SKRIPSI

APLIKASI PENGAMANAN DATA DIGITAL DALAM MEDIA AVI MENGGUNAKAN TEKNIK *STEGANOGRAPHY*

Disusun Oleh

ACHMAD TORIQUL HUDA

07.12.577

JURUSAN TEKNIK ELEKTRO S-1

KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI NASIONAL MALANG

2012

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Московский государственный технический университет
имени Н.Э. Баумана»

А.Н. Ширяев

ВЕРОЯТНОСТЬ

Издание 4-е, переработанное и дополненное
Том 1

Москва
Издательство МГТУ им. Н.Э. Баумана
2014

LEMBAR PERSETUJUAN

APLIKASI PENGAMANAN DATA DIGITAL DALAM MEDIA AVI MENGGUNAKAN TEKNIK *STEGANOGRAPHY*

SKRIPSI

Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Teknik Komputer dan Informatika Strata Satu (S-1)

Disusun oleh :

ACHMAD TORIQUL HUDA

07.12.577

Mengetahui,

Ketua Jurusan Teknik Elektro S-1

Ir. Yusuf Ismail Nakhoda, MT

NIP.Y.1018800189

Diperiksa dan Disetujui

Dosen Pembimbing I	Dosen Pembimbing II
Joseph Dedy Irawan, ST, MT	**Sotyohadi, ST**
NIP. 19740416 200501 1 002	**NIP.Y.1039700309**

JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2012

SURAT PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini :

Nama : ACHMAD TORIQUL HUDA

NIM : 07.12.577

Program Studi : TEKNIK ELEKTRO

Konsentrasi : TEKNIK KOMPUTER DAN INFORMATIKA S-1

Dengan ini menyatakan bahwa Skripsi yang saya buat adalah hasil karya sendiri, tidak merupakan plagiasi dari karya orang lain. Dalam Skripsi ini tidak memuat karya orang lain, kecuali dicantumkan sumbernya sesuai dengan ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat, dan apabila di kemudian hari ada pelanggaran atas surat pernyataan ini, saya bersedia menerima sangsinya.

Malang, 12 Maret 2012

Yang membuat Pernyataan,

Achmad Toriqul Huda
07.12.577

SKRIPSI

APLIKASI PENGAMANAN DATA DIGITAL DALAM MEDIA AVI MENGGUNAKAN TEKNIK *STEGANOGRAPHY*

Disusun Oleh

ACHMAD TORIQUL HUDA

07.12.577

JURUSAN TEKNIK ELEKTRO S-1

KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

INSTITUT TEKNOLOGI NASIONAL MALANG

2012

ABSTRAK

APLIKASI PENGAMANAN DATA DIGITAL DALAM MEDIA AVI MENGGUNAKAN TEKNIK *STEGANOGRAPHY*

Achmad Toriqul Huda, NIM 07.12.577
Dosen Pembimbing : Joseph Dedy Irawan, ST, MT
Sotyohadi, ST

Steganography merupakan ilmu dan seni yang mempelajari cara penyembunyian informasi rahasia ke dalam suatu media sedemikian rupa sehingga manusia tidak dapat menyadari keberadaan pesan tersebut. Informasi rahasia disembunyikan dengan cara disisipkan pada suatu media seperti image, audio, dan video sehingga tidak terlihat bahwa dalam media tersebut disembunyikan suatu pesan informasi.

Metode Least Significant Bit (LSB) adalah suatu metode steganografi yang dapat menyisipkan suatu data ke dalam cover penyisipan seperti image, audio, dan video dengan cara mengubah bit-bit yang paling rendah dari suatu media penyisipan, sehingga file steganografi yang dihasilkan tidak akan kelihatan perubahannya meskipun telah disisipi suatu data.

Steganografi pada video dengan menggunakan metode LSB menyembunyikan bit-bit dari berkas rahasia pada bit-bit terendah dari frame video. Penyembunyian ini pada dasarnya memberikan pengaruh terhadap berkas cover frame video, tetapi karena perubahan yang terjadi sangat kecil sehingga tidak tertangkap oleh indera manusia.

Kata Kunci: Steganografi video, video, LSB, Least Significant Bit

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, atas karunia yang telah dilimpahkan-Nya sehingga penulis dapat menyelesaikan tugas akhir dengan judul **"APLIKASI PENGAMANAN DATA DIGITAL DALAM MEDIA AVI MENGGUNAKAN TEKNIK *STEGANOGRAPHY*"**.

Selanjutnya pada kesempatan ini penulis juga menyampaikan rasa terimakasih yang sebesar-besarnya kepada pihak – pihak yang telah banyak membantu penulis selama penyusunan tugas akhir, diantaranya :

1. Bapak Ir. Yusuf Ismail Nahkoda, MT selaku Ketua Jurusan Teknik Elektro S-1 ITN Malang.
2. Bapak Dr. Aryuanto Soetedjo, ST, MT selaku Sekertaris Jurusan Teknik Elektro S-1 ITN Malang.
3. Bapak Joseph Dedy Irawan, ST, MT selaku Dosen Pembimbing I
4. Bapak Sotyohadi, ST selaku Dosen Pembimbing II
5. Bapak Ahmad Faisol, ST selaku Dosen Wali.
6. Kedua orang tua dan keluarga besar yang telah memberikan do'a, motivasi, dukungan dan nasehat.
7. Seluruh dosen dan pegawai ITN Kampus 2 Malang.
8. Terima kasih kepada sahabat – sahabatku: Wendi, Pras, Vico, Risky, Reni, Gofur, Aris, Becca, Devi, Dendy, Awang yang telah ikut mendukung.
9. Semua pihak yang telah membantu penulis dalam menyelesaikan skripsi ini yang tidak bisa penulis sebutkan satu persatu.

Tentunya laporan tugas akhir ini masih memiliki banyak kekurangan dan kelemahan dalam penyusunannya. Oleh karena itu penulis mengharapkan kritik dan saran yang membangun demi kesempurnaan penyusunan tugas akhir ini. Besar harapan penulis laporan tugas akhir ini dapat bermanfaat bagi semua pihak.

Malang, Februari 2012

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar Belakang

Keamanan suatu informasi dewasa ini makin menjadi sebuah kebutuhan vital dalam berbagai aspek kehidupan. Suatu informasi akan memiliki nilai lebih tinggi apabila menyangkut tentang aspek-aspek keputusan bisnis, keamanan, ataupun kepentingan umum. Dimana informasi-informasi tersebut tentunya akan banyak diminati oleh berbagai pihak yang juga memiliki kepentingan di dalamnya.

*Steganography*merupakan ilmu yang mempelajari, meneliti dan mengembangkan seni menyembunyikan suatu informasi. *Steganography* dapat di golongkan sebagai salah satu bagian dari ilmu komunikasi. Kata *steganography* berasal dari bahasa yunani yang berarti "tulisan tersembunyi". Pada era informasi digital, *steganography* merupakan teknik dan seni menyembunyikan informasi digital di balik informasi digital lainnya, sehingga informasi digital yang sesungguhnya tidak kelihatan.

Secara teori, semua file umum yang ada dalam komputer dapat digunakan sebagai media, seperti gambar berformat JPG, GIF, BMP, atau di dalam musik MP3 atau WAV, bahkan didalam sebuah film dengan format MPEG atau AVI. Semua dapat dijadikan tempat bersembunyi, asalkan file tersebut memiliki bit-bit data redundan yang dapat di modifikasi. Setelah dimodifikasi file tersebut tidak akan banyak terganggu fungsinya dan kualitasnya tidak akan jauh berbeda dengan aslinya.

Steganography yang berkembang saat ini adalah *steganography* pada image dan audio, akan tetapi, kebanyakan pada *steganography* image dan *steganography* audio, data yang disembunyikan hanya terbatas pada inputan teks.

Berdasarkan pada kelemahan yang ada pada *steganography* pada image dan audio tersebut, dalam tugas akhir ini akan dibahas *steganography* pada video yang dapat menyembunyikan data, yang tidak terbatas pada inputan teks. Padadasarnya video merupakan gabungan dari *image* yang "bergerak" dan audio.

Penggunaan video sebagai media penyimpanan data akan lebih banyak menampung besar ukuran data yang akan disisipkan. Sehingga pengguna data dapat menyembunyikan datanya lebih maksimal. Disamping itu video merupakan aliran dari image, sehingga distorsi pada salah satu frame image tidak terlihat dengan mudah oleh mata manusia.

Penggunaan teknologi *steganography* ini diharapkan dapat membantu upaya dalam peningkatan pengamanan pengiriman informasi dan mempermudah perlindungan atas hak cipta hasil karya media elektronik.

1.2 Rumusan Masalah

Berdasarkan latar belakang permasalahan di atas, maka masalah yang dirumuskan yaitu :

1. Bagaimana teknik penyisipan pesan rahasia ke dalam media citra pada video dengan menggunakan teknik *steganography*
2. Bagaimana membangun suatu aplikasi *steganography* untuk menyembunyikan pesan rahasia dan dapat melindungi keamanan datanya yang disisipkan pada video, sehingga kerahasiaan datanya tetap terjamin aman dan secara kasat mata tidak terjadi perubahan pada video tersebut meskipun telah disisipi data. Di samping itu, data yang telah disisipkan juga dapat dipisahkan kembali dari video.

1.3 Tujuan Penelitian

Sesuai dengan rumusan masalah tersebut maka tujuan dari penulisan skripsi ini adalah :

1. Memberikan solusi bagaimana teknik penyisipan pesan rahasia ke dalam media citra pada video dengan menggunakan teknik *steganography*.
2. Untuk membangun suatu perangkat lunak yang dapat menyembunyikan dan melindungi keamanan informasi rahasia yang disisipkan ke dalam video digital.
3. Untuk membangun suatu perangkat lunak yang dapat memisahkan pesan rahasia dari video digital yang telah disisipi pesan dengan tidak mengurangi nilai dari informasi pesan rahasia tersebut.

1.4 Batasan Masalah

Agar permasalahan mengarah sesuai dengan tujuan yang diharapkan, maka pembahasan dibatasi oleh hal-hal sebagai berikut :

1. Video yang dihasilkan dari proses *steganography* adalah video berformat AVI yang tidak dikompresi.
2. Data digital yang akan disisipkan adalah data digital yang berekstensi (*.docx), (*.xlsx), (*.txt), dan (*.jpg).
3. Tidak membahas stream audio yang ada pada video.
4. Berkas atau data rahasia yang disisipkan atau disembunyikan pada video hanya satu file.
5. Metode yang digunakan adalah *Least Significant Bit* (LSB)
6. Tidak membahas tentang steganalysis dan kriptografi.

1.5 Metodologi Penelitian

Metodologi yang digunakan dalam penulisan tugas akhir ini adalah sebagai berikut :

1. Tahap pengumpulan data

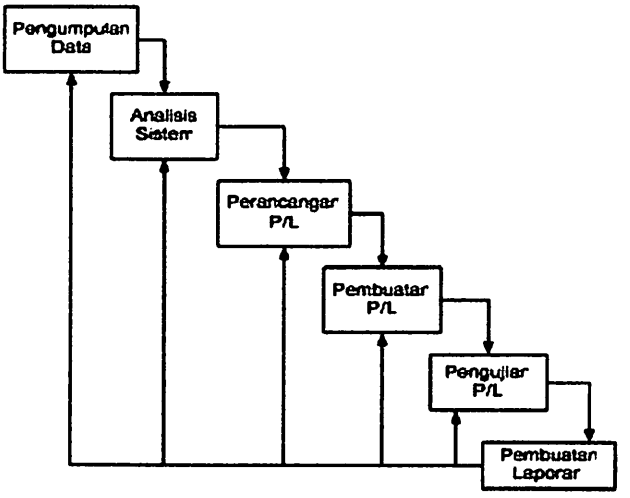
Metode pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut :

a. Studi ke perpustakaan

Studi kepustakaan dilakukan untuk mencari informasi tentang *steganography* video dan prosesnya. Sumber kepustakaan diambil dari buku-buku yang berkaitan dengan *steganography* video dan informasi yang juga dapat diperoleh dari internet.

2. Metode pengembangan perangkat lunak

Tahapan yang akan dilakukan dalam pembangunan sistem ini ialah dengan menggunakan metode *waterfall* seperti diperlihatkan pada gambar 1.1 yaitu :



Gambar 1.1 Diagram metode *waterfall*

Penjelasan dari setiap tahap dari proses pada diagram waterfall diatas adalah sebagai berikut:

a. Pengumpulan data

   Pengumpulan data berupa literatur atau berupa karya-karya ilmiah yang berhubungan dengan penelitian yang dikerjakan.

b. Analisis sistem

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

c. Perancangan Perangkat Lunak

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

d. Pembuatan Perangkat Lunak

   Pembuatan Perangkat Lunak pengolahan data akan dilakukan dengan penggunaan bahasa pemograman yang tepat, efektif dan efisien dengan memperhatikan teknologi komputerisasi.

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.

f. Pembuatan Laporan

Hasil penelitian yang dilakukan akan didokumentasikan untuk kebutuhan lebih lanjut guna pengembangan selanjutnya apabila dibutuhkan.

1.6 Sistematika Penulisan

Untuk mempermudah dan memahami pembahasan penulisan skripsi ini, maka sistematika penulisan disusun sebagai berikut :

BAB I :PENDAHULUAN

Berisi Latar Belakang, Rumusan Masalah, Tujuan Penelitian, Pembatasan Permasalahan, dan Sistematika Penulisan.

BAB II : LANDASAN TEORI

Berisi tentang teori - teori yang berhubungan dengan penelitian yang dilakukan.

BAB III : ANALISA DAN PERANCANGAN SISTEM

Dalam bab ini berisi mengenai analisa kebutuhan sistem baik software maupun hardware yang diperlukan untuk membuat kerangka global yang menggambarkan mekanisme dari sistem yang akan dibuat.

BAB IV : IMPLEMENTASI DAN PENGUJIAN SISTEM

Berisi tentang implementasi dari perancangan sistem yang telah dibuat serta pengujian terhadap sistem tersebut.

BAB V : PENUTUP

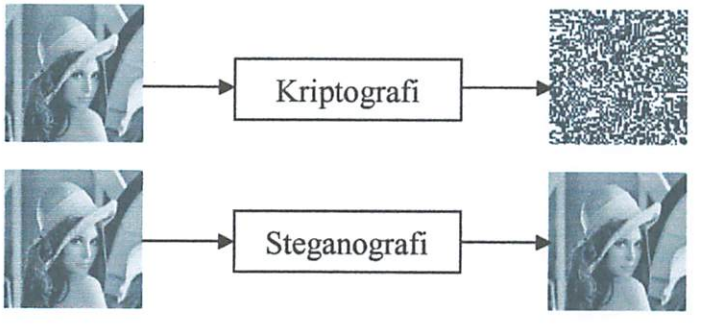
Merupakan bab terakhir yang memuat intisari dari hasil pembahasan yang berisikan kesimpulan dan saran yang dapat digunakan sebagai pertimbangan untuk pengembangan penulisan selanjutnya.

BAB II
LANDASAN TEORI

2.1 Steganography

Seteganografi (*steganography*) merupakan ilmu yang mempelajari, meneliti, dan mengembangkan seni menyembunyikan suatu informasi. Kata "*steganography*" berasal dari bahasa Yunani *steganos*, yang artinya "tersembunyi atau terselubung", dan *graphein*, yang artinya "menulis", sehingga kurang lebih secara keseluruhan artinya adalah tulisan yang disembunyikan (Munir, 2006).

Steganography membutuhkan dua properti: wadah penampung dan data rahasia yang akan disembunyikan. *Steganography* 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.*Steganography* berbeda dengan kriptografi, dimana pihak ketiga dapat mendeteksi adanya data (*chipertext*), karena hasil dari kriptografi berupa data yang berbeda dari bentuk aslinya dan biasanya datanya seolah-olah berantakan, tetapi dapat dikembalikan ke bentuk semula (Suseno, 2002).Ilustrasi mengenai perbedaan kriptografi dan *steganography* dapat dilihat pada Gambar 2.1.

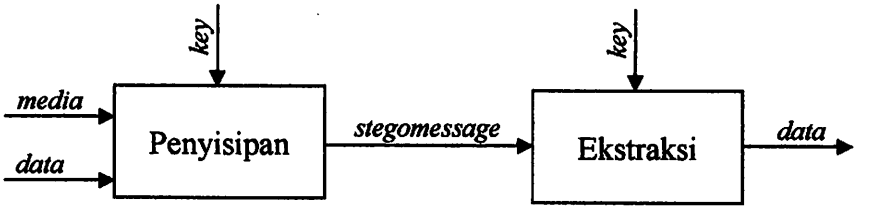


Gambar 2.1Ilustrasi kriptografi dan *steganography* pada citra digital.

Steganography membahas bagaimana sebuah pesan dapat disisipkan ke dalam sebuah berkas media sehingga pihak ketiga tidak menyadarinya.*Steganography* memanfaatkan keterbatasan sistem indera manusia seperti mata dan telinga.Dengan adanya keterbatasan inilah, metoda

steganography ini dapat diterapkan pada berbagai media digital. Hasil keluaran dari *steganography* ini memiliki bentuk persepsi yang sama dengan bentuk aslinya, tentunya persepsi disini sebatas oleh kemampuan indera manusia, tetapi tidak oleh komputer atau perangkat pengolah digital lainnya.

Steganography digital menggunakan media digital sebagai wadah penampung, misalnya citra, suara, teks, dan video.Sedangkan data rahasia yang disembunyikan dapat berupa berkas apapun.Media yang telah disisipi data disebut *stegomessage*. Proses penyembunyian data ke dalam media disebut penyisipan (*embedding*), sedangkan proses sebaliknya disebut ekstraksi. Proses tersebut dapat dilihat pada Gambar 2.2. Penambahan kunci yang bersifat opsional dimaksudkan untuk lebih meningkatkan keamanan.



Gambar 2.2 Proses penyisipan dan ekstraksi dalam *steganography*.

2.1.1 Sejarah Steganography

Steganography berasal dari bahasa Yunani yaitu *steganos*, yang artinya tersembunyi atau terselubung, dan *graphein*, yang artinya menulis, sehingga kurang lebih secara keseluruhan artinya adalah tulisan yang disembunyikan. Secara umum *steganography* merupakan seni atau ilmu yang digunakan untuk menyembunyikan pesan rahasia dengan segala cara sehingga selain orang yang dituju, orang lain tidak akan menyadari keberadaan dari pesan rahasia tersebut.

Steganography sudah digunakan sejak dahulu kala sekitar 2500 tahun yang lalu untuk kepentingan politik, militer, diplomatik, serta untuk kepentingan pribadi sebagai alat. Beberapa contoh penggunaan *steganography* pada masa lampau:

1. Steganografi pada tubuh budak: Pada jaman Yunani Kuno, Herodotus (sejarawan Yunani) menyampaikan pesan dengan cara mencukur kepala budak dan mentato kepalanya dengan pesan tersebut. Kemudian saat

rambutnya tumbuh kembali, budak dikirimkan pada tempat tujuan dan saat rambutnya kembali digunduli maka pesan akan terbaca.

2. Steganografi dalam lapisan lilin: Masih pada jaman Yunani Kuno, pesan rahasia di ukir pada kayu kemudian diberi lapisan lilin untuk menutupi pesan tersebut, dengan begitu pesan dapat disampaikan tanpa menimbulkan kecurigaan. Penggunaan tinta yang tidak terlihat pada pesan lainnya.
3. Steganografi pada kertas: Pada Perang Dunia II, pemerintah Amerika Serikat menulis pesan rahasia pada tentaranya yang di tawan oleh Jerman dengan menggunakan tinta tak terlihat di atas atau di bagian kosong pesan lainnya dan untuk mendeteksinya digunakan air.
4. Steganografi dengan *microdots*: Masih pada Perang Dunia II, agen mata-mata menggunakan *microdots* untuk mengirimkan informasi. Penggunaan teknik ini biasa digunakan pada *microfilm chip* yang harus diperbesar sekitar 200 kali.
5. Steganografi dunia digital: Pada suatu pengarahan yang dilakukan oleh FBI pada akhir September 2001, asisten direktur FBI, Ron Dick menyatakan kemungkinan para pembajakpesawat pada serangan 11 September ke gedung *World TradeCenter* menggunakan teknologi internet seperti website perusahaan palsu atau website porno, e-mail, chat rooms, bulletin boards untuk mengkoordinasi serangan.

2.1.2 Manfaat Steganography

Steganography adalah sebuah pisau bermata dua, ia bisa digunakan untuk alasan-alasan yang baik, tetapi bisa juga digunakan sebagai sarana kejahatan. *Steganography* juga dapat digunakan sebagai salah satu metode watermarking pada *image* untuk proteksi hak cipta, seperti juga digital *watermarking* (*fingerprinting*).*Steganography* juga dapat digunakan sebagai penggantihash. Dan yang terutama, seperti disebutkan sebelumnya, *steganography* dapat digunakan untuk menyembunyikan informasi rahasia, untuk melindunginya dari pencurian dan dari orang yang tidak berhak untuk mengetahuinya. Sayangnya, *steganography* juga dapat digunakan untuk mencuri data yang disembunyikan

pada data lain sehingga dapat dikirim ke pihak lain, yang tidak berhak, tanpa ada yang curiga. *Steganography* juga dapat digunakan oleh para teroris untuk saling berkomunikasi satu dengan yang lain. Sehubungan dengan keamanan sistem informasi, *steganography* hanya merupakan salah satu dari banyak cara yang dapat dilakukan untuk menyembunyikan pesan rahasia. *Steganography* lebih cocok digunakan bersamaan dengan metode lain untuk menciptakan keamanan yang berlapis. Sebagai contoh *steganography* dapat digunakan bersama dengan enkripsi. Windows dan Unix jugamenggunakan *steganography* dalam mengimplementasikan *hidden directory.*

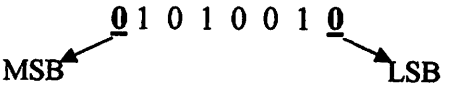
2.1.3 Faktor Penting Dalam Steganography

Penyembunyian data rahasia ke dalam media digital mengubah kualitas media tersebut. Faktor penting dalam penyembunyian data diantaranya adalah:

1. *Fidelity.* Mutu media penampung tidak jauh berubah. Setelah penambahan data rahasia, pengamat tidak mengetahui kalau di dalam media tersebut terdapat data rahasia.
2. *Recovery.* Data yang disembunyikan harus dapat diungkapkan kembali (*recovery*). Karena tujuan steganografi adalah menyembunyikan data, maka sewaktu-waktu data rahasia didalam media penampung harus dapat diambil kembali untuk digunakan lebih lanjut.

2.1.4 Metode Least Significant Bit (LSB)

Metode steganografi yang paling umum pada tipe berkas citra adalah *least significant bit* (LSB). Metode ini menyembunyikan data dengan mengganti bit-bitdata yang paling tidak berarti di dalam cover dengan bit-bit data rahasia. Pada susunanbit di dalam sebuah *byte* (1 *byte* = 8 bit), ada bit yang paling berarti *most significant bit* (MSB) dan bit yang paling kurang berarti *least significant bit* (LSB).



Gambar 2.3 Susunan bit pada sebuah *byte.*

Bit yang cocok untuk diganti adalah bit LSB, sebab perubahan tersebut hanya mengubah nilai *byte* satu lebih tinggi atau satu lebih rendah dari nilai sebelumnya. Misalkan pada cover citra, *byte* tersebut menyatakan warna merah, maka perubahan satu bit LSB tidak mengubah warna merah tersebut secara berarti, apalagi mata manusia tidak dapat membedakan perubahan yang kecil. Sebagai contoh, misalnya terdapat data *raster original* file citra yang akan digunakan sebagai cover sebagai berikut:

00100111 11101001 11001000
00100111 11001000 11101001
11001000 00100111 11101001

Sedangkan representasi biner huruf A adalah 01000001, dengan menyisipkannya ke dalam pixel di atas maka akan dihasilkan

00100110 11101001 11001000
00100110 11001000 11101000
11001000 00100111 11101001

Terlihat hanya tiga bit rendah yang berubah (cetak tebal), untuk penglihatan mata manusia sangatlah mustahil untuk dapat membedakan warna pada file citra yang sudah diisi pesan rahasia jika dibandingkan dengan file citra asli sebelum disisipi dengan pesan rahasia.

2.1.5 Metode Discrete Cosine Transform (DCT)

DCT digunakan terutama pada berkas gambar JPEG, untuk mentransformasikan blok 8x8 pixel yang berurutan dari gambar menjadi 64 koefisien DCT. Setiap koefisien DCT F(u,v) dari blok 8x8 pixel gambar f(x,y) dihitung sebagai berikut :

$$F(u,v) = \frac{1}{4}C(u)C(v)\left[\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y) * cos\frac{(2x+1)u\pi}{16} cos\frac{(2y+1)v\pi}{16}\right]$$

Dengan C(x) =1/√2 jika x=0 dan C(x)=1 jika x=1.

Setelah koefisien-koefisien diperoleh, dilakukan proses kuantisasi sebagai berikut :

$$F^Q(u,v) = \left\lfloor \frac{F(u,v)}{Q(u,v)} \right\rfloor$$

dengan Q(u,v) adalah 64-elemen dari tabel kuantisasi.

Sebagai contoh, berikut merupakan algoritma sederhana untuk menyembunyikan pesan di dalam image JPEG :

Input : pesan, cover image

Output : stego

while (masih ada data untuk di-embed) **do**

ambil koefisien DCT selanjutnya dari cover image (DCT)

if koefisien < nilai treshold **then**

ambil bit selanjutnya dari pesan

ganti bit koefisien DCT dengan bit pesan tersebut

end if

masukkan DCT ke stego (invers DCT)

end while

Walaupun image yang dikompresi dengan lossy compression akan menimbulkan kecurigaan karena perubahan LSB akan terlihat jelas, pada metode ini hal ini tidakakan terjadi karena metode ini terjadi di domain frekuensi di dalam *image*, bukan pada domain spasial, sehingga tidak akan ada perubahan yang terlihat pada cover image.

2.1.6 Steganography Pada File Video

Seperti dikatakan sebelumnya, video merupakan kumpulan dari *image* yang "bergerak", jadi sebagian besar metode yang digunakan pada *imagesteganography* dapat digunakan pada video *steganography*. Dapat dikatakan bahwa video *steganography* merupakan turunan dari *imagesteganography*.

Metode yang dapat digunakan dalam video *steganography* yaitu LSB (*Least Significant Bit*) dan DCT (*Discrete Cosine Transform*). Perbedaan antara kedua metode tersebut adalah, jika menggunakan metode LSB, file yang

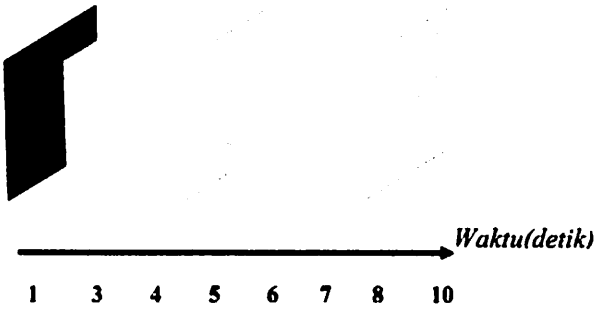
digunakan harus bersifat *lossless compression*, dikarenakan pada LSB pada proses penyembunyiannya menggunakan bit LSB pada file induk untuk menyembunyikan bit dari data pesan, sehingga jika dilakukan proses kompresi pada file induk maka data pesan akan hilang. Untuk metode DCT dapat digunakan pada file yang bersifat *lossy compression*, karena pada metode ini proses penyembunyian dilakukan pada frekuensi domain file induk, sehingga tidak akan mempengaruhi jika dilakukan proses kompresi pada file induk.

Pada steganografi video dengan menggunakan metode LSB menyembunyikan bit-bit dari berkas rahasia pada bit-bit segmen *frame* video. Penyembunyian ini pada dasarnya memberikan pengaruh terhadap berkas cover *frame* video, tetapi karena perubahan yang terjadi sangat kecil sehingga tidak tertangkap oleh indera manusia. Sebagai contoh, untuk citra bitmap 24-bit yang berukuran 256 x 256 piksel terdapat 65536 piksel dan setiap pikselnya berukuran 3 *byte* (R=1 *byte*, G=1 *byte* dan B=1 *byte*), berarti seluruhnya ada 65536 x 3 = 196608 *byte*. Karena dalam setiap *byte* hanya bisa menyembunyikan 1 bit pada LSB-nya, maka ukuran berkas rahasia maksimum yang dapat disimpan pada citra tersebut adalah 196608 / 8 = 24576 *byte* atau 1/8 dari ukuran citra tersebut.

Keuntungan dari video *steganography* adalah banyaknya data yang dapat disembunyikan didalamnya, serta fakta bahwa video merupakan "*streams*" dari gambar-gambar menyebabkan adanya distorsi pada salah satu *frame* gambar tidak akan dilihat dengan mudah dengan mata manusia. Akan tetapi, semakin banyak data pesan yang disembunyikan, bukan hal yang mustahil jika perubahan pada video menjadi semakin mudah terlihat.

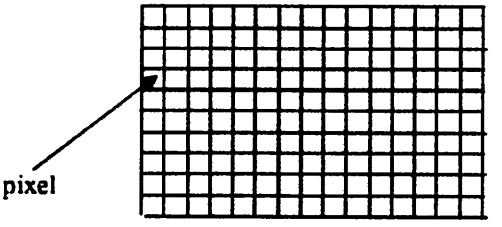
2.2 Struktur Video Digital

Video digital pada dasarnya tersusun atas serangkaian *frame* yang ditunjukan pada gambar 2.4. Rangkaian *frame* tersebut ditampilkan pada layar dengan kecepatan tertentu, bergantung kepada laju *frame* yang diberikan (dalam *frame*/detik). Jika laju *frame* cukup tinggi, maka manusia dapat menangkap gambar per *frame* secara berkelanjutan seolah-olah melihat gambar bergerak.



Gambar 2.4 Aliran*Frame.*

Masing masing *frame* merupakan gambar atau citra digital.Suatu citra digital direpresentasikan dengan sebuah matriks yang masing-masing elemennya merepresentasikan nilai intensitas. Jika I adalah matriks dua dimensi, I(x,y) adalah nilai intensitas yang sesuai pada posisi baris x dan kolom y pada matriks tersebut. Titik-titik ditempat citra sampling disebut picture element, atau sering dikenal dengan istilah *pixel* (piksel) yang ditunjukan pada gambar 2.5.



Gambar 2.5 *Pixel* (*Picture Element*).

Karakteristik suatu video digital ditunjukan oleh resolusi atau dimensi *frame*, kedalaman *pixel*, laju *frame*. Karakteristik-karakteristik ini akan memperlihatkan antara kualitas video dan jumlah bit yang dibutuhkan untuk menyimpan besarnya data atau mengirimkan data melalui jaringan. Beberapa penjelasan mengenai karakteristik video digital sebagai berikut.

2.2.1 Resolusi (*Resolution*)

Resolusi atau dimensi *frame* adalah ukuran suatu *frame*. Resolusi dinyatakan dalam *pixel* x *pixel*. Semakin tinggi resolusi maka akan semakin baik kualitas video tersebut, dalam arti bahwa dalam ukuran fisik yang sama, video dengan resolusi yang tinggi akan lebih detail terlihat. Resolusiyang tinggi akan

mengakibatkan jumlah bit yang diperlukan untuk penyimpanan data menjadi meningkat.

Gambar 2.6 Resolusi Dimensi *Frame*.

Pada gambar 2.6 terdapat dua *frame* dengan resolusi berbeda. *frame* sebelah kiri memiliki resolusi 240 *pixel* x 344 *pixel*, sedangkan *frame* sebelah kanan memiliki resolusi 178 *pixel* x 255 *pixel*.

2.2.2 Kedalaman Bit (*Bit Depth*)

Kedalaman bit menentukan jumlah bit yang digunakan untuk mereprentasikan tiap *pixel* pada sebuah *frame*. Kedalaman bit dinyatakan dalam bit/*pixel*. Semakin banyak jumlah bit yang digunakan untukmerepresentasikan sebuah *pixel* yang berarti semakin tinggi kedalaman *pixel*-nya, maka semakin tinggi pula kualitasnya, dengan bayaran jumlah bit yang dibutuhkan menjadi lebih tinggi.

Dengan satu *byte* (8-bit) untuk tiap *pixel*, diperoleh 28 atau 256 level intensitas. Dengan level intensitas sebanyak itu, umumnya mata manusia sudah dapat dilihat dengan cukup jelas. Kedalaman *pixel* paling rendah terdapat pada binary-value image yang hanya menggunakan 1 bit untuk tiap *pixel*, sehingga hanya ada dua kemungkinan bagi tiap *pixel*, yaitu 0 (hitam) atau 1 (putih).

Gambar 2.7 Variasi Kedalaman *Pixel.*

Sebagai contoh, dalam gambar 2.7, suatu citra yang sama direpresentasikan dalam beberapa kedalaman *pixel.* Pada gambar kiri, digunakan 16 bit untuk tiap *pixel.* Pada gambar tengah, bit yang digunakan untuk merepresentasikan sebuah *pixel* dikurangi menjadi 8 bit per *pixel.*Sedangkan pada gambar kanan, digunakan 4 bit per *pixel.* Terlihat bahwa semakin sedikit jumlah bit yang digunakan untuk tiap *pixel*, maka semakin turun pula kualitasnya.

2.2.3 Laju *Frame* (*Frame Rate*)

Frame rate menunjukan jumlah *frame* yang digambar setiap detik, dan dinyatakan dengan *frame* per detik. Sehubungan dengan laju *frame* ini, ada duahal yang perlu diperhatikan, yaitu kehalusan gerakan (smooth motion) dan kilatan (flash). Kehalusan gerakan ditentukan oleh jumlah *frame* yang berbeda per detik. Untuk mendapatkan gerakan yang halus video digital setidaknya harus menampilkan sedikitnya 25 *frame* per detik. Kilatan ditentukan oleh jumlah berapa kali layar digambar per detik.

2.2.4 Representasi Warna

Pada video digital, umunya data video dipisahkan menjadi komponen-komponen, baik untuk komponen warna maupun komponen kecerahan. Penyajian semacam ini disebut komponen video. Pada komponen video, tiapkomponen dipisahkan dengan cara tertentu. Beberapa cara pemisahan komponen tersebut adalah RGB,YUV dan YIQ.

RGB

Data video dapat dipisahkan menjadi komponen-komponen untuk masing-masing warna, yaitu merah (*red*), hijau (*green*) dan biru (*blue*).Warna tiap *pixel* ditentukan oleh kombinasi intensitas dari masing-masing komponen warna. Sebagai contoh, pada RGB 24 bit, masing-masing komponen warna

dinyatakan dalam 8 bit atau 256 level. Warna biru langit direpresentasikan dengan R=181, G=189 dan B=249.

YUV

Pemisahan komponen tidak hanya dilakukan dengan pemisahan warna, namun dapat juga dilakukan dengan memisahkan komponen kecerahan (*luminance*) dan komponen warna (*crominance*). Pada format PAL, sinyal kecerahan dinyatakan dengan Y, sedangkan dua sinyal warna dinyatakan dengan U dan V.

Masing-masing komponen tersebut diperoleh dengan mengirimkan masukan RGB dengan rumus :

$$Y = 0{,}299R + 0{,}587G + 0{,}114B \quad \text{(Persamaan 3.1)}$$

$$U = (B - Y) \times 0{,}493 \quad \text{(Persamaan 3.2)}$$

$$V = (R - Y) \times 0{,}877 \quad \text{(Persamaan 3.3)}$$

YIQ

Pemisahan sinyal video menjadi luminance dan crominance dapat dilakukan juga sesuai dengan format *National Television Systems Comitte* (NTSC), *luminance* dinyatakan dengan Y, dan dua *crominance* dinyatakan dengan I dan Q.

Masing-masing komponen tersebut diperoleh dengan mentransformasikan RGB dengan rumus :

$$Y = 0{,}299R + 0{,}587G + 0{,}114B \quad \text{(Persamaan 3.4)}$$

$$I = 0{,}596R + 0{,}275G + 0{,}321B \quad \text{(Persamaan 3.5)}$$

$$Q = 0{,}212R + 0{,}523G + 0{,}311B \quad \text{(Persamaan 3.6)}$$

2.3 AVI (Audio Video Interleave)

AVI (*Audio Video Interleave*) merupakan format berkas (*file*) video buatan Microsoft. Format ini merupakan salah satu format video tertua yang diperkenalkan Microsoft sejak tahun 1992 sebagai format *playback audio video* pada system operasi windows. AVI sering dikatakan sebagai format video yang belum terkompresi, dikarenakan format ini akan menghasilkan ukuran file yang

sangat besar. Format ini juga dipakaisebagaikualitas terbaik yang digunakan untukmenentukan hasil akhir dari video.Pada file AVI terdapat 3 bagian utama yang merupakan komponen penyusun audio dan visual pada file AVI.

2.3.1 AVI Header

File AVI dimulai dengan *header* utama. Pada file AVI, *header* ini ditandai dengan 4 karakter kode. *Header* mengandung informasi utama yang terdapat pada file AVI, seperti jumlah *stream* pada file, juga tinggi dan lebar dari sekuensial file AVI. Susunan / isi *header* file AVI dalam *memory*:

```
typedef struct {
    DWORD dwMaxBytesPerSec;
    DWORD dwFlags;
    DWORD dwCaps;
    DWORD dwStreams;
    DWORD dwSuggestedBufferSize;
    DWORD dwWidth;
    DWORD dwHeight;
    DWORD dwScale;
    DWORD dwRate;
    DWORD dwLength;
    DWORD dwEditCount;
    TCHAR szFileType[64];
} AVIFILEINFO;
```

- dwMaxBytesPerSec.

  Mengidentifikasikan kecepatan maksimal data per-detik dari file AVI.

- dwFlags.

  Berisikan kode untuk metode penggunaan file AVI (seperti HASINDEX,MUSTUSEINDEX, COPYRIGHTED).

- dwCaps.

  Berisikan kode untuk metode kemampuan file AVI (seperti CAN READ,CANWRITE, ALLKEYFRAME, NOCOMPRESSION).

- dwStreams.

  Jumlah *stream* pada file AVI, contohnya pada file yang mempunyai audiodan visual akan mempunyai 2 *stream*.

- dwSuggestedBufferSize.

  Ukuran *buffer* yang digunakan untuk menyimpan data file AVI pada *memory*.

- dwWidth.

Lebar (dalam piksel) dari file AVI.

- dwHeight.

  Tinggi (dalam piksel) dari file AVI.

- dwScale.

  Skala waktu yang digunakan pada keseluruhan file AVI.

- dwRate.

  Jumlah sampel dari file AVI. Untuk mendapatkan nilai sampel per detik dengan membagi nilai dwRate dengan dwScale.

- dwLength.

  Panjang / lama dari file AVI, didefinisikan berdasarkan dwScale dan dwRate.

- dwEditCount.

  Jumlah *stream* yang ditambahkan atau dihilangkan dari file.

- szFileType.

  Deskripsi dari file yang digunakan.

