ISSN 1858-4853

KEAMANAN DAN PENYISIPAN PESAN RAHASIA PADA GAMBAR DENGAN ENKRIPSI *BLOWFISH* DAN STEGANOGRAFI *END OF FILE*

M. Mirsa Hariady [1], Addy Suyatno[2], Indah Fitri Astuti[3]

[1,2,3]Program Studi Ilmu Komputer, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Mulawarman
Jalan Barong Tongkok Kampus Gunung Kelua Samarinda, Kalimantan Timur
Email : mirzahariady@gmail.com[1], addysuyatno@yahoo.com[2], indahfitriastuti@yahoo.com[3]

ABSTRAK

Ancaman penyadapan banyak sekali terjadi sekarang, bukan hanya antara orang ke orang yang melakukannya, bahkan sampai antara negara bertetangga terindikasi juga melakukan aksi saling sadap rahasia-rahasia negara yang menyebabkan terjadinya hubungan diplomasi yang kurang baik. Tentunya hal itu menjadi kerugian bagi semua pihak, terlebih oleh pihak yang disadap, sehingga suatu teknik pengamanan data dapat diterapkan untuk mengantisipasi ancaman tersebut seperti teknik kriptografi dan steganografi. Kriptografi adalah suatu ilmu dan seni untuk menjaga keamanan pesan saat pesan dikirim dari suatu tempat ke tempat yang lain. Steganografi adalah teknik pengamanan informasi dengan cara menyembunyikan pesan pada file gambar, audio, ataupun video yang disebut sebagai berkas pembawa. Pada penelitian ini hanya menggunakan file gambar sebagai berkas pembawanya. Penelitian ini mengkombinasikan dua teknik pengamanan data yaitu kriptografi algoritma *blowfish* dan steganografi *end of file.* Pesan rahasia akan melalui proses enkripsi dan menghasilkan cipherteks, kemudian cipherteks akan disisipkan kedalam gambar. Penelitian ini menghasilkan aplikasi keamanan dan penyisipan pesan rahasia dan menunjukkan bahwa pengguna dapat mengamankan informasi rahasia dibalik gambar dengan tidak merusak kualitas gambar tersebut.

Kata kunci : Kriptografi, *Blowfish*, Steganografi, *End Of File.*

1. PENDAHULUAN

a. Latar Belakang

Kebutuhan setiap individu akan informasi berbanding lurus dengan pesatnya perkembangan teknologi informasi. Informasi yang berfungsi untuk memberi wawasan dan pengetahuan sering ditulis dan disimpan seseorang dalam bentuk dokumen atau media lainnya seperti berkirim pesan melalui SMS, email dan sejenisnya yang bersifat pribadi dan bisa dibaca serta mudah dipahami. Internet sebagai salah satu hasil berkembangnya teknologi merupakan sistem jaringan terluas saat ini yang mudah diakses oleh semua orang untuk saling bertukar data dan informasi. Sedangkan informasi yang dikirim tidak hanya informasi yang boleh dibaca semua orang, tetapi ada juga informasi yang bersifat rahasia dan hanya orang-orang atau badan-badan tertentu saja yang boleh membacanya, sehingga informasi menjadi sangat rentan untuk diketahui, diambil atau bahkan dimanipulasi dan disalahgunakan oleh pihak lain yang tidak berhak.

Ancaman penyadapan banyak sekali terjadi sekarang, bukan hanya antara orang ke orang yang melakukannya, bahkan antara negara bertetangga terindikasi juga melakukan aksi saling sadap yang berakibat terjadinya hubungan diplomasi yang kurang baik. Tentunya hal itu menjadi kerugian bagi semua pihak, terlebih oleh pihak yang disadap.

Pengamanan dan kerahasiaan data merupakan salah satu aspek penting dari suatu sistem informasi. Sehingga informasi hanya bisa diakses oleh pemilik informasi atau *user* yang telah ditentukan oleh pemilik informasi. Kriptografi adalah ilmu dan seni untuk menjaga kerahasian pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Teknik pengaman data lain yang juga cukup populer adalah teknik steganografi. Teknik ini berbeda dengan teknik kriptografi yang masih menimbulkan kecurigaan karena pesan yang disamarkan dapat dengan mudah dikenali, steganografi lebih mengurangi kecurigaan, karena pesan yang disamarkan disembunyikan ke dalam file. Jika pesan rahasia disamarkan menggunakan teknik kriptografi lalu disisipkan menggunakan teknik steganografi memungkinkan pesan rahasia yang ingin dikirim ke orang lain akan lebih terjaga keamananya.

Enkripsi algoritma *blowfish* merupakan salah satu dari kriptografi simetris yang dalam proses enkripsi dan dekripsi menggunakan kunci yang sama. Steganografi metode EOF (*End Of File*) merupakan teknik substitusi dalam steganografi yang menyisipkan data pada akhir *file.*

b. Batasan Masalah

Berdasarkan dari latar belakang yang telah diuraikan, agar permasalahan tidak meluas, maka masalah dibatasi pada :

ISSN 1858-4853

1. File yang akan disisipkan pesan berupa gambar dengan ekstensi jpg, png, gif, dan bmp.
2. *Input* sistem berupa teks dan gambar.
3. *Output* sistem berupa gambar stego yang telah berisi pesan ter-enkripsi.
4. Gambar stego tidak boleh di-*edit*.

c. Tujuan Penelitian

Tujuan penelitian ini adalah *user* dapat melakukan pengamanan data yang akan dikirim dengan menggunakan aplikasi pengiriman pesan rahasia yang telah ter-enkripsi dan disisipkan di dalam sebuah media berupa gambar.

d. Manfaat Penelitian

Manfaat dari aplikasi keamanan dan penyisipan pesan rahasia pada media gambar berbasis web yaitu, dapat membantu penggunanya untuk mengirimkan informasi atau data–data yang bersifat rahasia agar sampai ke tangan penerima tanpa menimbulkan kecurigaan pada pihak lain. Pertama, informasi atau pesan rahasia terlebih dahulu diubah menjadi sandi khusus, sehingga pesan akan lebih terjaga kerahasiaannya. Kemudian, pesan yang telah ter-enkripsi disisipkan kedalam media berupa gambar supaya pihak lain tidak menyadari bahwa terdapat pesan didalam media gambar tersebut. Aplikasi pesan rahasia ini tidak membatasi kapasitas pesan yang akan disisipkan pada gambar dan juga tidak mengubah kualitas gambar, sehingga tidak akan menimbulkan kecurigaan.

