ISSN 0216-4728

PENGAMANAN ENKRIPSI BERKAS MENGGUNAKAN ALGORITMA DATA STANDAR ENKRIPSI

Atmanu Budhiarto[1]
Didin Mukodim[2]
Jurusan Sistem Informasi, Fakultas Ilmu Komputer
Universitas Gunadarma
[2]didin@staff.gunadarma.ac.id

ABSTRAK

Saat ini informasi telah menjadi bagian yang sangat penting dalam suatu aktivitas. Sebagai contoh, dalam suatu proses pengambilan keputusan, informasi dapat mengurangi ketidakpastian dan dapat menciptakan alternatif keputusan yang lebih baik. Kerahasiaan dari suatu informasi yang memiliki nilai sangat penting atau sensitif dapat dilindungi dengan menggunakan kriptografi. Untuk melindungi informasi yang berupa berkas diperlukan suatu perangkat lunak yang dapat melakukan proses enkripsi terhadap berkas-berkas. Perangkat lunak tersebut menggunakan suatu algoritma kriptografi untuk mengenkripsi berkas. Algoritma kriptografi yang digunakan pada penulisan ini adalah Data Standar enkripsi atau disingkat DSE. DSE mengenkripsi data dalam ukuran 64-bit dengan menggunakan kunci berukuran 56-bit. Blok data 64-bit diperoleh dengan mengambil byte-byte dan mengubahnya ke dalam bit. Blok bit data yang telah diproses selanjutnya diubah kembali ke dalam byte dan disimpan pada berkas hasil bserta dengan kunci yang digunakan. Berkas hasil tersebut memiliki ekstensi .ECR. Kunci yang disimpan pada berkas hasil tersebut dienkripsi dengan menggunakan MD5 (Message Digest 5). Algoritma DSE yang diimplementasikan pada perangkat lunak menggunakan mode operasi Cipher Block Chaining. Basis data digunakan untuk menyimpan daftar pengguna yang dapat menggunakan perangkat lunak tersebut beserta kata sandi logonnya. Dengan digunakannya penggunaname dan kata sandi logon untuk mengakses perangkat lunak maka akan meningkatkan keamanan dari data terenkripsi dan perangkat lunak itu sendiri. Selain itu keamanan data terenkripsi lebih tinggi karena algoritma DSE menggunakan mode operasi CBC sehingga ciphertext yang dihasilkan akan selalu berbeda untuk data yang sama. Karena dapat mengenkripsi semua tipe berkas maka perangkat lunak tersebut memiliki fleksibilitas dalam melindungi informasi yang ada.

Kata kunci : Enkripsi, Data Standar enkripsi, Kriptografi, Visual Basic.

PENDAHULUAN

Pentingnya nilai dari suatu informasi mengakibatkan informasi tersebut hanya dapat diakses oleh pihak tertentu saja. Perlindungan terhadap nilai informasi agar tidak diubah atau jatuh ke pihak yang tidak berwenang harus didukung oleh tingkat keamanan yang memadai. Jika suatu informasi sampai jatuh ke pihak yang tidak berhak atau informasi tersebut telah diubah sehingga menjadi palsu, maka akan dapat menimbulkan kerugian yang besar.

Bentuk data atau informasi yang cukup umum digunakan dalam lingkup komputer yaitu berupa berkas. Berkas-berkas yang biasa digunakan dalam komputer dapat memiliki format yang beragam, mulai dari berkas yang berupa teks hingga berkas yang berupa gambar bahkan audio/video.

