PENGAMANAN PESAN STEGANOGRAFI DENGAN METODE LSB BERLAPIS ENKRIPSI

Tri Ismardiko Widyawan
Fakultas Ilmu Komputer, Universitas Esa Unggul
Jalan Arjuna Utara No.9, Kebon Jeruk, Jakarta 11510
ismardiko@esaunggul.ac.id

Abstracts

Security in communication is a top priority in the use of steganography. This concept allows us to convey messages to people we want through a media without arousing suspicion to others. In this case the media used is digital media. More specifically digital media used are image media because images are the most popular media in secret message hiding. The increasingly widespread use of the internet as a communication medium is the reason why this message insertion technique uses the PHP programming language that is currently popular as a programming language supporting web-based applications. The LSB (Least Significant Byte) method is a message hiding technique in steganography where the hidden message is hidden by replacing the data bits in the image segment with secret message bits. Secret message bits are inserted in the low bit or the rightmost bit in the pixel composing the image consisting of red, green and blue (RGB), each of which has an 8-bit value of 0 to 255 with binary format 00000000 to 11111111. Thus, 3 bits of data can be inserted in each pixel image available. The LSB method is then applied in a library which is then named as a stegger which has the function to insert messages in images using the PHP programming language. To add protection to messages sent, an encryption algorithm is added that has been arranged in a class named secrypt that is accompanied by keyword inclusion so that the message sent has a higher level of security.

Keywords*: steganography, PHP, encryption, stegger, secrypt*

Abstraksi

Keamanan dalam berkomunikasi merupakan prioritas utama dalam penggunaan steganografi. Konsep ini memungkinkan kita untuk menyampaikan pesan kepada orang-orang yang kita inginkan melalui suatu media tanpa menimbulkan kecurigaan kepada orang lain. Dalam hal ini media yang digunakan adalah media digital. Secara lebih spesifik media digital yang dipakai adalah media gambar karena gambar adalah media yang paling populer dalam penyembunyian pesan secara rahasia. Makin maraknya penggunaan internet sebagai media komunikasi adalah alasan mengapa teknik penyisipan pesan ini menggunakan bahasa pemrograman PHP yang saat ini populer sebagai bahasa pemrograman pendukung aplikasi berbasis web. Metode LSB (Least Significant Byte) merupakan teknik penyembunyian pesan dalam steganografi dimana penyembunyian pesan rahasia dilakukan dengan mengganti bit-bit data dalam segmen gambar dengan bit-bit pesan rahasia. Bit-bit pesan rahasia disisipkan pada bit rendah atau bit paling kanan dalam pixel penyusun gambar yang terdiri dari warna merah, hijau dan biru (RGB) yang masing masing memiliki nilai 8 bit bernilai 0 sampai dengan 255 dengan format biner 00000000 sampai dengan 11111111. Dengan demikian, 3 bit data dapat disisipkan pada tiap-tiap pixel gambar yang tersedia. Metode LSB kemudian diterapkan dalam sebuah library yang kemudian dinamakan sebagai stegger yang memiliki fungsi untuk menyisipkan pesan dalam gambar dengan menggunakan bahasa pemrograman PHP. Untuk menambahkan proteksi pada pesan yang dikirim maka ditambahkan sebuah algoritma enkripsi yang telah disusun dalam sebuah class bernama secrypt yang disertai penyertaan kata kunci agar pesan yang dikirim tersebut memiliki tingkat keamanan yang lebih tinggi.

Kata kunci: *steganografi, PHP, enkripsi, stegger, secrypt*

Pendahuluan

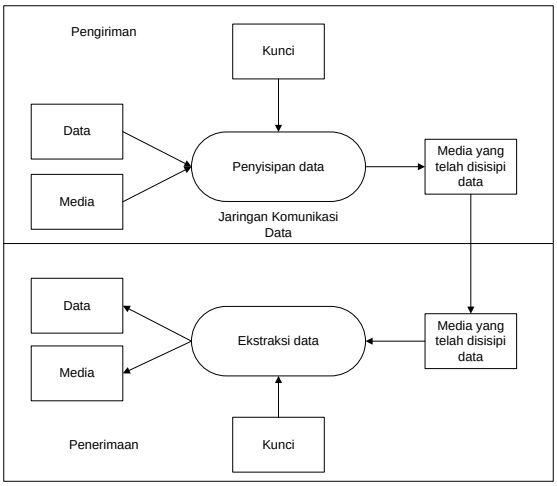
Pada dasarnya manusia diciptakan untuk saling berinteraksi satu sama lain, dan salah satu cara berinteraksi adalah dengan melakukan pertukaran informasi. Awalnya manusia saling bertukar informasi dengan berbicara secara langsung, kemudian menggunakan media tulisan berupa surat. Seiring berkembangnya media maka dikenallah media gambar. Dengan gambar sebuah informasi dapat disajikan dengan mudah dan memiliki kesan estetik.

Seiring perkembangan zaman, saat ini manusia memasuki era *internet*, dimana perkembangan pertukaran informasi semakin berkembang pesat. Kini informasi berupa pesan dan gambar dapat dikirim melalui *e-mail*. Sebagian besar data yang menyimpan informasi bagi tiap individu yang dipertukarkan membutuhkan pengamanan untuk menjaga integritas pesan tersebut agar aman dan tidak dapat disalahgunakan oleh pihak yang tidak berhak mengetahui informasi dari data tersebut.

Terdapat beberapa cara untuk mengamankan data yaitu dengan cara kriptografi dan steganografi yang dapat diimplementasikan dalam sebuah aplikasi untuk menyamarkan pesan.