2. TINJAUAN PUSTAKA

a. Kriptografi

Kriptografi berasal dari bahasa Yunani yang terdiri atas dua kata, yaitu *crypto* dan *graphia*. *Crypto* yang mempunyai arti rahasia (*secret*) dan *graphia* yang mempunyai arti menulis (*writing*). Kriptografi adalah ilmu yang berguna untuk mengacak (kata yang lebih tepat adalah *masking*) data sedemikian rupa sehingga tidak bisa dibaca oleh pihak ketiga. Tentu saja data yang diacak harus bisa dikembalikan ke bentuk semula oleh pihak yang berwenang (Fidens, 2006).

Pada dasarnya kriptografi terdiri dari beberapa komponen sebagai berikut (Ariyus, 2006) :

1. Enkripsi : merupakan hal yang sangat penting dalam kriptografi sebagai pengamanan atas data yang dikirim agar rahasianya terjaga. Pesan aslinya disebut *plaintext* yang diubah menjadi kode-kode yang tidak dimengerti yang disebut dengan *chipertext*. Untuk mengubah *plaintext* kedalam *chipertext* digunakan algoritma yang bisa mengkodekan data yang diinginkan.
2. Dekripsi : merupakan kebalikan dari enkripsi, pesan yang telah dienkripsi dikembalikan ke bentuk asalnya, yang disebut dengan dekripsi pesan.
3. Kunci : berfungsi untuk melakukan enkripsi dan dekripsi. Kunci terbagi menjadi dua, yakni kunci pribadi (*private key*) dan kunci umum (*public key*).
4. *Chipertext* : merupakan suatu pesan yang sudah melalui proses enkripsi. Pesan yang ada pada *chipertext* tidak bisa dibaca karena berisi karakter-karakter yang tidak memiliki makna (arti).
5. *Plaintext* : sering juga disebut sebagai *cleartext*, merupakan suatu pesan bermakna yang ditulis atau diketik (pesan asli) dan *plaintext* itulah yang kemudian akan diproses menggunakan algoritma kriptografi tertentu agar menjadi *chipertext*.
6. Pesan : pesan ini bisa berupa data atau informasi yang dikirim (melalui kurir, saluran komunikasi data, dan sebagainya) atau yang disimpan di dalam media perekaman (kertas, *storage*, dan sebagainya).
7. Kriptanalisis : bisa diartikan sebagai analisis sandi atau suatu ilmu untuk mendapatkan *plaintext* tanpa harus mengetahui kunci secara wajar. Jika suatu *chipertext* berhasil menjadi *plaintext* tanpa menggunakan kunci yang sah, maka proses tersebut dinamakan *breaking code* yang dilakukan oleh para kriptanalis. Analisis sandi juga mampu menemukan kelemahan kunci atau *plaintext* dari *chipertext* yang dienkripsi menggunaan algoritma tertentu

b. *Blowfish*

Blowfish merupakan salah satu jenis kriptografi kunci simetris yang proses enkripsi dan dekripsinya menggunakan kunci yang sama (Munir, 2006). *Blowfish* (*OpenPGP.Cipher.4*) merupakan enkripsi yang metode enkripsinya mirip dengan DES, diciptakan oleh seorang *Cryptanalyst* bernama Bruce Schneier, Presiden perusahaan Counterpane Internet Security, Inc (Perusahaan konsultan tentang kriptografi dan keamanan Komputer) dan dipublikasikan tahun 1994. Dibuat untuk digunakan pada komputer yang mempunyai microposesor besar (32-bit keatas dengan *cache* data yang besar).

Blowfish dikembangkan untuk memenuhi kriteria desain yang cepat dalam implementasinya dimana pada keadaan optimal dapat mencapai 26 *clock cycle per byte*, kompak dimana dapat berjalan pada memori kurang dari 5 KB, sederhana dalam algoritmanya sehingga akan mudah mengetahui kesalahannya dan keamanan yang variabel dimana panjang kunci bervariasi (minimum 32 bit, maksimum 448 bit, *Multiple* 8 bit, *default* 128 bit). *Blowfish* dioptimasikan untuk berbagai aplikasi dimana kunci tidak sering berubah, seperti pada jaringan komunikasi atau enkripsi file secara otomatis.

Dalam pengimplementasiannya dalam komputer ber-*micro processor* 32-bit dengan *cache* data yang besar (Pentium dan Power PC) *blowfish*

ISSN 1858-4853

terbukti jauh lebih cepat dari DES. Tetapi *blowfish* tidak cocok dengan aplikasi dengan perubahan kunci yang sering atau sebagai fungsi hast satu arah seperti pada aplikasi *packet switching*. *Blowfish* pun tidak dapat digunakan pada aplikasi kartu pintar (*smart card*) karena memerlukan memori yang besar. *Blowfish* termasuk dalam enkripsi block Cipher 64-bit dangan panjang kunci yang bervariasi antara 32-bit sampai 448-bit. Algoritma *Blowfish* terdiri atas dua bagian :

1. *Key-Expansion*

Berfungsi merubah kunci (Minimum 32-bit, Maksimum 448-bit) menjadi beberapa array subkunci (*subkey*) dengan total 4168 *byte*.

2. Enkripsi Data

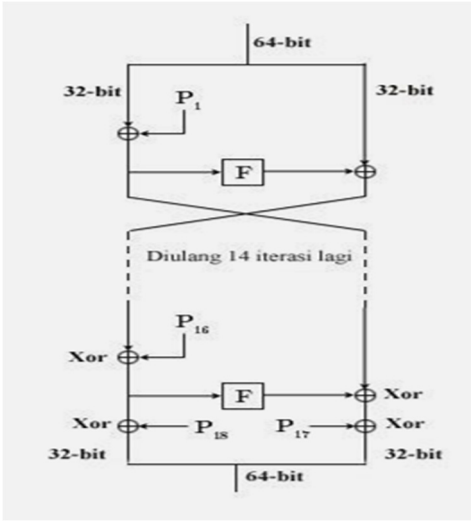
Terdiri dari iterasi fungsi sederhana (*Feistel Network*) sebanyak 16 kali putaran. Setiap putaran terdiri dari permutasi kunci-*dependent* dan substitusi kunci- dan data-*dependent*. Semua operasi adalah penambahan (*addition*) dan XOR pada variabel 32-bit. Operasi tambahan lainnya hanyalah empat penelusuran tabel (*table lookup*) array berindeks untuk setiap putaran.

c. *Algoritma Blowfish*

Blowfish menggunakan subkunci yang besar sehingga kunci tersebut harus dihitung sebelum enkripsi atau dekripsi data. *Blowfish* adalah algoritma yang menerapkan jaringan Feistel (*Feistel Network*) yang terdiri dari 16 putaran. Untuk alur algoritma enkripsi dengan metode *Blowfish* dijelaskan :

