IMPLEMENTASI ALGORITMA *VIGENERE CIPHER* DAN METODE *LEAST SIGNIFICANT BIT* (LSB) DALAM PERANCANGAN APLIKASI STEGANOGRAPHY UNTUK MENYISIPKAN PESAN TEKS PADA GAMBAR

Riki Sugiarto[1)], Mufti[2)]
Jurusan Teknik Informatika. Fakultas Teknologi Informasi. Universitas Budi Luhur
Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260
E-mail: rikisugiarto009@gmail.com[1)], muftyhayat@gmail.com[2)]

Abstrak

Pengamanan kerahasiaan citra digital pada Sekolah Dasar Negeri Pengasinan IX Kota Bekasi masih memberikan celah untuk adanya kebocoran informasi. Saat ini perkembangan teknologi informasi berkembang dengan sangat pesat, maka taraf pengamanan informasi harus disesuakan. Oleh karena itu dibutuhkan suatu aplikasi yang dapat menjaga kerahasiaan informasi citra tersebut. Aplikasi yang dapat menjaga kerahasiaan informasi citra adalah kriptografi dan steganografi. Dengan penggabungan algoritma kriptogorafi dan steganografi, sistem pengamanan informasi citra digital akan menjadi lebih aman. Algoritma kriptografi yang digunakan adalah algoritma Vigenere Cipher. Algoritma Vigenere Cipher ini menghasilkan setiap karakter pesan pada plainteks berkorespondensi dengan lebih dari satu karakter pada cipherteks. Algoritma steganografi yang digunakan adalah algoritma LSB yang menggunakan gambar sebagai media penampungnya. Algoritma LSB ini Algoritma steganografi Least Signifikan Bit (LSB) mengubah nilai pixel pada citra dalam representasi biner. Perubahan dapat dilakukan dengan berbagai cara dan algoritma, misalnya mengubah nilai biner 0 menjadi 1 atau sebaliknya. Teknik ini memanfaatkan karakteristik penglihatan manusia yang tidak dapat melihat perubahan pola biner yang terjadi pada gambar. Pada penelitian ini data image yang digunakan adalah citra dengan format JPG (*Joint Photographic Group*). Dengan adanya teknik kriptografi pada media teks dan steganografi pada media gambar, maka pengiriman suatu pesan yang bersifat rahasia akan memiliki tingkat keamanan yang sangat tinggi karena citra sudah terenkripsi dan perubahan pola biner tidak dapat dideteksi langsung oleh indera penglihatan manusia. Hasil riset ini membuktikan bahwa implementasi kriptografi dan steganografi teks kedalam gambar menggunakan metode Vigenere Cipher dan LSB berbasis desktop pada SDN Pengasinan IX Kota Bekasi dapat dengan sukses diimplementasikan sebagai solusi bagi kebutuhan keamanan data.

Kata kunci: Steganografi, Kriptografi, Metode LSB, dan Vigenere Cipher

1. PENDAHULUAN

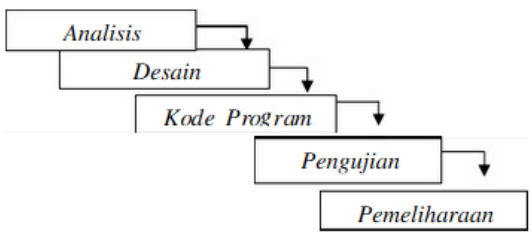
Keamanan dan kerahasian pada data ini merupakan suatu gagasan aspek yang sangat penting dalam sebuah sistem informasi pada saat ini. Disebabkan pesatnya perkembangan ilmu pengetahuan dan teknologi yang memungkinkan munculnya teknik-teknik baru, yang disalahgunakan oleh pihak-pihak tertentu yang mengancam keamanan dari sistem informasi tersebut. Jatuhnya informasi ke tangan pihak lain dapat menimbulkan kerugian bagi pemilik informasi.

Karena itu muncul suatu gagasan yang mengacu pada permasalahan tersebut, yaitu untuk membuat suatu sistem keamanan yang dapat melindungi data yang dianggap penting dengan penyandian data, serta membuat kunci rahasia untuk dapat membuka data tersebut yang sulit untuk di deteksi oleh pihak yang tidak berhak, serta bisa di sisipkan kedalam media lain seperti video, gambar, audio dan lain-lain.

