e-ISSN: 2549-7952
p-ISSN: 2580-3336

Sistem Keamanan Data Teks dengan Steganografi Citra Gambar Menggunakan Algoritma End Of File

Arvin Argananta Gilbijatno[1], Patmi Kasih[2],
[1,2,]Teknik Informatika, Fakultas Teknik, Universitas Nusantara PGRI Kediri
E-mail: *[1]*aargananta@gmail.com*, [2] *fatkasih@gmail.com*

Abstrak-*Steganografi merupakan ilmu dan seni yang mempelajari cara penyembunyian informasi pada suatu media sedemikian rupa sehingga keberadaannya tidak terdeteksi oleh pihak lain yang tidak berhak atas informasi tersebut. Pengguna pertama (pengirim pesan) dapat mengirim media yang telah disisipi informasi rahasia tersebut melalui jalur komunikasi publik, hingga dapat diterima oleh pengguna kedua (penerima pesan). Penerima pesan dapat mengekstraksi informasi rahasia yang ada di dalamnya. Pada penelitian ini sistem dibuat dengan menggunakan metode End of File untuk proses penyisipan dan ekstraksi pesan. Sistem keamanan data dibuat bekerja dengan cara melakukan enkripsi teks ASCII dan citra gambar. Lalu menggabungkan keduanya dalam suatu media dengan menggunakan metode end of file. Pesan yang berupa plaintext (data teks) akan di ubah ke ASCII kemudian disisipkan pada akhir dari media citra yang digunakan. Dengan adanya metode ini memungkinkan kita bisa saling bertukar informasi tanpa adanya rasa khawatir pesan rahasia yang kita kirim diketahui oleh orang yang tidak berhak menerimanya. Pada pengujian dihasilkan karakteristik metode end of file adalah mampu menampung lebih banyak data atau pesan sehingga memungkinkan dapat menyisipkan lebih banyak pesan yang akan disisipkan. Tapi tetap ditentukan oleh ukuran panjang pesan yang akan disisipkan agar tidak mempengaruhi citra penampung yaitu image.*

Kata Kunci : *Steganografi, Data Teks, Gambar, End Of File (EOF)*

1. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi berperan penting dalam mempermudah masyarakat untuk saling bertukar informasi. Namun informasi yang bersifat rahasia saat ini rentan dicuri oleh orang yang tidak bertanggung jawab. Karena pada dasarnya pengiriman informasi dilakukan tanpa adanya pengamanan terhadap konten yang dikirim. Ketika terkena penyadapan, maka data dapat langsung dibaca oleh penyadap.

Pengamanan informasi bisa dilakukan dengan berbagai cara, salah satunya adalah penyandian pesan menggunakan kode kode yang rumit ataupun acak. Oleh sebab itu diperlukan ilmu yang mempelajari keamanan informasi. Dan ilmu yang mempelajari sistem keamanan informasi tersebut adalah kriptografi. Kriptografi berasal dari dua kata, yaitu *cryptos* dan *graphein. Cryptos* berarti rahasia dan *graphein* berarti tulisan, sehingga menurut bahasa, kriptologi adalah tulisan rahasia.

Menurut (Eko Arryawan, 2010) kriptografi adalah "ilmu untuk menyembunyikan isi pesan yang disandikan sehingga tidak diketahui apa isi pesan tersebut". Pada kriptografi terdapat dua proses utama yakni *encoding* dan *decoding*. "Proses *encoding* dan *decoding* diatur oleh satu atau lebih kunci kriptografi" (Wirdasari, 2008). Dalam kriptografi pesan asli yang akan dikirim terlebih dahulu dikodekan, proses ini disebut Enkripsi. Sementara itu, untuk mengembalikan ke bentuk pesan asli disebut Dekripsi. Pesan asli disebut *Plaintext* dan pesan yang sudah dirahasiakan disebut *chipertext.*

Namun disisi lain kriptografi dapat menimbulkan kecurigaan pada orang yang membaca data terenkripsi. Teknik lain sebagai upaya pengamanan data adalah Steganografi. Cara kerja dari steganografi adalah menyamarkan pesan rahasia pada suatu media digital dengan teknik penyisipan.

Penelitian ini terinspirasi keinginan untuk membuat suatu sistem keamanan data teks dengan menerapkan metode *End Of File* dengan teknik steganografi berbasis web. Selain itu peneliti ingin mengetahui bagaimana membuat sistem keamanan data berupa teks yang nantinya akan disisipkan pada citra gambar. Dalam penelitian ini digunakan data awal atau pesan rahasia yang disembunyikan berformat teks yang berupa citra gambar.

Tujuan dari penelitian pembuatan sistem keamanan data ini adalah merancang dan membangun aplikasi sistem keamanan data dengan implementasi metode *End Of File* dalam teknik steganografi. Menganalisis perbedaan besar ukuran *file* citra sebelum dan sesudah disisipkan pesan.

Manfaat yang diinginkan peneliti dengan dibangunnya aplikasi ini adalah menjaga keamanan pesan rahasia yang ingin disampaikan, mengkombinasikan antara Kriptografi dan Steganografi untuk mengamankan pesan yang disisipkan di dalam citra, menghindari kecurigaan publik pada sebuah kata maupun kalimat acak.

