

## BAB IV

### PERANCANGAN DAN HASIL IMPLEMENTASI

#### 4.1 Analisa Kebutuhan Aplikasi Enkripsi Teks

Analisis kebutuhan sistem bertujuan untuk melakukan identifikasi persoalan yang mungkin akan muncul dalam pembuatan aplikasi. Hal ini dilakukan agar dalam proses perancangan aplikasi tidak terjadi kesalahan-kesalahan yang berarti sehingga aplikasi yang dihasilkan dapat berjalan dengan baik dan selesai tepat waktu seperti yang telah ditentukan. Dalam analisis kebutuhan sistem ini akan dibahas mengenai kebutuhan perangkat lunak (*software*), kebutuhan perangkat keras (*hardware*), sedangkan pemodelan sistem yang digunakan akan dibahas pada bagian selanjutnya. Kebutuhan-kebutuhan tersebut antara lain:

a. Analisis kebutuhan perangkat lunak (*software*):

- Sistem operasi Microsoft Windows 7  
Sistem operasi yang digunakan adalah Microsoft Windows 7, karena sistem operasi ini terbukti sebagai salah satu sistem operasi terbaik yang ada sekarang.
- Software Microsoft Visual Basic 6. 0  
Bahasa *pemrograman* yang digunakan untuk perancangan sistem aplikasi adalah Microsoft Visual Basic 6. 0, dikarenakan bahasa ini merupakan bahasa mudah dipelajari dan mendukung dalam perhitungan aritmatika dan logika yang dibutuhkan dalam pembuatan aplikasi.
- Software Microsoft Word 2010  
Software ini digunakan dalam penelitian untuk mendukung aplikasi sebagai salah satu fasilitas dalam penyimpanan hasil enkripsi dan dekripsi data.
- Software Rational Rose 03  
Kebutuhan akan pembuatan UML untuk mendukung proram aplikasi pada tahap pembuatan model.

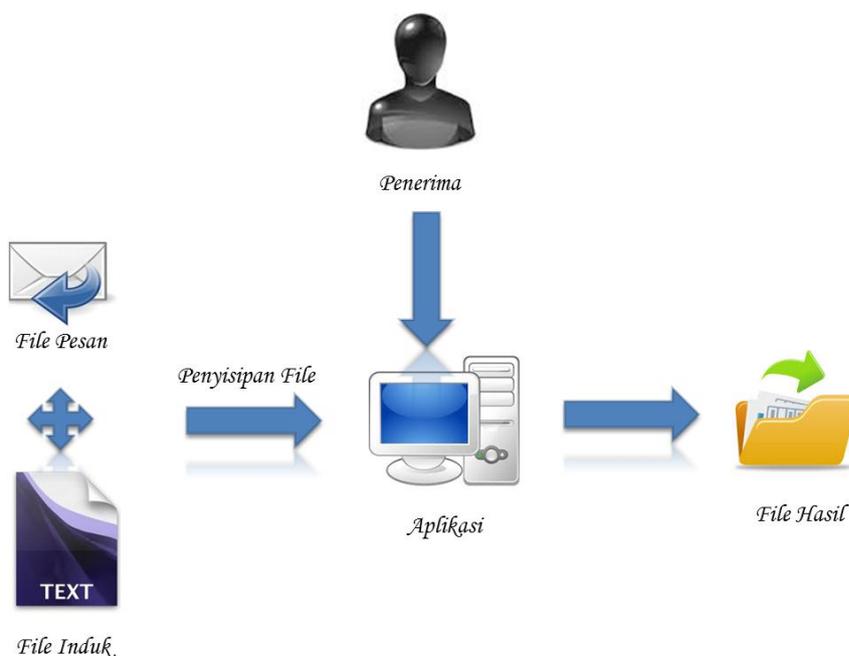
- b. Spesifikasi kebutuhan perangkat keras :
- Prosesor minimal dual core.
  - Harddisk minimal 100 GB.
  - Memori DDR minimal 512 MB.
  - Layar display minimal 14'.
  - Satu buah keyboard dan mouse standard.

#### 4.2 Prosedur Persiapan Pembuatan Aplikasi

Langkah-langkah pembuatan aplikasi Kripto dalam penelitian ini adalah sebagai berikut :

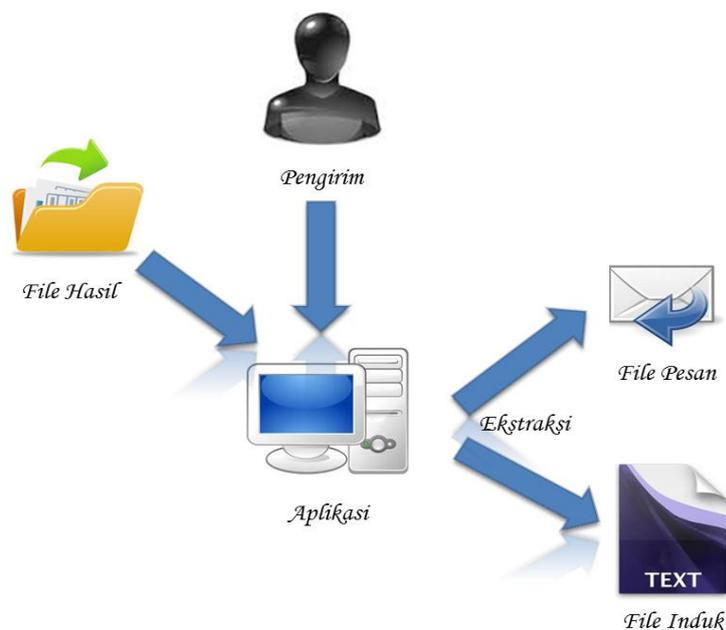
- Mengumpulkan materi-materi yang dibutuhkan.
- Melakukan pembatasan terhadap materi tersebut.
- Mempersiapkan dan melakukan instalasi perangkat keras.
- Mempersiapkan dan melakukan instalasi perangkat lunak sesuai dengan spesifikasi kebutuhan.

Setelah langkah-langkah persiapan aplikasi dilakukan, skema yang diharapkan menampilkan hasil maksimal di desain sebagai berikut:



**Gambar 13 : Skema Proses Penyisipan File**

Pada gambar 12, dapat diasumsikan bahwa baik *file* pesan maupun *file* induk dapat berupa *file* dengan format . *txt*, . *doc*, atau . *docx*, sehingga format dari *file* hasil mengikuti format *file* induk.



**Gambar 14 : Skema Proses Ekstraksi File**

### 4.3 Unit Bahasa Pemodelan

#### 4.3.1 Use Case Diagram

Dalam pembuatan *use case diagram* ini, sangat penting sekali untuk mengetahui kandidat dari *actor-actor* yang pada umumnya dapat ditentukan melalui beberapa pertanyaan berikut, antara lain yaitu siapakah yang menggunakan sistem, apa yang dipengaruhi oleh kehadiran sistem tersebut, masalah apa yang dikerjakan oleh sistem yang sedang dikembangkan, untuk siapakah dan untuk apa sistem tersebut dikembangkan, dan bagaimanakah *user* dapat menggunakan sistem?.

Dalam bahasa pemodelan ini, penulis menggunakan 2 (dua) buah aktor yaitu pengirim dan penerima. Aktor tersebut mempunyai karakteristik yang berbeda dalam hal menggunakan aplikasi dan menggunakan *file* yang telah di ketahui adanya proses.

Pengirim merupakan orang yang mengirimkan pesan dalam bentuk *file*, kemudian *file* tersebut akan dienkripsi dan disisipkan pada *file* induk, sehingga

orang lain tidak dapat mengetahui adanya *file* sisipan tersebut. Sedangkan penerima, merupakan orang yang menerima *file*. *File* sisipan yang dikirimkan oleh pengirim tadi dapat dibuka oleh penerima melalui *program* atau sandi yang sama. Proses ini merupakan proses pemisahan *file* induk dengan *file* pesan yang telah disisipkan oleh pengirim.

