RANCANG BANGUN *MULTIFILE LOCKER APPLICATION* MENGGUNAKAN METODE DATA *ENCRYPTION STANDARD*

[1)]Rifaidi Akbar, [2)]Zainal Arifin &[3)] Dyna Marisa Khairina

[1,2,3)]Program Studi Ilmu Komputer, FMIPA, Universitas Mulawarman
Email : rifaidiakbar@yahoo.com[1)], zainal_arifin@fmipa.unmul.ac.id[2)], dyna.ilkom@gmail.com[3)]

ABSTRAK

Informasi yang berfungsi untuk memberi wawasan dan pengetahuan. Informasi disimpan dalam bentuk file dokumen atau media lainnya ke dalam penyimpanan komputer yang dapat dibaca dan dipahami oleh siapapun. Sering kali beberapa informasi yang bersifat rahasia dan hanya boleh dibaca dan diakses oleh user sendiri maupun orang tertentu. Salah satu cara untuk melindungi kerahasiaan informasi yaitu dengan menggunakan metode kriptografi untuk mengenkripsi file yang berisi informasi sehingga file menjadi teracak dan tidak dimengerti lagi maknanya. Salah satu algoritma kriptografi adalah metode Data Encryption Standard (DES) yang merupakan algoritma kriptografi modern dengan inputan kata kunci simetris dimana hanya membutuhkan satu kunci yaitu private key untuk mengenkripsi dan mendekripsinya.

File yang diamankan digabung membentuk satu file baru yaitu adalah file hasil. File hasil adalah file dari hasil proses pengamanan dengan ekstensi EXE (*.exe) yaitu tipe executable file. File hasil diamankan dengan password untuk membuka dan memecahnya, sedangkan file yang diamankan dienkripsi dengan menerapkan metode Data Encryption Standard. Proses enkripsi pada file mengubah struktur asli hexa dari file sehingga file dalam keadaan teracak dan sulit dipahami. Diperlukan proses dekripsi untuk mengembalikan menjadi file yang dapat diakses, dibaca dan dipahami lagi.

Kata Kunci : Kriptografi, *Multifile Locker Application*, *Data Encryption Standard*, Enkripsi *File*, *Executable File*.

PENDAHULUAN

Informasi sangat dibutuhkan pada saat ini. Informasi yang berfungsi untuk memberi wawasan dan pengetahuan sering ditulis dan disimpan seseorang dalam bentuk dokumen atau media lainnya di dalam penyimpanan komputer sehingga bisa dibaca dan mudah dipahami oleh siapapun. Sering kali beberapa informasi yang bersifat rahasia dan hanya boleh dibaca dan diakses oleh *user* sendiri maupun orang tertentu. Informasi menjadi sangat rentan untuk diketahui, diambil atau bahkan dimanipulasi dan disalahgunakan oleh pihak lain yang tidak berhak. Untuk melindungi kerahasiaan informasi tersebut mencegah dari jatuhnya informasi kepada pihak-pihak lain yang tidak berkepentingan, diperlukan pengamanan. Pengamanan dan kerahasiaan data merupakan salah satu aspek penting dari suatu sistem informasi. Sehingga informasi ditentukan hanya bisa diakses oleh pemilik informasi atau *user* yang telah ditentukan pemilik informasi. Pengamanan pada informasi dilakukan dengan cara Enkripsi. Enkripsi berfungsi untuk mengamankan sebuah informasi atau pesan (*plaintext*) menjadi pesan yang tersembunyi (*ciphertext*). Sedangkan untuk membalikkan proses dari enkripsi agar informasi kembali ke karakter semula disebut Dekripsi.

Untuk mempelajari cara dan proses enkripsi dan dekripsi maka diterapkanlah Kriptografi.Kriptografi adalah ilmu dan seni untuk menjaga kerahasian pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya[2]. Kriptografi juga untuk tujuan mempelajari tentang keamanan kerahasiaan informasi, integritas data, autentifikasi, dan nirpenyangkalan[6].Kriptografi terbagi 2 macam yaitu kriptografi klasik dan kriptografi modern. Kriptografi klasik mengenkripsi karakter per karakter dengan menggunakan alfabet. Sedangkan kriptografi modern beroperasi pada *string biner*. Salah satu dari metode kriptografi yaitu metode *Data Encryption Standard* (DES). Metode ini tergolong dalam metode kriptografi modern dimana beroperasi pada mode bit ketimbang karakter. Sehingga semua data dan informasi yang tersimpan dalam media *file* akan dienkripsi diubah dalam bentuk rangkaian bit biner, kemudian menjadi cipherteks dalam rangkaian bit biner juga. *File* yang sudah dienkripsi akan memiliki rangkaian bit yang diacak, sehingga susah dibaca dan dipahami secara langsung tanpa ada pemecah kunci dan aplikasi untuk dekripsi *file* tersebut. Metode ini juga termasuk algoritma simetris dimana hanya membutuhkan satu kunci yaitu *private key* untuk mengenkripsi dan mendekripsinya.

