BAB III

ANALISIS DAN PERANCANGAN

Analisis dan perancangan sistem berfungsi untuk mempermudah dalam memahami dan menyusun tahapan selanjutnya yang akan dilakukannya penguraian 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 Analisis Masalah

Analisis ini mencakup semua permasalahan yang terjadi pada teknik *steganography* video ini seperti bagaimana proses penyisipan / *embed* pesan pada video, proses ekstraksi pesan, penggunaan password untuk menambah keamanan, serta proses pengukuran kualitas video untuk mengetahui bagaimana perbedaan video yang dihasilkan.

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,

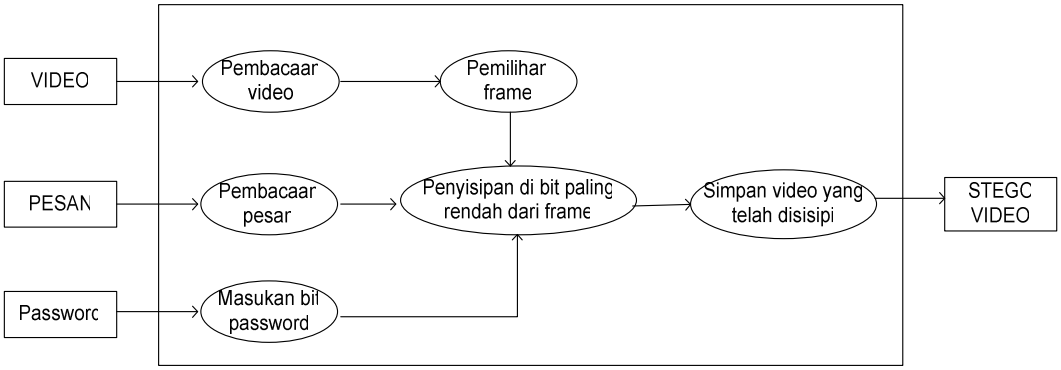
G=1 byte dan B=1 byte), berarti seluruhnya ada 65536 x 3 = 196608 byte. Karena dalam setiap byte hanya bisa menyembunyikan 1 bit pada LSB-nya, maka ukuran berkas rahasia maksimum yang dapat disimpan pada citra tersebut adalah 196608/ 8 = 24576 byte atau 1/8 dari ukuran citra tersebut.

3.1.1 Penyisipan pesan pada video

Sistem untuk penyisipan pesan pada video, membutuhkan masukan berupa video sebagai media penyisipan, pesan yang ingin disisipkan, serta password sebagai pengaman. Video yang digunakan sebagai media penyisipan pesan hanya video yang berformat (*.avi) yang belum terkompresi. 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 BMP.
2. Melakukan pembacaan terhadap file teks untuk mempersiapkan bit-bit file teks yang akan disisipkan pada video. Pesan yang disisipkan kedalam video bisa berformat apa saja tapi ukurannya tidak melebihi daya tampung penyimpanan dari video tersebut.
3. Menyisipkan bit-bit file pesan kedalam bit yang paling rendah dari salah satu 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 3.1 Diagram proses penyisipan pesan

3.1.2 Ekstraksi pesan yang disisipkan pada video

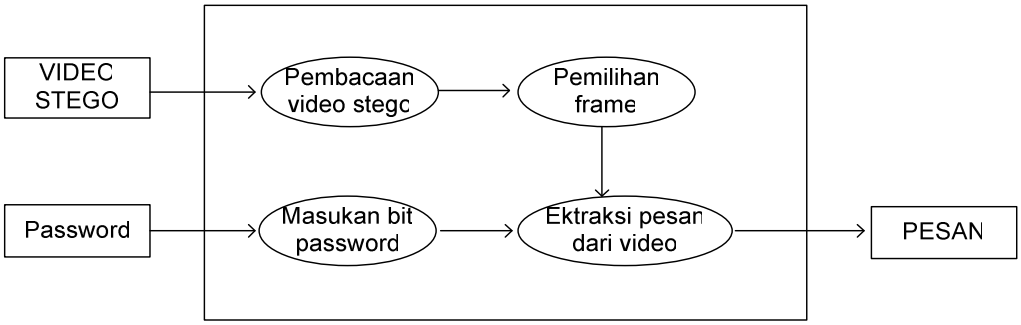
Sistem untuk mengekstraksi pesan pada video memerlukan dua buah masukan yaitu video yang mengandung pesan, serta password yang diinputkan sebagai pengamannya. Video ini memiliki format yang sama pada saat penyisipan yaitu video yang berformat *.avi.

Proses ekstraksi pesan dimulai dengan pemilihan frame dan koefesien pada video yang akan dibaca. Kemudian pesan didalamnya dibaca, menjadi pesan dalam bentuk acak, password yang diinputkan akan menjadi penentu kebenaran pesan, dimana deretan bilangan acak yang dibangkitkan oleh password akan mengatur bagaimana urutan pesan tersebut. Hanya password yang digunakan pada saat penyisipan yang dapat menghasilkan pesan asli kembali. 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.



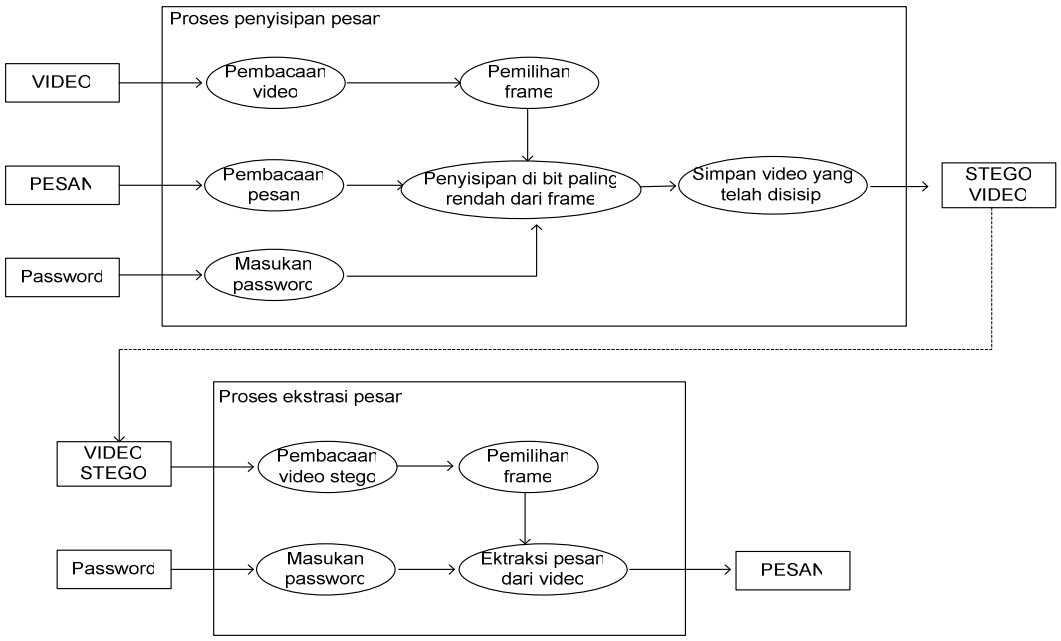
Gambar 3.2 Diagram proses ekstraksi pesan

3.2 Analisis Sistem

Dalam tahap analisis sistem dilakukan penguraian dari suatu sistem yang sedang berjalan. Proses ini bertujuan untuk mengidentifikasi kebutuhan sistem dan mengevaluasi permasalahan yang ada sehingga ditemukan kelemahan-kelemahan yang diharapkan untuk menunjang kemungkinan pengembangan sistem yang telah ada.

