# Implementasi Keamanan File dengan Kompresi Huffman dan Kriptografi menggunakan Algoritma RC4 serta Steganografi menggunakan End of File Berbasis Desktop pada SMK Negeri 3 Kota Tangerang

Nurhardian[#1], Ahmad Pudoli[#2]

[#]*Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur*
*Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260*
*Telp. (021) 5853753, Fax. (021) 5866369*

[1]ardy.game@gmail.com

[2]ahmad.pudoli@explorindo.com

***Abraksi - Sekolah SMK Negeri 3 Kota Tangerang merupakan*** **salah satu sekolah favorit di Kota Tangerang, sekolah ini mempunyai 5 (lima) jurusan diantaranya Akomodasi Perhotelan, Tata Busana, Tata Boga, Tata Kecantikan dan Teknik Komputer dan Jaringan. Masalah keamanan dan kerahasiaan data soal merupakan hal yang sangat penting bagi sekolah ini. Dalam pembuatan dan penyimpanan soal-soal tersebut tentunya membutuhkan kerahasiaan antara pihak murid dan guru agar tetap terjaga dengan baik dan tidak mengalami pencurian ataupun kecurangan oleh pihak yang tidak berhak. Untuk itu dibutuhkan aplikasi keamanan data terhadap soal tersebut supaya terhindar dari hal-hal yang tidak diinginkan. Salah satu cara untuk mengamankan file soal maka dibangunlah sebuah aplikasi yang memiliki tingkat keamanan yang cukup tinggi untuk menjamin keamanan dan kerahasian data soal pada sekolah SMK Negeri 3 Kota Tangerang. Sehingga untuk mencegah terjadinya hal-hal yang tidak diinginkan tersebut salah satu cara adalah dengan memanfaatkan kompresi Huffman, kriptografi RC4 dan steganografi EoF, yaitu salah satu teknik mengkompresi, mengenkripsi dan menyisipkan data sehingga orang luar tidak dapat melihat data asli dari file soal yang disembunyikan. File soal yang sudah di-embed dapat dikembalikan seperti semula yaitu dengan cara di-retrieve, sehingga data asli dari file soal yang di-embed dapat dilihat seperti semula. Pada penelitian ini, kompresi yang digunakan adalah Huffman, kriptografi yang digunakan adalah algoritma *Rivest Code* 4 (RC4) dan steganografi dengan menggunakan metode End Of File (EOF). Tujuan dibuatnya sistem keamanan file atau data yang berupa text, Microsoft Word, Microsoft Excel dan PDF dari pencurian, kerusakan dan penyalahgunaan data tanpa merusak keaslian file soal tersebut. Aplikasi ini dibangun dengan menggunakan bahasa pemrograman java berbasis desktop. Dengan aplikasi keamanan file ini diharapkan dapat melindungi data penting pada Sekolah SMK Negeri 3 Kota Tangerang.**

**Kata kunci : Kriptografi, Rivest Code 4, Steganografi, End Of File, Kompresi, Huffman**

## I. PENDAHULUAN

Seiring dengan perkembangannya teknologi dan komunikasi yang begitu pesat, memudahkan kita untuk melakukan pertukaran dengan data orang lain secara cepat. Namun terkadang keamanan dalam pertukaran data tersebut kurang disadari oleh kita sehingga terjadi pencurian data. Dengan adanya pencurian data maka aspek keamanan dalam pertukaran informasi serta penyimpanan data dianggap penting.

Sekolah saat ini telah menjadi aset penting yang mana disekolah itu sendiri memiliki informasi dan *file-file* yang sangat rahasia. Sekolah Menengah Kejuruan (SMK) 3 Kota Tangerang merupakan sekolah kejuruan yang memiliki jurusan Perhotelan, Tata Kecantikan, Tata Boga, Tata Busana, Teknik Komputer dan Jaringan. Khususnya pada Bidang Tata Usaha bagian admin yang menyimpan *file* soal yang harus terjaga keamanannya dari pihak yang tidak bertanggung jawab. Banyak *file* soal yang bersifat rahasia dan tidak bisa dipergunakan atau dirubah oleh pihak yang tidak berhak. Untuk mengamankan *file* maka dapat menggunakan kriptografi, steganografi dan kompresi. Oleh Karena itu, pengguna *file* soal membutuhkan bantuan untuk keamanan

akan *file* soal yang disimpannya. Penerapan kriptografi pada SMKN 3 Kota Tangerang akan difokuskan bagaimana kriptografi dan steganografi dapat mengamankan *file* soal yang tersimpan menjadi aman sampai dengan dokumen dibuka oleh pihak yang berhak untuk membukanya.

Berdasarkan uraian diatas, penulis ingin membangun suatu aplikasi keamanan data dengan kompresi Huffman yang berfungsi untuk memperkecil ukuran data dari data aslinya, algoritma kriptografi RC4 yang merupakan metode penyandian pesan teks yang melakukan enkirpsi per-*bit* dan dikombinasikan dengan menggunakan metode steganografi EOF (*End of File*) yang merupakan salah satu teknik yang menyisipkan data pada akhir *file*, untuk mengamankan sebuah *file* soal agar tidak bisa dibaca oleh orang lain selain pemilk *file* tersebut dan menghasilkan aplikasi pengamanan dokumen berbasis *java desktop* yang mudah digunakan oleh pengguna.