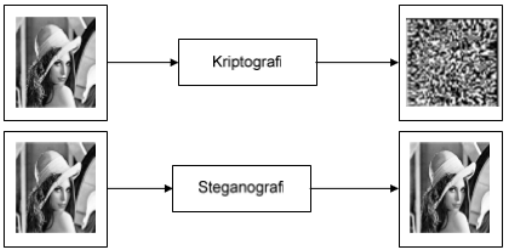
Konsep Steganografi

Metode Steganografi sedemikian rupa dalam menyembunyikan isi suatu data di dalam suatu sampul mediaatau data digital lain yang tidak dapat diduga oleh orang biasa sehingga tidak menimbulkan kecurigaan kepada orang yang melihatnya. Gambar 2 adalah ilustrasi dasar dari konsep Steganografi.



Gambar 1
Ilustrasi Dasar Konsep Steganografi

Steganografi adalah seni dan ilmu menulis pesan tersembunyi dengan suatu cara sehingga selain si penerima yang dimaksud tak ada satupun orang yang mengetahui atau menyadari bahwa suatu pesam tersimpan. Sebaliknya, kriptografi adalah menyamarkan arti suatu pesan tanpa menyembunyikan keberadaanya dan membuat siapapun menyadari bahwa ada sesuatu yang mencurigakan dari pesan tersebut. Kelebihan steganografi dibanding kriptografi adalah pesan-pesannya tidak menarik perhatian orang lain. Namun pada saat ini seringkali kita temukan penggunaan steganografi dan kriptografi secara bersamaan untuk menjamin kerahasiaan sebuah pesan.



Gambar 2
Ilustrasi Dasar Konsep Steganografi

Pada dasarnya citra digital (diskrit) dihasilkan dari citra analog (kontinu) melalui digitalisasi. Digitalisasi citra analog terdiri atas *sampling* dan kuantisasi (*quantization*). *Sampling* adalah pembagian citra ke dalam elemen-elemen diskrit (piksel), sedangkan kuantisasi adalah pemberian nilai intensitas warna pada setiap piksel dengan nilai yang berupa bilangan bulat[1]. Dan citra tersebut adalah citra biner, citra grayscale dan citra berwarna.

Adapun jenis-jenis gambar yang dapat disisipi pesan dalam steganografi adalah sebagai berikut :

a. JPG / JPEG (*Joint Photographic Experts Assemble*)

JPG adalah jenis data yang dikembangkan oleh Joint Photographic Experts Assemble (JPEG) yang dijadikan standar untuk para fotografer profesional

b. GIF (*Graphics Interchange Format*)

GIF, sama seperti JPG, adalah format gambar yang sudah cukup lama digunakan dan salah satu yang umum dipakai di internet. GIF adalah kepanjangan dari Graphics Interchange Format. GIF secara alami adalah gambar dengan 8-bit warna, berarti mereka dibatasi oleh palet sebanyak 256 jenis warna, yang dapat dipilih dari model RGB dan disimpan ke *Color Look Up Tablet* (CLUT), atau sederhananya "*Color Table*". Mereka itu sejatinya adalah palet warna standar, seperti palet "*Web Safe*".Selain bisa transparansi, GIF juga mendukung animasi gambar yang membatasi tiap form nya pada 256 warna standar. Dan karena sifatnya yang tidak pecah-pecah, GIF bisa digunakan untuk menjaga baris dalam tipografi tetap rapi, dan juga bentuk-bentuk geometri.

c. PNG (*Portable Network Graphics*)

PNG adalah kepanjangan dari Portable Network Graphics. Dikembangkan sebagai alternatif lain untuk GIF, yang menggunakan paten dari LZW–algoritma kompresi. PNG adalah format gambar yang sangat baik untuk grafis internet, karena mendukung transparansi didalam perambah (browser) dan memiliki keindahan tersendiri yang tidak bisa diberikan GIF atau bahkan JPG. Bisa disebut sebagai salah satu format yang merupakan gabungan dari format JPG dan GIF. Untuk tipe ini mampu untuk gradiasi warna. Tipe file PNG merupakan solusi kompresi yang powerful dengan warna yang lebih banyak (24 bit RGB + alpha). Berbeda dengan JPG yang menggunakan teknik kompresi yang menghilangkan data, file PNG menggunakan kompresi yang tidak menghilangkan data (lossles compression). Kelebihan file PNG adalah adanya warna transparan dan alpha. Warna alpha memungkinkan sebuah gambar transparan, tetapi gambar tersebut masih dapat dilihat mata seperti samar-samar atau bening.

d. BMP (Bitmap)

Bitmap adalah representasi dari citra grafis yang terdiri dari susunan titik (pixel) yang tersimpan di memori komputer. Nilai setiap titik diawali oleh satu bit data (untuk gambar hitam putih) atau lebih (untuk gambar berwarna). Kerapatan titik-titik tersebut dinamakan resolusi, yang menunjukkan seberapa tajam gambar ini ditampilkan, ditunjukkan dengan jumlah baris dan kolom (contoh 1024×768).

e. TIFF (*Tagged Image Format File*)

TIFF merupakan format gambar terbaik dengan pengertian bahwa semua data dan informasi (data RGB, data CMYK, dan lainnya) yang berkaitan dengan koreksi atau manipulasi terhadap gambar tersebut tidak hilang. Format TIFF biasa digunakan untuk kebutuhan pencetakan dengan kualitas gambar yang sangat tinggi sehingga ukuran berkas untuk format ini biasanya sangat besar, karena dalam file ini gambar tidak dikompresi. Format ini mampu menyimpan gambar dengan kualitas hingga 32 bit. Format berkas TIFF juga dapat digunakan untuk keperluan pertukaran antar platform (PC, Macintosh, dan Silicom Graphic). Format ini juga mudah digunakan untuk transfer antar program.

