Implementasi Algoritma Kriptografi Blowfish dan Metode Steganografi End Of File (EOF) untuk Keamanan Data

Cucu Tri Eka Yuliana[1]
[1,3] *Jurusan Teknik Informatika, FASILKOM UDINUS*
Jln. Nakula 1 No 5-11 Semarang 50131 INDONESIA
[1]cucutriekayuliana@gmail.com

Abstract-**Cryptography is the science of encryption technique where the data is scrambled using an encryption key to be something that is not easily read by someone who does not have the decryption key. But now, cryptanalyst (science and art to solve without knowing the cryptographic keys used) has grown, so many algorithms that can be solved, so the security level of cryptography is reduced. So that one of them needed steganographic techniques to enhance security by hiding the message that has been encrypted. Steganography is the process of hiding a message in an image. Steganography can be said to be more secure because it does not randomize, so the inserted file is not suspicious, although each method has its advantages and disadvantages. In this study, an encryption implementation using Blowfish cryptography algorithm is created and then it is hidden in an image by using End of File (EOF) method. It is aimed to increase the message security because the encryption message will be hidden in an image which the difference is invisible. The implementation of this study is applied by using a source in Visual Basic 6.0 programming language. The result is a picture which does not show the secret message in it.**

I. PENDAHULUAN

Dunia informatika saat ini berkembang sangat pesat dan membawa dunia ke era teknologi, karena itulah saat ini informasi menjadi sangat penting [1]. Maka mulai bermunculan berbagai media untuk mengakses informasi dan mengelola informasi sehingga banyak kemudahan-kemudahan yang kemudian didapat untuk mengakses suatu informasi.

Dari kemudahan akses informasi ini membawa pengaruh terhadap tingkat keamanan informasi yang dapat berakibat pada keamanan data itu sendiri. Sehingga informasi penting yang seharusnya rahasia menjadi sangat beresiko diketahui dan disalahgunakan oleh pihak lain yang tidak berhak. Hal itu menyebabkan munculnya rasa ketidakyakinan pada penerima maupun pengirim informasi tersebut. Adanya kekhawatiran penerima apakah informasi benar adanya dari sumber yang dimaksudkan, dan keraguan pengirim akankah informasi yang dikirim diterima oleh pihak yang dimaksudkan.

Bahkan berdasarkan CSI/FBI *Computer Crime and Security Survey* 2005, mayoritas perusahaan memberikan anggaran untuk pengembangan sistem dan keamanan data sekitar 5% dari total anggaran perusahaan tersebut. Salah satu contoh masalah keamanan komputer yang pernah terjadi di Amerika, terjadi pada Maret 2005. Seorang mahasiswi dari UCSB dituduh melakukan kejahatan mengubah data-data nilai ujiannya (dan beberapa mahasiswa lainnya). Mahasiswi itu melakukan hal tersebut dengan mencuri identitas dua orang profesor [2].

Contoh lainnya di Indonesia, ada beberapa kasus sehubungan dengan kejahatan komputer. Salah satu kasusnya adalah seorang *cracker* Indonesia (yang dikenal dengan nama hc) tertangkap di Singapura ketika mencoba menjebol sebuah perusahaan di Singapura, September dan Oktober 2000. Setelah berhasil membobol bank Lippo, kembali Fabian Clone beraksi dengan menjebol web milik Bank Bali. Perlu diketahui bahwa kedua bank ini memberikan layanan *Internet banking* [2].

Oleh karena itu dengan adanya kriptografi diharapkan data akan tetap terjaga kerahasiaannya hingga informasi tersebut sampai pada tujuannya, serta memberikan keyakinan pada penerima bahwa informasi tersebut memang benar berasal dari pengirim yang tepat begitu pula sebaliknya pengirim yakin bahwa penerima informasi adalah pihak yang tepat.

Kriptografi akan merahasiakan informasi dan menyandikannya ke dalam informasi acak yang tidak dimengerti kecuali oleh penerima yang tepat. Hal ini dikarenakan kriptografi itu sendiri ialah ilmu dan seni untuk menjaga keamanan pesan atau data [3]. Kriptografi mengubah sebuah data atau informasi menjadi pesan acak yang sesuai dengan kunci pengacaknya. Dimana kunci ini diketahui oleh pengirim dan penerima informasi saja.

Tentunya pengaplikasian kriptografi akan mengamankan sebuah informasi. Namun, saat ini kriptanalisis atau kemampuan untuk memecahkan kriptografi turut berkembang pesat, telah banyak metode kriptografi yang terpecahkan. Sehingga metode pengamanan informasi pun harus terus dikembangkan.

Sedangkan steganografi menjadi salah satu alternatif untuk menambah keamanan yang telah didapat dari kriptografi, steganografi adalah teknik menyembunyikan data rahasia di dalam wadah (media) digital sehingga keberadaan data rahasia tersebut tidak diketahui oleh orang lain [4]. Penyisipan ini dapat diterapkan pada berbagai media seperti gambar, suara dan video. Jika pesan disisipkan pada media dan tidak merubah bentuk media tersebut, maka tingkat keamanan pesan akan

meningkat karena tidak adanya kecurigaan keberadaan pesan pada media tersebut.

