e-ISSN: 2955-7002
Jurnal Informasi Komputer Logika
Vol.2, No.4, Thn. 2022

ALGORITMA TRIPPLE DES DALAM PENGAMANAN FILE DENGAN USB FLASHDISK

Eferoni Ndruru[1], Tomy Satria Alasi[2]
[1]Program Studi Manajemen Informatika, Universitas Budidarma, Medan
email: ronindruru@gmail.com
[2]Program Studi Teknik Informatika,STMIK Logika, Medan
[2]email: tomysatriaalasi@live.com

ABSTRAK

Salah satu metode penyandian yang cukup baik untuk digunakan adalah metode TrippleDes. Dengan menggunakan metode TrippleDes, kata kunci akan dienkripsi terlebih dahulu pada saat disimpan untuk kemudian didekripsi pada saat proses verifikasi.

Kata kunci yang dipergunakan merupakan Key dari Flash Disk sehingga proses keamanan datanya akan menjadi lebih baik, untuk proses enkripsi dekripsi File Flash disk harus dikenali terlebih dahulu jika tidak ada flash disk maka proses enkripsi dan dekripsi tidak dapat dilakukan. 3DES (Triple Data Encryption Standard) merupakan suatu algoritma pengembangan dari algoritma DES (Data Encryption Standard). Pada dasarnya algoritma yang digunakan sama, hanya pada 3DES dikembangkan dengan melakukan enkripsi dengan implementasi algoritma DES sebanyak tiga kali. 3DES memiliki tiga buah kunci yang berukuran 168-bit (tiga kali kunci 56-bit dari DES). Pada algoritma 3DES dibagi menjadi tiga tahap, setiap tahapnya merupakan implementasi dari algoritma DES.

Kata Kunci : Pengamanan File,Algoritma 3DES, FlashDisk

1. Pendahuluan

1.1 Latar Belakang Masalah

Dalam era globalisasi sekarang ini, komputer merupakan alat yang sangat dibutuhkan oleh banyak instansi dan perusahaan-perusahaan milik negara maupun swasta. Pemakaian komputer dewasa ini semakin berkembang pesat di segala bidang sesuai dengan kemajuan zaman. Hal ini ditandai dengan semakin bertambahnya jenis-jenis sistem operasi komputer yang digunakan mulai dari *Microsoft Windows, Mac OS* hingga *Linux* [1].

Agar dapat menghasilkan pengamanan komputer berbasis kata kunci yang memiliki tingkat kerahasiaan yang tinggi, haruslah dilakukan penyandian terhadap kata sandi tersebut baik untuk proses penguncian maupun proses pembukaan kunci pengamanan. Salah satu metode penyandian yang cukup baik untuk digunakan adalah metode *TrippleDes*[2]. Dengan menggunakan metode *TrippleDes,* kata kunci akan dienkripsi terlebih dahulu pada saat disimpan untuk kemudian didekripsi pada saat proses verifikasi. Kata kunci yang dipergunakan merupakan *Key* dari *Flash Disk* sehingga proses keamanan datanya akan menjadi lebih baik, untuk proses enkripsi dekripsi *File Flash disk* harus dikenali terlebih dahulu jika tidak ada flash disk maka proses enkripsi dan dekripsi tidak dapat dilakukan[3].

Dalam kriptografi terdapat beberapa algoritma yang dapat mengunci data dan masih banyak orang yang belum mengerti bagaimana mengunci ataupun mengamankan sebuah file sehingga tidak dapat dilihat oleh orang lain. Hal ini disebabkan rumitnya prosedur pengamanan komputer jika menggunakan fasilitas yang disediakan oleh masingmasing sistem operasi[4]. Untuk itu, dibutuhkan sebuah aplikasi yang dapat dengan mudah dan cepat mengunci dan mengamankan komputer pengguna dengan menggunakan kata kunci yang diinputkan ke dalamnya. Algoritma Elgamal menekankan pada permasalahan Algoritma diskrit. Dengan permasalahan tersebut maka cipherteks hasil enkripsi Elgamal akan sangat sulit di kriptanalis. Matematika diskrit yang dimaksud dalam kriptografi Elgamal adalah mencari sebuah bilangan pangkat (x), pada sebuah bilangan bulat (g). Dimana bilangan tersebut merupakan bilangan bulat lainnya (y) jika di mod dengan bilangan p (bilangan prima). kerumitannya terletak pada masalah diskrit karena melibatkan bilangan prima p sebagai variabel modulo dan x adalah bilangan yang dicari berupa bilangan pangkat (Winda, 2018).