Sebuah pesan yang akan dikirimkan diubah terlebih dahulu menjadi kode biner dan dimasukkan ke dalam kode biner data lain yang menjadi media atau sampulnya. Lalu kedua kode biner tersebut dikodekan sehingga menjadi satu kesatuan tanpa mengubah integritas media yang ditumpangi. Selanjutnya data tersebut dikirimkan dan diterima oleh si penerima pesan. Penerima pesan lalu mengkodekan kembali pesan tersebut sehingga pesan bisa dibaca.

Sebagai contoh, pengirim pesan mulai dengan berkas citra biasa, lalu mengatur warna setiap piksel ke-50 untuk menyesuaikan suatu huruf dalam alphabet (perubahannya begitu halus sehingga tidak ada seorangpun yang menyadarinya jika ia tidak benar-benar memperhatikannya).

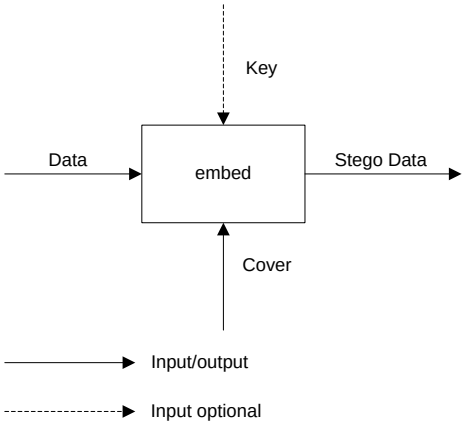
Sebuah pesan steganografi (*plaintext*), biasanya pertama-tama dienkripsikan dengan beberapa arti tradisional, yang menghasilkan *ciphertext*. Kemudian, *covertext* dimodifikasi dalam beberapa cara sehingga berisi *ciphertext*, yang menghasilkan *stegotext*. Contohnya, ukuran huruf, ukuran spasi, jenis huruf, atau karakteristik *covertext* lainnya dapat dimanipulasi untuk membawa pesan tersembunyi; hanya penerima (yang harus mengetahui teknik yang digunakan) dapat membuka pesan dan mendekripsikannya.

Steganografi terdiri atas dua teknik yaitu :

a. Teknik Penyembunyian data

Penyembunyian data dilakukan dengan mengganti bit-bit data di dalam segmen citra dengan bit-bit data rahasia. Hingga saat ini sudah banyak dikemukakan oleh para ilmuwan metode-metode penyembunyian data. Metode yang paling sederhana adalah metode modifikasi LSB. Pada susunan bit di dalam sebuah byte (1byte = 8bit), ada bit yang paling berarti (most significant bit atau MSB) dan bit yang paling kurang berarti yaitu LSB.

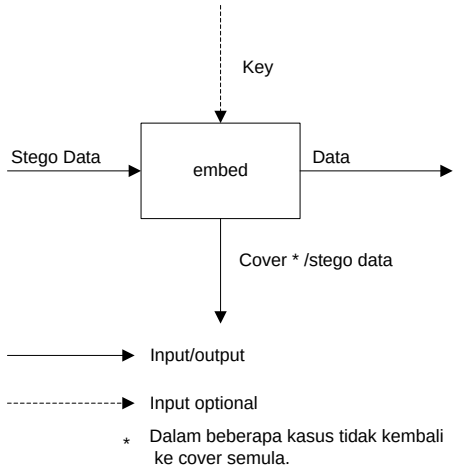
Misalnya pada byte 11010010, bit 1 yang pertama (digaris bawahi) adalah bit MSB dan bit 0 yang terakhir (digaris bawahi) adalah bit LSB. Bit yang cocok untuk diganti adalah bit LSB, sebab perubahan tersebut hanya mengubah nilai byte tersebut satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya. Misalkan byte tersebut menyatakan warna tertentu, maka perubahan satu bit LSB tidak mengubah warna keabuan tersebut secara berarti. Selain itu, mata manusia tidak dapat membedakan perubahan yang kecil.



Gambar 3
Penyembunyian Data

b. Teknik Pengungkapan Data

Data yang disembunyikan di dalam citra dapat dibaca kembali dengan cara pengungkapan (reveal atau extraction). Dengan cara mengumpulkan kembali bit-bit data rahasia yang bertaburan di dalam citra.



Gambar 4
Pengungkapan Data

Beberapa metode untuk membuat suatu steganografi yaitu Least Significant Bit (LSB), Algorithms and Transformation, Redundant Pattern Encoding, Spread Spectrum method dan End Of File. Metode-metode tersebut digunakan dalam steganografi dalam media dan fungsi yang berbeda-beda untuk memaksimalkan pengamanan suatu data (informasi) agar menjadi rahasia. Dalam pembangunan metode yang digunakan yaitu Least Signifacant Bit (LSB) yang berfungsi sebagai tempat penyisipan data.

Metode ini banyak digunakan karena metode ini paling sederhana dan mudah diimplementasikan. Media penampung yang paling sering digunakan dalam mengimplementasikan steganography adalah gambar. Kehandalan penggunaan file gambar dibandingkan dengan media lain adalah kualitas gambar yang telah disisipi pesan rahasia tidak berbeda jauh dengan kualitas citra aslinya.