**Tabel 4 : Skenario use case Proses Kripto dan Stego**

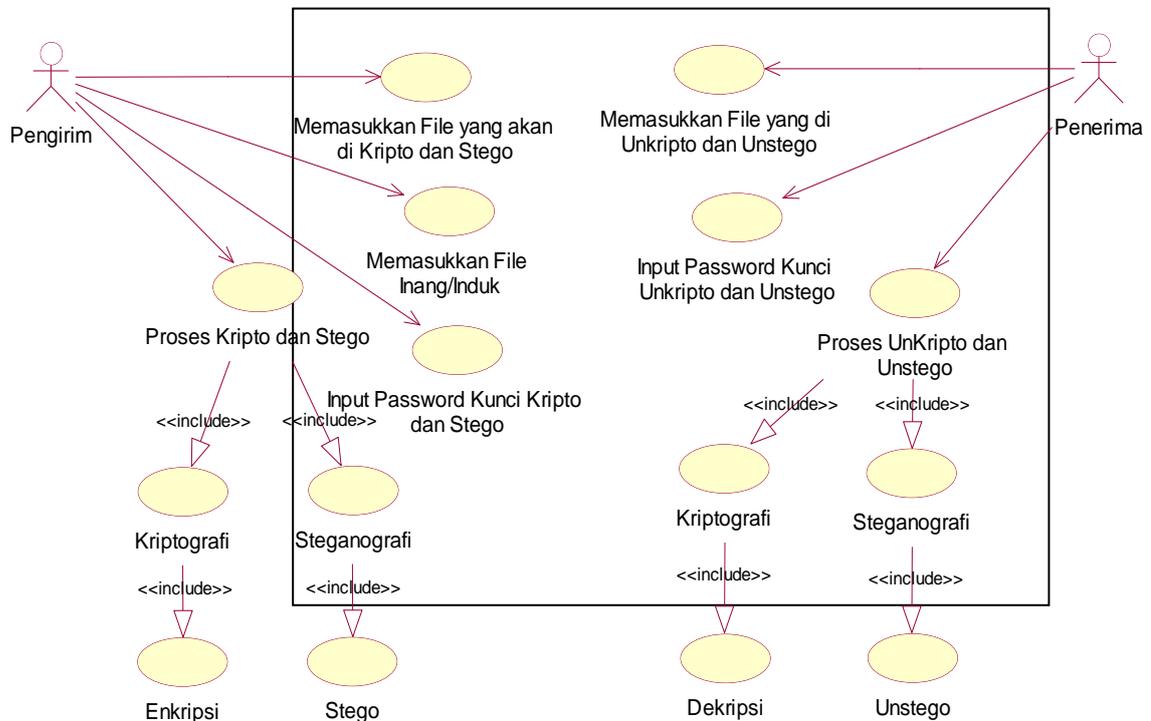
<i>Use-case :</i>	<ul style="list-style-type: none"> <li>• Proses Kripto dan Stego</li> </ul>
<i>Primary actor :</i>	<ul style="list-style-type: none"> <li>• Pengirim</li> </ul>
<i>Goal :</i>	<ul style="list-style-type: none"> <li>• <i>File</i> berhasil disembunyikan</li> </ul>
<i>Precondition :</i>	<ul style="list-style-type: none"> <li>• Memasukkan nama <i>file</i> asli yang akan disembunyikan</li> <li>• Memasukkan nama <i>file</i> induk tempat <i>file</i> tersembunyi</li> <li>• Memasukkan kata sandi/kata kunci</li> </ul>
<i>Trigger :</i>	<ul style="list-style-type: none"> <li>• Pengirim ingin menyembunyikan <i>file</i></li> </ul>
<i>Skenario:</i>	<ol style="list-style-type: none"> <li>1 Pengirim membuka aplikasi Steganographer.</li> <li>2 Pengirim memasukkan nama <i>file</i> yang akan disembunyikan.</li> <li>3 Pengirim memasukkan nama <i>file</i> untuk tempat <i>file</i> disembunyikan.</li> <li>4 Pengirim memberikan kata kunci sebagai pelengkap sandi.</li> <li>5 Pengirim menekan tombol Stego untuk memprosesnya.</li> </ol>
<i>Alternate Flow :</i>	<ul style="list-style-type: none"> <li>• Pengirim keluar dari aplikasi</li> </ul>
<i>Priority :</i>	<ul style="list-style-type: none"> <li>• Moderate priority</li> </ul>
<i>Frequency of use :</i>	<ul style="list-style-type: none"> <li>• Frequent</li> </ul>

**Tabel 5 : Skenario use case Proses Unkripto dan Unstego**

<i>Use-case :</i>	<ul style="list-style-type: none"> <li>• Proses Unkripto dan Unstego</li> </ul>
<i>Primary actor :</i>	<ul style="list-style-type: none"> <li>• Penerima</li> </ul>
<i>Goal :</i>	<ul style="list-style-type: none"> <li>• Mengembalikan <i>file</i> yang tersembunyi</li> </ul>
<i>Precondition :</i>	<ul style="list-style-type: none"> <li>• Memasukkan nama <i>file</i> hasil stego untuk dikembalikan seperti semula</li> </ul>

<ul style="list-style-type: none"> <li>Memasukkan kata sandi/kata kunci</li> </ul>
<i>Trigger :</i> <ul style="list-style-type: none"> <li>Penerima ingin mengembalikan <i>file</i> yang disembunyikan</li> </ul>
<i>Skenario:</i> <ol style="list-style-type: none"> <li>Penerima membuka aplikasi Steganographer.</li> <li>Penerima memasukkan nama <i>file</i> yang disembunyikan.</li> <li>Penerima memberikan kata kunci sebagai pelengkap sandi.</li> <li>Penerima menekan tombol Unstego untuk memprosesnya.</li> </ol>
<i>Alternate Flow :</i> <ul style="list-style-type: none"> <li>Penerima keluar dari aplikasi</li> </ul>
<i>Priority :</i> <ul style="list-style-type: none"> <li><i>Moderate priority</i></li> </ul>
<i>Frequency of use :</i> <ul style="list-style-type: none"> <li><i>Frequent</i></li> </ul>

Berikut ini merupakan model fungsionalitas sistem yang digambarkan secara utuh melalui *use case* dilihat dari pengguna yang berada di luar sistem (*actor*). Dalam *use case* dibawah ini dijelaskan adanya hubungan antara *actor*, relasi, dan spesialisasi yang terjadi pada aplikasi penyembunyian pesan.

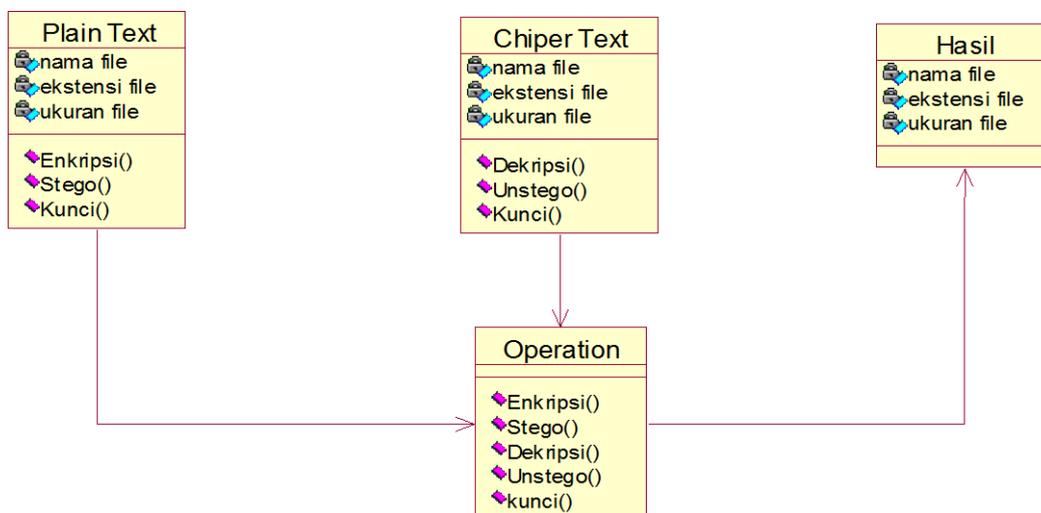


Gambar 15 : Use Case

Pada *use case* diatas digambarkan bahwa pengirim harus memasukkan *file* akan diproses melalui metode kriptografi dan steganografi di amana proes ini dikenal dengan pemasukan *file* pesan. Langkah kedua yaitu memasukkan *file* induk, sehingga *file* pesan tadi dapat disisipkan. Selanjutnya pengirim harus memasukkan kunci/sandi, sehingga pesan tersebut aman dari orang-orang yang tidak berkepentingan. Selanjutnya sistem akan memproses kedua *file* tersebut melalui proses kriptografi dan steganografi. Pada bagian penerima, *actor* melakukan proses pemisahan *file* dengan cara memasukkan *file* yang akan dipisahkan dan memasukkan kunci/sandi. Kemudian sistem akan melakukan pemisahan melalui proses kriptografi dan steganografi.

#### 4.3.2 Class Diagram

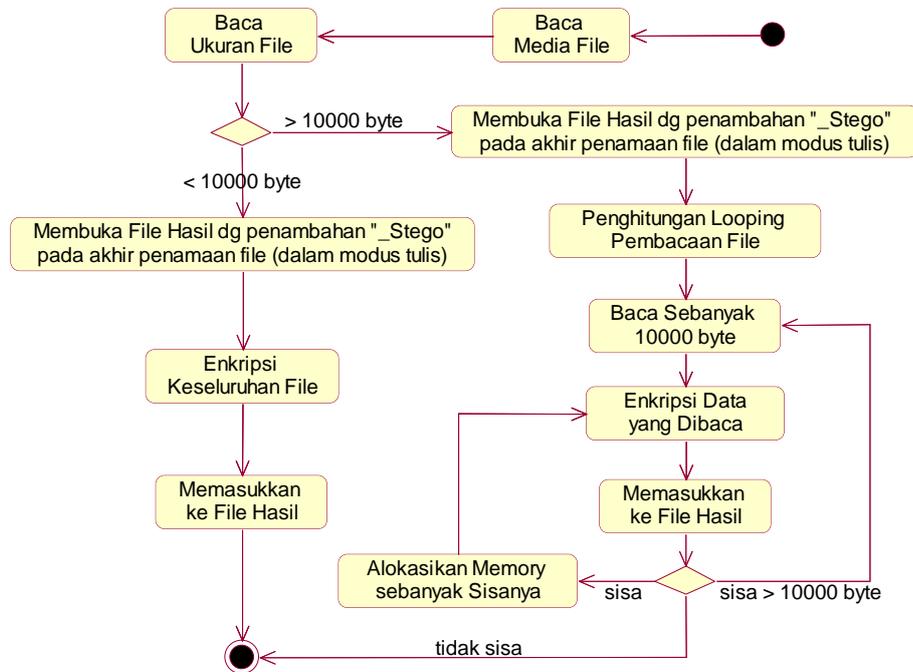
Pada *class diagram*, penulis menggunakan 4 macam kelas yaitu kelas *plain text*, *chiper text*, *operation* dan hasil. Kelas-kelas tersebut saling berhubungan dan mempunyai keterkaitan. Di bawah ini merupakan gambar *class diagram* yang penulis maksud:



**Gambar 16 : Class Diagram**

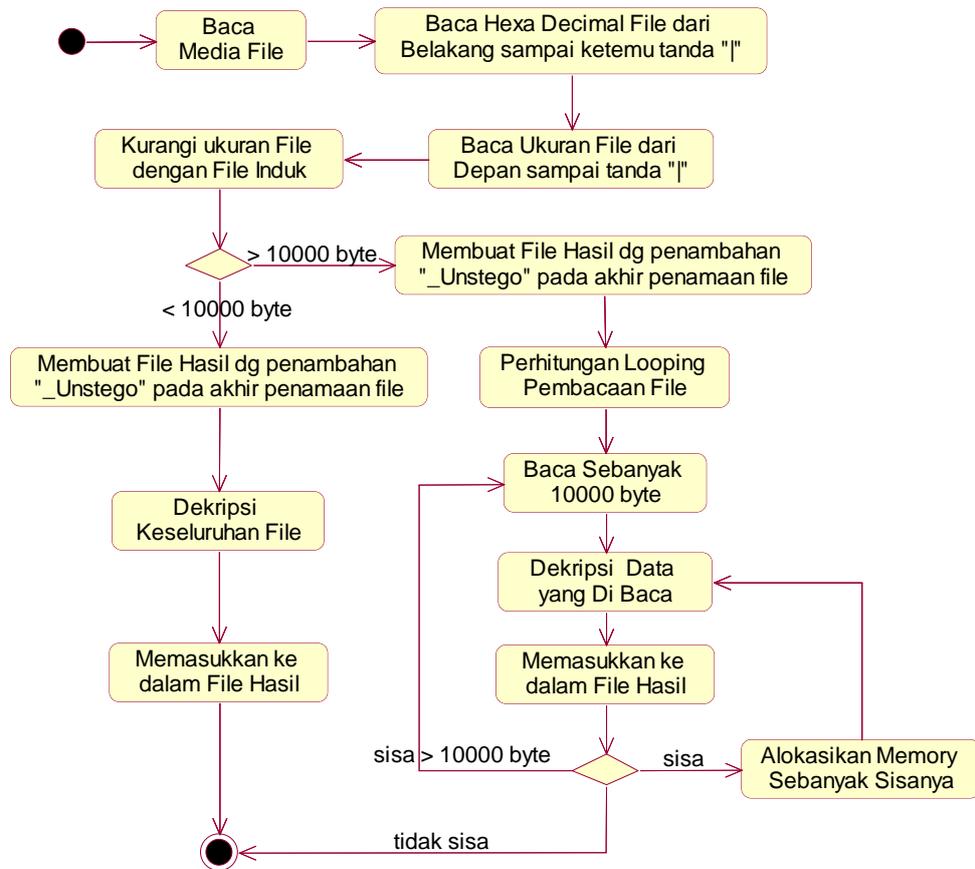
### 4.3.3 Activity Diagram

*Activity Diagram* yang penulis buat saat ini menggunakan 2 macam model diagram yaitu diagram saat krypto dan stego serta diagram pada saat *unkripto* dan *unstego*.

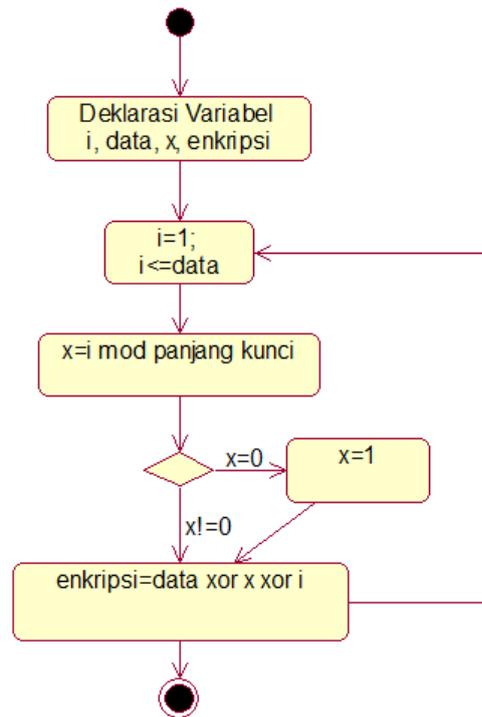


**Gambar 17 : Activity Diagram Kripto dan Stego**

Gambar 4. 5 menunjukkan alur yang berjalan pada saat *file* yang akan di proses dengan *file* induknya. Pada gambar selanjutnya penulis membuat alur yang berjalan pada saat *file* hasil gabungan di pecah kembali menjadi *file* asli.



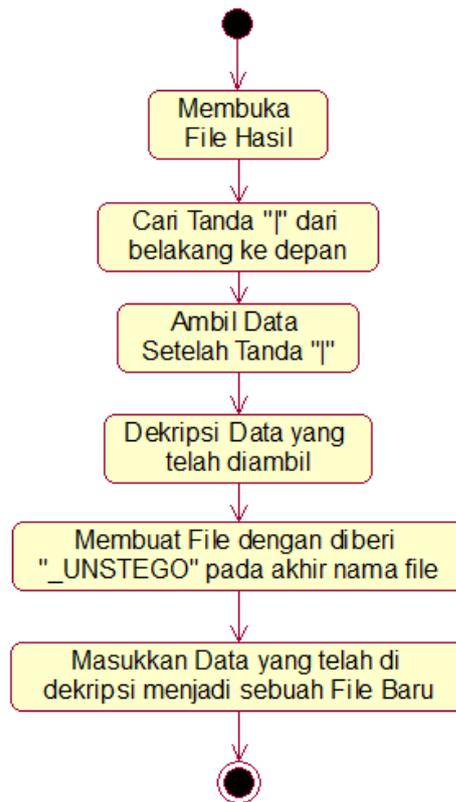
**Gambar 18 : Activity Diagram Unkripto dan Unstego**



**Gambar 19 : Activity Diagram Enkripsi dan Dekripsi**



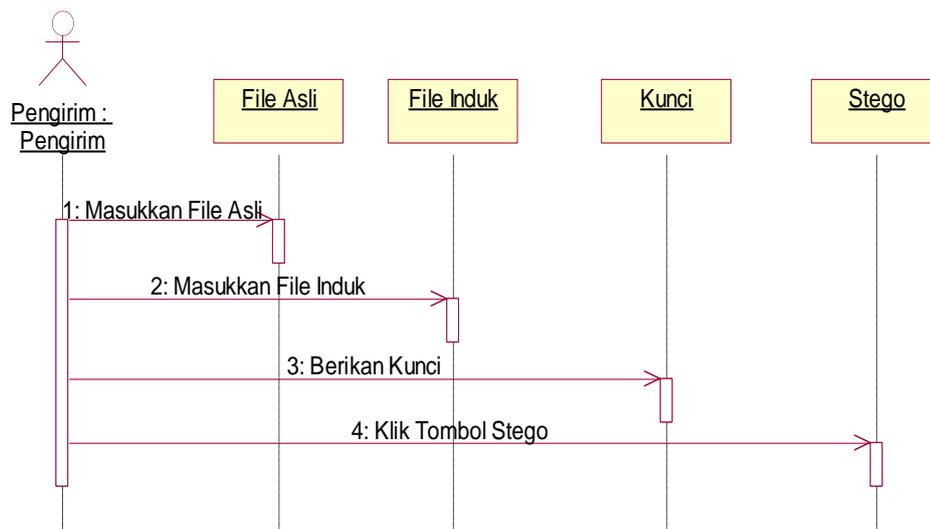
**Gambar 20 : Activity Diagram Stego pada File**



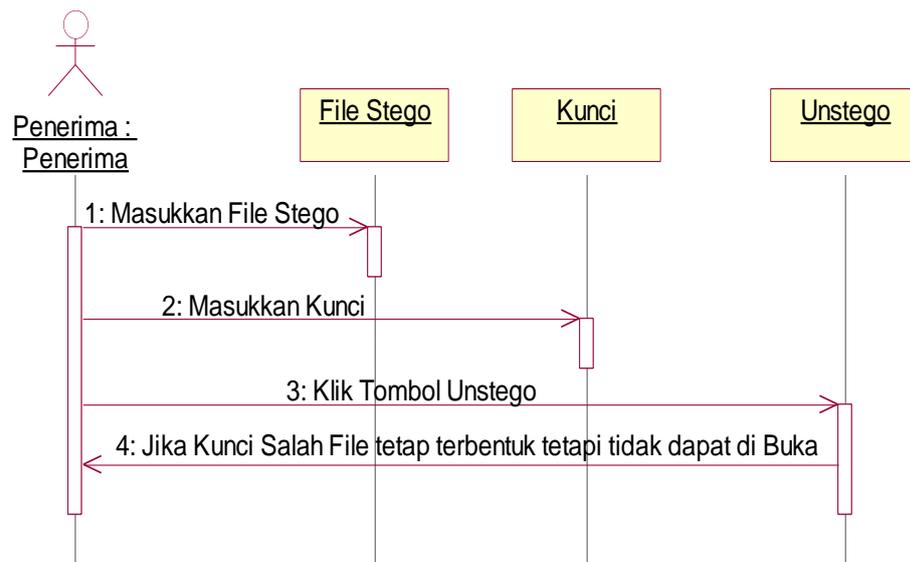
**Gambar 21 : Activity Diagram Unstego pada File**

#### 4.3.4 Sequence Diagram

Penulis juga membuatkan model *sequence* untuk memperjelas dari model-model sebelumnya. Dalam *sequence diagram* juga terdapat 2 bentuk model yang penulis buat yaitu diagram pada saat *Kripto Stego* dan *Unkripto Unstego*. Di bawah ini merupakan model-model yang telah penulis buat supaya memudahkan pengguna dalam menggunakan program yang nantinya akan dibuat.



