BAB 4
IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

Implementasi sistem merupakan tahap dimana sistem dimplementasikan dan siap untuk digunakan sesuai dengan perancangan yang telah dibuat. Implementasi sistem dilakukan pada tiga tahap yaitu implementasi perangkat keras, implementasi perangkat lunak, dan implementasi antarmuka.

4.1.1 Implementasi Perangkat Keras

Implementasi perangkat keras merupakan relisasi dari perancangan kebutuhan perangkat keras. Aplikasi steganografi ini d2mplementasikan pada perangkat keras dengan spesifikasi seperti pada tabel 4.1.

Tabel 4. 1 Spesifikasi Perangkat Keras

Perangkat Keras	Spesifikasi
CPU	Intel(R) core(TM) 2 duo CPU T6570 2.10GHz
RAM	2048 Mb
Harddisk	320 Gb
Resolusi Layar	1366 x 768
VGA	nVIDIA GeForce G 310M 512MB

4.1.2 Implementasi Perangkat Lunak

Implementasi perangkat lunak merupakan proses instalasi perangkat lunak, sehingga dapat beroperasi dengan benar. Perangkat lunak yang digunakan yaitu seperti pada tabel 4.2.

Tabel 4. 2 Spesifikasi Perangkat Lunak

Perangkat Lunak	Keterangan
Sistem Operasi	Windows 8.1 Professional
Bahasa Pemrograman	Java
Tools Pemrograman	Java NetBeans IDE 7.1
Perangkat Pendukung	JDK 7
Picture Viewer	Windows Photo Viewer
Audio Player	Aimp 3

Document Viewer	Notepad, MS Word 2013, dan Foxit Reader
Picture Editor	Adobe Photoshop CS6

4.1.3 Implementasi Antarmuka

Implementasi antarmuka merupakan realisasi desain tampilan antarmuka dari setiap menu yang telah dirancang sebelumnya kedalam bahasa pemrograman.

Berikut ini adalah implementasi antarmuka sistem yang terlihat pada Tabel 4.3.

Tabel 4. 3 Implementasi Antarmuka

Menu	Use Case	File
Home	-	HomeRPP.java
Penyisipan	Pemilihan pesan, pemilihan citra, enkripsi, pembuatan kunci, penyalinan kunci, penyisipan,save file.	HomeRPP.java, Message.java, Image.java, steganoRPP.java, AES.java.
Ekstraksi	Pemilihan citra stego, ekstrak, save file, pemasukan kunci, dekripsi.	HomeRPP.java, Message.java, Image.java, steganoRPP.java, AES.java.

Antarmuka dari aplikasi steganografi ini, terdiri dari tiga menu utama, yaitu menu *home*, menu penyisipan, dan menu ekstraksi.

1. Tampilan Menu *Home*

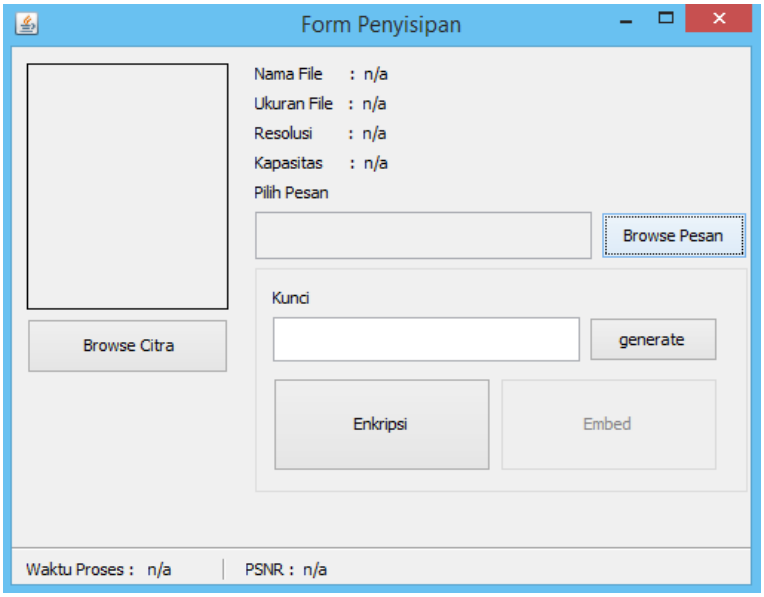
Tampilan menu *home* dari aplikasi steganografi dapat dilihat pada gambar 4.1.



Gambar 4. 1 Tampilan Menu Home

2. Tampilan Menu Penyisipan

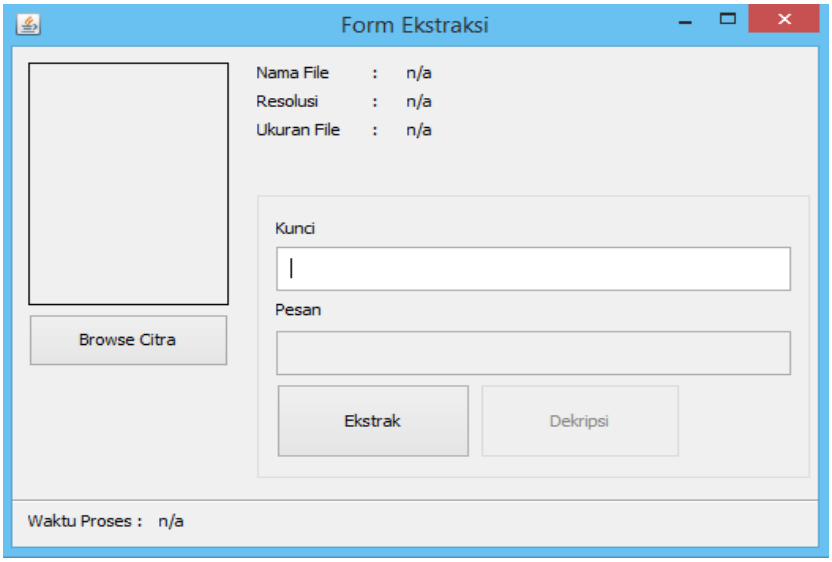
Tampilan menu penyisipan dari aplikasi steganografi dapat dilihat pada gambar 4.2.