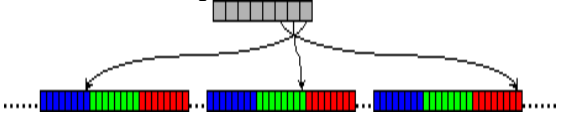
Metode Lsb

LSB

Metoda yang digunakan untuk menyembunyikan pesan pada media digital tersebut berbeda-beda. Contohnya, pada berkas image pesan dapat disembunyikan dengan menggunakan cara menyisipkannya pada bit rendah atau bit yang paling kanan (LSB) pada data pixel yang menyusun file tersebut. Pada berkas bitmap 24 bit, 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 berkas bitmap 24 bit kita dapat menyisipkan 3 bit data.

Kekurangan dari LSB Invertion: Dapat diambil kesimpulan dari contoh 8 bit pixel, menggunakan LSB Insertion dapat secara drastis mengubah unsur pokok warna dari pixel. Ini dapat menunjukkan perbedaan yang nyata dari cover image menjadi stego image, sehingga tanda tersebut menunjukkan keadaan dari steganografi. Variasi warna kurang jelas dengan 24 bit image, bagaimanapun file tersebut sangatlah besar. Antara 8 bit dan 24 bit image mudah diserang dalam pemrosesan image, seperti cropping (kegagalan) dan compression (pemampatan).

Keuntungan dari LSB Insertion : Keuntungan yang paling besar dari algoritma LSB ini adalah cepat dan mudah. Dan juga algoritma tersebut memiliki software steganografi yang mendukung dengan bekerja di antara unsur pokok warna LSB melalui manipulasi pallete.



Gambar 5
Least Significant Bit

Berikut adalah konsep *Significant Bit*. Untuk setiap byte pesan kita harus :

a. Ambil pixel
b. Dapatkan bit pertama dari byte pesan
c. Dapatkan salah satu komponen warna pixel
d. Dapatkan bit pertama dari komponen warna
e. Jika warna bit berbeda dari bit pesan, set/reset
f. Lakukan hal yang sama untuk tujuh bit lainnya.

Stegger

Stegger adalah sebuah *class library open source* yang berlisensi. Dan untuk membuat aplikasi pengamanan pesan dengan steganografi berbasis web ini kami menggunakan *library* tersebut. Stegger mengambil keuntungan dari konsep dasar gambar untuk menyembunyikan data dengan cara mengubah nilai setiap warna primer yang ada pada setiap pixel baik 1 atau 0. Seperti yang kita tahu bahwa semua digital data hanya berupa urutan dari kedua angka ini yakni 1 dan 0, ada beberapa cara untuk mengubah sebuah 1 dan 0, cara yang digunakan oleh stegger adalah mengubah angka berdasarkan urutan ganjil maupun genap. Langkah ini memungkinkan kita untuk menyimpan 1 bit data pada setiap warna primer yang berarti kita bisa menyimpan 3 bit symbol data didalam setiap pixels.

Sekarang 3 bit per pixels terdengar kurang untuk menyembunyikan data. Tapi satu hal yang perlu di perhatikan adalah jika sebuah gambar dengan ukuran 800 x 600 pixels mempunyai 480000 pixels. Berarti kita bisa menyimpan sekitar 1440000 bit data atau sekitar 175 KB.

Terdapat beberapa limitasi pada metode ini. File gambar yang bisa digunakan adalah gambar yang lossless atau dengan katalain gambar yang memanfaatkan semua bit. Jika tidak maka semua data yang secara simbolik tersimpan di dalam gambar akan terhapus atau hancur. Oleh sebab itu metode ini tidak bisa digunakan pada gambar GIF dan JPEG oleh sebab itu Stegger menyimpan semua gambar yang terencode sebagai PNG.

PNG merupakan ekstensi untuk file gambar dengan kualitas baik,ringkas,kapasitas penyimpanan terkompresi untuk gambar raster, PNG Menyediakan paten yang bebas untuk sebagai solusi alternative pengganti GIF dan TIFF seperti yang banyak digunakan. Gambar dengan Warna Terindex, Grayscale,Warna Asli di dukung dengan file extensi ini

Selain itu terdapatnya sebuah tambahan alpha channel untuk mengukur kedalaman gambar antara 1 sampai 16 bit. PNG dirancang untuk dapat di lihat pada kebanyakan aplikasi online seperti world wide web, dimana dapat di stream pada tampilan yang berubah ubah. PNG merupakan file yang sangat kokoh , disamping menyediakan fasilitas integrity checking yang berguna untuk memeriksa keutuhan file. PNG juga dapat menyimpan Gamma dan kromatisitas data untuk meningkatkan pencocokan pada platform warna yang berbeda.

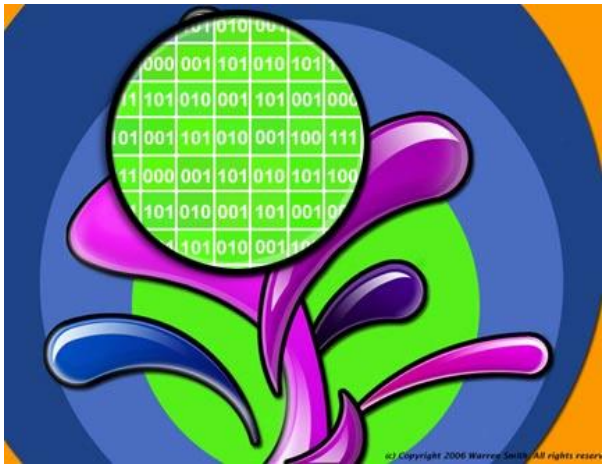
Berikut adalah gambar penjelasan detail bagaimana cara stegger menyimpan data.

