BAB III

PERANCANGAN SISTEM

3.1. Analisis Sistem

Analisis sistem ini merupakan uraian dari suatu sistem perangkat lunak dan mengidentifikasi permasalahan dari kebutuhan-kebutuhan yang diharapkan sehingga dapat dipergunakan dengan baik. Analisis sistem ini memiliki tujuan dalam penerapan kriptografi dan steganografi, yaitu proses penyandian pesan teks rahasia dan penyisipan pesan ciphertext pada file image. Secara umum proses pada sistem ini ada dua, yaitu proses penyisipan pesan (encoding) menggunakan algoritma AES dan proses ekstraksi pesan(decoding) menggunakan Metode Lsb. File image dalam sistem penyisipan ini berformat BMP (*Bitmap Picture*), JPG (*Joint Photographic Group*), PNG (*Portable Network Graphics*).

3.2.1. Deskripsi Masalah

Perkembangan teknologi informasi sangat pesat dan semakin memudahkan penggunanya dalam berkomunikasi atau beraktifitas melalui media, dalam berkomunikasi akan melibatkan pengirim dan penerima pesan. Semakin pesat perkembangan teknologi semakin pesat pula pelaku kejahatan yang terjadi, salah satunya dalam pencurian data atau meretas data, untuk itu perlu adanya peningkatan kemanaan data guna menghindari perilaku kejahatan tersebut. Menghindari kejahatan peretasan dan pencurian data tersebut, data-data diamankan dalam sebuah file image melalui sebuah aplikasi yang bisa menyimpan file image yang berbeda formatnya.

3.2.2. Pemecahan Masalah

Sebagaimana pendeskripsian masalah-masalah yang telah dipaparkan diatas, maka solusi untuk menyelesaikan permasalahan tersebut yaitu dimana untuk menjaga keamanan pesan rahasia maka dibuat aplikasi *mobile* untuk mengamankan sebuah pesan rahasia dengan menysisipkan pesan kedalam gambar dan penambahan fitur format file seperti format BMP, PNG dan JPG.

3.2. Analisis Kebutuhan

3.2.1. Analisis Kebutuhan Perangkat Lunak (*Software)*

Spesifikasi perangkat Lunak minimum yang dibutuhkan dalam pembuatan Aplikasi ini adalah sebagai berikut :

a. Sistem Operasi *Windows 7*
b. Android Studio
c. Java SE *Development Kit 7*
d. AVD (*Android Virtual Device*)
e. Balsamiq Mockups *Version* 3.5.8
f. StarUML Versi 3.0
g. *Smartphone* android versi 5.0

3.2.2. Analisis Kebutuhan Perangkat Keras (*Hardware)*

Spesifikasi Perangkat keras minimum yang digunakan pada pembuatan Aplikasi ini adalah sebagai berikut :

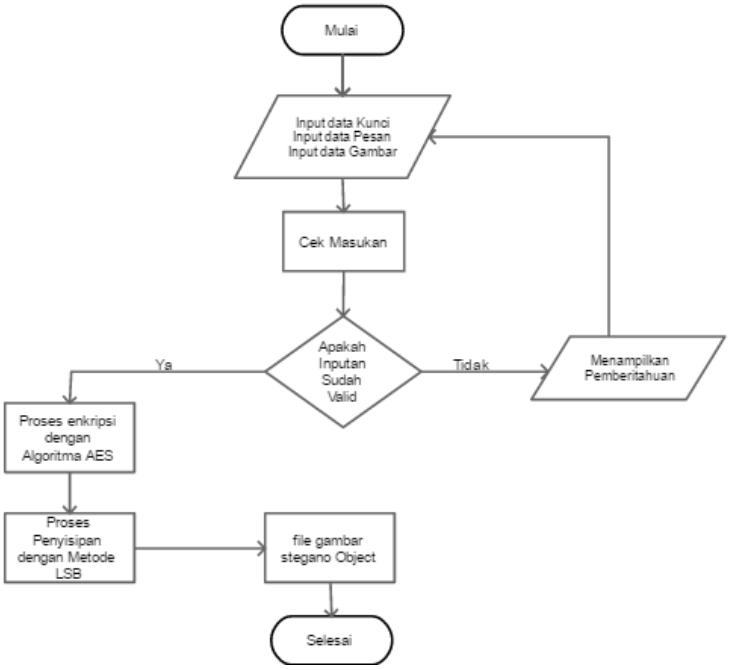
a. Processor 2.10GHz
b. Memory (RAM) 2 Gb
c. Hardisk 320 GB.