Berdasarkan latar belakang diatas, maka disini peneliti akan membahas teknik kriptografi algoritma *Vigenere Cipher* dan steganografi pada gambar menggunakan metode *Least Significant Bit (LSB).* Maka dengan ini peneliti akan mencoba menjelaskan tentang bagaimana menyisipkan pesan rahasia berupa teks pada sebuah gambar. Dalam steganografi gambar ini menggunakan metode *Least Significant Bit (LSB),* Hal ini dikarenakan pada metode *LSB* ini dapat menyimpan file gambar pada bit yang paling rendah sehingga hasil dari gambar steganografi tidak akan kelihatan terjadi perubahan. Pada sebuah gambar menjelaskan tentang bagaimana mengembangkan agar pesan yang disembunyikan tidak hanya berupa pesan rahasia tapi pesan yang telah dienkripsikan menggunakan algoritma *Vigenere Cipher*. Dengan menambahkan Steganografi kedalam pesan rahasia yang disembunyikan kedalam gambar akan memperkuat keamanan dari pesan rahasia. Dengan penerapan teknik kriptografi dan steganografi ini diharapkan dapat membantu dalam proses penyimpanan data nilai rapot dengan aman.

2. METODE PENELITIAN

Tahapan yang akan dilakukan oleh peneliti dalam pembangnan aplikasi sistem ini dengan menggunakan sebuah metode waterfaal yang dapat dilihat pada gambar di bawah ini:



Gambar 1. Metode Waterfall

Penjelasan yang dimaksud dari setiap point dari proses yang terdapat pada gambar 1.1. metode waterfall di atas sebagai berikut:

a. Pengumpulan data

Pengumpulan data yang dipake oleh sang peneliti pada penelitian inimenggunakan metode studi literatur atau berupa karya-karya ilmiah yang berhubungan dengan penelitian yang dikerjakan.

1) Metode Kepustakaan, metode ini dilakukan oleh peneliti dengan cara mengumpulkan data serta mencari dan membaca buku-buku referensi serta jurnal yang berkaitan dengan penelitian ini.
2) Metode Wawancara, metode ini adalah suatu cara memperoleh data dengan mendatangkan langsung ke sekolah untuk Tanya-Jawab mengenai permasalahan yang dihadapi dan mendiskusikan aplikasi

b. Analisis Sistem

Analisis sistem akan ditetapkan dengan memperhatikan kapasitas informasi dan proses pengolahan data yang diperlukan.

c. Perancangan Perangkat lunak

Perancangan pada Perangkat Lunak pengolahan data meliputi struktur data, menu dan prosedur pengoperasian.

1) Citra yang akan di proses adalah citra dengan jenis format TXT.
2) Gambar yang dihasilkan akan memliki jenis format PNG.

d. Pembuatan Perangkat Lunak

Pembuatan Perangkat Lunak pengolahan data akan dapat dilakukan dengan menggunakan bahasa pemograman yang tepat, efektif dan efisien dengan memperhatikan teknologi komputerisasi dengan cara menggunakan sebuah Bahasa pemrograman yang diintegrasikan dengan aplikasi Netbeans.

e. Pengujian Perangkat Lunak

Pengujian terhadap Perangkat Lunak pengolahan data akan dilakukan dengan sistematis yaitu secara parsial pada tahapan pembuatan dan pengujian akhir terintegrasi secara menyeluruh untuk memastikan bahwa pemograman yang dirancang telah memenuhi hasil yang diinginkan.

1) Enkripsi citra tertentu.
2) Proses steganografi yaitu penyisipan citra yang sudah ter-enkripsi kedalam gambar.
3) Ekstraksi steganografi yaitu suatu mengeluarkan citra yang telah disisipkan kedalam gambar.
4) Proses dekripsi citra hasil ekstraksi steganografi.

f. Pembuatan Laporan

Hasil pada sebuah penelitian yang dilakukan sang peneliti ini dapat didokumentasikan untuk sebuah kebutuhan yang lebih lanjut guna pengembangan secara selanjutnya apabila dibutuhkan.

2.1 Metode *Least Significant Bit* (LSB)

Metode Least Sgnificant Bit (LSB), Merupakan Strategi penyembunyian data didalam citra yang digunakan untuk menyisipkan sebuah citra kedalam sebuah media. Dimana sebuah bit terdaoat sebuah data citra yang akan digantikan dengan bit paling rendah dalam sebuah media citra. Pada file citra 24 bit setiap piksel yang terdapat pada citra terdiri dari susnan tiga warna, yaitu warna merah, warna hijau, dan warna biru (RGB) yang tersusun masing-masing bilangan 8 bit (1 *byte*) yang dimulai dari 0 sampai dengan 255 atau dengan sebuah bilangan biner dari 00000000 sampai dengan 11111111. Informasi dari sebuah warna biru itu sendiri berada pada bit 1 sampai dengan bit ke 8, dan informasi warna hijau itu sendiri berada pada bit 9 sampai dengan bit ke 16, sedangkan informasi warna merah itu sendiri berada pada bit 17 sampai dengan bit 24.