Gambar 6
Contoh Gambar

Gambar 7
Pixel dalam Gambar

Gambar 7
Tiap Pixel Terdiri Atas Banyak Warna

Gambar 8
Tiap Warna terdiri dari 3 warna primer yakni Merah, Hijau dan Biru



Gambar 9
Tiap Warna memiliki nilai biner

Gambar 10
Tiap Pixel disisipi data yang akan disembunyikan

Kita bisa menyimpan tiga karakter data dalam 8 pixel dengan asumsi tiap karakter mengambil tempat sampai satu byte yang terdiri dari 8 bit. Sebagai contoh, karakter 'A' pada gambar diatas disimpan dalam format biner 00001010.

Enkripsi

1. Enkripsi

Untuk meembuat keamanan berlapis, ditambahkan enkripsi di dalam steganografi. Enkripsi sendiri adalah proses mengamankan suatu informasi dengan membuat informasi tersebut tidak dapat dibaca tanpa "kunci" yang telah ditentukan sebelumnya. Enkripsi banyak digunakan dalam kepentingan militer maupun agen pemerintah yang memang bertujuan untuk menjaga kerahasiaan informasi. Namun saat ini enkripsi telah digunakan untuk kebutuhan yang lebih luas, seperti pembayaran online untuk situs e-commerce.

Enkripsi mempunyai algoritma untuk mengenkripsi data. Data yang telah terenkripsi disebut sebagai *ciphertext*. Rumus ini memerlukan sebuah variabel untuk mengembalikan data tersebut kembali ke bentuk asal. Variabel ini biasa disebut kunci. Tanpa kunci, seseorang sangat sulit bahkan hampir tidak mungkin, untuk dapat memecahkan kode enkripsi tersebut. Maka kunci ini memegang peranan vital di dalam enkripsi.

Enkripsi terbagi menjadi dua: simetris dan asimetris (juga disebut sebagai public key). Enkripsi simetris memungkinkan sebuah file dijalankan melalui program dan membuat sebuah kunci untuk

mengacak file tersebut. Kemudian file terenkripsi dan kunci dikirimkan secara terpisah kepada penerima. Penerima menjalankan aplikasi enkripsi yang sama dan menggunakan kunci yang diberikan untuk menyatukan kembali file yang telah diacak. Kelebihan enkripsi simetris adalah sangat mudah dan sangat cepat dalam penggunaannya, tetapi tidak seaman enkripsi asimetris, karena jika kunci tersebut jatuh ke tangan orang lain, maka mudah untuk menyatukan file.

Berbeda dengan enkripsi simetris, enkripsi asimetris lebih rumit tapi lebih aman. Hal ini dikarenakan dibutuhkan dua kunci yang saling berhubungan untuk membuka file. Kunci tersebut adalah kunci publik dan kunci pribadi. Kunci publik disediakan bagi siapa saja yang ingin dikirimkan informasi yang terenkripsi. Namun, kunci tersebut hanya dapat digunakan untuk mengkodekan data. Jika ingin mendekodekan data, maka dibutuhkan kunci pribadi yang disimpan oleh pemilik kunci. Kelebihan dari enkripsi asimetris adalah tingkat keamanannya sangat tinggi, tapi kekurangannya adalah dibutuhkan proses dan waktu yang lebih banyak untuk mengenkripsi dan mendekripsi data.

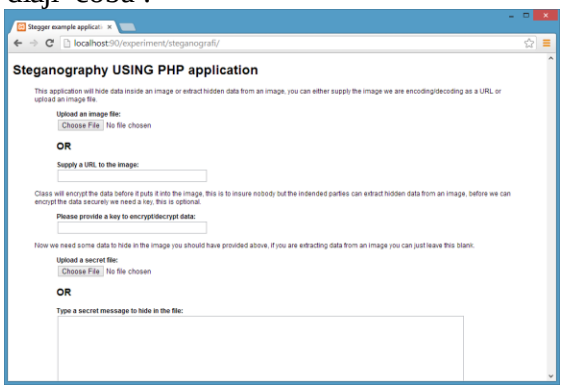
2. Secrypt

Untuk menjaga keamanan dari data maka setiap informasi harus dienkripsi. Pada stegger fungsi yang dipanggil untuk enkripsi data adalah secrypt. Yaitu dengan cara memkombinasikan antara informasi yang diberikan dengan sebuah key.

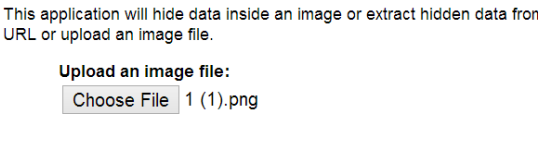
Implementasi

Proses Encode

Berikut ini adalah mekanisme *encoding* dari program yang memiliki fungsi steganografi dan enkripsi berbasis web untuk diuji coba :

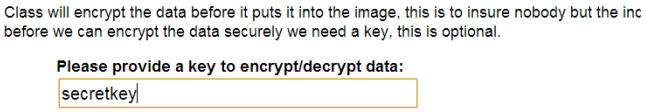


Gambar 11
Overview Untuk Encode aplikasi



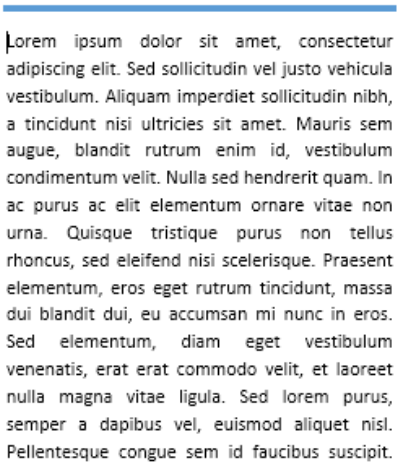
Gambar 12
Upload gambar

Pada aplikasi akan ditampilkan menu pilihan untuk mengupload file image seperti gambar diatas.



