PERANCANGAN ENKRIPSI PADA CITRA BITMAP DENGAN ALGORITMA DES, TRIPLE DES, DAN IDEA

Agustinna Yosanny

Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Bina Nusantara University
Jln. K.H. Syahdan No. 9, Palmerah, Jakarta Barat 11480
ayosanny@binus.edu

ABSTRACT

Rapid development of Information Technology causes information can access easier without space boundaries. Image is a form of information, which contains many information. Some image contains confidential information that cannot distribute to unauthorized persons. Therefore, image encryption application is create to encrypt part of image that has confidential information. The image encryption application is encrypt image by applying DES, Triple DES, and IDEA algorithms. The research was applying analysis and design methodology. The analysis methodology was undertaken through literature study and algorithm research and testing. The design methodology was undertaken through database, features, system, and screen layout design. Results of the research are image encryption application that can encrypt part of image and or all of image by applying three algorithms such as DES, Triple DES, and IDEA. This application is show comparation of these algorithms. In conclusion, method of encryption can apply to image, so that confidential information of the image can protect from unauthorized person.

Keywords: *image, encryption, DES, triple DES, IDEA.*

ABSTRAK

Perkembangan teknologi informasi yang pesat dewasa ini menyebabkan semakin mudahnya informasi diakses tanpa adanya batasan ruang. Salah satu bentuk informasi adalah gambar (citra/image), dimana sebuah citra dapat mengandung banyak informasi. Oleh karena itu diperlukan suatu program aplikasi yang dapat menyembunyikan bagian-bagian dari citra yang bersifat rahasia. Pada penulisan ini membahas enkripsi sebagian pada citra dengan menggunakan algoritma enkripsi DES, Triple DES dan IDEA. Metode yang digunakan dalam penyajian skripsi ini adalah metode analisis dan perancangan. Metode analisa berupa studi pustaka dan melalukan penelitian terhadap algoritma DES, Tripe DES, dan IDEA. Metode perancangan berupa perancangan database, fitur, sistem, dan layar. Hasil dari penelitian ini adalah sebuah program aplikasi yang dapat mengenkripsi sebagian atau seluruh citra dengan tiga buah algoritma, yaitu: DES, Triple DES dan IDEA. Dari hasil penelitian dapat disimpulkan bahwa teknik enkripsi dapat diterapkan pada citra untuk melindungi bagian dari citra tersebut.

Kata kunci: *citra, enkripsi, DES, triple DES, IDEA*

PENDAHULUAN

Perkembangan teknologi informasi yang semakin pesat membuat data dan informasi semakin mudah diaskses. Bentuk-bentuk informasi yang ada dapat berupa teks, citra (*image*), suara, *executable file*, video, dan lain sebagainya. Namun adakalanya, perlu untuk menjaga kerahasiaan data dan informasi dari pihak-pihak yang tidak berwenang baik untuk alasan komersial maupun keamanan. Oleh karena itu, maka diperlukan teknik untuk menjaga kerahasiaan tersebut yang dikenal dengan teknik enkripsi.

Enkripsi memiliki bermacam-macam teknik dimana setiap teknik memiliki keunggulan dan kekurangan masing-masing, namun unsur yang paling penting dalam teknik enkripsi adalah bahwa informasi yang telah dienkripsi atau disembunyikan harus dapat didekripsi atau dibuka kembali tanpa merusak keutuhan dari informasi tersebut, atau dengan kata lain teknik enkripsi tersebut harus aman digunakan. Unsur lain yang tidak kalah pentingnya adalah bahwa teknik enkripsi tersebut harus memiliki ketahanan yang tinggi terhadap tindakan-tindakan yang bertujuan untuk memecahkan kode enkripsi tersebut.

Pada suatu keadaan tertentu, informasi yang ingin disembunyikan mungkin terintegrasi dengan informasi yang ingin dipublikasikan, salah satu bentuk informasi seperti ini adalah citra. Sebuah citra dapat mengandung banyak sekali informasi dan terkadang terdapat informasi-informasi (bagian dari citra) yang rahasia dan perlu disembunyikan. Oleh karena itu, perlunya teknik enkripsi yang dapat menyembunyikan baik sebagian maupun keseluruhan informasi dari citra tersebut dari pihak-pihak yang tidak berwenang.

Tujuan dari pengembangan aplikasi ini adalah rancangan dan analisa enkripsi pada citra (i*mage*) bitmap dengan menggunakan algoritma DES, *Triple* DES dan IDEA. Sedangkan Manfaat yang ingin diperoleh adalah memberikan solusi dalam menyembunyikan sebagian atau seluruh citra (*image*) yang didistribusikan melalui jaringan baik intranet maupun internet untuk alasan komersial atau keamanan.

Uraian Pustaka

Citra

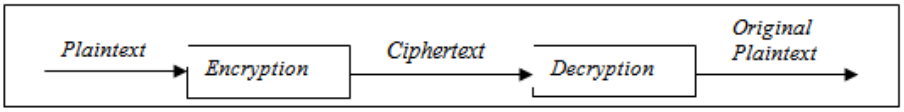
Citra adalah representasi visual yang terdiri dari sekumpulan *pixel* atau titik berwarna dalam bentuk dua dimensi. Menurut Gonzalez (2008, p.1), citra merupakan representasi dua dimensi (2-D) dari intensitas cahaya yang dinyatakan dengan fungsi *f(x,y)*, dimana *x* dan *y* merupakan koordinat spasial dan nilai fungsi *f* menunjuk pada titik *(x,y)*. *Pixel* adalah unit dasar dari *digital image*. Pada citra berformat *bit*map, *pixel-pixel* adalah titik-titik yang digunakan untuk membangun suatu citra. *Pixel* terdiri dari tiga komponen yaitu R (*Red*), G (*Green*) dan B (*Blue*), yang masing-masing merupakan warna dasar cahaya.

Kualitas sebuah citra ditentukan oleh resolusi (banyaknya *pixel* yang menghasilkan sebuah citra dalam sebuah layar atau *printer*). Semakin banyak jumlah *pixel*-nya, maka semakin tinggi resolusinya dan akan dihasilkan citra yang lebih baik dan lebih halus. Resolusi yang ideal merupakan keseimbangan antara kualitas dengan ukuran penyimpanan citra tersebut.