Pada umumnya prinsip kerja sistem aplikasi *steganography* pada video ini sama seperti halnya aplikasi *steganography* pada video lain yang telah ada, hanya

saja pada sistem aplikasi yang dibangun ini dengan menggunakan metode *Least Significant Bit* (LSB) dimana metode ini dapat menyembunyikan bit-bit dari berkas rahasia pada bit-bit segmen terendah dari frame video



Gambar 3.3 Arsitektur sistem *steganography* video

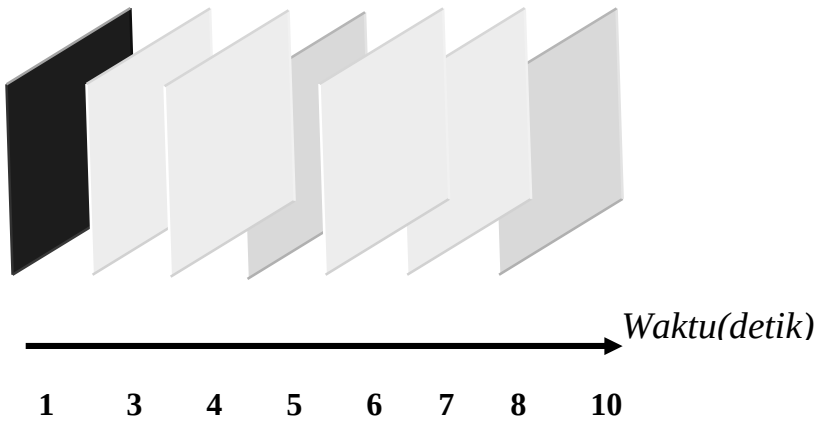
Fungsi utama dari aplikasi *steganography* video ini adalah supaya dapat menyisipkan pesan kedalam video, serta dapat mengekstraksinya kembali pesan tersebut. Objek masukan untuk modul proses penyisipan pesan yaitu video yang berformat *.avi yang belum terkompres, pesan rahasia bisa berupa file teks yang berformat apa saja tapi ukurannya tidak melebihi daya tampung penyimpanan dari video tersebut, 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 sudah memiliki pesan dan password. Pesan yang dibaca dari video masih dalam bentuk acak, sehingga masih harus diurutkan

kembali. Dengan password masukan, urutan data dapat ditentukan sehingga pesan akan terbaca dengan benar yang sekaligus menjadi keluaran dari modul ini.

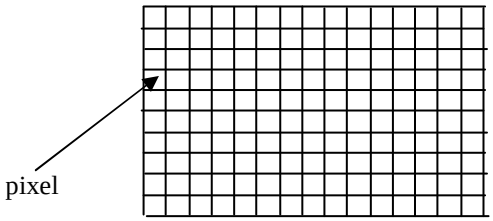
3.2.1 Struktur Video digital

Video digital pada dasarnya tersusun atas serangkaian *frame* yang ditunjukan pada gambar 3.4. 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 3.4 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 disampling disebut *picture element*, atau sering dikenal dengan istilah *pixel* (piksel) yang ditunjukan pada gambar 3.5.



Gambar 3.5 *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.

3.2.1.1 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. Namun resolusi yang tinggi akan mengakibatkan jumlah *bit* yang diperlukan untuk penyimpanan data dan pengiriman datanya menjadi meningkat.



Gambar 3.6 Resolusi dimensi *frame*

Pada gambar 3.6 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*.

3.2.1.2 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 untuk merepresentasikan 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 2^8 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 3.7 Variasi kedalaman *pixel*

Sebagai contoh, dalam gambar 3.7, 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

3.2.1.3 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 dua hal 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.

3.2.1.4 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, tiap komponen 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 :

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

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

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

Oleh karena itu persepsi mata manusia lebih peka kepada kecerahan dari pada warna, maka *crominance* cukup di sampling separuh dari *luminance*.

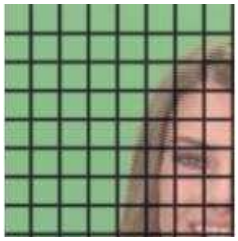
3.2.2 Metode Least Significant Bit (LSB)

Pada pembahasan ini akan dilakukan penerapan perhitungan proses penyimpanan berkas teks secara manual dengan menggunakan metode LSB. Sebagai contoh ilustrasi, digunakan sebuah citra bitmap 24-bit dibawah ini untuk digunakan sebagai *cover*.



Gambar 3.8 Citra bitmap 24-bit

Proses penyimpanan dengan menggunakan LSB ini pada dasarnya adalah memanipulasi piksel-piksel yang disusun oleh kombinasi tiga komponen warna RGB. Setiap komponen warna tersebut terdiri dari 8-bit dan hanya menyimpan 1-bit pada bit paling tidak berartinya, sehingga setiap piksel hanya dapat menyimpan 3-bit data saja. Piksel-piksel tersebut direpresentasikan sebagai elemen-elemen matriks seperti pada Gambar 3.9 dibawah ini.



Gambar 3.9 Matriks piksel citra

Misalkan akan disimpan sebuah huruf “A” (ASCII code 65, yaitu 01000001) pada citra di atas.

Data matriks piksel citra pada piksel ke-1, 2, 3, 4 … *n* =

00100111 11101001 11001000 00100111 11001000 11101001 11001000

00100111 11101001 00100111 11101001 11001000 … 00100111

Data yang akan disimpan adalah 01000001

Setiap bit yang merepresentasikan huruf “A” tersebut akan disimpan pada bit-bit terendah dari komponen warna pada matriks piksel citra di atas secara berurutan (diberi tanda garis bawah). Setelah semua tersimpan maka akan dihasilkan:

00100110 11101001 11001000 00100110 11001000 11101000 11001000

00100111 11101001 00100111 11101001 11001000 … 00100111

Terlihat hanya tiga bit yang berubah (ditandai cetak tebal dan garis bawah).

3.3 Analisis Lingkungan Implementasi

Untuk membangun aplikasi steganografi video ini diperlukan beberapa komponen untuk mendukung kinerja dari sistem tersebut mulai dari kebutuhan *software* / perangkat lunak sampai pada kebutuhan *hardware* / perangkat keras di mana masing–masing komponen tersebut saling bekerja sama untuk mencapai apa yang menjadi tujuan pembangunan sistem.