Gambar 4. 2 Tampilan Menu Penyisipan

3. Tampilan Menu Ekstraksi

Tampilan menu ekstraksi dari aplikasi steganografi dapat dilihat pada gambar 4.3.



Gambar 4. 3 Tampilan Menu Ekstraksi

4.2 Pengujian Sistem

Tahap selanjutnya setelah mengimplementasikan perancangan kedalam program yaitu pengujian sistem. Pengujian sistem merupakan tahapan dimana sistem diuji untuk mengetahui apakah sistem yang dibangun telah sesuai dengan perancangan yang d2nginkan atau tidak, dan untuk mengetahui apakah tujuan yang d2nginkan telah tercapai atau belum. Rencana pengujian sistem akan dilakukan menggunakan dua metode yaitu pengujian *white box* dan *black box*.

4.2.1 Rencana Pengujian

Rencana pengujian yang akan dilakukan pada aplikasi steganografi ini yaitu seperti pada tabel 4.4.

Tabel 4. 4 Rencana Pengujian Sistem

No	Proses	Jenis Pengujian
1	Enkripsi	*White Box*
2	*Key Schedule*	*White Box*
3	Dekripsi	*White Box*
4	Penyisipan	*Black Box* : pengujian kapasitas, performansi, dan kualitas citra
5	Ekstraksi	*Black Box* : pengujian performansi, dan ketahanan

4.2.2 Pengujian *White Box*

Pengujian *white box* bertujuan untuk mengetahui kinerja logika yang dibuat pada sebuah sistem apakah berjalan dengan baik atau tidak. Pengujian *white box* yang akan dilakukan pada aplikasi yaitu menggunakan *flow graph* yang digunakan untuk menggambarkan alur dari algoritma dan *graph matrix* yang digunakan untuk menggenerasi *flow graph*. Algoritma yang akan diujikan pada pengujian *white box* ini yaitu pengujian enkripsi *AES*, dekripsi *AES*, dan *kunci AES*.

4.2.1.1 Pengujian Enkripsi AES

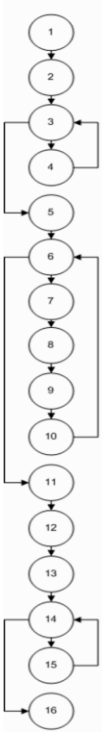
Pengujian enkripsi AES dilakukan pada algoritma enkripsi seperti pada tabel 4.5.

Tabel 4. 5 Algoritma Enkripsi AES

Line	Source
1	Byte[ ] tmp = new byte[in.length];
2	Byte[ ][ ] state = new byte[4][Nb()]; for (int i
3	= 0; i < in.length; i++) { state[i / 4][i % 4] =
4	in[i % 4 * 4 + i / 4];
	}
5	state = addRoundKey(state, w, 0);
6	for (int round = 1; round < Nr(); round++) { state =
7	subBytes(state); state = shiftRows(state); state =
8	mixColumns(state); state = addRoundKey(state, w, round);
9	}
10	state = subBytes(state);
11	state = shiftRows(state);
12	state = addRoundKey(state, w, Nr()); for (int i =
13	0; i < tmp.length; i++) { tmp[i % 4 * 4 + i / 4] =
14	state[i / 4][i % 4];
15	} return
	tmp;
16	

1. *Flow Graph* Enkripsi *AES*

Berdasarkan pada tabel 4.5 maka dapat dibentuk *flow graph* seperti pada gambar 4.4.



Gambar 4. 4 Flow Graph

Enkripsi AES Hitung cyclomatic complexity yaitu sebagai berikut:

Region = 4

V(G) = Edge – Node + 2

= 18 – 16 + 2

= 4

Berdasarkan pada hasil *cyclomatic complexity* maka didapat empat *independent path* yaitu:

1 = 1,2,3,4,3,5,6,7,8,9,10,6,11,12,13,14,15,14,16

2 = 1,2,3,5,6,7,8,9,10,6,11,12,13,14,15,14,16

3 = 1,2,3,5,6,11,12,13,14,15,14,16

4 = 1,2,3,5,6,11,12,13,14,16

2. *Graph Matrix* Enkripsi *AES*

Graph matrix enkripsi *AES* terlihat pada tabel 4.6.

Tabel 4. 6 Graph Matrix Enkripsi AES

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	sum
1		1															0
2			1														0
3				1	1												1
4			1														0
5						1											0
6							1				1						1
7								1									0
8									1								0
9										1							0
10						1											0
11												1					0
12													1				0
13														1			0
14															1	1	1
15														1			0
16																	0
Total																	**3**

V(G) = Jumlah *Graph Matrix* + 1

= 3 + 1 = 4

Berdasarkan pengujian yang dilakukan pada setiap metode, dihasilkan nilai *Cyclomatic Complexity* yang sama yaitu 4, Maka dapat disimpulkan bahwa pengujian *white box* pada proses enkripsi AES berjalan dengan baik, karena setiap pengujian menghasilkan nilai yang sama.

4.2.1.2 Pengujian Dekripsi AES

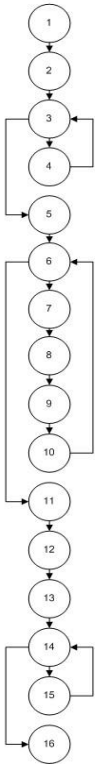
Pengujian dekripsi AES dilakukan pada algoritma dekripsi seperti pada tabel 4.7.