File atau disebut dengan Berkas didefinisikan sebagai kumpulan catatan yang sama disimpan

pada perangkat penyimpanan sekunder komputer[8]. *File*ini dapat diakses lebih dari satu proses, dapat dibaca, dan bahkan menulis yang baru. *File* biasanya diperoleh dari informasi yang sudah kita buat dalam bentuk dokumen digital, ataupun unggahan dari jaringan internet secara langsung ataupun lampiran sisipan dari email.

Adapun penelitian sebelumnya mengenai pengaman berkas menggunakan metode *Data Encryption Standard* telah dilakukan oleh Budhiarto (2006) dengan judul "Pengamanan Enkripsi Berkas Menggunakan Algoritma Data Standar Enkripsi". Pengamanan dilakukan dengan mengenkripsi pada satu *file* menerapkan metode *Data Encryption Standard* sehingga *file* tidak bisa diakses, dibaca dan dipahami.

Pada penelitian penulis menambahkan teknik pengamanannya. Cara pengamanannya yaitu dengan mengenkripsi dua *file* atau lebih secara bersamaan, sehingga*file* menjadi teracak dan tidak dimengerti lagi maknanya. Dan selanjutnya *file-file* yang sudah teracak sebelumnya digabungkan menjadi satu *file* baru yaitu *file hasilexecutable file* yang berekstensi EXE (*.exe). *File* juga diamankan dengan menambahkan *password* yang digunakan apabila ingin memecah kembali *file-file* yang telah digabungkan. Teknik penguncian *file* ini disebut teknik *file locker*, karena *file* diamankan lebih banyak maka menjadi *Multifile Locker*. Untuk memecah kembali *file* yang sudah digabungkan sebelumnya pada *file* EXE (*.exe) dibuka dengan *password*, selanjutnya *file-file* yang sudah teracak dapat dibuka dan dibaca kembali dengan cara membalik proses enkripsinya yaitu didekripsi dengan kata kunci yang sama pada saat proses enkripsi sebelumnya.

TINJAUAN PUSTAKA

Rancang Bangun

Rancang Bangun adalah tahap dari setelah analisis dari siklus pengembangan sistem yang merupakan pendefinisian dari kebutuhan-kebutuhan fungsional, serta menggambarkan bagaimana suatu sistem dibentuk yang dapat berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi, termasuk menyangkut mengkonfigurasikan dari komponen-komponen perangkat keras dan perangkat lunak dari suatu sistem[3].

Multifile

Multi dalam bahasa Inggris disebut dengan *multiple* yang artinya lebih dari satu. Sedangkan *file* atau disebut dengan berkas didefinisikan sebagai sebuah koleksi informasi berkaitan yang diberi nama dan disimpan di dalam penyimpanan sekunder (*secondary storage*) komputer[7]. Jadi dapat diartikan *Multifile* adalah kumpulan koleksi informasi yang lebih dari satu kemudian disimpan dalam penyimpanan sekunder komputer.

Biasanya sebuah *file* merepresentasikan data atau program. Adapun jenis-jenis dari file[7]:

1. *Text file* : urutan dari karakter-karakter yang diatur menjadi barisan dan mungkin halaman.
2. *Source file* : urutan dari berbagai *subroutine* dan fungsi yang masing-masing kemudian diatur sebagai deklarasi-deklarasi diikuti oleh pernyataan-pernyataan yang dapat dieksekusi.
3. *Object file*: urutan dari *byte-byte* yang diatur menjadi blok-blok yang dapat dipahami oleh penghubung sistem.
4. *Executable file*: kumpulan dari bagian-bagian kode yang dapat dibawa ke memori dan dijalankan oleh *loader*.

Locker Application

Locker berasal dari bahasa Inggris yang berarti diambil dari kata *lock* yaitu mengunci, sehingga *locker* adalah pengunci. *Locker* juga diartikan sebuah perangkat yang dioperasikan dengan kunci, kombinasi, atau *keycard* dan digunakan untuk memegang, menutup atau mengamankan[9].

Sedangkan *Application* berasal dari bahasa Inggris yaitu aplikasi yang artinya penerapan, lamaran ataupun penggunaan. Menurut Nugroho (2007) "Aplikasi adalah program yang ditulis untuk melaksanakan tugas khusus dari pengguna. Jenis program ini mempunyai sifat pasti tentang pemrosesan yang harus dilakukan *file* data yang harus di proses guna menyelesaikan suatu pekerjaan".

Menurut Abdul Kadir (2003), "Aplikasi adalah suatu program yang dibuat oleh pemakai yang ditujukan untuk kepentingan khusus".