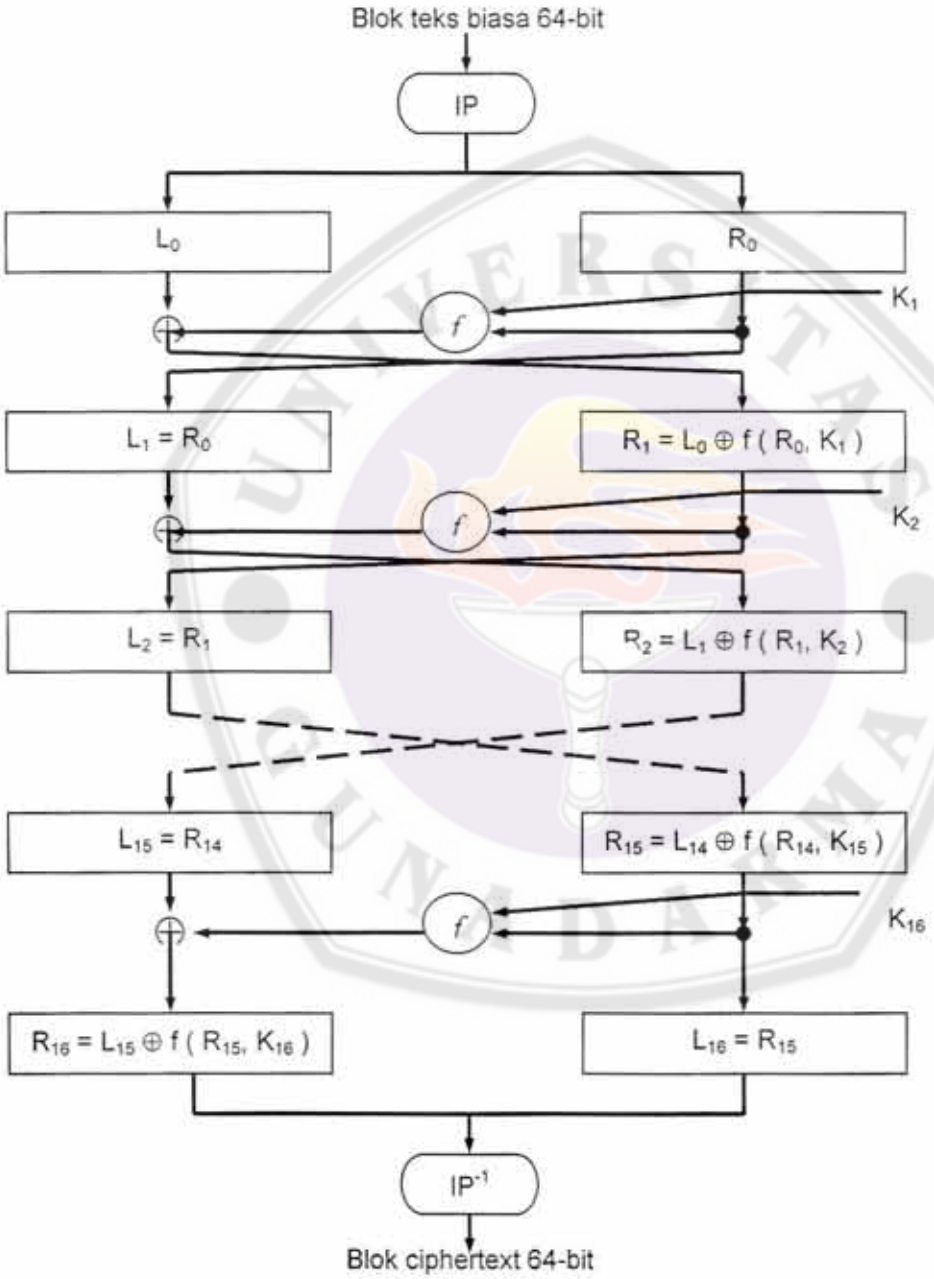
Salah satu metode yang dapat digunakan untuk melakukan pengamanan terhadap data atau informasi yaitu dengan menggunakan kriptografi. Kriptografi merupakan ilmu dan seni untuk mengacak suatu pesan, sehingga pesan tersebut hanya dapat dipahami dan dimengerti oleh pihak-pihak yang memiliki hak saja. Dalam kriptografi terdapat dua macam proses, yaitu enkripsi dan dekripsi. Proses enkripsi digunakan untuk mengacak pesan ke dalam suatu bentuk tertentu. Sedangkan dekripsi digunakan untuk mengubah kembali pesan yang telah diacak ke bentuk aslinya.

Kriptografi pada berkas digunakan untuk mengacak nilai- nilai bit yang terdapat di dalamnya. Nilai-nilai tersebut diacak secara per blok dengan menggunakan suatu algoritma kriptografi. Algoritma Data Standar Enkripsi

(DSE) yang diimplementasikan dapat digunakan untuk mengacak berkas-berkas dengan berbagai tipe.