Kriptografi

Menurut Schneiner (1996), *cryptography* adalah seni dan ilmu dalam mengamankan pesan. Dalam dunia kriptografi, pesan disebut *plaintext* atau *cleartext*. Proses untuk menyamarkan pesan

dengan cara sedemikian rupa untuk menyembunyikan isi aslinya disebut enkripsi. Pesan yang telah dienkripsi disebut *ciphertext*. Proses Pengembalian sebuah *ciphertext* ke *plaintext* disebut dekripsi.



Gambar 1 Konsep Dasar dari Enkripsi dan Dekripsi (Schneier, 1996)

Dalam algoritma enkripsi diperlukan kunci yang merupakan kode yang digunakan untuk mengenkripsi maupun mendekripsi. Panjang kunci dan algoritma yang kuat merupakan dua hal yang penting untuk tetap menjaga kerahasiaan.

DES

DES adalah *cipher* blok yang mengenkripsi data dalam blok 64-*bit*. Sebuah blok 64-*bit* dari *plaintext* sebagai *input* ke dalam algoritma tersebut akan menghasilkan blok 64-*bit ciphertext*. Untuk proses enkripsi dan dekripsi menggunakan algoritma yang sama kecuali dalam pengaturan kunci. Panjang kunci yang digunakan adalah 56-*bit*, namun panjang kunci sebenarnya yang dimasukkan adalah 64-*bit* karena setiap *bit* kelipatan 8 tidak digunakan dalam algoritma namun hanya digunakan untuk *parity check*. Kunci dapat berupa angka 56-*bit* apa saja dan dapat diubah sewaktu-waktu.

Pada level yang paling sederhana, algoritma DES dapat dikatakan hanya berupa kombinasi dari 2 teknik dasar enkripsi, konfusi dan difusi. Dasar dari pembentukan blok dari DES adalah kombinasi tunggal dari teknik-teknik ini yang berupa sebuah subsitusi yang diikuti oleh sebuah permutasi terhadap *plaintext* yang didasarkan pada kunci. Ini dikenal sebagai sebuah *round*. Sebuah DES memiliki 16 *round*, yang melakukan kombinasi teknik yang sama terhadap *plaintext* selama 16 kali.

Triple DES

Triple DES menggunakan dua buah kunci dalam tiga proses pelaksanaan untuk algoritma DES dapat dilihat pada gambar di bawah ini. Fungsi ini menggunakan *multiple* enkripsi EDE (*Encrypt – Decrypt – Encrypt*) berurutan. Rumusan algoritma *Triple* DES adalah:

$C = E_{k1} [D_{k2} [E_{k1} [P]]]$

Algoritma EDE pada *Triple* DES ini menggunakan dua kunci (k_1 dan k_2) dimana k_1 digunakan pada proses enkripsi pertama, lalu hasil enkripsi ini didekripsi dengan menggunakan k_2 dan yang terakhir dilakukan enkripsi kembali dengan menggunakan kunci yang dipakai pertama kali untuk enkripsi, yaitu k_1. Algoritma EDE ini menggunakan prinsip kerja yang sama dengan DES.

IDEA

Algoritma IDEA menggunakan kunci 128-bit dan secara umum algoritma ini sangat aman. Proses enkripsi dari algoritma IDEA menggunakan kunci 52-*subkey*, enam untuk delapan putaran dan empat lainnya untuk transformasi *output*. Mula-mula 128-*bit* kunci dibagi menjadi 8-*subkey* masing-masing berukuran 16-*bit*. Kedelapan *subkey* awal akan digunakan oleh algoritma tersebut dimana enam digunakan untuk putaran pertama dan dua yang ada di awal akan digunakan untuk perulangan kedua. Lalu kunci tersebut dirotasi 25-*bit* ke kiri dan dilakukan pembagian lagi menjadi delapan *subkey*. Empat yang ada diawal digunakan pada perulangan dua, empat yang di belakang digunakan

pada perulangan tiga. Lalu kunci tersebut dirotasi lagi 25-*bit* ke kiri untuk mendapatkan 8-*subkey* berikutnya, dan seterusnya sampai akhir dari algoritma.

Kumpulan data 64-*bit* dibagi menjadi empat bagian yang disebut sebagai *sub-block* dengan ukuran 16-*bit* yaitu : X_1, X_2, X_3, dan X_4. Empat *sub-block* tersebut menjadi *input* pada perulangan pertama dari algoritma kesemuanya terdiri dari delapan putaran. Dalam tiap perulangan keempat *sub-block* tadi mengalami operasi XOR, kemudian dijumlahkan, dan dikalikan antara yang satu dengan yang lainnya dan dengan enam kunci bagian (*subkey*) yang berukuran 16-*bit*. Di sela-sela masing-masing perulangan, *sub-block* kedua dan ketiga ditukar. Akhirnya, keempat *sub-block* tadi dikombinasikan dengan keempat *subkey* dalam suatu transformasi *output*.

Tahap selanjutnya adalah memproses data dengan gabungan beberapa operasi aljabar. Operasi ini terdiri dari delapan perulangan dimana urutan operasinya adalah sebagai berikut: (1)Mengalikan X_1 dengan *subkey* pertama, dengan modulo $2^{16}+1$, (2)Menjumlahkan X_2 dengan *subkey* kedua, dengan modulo 2^{16}, (3)Menjumlahkan X_3 dengan *subkey* ketiga, dengan modulo 2^{16}, (4)Mengalikan X_4 dengan *subkey* keempat, dengan modulo $2^{16}+1$, (5)Melakukan operasi XOR dari hasil langkah pertama dan ketiga, (6)Melakukan operasi XOR dari hasil langkah kedua dan keempat, (7)Mengalikan hasil dari langkah kelima dengan *subkey* kelima, dengan modulo $2^{16}+1$, (8)Menjumlahkan hasil dari langkah keenam dan ketujuh, dengan modulo 2^{16}, (9)Mengalikan hasil dari langkah kedelapan dengan *subkey* keenam, dengan modulo $2^{16}+1$, (10)Menjumlahkan hasil dari langkah ketujuh dan kesembilan, dengan modulo 2^{16}, (11)Melakukan operasi XOR dari hasil langkah kesatu dan kesembilan, (12)Melakukan operasi XOR dari hasil langkah ketiga dan kesembilan, (13)Melakukan operasi XOR dari hasil langkah kedua dan kesepuluh, dan (14)Melakukan operasi XOR dari hasil langkah keempat dan kesepuluh.