Pemilihan *Flashdisk* sebagai kunci enkripsi juga sangat penting dan ini merupakan alasan utama kenapa penulis memilih menggunakan *Flashdisk* sebagai kunci enkripsinya.

Flashdisk digunakan sebagai kunci terhadap *File* yang akan di enkripsi, penggunaan *Flashdisk* akan lebih aman dikarenakan setiap *Flashdisk* tidak ada memiliki serial yang sama, tanpa *flashdisk* mustahil *file* tersebut

dapat di enkripsi ataupun di deskripsi.Dalam penelitian ini hasil yang di dapat berupa aplikasi berupa dengan pengamanan data dengan menggunakan flasdisk.

2. Teoritis

2.1 Pengertian Perancangan Aplikasi

Perancangan adalah merupakan awal dalam *fase* pengembangan sistem untuk setiap produk keteknikan atau sistem (Tavri D. Mahyuzir, 2000: 78). Perancangan adalah proses penggunaan berbagai teknik dan prinsip untuk tujuan mendefinisikan proses atau sistem secara detail.

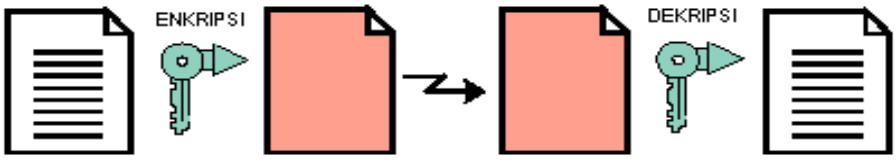
2.2 Pengertian Aplikasi

Aplikasi merupakan rangkaian kegiatan atau perintah untuk dieksekusi oleh komputer atau suatu perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna

Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau *suite* aplikasi *(application suite)*.

2.3 Pengenalan Kriptografi

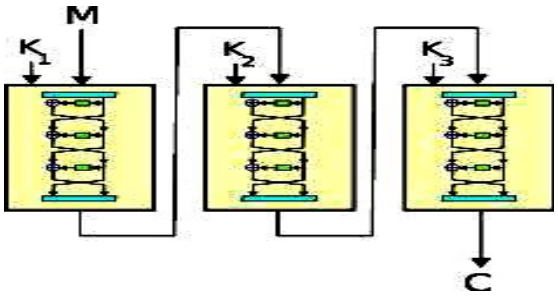
Secara etimologi (ilmu asal usul kata), kata kriptografi berasal dari gabungan dua kata dalam bahasa Yunani yaitu "*kriptos*" dan "*graphia*". Kata *kriptos* digunakan untuk mendeskripsikan sesuatu yang disembunyikan, rahasia atau misterius. Sedangkan kata *graphia* berarti tulisan[5]. Kriptografi didefinisikan sebagai ilmu dan pelajaran untuk tulisan rahasia dengan pertimbangan bahwa komunikasi dan data dapat dikodekan untuk mencegah dari mata-mata atau orang lain yang ingin mengetahui isinya, dengan menggunakan kode-kode dan aturan-aturan tertentu dan metode lainnya sehingga hanya orang yang berhak yang dapat mengetahui isi pesan sebenarnya [6].



Gambar 2.1 Konsep Dasar dari Enkripsi dan Dekripsi

2.3 Triple Data Encryption Standard

3DES (*Triple Data Encryption Standard*) merupakan suatu algoritma pengembangan dari algoritma DES (*Data Encryption Standard*)[7]. Pada dasarnya algoritma yang digunakan sama, hanya pada 3DES dikembangkan dengan melakukan enkripsi dengan implementasi algoritma DES sebanyak tiga kali. 3DES memiliki tiga buah kunci yang berukuran 168-bit (tiga kali kunci 56-bit dari DES). Pada algoritma 3DES dibagi menjadi tiga tahap, setiap tahapnya merupakan implementasi dari algoritma DES[8].