Jika kriptografi dan steganografi berdiri sendiri-sendiri, maka akan ada kemungkinan dapat terpecahkan. Sehingga penulis melakukan penelitian untuk menggabungkan antara kriptografi dan steganografi agar keamanan informasi dapat lebih terjamin. Maka tema tugas akhir ini adalah "Implementasi Algoritma Kriptografi *Blowfish* Dan Metode Steganografi *End Of File* (EOF) Untuk Keamanan Data" sebagai bahan pertimbangan dalam proses pengamanan data agar tidak terjadi pencurian atau pengrusakan data.

II. STUDI PUSTAKA

2.1. Penelitian Terkait

Penulis memulai penelitiannya dengan terlebih dahulu melakukan studi kepustakaan dari penelitian-penelitian dan sumber-sumber lain. Dari studi kepustakaan itu, penulis menemukan beberapa penelitian yang mendorong penulis untuk mengangkat tema seperti di atas. Penelitian tersebut membahas tentang topik yang terkait dengan penelitian penulis, antara lain adalah penelitian mengenai salah satu algoritma yang digunakan penulis atau kedua algoritma yang akan diangkat oleh penulis.

Penelitian pertama yang berkaitan dengan laporan ini berjudul "Aplikasi Kriptografi File Menggunakan Algoritma Blowfish", ditulis oleh Suriski Sitinjak, Yuli Fauziah dan Juwairiah. Penelitian tersebut menerapkan algoritma *Blowfish* untuk mengenkripsi dan dekripsi *file*. Hasil dari penelitian tersebut berupa aplikasi "*BlowCrypt*", aplikasi tersebut dapat mengenkripsi *file* dalam bentuk teks, gambar, suara, video juga *archieve* seperti .zip dan .rar. Kesimpulan yang dapat ditarik dari penelitian tersebut ialah kecepatan enkripsi/dekripsi tergantung dari besarnya ukuran *file*, semakin besar ukuran *file*, semakin banyak waktu yang dibutuhkan. Terjadi penambahan *byte* pada *file* hasil enkripsi sehingga mengakibatkan ukuran *file* enkripsi dan *file* asli sedikit berbeda. Namun, jika telah di dekripsi, maka ukuran *file* akan kembali seperti semula [5].

Penelitian kedua berjudul "Studi Pustaka Untuk Steganografi Dengan Beberapa Metode". Penelitian kedua ini ditulis oleh Yogie Aditya, Andhika Pratama dan Alfian Nurlifa. Penelitian tersebut membandingkan antara dua metode steganografi yang sering digunakan, yaitu *Least Significant Bit (LSB)* dan *End Of File (EOF).* Kesimpulan yang dapat ditarik dari penelitian tersebut adalah metode LSB tidak mengubah ukuran *file* setelah disisipi pesan namun metode LSB tersebut banyak menurunkan kualitas gambar. Sedangkan metode EOF lebih baik karena kualitas gambar tetap terjaga meskipun ukuran *file* akan menjadi lebih besar [6].

Penelitian ketiga berjudul "Penerapan Algoritma *Blowfish* Pada Proses Steganografi Metode EOF" ditulis oleh Paskalis Andrianus Nani. Penelitian tersebut memadukan kelebihan algoritma *Blowfish* dan metode *End Of File (EOF),* dimana pesan rahasia dienkripsi terlebih dahulu menggunakan algoritma *Blowfish*, lalu setelah itu disisipkan pada citra digital menggunakan metode *End Of File*. Kesimpulan dari penelitian tersebut adalah steganografi dengan metode *End Of File* merupakan teknik yang relatif mudah dimengerti, namun pesan yang terdapat di dalamnya dapat dengan mudah diekstrak oleh pihak yang tidak bertanggungjawab. Namun dengan mengenkripsi pesan rahasia sebelum disisipkan ke dalam citra dapat meningkatkan tingkat keamanan pesan. Hal itu didukung pula dengan fakta bahwa hingga saat ini belum ada peneliti yang dapat menemukan celah keamanan dari algoritma *Blowfish*. Hasil akhir dari penelitian tersebut menunjukkan bahwa ukuran citra setelah disisipi sebuah pesan rahasia akan bertambah besar [7]. Perbedaan penelitian tersebut dengan penelitian penulis ialah penelitian sebelumnya hanya mengenkripsi *file* berekstensi .txt, sedangkan penelitian penulis akan mengembangkannya dengan mengenkripsi *file* berekstensi .doc dan .docx, karena saat ini ekstensi yang sering digunakan untuk pembuatan soal ialah ekstensi tersebut.

2.2. Tinjauan Pustaka

A. Kriptografi

Kriptografi adalah ilmu mengenai teknik enkripsi dimana data diacak menggunakan suatu kunci enkripsi menjadi sesuatu yang sulit dibaca oleh seseorang yang tidak memiliki kunci dekripsi [10]. Enkripsi adalah proses merubah plainteks (pesan asli) menjadi suatu pesan tersandi (cipherteks).

B. Steganografi

Steganografi adalah seni untuk menyembunyikan pesan di dalam media digital sedemikian rupa sehingga orang lain tidak menyadari ada sesuatu pesan di dalam media tersebut [12].

