STEGANOGRAFI VIDEO MENGGUNAKAN METODE *END OF FILE* (EOF)

[1]Septya Maharani, [2]Ismiatul Maula, [3]Zainal Arifin
[1,2,3]Program Studi Ilmu Komputer, Fakultas Ilmu Komputer dan Teknologi Informasi
Universitas Mulawarman, Kalimantan Timur
Email: [1]septyamaharani@gmail.com, [2]smiatulmaula@gmail.com, [3]zainal.ilkom.unmul@gmail.com

Abstrak. *Steganografi adalah ilmu untuk menyembunyikan pesan atau informasi dalam suatu media, seperti teks, gambar, audio ataupun video yang bertujuan untuk menghindari kecurigaan dari orang yang tidak berhak. Untuk itu diperlukan sebuah perangkat lunak yang dapat menyembunyikan informasi yang bersifat rahasia pada sebuah media yaitu video.Metode End of file (EOF) merupakan salah satu teknik yang dapat digunakan untuk menyembunyikan informasi atau pesan rahasia yang dikirimkan pada orang lain. Penerapan metode EOF pada video dilakukan dengan menyisipkan pesan pada akhir file dengan menambahkan kunci pada proses encoding dan decoding.Hal ini bertujuan untuk mengamankan informasi rahasia yang telah disembunyikan pada video agar dapat terjaga dan hanya dapat dibuka oleh pengguna yang memiliki kunci dan aplikasi steganografi tersebut.*

Kata Kunci: Steganografi, Video, Encoding, Decoding, EOF

Kerahasiaan sebuah informasi berupa pesan yang dimiliki oleh pengguna merupakan hal penting dalam pertukaran pesan agar pesan tersebut hanya dapat diberikan kepada penggunatertentu yang memiliki hak untuk mengakses pesan tersebut. Saat pengguna akan melakukan pengiriman data, maka data tersebut tidak dapat dijamin keamanannya. Seiring dengan berkembangnya teknologi saat ini, ada banyak cara yang digunakan untuk menyadap sebuah informasi penting. Pihak lain yang tidak bertanggung jawab dapat melihat, membaca ataupun mengubah konten asli dari informasi, sehingga di perlukan sebuah cara untuk memastikan kerahasiaan informasi yang akan dikirim sampai kepada pihak yang seharusnya menerima informasi tersebut. Salah satu cara yang dapat digunakan untuk menyembunyikan informasi rahasia dari penyadapan adalah steganografi.

Steganografi adalah ilmu untuk menyembunyikan pesan atau informasi dalam suatu media, seperti teks, gambar, audio ataupun video yang bertujuan untuk menghindari kecurigaan dari orang yang tidak berhak. Steganografi selalu memiliki dua proses, yaitu*encoding* dan *decoding*. *Encoding* merupakan proses penyisipan pesan kedalam sebuah media penampung (*cover*) sedangkan *decoding* adalah proses ekstraksi pesan dari media penampung (*cover*) [1].

Salah satu metode *steganografi* yang dapat digunakan untuk menyembunyikan teks pesan rahasia kedalam sebuah *file* video digital adalah metode *End of file* (EOF). EOF merupakan teknik penyembunyian pesan rahasia pada akhir *file* menggunakan kunci yang sama.

Berdasarkan uraian tersebut, mendorong peneliti untuk menerapkan metode EOF dalam perangkat lunak (*software*) yang dapat membantu proses penyembunyian pesan rahasia pada video. Dalam aplikasi ini video akan di pecah menjadi *frame-frame* gambar, kemudian pada satu *frame* akan disisipkan pesan teks. *Frame* yang telah disisipkan pesan akan digabung kembali menjadi sebuah *file* video. Video digital dipilih sebagai *cover* stego karena video saat ini banyak digunakan sebagai sarana pertukaran informasi yang lebih menarik daripada citra digital.

Batasan Masalah

Berdasarkan latar belakang yang telah diuraikan, agar tidak menyimpang dari tujuan, penulis memberikan batasan-batasan masalah yaitu :

1. Format video yang digunakan adalah *.mp4.
2. Pesan teks maksimal 200 karakter.
3. Durasi video maksimal 60 detik.
4. Metode yang digunakan yaitu EOF.
5. *Size* video ketika disisipkan pesan akan bertambah.

