ENKRIPSI *FILE* MENGGUNAKAN METODE STEGANOGRAFI DAN *XML SERIALIZER* UNTUK PENGAMANAN *FILE* BERBASIS *ONLINE*

Tri Aji Nugroho [1], Soetam Rizky Wicaksono [2]
[1] Program Studi Sistem Informasi, STIKOM Surabaya
[2] Program Studi Sistem Informasi, STIKOM Surabaya, email: soetam@stikom.edu

Abstract: With a lot more people using communication service in internet, makes a lot of problems occur. Many people look for the way to protect communicated information, especially for company confidential document, and with plain sending method it can be hack on the way to the destination. This paper uses steganography and XML Serialized to give some options for protecting the document through the web based application. Also with implementation of XML Web service as link between web application and desktop application, the confidential document will be better kept since that the document will be more secure, and also the authentication of the data is more guaranteed.

Keywords: Steganography, XML Serializer, XML Web Service

Berbagai macam layanan komunikasi tersedia di *internet*, diantaranya adalah *web*, *e-mail*, milis, *news-group*, *e-learning* dan sebagainya. Dengan semakin maraknya orang memanfaatkan layanan komunikasi di *internet* tersebut, maka permasalahanpun bermunculan, apalagi ditambah dengan adanya *hacker* dan *cracker*. Banyak orang kemudian berusaha menyiasati bagaimana cara mengamankan informasi yang dikomunikasikannya atau menyiasati bagaimana cara mendeteksi keaslian dari informasi yang diterimanya.

Studi kasus yang ada saat ini adalah pada PT. Grafindo Media Pratama, di mana PT. Grafindo Media Pratama adalah sebuah perusahaan percetakan yang memiliki kantor cabang di semua kota besar di pulau Jawa dan juga melayani pemesanan buku untuk luar pulau. Untuk mem-buat laporan di kantor pusat, kantor pusat meminta data-data dari kantor-kantor cabang dan dikirimkan melalui *internet* melalui *upload* pada suatu situs. Namun, *file* yang dikirimkan tidak mendapat-kan proses perlindungan data sama sekali. Hal ini sangat berbahaya karena jenis *file* yang dikirimkan termasuk rahasia. Maka, untuk lebih melindungi ke-amanan data dari *file* yang dikirimkan tersebut, saat *file excel* atau *word* melalui proses *upload*, *file excel* atau dokumen tersebut akan melalui proses steganografi sehingga akan menjadi sebuah *file* gambar yang kemudian akan dilakukan proses *posting* atau ditampilkan di halaman *web*.

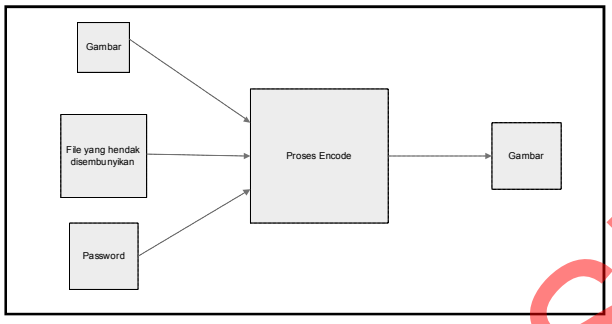
Kemudian ketika *file* gambar tersebut dilakukan *download,* maka proses tersebut akan melalui sebuah proses enkripsi lagi, yaitu *XML Serializer* yang mana proses tersebut akan dilakukan secara majemuk, sehingga proses pemecahan akan menjadi lebih sulit, sehingga *file* yang telah melalui proses pada aplikasi ini akan lebih aman. Kemudian saat hendak melakukan dekripsi harus melalui proses *login* sehingga hanya *user* yang berkepentingan yang dapat mengakses *file* tersebut.

LANDASAN TEORI

Steganografi merupakan seni penyembunyian pesan ke dalam pesan lainnya sedemikian rupa, sehingga orang lain tidak menyadari ada sesuatu di dalam pesan tersebut. Kata steganografi (*steganography*) berasal dari bahasa Yunani yaitu *steganos* yang artinya tersembunyi atau terselubung dan *graphein*, yang artinya menulis, sehingga kurang lebih artinya adalah "menulis tulisan yang ter-sembunyi atau terselubung" (Sellars, 1996). Teknik ini meliputi banyak sekali metoda komunikasi untuk menyem-bunyikan pesan rahasia. Metoda ini termasuk tinta yang tidak tampak, *microdots*, pengaturan kata, tanda tangan digital, jalur tersembunyi dan komunikasi spektrum lebar.

Steganografi pada gambar biasanya menggunakan teknik penggantian *LSB*. Bagi *computer* gambar adalah *file* yang berisi kumpulan warna dan intensitas cahaya pada daerah yang berbeda. Dengan menggunakan penggantian *LSB*, maka untuk mendapatkan hasil terbaik sebaiknya digunakan 24 bit *Bitmap*, dikarenakan ukurannya yang besar dan memiliki resolusi tinggi. Dengan ukuran yang besar maka pesan yang dapat dibawa semakin besar dan dengan resolusi tinggi tidak akan terlihat perubahan yang signifikan. Namun, pada kenyataannya banyak yang menggunakan 8 bit *Bitmap* atau dengan menggunakan format lainnya seperti *GIF, JPEG,* atau *PNG* untuk menghindari kecurigaan.