## II. LANDASAN TEORI

### 2.1. Kompresi Data

Kompresi data (pemampatan data) merupakan suatu teknik untuk memperkecil jumlah ukuran data (hasil kompresi) dari data aslinya. Pemampatan data umumnya diterapkan pada mesin komputer, hal ini dilakukan karena setiap simbol yang dimunculkan pada komputer memiliki nilai bit-bit yang berbeda. Misal pada ASCII setiap simbol yang dimunculkan memiliki panjang bit 8 bit, misal kode A pada ASCII mempunyai nilai desimal = 65, jika dirubah dalam bilangan biner menjadi 010000001. Pemampatan data digunakan untuk mengurangkan jumlah bit-bit yang dihasilkan dari setiap simbol yang muncul. Dengan pemampatan ini diharapkan dapat mengurangi (memperkecil ukuran data) dalam ruang penyimpanan[1].

### 2.2. Algoritma Huffman

Pengkodean dengan metode Huffman dibangun dari panjang variabel kode-kode yang disusun dari bit-bit. Simbol dengan probabilitas yang tinggi akan memperoleh kode-kode paling pendek sedangkan simbol dengan probabilitas paling rendah akan memperoleh kode terpanjang. Contoh untuk string 'NURHARDIAN' mempunyai panjang bit sebanyak 80 bit karena 1 karakter dikodekan dengan 8 bit (ASCII) akan diperoleh jumlah bit untuk tiap simbolnya dengan jumlah yang lebih sedikit atau bitnya lebih pendek yaitu 27 bit, sehingga secara otomatis ukuran filenya berkurang [1].

Kode Huffman digunakan secara luas dan sangat efektif untuk kompresi data. Bisa menghemat 20%-90% dari ukuran semula, tergantung tipe karakter yang dikompresi. Algoritma huffman menggunakan tabel yang menyimpan frekuensi kemunculan dari masing-masing simbol yang digunakan dalam file tersebut dan kemudian mengkodekannya dalam bentuk biner [2].

### 2.3. Kriptografi

#### 2.3.1. Definisi Kriptografi

Kata Kriptografi berasal dari bahasa Yunani yang terdiri dari 2 (dua) buah kata yaitu *crypto* dan *graphia.* Kata *crypto* berarti *secret* (rahasa) sedangkan *graphia* berarti *writing* (tulisan). Berarti secara umum makna dari kata kriptografi adalah tulisan rahasia. Kriptografi pada awalnya dijabarkan sebagai ilmu yang mempelajari bagaimana cara menyembunyikan pesan. " Kriptografi adalah Sebuah teknik rahasia dalam penulisan, dengan karakter khusus, dengan mengguanakan huruf dan karakter di luar bentuk aslinya, atau dengan metode-metode lain yang hanya dapat dipahami oleh pihak-pihak yang memproses kunci, juga semua hal yang ditulis dengan cara seperti ini." Jadi, secara umun dapat diartikan sebagai seni menulis atau memecahkan cipher [3].

#### 2.3.2. Sejarah Kriptografi

Kriptografi mempunyai sejarah yang panjang dan menakjubkan. Pada zaman Romawi Kuno, telah ada alat untuk mengirim pesan rahasia dengan nama *Scytale* yang digunakan oleh tentara Sparta. *Scytale* merupakan alat yang memiliki pita panjang dari daun *Papyrus* dan sebatang silinder. Pesan ditulis diatas pita yang dililitkan dari batang silinder lalu dikirim. Untuk membaca pesan, pita tersebut dililitkan kembali pada sebatang silinder yang diameternya sama sehingga yang menjadi kunci pada *Scytale* adalah diameter silindernya.

#### 2.3.3. Tujuan Kriptografi

Aspek-aspek keamanan didalam kriptografi adalah :

1) *Confidentiality* (kerahasiaan)

Kerahasiaan menjamin data-data tersebut hanya bisa diakses oleh pihak-pihak tertentu saja. Kerahasiaan bertujuan untuk melindungi suatu informasi dari semua pihaik yang tidak berhak atas informasi tersebut.

2) *Authentication* (Otentikasi)

Otentikasi merupakan identifikasi yang dilakukan oleh masing-masing pihak yang saling berkomunikasi. Penerima pesan dapat memastikan keaslian pengirimnya.

3) *Integrity* (Integritas)

Integritas menjamin setiap pesan yang dikirim pasti sampai pada penerimanya tanda ada bagian dari pesan tersebut yang diganti, diduplikasi, dirusak, diubah urutannya dan ditambahkan. Integritas data bertujuan untuk mencegah terjadinya pengubahan informasi oleh pihak-pihak yang tidak berhak atas informasi tersebut.

4) *Non – Repudiation* (Tanpa Penyangkalan)

Pengirim tidak mengelak bahwa dia telah mengirim pesan, penerima juga tidak dapat mengelak bahwa dia telah menerima pesan tersebut.