Tabel 4. 7 Algoritma Dekripsi AES

Line	Source
1	byte[ ] tmp = new byte[in.length];
2	byte[ ][ ] state = new byte[4][Nb()]; for (int i = 0;
3	i < in.length; i++) {
4	state[i / 4][i % 4] = in[i % 4 * 4 + i / 4];
	}
5	state = addRoundKey(state, w, Nr());
6	for (int round = Nr() - 1; round >= 1; round--) { state =
7	invSubBytes(state); state = invShiftRows(state); state =
8	addRoundKey(state, w, round); state = invMixColumns(state);
9	}
10	state = invSubBytes(state); state =
	invShiftRows(state); state =
11	addRoundKey(state, w, 0); for (int i =
12	0; i < tmp.length; i++) {
13	tmp[i % 4 * 4 + i / 4] = state[i / 4][i % 4];
14	} return
15	tmp;
16	

1. *Flow Graph* Dekripsi *AES*

Berdasarkan pada tabel 4.7 maka dapat dibentuk *flow graph* seperti pada gambar 4.5.



Gambar 4. 5 Flow Graph

Dekripsi AES Hitung cyclomatic complexity yaitu sebagai berikut:

Region = 4

V(G) = Edge – Node + 2

= 18 – 16 + 2

= 4

Berdasarkan pada hasil *cyclomatic complexity* maka didapat empat *independent path* yaitu:

1 = 1,2,3,4,3,5,6,7,8,9,10,6,11,12,13,14,15,14,16

2 = 1,2,3,5,6,7,8,9,10,6,11,12,13,14,15,14,16

3 = 1,2,3,5,6,11,12,13,14,15,14,16

4 = 1,2,3,5,6,11,12,13,14,16

2. *Graph Matrix* Dekripsi *AES*

Graph matrix dekripsi *AES* terlihat pada tabel 4.8.

Tabel 4. 8 Graph Matrix Dekripsi AES

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**sum**
1		1															0
2			1														0
3				1	1												1
4			1														0
5						1											0
6							1				1						1
7								1									0
8									1								0
9										1							0
10						1											0
11												1					0
12													1				0
13														1			0
14															1	1	1
15														1			0
16																	0
Total																	**3**

V(G) = Jumlah *Graph Matrix* + 1

= 3 + 1

= 4

Berdasarkan pengujian yang dilakukan pada setiap metode, dihasilkan nilai *Cyclomatic Complexity* yang sama yaitu 4, Maka dapat disimpulkan bahwa pengujian *white box* pada proses dekripsi AES berjalan dengan baik, karena setiap pengujian menghasilkan nilai yang sama.

4.2.1.3 *Pengujian* generateSubKeys *AES*

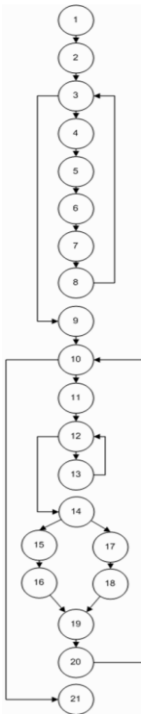
Pengujian *generate Sub Keys AES* dilakukan pada algoritma *generate Sub Keys* seperti pada tabel 4.9.

Tabel 4. 9 Algoritma generate Sub Keys AES

Line	Source
1	byte[][] tmp = new byte[nb * (nr + 1)][4];
2	int i = 0; while (i < Nk) {
3	tmp[i][0] = key[i * 4];
4	tmp[i][1] = key[i * 4 + 1];
5	tmp[i][2] = key[i * 4 + 2];
6	tmp[i][3] = key[i * 4 + 3];
7	i++; } i = Nk;
8	while (i < Nb * (Nr + 1)) {
9	byte[] temp = new byte[4]; for (int k = 0; k < 4; k++) { temp[k]
10	= tmp[i - 1][k];
11	}
12	if (i % Nk == 0) {
13	temp = subWord(rotWord(temp));
14	temp[0] = (byte) (temp[0] ^ (Rcon[i / nk] & 0xff));
15	} else if (nk > 6 && i % nk == 4) { temp =
16	subWord(temp);
17	}
18	tmp[i] = xor_Func(tmp[i - Nk], temp);
19	i++; } return tmp;
20	
21	

1. ***Flow Graph Generate Sub Key AES***

Berdasarkan pada tabel 4.9 maka dapat dibentuk *flow graph* seperti pada gambar 4.6.



Gambar 4. 6 Flow Graph

generate sub key AES Hitung cyclomatic complexity yaitu sebagai berikut:

Region = 5

V(G) = Edge – Node + 2

= 24 – 21 + 2

= 5

Berdasarkan pada hasil *cyclomatic complexity* maka didapat lima *independent path* yaitu:

1 = 1,2,3,4,5,6,7,8,3,9,10,11,12,13,12,14,15,16,19,20,10,21

2 = 1,2,3,4,5,6,7,8,3,9,10,11,12,13,12,14,17,18,19,20,10,21

3 = 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,19,20,10,21

4 = 1,2,3,4,5,6,7,8,9,10,11,12,13,14,17,18,19,20,10,21

5 = 1,2,3,4,5,6,7,8,9,10,21

2. *Graph Matrix generate sub key AES*

Graph matrix generate sub key AES terlihat pada tabel 4.10.