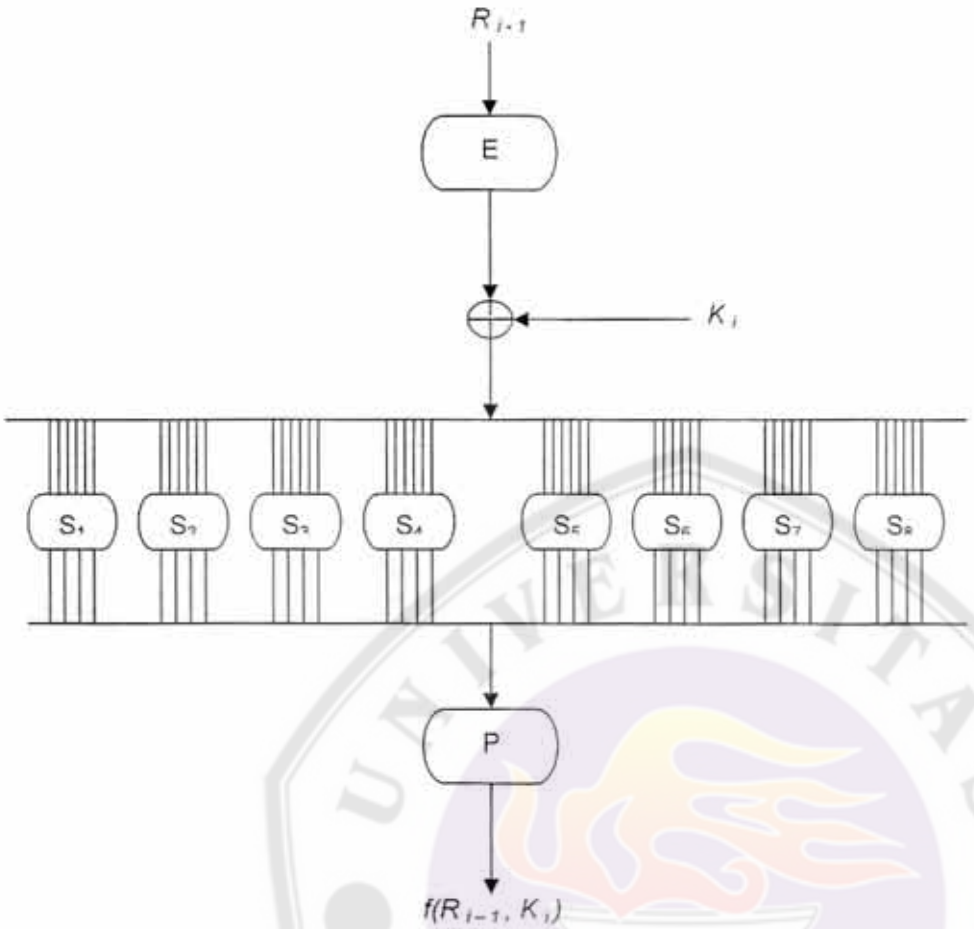
LANDASAN TEORI

Algoritma Enkripsi DSE. Data Standar Enkripsi atau disingkat dengan DSE, merupakan algoritma yang beroperasi dengan menggunakan teknik blok cipher. DSE mengenkripsi data yang terbagi ke dalam blok sebesar 64-bit dengan menggunakan kunci yang panjangnya 56-bit. Blok 64-bit tersebut dimanipulasi menggunakan tabel permutasi, operasi bit XOR, iterasi fungsi *f* sebanyak 16 kali serta tabel S-boxes. Gambar berikut ini menunjukkan algoritma dari DSE.



Gambar 1. Algoritma Enkripsi DSE.