Gambar 2.2 Algoritma 3DES

2.3.1 Pemilihan Kunci

Ada dua pilihan untuk pemilihan kunci eksternal algoritma 3DES, yaitu:

a. K_1, K_2, dan K_3 adalah kunci-kunci yang saling bebas

$$\mathbf{K_1 \neq K_2 \neq K_3 \neq K_1}$$

b. K_1 dan K_2 adalah kunci-kunci yang saling bebas, dan K_3 sama dengan K_1

$$\mathbf{K_1 \neq K_2} \text{ dan } \mathbf{K_3 = K_1}$$

(*NIST, 2004*)

2.3.2 Proses Enkripsi dan Dekripsi

Proses enkripsi dan dekripsi algoritma 3DES dapat dicapai dengan beberapa cara, yaitu:

Tabel 2.1 Cara pengenkripsian dan pendekripsian

Cara	Enkrip	Dekrip
1	DES – EDE2 ▪ $K_1 \neq K_2$, $K_3 = K_1$	DES – DED2 ▪ $K_1 \neq K_2$, $K_3 = K_1$
2	DES – EEE2 ▪ $K_1 \neq K_2$, $K_3 = K_1$	DES – DDD2 ▪ $K_1 \neq K_2$, $K_3 = K_1$
3	DES – EDE3 ▪ $K_1 \neq K_2 \neq K_3 \neq K_1$	DES – DED3 ▪ $K_1 \neq K_2 \neq K_3 \neq K_1$
4	DES – EEE3 ▪ $K_1 \neq K_2 \neq K_3 \neq K_1$	DES – DDD3 ▪ $K_1 \neq K_2 \neq K_3 \neq K_1$

3. Analisa dan Perancangan

a. Analisa *Triple DES*

Dalam kriptografi, *Triple DES* adalah sebuah *cipher* blok yang dibentuk oleh DES dengan menggunakannya tiga kali. Ketika diketahui bahwa kunci berukuran 56-bit dari DES tidak cukup kuat untuk menjaga dari *brute force attacks, Triple DES* dipilih sebagai cara simpel untuk memperbesar ukuran kunci tanpa perlu mengganti algoritma. Penggunaan dari tiga tahap tersebut penting untuk mencegah *meet-in-themiddle attacks* yang efektif untuk digunakan terhadap enkripsi *Double DES*. Catat bahwa *DES* bukanlah sebuah grup (dalam matematika), karena jika merupakan grup, pembangunan *Triple DES* akan ekivalen dengan operasi *Single DES* yang berarti tidak lagi aman. Variasi paling simpel dari *Triple DES* adalah:

$$\mathbf{DES\ (k_3;DES(k_2;DES(k_1;M)))}$$

di mana M adalah blok pesan yang akan dienkripsi, k_1, k_2, dan k_3, adalah kunci DES.Variasi ini umumnya diketahui sebagai EEE karena ketiga operasi *DES* adalah proses enkripsi. Untuk menyederhanakan operasi antara *DES* dan *TDES*, langkah tengah biasanya diganti dengan proses dekripsi (*EDE mode*):

$$\mathbf{DES(k3;DES^{-1}(k_2;DES(k_1;M)))}$$

maka sebuah enkripsi *DES* dengan kunci k dapat direpresentasikan sebagai *TDES-EDE* dengan $k_1=k_2=k_3=k$. Pemilihan proses dekripsi pada langkah tengah tidak mempengaruhi keamanan dari algoritma.

3.1.1 Proses Enkripsi *Triple DES*

Untuk proses enkripsi *Triple DES*, plaintext di transformasikan secara berulang kali selama beberapa putaran. Banyaknya transformasi putaran (Nr) tergantung dari nilai Nk dan Nb. Nk yaitu panjang kunci dibagi 32, sedangkan Nb yaitu panjang blok dibagi 32.