Tujuan Penelitian

Tujuan dari penelitian ini adalah membangun perangkat lunak (*software)* yang

dapat menyembunyikan pesan untuk penyamaran informasi data menggunakan metode EOF.

Manfaat Penelitian

Berdasarkan latar belakang dan rumusan masalah yang dibahas, maka manfaat dari penelitian ini adalah:

1. Perangkat lunak (*software*) dapat digunakan untuk menyembunyikan pesan dalam sebuah video sehingga informasi yang bersifat rahasia.
2. Sebagai referensi untuk penelitian lain menggunakan EOF.

Steganografi

Steganografi adalah ilmu dan seni menyembunyikan pesan rahasia (*hiding image*) sedemikian sehingga keberadaan pesan tidak terdeteksi oleh indra manusia. Kata *steganografi* berasal dari bahasa yunani yang berarti "tulisan tersembunyi" (*covered writing*). *Steganografi* membutuhkan dua properti yaitu wadah penampung dan data rahasia yang akan disembunyika. Steganografi digital menggunakan media digital sebagai wadah penampung, misalnya, citra, suara, teks, dan video. Data rahasia yang disembunyikan juga dapat berupa citra, suara, teks, atau video [2].

Dalam bidang keamanan komputer, *steganography* digunakan untuk menyembunyikan data rahasia, saat enkripsi tidak dapat dilakukan atau bersamaan dengan enkripsi. Walaupun enkripsi berhasil dipecahkan (*decipher*), pesan atau data rahasia tetap tidak terlihat. Pada *cryptography*, pesan disembunyikan dengan "diacak" sehingga pada kasus-kasus tertentu dapat dengan mudah mengundang kecurigaan, sedangkan pada *steganografi* pesan "disamarkan" dalam bentuk yang relatif lebih "aman" sehingga tidak terjadi kecurigaan itu [3]

Dalam proses steganografi terdapat beberapa kriteria yang harus dipenuhi, kriterianya adalah sebagai berikut [2]:

1. *Imperceptibility*. Keberadaan pesan *rahasia* tidak dapat dipersepsi oleh inderawi. Misalnya, jika *covertext* berupa citra, maka penyisipan pesan membuat citra *stegotext* sukar dibedakan oleh mata dengan citra *covertext*-nya. Jika *covertext* berupa audio, maka indera telinga tidak dapat mendeteksi perubahan pada audio *stegotext*-nya.
2. *Fidelity*. Mutu *stegomedium* tidak berubah banyak akibat penyisipan. Perubahan tersebut tidak dapat dipersepsi oleh inderawi. Misalnya, jika *covertext* berupa citra, maka penyisipan pesan membuat citra *stegotext* sukar dibedakan oleh mata dengan citra *covertext*-nya. Jika *covertext* berupa audio, maka audio stegotext tidak rusak dan indera telinga tidak dapat mendeteksi perubahan tersebut.
3. *Recovery*. Pesan yang disembunyikan harus dapat diungkapkan kembali. Karena tujuan *Steganografi* adalah *data hiding*, maka sewaktu-waktu pesan rahasia di dalam *stegotext* harus dapat diambil kembali untuk digunakan lebih lanjut.

Metode *End of file* (EOF)

Metode EOF merupakan salah satu metode yang digunakan dalam steganografi. Metode ini menggunakan cara dengan menyisipkan data pada akhir *file*. Sehingga, tidak akan mengganggu kualitas data awal yang akan disisipkan pesan. Namun, ukuran *file* setelah disisipkan pesan rahasia akan bertambah. Sebab, ukuran *file* yang telah disisipkan pesan rahasia sama dengan ukuran *file* sebelum disisipkan pesan rahasia yang disisipkan. Untuk mengenai data yang disisipkan pada akhir *file*, diperlukan suatu tanda pengenal atau simbol pada awal dan akhir data yang akan disisipkan [4].

EOF menggunakan karakter yang berbeda sebagai penanda awal penyisipan pesan dan penanda akhir penyisipan pesan. Metode EOF menggunakan kelemahan indera manusia yang tidak sensitif sehingga seakan-akan tidak ada perbedaan yang terlihat antara sebelum atau sesudah pesan disisipkan [5].