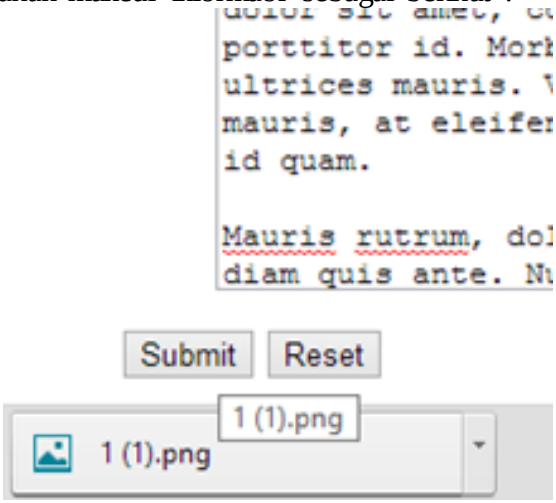
Gambar 13
Kata Kunci

Jika telah dimasukkan file gambar yang ingin disisipi pesan kemudian aplikasi memerintahkan untuk memasukan kata kunci seperti yang ada pada gambar di atas. Setelah itu dilakukan penginputan pesan sebagai berikut:


Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed sollicitudin vel justo vehicula vestibulum. Aliquam imperdiet sollicitudin nibh, a tincidunt nisi ultricies sit amet. Mauris sem augue, blandit rutrum enim id, vestibulum condimentum velit. Nulla sed hendrerit quam. In ac purus ac elit elementum ornare vitae non urna. Quisque tristique purus non tellus rhoncus, sed eleifend nisi scelerisque. Praesent elementum, eros eget rutrum tincidunt, massa dui blandit dui, eu accumsan mi nunc in eros. Sed elementum, diam eget vestibulum venenatis, erat erat commodo velit, et laoreet nulla magna vitae ligula. Sed lorem purus, semper a dapibus vel, euismod aliquet nisl. Pellentesque congue sem id faucibus suscipit.

Gambar 14
Pesan

Langkah terakhir yang perlu dilakukan adalah men-*submit* pesan tersebut. Setelah semua langkah itu dilakukan maka akan muncul informasi sebagai berikut :



Gambar 15
Konfirmasi

Setelah gambar yang berisi pesan terdownload, setelah dibuka maka yang akan tertampil adalah isi dari pesan yang sudah di masukan sebelumnya seperti yang terlihat di gambar 14.

Gambar
Sebelum di encode

Gambar
Sesudah di encode

Gambar 16
Perbandingan hasil gambar

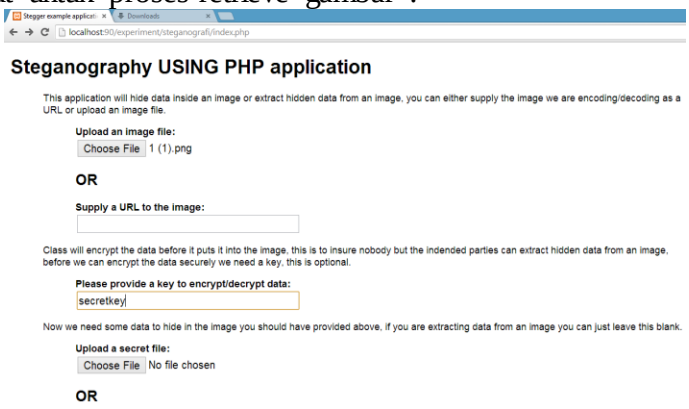
Proses Decode

Jika dilihat dengan kasat mata gambar 16 terlihat sama sekali tidak mengalami perubahan. Namun yang sebenarnya terjadi, pesan pada gambar 15 mengalami enkripsi yang dikerjakan oleh secrypt seperti yang terlihat di gambar 17 berikut :

o5ckHqKTthyfi59LtbBttW5VuOw+08b3drRKn2RqiMzkwu
Wi2SGBMWbMM7JMvQWzmWC7iWgmKCWo5z1WWq9
WzU2rCseOAKsUjf4hMrzNAjqGvFqXvIB4jrD9Z1jqtGe13J
7ppEwF+b4Yci+rWSlRE3gLnTKVl2Nb3eGk+C56V4yRAOG
1GYebI0tx5R6RTISw4Z9rsD4EzIBkEEJfussrSegxcRyP2yOm
V5VOGIwjZxGOud8ak/KMMn3kwGCkvtFZxC2AeMliJ7hH
kRnG6X5tussuwDDU5YfOHCsLC2oSfiTCQRkx2ti4/Tfm5rQ
wBjik2aQS5M1KtGbE8nAoGr8obMRNF/fMuthU5X413TT
5OOZvYamEBg6nP69YpydL+PqlxmzjcRDwrSajwxhNrcdBC
/MdzmTBscTBeyyMDgr8CPH+Z1Tnj1Mz2cnf/tCPj507Cy/
vRiIdvAc/hDyH6Atb/ZSH0+a/gsmLBsYm90Clbbo7crhYbm
ZEgGzcOROdTudxveLVo11N0TzSL18vxf5zXJ8crsluXzZFm9
WYFKYb0axFTSNxPHJSdY4EfNmlLVZfqGYWcJ8eZCD7jW2
UaJ+0/ItTPxGJ6YDliEMBTGLHbqrx3q2T1dAB+Mk4Fvv2P
HcV/epo+lpxlyaOKvgpUmP0RotrfhmfbncguC794zXXc0N/