d. *Mouse, Keyboard,* dan *Monitor*

e. Ponsel *smartphone* android

3.3. Analisis Algoritma

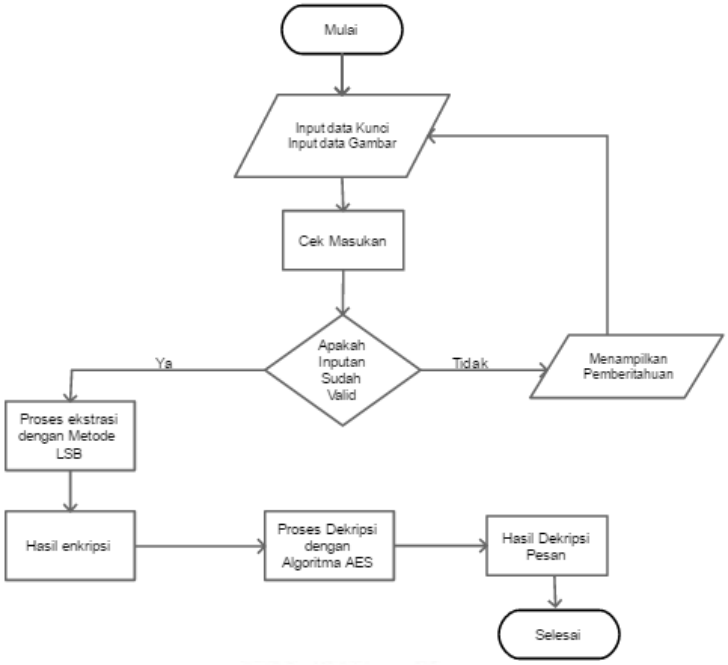
Analisis algoritma dibutuhkan untuk mengetahui kinerja algoritma secara sistematis dalam melakukan sebuah enkripsi dan dekripsi pada algoritma AES (*Advanced Encryption Standard*) 256 bit dan Teknik penyisipan pesan dengan menggunakan metode steganografi LSB. Berikut proses enkripsi dan dekripsi dengan algoritma AES (*Advanced Encryption Standard*) 256 bit yang digambarkan dengan flowchart , dapat dilihat pada Gambar 3.1 dibawah ini :



Gambar 3. 1 Flowchart Enkripsi

Gambar 3.1 merupakan flowchart enkripsi dengan algoritma AES *(Advanced Encryption Standard*) 256 bit yang mana diimplementasikan dalam *website*. Dapat dilihat pada gambar tersebut, alur dari aplikasi ini adalah pertama apabila *user* akan memulai enkripsi, *user* akan melakukan *input text, input key* dan memasukan

gambar. Artinya *input text* dan *input key* akan masuk ke sistem, kemudian akan dilakukan proses enkripsi dengan algoritma AES, setelah dienkripsi hasil tersebut akan disisipkan kedalam file gambar untuk diproses menggunakan metode LSB(*Least Significant Bit*). Kemudian gambar akan secara otomatis terunduh.



Gambar 3. 2 Flowchart Dekripsi

Kemudian untuk proses dekripsi, user akan mengimport gambar, setalah terimport, user akan diminta untuk memasukan *key*. *Key* yang dipakai akan sama dengan *key* pada waktu enkripsi. kemduain user harus melakukan *extracet* untuk memisahkan gambar dengan pesan yang telah disisipkan, setelah hasil *extracet* dilakukan dekripsi untuk pengembalian pesan yang telah dienkripsi dengan algoritma AES. Berikut ini merupakan contoh penerapan enkripsi dan dekripsi algoritma AES (*Advanced Encryption Standard*) 256 bit :

Pertama yang dilakukan adalah *key schedulling* pada proses ini merupakan pembentukan kunci yang terdiri dari empat tahap *yaitu rot word, sub word,* proses XOR dengan nilai *R-Con*, dan proses XOR dengan word sebelumnya. Pertama, masukkan key, proses dilakukan dalam heksamdesimal, dan beberapa proses membutuhkan konversi ke biner, yaitu pada proses XOR.