### 2.4. Algoritma Kriptografi

Algoritma kriptografi merupakan langkah-langkah logis bagaimana menyembunyikan pesan dari orang-orang yang tidak berhak atas pesan tersebut dengan melakukan pembangkitan kunci, enkripsi dan dekripsi. Dalam beberapa

metode kriptografi terdapat perbedaan antara fungsi enkripsi dan fungsi dekripsi.[4]

### 2.5. Algoritma Rivest *Code* 4 (RC4)

#### 2.5.1. Sejarah Singkat RC4

RC4 pertama kali didesain oleh Ron Rivest yang berasal dari Laboratorium RSA pada tahun 1987.RC sendiri mempunyai singkatan resmi yaitu "*Rivest Chiper*", namun juga dikenal sebagai "*Ron's Code*". RC4 sebenarnya dirahasiakan dan t idak dipublikasikan kepada khalayak ramai, akan tetapi pada September 1994, kode tersebut dikirim oleh seseorang yang tidak diketahui ke milist Chypermunks dan menyebar ke banyak situs *internet*. Kode yang bocor tesebut akhirnya dikonfirmasi sebagai RC4 karena memiliki *output* yang sama dengan *software* dengan lisensi RC4 di dalamnya. Karena algoritma sudah diketahui, RC4 tidak lagi menjadi rahasia dagang. Nama RC4 sudah dipatenkan, sehingga sering disebut sebagai "ARCFOUR" atau "ARC4" (*Alleged* RC4) untuk menghindari pematenan. RSA *Security* tidak pernah secara resmi merilis algoritma tersebut, namun Rivest secara pribadilah yang merilisnya dengan menghubungkan Wikipedia Inggris ke catatan-catatan yang ia punya.

#### 2.5.2. Deskripsi Mengenai RC4

RC4 merupakan metode penyandian pesan teks yang melakukan enkripsi per bit sehingga kelebihan dari metode ini kerusakan pada satu bit tidak mempengaruhi keseluruhan isi pesan. Pada RC4 dihasilkan pseudo random stream bit. Seperti halnya stream cipher lainnya, algoritma RC4 ini dapat di gunakan untuk mengenkripsi dengan mengkombinasikannya dengan plainteks menggunakan *Excusive-or* (Xor). Untuk proses dekripsi dilakukan cara yang sama dengan kunci yang sama, karena Xor merupakan fungsi simetrik. Secara garis besar proses algoritma RC4 dibagi menjadi dua bagian, yaitu *Key Scheduling Algorithm* (KSA) dan *Pseudo Random Generation Algorithm* (PRGA) [5].

#### 2.5.3. Algoritma Enkripsi RC4

RC4 adalah *cipher* aliran yang digunakan secara luas pada sistem keamanan seperti protokol SSL (*Secure Socket Layer*). Algoritma kriptografi ini sederhana dan mudah diimplementasikan. RC4 dibuat oleh Ron Rivers dari Laboratiorium RSA (RC adalah singkatan dari Ron's *Code*). RC4 membangkitkan aliran kunci (*keystream*) yang kemudian di-XOR-kan dengan *plaintext* pada waktu enkripsi (atau di-XOR-kan dengan *bit-bit ciphertext* pada waktu dekripsi) [6].

1) Inisialisasi *array* S-*box* pertama, S[0],S[1],.....,S[255]. Diisi dengan bilangan 0 sampai 255, sehingga *array* S-*box array* S berbentuk S[0]=0, S[1] = 1,...., S[255]=255.
   For r = 0 to 255
   S[r]=r
2) Inisialisasi *array* kunci (S-*box* lain), misal *array* kunci K dengan panjang 256. Jika panjang kunci K < 256, maka di lakukan *padding* yaitu penambahan *byte* sehingga panjang kunci menjadi 256 *byte*. Misalnya K = "abc" yang hanya terdiri dari 3 *byte* (3 huruf), maka lakukan *padding* dengan penambahan *byte* (huruf) semu, misalnya K = "abcabcabcabc....." sampai panjang K mencapai 256 *byte*, sehingga S-*box Array* kunci K berbentuk K[0],K[1],....,K[255].
   For i = 0 to 255
   K[i] = Kunci [i mod length];
3) Permutasi terhadap nilai-nilai di dalam *array* S dengan cara menurkarkan isi *array* S[I] dengan S[j], prosesnya adalah sebagai berikut :
   j = 0
   For i = 0 to 255
   j = (j +S[i] + K[j]) mod 256
   Isi S[i] dan isi S[J] ditukar
4) Membangkitkan aliran kunci (*keystream*) selanjutnya digunakan untuk enkripsi.
   i = j = 0
   i = (i + 1) mod 256
   j = (j + S[j]) mod 256
   isi S[i] dan S[j] ditukar
   t = (S[i] + S[j]) mod 256
   K = S[t];
5) Kunci aliran K kemudian digunakan untuk mengenkripsi *plaintext* ke-idx sehingga didapatkan *ciphertext*, sedangkan untuk mendapatkan *plaintext* dengan cara *ciphertext* di-XOR-kan dengan kunci yang sama dengan proses enkripsi.

