BAB III

ANALISIS DAN PERANCANGAN SISTEM

Analisis ditujukan untuk memberikan gambaran secara umum terhadap perangkat lunak. Hasil analisis akan menjadi dasar untuk melakukan perancangan atau desain perangkat lunak sesuai kebutuhan sistem.

3.1 Analisis Sistem

Program aplikasi perbandingan steganografi ini dapat dibuat dengan adanya suatu analisis terlebih dahulu. Analisis sistem dilakukan dengan mendefinisikan masalah yang ada, menganalisis dan mengamati secara keseluruhan bagaimana program aplikasi ini terbentuk. Dalam pembuatan program aplikasi ini diperlukan suatu analisis yang benar agar dihasilkan suatu program aplikasi yang sesuai dengan yang diharapkan.

Secara umum analisis yang akan dilakukan adalah mengenai lingkungan dari sistem dimana program aplikasi ini akan digunakan dan siapa yang menggunakannya. Analisis lingkungan dari sistem menguraikan sejumlah entitas yang ada hubungannya dengan sistem dan berpengaruh terhadap sistem.

3.2 Analisis Masalah

Pada umumnya masalah yang timbul dalam proses steganografi audio hampir sama yaitu jumlah informasi yang akan disisipkan terbatas, masih layak atau tidaknya berkas audio yang telah disisipkan informasi itu didengar, ketahanan berkas (*Robustness*), serta kecepatan proses penyisipan maupun pengekstrakan informasi dalam proses steganografi tersebut.

3.3 Analisis Proses

Dalam analisis proses ini akan membahas tentang proses penyisipan dan ekstraksi secara umum, serta proses dari algoritma *Rijndael* serta metode steganografi phase coding dan echo data hiding.

3.3.1 Proses Penyisipan

Dalam proses penyisipan ini terdapat beberapa proses yang akan dilakukan yaitu:

1. Untuk pesan rahasia terlebih dahulu akan mengalami proses enkripsi menggunakan algoritma Rijndael sebelum disisipkan pada berkas audio *cover*.
2. Berkas audio *cover* yang masih berbentuk sinyal terlebih dahulu akan ditransformasikan ke dalam bit-bit digital.
3. Proses penyisipan akan dilakukan menggunakan dua buah teknik yang akan dibandingkan yaitu teknik *phase coding* dan *echo data hiding*. Dalam proses ini akan menghasilkan dua buah berkas audio yaitu berkas audio dengan

teknik steganografi *phase coding* dan teknik *echo data hiding* dengan masing-masing kunci (*stego key*) untuk nantinya dipakai dalam proses ekstraksi.

3.3.2 Proses Ekstraksi

Dalam proses ekstraksi, proses yang dilakukan adalah kebalikan dari proses penyisipan yaitu:

1. Berkas audio yang telah disisipi pesan dan masih berbentuk sinyal terlebih dahulu diubah ke dalam bit-bit digital.
2. Proses ekstraksi dipilih dan dilakukan berdasarkan teknik yang digunakan pada saat proses penyisipan yaitu menggunakan tekhik *Phase Coding* dan *Echo Data Hiding* serta memasukkan *stego key* yang dihasilkan saat proses penyisipan.
3. Untuk berkas audio akan di transformasikan kembali ke bentuk sinyal.
4. Pesan teks yang telah diekstrak harus melalui proses dekripsi terlebih dahulu agar teks tersebut nantinya dapat dimengerti.

3.3.3 Proses Enkripsi

Enkripsi yang digunakan dalam mengamankan pesan informasi teks menggunakan algoritma rijndael 128 bit.

Ada pun prosedur algoritma Rijndael adalah sebagai berikut:

1. Teks yang akan diinputkan terlebih dahulu diubah ke dalam bentuk hexadesimal.

2. Dalam proses penyandian ini akan menggunakan kunci (*password*) yang tetap, dalam hal ini untuk kunci enkripsi tidak dibuat fleksibel.
3. Proses penyandian melakukan operasi sebanyak sepuluh ronde. Untuk sembilan ronde (*round* 1-9) terdiri tas proses *add round key,sub bytes, shiftrows*, dan *mix colums*. Sedangkan pada ronde kesepuluh hanya melakukan *proses sub bytes, shift rows dan add round key*.

Sebagai contoh diperlihatkan pada blok masukan dan kunci berikut :

Masukan	32	43	F6	A8	88	5A	30	8D	31	31	98	A2	E0	37	07	34
Kunci	2B	7E	15	16	28	AE	D2	A6	AB	F7	15	88	09	CF	4F	3C

Karena panjang masukan dan kunci adalah 128 bit, diperoleh nilai Nk = Nb = 4, direpresentasikan dalam bentuk matriks sebagai berikut.

$$\text{Masukan} = \begin{bmatrix} 32 & 88 & 31 & E0 \\ 43 & 5A & 31 & 37 \\ F6 & 30 & 98 & 07 \\ A8 & 8D & A2 & 34 \end{bmatrix} \qquad \text{Kunci} = \begin{bmatrix} 2B & 7E & 15 & 16 \\ 28 & AE & D2 & A6 \\ AB & F7 & 15 & 88 \\ 09 & CF & 4F & 3C \end{bmatrix}$$

a. Proses *Sub bytes*

   Proses *sub bytes* adalah proses pensubtitusian masukan berdasarkan kotak-S. Kotak-S dapat dilihat pada tabel 3.1.

Tabel 3.1 kotak-s (s-box)

hex		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Koordinat x merepresentasikan digit pertama dari bilangan heksadesimal, sedangkan koordinat y merepresentasikan digit ke dua. Misalnya untuk bilangan heksadesimal 08, maka 0 berada di koordinat x dan 8 berada di koordinat y.

Jika masukan dinyatakan dengan S dan hasil subtitusi dinyatakan dengan S',maka:

$$S=\begin{bmatrix} \underline{32} & 88 & 31 & E0 \\ 43 & 5A & 31 & 37 \\ F6 & 30 & 98 & 07 \\ A8 & 8D & A2 & 34 \end{bmatrix} \quad \text{maka} \quad S_1=\begin{bmatrix} \underline{23} & C4 & C7 & E1 \\ 1A & BE & C7 & 9A \\ 42 & 04 & 46 & C5 \\ C2 & 5D & 3A & 18 \end{bmatrix}$$

b. Proses *Shift Rows*

Transformasi selanjutnya adalah pergeseran yang disebut *Shift Rows* untuk proses enkripsi, pergeseran dilakukan ke sebelah kiri. Jumlah pergeseran tergantung pada urutan baris (0, 1, 2 atau 3) dari matriks *state.* Artinya baris 0 tidak mengalamami pergeseran dan baris terakhir digeser 3 byte ke kiri. Pada proses dekripsi transformasi disebut *Inv. Shift Rows* dan pergeseran dilakukan ke sebelah kanan. Junlah pergeseran sama dengan proses enkripsi.

$$S_1 = \begin{bmatrix} 23 & C4 & C7 & E1 \\ 1A & BE & C7 & 9A \\ 42 & 04 & 46 & C5 \\ C2 & 5D & 3A & 18 \end{bmatrix} \quad \text{Maka} \quad S_2 = \begin{bmatrix} 23 & C4 & C7 & E1 \\ BE & C7 & 9A & \underline{1A} \\ 46 & C5 & \underline{42} & \underline{04} \\ 18 & \underline{C2} & \underline{5D} & \underline{3A} \end{bmatrix}$$

c. Proses *Mix Column*

Mix Column mengubah nilai dari setiap byte dengan mengambil 4 byte dan mengkombinasikannya untuk menghasilkan 4 byte baru. Untuk menjamin bahwa setiap byte berbeda (sekalipun keempat byte adalah sama), proses pengkombinasian dilakukan dengan mengalikan tiap byte dengan konstanta yang berbeda.

