p-ISSN : 2443-2210
e-ISSN : 2443-2229

Implementasi Algoritma *Caesar Cipher* Dan Steganografi *Least Significant Bit* Untuk *File* Dokumen

http://dx.doi.org/10.28932/jutisi.v6i3.2817

Irvan Maulana Yusup ✉ #1, Carudin*2, Intan Purnamasari#3
#*Jurusan Teknik Informatika, Universitas Singaperbangsa Karawang, Telukjambe Timur, Karawang*
[1]Irvan.16118@student.unsika.ac.id
[2]carudin@staff.unsika.ac.id
[3]Intan92@staff.unsika.ac.id

Abstract — **Security and confidentiality of a file are important aspects because the owner of the file does not want the data to be known by irresponsible parties. To keep the file secret and secure there are techniques called cryptographic algorithms and steganographic algorithms. Cryptographic algorithms are a way to change the contents of the file to be incomprehensible, while steganographic algorithms are a way of inserting files that you want to keep secret with other file types such as images, sounds, or videos. One type of cryptography is Caesar Cipher and steganography is Least Significant Bit (LSB). Caesar Cipher is a way of securing and keeping the contents of a file secret by shifting letters, while the Least Significant Bit (LSB) is a method of insertion by replacing the rightmost or backmost bits. This research uses waterfall software development with the stages consist of needs analysis, system design, implementation, and testing. The program code is written in Java language and uses the Netbeans 8.2 application. The result of the research is that with 10 research materials, 5 document files (* .doc) and 5 image files (* .png), only 2 files of each research material can be processed by this software. The tests carried out included testing the functions and steganographic criteria such as Fidelity, Recoverable, and Robustness.**

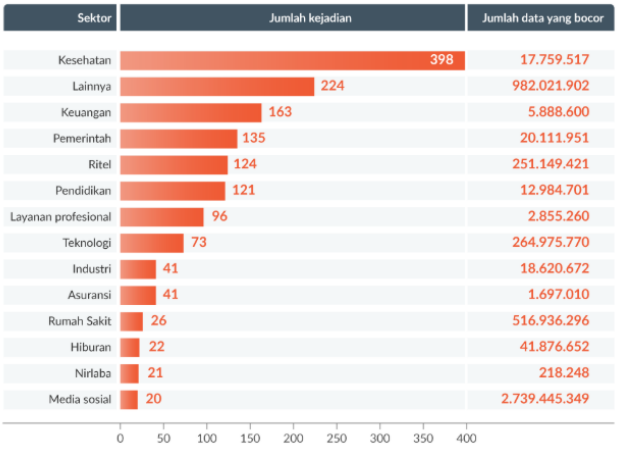
Keywords — ***Algorithm, Caesar Cipher; Least Significant Bit* (LSB); Netbeans;**

I. PENDAHULUAN

Dalam perkembangan teknologi dan komunikasi di jaman ini begitu pesat, salah satu manfaat yang terasa adalah saling bertukar data maupun informasi begitu cepat [1].

Data atau informasi ada yang bersifat umum, yang artinya bisa dilihat atau diakses oleh banyak orang. Ada juga informasi yang bersifat pribadi atau rahasia, yang artinya tidak boleh dilihat oleh banyak orang, hanya orang tertentu yang bisa mengaksesnya. Maka kerahasiaan dan keamanan ini menjadi faktor penting untuk menjaga data dalam pertukaran data [2]. Jika faktor keamanan kerahasiaan tidak diperhatikan makan akan terjadi suatu pencurian data oleh pihak yang tidak bertanggung jawab seperti halnya kasus tahun 2018.

Suatu kasus pada tahun 2018 peretasan data pribadi rawan terjadi, sekitar 234,49 juta data pribadi diretas oleh pihak luar maupun dari pihak dalam. Sumber kebocoran data kemungkinan akibat sistem tidak aman, peretasan dari luar, peretasan dari dalam, dan beragam sumber yang diketahui, bisa dilihat pada gambar 1 berbagai sektor dan jumlah kebocoran data setiap sektor [3].



Gambar 1 Kebocoran data dari berbagai sektor

Maka dari itu dibutuhkan ilmu untuk menjaga kerahasiaan data dan informasi ini. Dalam penjagaan data pribadi ada ilmu yang mempelajari cara-cara pengamanan data untuk menjaga keamanan data tersebut, ilmu tersebut biasa dikenal dengan kriptografi. Kriptografi (*cryptography*) adalah penggabungan kata "*Crypto*" yang berarti rahasia dan "*graphy*" yang berarti tulisan, bisa disimpulkan bahwa kriptografi adalah seni dan seni untuk menjaga keamanan data dari pihak ketiga, yang hanya bisa diketahui oleh pihak pengirim dan penerima [4]. Dalam proses kriptografi ada 2

proses yaitu proses enkripsi dan dekripsi. Enkripsi adalah proses diubahnya isi pesan yang dimengerti menjadi tidak bisa dimengerti lagi, karena isi sudah diubah. Sedangkan proses Dekripsi adalah proses di mana pesan yang sudah diubah dikembalikan keasliannya, agar dapat dimengerti kembali seperti sedia kala.