Tabel 4. 10 Graph Matrix generate sub key AES

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	sum
1		1																				0
2			1						1													1
3				1																		0
4					1																	0
5						1																0
6							1															0
7								1														0
8			1						1													1
9										1												0
10											1										1	1
11												1										0
12													1	1								1
13														1								0
14															1		1					1
15																1						0
16																			1			0
17																		1				0
18																			1			0
19																				1		0
20										1												0
21																						0
Total																						4

V(G) = Jumlah *Graph Matrix* + 1

= 4 + 1

= 5

Berdasarkan pengujian yang dilakukan pada setiap metode, dihasilkan nilai *Cyclomatic Complexity* yang sama yaitu 5, Maka dapat disimpulkan bahwa pengujian white box pada proses *generate sub key* AES berjalan dengan baik, karena setiap pengujian menghasilkan nilai yang sama.

4.2.3 Pengujian *Black Box*

Pengujian yang dilakukan yaitu terdiri dari pengujian kapasitas citra digital, pengujian performansi, pengujian kualitas citra digital, dan pengujian ketahanan citra digital.

4.2.4 Pengujian Program

Pengujian ini dilakukan untuk mengetahui apakah rancangan dan implementasi yang sudah dilakukan berjalan sesuai dengan prosedur yang diinginkan atau tidak.

Pengujian ini dilakukan terhadap dua format gambar yaitu bitmap dan png. Citra yang akan dipakai sebagai media penampung dapat dilihat pada tabel 4.12 Pesan yang akan disisipkan dapat dilihat pada tabel 4.11.

Tabel 4. 11 Citra Bitmap, PNG dan JPG sebagai media penampung

No	Citra Bitmap	Resolusi	Ukuran Citra (byte)
1	 america.bmp	512 x 512	589,880
2	 carton.bmp	512 x 512	589,880
3	 flowers.bmp	512 x 512	491,576
4	 reaper.bmp	512 x 512	589,880
5	 Windows8.bmp	512 x 512	589,880

No	Citra PNG	Resolusi	Ukuran Citra (byte)
1	 Spiderman .png	512 x 512	192,257
2	 Pasteur .png	512 x 512	111,773
3	 carton.png	512 x 512	297,815
4	 walpaper.png	512 x 512	211,850
5	 natur.png	512 x 512	471,277

No	Citra JPG	Resolusi	Ukuran Citra (byte)
1	 Asus.jpg	512 x 512	42,230

2	 **Pasteur.jpg**	**512 x 512**	**143,783**
3	 **Walpaper.jpg**	**512 x 512**	**135,654**
4	 **Saiya.jpg**	**512 x 512**	**148,251**
5	 **Spiderman.jpg**	**512 x 512**	**94,982**

Tabel 4. 12 Pesan yang akan disisipkan

No	Nama File Pesan Text	Ukuran File Pesan Teks (byte)
1	PUSTAKA.txt	3,475
2	BAB II.txt	37,539
3	Pesan.doc	30,671
4	RPP.docx	72,173
5	FRS.doc	189,952
6	Buku.docx	192,168
7	Stegano.doc	240,733
8	BAB III.doc	985,398

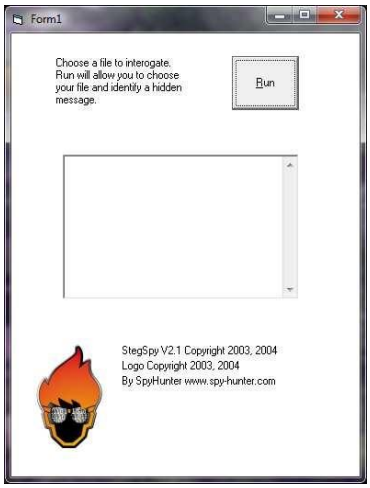
4.2.5 Pengujian Keamanan Pesan

Pengujian keamanan pesan dilakukan untuk mengetahui aman atau tidaknya pesan yang tersisip pada citra, pengujian keamanan ini dilakukan menggunakan aplikasi *steganalysis* yaitu *StegSpy 2.1*. Pengujian ini dilakukan terhadap dua format gambar yaitu bitmap dan png.

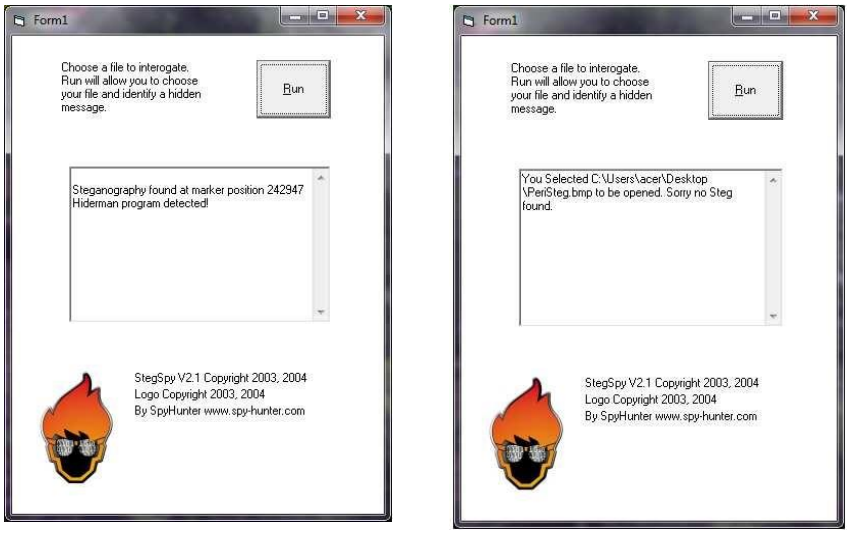
StegSpy dapat mengidentifikasi suatu *file stego* dimana *Steg-spy* ini dapat mendeteksi steganografi yang digunakan untuk menyembunyikan informasi/pesan. Versi yang paling terakhir juga mengidentifikasi lokasi isi pesan

yang disembunyikan dimana dengan cara menemukan kunci yang digunakan pada saat proses penyisipan. *StegSpy* sekarang ini mengidentifikasi program-program steganografi yang ada seperti: *Hiderman, JPHideandSeek*, dan program-program aplikasi data *hiding*/penyembunyian data lainnya.