$$S_2 = \begin{bmatrix} 23 & C4 & C7 & E1 \\ BE & C7 & 9A & 1A \\ 46 & C5 & 42 & 04 \\ 18 & C2 & 5D & 3A \end{bmatrix} \bullet \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} = S_3 = \begin{bmatrix} 45 & 23 & D2 & 77 \\ A2 & F2 & 11 & 56 \\ 36 & 14 & B2 & 73 \\ 41 & 35 & 89 & A7 \end{bmatrix}$$

d. Add Round Key

Dalam proses ini melibatkan kunci yang dimasukkan di awal.

$$S_3 = \begin{bmatrix} 45 & 23 & D2 & 77 \\ A2 & F2 & 11 & 56 \\ 36 & 14 & B2 & 73 \\ 41 & 35 & 89 & A7 \end{bmatrix} \oplus \text{Kunci} = \begin{bmatrix} 2B & 7E & 15 & 16 \\ 28 & AE & D2 & A6 \\ AB & F7 & 15 & 88 \\ 09 & CF & 4F & 3C \end{bmatrix}$$

$$= \begin{bmatrix} DA & F1 & 25 & 96 \\ 88 & 46 & D2 & E1 \\ A4 & 18 & D5 & 22 \\ 55 & 78 & 63 & 67 \end{bmatrix} = \text{New Round Key}$$

3.3.4 Analisis Steganografi Phase Coding

A. Penyisipan

1. Pemrosesan awal pada pesan

Setelah file pesan teks1.txt yang berisi ‘Klaveryus Irvan’ dibaca, maka yang pertama akan dilakukan oleh sistem adalah mengubah karakter-

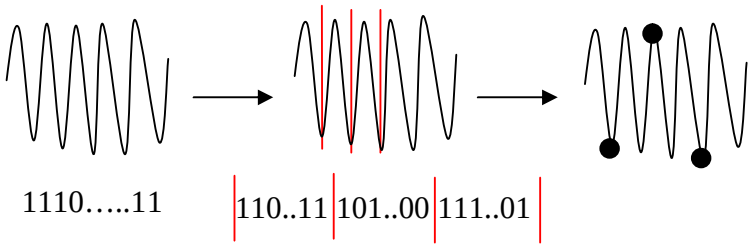
karakter yang ada dalam file tersebut ke dalam bentuk biner dan menyimpannya ke dalam sebuah *array*. Pemrosesan awal ini juga akan melakukan perhitungan jumlah bit dari pesan yang nantinya akan digunakan dalam beberapa proses lainnya.



Gambar 3.1 Perubahan Teks ke Bit

2. Pemrosesan awal pada berkas cover

Pemrosesan awal ini diawali dengan memecah berkas audio cover yaiu fanfare.wav dengan panjang 25742 bit menjadi 120 segmen sesuai panjang bit pesan, kemudian melakukan FFT pada tiap segmen dan menyimpan nilai fase dan juga magnitude dari sample data audio ke dalam dua array yang berbeda. Namun karena mod(25742,120) tidak sama dengan 0, maka dicari bilangan terdekat setelah 120 yang habis membagi panjang sinyal. Dalam kasus ini bilangan tersebut bernilai 121. Kemudian sinyal dibagi ke dalam 121 segmen yang masing-masing panjangnya adalah 211 (hasil pembagian panjang audio dengan bilangan pembagi) di mana nantinya 120 segmen pertama dari sinyal ini akan disisipi pesan.



Gambar 3.2 Proses Pembagian Segmen dan FFT

Dari hasil FFT tadi, kemudian dihitung nilai fase dan magnitudenya seperti pada tabel 3.2.

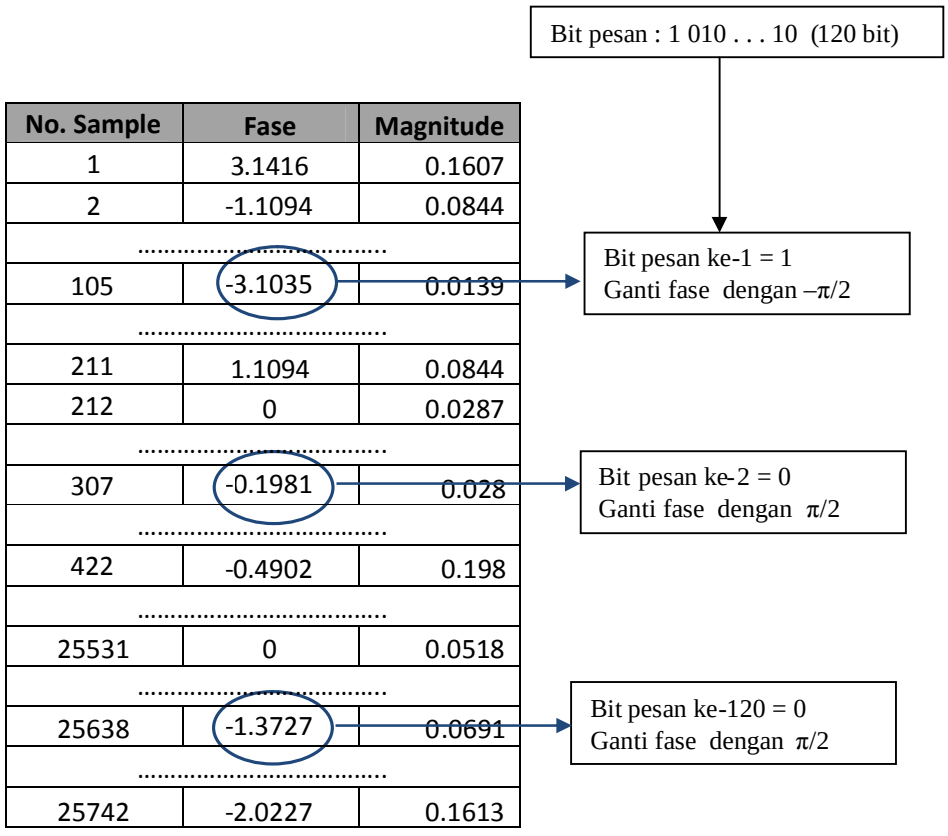
Tabel 3.2 Nilai Fase dan Magnitude dari berkas *Cover*

No. Sample	Fase	Magnitude	
1	3.1416	0.1607	Segmen ke 1
2	-1.1094	0.0844	
.....			
105	-3.1035	0.0139	
.....			
211	1.1094	0.0844	
212	0	0.0287	Segmen ke 2
.....			
307	-0.1981	0.028	
.....			
422	-0.4902	0.198	
.....			
25531	0	0.0518	Segmen ke 211
.....			
25638	-1.3727	0.0691	
.....			
25742	-2.0227	0.1613	

3. Penyisipan fase referensi dari pesan ke sinyal audio

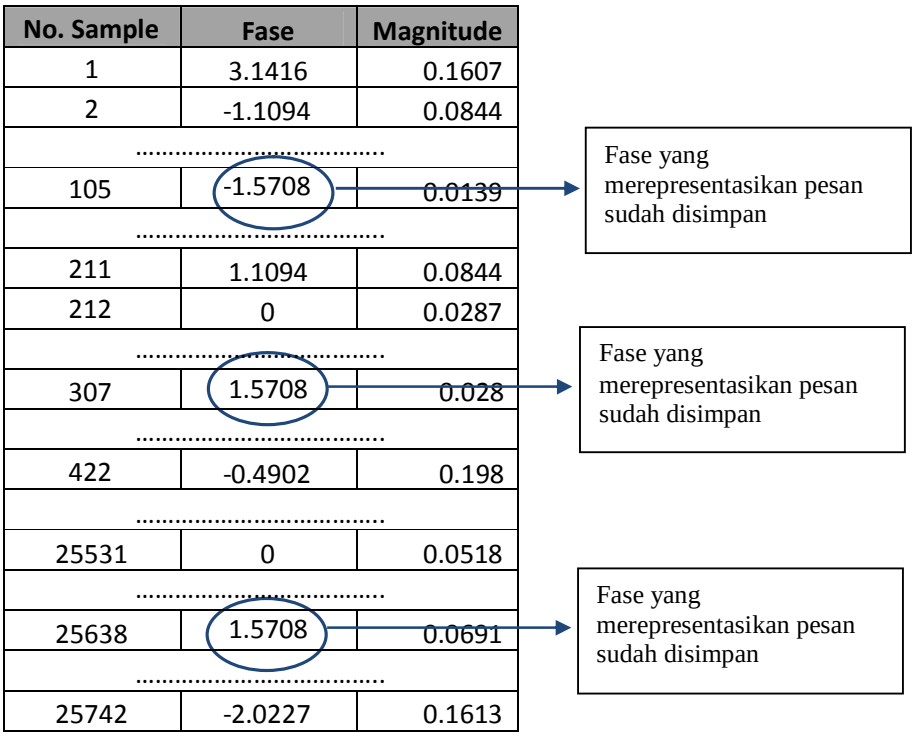
   Pada proses ini, nilai fase audio yang akan diganti dengan fase yang

