

SISTEM PENGAMANAN DATA GAMBAR MENGGUNAKAN RC4 DAN EOF PADA MEDIA VIDEO MP4 BERBASIS JAVA DESKTOP PADA KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN

Abid Fikriyan¹⁾, Sri Mulyati²⁾

Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260
E-mail : 1511502781@student.budiluhur.ac.id¹⁾, Sri.mulyati@budiluhur.ac.id²⁾

Abstrak

Masalah keamanan dan kerahasiaan data merupakan hal yang sangat penting baik dalam suatu organisasi komersial, perguruan tinggi, lembaga pemerintahan, maupun individu. Keamanan data gambar pada Kementerian Pendidikan dan Kebudayaan akan sangat berharga bila di dalamnya terdapat sebuah data dan tidak boleh diketahui oleh orang lain. Sebab itu dibutuhkan aplikasi yang dapat menjaga kerahasiaan terhadap data tersebut. Oleh karena itu dibangun sebuah aplikasi yang memiliki tingkat keamanan yang cukup tinggi untuk menjamin keamanan dan kerahasiaan data – data penting pada Kementerian Pendidikan dan Kebudayaan. Hal ini dilakukan untuk menjaga agar data tersebut jangan sampai di baca apalagi dibajak atau diubah oleh pihak yang tidak berwenang. Sehingga untuk mencegah terjadinya hal – hal yang tidak di inginkan tersebut salah satu cara adalah dengan memanfaatkan kriptografi dan steganografi, yaitu salah satu teknik mengenkripsi data sehingga orang luar tidak dapat melihat data asli dari *file* gambar yang sudah di enkripsi dapat dilakukan dengan cara mengembalikannya ke bentuk semula yaitu dengan cara di dekripsi, sehingga data asli dapat di lihat seperti semula. Pada tugas akhir ini, kriptografi yang di gunakan adalah algoritma *Rivest Code 4* (RC4) dan steganografi dengan menggunakan metode *End Of File* (EOF). Tujuan di buatnya sistem keamanan *file gambar* atau data dalam gambar tersebut agar arsip tidak dicuri, mengalami kerusakan dan penyalahgunaan data. Aplikasi ini dibangun dengan menggunakan bahasa pemrograman *java* berbasis *desktop*. Dengan sistem keamanan *file gambar* ini di harapkan dapat melindungi data penting pada Kementrian pendidikan dan Kebudayaan.

Kata kunci: Kriptografi, RC4, Steganografi, EOF

1. PENDAHULUAN

Pesatnya perkembangan teknologi yang saat ini terjadi di dunia perkantoran maupun pemerintahan yang sangat bahaya apabila data-data yang tersebar luas jika di curi oleh seseorang atau pihak yang tidak mempunyai wewenang untuk memiliki data tersebut.

Jika data yang berupa file, gambar harus di amankan maka sebuah sistem yang akan mengamankan sebuah data-data tersebut. Tidak akan di ketahui oleh banyak khalayak, dan sulit di ketahui oleh orang lain. Kecuali yang mempunyai izin dan berkepentingan dalam melihat atau memiliki data tersebut.

Kementerian Pendidikan dan Kebudayaan adalah instansi pemerintah yang bergerak dalam bidang pendidikan dan kebudayaan dimana banyak file gambar hasil scan yang berupa surat menyurat, untuk kenaikan pangkat dan lain-lain. File gambar tersebut penting harus terjaga kerahasiaannya dari pihak yang tidak berhak untuk mengaksesnya. Salah satu cara yang digunakan untuk menjamin kamanan data atau informasi adalah dengan menggunakan steganografi dan kriptografi. Penerapan steganografi dan kriptografi pada Kementerian Pendidikan dan Kebudayaan akan difokuskan kepada pengamanan data *file gambar* yang tersimpan menjadi aman

sampai dengan dapat diakses oleh pihak yang berhak mengaksesnya

metode yang digunakan adalah metode algoritma kriptografi RC4 (*Rivest Code 4*) dengan kombinasi steganografi EOF (*End Of File*). metode steganografi EOF dengan kombinasi algoritma kriptografi RC4 dikarenakan teknik menyimpan atau menyamarkan keberadaan pesan rahasia dalam suatu media penampung dan sistem penyandian pada algoritma ini memiliki mekanisme kerja yang cukup sederhana dan mudah dimengerti namun tetap kokoh dalam tugasnya menyamarkan data.