Dalam proses *encoding* program ini dibutuhkan tiga buah *input* yaitu gambar, *password* dan *file* yang hendak disembunyikan. Pada dasarnya, proses *encoding* dapat digambarkan dengan diagram pada Gambar 1.

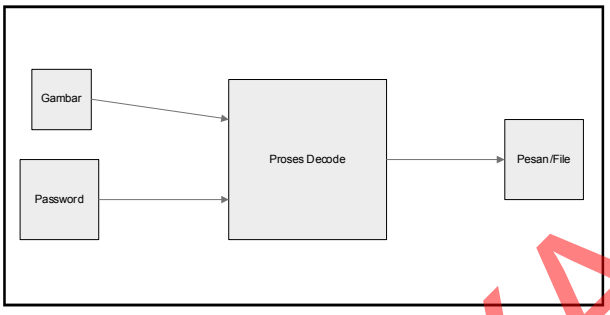


Gambar 1 Proses *Encoding*

Pada proses *encoding* yang pertama kali yang dilakukan adalah merubah *password* yang diberikan menjadi bilangan integer yang dipakai sebagai bilangan acak, di mana bilangan acak tersebut dipakai sebagai posisi penulisan data yang disembunyikan. Jadi, data yang disembunyikan tidak bisa dibaca dengan program lain meskipun program tersebut juga menggunakan teori steganografi.

Proses selanjutnya adalah menghitung panjang karakter dari pesan yang disembunyikan, kemudian menyembunyikan panjang karakter dan pesan ke dalam gambar. Dalam proses penyembunyian ini, diperlukan bilangan *random* yang digunakan sebagai posisi dari *color-channel* yang diedit. Proses penyembunyian tersebut dilakukan berulang-ulang sepanjang *file* yang dimasukkan *user*. Kemudian gambar yang telah diedit tersebut disimpan kembali sebagai *output* proses *encoding*.

Proses *decoding* merupakan kebalikan dari proses *encoding*. Proses ini berguna untuk memperoleh pesan atau *file* yang disembunyikan dalam sebuah gambar. Proses *decoding* dapat dilihat pada Gambar 2.



Gambar 2 Proses *Decoding*

Proses *decoding* memerlukan gambar dan *password* sebagai *input*. Jika gambar dan *password* tersebut benar, maka pada posisi yang ditentukan oleh bilangan *random* tersebut akan didapat suatu bilangan *integer* yang merupakan panjang pesan/*file* yang disembunyikan. Kemudian proses dilakukan berulang-ulang hingga semua karakter dari pesan/*file* yang disembunyikan diperoleh.

Serialisasi adalah proses yang berjalan *run time* untuk mengkonversi obyek ke dalam bentuk sekuensial *byte* secara linier (Esposito, 2003). Kegunaan utama dari sebuah proses serialisasi adalah pemrosesan lebih lanjut hasil serialisasi ke sebuah bentuk blok memori untuk ditransfer melalui jaringan dengan protokol yang umum.

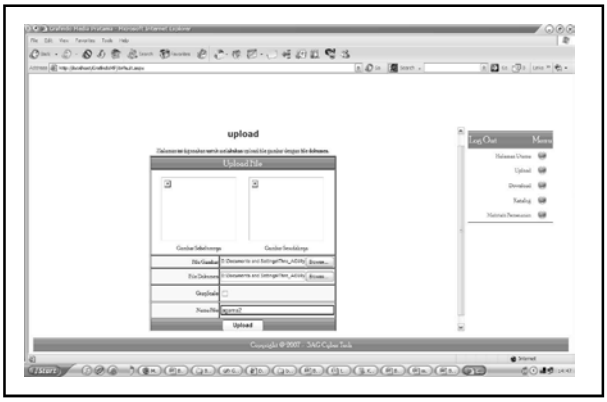
Sebuah proses serialisasi dapat menghasilkan tiga macam bentuk output antara lain:

1. *Binary*
2. *Simple Object Access Protocol* (*SOAP*)
3. *XML*

PEMBAHASAN

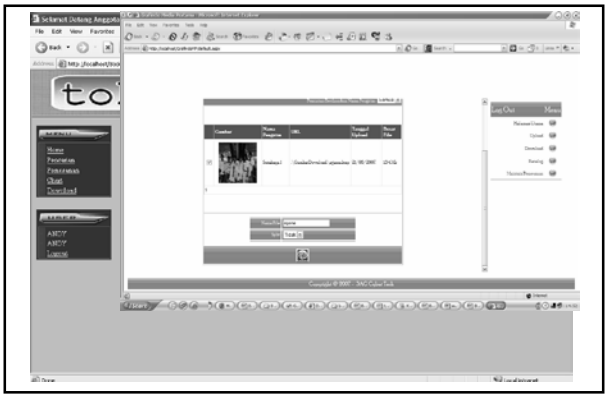
Pada tahap ini, diuraikan hasil dan pembahasan penelitian terhadap perangkat lunak dimulai dari masukan ke proses validasi terhadap perangkat lunak.