merepresentasikan pesan adalah nilai fase yang terdapat pada index $xN+pos$ ($0 \leq x \leq X$) di mana X adalah jumlah bit pesan, N adalah panjang segmen, dan *pos* adalah $N/2$ untuk N genap dan $(N-1)/2$ untuk N ganjil. Pada penyisipan text1.txt ke file fanfare.wav nilai pos adalah (211-1)/2 = 105. Nilai fase audio dengan index tadi akan diganti dengan $\pi/2$ jika bit pada array pesan dengan index yang berkesesuaian bernilai 0 dan $-\pi/2$ jika bitnya bernilai 1.



No. Sample	Fase	Magnitude
1	3.1416	0.1607
2	-1.1094	0.0844
.....		
105	-3.1035	0.0139
.....		
211	1.1094	0.0844
212	0	0.0287
.....		
307	-0.1981	0.028
.....		
422	-0.4902	0.198
.....		
25531	0	0.0518
.....		
25638	-1.3727	0.0691
.....		
25742	-2.0227	0.1613

Gambar 3.2 Penyisipan Pesan Teks Pada *Phase Coding*



No. Sample	Fase	Magnitude
1	3.1416	0.1607
2	-1.1094	0.0844
.....		
105	-1.5708	0.0139
.....		
211	1.1094	0.0844
212	0	0.0287
.....		
307	1.5708	0.028
.....		
422	-0.4902	0.198
.....		
25531	0	0.0518
.....		
25638	1.5708	0.0691
.....		
25742	-2.0227	0.1613

Gambar 3.3 Akhir Penyisipan Teks pada *Phase Coding*

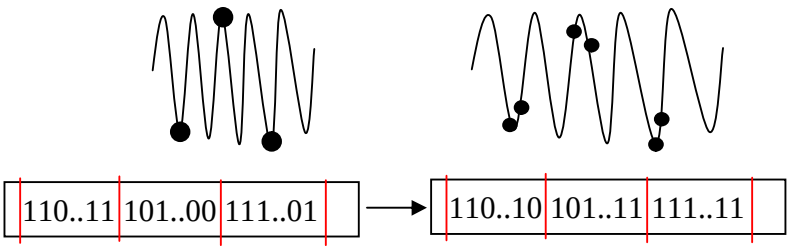
4. Membangun matriks audio

Nilai-nilai pada array magnitude yang tadi disimpan pada proses no.2 kemudian digabungkan dengan nilai-nilai pada array fase audio yang telah mengalami penyisipan fase pesan dengan menggunakan rumus :

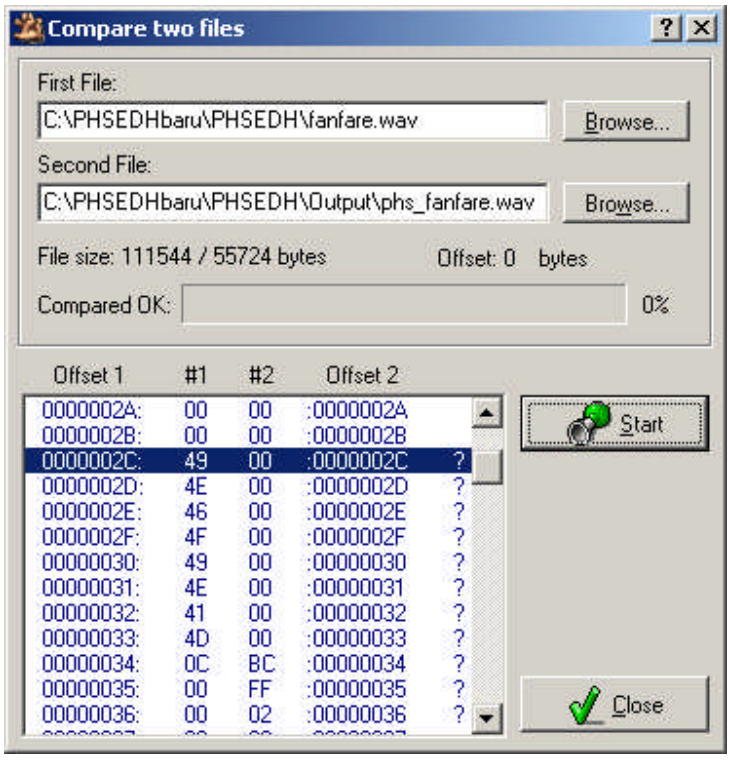
$$new_signal=mg.*exp(i*ph)$$

Di mana new_signal adalah hasil penggabungan array magnitude dan fase, mg adalah array magnitude dan ph adalah array fase yang telah mengalami penyisipan. Array new_signal ini kemudian digabung kembali dengan channel yang hanya mengalami proses FFT, untuk selanjutnya

dilakukan fungsi IFFT (Inverse Fast Fourier Transformation) pada penggabungan array ini untuk akhirnya mendapatkan sinyal audio yang baru yang telah disisipi pesan. Sinyal audio yang sudah dibangkitkan ini kemudian disimpan dengan format .wav.



Gambar 3.4 Perubahan Bit pada Stego-Audio *Phase Coding*



Gambar 3.4 *Capture* Perubahan Bit pada Stego-Audio *Phase Coding*

3.3.5 Analisis Steganografi *Echo Data Hiding*

1. Pemrosesan awal pada pesan

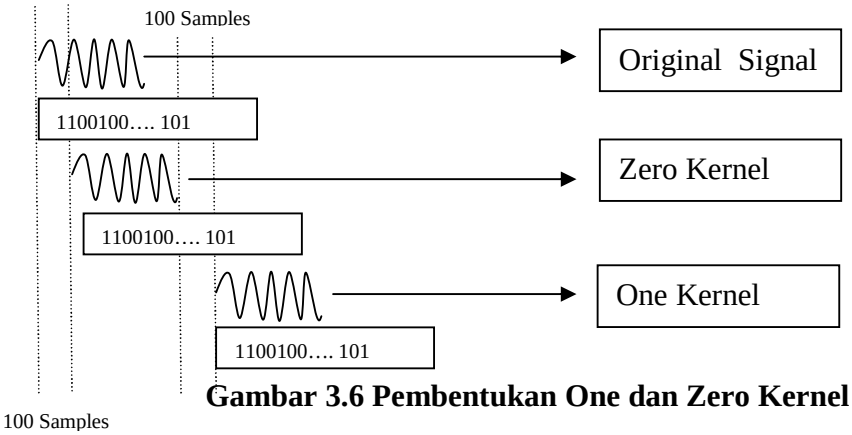
Proses pembacaan pesan sama seperti pada teknik Phasee Coding. Setelah file pesan teks1.txt yang berisi ‘Klaveryus Irvan’ dibaca, maka yang pertama akan dilakukan oleh sistem adalah mengubah karakter-karakter yang ada dalam file tersebut ke dalam bentuk biner dan menyimpannya ke dalam sebuah *array*. Pemrosesan awal ini juga akan melakukan perhitungan jumlah bit dari pesan yang nantinya akan digunakan dalam beberapa proses lainnya.