Hasil dari perulangan ini adalah empat *sub-block* yang dihasilkan oleh langkah ke-11, 12, 13, dan 14. Tukar dua blok yang ada di sebelah dalam kecuali untuk putaran terakhir dan hasilnya akan menjadi *input* bagi perulangan berikutnya.

Setelah melewati delapan perulangan, maka akan dilakukan transformasi *output* yang terakhir yaitu: (1)Mengalikan X_1 dengan *subkey* pertama, dengan modulo $2^{16}+1$, (2)Menjumlahkan X_2 dengan *subkey* ketiga, dengan modulo 2^{16}, (3)Menjumlahkan X_3 dengan *subkey* kedua, dengan modulo 2^{16}, dam (4)Mengalikan X_4 dengan *subkey* keempat, dengan modulo $2^{16}+1$. Akhirnya, keempat *sub-block* tersebut digabungkan kembali sehingga menghasilkan teks yang sudah disandi rahasiakan atau disebut dengan *ciphertext*.

METODE

Metode yang digunakan dalam penerapan algoritma enkripsi pada citra ini adalah metode pustaka, analisis, dan perancangan. Metode yang digunakan dalam penelitian ini meliputi empat tahapan. Pertama adalah analisa kebutuhan user. Pada tahap ini dilakukan studi pustaka dan penelitian terhadap ketiga algoritma yang digunakan, yaitu DES, Triple DES, dan IDEA. Kedua adalah metode perancangan. Dari hasil analisa kebutuhan *user*, maka dibuatlah perancangan aplikasi sebagai program enkripsi sebagian atau keseluruhan citra dengan menerapkan algoritma DES, Triple DES, dan IDEA. Dalam proses perancangan aplikasi ddilakukan perancangan-perancangan seperti perancangan fitur, perancangan sistem, perancangan database, dan perancangan layar dari aplikasi yang diusulkan.

Ketiga adalah metode implementasi. Implementasi dilakukan untuk mendapatkan *feedback* dan perbaikan *bug* yang mungkin muncul saat implementasi. Keempat adalah tahap evaluasi. Pada

tahap ini dilakukan evaluasi terhadap aplikasi dari segi usability dan fitur aplikasi supaya mendapatkan hasil yang memuaskan dan aplikasi dapat digunakan.

HASIL DAN PEMBAHASAN

Citra (*image*) merupakan salah satu bentuk informasi yang sudah banyak digunakan saat ini. Untuk itu diperlukan suatu teknik untuk menjaga keamanan dalam pendistribusian maupun publikasi citra tersebut dari pihak-pihak yang tidak berkepentingan. Teknik yang digunakan adalah teknik enkripsi. Teknik ini dapat langsung diterapkan pada *file* citra secara fisik sehingga *file* citra tersebut tidak dapat dibuka oleh pihak penyunting citra manapun. Masalah baru lainnya yang dihadapi adalah dalam sebuah citra mungkin mengandung lebih dari sebuah informasi yang penting dan ingin dirahasiakan dari pihak-pihak yang tidak berwenang. Semakin meningkatnya teknologi komputer saat ini memungkinkan banyak algoritma enkripsi yang dulu dikira aman dapat dipecahkan oleh komputer yang semakin meningkat kecepatannya, dan sama sekali tidak ada jaminan bahwa algoritma enkripsi yang sekarang dikira aman dan tidak dapat dipecahkan untuk tahun-tahun mendatang.

Berikut ada beberapa aplikasi pengenkripsian citra yang telah beredar yaitu ImageCrypt (sebuah aplikasi enkripsi pada citra, dapat mengenkripsi citra dengan tipe apa saja, namun *file* yang telah dienkripsi tidak dapat dibuka oleh *image editor*), CryptBMP (sebuah aplikasi enkripsi pada citra bitmap, *file* yang telah dienkripsi dapat dilihat oleh citra *editor*, namun hanya dapat mengenkrip seluruh citra), CryptJPG (sebuah aplikasi enkripsi pada citra JPEG, *file* yang telah dienkripsi tidak dapat dibuka oleh *image editor*), XFORM (sebuah aplikasi enkripsi citra bitmap, citra yang dienkrip harus berukuran bujur sangkar, *file* citra yang telah dienkrip tidak dapat dibuka oleh *image editor*), dan Fitin (sebuah aplikasi enkripsi pada citra GIF, citra dapat dienkrip sebagian, dan dapat dibuka pada *image editor*, namun pendekripan antara bagian-bagian citra yang dienkripsi tidak independen, dengan kata lain, citra yang terakhir dienkripsi harus didekripsi terlebih dahulu). Aplikasi-aplikasi tersebut masih terdapat kelemahan-kelemahan yang tidak dapat menyelesaikan masalah yang telah disebutkan sebelumnya, seperti kemampuan enkripsi sebagian pada citra, *file* yang dirusak sehingga tidak dapat dibuka oleh *image editor*, dan tidak *independen* antara bagian-bagian citra yang dienkripsi.

Untuk memecahkan masalah penyembunyian citra tersebut, maka dibuat suatu aplikasi yang dapat menyembunyikan sebagian atau keseluruhan citra dengan menggunakan teknik enkripsi yang sudah standar. Teknik enkripsi ini harus dapat mengenkripsi sebagian atau seluruh bagian citra yang dianggap penting dan bersifat rahasia, karena ada citra yang tidak seluruh bagiannya bersifat rahasia. Setiap melakukan enkripsi harus disertai dengan penggunaan kunci yang berbeda-beda dan penggunaan teknik yang berbeda-beda pula supaya tidak mudah bagi orang yang tidak berwenang untuk memecahkan pengenkripsian pada citra yang sudah dilakukan.