C. Algoritma Blowfish

Blowfish merupakan metode enkripsi yang mirip dengan DES yang diciptakan oleh Bruce Schneier yang ditujukan untuk mikroprosesor besar (32 bit ke atas dengan *cache* data yang besar) [7].

Blowfish dikembangkan untuk memenuhi kriteria desain sebagai berikut :

1. Cepat, pada implementasi yang optimal *Blowfish* dapat mencapai kecepatan 26 *clock cycle per byte*.
2. Kompak, *Blowfish* dapat berjalan pada memori kurang dari 5KB.
3. Sederhana, *Blowfish* hanya menggunakan operasi yang simpel: penambahan (*addition*), XOR, dan penelusuran tabel (*table lookup*) pada *operand* 32 bit. Desainnya mudah untuk dianalisa yang membuatnya resisten terhadap kesalahan implementasi.
4. Keamanan yang variabel, panjang kunci *Blowfish* dapat bervariasi dan dapat mencapai 448 bit (56 *byte*).

D. End Of File (EOF)

Metode *End Of File* (EOF) merupakan salah satu metode yang digunakan dalam steganografi. Teknik ini digunakan dengan cara menyisipkan data pada akhir *file* [14]. Sehingga tidak akan mengganggu kualitas data awal yang akan disisipkan pesan.

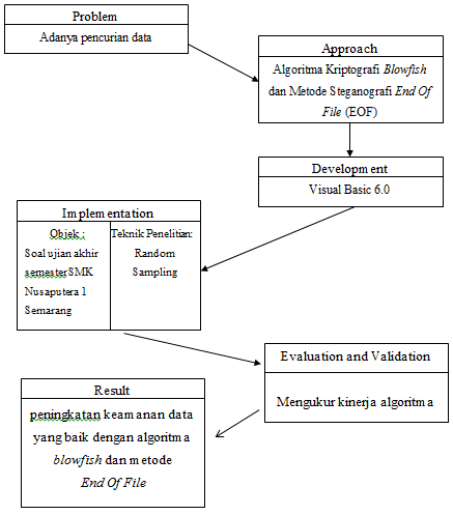
Namun, ukuran *file* yang setelah disisipkan pesan akan bertambah. Sebab ukuran *file* sebelum disisipkan pesan rahasia ditambah dengan ukuran pesan rahasia yang disisipkan. Untuk mengenal data yang disisipkan pada akhir *file*, diperlukan suatu tanda pengenal atau simbol pada awal dan akhir data yang akan disisipkan.

Menurut Paskalis A. N. proses penyisipan pesan dengan metode *End of File* ini dapat ditulisan sebagai berikut [7]:

1. Inputkan pesan yang akan disisipkan
2. Ubah pesan menjadi kode-kode desimal.
3. Inputkan citra *grayscale* yang akan disisipi pesan.
4. Dapatkan nilai derajat keabuan masing- masing piksel.
5. Tambahkan kode desimal pesan sebagai nilai derajat keabuan diakhir citra.
6. Petakan menjadi citra baru.

2.3. Keranga Pemikiran

Kerangka pemikiran pada penelitian Implementasi Algoritma Kriptografi *Blowfish* Dan Metode Steganografi *End Of File* (EOF) Untuk Keamanan Data ini dapat diilustrasikan pada diagram berikut ini :



Gambar 1. Kerangka Pemikiran

III METODE PENELITIAN

3.1 Instrumen Penelitian

Penelitian ini dilakukan berdasarkan permasalahan yang telah di uraikan pada bab sebelumnya. Adapun bahan dan peralatan guna penelitian ini adalah :

A. Bahan

Bahan penelitian ini yaitu data yang dibutuhkan berupa soal-soal ujian pada SMK Nusaputera 1 Semarang. Data soal ujian itu merupakan soal ujian akhir semester ganjil tahun ajaran 2013/2014 dengan ekstensi .doc dan .docx.

B. Peralatan

Tujuan dari deskripsi peralatan adalah untuk mengetahui kebutuhan sistem agar mempermudah perancangan. Peralatan ini meliputi kebutuhan perangkat keras dan perangkat lunak sistem. Kebutuhan-kebutuhan di bawah ini merupakan kebutuhan minimal dari sistem :

a) Kebutuhan Perangkat Keras :
   - Prosesor Pentium IV.
   - Harddisk 20 GB.
   - Memori DDR 128 MB.
   - Layar Display 14'.
   - Satu buah mouse dan keyboard standart.

b) Kebutuhan Perangkat Lunak :
   - Sistem operasi Microsoft Windows XP.

     Pada penelitian ini sstem operasi minimal yaitu Microsoft Windows XP, dikarenakan untuk pembuatan sistem menggunakan bahasa pemrograman *visual basic* sudah dapat berjalan pada sistem operasi ini.
   - Software Microsoft Visual Studio 6.0

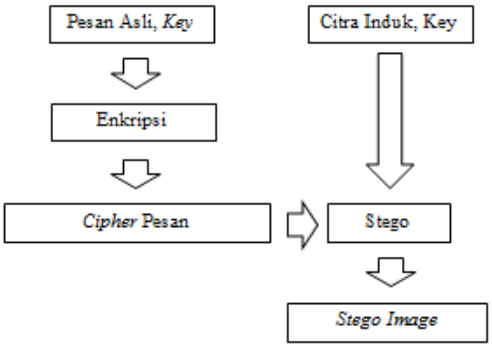
     Bahasa pemrograman yang digunakan untuk perancangan sistem adalah *Visual Basic* 6.0 karena bahasa ini merupakan bahasa yang sering digunakan dan dapat menjalankan perhitungan aritmatika dan logika yang dibutuhkan dalam pembuatan sistem.
   - Software Microsoft Word 2007

     Software pengelola data yang dianjurkan sebagai spesifikasi minimal adalah Microsoft Word 2007, karena saat ini ekstensi yang sering digunakan adalah .docx yang dapat dijalankan pada software ini, dan software ini sendiri juga dapat menjalankan data dengan format .doc.