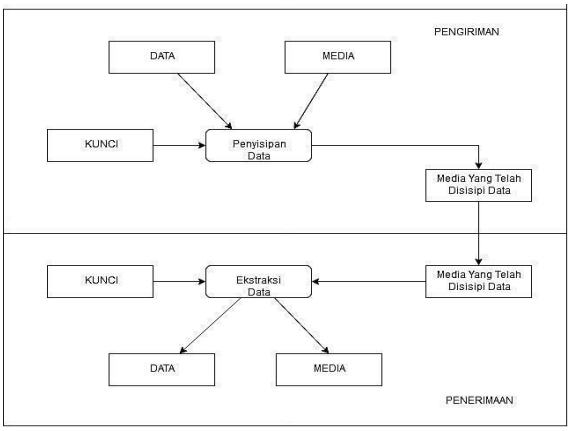
2. METODE PENELITIAN

2.1 Steganografi

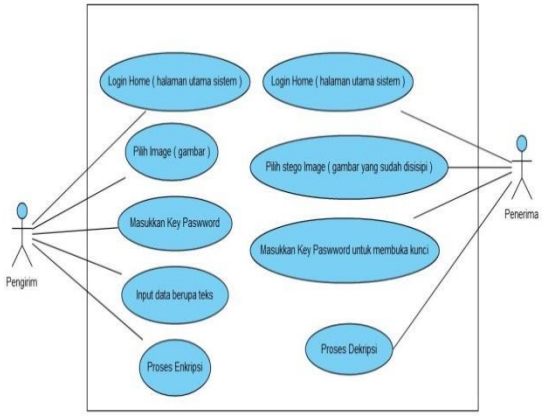
Kata Steganografi berasal dari bahasa Yunani "*steganos"*, yang artinya tersembunyi atau terselubung dan "*graphein"* artinya menulis. Menurut (Munir, 2009) Steganografi adalah ilmu untuk menyembunyikan suatu pesan rahasia sehingga keberadaan pesan tersebut menjadi tidak dapat dideteksi oleh indra manusia. Jadi Steganografi adalah sebuah teknik yang digunakan untuk menyisipkan data ke dalam sebuah media yang

e-ISSN: 2549-7952
p-ISSN: 2580-3336

bertujuan untuk menyembunyikan keberadaan data rahasia dari pihak-pihak yang tidak berkepentingan.



Gambar 1. Konsep dasar Steganografi



Gambar 2. *Use Case Diagram* Pengirim dan Penerima

Beberapa hal yang harus diperhatikan dalam penyembunyian data, adalah :

1. Tidak dapat dipersepsi *(Imperceptibility)*
2. Ketepatan *(Fidelity)*
3. Ketahanan *(Robustness)*
4. Kapasitas *(Capacity)*
5. Pemulihan *(Recovery)*

2.2 Data Text dan Citra Digital

Text merupakan sekumpulan karakter terdiri dari huruf-huruf, angka-angka (A-Z, a-z, 0-9) dan simbol-simbol lainya seperti %, &, ^, =, @, $, !, * dan lain-lain, dengan menggunakan kode ASCII setiap karakter dari text berjumlah 8-bit atau 1 byte.

Citra digital adalah gambar dua dimensi yang bisa ditampilkan pada layar komputer sebagai himpunan atau diskrit nilai digital yang disebut pixel atau picture elements. Dalam tinjauan matematis, citra merupakan fungsi kontinu dari intensitas cahaya pada bidang dua dimensi. Jenis-jenis warna pada citra digital adalah hitam putih (monochrome), Hitam Putih dan abu abu (Grayscale), dan citra digital berwarna (RGB).

2.3 Perencanaan Sistem

Sistem ini dirancang dengan user satu adalah pengirim sebagai pembuat dan pengelola pesan rahasia sebelum data tersebut dienkripsi, dikirim dan disisipkan dalam media gambar. Selanjutnya user dua sebagai penerima pesan yaitu orang tertentu yang bisa membuka pesan rahasia setelah data yang diterima diekstraksi atau didekripsi terlebih dahulu.

a. Data Input
   Dalam penelitian ini yang dijadikan data input adalah data berupa teks dan citra gambar
b. Gambaran Proses
   Data Input yang akan akan diproses, yaitu penyisipan pesan rahasia ke dalam citra digital sebagai pengirim (enkripsi end of file), dan proses ekstraksi pesan (dekripsi *stego image*) sebagai pihak penerima.
   1. Penyisipan pesan teks sebagai enkripsi pihak pengirim :
      a) Masukan pertama yaitu media penampung berupa citra gambar
      b) Masukan kedua yaitu pesan rahasia berupa pesan teks
      c) Proses steganografi enkripsi dengan metode *End Of File*
      d) Menghasilkan citra digital berupa gambar yang telah disisipi dengan data berupa teks atau biasa disebut *stego image*
   2. Proses Ekstraksi Pesan Sebagai dekripsi pihak penerima :
      a) Masukkan stego image yaitu hasil dari enkripsi awal
      b) Proses ekstraksi pesan
      c) Menghasilkan Pesan rahasia dan Gambar awal

c. Data Output
   Data yang dihasilkan setelah proses stenografi adalah data berupa sebuah file gambar yang berisi pesan asli (teks) yang sebelumnya telah dienkripsi dan disisipkan pada citra gambar.