Dapat dikatakan *Locker Application* adalah suatu program yang mempunyai tugas khusus untuk proses pengamanan yang dioperasikan dengan kunci atau kombinasi.

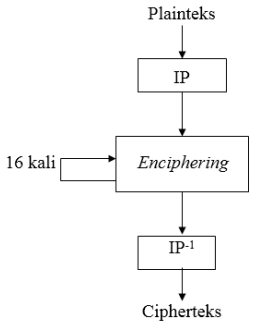
Data Encryption Standard (DES)

DES termasuk ke dalam sistem kriptografi simetri dan tergolong jenis *cipher blok*. DES beroperasi pada ukuran blok 64 bit. DES mengenkripsikan 64 bit plainteks menjadi 64 bit cipherteks dengan menggunakan 56 bit kunci internal (*internal key*) atau upa-kunci (*subkey*). Kunci internal dibangkitkan dari kunci eksternal (*external key*) yang panjangnya 64 bit[6].

Skema global dari algoritma DES yaitu (lihat Gambar 2.1):

1. Blok plainteks dipermutasi dengan matriks permutasi awal (*initial permutation* atau IP).
2. Hasil permutasi awal kemudian di-*enciphering* sebanyak 16 kali (16 putaran). Setiap putaran menggunakan kunci internal yang berbeda.

3. Hasil *enciphering* kemudian dipermutasi dengan matriks permutasi balikan (*invers initial permutation* atau IP^{-1}) menjadi blok cipherteks.



Gambar 1. Skema Global Algoritma *Data Encryption Standard*[6]

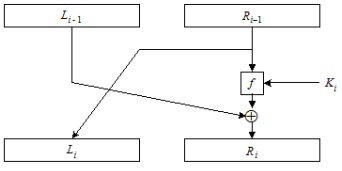
Di dalam proses *enciphering*, blok plainteks terbagi menjadi dua bagian, kiri (*L*) dan kanan (*R*), yang masing-masing panjangnya 32 bit. Kedua bagian ini masuk ke dalam 16 putaran DES. Pada setiap putaran *i*, blok *R* merupakan masukan untuk fungsi transformasi yang disebut *f*. Pada fungsi *f*, blok *R* dikombinasikan dengan kunci internal K_i. Keluaran dai fungsi *f* di-XOR-kan dengan blok *L* untuk mendapatkan blok *R* yang baru. Sedangkan blok *L* yang baru langsung diambil dari blok *R* sebelumnya. Ini adalah satu putaran DES. Secara matematis, satu putaran DES dinyatakan sebagai:

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

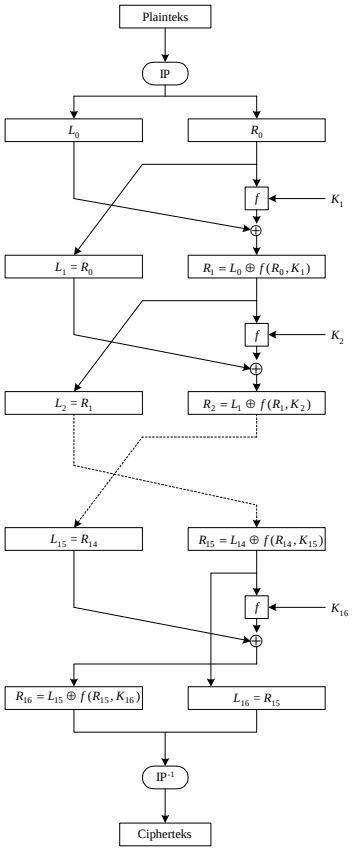
Dimana:

1. L_{i-1} adalah blok yang sedang giliran tidak dienkripsi.
2. $\oplus$ adalah operasi *exclusive or* secara *bitwise*.
3. *f* adalah fungsi *cipher* yang akan dijelaskan.
4. R_{i-1} adalah blok yang sedang giliran dienkripsi.
5. K_i adalah kunci untuk putaran *n*.

Gambar 3 memperlihatkan skema algoritma DES yang lebih rinci. Satu putaran DES merupakan model jaringan *Feitsel* (Lihat gambar 2). Perlu dicatat dari gambar 2 bahwa jika (L_{16}, R_{16}) merupakan keluaran dari putaran ke-16, maka (R_{16}, L_{16}) merupakan pra-cipherteks (*pre-cipertext*) dari *enciphering* ini. Ciperteks yang sebenarnya diperoleh dengan melakukan permutasi awal, IP^{-1}, terhadap blok pra-cipherteks.