ISSN 0216-4728



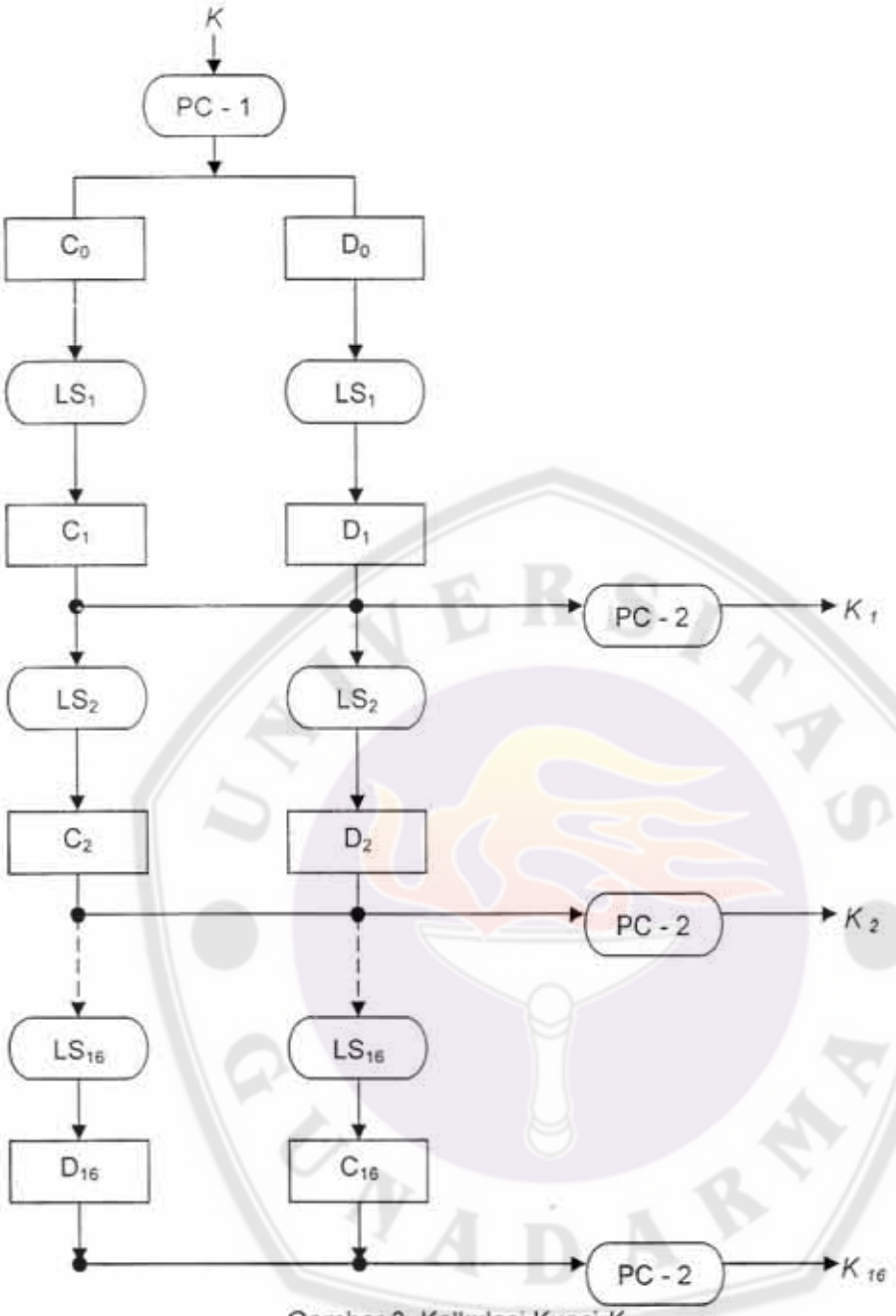
Gambar 2. Fungsi $f(R_{i-1}, K_i)$

PEMBAHASAN

Pembentukan Blok Data 64-bit. Data yang terdapat pada suatu berkas diproses oleh DSE dalam satuan bit. Bit data tersebut dikelompokkan ke dalam blok data yang masing-masing blok memiliki panjang 64-bit. Karena data yang terdapat di dalam berkas berada dalam satuan byte, maka agar dapat diproses oleh DSE data tersebut harus diubah dulu ke dalam satuan bit. Karena dalam 1 byte data terdiri dari 8-bit, maka data yang dapat diproses oleh DSE adalah sebanyak 8 byte untuk satu kali proses(1 byte = 8-bit maka 8 byte = 64-bit).

Untuk penentuan jumlah blok yang dapat dibuat dari panjang byte yang ada pada berkas dilakukan dengan cara menghitung jumlah blok menggunakan Persamaan (1).

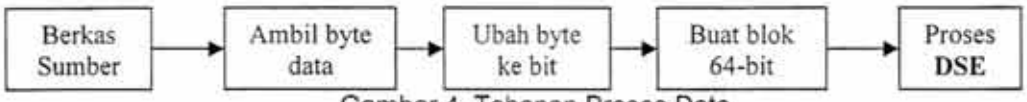
Pembangkitan Kunci. Setiap proses enkripsi atau dekripsi selalu membutuhkan suatu nilai tertentu yang digunakan untuk melaksanakan proses secara tepat nilai tersebut merupakan sebuah kunci. Algoritma DSE menggunakan kunci dengan panjang 56-bit. Sebelum dapat digunakan kunci tersebut harus melalui sejumlah proses agar menghasilkan nilai yang tepat. Kunci yang diinput pada awalnya memiliki panjang 64-bit. Setelah diproses dengan tabel PC1, ukuran kunci tersebut berkurang hingga menjadi 56-bit. Hal ini disebabkan karena bit pada posisi 8, 16, 24, 32, 40, 48, 56 dan 64 tidak dipergunakan oleh tabel PC1. Modul program yang digunakan pada pembangkitan kunci di antaranya InitialKey( ) dan GenKey( ).



Gambar 3. Kalkulasi Kunci K_i

$$JumlahBlok = \frac{panjangbytedata}{8}; \quad Sisabyte = panjangbytedataMod8, \text{ jika sisa byte=0}$$

$$JumlahBlok = jumlahblok + 1; \quad \text{jika sisa byte}\neq 0 \qquad (1)$$



Gambar 4. Tahapan Proses Data.

Enkripsi Kunci. Kunci yang digunakan pada proses enkripsi berkas, sebelum disimpan ke berkas hasil akan dienkripsi terlebih dulu menggunakan fungsi *hash* MD5 (Message Digest 5). Fungsi *hash* merupakan enkripsi satu arah sehingga teks biasa yang telah dienkripsi tidak akan dapat didekripsi. Enkripsi dengan algoritma MD5 tersebut akan menghasilkan blok output sebesar 128-bit(16 byte).
Kunci tersebut perlu untuk dienkripsi karena akan disimpan ke dalam berkas terenkripsi bersama dengan data terenkripsi dan akan digunakan lagi pada saat proses dekripsi berkas yaitu dengan cara membandingkannya dengan kunci input saat proses dekripsi. Proses enkripsi kunci di dalam kode program terdapat pada modul Enkripsi_Kunci( ).