2.3.2 AVI Stream

AVI stream terdiri dari 2 jenis yaitu *stream* video dan *stream* audio, *stream* audio tidak harus terdapat di dalam suatu file AVI. Isi / susunan *stream* dalam *memory*:

```
typedef struct {
      DWORD fccType;
      DWORD fccHandler;
      DWORD dwFlags;
      DWORD dwCaps;
    WORD  wPriority;
    WORD  wLanguage;
      DWORD dwScale;
      DWORD dwRate;
      DWORD dwStart;
      DWORD dwLength;
      DWORD dwInitialFrames;
      DWORD dwSuggestedBufferSize;
      DWORD dwQuality;
      DWORD dwSampleSize;
    RECT  rcFrame;
      DWORD dwEditCount;
      DWORD dwFormatChangeCount;
      TCHAR szName[64];
} AVISTREAMINFO;
```

- fccType.

  Kode 4 karakter yang mendefinisikan tipe dari *stream.*

- fccHandler.

  Kode 4 karakter dari *handle* yang menangani kompresi pada saat file disimpan.

- dwFlags.

  Kode untuk metode penggunaan stream (seperti AVISTREAMINFODISABLED, AVISTREAMINFO ENABLED).

- dwCaps.

  Kode untuk metode kemampuan *stream.*

- dwPriority.

  Prioritas dari *stream.*

- dwLanguage.

  Bahasa yang digunakan pada *stream.*

- dwScale.

  Skala waktu yang digunakan pada keseluruhan *stream.*

- dwRate.

  Jumlah sampel dari *stream.*Untuk mendapatkan nilai sampel per detik dengan membagi nilai dwRate dengan dwScale.

- dwStart.

  Nilai frame awal dari *stream.*

- dwLength.

  Panjang / lama dari *stream.*

- dwInitialFrames.

  Bagian ini menspesifikasikan ukuran dari pergeseran audio dan video data pada file AVI.

- dwSuggestedBufferSize.

  Ukuran *buffer* yang digunakan untuk menyimpan data file AVI pada memory.

- dwQuality.

  Indikator kualitas data video pada *stream,* ukuran indikator antara 0

sampai 10000.

- dwSampleSize.

  Ukuran dalam *byte* untuk satu sampel data dalam *stream.*

- rcFrame.

  Dimensi yang menunjukkan nilai dari posisi tepi, tinggi dan lebar dari video.

- dwEditCount.

  Jumlah dari proses pengeditan *stream* yang pernah dilakukan.

- dwFormatChangeCount.

  Jumlah dari proses pengeditan format *stream* yang pernah dilakukan.

- szName.

  Deskripsi dari *stream.*

2.3.3 AVI Frame

Frame pada *stream* video yang terdapat pada file AVI merupakan suatu data yang berbentuk DIB (*Device Independent Bitmap*). Bentuk DIB secara umum terdiri dari 3 bagian utama.

2.3.3.1 BITMAP FILE HEADER

Struktur BITMAP FILE HEADER mengandung informasi tentang tipe, ukuran, dan layout dari file DIB.

```
typedef struct tag BITMAPFILEHEADER {
    WORD    bfType;
    DWORD   bfSize;
    WORD    bfReserved1;
    WORD    bfReserved2;
    DWORD   bfOffBits;
} BITMAPFILEHEADER;
```

- bfType.

  Menspesifikasikan tipe dari file.

- bfSize.

  Menspesifikasikan ukuran keseluruhan dari file.

- bfReserved.

  Berisikan data cadangan, biasanya di beri nilai 0.

- bfOffbit.

  Menspesifikasikan besarnya *offset* dari BITMAPFILEHEADER ke data bitmap yang sebenarnya.

2.3.3.2 BITMAP INFO

Pada bitmap info terdiri dari 2 struktur utama yang dapat dijabarkan menjadi lebih detail lagi yaitu BITMAP INFO HEADER dan RGBQUAD.

```
typedef struct tagBITMAPINFO {
    BITMAPINFOHEADER    bmiHeader;
    RGBQUAD             bmiColors[1];
} BITMAPINFO;
```

1. BITMAPINFOHEADER

Class ini berisikan informasi tentang dimensi dan format warna dari DIB.

```
typedef struct tagBITMAPINFOHEADER{
    DWORD  biSize;
    DWORD  biWidth;
    DWORD  biHeight;
    WORD   biPlanes;
    WORD   biBitCount;
    DWORD  biCompression;
    DWORD  biSizeImage;
    DWORD  biXPelsPerMeter;
    DWORD  biYPelsPerMeter;
    DWORD  biClrUsed;
    DWORD  biClrImportant;
} BITMAPINFOHEADER;
```

- biSize.

  Menspesifikasikan besar dari BITMAPINFOHEADER dalam *byte*.

- biWidth.

  Menspesifikasikan lebar dari bitmap dalam piksel.

- biHeight.

  Menspesifikasikan tinggi dari bitmap dalam piksel.

- biPlanes.

  Menspesifikasikan jumlah *plane* dalam bitmap, bernilai 1.

- biBitcount.

  Menspesifikasikan jumlah bit per piksel, bernilai 1, 4, 8, 16, 24 bit.

- biCompression.

  Menspesifikasikan tipe kompresi pada bitmap.

- biSize*Image*.

  Menspesifikasikan ukuran dari data gambar pada bitmap.

- biXPelsPerMeter.

  Menspesifikasikan resolusi horisontal dalam piksel per meter pada *device*tujuan untuk bitmap.

- biYPelsPerMeter.

  Menspesifikasikan resolusi vertikal dalam piksel per meter pada *device* tujuan untuk bitmap.

- biClrUsed.

  Menspesifikasikan jumlah index warna yang digunakan pada tabel warna.

- biClrImportant.

  Menspesifikasikan jumlah index warna yang dianggap penting untuk menampilkan bitmap.

2. RGBQUAD

Class ini mendeskripsikan warna yang berisikan hubungan intensitas antara merah (*red*), hijau (*green*), biru (*blue*). Biasanya digunakan sebagai tabel warna untuk bitmap yang berformat 4 dan 8 bit per piksel.

```
typedef struct tagRGBQUAD {
    BYTE    rgbBlue;
    BYTE    rgbGreen;
    BYTE    rgbRed;
    BYTE    rgbReserved;
```

```
} RGBQUAD;
```

- rgbRed.

  Menspesifikasikan intensitas dari warna merah.
- rgbGreen.

  Menspesifikasikan intensitas dari warna hijau.
- rgbBlue.

  Menspesifikasikan intensitas dari warna biru.
- rgbReserved.

  Bit cadangan dan biasanya diberi nilai 0 pada format rgb 5-5-5, akan tetapiakan berisi bagian dari bit red apabila format rgb nya 5-6-5.

2.3.3.3 Data Pixel

Data dari gambar dalam DIB dibentuk berdasarkan biBitCount (*bit per piksel*) yang terdapat dalam informasi bitmap. Untuk mendapatkan *pointer* dari data piksel awal yang terdapat dalam DIB digunakan *pointer* dari BITMAPINFO ditambah dengan besar ukuran dari BITMAPINFO ditambahkan dengan besar ukuran dari RGBQUAD.

*Bitmap_data = (LPBYTE)BitmapInfo + BitmapInfo->biSize + (biClrUsed * sizeof(RGBQUAD);*

Ukuran dari *bit per piksel* menentukan struktur dari data yang disimpandalam DIB.

- 1 bit per piksel.

  Gambar berupa *monochrome* (hitam putih), dimana tiap bit menentukan hitam / putihnya piksel dalam gambar.
- 4 bit per piksel.

  Tiap 4 bit menentukan nilai index warna dari tiap piksel yang kemudian dibaca melalui tabel warna yang ada.
- 8 bit per piksel.

  Tiap 8 bit / 1 *byte* menentukan nilai index warna dari tiap piksel yang kemudian dibaca melalui tabel warna yang ada.
- 16 bit per piksel.

  Tiap 16 bit / 2 *byte* menentukan nilai warna piksel yang terdiri dari

merah,hijau dan biru. Susunan bit warna terbalik dari biru (5 bit), hijau (6 bit), merah (5 bit). Dan gambar 16 bit tidak memiliki RGBQUAD (tabel warna).

- 24 bit per piksel.

  Tiap 24 bit / 3 *byte* menentukan nilai warna piksel.*Byte* pertama menentukan warna biru, *byte* kedua warna hijau, dan *byte* ketiga warna merah. Gambar 24 bit tidak memiliki RGBQUAD (tabel warna).

2.4 Format Berkas Bitmap (BMP)

Citra disimpan di dalam berkas dengan format tertentu. Format citra yang baku di lingkungan sistem operasi Microsoft Windows dan IBM OS/2 adalah berkas bitmap (BMP). Saat ini format BMP memang "kalah" populer dibandingkan format JPG atau GIF.Hal ini karena berkas BMP pada umumnya tidak di kompresi, sehingga ukuran berkasnya relatif lebih besar dari pada berkas JPG maupun GIF.Hal ini juga yang menyebabkan format BMP sudah jarang digunakan.

Meskipun format BMP mempunyai kekurangan dari segi ukuran berkas, namun format BMP mempunyai kelebihan dari segi kualitas gambar.Citra dalam format BMP lebih bagus dari pada citra dalam format lainnya karena citra dalam format BMP umumnya tidak dikompresi sehingga tidak ada informasi yang hilang. Terjemahan bebas dari bitmap adalah berkas bit. Artinya nilai intensitaspixel di dalam citra dipetakan ke sejumlah bit tertentu. Peta bit yang umum adalah 8, artinya setiap pixel panjangnya 8 bit. Delapan bit ini merepresentasikan nilai intensitas pixel. Dengan demikian ada sebanyak $2^8 = 256$ derajat keabuan, mulai dari 0 sampai 255.

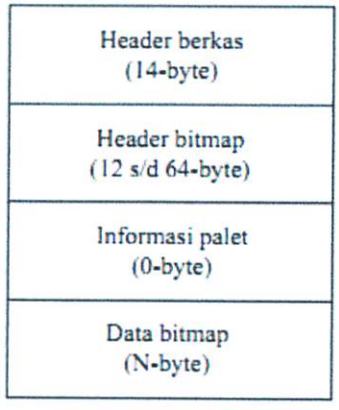
Citra dalam format BMP ada tiga macam: citra biner, citra berwarna dan citra hitam-putih. Citra biner hanya mempunyai dua nilai keabuan, 0 dan 1. Oleh karena itu, 1 bit sudah cukup untuk merepresentasikan nilai pixel.Citra berwarna adalah citra yang lebih umum.Warna yang terlihat pada citra bitmap merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau dan biru.Setiap pixel disusun oleh tiga komponen warna, yaitu R (*red*), G (*green*) dan B (*blue*).Kombinasi dari ketiga warna tersebut menghasilkan warna yang khas untuk pixel yang

bersangkutan. Pada citra 256 warna, setiap pixel panjangnya 8 bit, tetapi komponen warna RGB-nya disimpan di dalam tabel RGB yang disebut *pallete*. Setiap komponen panjangnya 8 bit, jadi ada 256 nilai keabuan untuk warna merah, 256 untuk warna hijau dan 256 untuk warna nilai keabuan untuk warna biru. Nilai setiap pixel tidak menyatakan derajat keabuannya secara langsung, tetapi nilai pixel menyatakan indeks tabel RGB yang memuat nilai keabuan merah (R), nilai keabuan hijau (G), dan nilai keabuan biru (B) untuk pixel yang bersangkutan. Pada citra hitam-putih, nilai R = G = B untuk menyatakan bahwa citra hitam-putih hanya mempunyai satu kanal warna. Citra hitam-putih umumnyaadalah citra 8-bit.

Citra yang lebih kaya warna adalah citra 24-bit.Setiap pixel panjangnya 24-bit, karena setiap pixel langsung menyatakan komponen warna merah, komponen warna hijau dan komponen warna biru.Masing-masing komponen panjang 8-bit. Citra 24-bit disebut juga citra 16 juta warna, karena ia mampu menghasilkan 2^{24} = 16.777.216 kombinasi warna.

Saat ini beredar tiga versi berkas bitmap, yaitu berkas bitmap versi lama dari Microsoft Windows atau IBM OS/2, berkas bitmap versi baru dari Microsoft Windows dan yang ketiga adalah berkas bitmap versi baru dari IBM OS/2. Yang membedakan dari ketiga versi diatas adalah panjang *header*-nya.*Header* adalah data yang terdapat pada bagian awal berkas citra.Data dalam *header* berguna untuk mengetahui bagaimana citra dalam format bitmap dikodekan dan disimpan.Data di dalam *header* misalnya panjang citra, lebar citra, kedalaman pixel, dan sebagainya.

Setiap berkas bitmap terdiri atas *header* berkas, *header* bitmap, informasi palet, dan data bitmap.



Gambar 2.8 Struktur berkas citra bitmap.

Ukuran header berkas sama untuk semua versi, yaitu 14 byte. Tabel 2.1 memperlihatkan filed-filed penyusunan *header* berkas.

Tabel 2.1 *Header* berkas bitmap (Panjang = 14 *byte*)

Byte ke-	Panjang (byte)	Nama	Keterangan
1-2	2	Bmp Type	Tipe berkas bitmap: BA=bitmap array, CI = icon BM= bitmap, CP= color pointer PT= pointer
3-6	4	Bmp Size	Ukuran berkas bitmap
7-8	2	XhotSpot	X hotspot untuk kursor
9-10	2	YhotSpot	Y hotspot untuk kursor
11-14	4	OffBits	Ofset ke awal data bitmap (dalam byte)

Ukuran *header* bitmap berbeda-beda untuk setiap versi. Untuk berkas bitmap versi lama, *header* bitmap berukuran 12 *byte* (Tabel 2.2), untuk berkas versi baru dari Microsoft Windows, *header* bitmap berukuran 40 byte (Tabel 2.3), dan untuk berkas bitmap versi baru dari IBM OS/2, *header* bitmap berukuran 64 *byte* (Tabel 2.4).

Tabel 2.2 *Header* bitmap versi lama dari Microsoft Windows (12 *byte*)

Byte ke-	Panjang (byte)	Nama	Keterangan
1-4	4	HdrSize	Ukuran header dalam ukuran byte
5-6	2	Width	Lebar bitmap dalam ukuran pixel
7-8	2	Height	Tinggi bitmap dalam ukuran pixel
9-10	2	Planes	Jumlah plane (umum-nya selalu satu)
11-12	2	BitCount	Jumlah bit per pixel

Tabel 2.3*Header* bitmap versi baru dari Microsoft Windows (40 *byte*)

Byte ke-	Panjang (byte)	Nama	Keterangan
1-4	4	HdrSize	Ukuran header dalam ukuran byte
5-8	4	Width	Lebar bitmap dalam ukuran pixel
9-12	4	Height	Tinggi bitmap dalam ukuran pixel
13-14	2	Planes	Jumlah plane (umum-nya selalu satu)
15-16	2	BitCount	Jumlah bit per pixel
17-20	4	Compression	0 = tak dikompresi, 1 = dikompresi
21-24	4	ImgSize	Ukuran bitmap dalam byte
25-28	4	HorzRes	Resolusi horizontal
29-32	4	VertRes	Resolusi vertical
33-36	4	ClrUsd	Jumlah warna yang digunakan
37-40	4	ClrImportant	Jumlah warna yang penting

Tabel 2.4 *Header* bitmap versi baru dari IBM OS/2 (64 *byte*)

Byte ke-	Panjang (byte)	Nama	Keterangan
1-4	4	HdrSize	Ukuran header dalam ukuran byte
5-8	4	Width	Lebar bitmap dalam ukuran pixel
9-12	4	Height	Tinggi bitmap dalam ukuran pixel
13-14	2	Planes	Jumlah plane (umum-nya selalu satu)
15-16	2	BitCount	Jumlah bit per pixel
17-20	4	Compression	0 = tak dikompresi, 1 = dikompresi
21-24	4	ImgSize	Ukuran bitmap dalam byte
25-28	4	HorzRes	Resolusi horizontal
29-32	4	VertRes	Resolusi vertical
33-36	4	ClrUsd	Jumlah warna yang digunakan
37-40	4	ClrImportant	Jumlah warna yang penting
41-42	2	Units	Satuan pengukuran yang dipakai
43-44	2	Reserved	Filed cadangan
45-46	2	Recording	Algoritma perekaman
47-48	2	Rendering	Algoritma halftoning
49-52	4	Size1	Nilai ukur 1
53-56	4	Size2	Nilai ukur 2

57-60	4	ClrEncoding	Pengkodean warna
61-64	4	Identifier	Kode yang digunakan aplikasi

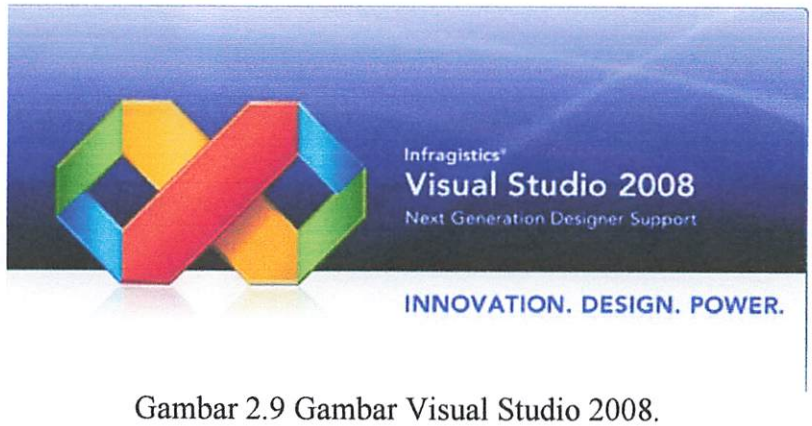
Informasi palet warna terletak sesudah *header* bitmap, Informasi palet warna dinyatakan dalam suatu tabel RGB.Setiap *entry* pada tabel terdiri atas tiga buah *field*, yaitu R (*red*), G (*green*), dan B (*blue*).

Tabel 2.5 Panjang informasi palet untuk setiap versi berkas bitmap

Citra m bit	Versi lama (Windows)	Versi baru (Windows)	Versi baru (OS/2)
Citra 4-bit	48 byte	64 byte	64 byte
Citra 8-bit	768 byte	1024 byte	1024 byte
Citra 24-bit	0 byte	0 byte	0 byte

Data bitmap diletakan sesudah informasi palet.Setiap elemen data bitmap panjangnya 3-*byte*, masing-masing menyatakan komponen RGB.Data bitmap ini merupakan data berkas bitmap yang sebenarnya, pada bagian inilah informasi dapat disimpan dengan metode LSB.

2.5 Microsoft Visual Studio 2008



Gambar 2.9 Gambar Visual Studio 2008.

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap yang dapat digunakan untuk melakukan pengembangan aplikasi, baik aplikasi bisnis, aplikasi *personal* maupun komponen aplikasinya, dalam bentuk aplikasi *console*, aplikasi windows, ataupun aplikasi Web.Visual Studio mencakup *compiler*, SDK, *Intgreted Development Environment* (IDE), dan dokumentasi (umumnya berupa

MSDN Library).*Compiler* yang dimasukkan ke dalam paket Visual Studio antara lainVisual C++, Visual C#, Visual Basic, Visual Basic.NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe.

2.6 Visual Basic .NET

Microsoft Visual Basic .NET adalah sebuah perangkat lunak untuk mengembangkan dan membangun aplikasi yang bergerak di atas system.NET Framework, dengan menggunakan bahasa BASIC.Dengan menggunakan perangkat ini, para *programmer* dapat membangun aplikasi Windows Form, Aplikasi web berbasis ASP.NET, dan juga aplikasi*command-line*.Bahasa Visual Basic .NET sendiri menganut paradigma bahasa pemrograman berorientasi objek yang dapat dilihat sebagai evolusi dari Microsoft Visual Basic versi sebelumnya yang diimplementasikan di atas .NET Framework.Peluncurannya mengundang kontroversi, mengingat banyak sekali perubahan yang dilakukan oleh Microsoft.

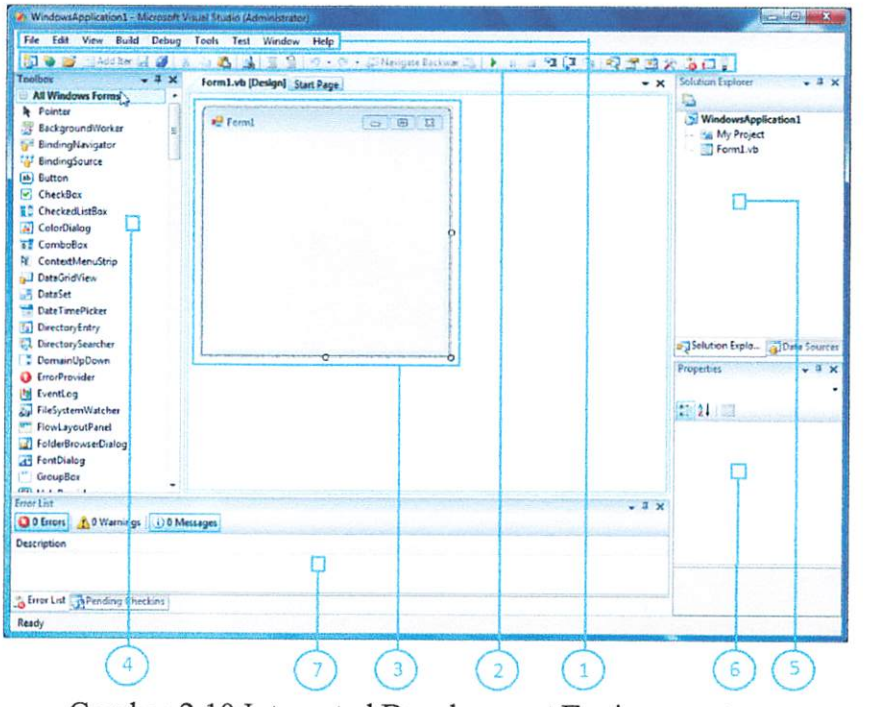
2.6.1 Integrated Development Environment (IDE)

Intergreted Development Environment (IDE) adalah program komputer yang memiliki beberapa fasilitas yang diperlukan dalam pembangunan perangkat lunak.Tujuan dari IDE adalah untuk menyediakan semua utilitas yang diperlukan dalam membangun perangkat lunak.Sebuah IDE setidaknya memiliki fasiltas *Editor, Compiler, Linker* dan *Debugger*.Sampai tahap tertentu IDE dapat membantu memberikan saran yang mempercepat penulisan.Pada saat penulisan kode, IDE juga dapat menunjukkan bagian-bagian yang jelas mengandung kesalahan.

Tampilan IDE secara keseluruhan ditunjukkan pada Gambar 2.10.

1. Menu Bar, adalah suatu menu yang terdiri dari beberapa menu utama, masing-masing memiliki sub menu dan perintah lengkap dengan *shortcut key*.
2. Toolbar Standart, adalah suatu baris menu yang mempunyai fungsi yang sama pada setiap Tool standart pada umumnya, seperti fungsi untuk menyimpan, meng-*copy*, menambah *project* baru, mengatur tampilan program dan masih banyak lagi.

3. Form Design, adalah suatu lembar *form* yang berfungsi untuk merancang tampilan aplikasi secara visual dengan menempatkan control-kontrol yang diperlukan.
4. Toolbox, adalah suatu jendela yang berfungsi untuk menampung komponen-komponen standart.
5. Solution Explorer, adalah suatu jendela yang berfungsi untuk menampilkan *object* yang digunakan untuk membuat aplikasi, seperti *form*, *class*, *module*, dan *object* lainnya.
6. Properties Window, adalah suatu jendela yang berfungsi untuk mengatur nilai properties dari masing-masing komponen yang akan digunakan.
7. Error list, adalah suatu jendela yang berfungsi untuk menampilkan setiap kesalahan dari pembuatan kode program suatu aplikasi.

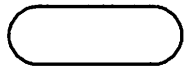


Gambar 2.10 Integrated Development Environment

2.7 Flowchart

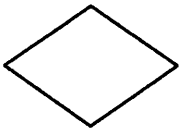
Flowchart adalah representasi grafik dari langkah-langkah yang harus diikuti dalam menyelesaikan suatu permasalahan yang terdiri atas sekumpulan simbol, dimana masing-masing simbol merepresentasikan suatu kegiatan tertentu.*Flowchart* diawali dengan penerimaan input, pemrosesan *input*, dan

diakhiri dengan penampilan *output*. Pada Tabel 2.6 akan ditunjukkan symbol-

SIMBOL	NAMA	FUNGSI
	**TERMINATOR**	Permulaan/akhir program
	GARIS ALIR (FLOW LINE)	Arah aliran program

simbol dari *flowchart*dan fungsinya:

Tabel 2.6 Tabel Simbol - Simbol Flowchart

	**PREPARATION**	Proses inisialisasi/pemberian harga awal
	**PROSES**	Proses perhitungan/proses pengolahan data
	**INPUT/OUTPUT DATA**	Proses input/output data, parameter, informasi
	**PREDEFINED PROCESS (SUB PROGRAM)**	Permulaan sub program/proses menjalankan sub program
	**DECISION**	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	**ON PAGE CONNECTOR**	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	**OFF PAGE CONNECTOR**	Penghubung bagian-bagian flowchart yang berada pada halaman berbeda

BAB III
ANALISA DAN PERANCANGAN SISTEM

Analisis dan perancangan sistem berfungsi untuk mempermudah dalam memahami dan menyusun tahapan selanjutnya yang akan dilakukan, menguraikan dari suatu sistem yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan-permasalahan sehingga ditemukan kelemahan, kesempatan, dan hambatan yang terjadi serta kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya.

3.1 Analisa Sistem

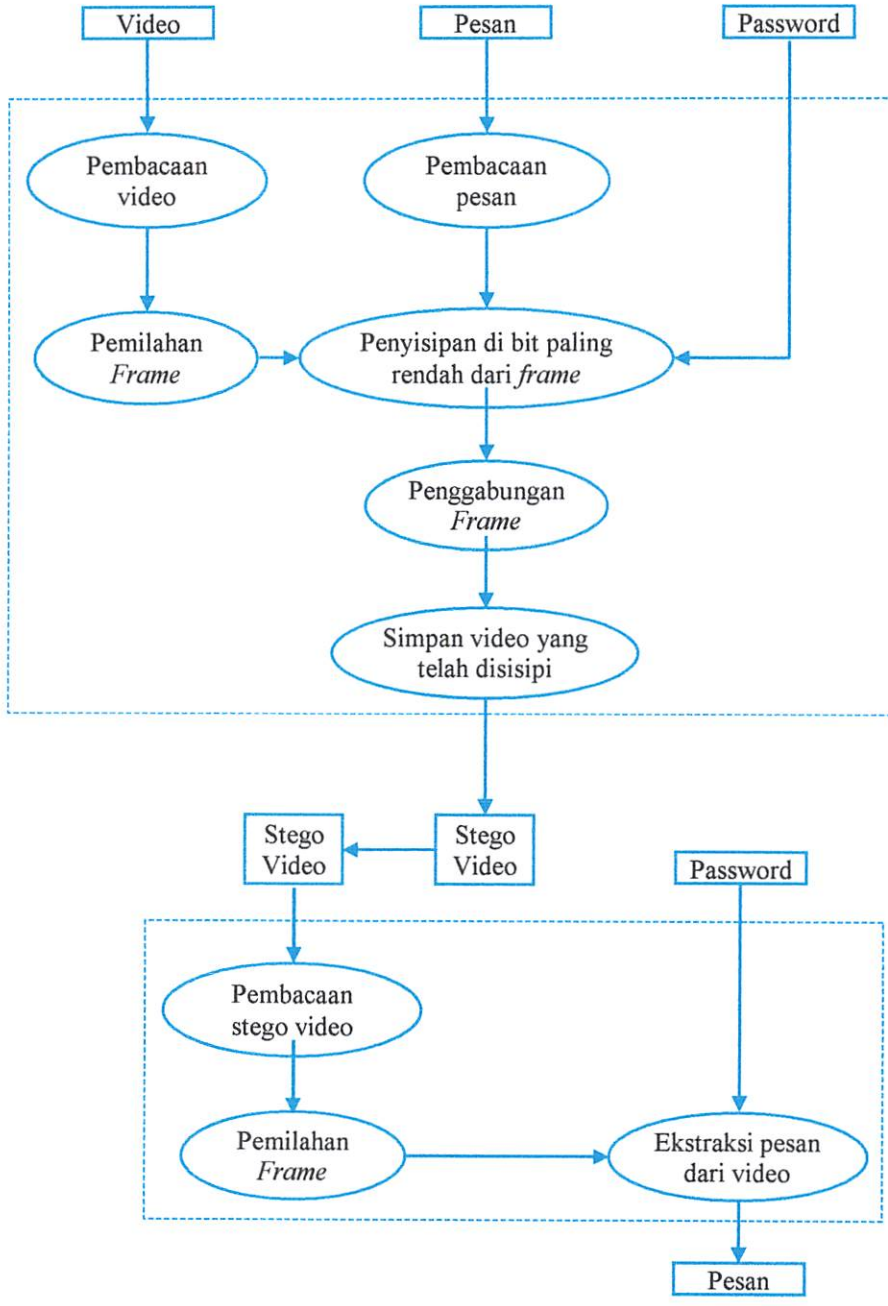
Aplikasi yang akan dibuat pada tugas akhir ini adalah sebuah aplikasi *steganography*dengan video AVI sebagi *cover object* atau wadah, dimana fungsi utama dari aplikasi *steganography* video ini adalah supaya dapat menyisipkan data digital kedalam video, serta dapat mengekstrak kembali data digital tersebut tanpa mngurangi informasi didalamnya.

3.2 Alur Sistem

Objek masukan untuk modul proses penyisipan pesan yaitu video yang berformat *.avi, pesan rahasia bisa berupa file digital yang berekstensi (*.docx), (*.xlsx), (*.txt), dan (*.jpeg), dan juga inputan teks biasa, dan password sebagai pengaman. Sedangkan keluaran dari proses penyisipan ini adalah video yang sudah memiliki pesan rahasia didalamnya.

Sedangkan pada modul proses ekstraksi pesan, objek masukan berupa video yang berformat (*.avi) yang tidak dikompres yang sudah memiliki pesan didalamnya dan juga password yang digunakan pada saat proses penyisipan pesan.

Untuk lebih jelasnya pada Gambar 3.1 akan diberikan arsitektur dari sistem *steganography* video yang akan dibuat :

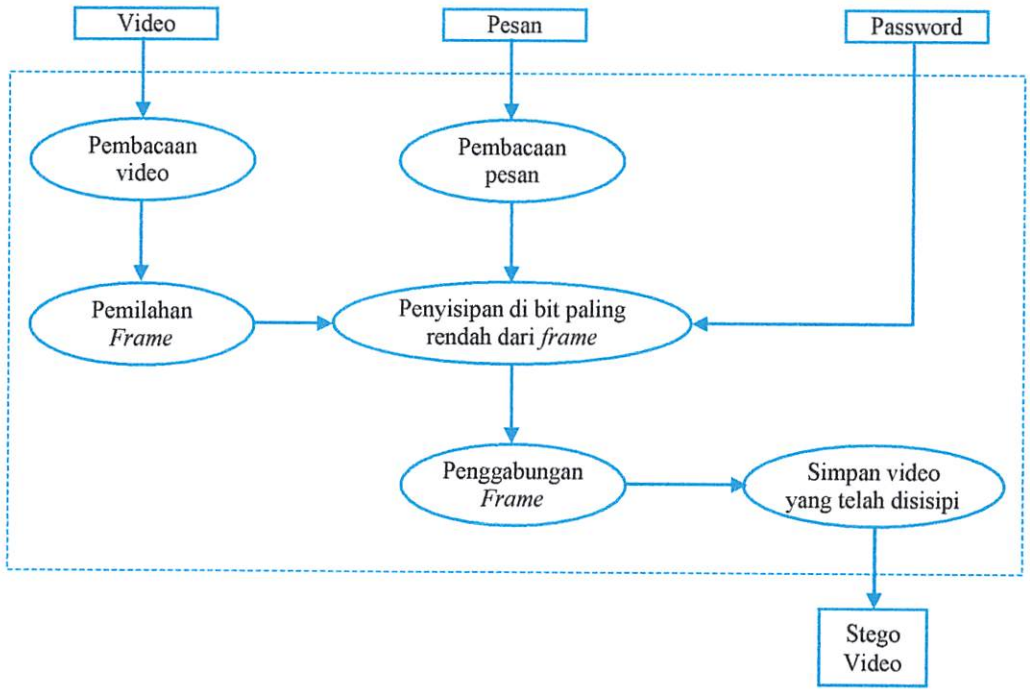


Gambar 3.1 Arsitektur Sistem *Steganography* Video

3.2.1 Sistem Penyisipan Pesan

Sistem untuk penyisipan pesan pada video, membutuhkan masukanberupa video sebagai media penyisipan, pesan yang ingin disisipkan, sertapassword sebagai pengaman. Video yang digunakan sebagai media penyisipan pesan hanya video yang berformat (*.avi) yang tidak di kompresi. Aktifitas yang akan dilakukan pada proses penyisipan pesan ini adalah sebagai berikut :

1. Melakukan pembacaan terhadap file video yang berformat (*.avi), untuk mempersiapkan file video tersebut yang akan digunakan sebagai media penyisipan pesan. File video tersebut diubah menjadi kumpulan *frame-frame*. Setiap *frame* dalam video diubah menjadi sebuah file gambar bitmap (*.bmp).
2. Melakukan pembacaan *byte*terhadap datadigitalatau teks biasa, yang selanjutnya *byte* tersebut di konversi menjadi bilangan biner. Bit-bit dari bilangan biner inilah yang akan disisipkan pada video.
3. Menyisipkan bit-bit file pesan kedalam bit yang paling rendah dari *frame* video yang digunakan sebagai tempat penyisipan yang dihasilkan pada langkah pertama, password disisipkan pada *frame* pertama sedangkan untuk pesan rahasia akan disisipkan pada *frame* berikutnya.
4. Menggabungkan kembali kumpulan *frame-frame* tersebut yang telah disisipi pesan sehingga menjadi video yang mengandung pesan.

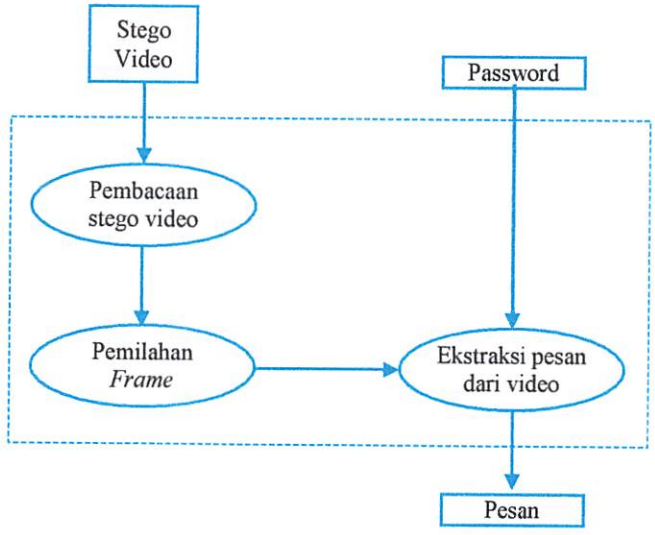


Gambar 3.2DiagramProses Penyisipan Pesan.

3.2.2 Sistem Ekstraksi Pesan

Sistem untuk mengekstraksi pesan pada video memerlukan dua buah masukan yaitu video yang mengandung pesan, serta password yang diinputkan pada saat penyisipan pesan. Aktifitas yang dilakukan pada proses ini adalah sebagai berikut:

1. Melakukan pembacaan terhadap file video yang telah disisipi pesan yang dihasilkan pada proses penyisipan pesan. File video tersebut diubah menjadi kumpulan *frame-frame*. Setiap *frame* dalam video tersebut di ubah menjadi kumpulan file BMP.
2. Memeriksa masukan password apakah sudah sama dengan password yang tersimpan di file BMP pertama. Jika sama maka dilanjutkan ke proses selanjutnya, jika tidak sama maka akan menampilkan pesan kesalahan.
3. Mendeteksi bit data pada *frame* kedua dan seterusnya yang mengandung kode data pesan.
4. Menuliskan bit-bit data yang telah diekstraksi menjadi sebuah file.

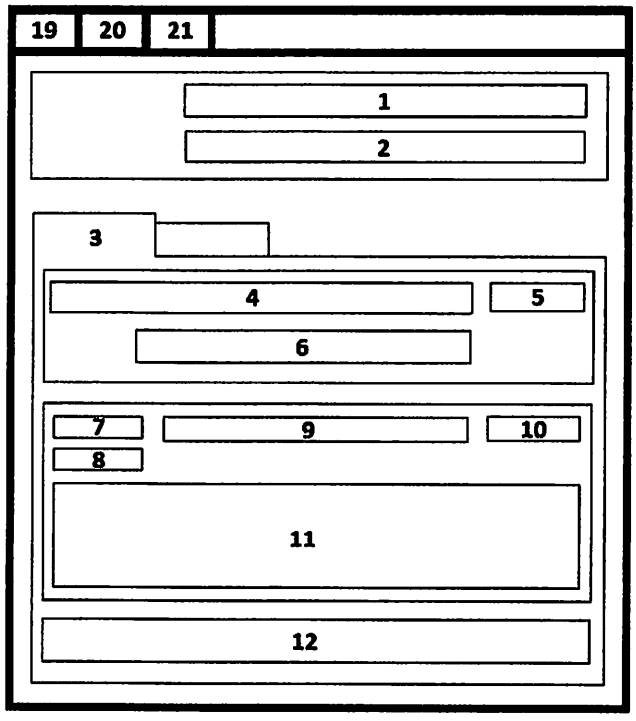


Gambar 3.3DiagramProses Ekstraksi Pesan.

3.3 Desain Antarmuka Aplikasi

Dalam perancangan pembuatan perangkat lunak untuk pengamanan data digunakan program *Microsoft Visual Studio 2008*. Aplikasi ini akan memproses pengamanan data digital dalam media avi. Hasil dari program aplikasi ini dapat untuk menyimpan atau menyisipkan file data digital. Berikut desain antarmuka dari aplikasi yang akan dibuat.

3.3.1 Desain Form Penyisipan pesan



Gambar 3.4 Desain Form Penyisipan

1. TextBox

   Digunakan untuk menginputkan password sebagai pengaman.
2. TextBox

   Digunakan untuk mengonfirmasi dari inputan password pertama.
3. TabControl

   Tab yang digunakan pada saat penyisipan data digital
4. TextBox

   Alamat (*path*) dari video yang akan disisipi pesan

5. Button

   Digunakan untuk mencari file video avi yang akan disisipi pesan

6. Button

   Button ini akan aktif jika sudah terdapat video yang akan disisipi pesan, digunakan untuk membaca atribut dari videotersebut.

7. RadioButton

   Digunakan jika pesan yang akan disisipkan berupa file.

8. RadioButton

   Digunakan jika pesan yang akan disisipkan berupa teks biasa.

9. TextBox

   Alamat (*path*) dari pesan yang berupa file.

10. Button

    Digunakan untuk mencari file yang akan disembunyikan.