Gambar 17
Hasil Enkripsi Pesan

Berikut ini adalah layout untuk proses retrieve gambar :

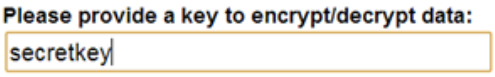


Gambar 18
Overview Untuk Decode Gambar

Setelah itu yang perlu dilakukan adalah mengupload gambar yang telah berisi pesan rahasia :

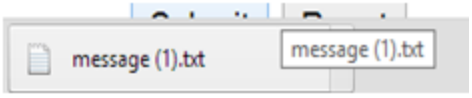
Gambar 19
Upload Gambar Berisi Pesan Rahasia

Langkah selanjutnya adalah menginput secret key yang nantinya akan membaca pesan terenkripsi yang tersimpan dalam gambar



Gambar 20
Input Sercet Key

Setelah dilakukan submit maka pesan secara otomatis terdownload :



Gambar 21
Pesan Rahasia yang terdownload

Seperti yang terlihat pada gambar 16, secara kasat mata gambar biasa dengan gambar berisi pesan rahasia tidak memiliki perbedaan, namun bila dilakukan komparasi secara lebih mendalam maka dapat kami temukan perbedaannya. Komparasi yang kami lakukan adalah komparasi secara objective. Sebelumnya kami menyisipkan pesan rahasia yang sama pada lima gambar dengan ukuran berbeda sebagai pembanding dengan detil sebagai berikut :

Tabel 1
Gambar 1.png

File Name	1.png
File Size in Byte	15.783 Bytes
File Extension	PNG
Pixel Dimension	100x75 Pixels
MD5 Checksum	E04F0C43D841E846 C6F598893D8256F7
SHA 1 Checksum	255365C10AF0C1E2 623F053CCBE5DA9 36BD7BF9A

Tabel 2
Gambar 2.png

File Name 2.png

File Size in Byte	37.449 Bytes
File Extension	PNG
Pixel Dimension	200x150 Pixels
MD5 Checksum	84D441C639D65F2 D3DEBFA709CABF D38
SHA 1 Checksum	07623B5EE3C716C8 307F6C1B3C6E45A 39F034883

Tabel 3
Gambar 3.png

File Name	3.png
File Size in Byte	154.842 Bytes
File Extension	PNG
Pixel Dimension	300x225 Pixels
MD5 Checksum	A630750BD2EE4D1 93845B03B06AAD1 E9
SHA 1 Checksum	4DB16410E3367513 B061BF96770D5C8 345B95100

Tabel 4
Gambar 4.png

File Name	4.png
File Size in Byte	213,562 Bytes
File Extension	PNG
Pixel Dimension	400x300 Pixels
MD5 Checksum	2982CCD1F6FBE85 FB80F675A967F22 A6
SHA 1 Checksum	9245F34E34B82421 0BBF06597094FB0 B53EF8596

Tabel 5
Gambar 5.png

File Name	5.png
File Size in Byte	297,255 Bytes
File Extension	PNG
Pixel Dimension	500 X 311 Pixels
MD5 Checksum	E04F0C43D841E846C6F 598893D8256F7
SHA 1 Checksum	414578BEA1B93E18CF2 99EF09AD41430BA0D8 CA5

Kelima gambar tersebut memiliki ukuran dan dimensi pixel yang berbeda satu sama lain, oleh sebab itu sesuai dengan teori yang dikemukakan bahwa setiap dalapan pixel dapat menyimpan 3 byte. Maka besar alokasi pesan yang bisa disembunyikan dapat dirumuskan sebagai berikut :

Jumlah Byte = (luas gambar / 8) * 3

Dengan catatan bahwa luas gambar adalah panjang pixel dikali dengan lebar pixels. Berdasarkan rumus tersebut maka dapat diperoleh hasil sebagai berikut :

Tabel 6
list perbandingan gambar

Gambar	Luas	Daya Tampung bit	Besar Pesan byte	Perkiraan Karakter
1.PNG	7500	937.5	2812.5	2800
2.PNG	30000	3750	11250	11250
3.PNG	76500	9562.5	28687.5	28687
4.PNG	120000	15000	45000	45000
5.PNG	155500	19437.5	58312.5	58300

Untuk menguji gambar yang dipakai adalah gambar yang sudah disiapkan sebelumnya dan menggunakan kata kunci sebagai kombinasi enkripsi yakni "*secretkey*". Untuk informasi yang digunakan diambil dari lipsum generator dimana bisa mengeluarkan informasi sesuai dengan bytes yang di inginkan.

Berikut adalah hasil komparasi kelima gambar dalam tabel setelah dilakukan penyisipan pesan terenkirpsi dengan metode komparasi secara obyektif :

1. Melihat isi dari matrix dengan menggunakan matlab

50	49	17	20	18	10	13
39	41	30	30	18	10	12
43	53	30	11	13	8	9
24	33	23	12	17	20	14

Gambar 19
Matriks gambar 1.png sebelum di encode (4 Baris terakhir yang ditampilkan)

50	48	17	20	18	10	13
38	41	31	30	18	10	12
42	53	31	11	12	9	8
24	32	23	12	16	21	15

Gambar 20
Matriks gambar 1.png sudah di encode (4 Baris terakhir yang ditampilkan)

Jika dilihat maka terdapat pebedaan nilai matrix pada baris terakhir.