Adapun program aplikasi Steg-spy yang dimaksud didapat dilihat pada gambar 4.10. Dimana tombol *Run* berfungsi untuk membuka file steganografi sekaligus melakukan deteksi dan cracking terhadap file steganografi.



Contoh apabila *stegspy* ini dapat mengidentifikasi *file stego* yang ada pada berkas citra penampung seperti pada gambar 4.10



File Stego Terdeteksi File Stego Tidak Terdeteksi

Gambar 4. 7 Hasil Deteksi File Stego

Pengujian keamanan pesan pada citra bitmap dilakukan pada lima citra bitmap dengan ukuran 512 x 512 pixels yang telah disisipi pesan. Pesan yang disisipkan dapat dilihat pada tabel 4.13.

Tabel 4. 13 Pengujian keamanan pesan pada citra BMP

No	File pesan text	Ukuran file pesan text (byte)	Citra				
			America .bmp	Carton .bmp	Flower .bmp	Reaper .bmp	Windows8 .bmp
1	PUSTAKA.txt	3,475	Tidak Terdeteksi	Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi
2	BAB II.txt	37,539	Tidak Terdeteksi	Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Terdeteksi
3	Pesan.doc	30,671	Tidak Terdeteksi	Terdeteksi	Tidak Terdeteksi	Terdeteksi	Terdeteksi
4	RPP.docx	72,173	Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Terdeteksi	Terdeteksi
5	FRS.doc	189,952	Tidak Terdeteksi	-	Tidak Terdeteksi	Terdeteksi	Terdeteksi
6	Buku.docx	192,168	Terdeteksi	-	Tidak Terdeteksi	Terdeteksi	Terdeteksi
7	Stegano.doc	240,733	Terdeteksi	Terdeteksi	Tidak Terdeteksi	Terdeteksi	Tidak Terdeteksi
8	BAB III.doc	985,398	Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Terdeteksi	Terdeteksi

Pengujian keamanan pesan pada citra png dilakukan pada lima citra png dengan ukuran 512 x 512 pixels yang telah disisipi pesan. Pesan yang disisipkan dapat dilihat pada tabel 4.14.

Tabel 4. 14 Pengujian Keamanan Pesan Pada Citra PNG

No	File pesan text	Ukuran file pesan text (byte)	Citra				
			America .png	Carton .png	Flower .png	Reaper .png	Windows8 .png
1	PUSTAKA.txt	3,475	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi
2	BAB II.txt	37,539	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Terdeteksi
3	Pesan.doc	30,671	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi
4	RPP.docx	72,173	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi
5	FRS.doc	189,952	Tidak Terdeteksi	Tidak Terdeteksi	Terdeteksi	Terdeteksi	Tidak Terdeteksi
6	Buku.docx	192,168	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Terdeteksi	Tidak Terdeteksi
7	Stegano.doc	240,733	Tidak Terdeteksi	Tidak Terdeteksi	Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi
8	BAB III.doc	985,398	Tidak Terdeteksi	Tidak Terdeteksi	Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi

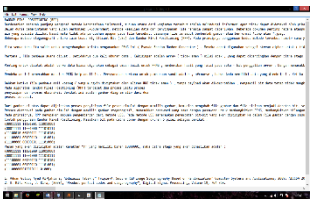
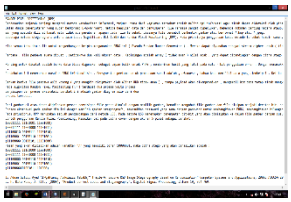
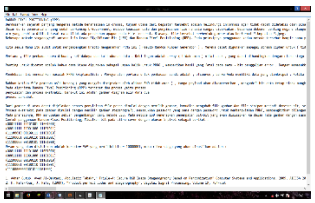
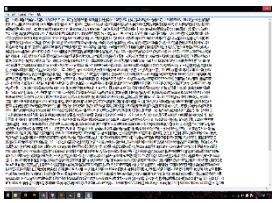
Pengujian keamanan pesan pada citra png dilakukan pada lima citra jpg dengan ukuran 512 x 512 pixels yang telah disisipi pesan. Pesan yang disisipkan dapat dilihat pada tabel 4.15.

Tabel 4. 15 Pengujian keamanan pesan pada citra JPG

No	File pesan	Ukuran file	Citra

	text	pesan text (byte)	Ganesa Steg.png	Green Steg.png	Kota Steg.png	Ternak Steg.png	Anime Steg.png
1	PUSTAKA.txt	3,475	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi
2	BAB II.txt	37,539	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Terdeteksi
3	Pesan.doc	30,671	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi
4	RPP.docx	72,173	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi
5	FRS.doc	189,952	Tidak Terdeteksi	Tidak Terdeteksi	Terdeteksi	Terdeteksi	Tidak Terdeteksi
6	Buku.docx	192,168	Tidak Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi	Terdeteksi	Tidak Terdeteksi
7	Stegano.doc	240,733	Tidak Terdeteksi	Tidak Terdeteksi	Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi
8	BAB III.doc	985,398	Tidak Terdeteksi	Tidak Terdeteksi	Terdeteksi	Tidak Terdeteksi	Tidak Terdeteksi

Tabel 4. 16 Pengujian Keamanan Pesan Menggunakan Kunci Simetris

	Data Normal (Kunci Enkripsi sama dengan Kunci Dekripsi)	Data Salah (Kunci Enkripsi berbeda dengan Kunci Dekripsi)
Citra (citrastega.bmp)		
Pesan (pesan.txt)		
Kunci Enkripsi	mAYUmScDUPrmHj4JU b0SL0QMyR7lxdfa	mAYUmScDUPrmHj4JU b0SL0QMyR7lxdfa
Kunci Dekripsi	mAYUmScDUPrmHj4JUb0 SL0QMyR7lxdfa	mAYUmScDUPrmHj4J Ub0SL0QMyR7lxdfa
Hasil		