3.2Metode yang diusulkan

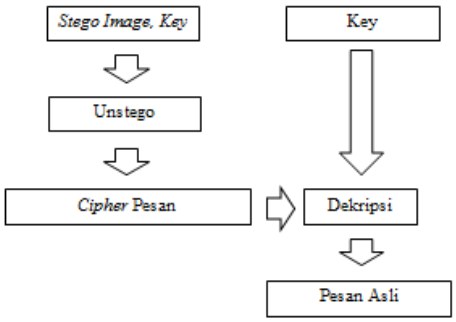
Metode yang diusulkan untuk proses seperti yang telah dijelaskan di atas yaitu Algoritma Kriptografi *Blowfish* dan Metode Steganografi *End Of File* (EOF).

- Prosedur Enkripsi Data yang Diusulkan



Gambar 2. Metode Enkripsi Data

- Prosedur Dekripsi Data yang Diusulkan



Gambar 3. Metode Dekripsi Data

IV HASIL PENELITIAN DAN PEMBAHASAN

4.1 Algoritma Blowfish

Berdasarkan artikel yang ditulis oleh Suriski Sitinjak, Yuli Fauziah dan Juwairiah [5] Blowfish diciptakan oleh seorang *Cryptanalyst* bernama Bruce Scheneir, Presiden perusahaan *Counterpane Internet Security, Inc* (Perusahaan konsultan tentang kriptografi dan keamanan komputer) dan dipublikasikan tahun 1994.

Algoritma blowfish memiliki 2 bagian, yaitu ekspansi kunci dan enkripsi data:

1. Ekspansi kunci

   Pada tahap ini, akan mengubah minimum kunci menjadi *array* sub kunci sebanyak 4168 *byte*.
2. Enkripsi data

Enkripsi data ini dilakukan sebanyak 16 kali putaran dan masukannya adalah data X yang terdiri dari 64 bit, dimana setiap putaran menggunakan operasi

XOR. Untuk setiap putaran, pertamaXOR-kan blok kiri dengan sub kunci untuk *round*/putaran itu.

4.2 *Langkah Kerja Blowfish*

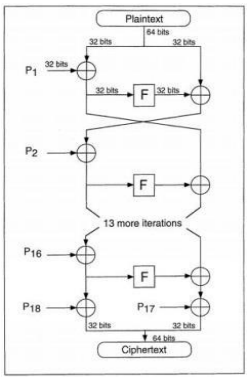
A. Proses Ekspansi Kunci

1. Inisialisasi P-array yang pertama dan juga empat S-box, berututan, dengan string yang telah pasti.
   String tersebut terdiri dari digit-digit heksadesimal dari phi, tidak termasuk angka tiga di awal.
   Contoh :
   P1= 0x243f6a88
   P2= 0x85a308d3
   P3= 0x13198a2e
   P4= 0x03707344
   dan seterusnya sampai dengan P18.
2. XOR-kan P1 dengan 32-bit awal kunci, XOR-kan P2 dengan 32-bit berikutnya dari kunci, dan seterusnya untuk semua bit kunci. Jika panjang kunci ternyata kurang dari jumlah Pbox, maka siklus perhitungan akan diulangi hingga semua P ter-XOR-kan.
3. Enkripsikan string yang seluruhnya nol (all-zero string) dengan algoritma Blowfish, menggunakan subkunci yang telah dideskripsikan pada langkah 1 dan 2.
4. Gantikan P1 dan P2 dengan keluaran dari langkah 3
5. Enkripsikan keluaran langkah 3 menggunakan algoritma Blowfish dengan subkunci yang telah dimodifikasi.
6. Gantikan P3 dan P4 dengan keluaran dari langkah 5.
7. Lanjutkan langkah-langkah di atas, gantikan seluruh elemen P-array dan kemudian keempat S-box secara berurutan, dengan hasil keluaran algoritma Blowfish yang terus-menerus berubah.