Adapun langkah-langkah encoding menggunakan metode *End Of File* [6] :

1. Proses encoding dimulai dengan pesan yang akan disisipkan. Pesan diubah kedalam bentuk biner denganrepresentasi 1 atau 0.
2. Kemudian disisipkan angka 1 didepan rangkaian biner tersebut.langkah selanjutnya rangkaian biner tersebut dikonversikan menjadi bilangan decimal dan meghasilkan sebuah bilangan yang dinamakan dengan *m.*
3. Menghitung jumlah warna yang terdapat pada berkas RGB yang menjadi objek steganografi dan akan menghasilkan sebuah

bilangan. Bilangan tersebut dinamakan dengan In, maka apabila *m>n!-1* maka pesan yang akan disisipkan berukuran terlalu besar sehingga proses penyisipan tidak dapat dilakukan.

4. Warna dalam palet warna diurutkan sesuai dengan urutan '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.
5. Setelah itu lakukan proses iterasi terhadap variable *i* adalah 1 sampai n. Setiap warna dengan urutan *n – i* dipindahkan ke posisi baru yaitu m mod i, kemudian *m* dibagi dengan *i*.
6. Kemudian palet warna yang baru hasil iterasi pada langkah ke-4 dimasukkan kedalam palet warna berkas RGB. Apabila ada tempat yang diisi oleh dua buah warna, maka warna sebelumnya yang menempati tempat tersebut akan digeser satu tempat ke samping.
7. Apabila ternyata besar dari palet warna yang baru lebih kecil dari 256 maka palet warna akan diisi dengan warna terakhir dari palet warna sebelumnya.
8. Kemudian berkas RGB akan dikompresi ulang dengan palet warna yang baru, untuk menghasilkan berkas yang baru.

Sebagai contoh pada sebuah citra grayscale 6x6 piksel disisipkan pesan yang berbunyi "aku". Untuk menandai akhir pesan digunakan karakter yang jarang dipakai, misalnya karakter #. Sehingga pesan yang dimaksud adalah "#aku". Kode ASCII dari pesan diberikan sebagai berikut [4]:

97 107 117 35

Misalkan matrik tingkat derajat keabuan citra :

96	0	7	82	01	0
7	00	00	0	0	0
5	50	5	00	5	8
76	6	7	00	5	00
01	4	50	0	00	0
4	6	9	25	90	00

Kode biner pesan disisipkan diakhir citra sehingga citra menjadi:

96	0	7	82	01	0
7	00	00	0	0	0
5	50	5	00	5	8
76	6	7	00	5	00
01	4	50	0	00	0
4	6	9	25	90	00
7	**07**	**17**	**5**		

Video digital

Video digital adalah berkas komputer yang digunakan untuk menyimpan kumpulan berkas *digital* seperti *video*, *audio*, metadata, informasi, pembagian *chapter*, dan judul sekaligus, yang dapat dimainkan atau digunakan melalui perangkat lunak tertentu pada komputer [7]

Video digital sebenarnya terdiri atas serangkaian gambar digital yang ditampilkan dengan cepat pada kecepatan yang konstan. Dalam konteks video, gambar ini disebut *frame*. Satuan ukuran untuk menghitung *frame* rata-rata yang ditampilkan disebut *frameper second* (FPS). Setiap *frame* merupakan gambar digital yang terdiri dari dari raster *pixel*. Gambar digital akan memeiliki lebar sebanyak W *pixel* dan tinggi sebanyak H *pixel*. Oleh karena itu, dapat dikatakan bahwa *frame size* adalah Weight x Height [7].

I. Metodologi

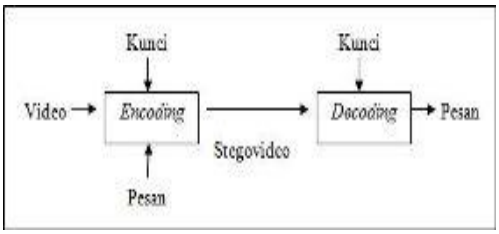
Deskripsi Sistem

Sistem steganografi pada video menggunakan metode *end of file* merupakan salah satu teknik pengamanan data dari pihak yang tidak berwenang. Data yang diamankan berupa data teks yang disebut sebagai pesan rahasia.