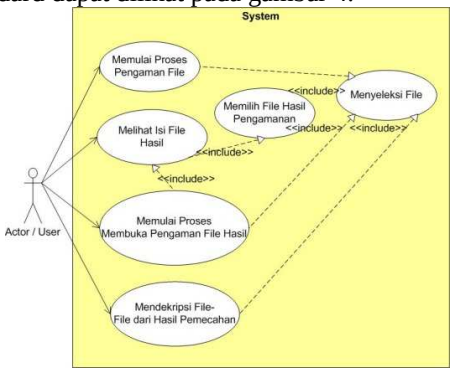
Gambar 2. Jaringan *Feistel* untuk satu putaran DES[6]



Gambar 3. Algoritma Enkripsi dengan DES Permutasi Awal[6]

HASIL DAN PEMBAHASAN

Use case diagram Multifile Locker Application menggunakan metode Data Encryption Standard dapat dilihat pada gambar 4.



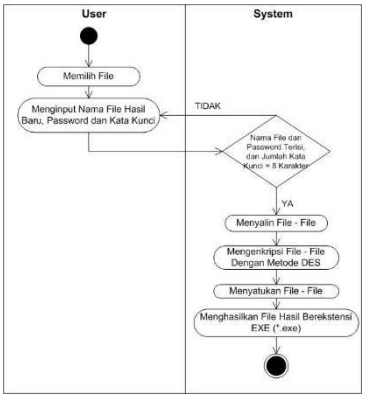
Gambar 4.*Use Case* Multifile Locker Application

Pada gambar 4*user* merupakan pemilik *file* dan pada saat *user* memulai proses pengamanan, terlebih dahulu menyeleksi *file* yang ingin diamankan kemudian menginputkan *password* dan kata kunci enkripsi dan selanjutnya memulai proses pengamanan sehingga memperoleh *file* hasil. Pada proses membuka pengamanan *file* hasil, user perlu memilih *file* hasil dari proses pengamanan

sebelumnya serta menginput *password* dan kata kunci enkripsi yang juga digunakan pada saat proses pengamanan sebelumnya. Setelah itu *file* akan ditampilkan dan *user* memulai proses membuka pengamanan. Hasil dari proses adalah berupa *file-file* yang bisa diakses, dibaca, dan dipahami. Pada proses pendekripsi *file*, *file* yang diseleksi untuk didekripsi adalah *file* yang sebelumnya hanya melalui proses pemecahan tanpa melalui proses membuka pengamanan enkripsi pada *file*. Sehingga perlu proses dekripsi agar file bisa diakses, dibaca, dan dipahami kembali.

Proses Pengamanan File

Pada proses pengamanan *file*, *user* harus menginputkan *password* dan kata kunci. *Password* memberikan keamanan pada saat akan memecah *file* hasil, sedangkan kata kunci adalah kata kunci untuk proses pengenkripsian pada *file-file* yang telah diseleksi.Kata kunci yang diinputkan pada proses enkripsi menggunakan *Data Encryption Standard* dengan jumlah bit kunci harus 64 bit sehingga harus menginputkan 8 karakter karena dalam satu karakter memiliki jumlah 8 bit. *Activitydiagram* untuk proses ini dapat dilihat pada gambar 5.

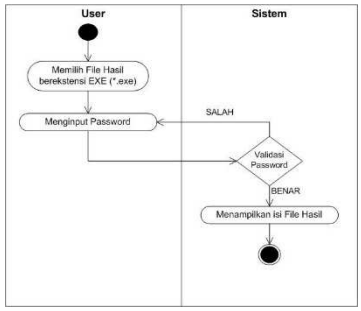


Gambar 5.*Activity Diagram* Mulai Proses Pengamanan *File*

Dapat dilihat pada gambar 5 setelah *user* menginputkan *file*, nama *file* hasil, *password* dan kata kunci sistem terlebih dahulu menyalin *file-file* yang akan diamankan kemudian mengenkripsi dilanjutkan dengan penyatuan *file-file* sehingga menjadi *file* hasil yang berekstensi EXE (*.exe).

Proses Lihat Isi File Hasil

Activity diagram Lihat Isi *File* Hasil dapat dilihat pada gambar 6.

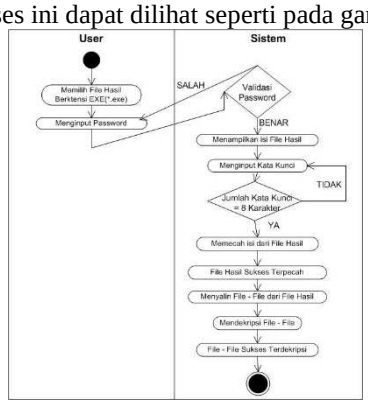


Gambar 6.*Activity Diagram* Lihat Isi *File* Hasil

Pada gambar 6 adalah proses sistem menampilkan *file-file* yang disatukan sebelumnya yang ada pada *file* hasil. *User* hanya perlu menginputkan *password* dan kemudian divalidasi oleh sistem. Apabila *password* benar, maka info dari *file-file* akan ditampilkan.