3. 3.1 Perangkat Keras Pendukung Penelitian

Spesifikasi *hardware* minimum yang dapat digunakan untuk menjalankan aplikasi ini kurang lebih memiliki spesifikasi sebagai berikut :

1. *Processor* Pentium II.
2. RAM 128 MB.
3. VGA 64 MB.
4. *Harddisk* 10 GB.
5. *Monitor*
6. *Mouse* dan *Keyboard*.

3.3.2. Perangkat Lunak Pendukung Penelitian

Perangkat lunak yang digunakan untuk menjalankan aplikasi ini memiliki spesifikasi sebagai berikut :

1. Windows XP SP2
2. Borland Delphi 7.0
3. Virtual Dub

3.4 Perancangan Sistem

Perancangan sistem adalah suatu proses yang menggambarkan bagaimana suatu sistem dibangun untuk memenuhi kebutuhan pada tahap analisis. Perancangan sistem bertujuan untuk memberikan gambaran sistem yang akan dibuat. Dengan kata lain perancangan sistem didefinisikan sebagai penggambaran atau pembuatan sketsa dari beberapa elemen yang terpisah kedalam satu kesatuan

yang utuh dan berfungsi. Selain itu juga perancangan bertujuan untuk lebih mengarahkan sistem yang terinci, yaitu pembuatan perancangan yang jelas dan lengkap yang nantinya akan digunakan untuk pembuatan simulasi. Aplikasi *steganography* ini dibuat dengan sederhana, sehingga diharapkan *user* dapat dengan mudah menggunakan aplikasi *steganography* ini.

3.4.1 Perancangan Prosedural

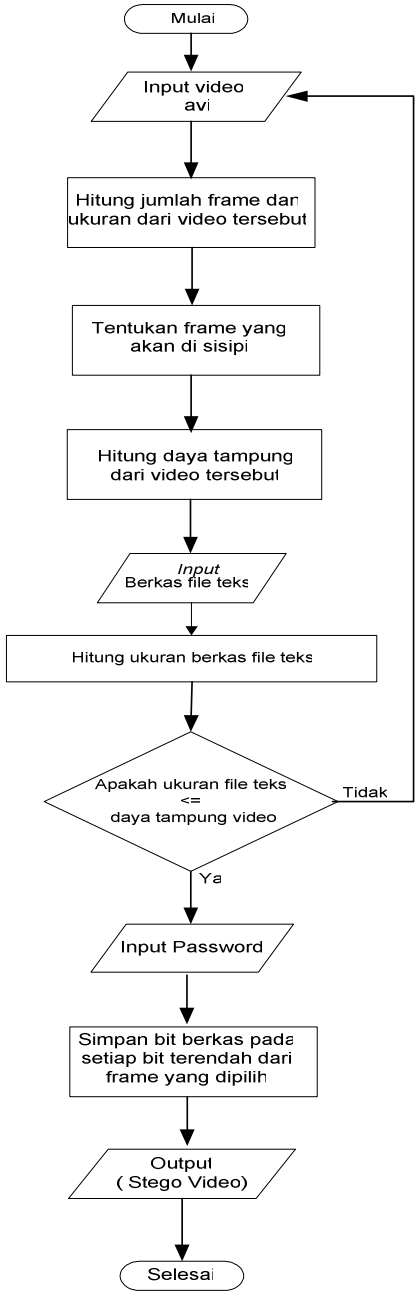
Pada bagian ini akan dibahas bagaimana prosedur dan aliran dokumen yang sedang berjalan dari satu entitas ke entitas lain yang digambarkan dalam bentuk *flowmap.* Berdasarkan hasil penelitian dan observasi pada sistem yang sedang berjalan, terdapat dua prosedur pada sistem manual yang sedang berjalan yaitu proses penyisipan (*embed*) dan proses ekstraksi (*extract*).

3.4.1.1 *Flowmap* Proses Penyisipan

Pada prosedur ini kegiatan yang dilakukan yaitu :

1. Inputkan video yang berformat *.avi yang belum terkompres, lalu hitung ukuran jumlah frame, kemudian menentukan frame yang akan dijadikan sebagai media penyisipan.
2. Inputkan file teks yang akan disisipkan.
3. Inputkan password sebagai pengamanan, dimana password tersebut dapat digunakan pada saat proses ekstraksi.
4. Tentukan alamat dan nama file yang akan menyimpan file video stego hasil proses penyisipan tersebut.

5. Jika semua inputan yang diperlukan sudah diinputkan, maka proses penyisipan dapat terjadi yaitu dengan cara menyisipkan bit berkas pada setiap bit yang paling rendah dari frame, jika belum maka proses penyisipan tidak akan terjadi.



Gambar 3.10 Flowmap Proses Penyisipan / *Embed*

3.4.1.2 *Flowmap* Proses Ekstraksi

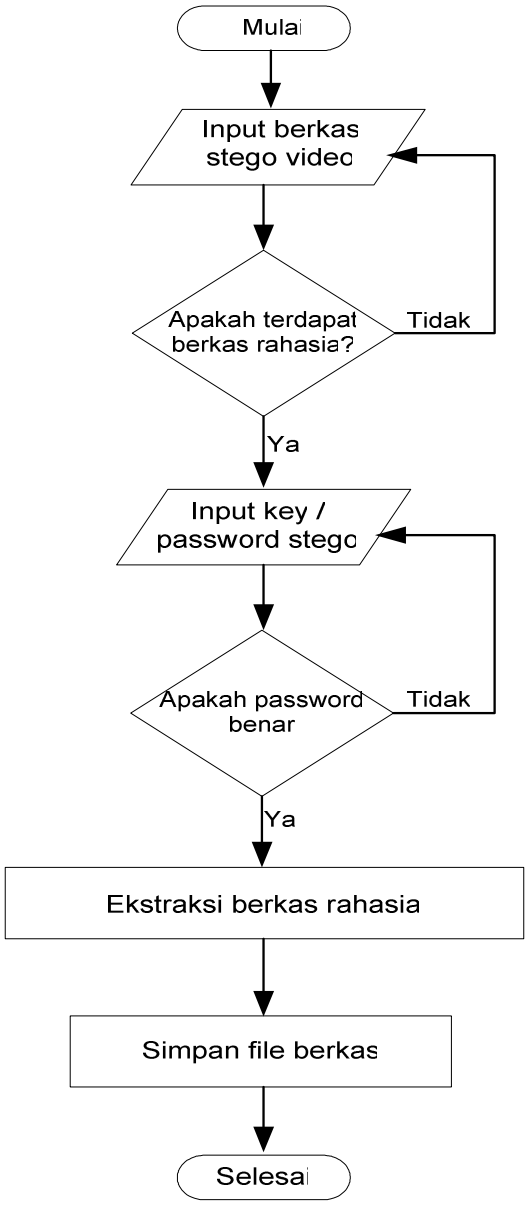
Pada prosedur ini kegiatan yang dilakukan yaitu :

1 Inputkan video stego yang berformat *.avi yang sudah mengandung file teks.

2 Inputkan password, apakah password yang di masukan sama dengan password yang diinputkan pada saat proses penyisipan, jika sama maka lanjutkan ke proses selanjutnya jika tidak sama maka inputkan lagi password.