Terdapat dua proses utama dalam perangkat lunak ini yaitu proses penyisipan pesan rahasia pada sebuah media penampung yang disebut *Encoding* dan proses pengambilan pesan pada media penampung disebut *decoding*. Setelah melalui proses *Encoding* maka pesan rahasia dapat dikirim ke pengguna

lain yang memiliki kunci. Kemudian pesan rahasia akan melalui proses *decoding* untuk agar dapat dibaca oleh pengguna lain yang memiliki kunci.

Tahapan proses dari sistem ditunjukkan pada gambar 1



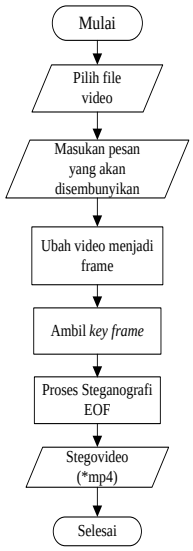
Gambar 1. Sistem Steganografi Pada Video

Steganografi pada video melakukan proses *encoding* dan *decoding* pada *frame* yang menyusun video. Sebelum melakukan proses *encoding* dan *decoding*, video akan dipecah menjadi audio dan *frame-frame.* kemudian diambil sebuah*frame* pertama untuk dilakukan proses *encoding* dan *decoding*. Setelah proses *encoding* maka audio dan semua *frame* akan digabungkan menjadi video kembali.

Perancangan Sistem

Tahapan ini merupakan tahapan perancangan sistem yang bertujuan memberikan gambaran jelas dan rancang bangun sistem yang akan dibuat. Tahapan desain dapat dilihat pada *flowchart* yaitu :

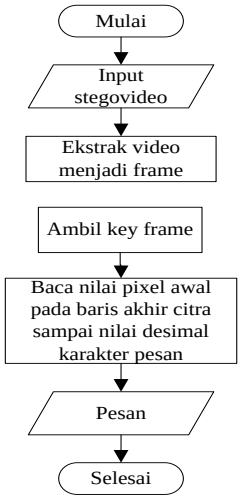
1. Proses *Encoding* : penyisipan pesan



Gambar 2. *Flowchart* Proses *Encoding* Pesan

Flowchart pada gambar 2 menjelaskan dimana pengguna dapat memasukkan data berupa video dan pesan rahasia. Video yang dipilih kemudian diekstrak menjadi frame-frame. Selanjutnya data yang masuk pada sistem akan dilakukan proses steganografi video yaitu *encoding* dan *decoding* menggunakan metode EOF yang hasil keluarannya berupa *stegovideo* yang dapat disimpan oleh pengguna.

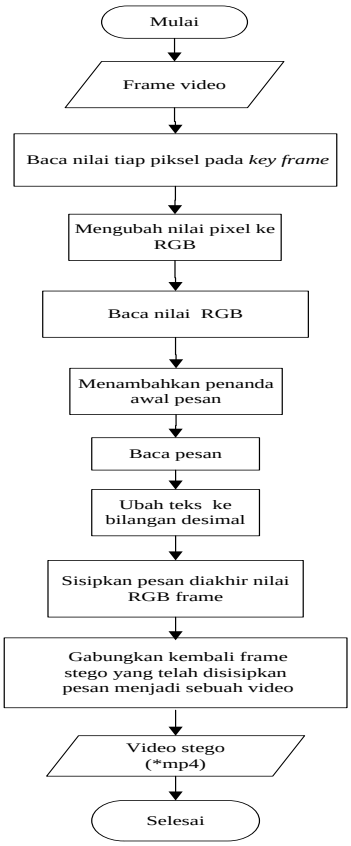
2. Proses *decoding* : pengambilan pesan



Gambar 3. *Flowchart* Proses *Decoding*Pesan

Pada gambar 3 menjelaskan dimana pengguna melakukan proses *decoding* pesan atau pengembalian pesan asli dengan memasukkan stegovideo dan mengekstaknya menjadi frame-frame. Kemudian ambil frame dan baca nilai pixel awal pada baris akhir citra sampai nilai desimal karakter pesan. setelah proses tersebut maka pesan yang disisipkan akan terbaca.