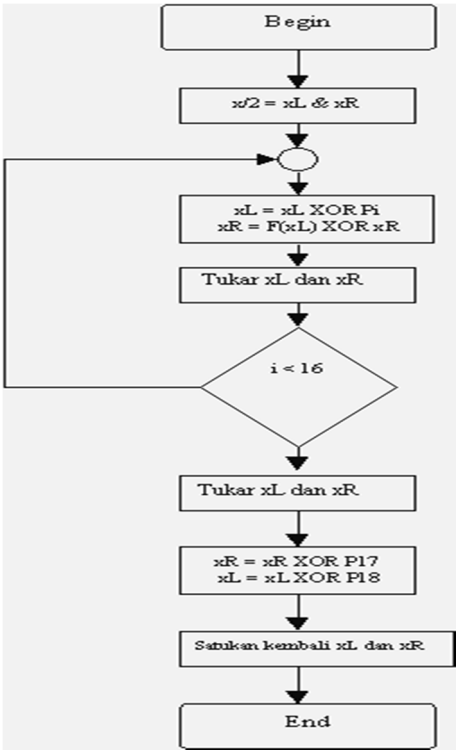
1. Bentuk inisial P-*array* sebanyak 18 buah (P1,P2,.....P18) masing-masing bernilai 32-bit.
   Array P terdiri dari delapan belas kunci 32-bit subkunci :
   $P_1,P_2,.....,P_{18}$
2. Bentuk S-*box* sebanyak 4 buah masing-masing bernilai 32-bit yang memiliki masukan 256.
   Empat 32-bit S-*box* masing-masing mempunyai 256 entri :
   $S_{1,0},S_{1,1},.....,S_{1,255}$
   $S_{2,0},S_{2,1},.....,S_{2,255}$
   $S_{3,0},S_{3,1},.....,S_{3,255}$
   $S_{4,0},S_{4,1},.....,S_{4,255}$
3. *Plaintext* yang akan dienkripsi diasumsikan sebagai masukan, *Plaintext* tersebut diambil sebanyak 64-bit, dan apabila kurang dari 64-bit maka kita tambahkan bitnya, supaya dalam operasi nanti sesuai dengan datanya.
4. Hasil pengambilan tadi dibagi 2, 32-bit pertama disebut XL, 32-bit yang kedua disebut XR.
5. Selanjutnya lakukan operasi XL = XL xor Pi dan XR = F(XL) xor XR
6. Hasil dari operasi diatas ditukar XL menjadi XR dan XR menjadi XL.
7. Lakukan sebanyak 16 kali, perulangan yang ke-16 lakukan lagi proses penukaran XL dan XR.
8. Pada proses ke-17 lakukan operasi untuk XR = XR xor P17 dan XL = XL xor P18.
9. Proses terakhir satukan kembali XL dan XR sehingga menjadi 64-bit kembali.

Blowfish menggunakan jaringan Feistel yang terdiri dari 16 buah putaran. Skema jaringan Feistel dapat dilihat di gambar 2.2 :



Gambar 1. Jaringan Feistel Algoritma *Blowfish* (Sitinjak, 2010)

Diagram alur (*flowchart*) dari algoritma *blowfish* pada gambar 2 :



Gambar 2. *Flowchart Blowfish* (Erikawati, 2010)

ISSN 1858-4853

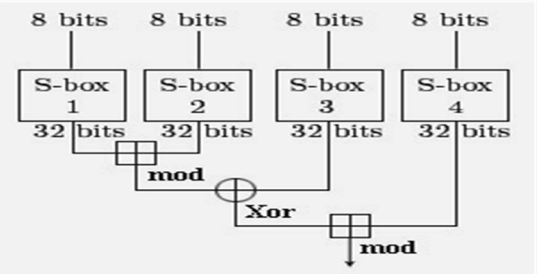
Algoritma *Blowfish* memiliki keunikan dalam hal proses dekripsi, yaitu proses dekripsi dilakukan dengan urutan yang sama persis dengan proses enkripsi, hanya saja pada proses dekripsi P1, P2, …, P18 digunakan dalam urutan yang terbalik. Dalam algoritma *Blowfish* juga terdapat fungsi F. Fungsi F adalah :

Bagi XL, menjadi empat bagian 8-bit : a,b,c dan d.

F(XL) = ((S1,a + S2,b mod 2^{32})xor S3,c) + S4,c mod 2^{32}

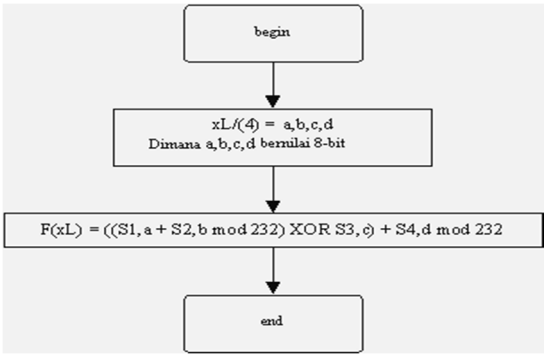
Subkunci dihitung menggunakan algoritma *blowfish*, metodenya yaitu :

1. Pertama-tama inilialisasi P-*array* dan kemudian empat S-*box* secara berurutan dengan string yang tetap. String ini terdiri atas digit hexadesimal dari Pi.
2. XOR P1 dengan 32-bit pertama kunci, XOR P2 dengan 32-bit kedua dari kunci dan seterusnya untuk setiap bit dari kunci (sampai P18). Ulangi terhadap bit kunci sampai seluruh P-array di XOR dengan bit kunci.
3. Enkrip semua string nol dengan algoritma Blowfish dengan menggunakan subkunci seperti dijelaskan pada langkah (1) dan (2).
4. Ganti P1 dan P2 dengan keluaran dari langkah (3).
5. Enkrip keluaran dari langkah (3) dengan algoritma *Blowfish* dengan subkunci yang sudah dimodifikasi.
6. Ganti P3 dan P4 dengan keluaran dari langkah (5).
7. Lanjutkan proses tersebut, ganti seluruh elemen dari P-*array*, kemudian seluruh keempat S-*box* berurutan, dengan keluaran yang berubah secara kontiyu dari algoritma *Blowfish*. Fungsi F dalam *blowfish* bisa dilihat pada gambar 3 :



Gambar 3. Fungsi F dalam *Blowfish*
(Sumber : Sitinjak, 2010)

Diagram alur *(Flowchart)* dari fungsi F bisa dilihat pada gambar 4 :



Gambar 4. *Flowchart* F Fungsi
(Sumber : Erikawati, 2010)

Total yang diperlukan adalah 521 iterasi untuk menghasilkan semua subkunci yang dibutuhkan. Aplikasi kemudian dapat menyimpan subkunci ini dan tidak membutuhkan langkah-langkah proses penurunan berulang kali, kecuali kunci yang digunakan berubah. Untuk deskripsi sama persis dengan enkripsi, kecuali pada P-array (P1,P2,.....,P18) digunakan dengan urutan terbalik atau di inverskan.