Caesar Cipher adalah salah satu dari banyaknya algoritma kriptografi yang ada, cara kerja *Caesar Cipher* ini yaitu dengan cara menggeser semua karakter dengan nilai karakter yang sama [5]. Dalam penelitian Yuningrat Dwi Putri dan kawan-kawan dengan judul Penerapan Kriptografi *Caesar Cipher* Pada Fitur *Chatting* Sistem Informasi *Freelance* menyatakan bahwa algoritma *Caesar Cipher* adalah suatu teknik enkripsi yang paling banyak digunakan dalam pengamanan data, dan penelitian ini juga, berhasil menerapkan algoritma *Caesar Cipher* pada sistem informasi *freelance* dengan menggunakan bahasa pemrograman PHP. Pada enkripsi dan dekripsi memakai metode *end to end*, pengirim dan penerima pesan tidak perlu menggunakan kunci.

Ada juga yang dikenal dengan steganografi (*Steganography*), steganografi ini sedikit berbeda dengan kriptografi, jika kriptografi adalah mengubah isi pesan menjadi tidak dimengerti, maka steganografi adalah seni dan juga ilmu di mana isi pesan akan disembunyikan [6]. Biasanya isi pesan akan disembunyikan dengan cara menyisipkan media lain seperti gambar.

Algoritma *Least Significant Bit* (LSB) dalam penyisipannya atau menyembunyikan pesannya dengan citra digital yaitu melakukan penggantian bit akhir [7]. Dalam penelitian Salkin Lutfi dan Rosihan dalam penelitiannya yang berjudul Perbandingan Metode Steganografi LSB (*Least Significant Bit*) Dan MSB (*Most Significant Bit*) Untuk Menyembunyikan Informasi Rahasia Ke dalam Citra Digital menyatakan bahwa metode steganografi *Least Significant Bit* (LSB) adalah metode yang populer dalam penyisipan citra digital. Hasil dari analisis perbandingan LSB (*Least Significant Bit*) Dan MSB (*Most Significant Bit*) bahwa LSB lebih baik dari pada MSB, karena *pixel* dari citra yang dihasilkan oleh LSB tidak mengubah warna secara drastis dibandingkan MSB.

Berdasarkan penelitian sebelumnya dalam pengamanan data pribadi atau rahasia bisa menggunakan teknik enkripsi dan steganografi, algoritma *Caesar Cipher* dan *Least Significant Bit* (LSB) adalah metode yang paling banyak digunakan, maka peneliti memutuskan membuat judul penelitian "Implementasi Algoritma *Caesar Cipher* dan Steganografi *Least Significant Bit* (LSB) Untuk *File* Dokumen (.DOC)", di mana dalam penelitian ini akan membahas proses enkripsi dan dekripsi pada suatu *file* dokumen pribadi atau rahasia menggunakan algoritma *Caesar Cipher* dan *Least Significant Bit* (LSB).

II. TINJAUAN PUSTAKA

A. Algoritma

Algoritma merupakan dasar pengetahuan untuk bisa cabang ilmu komputer. Algoritma yakni suatu urutan langkah logis dalam mengatasi suatu masalah, dan penyelesaian masalah itu disusun dengan logis dan sistematis [8]. Walaupun algoritma sering dikenal di ilmu komputer, tapi algoritma juga bisa ditemukan di kehidupan sehari-hari, contoh sederhananya seperti membuat makanan, dengan menyediakan alat masak, bahan makanan, yang dilakukan secara sistematis.

B. Kriptografi

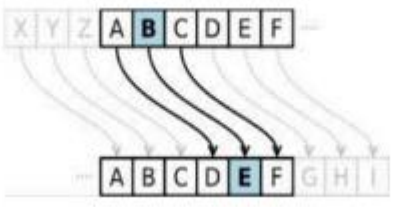
Kriptografi (*cryptography*) berasal dari kata Yunani di mana kata "*cryptos*" yang artinya (*secret*) yakni rahasia, jika kata "*graphein*" artinya (*writing*) yakni tulisan [9]. Ada beberapa definisi kriptografi dari para ahli, yaitu:

a. Menurut Meyer : Kriptografi merupakan ilmu dan seni dalam menjaga suatu kerahasiaan pesan, cara penyandiannya ke dalam bentuk yang tidak bisa dipahami lagi maknanya.
b. Menurut Schneier : Kriptografi adalah ilmu dan seni dengan tujuan adalah mengamankan pesan.
c. Menurut Menez : Kriptografi yakni ilmu yang mempelajari banyak teknik matematika yang berkaitan dengan aspek keamanan informasi contohnya kerahasiaan, integritas data, serta autentikasi.

Bisa disimpulkan dari beberapa definisi di atas tentang kriptografi, Kriptografi adalah penggabungan ilmu dan seni dengan cara unik atau tersendiri dalam mengamankan pesan pribadi atau bersifat rahasia.

C. Caesar Cipher

Caesar Cipher adalah salah satu metode kriptografi klasik, alasan metode ini dinamakan *Caesar Cipher* adalah saat Julius Caesar berkomunikasi dengan para panglimanya [4]. Metode ini jenis *cipher* substitusi, dalam setiap huruf plainteksnya digantikan huruf lain. Contohnya pergeseran huruf adalah 3, maka huruf A menjadi huruf D, huruf B menjadi huruf E, dan seterusnya. Untuk memperjelas gambarannya, bisa lihat gambar 2 di bawah ini.