2.4 Arsitektur dan Desain Sistem

Secara umum program steganografi ini mempunyai fungsi untuk menyembunyikan informasi berupa pesan teks di balik data citra, Dalam hal ini media yang digunakan adalah citra digital. Dan harus diperhatikan bahwa perubahan pada citra penampung yang telah termodifikasi tidak boleh terlalu terlihat, agar suatu kerahasiaan dari informasi yang ada dalam file citra digital tetap terjaga *(integrity)*.
Perencanaan dalam sistem keamanan data steganografi ini dibagi beberapa *subsistem* yaitu :

a. Use Case Diagram
   Pemodelan sistem menggunakan *Use Case Diagram* antara pengirim dan penerima pesan dapat dilihat pada gambar 2.
   Dari *Use Case Diagram* gambar 2 dapat kita simpulkan bahwa *user* (Pengirim/Penerima) dapat melakukan 6 hal:

e-ISSN: 2549-7952
p-ISSN: 2580-3336

1. Login ke sistem
2. Masukkan key antar Pengirim dan Penerima
3. Menentukan pesan rahasia yang akan dibuat
4. Menentukan Citra Masukan.
5. Melakukan proses penyisipan pesan (berlaku jika user sebagai pengirim pesan)
6. Melakukan proses ekstraksi pesan (berlaku jika user sebagai penerima pesan)

b. Activity Diagram

Proses sistem yang dilakukan untuk menyisipkan pesan digambarkan pada diagram gambar 3. Pada proses ini masukkan input file berupa pesan teks dan citra penampung berupa gambar. Saat proses enkripsi penyisipan, sistem akan merubah file file masukan tadi terlebih dahulu menjadi biner lalu hexadesimal. Dan tahap selanjutnya adalah menyisipkan pesan yang terenkripsi tadi pada akhir nilai hexadesimal citra gambar.

Selanjutnya pada gambar 4, dari diagram proses ekstraksi, citra masukan pesan tersebut adalah berupa citra yang mengandung pesan didalamnya yaitu *Stego Image*. Lalu *user* hanya perlu memilih pilihan menu dekripsi untuk mengubah dan membalik *stego image* menjadi file file masukan pertama. Terakhir, sistem akan menampilkan gambar awal dan isi dari pesan awal yang di rahasiakan.

c. Flowchart Program

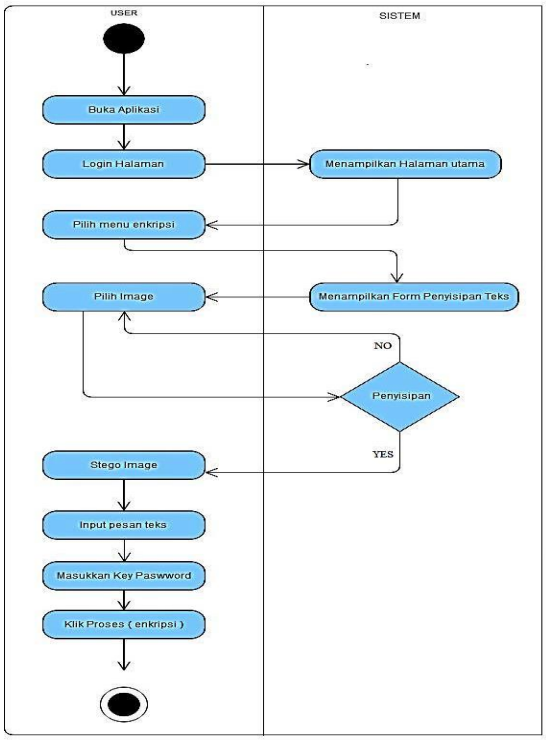
Secara keseluruhan alur aplikasi yang dibuat dalam penelitian ini digambarkan dalam bentuk flowchart diagram pada gambar 5 dan gambar 6.

Dari gambar 5, dapat dijelaskan bahwa langkah awal pada tahap ini adalah memilih citra. Citra yang dipilih merupakan citra gambar RGB yang mana dalam citra tersebut akan diubah menjadi citra biner Inputan selanjutnya adalah pesan berupa teks yang nantinya akan disisipkan. Karakter pesan yang akan disisipkan ke dalam citra harus sudah dalam bentuk sekumpulan kode ASCII, sebab letak kode ASCII tersebut akan ditempatkan pada akhir baris citra *cover*. Dan masukkan juga key password sebagai ketentuan proteksi yang hanya diketahui oleh pengirim dan penerima pesan nantinya. Terdapat keterangan jika gambar bukan merupakan *stego image* maka proses tidak bisa dijalankan. Namun jika gambar belum pernah disisipi pesan apapun maka proses enkripsi bisa dijalankan.