Dalam menjelaskan metode ini, digunakan citra digital sebagai stegamedium. Pada setiap *byte* terdapat dari sebuah bit yang tidak signifikan. Misalnya pada *byte* 000110001, maka bit *Least Significant Bit*, sebab pengubahan bit tersebut hanya akan dapat mengubah nilai *byte* nya yang menjadi satu nilai lebih tinggi atau juga bisa satu nilai lebih rendah. Sebagai contoh dari pembahasan ini, urutan bit berikut ini menggambarkan 3 piksel yang terdapat pada stegomedium 24 – bit.

piksel 1 = (00100111 11101001 11001000)

piksel 2 = (00100111 11001000 11101001)

piksel 3 = (11001000 00100111 11101001)

Pesan yang akan disisipkan adalah sebuah karakter yang bernilai “A”, yang mempunyai nilai biner-nya adalah **01000001 (ASCII)** maka dapat menghasilkan *stego image* dengan urutan bit sebagai berikut.

piksel 1 = (0010011**0** 11101001 11001000)

piksel 2 = (0010011**0** 11001000 11101000)

piksel 3 = (11001000 00100111 11101001)

Terlihat hanya pada tiga bit terendah yang akan berubah (bit dengan yang digaris bawah), untuk mata manusia tidak akan tampak perubahan yang terlihat. Secara nilai rata-rata dengan metode ini hanya terdapat setengah dari data bit yang rendah yang akan dapat berubah, sehingga bila akan dibutuhkan dapat menggunakan bit rendah yang kedua bahkan dapat pada bit rendah ketiga.

2.2 Vigenere Cipher

Vigenere Cipher berasal dari nama penemunya yaitu Blaise de Vigenere adalah seorang kriptografer asal dari *France*. *Vigenere Cipher* adalah sebuah

pengembangan dari Caesar Cipher. Pada Caesar cipher itu sendiri, setiap pada huruf plainteks selalu akan digantikan dengan huruf lain yang memiliki perbedaan tertentu pada urutan terdapat pada alphabet. Sedangkan pada *vigenere cipher*, pada setiap karakter pada pesan plainteks yang berkorespondensi dengan lebih dari satu karakter yang terdapat pada cipherteks. Misalkan huruf A pada plainteks dapat berubah menjadi huruf R atau S pada cipherteks yang dapat berkaitan, tergantung pada kunci yang akan digunakan oleh admin.

Algoritma *Vigenere Cipher* menggunakan sebuah persegi *vigenere yang* untuk melakukan sebuah enkripsi dan deskripsi. Deretan pada huruf mendatar yang terletak pada bagian atas persegi menyatakan plainteks, sedangkan deretan pada huruf menurun pada bagian yang terdapat disebelah kiri tabel persegi menyatakan sebuah kunci. Pada setiap baris yang terdapat di dalam persegi menyatakan huruf-huruf cipherteks yang dapat diperoleh dengan Caesar cipher, yang dimana jumlah pergeseran huruf plainteks ditentukan pada nilai numeric huruf pada kunci itu tersebut (yaitu A=0, B=1, C=2 ….. Z=25). Jika terdapat panjang dalam sebuah kunci yang akan digunakan lebih pendek dari sebuah plainteks, maka kunci tersebut akan diulang secara periodic sepanjang plainteks.

Tabel 1. Persegi *Vigenere Cipher*

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Langkah-langkah yang digunakan untuk sebuah proses enkripsi pada metode *vigenere Cipher*, sebagai berikut:

a. Menghilangkan spasi dan tanda baca pada plainteks.
b. Menuliskan kunci secara periodic sepanjang karakter plainteks.
c. Mengenkripsikan setiap karakter plainteks dengan sebuah karakter kunci menggunakan sebuah persegi *Vigenere Cipher*. Cipherteks itu sendiri akan diperoleh dari perpotongan antara yang dimaksud pada persegi kolom karakter plainteks.

Untuk proses deskripsinya merupakan kebalikan dari proses sebuah enkripsi. Plainteks dapat diperoleh dari perpotongan baris karakter kunci dengan karakter cipherteks yang dimaksud pada sebuah persegi *Vigenere Cipher*. Sebagai contoh, untuk sebuah plainteks "Skripsi Steganografi 2019" dengan kunci "budiluhur", mengacu pada tabel *Vigenere Cipher* akan menghasilkan sebuah cipherteks sebagai berikut:

Plainteks : Skripsi Steganografi 2019

Kunci : budiluhur

Cipherteks : 0{nztJzCgTNholHvAIbzl7dk8t

Secara matematis, misalkan P_i adalah karakter ke – i pada plainteks, C_i adalah karakter ke- i pada cipherteks, dan K_i adalah karakter ke – i pada kunci, maka sebuah enkripsi pada metode *vigenere cipher* dapat dinyatakan sebagai berikut:

$$C_i = (P_i + K_i) \bmod 26$$

Sedangkan untuk deskripsi pada *vigenere cipher* dapat dinyatakan sebagai:

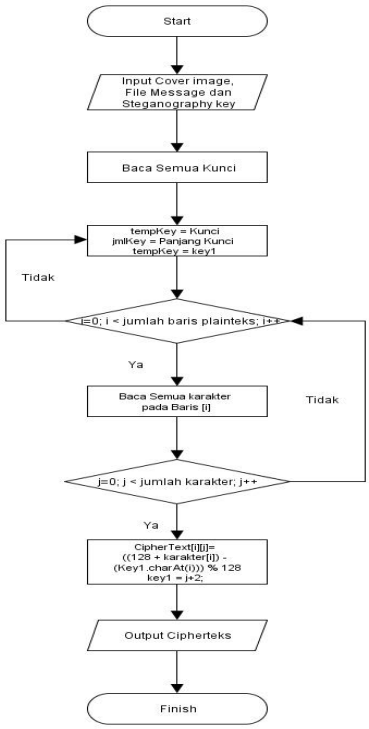
$$P_i = (C_i + K_i) \bmod 26$$

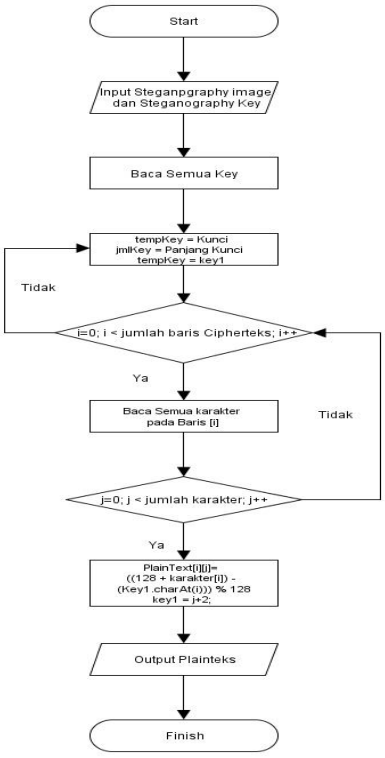
Maka harus diperhatikan kalau angka 26 pada persamaan (2.1) dan (2.2) akan disebabkan karena banyak huruf yang akan dienkripsi hanya abjad biasa, yaitu 26 huruf. Maka apabila banyak karakter yang akan dienkripsi misalnya berupa karakter berbentuk ASCII, maka angka 26 digantikan dengan nilai 256.

3. HASIL DAN PEMBAHASAN

3.1 *Flowchart* Proses Enkripsi *Vigenere Cipher*

Flowchart ini merupakan alur dari proses mengenkripsi dengan algoritma *Vigenere Cipher*. Berikut ini adalah *flowchart* untuk proses mengenkripsi data dengan algoritma *Vigenere Cipher*:

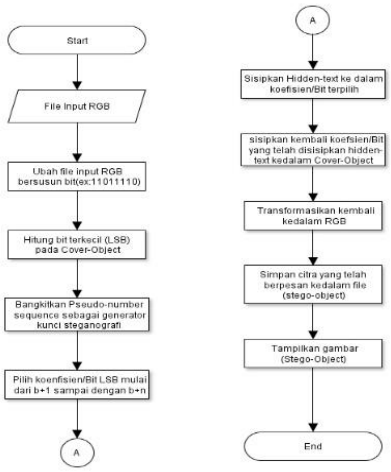




Gambar 1. Proses Enkripsi *Vigenere Cipher*

3.2 *Flowchart Proses Encode Least Significant Bit (LSB)*

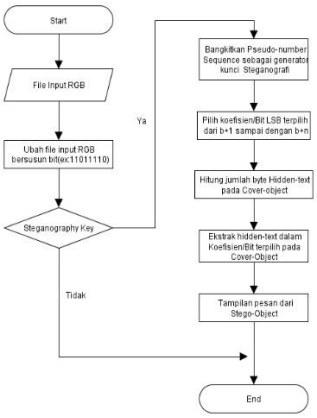
Flowchart ini merupakan alur dari proses encode data dengan metode *Least Significant Bit (LSB).* Berikut ini adalah *flowchart* untuk proses encode data menggunakan metode *Least Significant Bit (LSB).*



Gambar 2. Proses Encode Least Significant Bit (LSB)

3.3 *Flowchart* Proses Decode *Least Significant Bit* (LSB)