Proses Enkripsi. Algoritma Data Standar enkripsi yang diimplementasikan menggunakan mode operasi CBC. Pada mode ini, sebelum blok data 64-bit dienkripsi maka terlebih dulu dilakukan operasi bit XOR terhadap blok data tersebut. Operasi bit XOR dilakukan antara blok data 64-bit dengan blok ciphertext yang dihasilkan dari enkripsi blok 64-bit data sebelumnya. Untuk menentukan ciphertext pada proses enkripsi menggunakan Persamaan (2). Sedangkan untuk menentukan teks biasa pada proses dekripsi rumus yang digunakan Persamaan (3).

$$C_i = E_k(M_i \oplus C_{i-1}) \qquad (2)$$

$$M_i = D_k(C_i) \oplus C_{i-1} \qquad (3)$$

Tabel 1 menunjukkan contoh teks biasa yang dienkripsi dengan algoritma DSE dengan mode operasi CBC.

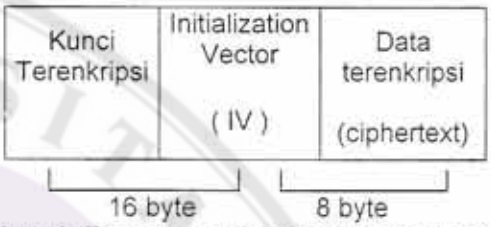
Tabel 1. Teks biasa yagn dienkripsi dengan algoritma DSE.

Teks biasa	:	"FORGIVEN_AND_FORGOTTEN"
Kunci	:	"PASOPATI"
Ciphertext	:	ˆ†yO#aof□Uê...†‹I□□·Ñ8ĩJ

Penyimpanan Hasil Enkripsi ke Berkas. Berkas hasil yang dibuat memiliki tipe atau ekstensi tersendiri, dalam hal ini digunakan ekstensi **.ECR**. Semua informasi yang berhubungan dengan tipe berkas ini akan disimpan di registry Windows pada key HKEY_CLASSES_ROOT\ECRFILE pada saat proses instalasi perangkat lunak.

Gambar 5. Icon untuk berkas .ECR.

Nama file menggunakan nama dari berkas asli beserta ekstensinya. Sebagai contoh jika berkas yang akan dienkripsi memiliki nama "Samples.txt", maka setelah dilakukan proses enkripsi berkas hasil yang terbentuk memiliki nama "Samples.txt.ecr". Format penulisan dari berkas yang dihasilkan dapat dilihat pada Gambar 6.



Gambar. 6. Format penulisan berkas hasil enkripsi.

Kunci terenkripsi memiliki panjang yang tetap yaitu sebesar 16 byte(128-bit). Sedangkan untuk IV juga memiliki panjang yang tetap yaitu sebesar 8 byte(64-bit). Kunci dan IV disertakan dalam berkas hasil enkripsi karena kedua nilai tersebut akan digunakan kembali pada saat proses dekripsi.

Struktur Basis data. Basis data digunakan untuk menyimpan nama dan kata sandi dari pengguna perangkat lunak ini. Data tersebut digunakan untuk melakukan perbandingan antara nama dan kata sandi yang diinput saat hendak menggunakan perangkat lunak ini dengan nama dan kata sandi yang ada di basis data. Basis data yang digunakan adalah Krypto.mdb yang terdiri dari satu buah tabel, seperti yang ditunjukkan Tabel 2.

Tabel 2. Basis data

Nama Tabel : Pengguna
Kunci : pengguna_nm
Indeks : Penggunadx

No	Nama Field	Tipe	Panjang	Keterangan
1.	pengguna_nm	Karakter	50	Nama pengguna
2.	pwd	Karakter	50	Kata sandi pengguna

Untuk meningkatkan keamanan terhadap akses ke perangkat lunak, kata sandi disimpan dalam bentuk terenkripsi. Sedangkan nama pengguna tetap dalam bentuk aslinya. Enkripsi kata sandi menggunakan MD5.

Dekripsi Berkas dengan Data Standar enkripsi. Untuk proses dekripsi digunakan algoritma yang sama dengan proses enkripsi. Hanya ada perbedaan dalam hal urutan penggunaan kunci saat proses dilakukan. Proses dekripsi menggunakan kunci dengan urutan menurun, mulai dari kunci ke-16 hingga kunci ke-1. Iterasi ke-1 dengan demikian menggunakan kunci ke-16, iterasi ke-2 menggunakan kunci ke-15 dan seterusnya.