Input *key* : iniadalahenkripsialgoritmaaes256

Dari *key* tersebut, diubah terlebih dahulu ke bilangan heksadesimal dan dimasukkan kedalam blok-blok matrik (array) :

69	64	68	72	69	6F	6D	73
6E	61	65	69	61	72	61	32
69	6C	6E	70	66	69	61	35
61	61	6B	73	67	74	65	36

Karena AES selalu menggunakan input sepanjang 128 bit maka blok array kunci chiper tersebut dibagi menjadi 2 bagian, yang sekaligus menjadi kunci ronde ke-0 dan kunci ronde ke-1.

$$\begin{bmatrix} 69 & 64 & 68 & 72 \\ 6E & 61 & 65 & 69 \\ 69 & 6C & 6E & 70 \\ 61 & 61 & 6B & 73 \end{bmatrix} \qquad \begin{bmatrix} 69 & 6F & 6D & 73 \\ 61 & 72 & 61 & 32 \\ 66 & 69 & 61 & 35 \\ 67 & 74 & 65 & 36 \end{bmatrix}$$

Kunci Round 0 Kunci Round 1

Untuk mencari Round ke 2 digunakan transformasi RotWord atau menggeser blok paling atas keblok paling bawah pada kolom terakhir kunci Round-1

73	→	32	Kemudian hasil dari RotWord dilakukan Sub Bytes menggunakan tabel substitusi Sub Bytes
32		35	
35		36	
36		73	

32	→		x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
35		0x	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
36		1x	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
73		2x	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
		3x	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
		4x	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
		5x	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
		6x	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
		7x	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
		8x	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
		9x	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
		ax	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
		bx	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
		cx	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
		dx	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
		ex	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
		fx	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Maka hasil dari s-Box adalah

23
96
05
8F

Setelah hasil S-Box diketahui, selanjutnya hasil S-Box di XOR dengan Rcon di XOR dengan Kolom ke 1 pada Round-0.

Rcon merupakan nilai round constanta pada setiap putaran. Dan nilai setiap putaran pun berbeda-beda.

$$\begin{bmatrix}23\\96\\05\\8f\end{bmatrix} \oplus \begin{bmatrix}01\\00\\00\\00\end{bmatrix} \oplus \begin{bmatrix}69\\6e\\69\\61\end{bmatrix} = \begin{bmatrix}4b\\f8\\6c\\ee\end{bmatrix}$$

Untuk menghitungnya, bilangan heksadesimal di ubah menjadi bilangan bilangan biner 4 bit.

00100011	10010110	00000101	10001111
00000001	00000000	00000000	00000000
01101001	01101110	01101001	01100001
XOR	XOR	XOR	XOR
01001011	11111000	01101100	11101110

Maka didapat hasil Kolom 1 pada Round -2 :

4B
F8
6C
EE

Untuk mendapatkan kolom kedua dilakukan XOR antara hasil Kolom 1 dengan kolom 2 pada Round 0, begitu pun seterusnya, untuk Kolom 3 dilakukan XOR antara hasil Kolom 2 dengan kolom 3 Round 0. Dan untuk menghasilkan kolom 4 dilakukan XOR hasil kolom 3 dengan kolom 4 Round 0.

Kolom 2 pada Round – 2 :

$$\begin{bmatrix} 4b \\ f8 \\ 6c \\ ee \end{bmatrix} \oplus \begin{bmatrix} 00 \\ 00 \\ 00 \\ 00 \end{bmatrix} \oplus \begin{bmatrix} 64 \\ 61 \\ 6c \\ 61 \end{bmatrix} = \begin{bmatrix} 2f \\ 99 \\ 00 \\ 8f \end{bmatrix}$$