4.2.6 Pengujian Kualitas Citra

Pengujian kualitas citra dilakukan untuk mengetahui kualitas citra yang telah disisipi pesan. Kualitas pada citra dapat diketahui menggunakan *Peak Signal to Noise Ration* (PSNR) yang merupakan standar pengukuran error pada pengolahan citra. Kualitas citra yang telah dimanipulasi pada umumnya mempunyai nilai

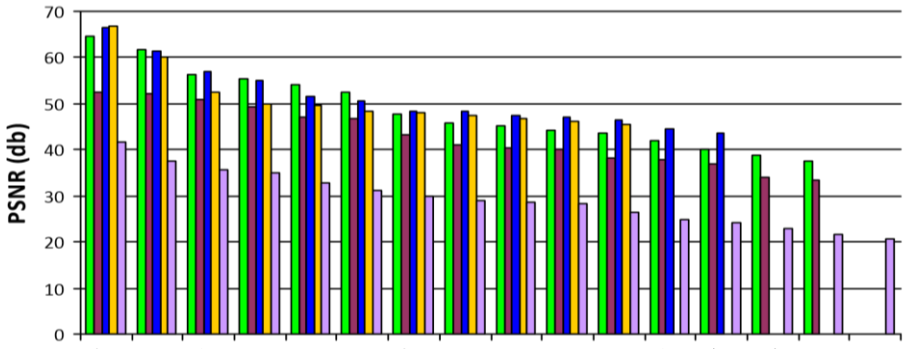
PSNR yang berada pada rentang antara 20 – 40db .Jika PSNR yang tinggi maka kualitas citra akan lebih baik. Pengujian dilakukan terhadap dua format citra yaitu bitmap, png dan jpg.

1. Pengujian Kualitas Citra Bitmap

Pengujian kualitas pada citra bitmap dilakukan pada lima citra bitmap dengan ukuran 512 x 512 pixels. Pengujian ini dilakukan menggunakan citra penampung yang dapat dilihat pada tabel 4.11, sedangkan pesan yang disisipkan dapat dilihat pada tabel 4.17.

Tabel 4. 17 Pengujian Kualitas Citra Bitmap

No	File pesan text	Ukuran file pesan text (byte)	Nilai PSNR Citra (db)				
			america.bmp	carton.bmp	flower.bmp	reaper.bmp	Windows8.bmp
1	Abstrak.txt	832	64,62	52,64	66,51	66,96	41,56
2	Gif.txt	1.600	61,76	52,34	61,35	59,99	37,67
3	Sistem.txt	3.904	56,26	50,81	56,98	52,54	35,54
4	Netbeans.txt	5.056	55,25	49,45	55,01	50,10	35,08
5	Latar.txt	8.192	54,04	47,17	51,66	49,61	32,64
6	Bitmap.txt	12.456	52,45	46,77	50,69	48,27	31,21
7	Aplikasi.txt	28.224	47,77	43,12	48,42	47,92	29,84
8	Fisika.txt	51.288	45,90	41,04	48,24	47,31	29,04
Rata-Rata PSNR			**48,64**	**42,94**	**51,38**	**51,03**	**29,37**
			44,67				



Dari pengujian tabel 4.17 dapat dilihat pada grafik seperti pada gambar

Pesan yang disisipkan

Gambar 4. 8 Grafik Pengujian Kualitas Citra Bitmap

Pengujian kualitas citra bitmap seperti pada tabel 4.18 menunjukan kualitas citra yang telah disisipi pesan mempunyai nilai PSNR rata-rata 44,67 db.

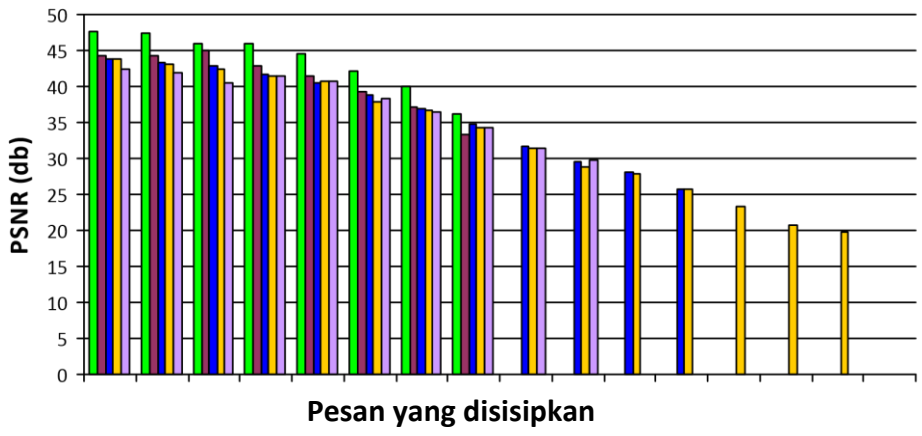
2. Pengujian Kualitas Citra PNG

Pengujian kualitas pada citra png ini dilakukan terhadap citra dengan resolusi 512 x 512 pixels. Pengujian ini dilakukan menggunakan citra penampung yang dapat dilihat pada tabel 4.11, sedangkan pesan yang disisipkan dapat dilihat pada tabel 4.18.