**Gambar 22 : Sequence Diagram Kripto dan Stego**



**Gambar 23 : Sequence Diagram Unkripto dan Unstego**

#### 4.4 Desain Input Output (I/O)

Masukkan/input merupakan awal dimulainya proses informasi. Bahan mentah dari informasi adalah data yang terjadi dari transaksi yang dilakukan melalui sebuah aplikasi yang dibuat. Data hasil dari transaksi merupakan dasar untuk memasukkan suatu sistem informasi. Dokumen dasar biasanya berbentuk formulir yang digunakan untuk menangkap (*Capture*) data yang terjadi.

Desain *input* merupakan produk dari aplikasi yang dapat dilihat dan digunakan untuk memasukkan data-data yang akan diolah dan nantinya akan menghasilkan informasi-informasi yang diperlukan maupun menghasilkan data-data untuk diolah menjadi informasi bentuk yang lain, sedangkan desain *output* merupakan hasil pengolahan data yang telah diinput.

Dalam aplikasi ini, akan digunakan *software* Visual Basic 6. 0. Berikut ini merupakan desain input output yang telah di buat adalah sebagai berikut :



**Gambar 24 : Storyboard Tampilan Awal**

Pada gambar *Storyboard* di atas, terdapat garis silang yang mewakili gambar *background*. Sedangkan, menu utama dan submenu dalam gambar 4. 12 dapat dijabarkan sebagai berikut:

a. Menu

Pada Menu, terdapat 2 buah sub menu yaitu *Stego* dan *Unstego*.

b. Submenu *Stego*

Submenu *Stego* merupakan submenu yang digunakan untuk menyembunyikan *file*.

c. Submenu *Unstego*

Submenu *Unstego* merupakan submenu yang digunakan untuk mengembalikan *file* yang telah disembunyikan sebelumnya.

d. Submenu *About Us*

Submenu *About Us* merupakan submenu yang berisi tentang data diri pembuat program.

e. Submenu Keluar

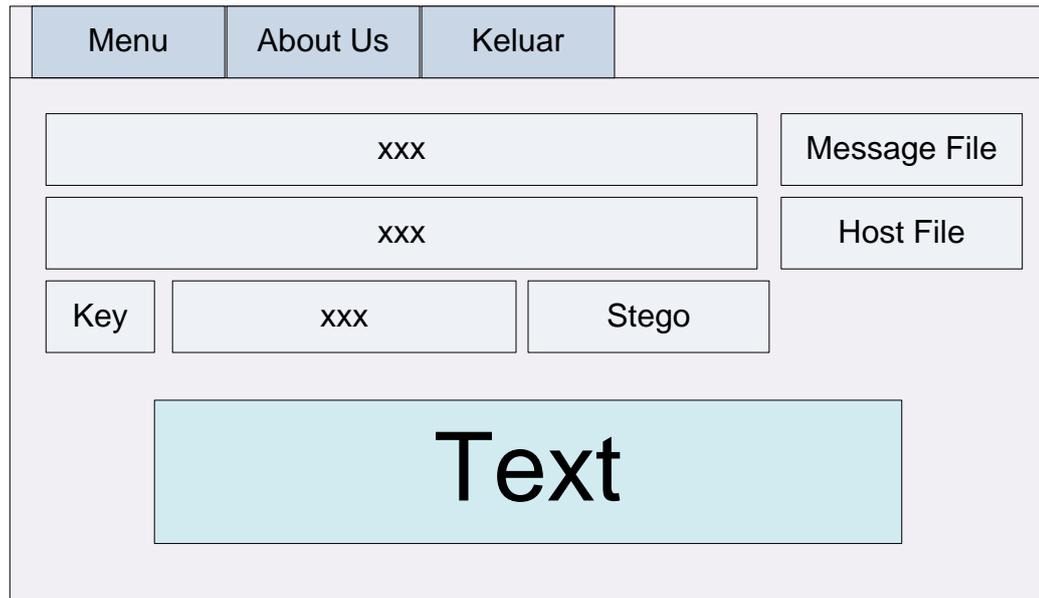
Submenu Keluar merupakan submenu yang digunakan untuk keluar dari aplikasi.

#### 4.4.1 Desain *Input Output (I/O)* Submenu *Stego*



**Gambar 25 : Storyboard Menu Utama**

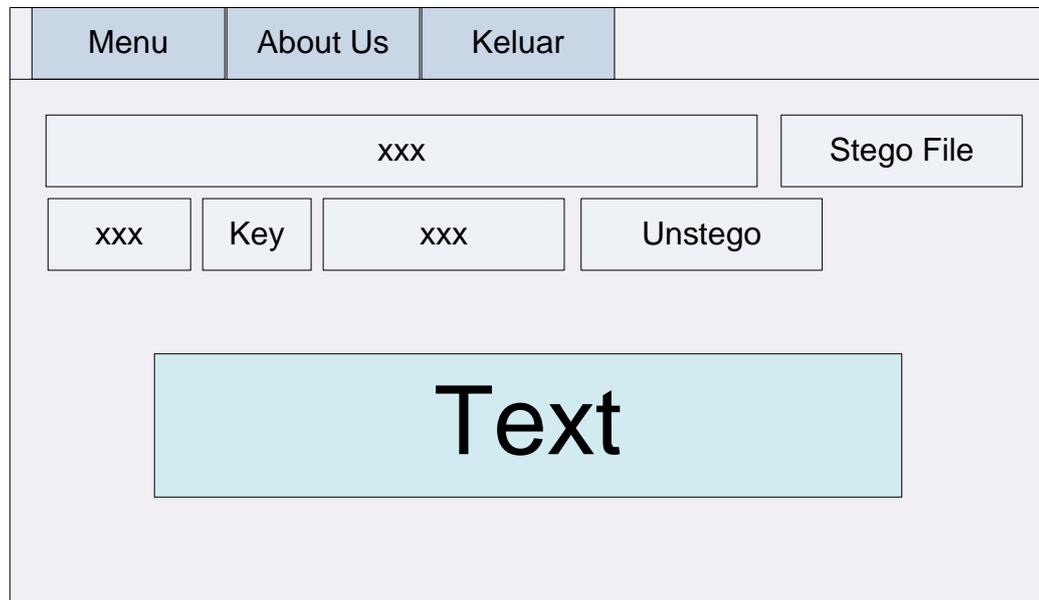
Pada menu utama, pada pilihan *Menu*, pada saat diklik akan muncul submenu *Stego dan Unstego*. Pada submenu *Stego*, apabila diklik akan muncul tampilan seperti berikut ini.



**Gambar 26 : Storyboard Submenu Stego**

Pada submenu *stego* terdapat 3 (tiga) buah *text box* yaitu 2 *text box* untuk memasukkan *file* dan 1 (satu) *text box* untuk kunci, 3 (tiga) *command* yaitu 2 (dua) *command* untuk mengambil *file* dan 1 (satu) *command* untuk proses *stego* serta 1 (satu) label untuk kunci.

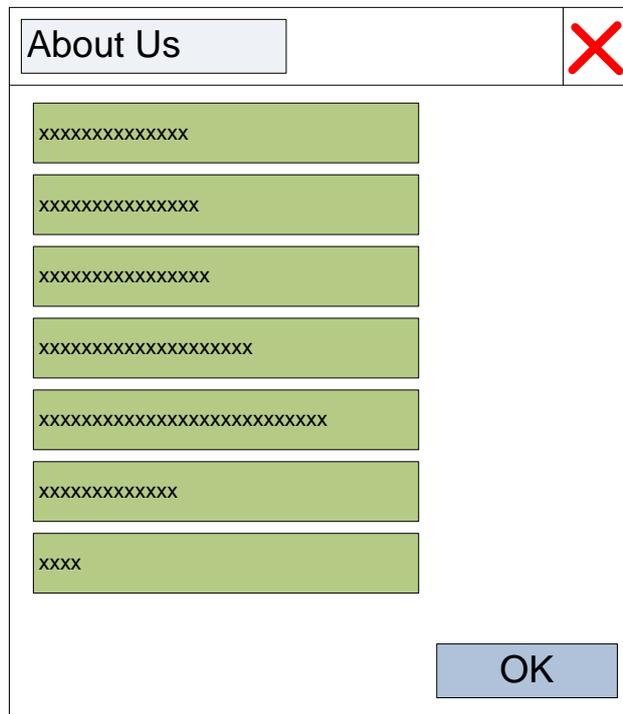
#### 4.4.2 Desain *Input Output (I/O)* Submenu *Unstego*



**Gambar 27 : Storyboard Submenu *Unstego***

Pada submenu *unstego* terdapat 3 (tiga) *text box* yaitu untuk memasukkan *file*, memasukkan kata kunci dan untuk memberikan ekstensi *file* yang nantinya akan terbentuk. Selain itu terdapat 2 (dua) *command* yaitu untuk mengambil *file* dan proses *unstego*. Kemudian terdapat 1 (satu) label untuk kunci.