Kolom 3 pada Round – 2 :

$$\begin{bmatrix} 2f \\ 99 \\ 00 \\ 8f \end{bmatrix} \oplus \begin{bmatrix} 00 \\ 00 \\ 00 \\ 00 \end{bmatrix} \oplus \begin{bmatrix} 68 \\ 65 \\ 6e \\ 6b \end{bmatrix} = \begin{bmatrix} 47 \\ fc \\ 6e \\ e4 \end{bmatrix}$$

Kolom 4 pada Round – 2 :

$$\begin{bmatrix}47\\ fc\\ 6e\\ e4\end{bmatrix} \oplus \begin{bmatrix}00\\ 00\\ 00\\ 00\end{bmatrix} \oplus \begin{bmatrix}72\\ 69\\ 70\\ 73\end{bmatrix} = \begin{bmatrix}35\\ 95\\ 1e\\ 97\end{bmatrix}$$

Maka telah di dapat :

$$\begin{bmatrix}69 & 64 & 68 & 72\\ 6E & 61 & 65 & 69\\ 69 & 6C & 6E & 70\\ 61 & 61 & 6B & 73\end{bmatrix} \qquad \begin{bmatrix}69 & 6F & 6D & 73\\ 61 & 72 & 61 & 32\\ 66 & 69 & 61 & 35\\ 67 & 74 & 65 & 36\end{bmatrix} \qquad \begin{bmatrix}48 & 2F & 47 & 35\\ F8 & 99 & FC & 95\\ 6C & 00 & 6E & 1E\\ EE & 8F & E4 & 97\end{bmatrix}$$

Round 0 Round 1 Round 2

Selanjutnya, untuk mencari Round – 3 dilakukan XOR antara kolom 4 dari Round 2 dengan kolom 1 Round – 1, dan seterusnya hasil dari setiap kolom Round – 3 akan di XOR kan dengan kolom Round – 1.

$$\begin{bmatrix}35\\ 95\\ 1e\\ 97\end{bmatrix} \oplus \begin{bmatrix}69\\ 61\\ 6c\\ 67\end{bmatrix} = \begin{bmatrix}5c\\ f4\\ 72\\ f0\end{bmatrix} \qquad \begin{bmatrix}5c\\ 4f\\ 72\\ f0\end{bmatrix} \oplus \begin{bmatrix}6f\\ 72\\ 69\\ 74\end{bmatrix} = \begin{bmatrix}33\\ 86\\ 1b\\ 84\end{bmatrix}$$

Kolom 1 Kolom 2

$$\begin{bmatrix}33\\ 86\\ 1b\\ 84\end{bmatrix} \oplus \begin{bmatrix}6d\\ 61\\ 61\\ 65\end{bmatrix} = \begin{bmatrix}5e\\ e7\\ 7a\\ e1\end{bmatrix} \qquad \begin{bmatrix}5e\\ e7\\ 7a\\ e1\end{bmatrix} \oplus \begin{bmatrix}73\\ 32\\ 35\\ 36\end{bmatrix} = \begin{bmatrix}2d\\ d5\\ 4f\\ d7\end{bmatrix}$$

Kolom 3 Kolom 4

Round – 3 didapatkan hasil :

$$\begin{bmatrix}5C & 33 & 5E & 2D\\ F4 & 86 & E7 & D5\\ 72 & 1B & 74 & 4F\\ F0 & 84 & E1 & D7\end{bmatrix}$$

Maka hasil Round :

$$\begin{bmatrix}69 & 64 & 68 & 72\\ 6E & 61 & 65 & 69\\ 69 & 6C & 6E & 70\\ 61 & 61 & 6B & 73\end{bmatrix} \qquad \begin{bmatrix}69 & 6F & 6D & 73\\ 61 & 72 & 61 & 32\\ 66 & 69 & 61 & 35\\ 67 & 74 & 65 & 36\end{bmatrix} \qquad \begin{bmatrix}48 & 2F & 47 & 35\\ F8 & 99 & FC & 95\\ 6C & 00 & 6E & 1E\\ EE & 8F & E4 & 97\end{bmatrix}$$

Round 0 Round 1 Round 2

$$\begin{bmatrix} 5C & 33 & 5E & 2D \\ F4 & 86 & E7 & D5 \\ 72 & 1B & 74 & 4F \\ F0 & 84 & E1 & D7 \end{bmatrix}$$

Round 3

Pada proses pencarian kunci Round – 4 diilakukan seperti pada proses pencarian kunci Round – 2, yaitu menggunakan transformasi RotWrod, SubBytes, kemudian XOR dengan tabel Rcon kolom ke 2 .