Misalkan plaintext: 00112233445566778899aabbccddeeff

Kunci : 000102030405060708090a0b0c0d0e0f

1. Mengekspansi Kunci:

W1 = 00010203 W2 = 04050607

W3 = 08090a0b W4 = 0c0d0e0f

Tabel 3.1 Round Constanta (Rcon)

0	0	0	0	1	2	4	8	1	3
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Tabel 3.2 Round Key Enkripsi

Cipher Key				Round Key I				Round Key II				Round Key III			
0	4	8	c	6	2	a	6	6	4	e	8	6	2	c	4
1	5	9	d	a	f	6	b	2	d	b	0	f	2	9	9
2	6	a	e	4	2	8	6	f	d	5	3	4	9	c	f

3	7	b	f	d	a	1	a	b	1	0	e	e	f	f	1

Round Key IV				Round Key V				Round Key VI				Round Key VII			
7	5	9	d	c	9	0	d	e	7	7	a	4	3	4	e
7	5	c	5	a	f	3	6	9	6	5	3	9	f	a	9
7	e	2	d	3	d	f	2	f	2	d	f	0	2	f	0
c	3	c	d	8	b	7	a	d	6	1	b	a	c	d	6

Round Key VIII				*Round Key* IX				*Round Key* X			
7	4	0	e	4	0	0	e	3	3	3	d
3	c	6	f	9	5	3	c	1	4	7	b
7	5	a	a	2	7	d	7	d	a	7	0
5	9	4	2	1	8	c	e	f	7	b	5

2. Melakukan transformasi putaran sebanyak Nr kali sebagai berikut: Nk =128/32 =4
   Nb =128/32 =4
   Maka Nr=10 putaran.
   Putaran:

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} 63 & 09 & cd & ba \\ 53 & 65 & 70 & ca \\ e0 & e1 & b7 & d0 \\ 8c & 04 & 51 & e7 \end{bmatrix} = \begin{bmatrix} 5f & 57 & f7 & 1d \\ 72 & f5 & be & b9 \\ 64 & bc & 3b & f9 \\ 15 & 92 & 29 & 1a \end{bmatrix}$$

Add Round Key

Hasil MixColum Round Key I Hasil

$$\begin{bmatrix} 5f & 57 & f7 & 1d \\ 72 & f5 & be & b9 \\ 64 & bc & 3b & f9 \\ 15 & 92 & 29 & 1a \end{bmatrix} \begin{bmatrix} d6 & d2 & da & d6 \\ aa & af & a6 & ab \\ 74 & 72 & 78 & 76 \\ fd & fa & f1 & fa \end{bmatrix} = \begin{bmatrix} 89 & 85 & 2d & cb \\ d8 & 5a & 18 & 12 \\ 10 & ce & 43 & 8f \\ e8 & 68 & d8 & ed \end{bmatrix}$$

Untuk putaran dari *Triple DES dilakukan sebanyak* 10 putaran Dari beberapa langkah di atas diperoleh ciphertext sebagai berikut: **69c4e0d86a7b0430d 8cdb78070b4c55a**

3.1.2 Proses Dekripsi *Triple DES*

Pada proses dekripsi Triple DES hal-hal kunci dan *ciphertext* harus diketahui. Misalkan c*iphertext*: 69c4e0d86a7b0430d8cdb78070b4c55a

kunci : 000102030405060708090a0b0c0d0e0f maka proses pendekripsian nya adalah sebagai berikut:

1. MengkspansiKunci
   Kunci yang telah diekspansi:
   W1 = 00010203 W2 = 04050607
   W3 = 08090a0b W4 = 0c0d0e0f
   Proses pencariannya sebagai berikut

Tabel 3.3 Round Key Dekripsi

Cipher Key				Round Key I				Round Key II				Round Key III			
0	4	8	c	6	2	a	6	6	4	e	8	6	2	c	4
1	5	9	d	a	f	6	b	2	d	b	0	f	2	9	9
2	6	a	e	4	2	8	6	f	d	5	3	4	9	c	f
3	7	b	f	d	a	1	a	b	1	0	e	e	f	f	1