Berdasarkan latar belakang masalah maka dapat dirumuskan sebagai masalah berikut:

- Bagaimana mengamankan isi data khususnya dalam bentuk *file gambar* dari pencurian dengan cara mengimplementasikan kriptografi data menggunakan algoritma enkripsi RC4 (*Rivest Code 4*) dan steganografi EOF (*End Of File*) ?.
- Bagaimana cara mengembalikan data yang telah di enkripsi menjadi data yang asli tanpa mengalami perubahan sedikitpun?.

Tujuan penulisan tugas akhir ini adalah membuat aman data rahasia perusahaan dari pihak yang tidak bertanggung jawab dan tidak berhak menerima informasi tersebut, dengan membuat aplikasi pengamanan data

menggunakan algoritma kriptografi RC4 dan steganografi EOF.

Aspek-aspek keamanan didalam kriptografi adalah:

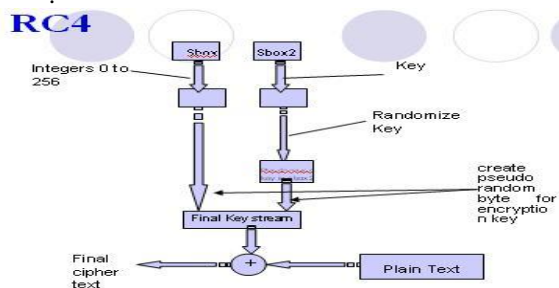
- 1) *Confidentiality* (Kerahasiaan)
 Layanan yang berfungsi untuk menjaga dan mengamankan agar data tidak dapat di lihat.
- 2) *Data Integrity* (Integritas)
 Layanan yang menjamin bahwa pesan aman apa tidak, apabila telah di kirim atau di terima. Mengalami perubahan atau tidak jika di buka data nya?"
- 3) *Authenticacion* (otentikasi)
 Layanan yang berhubungan dengan identifikasi, baik mengidentifikasi kebenaran pihak-pihak yang berkomunikasi (*user authentication* atau *entity authentication*) maupun mengidentifikasi kebenaran sumber pesan (*origin authentication*).
- 4) *Non-repudiation* (Nirpenyangkalan)
 Layanan untuk mencegah entitas yang berkomunikasi melakukan penyangkalan, yaitu pengirim pesan menyangkal melakukan pengiriman atau penerima pesan menyangkal telah menerima pesan[1].

2. METODE PENELITIAN

Ada banyak model dan metode enkripsi, salah satu diantaranya adalah enkripsi dengan metode algoritma *Rivest Code4* (RC4) .Model ini merupakan salah satu algoritma kunci simetris yang berbentuk *streamchipper* .Algoritma ini dibuat dan publikasikan pada tahun 1987 oleh *Ronald Rivest* dan terus berkembang sampai tahun 1994 sampai saat ini.

2.1. Studi Literatur

rangkaiannya proses yang dijalankan untuk mengenkripsi atau mendekripsi:



Gambar 1. Proses Enkripsi Dan Dekripsi RC4[1]

Hampir semua algoritma steganografi memakai kombinasi dari bidang jenis teknik untuk melakukan sebuah tugas dalam penyelubungan pesan rahasia dalam sebuah selubung berkas. Sebuah program steganografi dibutuhkan untuk melakukan hal-hal berikut (baik implisit melalui suatu perkiraan maupun eksplit melalui sebuah perhitungan),

menemukan kelebihan bit dalam selubung *file* yang dapat digunakan untuk menyelubungi pesan rahasia didalamnya, memilih beberapa diantaranya untuk digunakan dalam menyelubungi data dan penyelubungan data dalam bit dipilih sebelumnya.

Ini adalah gambar struktur untuk steganografi apabila dilihat dari bagian file.



Gambar 2. Struktur awal steganografi dengan EOF.

Setiap bagian pada sebuah data dapat kita baca dengan memakai dua parameter, posisi awal dan panjang bagian tersebut. Dengan struktur diatas, kita dapat membaca posisi awal [Data Header] yang isinya meliputi :

- a. Posisi awal [DATA] pada *file*.
- b. Panjang [DATA] pada *file*.