d. *Steganografi*

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 tersembunyi atau terselubung" (Ariyus, 2009). *Steganography* merupakan salah satu cabang ilmu dari *cryptography* (Ariyus, 2009). Tetapi *steganography* berbeda dengan *cryptography*, letak perbedaannya adalah pada hasil keluarannya. Hasil dari kriptografi biasanya berupa data yang berbeda dari bentuk aslinya dan biasanya data seolah-olah berantakan, sehingga tidak dapat diketahui informasi apa yang terkandung didalamnya (namun sesungguhnya dapat dikembalikan ke bentuk semula lewat proses dekripsi), sedangkan hasil keluaran dari steganografi memiliki bentuk persepsi yang sama dengan bentuk aslinya. Kesamaan persepsi tersebut adalah oleh indera manusia (khususnya visual), namun bila digunakan komputer atau perangkat pengolah digital lainnya dapat dengan jelas dibedakan antara sebelum proses dan setelah proses.

Secara umum steganografi merupakan seni atau ilmu yang digunakan untuk menyembunyikan pesan rahasia dengan segala cara sehingga selain orang yang dituju, orang lain tidak akan menyadari keberadaan dari pesan rahasia tersebut. Steganografi menggunakan sebuah berkas yang disebut dengan *cover* atau biasa disebut dengan *carrier*, tujuannya sebagai pembawa dari pesan yang dirahasiakan. Banyak format *carrier* yang dapat dijadikan media untuk menyembunyikan pesan, diantaranya, format *image* (gambar), audio, dan format lainnya (.pdf, .html, video, dan lain-lain).

ISSN 1858-4853

Dari definisi diatas, maka dapat disimpulkan bahwa steganografi dibuat untuk membantu mengamankan informasi dengan cara menyembunyikan pesan pada media gambar, audio, ataupun video, agar pihak lain tidak mengetahui keberadaan informasi rahasia tersebut, kecuali si pengirim pesan dan penerima pesan.

Terdapat beberapa istilah yang berkaitan dengan steganografi:

1. *Hiddentext* atau *embedded message*; pesan yang disembunyikan.
2. *Covertext* atau *cover-object*; pesan yang digunakan untuk menyembunyikan *embedded message*.
3. *Stegotext* atau *stego-object*; pesan yang sudah berisi *embedded message.*

e. *Steganografi End of File*

Teknik EOF atau *End Of File* merupakan salah satu teknik yang digunakan dalam steganografi. Teknik ini menggunakan cara dengan menyisipkan data pada akhir file. Teknik ini dapat digunakan untuk menyisipkan data yang ukurannya sesuai dengan kebutuhan. Ukuran file yang telah disisipkan data sama dengan ukuran file sebelum disisipkan data ditambah dengan ukuran data yang disisipkan ke dalam file tersebut. Dalam teknik ini, data disisipkan pada akhir file dengan diberi tanda khusus sebagai pengenal start dari data tersebut dan pengenal akhir dari data tersebut.

Pada sebuah citra grayscale 6x6 piksel disisipkan pesan yang berbunyi “aku”. Untuk menandai akhir pesan digunakan karakter yang jarang dipakai, misalnya karakter #. Sehingga pesan yang dimaksud adalah “#aku”.

Kode ASCII dari pesan diberikan sebagai berikut:

97 107 117 35

Misalkan matrik tingkat derajat keabuan citra :

196	10	97	182	101	40
67	200	100	50	90	50
25	150	45	200	75	28
176	56	77	100	25	200
101	34	250	40	100	60
44	66	99	125	190	200

Kode biner pesan disisipkan diakhir citra sehingga citra menjadi:

196	10	97	182	101	40
67	200	100	50	90	50
25	150	45	200	75	28
176	56	77	100	25	200
101	34	250	40	100	60
44	66	99	125	190	200
97	**107**	**117**	**35**		

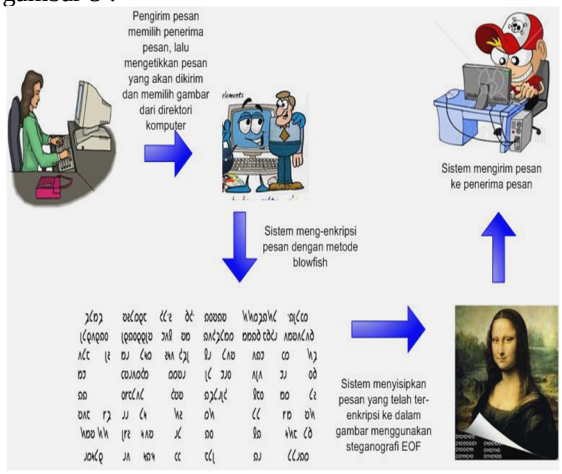
3. HASIL DAN PEMBAHASAN

a. Deskripsi Sistem

Keamanan dan penyisipan pesan rahasia pada gambar dengan enkripsi *Blowfish* dan Steganografi *End of file* berbasis web adalah suatu aplikasi yang digunakan untuk meyembunyikan pesan rahasia dibalik gambar. Aplikasi ini menggunakan algoritma steganografi *end of file*, yaitu menyisipkan pesan di akhir file berkas pembawanya yang berupa gambar, sehingga pengguna dapat menyisipkan pesan tanpa dibatasi kapasitasnya. Sebelum penyisipan, pesan rahasia diubah menjadi sandi khusus, sehingga pesan akan lebih terjaga kerahasiaannya.

Terdapat empat proses utama dalam aplikasi ini, yaitu enkripsi (penyandian pesan), *encoding* (penyisipan pesan kedalam media berupa gambar), *decoding* (pengungkapan pesan yang terdapat dalam gambar stego), dan dekripsi (penerjemahan pesan yang masih terenkripsi). Setelah melalui proses enkripsi dan *encoding*, maka pesan rahasia bisa dikirim ke pengguna lain yang telah terdaftar dalam *database*. Kemudian pesan rahasia akan melalui proses *decoding* dan dekripsi agar dapat dibaca oleh penerima pesan.