Gambar 3.5 Perubahan Teks ke Bit

2. Pembuatan One Kernel dan Zero Kernel

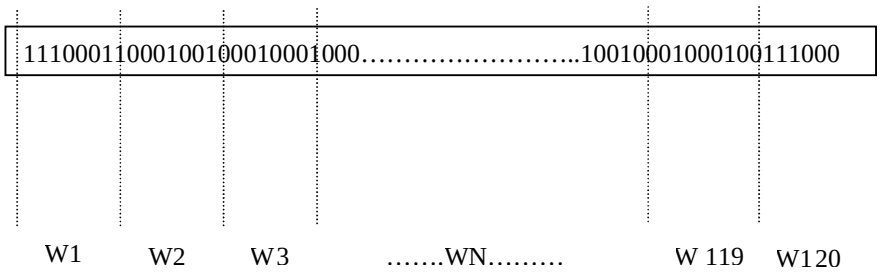
Pembuatan *One kernel* dan *Zero Kernel* tergantung dari sinyal asli yakni sinyal dari berkas audio *cover*. *Zero kernel* didapatkan dengan cara mendelay sinyal asli sepanjang 100 sample, dan *one kernel* didapatkan dengan cara mendelay sinyal asli sebesar 100+panjang data(dalam bit) sample.



Gambar 3.6 Pembentukan One dan Zero Kernel

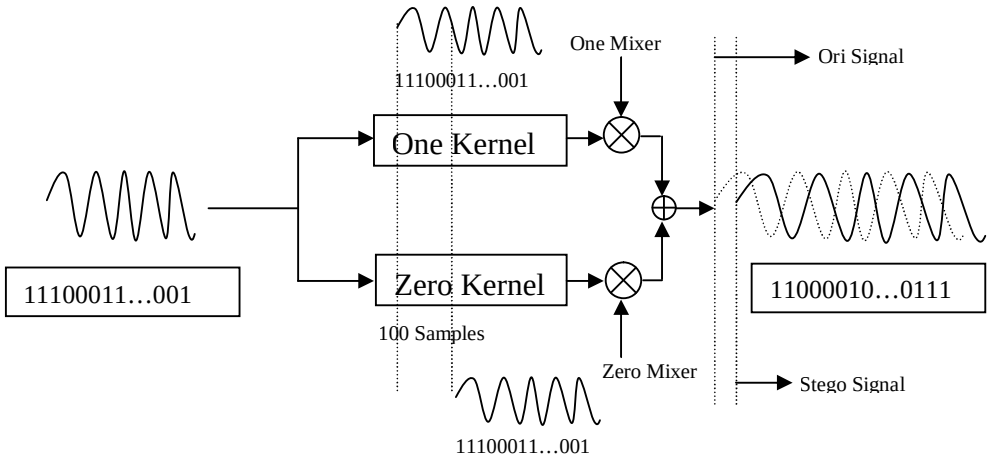
3. Pembuatan Window

Berkas audio *cover* dibagi menjadi beberapa *window* berdasarkan panjang data teks yang akan disisipkan.

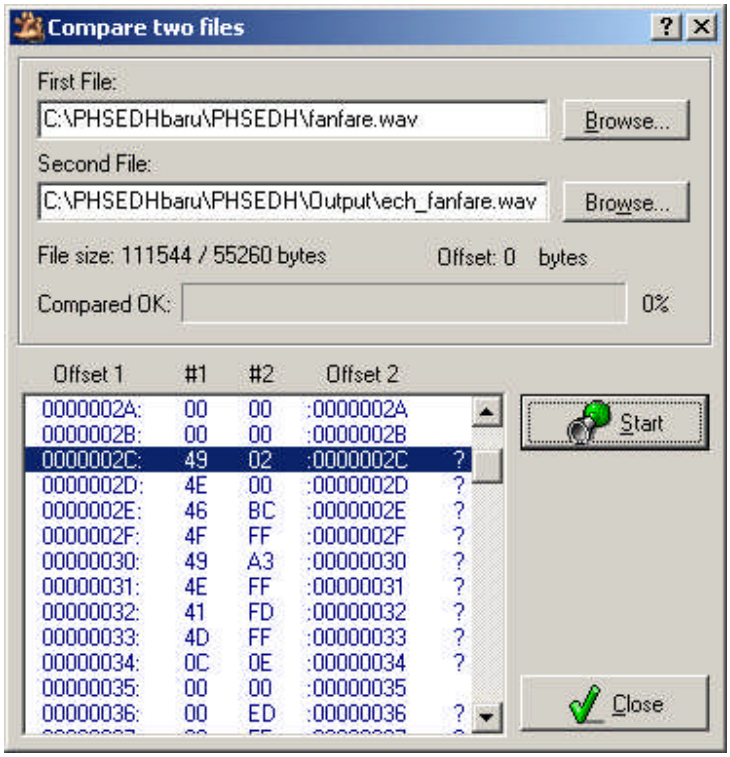


Gambar 3.7 Pembagian Window

4. Dua waktu *delay* digunakan untuk *encoding* data yang disisipkan, biner 0 akan diencodingkan dengan *delay* sebesar *offset*, dan biner 1 akan diencodingkan sebesar *offset* + *delta*. Mixer segmen sinyal asli dengan *mixer one* apabila data yang disisipkan adalah bit 1 dan *mixer zero* jika bit 0 untuk proses penyisipan



Gambar 3.8 Penyisipan pada *Echo Data Hiding*



Gambar 3.9 *Capture* Perubahan Bit pada Stego-Audio *Echo Data Hiding*

3.4 Analisis Kebutuhan Fungsional

Analisis non fungsional bertujuan untuk menghasilkan spesifikasi kebutuhan non fungsional. Spesifikasi kebutuhan non fungsional adalah spesifikasi yang rinci mengenai hal-hal yang akan dilakukan sistem ketika di implementasikan. Analisis kebutuhan ini diperlukan untuk menentukan keluaran yang akan dihasilkan sistem, masukan yang diperlukan sistem, lingkup proses yang digunakan untuk mengolah masukan menjadi keluaran. Kebutuhan ini terdiri dari kebutuhan perangkat lunak (*software*) dan kebutuhan perangkat keras (*hardware*).

3.4.1 Analisis Kebutuhan Perangkat Lunak

Kebutuhan *software* merupakan kebutuhan akan perangkat lunak yang digunakan untuk membangun program aplikasi steganografi ini. Diantara kebutuhan terebut adalah :

1. Sistem operasi berbasis windows, yang penulis gunakan adalah sistem operasi Windows XP SP 3.
2. *Tools* yang penulis gunakan adalah MATLAB R2009a.

3.4.2 Analisis Kebutuhan Perangkat Keras

Sedangkan untuk kebutuhan hardware diantaranya yaitu :

1. Processor intel core duo 2.0 Ghz
2. Harddisk 120 Gb
3. RAM 2 Gb
4. Soundcard onboard
5. Speaker (stereo)
6. Seperangkat I/O (mouse,keyboard,monitor)

3.4.3 Analisis pengguna

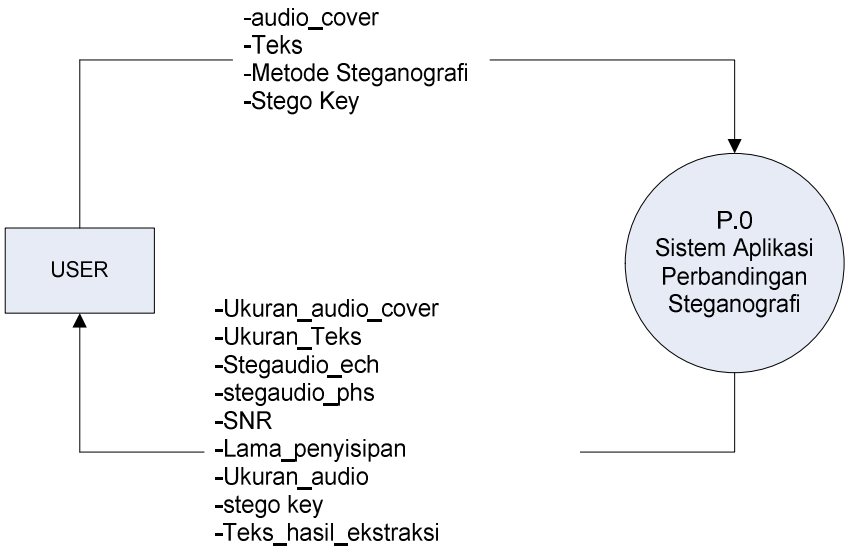
Pengguna aplikasi ini adalah orang yang menguasai sistem operasi windows xp,dan mempunyai pemahaman tentang steganografi dan kriptografi minimal defenisi dari kedua hal tersebut.