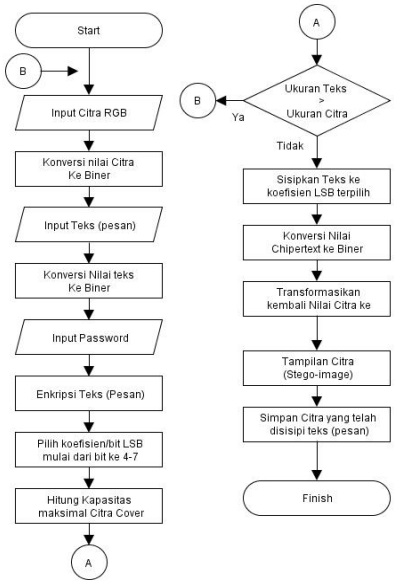
Flowchart ini merupakan alur dari proses decode data dengan metode *Least Significant Bit (LSB).* Berikut ini adalah *flowchart* untuk proses decode data menggunakan metode *Leastt Significant Bit (LSB).*



Gambar 3. Proses Decode Least Significant Bit (LSB)

3.4 Diagram Proses Alir Penyisipan Teks ke Dalam Gambar

Diagram alir pentisipan teks ke dalam gambar yang meggunakan teknik Least Significant Bit pada aplikasi steganografi dapat dijelaskan dalam diagram alir di bawah ini:



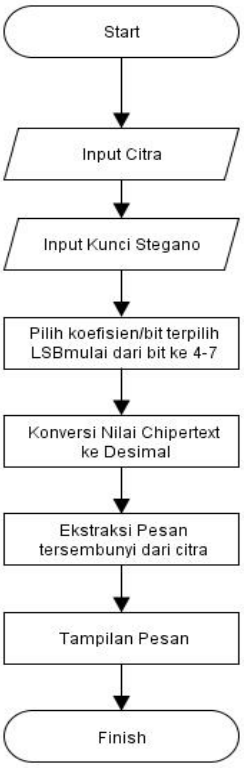
Gambar 4. Proses Alir Penyisipan Teks ke Dalam Gambar

Pada penjelasan flowchart tentang proses alir penyisipan teks ke dalam gambar, gambar tersebut telah dipilih oleh peneliti, nilai pixel citra *(Cover-Image)* akan selalu dikonversi kedalam bentuk biner (8-bit). Kemudian itu akan memasukan atau memilih sebuah file teks *(plaintext)* yang akan disiisipkan serta akan dilanjutkan dengan membangkitkan *pseudo-number* Vigenere Cipher. Kemudian proses *embedding* selesai makaa pada sistem akan menghitung kapasitas kemampuan yang terdapat pada *cover-image* maka pada siebuah sistem akan menolak untuk melanjutkan dan akan meminta untuk mengurangi jumlah pada pesan. Jika *cover-image* dapat menampung maka sebuah sistem akan memproses melanjutkan mengkonversi nilai pada bilangan Biner *Chiper-text (Stego-image)* ke dalam nilai bilangan decimal dan akan dilanjutkan

konversi nilai decimal ke nilai pixel dan pada akhirnya akan menyimpan *(save) File Cover-image* yang telah disisipkan sebuah pesan *(Chiper-text)*.

3.5 Diagram Proses Alir Ekstraksi Teks Yang Terdapat Pada Sebuah Gambar

Diagram alir ekstraksi teks yang terdapat pada sebuah gambar yang meggunakan teknik Least Significant Bit pada aplikasi steganografi dapat dijelaskan dalam diagram alir di bawah ini:



Gambar 5. Proses Alir Ekstraksi Teks Yang Terdapat Pada Sebuah Gambar

Pada proses ekstraksi teks yang terdapat pada sebuah image akan dipilih *stego-image* dengan nilai pixel yang terdapatnya, akan dikonversi ke bilangan Biner (8-bit) dan akan selalu dilanjutkan dengan membangkitkan sebuah *pseudo-number* Vigenere Cipher. Selanjutnya mengambi nilai yang terdapat pada bit terakhir (LSB) yang di setiap pexelnya, dan nilai biner akan dikonversi dari sebuah bilangan biner ke decimal dan pesan *(Plain-text)* akan ditampilkan.

3.6 Algoritma Antarmuka Encode

Pada Gambar 6 merupakan gambar Algoritma *antarmuka encode* yang menjelaskan bahwa menu ini adalah proses enkripsi teks yang akan di sisipkan kedalam gambar

```
Tampilan Antarmuka Encode
Tampilan Form Encode
Input Pilih
If Pilih = Browser Cover Image Then
    If pilih = cover image berformat .jpg then
        Input Cover image berformat .jpg
        Kembali kehalaman Form Encode
    Else
        Kembali ke halaman pilih cover image
    End if
Else if Pilih = Browser File Message then
    If pilih = File Message berformat .txt then
        Input File Message berformat .txt
        Kembali Kehalaman Form Encode
    Else
        Kembali ke halaman pilih File Message
    End if
Input Steganography key
Else if pilih = Embedding image then
        Proses Enkripsi Vigenere Cipher
        Proses Encode List Significant Bit(LSB)
        Save nama File
        Input File Nama untuk penyimpanan hasil encode
        If pilih = ok then
            Tampilan pesan Notifikasi Success
            Output Gambar Hasil Encode
            Kembali kehalaman Form Encode
        Else if Pilih = cancel Then
            Kembali kehalaman form Encode
        Else
            Kembali Kehalaman Input file nama
        End if
Else if pilih = New Then
        Kembali kehalaman Form encode
    End If
Else if Pilih = Close Then
End if
```