### 2.6. Steganografi

#### 2.6.1. Pengertian Steganografi

Steganografi berasal dari bahasa Yunani yaitu *stegos* yang berarti penyamaran dan *graphia* yang berarti tulisan. Steganografi digunakan untuk menyembunyikan informasi rahasia ke dalam suatu media sehingga keberadaan pesan tersebut tidak diketahui oleh orang lain. Steganografi bertujuan untuk menghilangkan kecurigaan dengan cara menyamarkan pesan tersebut. [7]

#### 2.6.2. Sejarah Steganografi

Steganografi berasal dari bahasa Yunani yang berarti tertutup atau tulisan ternsembunyi. Steganografi sudah dikenal sejak 440 SM. Herodutus menyebutkan salah satu contoh steganografi adalah Histiaeus mencukur kepada budak yang paling dipercayainya dan mentatokan sebuah pesan di atasnya. Setelah rambutnya tumbuh, kemudian budak tersebut diutus untuk membawa pesan rahasia di balik rambutnya.[8]

#### 2.6.3. Tujuan Steganografi

Tujuan dari steganografi adalah merahasiakan atau menyembunyikan keberadaan dari sebuah pesan tersembunyi atau sebuah informasi. Dalam prakteknya, kebanyakan pesan disembunyikan dengan membuat perubahan tipis terhadap data *digital* lain yang sisinya tidak akan menarik perhatian dari penyerang potensial, sebagai contoh sebuah gambar yang terlihat tidak berbahaya. Perubahan ini bergantung pada kunci (sama pada kriptografi).

Dan pesan untuk disembunyikan orang yang menerima gambar kemudian dapat menyimpulkan informasi terselubung

dengan cara mengganti kunci yang benar ke dalam algoritma yang digunakan.[9]

### 2.7. Metoda End of File

Metode *End of File (EOF)* merupakan salah satu teknik yang menyisipkan data pada akhir *file*. Teknik ini dapat digunakan untuk menyisipkan data yang ukurannya sama dengan ukuran *file* sebelum disisipkan kedalam *file* tersebut. Dalam teknik EoF, data yang disisipkan pada akhir *file* diberi tanda khusus sebagai pengenal *start* dari data tersebut dan pengenal akhir dari data tersebut.[10]

### 2.8. Steganografi Pada Media Audio

Steganografi dapat diimplementasi pada media audio digital. Ketika berurusan dengan transmisi sinyal audio, ada hal utama yang harus diperhatikan, yaitu bentuk representasi audio memiliki dua karakteristik utama, yaitu *sampe quantiztion method* (metode quantisasi) dan temporal *sample rate*. Metoda quantisasi menyatakan representasi sampel audio berdasarkan kualitas digitalnya, misalnya WAV (*Windown Audio Visual*). Temporal *sampling rate* yaitu kecepatan yang dapat dihitung untuk melakukan *sampling* (pengambilan sampel) audio secara periodik. [11]

## III. ANALISIS MASALAH DAN RANCANGAN PROGRAM

### 3.1. Analisis Masalah

Dokumen soal merupakan data yang sangat penting bagi sekolah SMK Negeri 3 Kota Tangerang. Oleh karena itu, sebuah dokumen seharusnya dijaga keasliannya dan kerahasiaannya agar tidak disalahgunakan oleh orang yang tidak bertanggung jawab. Dikarenakan keamanan dokumen disini masih sangat kurang, sehingga terjadinya pencurian dokumen oleh orang yang tidak bertanggung jawab dan menyebarkannya. Salah satu cara untuk mengamankan sebuah dokumen yaitu dengan mengubah dokumen asli menjadi dokumen yang tidak bisa dibaca oleh orang lain atau sering disebut dengan enkripsi.

### 3.2. Penyelesaian Masalah

Untuk memecahkan masalah diatas, maka dibuatlah aplikasi pengamanan data yang dapat menjaga kerahasiaan dari orang yang tidak bertanggung jawab. Aplikasi tersebut nantinya dapat mengefisienkan penyimpanan, dokumen tersebut akan dikompresi. Mengubah sebuah *file* dokumen menjadi *file* yang isinya tidak bisa dibaca dan dokumen tersebut terjaga kerahasiaannya. Lalu, untuk menyembunyikan *file* yang sudah terjaga kerahasiaannya maka disisipkan ke dalam audio. Kemudian mengembalikan dokumen tersebut menjadi seperti semula tanpa mengalami perubahan sedikitpun.

### 3.3. Kebutuhan Sistem

Kebutuhan sistem yang akan dibangun pada aplikasi ini adalah sebagai berikut :

a. Proses pengamanan data dilakukan menggnakan aplikasi berbasis *desktop*.
b. Keamanan dan konsistensi ini data harus terjamin.
c. Aplikasi mampu mengefisienkan media penyimpanan dokumen dengan cara dikompresi.
d. Aplikasi mampu mengubah data asli menjadi data acak dan juga mampu mengembalikan data acak tersebut kembali menjadi data asli tanpa adanya perubahan pada isi data tersebut.
e. Aplikkasi mampu menyisipkan data acak ke dalam audio dan juga mampu mengembalikan data acak tersebut.

