi tersebut dapat dibatasi.

Citra dengan kualitas *high-color* yang dikompres biasanya akan menggunakan metode *lossy compression*, sebagai contoh untuk citra JPEG. Algoritma JPEG akan pertama-tama melakukan transformasi pixel menjadi ruang *luminance chrominance. Chrominance* ini lalu *di-downsampled* hal ini mungkin karena HVS *(human vision system)* jauh kurang sensitif atas *chrominance* dibandingkan dengan perubahan *luminance* sehingga ukuran dari data menjadi berkurang. *Discrete Cosine Transform* lalu diaplikasikan pada kelompok pixel dengan ukuran 8 x 8.

Aplikasi steganografi pada umumnya beroperasi setelah langkah perhitungan, sebagai contoh adalah aplikasi Jpeg-Jste, dan SysCoP. SysCoP menggunakan generator rentetan posisi. Masukan dari generator tersebut adalah data citra dan sebuah kunci eksternal, dan keluaran yang dihasilkan adalah suatu rentetan posisi yang dapat dipilih untuk menentukan posisi blok tempat pesan rahasia disembunyikan.

Pada kasus ini, bloknya terdiri dari pixel dengan ukuran 8 x 8, yang dapat *contigous* blok adalah suatu kotak dalam citra atau terdistriubsi, dimana pixel akan dipilih secara acak. Sebuah bit label akan ditambahkan melalui pengaturan atas hubungan

spesifik antara ketiga elemen kuantitas dari suatu blok, dan algoritma yang mengandung suatu mekanisme pengecekan untuk mengetes apakah blok yang asli mampu atau tidak untuk menyimpan informasi tersebut, dan berapa banyak modifikasi yang dibutuhkan untuk menyimpan 1 bit informasi diantara pixel-pixelnya.

Metode popular pada domain frekuensi yaitu dengan memodifikasi ukuran relative dari 2 atau lebih koefesien DCT pada blok citra, dan menambahkan 1 bit informasi pada tiap blok. Kedua koefisien tersebut harus berkorespondensi dengan fungsi cosine dan dengan frekuensi tengah yang berarti informasi disimpan pada suatu bagian signifikan dari suatu sinyal. Algoritma yang digunakan harus kebal terhadap kompresi JPEG, sehingga koefisien DCT dengan nilai kuantitas yang sama harus dipilih, sesuai dengan kuantitas tabel dari JPEG.

Pada domain frekuensi, proses penambahan informasi biasanya mampu menyimpan lebih sedikit informasi ke dalam citra, tidak ada batas yang pasti atas ukuran dari objek yang ditambahkan seperti pada kasus LSB, dimana jumlah dari pixel dan kedalaman warna ditentukan oleh ukuran maksimum dari data tambahan tersebut (dan tentunya perubahan yang terjadi selama penambahan informasi akan tidak terlihat). Pada kasus operasi domain transformasi, proses penambahan informasi dapat tampak jika ukuran data yang ditambahkan terlalu besar, dan batas yang diberikan atas ukuran data tambahan yang tidak akan mengubah properti visual dari citra tersebut adalah *image dependent*. Berikut adalah gambar yang menunjukkan hasil dari proses *embedding* pada domain transformasi.

Gambar Citra 30 KB dengan data tersembunyi bernilai "jhps"

Gambar Citra 50 KB dengan data tersembunyi bernilai "jhps"

Gambar Citra 60 KB dengan data tersembunyi bernilai "jhps"

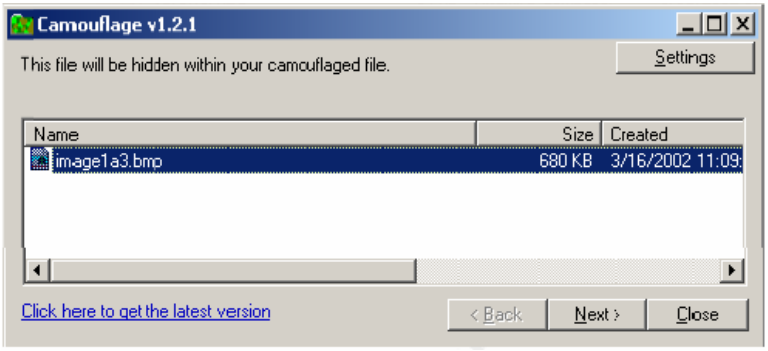
Pada kasus diatas jika gambar jam dengan ukuran 50 KB yang berisi data kemudian memodifikasi property *visible* dari citra, maka saat citra-stego dibandingkan dengan citra asli akan memungkinkan untuk dideteksi adanya perbedaan.