B. Proses Enkripsi

1. Berdasarkan metode enkripsi blowfish, untuk langkah awal yaitu file inputan atau file soal akan dirubah menjadi *binary*. Proses enkripsi dan dekripsi blowfish dilakukan dengan pemecahan file menjadi per 64 bit. Yang kemudian 64 bit ini diinisialkan "x".
2. Masukan data "x" 64 bit ini kemudian dipecah menjadi dua bagian yang disebut dengan XR (X *Right*) dan XL (X *Left*) masing-masing 32 bit.
3. Menurut buku "Pengantar Ilmu Kriptografi" karangan Dony Ariyus [9], proses enkripsi maupun dekripsi dilakukan dengan iterasi fungsi sederhana sebanyak 16 kali. Berikut rumus enkripsi :
   XL = XL XOR p[i]
   XR = F(XL) XOR XR
   Tukar XL dan XR
   Dimana rumus F adalah $F(X_L) = ((S[1,a] + S2[2,b] \bmod 2^{32}) \text{ XOR } S[3,c]) + S[4,d]$
4. Setelah iterasi ke-16, tukar XL dan XR lagi untuk melakukan pembatalan penukaran terakhir.
5. Dari semua langkah diatas, langkah terakhir adalah kembali menggabungkan XR dan XL untuk mendapatkan cipherteks.

Proses enkripsi dengan algoritma *blowfish* ini dapat digambarkan melalui bagan di bawah ini :



Gambar 4.Alur Proses Enkripsi *Blowfish*

6. Untuk Proses Dekripsi, langkah sama persis dengan proses enkripsi, hanya saja urutan Pbox digunakan dengan urutan terbalik.

C. Proses Stego

Teknik penyisipan data pada *End Of File* yaitu menyisipkan data rahasia pada akhir file [14]. Teknik ini sendiri ditujukan agar tidak mengurangi kualitas dari gambar induk sebelum dan sesudah penyisipan. Dimana jika diruntutkan, maka proses penyisipan gambar yakni seperti di bawah ini :

1. Input pesan yang akan disisipkan.
2. Ubah pesan menjadi kode heksadesimal.
3. Buka *file* hasil stego dengan mode *write*. Tambahkan kode heksadesimal pesan diakhir citra setelah tanda "|".

D. Proses Unstego

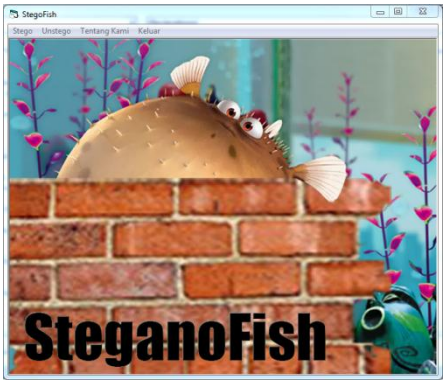
Proses unstego merupakan proses dimana pesan rahasia di dalam gambar stego akan dikeluarkan. Berikut merupakan langkah penulis untuk proses unstego :

1. Input gambar stego.
2. Ubah menjadi kode heksadesimal.
3. Baca ukuran *file* sampai dengan tanda "|".
4. Kurangi ukuran *file* dengan *file* induk.
5. Buka *file* hasil unstego dengan mode *write*.
6. Letakkan kode heksadesimal mulai dari tanda "|" sampai akhir ke dalam *file* hasil.

4.3 Implementasi

Halaman Utama Aplikasi SteganoFish

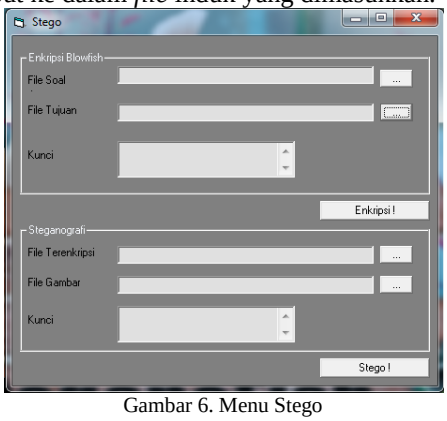
Pada halaman utama ini terdapat empat pilihan menu, yaitu Menu Stego, Menu Unstego, Menu Tentang Kami dan Menu Keluar dari Aplikasi.



Gambar 5. Menu Awal

Halaman Menu Stego

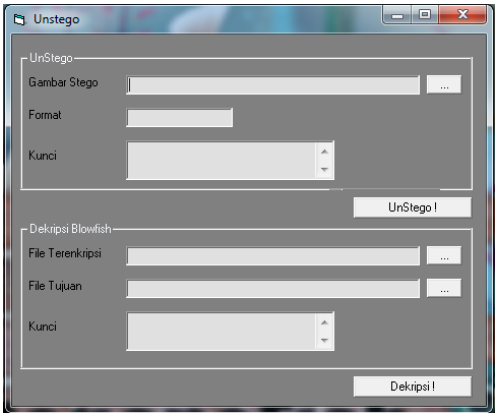
Halaman ini berfungsi untuk melakukan proses enkripsi dan stego *file*. Pada halaman ini *file* soal yang dimasukkan akan disimpan ke *file* tujuan setelah proses enkripsi. Kemudian *file* yang telah di enkripsi akan di masukkan untuk kemudian di lakukan proses stego atau penyembunyian *file* tersebut ke dalam *file* induk yang dimasukkan.