Bagaimana melihat posisi dan panjang data [Data Header] itu sendiri? Kita dapat menggunakan *looping* yang mencari penanda (*FLAG*) sebagai penentu posisi awal [Data Header] pada *file* media mulai dari awal *file*.

2.2. Analisis Masalah

Setiap instansi pemerintahan memiliki informasi yang sangat rahasia yang tidak boleh diketahui oleh orang lain, yang terdapat pada kementerian pendidikan dan kebudayaan adalah gambar dari hasil scan untuk surat kenaikan pangkat dan lain-lain. Kerahasiaan suatu informasi tersebut selalu menjadi masalah tersendiri bagi setiap lembaga pemerintahan dimana data dan informasi yang dihasilkan lembaga pemerintahan harus bisa diamankan. Tanpa keamanan yang maksimal, mengalami resiko kerugian akan terkena lembaga pemerintahan tersebut manakala data yang sangat penting dan rahasia tersebut hilang atau dicuri oleh orang yang tidak bertanggung jawab. Pertukaran informasi pada saat ini yang berkembang pesat membuat informasi semakin rawan untuk dicuri.

Sebuah solusi yang cukup ideal untuk mengatasi dan menyelesaikan permasalahan di atas adalah dengan membuat program aplikasi yang dapat mendukung keamanan dalam menjaga kerahasiaan *file* gambar pribadi, instansi atau organisasi dan lain sebagainya. Aplikasi tersebut nantinya dapat membuat sebuah *file* gambar tidak dapat terbaca karena isinya sudah diacak, lalu *file* gambar tersebut disembunyikan di dalam sebuah video sehingga data tersebut tidak bisa dilihat dengan kasat mata dan hanya diketahui oleh orang tertentu. Kemudian mengembalikan *file* gambar tersebut menjadi seperti semula tanpa mengalami perubahan sedikitpun.

Dengan adanya aplikasi ini diharapkan gambar hasil scan tersebut penting dapat disimpan dan dikirim ke pihak yang benar-benar berhak dan tidak disalahgunakan oleh pihak-pihak yang tidak bertanggung jawab.

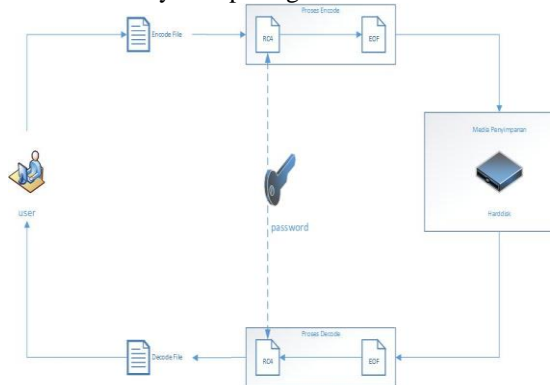
2.3. Perancangan Solusi

Aplikasi kriptografi dan steganografi yang sudah dibuat memiliki 5 buah bagian yaitu Menu *home*, menu *encode* dan *decode*, menu *help* dan menu *about*. menu *home* adalah tampilan awal dimana program ini dijalankan. didalam menu *encode* ada 4 pilihan *form* yaitu *form select file*, *form select video*, *form save directory* dan *form password* dan di dalam menu *decode* ada 3 pilihan *form* yaitu *form Select video*, *form Save Directory* dan *form password*. Menu *Help* adalah menu yang akan memberikan informasi kepada *user* tentang penggunaan aplikasi tersebut help memiliki 2 menu yaitu *encode* dan *decode*. Dan menu *about* akan memuat informasi tentang pembuat aplikasi.

Untuk membuat enkripsi, *user* harus menuju pada halaman *encode* dan mengisi semua form yang ada di halaman *encode* lalu membuat *password* untuk pengenkripan *File* tersebut. Setelah itu, *user* bisa melakukan proses enkripsi dengan menekan *Button Encode* yang telah disediakan di menu *encode*, setelah *file* gambar selesai di enkripsi untuk melakukan penyisipan kedalam *video* maka *user* dapat memilih tombol back untuk kembali kehalaman menu utama.