Tahapan proses dari sistem ditunjukkan pada gambar 5 :



Gambar 5. Deskripsi Tahapan Sistem

Seperti aplikasi lainnya, aplikasi pesan rahasia ini juga mempunyai aturan-aturan dalam penggunaanya. Aturan-aturan yang digunakan dalam aplikasi ini adalah :

1. Pengguna (pengirim dan penerima pesan) merupakan pengguna yang telah melakukan proses pendaftaran atau registrasi dan datanya berhasil tersimpan dalam *database*.
2. Pengguna hanya memiliki hak untuk mengirim dan menerima pesan antar sesama pengguna.
3. Administrator, selain memiliki hak yang sama dengan pengguna biasa, yaitu dapat mengirim dan menerima pesan, administrator juga bertugas untuk mengatur jalannya aplikasi dan membantu pengguna untuk menyelesaikan masalahnya, misalnya mengatur ulang

password pengguna. Administrator memiliki akses penuh dalam mengelola sistem.

b. Perancangan Sistem

Perancangan analisis sistem pada keamanan dan penyisipan pesan rahasia pada gambar dengan enkripsi *blowfish* dan steganografi *end of file* ini menggunakan *Data Flow Diagram* (DFD) dan *Flowchart*. DFD yang digunakan pada sistem ini terdiri dari 3 diagram, yakni Diagram Konteks, *Data Flow Diagram* Level 1, dan *Data Flow Diagram* Level 2.

- **Diagram Konteks**

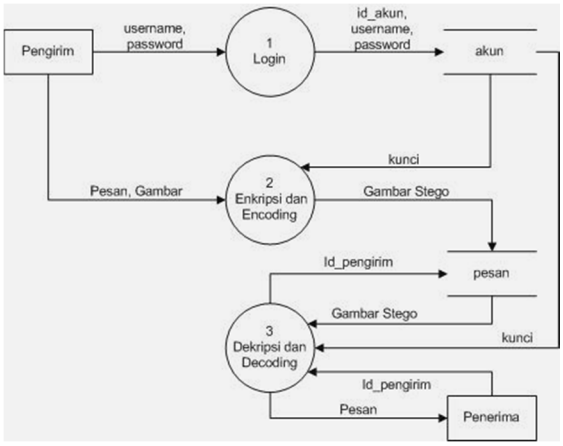
Dalam mengembangkan sistem ini dibuat sebuah diagram konteks yang menjelaskan bahwa pengirim dapat mengirim pesan rahasia dibalik gambar dengan menginputkan pesan berupa teks serta memilih gambar yang akan disisipkan pesan tersebut..

Gambar 6. Diagram Konteks

Setelah diproses oleh sistem, penerima akan menerima gambar stego (gambar yang telah disisipkan pesan) dan dapat membaca pesan yang diekstrak dari gambar tersebut

- **Data Flow Diagram Level 1**

Pertama harus *login* terlebih dahulu dengan menginputkan *username* dan *password*. Setelah pengirim berhasil melakukan proses *login*, maka pengirim dapat melakukan proses kedua dalam satu waktu, yaitu penyandian (enkripsi) dan penyisipan (*encoding*) pesan. Pengirim menginputkan pesan berupa teks kemudian memilih gambar yang selanjutnya akan melalui proses enkripsi dan *encoding* dengan bantuan kunci dari *database user*. Setelah melewati proses enkripsi dan *encoding*, semua inputan dari pengirim akan tersimpan dalam *database* pesan dalam bentuk gambar stego (gambar yang telah disisipkan pesan).



Gambar 6. *Data Flow Diagram* Level 1

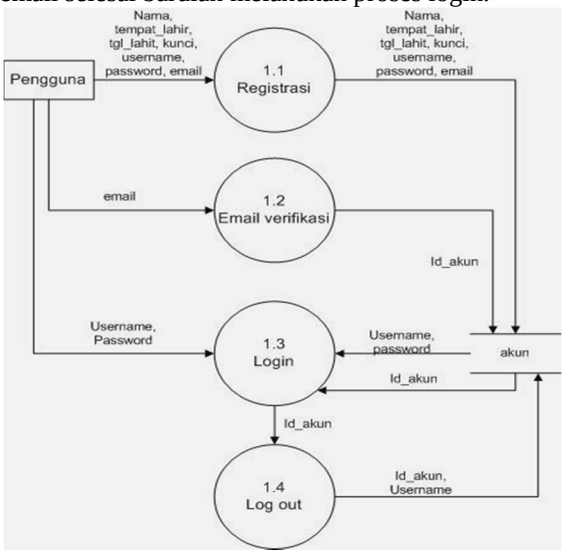
Penerima membuka gambar stego yang diterimanya dengan menggunakan proses *decoding* dan dekripsi. Sistem membaca pada *database user*, yang kemudian diproses untuk mengekstrak pesan dari dalam gambar stego.

- **Data Flow Diagram Level 2**

Pada tahap ini terdapat 3 bagian yaitu :

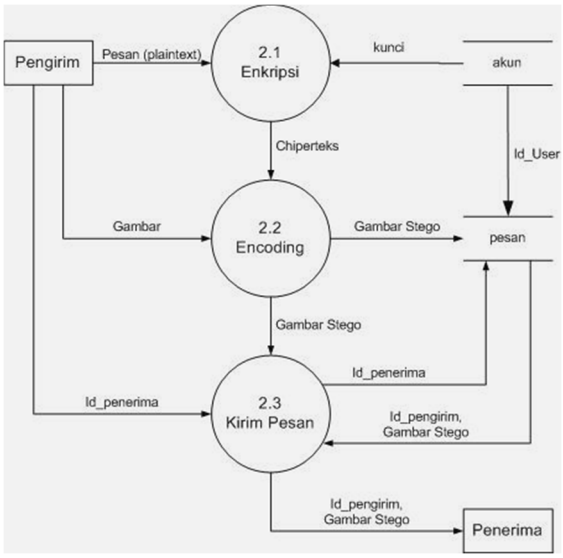
1. Proses Registrasi dan *Login.*
2. Proses Enkripsi dan *Encoding.*
3. Proses Dekripsi dan *Decoding.*

Pada proses pertama yang ditunjukan pada gambar 7, pengguna harus melakuka proses registrasi dengan cara menginputkan *emai*, nama lengkap, tempat dan tanggal lahir, kunci (menginputkan kata sesuai keinginan), *username* (nama yang akan digunakan untuk *login*) dan *password* untuk disimpan kedalam *database user*. Setelah proses registrasi dan melakukan verifikasi email selesai barulah melakukan proses login.



Gambar 7. *Data Flow Diagram* Level 2 Proses 1

Proses kedua adalah proses enkripsi dan *encoding* ditunjukan pada gambar 8.