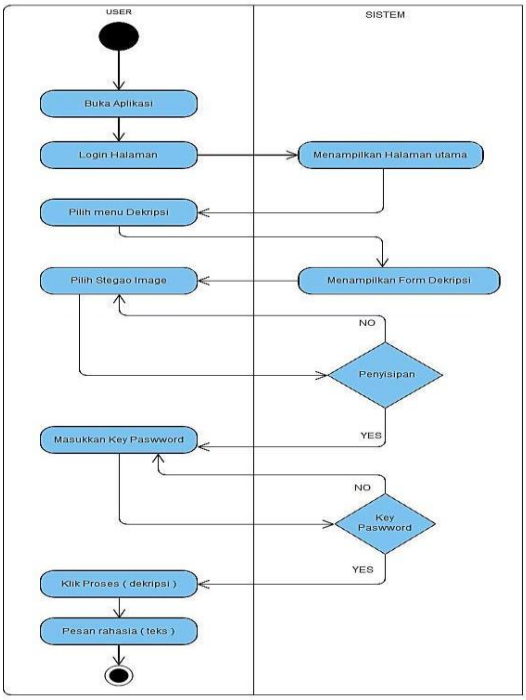
Proses ekstraksi pesan (dekripsi) bisa dilihat pada gambar 6. dan dapat dijelaskan bagaimana proses ekstraksi pesan yaitu tahap diungkapkannya kembali pesan yang telah disisipkan, sehingga penerima dapat memahami pesan yang terkandung didalam citra stego. Pada tahap ini terdapat beberapa tahapan yaitu memilih citra *stego image* dan memasukkan password. Jika password salah, program akan mengarahkan user pada pilihan "kembali" memilih *stego image* dan masukkan password sekali lagi. Jika password sudah benar, maka proses bisa dijalankan dan menghasilkan outputan berupa pesan teks yang dirahasiakan dan gambar awal sebelum disisipi.

2.5 Metode End Of File

Metode *End Of File* merupakan salah satu metode yang digunakan dalam steganografi. Metode ini menggunakan cara dengan menyisipkan data pada akhir *file*.



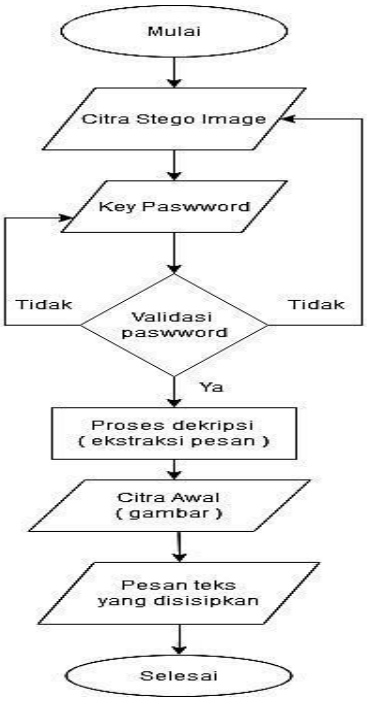
Gambar 3. *Activity Diagram* Penyisipan Teks



Gambar 4. *Activity Diagram* Ekstraksi Pesan

e-ISSN: 2549-7952
p-ISSN: 2580-3336

Gambar 5. Flowchart Proses Penyisipan teks



Gambar 6. Flowchart Proses Ekstraksi pesan

Metode *End of File* 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 data ditambah dengan ukuran data yang disisipkan ke dalam file tersebut. Dalam teknik *End Of File*, data yang disisipkan pada akhir file diberi tanda khusus sebagai pengenal start dari data tersebut dan sekaligus sebagai tanda pengenal akhir dari data tersebut. Prinsip kerja *End Of File* menggunakan karakter atau simbol khusus ctrl-z yang diberikan pada setiap akhir file.

Misalkan pada sebuah citra berukuran 5x5 *pixel* disisipkan pesan yakni “gajah”. Nilai desimal ASCII dari pesan diberikan sebagai berikut :

103 97 106 97 104

Misalkan matrik nilai desimal dari pixel citra adalah seperti gambar 7. Nilai desimal pesan disisipkan pada akhir citra menjadikan nilai seperti gambar 8.

162	78	123	212	110
35	76	191	148	68
212	234	107	56	98
122	76	112	92	107
271	178	154	177	102

Gambar 7. Matrik Decimal *Pixel* Citra Sebelum Disisipi

162	78	123	212	110
35	76	191	148	68
212	234	107	56	98
122	76	112	92	107
271	178	154	177	102
103	**97**	**106**	**97**	**104**

Gambar 8. Matrik Decimal *Pixel* Citra Setelah Disisipi

0	1	2	3	4	5
Warna A	Warna B	Warna C	Warna D	Warna E	Warna F
RGB	RGB	RGB	RGB	RGB	RGB
F7,47,47	47,91,f	47,9,b2	C,5b,1	13,34,11	15,B,B
247,71,71	71,145,247	71,240,111	235,235,135	115,52,13	245,249,249
16205639	4690423	4174674	15526791	7549969	16185549