Round Key IV				Round Key V				Round Key VI				Round Key VII			
7	5	9	d	c	9	0	d	e	7	7	a	4	3	4	e
7	5	c	5	a	f	3	6	9	6	5	3	9	f	a	9
7	e	2	d	3	d	f	2	f	2	d	f	0	2	f	0
c	3	c	d	8	b	7	a	d	6	1	b	a	c	d	6

Round Key VIII				Round Key IX				Round Key X			
7	4	0	e	4	0	0	e	3	3	3	d
3	c	6	f	9	5	3	c	1	4	7	b
7	5	a	a	2	7	d	7	d	a	7	0
5	9	4	2	1	8	c	e	f	7	b	5

2. Putaran

a. Inverse of Add Round Key

69 6ad8 70 13 e3 f3 4d 7a89 2b 3d

b. Inverse of Mix Column

Pada putaran pertama proses ini tidak digunakan.

c. Inverse Shift Row

d. Inverse of Sub row

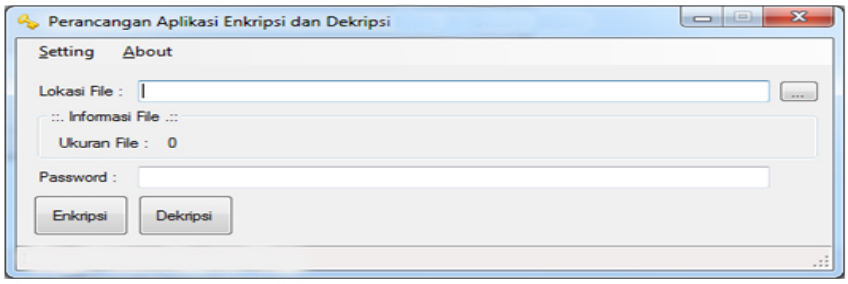
Proses *inverse* terus dilakukan sampai dengan 10 putaran sehingga hasil dari proses dekripsi, *plaintext* nya adalah sebagai berikut : **00112233445566778899aabbccddeeff**

4.2 Implementasi Sistem

Dalam pengembangan implementasi sistem, sangat diperlukan suatu metodologi yang dapat digunakan sebagai pedoman bagaimana dan apa yang harus dikerjakan selama implementasi ini. Dengan mengikuti metode dan prosedur-prosedur yang diberikan oleh suatu metodologi, maka implementasi sistem diharapkan dapat diselesaikan dengan baik. Dan implementasi sistem dalam aplikasi pengamanan ini mencakup spesifikasi kebutuhan perangkat keras (*hardware*) dan spesifikasi perangkat lunak (*software*).

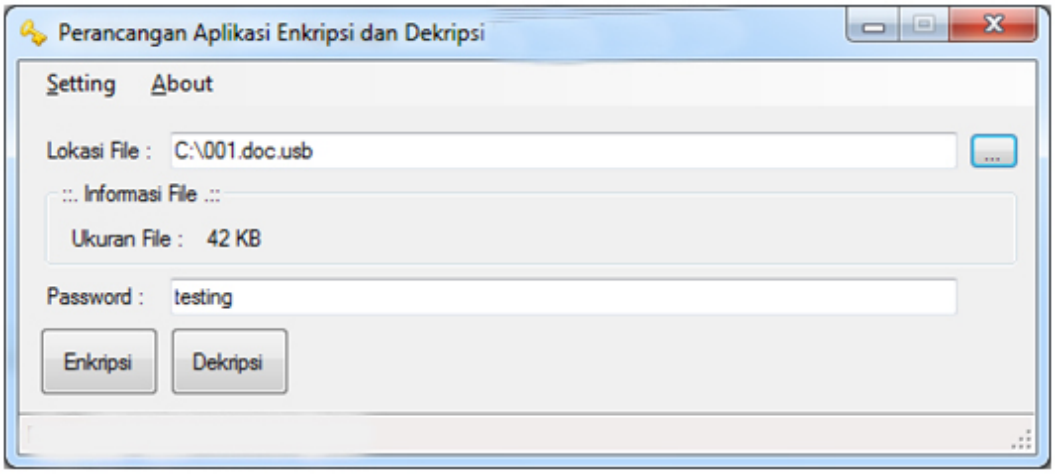
4.2.1 Pengujian Tampilan

Pengujian program digunakan untuk mencoba aplikasi yang sudah dirancang apakah sudah sesuai dengan yang diinginkan atau belum, perhatikan gambar dibawah ini