Transformasi RotWord terjadi pada kolom ke 4 kunci Round – 3, kemudian diSubBytes dengan tabel substitusi, dan terakhir adalah proses XOR yaitu hasil SubBytes di-XOR dengan kolom ke-1 kunci Round – 2 dan kolom ke 2 tabel Rcon, sehingga akan didapatkan kolom ke 1 kunci Round – 4 :

$$\begin{bmatrix} 2D \\ D5 \\ 4F \\ D7 \end{bmatrix} \longrightarrow \begin{bmatrix} D5 \\ 4F \\ D7 \\ 2D \end{bmatrix} \longrightarrow \begin{bmatrix} 03 \\ 84 \\ 0E \\ D8 \end{bmatrix} \oplus \begin{bmatrix} 48 \\ F8 \\ 6C \\ EE \end{bmatrix} \oplus$$

01	02	04	08	10	20	40
00	00	00	00	00	00	00
00	00	00	00	00	00	00
00	00	00	00	00	00	00

Maka hasil Round 4 kolom 1 : $\begin{bmatrix} 49 \\ 7C \\ 62 \\ 36 \end{bmatrix}$

dan untuk kolom selanjutnya dilakukan XOR dari hasil kolom sebelumnya dengan Round – 2. Maka hasil dari Round – 4 :

$$\begin{bmatrix} 49 & 66 & 20 & 15 \\ 7C & E5 & 19 & 8C \\ 62 & 62 & 0C & 12 \\ 36 & B9 & 5D & CA \end{bmatrix}$$

Selanjutnya untuk Round – 5 dilakukan XOR kolom 4 Round 4 dengan Round – 3, caranya sama seperti pada mencari Round – 3.

Untuk pencarian kunci Round 6,8,10,12, dan 14 menggunakan tabel Rcon kolom ke 3,4,5,6,dan ke 7. Caranya seperti pada mencari Round ke 2 dan 4. Dan untuk mencari Round – 7,9,11, dan 13 dilakukan cara XOR, seperti yang dilakukan untuk mencari Round 3 dan 5.

Maka hasil ke 14 Round dari chiperkey iniadalahenkripsialgoritmaaes256 adalah sebagai berikut :

$$\begin{bmatrix} 69 & 64 & 68 & 72 \\ 6E & 61 & 65 & 69 \\ 69 & 6C & 6E & 70 \\ 61 & 61 & 6B & 73 \end{bmatrix}$$

Round 0

$$\begin{bmatrix} 69 & 6F & 6D & 73 \\ 61 & 72 & 61 & 32 \\ 66 & 69 & 61 & 35 \\ 67 & 74 & 65 & 36 \end{bmatrix}$$

Round 1

$$\begin{bmatrix} 48 & 2F & 47 & 35 \\ F8 & 99 & FC & 95 \\ 6C & 00 & 6E & 1E \\ EE & 8F & E4 & 97 \end{bmatrix}$$

Round 2

$$\begin{bmatrix} 5C & 33 & 5E & 2D \\ F4 & 86 & E7 & D5 \\ 72 & 1B & 74 & 4F \\ F0 & 84 & E1 & D7 \end{bmatrix}$$

Round 3

$$\begin{bmatrix} 49 & 66 & 20 & 15 \\ 7C & E5 & 19 & 8C \\ 62 & 62 & 0C & 12 \\ 36 & B9 & 5D & CA \end{bmatrix}$$

Round 4

$$\begin{bmatrix} 49 & 7A & 24 & 09 \\ 78 & FE & 19 & CC \\ 60 & 7B & 0F & 40 \\ 3A & BE & 5F & 88 \end{bmatrix}$$

Round 5

$$\begin{bmatrix} 42 & 24 & 04 & 11 \\ 75 & 70 & 89 & 05 \\ A6 & C4 & C8 & DA \\ 37 & 8E & D3 & 19 \end{bmatrix}$$