Pengambilan Blok Ciphertext 64-bit. Untuk mengambil data berkas terenkripsi, terlebih dahulu harus dihitung panjang data. Panjang data terenkripsi dapat dihitung dengan menggunakan ketentuan berikut :

Panjang data = Panjang Berkas – (Panjang Kunci Terenkripsi + Panjang IV) (3)

Panjang kunci terenkripsi dan panjang IV memiliki nilai yang konstan yaitu masing-masing 16 dan 8. Nilai panjang kunci yang konstan diperoleh karena kunci ini dienkripsi dengan algoritma MD5. Algoritma tersebut menghasilkan ciphertext dengan panjang yang selalu tetap yaitu 16 byte(128-bit). Sedangkan untuk panjang IV telah ditentukan bahwa panjangnya harus 64-bit atau 8 byte. Sehingga panjang data terenkripsi dapat dihitung sebagai berikut :

Panjang data = Panjang Berkas – (16 + 8) (4)

Pembandingan Kunci Input dengan Kunci di Berkas. Pada bagian sebelumnya telah dijelaskan mengenai enkripsi kunci serta penyimpanannya ke berkas pada saat proses enkripsi. Pada proses dekripsi berkas, sebelum ciphertext dapat diuraikan maka pengguna harus memasukkan kunci yang tepat untuk dapat menjalankan proses dekripsi. Kunci ini disebut dengan kunci input.

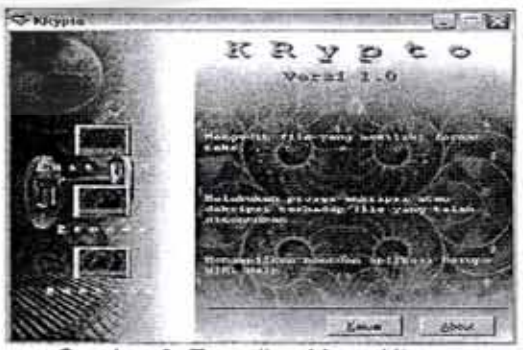
Kunci input tersebut selanjutnya akan dibandingkan dengan kunci terenkripsi yang ada di berkas. Untuk itu, maka kunci input akan dienkripsi dulu dengan menggunakan algoritma MD5. Selanjutnya baru dilakukan proses pembandingan antara nilai kunci input tersebut dengan nilai kunci di berkas. Jika keduanya memiliki nilai yang sama, maka proses dekripsi pesan akan dilakukan.

Tampilan Aplikasi. *Tampilan LogOn*. Tampilan ini ditujukan untuk membatasi pemakaian dari aplikasi ini. Aplikasi hanya dapat dipakai oleh pengguna yang memiliki nama pengguna di dalam aplikasi yang disimpan pada basis data Krypto.mdb. Untuk dapat mengakses aplikasi, pengguna harus memasukkan penggunaname dan kata sandi yang sesuai dengan yang ada di basis data aplikasi. Jika pengguna belum memiliki nama pengguna, maka harus melakukan registrasi terlebih dulu.

Gambar 7. Tampilan LogOn.

Gambar .8. Tampilan Registrasi.

Tampilan Menu Utama. Tampilan ini digunakan untuk mengakses tampilan lain yang ada di dalam aplikasi. Pada tampilan menu ini terdapat tombol yang dapat digunakan untuk membuka teks editor sederhana, memulai proses enkripsi atau dekripsi serta menampilkan berkas bantuan.



Gambar 9. Tampilan Menu Utama.

ISSN 0216-4728

Tampilan Proses. Tampilan proses merupakan tampilan utama yang digunakan untuk melakukan semua hal yang berkaitan dengan proses enkripsi atau dekripsi berkas. Melalui tampilan ini, proses enkripsi atau dekripsi tidak hanya dapat dilakukan pada satu berkas saja, tetapi juga dapat dilakukan pada beberapa berkas sekaligus.

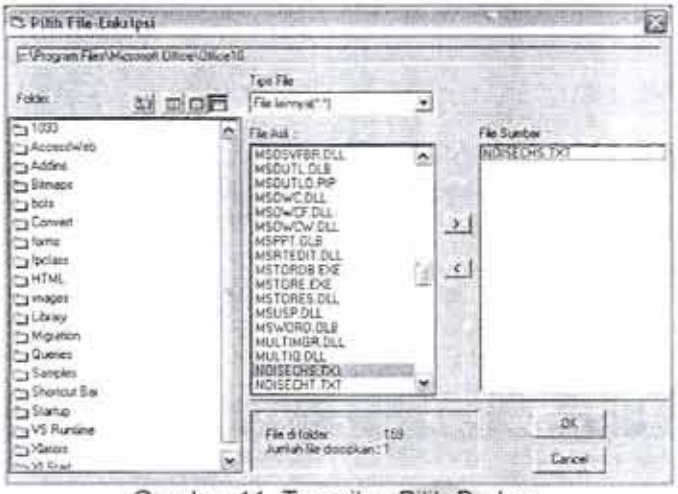
Gambar 10. Tampilan Proses.