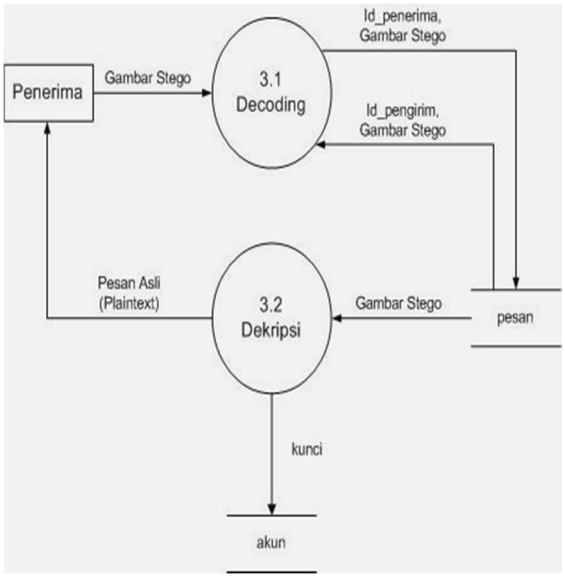


Gambar 8. *Data Flow Diagram* Level 2 Proses 2

ISSN 1858-4853

Setelah pengguna telah masuk pada aplikasi keamanan dan penyisipan pesan rahasia ini, Pengguna selaku pengirim menginputkan nama penerima pesan kedalam proses pengiriman pesan. Lalu, pengirim pesan menginputkan *plaintext* (pesan yang akan disisipkan), kemudian sistem mengambil kunci, tempat lahir dan tanggal lahir dari pengirim dan penerima pesan pada *database user*. Dimana kunci, tempat lahir dan tanggal lahir ini akan dikombinasikan sebagai kunci pesan untuk proses enkripsi atau penyandian pesan. Sistem akan membaca *plaintext* dan kunci pesan kemudian mengubah tiap-tiap karakter dari *plaintext* tersebut menjadi *ciphertext*. *Ciphertext* ini akan disisipkan kedalam gambar melalui proses *encoding* dan akan menghasilkan gambar stego atau gambar yang telah disisipkan pesan rahasia. Kemudian sistem akan mengirimkan pesan ke penerima.

Pada proses ketiga, proses dekripsi dan *decoding* merupakan proses kebalikan dari proses enkripsi dan decoding yaitu mengembalikan pesan pada bentuk asalnya (*plaintext),* ditunjukan pada gambar 9.

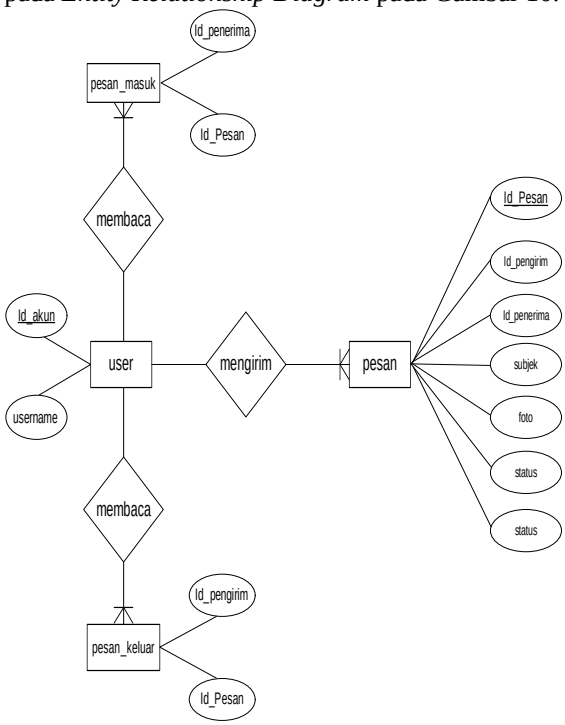


Gambar 9. *Data Flow Diagram* Level 2 Proses 3

pengguna atau penerima pesan harus melakukan *login* untuk dapat mengakses portal dan membaca pesan masuk. Gambar stego dibaca melalui proses *decoding*, yaitu proses pemisahan antara gambar dan pesan yang disisipkan di dalamnya. Proses *decoding* menghasilkan pesan yang masih ter-enkripsi (*ciphertext)* dan kemudian *ciphertext* itu harus melalui proses dekripsi untuk mengembalikan pesan yang teracak tersebut agar kembali kedalam bentuk aslinya sehingga pesan tersebut dapat dibaca oleh penerima pesan.

c. Perancangan Database

Database atau basis data merupakan sekumpulan informasi yang berguna, yang diorganisasikan dalam bentuk yang spesifik. Pada tahapan perancangan basis data, sistem ini menggunakan 3 tabel yang meliputi tabel akun, tabel pesan, tabel artikel. Tabel – tabel pada *database* sistem ini memiliki relasi atau hubungan. Hubungan antara tabel – tabel tersebut dapat dilihat pada *Entity Relationship Diagram* pada Gambar 10.



Gambar 10. *Entity Realtionship Diagram*

Rincian tabel-tabel yang digunakan pada Keamanan dan Penyisipan Pesan Rahasia Pada Gambar Dengan Enkripsi *Blowfish* dan Steganografi *End of File* :

1. Akun

Tabel akun yang ditunjukkan pada tabel 4.1 digunakan untuk menyimpan data akun pengguna aplikasi ini, tetapi akun memiliki 2 tipe akun yang berbeda, yaitu pengguna biasa dan pengguna administrator. Pengguna biasa memiliki hak akses terbatas, sedangkan pengguna administrator memiliki hak akses penuh. Nama Tabel : akun *Primary Key* : id_akun Jumlah Kolom : 12

Tabel 1. Tabel Akun

Nama Kolom	Tipe Data	Panjang
id_akun	varchar	10
tipe	enum	
nama	varchar	50
tempat_lahir	varchar	50
tgl_lahir	date	
kunci	varchar	200
username	varchar	20
password	varchar	100
email	varchar	100
status	enum	
token	varchar	20
tanggal	timestamp	

2. Pesan

Tabel pesan yang ditunjukkan pada tabel 4.2 digunakan untuk menyimpan data pesan.
Nama Tabel : pesan *Primary Key* : id_pesan
Jumlah Kolom : 9

Tabel 2. Tabel Pesan

Nama Kolom	Tipe Data	Panjang
id_pesan	Varchar	15
id_pengirim	Varchar	15
id_penerima	Varchar	15
subjek	Varchar	200
Foto	Varchar	50
status	Enum	
tanggal	timestamp	
pengirim_hapus	Enum	
penerima_hapus	Enum	

3. Artikel

Tabel artikel yang ditunjukkan pada tabel 4.3 digunakan untuk menyimpan data artikel yang dibuat oleh administrator.
Nama Tabel : artikel *Primary Key* : id_artikel
Jumlah Kolom : 5

Tabel 3. Tabel Artikel

Nama Kolom	Tipe Data	Panjang
id_artikel	Varchar	20
judul	Varchar	200
isi	Text	
tgl	timestamp	
foto	varchar	50

d. Implementasi dan Pengujian Sistem

Implementasi Sistem merupakan tahapan realisasi setelah rancangan aplikasi. Pengujian sistem merupakan tahapan yang terdapat pada sistem diuji dan dievaluasi.