Tabel 4. 18 Pengujian Kualitas Citra PNG

No	File pesan text	Ukuran file pesan text (byte)	Nilai PSNR Citra (db)				
			Ganesa.png	Green.png	Kota.png	Ternak.png	Anime.png
1	Abstrak.txt	832	47,52	44,33	43,79	43,80	42,42
2	Gif.txt	1.600	47,40	44,23	43,25	43,21	41,96
3	Sistem.txt	3.904	46,03	43,93	42,79	42,43	40,45
4	Netbeans.txt	5.056	45,89	42,85	41,64	41,37	41,41
5	Latar.txt	8.192	44,56	41,51	40,38	40,68	40,74
6	Bitmap.txt	12.456	42,26	39,20	38,77	37,96	38,40
7	Aplikasi.txt	28.224	40,04	37,19	36,98	46,57	36,33
8	Fisika.txt	51.288	36,20	33,28	34,85	34,25	34,35
Rata-Rata PSNR			**43,73**	**40,81**	**36,46**	**33,89**	**37,73**
			38,52				

Dari pengujian tabel 4.18 dapat dilihat pada grafik seperti pada gambar



Gambar 4. 9 Grafik Pengujian Kualitas Citra PNG

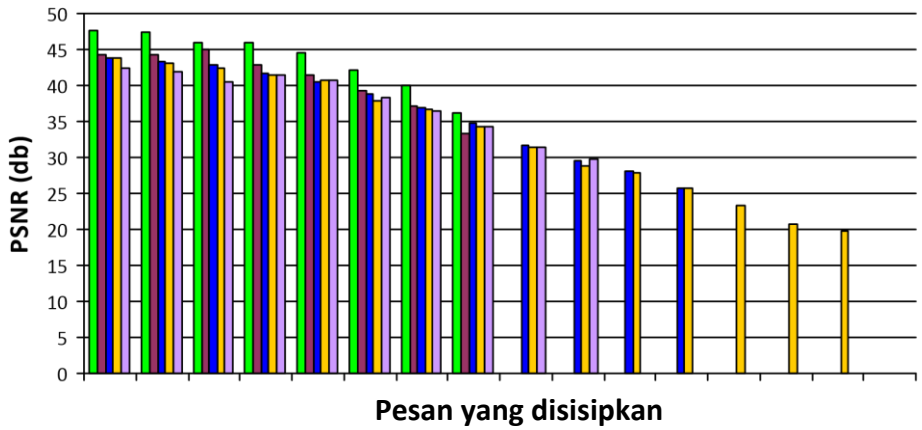
3. Pengujian Kualitas Citra JPG

Pengujian kualitas pada citra jpg ini dilakukan terhadap citra dengan resolusi 512 x 512 pixels. Pengujian ini dilakukan menggunakan citra penampung yang dapat dilihat pada tabel 4.11, sedangkan pesan yang disisipkan dapat dilihat pada tabel 4.19.

Tabel 4. 19 Pengujian Kualitas Citra JPG

No	File pesan text	Ukuran file pesan text (byte)	Nilai PSNR Citra (db)				
			Ganesa.png	Green.png	Kota.png	Ternak.png	Anime.png
1	Abstrak.txt	832	47,52	44,33	43,79	43,80	42,42
2	Gif.txt	1.600	47,40	44,23	43,25	43,21	41,96
3	Sistem.txt	3.904	46,03	43,93	42,79	42,43	40,45
4	Netbeans.txt	5.056	45,89	42,85	41,64	41,37	41,41
5	Latar.txt	8.192	44,56	41,51	40,38	40,68	40,74
6	Bitmap.txt	12.456	42,26	39,20	38,77	37,96	38,40
7	Aplikasi.txt	28.224	40,04	37,19	36,98	46,57	36,33
8	Fisika.txt	51.288	36,20	33,28	34,85	34,25	34,35
Rata-Rata PSNR			**43,73**	**40,81**	**36,46**	**33,89**	**37,73**
			38,52				

Dari pengujian tabel 4.19 dapat dilihat pada grafik seperti pada gambar



Gambar 4. 10 Grafik Pengujian Kualitas Citra PNG

Pengujian kualitas citra bitmap seperti pada tabel 4.19 menunjukan kualitas citra yang telah disisipi pesan mempunyai nilai PSNR rata-rata 38,52 db.

Dari tabel 4.19 nilai rata-rata PSNR pada citra berformat bitmap adalah 44,67 db, dan pada tabel 4.19 nilai rata-rata PSNR pada citra berformat png adalah 38,52

db. Dari nilai rata-rata kedua format citra tersebut, maka kualitas citra yang dihasilkan setelah disisipi pesan masih tergolong baik.

4.2.1.1 Pengujian Ketahanan

Pengujian ketahanan dilakukan untuk mengetahui ketahanan pesan yang telah tersisipi didalam citra, jika dilakukan manipulasi terhadap citra, dan untuk mengetahui ketahanan terhadap aplikasi *steganalysis.* Manipulasi yang dilakukan terhadap citra yaitu dengan melakukan *rotate left, rotate right, rotate 180^o, resize - 5%, Grayscale*, dan *Croping*. Pengujian ketahanan dilakukan terhadap empat format citra yaitu Bitmap, jpg dan png.

1. Pengujian Ketahanan Citra Bitmap

Hasil pengujian ketahanan pada citra bitmap terhadap manipulasi dapat dilihat pada tabel 4.20.

Tabel 4. 20 Pengujian Ketahanan Terhadap Manipulasi Citra Bitmap

Citra Bitmap	Rotate Left	Rotate Right	Rotate 180^o	Resize - 5%	Grayscale	Crop
1	Gagal	Gagal	Gagal	Gagal	Gagal	Gagal
2	Gagal	Gagal	Gagal	Gagal	Gagal	Gagal
3	Gagal	Gagal	Gagal	Gagal	Gagal	Gagal
4	Gagal	Gagal	Gagal	Gagal	Gagal	Gagal

Pengujian ketahanan terhadap aplikasi *Steganalysis* dilakukan menggunakan *tools* StegSpy2.1. Pengujian ini untuk mengetahui apakah pesan dapat terdeteksi oleh aplikasi steganalysis, atau tidak. Hasil pengujian seperti pada tabel 4.21.