3.5 Analisis Kebutuhan Fungsional

Kebutuhan fungsional adalah deskripsi dari aktivitas-aktivitas dan layanan yang harus disediakan oleh sistem, dimana kebutuhan suatu sistem itu tentunya akan mendukung daripada pengguna atau user sistem tersebut. Analisis kebutuhan fungsional ini meliputi diagram konteks dan *Data Flow Diagram* (DFD).

3.5.1 Diagram Konteks

Diagram konteks adalah diagram yang menunjukkan keterhubungan antara perangkat lunak dengan konteks eksternal di luar program. Pada diagram ini, input berupa perintah diterima dari pengguna melalui *keyboard* dan *mouse*, sedangkan output akan ditampilkan kepada pengguna melalui monitor.



Gambar 3.2 Diagram Konteks

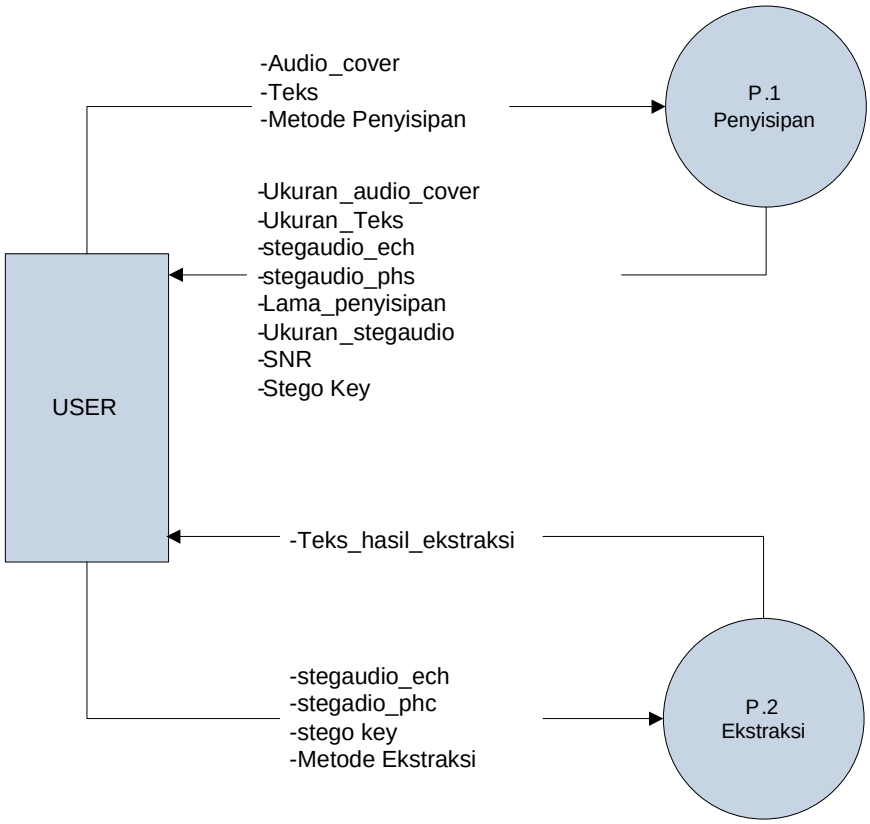
Tabel 3.2 adalah spesifikasi proses yang menjelaskan proses yang berlangsung dalam diagram konteks di atas.

Tabel 3.2 Spesifikasi Proses Diagram Konteks

No / Nama Proses	Input	Keterangan Proses	Output
P.0 / Sistem Aplikasi Perbandingan Steganografi	-audio_cover -Teks -Metode Steganografi -Stego Key	Sistem keamanan data yang dapat menyembunyikan teks ke dalam audio serta dapat mengungkapkan kembali pesan rahasia tersebut menggunakan dua buah metode yaitu Echo Data Hiding (EDH) dan Phase Coding (PhC)	-Ukuran_audio_cover -Ukuran_Teks -Audio_stego_ech -Audio_stego_phs -Lama_penyisipan -Ukuran_audio -Teks_hasil_ekstraksi

3.5.2 *Data Flow Diagram* Level 1

Dari diagram konteks sebelumnya, maka selanjutnya proses diuraikan menjadi bagian-bagian yang lebih kecil. Dalam aplikasi steganografi audio ini terdapat dua proses utama yaitu Penyisipan dan Ekstraksi. Proses tersebut dapat diuraikan dengan DFD level 1 berikut.



Gambar 3.3 *Data Flow Diagram Level* 1

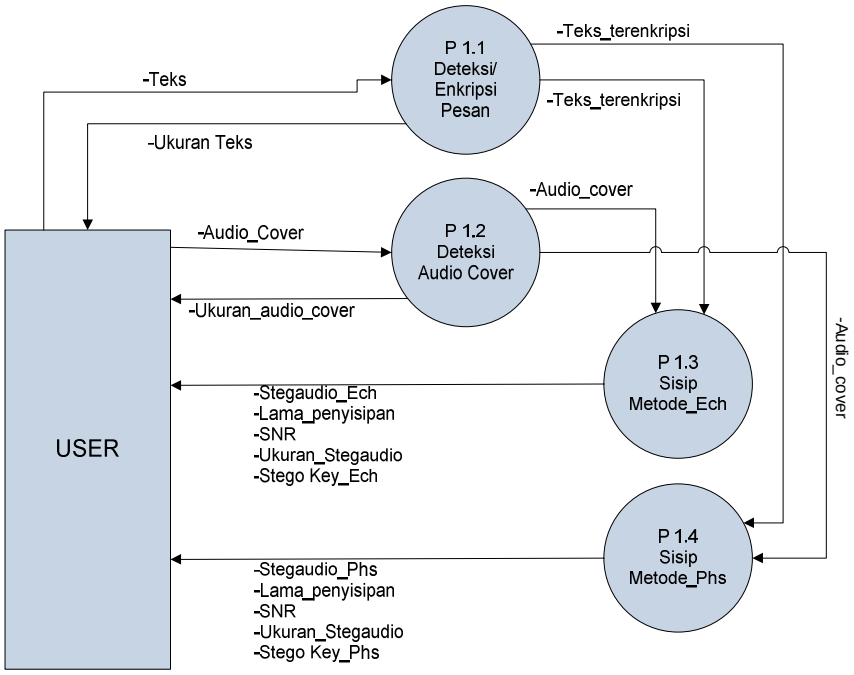
Tabel 3.3 Spesifikasi Proses DFD Level 1

No / Nama Proses	Input	Ket. Proses	Output
P.1 / Proses Penyisipan	-Audio_cover -Teks -Metode Penyisipan	Menyisipkan teks (yang terlebih dahulu di enkripsi) ke dalam berkas audio cover berdasarkan metode steganografi yang dipilih.	-Ukuran_audio_cover -Ukuran_Teks -stegaudio_ech -stegaudio_phs -Lama_penyisipan -Ukuran_stegaudio -SNR -Stego Key

P.2 / Proses Ekstraksi	-stegaudio_ech -stegadio_phc -stego key -Metode Ekstraksi	Mengekstrak teks dari dalam berkas audio hasil steganografi (stegaudio) dengan memasukkan stego key, dan melakukan proses dekripsi agar pesan yang tadinya tersandikan dapat dimengerti.	-Teks_Hasil_Ekstraksi

3.5.3 *Data Flow Diagram* Level 2 Proses 1 (Penyisipan)

Pada diagram level selanjutnya menggambarkan secara khusus tentang proses penyembunyian pesan rahasia ke dalam berkas audio *cover*. Input pada proses ini adalah password yang akan digunakan sebagai kunci enkripsi rijndael 128 bit, pesan yang akan disisipkan berupa teks berekstensi *txt* dan berkas WAV yang akan digunakan sebagai media *cover* pesan.