Algoritma enkripsi yang disertakan dalam aplikasi Enkripsi citra ini ada tiga pilihan dan semua algoritma tersebut telah diakui di dunia kriptografi sebagai algoritma yang dianggap aman, yaitu DES, *Triple* DES dan IDEA, dimana untuk mengenkripsi sebuah bagian dari citra harus memilih salah satu dari algoritma tersebut. Alasan diberikan pilihan ini untuk memperkecil resiko pemecahan algoritma enkripsi yang diterapkan pada citra dimana seorang *cryptanalyst* harus menebak teknik mana yang digunakan untuk mengenkripsi bagian dari citra tersebut.

Analisis Teknik Enkripsi

Teknik enkripsi dapat dinyatakan berhasil apabila citra yang sudah disembunyikan atau dienkripsi dapat dikembalikan seperti semula. Tujuan dari enkripsi suatu data atau informasi adalah supaya data atau informasi tersebut dapat terjaga kerahasiaannya walaupun data atau informasi

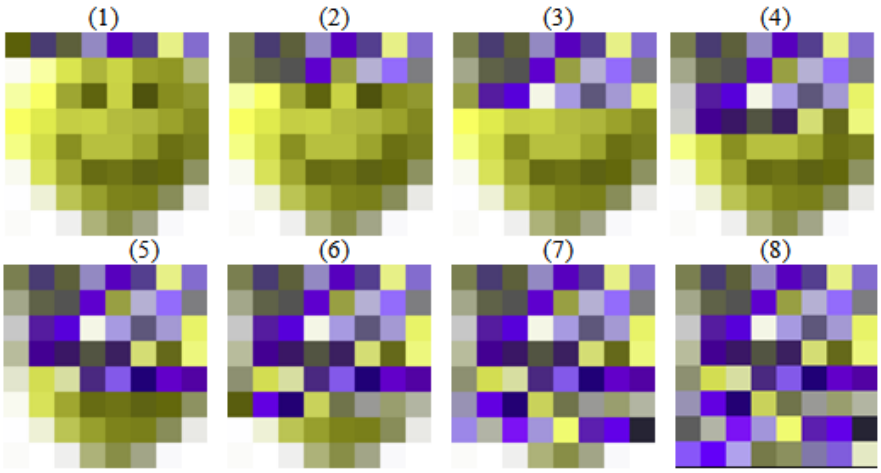
tersebut diambil oleh pihak yang tidak berwenang. Jadi dalam menerapkan teknik enkripsi selalu diasumsikan bahwa pihak manapun juga selalu dapat mengambil data atau informasi, bila data atau informasi tersebut tidak dapat diambil, maka tidak perlu melakukan enkripsi untuk penyembunyian data atau informasi tersebut.

Cara kerja suatu algoritma enkripsi dapat diketahui setiap orang, sehingga cara kerja suatu algoritma enkripsi tidak lagi menjadi suatu rahasia. Namun ini bukan berarti orang yang mengetahui cara kerja algoritma enkripsi dapat memecahkan suatu data atau informasi yang sudah dienkripsi. Kerahasiaan yang sebenarnya terletak pada kunci yang digunakan. Hanya orang yang memiliki kunci yang dapat mengetahui data atau informasi yang disembunyikan tersebut. Seorang *cryptanalyst* akan berusaha untuk mendapatkan kunci baik dengan cara paksa seperti dengan mencari kemungkinan kombinasi kunci satu persatu atau dengan algoritma tertentu. Suatu teknik enkripsi disebut sebagai teknik yang kuat bila tingkat kesulitan dalam mencari kunci yang digunakan sangat tinggi.

Analisis Teknik Mengenkripsi Citra

Pada bagian ini akan dijelaskan langkah-langkah mengenkripsi dan dekripsi pada citra dengan menggunakan tiga buah teknik yang telah dijelaskan. Pertama, buka citra yang ingin dienkripsi. Citra harus berupa *uncompressed* bitmap dengan ekstension bmp. Kedua, setelah citra dibuka, program akan mendeteksi apakah citra tersebut memiliki bagian-bagian yang dienkripsi oleh aplikasi tersebut. Dalam hal ini penanda yang menyatakan bahwa citra tersebut pernah dienkripsi adalah terdapatnya sebuah string ‘DEAOMS’ tepat diakhir *file*. Ketiga, bila ditemukan string tersebut, lanjutkan ke langkah 4, bila tidak lanjutkan ke langkah 6. Keempat, pembacaan dilanjutkan 1 *byte* sebelum string ‘DEAOMS’, nilai 1 *byte* tersebut menandakan jumlah enkripsi yang terdapat pada citra tersebut, jumlah maksimum adalah 255 buah. Masukkan nilai tersebut ke dalam sebuah variabel yang menandakan jumlah enkripsi yang tersedia. Kelima, lakukan perulangan sebanyak jumlah enkripsi, dalam setiap perulangan baca *byte-byte* sebelum 1 *byte* yang menandakan jumlah enkripsi. *Byte* yang dibaca setiap perulangan sebesar 17 *byte*. Informasi dalam 17 *byte* tersebut adalah posisi awal enkripsi, posisi akhir enkripsi, teknik enkripsi yang diterapkan. Tampilkan setiap informasi yang dibaca pada kotak informasi pada aplikasi. Keenam, tampilkan citra di tengah layar. Ketujuh, tunggu aktivitas dari *user*, bila *user* meng-*klik* dan *drag* di dalam wilayah citra, sebuah kotak akan tergambar mengikuti arah *mouse user*, posisi awal dan akhir dari kotak tersebut disimpan ke dalam variabel posisi awal dan akhir (x=kolom, y=baris). Posisi pilihan dari *user* secara otomatis akan dihitung jumlah *byte*-nya secara horizontal, jika jumlah *byte*-nya bukan merupakan kelipatan 8 maka secara otomatis akan diperbesar/diperkecil sehingga menjadi kelipatan 8.