Gambar 9. Tabel warna RGB

0	1	2	3	4	5
Warna A	Warna B	Warna C	Warna D	Warna E	Warna F

Gambar 10. Tabel grayscale

0	1	2	3	4	5
Warna A					
Warna F	Warna A				
Warna F	Warna A	Warna D			
	Warna F	Warna A	Warna D		
	Warna F	Warna A	Warna D	Warna C	
	Warna F	Warna B	Warna A	Warna D	Warna C

Gambar 11. Palet Index

e-ISSN: 2549-7952
p-ISSN: 2580-3336

Kelebihan dari metode end of file adalah tidak ada batasan dalam menambahkan informasi yang ingin disembunyikan, bahkan jika ukuran informasi itu melebihi ukuran citra penampung. Data informasi akan disembunyikan atau disisipkan di akhir *file* sehingga *file image* mungkin akan tampak ada perubahan dengan aslinya. Jika dapat dilihat mata maka perubahan ini akan tampak di baris bawah dari *image*.

a. Encode Data Teks

Proses encoding dimulai dengan pesan yang akan disisipkan. Pesan diubah kedalam bentuk biner dengan representasi 1 atau 0. Selanjutnya rangkaian biner tersebut dikonversikan menjadi bilangan desimal dan menghasilkan sebuah bilangan yang dinamakan dengan *m.*

Algoritma yang dilakukan dalam data teks adalah proses perubahan kalimat ke dalam bentuk ASCII. Sebagai contoh pesan yang akan disisipkan adalah "#aku". Maka kode ASCII dari pesan tersebut adalah :35 97 107 117

b. Encode Data Gambar

Proses Encode gambar adalah sebagai berikut :

1. Menghitung jumlah warna yang terdapat pada berkas RGB yang menjadi objek steganografi dan akan menghasilkan sebuah bilangan. Bilangan tersebut dinamakan dengan *n*, maka apabila *m>n!-1* maka pesan yang akan disisipkan berukuran terlalu besar sehingga proses penyisipan tidak dapat dilakukan.
2. Warna dalam palet warna diurutkan sesuai dengan urutan yang "natural". Setiap warna dengan *format* RGB dikonversikan kedalam bilangan integer dengan aturan (Merah * 65536 + Hijau * 256 + Biru). Kemudian diurutkan berdasarkan besar bilangan integer yang mewakili warna tersebut.
3. Setelah itu proses iterasi terhadap variabel *i* dengan nilai *i* adalah dari 1 sampai n. Setiap warna dengan urutan *n-i* dipindahkan ke posisi baru yaitu *m mod i*, kemudian m dibagi dengan *i*.
4. Kemudian palet warna yang baru hasil iterasi pada langkah ke-4 dimasukkan ke dalam palet warna berkas RGB.
5. Apabila ternyata besar dari palet warna yang baru lebih kecil dari 256 maka palet warna akan diisi dengan warna terakhir dari palet warna sebelumnya. Kemudian berkas RGB akan dikompresi ulang dengan palet warna baru, untuk menghasilkan berkas yang baru dengan ukuran dan gambar yang sama, namun telah disisipi pesan.

c. Kode Warna

Dalam sebuah gambar digital sering melihat kode warna yang terdiri dari tanda '#' dan 6 angka atau huruf di belakang tanda tersebut, kode warna tersebut dapat diterjemahkan menjadi kode warna RGB. Misalkan kita mempunyai kode warna "#0088FF", kita akan mencari beberapa porsi untuk warna merah (red), hijau (green), dan biru (blue).

d. Proses Decode

Adapun langkah-langkah proses *decode* atau mengekstrak pesan dari citra RGB yang telah disisipi pesan dengan metode *End Of File* adalah sebagai berikut:

1. Masukkan nomor sesuai dengan posisi setiap warna pada palet warna citra RGB yang telah disisipkan pesan
2. Warna diurutkan berdasarkan konversi RGB ke nilai integer dengan rumus:
   (Merah * 65536 + Hijau * 256 + Biru).
3. *m* diberi nilai 0
4. Iterasi variabel *i* dari *i+1* sampai *n-1*.
   a) m=m*(n-1) + posisi warna ke *i*
   b) iterasi variabel *j* dari *i* +1 sampai n-1
   c) jika posisi warna ke *j* > nilai posisi warna ke-i, maka posisi warna ke *i* dikurangkan 1
5. Setelah nilai *m* diperoleh, maka nilai *m* dikonversikan ke bilangan biner untuk memperoleh pesan asli kembali

e. Proses End Of File

Contoh penyisipan pesan "T" kedalam berkas RGB dengan jumlah warna pada palet warna sebanyak 6 buah, adalah sebagai berikut:

1. Pesan yang disisipkan adalah "T" yang diubah ke bentuk binary dengan pengkodean ASCII menghasilkan bilangan biner : 01010100. Untuk mendapatkan nilai M disisipkan angka 1 pada rangkaian biner maka: m = 1 + 01010100 2 = 101010100 2 = 340 10
2. Jumlah warna pada palet warna citra tersebut adalah 6, maka apabila 340 >6!-1 cara menghitungnya yaitu sebagai berikut:
   $_C$("T") = ("T" + "U") mod 256 = L
   T = 51 dan U = 52 (51+52) % 96 = 7
   $_C$ ("E") = ("E" + "K") mod 96 = p
3. Urutan warna pada palet warna citra tersebut secara "natural"ditunjukkan dari beberapa warna yang didapat dari besar nilai RGB. Contoh pada warna A, nilai *Red* dalam hexadecimal adalah f7, dan dikonversikan ke dalam desimal menjadi 247, nilai Green pada hexadecimal adalah 47 dan nilai Blue dalam hexadecimal adalah 47 dan dikonversikan kedalam desimal menjadi 71. Berdasarkan nilai-nilai tersebut didapat nilai "natural" dengan rumus sebagai berikut:
   (Red 65536 + Green 256 + Blue) sehingga didapat nilai integer yaitu: 16205639.
4. Iterasi variabel i mulai dari 1 sampai n:
   Warna indeks ke- (n-1) dipindahkan ke- (*m mod i), m* = 1)
   a) Untuk *i* = 1 *m* = 340, *m* =340/1 = 340 maka warna indeks ke-5 dipindahkan ke indeks ke-0 pada susunan palet warna yang baru.

e-ISSN: 2549-7952
p-ISSN: 2580-3336

b) Untuk $i = 2$ $m = 340$, $m = 340/2 = 170$ maka warna indeks ke-4 dipindahkan ke indeks ke-0 pada susunan palet warna yang baru.
c) Untuk $i = 3$ $m = 170$, $m = 170/3 = 56$ maka warna indeks ke-3 dipindahkan ke indeks ke-2 pada susunan palet warna yang baru.
d) Untuk $i = 4$ $m = 56$, $m = 56/4 = 14$, maka warna indeks ke-2 dipindahkan ke indeks ke-0 pada susunan palet warna yang baru.
e) Untuk $i = 5$ $m = 14$, $m = 14/5 = 2$ maka warna indeks ke-1 dipindahkan ke indeks ke-3 pada susunan palet warna yang baru.
f) Untuk $i = 6$ $m = 2$, $m = 2/6 = 0$ maka warna indeks ke-0 dipindahkan ke indeks ke-1 pada susunan palet warna yang baru.

5. Pada tahap berikutnya, apabila ada beberapa warna indeks yang menempati indeks yang sama, maka setiap warna yang menempati indeks tersebut akan bergeser sekali ke indeks berikutnya.

6. Urutan palet warna ini kemudian dimasukkan kedalam berkas citra RGB untuk menghasilkan citra yang telah disisipi pesan.

3. HASIL DAN PEMBAHASAN

Sistem keamanan data pesan dengan Steganografi End of File, terdapat 3 komponen penting menu utama yang mendasari.
Adalah sebagai berikut :

1. Menu Enkripsi
   Merupakan interface yang memungkinkan user untuk membuat (*button* tambah), analisa dan hapus pesan rahasia yang akan dibuat maupun yang telah dibuat.
   a) *Button* Tambah dalam form Enkripsi, Tombol menu yang digunakan user untuk membuat baru sebuah pesan rahasia
   b) *Button* Analisa dalam form Enkripsi, Tombol Menu yang digunakan user untuk memastikan atau mengecek pesan yang dibuat telah berhasil disisipkan di media gambar atau belum
   c) *Button* Hapus dalam form Enkripsi, Tombol menu yang digunakan user untuk menghapus pesan rahasia beserta media gambarnya.
   d) *Button* Pencarian, Memungkinkan user untuk mencari file file enkripsi yang diinginkan

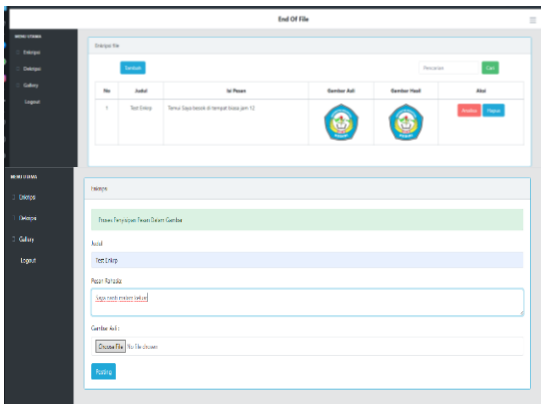
2. Menu Dekripsi
   Interface yang memungkinkan user untuk membuka (decrypt), analisa dan hapus pesan rahasia yang akan telah di enkripsi sebelumnya.
   a) *Button* Tambah dalam Menu Dekripsi, Menu yang digunakan user untuk memilih lalu membuka pesan rahasia dalam gambar
   b) *Button* Analisa dalam form Dekripsi, Menu yang digunakan user untuk memastikan dan membuktikan bahwa pesan rahasia sudah bisa di lepas dari media gambar.
   c) *Button* Hapus dalam form Dekripsi, Menu yang digunakan user untuk menghapus pesan rahasia beserta media gambar yang telah dibuka.

3. Menu Gallery
   Interface yang menyediakan tampilan pesan berupa gambar baik enkripsi maupun deskripsi beserta tanggal penyimpanan. Dan memungkinkan user untuk melihat langsung pesan rahasia yang ada di dalamnya dengan mengklik *button* buka pesan.
   a) *Button* Buka Pesan, Menu yang digunakan user untuk membuka secara langsung isi pesan rahasia yang ada di dalam gambar.
   b) *Button* Buka Pesan, Memungkinkan user untuk mencari file file enkripsi dan dekripsi yang tersimpan di gallery.