### 3.4. Analisa Kebutuhan Sistem

Adapun analisa kebutuhan sistem adalah sebagai berikut :

a. Aplikasi dapat memberikan fungsi otentifikasi *user* melalui proses *login*.
b. Aplikasi dapat memberikan layanan proses kompresi (pemampatan data).
c. Aplikasi dapat memberikan layanan proses enkripsi (pengacakan isi data).
d. Aplikasi dapat memberikan layanan proses embeed (penyisipan data ke dalam audio).
e. Aplikasi dapat memberikan layanan proses (mengembalikan data ke dalam bentuk file enkripsi).
f. Aplikasi dapat memberikan layanan proses dekripsi (mengembalikan isi data seperti semula).
g. Aplikasi dapat memberikan layanan dekompresi kompresi (pengembalian ukuran data).

### 3.5. Komponen Yang Digunakan

Komponen yang digunakan sebagai uji coba dan penelitian ini meliputi perangkat lunak dan perangkat keras, sebagai berikut.

a. Perangkat Keras

Dalam merancang dan membuat aplikasi kriptografi dan stegano *file* berbasis *dekstop* ini, dibutuhkan perangkat keras agar aplikasi dapat berjalan dengan baik. Spefisikasi perangkat keras yang digunakan adalah sebagai berikut :

1) Processor Intel(R) Core i3-4030U, 1.90GHz
2) Memory DDR3 4GB
3) Display 14” 1366 x 768 (64-bit)
4) Hard Drive 500GB data

b. Perangkat Lunak

Perangkat lunak yang digunakan terdiri dari perangkat lunak untuk mengembangkan aplikasi dan algoritma kompresi, kriptografi, steganografi. Spesifikasi perangkat lunak adalah sebagai berikut:

1) Windows 8.1
2) XAMPP v3.2.2
3) NetBeans 8.1

### 3.6. Perancangan Program

Program yang akan dibuat terdiri dari tujuh buah *Form*, yaitu terdiri dari *Form login, Master, Enkripsi, Dekripsi, Form akun, Form Credit, dan Help.*

Untuk dapat menggunakan aplikasi *user* harus daftar terlebih dahulu melalui admin dan untuk melakukan enkripsi *file*, *user* dapat memilih menu enkripsi. Pada menu ini, *user* diharuskan memilih *file* terlebih dahulu, baru melakukan proses enkripsi dan kompresi.

Namun *file* dokumen tidak boleh lebih besar dari ukuran *file* yang telah ditentukan, selanjutnya akan tampil *pop-up* memberi nama pada hasil *file* yang di enkripsi.

Sedangkan untuk mengembalikan *file* yang sudah di enkripsi menjadi *file* asli, *user* juga dapat memilih menu dekripsi. Serta ada menu *help* untuk membantu *user* dalam menggunakan program tersebut. Secara umum, rancangan program yang akan dibuat dapat dilihat pada gambar 1.

Gbr. 1 Arsitektur Kerja Aplikasi

### 3.7. Rancangan Basis Data

Berikut adalah struktur tabel yang terdapat pada aplikasi enkripsi dan dekripsi.

a. Tabel Basis Data
Login

TABEL I
TABEL LOGIN

| Id | Username | Password |
|---|---|---|
| PK | | |

b. Spesifikasi Basis Data
Login

Nama Tabel : admin
Isi : berisi data pengguna
Media : *harddisk*
Primary Key : Id
Foreign Key :

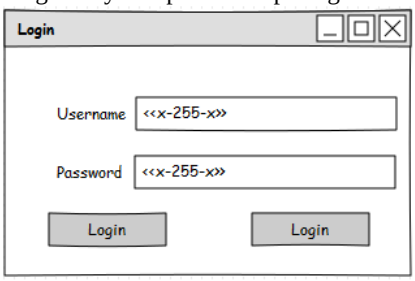
TABEL II
SPESIFIKASI DATA TABEL LOGIN

| No. | Nama Field | Type | Lebar | Keterangan |
|---|---|---|---|---|
| 1 | Id | Varchar | 11 | Kode User |
| 2 | Username | Varchar | 255 | Nama User |
| 3 | Password | Varchar | 255 | Password User |

### 3.8. Rancangan Layar

#### 3.8.1. Rancangan Layar Form Login

Rancangan Layar dapat dilihat pada gambar berikut ini.




Gbr. 2 Rancangan Layar *Form Login*

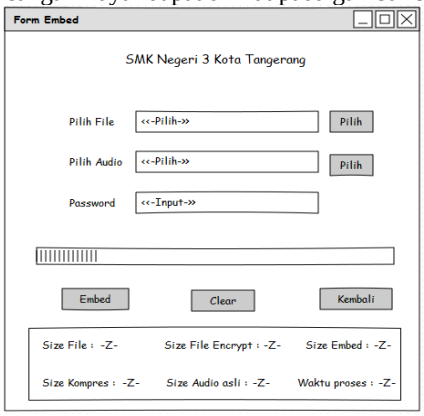
#### 3.8.2. Rancangan Layar Form Menu Utama

Rancangan Layar dapat dilihat pada gambar berikut ini.