Pengecekan secara vertikal tidak diperlukan karena pembacaan *byte* akan dilakukan perbaris. Untuk melakukan enkripdi atau dekripsi, *user* harus meng-*klik* pilihan menu utama yang tersedia. Kedelapan, berdasarkan posisi awal dan akhir yang telah dimasukkan ke variabel, lakukan perulangan dari y awal sampai y akhir dalam setiap perulangan terdapat perulangan lagi yang membaca *byte* dari posisi x awal sampai posisi x akhir. Setiap pembacaan memisahkan nilai *pixel* ke dalam variabel-variabel penampung warna merah, hijau dan biru, nilai tersebut dijadikan biner dalam *string*, dan satukan dengan variabel pada pembacaan berikutnya, bila jumlah pembacaan sudah mencapai kelipatan 8 maka kirimkan ketiga nilai biner tersebut ke modul enkripsi sesuai pilihan *user* melalui pemanggilan modul sebanyak 3 kali. Gambar di bawah ini memperlihatkan setiap langkah perubahan hasil enkripsi pada masing-masing baris.



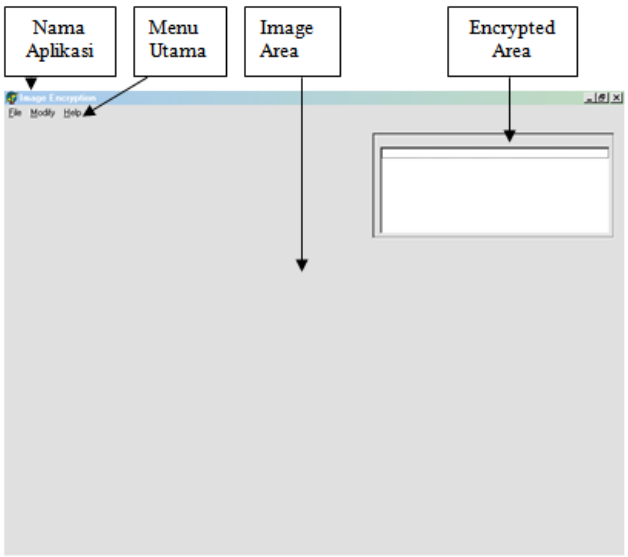
Gambar 2 Tahap Enkripsi Citra 8 x 8 *pixel*

Spesifikasi Sistem

Untuk menjalankan aplikasi *Image Encryption* diperlukan perangkat keras dan perangkat lunak. Spesifikasi kebutuhan perangkat keras adalah PC dengan prosesor minimum Pentium 166 MHz, memory dengan kapasitas minimum 32 MB, SVGA Card dengan kapasitas minimum 2 MB, keyboard, mouse, dan monitor. Sedangkan spesifikasi kebutuhan perangkat lunak adalah system operasi Microsoft Windows 98 ke atas.

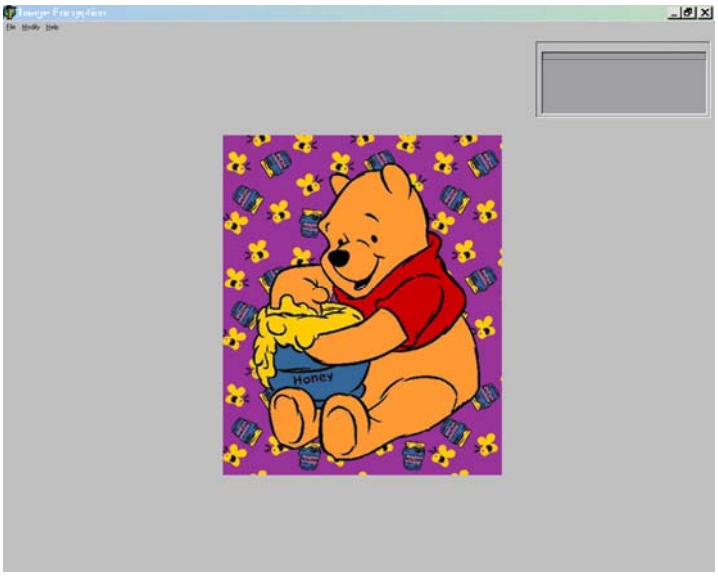
Prosedur Aplikasi *Image Encryption*

Cara untuk menjalankan aplikasi *Image Encrytion* adalah dengan menjalankan *executable* filenya yaitu ENCRYPT.EXE. Setelah itu akan muncul tampilan layar yang terdiri dari nama aplikasi, menu utama, *image area*, dan *encrypted area* seperti tampak pada gambar di bawah ini.