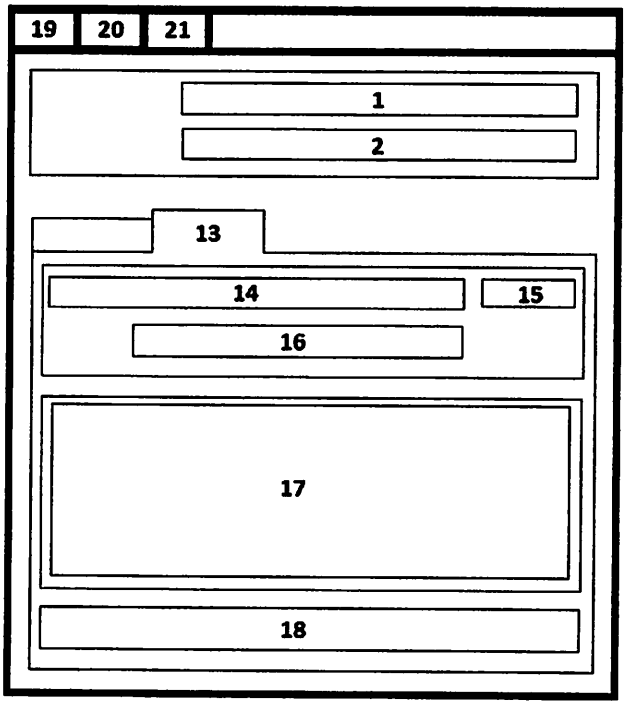
11. RichTextBox

    Digunakan untuk menginputkan pesan yang berupa teks biasa.

12. Button

    Digunakan untuk melakukan proses menyembunyikan data digital.

3.3.2 Desain Form Ekstraksi Pesan



Gambar 3.5 Desain Form Ekstraksi

1. TextBox

   Digunakan untuk menginputkan password sebagai pengaman.

2. TextBox

   Digunakan untuk mengonfirmasi dari inputan password pertama.

13. TabControl

    Tab yang digunakan pada saat ekstraksi data digital.

14. TextBox

    Alamat (*path*) dari video yang sudah terdapat pesan didalamnya.

15. Button

    Digunakan untuk mencari file video avi yang sudah disisipi pesan.

16. Button

    Button ini akan aktif jika sudah terdapat video yang sudah disisipi pesan, digunakan untuk membaca atribut dari video tersebut.

17. RichTextBox

    Digunakan untuk hasil ekstraksi data digital yang berupa teks biasa.

18. Button

    Digunakan untuk melakukan proses ekstraksi data digital.

19. Menu Help

    Digunakan untuk menu bantuan atau cara untuk menggunakan aplikasi ini.

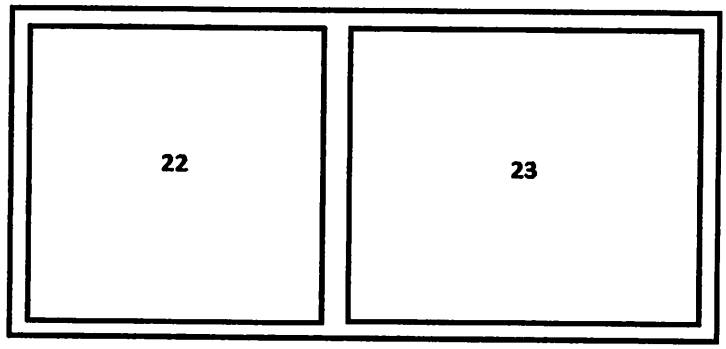
20. Menu About

    Digunakan untuk memanggil form about.

21. Menu Exit

    Digunakan untuk keluar dari aplikasi.

3.3.3 Desain Form Pembacaan Atribut Video



Gambar 3.6 Desain Form Pembacaan Atribut Video

22. Media Player

    Digunakan untuk memainkan video yang dipilih.

23. ListView

    Digunakan untuk menampilkan atribut dari video.

3.4 Perancangan Antarmuka

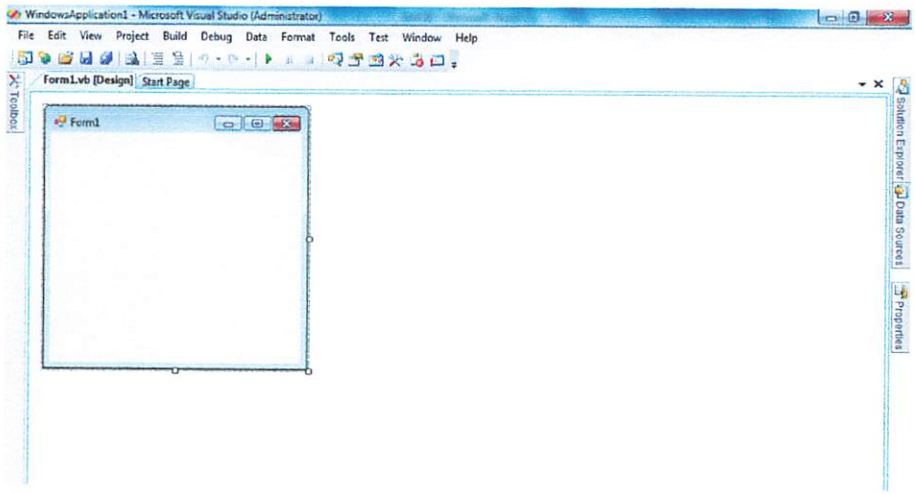
Perancangan antarmuka merupakan langkah awal dari pembuatan aplikasi ini. Mulai dari menentukan gambar, hingga menentukan tata letak dari komponen. Bahasa pemrograman yang dipakai dalam perancangan aplikasi ini menggunakan *Microsoft Visual Studio 2008.*

Dalam perancangan antarmuka ini, ada beberapa tahapan yang perlu dilakukan, yaitu:

3.4.1 Mendesain Tampilan Form Utama

Dalam mendesain tampilan, digunakan komponen-komponen yang ada pada ToolBox dari *Visual Studio 2008.*Berikut langkah-langkah dari pendesainan aplikasi.

- Langkah pertama dari mendesain tampilan form utama adalah membuka program *visual studio 2008.* Setelah dibuka, membuat project baru dengan mengklik *create project.* Setelah memasukkan nama aplikasi maka akan muncul IDE dari *visual studio 2008.*

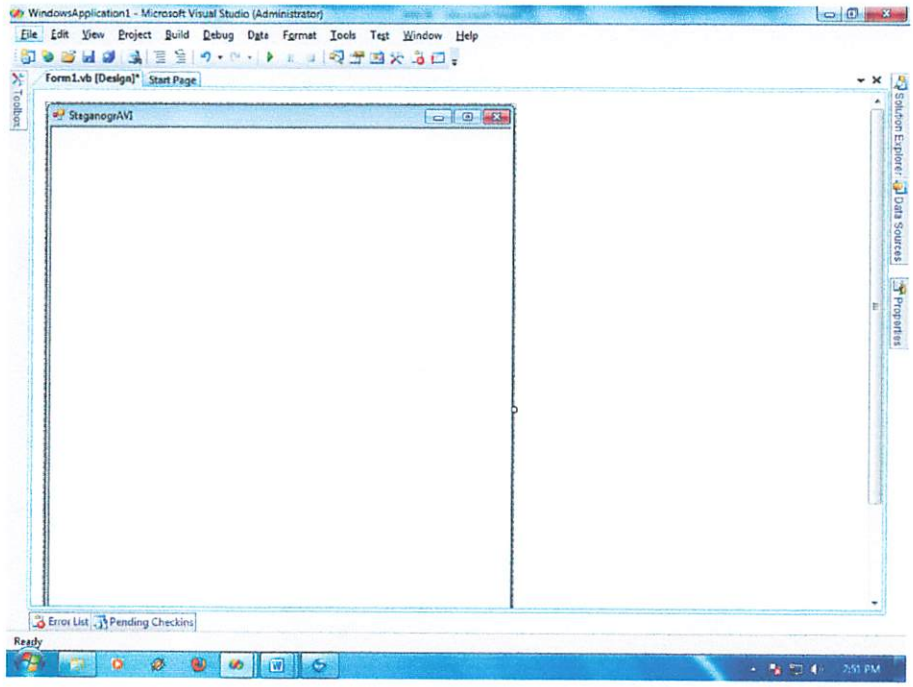


Gambar 3.7 TampilanIDE

- Setelah itu mengatur *properties* dari form dengan value sebagai berikut.

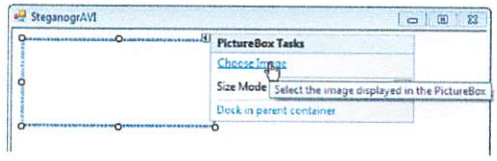
Tabel 3.1 Pengaturan*Properties* Pada Form Aplikasi

Properties	Value
Text	SteganogrAVI
StartPosition	CenterScreen
FormBorderStyle	Fixed3D
BackColor	InactiveBorder
Size	532, 693
MaximizeBox	False



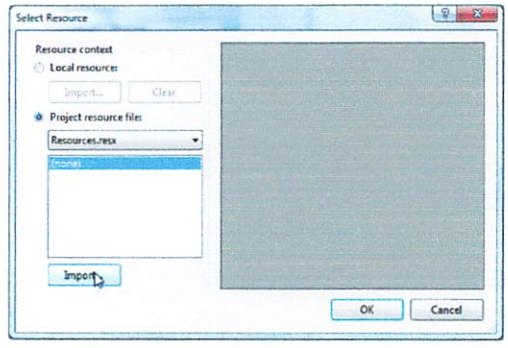
Gambar 3.8 Form Setelah Diatur *Properties*-nya

- Untuk menambahkan gambar pada form menggunakan komponen PictureBox yang ada pada ToolBox ( ), untuk memasukkan gambar klik tanda panah kecil pada atas kanan dari PictureBox, lalu pilih choose image.



Gambar 3.9 Komponen PictureBox Pada Form

Lalu akan muncul window Select Resource, kemudian klik import untuk memilih gambar yang akan dijadikan background.



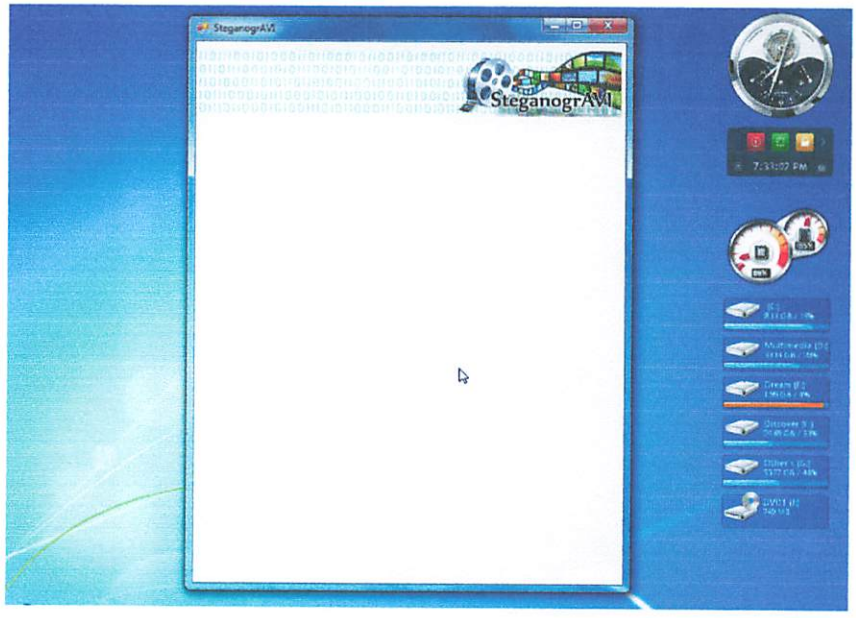
Gambar 3.10Window Select Resource

- Setelah di pilih dilanjutkan dengan mengklik OK pada jendela Select Resource lalu mengatur properties dari PictureBox tersebut dengan value sebagai berikut

Tabel 3.2 Pengaturan*Properties* Pada PictureBox

Properties	Value
BackColor	**Transparent**
Location	2, 10
Size	514, 82

Untuk melihat hasilnya dengan mengklik tombol (▶) yang ada pada ToolBar.

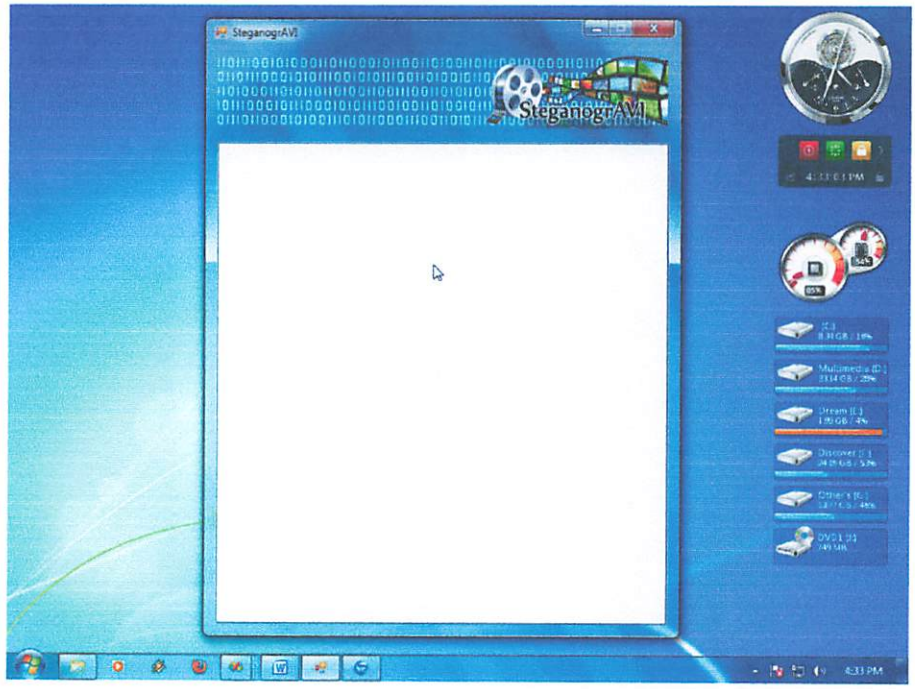


Gambar 3.11 Form Setelah Diberi Gambar

- Supaya tampilan kelihatan menarik, maka diberikan efek glass pada form. Caranya dengan mengimport file rtaGlassEffectLib.dll yang bisa didapatkan melalui internet. Setelah di import kemudian klik dua kali pada form untuk memanggil *event* Load pada form. Pada event tersebut kemudian dimasukkan kode berikut.

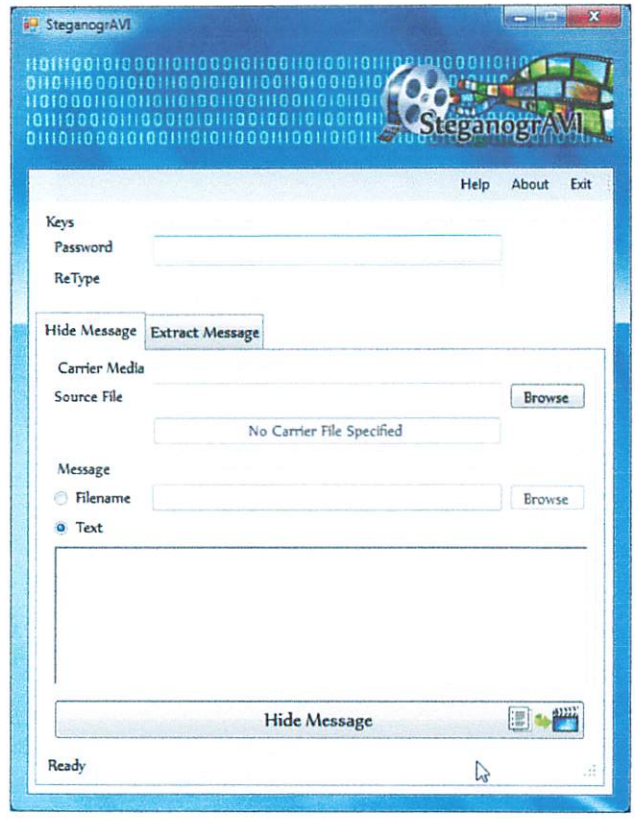
```
Dim glass AsNew rtaGlassEffectsLib.rtaGlassEffect
glass.TopBarSize = 110
glass.LeftBarSize = 6
glass.RightBarSize = 6
glass.BottomBarSize = 6
glass.ShowEffect(Me, PictureBox1)
```

Sehingga bila di jalankan hasilnya seperti Gambar 3.12.

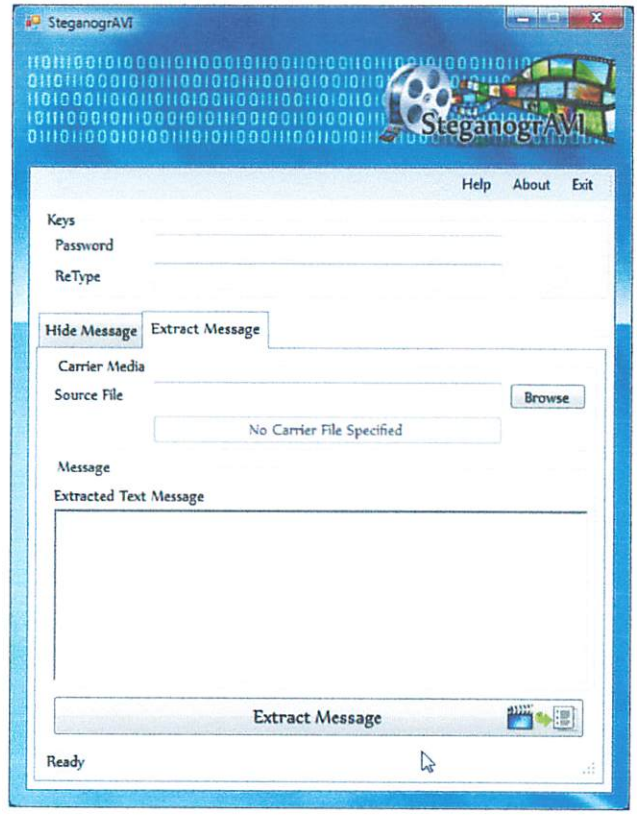


Gambar 3.12 Tampilan Dari Form Setelah Diberi Efek Glass

- Setelah itu dilanjutkan dengan menambahkan komponen-komponen seperti Button, TextBox, dan beberapa komponen lain yang diperlukan, sehingga desain dari antarmuka aplikasi terpenuhi.



Gambar 3.13 Tampilan Form Penyisipan Data Digital

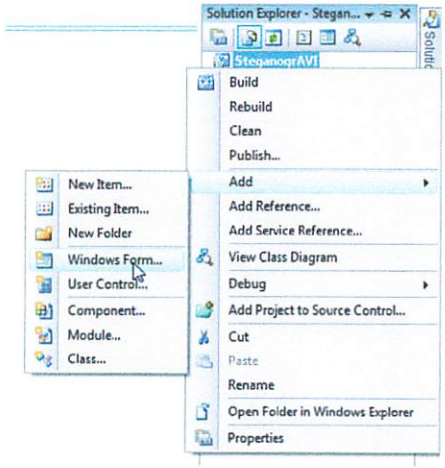


Gambar 3.14 Tampilan Form Ekstraksi Data Digital

3.4.2 Mendesain Tampilan Form VisualDialog

Setelah desain form penyisipan dan ekstraksi data digital selesai dilanjutkan dengan mendesain form yangakan digunakan untuk pembacaan atribut dari video.

- Langkah pertama dengan mengklik kanan pada nama projek yang ada pada Solution Explorer lalu pilih Add, kemudian pilih Windows Form.



Gambar 3.15 Menambahkan Form

- Setelah itu mengatur properties dari form dengan value sebagai berikut

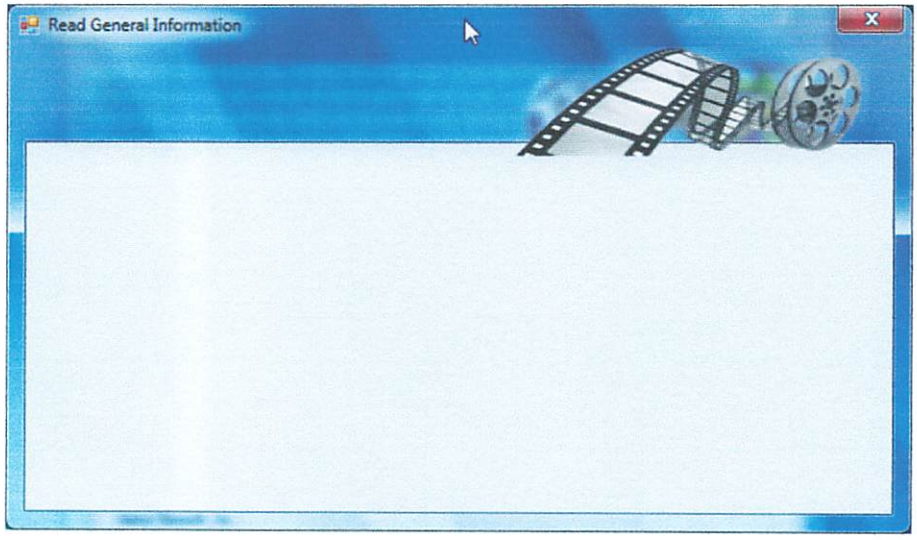
Tabel 3.3 Pengaturan Properties Pada Form

Properties	Value
Name	VisualDialog
Text	Read General Information
StartPosition	CenterScreen
FormBorderStyle	FixedSingle
BackColor	InactiveBorder
Size	626, 374
MaximizeBox	False
MinimizeBox	False

Selanjutnya menambahkan gambar dan juga *event* Load pada form, untuk menambahkan gambar caranya sama dengan sebelumnya. Untuk *event* Load kodenya seperti dibawah ini.

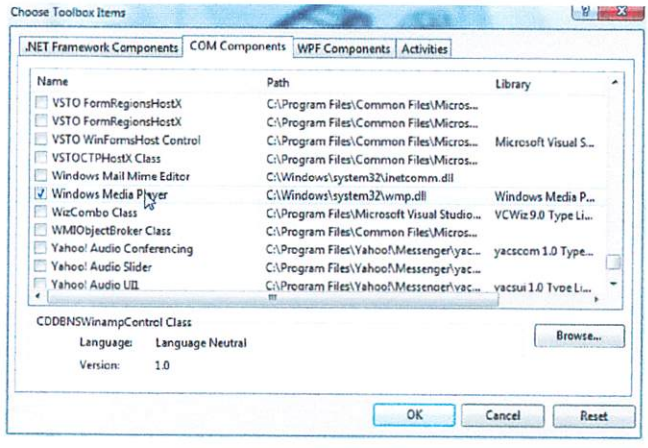
```
Dim glass AsNew rtaGlassEffectsLib.rtaGlassEffect
        glass.TopBarSize = 68
        glass.LeftBarSize = 6
        glass.RightBarSize = 6
        glass.BottomBarSize = 6
        glass.ShowEffect(Me, PictureBox1)
```

Untuk hasilnya seperti berikut ini.



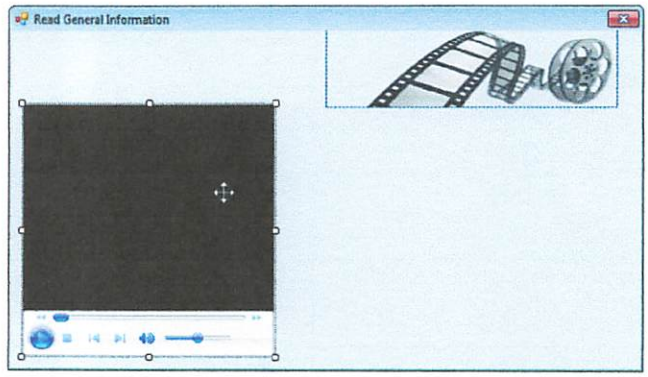
Gambar 3.16 Tampilan Awal Form VisualDialog

- Selanjutnya menambahkan *windows media player,* defaultnya *toolbox* dari visual studio tidak menampilkan komponen ini, untuk menampilkan komponen ini caranya klik kanan pada *toolbox* kemudian pilih *choose item*, setelah itu akan muncul window *Choose Toolbox Item*, kemudian pilih tab *COM Components*, lalu cari *windows media player*, kemudian centang, jika sudah lalu OK.



Gambar 3.17 Memilih Komponen Windows Media Player

- Setelah komponen windows media player ditambahkan, selanjutnya mengatur letak dari komponen tersebut.



Gambar 3.18 Mengatur Letak Komponen Windows Media Player

- Selanjutnya menambahkan ListView yang digunakan untuk list dari atribut video yang digunakan. Berikut pengaturan propertiesnya.

Tabel 3.4 Pengaturan Properties Pada ListView

Properties	Value
Name	listInfo
View	Details
GridLines	True
Size	326, 184
Location	10, 19

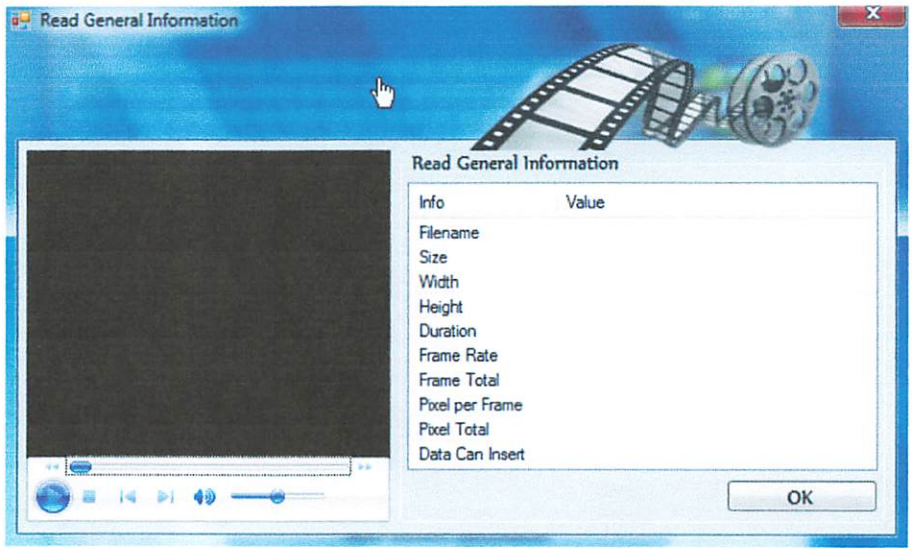
Setelah itu menambahkan 2 kolom dan juga item pada listview, caranya dengan mengklik tanda panah pada kanan atas dari listview. Berikut kolom dan item yang digunakan untuk pembacaan atribut

Kolom

1. Info
2. Item

Item

1. Filename
2. Size
3. Width
4. Height
5. Duration
6. Frame Rate
7. Frame Total
8. Pixel per Frame
9. Pixel Total
10. Data Can Insert



Gambar 3.19 Tampilan Form VisualDialog

3.5 Perancangan Sistem

Perancangan perangkat lunak disini meliputi dua bagian pokok yaitu proses penyisipan dan proses ekstraksi. Agar kedua proses tersebut dapat berjalan, maka dibutuhkan beberapa *class* yang berguna untuk mendukung kedua proses tersebut. Berikut akan dijelaskan fungsi dari berapa *class* tersebut, berikut kedua proses pada perancangan perangkat lunak.

3.5.1 Pembuatan Class Avi

Untuk mengakses file avi supaya dapat diproses tiap *frame*-nya dibutuhkan sebuah komponen, yang mana komponen ini berfungsi untuk mengakses file avi yang akan digunakan sebagai *cover object* pada perangkat lunak ini. Sayangnya, komponen tersebut tidak disediakan oleh *Visual Studio 2008*.Untunglah, windows menyediakan fungsi untuk melakukan hal ini. Fungsi untuk melakukan akses ke file avi, seperti membaca ataupun membuat file avi terdapat pada file avifil32.dll yang terdapat dalam folder sistem32 pada *windows*. Maka dari itu *class* Avi dibuat, dimana *class* ini merupakan seperangkat *method* yang digunakan untuk mengimport fungsi dari file avifil32.dll.Berikut adalah potongan kode dari *class* avi.

```
'menginisialisasi AVI Library
<DllImport("avifil32.dll")> _
PublicSharedSub AVIFileInit()
EndSub

'membuka file AVI
<DllImport("avifil32.dll", PreserveSig:=True)> _
PublicSharedFunction AVIFileOpen(ByRef ppfile AsInteger, _
ByVal szFile AsString, ByVal uMode AsInteger, _
ByVal pclsidHandler AsInteger) AsInteger
EndFunction
   ~
   ~
   ~
'Menutup file AVI yang telah dibuka
<DllImport("avifil32.dll")> _
PublicSharedFunction AVIFileRelease(ByVal pfile AsInteger)
As _ Integer
EndFunction

'menutup AVI Library
<DllImport("avifil32.dll")> _
PublicSharedSub AVIFileExit()
EndSub
```

3.5.2 Pembuatan Class AviReader

Pada file avi dapat berisi banyak streamdari empat jenis yang berbeda, yaitu stream video, stream audio, stream teks, dan stream midi. Pada penelitian ini hanya akan dilakukan pada stream video saja. Kegunaan dari *class* AviReader adalah memanggil fungsi-fungsi yang terdapat pada *class* Avi untuk mendapatkan stream video yang nantinya akan digunakan pada tahap penyisipan. Untuk dapat mengakses (membaca) stream video dari file avi, diperlukan langkah-langkah sebagai berikut:

- Inisialisasi file avi dengan menggunakan fungsi AVIFileInit. Fungsi ini akan menaikkan jumlah referensi pada *library*.

```
'Inisialisasi AVI Library
Avi.AVIFileInit()
```

- Membuka file avi dengan fungsi AVIFileOpen. Pada *class* ini mode yang digunakan adalah OF_SHARE_DENY_WRITE. Dimana mode ini berfungsi untuk membuka file avi dan menolak akses menulis pada file tersebut.

```
'Membuka file Avi
Avi.AVIFileOpen(aviFile, fileName, _
Avi.OF_SHARE_DENY_WRITE, 0)
```

- Membuka *stream* untuk file avi yang sudah dibuka dengan fungsi AVIStreamGetFrameOpen.

```
'Membuka stream
Avi.AVIStreamGetFrameOpen(aviStream, bih)
```

- Mengambil data (*frame*) dari file avi secara berkala dengan fungsi AVIStreamGetFrame.

```
'Mengambil frame
Avi.AVIStreamGetFrame(getFrameObject, firstFrame + _
position)
```

- Setelah selesai pengaksesan file avi, maka *frame* yang dibuka harus ditutup dengan fungsi AVIStreamGet*Frame*Close. *Stream* juga harus ditutup dengan fungsi AVIStreamRelease. File avi harus ditutup dengan fungsi AVIFileRelease. Dan yang terakhir memanggil fungsi AVIFileExit untuk mengurangi jumlah referensi pada *library*.

```
'Menutup semua stream, file dan library dari AVI
PublicSub Close()
If getFrameObject <> 0 Then
   Avi.AVIStreamGetFrameClose(getFrameObject)
   getFrameObject = 0
EndIf
If aviStream <> IntPtr.Zero Then
   Avi.AVIStreamRelease(aviStream)
   aviStream = IntPtr.Zero
EndIf
If aviFile <> 0 Then
   Avi.AVIFileRelease(aviFile)
   aviFile = 0
EndIf
Avi.AVIFileExit()
EndSub
```

3.5.3 Pembuatan Class AviWriter

Kegunaan *class* AviWriter disini adalah untuk membuat file avi baru yang mengacu pada informasi file avi *original*yang didapatkan pada *class* AviReader, seperti *framerate*dan gambar-gambar yang telah diekstrak yang nantinya akan digunakan pada file avi baru. Untuk membuat file avi baru deperlukan langkah-langkah sebagai berikut:

- Inisialisasi file avi dengan menggunakan fungsi AVIFileInit. Fungsi ini akan menaikkan jumlah referensi pada *library*.

```
'Inisialisasi AVI Library
Avi.AVIFileInit()
```

- Membuat file avi dengan fungsi AVIFileOpen. Pada *class* ini mode yang digunakan adalah OF_WRITE Or OF_CREATE. Dimana mode ini berfungsi untuk membuat file avi baru yang didalamnya sudah terdapat data digital yang telah disembunyikan.

```
'membuat file avi baru
Avi.AVIFileOpen(aviFile, fileName, Avi.OF_WRITE Or _
Avi.OF_CREATE, 0)
```

- Membentuk *stream* dengan menggunakan AVIFileCreateStream.

```
'Membentuk stream
Avi.AVIFileCreateStream(aviFile, aviStream, strhdr)
```

- Setelah *stream* dibuat dilanjutkan dengan mengisi informasi dalam stream tersebut dengan fungsi AVIStreamSetFormat.

```
'Membentuk stream
Avi.AVIStreamSetFormat(aviStream, 0, bi, _
Marshal.SizeOf(bi))
```

- Setelah itu dilakukan penyimpanan data *stream* tersebut dengan fungsi AVIStreamWrite.

```
'Menyimpan data stream
Avi.AVIStreamWrite(aviStream, countFrames, 1, _
bmpDat.Scan0, CType(stride * height, Int32), 0, _
0, 0)
```

- Setelah selesai pembuatan file avi, maka *stream* harus ditutup dengan fungsi AVIStreamRelease. File avi harus ditutup dengan fungsi AVIFileRelease. Dan yang terakhir memanggil fungsi AVIFileExit untuk mengurangi jumlah referensi pada *library*.

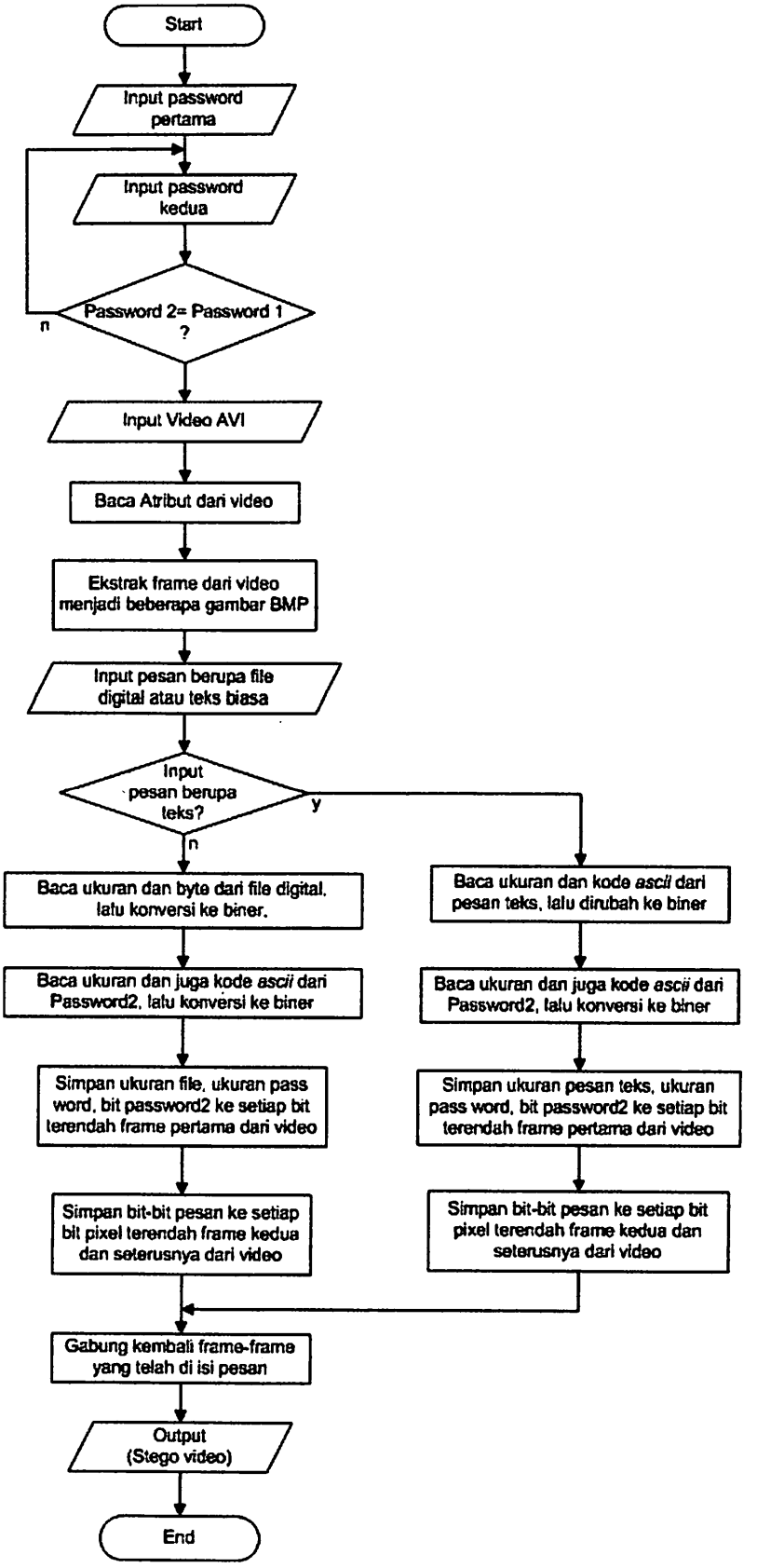
```
'Menutup semua stream, file dan library dari AVI
PublicSub Close()
If aviStream <> IntPtr.Zero Then
   Avi.AVIStreamRelease(aviStream)
   aviStream = IntPtr.Zero
EndIf
If aviFile <> 0 Then
   Avi.AVIFileRelease(aviFile)
   aviFile = 0
EndIf
Avi.AVIFileExit()
EndSub
```

3.5.4 Tahap Penyisipan

Pada proses penyisipan ini landasan yang penulis gunakan dalam perancangan program yaitu implementasi teknik steganografi pada file video. Dimana untuk file video yang dipilih adalah video dengan format avi 24-bit color. Untuk penyisipan pesannya menggunakan teknik steganografi yaitu metode *Least Significant Bit* (LSB).

Di dalam proses penyisipan ini terdapat beberapa proses seperti pengambilan video, proses pengambilan data informasi atau atribut video meliputi total *frame*, durasi, resolusi dan juga *framerate* dari file avi tersebut dan juga termasuk proses ambil citra bitmap dari masing-masing *frame* video avi, proses pembacaan data digital termasuk didalamnya password sebagai pengaman yang dirubah menjadi binary, proses penyisipan dan terakhir proses penggabungan dari citra bitmap menjadi file video avi. Dimana untuk masing-masing proses ini akan dijelaskan lebih lanjut pada sub bab berikutnya.

Pada Gambar 3.20akan diberikan diagram alir (*flowchart*) dari proses penyisipan pesan secara keseluruhan.



Gambar 3.20*Flowchart*Proses Penyisipan

Penjelasan dari diagram alir tersebut adalah sebagai berikut:

a. Proses input file video

Proses input file video avi dilakukan untuk memilih file yang ingin kita tanamkan pesan rahasia didalamnya atau sebagai cover objek. Pada proses ini setelah file avi di pilih selanjutnya dibaca atribut dari file avi tersebut.

b. Proses pembacaan atribut video

Pada proses pembacaan atribut video Avi, proses yang dilakukan meliputi pembacaan ukuran (*size*), pembacaan durasi, pembacaan resolusi, pembacaan*framerate*, serta atribut lainnya dari file avi tersebut.