Gambar 2 Proses pergeseran 3 huruf

Sebagai contoh lain ada sebuah kalimat yang menggunakan metode *Caesar Cipher* di mana kalimat adalah "awasi asterix dan temannya obelix" (abaikan kutip 2) makan setelah dienkripsi menjadi "DZDVL DVWHULA GDQ WHPDQQBA REHOLA", Pada Tabel I adalah

gambaran dari karakter asli atau *Plainteks* diubah menjadi karakter rahasia atau *Cipherteks* dengan cara menggeser karakter tersebut, pada Tabel I pergeseran *plainteks* sebanyak 3 karakter.

TABEL I
TABEL SUBSTITUSI MENGGESER 3 HURUF

Plainteks	abcdefghijklmnopqrstuvwxyz
Cipherteks	defghijklmnopqrstuvwxyzabc

Caesar Cipher dapat diformulasikan dengan fungsi matematika, kodekan dengan angka di mana A=0, B=1, C=2, ..., Z=25. Proses enkripsi (E) tentukan pergeserannya adalah 3 dengan melakukan penjumlahan plainteks (P) dengan 3 dalam modulus 26.

$$C = E(P) = (P + 3)\ mod\ 26 \quad \textbf{(1)}$$

Kemudian rumus dekripsinya (D) dengan rumus kebalikannya, yaitu mengurangi cipherteksnya (C) dengan 3 *modulus* 26.

$$P = D(P) = (C - 3)\ mod\ 26 \quad \textbf{(2)}$$

Maka dalam perhitungan untuk proses enkripsi bisa dihitung sebagai berikut:

P1=A=0 → C1=E(0)=(0+3) mod 26 =3=D
P2=W=22 → C2=E(22)=(22+3) mod 26 =25=Z
P3=A=0 → C3=E(0)=(0+3) mod 26 =3=D
P4=S=18 → C4=E(18)=(18+3) mod 26 =21=V
P5=I=8 → C5=E(8)=(8+3) mod 26 =11=L
Dan seterusnya

Setelah dicari semuanya maka dapat diperoleh cipherteksnya adalah "DZDVL DVWHULA GDQ WHPDQQBA REHOLA". Chiperteks ini akan dikembalikan atau didekripsi ke *plainteks* dengan perhitungan berikut:

C1=D=0 →P1=D(3)=(3-3) mod 26 =0=A
C2=Z=22 →P2=D(22)=(22-3) mod 26 =22=W
C3=D=0 →P3=D(3)=(3-3) mod 26 =0=A
C4=V=18 →P4=D(21)=(21-3) mod 26 =18=S
C5=L=8 →P5=D(11)=(11-3) mod 26 =8=I

Dan seterusnya. Untuk C12=A=0 P12=D(0)=(0-3) mod 26 =-3 mod 26=X, -3 mod 26 dihitung dengan cara |-3| mod 26=3, sehingga -3 mod 26=26-3=23 [9].

D. Steganografi

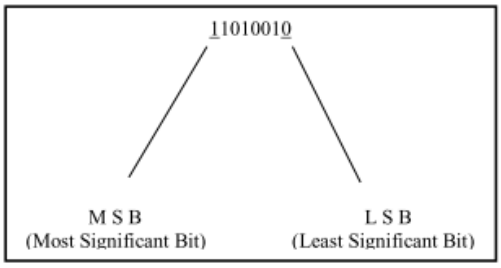
Steganografi katanya adalah asal dari bahasa Yunani, di mana "*steganos"* yang memilik makna tersembunyi dan "*graphien"* yang memiliki makna tulisan, maka steganografi adalah tulisan yang tersembunyi. Steganografi bisa dilihat sebagai pelengkap dari kriptografi, dan dalam pengeksekusiannya mengenkripsi dahulu *file* dokumen baru disembunyikan dengan steganografi [9].

Kelebihannya adalah tidak akan menarik perhatian pihak ketiga sehingga tidak menimbulkan rasa curiga, berbeda dengan kriptografi yang hasil enkripsinya menimbulkan kecurigaan oleh pihak ketiga.

E. Least Significant Bit (LSB)

Metode *Least Significant Bit* (LSB) atau biasa disebut dengan LSB suatu metode modifikasi steganografi suatu melakukan perubahan pada bit yang paling kanan atau bit yang kurang berarti. Media penampung LSB ini biasanya menggunakan citra digital atau gambar [9].

Dalam setiap *byte* (1 *btye* = 8 bit), susunan bit dalam setiap bit yakni $b_7b_6b_5b_4b_3b_2b_1b_0$, bit b_0 adalah bit yang kurang berarti atau *Least Significant Bit* (LSB) sedangkan bit b_7 adalah bit yang sangat berarti atau *Most Significant Bit* (MSB). Padan gambar 3 dibawah ini adalah gambaran dalam perbedaan antara LSB dan MSB.



Gambar 3 Susunan bit

F. File

File merupakan kumpulan data dan informasi yang saling berkaitan satu sama lain yang tersimpan di penyimpanan komputer, konsep pada *file* ada beberapa jenis seperti *numeric, character, binary* dan lainnya [8]. Setiap ekstensi *file* memiliki kegunaan tersendiri seperti di bawah ini:

a. *File* systen ada beberapa ekstensi seperti (*.sys), (*.com), (*.bat), (.*exe) dan masih banyak lagi. Kegunaan jenis *file* ini adalah menjalankan program komputer dan menjalankan program aplikasi.
b. *File* video, ekstensi jenis *file* video seperti (*.mp4), (*.avi), (*.flv) dan lainnya. Ekstensi *file* ini bisa menggambarkan bahwa jenis *file* video yang memiliki pemutaran yang berbeda.
c. *File* dokumen, jenis *file* dokumen seperti (*.pdf), (*.docx), (*.txt) dan sebagainya. Setiap jenis ekstensi memiliki aplikasi tersendiri untuk bisa diakses.
d. *File* suara, beberapa ekstensi *file* suara adalah (*.wav), (*.mp3), (*.rm) dan lain-lain. Seperti hal *file* lainnya bahwa jenis ekstensi ini juga mempunyai aplikasi tersendiri untuk bisa diaksesnya.