2.4. Rancangan Arsitektur Sistem

Encode adalah cara untuk menyembunyikan data yang telah di enkripsi dan di acak datanya. Rancangan yang akan dibuat dapat dilihat pada gambar aesitektur system pada gambar berikut ini.



Gambar 3. Rancangan alur sistem

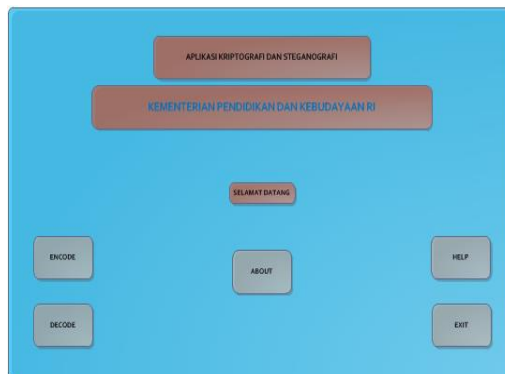
2.5. Rancangan Layar Aplikasi

Rancangan layar sangat penting dalam membuat suatu program. Oleh karena itu rancangan layar harus mudah dimengerti dan dipahami, agar dalam menggunakan program, pengguna merasa nyaman dalam menggunakannya sehingga rancangan layar tidak membuat bingung pengguna dan pengguna tidak mengalami kesulitan saat menggunakan program ini. Dalam program ini, akan digambarkan

rancangan layar masing-masing *form*, yaitu rancangan layar menu *home*, menu *encode* dan *decode*, menu *help* dan menu *About*.

a. Rancangan Layar Menu Utama

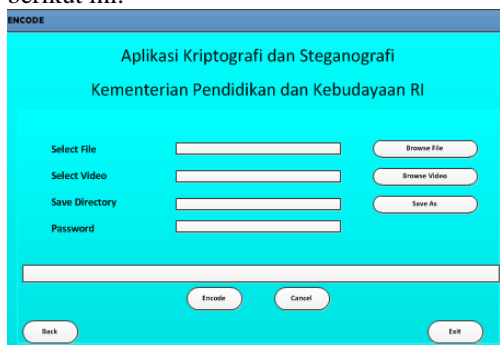
Fungsi Halaman *Home* berfungsi sebagai halaman awal yang memberikan akses kepada user dan menjadi *form* pertama yang ditampilkan pada saat *user* menggunakan aplikasi pengamanan data ini. Didalam layar halaman home terdapat beberapa pilihan menu yaitu *form encode*, *form decode*, *form about*, *form help* dan *form exit* seperti gambar berikut ini.



Gambar 5. Rancangan Layar Menu Utama

b. Rancangan Layar *encode*

Halaman *Encode* ini berfungsi untuk memilih *file* yang akan dienkripsi dan kemudian user harus memilih *file* video yang akan disisipi oleh *file* rahasia lalu *user* diminta untuk memilih *file directory* untuk menyimpan hasil dari proses enkripsi dan penyisipan *file* rahasia ke video lalu *user* diminta memasukkan *password* minimal 8 karakter agar *file* dapat memulai proses enkripsi dengan button yang telah disediakan pada halaman ini. Seperti gambar berikut ini.



Gambar 6. Rancangan Layar encode

c. Rancangan Layar *decode*

Halaman *Form file decode* adalah *form* yang berfungsi untuk melakukan dekripsi *file* atau mengubah *file* dari yang tidak terbaca menjadi ke bentuk semula dan memisahkan *file* dari video. *User* diminta untuk memilih sebuah video yang sudah disisipkan sebuah *file* tersembunyi dengan cara mengklik tombol

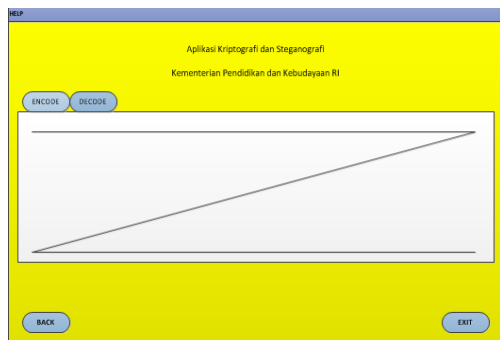
browse video. Setelah itu user harus memilih *file* directori tempat penyimpanan *file* hasil *decode* lalu *user* diminta memasukan *password* 8 digit yang telah dibuat Setelah itu *user* dapat mengklik tombol *decode* untuk menjalankan proses dekripsi. Tombol *cancel* untuk me-*reset* semua data *input*-an yang dimasukkan. Sedangkan menu *back* adalah untuk keluar dari *form* *file* *decode* dan kembali ke *form* menu utama. *Form* tersebut seperti gambar berikut ini:



Gambar 7. Rancangan Layar decode

d. Rancangan Layar *Help*

Halaman *Help* ini sebagai petunjuk bagi *user* agar dapat mempermudah dalam menjalankan aplikasi tersebut di halaman *help* ini terdapat *form* *help* *encode* dan *decode* Seperti gambar berikut ini.



Gambar 8. Rancangan Layar *Help*

e. Rancangan Layar *About*

Halaman layar *about* ini menjelaskan tentang program ini berisi tentang siapa yang membuatnya dan *button* *back* untuk keluar dari *form* tentang dan *button* *exit* untuk keluar dari aplikasi. Rancangan layar *Form* tentang terlihat seperti pada gambar di bawah ini.



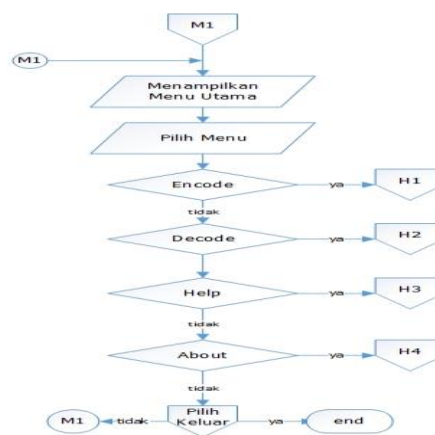
Gambar 9. Rancangan Layar *About*

2.6. *Flowchart*

Dalam pembuatan sistem aplikasi urutan proses pada aplikasi ini akan digunakan *flowchart* sebagai penjelasan. Dibawah ini akan diberikan sebagian *flowchart* aplikasi untuk masing-masing bagian.

a. *Flowchart* Menu Utama

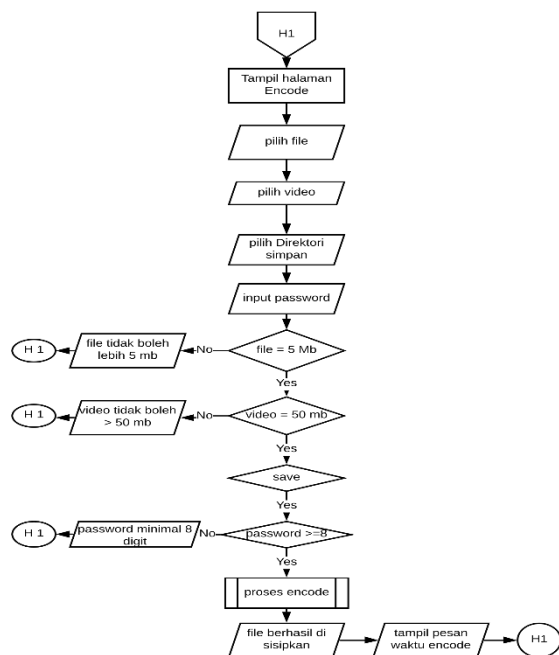
Flowchart layar *form* menu utama merupakan gambaran alur proses dari menu utama aplikasi. Pada proses ini, *user* dapat memilih aktivitas yang ingin dilakukan melalui menu-menu yang tersedia, berikut adalah gambar dari *flowchart* menu utama.