Setelah file avi di input, selanjutnya file avi tersebut akan diinisialisai terlebih dahulu dengan menggunakan komponen avifil32.dll, dalam hal ini dengan memanggil fungsi dari *class* Avi yang ada pada system operasi windows agar file avi yang kita buka dapat dikenali dan dapat ditampilkan. Dari komponen ini juga atribut dari file avi dapat di baca.

c. Proses ekstraksi *frame* dari video

Disinilah kegunaan dari *class* AviReader. Pada proses ekstraksi *frame* dari video, proses yang dilakukan meliputi proses penyalinan *frame-frame* avi ke dalam citra dengan ekstensi format(*.bmp). Pada proses ini penyalinan citra dilakukan satu persatu, sehingga disini akan terdapat pengulangan untuk proses penyalinan sebanyak jumlah *frame* keseluruhan yang didapat pada file avi. Pada proses penyalinannya gambar dalam citra dengan ekstensi format (*.bmp) disimpan sementara dalam folder dengan nama tempBitmap sampai gambar tersebut selesai diproses selanjutnya.

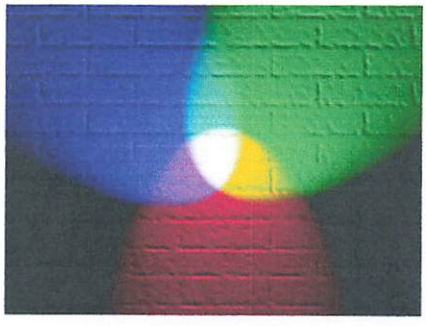
Setelah didapat citra image berupa bitmap dilakukan pembacaan *pixel*.Pembacaan *pixel* bertujuan untuk mendapatkan informasi yang terdapat pada citra.

Sebuah *pixel* adalah sampel dari pemandangan yang mengandung intensitas citra yang dinyatakan dalam bilangan bulat. Indeks baris dan kolom (x,y) dari sebuah *pixel*dinyatakan dalam bilangan bulat. *Pixel* (0,0) terletak pada sudut kiri atas pada citra, indeks x bergerak kekanan dan indeks y bergerak kebawah.

Setiap citra digital memiliki ukuran resolusi, resolusi merupakan suatu besaran yang biasanya dinyatakan dalam banyaknya titik atau *pixel* penyusunnya.Resolusi merupakan kerapatan *pixel* dari citra yang berrati banyaknya jumlah *pixel* yang menyusun citra tersebut.Secara keseluruhan, resolusi citra ditentukan oleh M x N *pixel* penyusunnya.

Pemilihan ukuran resolusi citra yang digunakan sangat berpengaruh pada kecepatan proses. Mengingat pada pengolahan citra proses dilakukan pada tiap-tiap *pixel*. Sehingga semakin besar *pixel* penyusunnya semakin banyak *pixel*yang harus diproses, dengan demikian semakin lambat pemrosesan pada citra.

Suatu *pixel* berisi informasi derajat keabuan (*gray level*) atau warna suatu citra.Model warna yang digunakan adalah warna merah (*red*), hijau (*green*), dan biru (*blue*) atau biasanya disebut dengan model warna RGB. Setiap warna yang lain dapat dihasilkan dari pencampuran ketiga warna tersebut, misalnya warna kuning dapat diperoleh dari warna merah dan hijau. Sehingga dari tiap-tiap *pixel* akan didapatkan tiga warna dasar penyusun tersebut. Nilai-nilai yang telah didapatkan itulah yang dapat diolah selanjutnya. Sehingga proses akses *pixel*(ambil dan simpan) ini merupakan awal dari pengolahan citra.



Gambar 3.21ModelWarna RGB

Nilai *pixel* pada citra digital diwakili dengan satu bilangan bulat, yaitu berkisar antara 0 sampai 255 pada citra dengan derajat keabuan 8 bit. Nilai tersebut merepresentasikan gradasi, kepekatan atau tingkat warna.Nilai 0 merepresentasikan warna hitam dan 255 merepresentasikan warna putih. Pada citra digital berwarna yang model warna RGB dengan derajat keabuan 8 bit tiap *pixel*-nya berisikan informasi warna merah, hijau, dan biru yang masing-masing memiliki derajat keabuan 8 bit, dengan kata lain memerlukan 24 bit utuk menyimpan data warna.

d. Proses pembacaan data digital

Pada proses ini terdapat dua kondisi yaitu apakah data digital berupa file digital atau hanya berupa inputan teks biasa.

Pada data digital dapat berupa file (*.docx), (*.xlsx), (*.txt), dan (*.jpeg), atribut yang dibaca meliputi *byte* penyusun dari file, dan ukuran dari file itu sendiri.*Byte* yang menyusun file tersebut, nantinya di konversi menjadi biner yang nantinya akan disisipkan pada *frame* video.

Sedangkan untuk data digital berupa teks biasa, pembacaan yang dilakukan hanya meliputi kode *ascii* dari teks tersebut, dan panjang (ukuran) dari teks tersebut. Dimana kode *ascii* yang didapat akan dirubah menjadi biner yang akan disisipkan pada *frame* video

e. Proses pembacaan password

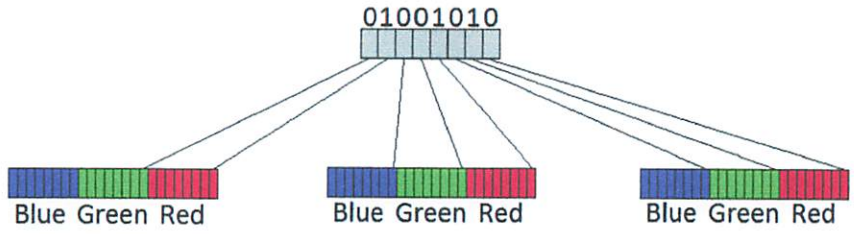
Pada proses ini,password yang diinputkan akan di konversi menjadi kode *ascii*yang selanjutnya dirubah menjadi biner. Bit-bit tersebut nantinya akan disisipkan pada *frame*pertama dari video.Termasuk panjang dari password tersebut.

f. Proses penyisipan data

Proses selanjutnya yang akan dilakukan adalah proses penyisipan pesan. File data digial yang sudah dirubah ke dalam bentuk biner akan disisipkan ke masing-masing bit terakhir dari RGB pada setiap *frame*.

Pada proses ini adalah melakukan proses pengulangan sebanyak jumlah *frame* yang dibutuhkan untuk proses penyisipan. Pada proses pengulangan ini bertujuan untuk memindahkan data biner dari satu *frame* ke *frame* yang lainnya jika pada *frame* sebelumnya jumlah data pesannya telah mencapai batas ukuran yang diperlukan dalam satu *frame*.

Selanjutnya dilakukan proses penyisipan di tiap-tiap bit pada tiap-tiap *pixel* citra dari masing-masing *frame* video avi. Dimana untuk proses ini dilakukan dengan operasi modulus, yaitu sisa dari pembagian. Untuk lebih jelasnya akan diberikan Gambar 3.22.



Gambar 3.22UrutanPenyisipan Bit Data Pesan Pada *Pixel* Gambar

Pada pesan yang berupa inputan teks biasa, bit-bit dari password, ukuran dari password, dan ukuran teks, disisipkan pada *frame* pertama dari video avi. Untuk bit-bit password disisipkan mulai *pixel* pertama dari *frame* pertama. Sedangkan untuk ukuran password dan ukuran file disembunyikan dengan merubah 2 *pixel* paling bawah kanan. Untuk bit-bit dari pesan sendiri akan disisipkan pada *frame* kedua dan seterusnya sampai berapa *frame* yang dibutuhkan untuk disisipi pesan tersebut.

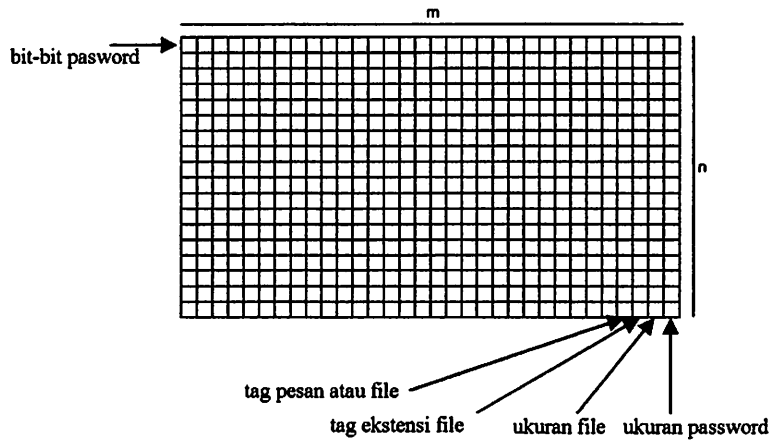
Sedangkan pada pesan yang berupa file digital, caranya hampir sama dengan pesan yang berupa teks biasa, yang membedakan adalah pemberian tag untuk membedakan pada tahap ekstraksi apakah pesan berupa file digital atau teks biasa. Pemberian tag ini di dilakukan dengan merubah pixel warna merah ke-4 dari paling kanan-bawah pada frame pertama, pada pesan yang berupa teks biasa pemberian tag dilakukan dengan merubah pixel tersebut dengan angka 0, sedangkan pada file digital dengan merubah pixel tersebut dengan angka 1.

Pada file digital itu sendiri terdapat beberapa ekstensi yang akan disisipkan pada video, yaitu (*.docx), (*.xlsx), (*.txt), (*.jpeg), dan (*.pdf). Supaya pada tahap ekstraksi, perangkat lunak dapat membedakan ekstensi (format) dari file tersebut, maka dilakukan pemberian tag juga pada frame pertama tersebut. Dengan cara merubah pixel warna merah ke-3 dari paling bawah kanan. Untuk file digital dengan ekstensi (*.docx) tag yang diberikan berupa nilai 1, untuk ekstensi (*.xlsx) berupa nilai 2, untuk ekstensi (*.jpeg) berupa nilai 3, dan untuk ekstensi (*.txt) berupa nilai 4.

Tabel 3.5 Pemberian Tag Pada Pesan.

Jenis Pesan	Pixel ke- (dari kanan-bawah)	Tag
Teks	4	0
File	4	1
(*.docx)	3	1
(*.xlsx)	3	2
(*.jpeg)	3	3
(*.txt)	3	4

Dengan demikian frame pertama dari video merupakan kumpulan informasi dari pesan yang telah disisipkan.



Gambar 3.23 Struktur penyisipan pesan pada frame pertama

Sedangkan untuk frame kedua dan seterusnya prosesnya yaitu menggunakan operasi modulus, dimana operasi modulus ini diterapkan pada panjang dari ukuran frame-frame tersebut untuk membagi bit-bit dari data digital kedalam setiap LSB dari *bytepixel* penyusun dari video. Berikut contoh perhitungan manual dalam proses ini.

Dimisalkan byte pixel penyusun dari sebuah frame video pada posisi (kolom,baris) adalah sebagai berikut:

Tabel 3.6 Byte Penyusun Dari Pixel Frame Video

	Pixel position(0,0)	Pixel position(1,0)	Pixel position(2,0)
Pixel merah	198	185	186
Pixel hijau	175	160	80
Pixel biru	45	80	60

Setelah itu byte-byte tersebut dirubah menjadi biner menjadi seperti berikut:

Tabel 3.7 Bit Penyusun Dari Pixel Frame Video

	Pixel position(0,0)	Pixel position(1,0)	Pixel position(2,0)
Pixel merah	11000110	10111001	10111010
Pixel hijau	10101111	10100000	01010000
Pixel biru	00101101	01010000	00111100

Dimisalkan byte dari sebuah data yang menyusunnya adalah 135, byte tersebut kemudian dirubah ke biner menjadi 10000111.

Setelah itu didapatkan index dari masing-masing bit tersebut dan dilakukan operasi modulus terhadap bit pixel dan mensubtitusi bit terakhir dari pixel tersebut dengan bit-bit penyusun dari data. Index dimulai dari 0.

```
Jika kolom modulus 3 = 0 maka
Index ke-7 dari pixel merah = index ke-0 dari bit data
   Index ke-7 dari pixel hijau = index ke-1 dari bit data
   Index ke-7 dari pixel biru = index ke-2 dari bit data
Jika kolom modulus 3 = 1 maka
   Index ke-7 dari pixel merah = index ke-3 dari bit data
   Index ke-7 dari pixel hijau = index ke-4 dari bit data
   Index ke-7 dari pixel biru = index ke-5 dari bit data
Jika kolom modulus 3 = 2 maka
   Index ke-7 dari pixel merah = index ke-6 dari bit data
   Index ke-7 dari pixel hijau = index ke-7 dari bit data
```

Sehingga bit-bit dari pixel penyusun frame akan berubah menjadi seperti dibawah ini:

Tabel 3.8 Bit Penyusun Dari Pixel Frame Video Setelah Disisipi

	Pixel position(0,0)	Pixel position(1,0)	Pixel position(2,0)
Pixel merah	1100011**1**	1011100**0**	1011101**1**
Pixel hijau	1010111**0**	1010000**0**	0101000**1**
Pixel biru	0010110**0**	0101000**1**	00111100

Proses ini diulang dengan batas ukuran dari data tersebut.

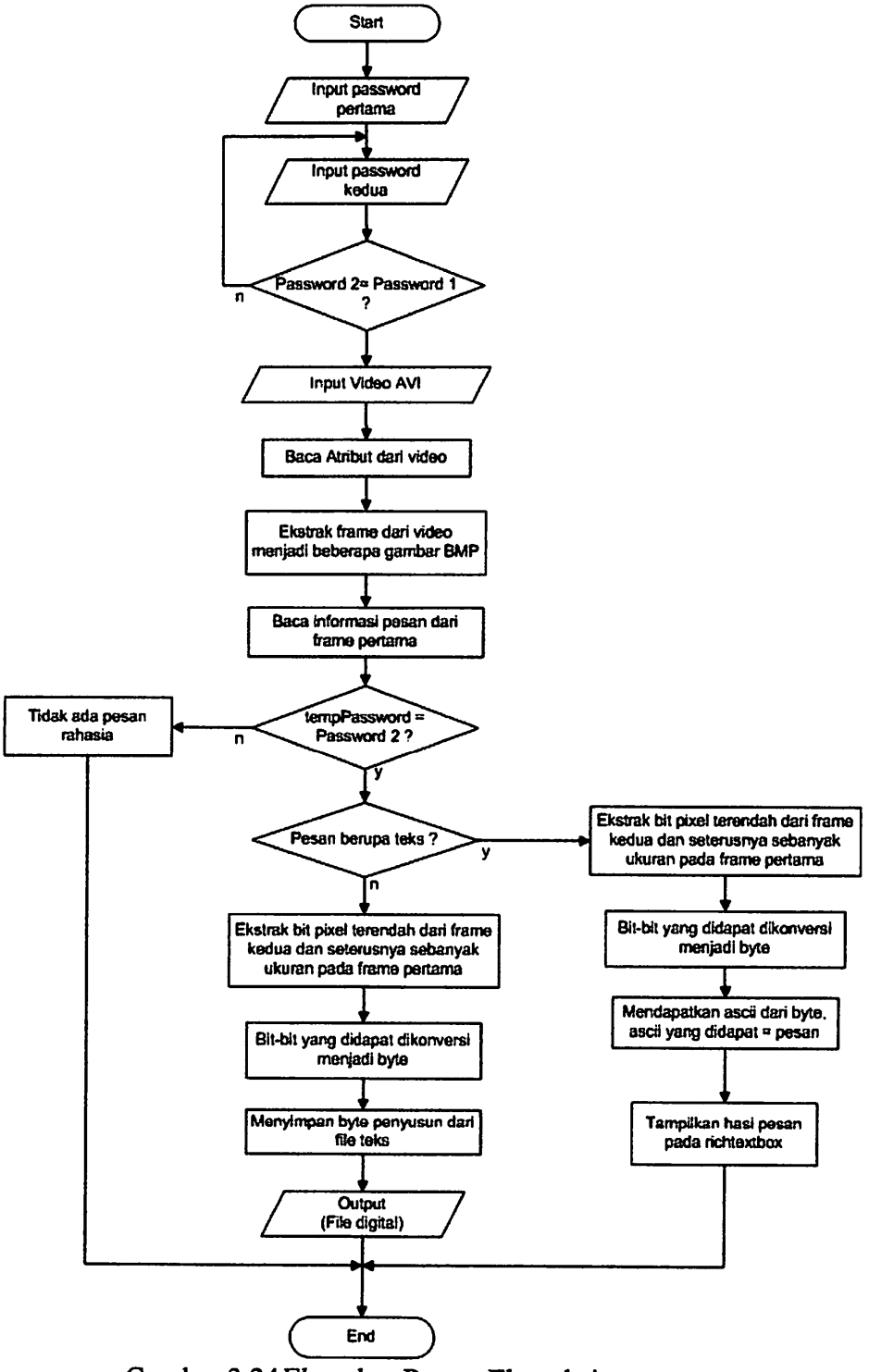
g. Proses penggabungan *frame* video

Proses terakhir adalah proses penyimpanan atau penggabungan gambar bitmap (*frame*) yang telah disisipi pesan ke dalam format avi.

Pada proses ini dibutuhkan *class* AviWriter dimana dalam *class* ini terdapat fungsi yang dapat mengubah file gambar bitmap menjadi file video dengan format avi. Pada *class* ini dilakukan pengulangan untuk mengubah gambar bitmap menjadi file video sebanyak jumlah *total frame*awal, dengan mengacu pada *framerate*awal dari file video. Sehingga video yang berisi pesan yang dihasilkan sama dengan file video sebelum diisi pesan.

3.5.5 Tahap Ekstraksi

Pada tahap ekstraksi ini proses yang dilakukan hampir sama dengan proses penyisipan pesan, yaitu meliputi proses pengambilan video, proses pembacaan atribut video dan juga termasuk proses ekstraksi *frame* dari video. Untuk lebih jelasnya akan diberikan diagram alir untuk proses ekstraksi.



Gambar 3.24*Flowchart*Proses Ekstraksi

Penjelasan dari diagram alir tersebut adalah sebagai berikut:

a. Proses ambil video

Proses ambil file avi dilakukan untuk memilih file yang sudah terdapat pesan rahasia didalamnya. Pada proses ini setelah file avi di pilih selanjutnya dibaca atribut dari file avi tersebut.

b. Proses pembacaan atribut video

Pada proses pembacaan atribut video Avi, proses yang dilakukan meliputi pembacaan ukuran (*size*), pembacaan durasi, pembacaan resolusi, pembacaan *framerate*, serta atribut lainnya dari file avi tersebut.

Setelah file avi di input, selanjutnya file avi tersebut akan diinisialisai terlebih dahulu dengan memanggil komponen avifil32.dll yang ada pada system operasi windows agar file avi yang kita buka dapat dikenali dan dapat ditampilkan. Dari komponen ini juga atribut dari file avi dapat di baca.

c. Proses ekstraksi *frame* dari video

Disinilah kegunaan dari *class* AviReader. Pada proses ekstraksi *frame* dari video, proses yang dilakukan meliputi proses penyalinan *frame-frame* avi ke dalam citra dengan ekstensi format(*.bmp). Pada proses ini penyalinan citra dilakukan satu persatu, sehingga disini akan terdapat pengulangan untuk proses penyalinan sebanyak jumlah *frame* keseluruhan yang didapat pada file avi. Pada proses penyalinannya gambar dalam citra dengan ekstensi format (*.bmp) disimpan sementara dalam folder dengan nama tempBitmap sampai gambar tersebut selesai diproses selanjutnya.

Setelah didapat citra image berupa bitmap dilakukan pembacaan *pixel*.Pembacaan *pixel* bertujuan untuk mendapatkan informasi yang terdapat pada citra.

d. Proses ekstraksi pesan

Pada proses ini yang pertama dilakukan adalah mengecek pada *frame* pertama terlebih dahulu. Apakah password yang telah disipkan pada frame pertama sama dengan password yang diinputkan pada tahap penyisipan. Jika tidak sama maka akan keluar pesan error yang berbunyi tidak ditemukan pesan pada video tersebut. Selanjutnya jika password yang diinputkan sama, langkah selanjutnya adalah mengecekapakah pada *frame* pertama tersebut terdapat tag pesan pada pixel warna merah ke-4 daripaling kanan-bawah berupa nilai 0 atau 1. Jika tag yang didapat berupa nilai 0 maka pesan yang disisipkan berupa teks biasa, sedangkan jika tag yang didapat berupa nilai 1 maka pesan yang disisipkan berupa file digital.Jika bentuk pesan yang disisipkan sudah diketahui maka ke langkah selanjutnya.

Jika pesan yang disisipkan berupa file digital, dilakukan pengecekan tag lagi pada pixel warna merah ke-3 dari paling kanan-bawah. Pengecekan ini berfungsi untuk mengetahui apakah ekstensi (format) yang ada pada file tersebut. Jika tag yang diberikan bernilai 1 maka file tersebut berformat (*.docx), jika 2 maka (*.xlsx), jika 3 maka (*.jpeg), dan jika 4 file tersebut berformat (*.txt). Setelah itu proses ekstraksi data digital pada *frame* kedua dan seterusnya dilakukan. Proses ekstraksi dilakukan dengan cara mengambil bit-bit terakhir dari *pixel* RGB sejumlah ukuran yang telah disimpan pada *pixel* paling kanan-bawah pada *frame* pertama. Apabila sudah didapatkan, bit-bit tersebut kemudian di konversi menjadi byte-byte yang menyusun sebuah file digital. Setelah itu dilakukan penyimpanan dari semua byte-byte penyusun tadi untuk menghasilkan sebuah pesan rahasia berupa file digital.

Sedangkan untuk data digital berupa teks biasa, dilakukan proses ekstraksi terlebih dahulu dari *frame* kedua dan seterusnya sejumlah ukuran yang telah disimpan pada *pixel* paling bawah kanan pada *frame* pertama. Apabila sudah didapatkan, bit-bit tersebut kemudian dikonversi menjadi byte yang merupakan kode *ascii* dari

pesan teks.Selanjutnya masing-masing kode *ascii* tersebut didapatkan nilai karakternya. Karakter-karakter tersebut merupakan pesan teks yang selanjutnya ditampilkanpada rich tekt box pada perangkat lunak.

Untuk melakukan prosesini caranya hampir sama dengan proses pada penyisipan.

Dimisalkan byte pixel penyusun dari sebuah frame video yang telah disisipi pesan pada posisi (kolom,baris) adalah sebagai berikut:

Tabel 3.9 Byte Penyusun Dari Pixel Frame Video

	Pixel position(0,0)	Pixel position(1,0)	Pixel position(2,0)
Pixel merah	199	184	187
Pixel hijau	174	159	81
Pixel biru	44	81	60

Setelah itu byte-byte tersebut dirubah menjadi biner menjadi seperti berikut:

Tabel 3.10 Bit Penyusun Dari Pixel Frame Video

	Pixel position(0,0)	Pixel position(1,0)	Pixel position(2,0)
Pixel merah	11000111	10111000	10111011
Pixel hijau	10101110	10100000	01010001
Pixel biru	00101100	01010001	00111100

Setelah itu didapatkan index dari masing-masing bit tersebut dan dilakukan operasi modulus terhadap bit pixel dan mengambil bit terakhir dari pixel. Bit yang didapat merupakan bit dari data. Index dimulai dari 0.

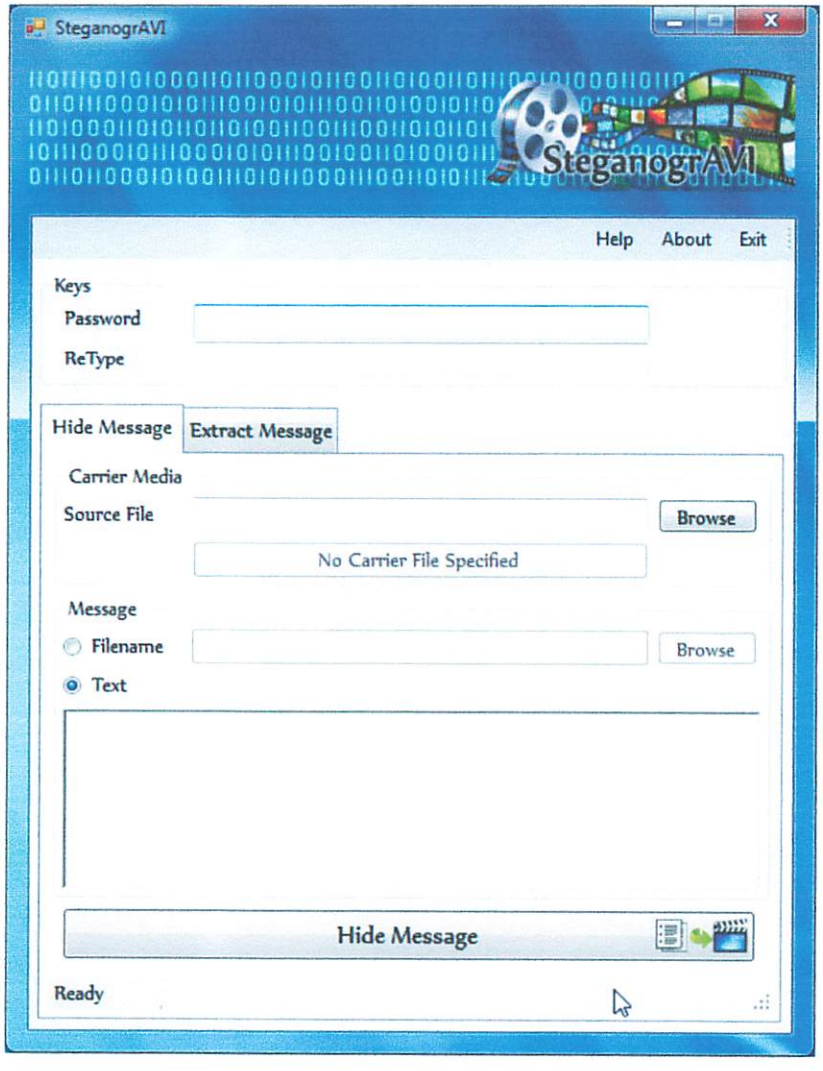
```
Jika kolom modulus 3 = 0 maka
index ke-0 dari bit data = Index ke-7 dari pixel merah
index ke-1 dari bit data = Index ke-7 dari pixel hijau
index ke-2 dari bit data = Index ke-7 dari pixel biru
Jika kolom modulus 3 = 1 maka
index ke-3 dari bit data = Index ke-7 dari pixel merah
index ke-4 dari bit data = Index ke-7 dari pixel hijau
index ke-5 dari bit data = Index ke-7 dari pixel biru
Jika kolom modulus 3 = 2 maka
index ke-6 dari bit data = Index ke-7 dari pixel merah
index ke-7 dari bit data = Index ke-7 dari pixel hijau
```

Setelah itu index dari 0 sampai 7 disusun kembali, menjadi 10000111, nilai tersebut kemudian dirubah menjadi decimal yaitu 135, dimana nilai tersebut merupakan byte penyusun dari data digital.

Proses tersebut diulang sampai batas ukuran data yang telah disisipkan pada frame pertama

BAB IV
IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada bab ini menjelaskan tentang implementasi dan pengujian sistem dari aplikasi yang telah dibuat. Implementasi meliputi proses-proses yang terdapat dalam aplikasi, *designinput* dan *output* serta hasil yang sesuai dengan target yang diharapkan. Pengujian sistem dilakukan dengan mengukur tingkat keberhasilan ekstraksi data digital yang disembunyikan, pengukuran tingkat keberhasilan dilakukan dengan melakukan uji coba aplikasi menggunakan data-data digital yang umum digunakan.Untuk lebih memperjelas gambaran tentang aplikasi yang telah dibuat berikut ini adalah gambar *design input* dan *output* aplikasi.



Gambar 4.1 Implementasi Desain Input Output

Pada pengujian sistem ini digunakan dua buah perangkat yaitu perangkat lunak dan juga perangkat keras. Dimana masing-masing perangkat yang digunakan akan dikelompokkan sebagai berikut:

Perangkat lunak yang digunakan untuk melakukan pengujian ini yaitu:

1. Sistem Operasi *Windows 7 Ultimate*
2. Microsoft Visual Studio 2008 yang digunakan untuk menjalankan program Steganografi
3. Program Steganografi
4. Virtual Dub 1.10

Sedangkan perangkat keras yang digunakan yaitu:

1. Processor Intel® Core™2 2.00 Gigaherzt
2. RAM 2.00 GB
3. VGA nVidia GeForce 512 MB
4. Monitor 15"
5. Hard disk 160 GB

4.1 Implementasi

4.1.1 Tahap Penyisipan

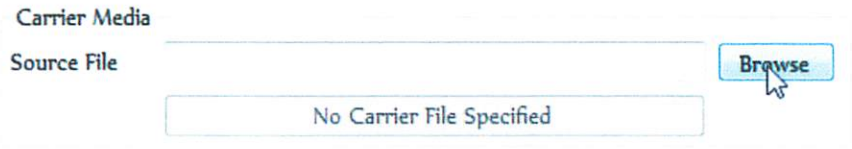
Tahap penyisipan ini digunakan untuk melakukan proses penyisipan berkas rahasia kedalam video dan juga melihat informasi tentang video yang akan digunakan sebagai media untuk penyisipan.

Proses dimulai dengan memasukkan Password sebagai pengaman pada *GroupBox*Keys. Berikut gambar dari proses ini.

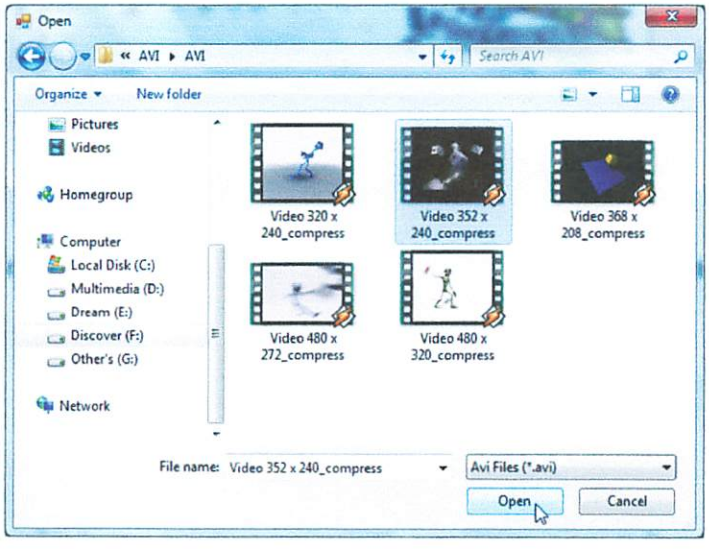
Gambar 4.2 GroupBox Keys

Selanjutnya pengguna memilih satu file video sebagai *cover object* atau wadah yang ingin digunakan untuk menyimpan data digital yang ingin disembunyikan. Untuk mengambil *cover object* dapat dilakukan dengan

menekan tombol "Browse" pada *GroupBox* Carrier media (Gambar 4.3).setelah itu akan muncul *Window Open Dialog* yang digunakan untuk memilih video yang akan digunakan sebagai *cover object* (Gambar 4.4)

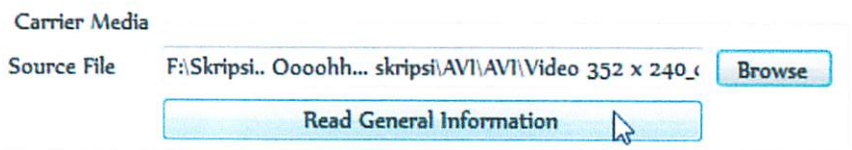


Gambar 4.3Menekan Tombol "Browse" Untuk Memilih Video

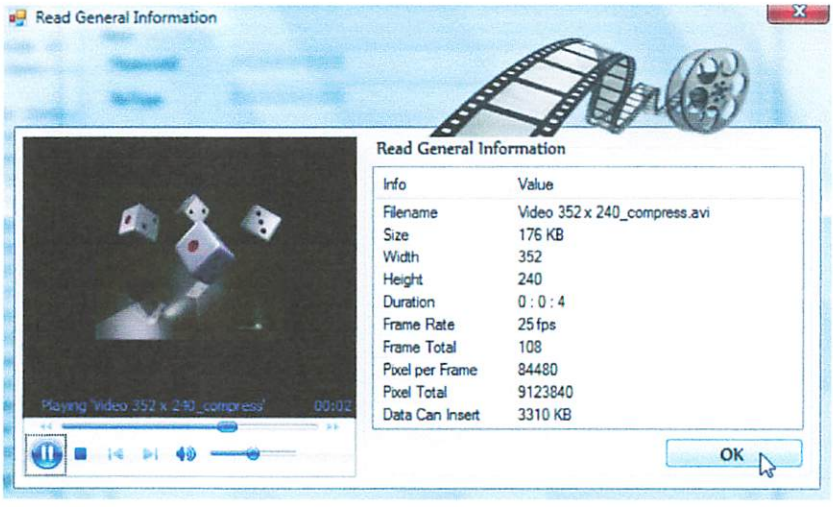


Gambar 4.4 Window Open Untuk Memilih Video

Setelah memilih video yang diinginkan, untuk melihat atribut yangada pada video dapat dilakukan dengan menekan tombol "Read General Information" (Gambar 4.5). Setelah ditekan maka akan muncul sebuah form yang memainkan video yang dipilih, berikut atribut-atribut dari video tersebut(Gambar 4.6).

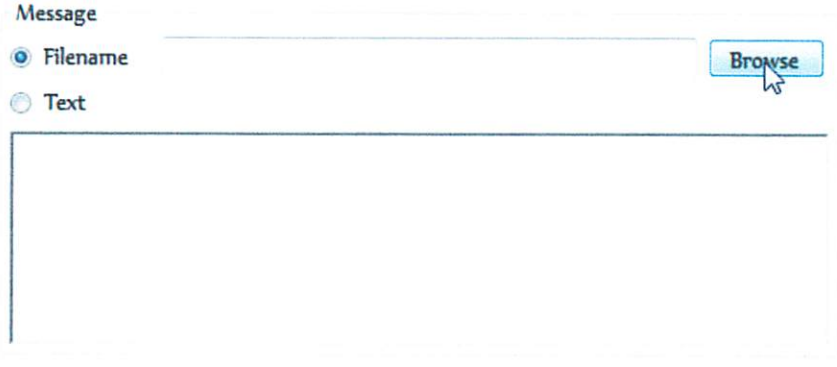


Gambar 4.5 Menekan Tombol "Read General Information"



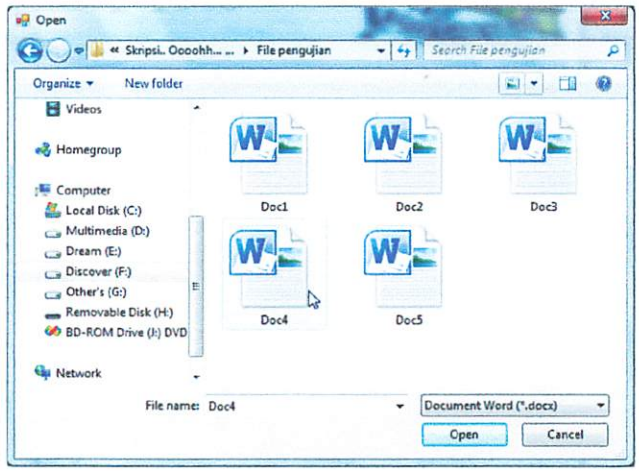
Gambar 4.6 AtributDari Video Yang Dipilih

Setelah itu memasukkan pesan yang berupa data digital, untuk data digital yang berupa teks biasa dapat menginputkan pada RichTextBox yang disediakan, jika berupa file dengan mengklik RadioButtonFilename lalu cari file yang akan disisipkan dengan menekan tombol "Browse".



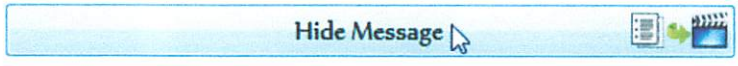
Gambar 4.7 Menekan Tombol "Browse" Untuk Memilih Data digital

Setelah itu akan muncul *WindowOpenDialog* yang digunakan untuk memilih file yang akan disisipkan

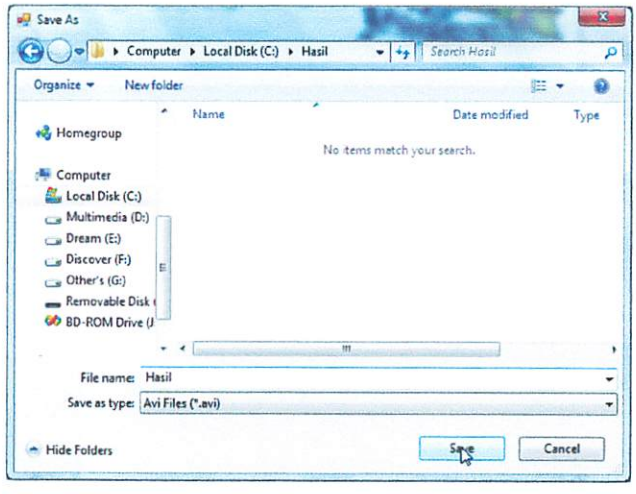


Gambar 4.8 Window Open untuk memilih video

Setelah selesai memilih data digital yang akan disisipkan pengguna menekan tombol Hide message (Gambar 4.9), untuk memulai proses penyisipan. Setelah ditekan, apabila tidak terdapat pesan error maka akan mucul *Window Save Dialog*yang digunakan untuk menyimpan video yang telah disisipi pesan (Gambar 4.10).

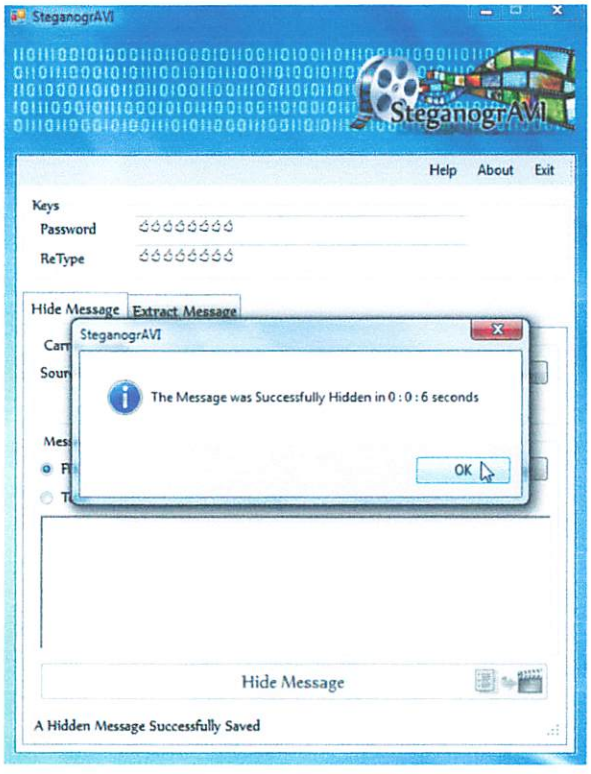


Gambar 4.9 Tombol Hide Message



Gambar 4.10 MenyimpanHasil Video Yang Telah Disisipi Pesan

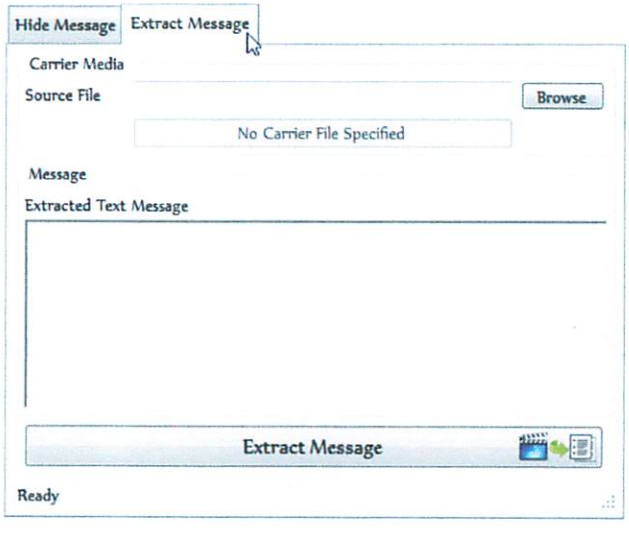
Setelah itu menunggu sampai ada pemberitahuan yang menyatakan bahwa penyisipan pesan telah berhasil.