Untuk memulai suatu proses yang baru, tombol Proses Baru pada toolbar ditekan atau dipilih perintah Proses Baru pada menu Berkas. Selanjutnya ditentukan proses dengan memilih Enkripsi dari submenu Pilih pada menu Proses. Lalu Pilih Berkas pada toolbar ditekan atau menu Berkas, hingga muncul tampilan Pilih Berkas. Berkas yang akan dienkripsi melalui tampilan ditentukan. Jika listview Proses telah terisi maka kotak Input Kunci akan diaktifkan. Langkah berikutnya, memasukkan kunci yang sama pada kotak Input Kunci dan kotak Konfirmasi Kunci. Lalu tombol Enkrip pada toolbar ditekan atau pilih Enkrip pada menu Proses. Selanjutnya akan ditampilkan *progressbar* di bagian kanan atas dari tampilan. *Progressbar* ini digunakan untuk memberikan informasi tentang tingkat kemajuan dari proses yang sedang berjalan.

Selama proses sedang berlangsung menu Berkas, Proses dan Setting serta beberapa tombol pada toolbar tidak diaktifkan, kecuali tombol Stop Proses. Tombol ini digunakan untuk menghentikan proses, apabila proses yang sedang dilaksanakan dianggap terlalu lambat dan ingin dihentikan. Jika proses telah selesai, maka listview Hasil akan terisi dengan daftar berkas yang telah terenkripsi. Berkas tersebut memiliki ekstensi .ecr. Selain itu icon pada panel ketiga statusbar akan berubah disertai dengan teks "Proses : OK".

Tampilan Pilih Berkas. Pada tampilan ini pengguna dapat menentukan berkas yang akan diproses. Langkah pertama untuk memilih berkas yang akan diproses yaitu dengan memilih direktori yang dikehendaki pada listview Pelipat. Selanjutnya ditentukan tipe dari berkas yang akan dipilih sehingga di filelist Berkas Asli akan tampak daftar nama berkas dengan tipe tersebut. Ditentukan nama berkas yang dipilih dengan menekan nama berkas tersebut. Pemilihan berkas yang ada menggunakan cara yang sama dengan aplikasi Windows pada umumnya. Untuk memilih seluruh berkas yang ada pada filelist, digunakan kombinasi tekan kiri tetikus dan tombol SHIFT. Sedangkan untuk memilih beberapa berkas tertentu, digunakan kombinasi tekan kiri tetikus dan tombol CTRL.

Untuk menyisipkan berkas yang telah dipilih ke dalam boks daftar Berkas Sumber, ditekan tombol Sisip. Sedangkan untuk menghapus berkas dari daftar pada boks daftar Berkas Sumber, ditekan tombol Hapus. Jika pada boks daftar Berkas Sumber sudah terdapat daftar berkas yang akan diproses, selanjutnya ditekan tombol OK.



Gambar 11. Tampilan Pilih Berkas.

Tampilan Pelipat. Pada menu Setting di tampilan Proses terdapat perintah Folder yang jika ditekan akan menampilkan tampilan pelipat. Tampilan ini digunakan untuk menentukan pelipat yang akan dipakai untuk meletakkan berkas hasil enkripsi. Jika pelipat tidak ditentukan, maka secara langsung akan digunakan pelipat dimana berkas aplikasi ini berada.

Gambar 12. Tampilan Pelipat.

Perubahan yang dilakukan terhadap lokasi pelipat akan disimpan di registry Windows pada key HKEY_CLASSES_ROOT\ECRFILE.

PENUTUP

Simpulan.

Penggunaan kriptografi untuk mengenkripsi suatu pesan atau informasi berupa berkas merupakan salah satu cara untuk mengamankan nilai suatu informasi. Dengan digunakannya teknik kriptografi maka informasi yang memiliki nilai sangat penting atau bahkan rahasia dapat dilindungi dari pihak-pihak yang tidak memiliki hak. Beberapa simpulan yang dapat diambil dari penulisan ini diantaranya adalah sebagai berikut:

Algoritma DSE dengan mode operasi Cipher Block Chaining(CBC) memiliki tingkat keamanan lebih tinggi daripada penggunaam algoritma tersebut dengan mode standar yaitu Electronic CodeBook(ECB).