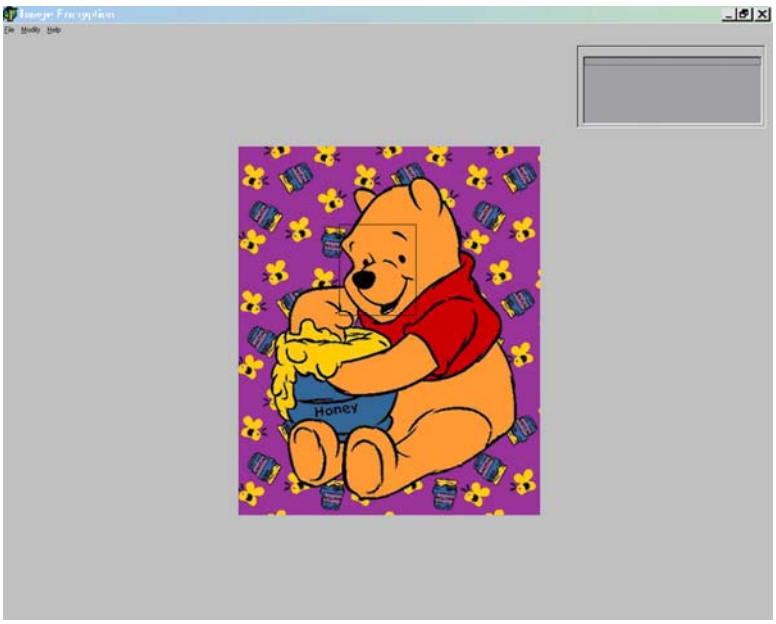


Gambar 3 Tampilan Layar Aplikasi *Image Encryption*

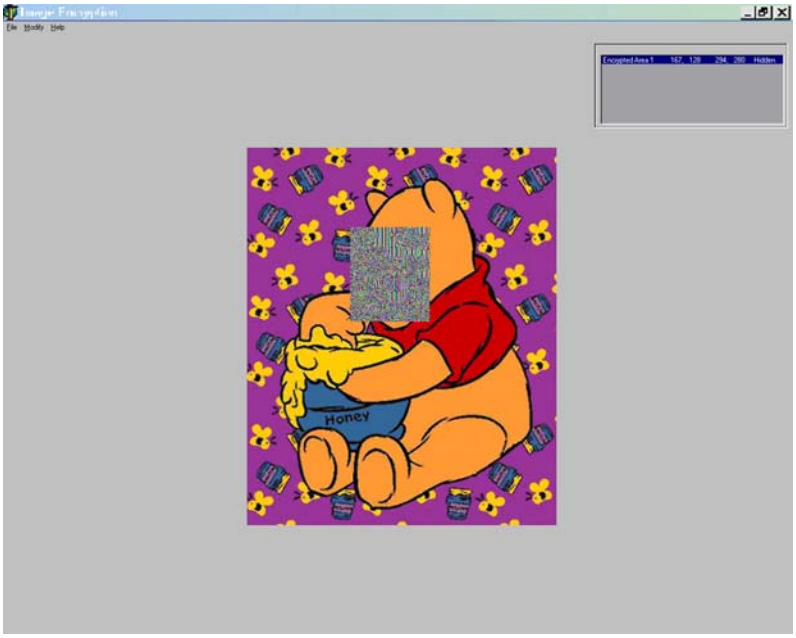
Dari gambar 3 di atas, dapat dilihat bahwa aplikasi *Image Encryption* memiliki tiga buah *menu*, yaitu *File*, *Modify*, dan *Help*. Pada *menu File* terdapat *submenu Open* (untuk membuka *file* citra), *Save* (untuk menyimpan / *rewrite file* citra yang sedang dibuka), *Save As* (untuk menyimpan *file* citra yang dibuka dengan nama atau lokasi penyimpanan yang berbeda), *Close* (untuk menutup *file* citra yang sedang dibuka), dan *Exit* (untuk keluar dari aplikasi). Untuk *menu Modify*, terdiri dari *submenu Encrypt* dengan tiga pilihan algoritma DES, Triple DES, dan IDEA (untuk melakukan enkripsi terhadap sebagian atau seluruh bagian citra dan *user* dapat memilih algoritma yang ingin digunakan) dan *submenu Decrypt* dengan tiga pilihan algoritma DES, Triple DES, dan IDEA (untuk melakukan proses dekripsi terhadap citra yang sudah terenkripsi). Pada *menu Help* terdapat *submenu Content* (untuk melihat panduan dalam menggunakan aplikasi ini) dan *About* (untuk melihat pembuat aplikasi ini).



Gambar 4 Tampilan Citra yang Dibuka



Gambar 5 Tampilan Citra yang Terblok untuk Dienkripsi



Gambar 6 Tampilan Layar Setelah Melakukan Enkripsi

Dari gambar 4, dapat dilihat tampilan layar pada saat citra ditampilkan, kemudian *user* dapat memilih bagian citra yang ingin disembunyikan atau dienkripsi (gambar 5). Pada saat enkripsi akan dilakukan, *user* dapat memilih algoritma enkripsi yang akan digunakan. Hasil citra yang telah dienkripsi akan dapat dilihat pada gamber 6.

Evaluasi Aplikasi *Image Encryption*

Pada tahap evaluasi ini dilakukan serangkaian pengujian aplikasi enkripsi pada citra. Pengujian yang dilakukan mencakup kecepatan waktu enkripsi untuk algoritma DES, *Triple* DES dan IDEA; keberhasilan tingkat dekripsi, tingkat keamanan dan keberhasilan pedekteksian data enkripsi pada citra yang sebelumnya pernah dienkripsi.

Dalam melakukan evaluasi kecepatan enkripsi, akan dipilih dua jenis citra, yaitu citra berwarna dan citra hitam putih. Dari masing-masing jenis citra akan dilakukan percobaan dengan mengambil sebagian dari citra tersebut dengan ukuran tertentu yang akan digunakan untuk membandingkan ketiga algoritma enkripsi.

Tabel 1 Evaluasi Perbandingan Kecepatan Waktu Jenis Citra Berwarna dan Citra Hitam Putih

Waktu (Detik)	**DES**		***Triple* DES**		**IDEA**	
	Berwarna	**Hitam Putih**	**Berwarna**	**Hitam Putih**	**Berwarna**	**Hitam Putih**
8 x 8 *pixel*	0.058	0.055	0.1703	0.1693	0.036	0.035
24 x 24 *pixel*	0.4256	0.4294	1.2419	1.2398	0.2332	0.2374
40 x 40 *pixel*	1.1527	1.1456	3.422	3.42	0.6308	0.6317
56 x 56 *pixel*	2.2553	2.2553	6.6222	6.6697	1.2268	1.2317
72 x 72 *pixel*	3.7172	3.7316	11.0028	10.8421	2.0089	2.0142

Dari hasil evaluasi di atas, dapat terlihat bahwa kecepatan waktu enkripsi citra berwarna hampir sama dengan kecepatan waktu enkripsi citra hitam putih. Hal ini disebabkan karena pengenkripsian terhadap citra menggunakan R, G, dan B dari masing-masing *pixel*, sehingga tidak mempengaruhi apapun termasuk kecepatan waktu. Kemudian juga dapat dilihat bahwa semakin banyak *pixel* yang harus dienkripsi, semakin besar waktu yang diperlukan. Dari perbandingan ketiga teknik diatas dapat pula terlihat bahwa teknik IDEA merupakan teknik yang tercepat.

Selain itu, juga dilakukan evaluasi tingkat keberhasilan dan mendapatkan hasil bahwa tingkat keberhasilan aplikasi *Image Encryption* adalah 100 % seperti dapat dilihat pada tabel 2 dan 3 di bawah ini.