3. SOFTWARE CAMOUFLAGE

Pada zaman sekarang ini, perusahaan-perusahaan telah diberi kuasa lebih untuk memonitor dan memeriksa arsip-arsip personal pegawainya. Dan dengan semakin menjamurnya perangkat lunak *spy* dengan tujuan tidak balk, pengguna komputer semakin membutuhkan keamanan untuk menjaga arsip-arsip yang mengandung informasi sensitif jauh dari pihak-pihak yang tidak berkepentingan. Keamanan elektronik juga sudah tidak dapat lagi dijamin siapa yang bisa mengetahui apabila terdapat pihak yang mengintai *email* atau memindai *hard drive* tanpa sepengetahuan orang yang bersangkutan? Alasan tersebutlah yang mendorong terbentuknya *software Camouflage* ini. *Software Camouflage* memungkinkan pengguna komputer untuk menyembunyikan arsip dengan mengacaknya dan melampirkannya ke dalam arsip lain.

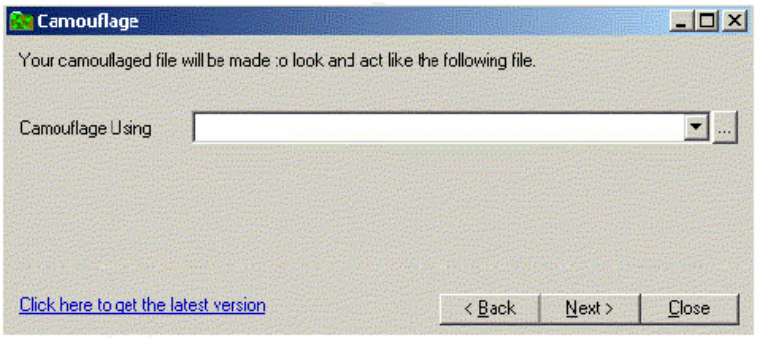
3.1. *Camouflaging* terhadap Arsip

Software Camouflage sangat mudah digunakan, hanya dengan meng-klik [l]kanan arsip dan memilih opsi *'Camouflage'* . Berikut adalah layar *Camouflage* yang muncul sesaat setelah pemilihan arsip untuk *di-camouflage.*



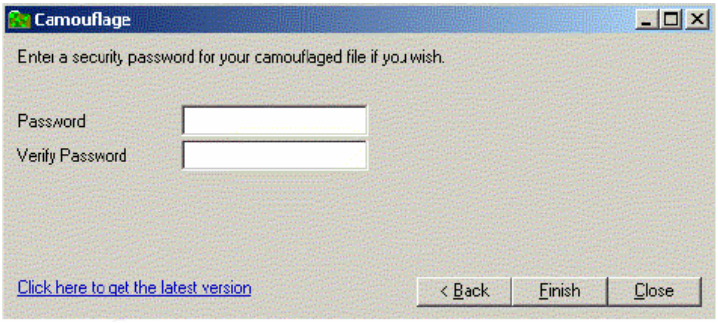
Gambar Layar deskripsi mengenai arsip yang akan di-*camouflage*

Citra yang dipilih “image1a3.bmp” yaitu citra yang akan di-*camouflage* ke dalam arsip lain yang akan ditentukan pada langkah selanjutnya. Ingat bahwa ukuran arsip citra yang asli yaitu 680 KB. Dengan menekan tombol ‘Next>’ akan memunculkan layar di bawah ini, yaitu layar untuk memilih arsip yang akan dilekatkan dengan arsip citra.



Gambar Layar untuk memilih arsip yang akan dilekatkan pada arsip yang akan di-*camouflage*

Layar akhir yang akan muncul dalam proses *camouflage* ini adalah cara lain untuk meyakinkan bahwa arsip tersebut aman. *Password* yang disimpan dalam arsip adalah opsional namun dibutuhkan untuk membuka arsip jika sebelumnya *password* digunakan untuk membuat arsip yang telah di-*camouflage*. Kemampuan untuk menambahkan *password* pada arsip yang di-*camouflage* minimal akan menambah banyak pekerjaan bagi seseorang yang memiliki program *Camouflage* namun lupa atau tidak memiliki *password* tersebut. Gambar 4 adalah kotak dialog *password* yang merepresentasikan proses finalisasi pembuatan arsip *camouflage*. Opsi ini akan selalu muncul namun *password* tidak harus diterapkan pada arsip tersebut.