Setiap proses enkripsi dengan mode CBC terhadap teks biasa yang sama akan selalu menghasilkan ciphertext yang berbeda walaupun menggunakan kunci yang sama berulangkali, sehingga tingkat keamanan ciphertext relatif lebih tinggi; ciphertext memiliki tingkat jaminan yang lebih tinggi untuk dapat didekripsi secara sah oleh penerima yang berhak karena nilai kunci dan IV (Initialization Vector) disertakan dalam berkas terenkripsi; kerahasiaan kunci lebih terjamin karena dienkripsi dengan algoritma MD5 yang merupakan fungsi enkripsi satu arah(one way encryption function), sehingga kunci tersebut tidak dapat didekripsi atau diuraikan kembali ke bentuk aslinya.

Pengendalian terhadap akses ke perangkat lunak enkripsi melalui penggunaan penggunaname dan kata sandi dapat meningkatkan keamanan dari data terenkripsi dan perangkat lunak itu sendiri. Dengan demikian penggunaan secara tidak sah dari

perangkat lunak tersebut untuk tujuan yang tidak sah pula dapat diminimalisasi atau bahkan dihindari.

Tingkat keamanan terhadap akses perangkat lunak relatif lebih baik karena penggunaan algoritma MD5 untuk mengenkripsi kata sandi pengguna; fleksibilitas dari perangkat lunak karena semua tipe berkas dapat dienkripsi dengan menggunakan perangkat lunak ini. Hasil dekripsi dari berkas terenkripsi akan sama dengan berkas aslinya.

Fasilitas bantuan dalam bentuk berkas Compiled HTML(.CHM) dapat membantu pengguna dan meminimalkan tingkat kesalahan dalam penggunaan perangkat lunak ini.

Saran

Ada beberapa langkah yang patut dipertimbangkan agar aplikasi ini dapat dikembangkan lagi, di antaranya menggunakan algoritma enkripsi yang lebih kompleks, baik itu berupa kombinasi algoritma enkripsi yang telah ada ataupun dengan membuat yang baru; melakukan proses enkripsi tidak hanya terbatas pada berkas tetapi juga pada suatu pelipat atau bahkan cakram keras; menyediakan fasilitas dekripsi otomatis(self decrypting) terhadap berkas terenkripsi yang dihasilkan. Hal ini ditujukan agar berkas tersebut tetap dapat didekripsi secara sah pada komputer yang tidak memiliki perangkat lunak ini; dan meningkatkan kinerja dari aplikasi sehingga menghasilkan waktu proses yang lebih cepat diantaranya dengan efisiensi kode program.

DAFTAR PUSTAKA

Adi Kurniadi, ***Pemrograman Microsoft Visual Basic 6***, Elex Media Komputindo, Jakarta, 2000.

Ananta Sjartuni, ***Dasar-dasar Pemrograman Visual Basic 5.0***, Elex Media Komputindo, Jakarta, 1999.

Anonim, ***Introduction To Cryptography and Cryptanalysis***. (http://theory.lcs.mit.edu/stasio/intro_crypto.htm).

Bruce Schneier, ***Applied Cryptography***, John Wiley & Sons, New York, 1996.

Bruce Schneier, et all., ***Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commercial Security***. (http://www.schneier.com/paper_keylength.html).

Dieter Gollman, et all., ***Computer Security***, West Sussex, New York 1982.

Dorothy Elizabeth, ***Cryptography and Data Security***, Addison Wesley, New York, 1982.

Gary Kessler, ***An Overview of Cryptography***. (http://www.garykessler.net/library/crypto.chm).

John Savard, ***Introduction To Cryptography***. (http://home.ecn.ab.ca/jsavard/crypto/intro.htm).

Michael Halvorson, ***Microsoft Visual Basic 6: Step by Step***, Elex Media Komputindo, Jakarta, 2000.

Sape Mullender, ***Distributed Systems***, Addison Wesley, New York, 1993.

Thomas J. Kakiay, ***Diktat Kuliah : Pengelolaan Sistem Terdistribusi***, Universitas Gunadarma.

Tri Amperiyanto, ***Bermain-main dengan Registry Windows***, Elex Media Komputindo, Jakarta, 2001.

US National Bureau of Standards, ***Data Standar enkripsi***, Federal Information Processing Standard (FIPS) Publication 46-2, Desember 1999. (http://www.itl.nist.gov/fipspubs/fip46-2.htm).

US National Bureau of Standards, ***DSE Modes of Operation***, Federal Information Processing Standard (FIPS) Publication 81, Desember 1980. (http://www.itl.nist.gov/fipspubs/fip81.htm).

US National Bureau of Standards, ***Guidelines for Implementing and Using the Data Standar enkripsi***, Federal Information Processing Standard (FIPS) Publication 74, April 1981. (http://www.itl.nist.gov/fipspubs/fip74.htm).

Wahana Komputer, ***Memahami Model Enkripsi dan Security Data***, ANDI Yogyakarta, Semarang, 2003.