3 Tentukan alamat dan nama file yang akan menyimpan file teks hasil proses ekstraksi tersebut.

4 Jika semua inputan yang diperlukan sudah diinputkan, maka proses ekstraksi dapat terjadi, jika belum di masukan maka proses ekstraksi tidak terjadi.



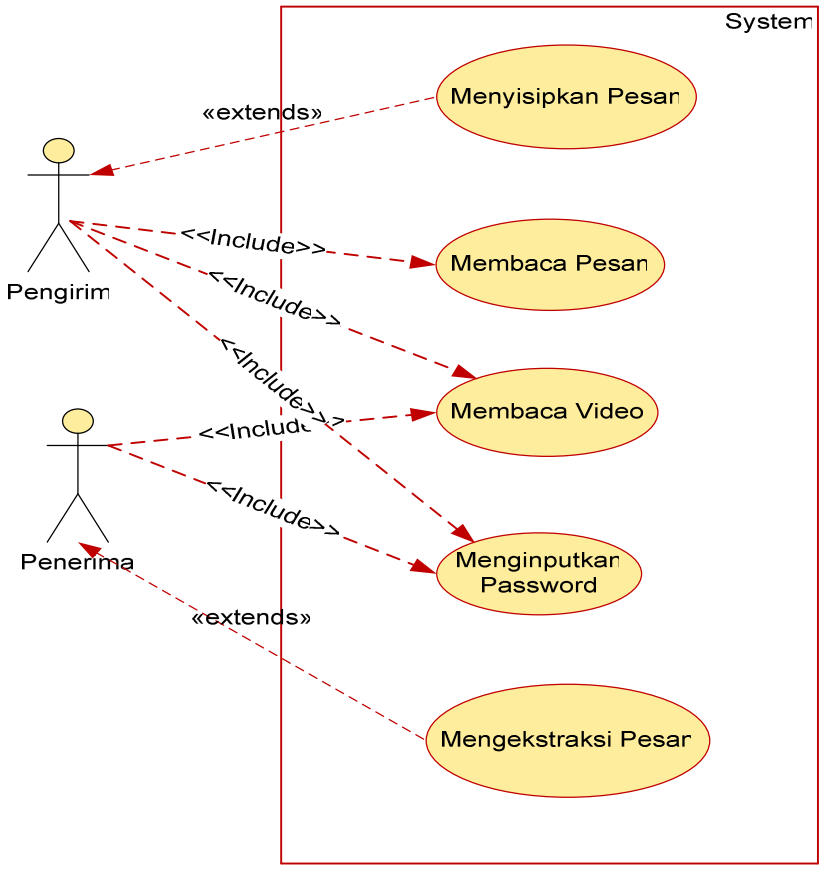
Gambar 3.11 Flowmap proses ekstraksi

3.4.2 Perancangan Berorientasi Objek

Perancangan yang dibuat akan menggunakan metode UML (*Unified Modelling Language*), diataranya : *use case diagram*, class diagram, *sequence diagram, activity diagram.*

3.4.2.1 Use case Diagram

Diagram *use-case* (*use-case diagram)* merupakan salah satu diagram untuk memodelkan aspek perilaku sistem. Masing-masing diagram *use-case* menunjukan sekumpulan *use-case*, *actor* dan hubungannya. Diagram *use-case* sangat penting untuk memvisualisasikan, menspesifikasikan, dan mendokumentasikan kebutuhan perilaku sistem. Diagram *use-case* juga merupakan pusat pemodelan perilaku sistem, subsistem dan kelas. Use case juga dapat digunakan mendeskripsikan interaksi antara pengguna sistem dengan sistem itu sendiri, dengan memberikan sebuah narasi tentang bagaimana sistem tersebut digunakan.



Gambar 3.12 use case *steganography* video

Perangkat lunak *steganography* video ini memiliki 5 buah use case dan 2 buah aktor. Aktor pengirim merupakan pengguna yang melakukan penyisipan/embed pesan rahasia kedalam video, sedangkan aktor penerima merupakan pengguna yang melakukan ekstraksi pesan rahasia yang telah disisipkan ke dalam video.

Use case menyisipkan pesan digunakan untuk melakukan menyisipkan data pesan kedalam video. Sedangkan use case membaca pesan digunakan untuk melakukan pembacaan pesan dan validasi ukuran pesan terhadap daya tampung video. Maka kedua use case ini hanya digunakan pada saat proses penyisipan.

Sedangkan untuk use case mengekstraksi pesan digunakan untuk mengekstraksi kembali pesan yang telah disisipkan kedalam video, maka use case ini hanya dapat digunakan pada proses ekstraksi.

Use case membaca video digunakan untuk melakukan pembacaan terhadap video dan validasi ukuran video serta apakah video tersebut sudah mengandung pesan atau belum. Dan use case menginputkan password digunakan pada saat proses penyisipan pesan sekaligus pada saat proses ekstraksi pesan, sehingga kedua use case ini terdapat pada kedua proses. Untuk penjelasan tiap use case secara lengkap dapat dilihat pada tabel skenario use case.

Tabel 3.1 skenario use case membaca video

Identifikasi	
Nama	Membaca Video
Deskripsi	Melakukan pembacaan, validasi dan parsing video.
Aktor	Pengirim dan penerima

Proses skenario	
Aksi Aktor	**Reaksi Sistem**
1.Pengirim atau penerima mengklik button browser input video pada main menu	2. Menampilkan layar browser file video .avi.
3. Pengirim atau penerima memilih file video lalu mengklik button open pada layar browser file video.	4. Sistem melakukan validasi ukuran video dan mengecek apakah video yang dipilih tersebut sudah diembed atau belum.
	5. Sistem memberikan informasi tentang video yang dipilih.
	6. Sistem mencatat semua info data video dan menyimpannya.
Kondisi akhir	Info video sudah tervalidasi dan tersimpan.

Tabel 3.2 skenario use case membaca pesan

Identifikasi	
Nama	Membaca Pesan
Deskripsi	Melakukan pembacaan pesan yang akan disisipkan
Aktor	Pengirim
Proses skenario	
Aksi Aktor	**Reaksi Sistem**
1. Pengirim mengklik button browser input file pesan teks pada main menu.	2. Sistem menampilkan layar browser file pesan.
3. Pengirim memilih file teks yang akan disisipkan lalu mengklik button open pada layar browser file pesan.	4. Sistem melakukan validasi ukuran dan format pesan yang disisipkan

	5. Sistem mencatat semua info data pesan dan menyimpannya.
Kondisi akhir	Info file pesan sudah tervalidasi dan tersimpan.