Gambar 6. Algoritma Antarmuka Encode

3.7 Algoritma Antarmuka Decode

Pada gambar 7 merupakan gambar Algoritma *Menu antarmuka decode* yang menjelaskan bahwa menu ini adalah proses deskripsi teks yang telah di sisipkan kedalam gambar.

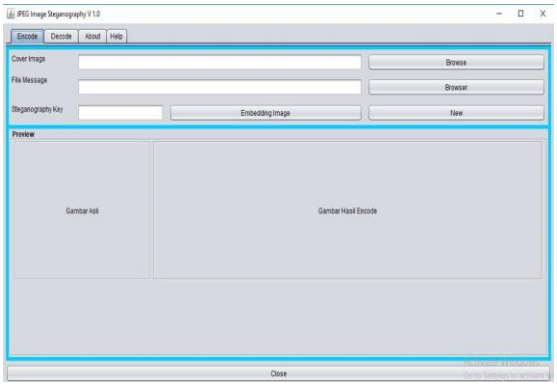
```
Tampilan Antamuka Decode
Tampilan Form Decode
Input Pilih
If Pilih = Browser Steganography Image Then
    If pilih = Steganography image berformat .jpg then
        Input Cover image berformat .jpg
        Output Object Gambar
        Kembali kehalaman Form Decode
    Else
        Kembali ke halaman pilih cover image
    End if
```

```
Input Steganography key
Else if pilih = Decoding image then
        Proses Encode List Significant Bit (LSB)
        Proses Enkripsi Vigenere Cipher
        Tampilan Pesan Success
        Tampilan Hasil Ciphertext dan Plainteks
        Kembali kehalaman Form decode
        End if
Else if pilih = Save then
      Input File nama untuk penyimpanan hasil cipherteks
      If pilih = oke then
                Tampilan Pesan Notifikasi Success
            Kembali Kehalaman Form Decode
            Else If pilih = Cancel Then
                Kembali Kehalaman Form Decode
            Else
                Kembali Kehalaman Input File Nama
            End if
Else If Pilih = New Then
        Kembali Kehalaman Form Encode
        End If
Else If Pilih = Close Then
End if
```

Gambar 7. Algoritma Antarmuka Decode

3.8 Tampilan Form Halaman Utama

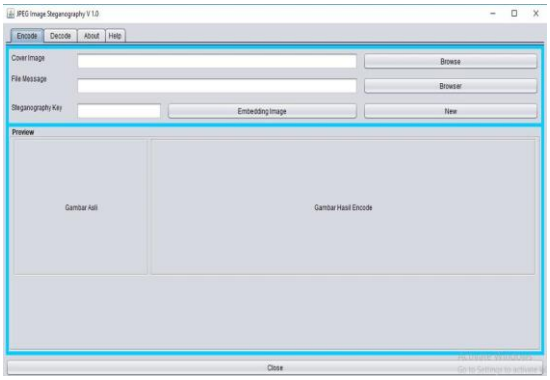
Pada Gambar 8 merupakan tampilan form halaman utama, user dapat melihat halaman utama yaitu berisi tampilan awal aplikasi



Gambar 8. Tampilan Form Halaman Utama

3.9 Tampilan Form *Encode*

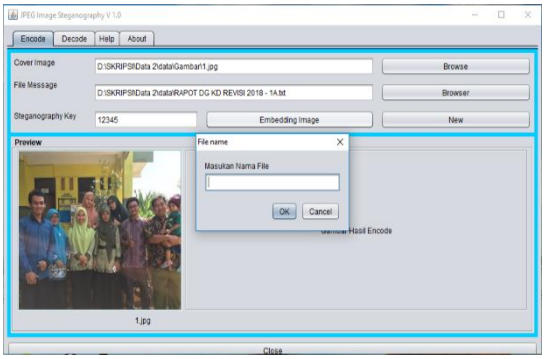
Pada gambar 9 merupakan tampilan *form encode*, pada tampilan ini berfungsi untuk proses enkripsi dan menyisipkan teks ke gambar. Tampilan ini terdapat *button browser* gambar yang berguna untuk mengambil gambar yang untuk menjadi *cover parent*, *button browser message* yang berguna untuk memasukan teks yang untuk disisipkan ke gambar, dan terdapat *input* steganography *key*.