Gambar 4.11 Proses Penyisipan Telah Berhasil

4.1.2 Tahap Ekstraksi

Tahap ekstraksi disini digunakan untuk mengekstraksi kembali berkas rahasia yang telah disisipkan ke dalam video.Langkah pertama adalah memilih Tab Extract Message.

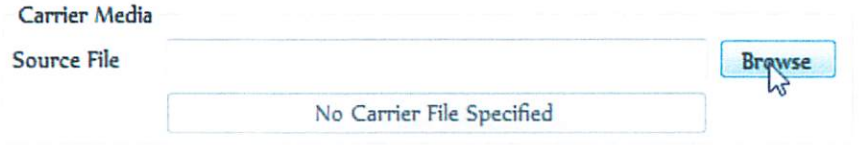


Gambar 4.12 Memilih Tab Extract Message

Langkah selanjutnya memasukkan password yang digunakan pada saat penyisipan.

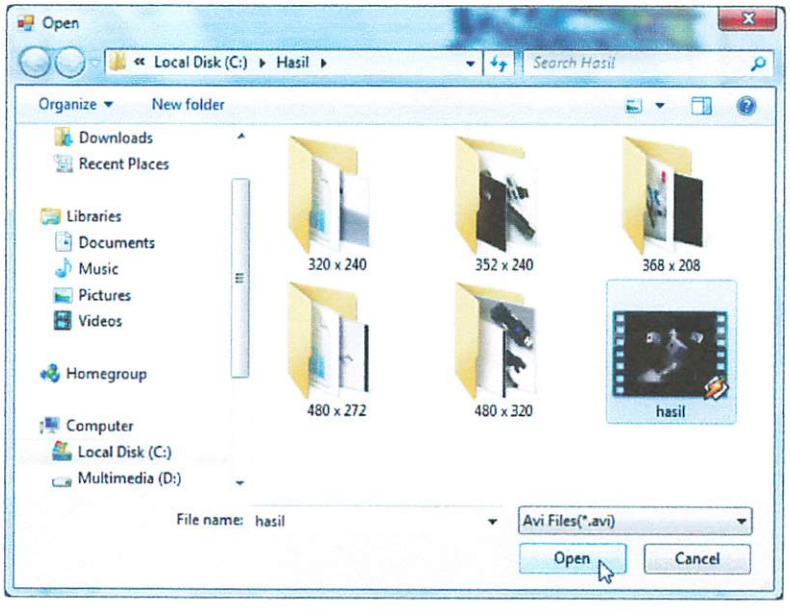
Gambar 4.13 Memasukkan Password

Selanjutnya memilih video yang telah disisipi berkas hasil dari tahap penyisipan, caranya sama pada tahap penyisipan. Yaitu menekan tombol "Browse" untuk mencari file video tersebut.



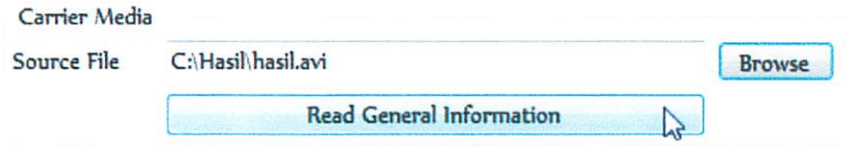
Gambar 4.14 Menekan Tombol "Browse" Untuk Memilih Video

Setelah itu akan muncul *Window Open Dialog*yang digunakan untuk memilih file video yang telah disisipi pesan.

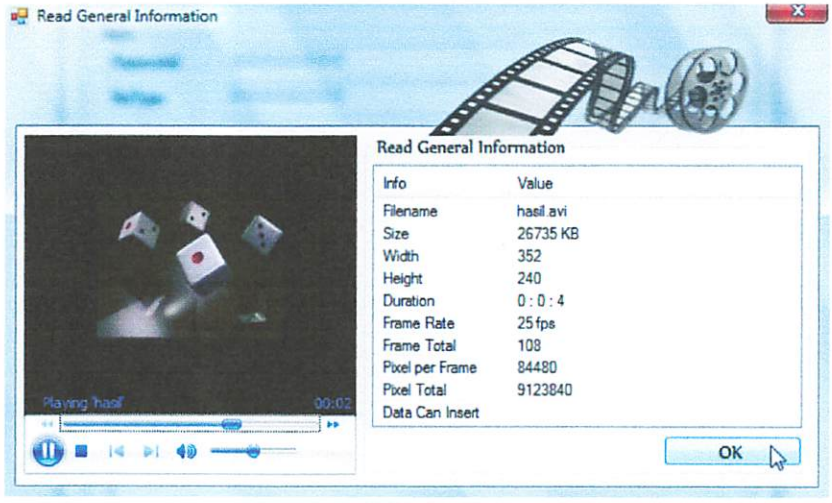


Gambar 4.15 Window Open Untuk Memilih Video

Setelah memilih video yang diinginkan, untuk melihat atribut yangada pada video dapat dilakukan dengan menekan tombol "Read General Information" (Gambar 4.16). Setelah ditekan maka akan muncul sebuah form yang memainkan video yang dipilih, berikut atribut-atribut dari video tersebut (Gambar 4.17).

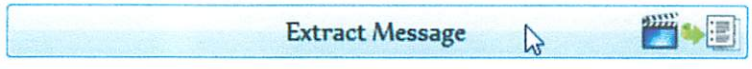


Gambar 4.16 Menekan Tombol "Read General Information"



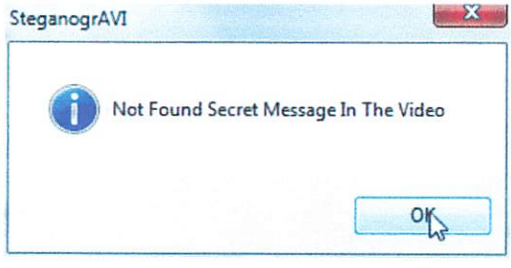
Gambar 4.17 Atribut Dari Video Yang Dipilih

Setelah itu menekan tombol Extract Message untuk memulai proses pengekstrakan.



Gambar 4.18 Tombol Extract Message

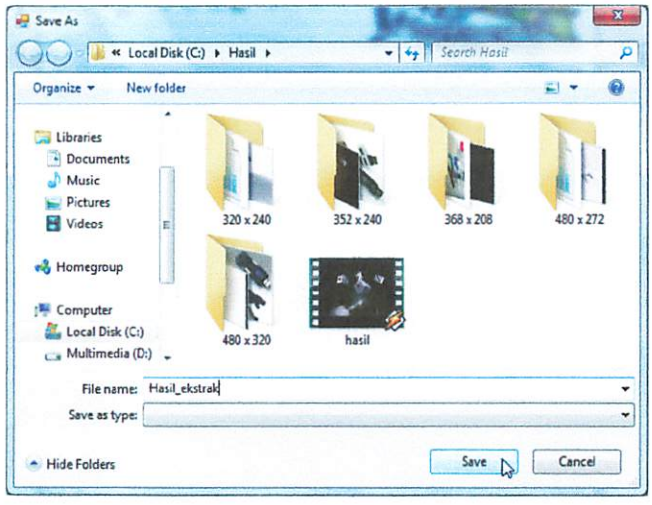
Jika password yang dimasukkan tidak sama dengan password pada saat tahap penyisipan, maka akan muncul pesan error yang berbunyi tidak ada pesan rahasia didalam video tersebut.



Gambar 4.19 Pesan Error jika Password tidak sama

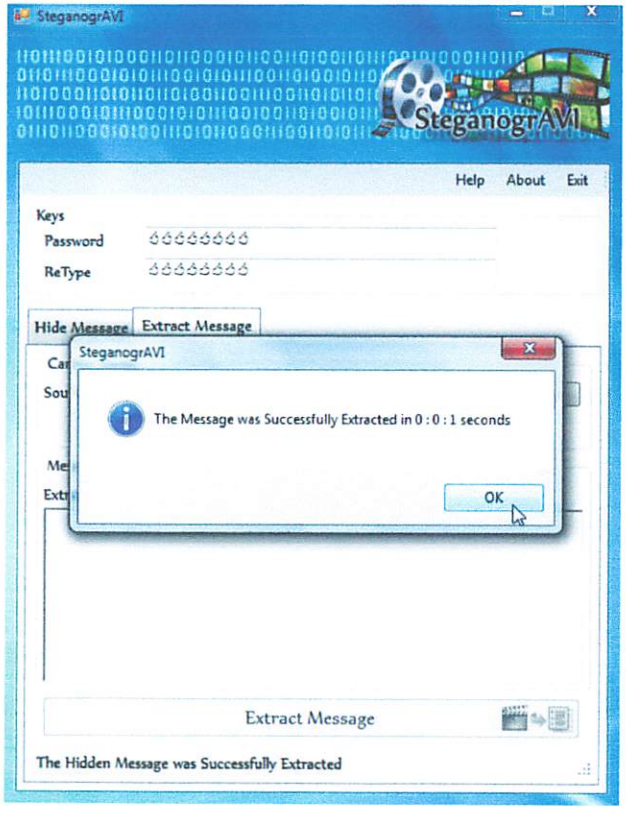
Jika password yang dimasukkan sama, maka akan dideteksi apakah bentuk dari data digital yang disisipkan. Jika berupa teks, maka akan ditampilkan teks

tersebut dalam RichTextBox yang disediakan, dikarenakan pada tahap penyisipan data digital yang disisipkan berupa file, maka setelah tombol ditekan otomatis akan muncul *Window Save Dialog*yang digunakan untuk menyimpan file tersebut. Setelah itu simpan dengan nama yang diinginkan.



Gambar 4.20 Window Save Dialog

Setelah itu proses ekstraksi akan berjalan, dan tunggu sampai keluar pesan bahwa ekstraksi pesan telah berhasil



Gambar 4.21 Proses Pengekstrakan Telah Berhasil

4.2 Pengujian Sistem

Pengujian sistem merupakan proses selanjutnya setelah implementasi perangkat lunak selesai dilakukan. Pengujian ini dilakukan untuk menguji berkas rahasia dan video avi yang digunakan.

4.2.1 Pengujian Video Avi Berdasarkan Resolusi

Pengujian ini dilakukan untuk menyisipkan data digital pada video berdasarkan resolusi dari video yang digunakan sebagai media penyisipan. Berikut ini 5 resolusi video yang akan digunakan sebagai media penyisipan yaitu:

1. Video beresolusi 320 x 240
2. Video beresolusi 352 x 240
3. Video beresolusi 368 x 208
4. Video beresolusi 480 x 272
5. Video beresolusi 480 x 320

4.2.1.1 Resolusi 320 x 240

Tabel 4.1 Hasil Pengujian Terhadap Video Avi Yang Beresolusi 320x240

No.	Info Video		Info Berkas		Proses	
	Ukuran Video	Jumlah Frame	Format	Ukuran	Penyisipan	Ekstraksi
1	17.8MB	81	*.txt	1 KB	Memenuhi	Memenuhi
2			*.docx	14 KB	Memenuhi	Memenuhi
3			*.xlsx	15 KB	Memenuhi	Memenuhi
4			*.jpg	15 KB	Memenuhi	Memenuhi

Video avi dengan nama file (Video 320 x 240.avi) yang beresolusi 320 x 240 piksel dengan ukuran filenya 17.8 MB dan mempunyai 81 frame dapat disisipi berkas rahasia dengan format (*.txt dengan ukuran filenya 1 KB, *.docx dengan ukuran filenya 14 KB, *.xlsx dengan ukuran filenya 15 KB, dan *.jpg dengan ukuran filenya 15 KB) dan video hasil penyisipannya dapat diekstraksi kembali untuk menghasilkan pesan yang telah disembunyikan (Tabel 4.1).

4.2.1.2 Resolusi 352 x 240

Tabel 4.2 Hasil Pengujian Terhadap Video Avi Yang Beresolusi 352x240

No.	Info Video		Info Berkas		Proses	
	Ukuran Video	Jumlah Frame	Format	Ukuran	Penyisipan	Ekstraksi
1	26.1MB	108	*.txt	7 KB	Memenuhi	Memenuhi
2			*.docx	17 KB	Memenuhi	Memenuhi
3			*.xlsx	12 KB	Memenuhi	Memenuhi
4			*.jpg	13 KB	Memenuhi	Memenuhi

Video avi dengan nama file (Video 352 x 240.avi) yang beresolusi 352 x 240 piksel dengan ukuran filenya 26.1 MB dan mempunyai 108 frame dapat disisipi berkas rahasia dengan format (*.txt dengan ukuran filenya 7 KB, *.docx dengan ukuran filenya 17 KB, *.xlsx dengan ukuran filenya 12 KB, dan *.jpg dengan ukuran filenya 13 KB) dan video hasil penyisipannya dapat diekstraksi kembali untuk menghasilkan pesan yang telah disembunyikan (Tabel 4.2).

4.2.1.3 Resolusi 368 x 208

Tabel 4.3 Hasil Pengujian Terhadap Video Avi Yang Beresolusi 352x208

No.	Info Video		Info Berkas		Proses	
	Ukuran Video	Jumlah Frame	Format	Ukuran	Penyisipan	Ekstraksi
1	23.7 MB	108	*.txt	7 KB	Memenuhi	Memenuhi
2			*.docx	24 KB	Memenuhi	Memenuhi
3			*.xlsx	12 KB	Memenuhi	Memenuhi
4			*.jpg	11 KB	Memenuhi	Memenuhi

Video avi dengan nama file (Video 368 x 208.avi) yang beresolusi 352 x 208 piksel dengan ukuran filenya 23.7 MB dan mempunyai 108 frame dapat disisipi berkas rahasia dengan format (*.txt dengan ukuran filenya 7 KB, *.docx dengan ukuran filenya 24 KB, *.xlsx dengan ukuran filenya 12 KB, dan *.jpg dengan ukuran filenya 12 KB) dan video hasil penyisipannya dapat diekstraksi kembali untuk menghasilkan pesan yang telah disembunyikan (Tabel 4.3).

4.2.1.4 Resolusi 480 x 272

Tabel 4.4 Hasil Pengujian Terhadap Video Avi Yang Beresolusi 480x272

No.	Info Video		Info Berkas		Proses	
	Ukuran Video	Jumlah Frame	Format	Ukuran	Penyisipan	Ekstraksi
1	37.0 MB	99	*.txt	8 KB	Memenuhi	Memenuhi
2			*.docx	23 KB	Memenuhi	Memenuhi
3			*.xlsx	12 KB	Memenuhi	Memenuhi
4			*.jpg	16 KB	Memenuhi	Memenuhi

Video avi dengan nama file (Video 480 x 272.avi) yang beresolusi 480 x 272 piksel dengan ukuran filenya 37.0 MB dan mempunyai 99 frame dapat disisipi berkas rahasia dengan format (*.txt dengan ukuran filenya 8 KB, *.docx dengan ukuran filenya 23 KB, *.xlsx dengan ukuran filenya 12 KB, dan *.jpg dengan ukuran filenya 16 KB) dan video hasil penyisipannya dapat diekstraksi kembali untuk menghasilkan pesan yang telah disembunyikan (Tabel 4.4).

4.2.1.5 Resolusi 480 x 320

Tabel 4.5 Hasil Pengujian Terhadap Video Avi Yang Beresolusi 480x320

No.	Info Video		Info Berkas		Proses	
	Ukuran Video	Jumlah Frame	Format	Ukuran	Penyisipan	Ekstraksi
1	43.1 MB	98	*.txt	10 KB	Memenuhi	Memenuhi
2			*.docx	15 KB	Memenuhi	Memenuhi
3			*.xlsx	14 KB	Memenuhi	Memenuhi
4			*.jpg	7 KB	Memenuhi	Memenuhi

Video avi dengan nama file (Video 480 x 320.avi) yang beresolusi 480 x 320 piksel dengan ukuran filenya 43.1 MB dan mempunyai 98 frame dapat disisipi berkas rahasia dengan format (*.txt dengan ukuran filenya 10 KB, *.docx dengan ukuran filenya 15 KB, *.xlsx dengan ukuran filenya 14 KB, dan *.jpg dengan ukuran filenya 7 KB) dan video hasil penyisipannya dapat diekstraksi kembali untuk menghasilkan pesan yang telah disembunyikan (Tabel 4.5).

4.2.2 Pengujian Terhadap Waktu Yang Dibutuhkan Untuk Melakukan Proses Penyisipan Dan Ekstraksi

Pengujian ini dilakukan untuk mengetahui waktu yang dibutuhkan untuk melakukan proses penyisipan dan ekstraksi terhadap data digital. Resolusi video yang digunakan untuk melakukan pengujian ini adalah 320 x 240 dengan ukuran yang berbeda.Sedangkan data digital yang disisipkan adalah yang berekstensi (*.docx) dengan ukuran 36 KB.

Tabel 4.6Hasil Pengujian Terhadap Waktu Yang Dibutuhkan

No.	Info Video		Info Berkas		Waktu Proses	
	Ukuran Video	Jumlah Frame	Format	Ukuran	Penyisipan	Ekstraksi
1	20.5 MB	93	*.docx	36 KB	0:00:03	0:00:01
2	40.3 MB	183	*.docx	36 KB	0:00:07	0:00:02
3	52.7 MB	240	*.docx	36 KB	0:00:09	0:00:03
4	83.0 MB	377	*.docx	36 KB	0:00:20	0:00:06
5	109 MB	499	*.docx	36 KB	0:00:23	0:00:09
6	215 MB	980	*.docx	36 KB	0:01:02	0:00:25
7	314 MB	1429	*.docx	36 KB	0:01:35	0:00:45
8	381 MB	1733	*.docx	36 KB	0:01:45	0:00:55
9	603 MB	2746	*.docx	36 KB	0:03:03	0:01:53
10	787 MB	3575	*.docx	36 KB	0:07:56	0:03:09

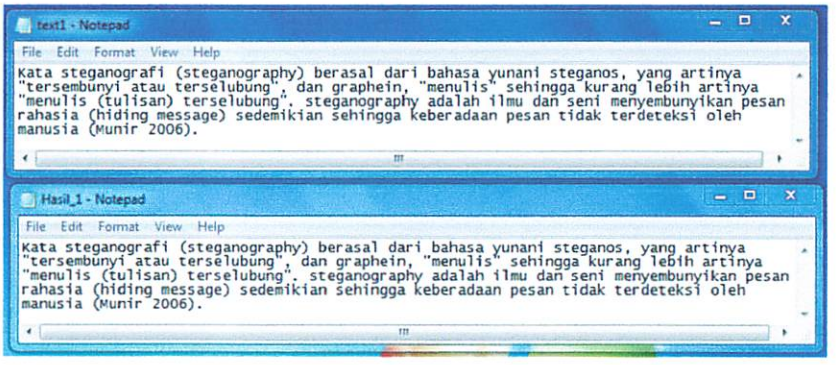
Dari beberapa pengujian diatas (Tabel 4.6), dapat terlihat bahwa semakin besar jumlah frame sebuah video maka waktu yang dibutuhkan untuk melakukan proses steganografi akan semakin lama, akan tetapi waktu tersebut tidak valid, dikarenakan proses looping dari proses steganografi dan sistem dari komputer yang digunakan itu sendiri tidak stabil.

4.2.3 Pengujian Terhadap Perbandingan Isi Dan ukuran Data Digital

Pengujian yang di lakukan disini yaitu dengan membandingkan isi dan ukuran dari berkas rahasia sebelum dan sesudah dilakukan proses steganografi. Pengujian ini dapat dikatakan berhasil apabila isi dan ukuran berkas sebelum dan sesudah dilakukan proses steganografi harus sama persis baik isi maupun ukuran filenya. Pengujian ini dilakukan berdasarkan hasil pengujian dari sub bab 4.2.1.1.

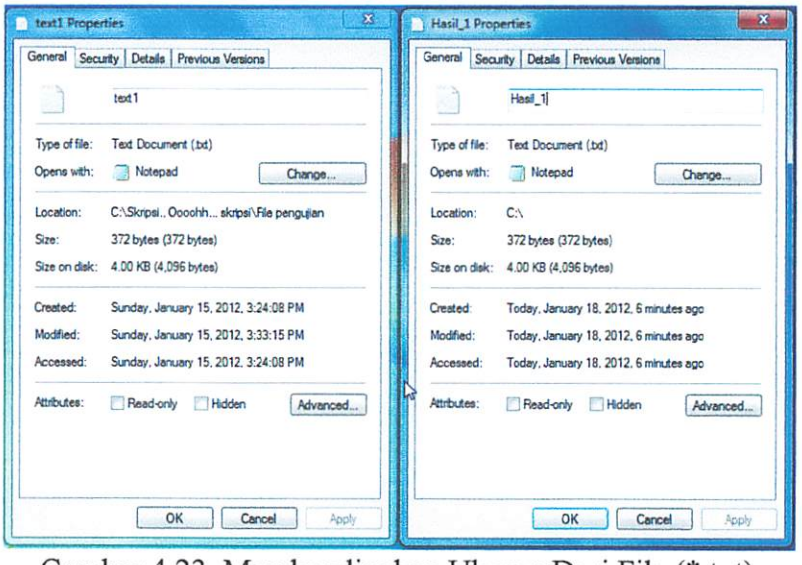
4.2.3.1 Data Digital Berekstensi (*.txt)

Sebuah data digital dengan nama file (text1.txt) yang disisipkan kedalam video avi dengan nama file (Video 320 x 240.avi) yang berukuran 17.8 MB. Setelah dilakukan pengekstrakan disimpan dengan nama Hasil_1.txt. kemudian dilakukan pembandingan terhadap isi dari kedua file tersebut.



Gambar 4.22Membandingkan Isi Dari File (*.txt)

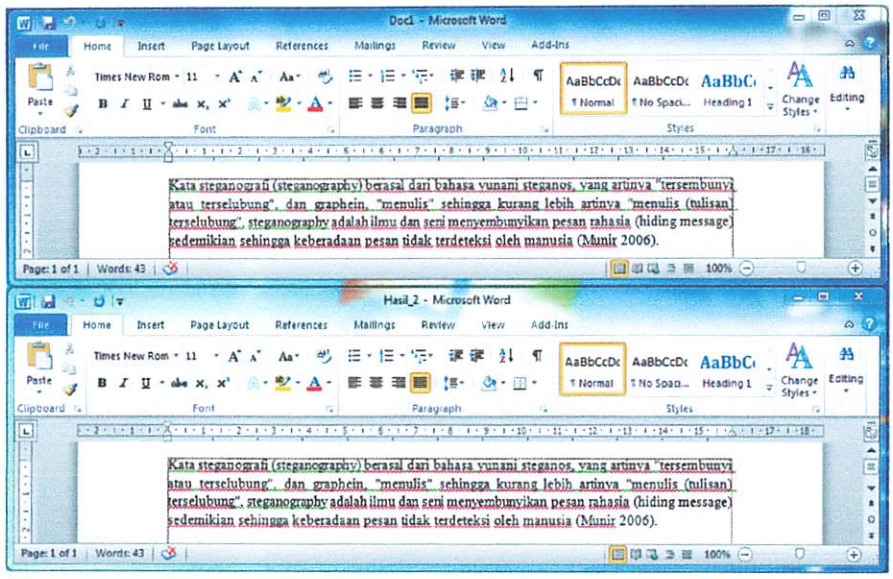
Setelah itu membandingkan ukuran dari file tersebut.



Gambar 4.23 Membandingkan Ukuran Dari File (*.txt)

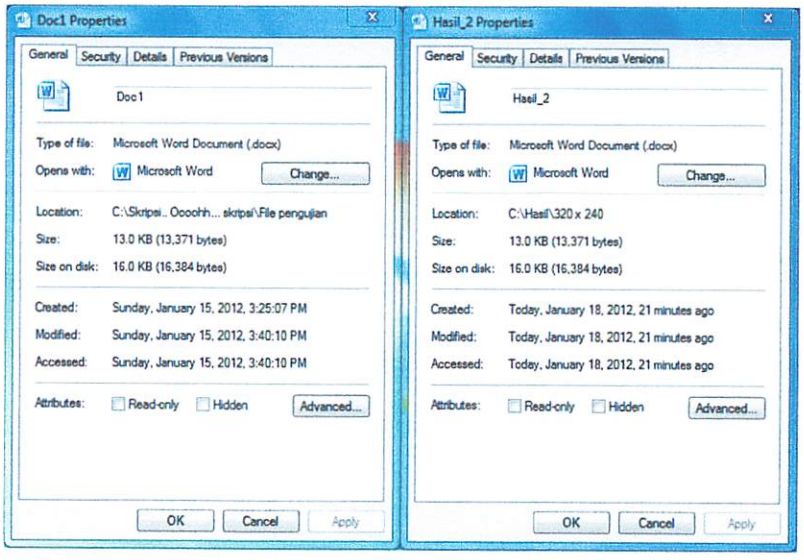
4.2.3.2 Data Digital Berekstensi (*.docx)

Sebuah data digital dengan nama file (Doc1.docx) yang disisipkan kedalam video avi dengan nama file (Video 320 x 240.avi) yang berukuran 17.8 MB. Setelah dilakukan pengekstrakan disimpan dengan nama Hasil_2.docx. kemudian dilakukan pembandingan terhadap isi dari kedua file tersebut.



Gambar 4.24Membandingkan Isi Dari File (*.docx)

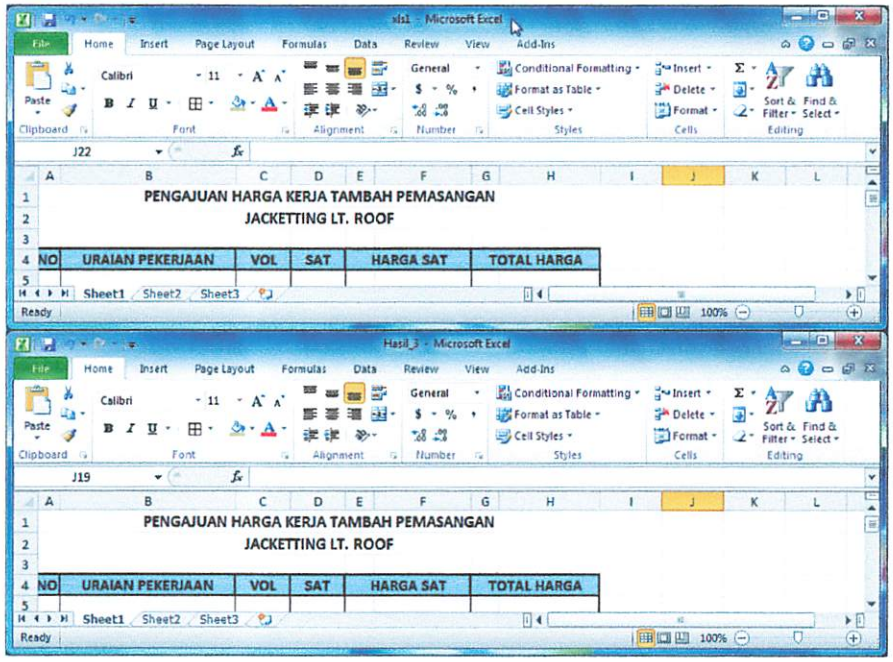
Setelah itu membandingkan ukuran dari file tersebut.



Gambar 4.25 Membandingkan Ukuran Dari File (*.txt)

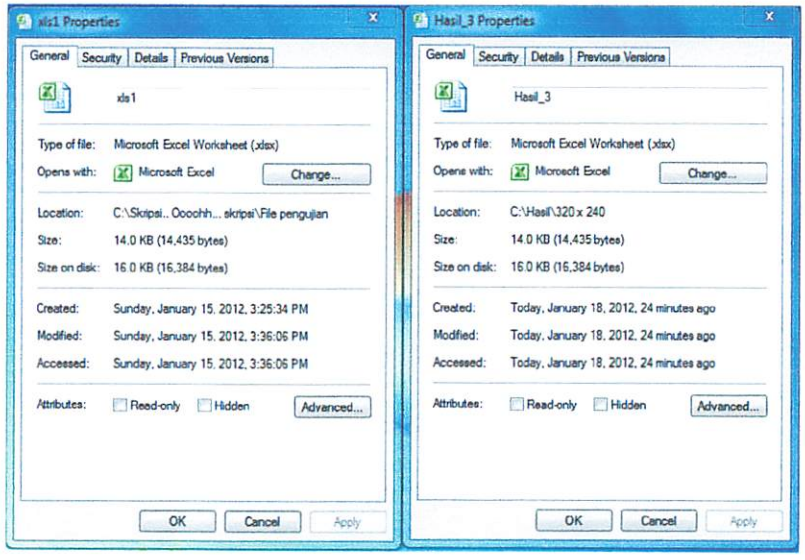
4.2.3.3 Data Digital Berekstensi (*.xlsx)

Sebuah data digital dengan nama file (xls1.xlsx) yang disisipkan kedalam video avi dengan nama file (Video 320 x 240.avi) yang berukuran 17.8 MB. Setelah dilakukan pengekstrakan disimpan dengan nama Hasil_3.xlsx. kemudian dilakukan pembandingan terhadap isi dari kedua file tersebut.



Gambar 4.26 Membandingkan Isi Dari File (*.xlsx)

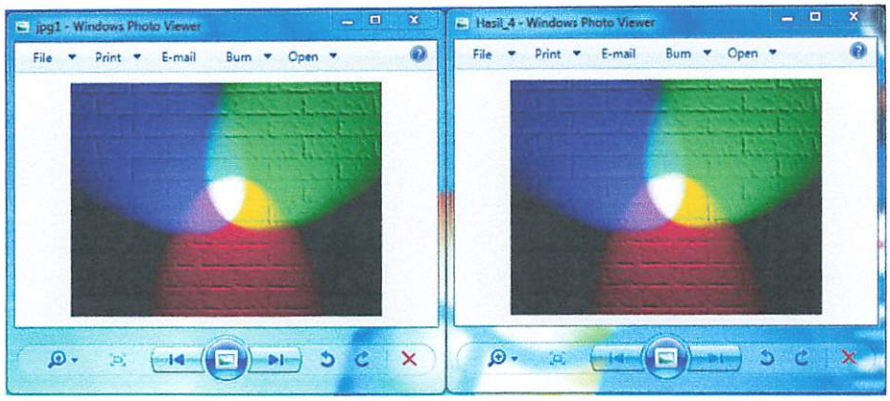
Setelah itu membandingkan ukuran dari file tersebut.



Gambar 4.27 Membandingkan Ukuran Dari File (*.xlsx)

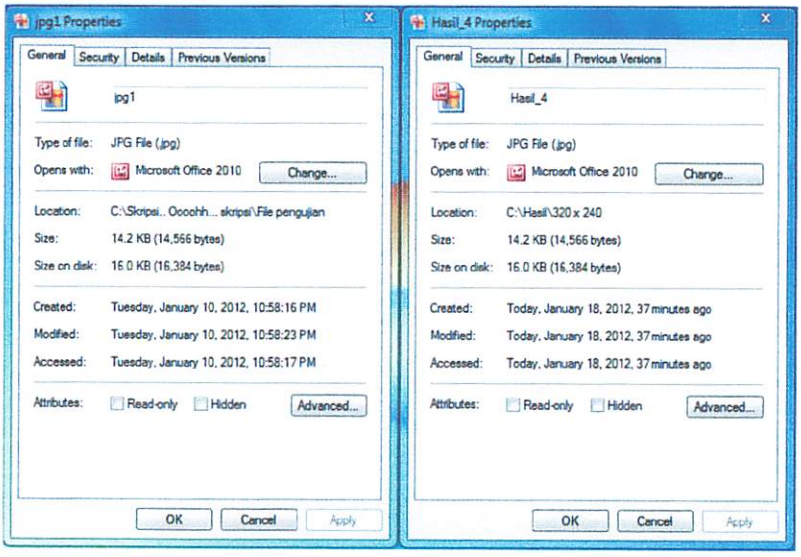
4.2.3.4 Data Digital Berekstensi (*.jpg)

Sebuah data digital dengan nama file (jpg1.jpg) yang disisipkan kedalam video avi dengan nama file (Video 320 x 240.avi) yang berukuran 17.8 MB. Setelah dilakukan pengekstrakan disimpan dengan nama Hasil_4.jpg. kemudian dilakukan pembandingan terhadap isi dari kedua file tersebut.



Gambar 4.28 Membandingkan Isi Dari File (*.jpg)

Setelah itu membandingkan ukuran dari file tersebut.



Gambar 4.29 Membandingkan Ukuran Dari File (*.jpg)

Hasil dari beberapa pengujian diatas menunjukkan bahwa isi dan ukuran dari semua berkas yang berhasil diproses sama persis dengan berkas rahasia sebelumnya sehingga pengujian ini dapat dikatakan berhasil.

4.2.4 Pengujian Terhadap Atribut Video

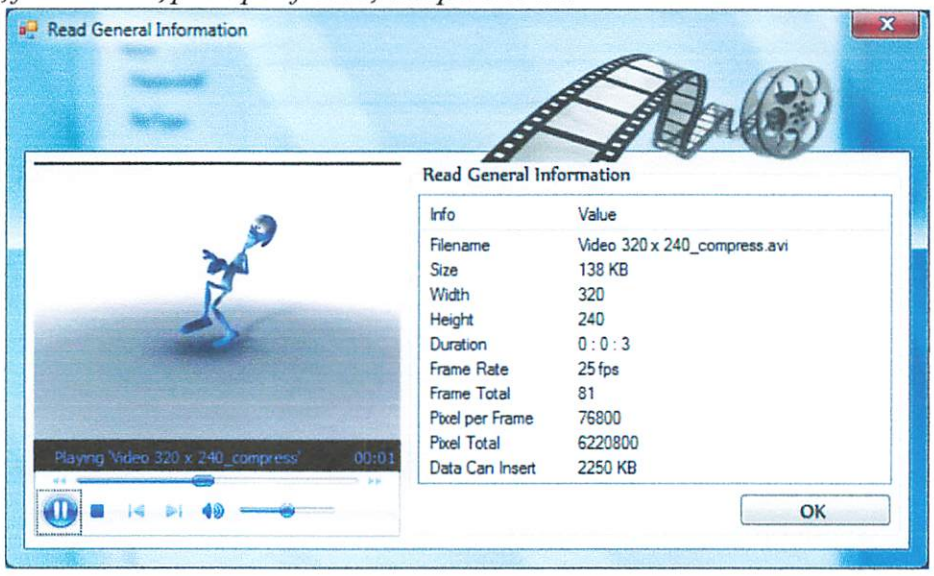
Pengujian ini dilakukan dengan memanfaatkan Form *Read General Information* yang terdapat pada aplikasi.Pada pengujian ini dilakukan dua kali pengujian, yang pertama pengujian dilakukan terhadap video yang di kompresi sebelum proses penyisipan, kemudian membandingkan atribut video tersebut

dengan atribut video sesudah proses penyisipan, yang kedua dilakukan terhadap video yang tidak dikompresi sebelum proses penyisipan, kemudian membandingkan atribut video tersebut dengan atribut video sesudah proses penyisipan.Untuk melakukan kompresi dan pendekompresian pada video digunakan software Virtual Dub 1. 10.

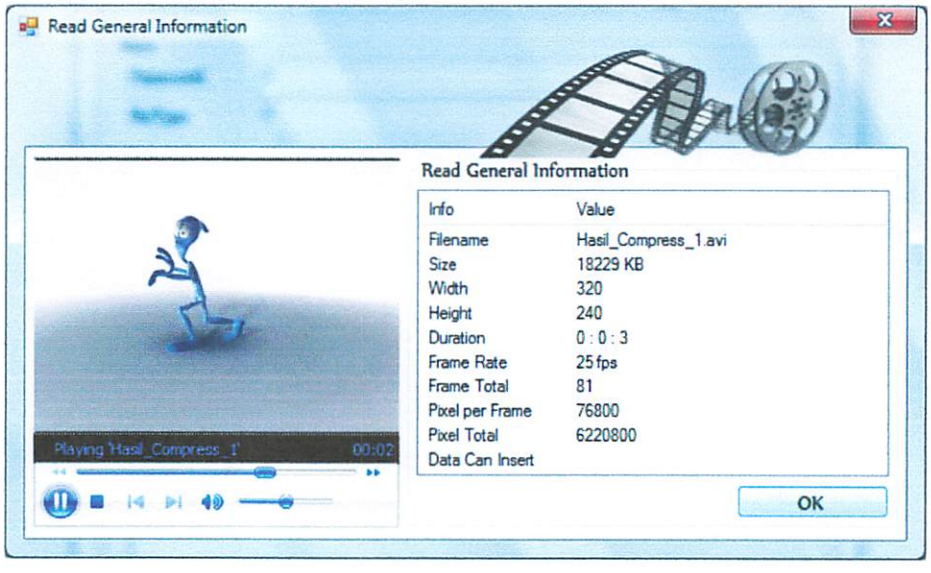
4.2.4.1 Video Yang Dikompresi

4.2.4.1.1 Video Dengan Resolusi 320x240

Pembacaan atribut meliputi ukuran dari video, resolusi, durasi, *frame rate*, *frame total*, *pixel per frame*, dan *pixel total*.