Tabel 2 Tingkat Keberhasilan Dekripsi pada DES

Percobaan	Ukuran Seleksi (*pixel*)				
	8 x 8	24 x 24	40 x 40	56 x 56	72 x 72
1	100%	100%	100%	100%	100%
2	100%	100%	100%	100%	100%
3	100%	100%	100%	100%	100%
4	100%	100%	100%	100%	100%
5	100%	100%	100%	100%	100%
6	100%	100%	100%	100%	100%
7	100%	100%	100%	100%	100%
8	100%	100%	100%	100%	100%
9	100%	100%	100%	100%	100%
10	100%	100%	100%	100%	100%

Tabel 3 Tingkat Keberhasilan Dekripsi pada *Triple* DES

Percobaan	Ukuran Seleksi (*pixel*)				
	8 x 8	24 x 24	40 x 40	56 x 56	72 x 72
1	100%	100%	100%	100%	100%
2	100%	100%	100%	100%	100%
3	100%	100%	100%	100%	100%
4	100%	100%	100%	100%	100%
5	100%	100%	100%	100%	100%
6	100%	100%	100%	100%	100%
7	100%	100%	100%	100%	100%
8	100%	100%	100%	100%	100%
9	100%	100%	100%	100%	100%
10	100%	100%	100%	100%	100%

Tabel 4 Tingkat Keberhasilan Dekripsi pada IDEA

Percobaan	Ukuran Seleksi (*pixel*)				
	8 x 8	24 x 24	40 x 40	56 x 56	72 x 72
1	100%	100%	100%	100%	100%
2	100%	100%	100%	100%	100%
3	100%	100%	100%	100%	100%
4	100%	100%	100%	100%	100%
5	100%	100%	100%	100%	100%
6	100%	100%	100%	100%	100%
7	100%	100%	100%	100%	100%
8	100%	100%	100%	100%	100%
9	100%	100%	100%	100%	100%
10	100%	100%	100%	100%	100%

Evaluasi tingkat keamanan dilakukan dengan menggunakan sebuah blok enkripsi 24 x 24 *pixel* dengan kunci 'Software' akan didekripsi dengan kunci yang berbeda-beda. Untuk *Triple* DES memiliki sifat yang sama dengan DES, yaitu bila kunci yang digunakan hanya memiliki perbedaan bit pada posisi kelipatan ke-8 dengan kunci yang sebenarnya, maka blok enkripsi akan terdekripsi 100 %, hanya saja pada *Triple* DES menggunakan 2 buah kunci sehingga kemungkinan untuk mendapatkan 2 buah kunci yang sesuai lebih kecil.

Tabel 5 Tingkat Keamanan untuk DES

Percobaan	Key yang digunakan (ASCII)	Hasil Yang terdekripsi (%)
1	Software	100%
2	Binusian	0%
3	Komputer	0%
4	Hoftware	0%
5	Softwaro	0%
6	Roftware	100%
7	Softwarf	0%
8	Softward	100%
9	Saftware	0%
10	Seftware	0%

Dari table 5 di atas, dapat terlihat bahwa pada algoritma DES terdapat beberapa kunci yang dapat menghasilkan *plaintext* yang sama, hal ini disebabkan karena perbedaan kunci yang digunakan dengan kunci yang sebenarnya terletak pada bit kelipatan ke delapan.

Tabel 6 Tingkat Keamanan untuk IDEA

Percobaan	Key yang digunakan (ASCII)	Hasil Plaintext (HEXA)
1	Software	100%
2	Binusian	0%
3	Komputer	0%
4	Hoftware	0%
5	Softwaro	0%
6	Roftware	0%
7	Softwarf	0%
8	Softward	0%
9	Saftware	0%
10	Seftware	0%

Dari hasil evaluasi tingkat keamanan dapat dilihat bahwa enkripsi dan dekripsi dengan algoritma IDEA yang paling aman.

Aplikasi enkripsi pada citra ini menyisipkan data kedalam *file* citra agar dapat mendeteksi area enkripsi pada citra pada saat citra tersebut dibuka kembali.

Tabel 7 Tingkat Keberhasilan Pendeteksian Data Enkripsi

Percobaan	Jumlah Enkripsi	Keberhasilan Pendeteksi data Enkripsi
1. *File* Citra yang tidak di-*save* dengan *image editor* yang lain	1	100%
	3	100%
	5	100%
2. *File* Citra yang telah di-*save* dengan image editor yang lain	1	0%
	3	0%
	5	0%

Pendeteksian *file* citra telah diuji berulang kali dan ternyata tingkat keberhasilan pendeteksian enkripsi adalah 100 %, namun apabila *file* citra yang telah disimpan menggunakan aplikasi enkripsi citra ini dibuka dan di*save* dengan *image editor* yang lain maka data-data enkripsi yang telah disisipkan pada *file* tersebut akan hilang dan tidak akan terdeteksi oleh aplikasi enkripsi citra ini pada saat *file* citra ini dibuka kembali.

Hal ini dikarenakan apabila sebuah *file* di*save* oleh *image editor*, maka *image editor* akan melakukan penulisan ulang terhadap *file* tersebut, sehingga informasi yang disisipkan diakhir *file* oleh aplikasi enkripsi yang dibuat akan hilang.

Untuk evaluasi tingkat keberhasilan dekripsi untuk *file* yang telah dimanipulasi, dapat dilihat bahwa *file* citra yang telah dienkripsi dengan aplikasi enkripsi sangat rentan terhadap manipulasi citra seperti *Brightness* dan *Contrast*. Hal ini dikarenakan perubahan nilai *pixel* pada bagian blok enkripsi, dimana terdapat perbedaan satu *pixel* saja pada sebuah *plaintext* akan menghasilkan *ciphertext* yang jauh berbeda, maka hal yang sama akan terjadi sebaliknya dimana terjadi sedikit perbedaan saja pada sebuah *ciphertext*, maka *plaintext* yang dihasilkan tidak akan kembali seperti semula.