Gambar 3.4 *Data Flow Diagram Level* 2 Proses 1 (penyisipan)

Tabel 3.4 adalah spesifikasi proses yang menjelaskan proses yang berlangsung dalam DFD level 2 proses 1 (penyisipan).

Tabel 3.4 Spesifikasi Proses DFD Level 2 Proses 1 (Penyisipan)

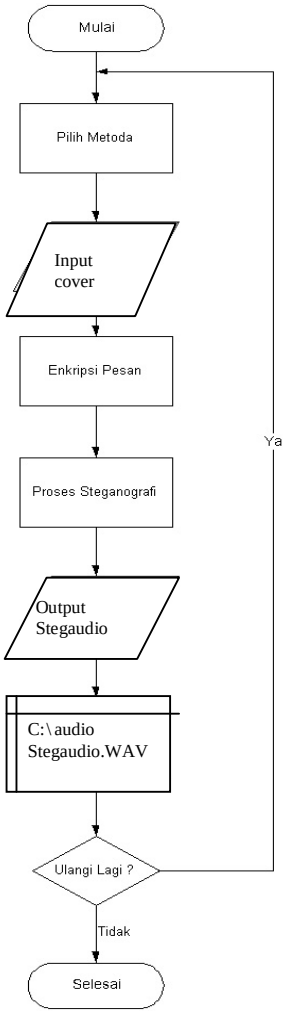
No / Nama Proses	Input	Ket. Proses	Output
P.1.1 Deteksi dan Enkripsi Pesan	-Teks	Mendeteksi Ukuran teks yang akan disembunyikan, serta mengenkripsi teks tersebut menggunakan algoritma rijndael 128 bit.	-Teks Terenkripsi
P.1.2 Deteksi audio *cover*	-Audio_ *Cover*	Menghitung ukuran dari berkas audio yang akan menjadi cover,dalam hal ini berkas audio WAV.	-Ukuran_ Audio_*cover*
P.1.3 Sisip Metode_ EDH	-Teks_terenkripsi -Audio_*Cover*	Proses menyisipkan teks yang telah terenkripsi ke dalam berkas audio cover menggunakan metode echo data hiding, di mana dalam proses ini juga akan meregenerate stego key yang akan digunakan dalam proses eksrtaksi, serta perhitungan SNR.	-Stegaudio_Ech -Lama_penyisipan -SNR -Ukuran_Stegaudio -Stego Key_Ech
P.1 .4/ Sisip Metode_Ph	-Teks_terenkripsi -Audio_*Cover*	Proses menyisipkan teks yang telah terenkripsi ke dalam berkas audio cover menggunakan metode Phase Coding, di mana dalam proses ini juga akan meregenerate stego key yang akan digunakan dalam proses eksrtaksi, serta perhitungan SNR	-Stegaudio_Phs -Lama_penyisipan -SNR -Ukuran_Stegaudio -Stego Key_Phs

3.6 Perancangan Perangkat Lunak

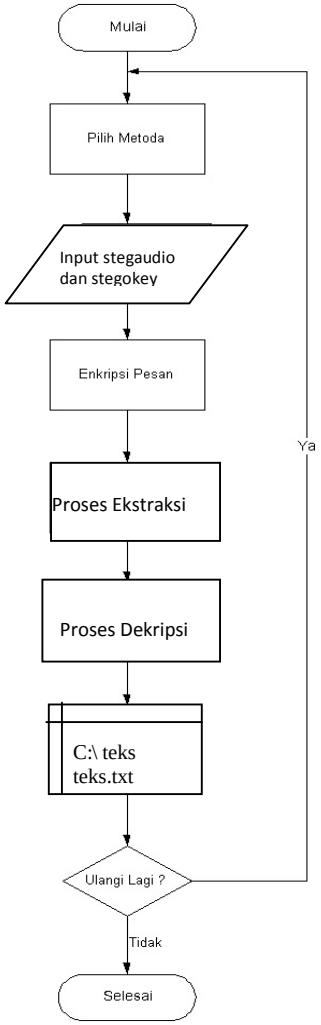
Perancangan adalah tahap mengimplementasikan keseluruhan hasil analisis yang dilakukan sebelumnya untuk menentukan kondisi akhir yang diharapkan serta merumuskan langkah-langkah dalam mencapai kondisi akhir tersebut. Perancangan dijadikan sebagai panduan yang dapat dipahami dengan mudah, memberikan gambaran yang jelas dan lengkap mengenai perangkat lunak yang akan dibangun.

3.6.1 Perancangan Prosedural

Ada beberapa proses yang akan dilakukan dalam sistem keamanan data dengan mengkombinasikan enkripsi dan steganografi ini. Secara umum, inti dari sistem keamanan ini adalah dapat berfungsi untuk menyisipkan pesan rahasia dan dapat mengungkapkan kembali pesan rahasia tersebut. Perancangan prosedural akan dibantu dengan menggunakan bagan alir (*flowchart*).