Gambar 4.30 Atribut Video Sebelum Disisipi Pesan

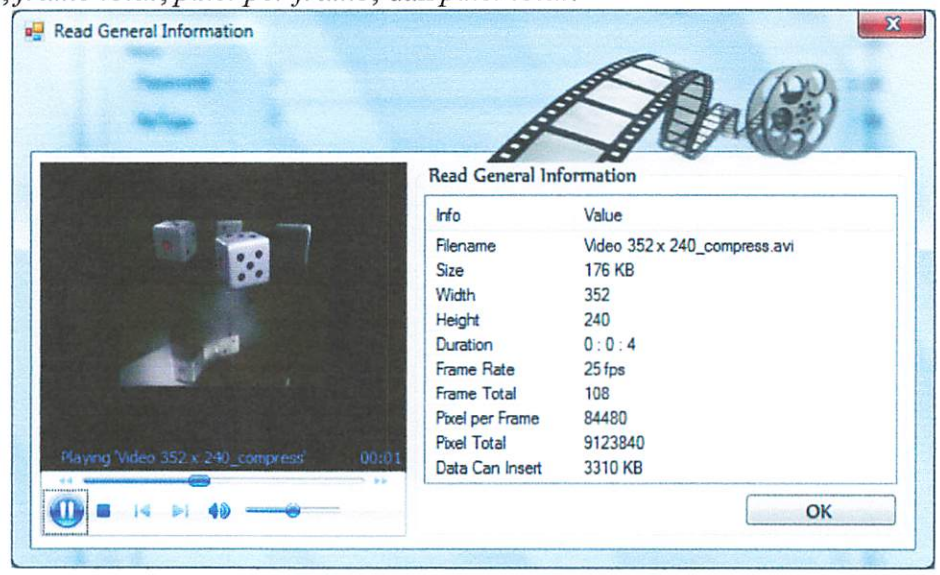


Gambar 4.31 Atribut Video Sesudah Disisipi Pesan

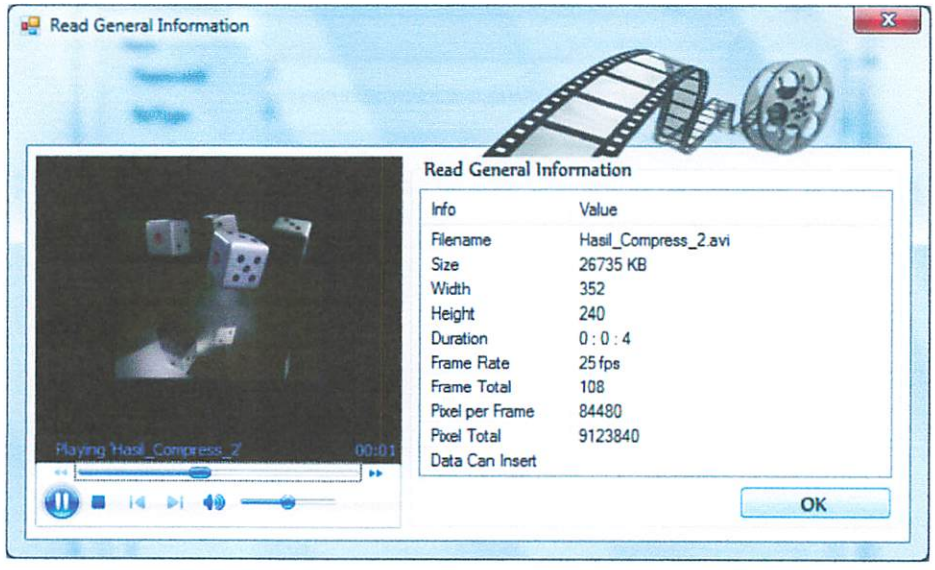
File video avi dengan nama Video 320x240_compress.avi sebelum disisipi pesan mempunyai ukuran 138KB dengan resolusi 320x240, *frame* total 81 dengan *frame rate* 25 fps, *pixel* total 6220800 dengan *pixel per frame* 76800, setelah disisipi pesan disimpan dengan nama Hasil_Compress_1.avi mempunyai ukuran 18229KB dengan resolusi 320x240, *frame* total 81 dengan *frame rate* 25 fps, *pixel* total 6220800 dengan *pixel per frame* 76800.

4.2.4.1.2 Video Dengan Resolusi 352x240

Pembacaan atribut meliputi ukuran dari video, resolusi, durasi, *frame rate*, *frame total*, *pixel per frame*, dan *pixel total*.



Gambar 4.32 Atribut Video Sebelum Disisipi Pesan

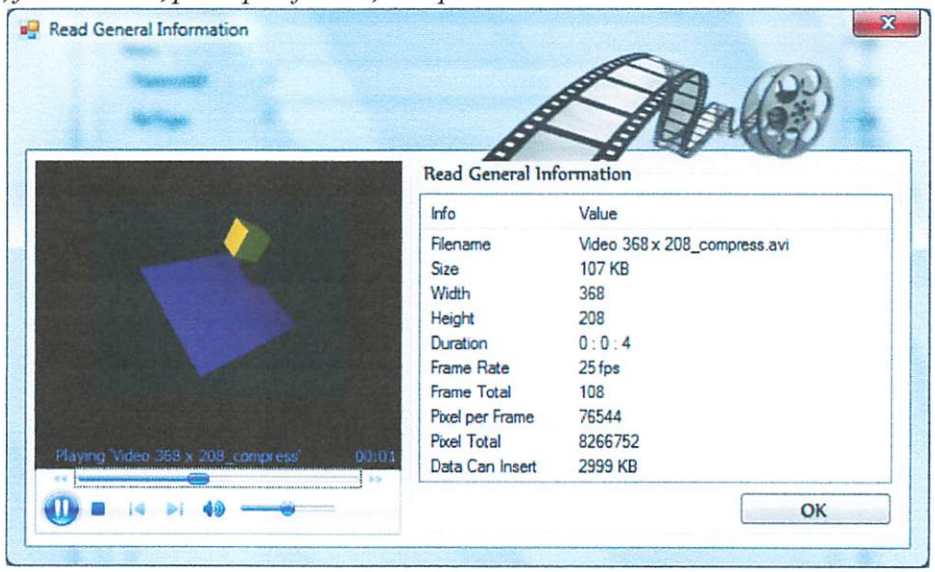


Gambar 4.33 Atribut Video Sesudah Disisipi Pesan

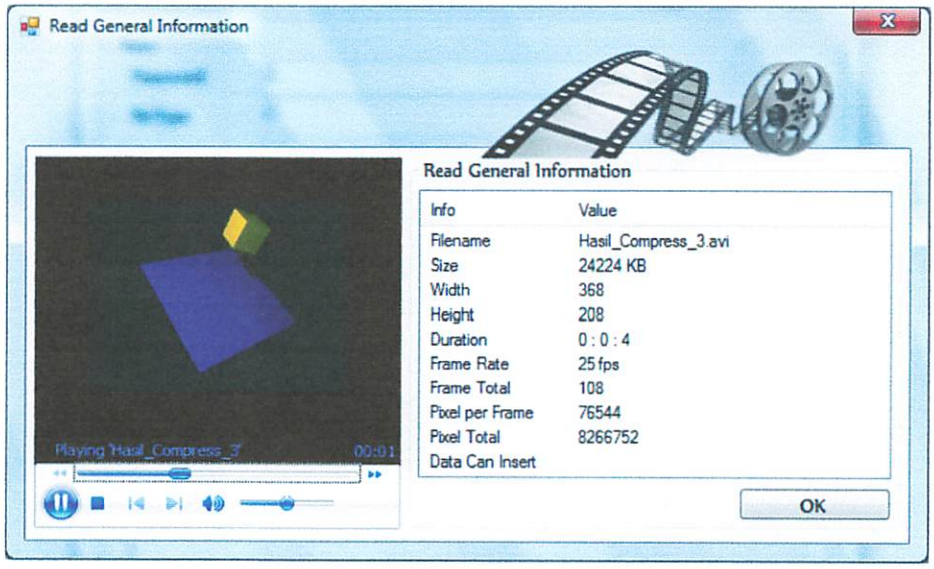
File video avi dengan nama Video 352x240_compress.avi sebelum disisipi pesan mempunyai ukuran 176 KB dengan resolusi 352x240, *frame* total 108 dengan *frame rate* 25 fps, *pixel* total 9123840 dengan *pixel per frame* 84480, setelah disisipi pesan disimpan dengan nama Hasil_Compress_2.avi mempunyai ukuran 26735 KB dengan resolusi 352x240, *frame* total 108 dengan *frame rate* 25 fps, *pixel* total 9123840 dengan *pixel per frame* 84480.

4.2.4.1.3 Video Dengan Resolusi 368x208

Pembacaan atribut meliputi ukuran dari video, resolusi, durasi, *frame rate*, *frame total*, *pixel per frame*, dan *pixel total*.



Gambar 4.34 Atribut Video Sebelum Disisipi Pesan

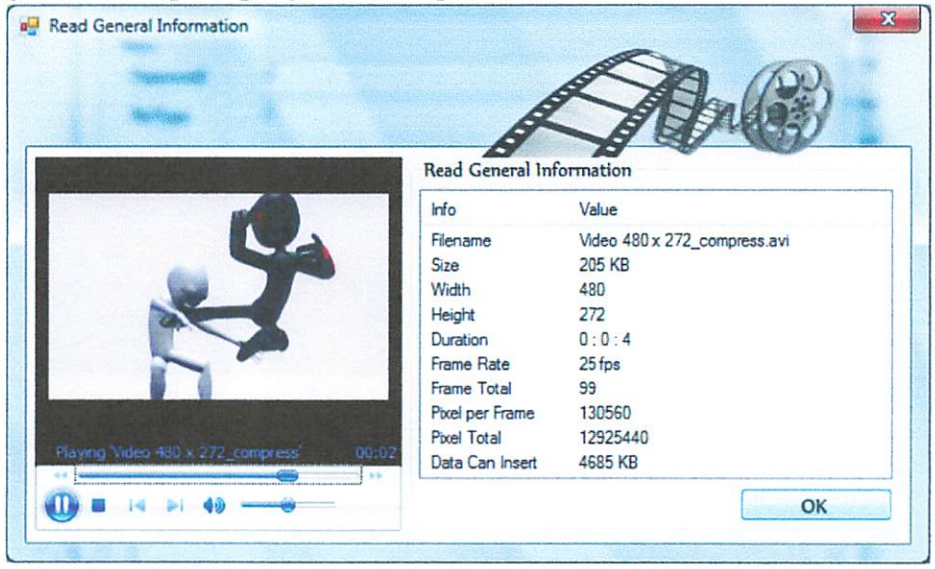


Gambar 4.35 Atribut Video Sesudah Disisipi Pesan

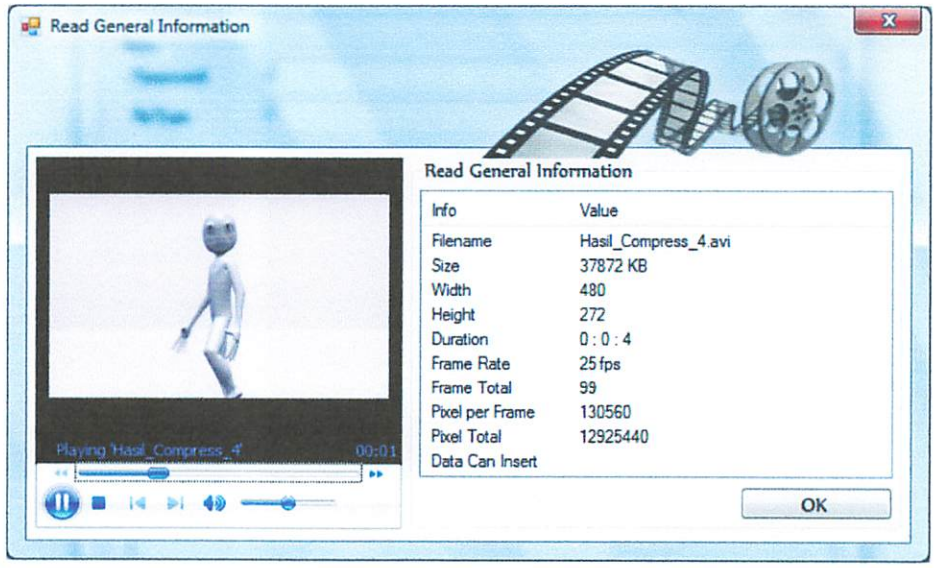
File video avi dengan nama Video 368x208_compress.avi sebelum disisipi pesan mempunyai ukuran 107 KB dengan resolusi 368x208, *frame* total 108 dengan *frame rate* 25 fps, *pixel* total 8266752 dengan *pixel per frame* 76544, setelah disisipi pesan disimpan dengan nama Hasil_Compress_3.avi mempunyai ukuran 24224 KB dengan resolusi 368x208, *frame* total 108 dengan *frame rate* 25 fps, *pixel* total 8266752 dengan *pixel per frame*76544.

4.2.4.1.4 Video Dengan Resolusi 480x272

Pembacaan atribut meliputi ukuran dari video, resolusi, durasi, *frame rate*, *frame total*, *pixel per frame*, dan *pixel total*.



Gambar 4.36 Atribut Video Sebelum Disisipi Pesan

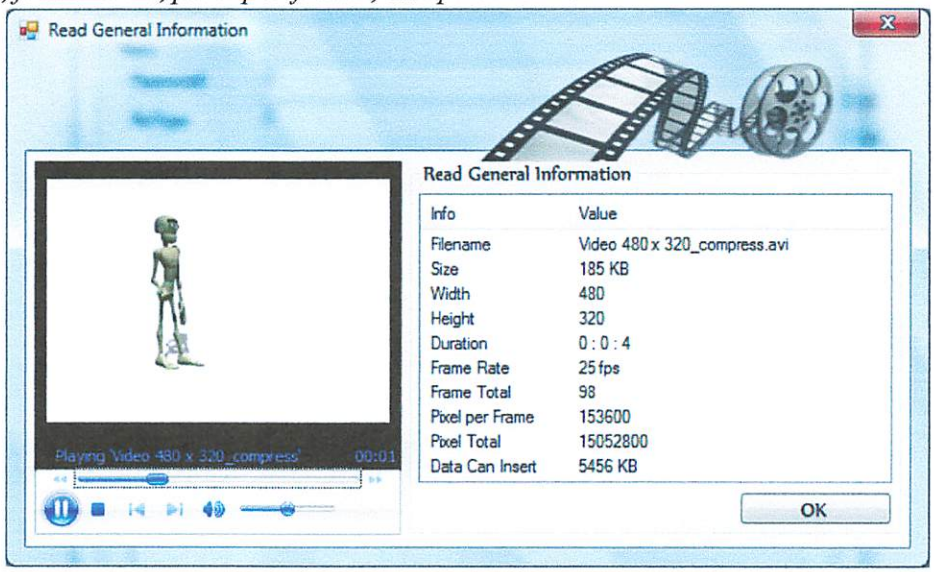


Gambar 4.37 Atribut Video Sesudah Disisipi Pesan

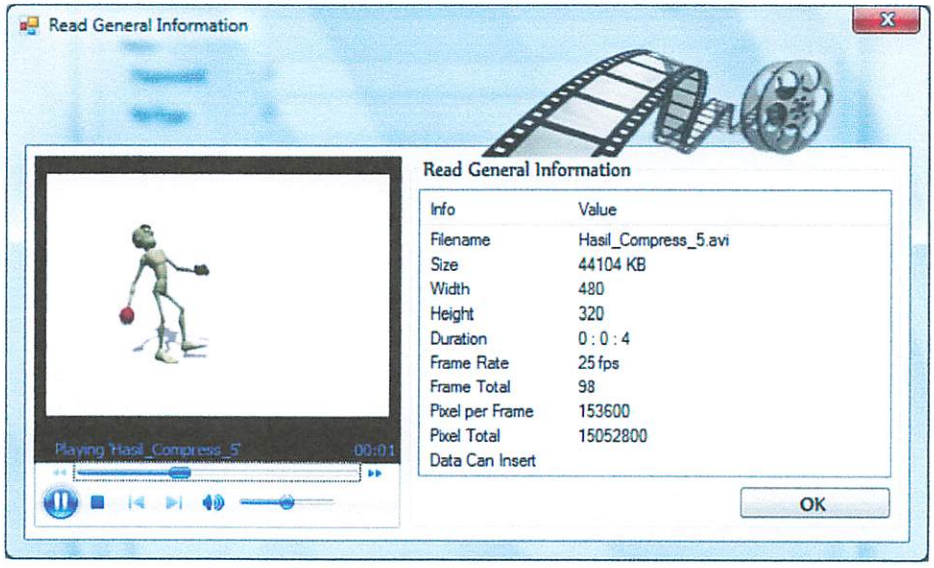
File video avi dengan nama Video 480x272_compress.avi sebelum disisipi pesan mempunyai ukuran 205 KB dengan resolusi 480x272, *frame* total 99 dengan *frame rate* 25 fps, *pixel* total 12925440 dengan *pixel per frame* 130560, setelah disisipi pesan disimpan dengan nama Hasil_Compress_4.avi mempunyai ukuran 37872 KB dengan resolusi 480x272, *frame* total 99 dengan *frame rate* 25 fps, *pixel* total 12925440 dengan *pixel per frame* 130560.

4.2.4.1.5 Video Dengan Resolusi 480x320

Pembacaan atribut meliputi ukuran dari video, resolusi, durasi, *frame rate*, *frame total*, *pixel per frame*, dan *pixel total*.



Gambar 4.38 Atribut Video Sebelum Disisipi Pesan



Gambar 4.39 Atribut Video Sesudah Disisipi Pesan

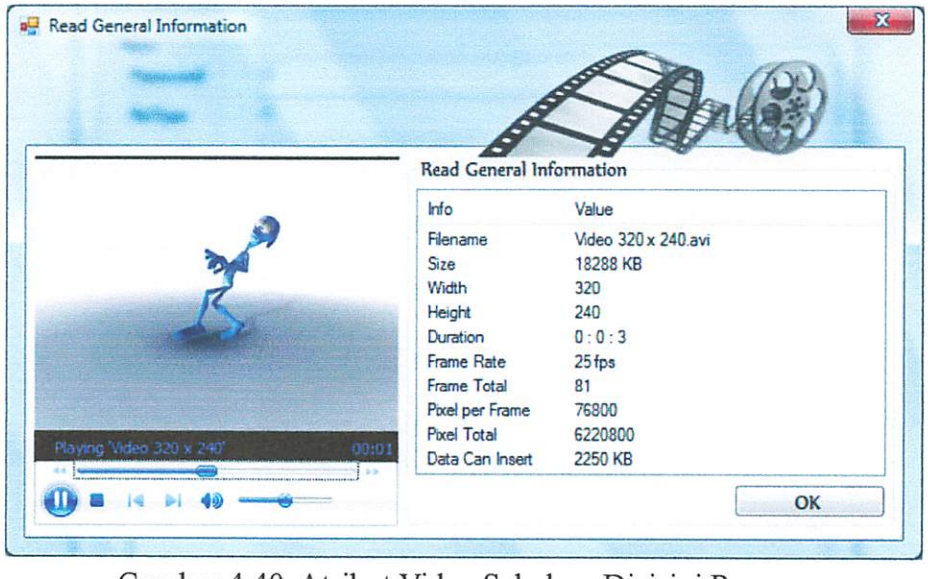
File video avi dengan nama Video 480x320_compress.avi sebelum disisipi pesan mempunyai ukuran 185 KB dengan resolusi 480x320, *frame* total 98 dengan *frame rate* 25 fps, *pixel* total 15052800 dengan *pixel per frame* 153600, setelah disisipi pesan disimpan dengan nama Hasil_Compress_5.avi mempunyai ukuran 44104 KB dengan resolusi 480x320, *frame* total 98 dengan *frame rate* 25 fps, *pixel* total 15052800 dengan *pixel per frame* 153600.

Dari beberapa pengujian diatas dapat terlihat atribut yang meliputi resolusi, durasi, *frame rate*,*frame total*, *pixel per frame*, dan *pixel total* dari video sebelum penyisipan dan sesudah penyisipan tidak mengalami perubahan meskipun telah mengalami proses steganografi dengan tersimpannya data digital didalamnya. Kecuali untuk ukuran (*size*) dari video, sebelum dilakukan proses penyisipan ukurannya kecil, setelah dilakukan proses penyisipan terjadi pembengkakan pada ukurannya yang besar. Hal ini dikarenakan file video yang dihasilkan dari aplikasi ini adalah video yang tidak terkompresi.

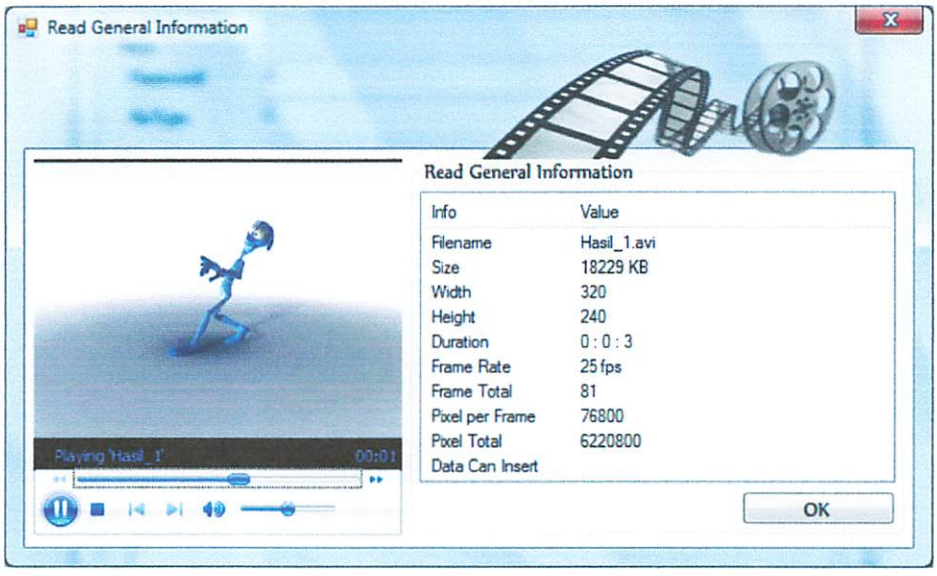
4.2.4.2 Video Yang Tidak Dikompresi

4.2.4.2.1 Video Dengan Resolusi 320x240

Pembacaan atribut meliputi ukuran dari video, resolusi, durasi, *frame rate*, *frame total*, *pixel per frame*, dan *pixel total*.



Gambar 4.40 Atribut Video Sebelum Disisipi Pesan

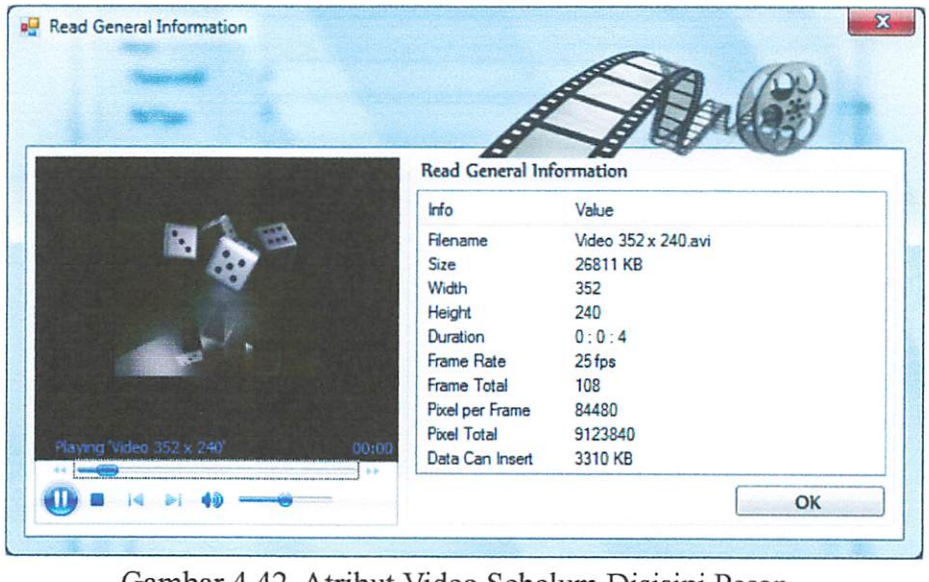


Gambar 4.41 Atribut Video Sesudah Disisipi Pesan

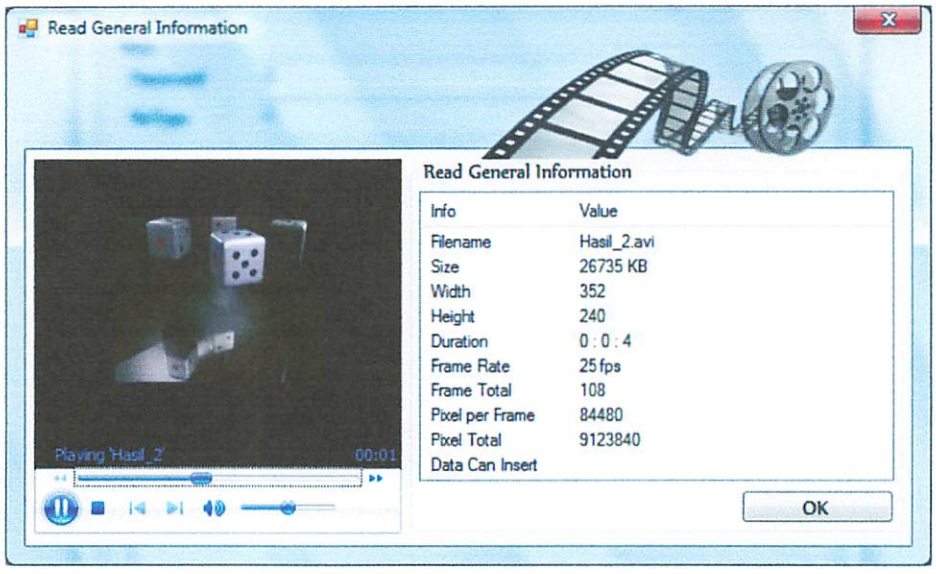
File video avi dengan nama Video 320x240.avi sebelum disisipi pesan mempunyai ukuran 18288KB dengan resolusi 320x240, *frame* total 81 dengan *frame rate* 25 fps, *pixel* total 6220800 dengan *pixel per frame*76800, setelah disisipi pesan disimpan dengan nama Hasil_1.avi mempunyai ukuran 18229KB dengan resolusi 340x240, *frame* total 81 dengan *frame rate* 25 fps, *pixel* total 6220800 dengan *pixel per frame*76800.

4.2.4.2.2 Video Dengan Resolusi 352x240

Pembacaan atribut meliputi ukuran dari video, resolusi, durasi, *frame rate*, *frame total*, *pixel per frame*, dan *pixel total*.



Gambar 4.42 Atribut Video Sebelum Disisipi Pesan

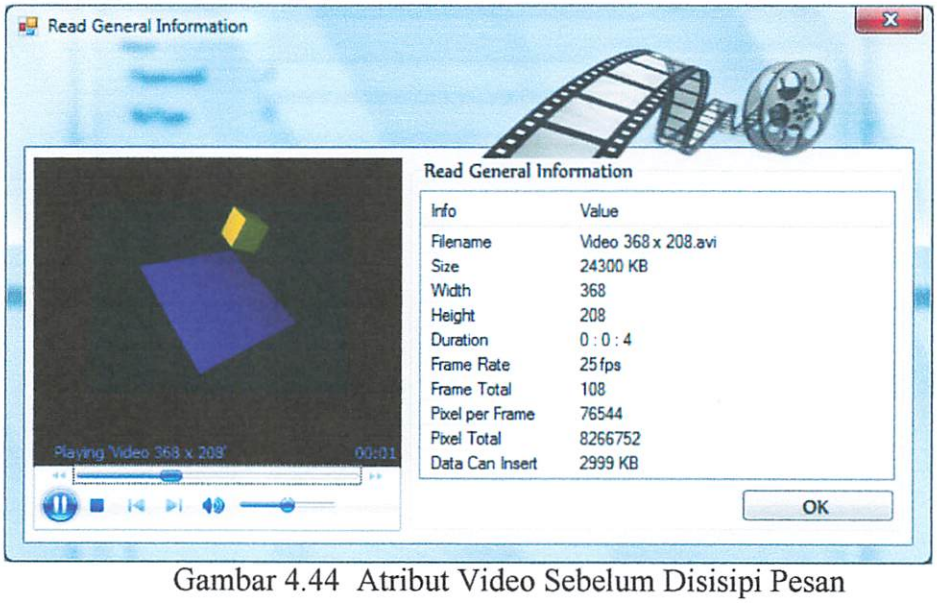


Gambar 4.43 Atribut Video Sesudah Disisipi Pesan

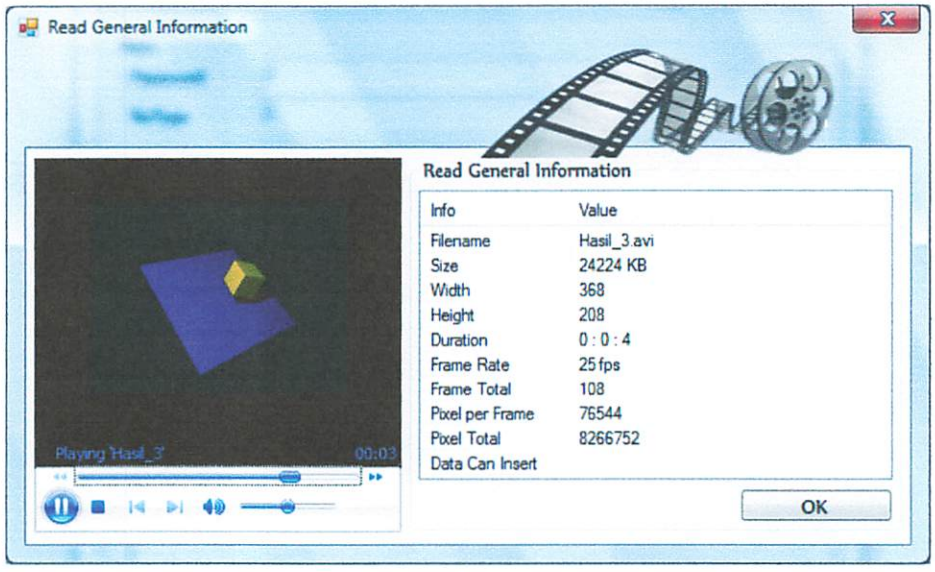
File video avi dengan nama Video 352x240.avi sebelum disisipi pesan mempunyai ukuran 26811 KB dengan resolusi 352x240, *frame* total 108 dengan *frame rate* 25 fps, *pixel* total 9123840 dengan *pixel per frame* 84480, setelah disisipi pesan disimpan dengan nama Hasil_2.avi mempunyai ukuran 26735 KB dengan resolusi 352x240, *frame* total 108 dengan *frame rate* 25 fps, *pixel* total 9123840 dengan *pixel per frame* 84480.

4.2.4.2.3 Video Dengan Resolusi 368x208

Pembacaan atribut meliputi ukuran dari video, resolusi, durasi, *frame rate*, *frame total*, *pixel per frame*, dan *pixel total*.



Gambar 4.44 Atribut Video Sebelum Disisipi Pesan

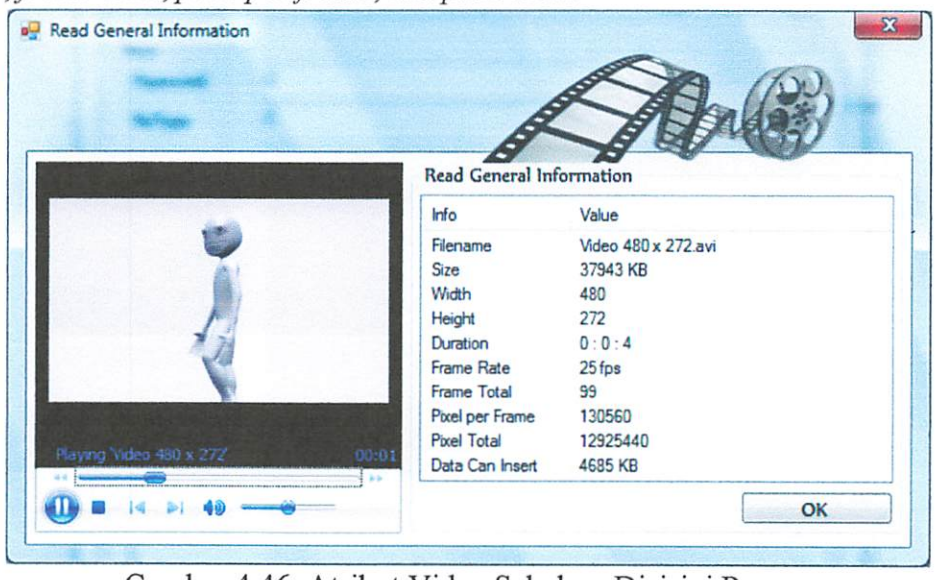


Gambar 4.45 Atribut Video Sesudah Disisipi Pesan

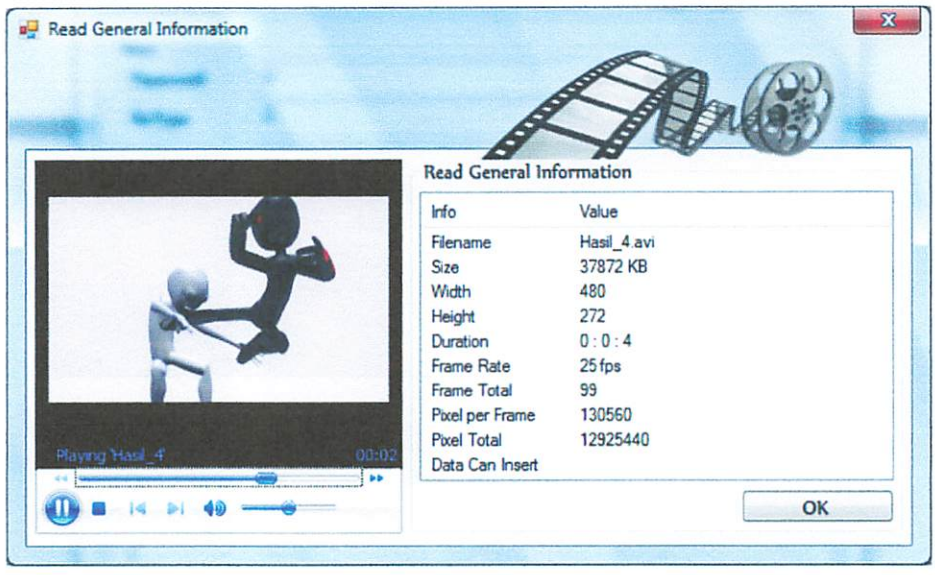
File video avi dengan nama Video 368x208.avi sebelum disisipi pesan mempunyai ukuran 24300 KB dengan resolusi 368x208, *frame* total 108 dengan *frame rate* 25 fps, *pixel* total 8266752 dengan *pixel per frame* 76544, setelah disisipi pesan disimpan dengan nama Hasil_3.avi mempunyai ukuran 24224 KB dengan resolusi 368x208, *frame* total 108 dengan *frame rate* 25 fps, *pixel* total 8266752 dengan *pixel per frame* 76544.

4.2.4.2.4 Video Dengan Resolusi 480x272

Pembacaan atribut meliputi ukuran dari video, resolusi, durasi, *frame rate*, *frame total*, *pixel per frame*, dan *pixel total*.



Gambar 4.46 Atribut Video Sebelum Disisipi Pesan

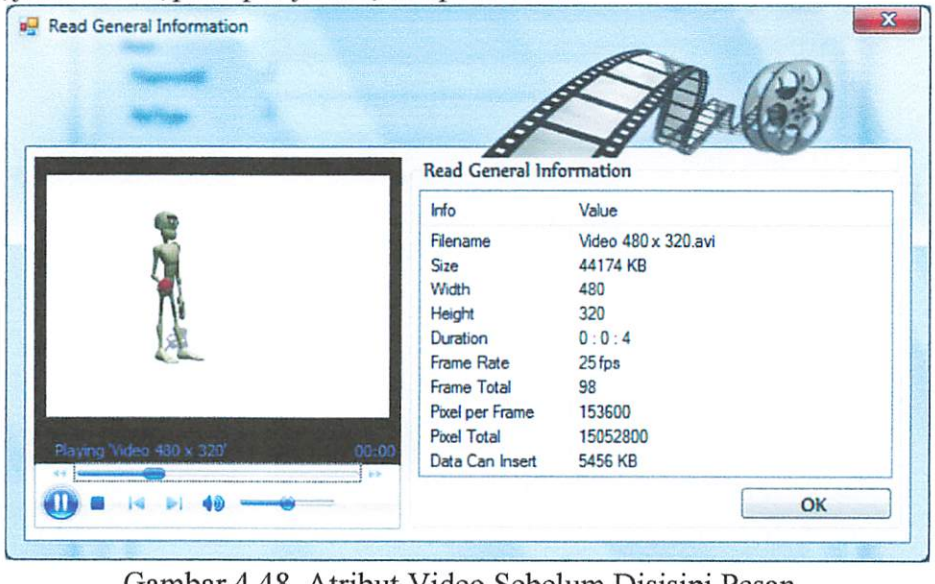


Gambar 4.47 Atribut Video Sesudah Disisipi Pesan

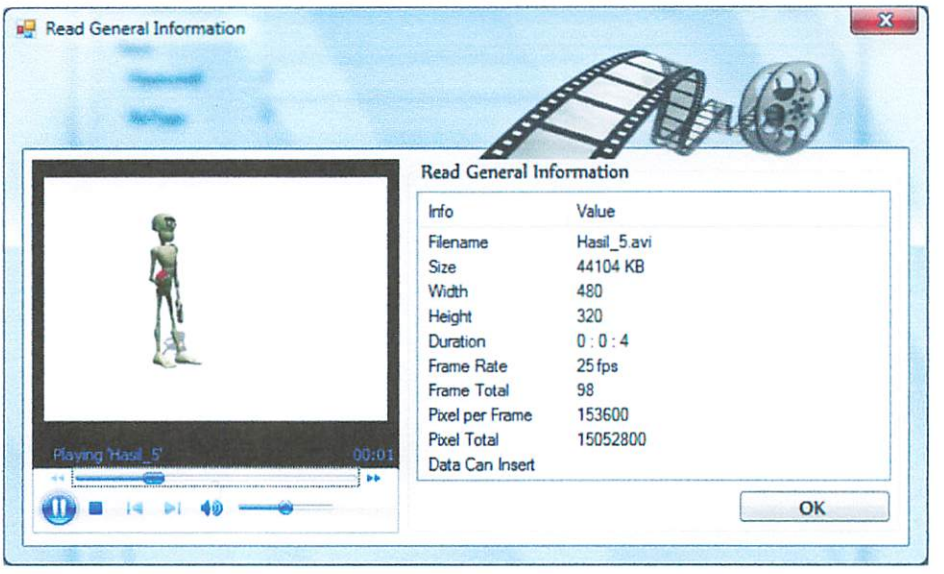
File video avi dengan nama Video 480x272.avi sebelum disisipi pesan mempunyai ukuran 37943 KB dengan resolusi 480x272, *frame* total 99 dengan *frame rate* 25 fps, *pixel* total 12925440 dengan *pixel per frame* 130560, setelah disisipi pesan disimpan dengan nama Hasil_4.avi mempunyai ukuran 37872 KB dengan resolusi 480x272, *frame* total 99 dengan *frame rate* 25 fps, *pixel* total 12925440 dengan *pixel per frame* 130560.

4.2.4.2.5 Video Dengan Resolusi 480x320

Pembacaan atribut meliputi ukuran dari video, resolusi, durasi, *frame rate*, *frame total*, *pixel per frame*, dan *pixel total*.



Gambar 4.48 Atribut Video Sebelum Disisipi Pesan



Gambar 4.49 Atribut Video Sesudah Disisipi Pesan

File video avi dengan nama Video 480x320.avi sebelum disisipi pesan mempunyai ukuran 44174 KB dengan resolusi 480x320, *frame* total 98 dengan *frame rate* 25 fps, *pixel* total 15052800 dengan *pixel per frame*153600, setelah disisipi pesan disimpan dengan nama Hasil_5.avi mempunyai ukuran 44104 KB dengan resolusi 480x320, *frame* total 98 dengan *frame rate* 25 fps, *pixel* total 15052800 dengan *pixel per frame*153600.