Tabel 3.3 skenario use case menyisipkan pesan

Identifikasi	
Nama	Menyisipkan Pesan
Deskripsi	Melakukan penyisipan pesan kedalam video.
Aktor	Pengirim
Prekondisi	Use case membaca video, use case membaca, dan use case menginput password telah dilakukan.
Proses skenario	
Aksi Aktor	**Reaksi Sistem**
1. Pengirim mengklik button embed pada main menu.	2. Sistem menampilkan form input password sebagai keamanan.
3. pengirim mengisi teks password dan mengklik button ok pada form input password	4. Sistem melakukan pembacaan terhadap password yang diinputkan tersebut.
	5. Sistem melakukan penyisipan atau embed data pesan dengan data video.
	6. Sistem menghasilkan video yang memiliki pesan dan menyimpannya.
Kondisi akhir	Video sudah memiliki pesan.

Tabel 3.4 skenario use case mengektraksi pesan

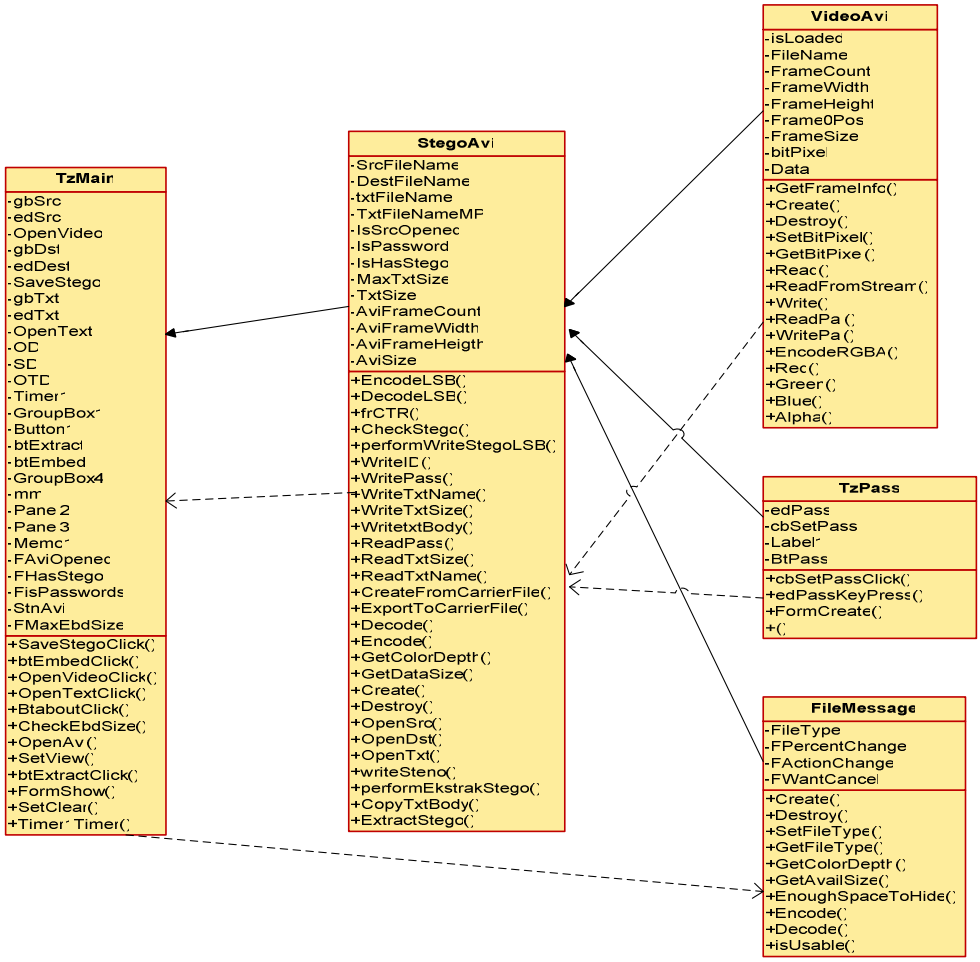
Identifikasi	
Nama	Mengektraksi Pesan
Deskripsi	Melakukan mengekstraksi pesan yang telah disisipkan ke dalam video.
Aktor	Penerima
Prekondisi	Use case membaca video dan use case menginputkan password sudah dilakukan.
Proses skenario	
Aksi Aktor	**Reaksi Sistem**
1. Penerima mengklik button extratc pada main menu	2. Sistem menampilkan form input password
3. penerima mengisi text password yang sama pada saat input password proses embed lalu klik button ok.	4. Sistem melakukan pembacaan terhadap password yang diinputkan
	5. sistem menampilkan layar browser save file teks.
6. penerima memilih folder sebagai tempat untuk menyimpan hasil ekstraksi lalu mengklik button save	5. Sistem melakukan mengekstraksi data pesan dari video.
	6. Sistem menghasilkan pesan dari hasil ekstraksi dan menyimpannya
Kondisi akhir	Proses ekstraksi pesan telah berhasil

Tabel 3.5 skenario use case Menginputkan Password

Identifikasi	
Nama	Menginputkan Password
Deskripsi	Sebagai password pada saat penyisipan pesan dan pada saat mengekstraksi pesan.
Aktor	Pengirim dan penerima
Prekondisi	Use case membaca video dan use case membaca pesan telah dilakukan.
Proses skenario	
Aksi Aktor	**Reaksi Sistem**
1. Pengirim atau penerima masukan password pada text edit input password yang ada pada form input password	2. Sistem memasukan password yang diketikan pengirim sebagai pengaman dalam proses penyisipan dan proses ekstraksi
	3. Sistem membaca password yang akan digunakan sebagai password dalam proses mengekstraksi pesan
	4. Sistem mengolah password dan menyimpannya
Kondisi akhir	Password untuk embed dan extract pesan telah tersimpan.

3.4.2.2 Class Diagram

Class diagram digunakan untuk mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat dalam sistem tersebut. *Class diagram* juga dapat menunjukan properti dan operasi sebuah *class* dan batasan-batasan yang terdapat dalam hubungan objek tersebut.



Gambar 3.13 class diagram aplikasi steganography video

Penjelasannya :

- Proses Penyisipan

Pada proses penyisipan membutuhkan 3 kelas masukan yaitu : kelas VideoAvi, kelas FileMessage, dan kelas TzPass.

Dimana kelas “VideoAvi” menangani segala proses terhadap video seperti validasi format video dan jumlah frame video yang dimasukan, mencari frame video yang tepat untuk dijadikan media penyisipan dan untuk menentukan kapasitas daya tampung dari video. Kelas “FileMessage” mengatur pembacaan

file teks yang akan diembed dan untuk mengecek apakah ukuran file teks tersebut sesuai dengan daya tampung video. Kelas “TzPass” untuk menyimpan password sebagai pengaman dari proses embed yang dapat digunakan sebagai password pada saat mengekstraksinya kembali. Kelas “StegoAvi” merupakan kelas yang mengatur proses penyisipan pesan kedalam video, dan juga digunakan untuk menyimpan hasil dari proses penyisipan.