2. Menghitung jarak pola antara gambar yang sebelum dan sesudah diencode dengan menggunakan ciri ciri
   a. Nilai Entropy
   b. Nilai Mean
   c. Nilai Intensitas Cahaya

Berikut hasil pengujian dengan cara menghitung jarak pola antar gambar.

Tabel 7
Jarak Pola

Gambar Sebelum di encode	**Sesudah di encode**
	
Nilai Entropy = 7.7923	Nilai Entropy = 7.7935
Nilai Mean = 100.3791	Nilai Mean = 100.3655
Nilai Intensitas Cahaya = 77.8493	Nilai Intensitas Cahaya = 77.8539
Jarak Pola antar gambar sebelum dan sesudah di encode = 0.0144	

Jika dilihat dari data diatas maka ada beberapa pola dari gambar yang berubah dengan nilai yang relative kecil. atau bisa dibilang hampir mendekati nilai 0.

3. Melihat perbedaan detail informasi dari gambar sebelum dan sesudah di encode dengan melihat aspek
   a. File Size dalam bytes
   b. Md5 Cheksum
   c. SHA1 Checksum

d. SHA 256 Checksum

Berikut Hasil pengujian dengan cara melihat informasi file. Hasil komparasi checksum file menunjukan perbedaan yang mencolok baik dari sisi besar file maupun nilai hash.

Tabel 8
Hasil komparasi checksum file

Gambar Sebelum di encode	**Hasil Komparasi**	**Sesudah di encode**
		
File Size **15,783 Bytes**	Lebih Kecil setelah di encode	File Size **13,422**
MD5 Checksum **E04F0C43D841E846C6F598893D8256F7**	Berbeda	MD5 Checksum **758A0F5F5830C7A1FEDA3FAFEF538368**
SHA 1 Checksum **255365C10AF0C1E2623F053CCBE5DA936BD7BF9A**	Berbeda	SHA1 Checksum **9684829643E0A4AAC0623EE432B26494F97D3FE2**

Hasil yang didapat adalah percobaan steganografi untuk file 1.png dengan ukuran 100 x 75 Pixels. berikut adalah hasil matriks percobaan untuk 5 gambar yang sudah didefinisikan sebelumnya menggunakan matlab.

Tabel 9
Hasil Matriks Percobaan

NO	**Filename**	**Nilai Matrix Terakhir sebelum di encode**	**Nilai Matrix Terakhir sesudah di encode**
1	1.png	63 65 65 65 67 68 67 66 71 48 51 47 24 33 23 12 17 20 14	62 64 65 65 66 69 66 66 70 49 51 47 24 32 23 12 16 21 15
2	2.png	144 143 143 142 140 139 136 135 135 133 131 130 128 129 127 126 125 123 123 123 122 120	145 143 143 142 141 138 137 134 134 132 130 130 128 128 127 127 125 123 122 123 123 120
3	3.png	60 52 39 38 43 74 110 126 110 77 103 131 99 87	61 52 38 38 43 75 111 127 111 77 102 130 99 87
4	4.png	214 211 226 230 208 176 103 89 91 89 52 58 51 45 57 39 48 64 76	214 211 226 230 208 176 103 89 90 89 52 58 50 45 57 38 49 65 77
5	5.png	64 65 65 62 55 53 56 55 52 48 50 53 53 53 53 52	65 65 64 62 54 52 57 55 53 48 50 53 52 52 53 53

Tabel 10
Perhitungan Jarak Antar Pola

NO	Filename	Jarak Antar Pola
1	1.png	**0.0144**
2	2.png	**0.0161**
3	3.png	**0.0133**
4	4.png	**0.0103**
5	5.png	**0.0081**

Dapat dilihat adanya perubahan jarak antar pola pada gambar setelah disisipi pesan namun dengan perbedaan yang sangat kecil yang rata-rata memiliki perbedaan mendekati nol.

Kesimpulan

Strategi keamanan berlapis pada steganografi dengan menggunakan metode LSB dengan pemanfaatan algoritma stegger dan enkripsi menggunakan secrypt telah meningkatkan keamanan informasi atau data yang disisipkan pada citra digital. Berikut kelebihan dari keamanan berlapis ini:

a. Metode LSB dapat menyembunyikan pesan yang sulit untuk dipecahkan.
b. Citra digital yang disisipkan dengan metode LSB ditambah dengan enkripsi akan makin sulit untuk dipecahkan oleh orang yang tidak berkepentingan.
c. Dibutuhkan *secret key* untuk mendekodekan enkripsi yang di*generate* dengan menggunakan secrypt.
d. Kunci untuk mendekodekan enkripsi secrypt berupa citra digital asli sebelum disisipi pesan, sehingga tidak menimbulkan kecurigaan.
e. Secret Key dikirimkan terpisa dengan pesan rahasia sehingga bila terjadi hal-hal yang tidak diinginkan maka kemungkinan pesan tersebut untuk terpecahkan masih sangat kecil.

Daftar Pustaka

Awcock, G.W. 1996. Applied Image Processing.Singapore. McGraw-Hill Book.

Pengertian dan Jenis Enkripsi, Shvoong, http://id.shvoong.com/

Provos, Niels, Honeyman. Peter. (2003), "Hide And Seek: An Introduction To Steganography", IEEE Computer Society.

Raghunathan, A., 2011. *Proofs in Cryptography*, Stanford University.