3. Proses steganografi EOF



Gambar 4. *Flowchart* Proses Steganografi *End of file*

Flowchart pada gambar 4 menjelaskan sitem steganografi pada video menggunakan metode EOF . Dari gambar 4 proses steganografi data masukan yang pertama adalah *frame* video yang akan dilakukan proses pembacaan nilai tiap *pixel* pada *frame*. Selanjutnya nilai *pixel* diubah menjadi nilai *Red Green Blue* (RGB). Setelah semua nilai *pixel* diubah menjadi nilai RGB, kemudian ditambahkan penanda diakhir nilai tersebut sebagai penanda awal pesan yang akan sisipkan. Kemudian sistem melakukan pembacaan pesan yang dimasukkan oleh pengguna dengan mengubah pesan menjadi ASCII. Pesan yang telah diubah menjadi ASCII ditambahkan setelah penanda akhir nilai *pixel*. Kemudian sistem akan memproses menjadi frame baru dan menggabungkan kembali *frame-frame* menjadi video stego.

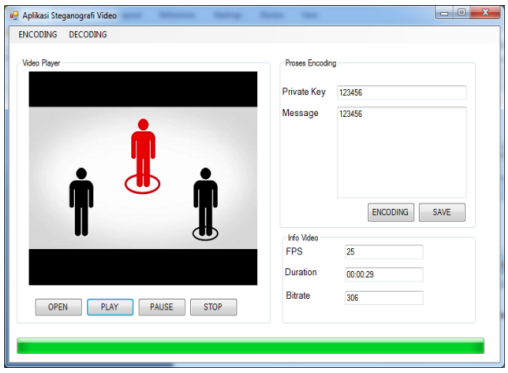
II. Hasil dan Pembahasan

Implementasi Sistem

Desain sistem aplikasi perangkat lunak steganografi video menggunakan metode *end of file* diimplementasikan menggunakan bahasa pemrograman vb dan mengaplikasikan teknologi .Net.

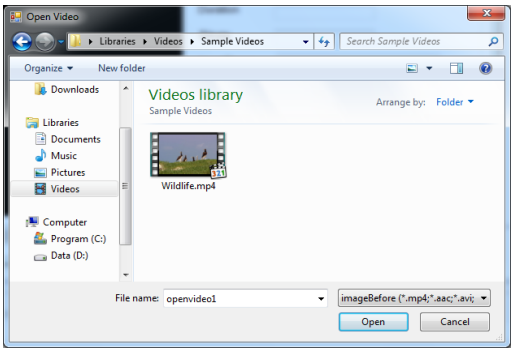
a. Implementasi *Encoding*

Desain awal antarmuka perangkat lunak dimulai dengan form menu *encoding* seperti gambar 5



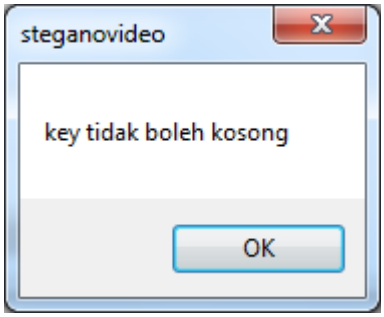
Gambar 5. Tampilan Awal Menu *Encoding*

Langkah pertama dalam melakukan proses *encoding* yaitu dengan menekan tombol *browse* untuk menampilkan *openfile dialog*, setelah itu *user* diminta untuk memilih video yang akan ditampilkan kedalam media player pada perangkat lunak. *Openfile dialog* hanya menampilkan *file* video berekstensi *.mp4. seperti gambar 6



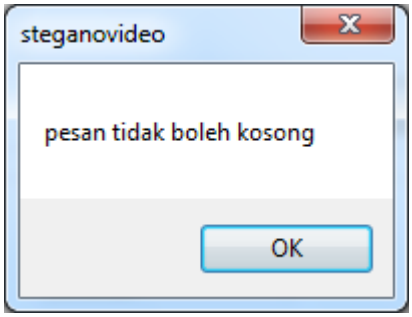
Gambar 6. Tampilan *Open dialog*

Selanjutnya *user* menginput *private key* pada *textbox* untuk melakukan penguncian hasil penyisipan agar tidak dapat dibaca pengguna lain. Apabila kunci tidak diinputkan maka perangkat lunak akan menampilkan *message box* bahwa user harus menginputkan kunci. Seperti gambar 7