Proses Membuka Pengamanan File Hasil

Untuk memulai proses membuka pengamanan, terlebih dahulu memilih *file* hasil. Pada proses ini terdapat untuk menginputkan *password* dan kata kunci. Adapun *activity diagram* dari proses ini dapat dilihat seperti pada gambar 7.

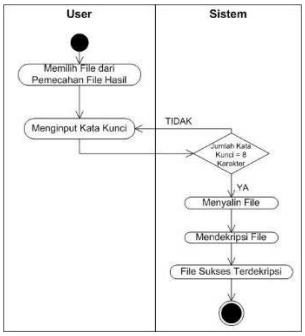


Gambar 7.*Activity Diagram* Mulai Proses Membuka Pengamanan *File* Hasil

Pada gambar 7 untuk memulai proses membuka pengamanan, *user* terlebih dahulu melakukan pengecek terhadap *file* hasil, setelah *password* yang diinputkan benar kemudian dilanjutkan dengan menginputkan 8 karakter kata kunci dekripsi. Dan selanjutnya sistem melakukan pemecahan terhadap *file* hasil dan kemudian menyalin *file* dari hasil pemecahan. *File* kemudian didekripsi untuk mengembalikan ke *file* yang bisa diakses, dibaca dan dipahami kembali.

Proses Dekripsi File-file dari Hasil Pemecahan

Proses pendekripsiandilakukan apabila *file* sudah dipecahkan tidak melalui sistem. Jadi hanya memilih *file* dari hasil pemecahan dan menginput kata kunci dekripsinya, sehingga mengembalikan ke *file* semula dari *file* yang teracak *hexa*-nya. *Activity diagram* dari proses pendekripsian ini dapat dilihat seperti pada gambar 8.



Gambar 8.*Activity Diagram* Dekripsi *File-file* Dari Hasil Pemecahan

Pada gambar 8*user* hanya memilih *file* dari hasil pemecahan, *file* ini diperoleh ketika pemecahan dilakukan tanpa menggunakan sistem. Dan kemudian menginputkan kata kunci dekripsi sebanyak 8 karakter. *File* disalin kemudian dilanjutkan dengan mendekripsi dan *file* sukses didekripsi.

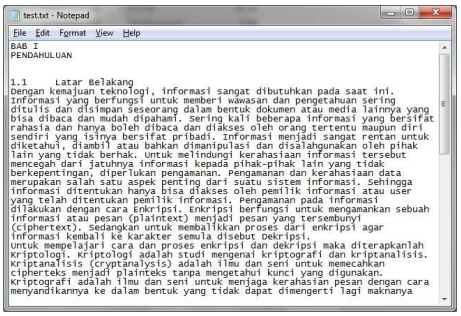
PENGUJIAN SISTEM

Pengujian pada sistem difokuskan pada tahapan pengamanan *file*, membuka pengamanan dan dekripsi *file*. Pengujian dilakukan untuk menguji tingkat keberhasilan, apakah *file* bisa diamankan sehingga *file* terkunci dan tidak bisa dibaca karena terenkripsi dan kemudian hanya bisa diakses oleh *user* sendiri. Dan pada saat memulai membuka pengamanan, pengamanan dibuka dan *file-file* bisa dikembalikan dalam bentuk semula sehingga menjadi *file* yang bisa dibaca kembali. Berikut beberapa *file* yang dijadikan bahan dalam pengujian dapat dilihat pada tabel 1.

Tabel 1*File-file* Bahan Pengujian

No.	Nama File	Format	Size (Byte)
1	10. BAB I.docx	DOCX	27489
2	flanel.jpg	JPG	88908
3	test.txt	TXT	4323
		Jumlah Total	**120720**

Adapun tampilan salah satu file yaitu ‘test.txt’ sebelum melakukan pengamanan dapat dilihat pada gambar 9.



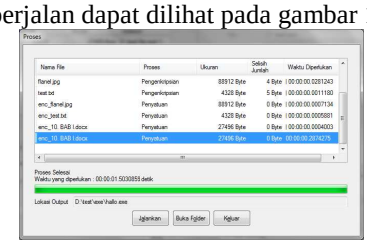
Gambar 9. Tampilan File Sebelum Pengamanan

Pengujian Memulai Proses Pengamanan File

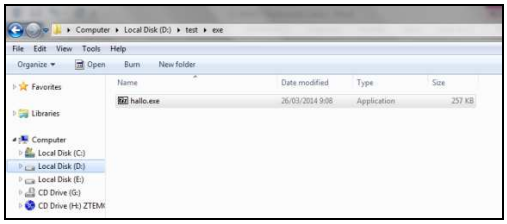
Pengujian dilakukan dengan menggunakan *file-file* pada tabel 1 diinputkan pada *textfield* :

Nama File Hasil : hallo
Password : samarinda
Kata Kunci Enkripsi : komputer

Dengan kata kunci enkripsi yang diinputkan harus sebanyak 8 karakter. Kemudian dilanjutkan dengan memilih lokasi folder *output file* hasil. Setelah semua *file* sudah diseleksi kemudian dipilih dan semua data telah diiputkan. Setelah itu memulai proses pengamanan kemudian setelah proses selesai maka tampilan beberapa data pada saat proses berjalan dapat dilihat pada gambar 10.