Berikut ini contoh gambar *user* hendak mengirimkan dokumen dan gambar ke *web server*.



Gambar 3 Tampilan *Upload* Dokumen

Kemudian *user* bisa melakukan *download* gambar melalui halaman *download*, ketika proses *download* dilakukan, maka gambar akan mengalami proses serialisasi menjadi *XML*.



Gambar 4 Tampilan *Download* Gambar

Apabila *file* terlalu besar, *user* dapat memilih untuk melakukan pemotongan *file* terlebih dahulu. Dengan metode *split* ini *file* gambar tersebut akan dipecah menjadi beberapa *file* berekstensi .3AG. Setelah itu, akan muncul halaman baru yang berisi *link* dari *file split* tersebut. Kemudian apabila data-data *user* tervalidasi dengan benar maka akan tampil *form* utama yang membaca isi dari *file* dengan mendekripsi *file* tersebut.

Gambar 5 Tampilan Utama Aplikasi *Desktop*

Setelah mengalami proses steganografi, *bit depth* dari *file* gambar akan berubah menjadi 32 *bit*. Yang dimaksud dengan *bit depth* adalah adalah jumlah *bit* yang digunakan untuk mempresentasikan tiap titik dalam representasi citra grafis. Makin besar jumlah *bit* yang digunakan untuk mempresentasikan suatu titik, semakin banyak warna dan atau bayangan abu-abu yang dapat dibuat. Sehingga, semakin besar *bit depth* maka gambar akan lebih jelas atau jernih.

SIMPULAN

Secara umum aplikasi enkripsi *file* menggunakan metode Steganografi dan *XML Serializer* untuk pengamanan *file* berbasis *online* ini telah berfungsi sebagaimana yang diharapkan. Untuk itu dapat diambil beberapa kesimpulan dari sistem ini sebagai berikut:

1. Tercapainya tujuan pembuatan yaitu melindungi suatu *file* format *excel* dan *word* serta menyamarkannya ke dalam *file* gambar dengan metode Steganografi dan *XML Serializer*.
2. Kecepatan proses steganografi dengan cara menyisipkan *byte* secara langsung ke *byte* warna gambar lebih cepat daripada penggunaan *LSB*.
3. Hasil enkripsi dengan menggunakan steganografi akan menghasilkan pembengkakan *file* sebesar 59,8% tergantung dari ukuran *file* dan *bit depth* dari *file* yang bersangkutan.
4. *File ciphertext* tidak dapat dilakukan *dekripsi* apabila aplikasi tidak terhubung dengan *server* yang menampung *XML Web services*.
5. *File* hasil steganografi akan mengalami kenaikan *bit depth* menjadi 32 *bit* dikarenakan penggunaan *bitmap*.

Sedangkan, saran pengembangan dari *prototipe* ini adalah sebagai berikut:

1. Aplikasi dapat dikembangkan dengan menggabungkan penggunaan algoritma enkripsi sebelum proses steganografi sehingga jauh lebih aman.
2. Aplikasi dapat dikembangkan untuk menyisipkan *file* lain selain *file word* dan *excel*.
3. Aplikasi dapat dikembangkan dengan menambah proses kompresi *file*. Dikarenakan apabila *file* yang dilakuan proses *upload* besar, maka waktu yang dibutuhkan lama, sehingga dengan dilakukan kompresi *file* diharapkan *file* menjadi lebih kecil dan waktu yang dibutuhkan menjadi lebih cepat.
4. Penggunaan *XML Serializer* menjadi lebih dinamis, tidak dibatasi oleh jumlah *file* melainkan oleh besar *file* yang hendak dilakukan *download.*

RUJUKAN

Ananta, DE. 2003. *Skripsi:Pembuatan Program Aplikasi Penyembunyian Data Dengan Metode Steganography*. Skripsi Tidak Diterbitkan. Surabaya: Program Strata Satu Sarjana Komputer STIKOM Surabaya.

Deitel. 2002. *Visual Basic.NET, How To Program*. New Jersey: Prentice Hall.

Esposito, D. 2003. *Applied XML Programming for Microsoft.NET*. Washington: Microsoft Press.

Johnson, NF. & Jajodia, S. 1998. *Exploring Steganography: Seeing the Unseen*, (Online), (www.jjtc.com/pub/r2026.pdf, diakses pada 2 September 2006).

Rusiawan, D. 2003. *Tinjauan Aspek Keamanan Sistem Web Service*. Skripsi. Bandung : Program Studi Magister Teknologi Informasi ITB.

Sellars, D. *An Introduction to Steganography*, (Online), (http://www.cs.uct.ac.za/courses/CS400W/NIS/papers99/dsellars/stego.htm, diakses pada 2 September 2006)