- Proses Ekstraksi

Pada proses ekstraksi membutukan 2 kelas masukan yaitu kelas VideoAvi dan kelas TzPass.

Dimana Kelas “VideoAvi” digunakan untuk melakukan pembacaan dan pengecekan terhadap video yang diinputkan apakah sudah mengandung pesan atau tidak. Kelas ”TzPass” digunakan melakukan pembacaan terhadap password yang diinputkan apakah sama dengan password yang inputkan pada saat proses penyisipan. Jika sama maka video tersebut dapat diekstrak, jika tidak sama maka video stego tersebut tidak dapat diekstrak. Kelas ”FileMessage” digunakan untuk menyimpan file teks hasil dari proses ekstrak video stego. Kelas ”StegoAvi” digunakan untuk proses ektraksi file teks dari video stego

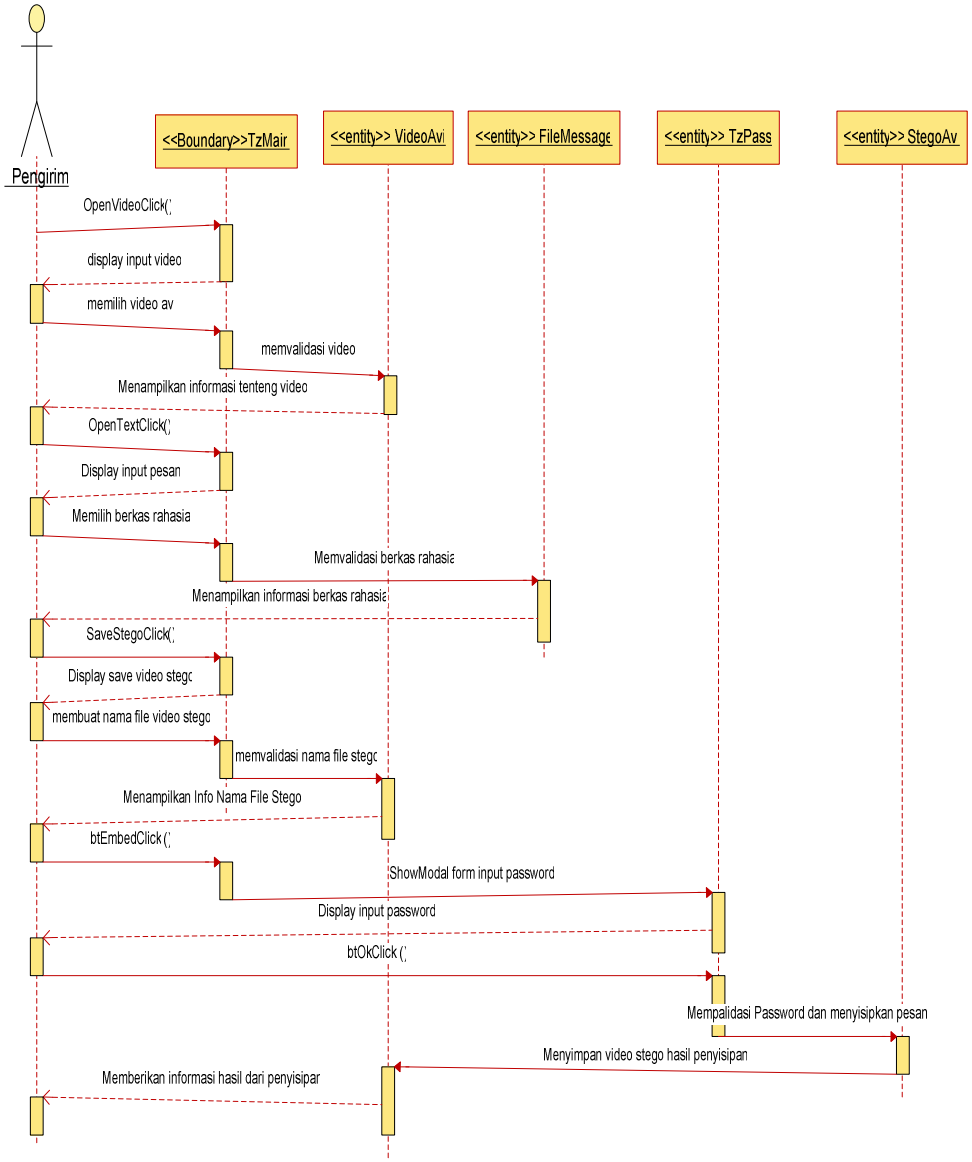
3.4.2.3 *Sequence Diagram*

Dalam metodologi berorientasi objek untuk menunjukkan bagaimana suatu objek berkomunikasi dengan objek yang lain dengan memperhatikan urutan waktu yang dimodelkan didalam *sequence diagram*.

Sequence diagram terdiri dari objek-objek yang dipresentasikan dalam

pembelajaran didalamnya memuat nama dari objek, pengenalan objek, baca objek yang direpresentasikan dengan garis panah dan *time* (waktu) yang direpresentasikan dengan *vertical progression*.

3.4.2.3.1 *Sequence Diagram* Proses Penyisipan / *Embed*



Gambar 3.14 Sequence Diagram Proses Penyisipan / *Embed*

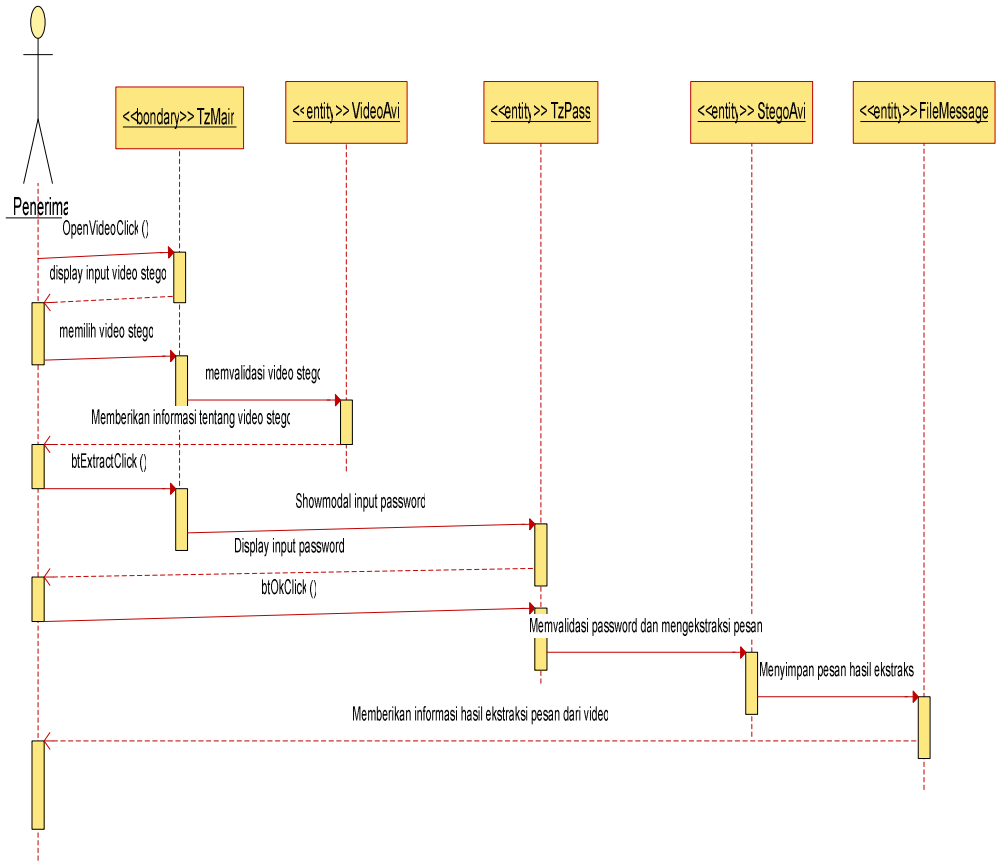
Penjelasan dari gambar 3.14 sebagai berikut :