Gambar 10. *Flowchart* Menu Utama

b. *Flowchart* *encode*

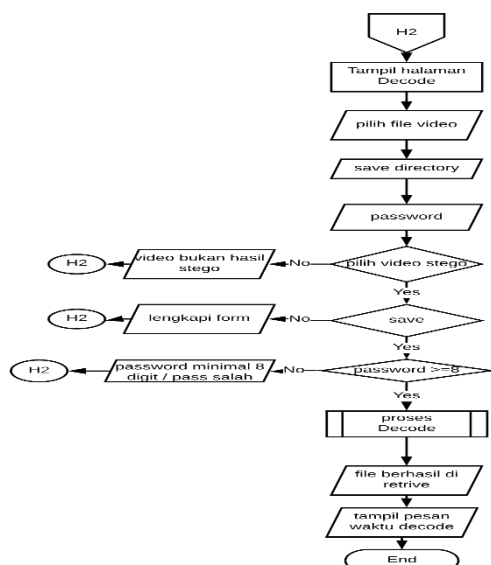
Flowchart Halaman *Encode* pada proses ini *user* dapat melakukan proses *encode* dan dapat melakukan penyisipan *file* rahasia didalam sebuah *file* video sebagai penampung *file* rahasia yang akan disisipkan. Kemudian *user* membuat *password*, Ketika *user* telah memasukan data dan memilih *button* *encode* maka proses *encode* akan berjalan, jika *user* meng klik *button* *cancel* maka proses *encode* akan digagalkan, apabila *user* memilih *button* *back* maka akan seperti ke menu utama dan jika *user* memilih *button* *exit* maka *user* akan keluar dari aplikasi. pada gambar. ini menunjukan *flowchart* pada *form* *encode* *file*.



Gambar 11. Flowchart Menu Encode

c. Flowchart Menu Decode

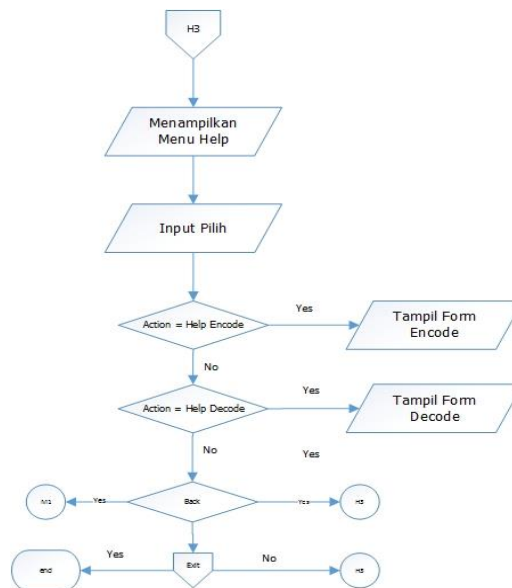
Flowchart layar form file decode merupakan gambaran alur proses dari form file decode. Pada proses ini user dapat menjalankan proses decode dengan cara memasukkan data yang ada pada panel input data decode. Data yang telah diacak dan disisipkan akan kembali seperti awal sebelum di-encode dan akan terlepas dari sisipan video. jika user memilih button cancel maka proses decode akan dibatalkan, apabila user memilih button back maka akan kembali ke menu utama dan jika user memilih button exit maka user akan keluar dari aplikasi. pada gambar ini menunjukan flowcart pada form decode file.



Gambar 12. Flowchart Menu Decode

d. Flowchart Menu Help

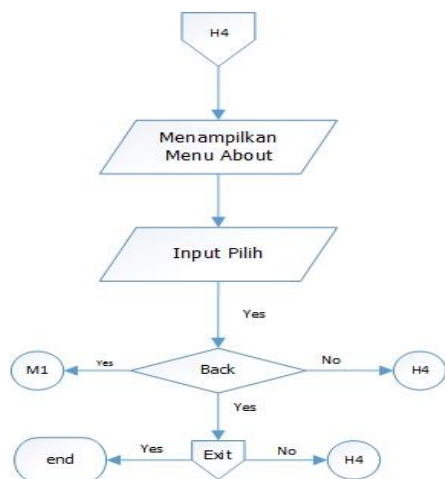
Halaman Help ini sebagai petunjuk bagi user agar dapat mempermudah dalam menjalankan aplikasi tersebut terdapat 2 sub menu yaitu menu encode dan decode apabila user memilih button back maka akan kembali ke menu utama dan jika user memilih button exit maka user akan keluar dari aplikasi. Seperti gambar ini menunjukan flowchart pada form help.



Gambar 13. Flowchart Menu Help

e. Flowchart Menu About

Halaman ini merupakan identitas program, yang berisi tentang judul program, nama pembuat program serta instansi yang bersangkutan, apabila user memilih button back maka akan kembali ke menu utama dan jika user memilih button exit maka user akan keluar dari aplikasi. Seperti pada gambar berikut.