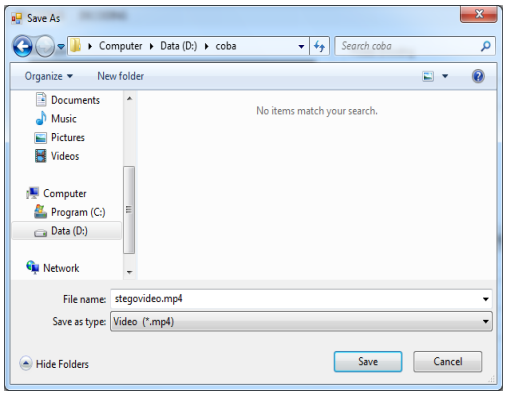
Gambar 7. *Message Box*

Setelah itu user diminta untuk mengisi pesan yang akan disembuyikan ke dalam *textbox* pesan. apabila pesan tidak diisi maka perangkat lunak akan menampilkan message box bahwa *user* harus mengisi pesan pada *textbox* pesan seperti gambar 8



Gambar 8. *Message Box*

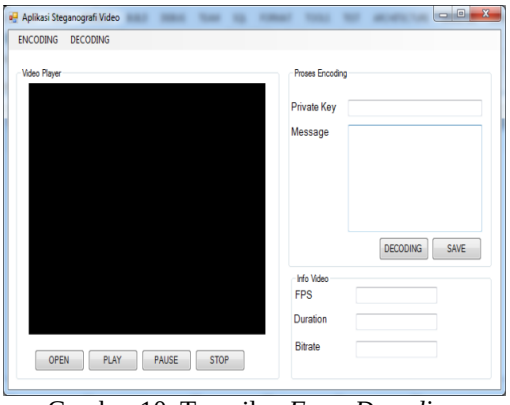
Setelah melakukan input data yang dibutuhkan oleh perangkat lunak, maka perangkat lunak akan melakukan proses penyisipan pesan pada video. Progress proses penyisipan pesan ini dapat lihat pada progress bar yang ada pda perangkta lunak. Ketika proses selesai maka user dapat menyimpan video yang telah disisipkan pesan dengan menekan tombol save. Tombol save akan memunculkan save dialog seperti gambar 9



Gambar 9. Tampilan *Save Dialog*

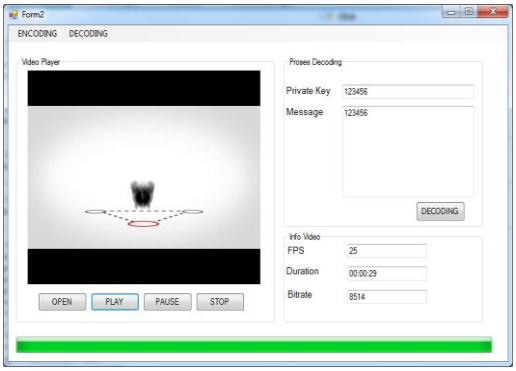
b. Implementasi *decoding*

Form selanjutnya adalah form *decoding*. Menu pada form *decoding* hampir sama dengan dengan form *encoding*, yang berbeda hanya pada tombol proses. Jika pada form *encoding* terdapat tombol *encoding* maka pada form *decoding* terdapat tombol *decoding*. Media *player* dan 5 *textbox* yang digunakan juga sama seperti pada form *encodig*. *Form decoding* dapat dilihat pada gambar 10



Gambar 10. Tampilan *Form Decoding*

Pada tombol *decoding* akan dilakukan proses pembacaan pesan yang telah disisipkan pada video. Setelah proses pembacaan maka pesan yang disisipkan pada video akan muncul pada *textbox message* seperti gambar 11



Gambar 11. Hasil Proses *Decoding*

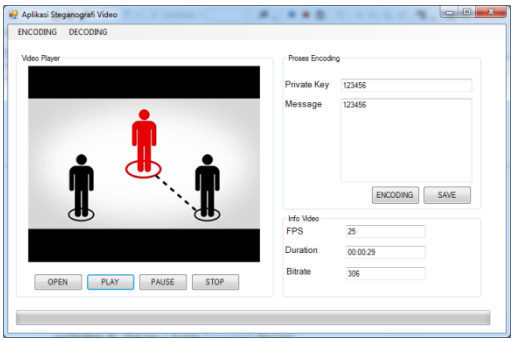
Pengujian Sistem

Pengujian sistem pada tahapan *encoding* dan *decoding* dilakukan untuk menguji tingkat keberhasilan perangkat lunak, apakah teks pada pesan rahasia dapat diencoding ke dalam video dan teks yang telah diencoding pada video dapat diambil kembali dengan tahapan *decoding*.