Tabel 8 Manipulasi *Brightness*

Jumlah *Brightness* yang ditambahkan pada *file* citra	Keberhasilan Pendekripsian pada *file* citra
1	0%
3	0%
5	0%

Tabel 9 Manipulasi *Contrast*

Jumlah *Contrast* yang ditambahkan pada *file* citra	Keberhasilan Pendekripsian pada *file* citra
1	0%
3	0%
5	0%

Berikut diberikan beberapa perbandingan aplikasi yang telah dibuat dengan aplikasi-aplikasi enkripsi citra lain yang telah ada, adapun kriteria perbandingan terdiri dari (1) kemampuan enkripsi sebagian pada citra, kriteria ini betujuan untuk mengevaluasi apakah enkripsi yang dilakukan dapat dilakukan pada sebagian dari citra sesuai keinginan pengguna aplikasi; (2) keutuhan fisik *file* citra, kriteria ini bertujuan untuk mengevaluasi apakah setelah *file* dienkripsi oleh aplikasi enkripsi citra, *file* tersebut masih dapat dibuka oleh *image editor* atau *image viewer* yang lain; (3) jenis *file* citra yang dapat dienkripsi, kriteria ini bertujuan untuk mengevaluasi jenis-jenis *file* citra mana saja yang dapat

dienkripsi oleh aplikasi enkripsi pada citra seperti BMP, GIF, JPEG dan lain sebagainya; (4) independen antara bagian-bagian yang dienkripsi, kriteria ini bertujuan untuk mengevaluasi jika aplikasi enkripsi citra dapat mengenkripsi sebagian pada citra, apakah bagian-bagian yang telah terenkripsi tersebut tidak saling tergantung satu sama lain. Apabila sebuah bagian enkripsi harus didekripsi terlebih dahulu sebelum dapat mendekripsi bagian enkripsi yang lain, berarti hubungan antara bagian-bagian enkripsi dalam citra tersebut tidak independen.

Tabel 10 Perbandingan Aplikasi *Image Encryption* dengan aplikasi enkripsi lainnya

Feature	**Nama Aplikasi**				
	ImageCrypt	**CryptBMP**	**CryptJPG**	**Fitin**	**Image Encryption**
Enkripsi sebagian pada citra	Tidak	Tidak	Tidak	Ya	Ya
Keutuhan fisik *file* citra	Tidak	Ya	Tidak	Ya	Ya
Jenis *file* citra yang dapat dienkripsi	Semua Jenis	Bitmap	JPG	GIF	Bitmap
Independen antara bagian enkripsi	Tidak	Tidak	Tidak	Tidak	Ya

SIMPULAN

Melalui penelitian ini, dapat ditarik kesimpulan berikut ini. Pertama, penerapan algoritma IDEA pada citra merupakan algoritma yang paling baik dilihat dari segi kecepatan dan keamanan. Kelebihan algoritma IDEA dari segi keamanan salah satunya dikarenakan menggunakan kunci 128-bit secara maksimal, sehingga memberikan tingkat kesulitan yang lebih tinggi untuk dipecahkan dibandingkan DES dan *Triple* DES. Kedua, pendeteksian terhadap citra yang terenkrip 100% berhasil apabila *file image* tersebut tidak pernah disimpan dengan menggunakan aplikasi *image editor* atau *image viewer* yang lain, seperti: Adobe Photoshop, ACDSee, Microsoft Paint, dan lain sebagainya. Ketiga, citra yang telah dienkripsi rentan terhadap modifikasi, seperti: *brightness*, *contrast*, dan lain sebagainya. Hal ini disebabkan karena ketika citra tersebut dimodifikasi, maka RGB pada *pixel* akan berubah sehingga pada saat akan melakukan dekripsi, citra tersebut tidak akan bisa kembali ke dalam keadaan semula.

Adapun saran yang diusulkan adalah sebagai berikut. Pertama, algoritma enkripsi yang diteliti terbatas pada algoritma kunci rahasia dimana kunci untuk mengenkrip sama dengan kunci untuk mendekrip, hal ini menyebabkan siapa saja yang mengenkrip sebuah citra pasti dapat mendekrip citra tersebut. Diharapkan agar dikembangkan penelitian terhadap algoritma kunci umum, dimana kunci untuk mengenkrip berbeda dengan kunci untuk medekrip, sehingga dapat diatur sedemikian rupa pihak-pihak mana saja yang hanya dapat mengenkrip atau mendekrip sebuah citra. Kedua, penerapan aplikasi *Image Encryption* ini hanya menggunakan jenis *file bitmap*. Diharapkan aplikasi tersebut dapat dikembangkan lebih lanjut agar dapat digunakan untuk jenis *file* citra lainnya seperti JPEG, GIF, PSD dan lain sebagainya. Ketiga, citra yang dienkripsi dengan aplikasi *Image Encryption* ini rentan terhadap modifikasi seperti *Brightness* dan *Contrass*. Diharapkan aplikasi ini dapat dikembangkan lebih lanjut sehingga citra yang telah dienkripsi dapat lebih tahan terhadap modifikasi.

DAFTAR PUSTAKA

Cantu, M. (2008). *Delphi 2009.* Piacenza, Italy: CreateSpace.

Gonzalez, R. C., & Woods, R. E. (2008). *Digital Image Processing.* New Jersey: Pearson Education, Inc.

Hoffman, N. A. *Simplified IDEA Algorithm.*

IMLJH. (2010). *Simplified Version Of The DES (Data Encryption Standard) in C#*. Retrieved from Code Project: http://www.codeproject.com/KB/cs/SDES.aspx

Jennie. *Data Encryption Standard*, retrieved from www.uow.edu.au/~jennie/CSCI971/Cs47103.ppt

Kammer, R. G., & Daley, W. M. (1999). Data Encryption Standard (DES). *Federal Information Processing Standards Publication* , 8 - 15.

Schneier, B. (1996). *Applied Cryptography, Protocols, Algorithms, and Source in C.* USA: John Wiley and Sons.

Sonka, M., Hlavac, V., & Boyle, R. (2007). *Image Processing, Analysis and Machine Vision.* CL Engineering.

Sukamaaji, A., & Rianto. (2008). *Jaringan Komputer*. Yogyakarta: Andi.