Gambar 14. Flowchart halaman About

3. HASIL DAN PEMBAHASAN

Agar aplikasi dapat berjalan sesuai dengan harapan, maka diperlukan pembuatan *interface* yang menarik dan dilakukannya berbagai macam pengujian. Berikut adalah *interface* dan beberapa pengujian yang telah dilakukan terhadap aplikasi.

3.1. Tampilan Layar

Pada bagian ini, akan diuraikan mengenai tampilan layar aplikasi kriptografi ini mulai dari pertama kali dijalankan sampai selesai dijalankan. Berikut ini akan diberikan penjelasan dan gambar mengenai tampilan-tampilan yang ada pada aplikasi ini.

a. Tampilan Layar Menu Utama

Tampilan layar dari *form* Menu Utama pada gambar dibawah ini muncul pada saat pertama kali aplikasi dijalankan dan terlihat beberapa menu yang bisa digunakan seperti *Encode*, *Decode*, *About*, *Help* dan *Exit* untuk keluar dari aplikasi.



Gambar 14. Tampilan Layar Menu Utama

b. TampilanLayar *Encode*

Tampilan layar dari *form Encode File* pada gambar dibawah ini muncul pada saat menu *Encode File* dijalankan.



Gambar 15. Tampilan Layar Menu *Encode*

c. Tampilan Layar Menu *Decode*

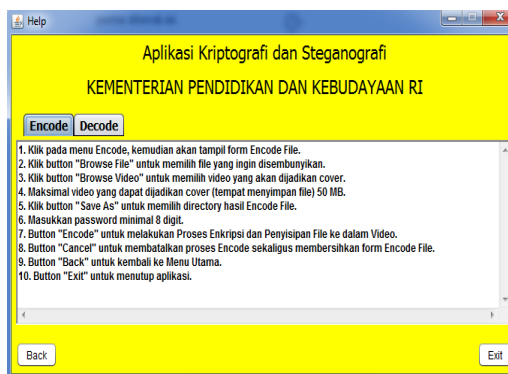
Tampilan layar dari *Form Decryption File* pada gambar dibawah ini muncul pada saat menu *Decode File* dijalankan.



Gambar 16. Tampilan Layar Menu *Decode*

d. Tampilan Layar *Help*

Pengguna dapat melihat informasi cara penggunaan aplikasi dengan melihat tampilan layar *form help* ini yang dibagi menjadi dua. Tampilan layar *form help* dapat dilihat pada gambar dibawah ini.



Gambar 17. Tampilan Layar *Help*

e. Tampilan Layar Menu *About*

Pada tampilan layar *form About*, pengguna dapat melihat informasi tentang aplikasi kriptografi ini. Tampilan layar *form about* dapat dilihat pada gambar berikut.

Gambar 18. Tampilan Layar *About*

3.2. Pengujian Kecepatan Aplikasi

Dalam pengujian kali ini akan dibahas perbandingan antara proses *encode* dan *decode* antara *file* gambar **jpg* dan **png*. Pengujiannya yaitu meliputi ukuran awal *file* yang ingin *encode*, ukuran *file* yang telah di *encode* dengan algoritma RC4.

Waktu proses *encode*, waktu proses *decode* dan hasil yang dicapai dalam proses *encode* maupun *decode*. Adapun hasil uji coba yang dilakukan tercatat pada Tabel 4.1 untuk uji coba proses *encode* dan untuk uji coba hasil proses *decode*, seperti berikut.

Tabel 1. Pengujian Kecepatan Aplikasi

No.	Proses	Waktu yang Dibutuhkan
1.	Membuka aplikasi	33.02 detik
2.	Membuat <i>file encode</i> (1)	2,334 detik
3.	Membuat <i>file decode</i> (1)	15,327 detik
4.	Membuat <i>file encode</i> (2)	6,248 detik
5.	Membuat <i>file decode</i> (2)	36,744 detik

3.3. Evaluasi Aplikasi

Berdasarkan hasil uji coba yang di lakukan dengan *file* format JPG dan PNG dari sistem yang sudah berjalan yaitu sebagai berikut :

- a. Kelebihan Aplikasi:
 1. Dapat dijalankan dengan ekstensi **.jpg* dan **.png*
 2. *File* yang sudah di *encode* tidak dapat dibaca atau dibuka kembali sebelum di *decode*.
 3. Isi *file* dari hasil *decode* tidak membuat perubahan atau seperti *file* asli.
- b. Kekurangan Aplikasi:
 1. Semakin besar size sebuah data maka prosesnya akan semakin lama.
 2. Waktu yang diperlukan untuk proses *decode* masih kurang cepat.
 3. Ukuran *file* yang dienkripsi hanya dibatasi 5 MB saja.