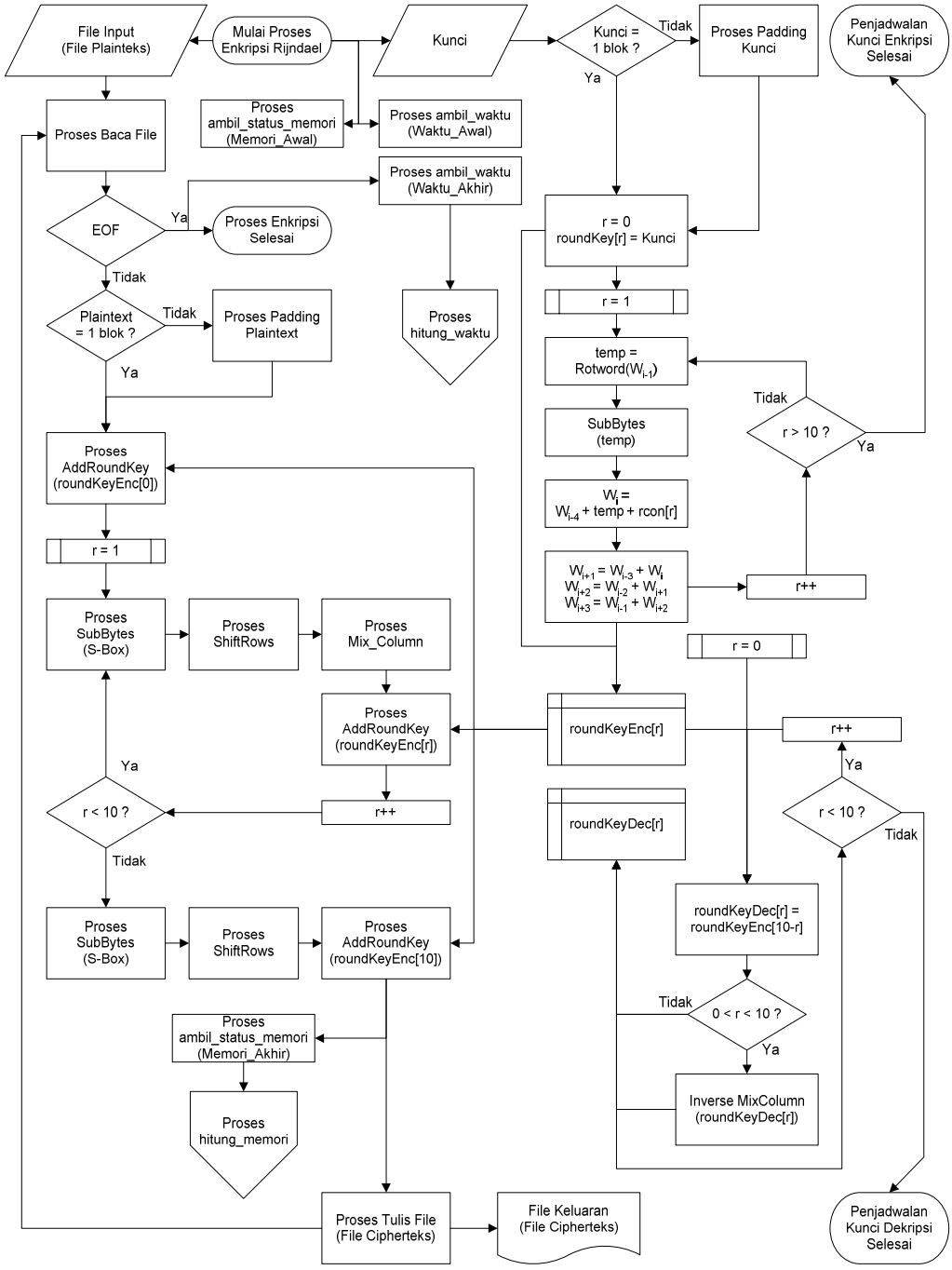


Gambar 3.5 *Flowchart* Proses Penyisipan



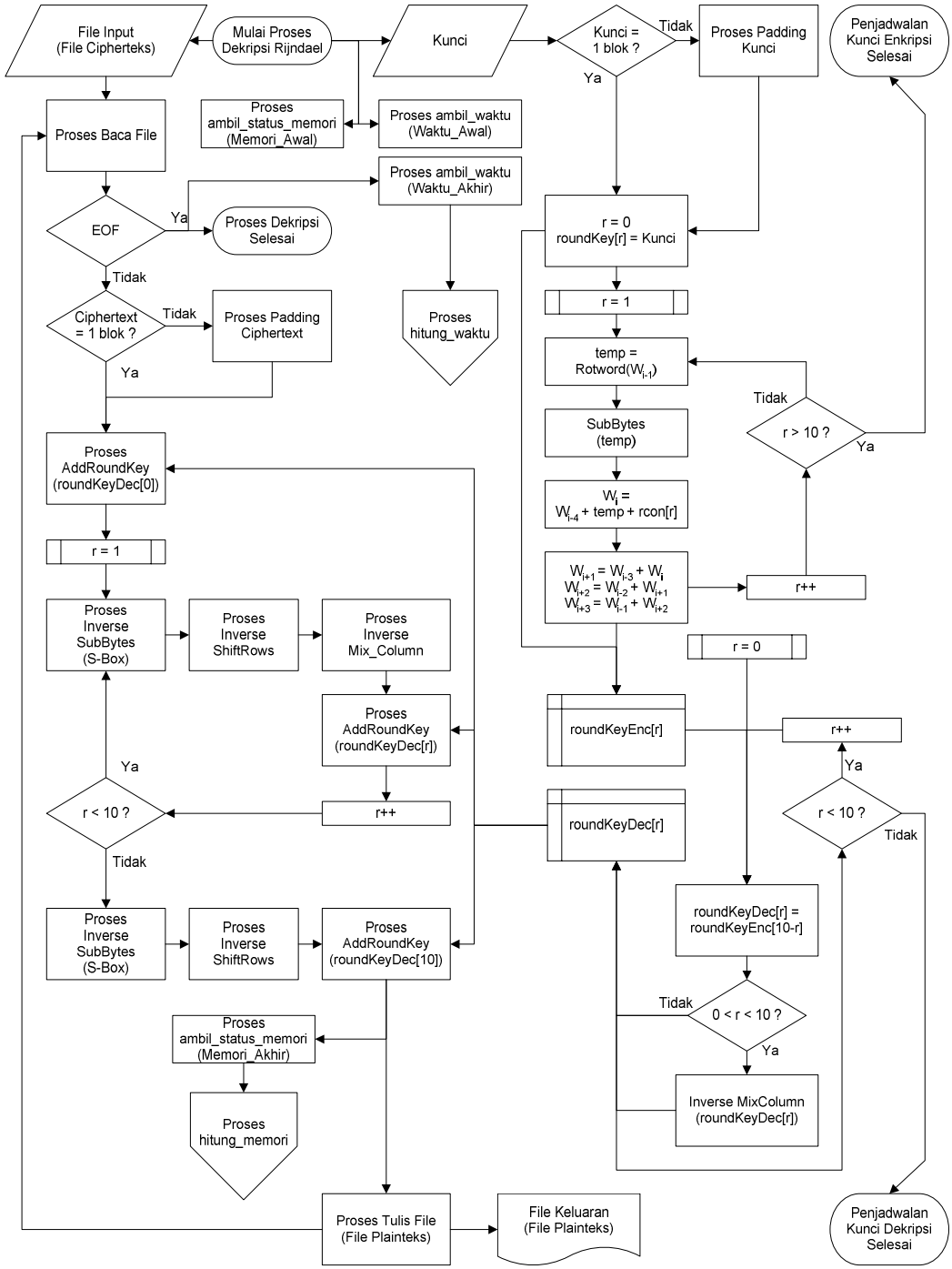
Gambar 3.6 *Flowchart* Proses Ekstraksi

Proses Enkripsi Algoritma RIJNDAEL



Gambar 3.7 *Flowchart* Enkripsi *Rijndael*

Proses Dekripsi Algoritma RIJNDAEL



Gambar 3.8 *Flowchart* Dekripsi *Rijndael*

3.6.2 Perancangan Antarmuka

Beberapa prinsip dari perancangan antarmuka yang baik telah dikembangkan dan banyak di antaranya menekankan pada antarmuka yang *user friendly* sehingga mudah digunakan. Penelitian pada bidang ini sering disebut sebagai *human factors engineering* atau *ergonomics*. Berikut adalah elemen-elemen yang harus dipertimbangkan dalam perancangan antarmuka.

1. Desain layar

Beberapa hal yang perlu dipertimbangkan dalam desain layar yaitu mengenai warna serta desain grafis yang digunakan pada antarmuka sistem dengan mengedepankan aspek *ergonomic* dan kenyamanan pengguna.

2. Umpan balik

Aspek yang paling penting dalam umpan balik (*feed back)* adalah waktu respon, yaitu waktu antara saat *user* memasukkan data dengan respon yang diberikan oleh sistem.

3. Bantuan

Pada saat *user* sedang mengoperasikan sistem, seringkali mengalami kesulitan dan tidak mengetahi apa yang harus dikerjakan berikutnya. Perancangan antarmuka Bantuan menyediakan langkah-langkah dalam menggunakan aplikasi.