Dan masih banyak jenis ekstensi lainnya sesuai dengan perannya masing-masing.

G. Citra Digital

Citra merupakan suatu representasi bentuk dua dimensi menjadi bentuk nyata tiga dimensi, perwujudan citra mempunyai banyak macam dari gambar hitam putih sampai dengan gambar bergerak yang ada di televisi [10]. Citra mempunyai 2 macam yakni citra digital yaitu citra yang dihasilkan oleh digitalisasi contohnya kamera digital, kemudian ada citra *continue* yakni citra yang proses hasilnya dari optik yang dikirim oleh sinyal analog,

contohnya kamera analog [8]. Berikut ini adalah beberapa jenis format citra yang sering dijumpai:

a. *Portable Network Graphics* (PNG) ekstensi (*.png) yakni format penyimpanan untuk citra terkompresi, format ini bisa digunakan dalam grayscale, palet warna hingga *full color* [11].
b. *Joint Photographic Experts Group* (JPEG) dengan ekstensi (.jpeg) yakni teknik kompresi yang menurunkan hasil kualitas citra atau gambar turun atau lossy compression. Format JPEG ini kurang baik digunakan citra atau gambar artistik [8].
c. *Graphics Interchange Format* (GIF) dengan ekstensi (*.gif) biasa dipakai untuk warna 8 bit, biasa dijumpai pada aplikasi web [11].

H. Pengujian Kualitas Media Citra Hasil Steganografi

Pengujian kualitas media hasil steganografi merupakan pengujian yang dicocokkan dengan kriteria-kriteria yang dicermati dalam penyembunyian *file* dokumen dengan metode steganografi [8]. Kriteria-kriteria tersebut diantaranya:

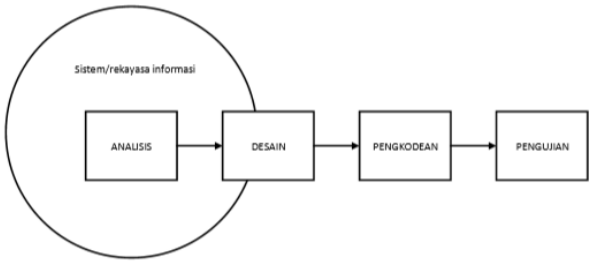
a. *Fidelity*, pengujian ini merupakan pengujian pada aspek mutu media hasil steganografi, biasanya metode pengujian ini menggunakan nilai *Mean Square Error* (MSE) dan *Peak Signal to Noise Rasio* (PSNR).
b. *Robustness*, *file* yang disembunyikan bisa bertahan terhadap dalam media tempat penampung *file* dokumennya di manipulasi.
c. *Recovery*, *file* yang telah disembunyikan ke dalam media harus bisa dikembalikan seperti sedia kala.

III. METODE PENELITIAN

Siklus Hidup Perangkat Lunak atau dikenal dengan istilah *Software Development Life Cycle* (SDLC) yakni merupakan salah satu metodologi proses tahapan dalam mengembangkan perangkat lunak, dengan tujuan utama metodologi ini adalah menciptakan produk berkualitas tinggi [12] [13] [14].

Model air terjun atau dikenal dengan model *waterfall*, model *waterfall* ini menyediakan tahapan perangkat lunak berurutan di mana tahapannya adalah analisa, desain, pengkodean, pengujian. Model ini sesuai dengan pengembangan perangkat lunak di mana rencana kebutuhan tidak banyak berubah [8].

Dalam penelitian ini metodologi untuk menerapkan algoritma yang dipakai adalah *Software Development Life Cycle* (SDLC) dengan model *waterfall*, pada gambar 4 merupakan gambaran tahapan-tahapan yang dilakukan.



Gambar 4 Tahapan *waterfall*

A. Analisa kebutuhan

Dalam tahapan ini peneliti membaca referensi dari buku, jurnal, skripsi, berita, sumber referensi lainnya. Kemudian mengidentifikasikan kebutuhan dari referensi yang telah dibaca, maka peneliti mengidentifikasikan kebutuhan keseluruhan sebagai berikut:

a. Analisa kebutuhan fungsional
b. Analisa kebutuhan non-fungsional
c. Analisa kebutuhan pengguna (*user*)
d. Analisis algoritma kriptografi *Caesar Cipher*
e. Analisis steganografi *Least Significant Bit* (LSB)

B. Desain Sistem

Dalam tahapan ini peneliti melakukan desain perancangan perangkat lunak yang akan dibuat, ada beberapa perancangan yang akan dilakukan antara lain:

a. Perancangan diagram sistem, perancangan diagram ini akan menggunakan *Unified Modeling Language* (UML), perancangan ditujukan untuk interaksi antara pengguna dengan sistem yang akan dibuat. Diagram yang akan digunakan adalah *use case* diagram dan *activity* diagram dan *sequence* diagram.
b. Perancangan *flowchart*, perancangan menyampaikan gambaran alur proses dari awal sampai akhir perangkat lunak berjalan.
c. Perancangan tampilan antarmuka, perancangan membuat desain tampilan dari perangkat lunak yang dibuat.