Gbr. 3 Rancangan Layar *Form* Menu Utama

#### 3.8.3. Rancangan Layar Form Embed

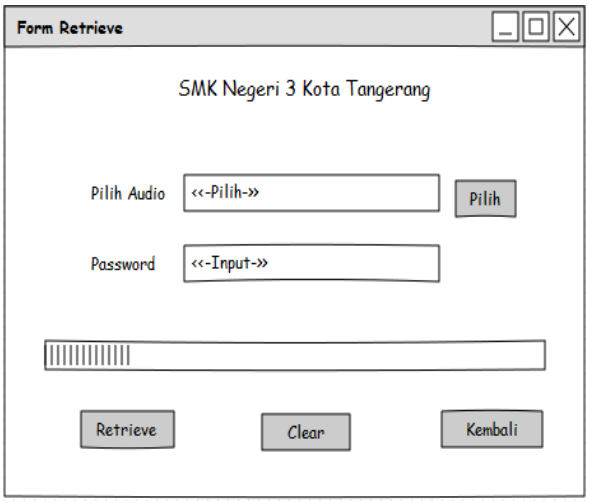
Rancangan Layar dapat dilihat pada gambar berikut ini.




Gbr. 4 Rancangan Layar *Form* Embed

#### 3.8.4. Rancangan Layar Form Retrieve

Rancangan Layar dapat dilihat pada gambar berikut ini.

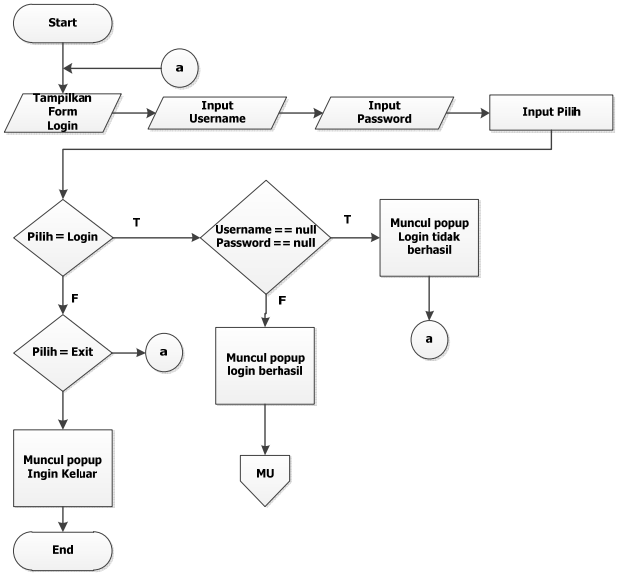



Gbr. 5 Rancangan Layar *Form* Retrieve

## 3.9. Flowchart

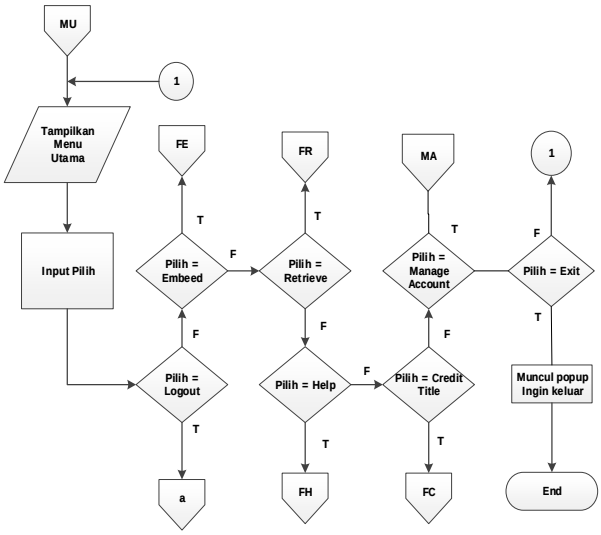
Berikut ini adalah *flowchart* yang digunakan untuk menelusuri proses program pada aplikasi pengamanan data ini:

### 3.9.1. Flowchart Form Login



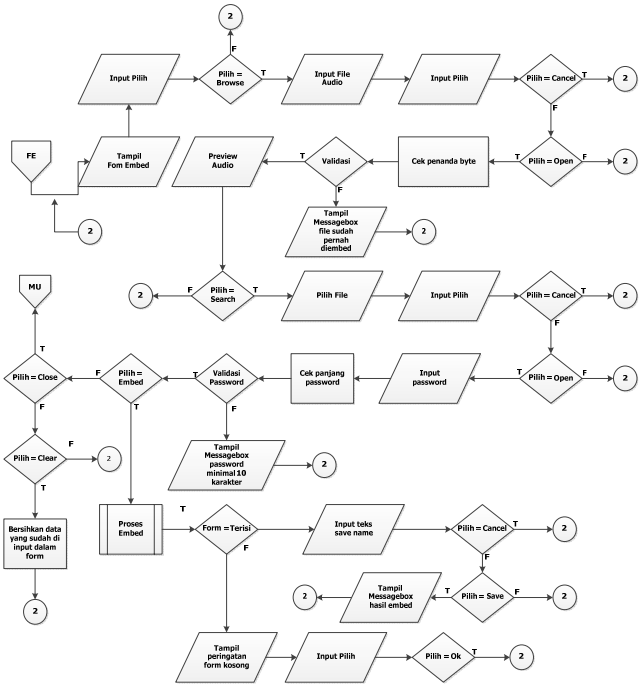

Gbr. 6 Flowchart *Form Login*

### 3.9.2. Flowchart Menu Utama



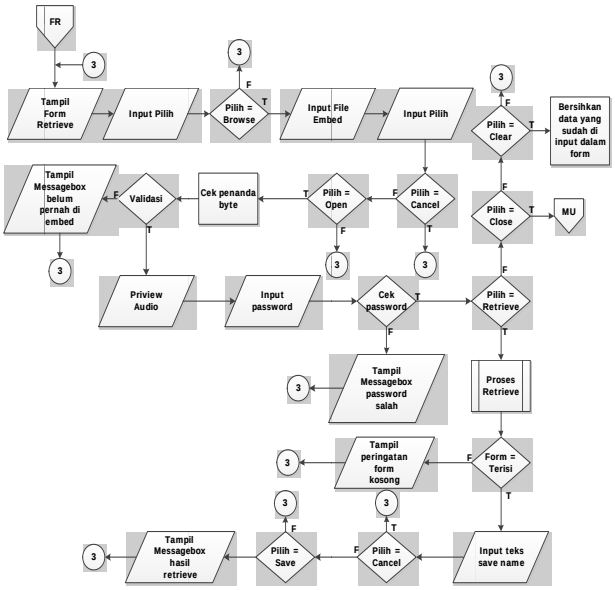

Gbr. 7 *Flowchart Form* Menu Utama

### 3.9.3. Flowchart Form Embed




Gbr. 8 *Flowchart Form* Embed

### 3.9.4. Flowchart Form Retrieve




Gbr. 9 *Flowchart Form* Retrieve

## IV. IMPLEMENTASI DAN UJI COBA PROGRAM

### 4.1. Spesifikasi Perangkat Keras dan Perangkat Lunak

Aplikasi pengamanan data ini dapat berjalan dengan baik apabila kebutuhan *hardware* dan *software* dapat terpenuhi dengan baik agar kinerja aplikasi ini berjalan dengan baik. Berikut adalah spesifikasi perangkata keras (*hardware*) dan perangkat lunak (*software*) yang bisa mendukung aplikasi ini:

**a. Spesifikasi Perangkat Keras**

Berikut ini adalah spesifikasi perangkat keras yang akan digunakan dalam aplikasi ini:

TABEL III
SPESIFIKASI PERANGKAT KERAS

| NO | Perangkat | Kebutuhan |
|---|---|---|
| 1 | CPU | Intel(R) Core (TM) i3-4030U CPU @ 1.90GHz |
| 2 | Hardisk | 500 GB |
| 3 | RAM | 4.00 GB |
| 4 | Monitor | 14” |
| 5 | Keyboard | Internal Keyboard Laptop |

**b. Spesifikasi Perangkat Lunak**

Berikut ini adalah spesifikasi perangkat lunak yang akan digunakan dalam aplikasi ini:

TABEL IV
SPESIFIKASI PERANGKAT LUNAK

| No | Perangkat | Kebutuhan |
|---|---|---|
| 1 | Sistem Operasi | Windows 8.1 64 bit |
| 2 | Tools | Netbeasn IDE 8.1 |
| 3 | Input | *File* (*.doc, *.docx, *.xls, *.xslx, *.pdf) |
| 4 | Ouput | Audio (*.Mp3 dan .Wav) |

### 4.2. Pengujian Aplikasi

Dalam hal pengujian kali ini akan dibahas mengenai perbandingan antara proses embed dan retrieve *file*. *File* yang diuji meliputi jenis *file* yang berformat .doc, docx, xls, .xlsx, .pdf, dan .txt. Pengujiannya yaitu antara lain perbandingan antara ukuran media, ukuran *file*, ukuran audio hasil *embed dan* waktu *embed*.

#### 4.2.1. Tabel Pengujian *Embed*

Di bawah ini adalah hasil pengujian proses embed, pengujiannya antara lain kunci yang digunakan, perbandingan antara ukuran media, ukuran file, ukuran gambar hasil embed dan waktu embed.