#### 4.4.3 Desain *Input Output (I/O)* Submenu *About Us*

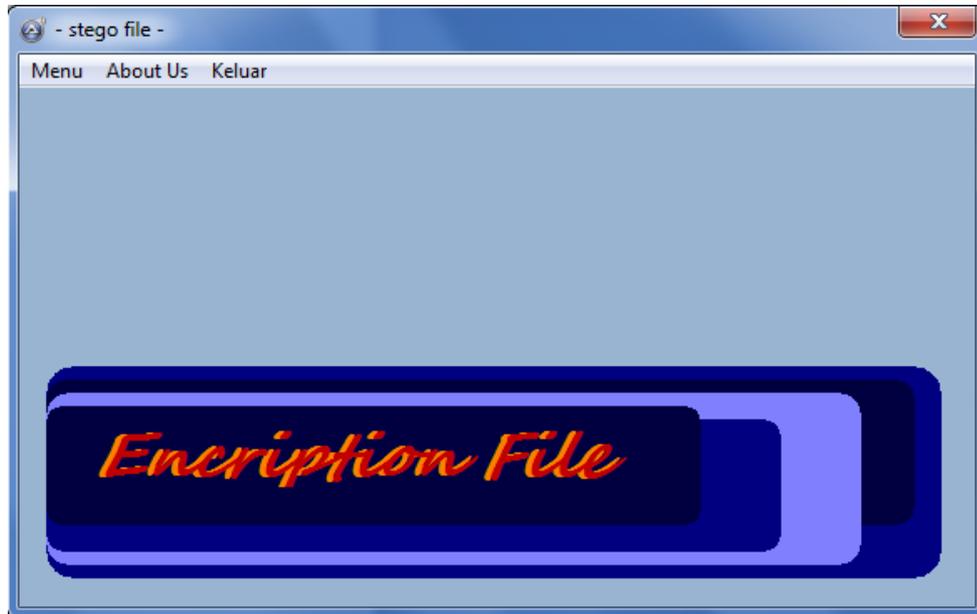


**Gambar 28 :** *Storyboard* Submenu *About Us*

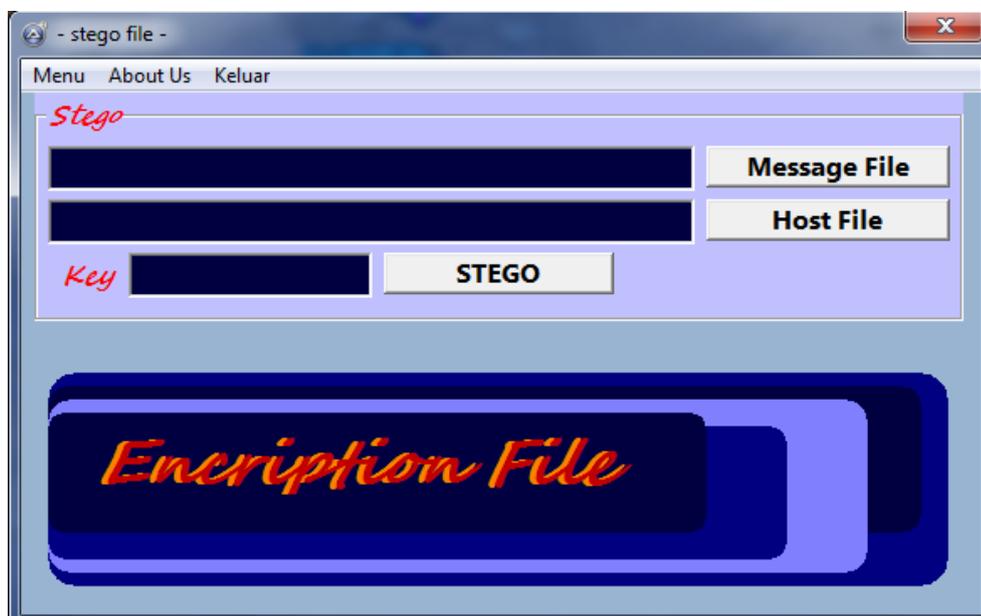
Pada submenu *about us* tanda “X” yang terletak pada bagian kanan atas adalah tombol untuk keluar dari submenu, garis silang besar yang ada di tengah merupakan isi dari biodata pembuat program dan tombol “OK” pada kanan bawah merupakan tombol untuk mengakhiri submenu.

#### 4.5 Implementasi

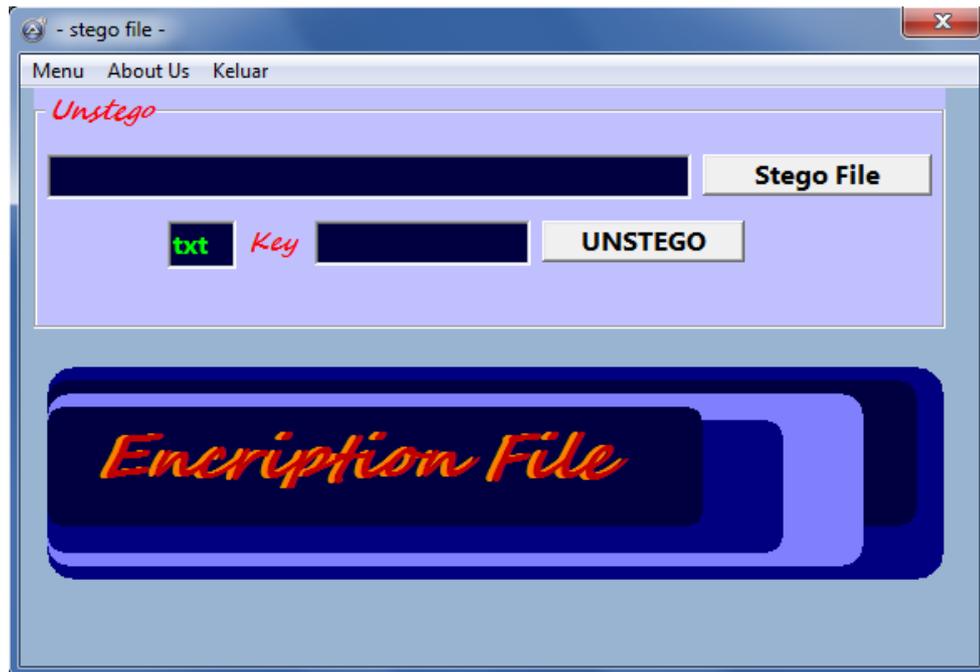
Program *Kripto* dan *Stego* ini dimulai dengan menekan tombol “Menu” yang ada pada sisi kiri atas aplikasi dilanjutkan memilih “Stego” terlebih dahulu.