C. Implementasi

Pada tahapan ini melakukan penerjemahan dari desain sistem ke kode-kode program untuk dimasukkan ke aplikasi pembuat perangkat lunak, perangkat lunak yang dipilih untuk perancang perangkat lunak adalah *Nerbeans* IDE 8.2 dengan bahasa pemrograman *java*.

D. Pengujian Program

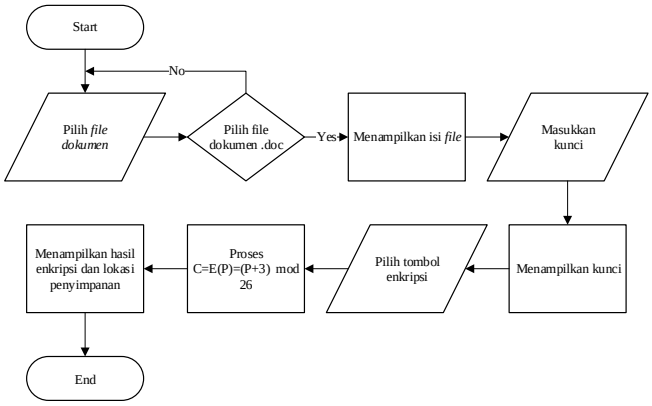
Pada tahap akhir ini peneliti melakukan pengujian pada perangkat lunak. Pengujian ini terbagi 2, yaitu pengujian terhadap enkripsi dan dekripsi pada algoritma kriptografi *Caesar Cipher*, dan pengujian terhadap algoritma steganografi *Least Significant Bit* (LSB) dengan kriteria *fidelity, robustness dan recovery.*

p-ISSN : 2443-2210
e-ISSN : 2443-2229

IV. HASIL DAN PEMBAHASAN

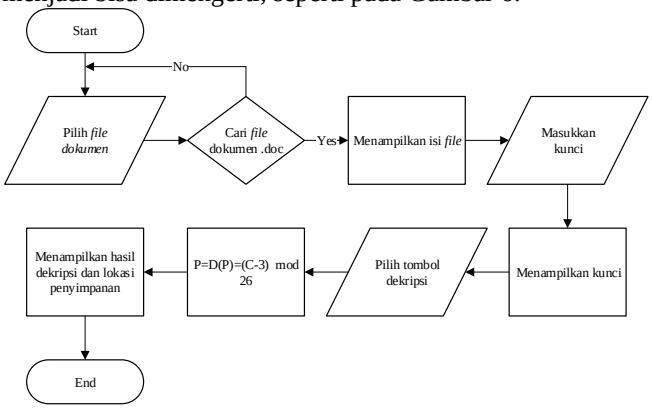
Dari hasil penelitian yang dilakukan bisa lihat gambaran untuk enkripsi, dekripsi, *embedding* atau penyisipan, dan *extraction* atau ekstrak

Proses dari enkripsi ini dengan cara menggeser huruf, di mana jumlah penggeseran tergantung dari kunci yang dimasukkan oleh pengguna, jika pengguna memasukkan kunci 8 maka akan melakukan pergeseran karakter sebanyak 8, jika kuncinya 1 maka bergeser 1 dan seterusnya. Pengguna hanya perlu memasukkan *file*, kunci dan menekan tombol fungsi dan biarkan sistem yang bekerja seperti pada Gambar 5.



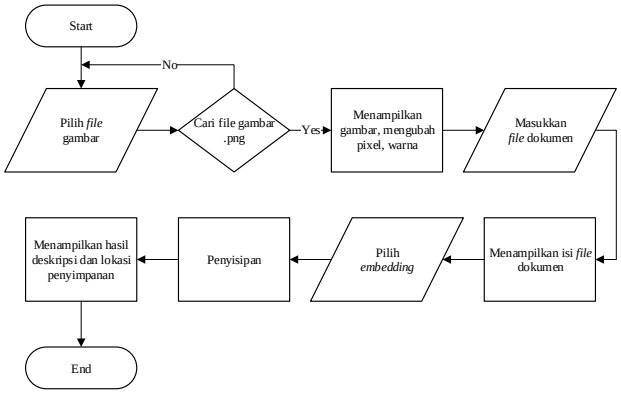
Gambar 5 *Flowchart* Enkripsi

Cara kerja dekripsi pun sama dengan enkripsi hanya saja tujuannya adalah untuk mengembalikan isi *file* tersebut menjadi bisa dimengerti, seperti pada Gambar 6.



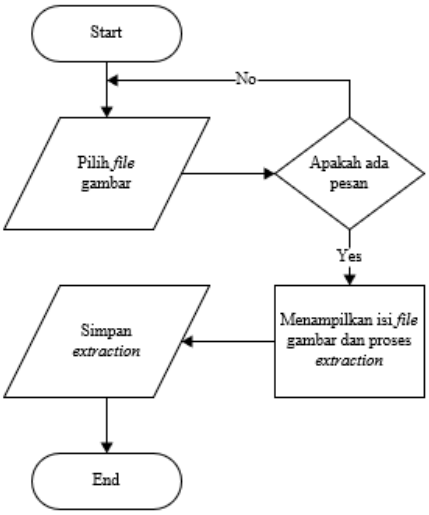
Gambar 6 *Flowchart* Dekripsi

Gambar 7 adalah gambaran bagaimana proses perangkat lunak ini untuk menyisipkan *file* dokumen ke dalam *file* gambar.