Tabel 4. 21 Pengujian Terhadap Aplikasi Steganalysis Citra Bitmap

Citra	Hasil Deteksi
1	Terdeteksi
2	Terdeteksi
3	Terdeteksi
4	Terdeteksi

Pengujian ketahanan terhadap manipulasi yang dilakukan dengan hasil seperti pada tabel 4.20, menunjukan bahwa citra *bitmap* tidak tahan terhadap manipulasi. Sedangkan pada tabel 4.46 pengujian terhadap aplikasi *steganalysis*, citra *bitmap* yang tersisipi pesan dapat terdeteksi oleh aplikasi *steganalysis*.

2. Pengujian Ketahanan Citra JPG

Hasil pengujian ketahanan pada citra JPG terhadap manipulasi dapat dilihat pada tabel 4.22.

Tabel 4. 22 Pengujian Ketahanan Terhadap Manipulasi Citra JPG

Citra JPG	Rotate Left	Rotate Right	Rotate 180 º	Resize - 5%	Grayscale	Crop
1	Gagal	Gagal	Gagal	Gagal	Gagal	Gagal
2	Gagal	Gagal	Gagal	Gagal	Gagal	Gagal
3	Gagal	Gagal	Gagal	Gagal	Gagal	Gagal
4	Gagal	Gagal	Gagal	Gagal	Gagal	Gagal

Pengujian ketahanan terhadap aplikasi *steganalysis* dilakukan menggunakan *tools* StegSpy2.1. Pengujian ini untuk mengetahui apakah pesan dapat terdeteksi oleh aplikasi steganalysis, atau tidak. Hasil pengujian seperti pada tabel 4.23.

Tabel 4. 23 Pengujian Terhadap Aplikasi Steganalysis Citra JPG

Citra	Hasil Deteksi
1	Terdeteksi
2	Terdeteksi
3	Terdeteksi
4	Terdeteksi

Pengujian ketahanan terhadap manipulasi yang dilakukan dengan hasil seperti pada tabel 4.20, menunjukan bahwa citra *JPG* tidak tahan terhadap manipulasi. Sedangkan pada tabel 4.21 pengujian terhadap aplikasi *steganalysis*, citra *JPG* yang tersisipi pesan terdeteksi oleh aplikasi *steganalysis*.

3. Pengujian Ketahanan Citra PNG

Hasil pengujian ketahanan pada citra PNG terhadap manipulasi dapat dilihat pada tabel 4.24.

Tabel 4. 24 Pengujian Ketahanan Terhadap Manipulasi Citra PNG

Citra PNG	Rotate Left	Rotate Right	Rotate 180 º	Resize - 5%	Grayscale	Crop
1	Gagal	Gagal	Gagal	Gagal	Gagal	Gagal
2	Gagal	Gagal	Gagal	Gagal	Gagal	Gagal
3	Gagal	Gagal	Gagal	Gagal	Gagal	Gagal
4	Gagal	Gagal	Gagal	Gagal	Gagal	Gagal

Pengujian ketahanan terhadap aplikasi *steganalysis* dilakukan menggunakan *tools* StegSpy2.1. Pengujian ini untuk mengetahui apakah pesan dapat terdeteksi oleh aplikasi steganalysis, atau tidak. Hasil pengujian seperti pada tabel 4.25.

Tabel 4. 25 Pengujian Terhadap Aplikasi Steganalysis Citra PNG

Citra	Hasil Deteksi
1	Terdeteksi
2	Terdeteksi
3	Terdeteksi
4	Terdeteksi

Pengujian ketahanan terhadap manipulasi yang dilakukan dengan hasil seperti pada tabel 4.23, menunjukan bahwa citra *PNG* tidak tahan terhadap manipulasi. Sedangkan pada tabel 4.24 pengujian terhadap aplikasi *steganalysis*, citra *PNG* yang tersisipi pesan terdeteksi oleh aplikasi *steganalysis.*

Tabel 4. 26 Ketahanan Citra Terhadap Manipulasi

Citra	Rotate Left	Rotate Right	Rotate 180 º	Resize - 5%	Grayscale	Crop
Bitmap	Gagal	Gagal	Gagal	Gagal	Gagal	Gagal
JPG	Gagal	Gagal	Gagal	Gagal	Gagal	Gagal
PNG	Gagal	Gagal	Gagal	Gagal	Gagal	Gagal

Tabel 4. 27 Ketahanan Citra Terhadap Steganalysis

Citra	Hasil Deteksi
Bitmap	Terdeteksi

JPG	Terdeteksi
PNG	Terdeteksi

4.2.7 Kesimpulan Hasil Pengujian

Berdasarkan pada pengujian yang dilakukan maka didapatkan kesimpulan sebagai berikut :

1. Keamanan pada citra stego menggunakan metode RPP masih bisa terdeteksi oleh aplikasi *StegSpy*, Namun walaupun pesan bisa terdeteksi oleh aplikasi *StegSpy*, pesan tidak akan bisa langsung di baca oleh pihak ketiga. Karena sebelum pesan disisipkan, pesan dienkripsi terlebih dahulu menggunakan algoritma AES.
2. Kapasitas penyimpanan pesan menggunakan metode *Random Pixel Positioning* (RPP) cukup besar.
3. Kualitas citra yang telah disisipi pesan menggunakan metode *Random Pixel Positioning (RPP)* tergolong baik.
4. Metode *Random Pixel Positioning* (RPP) tidak tahan terhadap menipulasi citra dan serangan *steganalysis*.