4. Pengendalian kesalahan

Perancangan antarmuka harus memperhatikan pengendalian kesalahan yang dapat berupa sebagai berikut:

a. Pencegahan kesalahan

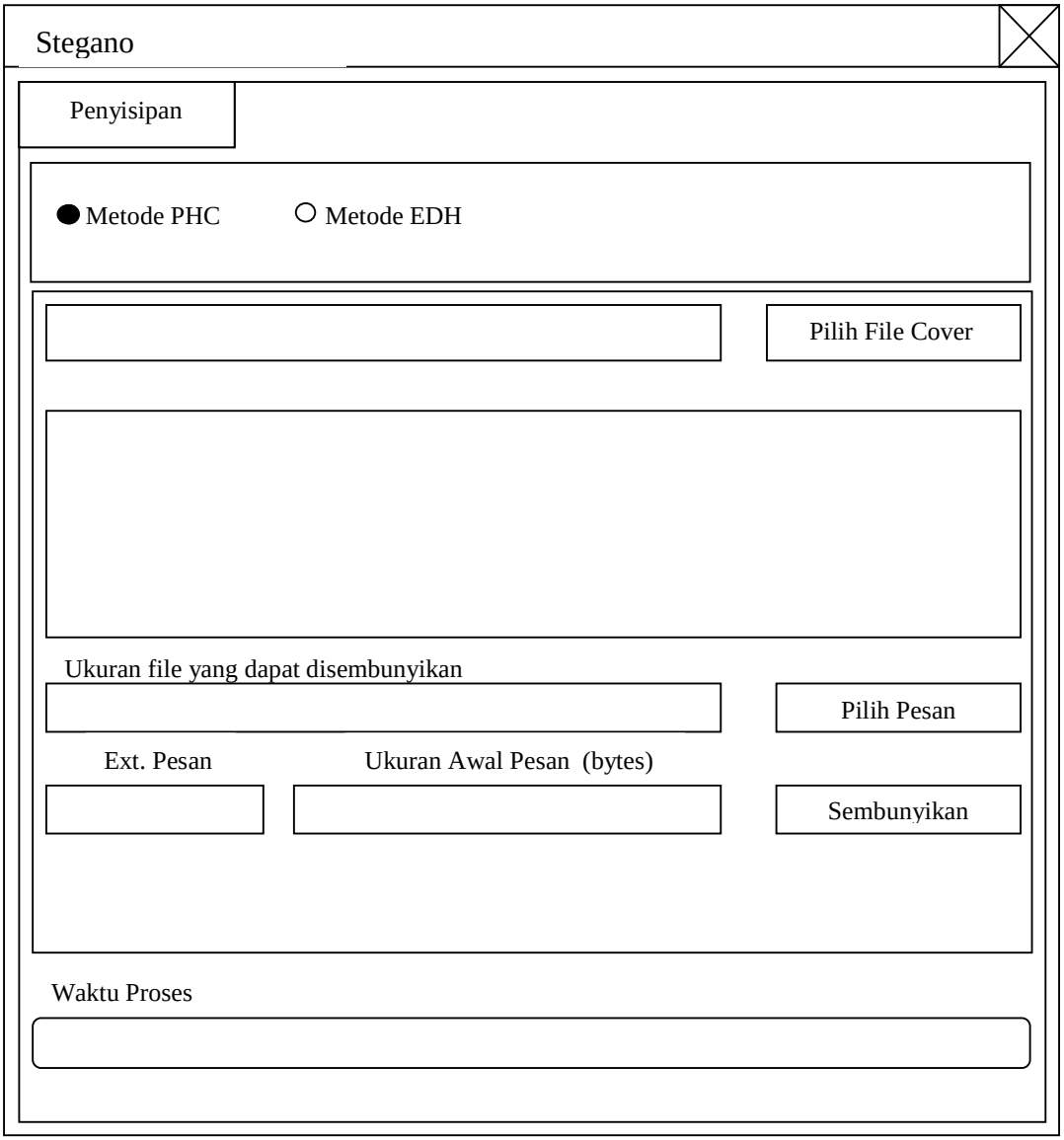
Sistem harus menyediakan instruksi yang jelas kepada pengguna tentang apa yang harus dilakukan sehingga pengguna tidak melakukan kesalahan yang seharusnya tidak perlu terjadi.

b. Pendeteksian kesalahan.

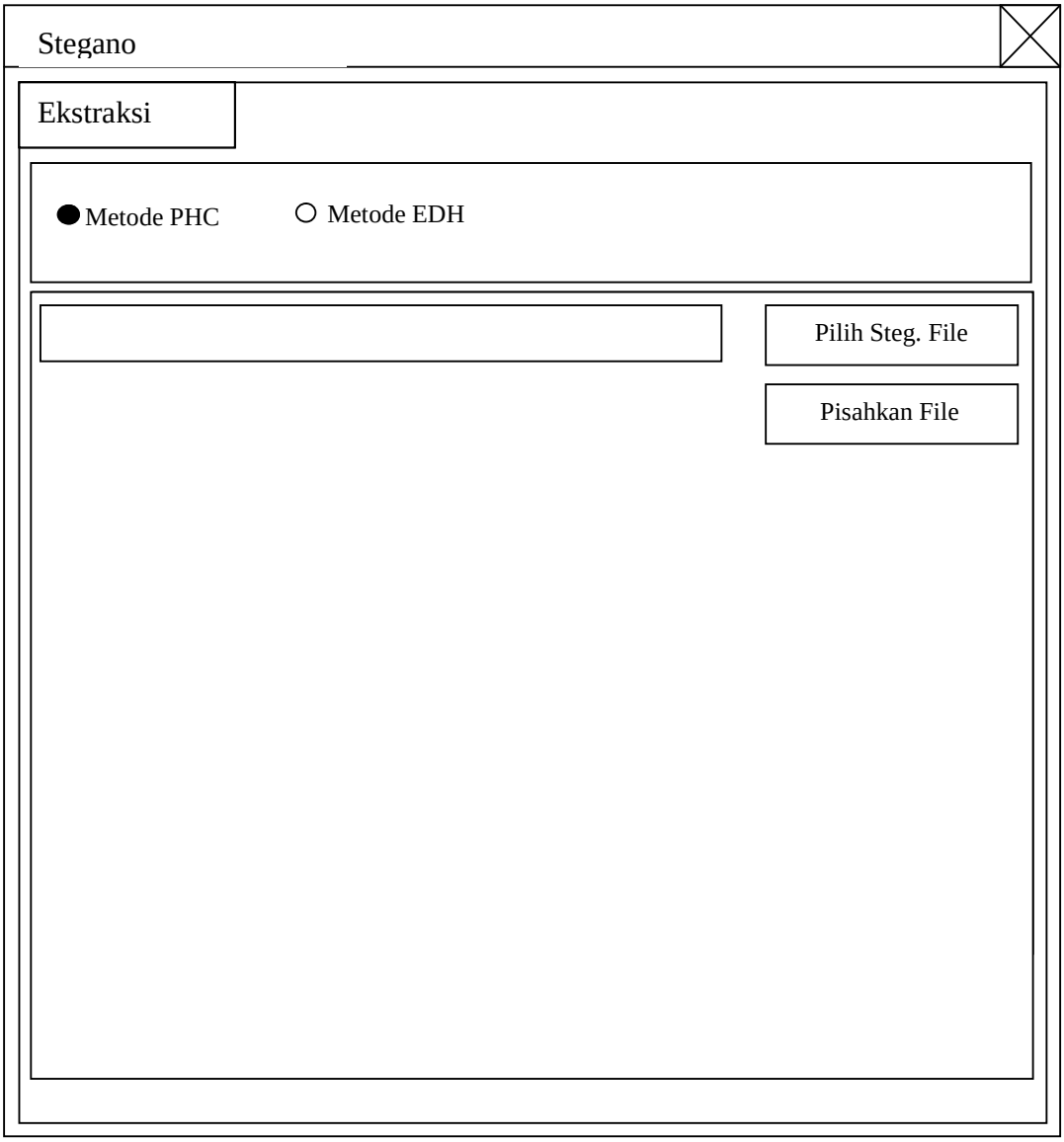
Jika suatu kesalahan terjadi, sistem harus dapat mengidentifikasi kesalahannya dengan jelas dan dapat menampilkan berita kesalahan ini, misalnya “Password tidak sesuai!”, atau “Fatal Error! Sistem dihentikan”.

c. Pembetulan kesalahan.

Jika suatu data yang dimasukkan salah, maka sistem harus dapat memberikan kesempatan kepada *user* untuk dapat mengoreksinya.



Gambar 3.9 Desain Tampilan Penyisipan



Gambar 3.10 Desain Tampilan Ekstraksi