Round 6

$$\begin{bmatrix} 58 & 22 & 06 & 0F \\ 7D & 83 & 9A & 56 \\ BA & C1 & CE & 8E \\ 23 & 9D & C2 & 4A \end{bmatrix}$$

Round 7

$$\begin{bmatrix} FB & DF & DB & CA \\ 6C & FC & 75 & 70 \\ 70 & B4 & 7C & A6 \\ 41 & CF & 1C & 05 \end{bmatrix}$$

Round 8

$$\begin{bmatrix} 92 & B0 & B6 & B9 \\ 0D & 8E & 14 & 42 \\ 1C & DD & 13 & 9D \\ 26 & BB & 79 & 33 \end{bmatrix}$$

Round 9

$$\begin{bmatrix} C7 & 18 & C3 & 09 \\ 32 & CE & BB & CB \\ B3 & 07 & 7B & DD \\ 17 & D8 & C4 & C1 \end{bmatrix}$$

Round 10

$$\begin{bmatrix} 9B & 2B & 9D & 24 \\ C6 & 48 & 5C & 1E \\ C1 & 1C & 0F & 92 \\ E7 & 5C & 25 & 16 \end{bmatrix}$$

Round 11

$$\begin{bmatrix} 95 & 8D & 4E & 47 \\ 7D & B3 & 08 & C3 \\ F4 & F3 & 88 & 55 \\ 21 & F9 & 3D & FC \end{bmatrix} \quad \begin{bmatrix} DC & F7 & 6A & 4E \\ 05 & 4D & 87 & 0F \\ 94 & 88 & 11 & 15 \\ 1B & 47 & 62 & 74 \end{bmatrix} \quad \begin{bmatrix} A3 & 2E & 60 & 27 \\ 24 & 97 & 9F & 5C \\ 66 & 95 & 1D & 48 \\ 0E & F7 & CA & 36 \end{bmatrix}$$

Round 12 Round 13 Round 14

Proses Enkripsi

Proses enkripsi AES terdiri dari 4 transformasi, yaitu AddRoundKey, SubBytes, ShiftRows, dan MixColoums. Input berupa teks diubah menjadi data-data digital dan direpresentasikan kedalam bentuk matriks. Algoritma AES beroperasi pada blok chiper, sehingga matriks akan dipartisi menjadi beberapa blok matriks input sebesar 256 bit dan elemennya direpresentasikan kedalam bentuk heksadesimal. Cara untuk pengambilan blok dengan cara menyusun seluruh elemen matriks plaintext menjadi array dan membaginya menjadi beberapa partisi 256 bit. Jika pada akhir partisi tidak mencapai 256 bit, maka akan ditambahkan dummy berupa elemen 0 sampai jumlah menjadi 256 bit. Berikut contoh perhitungan dari input teks yang berisi "akuadalahkriptografiblalabla" , maka akan diubah menjadi data-data digital dan di binerkan menjadi :

61	64	68	70	72	00	00	00
6b	61	6b	74	61	00	00	00
75	69	72	6f	66	00	00	00
61	61	69	67	69	00	00	00

Setelah diurutkan menjadi array, akan dipartisi menjadi beberapa blok dengan panjang 16 byte, dengan cara pengambilan 4 byte pertama menjadi kolom pertama, 4 byte kedua menjadi kolom kedua dan seterusnya. Matriks input pertama merupakan 4 kolom pertama yang direpresentasikan dalam heksadesimal.

Blok 16 byte pertama dalam heksadesimal :

61	64	68	70
6b	61	6b	74
75	69	72	6f
61	61	69	67

Chiperkey :