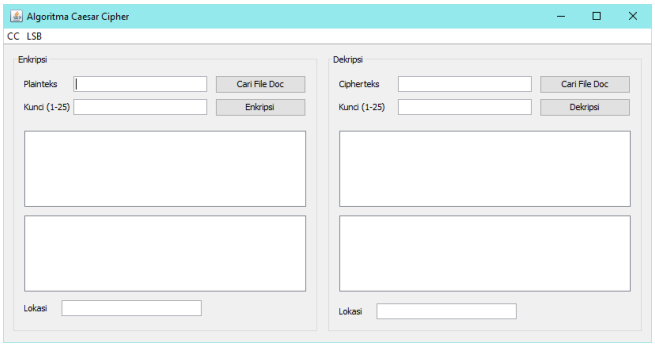
Gambar 7 *Flowchart Embedding*

Setelah proses *embedding* maka untuk mengeluarkan *file* dokumen di dalam *file* gambar, pada gambar 8 ditunjukkan peran *extraction* yang mengembalikan *file* dokumen ini.



Gambar 8 *Flowchart Extraction*

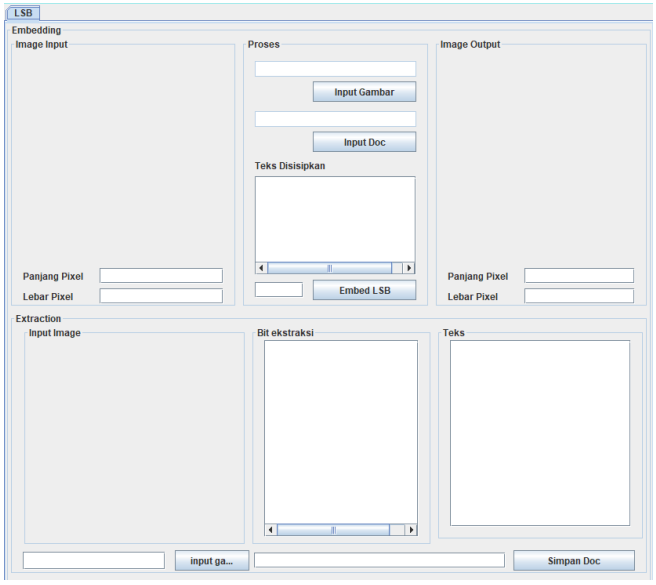
Setelah perancangan dan desain sistem telah dilaksanakan maka mengimplementasikannya. Gambar 9 adalah tampilan dari menu enkripsi-dekripsi setelah menerapkan tampilan dari rancangan *user interface*.



Gambar 9 Tampilan Enkripsi-Dekripsi

JuTISI
Jurnal Teknik Informatika dan Sistem Informasi

Gambar 10 adalah tampilah dari menu *embedding-extraction* setelah melakukan pengkodean dalam aplikasi *netbeans* 8.2 dan menerapkan rancangan antarmuka atau *user interface.*



Gambar 10 Tampilan *Embedding-Extraction*

Pengujian Fungsi Enkripsi-Dekripsi fungsi ini untuk mencari tahu apakah fungsi dari enkripsi-dekripsi ini berjalan dengan baik atau ada suatu masalah. Berikut ini pada Tabel II adalah hasil pengujian menggunakan bahan penelitian *file* dokumen.

TABEL II
HASIL ENKRIPSI

No.	Nama *file* dan ekstesni	Ukuran (KiloByte)	Keterangan
1.	cipherCaesar1.doc	2	Berhasil
2.	cipherCaesar2.doc	2	Berhasil
3.	cipherCaesar3.doc	3	Berhasil
4.	cipherCaesar4.doc	3	Berhasil
5.	cipherCaesar5.doc	3	Berhasil

Hasil pengujian fungsi enkripsi semua *file* dokumen bisa di enkripsi sesuai mestinya.

TABEL III
HASIL DEKRIPSI

No.	Nama *file* dan ekstesni	Ukuran (KiloByte)	Keterangan
1.	1cipherCaesar.doc	2	Berhasil
2.	2cipherCaesar.doc	2	Berhasil
3.	3cipherCaesar.doc	2	Berhasil
4.	4cipherCaesar.doc	2	Berhasil
5.	5cipherCaesar.doc	2	Berhasil

Yang terlihat pada Tabel III, bahwa pengujian fungsi enkripsi-dekripsi semua bahan uji *file* dokumen berhasil menjalankan prosesnya seperti yang diharapkan.