Gambar Layar akhir proses *camouflage*

Untuk melengkapi proses *camouflage*, tekan tombol 'Finish' dan sebuah arsip baru telah diciptakan. Keseluruhan proses *camouflage* ini memakan tidak lebih dari 1 menit. Arsip citra atau teks tersebut tidak memiliki perbedaan dengan yang asli. Jika suatu citra di-*camouflage* menjadi suatu dokumen MS Word, maka saat arsip tersebut dibuka akan muncul dan berlaku seperti arsip Word biasa. Kemampuan ini berlaku untuk hamper keseluruhan tipe arsip. Namun tentunya hal ini sedikit menimbulkan kecurigaan apakah arsip asli telah berhasil di-*camouflage* ke arsip yang baru. Untuk mengatasi hal tersebut, sebaiknya terdapat aspek lain dari arsip yang bertujuan untuk mengalamatkan bahwa arsip tersebut telah melalui proses *camouflage*.

3.2. *Uncamouflaging* terhadap Arsip

Semudah proses untuk melakukan *camouflage* terhadap suatu arsip, langkah untuk melakukan *uncamouflage* bahkan dilakukan hanya dalam 1 langkah. Untuk *uncamouflaging* arsip, cukup klik kanan pada arsip yang bersangkutan dan **pilih** opsi *Uncamouflage*.

Saat memilih opsi *uncamouflage*, jika arsip tersebut mengandung *password*, maka *password* perlu diketikkan untuk menyelesaikan proses. Pada arsip ini, sebuah arsip teks 1 KB di-*camouflage* menjadi arsip citra 678 KB (image1a3.bmp) dan arsip citra kemudian di-*camouflage* kembali dalam arsip citra 679 KB (image1a.bmp). Hasil dari *camouflage* akan meningkatkan ukuran arsip, yang pada poin tertentu dapat menentukan apakah suatu arsip telah di-*camouflage*.

3.3. Kekurangan pada *Camouflage*

Software Camouflage memiliki proses yang sangat buruk saat dilakukannya proses *uninstall* dan meninggalkan suatu *fingerprint* yang sangat jelas. Setelah proses *uninstall* selesai, suatu pencarian pada *registry* akan membuka sejumlah kehadiran informasi yang tidak dibuang. Yang lebih penting, catatan yang tertinggal tersebut masih mengandung data yang mengindikasi bahwa arsip telah digunakan dengan program *Camouflage*.

Walaupun arsip tidak dapat di- *camouflage* menggunakan program, namun proses *install* ulang yang sederhana akan memecahkan masalah tersebut. Pengaturan *registry*, data gabungan yang menghasilkan arsip dengan ukuran yang lebih besar, dan data terenkripsi yang berada pada arsip normal membuat *software Camouflage* sangat mudah untuk dideteksi.

4. ANALISA PROGRAM

Software Camouflage digunakan untuk menyimpan pesan rahasia dalam suatu arsip yang lain dan *software camouflage* ini jugs mensuport berbagai macam format *file* salah satunya format .bmp. Format tersebut biasanya menggunakan metode LSB *(Least Significant Bit),* metode yang biasanya diparameterisasi oleh suatu kunci-stego kedalam arsip lainnya dengan sedikit pengacakan atau enkripsi.

Metode LSB *(Least Significant Bit)* merupakan metode yang paling sederhana dan paling mudah diimplementasikan. Sebagai contoh untuk metode ini menggunakan cover yang berupa image sebagai tempat untuk menyisipkan pesan dan besar kecilnya pesan yang disisipkan ini tergantung dari tempat **untuk** menyisipkan pesan tersebut. Model dari penyisipan metode LSB ini terletak pada akhir bit sebab modifikasi hanya mengubah nilai *byte* tersebut satu lebih tinnggi atau satu lebih rendah dari nilai sebelumnya.

Cover yang digunakan untuk penyisipan misalnya memiliki ukuran antara 8 bit atau 24 hit. Untuk menyembunyikan suatu gambar dalam setiap byte dari gambar 24-bit, dapat disimpan 3 byte dalam setiap pixel (atau 1 by komponen Red, Green, Blue). Misalnya gambar dengan ukuran 1,024 x 768 mempunyai potensi untuk disembunyikan seluruhnya dari 2,359,296 bit (294,912 byte) pada informasi te untuk setiap.