Pengujian video secara visual dilakukan dengan dua cara yaitu memutar file video dengan menggunakan media payer biasa dan

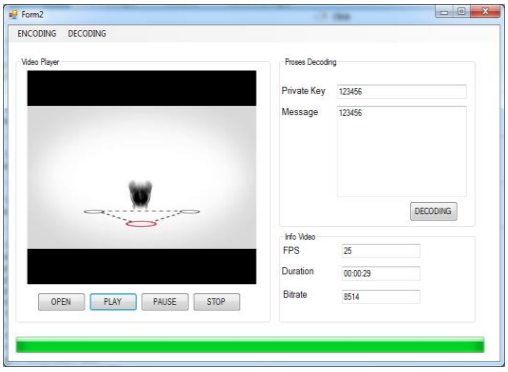
media player yang terdapat pada perangkat lunak. Media player yang digunakan untuk melakukan pengujian adalah media *player classic*. Dari hasil pengujian terlihat file video dapat diputar dengan baik tanpa ada gangguan dan pesan yang disisipkan tidak terlihat.

Pada gambar 12 dapat dilihat bahwa pada video sebelum disisipkan pesan memiliki informasi jumlah fps sebanyak 25 fps, dengan durasi video selama 29 detik dan jumlah bitrate 306 kb/s



Gambar 12. video sebelum diencoding

Sedangkan pada gambar 13 informasi video yang telah disisipkan pesan mengalami perubahan bitrate menjadi 8514 kb/s.



Gambar 13. Hasil Pengujian Perangkat Lunak

III. Simpulan

Kesimpulan yang dapat diambil berdasarkan dari penelitian steganografi video menggunakan metode *end of file* adalah:

1. Penelitian meghasilkan aplikasi steganografi video menggunakan metode *end of file*
2. Pengguna dari aplikasi ini dapat melakukan pengamanan informasi rahasia dengan menyisipkannya pada video.
3. Berdasarkan penelitian diketahui bahwa lama proses untuk melakukan proses steganografi menggunakan metode EOF ditentukan oleh jumlah frame yang menyusun video.

Berdasarkan hasil penelitian, sangat disadari oleh penulis bahwa masih banyak kekurangan dan kelemahan. Saran yang dapat diberikan adalah :

1. Stego video yang dihasilkan mengalami perubahan pada ukuran *file*, sehingga diperlukan metode lain untuk mengatasi perubahan ukuran tersebut.
2. Perangkat lunak dapat dikembangkan menggunakan metode lain dan dapat dijadikan model untuk pengembangan perangkat lunak yang lebih baik lagi.

IV. Daftar Pustaka

[1] Agutaviana, ilmia. 2012. Aplikasi pesan rahasia berbasis web menggunakan vigenere chiper dan steganografi *end of file* . Skripsi. Universitas mulawarman.

[2] Munir, R. 2006. *Kriptografi*. Bandung : Penerbit Informatika.

[3] Sutoyo, T dan kawan-kawan. 2009. *Teori Pengolahan Citra Digital*. Yogyakarta: Penerbit Andi.

[4] Hariady, M. Mirsa. 2015. Keamanan Dan Penyisipan Pesan Rahasia Pada Gambar Dengan Enkripsi Blowfish Dan Steganografi *End of file* . Skripsi. Universitas Mulawarman .

[5] Edisuryana, M., Isnanto, R.R., Somantri, M.. 2013. *Aplikasi Steganografi Pada Citra Berformat Bitmap Dengan Menggunakan Metode End Of File*. Jurnal Teknik Elektro. Universitas Diponegoro Semarang.

[6] Sandro Sembiring. 2013. Perancangan aplikasi steganografi untuk menyisipkan pesan teks pada gambar dengan metode end of file. Jurnal pelita informatika budi luhur, volume : IV, nomor: 2. STMIK Budi Darma Medan.

[7] Ian Chandra K. 2003. *Utiliti Audio/Video*. Jakarta : PT. Elex Media Komputindo.

Halaman ini sengaja dikosongkan.