TABEL V
HASIL UJI COBA PROSES EMBED APLIKASI

| No | *File* Dokumen | | *File* Audio | | | | Durasi *Embed* (Detik) |
|---|---|---|---|---|---|---|---|
| | Nama *File* | *Size* | Nama *File* | *Format* | *Size* Original | *Size File* Setelah *Embed* | |
| 1 | SISTEM KOMPUTER | 787 KB | One Ok Rock – The Beginning | Mp3 | 4.79 MB | 6.83 MB | 1.31 Detik |
| 2 | SISTEM KOMPUTER | 787 KB | Vista Windows Logon FUll | Wav | 0.86 MB | 2.88 MB | 0.66 Detik |
| 3 | SISTEM KOMPUTER | 787 KB | Payung Teduh – Angin Pujaan Hujan | Mp3 | 3.23 MB | 5.28 MB | 0.66 Detik |

#### 4.2.2. Tabel Pengujian Retrieve

Di bawah ini adalah hasil pengujian proses retrieve, pengujiannya antara lain ukuran *file* setelah *retrieve*, waktu *retrieve*.

TABEL VI
HASIL UJI COBA PROSES RETRIEVE APLIKASI

| NO | *File* Audio | | | *File* Dokumen Asli | | Durasi *Retrieve* (Detik) |
|---|---|---|---|---|---|---|
| | Nama *File* | *Format* | *Size* | Nama *File* | *Size* | |
| 1 | One Ok Rock – The Beginning | Mp3 | 6.83 MB | SISTEM KOMPUTER.docx | 787 KB | 4.1 Detik |
| 2 | Vista Windows Logon FUll | Wav | 2.88 MB | SISTEM KOMPUTER.docx | 787 KB | 0.65 Detik |
| 3 | Payung Teduh – Angin Pujaan Hujan | Mp3 | 5.28 MB | SISTEM KOMPUTER.docx | 787 KB | 0.76 Detik |

## V. PENUTUP

### 5.1. Kesimpulan

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

a. Dengan adanya aplikasi keamanan data ini, proses penyimpanan dan pertukaran *file* soal menjadi lebih aman.
b. Berdasarkan penelitian mengenai implementasi kompresi huffman, RC4 dan steganografi EOF yaitu penggunaan tiga teknik pengamanan data pada kompresi algoritma Huffman, kriptografi algoritma RC4 dan steganografi yang menggunakan metode EOF, dapat digunakan untuk meningkatkan keamanan *file* soal.
c. Tingkat keamanan data soal setelah diembed cukup terjaga, dengan kata lain file tidak berkurang atau mengalami kerusakan setelah proses embed data dilakukan.

### 5.2. 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* dokumen doc, docx, xls, xlsx dan pdf saja, namun bisa juga untuk *file* video, audio maupun *file* gambar.
b. Efisiensi dalam penyembunyian *file* diharapkan dapat lebih ditingkatkan cover penyisippan tidak terbatas hanya pada media audio saja.

## REFERENSI


[1] Wibowo, A. (2012). Kompresi data menggunakan metode huffman, *2012* (Semantik), 47–51.

[2] Cormen;Leiserson;Rivest , ”Introduction to Algorithms”, 1990, The MIT Press, Massachusetts

[3] Talbot dan Welsh. 2006. Karya ilmiah repository.usu.ac.id/bitstream/123456789/34717/4/Chapter II.pdf Diakses pada tanggal 05 April 2016.

[4] Ariyus. 2006. *Kriptografi Keamanan Data Dan Komunikasi*. Yogyakarta. Graha Ilmu. 2006.

[5] Hendrawati. Hamdani. Harsa, A. (2014). KEAMANAN DATA DENGAN MENGGUNAKAN ALGORITMA RIVEST CODE 4 ( RC4 ) DAN STEGANOGRAFI PADA CITRA DIGITAL. *INFORMATIKA Mulawarman*, *9*(1).

[6] Setyaningsih, E. (2013). IMPLEMENTASI SYSTEM SANDI STREAM CIPHER UNTUK PENGAMANAN DATA IMAGE Emy Setyaningsih. *Seminar Nasional Teknologi Informasi Dan Komputasi (Senastik)*, *2013*(Senastik), 84–91.

[7] Sembiring, S., Woods, G. R. E., & Processing, D. I. (2013). MENYISIPKAN PESAN TEKS PADA GAMBAR DENGAN METODE END OF FILE, 45–51.

[8] Munir, R., 2004. Sistem Kriptografi Kunci-Publik Departemen Teknik Informatika Institut Teknologi Bandung.

[9] Alatas, P., 2009. Implementasi teknik steganografi dengan metode lsb pada citra digital. , pp.1–25. Available at: http://www.gunadarma.ac.id/library/articles/graduate/computer-science/2009/Artikel_11104284.pdf.

[10] Anggraini, Yayuk. Shaka., 2014. PENERAPAN STEGANOGRAFI METODE END OF FILE ( EOF ) DAN ENKRIPSI METODE DATA ENCRYPTION STANDARD ( DES ) PADA APLIKASI PENGAMANAN DATA GAMBAR BERBASIS JAVA PROGRAMMING. , STMIK Dipanegara Makasar

[11] Lubis, A. R., Lidya, M. S., Budiman, M. A., & Utara, U. S. (2012). Perancangan Perangkat Lunak Steganografi Audio MP3 Menggunakan Metode Least Significant Bit ( LSB ) Dengan Visual Basic 6 . 0, *1*(1), 63–68.