Gambar 10. Tampilan Proses Pengamanan Selesai



Gambar 11. Hasil Dari Proses Pengamanan *File*

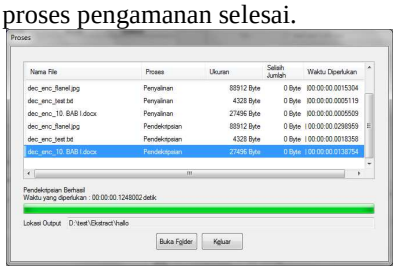
Dapat dilihat pada gambar 10 waktu proses pengamanan adalah 00:00:01.5030859 detik dan pada gambar 11*file* berhasil diamankan dan menjadi *file* hasil dengan *size* (ukuran) 257 KB.Kemudian *file* hasil dieksekusi maka akan muncul tampilan untuk menginputkan *password* seperti pada gambar 12.

Gambar 12. Tampilan Input *Password* Pada Saat *File* Hasil Dieksekusi

Pengujian Memulai Proses Membuka Pengamanan File

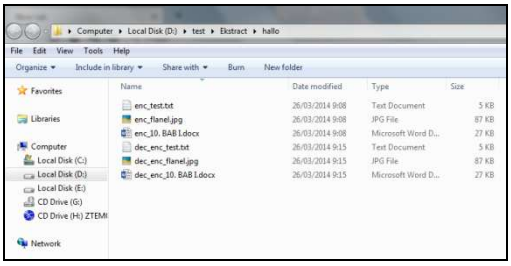
Pada pengujian proses membuka pengamanan *file*, *file* diinputkan adalah *file* hasil dari proses pengamanan sebelumnya. Jadi dalam pengujian ini *file* hasil yang digunakan adalah *file* ‘hallo.exe’ dengan *size* 257 KB (262984 Byte). Sebelum memulai proses membuka pengamanan, terlebih dahulu melakukan pengecekan sehingga ditampilkanlah *file-file* di *DataGridView* pada *file* hasil yang pada pengujian proses pengamanan *file* sebagai bahan uji. Setelah menginputkan *file* hasil dan *password* divalidasi benar oleh sistem maka diperoleh info dari *file-file*.Setelah menampilkan

info dari *file* pada *DataGridView* kemudian menginput kata kunci dan memilih lokasi folder *output* untuk *file-file* yang dibuka keamanannya. *File* yang dibuka keamanannya dengan cara memecah *file* hasil dan mendekripsi *file-file* hasil pemecahan menggunakan metode *Data Encryption Standard* (DES), kemudian memulai proses membuka pengamanan dan dapat dilihat pada gambar 4.36 data-data selama proses berjalan setelah proses pengamanan selesai.



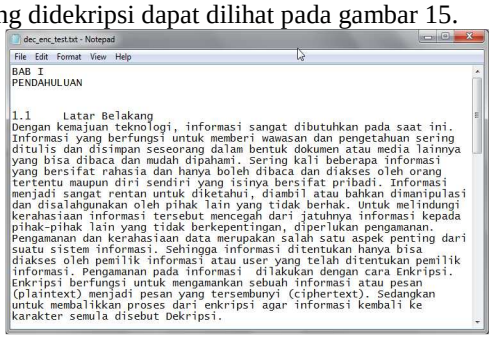
Gambar 13. Tampilan Proses Membuka Pengamanan Selesai

Dapat dilihat pada gambar 13 diperoleh waktu untuk membuka pengamanan *file* hasil yaitu 0.1248002 Second. Dan hasil dari proses adalah *file* dari hasil pemecahan dan *file* dari hasil pemecahan yang didekripsi. Adapun *file* dari hasil proses membuka pengamanan *file* hasil dapat dilihat pada gambar 14.



Gambar 14.*File* Dari Hasil Proses Membuka Pengamanan *File* Hasil

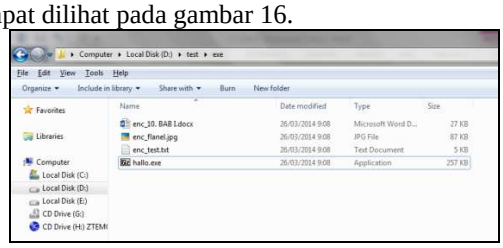
Nama *file* yang berawalan ‘dec’ adalah *file* yang didekripsi dan nama *file* yang berawalan ‘enc’ adalah *file* yang dienkripsi. Adapun tampilan salah satu dari *file* setelah proses membuka pengamanan yang didekripsi dapat dilihat pada gambar 15.