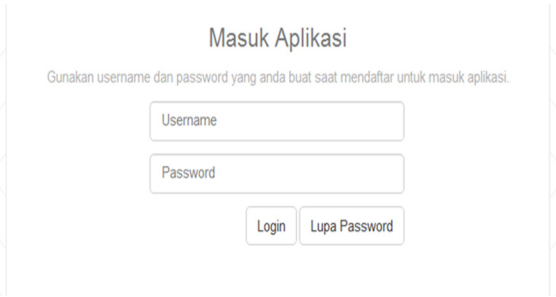
- **Sistem *User Login***

Pertama pengguna harus melakukan proses registrasi ditunjukan pada gambar 11.

Gambar 9. *Form* Registrasi

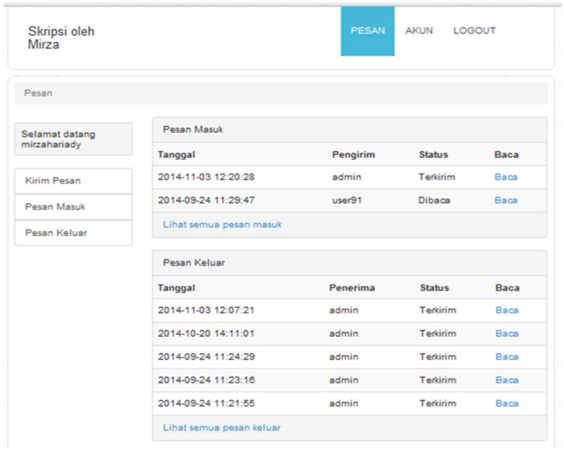
Setelah mengisi segala kolom lalu menekan tombol daftar, sehingga data akan tersimpan di *database.* Setelah registrasi selesai, pengguna harus melakukan verifikasi alamat *email.*

Pada proses login, pengguna diminta untuk memasukkan *username* dan *password* yang valid.



Gambar 10. *Form Login*

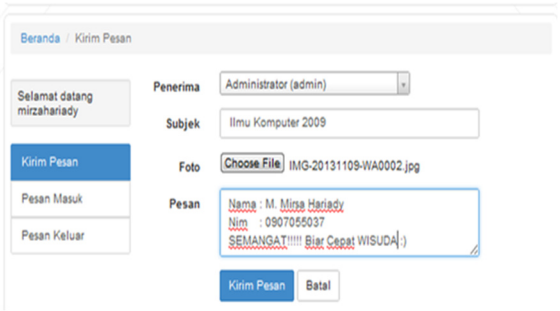
Ketika pengguna berhasil melakukan proses *login*, maka halaman pertama yang akan muncul pada portal adalah halaman pesan yang berisi pesan masuk dan pesan keluar, seperti pada gambar 11.



Gambar 11. *Form* Portal Pesan

- **Kirim Pesan dan Baca Pesan**

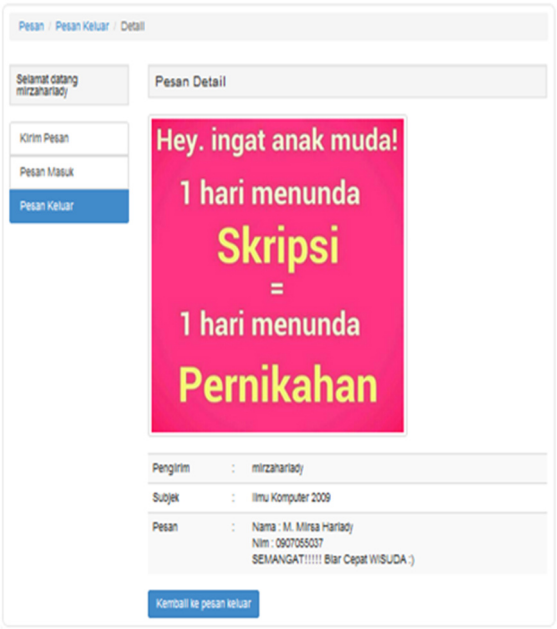
Pada proses mengirim pesan, pengguna harus memasukan alamat penerima, subjek, foto dan pesan. Kunci otomatis diproses oleh sistem dari gabungan kata kunci, tanggal lahir dan tempat lahir antara pengirim dan penerima.



Gambar 12. *Form* Kirim Pesan

ISSN 1858-4853

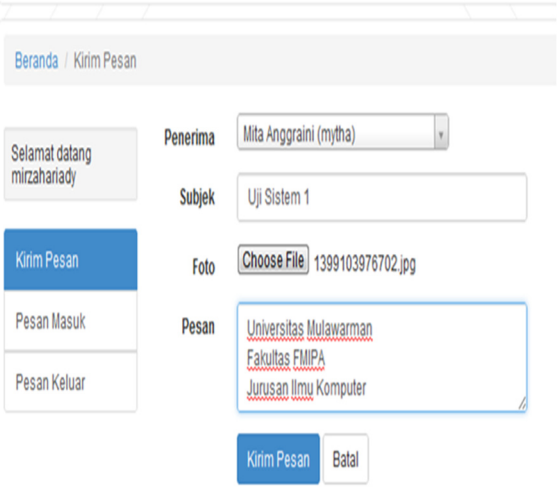
Setelah semua kolom terisi dan pesan terisi dan pesan telah dikirim ke penerima, maka penerima bisa membukanya di pesan masuk dan membuka detail pesan.



Gambar 13. *Form* Detail Pesan

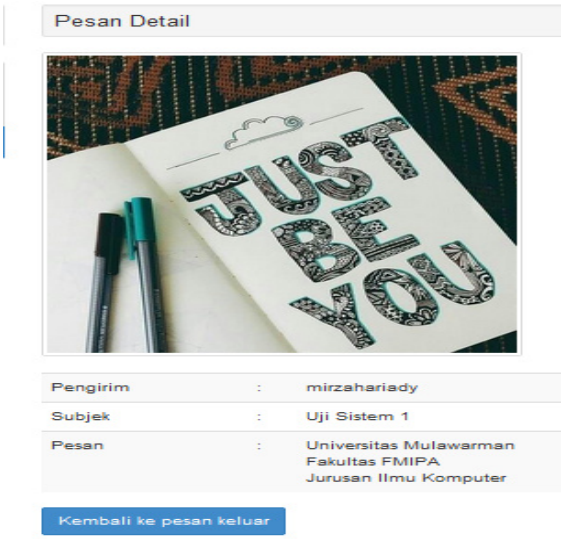
- **Pengujian Huruf Besar dan Kecil**

Pada pengujian ini dicoba meng-enkripsi huruf besar dan kecil, seperti yang ditunjukkan pada gambar 14.