Gambar 6. Menu Stego

Halaman Menu Unstego

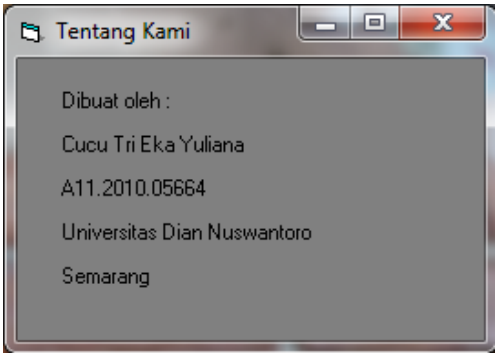
Pada halaman menu ini, berfungsi untuk mengeluarkan pesan tersembunyi dari dalam *file* gambar dan kemudian men-dekripsi menjadi *file* awal. Sebagai masukan, pengguna diharuskan memasukkan *file* gambar yang di dalamnya terdapat pesan rahasia, memasukkan format apa yang diinginkan untuk *file* setelah di-unstego, kemudian *file* yang terenkripsi serta *file* tujuan penyimpan hasil dekripsi dan kunci dekripsi *blowfish*.



Gambar 7. Menu Unstego

Halaman Tentang Kami

Halaman ini memuat informasi tentang pembuat aplikasi. Berikut tampilan menu halaman Tentang Kami :



Gambar 8. Menu Tentang Kami

4.4 Hasil Pengujian Aplikasi

Pengujian aplikasi dilakukan dengan metode *random sampling*, yakni mengambil masing-masing 25 sampel data soal dan 25 sampel data gambar induk guna diujikan pada aplikasi.

File kemudian diujikan pada aplikasi untuk melihat hasil akhir dari enkripsi dan stego yang diproses pada aplikasi. Untuk pengujian, penulis memasukkan kunci sama untuk tiap-tiap percobaan, kunci yang dimasukkan adalah “anna”.

Masing-masing data akan diujikan ke aplikasi. Pertama, data soal akan diproses dengan algoritma *blowfish*, proses ini disebut dengan proses enkripsi. Pada proses enkripsi bertujuan mengacak *file* soal atau dapat disebut pesan rahasia agar pihak lain yang tidak berwenang tidak dapat menemukan makna atau pesan rahasia di dalamnya.

Kemudian hasil dari proses enkripsi tadi akan diproses guna menyembunyikan *file* hasil enkripsi tersebut dalam sebuah citra atau gambar. Proses itu sendiri disebut dengan proses stego. Untuk kedua proses ini, baik enkripsi maupun stego, atau sebaliknya proses dekripsi maupun unstego, pengguna akan diminta memasukkan sebuah kunci. Dimana kunci ini berbeda antara kedua proses tersebut untuk meningkatkan keamanan, hanya saja untuk kunci proses enkripsi maupun dekripsi akan dibatasi dengan panjang maksimal 56 huruf saja, sesuai sengan ketentuan algoritma *blowfish* yang mengolah kunci sebanyak 448 bit maksimal.

Pada proses enkripsi soal yang menggunakan algoritma *blowfish*, maka akan dilakukan perhitungan panjang sesuai dengan jalannya algoritma blowfish dengan 16 kali perulangan dan jumlah Sbox sebanyak 4 kali 255, serta perhitungan XOR dengan Pbox sebanyak 18 buah. Jika dilihat dari panjangnya perhitungan, hal itu penulis perkirakan dapat menyebabkan beberapa hal seperti ukuran *file* bertambah besar dan kemungkinan meningkatnya tingkat pemakaian CPU.

Maka untuk membuktikan hal tersebut dapat terjadi atau tidak, maka penguji akan menjabarkan ukuran *file* soal setelah terenkripsi seperti di bawah ini :

Tabel 1. Tabel Pengujian

NO	FILE SOAL	UKURAN FILE SOAL	FILE INDUK	UKURAN FILE INDUK	UKURAN GAMBAR STEGO	PROSESOR *)
1	AGAMA KRISTEN KLS X.doc	612 KB	SAM_0026.JPG	2,46 MB	3,06 MB	49%
2	AGAMA KRISTEN KLS XI.doc	612 KB	SAM_0025.JPG	2,49 MB	3,09 MB	49%
3	AGAMA KRISTEN KLS XII.doc	616 KB	SAM_0024.JPG	2,62 MB	3,21 MB	49%
4	soal ag budha kelas XII.doc	636 KB	Koala.jpg	764 KB	1,36 MB	49%
5	soal ag budha kelas XI.doc	616 KB	SAM_0023.JPG	2,39 MB	2,99 MB	49%
6	Soal B IND X SMK TKJ 2.docx	1,12 MB	SAM_0022.JPG	2,68 MB	3,81 MB	50 %
7	SOAL IPA KLS X.doc	780 KB	Chrysanthemum.jpg	860 KB	1,60 MB	50%
8	SOAL IPA KLS XI.doc	680 KB	Jellyfish.jpg	760 KB	1,40 MB	50%
9	SOAL IPA KLS XII.doc	924 KB	SAM_0021.JPG	2,32 MB	3,22 MB	50%
10	soal ipa X.doc	616 KB	Lighthouse.jpg	552 KB	1,14 MB	52%
11	soal ipa XI.doc	620 KB	Desert.jpg	828 KB	1,41 MB	49%
12	Soal KIMIA X Genjil New.doc	648 KB	SAM_0020.JPG	2,47 MB	3,10 MB	51%
13	Soal KIMIA XI GANJIL new.docx	132 KB	SAM_0019.JPG	2,63 MB	2,76 MB	42%
14	Soal KIMIA XII GANJIL new.doc	1,51 MB	SAM_0018.JPG	2,49 MB	4,01 MB	49%
15	soal pkn X.docx	564 KB	Hydrangeas.jpg	584 KB	1,11 MB	49%
16	soal pkn XI.docx	560 KB	SAM_0017.JPG	2,46 MB	3,01 MB	49%
17	soal pkn XII.docx	560 KB	SAM_0090.JPG	2,5 MB	3,04 MB	49%
18	Soal UKAS fisika X.doc	1,42 MB	winnda.jpg	5 MB	6,43 MB	49%
19	Soal UKAS fisika XI.doc	820 KB	SAM_0088.JPG	2,51 MB	3,32 MB	49%
20	Soal UKAS fisika XII.doc	0,97 MB	SAM_0066.JPG	2,7 MB	3,67 MB	49%
21	UKAS B INGGRISsh X.docx	576 KB	muslimah.jpg	48 KB	620 KB	50%
22	UKAS KKPI X soal A.docx	560 KB	motor.jpg	161 KB	720 KB	41%
23	UKAS KKPI soal B.docx	20 KB	mia.JPG	28,3 KB	48 KB	1%
24	UKAS KKPI XI soal A.docx	52 KB	awan.jpg	60 KB	112 KB	23%
25	UKAS KKPI XI soal B.docx	52 KB	jajan.jpg	44 KB	96 KB	23%