Dari beberapa pengujian diatas dapat terlihat atribut yang meliputi resolusi, durasi, *frame rate*,*frame total*, *pixel per frame*, dan *pixel total* dari video sebelum penyisipan dan sesudah penyisipan tidak mengalami perubahan meskipun telah mengalami proses steganografi dengan tersimpannya data digital didalamnya. Kecuali untuk ukuran (*size*) dari video, pada video setelah proses penyisipan terjadi penyusutan yang tidak signifikan dari ukuran video sebelum dilakukan proses penyisipan.

4.2.5 Pengujian Terhadap Data Yang Ukurannya Lebih Kecil Atau Lebih Besar Dari Ukuran Yang Diperbolehkan

Pada pengujian ini dilakukan proses steganografi terhadap data yang ukurannya lebih kecil dan lebih besar dari ukuran yang diperbolehkan pada proses steganografi. Pada pengujian ini di gunakan video dengan nama movie.avi, dengan ukuran 252 KB, yang memiliki resolusi 160x116, mempunyai

frame total 4 dengan *framerate* 2 fps, pixel total 71680 dengan pixel per frame 17920, dengan data yang boleh disisipkan sebesar 19 KB.

Tabel 4.7 Hasil Pengujian Terhadap Data Yang Ukurannya Lebih Kecil

No.	Info Berkas		Keterangan Proses	
	Format	Ukuran	Penyisipan	Ekstraksi
1	*.txt	6.25 KB	Berhasil	Berhasil, Data hasil Ekstraksi sama dengan data asli.
2	*.docx	13.1 KB	Berhasil	Berhasil, Data hasil Ekstraksi sama dengan data asli.
3	*.xlsx	11.3 KB	Berhasil	Berhasil, Data hasil Ekstraksi sama dengan data asli.

Tabel 4.8 Hasil Pengujian Terhadap Data Yang Ukurannya Lebih Besar

No.	Info Berkas		Keterangan Proses	
	Format	Ukuran	Penyisipan	Ekstraksi
1	*.txt	20.0 KB	Berhasil	Berhasil, Data hasil Ekstraksi tidak sama dengan data asli.
2	*.docx	22.5 KB	Berhasil	Berhasil, Data hasil Ekstraksi *corrupt*
3	*.xlsx	21.3 KB	Berhasil	Berhasil, Data hasil Ekstraksi *corrupt*

Dari beberapa pengujian diatas dapat terlihat bahwa apabila file atau data yang akan disisipkan mempunyai ukuran yang lebih besar dari ukuran yang diperbolehkan, maka data tersebut apabila dilakukan proses ekstraksi, data tersebut akan *corrupt,* dikarenakan bit-bit dari pesan tidak semuanya disisipkan kedalam bit-bit dari video.

BAB V
PENUTUP

5.1 Kesimpulan

Setelah melakukan analisis dan implementasi sistem yang dilanjutkan dengan pengujian sistem, maka dari hasil implementasi dan pengujian tersebut maka dapat diambil beberapa kesimpulan yaitu:

1. Aplikasi *steganography* ini dapat dilakukan pada video beresolusi 320x240, 352x240, 368x208, 480x272, dan 480x320.
2. Semakin banyak jumlah *frame* dan juga jumlah *framerate* dari file avi mengakibatkan proses penyisipan dan ekstraksi file membutuhkan waktu yang lebih lama.
3. Data digital yang dihasilkan dari hasil ekstraksi tidak berubah baik isi maupun besar ukurannya sama seperti dengan file berkas asli yang disisipkan kedalam video, kecuali untuk data digital yang berekstensi (*.docx), jika file tersebut mengandung image dan format *sub numbering*, maka image yang ada akan rusak, dan diperlukan proses *recovery* untuk membuka file tersebut.
4. Untuk file video yang terkompresi akan dihasilkan file video yang berukuran sangat besar setelah dilakukan proses penyisipan. Sedangkan untuk file video yang tidak terkompresi akan dihasilkan video yang ukurannya tidak berubah secara signifikan.
5. Aplikasi ini baik digunakan untuk video yang tidak terkompresi. Karena dengan tidak berubahnya kualitas dan ukuran video stego yang signifikan dari video aslinya, memungkinkan orang lain tidak akan mengetahui bahwa ada suatu informasi didalam video tersebut.
6. Dengan menggunakan password pada proses penyisipan dan ekstraksi maka keamanan data digital yang disisipkan pada video dapat terjaga dengan aman.

5.2 Saran

Aplikasi steganography video yang dibangun ini masih jauh dari kata sempurna. Aplikasi ini masih perlu dikembangkan lagi supaya dapat menyisipkan pada berbagai format video yang sudah terkompres dan dapat menyisipkan berkas rahasia dalam ukuran yang besar, disamping itu penggunaan enkripsi pada file yang akan disisipkan, akan dapat lebih meningkatkan keamanan data.

DAFTAR PUSTAKA

[1] Hirin, A. M. (2011). *Belajar Tuntas VB.net 2010.* Jakarta: Penerbit PT. Prestasi Pustakaraya.

[2] J. Anderson, Ross (2001). *Security Engineering: A Guide to Building Dependable Distributed System.* Canada: John Wiley & Sons, Inc.

[3] Prasetyo, Deni (2009). *Perencanaan dan Implementasi Steganografi Citra Digital Pada Telepon Seluler.* Tugas Akhir 2009, ITN Malang.

[4] Purwanto, Heri (2009). *Penggunaan Steganografi Untuk Menyembunyikan Pesan Text Kedalam File Audio Berformat MP3.* Tugas Akhir 2009, ITN Malang.

[5] Putra, Darma (2010). *Pengolahan Citra Digital.* Yogyakarta: Penerbit Andi Offset.

[6] Sadeli, Muhammad (2009). *7 Jam Belajar Interaktif Visual Basic 2008 untuk Orang Awam.* Palembang: Maxikom.

[7] A. Riazi, Shafiee (2004). *Extracting AVI Frames.* From http://www.codeproject.com/KB/audio-video/ExtractAVIFrames.aspx?msg=2519906, 15 Oktober 2011.

[8] Alatas, Putri (2009). *Implementasi Teknik Steganografi Dengan Metode Lsb Pada Citra Digital.* From www.gunadarma.ac.id/library/articles/.../Artikel_11104284.pdf, *23 Juli 2011*

[9] Andreyono Sanjaya, Vicky (2006). *Pembuatan Perangkat Lunak Untuk Memperbaiki Citra Pada Video Digital.* From *digilib.petra.ac.id/.../jiunkpe-ns-s1-2006-26402150-7989-video_digital-chapter2.pdf, 13 Oktober2011.*

[10] Corinna John (2006). *A Simple C# Wrapper for the AviFile Library.* From http://www.codeproject.com/Articles/7388/A-Simple-C-Wrapper-for-the-AviFile-Library, 15 Oktober 2011.

[11] Henry, (2006). Video Steganography. From budi.insan.co.id/courses/security/2006/henry_report.pdf, 23 juli 2011.

[12] MSDN Library, http://msdn.microsoft.com/library/default.aspx, 23 Juli 2011.

[13] Steganografi, From http://id.wikipedia.org/wiki/Steganografi, 23 Juli 2011.

LAMPIRAN

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK
PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG
Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

BERITA ACARA UJIAN SKRIPSI
FAKULTAS TEKNOLOGI INDUSTRI

NAMA : ACHMAD TORIQUL HUDA
NIM : 07.12.577
JURUSAN : Teknik Elektro S-1
KONSENTRASI : Teknik Komputer dan Informatika
MASA BIMBINGAN : 15 Desember 2011 s/d 27 Juni 2012
JUDUL : **APLIKASI PENGAMANAN DATA DIGITAL DALAM MEDIA AVI MENGGUNAKAN TEKNIK *STEGANOGRAPHY***

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :

Hari : Kamis
Tanggal : 16 Februari 2012
Dengan Nilai : 83,5 (A)

PANITIA UJIAN SKRIPSI

Ketua Majelis Penguji,

Ir. Yusuf Ismail Nakhoda, MT
NIP.Y.1018800189

Sekretaris Majelis Penguji,

Dr. Eng. Aryuanto S, ST, MT
NIP.Y.1030800417

ANGGOTA PENGUJI

Dosen Penguji I

Sandy Nataly Mantja, S.Kom
NIP.P.1030800418

Dosen Penguji II

M. Ibrahim Ashari, ST, MT
NIP.P.1030100358

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK
PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG
Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata Satu (S-1) Jurusan Teknik Elektro Konsentrasi Teknik Komputer dan Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa:

NAMA : ACHMAD TORIQUL HUDA
NIM : 07.12.577
JURUSAN : Teknik Elektro S-1
KONSENTRASI : Teknik Komputer dan Informatika
MASA BIMBINGAN : 27 Desember 2011 s/d 27 Juni 2012
JUDUL : **APLIKASI PENGAMANAN DATA DIGITAL DALAM MEDIA AVI MENGGUNAKAN TEKNIK STEGANOGRAPHY**

No	Tanggal	Uraian	Paraf
1	Penguji I 16 - 02 - 2012	Bab II ditambahkan landasan teori tentang steganography video dengan menggunakan metode DCT, selanjutnya pada bab IV dilakukan perbandingan antara metode DCT dengan metode LSB dan ditulis pada kesimpulan	
		Ditambahkan kesimpulan tentang hal. 76 tergantung pada looping.	
		Daftar pustaka urut abjad dan dari website di bagian bawah	
		Ditambahkan pada pengujian tentang data yang lebih kecil atau lebih besar dalam bentuk tabel	
2	Penguji II 16 - 02 - 2012	Tambahkan keterangan pada gambar dan tabel.	

Disetujui,

Dosen Penguji I

Sandy Nataly Mantja, S.Kom
NIP.P.1030800418

Dosen Penguji II

M. Ibrahim Ashari, ST, MT
NIP.P.1030100358

Mengetahui,

Dosen Pembimbing I

Joseph Dedy Irawan, ST, MT
NIP. 19740416 200501 1 002

Dosen Pembimbing II

Sotyohadi, ST
NIP.Y.1039700309

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / T.Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : ACHMAD TORIQUL HUDA
NIM : 0712577
Perbaikan meliputi :

1. DIBAB IV TAMBAHKAN UJICOBA PEN DCT DAN DIBANDINGKAN MANA YG LEBIH AKURAT / BIDA DIBANDING LSB. DAN DITULIS DI KESIMPULAN BAB V JUGA ABSTRAKSI.

2. TAMBAHKAN DI KETIMPULAN TTG HAL 76 TERGANTUNG LOOPING

3. DAFTAR PUSTAKA URUT ABJAD DAN DARI WEBSITE DI BAB BELAKANG

4. TAMBAHKAN PADA PENGUJIAN TTG DATA YG LEBIH KECIL ATAU LEBIH BESAR DLM BENTUK TABEL.

Malang,

(SANDY NATALY)

INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

Formulir Perbaikan Ujian Skripsi

Dalam pelaksanaan Ujian Skripsi Janjang Strata 1 Jurusan Teknik Elektro Konsentrasi T. Energi Listrik / T. Elektronika / T.Infokom, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : Achmad toriqul Huda .
NIM : 071577
Perbaikan meliputi :

tambahkan keterangan pd gambar dan tabel

Malang,

(M. Ibrahim Ashari, ST MT)

INSTITUT TEKNOLOGI NASIONAL
Jl. Raya Karanglo, Km. 2
MALANG

PERMOHONAN PERSETUJUAN SKRIPSI

Yang betanda tangan dibawah ini :

N a m a	: ACHMAD TORIQUL HUDA
N I M	: 07.12.577
Semester	: VIII
Fakultas	: Teknologi Industri
Jurusan	: Teknik Elektro S-1
Konsentrasi	: ~~TEKNIK ELEKTRONIKA~~ ~~TEKNIK ENERGI LISTRIK~~ **TEKNIK KOMPUTER DAN INFORMATIKA** ~~TEKNIK KOMPUTER~~ ~~TEKNIK TELEKOMUNIKASI~~
Alamat	: JL. KI AGENG GRIBIG 67 MALANG

Dengan ini kami mengajukan permohonan untuk mendapatkan persetujuan untuk membuat ***SKRIPSI Tingkat Sarjana.*** Untuk melengkapi permohonan tersebut, bersama kami lampirkan persyaratan-persyaratan yang harus dipenuhi.
Adapun persyaratan-persyaratan pengambilan ***SKRIPSI*** adalah sebagai berikut :

1. Telah melaksanakan semua praktikum sesuai dengan konsentrasinya (...✓...)
2. Telah lulus dan menyerahkan Laporan Praktek Kerja (...✓...)
3. Telah lulus seluruh mata kuliah keahlian (MKB) sesuai konsentrasinya (...✓...)
4. Telah menempuh mata kuliah ≥ 134 sks dengan IPK ≥ 2 dan tidak ada nilai E (...✓...)
5. Telah mengikuti secara aktif kegiatan seminar skripsi yang diadakan Jurusan (.....)
6. Memenuhi persyaratan administrasi (.....)

Demikian permohonan ini untuk mendapatkan penyelesaian lebih lanjut dan atas perhatiannya kami ucapkan terima kasih.

Telah diteliti kebenaran data tersebut diatas
Recording Teknik Elektro

(.....)

Malang, 04-04.....2011
Pemohon

(ACHMAD TORIQUL HUDA)

Disetujui
Ketua Jurusan Teknik Elektro

Ir. Yusuf Ismail Nakhoda, MT
NIP. Y. 1018800189

Mengetahui
Dosen Wali

(.....)

Catatan :
Bagi mahasiswa yang telah memenuhi persyaratan mengambil SKRIPSI agar membuat proposal dan mendapat persetujuan dari Ketua Jurusan/Sekretaris Jurusan T. Elektro S-1

1. 463/138 : 3.52
2. .....
3. ~~Praktikum~~ Lengkap

Form. S-1a

JURUSAN TEKNIK ELEKTRO S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG

DAFTAR PRESTASI AKADEMIK PRAKTIKUM
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA

Nama Mahasiswa	:	ACHMAD TORIQUL HUDA
NIM	:	09 12 577
Tempat, Tanggal Lahir	:	MALANG, 21 FEBRUARI 1988
Jenjang	:	Strata 1 (S1)
Fakultas	:	Teknologi Industri
Jurusan / Program Studi	:	Teknik Elektro
Konsentrasi	:	Teknik Komputer dan Informatika

Praktikum Laboratorium	Kode	Nama Praktikum	SKS	Nilai
I	EL-2215	Fisika	1	B+
		Rangkaian Listrik		B+
		Rangkaian Logika dan Digital		B+
	27	Dasar Komputer dan Pemrograman		B
II	EL-4216	Dasar Elektronika	1	B+
		Dasar Sistem Telekomunikasi		B+
		Mikrokontroler		A
	28	Sistem Pengukuran		B+
		Dasar Sistem Kendali		A
III	EL-5316	Basis Data	1	B
		Administrasi Jaringan		B+
	19	Sistem Operasi		B
IV	EL-6317	Pemrograman Internet	1	B
		Pemrograman Objek		B
	29	Rekayasa Perangkat Lunak Sistem Informasi		A
V	EL-7318	Peripheral dan Antar Muka	1	B
		Pemrosesan Sinyal Digital		B
		Multimedia		B
PKN +	29	Pemrograman Jaringan		B+

95 = 1;10

Malang, _____

Recording
Jurusan Teknik Elektro S1

Puji Handayani

Institut Teknologi Nasional Malang
Fakultas Teknologi Industri
Jurusan Teknik Elektro S-1

LEMBAR PENGAJUAN JUDUL SKRIPSI
JURUSAN TEKNIK ELEKTRO S-1

Konsentrasi : Teknik Energi Listrik / Teknik Elektronika /Teknik Komputer & Informatika /Teknik Komputer / Teknik Telekomunikasi*)

1.	Nama Mahasiswa: Achmad Toeiqul Huda			Nim: 0712577
2.	Waktu Pengajuan	Tanggal: 26	Bulan: April	Tahun: 2011
3.	Spesifikasi Judul (berilah tanda silang)**) a. Sistem Tenaga Elektrik b. Energi & Konversi Energi c. Tegangan Tinggi & Pengukuran d. Sistem Kendali Industri e. Elektronika & Komponen f. Elektronika Digital & Komputer g. Elektronika Komunikasi h. lainnya			
4.	Konsultasikan judul sesuai materi bidang ilmu kepada Dosen*) Dr. Aryuanto, ST, MT	Ketua Jurusan Ir. Yusuf Ismail Nakhoda,MT NIP. Y. 1018800189		
5.	Judul yang diajukan mahasiswa:	Aplikasi Pengamanan Data Digital Dalam Media AVI Menggunakan Teknik Steganography		
6.	Perubahan judul yang disetujui Dosen sesuai materi bidang ilmu	.....		
7.	Catatan: .... Persetujuan Judul skripsi yang dikonsultasikan kepada Dosen materi bidang ilmu	Disetujui 28/4/2011 Dosen		

Perhatian:
1. Formulir pengajuan ini harap dikembalikan kepada jurusan paling lambat satu minggu setelah disetujui kelompok dosen keahlian dengan dilampirkan proposal skripsi beserta persyaratan skripsi sesuai form S-1
2. Keterangan: *) Coret yang tidak perlu
   **) dilingkari a, b, c, ...atau g sesuai bidang keahlian

Form S-2

INSTITUT TEKNOLOGI NASIONAL
JL. Raya Karanglo, Km. 2
MALANG

Lampiran : 1 (satu) berkas

Pembimbing Skripsi

Kepada : Yth. Bapak Joseph Dedy Irawan,ST,MT

Dosen Institut Teknologi Nasional Malang

Yang bertanda tangan di bawah ini:

Nama : Achmad Toriqul Huda

NIM : 07.12.577

Jurusan : Teknik Elektro S-1

Konsentrasi : Teknik Komputer dan Informatika

Dengan ini mengajukan permohonan, kiranya Bapak bersedia menjadi Dosen Pembimbing Utama untuk peyusunan Skripsi dengan judul (proposal terlampir):

" Aplikasi Pengamanan Data Digital Dalam Media AVI Menggunakan Teknik *Steganography* "

Adapun tugas tersebut sebagai salah satu syarat untuk menempuh Ujian Akhir Sarjana Teknik.

Demikian permohonan kami buat dan atas kesediaaan Bapak kami ucapkan terima kasih.

Mengetahui

Ketua Jurusan Teknik Elektro

Ir.Yusuf Ismail Nakhoda,MT
NIP. Y. 101.880.0189

Malang, 4 Mei 2011

Hormat kami

Achmad Toriqul Huda

Form S-3a

INSTITUT TEKNOLOGI NASIONAL
JL. Raya Karanglo, Km. 2
MALANG

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i:

Nama : Achmad Toriqul Huda

NIM : 07.12.577

Semester : VIII (Delapan)

Jurusan : Teknik Elektro S-1

Konsentrasi : Teknik Komputer dan Informatika

Dengan ini menyatakan bersedia/~~tidak bersedia~~*) Membimbing skripsi dari mahasiswa tersebut, dengan judul:

" Aplikasi Pengamanan Data Digital Dalam Media AVI Menggunakan Teknik *Steganography* "

Demikian surat pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Malang, Mei 2011
Kami yang membuat pernyataan

Joseph Dedy Irawan, ST, MT
NIP. 19740416 200501 1 002

Catatan :
Setelah disetujui agar formulir ini
Diserahkan mahasiswa/I yang bersangkutan
Kepada Jurusan untuk diproses lebih lanjut.
*) Coret yang tidak perlu

Form S-3b

INSTITUT TEKNOLOGI NASIONAL
JL. Raya Karanglo, Km. 2
MALANG

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa/i:

Nama : Achmad Toriqul Huda

NIM : 07.12.577

Semester : VIII (Delapan)

Jurusan : Teknik Elektro S-1

Konsentrasi : Teknik Komputer dan Informatika

Dengan ini menyatakan bersedia/ tidak bersedia*) Membimbing skripsi dari mahasiswa tersebut, dengan judul:

" Aplikasi Pengamanan Data Digital Dalam Media AVI Menggunakan Teknik *Steganography* "

Demikian surat pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Malang, Mei 2011
Kami yang membuat pernyataan

Sotyohadi, ST
NIP.Y. 1039700309

Catatan :
Setelah disetujui agar formulir ini
Diserahkan mahasiswa/I yang bersangkutan
Kepada Jurusan untuk diproses lebih lanjut.
*) Coret yang tidak perlu

Form S-3b

Institut Teknologi Nasional Malang
Fakultas Teknologi Industri
Jurusan Teknik Elektro S-1

BERITA ACARA SEMINAR PROPOSAL SKRIPSI
JURUSAN TEKNIK ELEKTRO S-1

Konsentrasi : Teknik Energi Listrik/Teknik Elektronika/ Teknik Komputer & Informatika*)

1.	Nama Mahasiswa: Achmad Toriqul Huda			Nim: 07 12 577
2.	Keterangan	Tanggal	Waktu	Tempat
	Pelaksanaan	24-05-2011	09.00- selesai	Ruang:
3.	Spesifikasi Judul (berilah tanda silang)**)			
	a. Sistem Tenaga Elektrik		e. Elektronika & Komponen	
	b. Energi & Konversi Energi		f. Elektronika Digital & Komputer	
	c. Tegangan Tinggi & Pengukuran		g. Elektronika Komunikasi	
	d. Sistem Kendali Industri		h. lainnya	
4.	Judul Proposal yang diseminarkan Mahasiswa	APLIKASI PENGAMANAN DATA DIGITAL DALAM MEDIA AVI MENGGUNAKAN TEKNIK STEGANOGRAPHY		
5.	Perubahan Judul yang diusulkan oleh Kelompok Dosen Keahlian	.....		
6.	Catatan:			
7.	Catatan:			

Persetujuan Judul Skripsi

Disetujui, Dosen Keahlian I	Disetujui, Dosen Keahlian II A. Fauzi, ST
Mengetahui, Ketua Jurusan. Ir. Yusuf Ismail Nakhoda,MT NIP. Y. 1018800189	Disetujui, Calon Dosen Pembimbing ybs Pembimbing I: ..... Pembimbing II: .....

Perhatian:
1. Keterangan: *) Coret yang tidak perlu
   **) dilingkari a, b, c, ...atau g sesuai bidang keahlian

Form S-3c

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK
PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG
Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 24 Nopember 2011

Nomor : ITN-805/I.TA/2/11
Lampiran : -
Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Sdr./i . **JOSEPH DEDY IRAWAN, ST, MT**
Dosen Institut Teknologi Nasional Malang

Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

Dengan hormat
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi
Untuk **Mahasiswa :**

Nama	: ACHMAD TORIQUL. H
Nim	: 0712577
Fakultas	: Teknologi Industri
Jurusan	: Teknik Elektro S-1
Konsentrasi	: Teknik **Komputer & Informatika**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu (enam) 6 bulan, terhitung mulai tanggal :

15 November 2011 s/d 15 April 2012

Sebagai satu syarat untuk menempuh ujian Sarjana Teknik,
Jurusan Teknik Elektro S-1
Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih

Ketua Jurusan
Teknik Elektro S-1

Ir. Yusuf Ismail Nakhoda, MT
Nip. Y 1018800189

Tembusan Kepada Yth :
1. Mahasiswa Yang Bersangkutan
2. Arsip
3. Coret yang tidak perlu

Form. S 4a

PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG
INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
PROGRAM PASCASARJANA MAGISTER TEKNIK
PT. BNI (PERSERO) MALANG
BANK NIAGA MALANG
Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

Malang, 24 Nopember 2011

Nomor : ITN-806/I.TA/2/11
Lampiran : -
Perihal : BIMBINGAN SKRIPSI

Kepada : Yth. Sdr./i . **SOTYOHADI, ST**
Dosen Institut Teknologi Nasional Malang

Dosen Pembimbing
Jurusan Teknik Elektro S-1
di
Malang

Dengan hormat
Sesuai dengan permohonan dan persetujuan dalam Proposal Skripsi
Untuk Mahasiswa :

Nama	: ACHMAD TORIQUL. H
Nim	: 0712577
Fakultas	: Teknologi Industri
Jurusan	: Teknik Elektro S-1
Konsentrasi	: Teknik **Komputer & Informatika**

Maka dengan ini pembimbingan tersebut kami serahkan sepenuhnya kepada Saudara/i selama masa waktu (enam) 6 bulan, terhitung mulai tanggal :

15 November 2011 s/d 15 April 2012

Sebagai satu syarat untuk menempuh ujian Sarjana Teknik,
Jurusan Teknik Elektro S-1
Demikian agar maklum dan atas perhatian serta bantuannya kami sampaikan terima kasih

Ketua Jurusan
Teknik Elektro S-1

Ir. Yusuf Ismail Nakhoda, MT
Nip. Y.1018800189

Tembusan Kepada Yth :
1. Mahasiswa Yang Bersangkutan
2. Arsip
3. Coret yang tidak perlu

Form. S 4a

INSTITUT TEKNOLOGI NASIONAL
Jl. Raya Karanglo, KM 2
MALANG

FORMULIR BIMBINGAN SKRIPSI

Nama : Achmad Toriqul Huda
Nim : 07.12.577
Masa Bimbingan : 15 November 2011 – 15 Mei 2012
Judul Skripsi : Aplikasi Pengamanan Data Digital Dalam Media AVI Menggunakan Teknik *Steganography*

No	Tanggal	Uraian	Paraf Pembimbing
1	15/12/2011	Demo Program	
2	21/12/2011	Konsultasi Revisi Program	
3	7/01/2011	Konsultasi Revisi Program	
4	16/01/2011	Konsultasi Laporan Bab I, II, III	
5	19/01/2011	Konsultasi laporan Bab IV, V	
6	26/01/2011	ACC seminar	
7		ACC kompre	
8			
9			
10			

Malang,
Dosen Pembimbing I

Joseph Dedy Irawan, ST, MT
NIP. 19740416 200501 1 002

Form S-4b

INSTITUT TEKNOLOGI NASIONAL
Jl. Raya Karanglo, KM 2
MALANG

FORMULIR BIMBINGAN SKRIPSI

Nama : Achmad Toriqul Huda
Nim : 07.12.577
Masa Bimbingan : 15 November 2011 – 15 Mei 2012
Judul Skripsi : Aplikasi Pengamanan Data Digital Dalam Media AVI Menggunakan Teknik *Steganography*

No	Tanggal	Uraian	Paraf Pembimbing
1	16/01/2012	Demo Program	
2	16/01/2012	Konsultasi Laporan Bab I, II, III	
3	19/01/2012	Konsultasi Revisi Laporan Bab I, III	
4	19/01/2012	Konsultasi Laporan Bab IV, V	
5	26/01/2012	Acc Makalah Seminar Hasil	
6			
7			
8			
9			
10			

Malang,
Dosen Pembimbing II

Sotyohadi, ST
NIP.Y.1039700309

Form S-4b

LISTING PROGRAM

➢ Avi.vb

```
Imports System
Imports System.Drawing
Imports System.Runtime.InteropServices

Namespace AviFile

PublicClass Avi
PublicShared PALETTE_SIZE AsInteger = 4 * 256
'RGBQUAD * 256 colours
PublicConst StreamtypeVIDEO AsInteger = 1935960438
'mmioStringToFOURCC("vids", 0)
PublicConst OF_SHARE_DENY_WRITE AsInteger = 32
PublicConst OF_WRITE AsInteger = 1
PublicConst OF_READWRITE AsInteger = 2
PublicConst OF_CREATE AsInteger = 4096
PublicConst BMP_MAGIC_COOKIE AsInteger = 19778
'ascii string "BM"
#Region"structure declarations"

<StructLayout(LayoutKind.Sequential, Pack:=1)> _
PublicStructure RECT
Public left As UInt32
Public top As UInt32
Public right As UInt32
Public bottom As UInt32
EndStructure

<StructLayout(LayoutKind.Sequential, Pack:=1)> _
PublicStructure BITMAPINFOHEADER
Public biSize As UInt32
Public biWidth As Int32
Public biHeight As Int32
Public biPlanes As Int16
Public biBitCount As Int16
Public biCompression As UInt32
Public biSizeImage As UInt32
Public biXPelsPerMeter As Int32
Public biYPelsPerMeter As Int32
Public biClrUsed As UInt32
Public biClrImportant As UInt32
EndStructure

<StructLayout(LayoutKind.Sequential, Pack:=1)> _
PublicStructure STREAMINFO
Public fccType As UInt32
Public fccHandler As UInt32
Public dwFlags As UInt32
Public dwCaps As UInt32
Public wPriority As UInt16
Public wLanguage As UInt16
Public dwScale As UInt32
Public dwRate As UInt32
Public dwStart As UInt32
Public dwLength As UInt32
Public dwInitialFrames As UInt32
Public dwSuggestedBufferSize As UInt32
Public dwQuality As UInt32
Public dwSampleSize As UInt32
Public rcFrame As RECT
Public dwEditCount As UInt32
Public dwFormatChangeCount As UInt32
<MarshalAs(UnmanagedType.ByValArray, SizeConst:=64)> _
Public szName As UInt16()
EndStructure
<StructLayout(LayoutKind.Sequential, Pack:=1)> _
PublicStructure BITMAPFILEHEADER
Public bfType As Int16
'"magic cookie" - must be "BM"
Public bfSize As Int32
Public bfReserved1 As Int16
Public bfReserved2 As Int16
Public bfOffBits As Int32
EndStructure

#EndRegion

#Region"method declarations"

'Initialize the AVI library
<DllImport("avifil32.dll")> _
PublicSharedSub AVIFileInit()
EndSub

'Open an AVI file
<DllImport("avifil32.dll", PreserveSig:=True)> _
```

```
PublicSharedFunction AVIFileOpen(ByRef ppfile AsInteger, ByVal szFile AsString, ByVal uMode AsInteger,
ByVal pclsidHandler AsInteger) AsInteger
EndFunction

'Get a stream from an open AVI file
<DllImport("avifil32.dll")> _
PublicSharedFunction AVIFileGetStream(ByVal pfile AsInteger, ByRef ppavi As IntPtr, ByVal fccType
AsInteger, ByVal lParam AsInteger) AsInteger
EndFunction

'Get the start position of a stream
<DllImport("avifil32.dll", PreserveSig:=True)> _
PublicSharedFunction AVIStreamStart(ByVal pavi AsInteger) AsInteger
EndFunction

'Get the length of a stream in frames
<DllImport("avifil32.dll", PreserveSig:=True)> _
PublicSharedFunction AVIStreamLength(ByVal pavi AsInteger) AsInteger
EndFunction

'Get information about an open stream
<DllImport("avifil32.dll")> _
PublicSharedFunction AVIStreamInfo(ByVal pAVIStream AsInteger, ByRef psi As STREAMINFO, ByVal lSize
AsInteger) AsInteger
EndFunction

'Get a pointer to a GETFRAME object (returns 0 on error)
<DllImport("avifil32.dll")> _
PublicSharedFunction AVIStreamGetFrameOpen(ByVal pAVIStream As IntPtr, ByRef bih As BITMAPINFOHEADER)
AsInteger
EndFunction

'Get a pointer to a packed DIB (returns 0 on error)
<DllImport("avifil32.dll")> _
PublicSharedFunction AVIStreamGetFrame(ByVal pGetFrameObj AsInteger, ByVal lPos AsInteger) AsInteger
EndFunction

'Create a new stream in an open AVI file
<DllImport("avifil32.dll")> _
PublicSharedFunction AVIFileCreateStream(ByVal pfile AsInteger, ByRef ppavi As IntPtr, ByRef
ptr_streaminfo As STREAMINFO) AsInteger
EndFunction

'Set the format for a new stream
<DllImport("avifil32.dll")> _
PublicSharedFunction AVIStreamSetFormat(ByVal aviStream As IntPtr, ByVal lPos As Int32, ByRef lpFormat
As BITMAPINFOHEADER, ByVal cbFormat As Int32) AsInteger
EndFunction

'Write a sample to a stream
<DllImport("avifil32.dll")> _
PublicSharedFunction AVIStreamWrite(ByVal aviStream As IntPtr, ByVal lStart As Int32, ByVal lSamples As
Int32, ByVal lpBuffer As IntPtr, ByVal cbBuffer As Int32, ByVal dwFlags As Int32, _
ByVal dummy1 As Int32, ByVal dummy2 As Int32) AsInteger
EndFunction

'Release the GETFRAME object
<DllImport("avifil32.dll")> _
PublicSharedFunction AVIStreamGetFrameClose(ByVal pGetFrameObj AsInteger) AsInteger
EndFunction

'Release an open AVI stream
<DllImport("avifil32.dll")> _
PublicSharedFunction AVIStreamRelease(ByVal aviStream As IntPtr) AsInteger
EndFunction

'Release an open AVI file
<DllImport("avifil32.dll")> _
PublicSharedFunction AVIFileRelease(ByVal pfile AsInteger) AsInteger
EndFunction

'Close the AVI library
<DllImport("avifil32.dll")> _
PublicSharedSub AVIFileExit()
EndSub

#EndRegion


EndClass

EndNamespace
```

➢ **AviReader.vb**

```
Imports System
Imports System.Drawing
Imports System.Drawing.Imaging
Imports System.Runtime.InteropServices
Imports System.IO

Namespace AviFile

'Extract bitmaps from AVI files
PublicClass AviReader

'position of the first frame, count of frames in the stream
Private firstFrame AsInteger = 0, cntFrames AsInteger = 0
'pointers
Private aviFile AsInteger = 0
Private getFrameObject AsInteger
Private aviStream As IntPtr
'stream and header info
Private streamInfo As Avi.STREAMINFO

PublicReadOnlyProperty CountFrames() AsInteger
Get
Return cntFrames
EndGet
EndProperty

PublicReadOnlyProperty FrameRate() As UInt32
Get
Return streamInfo.dwRate / streamInfo.dwScale
EndGet
EndProperty

PublicReadOnlyProperty BitmapSize() As Size
Get
ReturnNew Size(CInt(streamInfo.rcFrame.right), CInt(streamInfo.rcFrame.bottom))
EndGet
EndProperty

'Opens an AVI file and creates a GetFrame object
PublicSub Open(ByVal fileName AsString)
'Intitialize AVI library
            Avi.AVIFileInit()

'Open the file
Dim result AsInteger = Avi.AVIFileOpen(aviFile, fileName, Avi.OF_SHARE_DENY_WRITE, 0)

If result <> 0 Then
ThrowNew Exception("Exception in AVIFileOpen: "& result.ToString())
EndIf

'Get the video stream
            result = Avi.AVIFileGetStream(aviFile, aviStream, Avi.StreamtypeVIDEO, 0)

If result <> 0 Then
ThrowNew Exception("Exception in AVIFileGetStream: "& result.ToString())
EndIf

            firstFrame = Avi.AVIStreamStart(aviStream.ToInt32())
            cntFrames = Avi.AVIStreamLength(aviStream.ToInt32())

            streamInfo = New Avi.STREAMINFO()
            result = Avi.AVIStreamInfo(aviStream.ToInt32(), streamInfo, Marshal.SizeOf(streamInfo))

If result <> 0 Then
ThrowNew Exception("Exception in AVIStreamInfo: "& result.ToString())
EndIf

'Open frames

Dim bih AsNew Avi.BITMAPINFOHEADER()
            bih.biBitCount = 24
            bih.biClrImportant = 0
            bih.biClrUsed = 0
            bih.biCompression = 0
'BI_RGB;
            bih.biHeight = CType(streamInfo.rcFrame.bottom, Int32)
            bih.biWidth = CType(streamInfo.rcFrame.right, Int32)
            bih.biPlanes = 1
            bih.biSize = CType(Marshal.SizeOf(bih), UInt32)
            bih.biXPelsPerMeter = 0
            bih.biYPelsPerMeter = 0

            getFrameObject = Avi.AVIStreamGetFrameOpen(aviStream, bih)
'force function to return 24bit DIBS
'getFrameObject = Avi.AVIStreamGetFrameOpen(aviStream, 0); //return any bitmaps
If getFrameObject = 0 Then
```

```
ThrowNew Exception("Exception in AVIStreamGetFrameOpen!")
EndIf
EndSub

'Closes all streams, files and libraries
PublicSub Close()
If getFrameObject <> 0 Then
                Avi.AVIStreamGetFrameClose(getFrameObject)
                getFrameObject = 0
EndIf
If aviStream <> IntPtr.Zero Then
                Avi.AVIStreamRelease(aviStream)
                aviStream = IntPtr.Zero
EndIf
If aviFile <> 0 Then
                Avi.AVIFileRelease(aviFile)
                aviFile = 0
EndIf
            Avi.AVIFileExit()
EndSub

'Exports a frame into a bitmap file</summary>
PublicSub ExportBitmap(ByVal position AsInteger, ByVal dstFileName AsString)
Dim bmp As Bitmap = GetBitmap(position)
            bmp.Save(dstFileName, ImageFormat.Bmp)
            bmp.Dispose()
EndSub

PublicFunction GetBitmap(ByVal position AsInteger) As Bitmap
If position > countFrames Then
ThrowNew Exception("Invalid frame position: "& position)
EndIf

'Decompress the frame and return a pointer to the DIB
Dim dib AsInteger = Avi.AVIStreamGetFrame(getFrameObject, firstFrame + position)
'Copy the bitmap header into a managed struct
Dim bih AsNew Avi.BITMAPINFOHEADER()
            bih = DirectCast(Marshal.PtrToStructure(New IntPtr(dib), bih.[GetType]()),
Avi.BITMAPINFOHEADER)

If bih.biSizeImage < 1 Then
ThrowNew Exception("Exception in VideoStreamGetFrame")
EndIf

'copy the image

Dim bitmapData AsByte()
Dim address AsInteger = dib + Marshal.SizeOf(bih)
If bih.biBitCount < 16 Then
'copy palette and pixels
                bitmapData = NewByte(bih.biSizeImage + (Avi.PALETTE_SIZE - 1)) {}
Else
'copy only pixels
                bitmapData = NewByte(bih.biSizeImage - 1) {}
EndIf

            Marshal.Copy(New IntPtr(address), bitmapData, 0, bitmapData.Length)

'copy bitmap info
Dim bitmapInfo AsByte() = NewByte(Marshal.SizeOf(bih) - 1) {}
Dim ptr As IntPtr
            ptr = Marshal.AllocHGlobal(bitmapInfo.Length)
            Marshal.StructureToPtr(bih, ptr, False)
            address = ptr.ToInt32()
            Marshal.Copy(New IntPtr(address), bitmapInfo, 0, bitmapInfo.Length)

            Marshal.FreeHGlobal(ptr)

'create file header
Dim bfh AsNew Avi.BITMAPFILEHEADER()
            bfh.bfType = Avi.BMP_MAGIC_COOKIE
            bfh.bfSize = DirectCast(55 + bih.biSizeImage, Long)
'size of file as written to disk
            bfh.bfReserved1 = 0
            bfh.bfReserved2 = 0
            bfh.bfOffBits = Marshal.SizeOf(bih) + Marshal.SizeOf(bfh)
If bih.biBitCount < 16 Then
'There is a palette between header and pixel data
                bfh.bfOffBits += Avi.PALETTE_SIZE
EndIf

'write a bitmap stream
Dim bw AsNew BinaryWriter(New MemoryStream())

'write header
            bw.Write(bfh.bfType)
            bw.Write(bfh.bfSize)
            bw.Write(bfh.bfReserved1)
            bw.Write(bfh.bfReserved2)
```

```
            bw.Write(bfh.bfOffBits)
'write bitmap info
            bw.Write(bitmapInfo)
'write bitmap data
            bw.Write(bitmapData)

Dim bmp As Bitmap = DirectCast(Image.FromStream(bw.BaseStream), Bitmap)
Dim saveableBitmap AsNew Bitmap(bmp.Width, bmp.Height)
Dim g As Graphics = Graphics.FromImage(saveableBitmap)
            g.DrawImage(bmp, 0, 0)
            g.Dispose()
            bmp.Dispose()

            bw.Close()
Return saveableBitmap
EndFunction
EndClass

EndNamespace
```