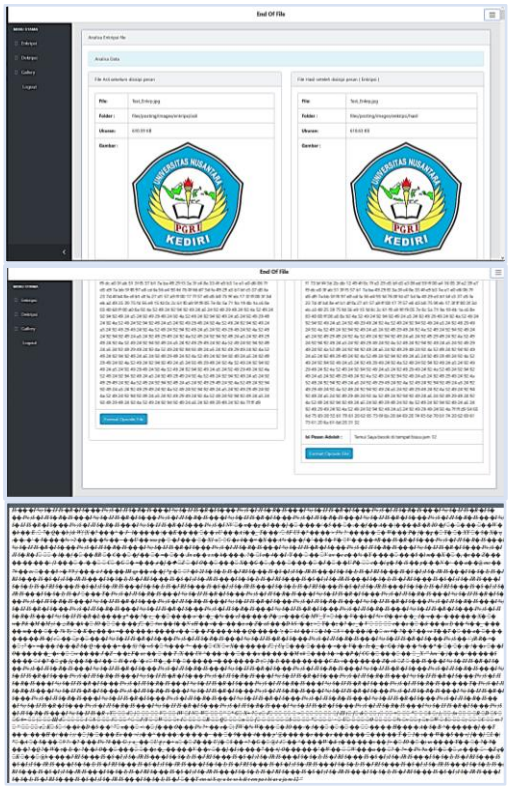
Pada sistem terdapat dua form/ layar kerja utama, pertama adalah hasil enkripsi berupa file gambar yang sudah tersisipi pesan teks rahasia di dalamnya, dan yang kedua adalah hasil deskripsi berupa pesan rahasia (awal) berbentuk teks dengan format ASCII dan citra gambar (awal) berformat JPEG maupun PNG. Secara rinci layar kerja yang ada dalam sistem adalah:

1. Form Enkripsi. Layar kerja ini digunakan untuk menambah maupun membuat pesan teks yang ingin dirahasiakan keberadaannya melalui media citra gambar sebagai penampungnya.
2. Form Analisa Enkripsi File. Form ini digunakan untuk melakukan analisa atau cek pesan teks yang telah disisipkan pada gambar apakah sudah berhasil atau belum. Tanda proses enkripsi berhasil adalah *size* gambar yang bertambah besar, jika format hexadecimal pada akhiran media gambar terlihat bertambah banyak di akhiran. Pengamatan lebih lanjut juga bisa dianalisa pada akhiran simbol opcode program.
3. Form hapus file terenkripsi dan dekripsi. Menu yang digunakan user untuk menghapus pesan rahasia beserta media gambar di dalamnya baik file enkripsi ataupun file dekripsi.
4. Form Tambah Dekripsi File. Form ini digunakan untuk membuka pesan teks rahasia yang telah terenkripsi sebelumnya dalam media gambar. Jadi dekripsi ini adalah kebalikan dari enkripsi.
5. Form Analisa File Dekripsi. Form ini digunakan oleh user untuk menganalisa file enkripsi yang sudah bisa dibuka atau dibalikkan dengan deskripsi. Tanda jika berhasilnya proses Dekripsi adalah ukuran atau *size* gambar yang mengecil dan juga jika format hexadecimal pada akhiran media gambar terlihat berkurang di akhir hexadecimal.
6. Form Gallery. Layar kerja yang menyediakan tampilan pesan berupa gambar baik enkripsi maupun deskripsi beserta tanggal penyimpanan. Dan memungkinkan user untuk melihat langsung pesan rahasia yang ada di dalamnya dengan mengklik *button* buka pesan.