Berdasarkan tabel 1 hasil *review* setelah proses enkripsi melalui ukuran *file* sesudah dan tingkat penggunaan CPU, maka dapat dilihat bahwa algoritma kriptografi *blowfish* ini tidak menambah besar ukuran *file* setelah enkripsi dan kalaupun bertambah hanya sedikit saja. Namun jika dilihat dari tingkat pemakaian CPU pada saat proses enkripsi, maka dapat dilihat adanya kenaikan yang cukup signifikan pada aktifitas pemakaian CPU.

Hal di atas dapat terjadi karena adanya perhitungan rumit algoritma ini yang diulang terus menerus dengan menggunakan banyak array Sbox maupun Pbox. Tentunya hal itu akan meningkatkan aktifitas pemakaian prosesor pada saat terjadi proses perhitungan atau biasa disebut enkripsi.

Sedangkan untuk hasil dari enkripsi ini sendiri adalah pengacakan isi *file* soal sehingga tidak dapat lagi terbaca. Untuk pengujian pada *file* berekstensi .doc, *file* dapat dibuka dan di dalamnya tidak lagi dapat terbaca seperti layaknya soal.

Tabel 1 di atas menunjukkan adanya perbedaan antara ukuran citra sebelum proses stego dan setelah proses stego, dimana ukuran akan bertambah besar. Hal ini disebabkan karena prinsip metode steganografi *end of file* ini sederhana, yaitu menambahkan *file* soal di belakang *file* citra induk. Dengan proses seperti itu, tentunya jumlah ukuran *file* itu akan bertambah. Pertambahan ukuran *file* ini dapat dirumuskan yaitu besar ukuran *file* soal ditambahkan dengan besar ukuran *file* citra itu sendiri. Seperti dapat dilihat di tabel 1, ukuran *file* koala.jpg setelah proses stego menjadi 1,36 MB. Jika dilihat ukuran data sebelum, yaitu *file* soal yang telah di enkripsi sebesar 634 KB jika ditambah dengan ukuran *file* induk sebelum stego sebesar 764 KB maka angka 1,36 MB adalah hasil dari penambahan ukuran kedua *file* itu.

Namun, meskipun terjadi penambahan besar ukuran *file* gambar, metode steganografi *end of* ini memiliki kelebihan dibanding dengan metode lainnya seperti dijelaskan pada kriteria steganografi yakni "*fidelity"* yang dapat dilihat di buku "Pengolahan Citra Digital" karangan Rinaldi Munir [4]. Yang mana *fidelity* ini berarti mutu dari citra setelah proses penyisipan tidak jauh berbeda dengan citra asli. Metode *end of file* ini memiliki kelebihan pada poin ini, secara kasat mata, tidak ada perbedaan dari kedua gambar sebelum dan sesudah proses stego.

Untuk membuktikan kelebihan metode steganografi *end of file* ini, perlu dilakukan perbandingan antara citra sebelum dan sesudah proses stego. Maka, berikut penulis tampilkan beberapa citra induk sebelum dan sesudah stego :

Gambar 9. Koala.jpg sebelum stego

Gambar 10. Koala.jpg setelah stego

Gambar 11. Nisa.jpg sebelum stego

Gambar 12. Nisa.jpg setelah stego

Dari pengujian yang telah dilakukan penulis, dapat menunjukkan bahwa untuk proses stego dengan metode steganografi *end of file* tidak akan merubah kualitas citra, namun akan memperbesar ukuran *file* karena merupakan penambahan antara ukuran *file* pesan rahasia dan *file* induk.

V KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari penelitian yang telah dilakukan, maka penulis dapat menarih beberapa kesimpulan, yaitu sebagai berikut :

1. Aplikasi SteganoFish dapat mengamankan data rahasia dengan baik, karena pengguna lain atau dalam obyek penelitian ini yaitu murid-murid tidak menyadari adanya pesan rahasia dalam sebuah gambar milik guru karena metode steganografi *end of file* tidak menampakkan perbedaan kualitas citra secara kasat mata.
2. Setelah melakukan proses enkripsi *blowfish*, dapat ditarik kesimpulan bahwa besar atau ukuran *file* tidak berubah, meskipun berubah, perbedaannya hanya sedikit saja.
3. Sementara setelah proses stego, dapat dilihat bahwa besar atau ukuran data akan bertambah banyak, besar penambahan ini sendiri minimal adalah jumlah ukuran *file* asli ditambahkan dengan ukuran citra induk.
4. Setelah proses dekripsi maupun unstego, ukuran data yang berubah tadi akan kembali seperti ukuran semula dan tidak berubah dari *file* asli sebelum terjadinya proses enkripsi dan stego.
5. Namun, selain kesimpulan di atas, penulis menemukan kelemahan dari aplikasi ini, yaitu aplikasi ini tidak dapat diterapkan untuk *file-file* yang berukuran sangat besar mengingat saat pengujian untuk *file* soal yang rata-rata memiliki ukuran sebesar 500 KB-1,5 MB saja aktifitas pemakaian prosesor meningkat drastis sampai lebih dari 50%. Jadi jika diterapkan pada ukuran data besar, maka dapat dimungkinkan akan terjadi peningkatan tingkat pemakaian prosesor sampai dengan 100%.

5.2 Penelitian Selanjutnya

Dari kesimpulan yang telah diuraikan penulis di atas, maka penulis memberikan saran untuk proses pengembangan lanjutan dan guna penyempurnaan aplikasi SteganoFish ini adalah :

1. Setelah dilakukan pengujian pada penelitian ini, maka penulis menyarankan penggunaan algoritma kriptografi *blowfish* untuk ukuran data besar dapat menggunakan spesifikasi prosesor besar. Karena pada pengujian penulis menggunakan prosesor *Dual Core* dan menunjukkan adanya aktifitas prosesor yang cukup tinggi saat proses enkripsi.
2. Atau jika tidak adanya spesifikasi yang lebih baik, maka disarankan untuk mengganti algoritma *blowfish* dengan algoritma lain yang proses enkripsinya tidak terlalu banyak perputaran dan array seperti *blowfish*.

DAFTAR PUSTAKA

[1] Rahman Chumaidi, "Studi dan Implementasi Algoritma Blowfish Untuk Enkripsi Email," Institut Teknologi Sepuluh Nopember, Surabaya, Makalah Tugas Akhir 2009.

[2] Lawrence A. Gordon, Martin P. Loeb, William Lucyshyn, and Robert Richardson, *CSI/FBI Computer Crime And Security Survey*.: Computer Security Institute, 2005.

[3] Rinaldi Munir, Kriptografi. Bandung: INFORMATIKA, 2007.

[4] Rinaldi Munir, Pengolahan Citra Digital. Bandung: INFORMATIKA, 2007.

[5] Suriski Sitinjak, Yuli Fauziah, and Juwairiah, "Aplikasi Kriptografi File Menggunakan Algoritma Blowfish," in Seminar Nasional Informatika 2010 (semnasIF 2010), Yogyakarta, 2010, pp. C-85.

[6] Yogie Aditya, Andhika Pratama, and Alfian Nurlifa, "Studi Pustaka Untuk Steganografi Dengan Beberapa Metode," in Seminar Nasional Aplikasi Teknologi Informasi 2010 (SNATI 2010), Yogyakarta, 2010, pp. G-34.

[7] Paskalis Andrianus Nani , "Penerapan Enkripsi ALgoritma Blowfish Pada Proses Steganografi EOF," Universitas Katolik Widya Mandira, Kupang,.

[8] Eko Hari Rachmawanto, "Teknik Keamanan Data Menggunakan Kriptografi Dengan Algoritma Vernam Chiper dan Steganografi Dengan Metode End Of File (EOF)," Universitas Dian Nuswantoro, Semarang, Laporan Tugas Akhir 2010.

[9] Dony Ariyus, Pengantar Ilmu Kriptografi Teori Analisis dan Implementasi. Yogyakarta: ANDI, 2008.

[10] Sentot Kromodimoeljo, Teori & Aplikasi Kriptografi.: SPK IT Consulting, 2010.

[11] Max Teja Ajie Cipta Widiyanto, "Teknik Penyembunyian Pesan File dan TXT Dengan Metode Enkripsi DES dan Enkripsi RC4," Universitas Dian Nuswantoro, Semarang, Laporan Tugas Akhir 2011.

[12] Sutojo T. et al., Teori dan Aplikasi Aljabar Linier & Matriks. UDINUS Semarang: ANDI Yogyakarta, 2010.

[13] Putri Alatas, "Implementasi Teknik Steganografi dengan Metode LSB Pada Citra Digital," Universitas Gunadarma, Jakarta, 2009.

[14] Henny Wandani, Muhammad Andri Budiman, and Amer Sharif, "Implementasi Sistem Keamanan Data dengan Menggunakan Teknik Steganografi End of File (EOF) dan Rabin Public Key Cryptosystem," Universitas Sumatera Utara , Medan,.

[15] H. Ary Setyadi, Pemrograman Visual Basic.: Portal Edukasi Indonesia.