➢ AviWriter.vb

```
Imports System
Imports System.IO
Imports System.Drawing
Imports System.Drawing.Imaging
Imports System.Runtime.InteropServices

Namespace AviFile

'Create AVI files from bitmaps
PublicClass AviWriter

Private aviFile AsInteger = 0
Private aviStream As IntPtr = IntPtr.Zero
Private frameRate As UInt32 = 0
Private countFrames AsInteger = 0
Private width AsInteger = 0
Private height AsInteger = 0
Private stride As UInt32 = 0
Private fccType As UInt32 = Avi.StreamtypeVIDEO ' vids
Private fccHandler As UInt32 = 1668707181 '"Microsoft Video 1" - Use CVID for default codec:
(UInt32)Avi.mmioStringToFOURCC("CVID", 0);

'Creates a new AVI file
PublicSub Open(ByVal fileName AsString, ByVal frameRate As UInt32)
Me.frameRate = frameRate

            Avi.AVIFileInit()

' OF_WRITE | OF_CREATE (winbase.h)
Dim hr AsInteger = Avi.AVIFileOpen(aviFile, fileName, Avi.OF_WRITE Or Avi.OF_CREATE, 0)
If hr <> 0 Then
ThrowNew Exception("Error in AVIFileOpen: "& hr.ToString())
EndIf
EndSub

'Adds a new frame to the AVI stream
PublicSub AddFrame(ByVal bmp As Bitmap)

            bmp.RotateFlip(RotateFlipType.RotateNoneFlipY)

Dim bmpDat As BitmapData = bmp.LockBits(New Rectangle(0, 0, bmp.Width, bmp.Height),
ImageLockMode.[ReadOnly], PixelFormat.Format24bppRgb)

If countFrames = 0 Then
'this is the first frame - get size and create a new stream
Me.stride = CType(bmpDat.Stride, UInt32)
Me.width = bmp.Width
Me.height = bmp.Height
                CreateStream()
EndIf

'pointer to the beginning of the image data
Dim result AsInteger = Avi.AVIStreamWrite(aviStream, countFrames, 1, bmpDat.Scan0, CType(stride *
height, Int32), 0, _
             0, 0)

If result <> 0 Then
ThrowNew Exception("Error in AVIStreamWrite: "& result.ToString())
EndIf

            bmp.UnlockBits(bmpDat)
            countFrames += 1
EndSub

'Closes stream, file and AVI library
```

```
PublicSub Close()
If aviStream <> IntPtr.Zero Then
                Avi.AVIStreamRelease(aviStream)
                aviStream = IntPtr.Zero
EndIf
If aviFile <> 0 Then
                Avi.AVIFileRelease(aviFile)
                aviFile = 0
EndIf
            Avi.AVIFileExit()
EndSub

'Creates a new video stream in the AVI file
PrivateSub CreateStream()
Dim strhdr AsNew Avi.STREAMINFO()
            strhdr.fccType = fccType
            strhdr.fccHandler = fccHandler
            strhdr.dwScale = 1
            strhdr.dwRate = frameRate
            strhdr.dwSuggestedBufferSize = CType(height * stride, UInt32)
            strhdr.dwQuality = 10000
'highest quality! Compression destroys the hidden message
            strhdr.rcFrame.bottom = CType(height, UInt32)
            strhdr.rcFrame.right = CType(width, UInt32)
            strhdr.szName = New UInt16(63) {}

Dim result AsInteger = Avi.AVIFileCreateStream(aviFile, aviStream, strhdr)
If result <> 0 Then
ThrowNew Exception("Error in AVIFileCreateStream: "& result.ToString())
EndIf

'define the image format

Dim bi AsNew Avi.BITMAPINFOHEADER()
            bi.biSize = CType(Marshal.SizeOf(bi), UInt32)
            bi.biWidth = CType(width, Int32)
            bi.biHeight = CType(height, Int32)
            bi.biPlanes = 1
            bi.biBitCount = 24
            bi.biSizeImage = CType(stride * height, UInt32)

            result = Avi.AVIStreamSetFormat(aviStream, 0, bi, Marshal.SizeOf(bi))
If result <> 0 Then
ThrowNew Exception("Error in AVIStreamSetFormat: "& result.ToString())
EndIf
EndSub

PublicSub saveImage(ByVal imageHide AsString, ByVal newFile AsString)
Dim aviReader AsNew AviReader
            aviReader.Open(imageHide)

Dim pathBitmap As [String] = "..\..\tempBitmap\"

Dim aviWriter AsNew AviWriter()
            aviWriter.Open(newFile, aviReader.FrameRate)

For n AsInteger = 0 To aviReader.CountFrames - 1
Dim image As Image = image.FromFile(pathBitmap & n &".bmp")
Dim bitmap As Bitmap = New Bitmap(image)
                aviWriter.AddFrame(bitmap)
                image.Dispose()
If File.Exists(pathBitmap & n &".bmp") Then
                    File.Delete(pathBitmap & n &".bmp")
EndIf
Next
            aviWriter.Close()
            aviReader.Close()

EndSub
EndClass

EndNamespace
```

➢ Converter.vb

```
Namespace AviFile
PublicClass Converter

PrivateShared m_IndexArr() AsInteger = {128, 64, 32, 16, 8, 4, 2, 1}

PublicSharedFunction StringToBinary(ByVal theStr AsString) AsString
Dim StrToBin AsString = ""
ForEach c AsCharIn theStr.ToCharArray
                StrToBin &= IntToBin(Asc(c))
Next c
Return StrToBin
EndFunction
```

```
PublicSharedFunction BinaryToString(ByVal theBin AsString) AsString
Dim BinToStr AsString = ""
For i AsInteger = 0 To theBin.Length - 1 Step 8
                BinToStr &= Chr(BinToInt8(theBin.Substring(i, 8).ToCharArray))
Next i
Return BinToStr
EndFunction

PrivateSharedFunction IntToBin(ByVal Number AsInteger) AsString
Dim IntToBinary AsString = ""
Dim Temp AsInteger = 1

DoUntil Temp > Number
                Temp <<= 1
Loop

While Temp > 0
If Number < Temp Then : IntToBinary &= "0"
Else
                    IntToBinary &= "1"
                    Number -= Temp
EndIf

                Temp >>= 1
EndWhile

            IntToBin = IntToBinary.PadLeft(8, "0")
EndFunction

PrivateSharedFunction BinToInt8(ByVal chars() AsChar) AsInteger
For i AsInteger = 0 To 7
If chars(i) = "1"Then BinToInt8 += m_IndexArr(i)
Next i
EndFunction
EndClass

EndNamespace
```

➢ Form1.vb

```
Imports System
Imports System.Drawing
Imports System.Collections
Imports System.ComponentModel
Imports System.Windows.Forms
Imports System.Data
Imports System.IO
Imports System.Text
Imports System.Runtime.InteropServices
Imports SteganogrAVI.AviFile
Imports VB = Microsoft.VisualBasic

PublicClass Form1

Dim Password1 AsString, Password2 AsString, imageHidePath AsString, imageExtractPath AsString,
loadedFilePath AsString, messageExt AsString, messageText AsString, saveToImage AsString, SaveFilePath
AsString
Dim messageFileSize AsLong, lenghtMessageText AsLong, lenghtPassword2 AsString, lenghtExt AsLong
Dim loadedImageHide0 As Image, loadedImageHide As Image, loadedImageExtract0 As Image,
loadedImageExtract As Image
Dim loadedBitmapHide0 As Bitmap, loadedBitmapHide As Bitmap, loadedBitmapExtract0 As Bitmap,
loadedBitmapExtract As Bitmap
Dim fileContainer AsByte()
Dim pathBitmap As [String] = "..\..\tempBitmap\"
Dim start AsDate, finish AsDate, time AsInteger, start1 AsDate, finish1 AsDate, time1 AsInteger

PrivateSub btnImageFileHide_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnImageFileHide.Click
If openFileDialog1.ShowDialog = DialogResult.OK Then
            imageHidePath = openFileDialog1.FileName
            txtImageHide.Text = imageHidePath
EndIf
EndSub

PrivateSub btnMessage_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnMessage.Click
If openFileDialog2.ShowDialog = DialogResult.OK Then
            loadedFilePath = openFileDialog2.FileName
            txtMessageFile.Text = loadedFilePath
            messageExt = justExt(loadedFilePath)
            lenghtExt = messageExt.Length
Dim finfo AsNew FileInfo(loadedFilePath)
            messageFileSize = finfo.Length
            fileContainer = File.ReadAllBytes(loadedFilePath)
EndIf
EndSub
```

```
PrivateFunction justExt(ByVal path AsString) AsString
Dim index AsInteger = path.LastIndexOf(".")
If index > 0 Then
            path = path.Substring(index)
EndIf
Return path
EndFunction

PrivateSub btnImageFileExtract_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnImageFileExtract.Click
If openFileDialog3.ShowDialog = DialogResult.OK Then
            imageExtractPath = openFileDialog3.FileName
            txtImageExtract.Text = imageExtractPath
EndIf
EndSub

#Region" Hide Message "
PrivateSub HideMessage(ByVal aviReader As AviReader)
        exportFrame(imageHidePath)
Dim countFrame AsInteger = aviReader.CountFrames

Dim pathBitmap0 AsString = pathBitmap &"0.bmp"
        loadedImageHide0 = Image.FromFile(pathBitmap0)
        loadedBitmapHide0 = New Bitmap(loadedImageHide0)
        Password2 = txtPass2.Text
        lenghtPassword2 = Password2.Length
Dim h AsInteger = aviReader.BitmapSize.Height
Dim w AsInteger = aviReader.BitmapSize.Width
Dim n AsInteger

If rdoMessageText.Checked Then

            messageText = txtMessageText.Text
            lenghtMessageText = messageText.Length

Dim changedBitmap0 As Bitmap = HideMsg(loadedBitmapHide0, Password2, lenghtPassword2, aviReader, True)
            setLenghtSize(changedBitmap0, lenghtMessageText * 8, lenghtPassword2 * 8, Nothing, 0)
            loadedImageHide0.Dispose()
If File.Exists(pathBitmap0) Then
                File.Delete(pathBitmap0)
EndIf
            changedBitmap0.Save(pathBitmap0)


If lenghtMessageText > 0 Then

                n = 1
While n < countFrame
                    loadedImageHide = Image.FromFile(pathBitmap & n &".bmp")
                    loadedBitmapHide = New Bitmap(loadedImageHide)
Dim changedBitmap As Bitmap = HideMsg(loadedBitmapHide, messageText, lenghtMessageText, aviReader,
True)

                    loadedImageHide.Dispose()
If File.Exists(pathBitmap & n &".bmp") Then
                        File.Delete(pathBitmap & n &".bmp")
EndIf
                    changedBitmap.Save(pathBitmap & n &".bmp")
If lenghtMessageText < w * h Then
ExitWhile
EndIf
                    n += 1
EndWhile
EndIf

Else

Dim changedBitmap0 As Bitmap = HideMsg(loadedBitmapHide0, Password2, lenghtPassword2, aviReader, True)
            setLenghtSize(changedBitmap0, messageFileSize, lenghtPassword2 * 8, messageExt, 1)
            loadedImageHide0.Dispose()
If File.Exists(pathBitmap0) Then
                File.Delete(pathBitmap0)
EndIf
            changedBitmap0.Save(pathBitmap0)

If messageFileSize > 0 Then

                n = 1
While n < countFrame
                    loadedImageHide = Image.FromFile(pathBitmap & n &".bmp")
                    loadedBitmapHide = New Bitmap(loadedImageHide)

Dim changedBitmap As Bitmap = HideMsg(loadedBitmapHide, Nothing, messageFileSize, aviReader, False)
                    loadedImageHide.Dispose()
If File.Exists(pathBitmap & n &".bmp") Then
                        File.Delete(pathBitmap & n &".bmp")
EndIf
                    changedBitmap.Save(pathBitmap & n &".bmp")
```

```
If messageFileSize < w * h Then
ExitWhile
EndIf
                    n += 1
EndWhile
EndIf

EndIf
EndSub

PrivateFunction HideMsg(ByVal inputBitmap As Bitmap, ByVal messageText AsString, ByVal messageLenght
AsLong, ByVal aviReader As AviReader, ByVal text AsBoolean)
Dim h AsInteger = aviReader.BitmapSize.Height
Dim w AsInteger = aviReader.BitmapSize.Width
Dim outputBitmap As Bitmap = inputBitmap
Dim convertedString AsString
Dim i AsInteger = 0, j AsInteger = 0, n AsInteger = 0
Dim red AsString() = NewString(7) {}
Dim t AsBoolean() = NewBoolean(7) {}
Dim rb AsBoolean() = NewBoolean(7) {}
Dim gb AsBoolean() = NewBoolean(7) {}
Dim bb AsBoolean() = NewBoolean(7) {}
Dim pixel AsNew Color()
Dim r AsByte, g AsByte, b AsByte

If text Then
          convertedString = Converter.StringToBinary(messageText)
Dim lenghtConvertedString AsString = convertedString.Length
          i = 0
          n = 0
While i < h AndAlso i * h < lenghtConvertedString AndAlso n < lenghtConvertedString
                j = 0
While j < w AndAlso i * h + j < lenghtConvertedString AndAlso n < lenghtConvertedString
                pixel = inputBitmap.GetPixel(j, i)
                r = pixel.R
                red = getBinary(r)

                red(7) = convertedString.Chars(n)

Dim result As Color = Color.FromArgb(getByte(red), pixel.G, pixel.B)
                outputBitmap.SetPixel(j, i, result)
                j += 1
                n += 1
EndWhile
              i += 1
              n = w * i
EndWhile

Else

          i = 0
While i < h AndAlso i * (h \ 3) < messageLenght
                j = 0
While j < (w \ 3) * 3 AndAlso i * (h \ 3) + (j \ 3) < messageLenght

                byte2bool(CByte(fileContainer(i * (h \ 3) + j \ 3)), t)

                pixel = inputBitmap.GetPixel(j, i)
                r = pixel.R
                g = pixel.G
                b = pixel.B
                byte2bool(r, rb)
                byte2bool(g, gb)
                byte2bool(b, bb)
If j Mod 3 = 0 Then
                    rb(7) = t(0)
                    gb(7) = t(1)
                    bb(7) = t(2)
ElseIf j Mod 3 = 1 Then
                    rb(7) = t(3)
                    gb(7) = t(4)
                    bb(7) = t(5)
Else
                    rb(7) = t(6)
                    gb(7) = t(7)
EndIf
Dim result As Color = Color.FromArgb(CInt(bool2byte(rb)), CInt(bool2byte(gb)), CInt(bool2byte(bb)))
                outputBitmap.SetPixel(j, i, result)
                j += 1
EndWhile
              i += 1
EndWhile

EndIf

Return outputBitmap
EndFunction
#EndRegion
```

```
#Region" Extract Message "
PrivateSub ExtractMessage(ByVal aviReader As AviReader)
        start = Now
        Timer1.Enabled = True
        btnExtract.Enabled = False
        ToolStripStatusLabel2.Text = "Extract the Message..."
        Application.DoEvents()
        exportFrame(imageExtractPath)
Dim countFrame AsInteger = aviReader.CountFrames
Dim h AsInteger = aviReader.BitmapSize.Height
Dim w AsInteger = aviReader.BitmapSize.Width
Dim i AsInteger, j AsInteger, n AsInteger, f AsInteger
Dim pixel AsNew Color()
Dim pathBitmap0 AsString = pathBitmap &"0.bmp"
        loadedImageExtract0 = Image.FromFile(pathBitmap0)
        loadedBitmapExtract0 = New Bitmap(loadedImageExtract0)
        Password2 = txtPass2.Text

        pixel = loadedBitmapExtract0.GetPixel(w - 1, h - 1)
Dim fSize AsLong = pixel.R + pixel.G * 100 + pixel.B * 10000
        pixel = loadedBitmapExtract0.GetPixel(w - 2, h - 1)
Dim tagSize AsLong = pixel.R + pixel.G * 100 + pixel.B * 10000
        pixel = loadedBitmapExtract0.GetPixel(w - 3, h - 1)
Dim getExt AsInteger = pixel.R
        pixel = loadedBitmapExtract0.GetPixel(w - 4, h - 1)
Dim tag AsInteger = pixel.R

Dim pass AsString = ""
Dim resPass AsString
Dim msgtext AsString = ""
Dim resText AsString
Dim red AsString
Dim t AsBoolean() = NewBoolean(7) {}
Dim rb AsBoolean() = NewBoolean(7) {}
Dim gb AsBoolean() = NewBoolean(7) {}
Dim bb AsBoolean() = NewBoolean(7) {}
Dim res AsByte() = NewByte((fSize) - 1) {}
Dim r AsByte, g AsByte, b AsByte
Dim temp AsByte

        i = 0
While i < h AndAlso i * h < tagSize AndAlso n < tagSize
            j = 0
            n = 0
While j < w AndAlso i * h + j < tagSize AndAlso n < tagSize
                pixel = loadedBitmapExtract0.GetPixel(j, i)
                r = pixel.R
                red = ByteToBool(r)

                pass &= red(7)

                j += 1
                n += 1
EndWhile
            i += 1
            n = w * i
EndWhile

        resPass = Converter.BinaryToString(pass)

        loadedImageExtract0.Dispose()
If File.Exists(pathBitmap0) Then
            File.Delete(pathBitmap0)
EndIf

If tag = 0 Then
If resPass = Password2 Then

                n = 1
While n < countFrame
                    loadedImageExtract = Image.FromFile(pathBitmap & n &".bmp")
                    loadedBitmapExtract = New Bitmap(loadedImageExtract)

                    i = 0
While i < h AndAlso i * h < fSize AndAlso f < fSize
                        j = 0
                        f = 0
While j < w AndAlso i * h + j < fSize AndAlso f < fSize
                            pixel = loadedBitmapExtract.GetPixel(j, i)
                            r = pixel.R
                            red = ByteToBool(r)

                            msgtext &= red(7)

                            j += 1
                            f += 1
EndWhile
                        i += 1
                        f = w * i
```

```
EndWhile
                    loadedImageExtract.Dispose()
If fSize < w * h Then
ExitWhile
EndIf
                    n += 1
EndWhile
                resText = Converter.BinaryToString(msgtext)
                txtExtractedMsgText.Text = resText
                Timer1.Enabled = False
                finish = Now
                time = DateDiff(DateInterval.Second, start, finish)
Dim countProcess AsString = countTime(time)
                ToolStripStatusLabel2.Text = "The Hidden Message was Successfully Extracted"
                MessageBox.Show("The Message was Successfully Extracted in "& countProcess &" seconds",
"SteganograVI", MessageBoxButtons.OK, MessageBoxIcon.Information)
                btnExtract.Enabled = True
                Application.DoEvents()
Else
                ToolStripStatusLabel2.Text = "Not Found Secret Message"
                MessageBox.Show("Not Found Secret Message In The Video", "SteganograVI",
MessageBoxButtons.OK, MessageBoxIcon.Information)
                btnExtract.Enabled = True
Return
EndIf

ElseIf tag = 1 Then
If resPass = Password2 Then
                Timer1.Enabled = False
                finish = Now
If saveFileDialog2.ShowDialog() = DialogResult.OK Then
                    SaveFilePath = saveFileDialog2.FileName
                    Timer1.Enabled = True
                    start1 = Now
Else
                    btnExtract.Enabled = True
Return
EndIf

                n = 1
While n < countFrame
                    loadedImageExtract = Image.FromFile(pathBitmap & n &".bmp")
                    loadedBitmapExtract = New Bitmap(loadedImageExtract)

                    i = 0
While i < h AndAlso i * (h \ 3) < fSize
                        j = 0
While j < (w \ 3) * 3 AndAlso i * (h \ 3) + (j \ 3) < fSize
                            pixel = loadedBitmapExtract.GetPixel(j, i)
                            r = pixel.R
                            g = pixel.G
                            b = pixel.B
                            byte2bool(r, rb)
                            byte2bool(g, gb)
                            byte2bool(b, bb)
If j Mod 3 = 0 Then
                                t(0) = rb(7)
                                t(1) = gb(7)
                                t(2) = bb(7)
ElseIf j Mod 3 = 1 Then
                                t(3) = rb(7)
                                t(4) = gb(7)
                                t(5) = bb(7)
Else
                                t(6) = rb(7)
                                t(7) = gb(7)
                                temp = bool2byte(t)
                                res(i * (h \ 3) + j \ 3) = temp
EndIf
                            j += 1
EndWhile
                        i += 1
EndWhile

                    loadedImageExtract.Dispose()
If fSize < w * h - 2 Then
ExitWhile
EndIf
                    n += 1
EndWhile

Dim ext AsString = ""
If getExt = 1 Then
                    ext = ".docx"
ElseIf getExt = 2 Then
                    ext = ".xlsx"
ElseIf getExt = 3 Then
                    ext = ".jpg"
ElseIf getExt = 4 Then
```

```
                    ext = ".txt"
EndIf

                File.WriteAllBytes(SaveFilePath & ext, res)
                Timer1.Enabled = False
                finish1 = Now
                time = DateDiff(DateInterval.Second, start, finish)
                time1 = DateDiff(DateInterval.Second, start1, finish1)
Dim totaltime AsInteger = time + time1
Dim countProcess AsString = countTime(totaltime)
                ToolStripStatusLabel2.Text = "The Hidden Message was Successfully Extracted"
                MessageBox.Show("The Message was Successfully Extracted in "& countProcess &" seconds",
"SteganograVI", MessageBoxButtons.OK, MessageBoxIcon.Information)
                btnExtract.Enabled = True
                Application.DoEvents()
Else
                ToolStripStatusLabel2.Text = "Not Found Secret Message"
                MessageBox.Show("Not Found Secret Message In The Video", "SteganograVI",
MessageBoxButtons.OK, MessageBoxIcon.Information)
                btnExtract.Enabled = True
Return
EndIf
EndIf
EndSub
#EndRegion

PrivateSub setLenghtSize(ByVal inputBitmap As Bitmap, ByVal fileSize AsLong, ByVal lengthPass AsLong,
ByVal ext AsString, ByVal tag AsInteger)
Dim outputBitmap As Bitmap = inputBitmap
Dim pixel AsNew Color()
Dim r AsByte, g AsByte, b AsByte
Dim w AsInteger = inputBitmap.Width
Dim h AsInteger = inputBitmap.Height
Dim tempFS AsLong = fileSize, tempLP AsLong = lengthPass
        r = CByte(tempFS Mod 100)
        tempFS = Math.Floor(tempFS / 100)
        g = CByte(tempFS Mod 100)
        tempFS = Math.Floor(tempFS / 100)
        b = CByte(tempFS Mod 100)
Dim fsLenColor As Color = Color.FromArgb(r, g, b)
        outputBitmap.SetPixel(w - 1, h - 1, fsLenColor)

        r = CByte(tempLP Mod 100)
        tempLP = Math.Floor(tempLP / 100)
        g = CByte(tempLP Mod 100)
        tempLP = Math.Floor(tempLP / 100)
        b = CByte(tempLP Mod 100)
Dim pLenColor As Color = Color.FromArgb(r, g, b)
        outputBitmap.SetPixel(w - 2, h - 1, pLenColor)

If ext = ".docx"Then
            r = 1
Dim extColor As Color = Color.FromArgb(r, pixel.G, pixel.B)
            outputBitmap.SetPixel(w - 3, h - 1, extColor)
ElseIf ext = ".xlsx"Then
            r = 2
Dim extColor As Color = Color.FromArgb(r, pixel.G, pixel.B)
            outputBitmap.SetPixel(w - 3, h - 1, extColor)
ElseIf ext = ".jpg"Then
            r = 3
Dim extColor As Color = Color.FromArgb(r, pixel.G, pixel.B)
            outputBitmap.SetPixel(w - 3, h - 1, extColor)
ElseIf ext = ".txt"Then
            r = 4
Dim extColor As Color = Color.FromArgb(r, pixel.G, pixel.B)
            outputBitmap.SetPixel(w - 3, h - 1, extColor)
EndIf

        r = CByte(tag)
Dim tagColor As Color = Color.FromArgb(r, pixel.G, pixel.B)
        outputBitmap.SetPixel(w - 4, h - 1, tagColor)
EndSub

PrivateFunction countFile(ByVal dir AsString) AsInteger
Dim cnt AsInteger
ForEach fi AsStringIn System.IO.Directory.GetFiles(dir)
            cnt += 1
Next
Return cnt
EndFunction

#Region" Conversion byte - binary "
PublicFunction getBinary(ByVal B AsByte) AsString()
Dim getB AsString() = NewString(7) {}
If B >= 128 Then
            B = B - 128
            getB(0) = "1"
Else
            getB(0) = "0"
```

```
EndIf
If B >= 64 Then
            B = B - 64
            getB(1) = getB(0) &"1"
Else
            getB(1) = getB(0) &"0"
EndIf
If B >= 32 Then
            B = B - 32
            getB(2) = getB(1) &"1"
Else
            getB(2) = getB(1) &"0"
EndIf
If B >= 16 Then
            B = B - 16
            getB(3) = getB(2) &"1"
Else
            getB(3) = getB(2) &"0"
EndIf
If B >= 8 Then
            B = B - 8
            getB(4) = getB(3) &"1"
Else
            getB(4) = getB(3) &"0"
EndIf
If B >= 4 Then
            B = B - 4
            getB(5) = getB(4) &"1"
Else
            getB(5) = getB(4) &"0"
EndIf
If B >= 2 Then
            B = B - 2
            getB(6) = getB(5) &"1"
Else
            getB(6) = getB(5) &"0"
EndIf
If B >= 1 Then
            B = B - 1
            getB(7) = getB(6) &"1"
Else
            getB(7) = getB(6) &"0"
EndIf
Return getB
EndFunction

PublicFunction getByte(ByVal B AsString())
Dim getB AsByte
Dim Bin AsString

If B(0) = "1"Then
            getB = 128
EndIf
        Bin = B(1)
If VB.Right(Bin, 1) = "1"Then
            getB = getB + 64
EndIf
        Bin = B(2)
If VB.Right(Bin, 1) = "1"Then
            getB = getB + 32
EndIf
        Bin = B(3)
If VB.Right(Bin, 1) = "1"Then
            getB = getB + 16
EndIf
        Bin = B(4)
If VB.Right(Bin, 1) = "1"Then
            getB = getB + 8
EndIf
        Bin = B(5)
If VB.Right(Bin, 1) = "1"Then
            getB = getB + 4
EndIf
        Bin = B(6)
If VB.Right(Bin, 1) = "1"Then
            getB = getB + 2
EndIf
        Bin = B(7)
If VB.Right(Bin, 1) = "1"Then
            getB = getB + 1
EndIf
Return getB
EndFunction

PublicFunction ByteToBool(ByVal B AsByte)
Dim getB AsString
If B >= 128 Then
            B = B - 128
            getB = "1"
```

```
Else
            getB = "0"
EndIf
If B >= 64 Then
            B = B - 64
            getB = getB &"1"
Else
            getB = getB &"0"
EndIf
If B >= 32 Then
            B = B - 32
            getB = getB &"1"
Else
            getB = getB &"0"
EndIf
If B >= 16 Then
            B = B - 16
            getB = getB &"1"
Else
            getB = getB &"0"
EndIf
If B >= 8 Then
            B = B - 8
            getB = getB &"1"
Else
            getB = getB &"0"
EndIf
If B >= 4 Then
            B = B - 4
            getB = getB &"1"
Else
            getB = getB &"0"
EndIf
If B >= 2 Then
            B = B - 2
            getB = getB &"1"
Else
            getB = getB &"0"
EndIf
If B >= 1 Then
            B = B - 1
            getB = getB &"1"
Else
            getB = getB &"0"
EndIf
Return getB
EndFunction

PrivateSub byte2bool(ByVal inp AsByte, ByRef outp AsBoolean())
If inp >= 0 AndAlso inp <= 255 Then
For i AsShort = 7 To 0 Step -1
If inp Mod 2 = 1 Then
                    outp(i) = True
                    inp -= 1
Else
                    outp(i) = False
EndIf
                inp /= 2
Next
Else
ThrowNew Exception("Input number is illegal.")
EndIf
EndSub

PrivateFunction bool2byte(ByVal inp AsBoolean()) AsByte
Dim outp AsByte = 0
For i AsShort = 7 To 0 Step -1
If inp(i) Then
                outp += CByte(Math.Pow(2.0, CDbl(7 - i)))
EndIf
Next
Return outp
EndFunction
#EndRegion

#Region" Button View "
PrivateSub btnView1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnView1.Click
Dim file AsString = txtImageHide.Text
Dim dlg As VisualDialog
        dlg = New VisualDialog(file, True)
        dlg.ShowDialog()
EndSub

PrivateSub btnView2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnView2.Click
Dim file AsString = txtImageExtract.Text
Dim dlg As VisualDialog
        dlg = New VisualDialog(file, False)
```

```
        dlg.ShowDialog()
EndSub
#EndRegion

#Region" Other Event "
PrivateSub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) HandlesMyBase.Load
'declare a variable of glass effect
Dim glass AsNew rtaGlassEffectsLib.rtaGlassEffect
        glass.TopBarSize = 110
        glass.LeftBarSize = 6
        glass.RightBarSize = 6
        glass.BottomBarSize = 6
        glass.ShowEffect(Me, PictureBox1)


Dim b AsInteger = 0
        b = countFile(pathBitmap)
For l AsInteger = 0 To b - 1
            File.Delete(pathBitmap & l &".bmp")
Next

EndSub

PrivateSub txtMessageFile_Enter(ByVal sender AsObject, ByVal e As System.EventArgs) Handles
txtMessageFile.Enter
        rdoMessageFile.Checked = True
EndSub

PrivateSub txtMessageText_Enter(ByVal sender AsObject, ByVal e As System.EventArgs) Handles
txtMessageText.Enter
        rdoMessageText.Checked = True
EndSub

PrivateSub txtImageHide_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
txtImageHide.TextChanged
If txtImageHide.Text = ""Then
            btnView1.Text = "No Carrier File Specified"
            btnView1.Enabled = False
Else
            btnView1.Text = "Read General Information"
            btnView1.Enabled = True
EndIf
EndSub

PrivateSub txtImageExtract_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles txtImageExtract.TextChanged
If txtImageExtract.Text = ""Then
            btnView2.Text = "No Carrier File Specified"
            btnView2.Enabled = False
Else
            btnView2.Text = "Read General Information"
            btnView2.Enabled = True
EndIf
EndSub

PrivateSub rdoMessageFile_CheckedChanged(ByVal sender AsObject, ByVal e As System.EventArgs) Handles
rdoMessageFile.CheckedChanged
        txtMessageFile.Enabled = True
        btnMessage.Enabled = True
EndSub

PrivateSub rdoMessageText_CheckedChanged(ByVal sender AsObject, ByVal e As System.EventArgs) Handles
rdoMessageText.CheckedChanged
        txtMessageFile.Enabled = False
        btnMessage.Enabled = False
EndSub

#EndRegion

PrivateSub btnHide_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnHide.Click

If txtPass1.Text = ""Then
            MessageBox.Show("Please enter The Password !", "SteganogrAVI", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            txtPass1.Focus()
Return
ElseIf txtPass2.Text = ""Then
            MessageBox.Show("Please Enter the Password !", "SteganogrAVI", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            txtPass2.Focus()
Return
ElseIf txtPass1.Text <> txtPass2.Text Then
            MessageBox.Show("Password not Match !", "SteganogrAVI", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            txtPass2.Focus()
Return
ElseIf txtImageHide.Text = ""Then
```

```
            MessageBox.Show("Please Select the Media !", "SteganograVI", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            txtImageHide.Focus()
Return
ElseIf txtMessageFile.Text = ""AndAlso txtMessageText.Text = ""Then
            MessageBox.Show("Please Enter the Message Text !"& vbLf &"Or Select a File Text !",
"SteganograVI", MessageBoxButtons.OK, MessageBoxIcon.Information)
            txtImageHide.Focus()
Return
Else
Dim aviReader AsNew AviReader()
Dim aviWriter AsNew AviWriter()
            aviReader.Open(imageHidePath)
If saveFileDialog1.ShowDialog() = DialogResult.OK Then
                saveToImage = saveFileDialog1.FileName
                ToolStripStatusLabel1.Text = "Hide the Message..."
                start = Now
                Timer1.Enabled = True
                btnHide.Enabled = False
                Application.DoEvents()
Else
Return
EndIf

            HideMessage(aviReader)

            aviWriter.saveImage(imageHidePath, saveToImage)

            aviReader.Close()
            aviWriter.Close()
            Timer1.Enabled = False
            finish = Now
            time = DateDiff(DateInterval.Second, start, finish)
Dim countProcess AsString = countTime(time)
            ToolStripStatusLabel1.Text = "A Hidden Message Successfully Saved"
            MessageBox.Show("The Message was Successfully Hidden in "& countProcess &" seconds",
"SteganograVI", MessageBoxButtons.OK, MessageBoxIcon.Information)
            btnHide.Enabled = True
            Application.DoEvents()
EndIf
EndSub

PrivateSub btnExtract_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnExtract.Click
If txtPass1.Text = ""Then
            MessageBox.Show("Please Enter the Password !", "SteganograVI", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            txtPass1.Focus()
Return
ElseIf txtPass2.Text = ""Then
            MessageBox.Show("Please Enter the Password !", "SteganograVI", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            txtPass2.Focus()
Return
ElseIf txtPass1.Text <> txtPass2.Text Then
            MessageBox.Show("Password not Match !", "SteganograVI", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            txtPass2.Focus()
Return
ElseIf txtImageExtract.Text = ""Then
            MessageBox.Show("Please Select the Media !", "SteganograVI", MessageBoxButtons.OK,
MessageBoxIcon.Information)
            txtImageExtract.Focus()
Return
Else
Dim aviReader AsNew AviReader()
            aviReader.Open(imageExtractPath)
            ExtractMessage(aviReader)
            aviReader.Close()
EndIf
EndSub

PrivateSub exportFrame(ByVal imagePath AsString)
Dim aviReader AsNew AviReader()
Dim path As [String] = "..\..\tempBitmap\"
        aviReader.Open(imagePath)

For n AsInteger = 0 To aviReader.CountFrames - 1
If File.Exists(path & n &".bmp") Then
                File.Delete(path & n &".bmp")
EndIf
            aviReader.ExportBitmap(n, path & n &".bmp")
Next
        aviReader.Close()
EndSub

PrivateSub SteganographyToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles SteganographyToolStripMenuItem.Click
Me.Close()
```

```
EndSub

PrivateSub HelpToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles HelpToolStripMenuItem.Click
        About.Show()
EndSub

PrivateSub AboutToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles AboutToolStripMenuItem.Click
        System.Windows.Forms.Help.ShowHelp(Me, "MANUALBOOK.pdf", HelpNavigator.AssociateIndex)
EndSub

PrivateFunction countTime(ByVal time AsInteger) AsString
Dim jam AsInteger = time \ 3600
Dim b AsInteger = jam * 3600
Dim menit AsInteger = (time - b) \ 60
Dim detik AsInteger = (time - b) Mod 60
Return (jam &" : "& menit &" : "& detik)
EndFunction

EndClass
```

➢ VisualDialog.vb

```
Imports System
Imports System.Drawing
Imports System.Windows.Forms
Imports System.Collections
Imports System.ComponentModel
Imports System.Data
Imports SteganogrAVI.AviFile

PublicClass VisualDialog

PrivateSub VisualDialog_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
HandlesMyBase.Load
'declare a variable of glass effect
Dim glass AsNew rtaGlassEffectsLib.rtaGlassEffect
        glass.TopBarSize = 68
        glass.LeftBarSize = 6
        glass.RightBarSize = 6
        glass.BottomBarSize = 6
        glass.ShowEffect(Me, PictureBox1)
EndSub

PublicSubNew(ByVal image AsString, ByVal hide AsBoolean)

        InitializeComponent()

        MediaPlayer.URL = image

Dim stream AsNew AviReader()
        stream.Open(image)

Dim h AsInteger, mn AsInteger, s AsInteger, frameRate AsInteger
Dim ukuran AsLong
Dim countPixels AsLong = 0
Dim aviCountFrames AsLong = 0
Dim sz As Size = stream.BitmapSize
        h = (CInt(stream.CountFrames / stream.FrameRate) \ 3600)
        mn = CInt(stream.CountFrames / stream.FrameRate) \ 60 - (h * 60)
        s = CInt(stream.CountFrames / stream.FrameRate) - mn * 60 - h * 3600
        ukuran = FileLen(image) / 1024
        frameRate = stream.FrameRate
        aviCountFrames = stream.CountFrames
        countPixels = sz.Width * sz.Height * aviCountFrames
Dim dataCanInsert AsDouble = ((((sz.Height * sz.Width * 3) * (aviCountFrames - 1)) \ 8.0) \ 1024)
Dim index AsInteger = image.LastIndexOf("\") + 1
If index > 0 Then
            image = image.Substring(index)
EndIf

        listInfo.Items(0).SubItems.Add(Convert.ToString(image))
        listInfo.Items(1).SubItems.Add(ukuran &" KB")
        listInfo.Items(2).SubItems.Add(sz.Width)
        listInfo.Items(3).SubItems.Add(sz.Height)
        listInfo.Items(4).SubItems.Add(h &" : "& mn &" : "& s)
        listInfo.Items(5).SubItems.Add(frameRate &" fps")
        listInfo.Items(6).SubItems.Add(aviCountFrames)
        listInfo.Items(7).SubItems.Add(sz.Width * sz.Height)
        listInfo.Items(8).SubItems.Add(countPixels)
If hide Then
            listInfo.Items(9).SubItems.Add(dataCanInsert.ToString() &" KB")
            stream.Close()
EndIf
        stream.Close()
EndSub
```

```
PrivateSub btnOk_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnOk.Click
        MediaPlayer.Ctlcontrols.stop()
        MediaPlayer.Dispose()
Me.Close()
EndSub
EndClass
```

➢ About.vb

```
Imports System
Imports System.Windows.Forms

PublicClass About

PrivateSub About_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) HandlesMyBase.Load
'declare a variable of glass effect
Dim glass AsNew rtaGlassEffectsLib.rtaGlassEffect
        glass.TopBarSize = 39
        glass.LeftBarSize = 123
        glass.RightBarSize = 6
        glass.BottomBarSize = 6
        glass.ShowEffect(Me, Label1, PictureBox1)
EndSub

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
Me.Close()
EndSub
EndClass
```