Gambar 15. Tampilan *File* Dari Hasil Dekripsi Setelah Proses Membuka Pengamanan *File* Hasil

Pengujian Pendekripsian File dari Hasil Pemecahan

File-file dari hasil pemecahan adalah *file* yang diperoleh ketika mengeksekusi langsung *file* hasil di penyimpanan komputer. Setelah mengeksekusi *file* hasil ‘hallo.exe’ dan diperoleh *file-file* yang dapat dilihat pada gambar 16.

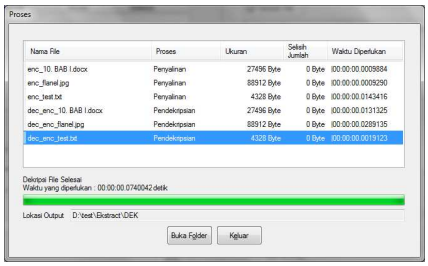


Gambar 16.*File* Dari Hasil Proses Membuka Pengamanan *File* Hasil

Adapun tampilan salah satu dari *file* setelah proses eksekusi langsung *file* hasil di penyimpanan komputer dapat dilihat pada gambar 17.

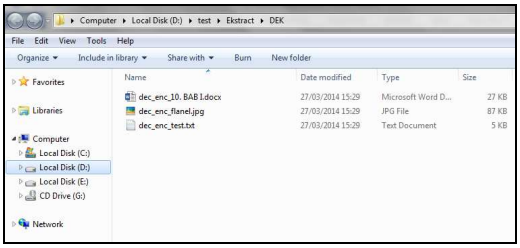
Gambar 17. Tampilan *File* Dari Hasil Didekripsi Setelah Proses Membuka Pengamanan *File* Hasil

Pada gambar 17 file dalam keadaan terenkripsi dan belum melalui proses dekripsi. Kemudian *file* diinputkan pada form ‘Unlocker’ pada tab ‘File-file’ dan setelah itu menginputkan kata kunci dekripsi dan lokasi folder *outputfile* dari hasil dekripsi. Setelah semua diinputkan, maka proses dekripsi*file* dimulai.

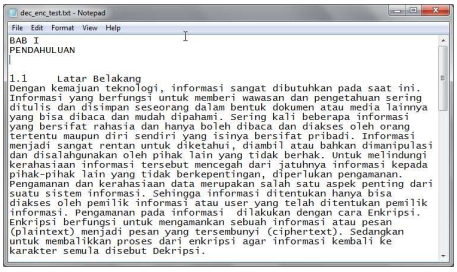


Gambar 18. Tampilan Proses Dekripsi Selesai

Pada gambar 18 dapat diperoleh bahwa waktu yang diperlukan untuk mendekripsi *file* yaitu 0,740042 detik. Nama *file* yang diberi awalan ‘dec’ menandakan bahwa *file* telah didekripsi. Proses dekripsi adalah untuk membalik dari proses enkripsi sehingga *file* dikembalikan menjadi *file* yang bisa diakses, dibaca dan dipahami lagi. Adapun file dari hasil pendekripsian dan tampilan file setelah proses dekripsi dapat dilihat pada gambar 19 dan gambar 20.



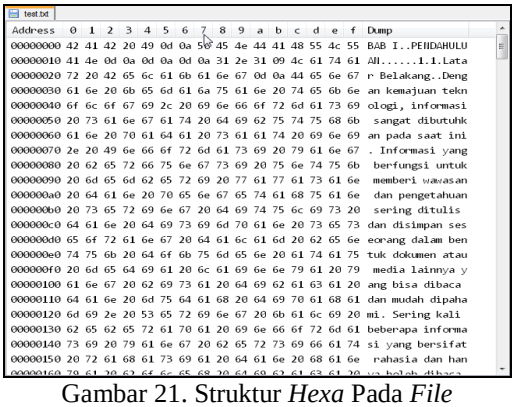
Gambar 19. *File* Dari Hasil Proses Dekripsi *File*



Gambar 20. Tampilan *File* Setelah Proses Dekripsi

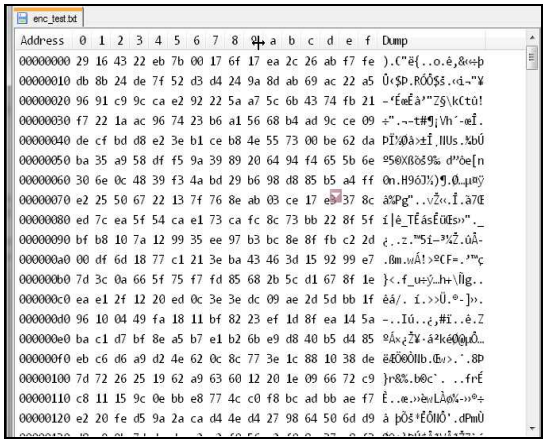
Perbandingan Struktur Hexa Pada File Di Setiap Tahap Proses

Pada pengujian ini adalah untuk melihat perbandingan perbedaan struktur *hexa* pada *file* saat proses pengamanan dan membuka pengamanan. Struktur *hexa* mengalami perubahan karena proses enkripsi dan dekripsi. Pengujian ini menggunakan salah satu *file* kemudian dibandingkan dengan kondisi *file* yang berbeda yaitu setelah dienkripisi dan didekripsi. Adapun struktur *hexa* pada *file* sebelum pengenkripsian dapat dilihat pada gambar 21.