69	64	68	72
6E	61	65	69
69	6C	6E	70
61	61	6B	73

Chiperkey didapat pada Round-0 atau pra enkripsi

1) Transformasi *AddRoundKey*

Untuk melakukan *AddRoundKey* dilakukan dengan mengXORkan input teks pada 16 byte pertama dengan *chiperkey* :

$$\begin{bmatrix}61\\6B\\75\\61\end{bmatrix} \oplus \begin{bmatrix}69\\6E\\69\\61\end{bmatrix} = \begin{bmatrix}08\\05\\8C\\00\end{bmatrix} \qquad \begin{bmatrix}64\\61\\69\\61\end{bmatrix} \oplus \begin{bmatrix}64\\61\\6C\\61\end{bmatrix} = \begin{bmatrix}00\\00\\05\\00\end{bmatrix}$$

$$\begin{bmatrix}68\\6B\\72\\69\end{bmatrix} \oplus \begin{bmatrix}68\\65\\6E\\6B\end{bmatrix} = \begin{bmatrix}00\\0E\\1C\\02\end{bmatrix} \qquad \begin{bmatrix}70\\74\\6F\\67\end{bmatrix} \oplus \begin{bmatrix}72\\69\\70\\73\end{bmatrix} = \begin{bmatrix}02\\1D\\1F\\14\end{bmatrix}$$

Maka hasil Transformasi *AddRoundKey* adalah :

$$\begin{bmatrix}08 & 00 & 00 & 02\\05 & 00 & 0E & 1D\\8C & 05 & 1C & 1F\\00 & 00 & 02 & 14\end{bmatrix}$$

Enkripsi dilakukan sampai dengan Round 14, memasuki iterasi pertama dari 14 iterasi dengan 9 iterasi pertama terdiri dari *transformasi sub bytes, shift rows, mix columns,* dan *add roundkey*. Namun pada Round – 14 dilakukan tanpa MixColumns.

2) Sub-Bytes

Sub Bytes dilakukan dengan cara mencocokkan hasil dari Transformasi *AddRoundKey* dengan Tabel substitusi

$$\begin{bmatrix}08 & 00 & 00 & 02\\05 & 00 & 0E & 1D\\8C & 05 & 1C & 1F\\00 & 00 & 02 & 14\end{bmatrix} \longrightarrow \begin{bmatrix}30 & 63 & 63 & 77\\6B & 63 & AB & A4\\64 & 6B & 9C & C0\\63 & 63 & 77 & FA\end{bmatrix}$$

3) Shift Rows

ShiftRows dilakukan dengan menggeser pada setiap baris matriks,

$$\begin{bmatrix}30 & 63 & 63 & 77\\6B & 63 & AB & A4\\64 & 6B & 9C & C0\\63 & 63 & 77 & FA\end{bmatrix} \longrightarrow \begin{bmatrix}30 & 63 & 63 & 77\\63 & AB & A4 & 6B\\9C & C0 & 64 & 6B\\FA & 63 & 63 & 77\end{bmatrix}$$

4) Mix Columns

MixColumns pada kolom ke satu

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} 30 \\ 63 \\ 9C \\ FA \end{bmatrix} = \begin{bmatrix} A3 \\ A8 \\ 65 \\ 55 \end{bmatrix}$$

b1,1 = $(\{02\}\cdot 30)\oplus(\{03\}\cdot 63)\oplus 9C\oplus FA = A3$

b2,1 = $30\oplus(\{02\}\cdot 63)\oplus(\{03\}\cdot 9C)\oplus FA = A8$

b3,1 = $30\oplus 63\oplus(\{02\}\cdot 9C)\oplus(\{03\}\cdot FA) = 65$

b4,1 = $(\{03\}\cdot 30)\oplus 63\oplus 9C\oplus(\{02\}\cdot FA) = 55$

MixColumns pada kolom ke dua

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} 63 \\ AB \\ C0 \\ 63 \end{bmatrix} = \begin{bmatrix} 98 \\ 16 \\ EC \\ 08 \end{bmatrix}$$

b1,2 = $(\{02\}\cdot 63)\oplus(\{03\}\cdot AB)\oplus C0\oplus 63 = 98$

b2,2 = $63\oplus(\{02\}\cdot AB)\oplus(\{03\}\cdot C0)\oplus 63 = 16$

b3,2 = $63\oplus AB\oplus(\{02\}\cdot C0)\oplus(\{03\}\cdot 63) = EC$

b4,2 = $(\{03\}\cdot 63)\oplus AB\oplus C0\oplus(\{02\}\cdot 63) = 08$

MixColumns pada kolom ke tiga

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} 63 \\ A4 \\ 64 \\ 63 \end{bmatrix} = \begin{bmatrix} 2D \\ E4 \\ AA \\ A3 \end{bmatrix}$$