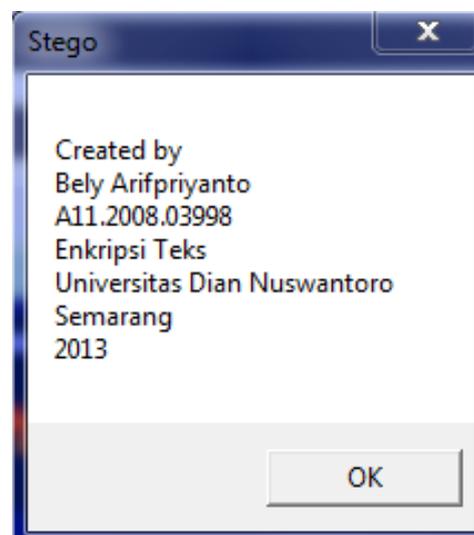
Gambar 29 : Tampilan Awal Program



Gambar 30 : Tampilan Menu Stego



Gambar 31 : Tampilan Menu *Unstego*

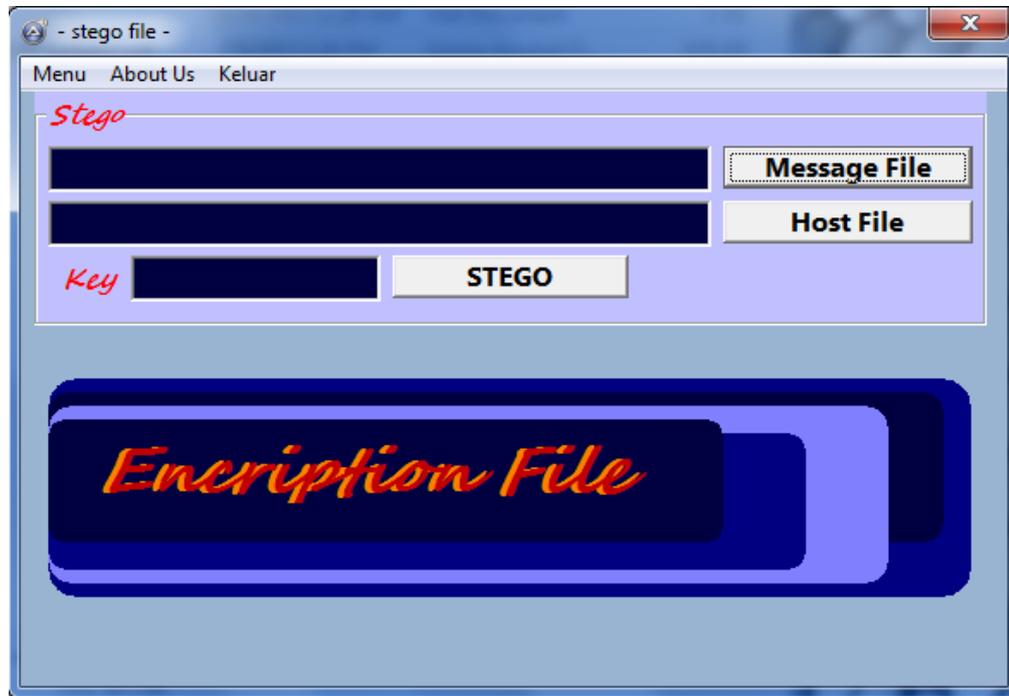


Gambar 32 : Tampilan *Message Box Menu About Us*

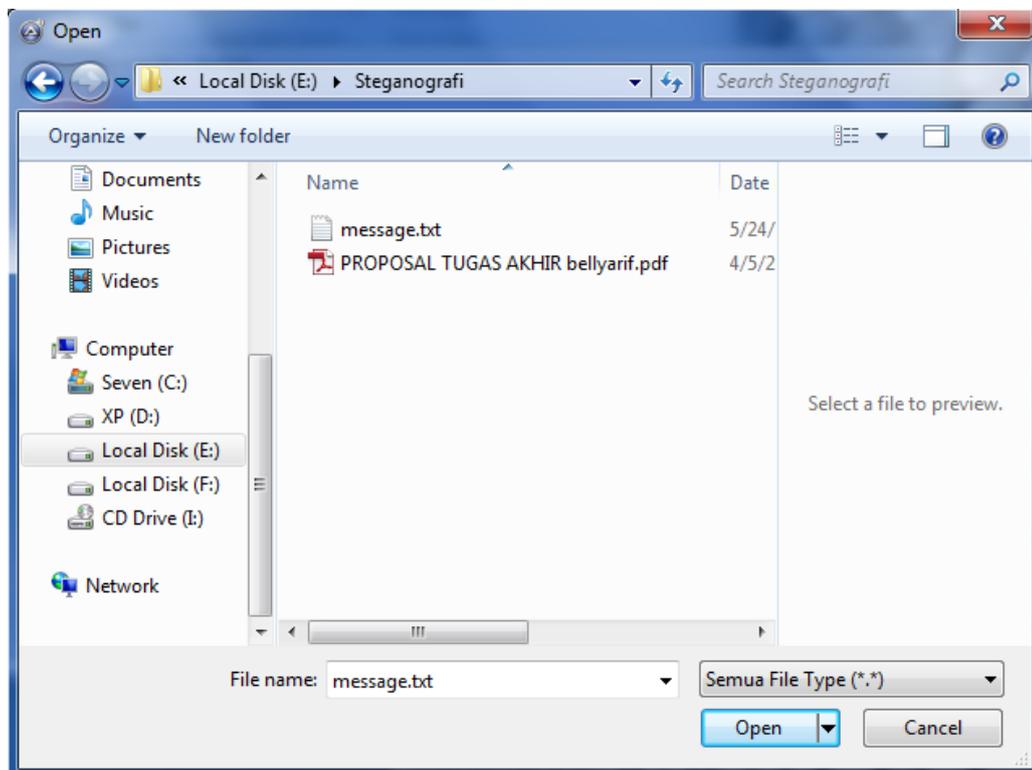
#### 4.6 Analisa Percobaan

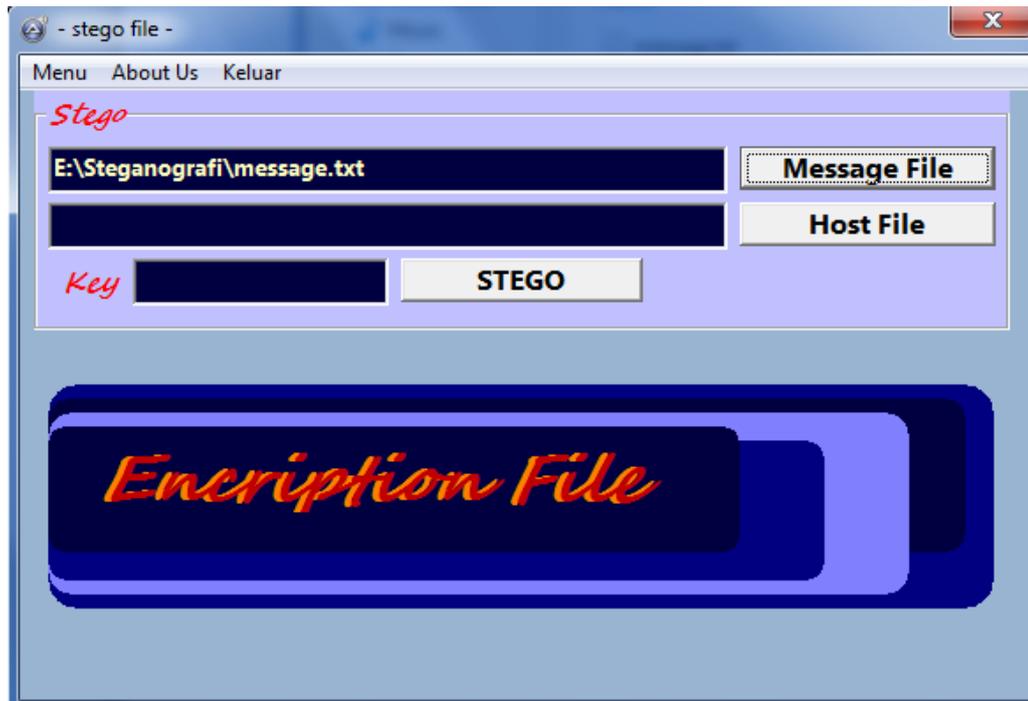
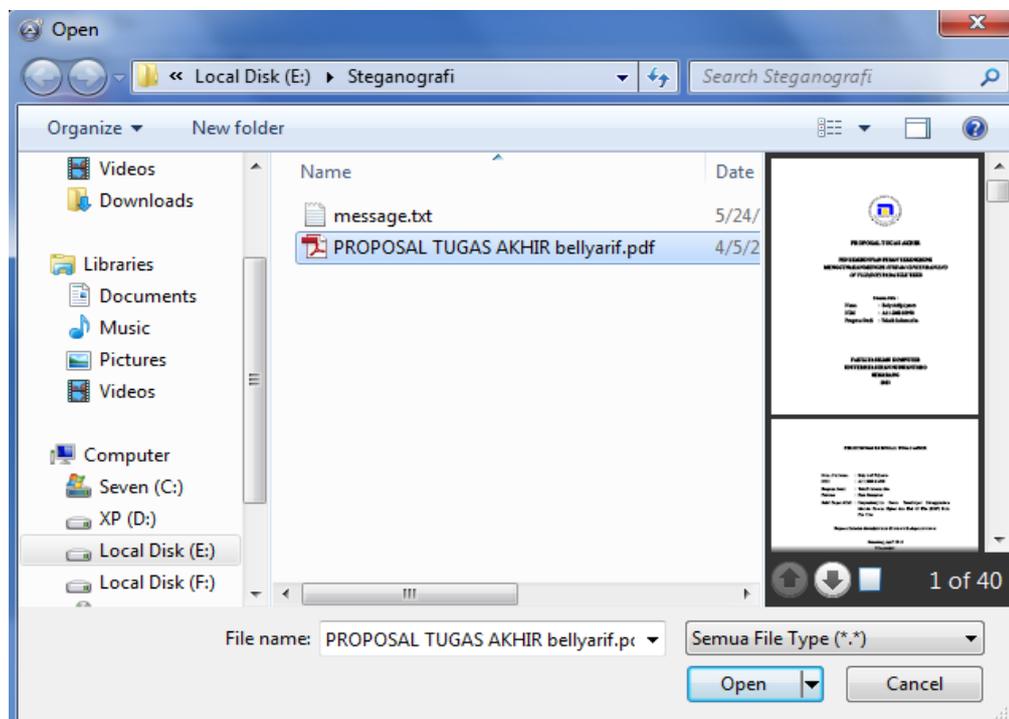
Setelah dilakukan implementasi pada aplikasi, maka hasil yang diperoleh akan diolah dan dianalisis sedemikian rupa untuk mendapatkan kesimpulan yang

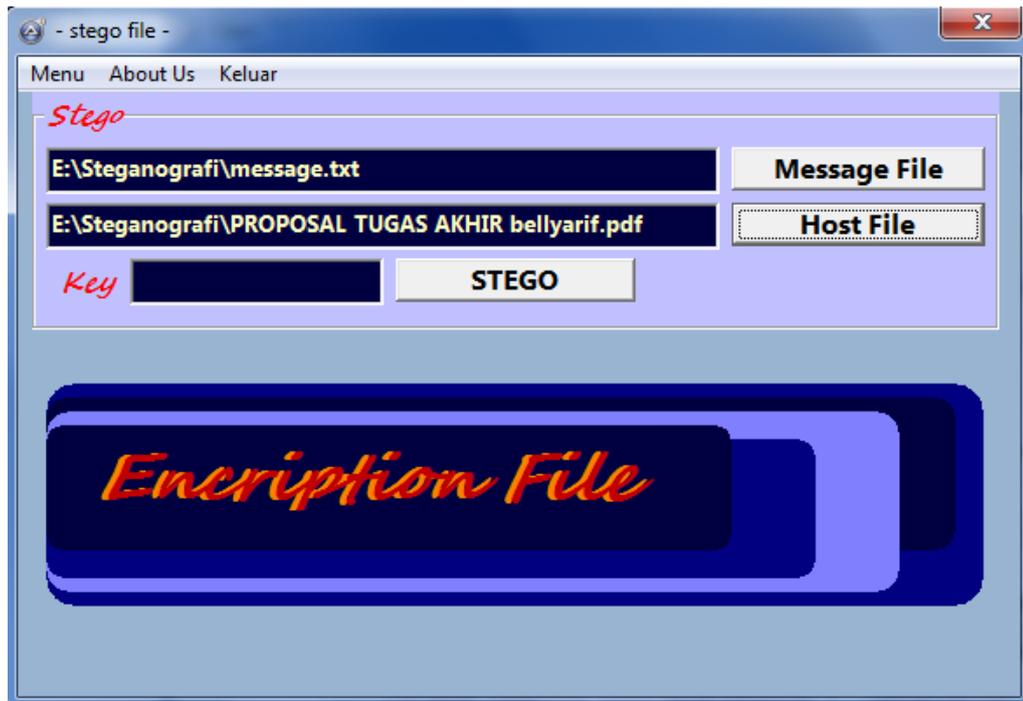
padat dan jelas. Berikut ini merupakan tampilan aplikasi penyisipan pesan secara berurutan:



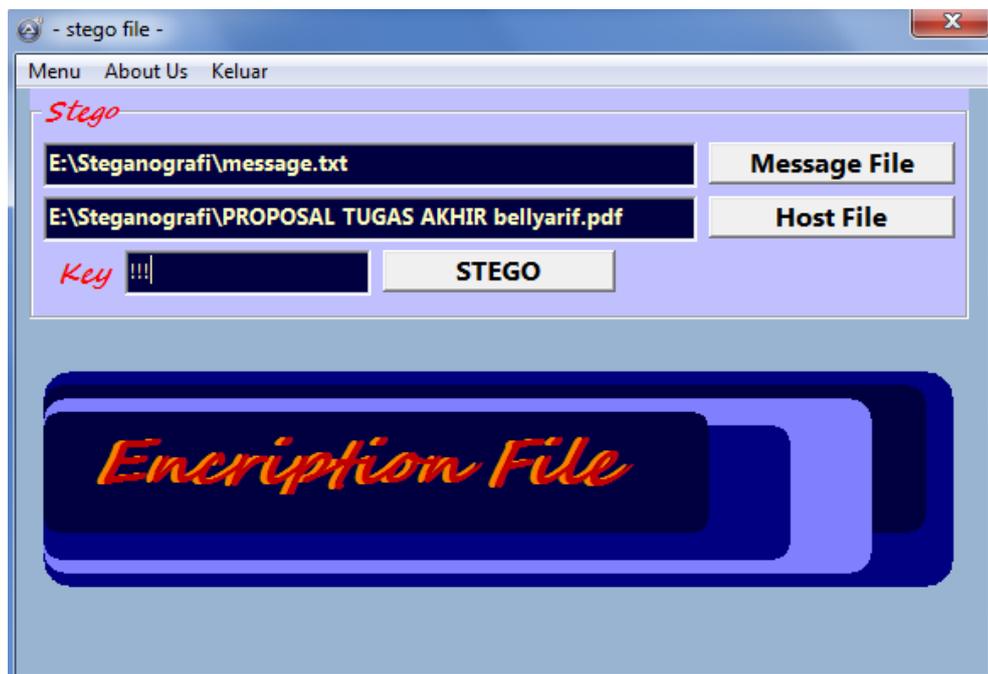
Gambar 33 : Tampilan Awal Proses Stego



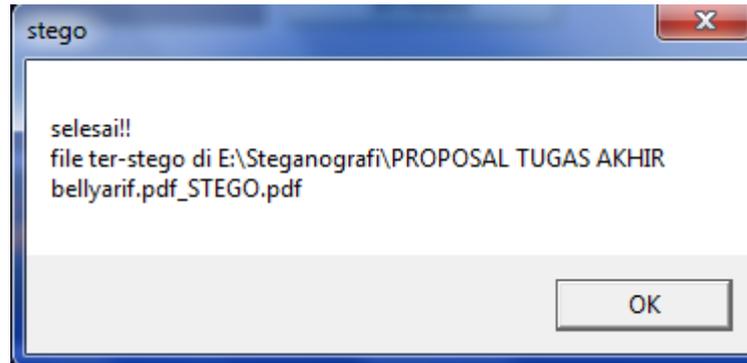
Gambar 34 : Memilih *Message File*Gambar 35 : *Message File* yang telah dipilihGambar 36 : Memilih *Host File*



Gambar 37 : *Host File* yang telah dipilih

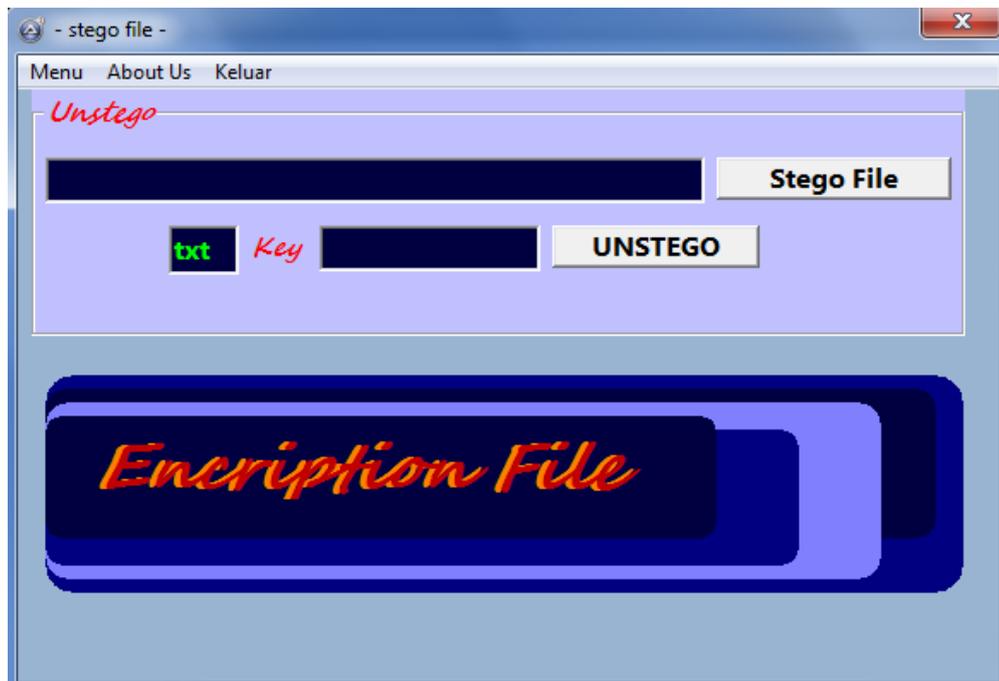


Gambar 38 : Memasukkan Kunci (*Key*)



**Gambar 39 : Pop Up Window Akhir Proses Penyisipan Pesan**

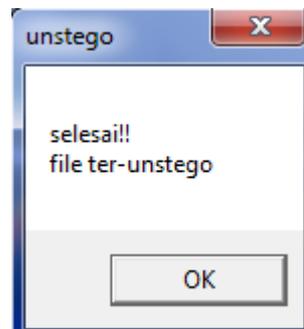
Pada akhir proses penyisipan pesan, akan muncul peringatan seperti pada gambar 4. 26. Setelah, proses penyisipan selsesai, kemudian berikut ini merupakan proses ekstraksi:



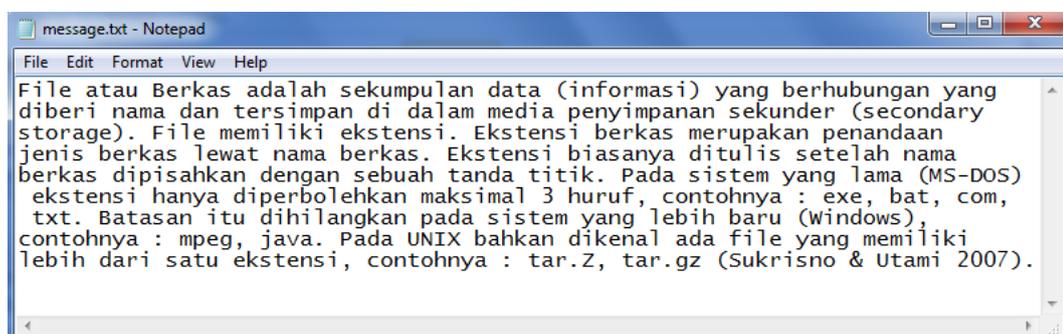
**Gambar 40 : Tampilan Awal Proses Unstego**



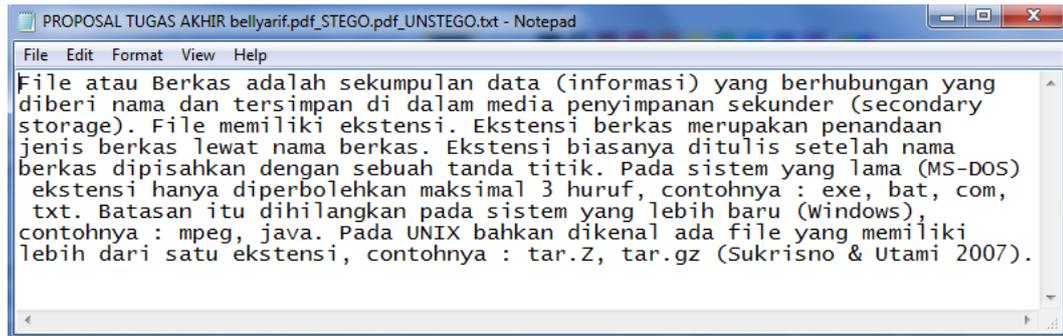
**Gambar 41 : Memasukkan *File Stego* dan Kunci (*Key*)**



**Gambar 42 : *Pop Up Window* Akhir Proses Ekstraksi**



**Gambar 43 : *File Pesan***



**Gambar 44 : File Hasil Proses Unstego**

#### 4.7 Pengujian Program (Testing)

Pada pengujian ini, apabila ukuran *file* pesan melebihi ukuran *file* induk, maka *file* induk akan pecah dan mengakibatkan *file* induk dapat terbaca oleh orang lain. Sedangkan apabila *file* induk diserang dengan mengubah format maka pesan akan hancur, tetapi apabila masih dengan format yang sama maka pesan masih dapat terbaca.

##### 4.7.1 Black Box Testing

Berikut ini merupakan beberapa aspek yang akan dikaji:

**Tabel 6 : Daftar Aspek Pengujian**

No.	Aspek Pengujian
1.	Apakah semua menu dapat dituju / diklik/ disorot secara tepat oleh <i>pointer mouse</i> ?
2.	Apakah setiap operasi <i>mouse</i> dikenali dengan baik oleh aplikasi yang akan meresponnya?

Berdasarkan hasil pengujian, dapat diperoleh kesimpulan bahwa performansi program cukup baik. Semua rancangan program telah tersusun dalam menu dengan tepat dan setiap kontrol yang terdapat dalam tiap – tiap menu juga dapat diakses secara tepat. *Mouse* dengan mudah dapat mengakses tiap menu dalam *program* secara tepat pula memberikan respon sesuai dengan konteks interaktifnya.