Gambar 9. Tampilan Form Encode

3.10 Tampilan Nama File

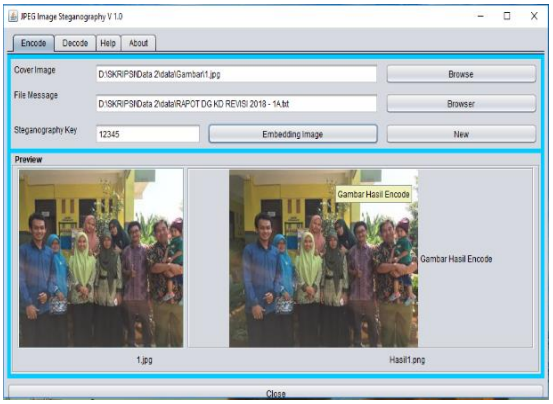
Pada gambar 10 merupakan tampilan file *name*, pada tampilan ini adalah tampilan yang menujukan informasi pemberian nama penyimpanan file yang terenkripsi.



Gambar 10. Tampilan Nama File

3.11 Tampilan Form Enkripsi

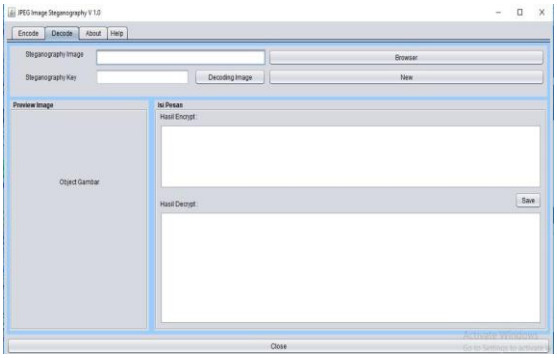
Pada gambar 11 merupakan tampilan form Enkripsi, pada tampilan ini berfungsi sebagai hasil proses yang telah berhasil di enkripsi. Tampilan ini terdapat gambar asli, berserta gambar yang telah di enkripsi yang menandakan enkripsi sukses terenkripsi.



Gambar 11. Tampilan Form Enkripsi

3.12 Tampilan Form *Decode*

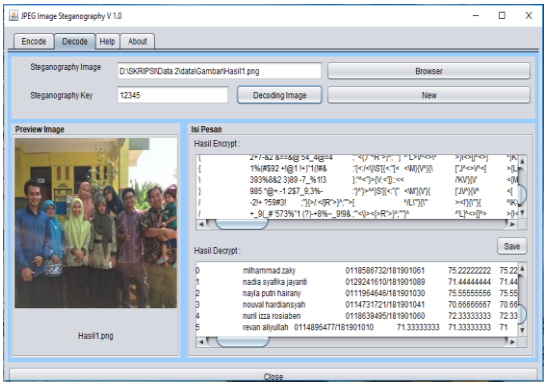
Pada gambar 12 merupakan tampilan *form decode*, pada tampilan ini berfungsi intuk proses deskripsi atau membalikan teks dari gambar. Tampilan ini terdapat *button browser Steganography image* yang berguna mengambil gambar yang telah di enkripsi, dan terdapat input *steganography key.*



Gambar 12. Tampilan Form Decode

3.13 Tampilan Form Ekstraksi

Pada gambar 13 merupakan tampilan form ekstraksi, pada tampiilan ini berfungsi sebagai proses mengektrak file *cover parent*. Berserta tampilan ini menjelaskan hasil isi pesan, yang terdapat hasil *encrypt* dan hasil *decrypt*.



Gambar 13. Tampilan *Form Ekstraksi*

3.14 Pengujian Sistem

Tahap pengujian yang dilakukan pada aplikasi ini antara lain pengujian Hasil Uji coba kecocokan, dan Hasil Uji lama Waktu Proses.

a. Hasil Uji Coba Kecocokan

Pada hal uji coba kecocokan ini dapat dilakukan pengujian untuk mengetahui sejauh mana tingkat kecocokan antara file teks dan gambar dari sistem yang dibuat.

Tabel 2. Pengujian Kapasitas Citra Cover JPG

					Decode		ok-an
Nama File	Size	Resolusi	Id	Size Stegano	Id	Size Ekstrak	True/ False
1.jpg	57.7kb	640x480	H1.png	80 kb	H1.png	47.6 kb	True
2.jpg	81,1 kb	400x300	H2.png	139 kb	H2.png	68,7 kb	True
3.jpg	184 kb	640x480	H3.png	286 KB	H3.Png	160 kb	True
4.jpg	265 kb	800x600	H4.png	511 kb	H4.png	235 kb	True
5.jpg	406 kb	1024x765	H5.png	921 kb	H5.Png	367 kb	True
6.jpg	873 kb	1600x1e200	H6.png	1,51 Mb	H6.PNG	859 kb	True
7.jpg	3,42 mb	3264x2448	H7.png	7,38 mb	H7.png	3,37 mb	True

Berdasarkan pada table 2 di atas terlihat semua file teks mempunyai nilai cek kecocokan yang bernilai True, karena semua file dapat terenkripsi kedalam cover image.

b. Hasil Uji Coba Lama Waktu Proses

Pada uji coba ini dilakukan pengujian untuk mengetahui lama waktu proses dari teks dan gambar dengan resolusi yang berbeda-beda.