Pengujian fungsi dari *embedding-extraction* tidak luput untuk melakukan pengujian fungsi, karena tujuannya apakah fungsi berjalan dengan baik atau ada suatu masalah. Berikut ini adalah hasil pengujian menggunakan bahan penelitian *file* gambar:

TABEL IV
HASIL *EMBEDDING*

No.	Nama *file*	Ukuran (KB)	Nama *file*	Ukuran (KB)	Ket.
1.	Dok-1.doc	29	Gam1.png	153	Berhasil
2.	Dok-2.doc	29	Gam2.png	145	Berhasil
3.	Dok-3.doc	34	Gam3.png	120	Gagal
4.	Dok-4.doc	32	Gam4.png	117	Gagal
5.	Dok-5.doc	32	Gam5.png	143	Gagal

Pada Tabel IV pengujian fungsi *embedding* bahwa *file* dokumen pada nomor 3 sampai dengan nomor 4 dengan *file* dokumen nomor 3 sampai dengan 5, dalam perangkat lunak ini belum bisa di *embedding.*

TABEL V
HASIL *EXTRACTION*

No.	Nama *file*	Ukuran (KB)	Nama *file*	Ukuran (KB)	Ket.
1.	1hasil embed .doc	2	1hasil embed .png	50	Berhasil
2.	2hasil embed .doc	2	2hasil embed .png	43	Berhasil

Dari Tabel V bisa disimpulkan pada fungsi *embedding-extraction* mengalami kegagalan dalam proses *embedding*, dikarenakan perangkat lunak ini hanya bisa menampung gambar dengan *pixel* 250x250 dan perubahan biner 8 bit, maka jumlah karakter yang bisa ditampung maksimal adalah 31, dalam hitungan kasarnya 31*8=248.

Pengujian Kriteria Steganografi, pengujian ini adalah untuk membandingkan atau mengkomparasi *file* sebelum di sisipi dan sesudah di sisipi, dan untuk mengetahui apakah perangkat lunak bisa lolos uji *fidelity*, *robustness* dan *recovery.*

Pengujian *Fidelity* adalah untuk mengetahui apakah *file* dari hasil dari perangkat lunak yang dibuat mengalami banyak perubahan atau tidak, pada Tabel VI ditunjukkan detail pengujian *Fidelity.*

TABEL VI
PENGUJIAN *FIDELITY*

No.	Kondisi awal		Hasil *Embedding*	
	Nama *File* dan Ukuran	**Nama Gambar dan Ukuran**	**Nama *File Embedding* dan Ekstensi**	**Uukuran (KB)**
1	Dok-1.doc (29 KB)	Gam1.png (153 KB)	1hasilembed.png	50
2	Dok-2.doc (29 KB)	Gam2.png (145 KB)	2hasilembed.png	43

Dalam Tabel VI bisa di artikan bahwa setelah melakukan pengujian dan komparasi bahwa hasil *embedding* mengalami banyak perubahan, seperti kondisi awal gambar berwarna dan setelah di *embedding* berubah menjadi keabuan atau tanpa warna. Yang berarti perangkat lunak ini tidak lolos dalam kriteria *fidelity*.

Pengujian *Robustness* adalah apakah hasil *embedding* dari perangkat lunak ini bisa *extraction*, jika *file* hasil *embedding* ini mengalami perubahan, seperti mengubah kontras, rotasi dan menambah objek, berikut pada Tabel VII hasil pengujian *Robustness.*

TABEL VII
PENGUJIAN *ROBUSTNESS*

No.	Nama file dan ekstesni	Jenis perubahan	Keterangan
1.	1hasilembed.png	Kontras	Gagal
2.	2hasilembed.png	Kontras	Gagal
3.	1hasilembed.png	Rotasi	Gagal
4.	2hasilembed.png	Rotasi	Gagal
5.	1hasilembed.png	Objek	Gagal
6.	2hasilembed.png	Objek	Gagal

Hasil dari pengujian ini bahwa hasil *embedding* yang telah mengalami perubahan masih bisa di *extraction,* tetapi hasil dari *extraction* menjadi *file* yang tidak bisa dipahami lagi.

Pengujian *Recovery* adalah menguji apakah *file* hasil *embedding* bisa di *extraction* atau dikembalikan. Untuk hasilnya bisa lihat di Tabel VIII.

TABEL VIII
PENGUJIAN *RECOVERY*

No.	Nama file dan ekstesni	Ukuran (KB)	Keterangan
1.	1hasilembed.png	50	Berhasil
2.	2hasilembed.png	43	Berhasil

Hasil pengujian *recovery* dalam fungsi *embedding-extraction*, bahwa *file* hasil *embedding* bisa di *extraction*. *File* dokumen yang telah disisipkan pun bisa kembali seperti semula walaupun format dari isi *file* tersebut menjadi tanpa format.

V. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian dalam pembangunan perangkat lunak untuk mengamankan dan merahasiakan dokumen rahasia bisa disimpulkan sebagai berikut:

File yang bersifat rahasia atau penting yang hanya ingin diketahui oleh pihak tertentu, bisa di amankan oleh perangkat lunak dengan bahasa pemrograman *java* yang menerapkan algoritma *Caesar Cipher* dan algoritma *Leas Significant Bit* (LSB). *File* dokumen yang dienkripsi oleh algoritma *Caesar Cipher* dilakukan dengan menggeser huruf yang ada dengan jumlah geser tergantung kunci yang dimasukkan. *File* yang di sisipkan dengan gambar menggunakan algoritma *Least Significant Bit* (LSB) dilakukan dengan mengubah bit paling kanan atau paling belakang pada *file* penampung dengan bit *file* dokumen. Dalam proses enkripsi bahwa semua bahan penelitian *file* dokumen bisa di enkripsi, tetapi tidak dengan proses *embedding* hanya 2 *file* dokumen dari total 5 *file* dokumen yang bisa di *embedding*. Dalam *file* dokumen yang akan disisipkan hanya bisa menyisipkan dengan jumlah maksimal 31 karakter. Karena *file* penampung dalam perangkat lunak ini hanya mempunyai ukuran pixel 250x250.