Tabel 7 : Ringkasan Hasil Pengujian

<i>Faktor Pengujian</i>	<i>Status Output</i>
<i>Tombol Submenu Stego</i>	✓
<i>Tombol Submenu Unstego</i>	✓
<i>Tombol Submenu About Us</i>	✓
<i>Penekanan Tombol</i>	✓

#### 4.7.2 White Box Testing

```
Private Sub Command3_Click() 'fungsi stego!!
```

```
Dim data As String .....(1)
```

```
Dim x As Long
```

```
Dim y As Long
```

```
Dim z As Long
```

```
Dim jpg As Long
```

```
Dim ext As String
```

```
Dim encjpg As String
```

```
encjpg = FileLen(Text2. Text) ..... (2)
```

```
ext = Mid(StrReverse(Text2. Text), 1, 4)
```

```
ext = StrReverse(ext)
```

```
jpg = FileLen(Text1. Text)
```

```
FileCopy Text2. Text, Text2. Text & "_STEGO" & ext ..... (3)
```

```
x = FileLen(Text1. Text) Mod 10000 ..... (4)
```

```
y = FileLen(Text1. Text) - x
```

```
Open Text1. Text For Binary Access Read As #1
```

```
Open Text2. Text & "_STEGO" & ext For Binary Access Write As #2
```

```
..... (5)
```

```
Put #2, FileLen(Text2. Text) + 1, ""
```

```
If jpg >= 10000 Then ..... (6)
```

```
For z = 1 To y Step 10000..... (7)
```

```
data = Space$(10000) ..... (8)
```

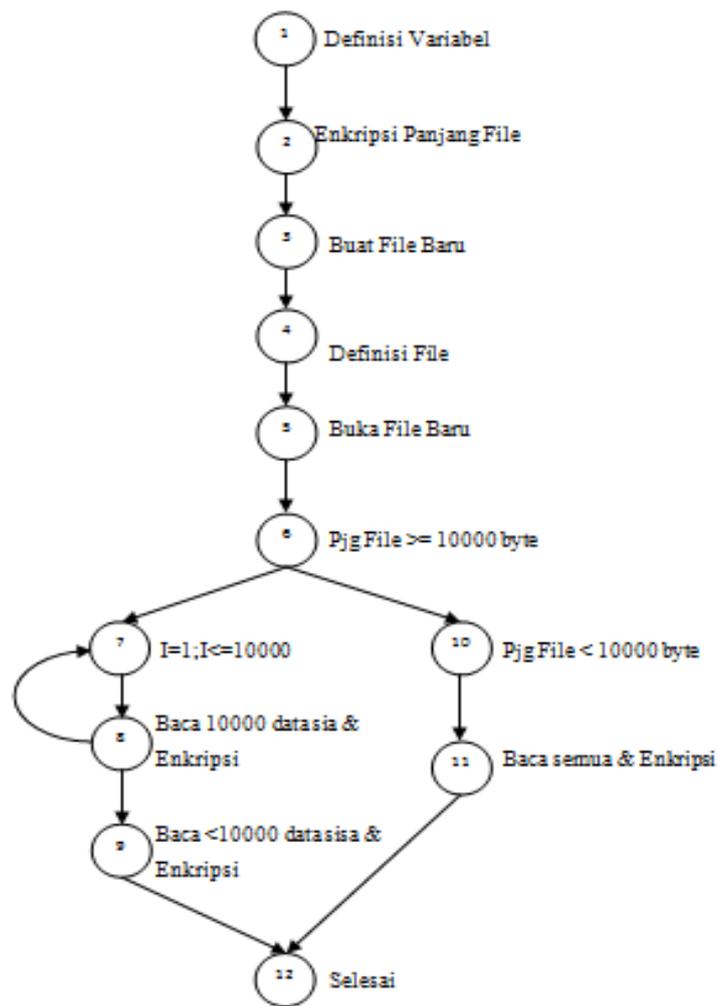
```
Get #1, z, data
```

```

Put #2, , encrypt(data, Text5. Text)
Next
y = x
data = Space$(y) .....(9)
Get #1, , data
Put #2, , encrypt(data, Text5. Text)
Put #2, , "|" & encrypt(encjpg, Text5. Text)
Else ..... (10)
data = Space$(jpg) .....(11)
Get #1, 1, data
Put #2, , encrypt(data, Text5. Text)
Put #2, , "|" & encrypt(encjpg, Text5. Text)
End If
Close #2
Close #1
MsgBox "selesai!!" & vbCrLf & "file ter-stego di " & Text2. Text &
"_STEGO" & ext, vbOKOnly, "stego" ..... (12)
End Sub

```

Dari penggalan *code program Kripto Stego* di atas didapatkan *graph* tertutup seperti gambar 4. 21 seperti dibawah ini.



**Gambar 45 : Graph Kripto Stego**

Dari *graph* tersebut di atas kemudian dapat dihitung *Cyclometric Complexity*-nya untuk menentukan berapa banyak jumlah *independent path* yang harus kita uji dari modul tersebut. *Cyclometric Complexity* dapat dihitung dengan rumus sebagai berikut:

- $V(G) = \text{jumlah edge} - \text{jumlah node} + 2$
- $V(G) = \text{jumlah kondisi} + 1$ , atau
- $V(G) = R_i$

Sehingga dengan menggunakan dua rumus di atas dapat dihitung  $V(G)$  dari modul tersebut sebagai berikut :

$$V(G)_a = \text{jumlah edge} - \text{jumlah node} + 2$$

$$= 13 - 12 + 2$$

$$= 3$$

$$V(G)b = \text{jumlah kondisi} + 1$$

$$= 2 + 1$$

$$= 3$$

Dari perbandingan di atas terbukti bahwa  $V(G)a = V(G)b$  yang membuktikan bahwa *program krypto stego* berjalan dengan baik pada aplikasi yang telah diuji cobakan.

#### 4.8 Kuisisioner

Pengujian selanjutnya dilakukan untuk mengetahui respon dari *user* terkait dengan *program* yang telah penulis buat sebelumnya dan telah diujicobakan. *User* yang menjadi responden adalah mahasiswa dan dosen pada Universitas Dian Nuswantoro Semarang. Kepada masing-masing responden peneliti membagikan kuisisioner yang berbentuk isian pilihan penilaian terhadap *software krypto* dan *stego* yang sebelumnya telah diujicobakan.

**Tabel 8 : Tabel kuisisioner**

No	Uraian Pertanyaan	Nilai				
		1	2	3	4	5
<b>A</b>	<b>Kemampuan Software</b>					
1	Apakah <i>file</i> dapat tersembunyi dengan baik ?					
2	Bagaimana kinerja <i>program</i> aplikasi ?					
3	Apakah semua <i>file</i> dapat di proses dengan baik ?					
4	Apakah <i>program</i> dapat mengembalikan <i>file</i> secara utuh ?					
<b>B</b>	<b>Interaksi Manusia dan Komputer</b>					
1	Apakah pengguna dapat menggunakan dengan baik ?					
2	Bagaimana bentuk desain <i>program</i> ?					
3	Apakah <i>program</i> dapat berjalan dengan lancar ?					
4	Bagaimana tampilan menu aplikasi ?					

Berikut ini adalah rincian dari hasil responden terhadap pengujian *program*.

**Tabel 9 : Tabel Hasil Pengujian**

No	Pertanyaan A				Pertanyaan B			
	1	2	3	4	1	2	3	4
1	5	4	5	5	3	5	5	4
2	5	5	4	5	5	5	5	4
3	5	4	5	5	4	5	5	4
4	5	5	5	5	4	4	5	4
5	5	5	5	5	5	4	5	4
6	5	4	5	4	4	5	5	5
7	5	5	5	4	5	5	5	5
8	5	5	4	5	5	5	5	5
9	5	5	5	5	4	5	5	5
10	5	5	5	5	5	4	5	5
11	5	4	4	5	5	5	5	5
12	5	5	5	5	4	4	5	5
13	5	5	5	5	4	4	5	4
14	5	5	5	4	5	5	5	5
15	5	5	5	5	4	4	5	4
16	5	4	5	5	5	4	5	4
17	5	5	5	4	5	5	5	5
18	5	5	5	5	4	5	5	4
19	5	5	5	4	5	5	5	5
Skala Kumulatif Rata-rata ( $x/19$ )	5	4. 73	4. 84	4. 73	4. 73	4. 63	5	4. 52

Kuisisioner ini dibuat untuk menguji hasil dari kekuatan program. Dalam hal ini, penulis membuat dua buah kategori yaitu kategori pertama tentang ketahanan atau kekuatan *software*, kategori yang kedua adalah interaksi manusia dan komputer yang maksudnya adalah bahwa apakah seorang *user* yang awam terhadap program dapat menggunakan program tersebut atau tidak. Dalam segi

penilaian, semakin besar angkanya maka semakin baik kualitas aplikasi dan semakin familiar dengan manusia. Adapun batasan penilaian berkisar dari angka 1 yang berarti sangat buruk, sampai dengan angka 5 yang berarti sangat baik.

Dari hasil pengujian kuisisioner tersebut, penulis dapat menyimpulkan bahwa dalam kategori pertama dan kategori kedua mempunyai skala kumulatif melebihi angka 4 semuanya. Hal ini membuktikan bahwa untuk program atau aplikasi dapat berjalan dengan baik. Selain itu, aplikasi juga dapat menyembunyikan pesan dan tidak tampak secara kasat mata. Pada kategori kedua, juga membuktikan bahwa aplikasi tersebut juga dapat dipahami oleh orang awam atau orang yang sebelumnya tidak mengetahui aplikasi tersebut.