1. User pengirim mengklik button OpenVideo pada TzMain, kemudian sistem akan menampilkan layar browser input video untuk memilih video yang diinputkan.
2. User pengirim memilih video yang akan di inputkan dan mengklik button open pada layar browser input video, kemudian sistem akan memanggil kelas VideoAvi untuk memvalidasi video dan memberikan informasi dari video yang pilih tersebut kepada user pengirim.
3. User pengirim mengklik button OpenText pada TzMain, kemudian sistem akan menampilkan layar browser input file pesan untuk memilih pesan yang akan disisipkan kedalam video.
4. User pengirim memilih file pesan yang akan disisipkan pada video dan mengklik button open pada layar browser input file pesan, kemudian sistem akan memanggil kelas FileMessage untuk memvalidasi file pesan dan memberikan informasi dari file pesan yang dipilih tersebut.
5. User pengirim mengklik button SaveStego, kemudian sistem akan menampilkan layar browser save file stego, User pengirim membuat nama file dan alamat folder untuk menyimpan hasil penyisipan lalu mengklik button save, kemudian sistem akan memanggil lagi kelas VideoAvi untuk memvalidasi dari nama file stego dan memberikan informasi alamat dan nama file hasil penyisipan kepada user pengirim.
6. User pengirim mengklik button btEmbed, kemudian sistem menampilkan form input password yang digunakan sebagai pengamanan, lalu user

pengirim mengisi password dan mengklik button OK, lalu sistem akan memanggil kelas TzPass untuk menginputkan dan memvalidasi password tersebut sebagai password penyisipan.

7. Kemudian sistem akan memanggil kelas StegoAvi untuk menyisipkan pesan kedalam video, kemudian menyimpan video stego dan menampilkan informasi hasil dari proses penyisipan tersebut.

3.4.2.3.2 *Sequence Diagram* Proses Ekstraksi



Gambar 3.15 Sequence Diagram Proses Ekstraksi

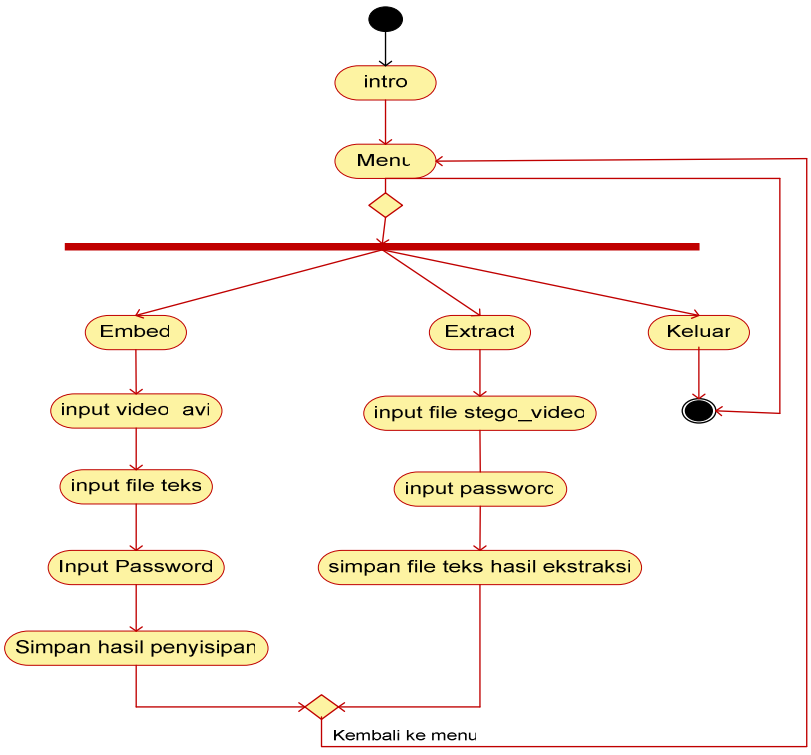
Penjelasan dari gambar 3.15 sebagai berikut :

1. User penerima mengklik button OpenVideo pada TzMain, kemudian sistem akan menampilkan layar browser input video stego untuk memilih video stego.
2. User penerima memilih file video stego yang akan di inputkan dan mengklik button open pada layar browser input video stego, kemudian sistem akan memanggil kelas VideoAvi untuk memvalidasi apakah video tersebut telah memiliki pesan atau tidak dan memberikan informasi dari video stego yang pilih tersebut.
3. User penerima mengklik button btExtract pada TzMain, kemudian sistem menampilkan browser save file teks untuk membuat nama file dan memilih alamat folder untuk menyimpan file teks hasil dari ekstraksi tersebut, lalu user penerima mengklik button save, kemudian sistem akan memanggil kelas FileMessage untuk memvalidasi nama file pesan dan memberikan informasi alamat file pesan hasil dari ekstraksi tersebut. Kemudian sistem menampilkan form input password.
4. User penerima masukan password yang sama dengan password pada saat penyisipan, lalu mengklik button OK pada form input password, kemudian sistem akan memanggil kelas TzPass untuk memvalidasi password yang diinputkan apakah sama dengan password pada saat penyisipan.
5. Sistem akan memanggil kelas StegoAvi untuk mengekstraksi pesan dari video. Lalu sistem memanggil kelas FileMessage untuk menyimpan pesan hasil dari proses ekstraksi dan memberikan informasi kepada user penerima.