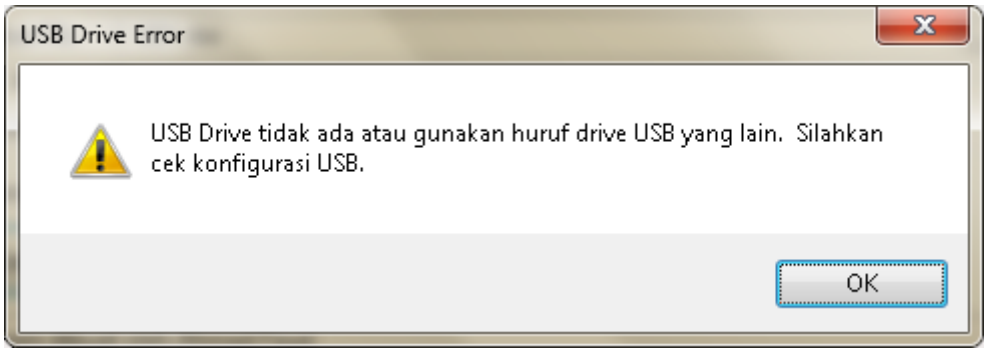
Gambar 4.1. Program Utama

Gambar diatas merupakan program utama yang dilakukan untuk proses enkripsi dan dekripsi *file*, untuk melakukan proses enkripsi tentunya harus mengambil *file* yang akan di enkripsi, *file* tersebut bisa *file Microsoft Word*, untuk mengambil *file*nya cukup dengan menekan tombol yang ada di bagian lokasi *file* dan mencari *file*nya sehingga hasilnya seperti dibawah ini:



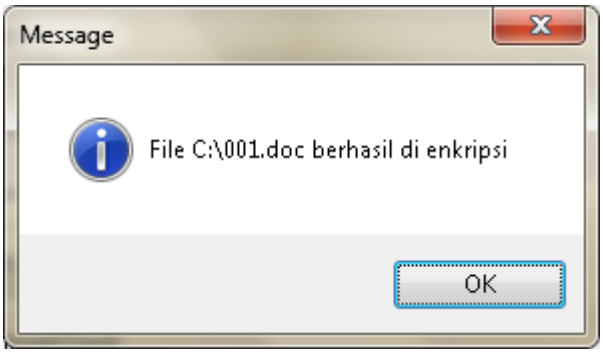
Gambar 4.2. Lokasi *File* yang Di Enkripsi

Gambar diatas menampilkan sebuah *file* yang akan dienkripsi dengan nama *file* 001.doc dengan ukuran file sebesar 42 KB, setelah menentukan file yang akan di enkripsi kemudian user memasukkan *password* sebagai kunci tambahan dari kunci USB *Flashdisk*, untuk proses pertama ini user akan mencoba melakukan proses enkripsi dengan tidak menggunakan USB *Flashdisk* dan ketika dilakukan proses enkripsi maka hasilnya akan muncul seperti dibawah ini:



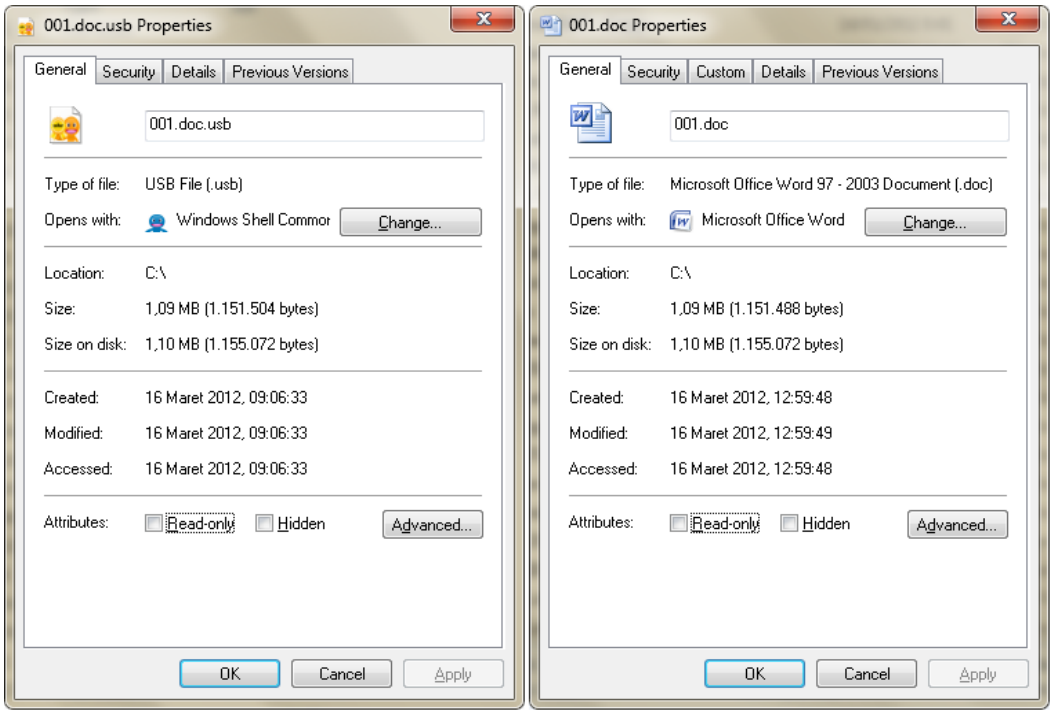
Gambar 4.3. Pesan Kesalahan USB Tidak dikenali

Pesan kesalahan yang muncul seperti pada gambar 4.3 merupakan pesan ketika proses enkripsi dilakukan tanpa adanya USB *Flash Disk* yang terpasang pada USB Komputer atau laptop, untuk menghindari munculnya pesan ini maka sebelum proses enkripsi USB *Flash Disk* harus terpasang pada USB *Port,* dan ketika sudah terpasang pada USB *Port* dan dilakukan proses enkripsi maka hasilnya akan seperti gambar dibawah ini:



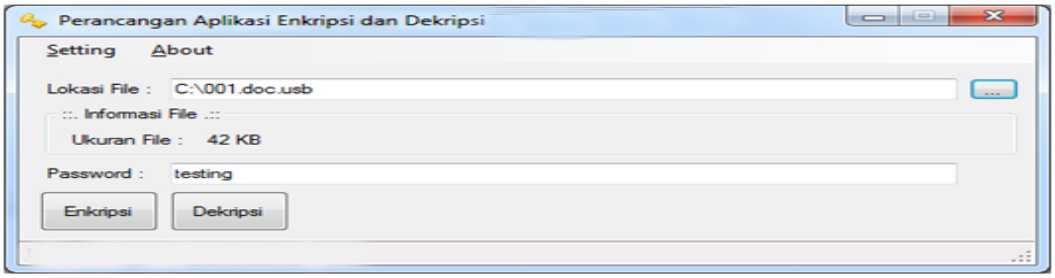
Gambar 4.4. Pesan Sukses Di Enkripsi

Proses enkripsi yang sudah sukses dilakukan akan menghasilkan sebuah *file* dengan nama baru yang menambahkan *ekstension* .USB, untuk perbandingan kapasitas sebelum dan sesudah dilakukan proses enkripsi dapat dilihat pada gambar dibawah ini:



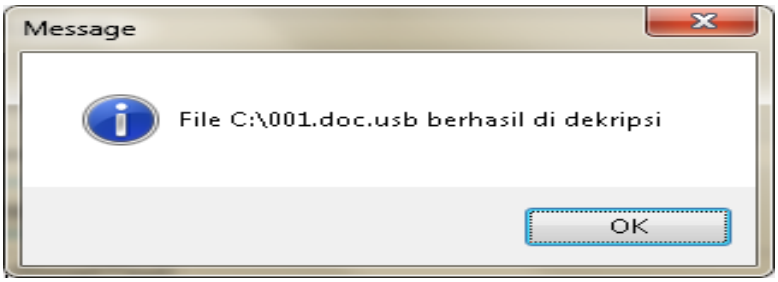
Gambar 4.5. Kapasitas Perbandingan *File*