Gambar 14. *Form* Uji Huruf Besar dan Kecil

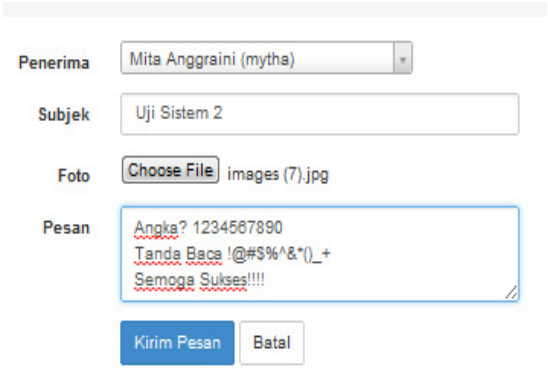
Pesan yang dikirim dengan menggunakan huruf besar dan kecil tidak mengalami masalah, seperti yang ditunjukkan pada gambar 15.



Gambar 15. *Form* Hasil Uji Huruf Besar dan Kecil

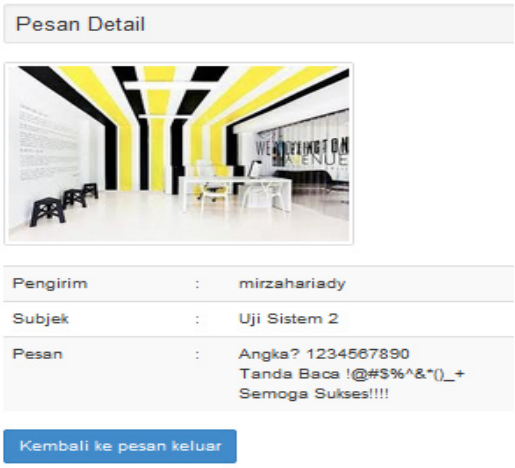
- **Pengujian Tanda Baca dan Angka**

Penggunaan tanda baca dan angka dapat digunakan pada aplikasi pesan rahasia seperti yang ditunjukkan pada gambar 16 dan 17.



Gambar 16. *Form* Uji Tanda Baca dan Angka

Tanda baca terbaca dengan baik, sehingga pesan yang tersampaikan tidak berubah sedikit pun.



Gambar 17. *Form* Hasil Uji Tanda Baca dan Angka

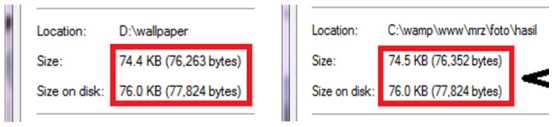
- **Perbandingan Gambar Asli dan Gambar Stego**

Proses penyisipan pesan pada aplikasi keamanan dan penyisipan pesan rahasia menggunakan metode penyisipan pada akhir file (*End Of File*). Dengan metode EOF, tidak banyak terjadi perubahan pada gambar yang belum disisipkan pesan (gambar asli) dengan gambar yang telah disisipkan pesan (gambar stego).

Gambar tidak mengalami perubahan dari kualitas ketajaman warna, seperti ditujukkakan pada gambar 18, tetapi mengalami perubahan pada ukurannya sesuai dengan besar pesan yang disisipkan, seperti yang ditunjukkan pada gambar 19.

Gambar 18. Perbandingan Kualitas Gambar Asli (kiri) dengan Gambar Stego (kanan)

Perbedaan ukuran gambar asli dengan gambar stego disebabkan oleh pesan yang disisipkan pada akhir file gambar stego sehingga gambar stego berkapasitas lebih besar dari kapasitas gambar aslinya.



Gambar 18. Perbandingan Ukuran Gambar Asli (kiri) dengan Gambar Stego (kanan)

4. KESIMPULAN

Berdasarkan hasil penelitian dan implementasi sistem keamanan dan penyisipan pesan dapat diambil kesimpulan :

1. Penelitian ini menghasilkan aplikasi keamanan dan penyisipan pesan rahasia pada gambar dengan enkripsi *blowfish* dan steganografi *end of file*.
2. Pengguna dari sistem ini dapat melakukan pengamanan informasi rahasia yang sebelumya di-enkripsi terlebih dahulu, kemudian disembunyikan didalam gambar sehingga pesan lebih terjamin kerahasiaannya.
3. Proses penyandian pesan dilakukan dengan metode *blowfish* yang dimana sampai saat ini belum ada Cryptanalysis yang dapat membongkar pesan tanpa kunci yang enkripsi oleh Blowfish.
4. Proses penyisipan pesan dengan menggunakan metode *end of file* menguntungkan bagi pengguna karena tidak memberikan batasan kapasitas untuk pesan yang akan disisipkan pada file gambar, tetapi akan mengalami perubahan pada ukuran gambar sesuai pesan yang disisipkan.
5. Kualitas dari gambar yang disisipi oleh pesan tidak berubah.

SARAN

Berdasarkan hasil penelitian, sangat disadari bahwa masih banyaknya kekurangan dan kelemahan. Saran yang dapat diberikan adalah :

1. Sistem ini tidak dapat mendekripsi pesan pada gambar yang telah disisipi pesan sebelumnya, sehingga diperlukan cara lain untuk memisahkan antara pesan lama dan pesan baru yang disisipkan didalam gambar.
2. Gambar stego yang dihasilkan mengalami sedikit perubahan pada ukuran file, sehingga diperlukan metode lain untuk mengatasi perubahan ukuran tersebut.
3. Sistem ini dapat dikembangkan menggunakan metode lain dan dapat dijadikan model untuk pengembangan sistem yang lebih baik lagi. Sehingga kelemahan algoritma yang digunakan semakin sedikit atau bila mungkin dihilangkan.

5. DAFTAR PUSTAKA

[1] Ariyus, D. 2006. *Computer Security*. Yogyakarta: Andi.
[2] Ariyus, D. 2009. *Keamanan Multimedia*. Yogyakarta: Andi.
[3] Erikawati, S. 2010. “Implementasi Algoritma Kriptografi Blowfish Untuk Keamanan Pada Dokumen Pada Microsoft Office”. Tugas Akhir Jurusan Teknik Informatika, STIMIK Amikom, Yogyakarta.
[4] Fidens, F. 2006. “Dasar Kriptografi”. Kuliah umum Ilmukomputer.com
[5] Krisnawati. 2008. “Metode Least Significant Bit (LSB) dan End Of File (EOF) untuk Menyisipkan Teks ke dalam Citra Grayscale”. Makalah disajikan pada Seminar Nasional Informatika, Yogyakarta.
[6] Munir, Rinaldi. 2006. *Kriptografi*. Bandung : Informatika.
[7] Sitinjak, Suriski. 2010. “Aplikasi Kriptografi File Menggunakan Algoritma Blowfish”. Skripsi UPN Veteran Yogyakarta.