Pada gambar 8-bit tidak diberikan untuk manipulasi LSB karena keterbatasan warnanya. Gambar cover harus lebih hati-hati diseleksi sehingga tidak akan *mem-broadcast* keberadaannya pada pesan yang ditempelkan. Ketika informasi disisipkan ke dalam LSB dari *raster data,* penunj uk kemasukan warna dalam palette yang diubah. Dalam suatu contoh, suatu palette sederhana empat warna dari putih, merah, biru dan hijau mempunyai posisi masukan palette yang sesuai secara berturut-turut dari 0 (00), 1 (01), 2 (10), dan 3 (11). Nilai *raster* dari empat pixel yang bersebelahan dari putih, putih, biru dan biru adalah 00 00 10 10. Penyembunyian nilai biner 1010 untuk perubahan bilangan 10 *raster data* ke 01 00 11 10, adalah merah, putih, hijau dan biru.

Untuk membbuat *hiddentext* tidak dapat dilacak maka bit-bit pesan tidak menggunakan *byte-byte* yang berurutan, namun bisa dipilih susunan bytenya secara acak. Misalnya jika terdapat 50 *byte* dan 10 bit data yang disembunyikan, maka *byte* yang diganti bit LSB-nya dapat dipilih secara acak misalnya 28, 11, 26, 4, 13, 6, 49, 30, 36,9. pembangkit bilangan acak seperti LCG dapat digunakan sebagi *pseudo-random-number-generator* (PRNG). Dalam hal ini nilai umpan untuk LCG berlaku sebagai kunci stegano.

5. KESIMPULAN DAN SARAN

Dari seluruh bahasan di atas maka dapat ditarik beberapa kesimpulan. Antara lain yaitu bahwa steganografi adalah suatu teknik untuk menyembunyikan informasi yang bersifat pribadi dengan sesuatu yang hasilnya akan tampak seperti informasi normal lainnya. Terdapat banyak metode yang dapat digunakan dalam mengaplikasikan steganografi, antara lain yaitu metode LSB (*least significant byte), spread spectrum*, kunci publik steganografi, domain transformasi, dan *embedding/injection.*

Salah satu dari aplikasi steganografi tersebut adalah *software Camouflage*. *Software Camouflage* memungkinkan pengguna komputer untuk menyembunyikan arsip dengan mengacaknya dan melampirkannya ke dalam arsip lain.

Arsip yang telah di-*camouflage* akan tampak dan berlaku seperti arsip normal lainnya, dan dapat disimpan ataupun dikirimkan sebagai email tanpa menarik perhatian. Selain itu *software Camouflage* juga memungkinkan pengguna untuk memberi password atas arsip yang di-camouflage tersebut. *Password* nantinya akan dibutuhkan kembali saat melakukan ekstraksi arsip tersebut.

Namun sayangnya, metode yang digunakan oleh *Camouflage* dalam menyembunyikan arsip ke dalam arsip lainnya sangat lemah dan mudah dipecahkan. Hanya dengan membandingkan arsip asli dengan arsip yang telah di-camouflage maka seseorang dapat dengan mudah mencari tahu password yang digunakan untuk uncamouflaging arsip tersebut. Dari kenyataan tersebut, tentunya *software Camouflage* tidak dapat lagi dinyatakan sebagai suatu aplikasi steganografi yang aman dan tidak dapat dideteksi.

Dan dari langkah pemecahan algoritma yang dilakukan terhadap aplikasi *Camouflage*, dapat diketahui bahwa *Camouflage* menyisipkan informasi rahasia pada akhir dari arsip lainnya. Maka dapat disimpulkan bahwa metode yang digunakan oleh *Camouflage* dalam menyembunyikan informasi adalah dengan metode *embedding/injection*.

Saran yang dapat diberikan untuk kelanjutan penelitian ini adalah dengan menambahkan metode-metode yang lain dan menggunakan *software camouflage* yang dibuat sendiri.

6. DAFTAR PUSTAKA

Anderson, R. J. — Petitcolas, F. A. P., *On The Limits of Steganography, IEEE Journal of*

Bartlett, John. The *Easy of Steganography and Camouflage GSEC V1.3*. Sans Institute, March 17th 2002.

Lenti, Jozsef. *Steganographic Methods*. Department of Control Engineering and Information Technology. Budapest University of Technology and Economics, June 5th 2000

Smith, J. R. – Comiskey, B. O., fjrs, *Modulation and Information Hiding in Images*, elwoodg@media.mit.edu Physics and Media GroupMIT Media Lab 20 Ames Street Cambridge, MA 02139 USA *Proceedings of the First Information Hiding Workshop*, Isaac Newton Institute, Cambridge, U.K., May 1996. Springer-Verlag *Lecture Notes in Computer Science* Volume 1174.