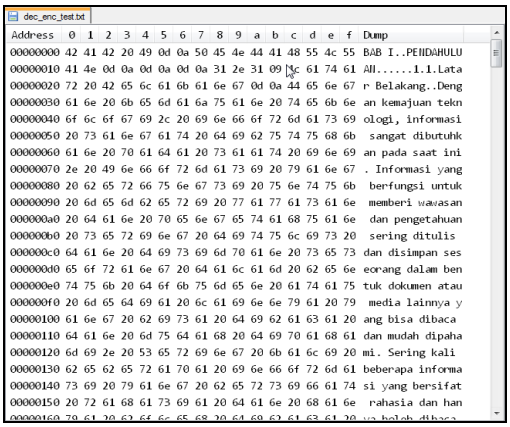
Gambar 21. Struktur *Hexa* Pada *File*

Untuk memperoleh *file* yang sudah terenkripsi yaitu dengan melakukan eksekusi pada *file* hasil di lokasi penyimpanan komputer dengan menginputkan *password* kemudian *file* hasil melakukan pemecahan sehingga *file-file* terenkripsi bisa diperoleh. Struktur *hexa* pada *file* terenkripsi memiliki banyak perubahan. Pada teksnya pun terlihat acak susah dipahami karena mengalami pengenkripsian. Adapun struktur *hexa* pada *file* setelah dienkripsi dapat dilihat pada gambar22.



Gambar 22. Struktur *Hexa* Pada *File* Setelah Dienkripsi

Dengan mengetahui kata kunci enkripsi maka file dapat didekripsi sehingga *file* dikembalikan menjadi file yang dapat diakses, dibaca dan dipahami lagi. Adapun struktur *hexa* pada *file* terenkripsi setelah didekripsi dapat dilihat pada gambar 23.



Gambar 23. Struktur *Hexa* Pada *File* Setelah Didekripsi

KESIMPULAN

Dari hasil penelitian yang telah dilakukan maka dapat diambil beberapa kesimpulan bahwa Rancang Bangun *Multifile Locker Application*Menggunakan Metode *Data Encryption Standar* berfungsi dengan cara yaitu satu *file*atau lebih diamankan dengan dua pengamanan yaitu pengamanan peram *file* dienkripsi dengan menerapkan metode *Data Encryption Standard* dan kemudian disatukan dalam satu *file* baru yaitu file hasil yang berekstensi EXE (*.exe) dengan tipe *executable file*. Pengamanan kedua *file* hasil dilindungi dengan*password*sehingga perlu mengetahui *password* saat membuka dan memecahnya.Proses enkripsi pada *file*menggunakan metode *Data Encryption Standard* mengubah struktur asli *hexa* dari *file* sehingga *file* dalam keadaan teracak dan sulit dipahami. Diperlukan proses dekripsi untuk

mengembalikan menjadi *file* yang dapat diakses, dibaca dan dipahami kembali.

[1] Budhiarto, A. 2006. "Pengamanan Enkripsi Berkas Menggunakan Algoritma Data Standar Enkripsi". **Skripsi**Jurusan Sistem Informasi, Fakultas IlmuKomputerUniversitas Gunadarma.

[2] Irwan, C. 2010. "Enkripsi Pada QR Code Tiket Dengan RSA".**Skripsi** Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung.

[3] Jogiyanto. 2005. *Analisis dan Desain*.Yogyakarta : Andi.

[4] Kadir, A. 2003. *Pengenalan Sistem Informasi*. Andi : Yogyakarta.

[5] Kendall dan Kendall. 2003.*AnalisisdanPerancanganSistem Edisi ke-5 Jilid 1*. Jakarta : PT. Prehallindo

[6] Munir, R. 2006. *Kriptografi*. Bandung : Informatika.

[7] Noor, R dan Effendi. 2005. "Supplement Chapter 12 File Management System : Sistem Berkas".Program Pasca Sarjana, Magister TeknologiInformasi,Universitas Indonesia, Jakarta.

[8] Wiederhold, G. 2001. "Database Design". Second Edition ; Restored for ACM.Chapter 5. Pp 2.

[9] http://www.thefreedictionary.com/lock(diakses pada 13.35, tanggal 31/12/2013)