3.4.2.4 *Activity Diagram*

Activity diagram didesain untuk memperlihatkan apa yang terjadi selama suatu proses dan operasi berlangsung. *Activity diagram* adalah suatu perluasan dari *state diagram*. *State diagram* menunjukan state-state suatu objek dan representasi aktivitas dinyatakan sebagai state yang dihubungkan dengan garis berarah.

Setiap aktivitas direpresentasikan dengan suatu *rounded rectangle*. Pemrosesan pada suatu aktivitas yang telah selesai akan dikirimkan ke aktivitas berikutnya secara otomatis untuk melakukan aktivitas berikutnya. Garis berarah menyatakan perpindahan aktivitas dari suatu aktivitas ke aktivitas berikutnya. Berikut ini merupakan *activity diagram* dari aplikasi *steganography* video, yang dapat dilihat pada gambar 3.26.

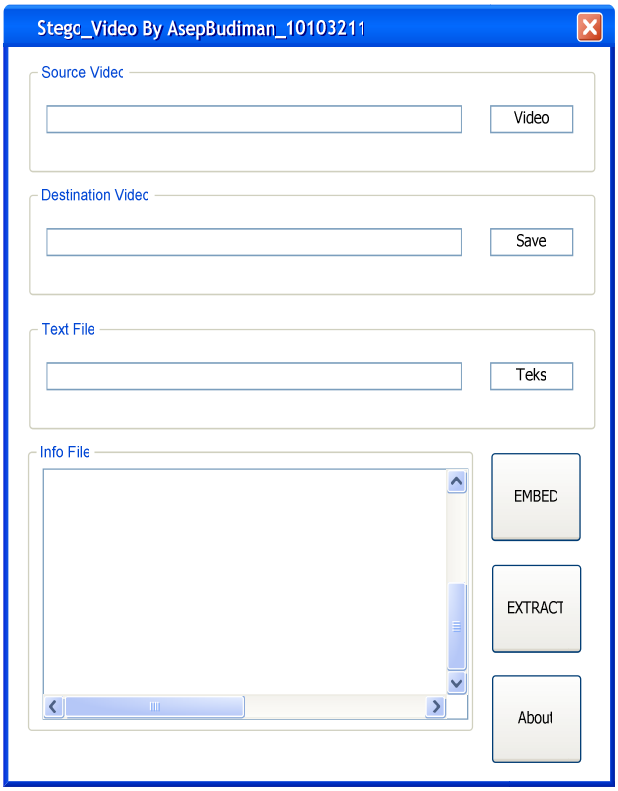


Gambar 3.16 Activity diagram dari aplikasi *steganography* video

3.4.3 Perancangan Antar Muka

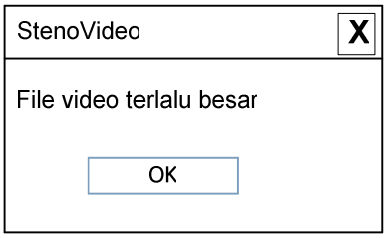
Perancangan antar muka dibutuhkan untuk mewakili keadaan sebenarnya dari aplikasi yang akan dibangun. Berikut akan disajikan perancangan antar muka dari aplikasi yang akan dibangun.

3.4.3.1 Perancangan Antar Muka Main Menu (M01)

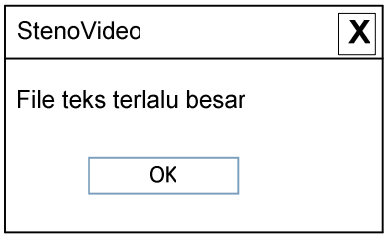
M01	NAVIGATOR
	Tampilan Main Menu >Klik button video untuk menginput video avi yang akan di jadikan sebagai media penyisipan jika video yang dimasukan terlalu besar maka akan keluar pesan A01 >Klik button Save untuk menyimpan berkas stego video hasil embed >Klik button Teks untuk menginput file teks yang akan disisipkan kedalam video bila file teks terlalu besar dan tidak cukup untuk disisipkan maka keluar pesan A02 >Klik button EMBED proses embedding dan akan muncul form M02 jika proses embed berhasil maka akan muncul pesanA03 >Klik button EXTRACT untuk proses ekstraksi dan akan muncul form M02 dan muncul pesan A04 dan jika password yang dimasukan salah maka akan muncul pesan A05 >Klik button About akan muncul M03 >Klik button (x) untuk keluar dari program
Ket Warna form sesuai dengan setting window Jenis huruf MS sant serif dengan ukuran 8 pt warna hitam ukuran form 600x450 pixel warna dan ukuran disesuaikan	

Gambar 3.17 perancangan antarmuka main menu

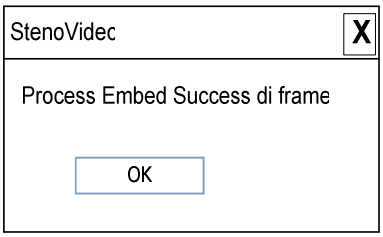
A01



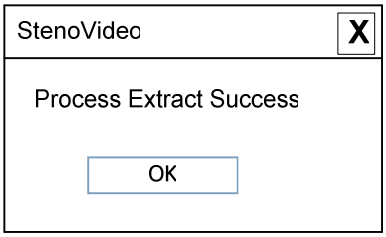
A02



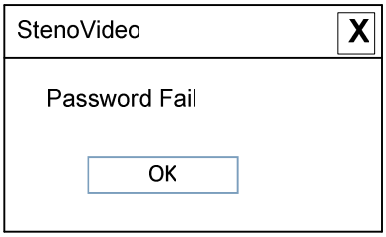
A03



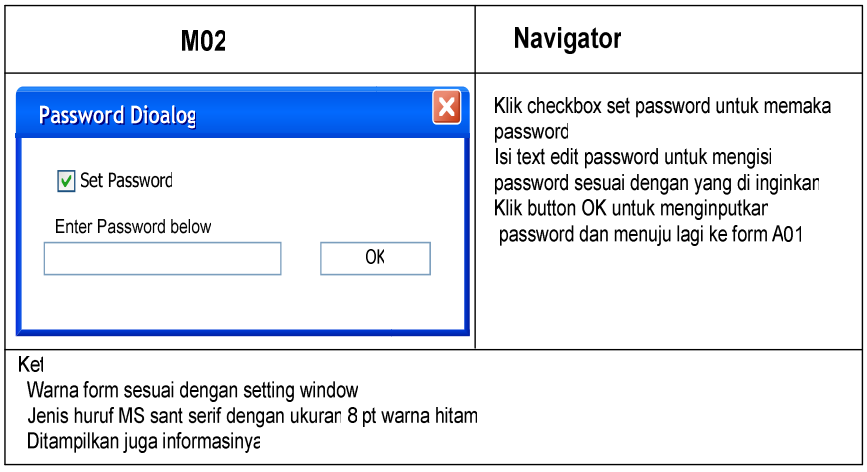
A04



A05



3.4.3.2 Perancangan antar muka form input password (M02)



Gambar 3.18 perancangan antarmuka form input password