4. Ukuran *File* video sebagai media penyisipan hanya dibatasi 50 MB saja.

3.4 Hasil Uji Coba Program

Input File	Size file Input	Nama File Video	Size File Video	Size File Hasil Encode	Waktu Proses Encode	Size File Hasil Decode	Waktu Proses Decode
Gambar sertifikat PNS untuk naik pangkat	1,29 MB	VIDEO MP4. (masjid)	25,4 MB	32,10 MB	2,334 detik	1,87 MB	15,327 detik
Gambar surat pengajuan dana	1,24 MB	VIDEO MP4. (rumah)	29,6 MB	35,58 MB	6,248 detik	1,66 MB	36,744 detik

4. KESIMPULAN

Berdasarkan perancangan, pembuatan, serangkaian uji coba dan analisa program dari aplikasi kriptografi ini, maka dapat diambil suatu kesimpulan antara lain :

- a. Dengan adanya aplikasi kriptografi, proses penyimpanan dan pertukaran informasi menjadi lebih aman.
- b. Proses dekripsi akan mengembalikan *file* seperti semula tanpa ada perubahan sedikitpun dengan menggunakan password yang sesuai.
- c. Waktu yang digunakan untuk melakukan proses enkripsi dan dekripsi berbanding lurus dengan ukuran *file* yang diproses (semakin kecil ukuran *file* yang diproses, semakin cepat proses enkripsi dan dekripsi dilakukan, semakin besar ukuran *file* yang diproses, semakin lama proses enkripsi dan dekripsi dilakukan).
- d. Sebaiknya untuk memudahkan proses enkripsi dan dekripsi menggunakan kapasitas RAM yang besar agar lebih cepat untuk melakukan proses enkrip dan dekrip pada *file*.

5. Saran

Adapun saran yang mungkin diperlukan untuk membuat aplikasi ini dapat berjalan lebih baik lagi antara lain :

- a. Aplikasi ini diharapkan dapat ditingkatkan kinerjanya sehingga tidak hanya dapat mengenkripsi *file* dengan ekstensi dokumen dan gambar saja, namun bisa juga untuk *file* video maupun audio.
- b. Waktu proses enkripsi dan dekripsi *file* yang rata-rata berukuran besar diharapkan dapat berjalan lebih cepat pada *hardware* yang lebih baik.

Dikembangkan menggunakan algoritma kompresi, agar ukuran *file* hasil enkripsi diharapkan dapat menjadi lebih kecil

DAFTAR PUSTAKA

- [1] Ariyus, D. (2008). Pengantar Ilmu Kriptografi: teori, analisis dan implementasi. Yogyakarta: ANDI.
- [2] Kusumaningsih, D., Pudoli, A., & Rahmadan, I. (2017). Steganografi Metode End Of File Untuk Keamanan Data, *9*(1), 47–55.
- [3] Bayu, A. E. (2014). Implementasi Keamanan Data Menggunakan Metode (End Of File) Pada PT Telekomunikasi Selular. Perpustakaan Universitas Budi Luhur, *5*, 53.
- [4] Hakim, E. L., Khairil, & Utami, F. H. (2014). Aplikasi Enkripsi Dan Deskripsi Data Menggunakan Algoritma Rc4 Dengan Menggunakan Bahasa Pemrograman Php. *Jurnal Media Infotama*, *10*(1), 1–7.
- [5] Munir, R., 2004. Sistem Kriptografi Kunci-Publik Departemen Teknik Informatika Institut Teknologi Bandung.
- [6] Marbun, M. (2015). Implementasi Sistem Pengamanan Data Barang Pada Pt . Matahari Putra Prima, Tbk. *Jurnal Mantik Penusa*, *18*(2), 1–10.