Setelah proses enkripsi sudah selesai maka langkah selanjutnya adalah melakukan proses dekripsi, proses dekripsi dilakukan untuk mengembalikan *file* hasil enkripsi ke bentuk *file* aslinya, untuk proses dekripsi perhatikan gambar dibawah pada halaman setelah ini:



Gambar 4.6. Proses Dekripsi

Perhatikan gambar diatas, untuk proses dekripsi harus mengambil *file* hasil enkripsi dan memasukkan *password* yang digunakan untuk proses enkripsi, jika sudah semua sudah di proses dengan menekan tombol dekripsi proses dekripsi akan dilakukan dan jika sukses akan muncul pesan seperti ini:



Gambar 4.7. Pesan Sukses Dekripsi

Gambar diatas menunjukkan bahwa proses dekripsi berhasil dilakukan dan *file* akan dikembalikan kedalam bentuk semula.Untuk melakukan pengaturan penggunaan USB *Flash Disk* sebagai *key* dapat diperhatikan gambar dibawah ini:

Gambar 4.8. *Form* Pengaturan Fungsi FlashDisk

Gambar diatas digunakan untuk mengatur drive yang digunakan sebagai *flashdisk.*

Kesimpulan

Setelah menyelesaikan perancangan perangkat lunak enkripsi dan dekripsi, penulis menarik beberapa kesimpulan sebagai berikut :

1. Penggunaan USB Flash disk sebagai alat proteksi sebuah file dengan cara mengenkripsinya.
2. Memproses file secara enkripsi dan deskripsi dengan menggunakan metode Tripple DES sehingga keamnan file lebih terjaga.
3. Program bisa dibuat dengan menggunakan bahasa pemrograman visual basic2008 dengan menerapkan algoritma Tripplee DES.

DAFTAR PUSTAKA

[1] G. Swain and S. K. Lenka, "Steganography using two sided, three sided, and four sided side match methods," *CSI Trans. ICT*, vol. 1, no. 2, pp. 127–133, 2013.

[2] E. Ndruru and T. Zebua, "Application of Text Message Held in Image Using Combination of Least Significant Bit Method and One Time Pad," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 13, no. 4, p. 323, 2019, doi: 10.22146/ijccs.46401.

[3] G. Swain and S. K. Lenka, "LSB array based image steganography technique by exploring the four least significant bits," in *International Conference on Computing and Communication Systems*, 2011, pp. 479–488.

[4] T. S. Alasi, "Implementasi Kriptografi Dengan Algoritma Ceasar Cipher Untuk Keamanan Data Microsoft Office Word Dan Excel," *J. Inf. Komput. Log.*, vol. 1, no. 2, 2019.

[5] and R. A. R. U. Marsal, F. Arnia, "Enkripsi dan Dekripsi Citra Menggunakan Modifikasi Algoritma Vigenere Cipher," *KITEKTRO J. Online Tek. Elektro*, vol. 3, no. 3, pp. 6–10, 2018.

[6] E. Apriliana, "Kombinasi Algoritma One Time Pad dengan Pembangkit Kunci Blum Blum Shub," *J. Inform. Bandung*, vol. 1, no. 70, pp. 11–20, 2016.

[7] M. Diana and T. Zebua, "Optimalisasi Beaufort Cipher Menggunakan Pembangkit Kunci RC4 Dalam Penyandian SMS," *J. Sains Komput. Inform.*, vol. 2, no. 1, pp. 12–22, 2018.

[8] I. Gede Nengah Bayu Darmawan, G. Made Arya Sasmita, and P. Wira Buana, "Pengembangan Metode Pendeteksi Modifikasi Citra Menggunakan Metode Error Level Analysis," *J. Ilm. Merpati (Menara Penelit. Akad. Teknol. Informasi)*, vol. 7, no. 1, p. 29, 2019, doi: 10.24843/jim.2019.v07.i01.p04.