File dokumen yang di enkripsi dari total 5 *file* dokumen semuanya bisa di dekripsi dengan baik. Dan *file* yang bisa di *embedding* oleh perangkat lunak ini hanya 2 *file*, maka saat di *extraction* 2 *file* ini bisa dikembalikan.

Ada beberapa saran yang ingin peneliti sampaikan, seperti diharapkan penelitian selanjutnya pada menu enkripsi-dekripsi bisa menggunakan semua karakter dan simbol, seperti angka, koma, titik, dan lain sebagainya. Pada penelitian selanjutnya pada menu *embedding-extraction* bisa menampung lebih banyak pixel. Pada menu *embedding-extraction* bisa menampung *file* selain gambar, seperti audio, video dan lainnya, bisa pula diterapkan pada *mobile programming*.

DAFTAR PUSTAKA

[1] I. A. Susanto dan A. Solichin, "Enkripsi Data Penggajian Dengan Algoritma Caesar Cipher Dan Vigenere Cipher Pada Pt . Kemasindo Cepat Nusantara," *SKANIKA*, vol. 1, no. 1, pp. 399-404, 2018.

[2] N. Azis, "Perancangan aplikasi enkripsi dekripsi menggunakan metode caesar chiper dan operasi xor," *Ikraith-Informatika,* vol. 2, no. 1, pp. 72-80, 2018.

[3] (2019) Beritagar.id. [Online]. Tersedia: https://beritagar.id/artikel/berita/maraknya-kebocoran-data-akun-jual-beli.

[4] Y. D. Putri, Rosihan dan S. Lutfi, "Penerapan Kriptografi Caesar Cipher Pada Fitur Chatting Sistem Informasi Freelance," *Jurnal Informatika dan Ilmu Komputer (JIKO),* vol. 2, no. 2, pp. 87-94, 2019.

[5] I. Gunawan, "Kombinasi Algoritma Caesar Cipher dan Algoritma RSA untuk pengamanan File Dokumen dan Pesan Teks," *InfoTekJar (Jurnal Nasional Informatika dan Teknologi Jaringan),* vol. 2, no. 2, pp. 124-129, 2018.

[6] A. Hafiz, "Steganografi Berbasis Citra Digital Untuk Menyembunyikan Data Menggunakan Metode Least Significant Bit (Lsb)," *Jurnal Cendikia*, vol. 17, pp. 194-198, 2019.

[7] S. Lutfi dan Rosihan, "Perbandingan Metode Steganografi LSB (Least Significant Bit) Dan MSB (Most Significant Bit) Untuk

Menyembunyikan Informasi Rahasia Kedalam Citra Digital," *JIKO (Jurnal Informatika dan Komputer),* vol. 02, no. 1, pp. 34-42, 2018.

[8] R. Ramdhani, "Teknik Steganografi End Of File (EOF) dan Algoritma Kriptografi Rijndael Untuk Keamanan File Dokumen," Skripsi, Universitas Singaperbangsa Karawang, Karawang, 2018.

[9] R. Munir, *Kriptografi*, Edisi Kedua, Bandung: Informatika, 2019.

[10] L. Widyawati, "Implementasi Metode Steganografi Slt-Dct Pada Citra Untuk Meningkatkan Kualitas Citra Steganografi," Skripsi, Universitas Islam Indonesia, Yogyakarta, 2019.

[11] E. Junianto dan M. Z. Zuhdi, "Penerapan Metode Palette Untuk Menentukan Warna Dominan Dari Sebuah Gambar Berbasis Android," *JURNAL INFORMATIKA,,* vol. 5, no. December, pp. 62-73, 2018.

[12] R. Ilyas dan Y. H. Chisnanto, "Pengembangan Sistem Informasi Penelitian LPPM Universitas Jenderal Achmad Yani Dengan Agile SDLC," *Konferensi Nasional Sistem Informasi (KNSI) 2018,* pp. 974-979, 2018.

[13] F. Supandi, W. D. P, Y. A. S. dan M. Sudir, "Analisis Resiko Pada Pengembangan Perangkat Lunak Yang Menggunakan Metode Waterfall Dan Prototyping," *Prosiding Seminar Dinamika Informatika 2018 (SENADI 2018),* 2018, pp. 83-86.

[14] T. K. Tia dan W. A. K, "Model Simulasi Pengembangan Perangkat Lunak Menggunakan Rational Unified Process (RUP)," *Teknika : Engineering and Sains Journa,* vol. 2, no. 1, pp. 33-40, 2018.

[15] Gunarto, A. Abdullah dan D. Irawan, "Model Matematis Turbin Pelton Dengan Menggunakan Bahasa Pemrograman Java," *Machine; Jurnal Teknik Mesin Vol.,* vol. 4, no. 2, pp. 9-14, 2018.

[16] I. Gunawan, Sumarno, E. Irawan dan H. S. Tambunan, "Pengamanan Berkas Dokumen Menggunakan Fungsi Algoritma Steganografi LSB," *ALGORITMA: Jurnal Ilmu Komputer dan Informatika,* vol. 02, no. 01, pp. 61-65, 2018.

[17] B. Santoso, "Analisis Webometrics terhadap Repositori Institusi Perguruan Tinggi Keagamaan Islam Negeri (PTKIN): Kajian terhadap 5 PTKIN di Indonesia," *Lentera Pustaka: Jurnal Kajian Ilmu Perpustakaan, Informasi dan Kearsipan,* vol. 5, no. 2, p. 121, 2019.