Tabel 3 Uji Coba Lama Waktu Proses

File Teks		*File Gambar*		*Waktu Proses(sec)*	
Nama	*Size*	*Nama*	*Resolusi*	*Encode*	*Decode*
Rapotdg kd Revisi 2018-1A.txt	*3.284 kb*	*1.jpg*	*320x240*	*0,045*	*0,021*
		2.jpg	*400x300*	*0,063*	*0,026*
		3.jpg	*640x480*	*0,129*	*0,064*
		4.jpg	*800x600*	*0,291*	*0,082*
		5.jpg	*1024x765*	*0,426*	*0,104*
		6.jpg	*1600x1200*	*0,860*	*0,111*
		7.jpg	*3264x2448*	*1,011*	*0,321*

3.15 Kelebihan Program

Pada penerapan Aplikasi Steganografi dapat menyisipkan teks ke dalam *file* gambar tanpa sedikitpun terlihat oleh mata telanjang. Serta Aplikasi ini selain bisa menyisipkan pesan tersembunyi, juga bisa mengekstrak kembali pesan tersembunyi itu dengan bentuk aslinya. Dan Aplikasi steganografi ini juga bisa menyisipkan file gambar yang beresolusi hingga 3264x2448, terlihat pada tabel 3 pengujian kapasitas citra cover JPG.

3.16 Kelebihan Program

Aplikasi steganografi ini hanya bisa di sisipkan file teks yang berformat *Text Document* (.txt) saja. Aplikasi steganografi ini juga hanya tersedia dalam versi *desktop* yang di buat oleh peneliti. Dan jugaAplikasi ini hanya dapat File Image sebagai wadah penampungnya.

4. KESIMPULAN

Dari hasil analisis ini peneliti memberikan kesimpulan yang diambil dari masalah yang terdapat pada pembahasan ini, yaitu dengan Telah berhasil dikembangkan aplikasi perangkat lunak yang dapat mengerjakan dan melakukan steganografi teks pada gambar dengan menggunakan metode List

INFO TEKS			
Nama File	Ukuran	Type Of File	
Rapot dg kd Revisi 2018- 1A.txt	3.284 Kb	Text Document (.txt)	
Info file Gambar	Hasil proses Encode	Hasil Proses	Hasil kecoc

Significant Bit (LSB) dan algoritma Vigenere Cipher. Kebutuhan fungsional dari perngkat lunak ini dapat memprosespenyisipan dan ekstraksi pesan teks sudah dapat dilakukan dengan baik dan benar. Serta aplikasi steganografi Dengan metode List Significant Bit (LSB) dapat dilakukan penyisipan pesan tersembunyi berupa teks ke dalam berkas citra digital berformat JPG dan dapat mengekstraksi sebuah teks tersembunyi tersebut dari dalam gambar. Namun secara kasat mata, perbedaan antara gambar sebelum dan sesudah disisipkan pesan tidak terlihat. Selain itu waktu yang dapat dibutuhkan untuk suatu proses enkripsi dan dekripsi dipengaruhi oleh kecepatan sebuah komputer yang di gunakan dan besar ukuran citra itu sendiri.

DAFTAR PUSTAKA

[1] Cahyadi, W. (2012). *Teori dan Aplikasi Pengolahan citra*. Penerbit Andi: Yogyakarta

[2] Mardhatillah, Dilla. (2017). *Implementasi Steganografi Least Significant Bit (LSB) dengan Algoritma Super Enkripsi pada File Citra.* Medan: Universitas Sumatera Utara.

[3] Munir, Rinaldi. (2006). *Kriptografi*. Penebit Informatika, Bandung.

[4] Nugroho, Djayadi dan Maftuhin. (2013). Penerapan steganografi pada file gambar(JPG) menggunakan metode LSB dengan aplikasi matlab. *Jurnal Sibernetika*, vol1, no. 1, pp.53- 58.

[5] Niria Laila, Anita Sindar RMS. (2018). Implementasi Steganografi LSB dengan Enkripsi Vigenere Cipher pada Citra. *Computer Science Informatika Journal*, vol. 1, no. 2, 2018. Medan: STMIK Pelita Nusantara.