e-ISSN: 2549-7952
p-ISSN: 2580-3336

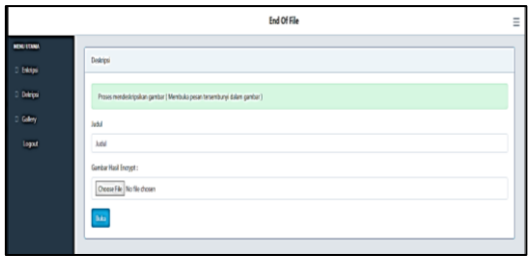


Gambar 12. Menu Enkripsi dengan Submenu Tambah Proses Enkripsi

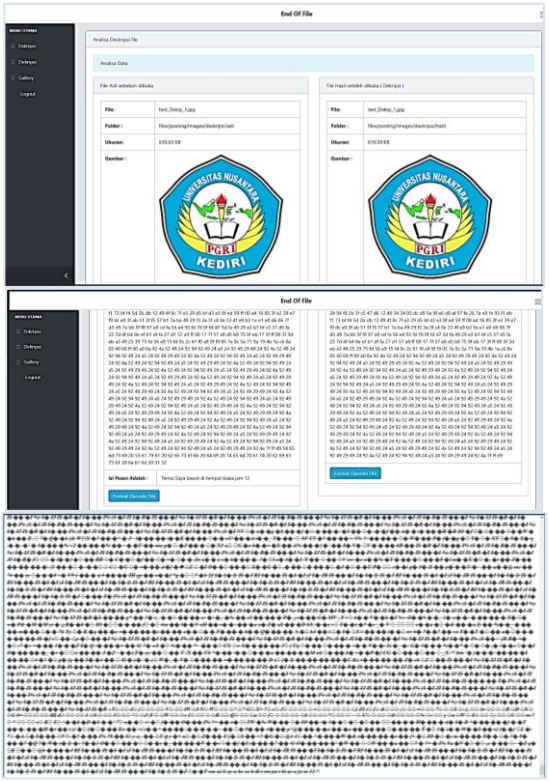


Gambar 13. Analisa Enkripsi Menggunakan *Size* Gambar, Hexadesimal dan Opcode.

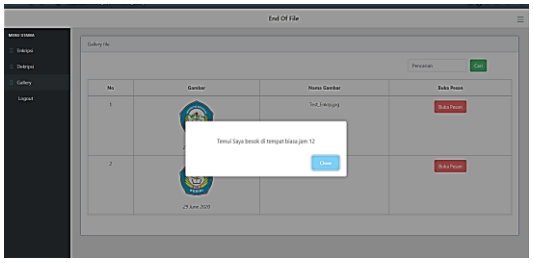
Gambar 14. Menu Hapus File Enkripsi Dekripsi



Gambar 15. Tambah File Dekripsi



Gambar 16. Analisa Dekripsi Menggunakan Size Gambar, Hexadecimal, dan Opcode



Gambar 17. Gallery Enkripsi dan Dekripsi

4. KESIMPULAN

Berdasarkan rangkaian kegiatan yang dilakukan dalam penelitian ini, mulai dari perencanaan, analisis permasalahan, perancangan, development sistem, uji coba dan analisa hasil, maka diperoleh beberapa kesimpulan sebagai berikut:

1. Penelitian ini berhasil membuat suatu sistem keamanan data pesan dalam gambar dengan menerapkan steganografi dengan metode *End Of File*, dimana sebuah pesan tek akan disisipkan ke dalam media penampung berupa gambar. Teks

e-ISSN: 2549-7952
p-ISSN: 2580-3336

dan gambar harus sama sama di jadikan format hexadecimal terlebih dahulu agar bisa disisipkan di akhir penanda citra.
2. Sistem keamanan data yang diterapkan adalah dengan cara melakukan enkripsi teks ASCII dan citra gambar. Lalu menggabungkan keduanya dalam suatu media dengan menggunakan metode end of file. Dengan konsep seperti ini, maka pesan yang dibuat bisa terjaga kerahasiaannya di dalam file.
3. Ukuran *pixel* citra dan jumlah karakter yang digunakan berpengaruh terhadap lamanya waktu proses *embedding* dan *extracting*. Semakin besar ukuran *pixel* citra dan jumlah karakter, maka akan semakin lama proses penyisipan dan ekstraksinya.

5. SARAN

Dalam pelaksanaan penelitian ini, peneliti menemukan beberapa pemikiran yang akhirnya tertuang dalam saran dan harapan yang dititipkan untuk peneliti selanjutnya, yaitu:

1. Aplikasi dapat dikembangkan lebih lanjut dengan menambahkan format gambar lain seperti *bitmap* (bmp) dan format penampung lain, seperti teks, *audio*, dan *video*.
2. Untuk Pengembangan lebih lanjut diharapkan dapat menambahkan sistem keamanan yang lebih baik pada perangkat lunak dalam meningkatkan keamanan sistem data yang seringkali dianggap sepele.

DAFTAR PUSTAKA

[1] Arifin, R., dan Oktaviana, L. T. 2013, *Implementasi Kriptografi dan Steganografi Menggunakan Algoritma RSA dan Metode LSB*, Universitas Malang.
[2] Ariyus, Dony. 2008, *Pengantar Ilmu Kriptografi.* Yogyakarta: CV. ANDI OFFSET.
[3] K. P. Adhiya dan S. A. Patil, “Hiding Text in Audio Using LSB Based Steganography, ”Information and Knowledge Management, vol. 2, no. 3, 2012.
[4] Rifki Sadikin, 2012, *Kriptografi Untuk Keamanan dan Implementasi Dalam Bahasa Java*. Yogyakarta: Andi Offset.
[5] Saputra, Joko. 2017, *Steganografi penyisipan teks pada citra menggunakan end of file,* Institut Teknologi Sepuluh November Surabaya.
[6] Sembiring, Sandro. 2013, *Perancangan Aplikasi Steganografi untuk menyisipkan pesan teks pada gambar dengan metode End Of File* STMIK Budi Darma Medan
[7] Munir Rinaldi, 2004, ”*Diktat Kuliah IF5054 Kriptografi : Steganografi dan Watermarking*”, Teknik Informatika ITB, Bandung, diakses dari http://informatika. stei. itb. ac. id/~rinaldi. munir/ Kriptografi/Steganografi dan Watermarking. pdf, akses 17 September 2019, pukul 13. 14 WIB.