b1,3 = $(\{02\}\cdot 63)\oplus(\{03\}\cdot A4)\oplus 64\oplus FA = 2D$

b2,3 = $63\oplus(\{02\}\cdot A4)\oplus(\{03\}\cdot 64)\oplus FA = E4$

b3,3 = $63\oplus A4\oplus(\{02\}\cdot 64)\oplus(\{03\}\cdot 63) = AA$

b4,3 = $(\{03\}\cdot 63)\oplus A4\ \oplus 64\oplus(\{02\}\cdot 63) = A3$

MixColumns pada kolom ke empat

$$\begin{bmatrix}02 & 03 & 01 & 01\\ 01 & 02 & 03 & 01\\ 01 & 01 & 02 & 03\\ 03 & 01 & 01 & 02\end{bmatrix}\begin{bmatrix}77\\ 6B\\ 6\mathrm{B}\\ 77\end{bmatrix} = \begin{bmatrix}4F\\ 6B\\ 53\\ 77\end{bmatrix}$$

b1,4 = $(\{02\}\cdot 77)\oplus(\{03\}\cdot 6B)\oplus 53\oplus 77 = 4F$

b2,4 = $77\oplus(\{02\}\cdot 6B)\oplus(\{03\}\cdot 6B)\oplus 77 = 6B$

b3,4 = $77\oplus 6B\oplus(\{02\}\cdot 6B)\oplus(\{03\}\cdot 77) = 53$

b4,4 = $(\{03\}\cdot 77)\oplus 6B\ \oplus 6B\oplus(\{02\}\cdot 77) = 77$

Maka hasil dari *MixColumns* = $\begin{bmatrix}02 & 03 & 01 & 01\\ 01 & 02 & 03 & 01\\ 01 & 01 & 02 & 03\\ 03 & 01 & 01 & 02\end{bmatrix}\begin{bmatrix}30 & 63 & 63 & 77\\ 63 & AB & A4 & 6B\\ 9C & C0 & 64 & 66\\ FA & 63 & 63 & 77\end{bmatrix}$

$$=\begin{bmatrix}A3 & 98 & 2D & 4F\\ A8 & 16 & E4 & 6B\\ 65 & FC & AA & 53\\ 55 & 08 & A3 & 77\end{bmatrix}$$

AddRoundKey dengan Round – 1, maka hasilnya :

$$\begin{bmatrix} A3 & 98 & 2D & 4F \\ A8 & 16 & E4 & 6B \\ 65 & FC & AA & 53 \\ 55 & 08 & A3 & 77 \end{bmatrix} \oplus \begin{bmatrix} 69 & 67 & 6D & 73 \\ 61 & 72 & 61 & 32 \\ 6C & 69 & 61 & 35 \\ 67 & 74 & 65 & 36 \end{bmatrix} = \begin{bmatrix} C4 & FF & 40 & 3C \\ C9 & 64 & 85 & 59 \\ 09 & 95 & CC & 66 \\ 32 & 7C & C6 & 41 \end{bmatrix}$$

SubBytes : $\begin{bmatrix} 74 & 16 & 09 & EB \\ DD & 43 & 97 & CB \\ 01 & 2A & 4B & 33 \\ 23 & 10 & B4 & 83 \end{bmatrix}$

ShiftRows : $\begin{bmatrix} 74 & 16 & 09 & EB \\ 43 & 97 & CB & DD \\ 4B & 33 & 01 & 2A \\ 83 & 23 & 10 & B4 \end{bmatrix}$

MixColumns : $\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} 74 & 16 & 09 & EB \\ 43 & 97 & CB & DD \\ 4B & 33 & 01 & 2A \\ 83 & 23 & 10 & B4 \end{bmatrix} = \begin{bmatrix} E5 & 9E & 45 & 2F \\ AF & 55 & 97 & 80 \\ 3F & 86 & F0 & A5 \\ 89 & D8 & F1 & A2 \end{bmatrix}$

AddRoundKey dengan Round – 2 :

$$\begin{bmatrix} E5 & 9E & 45 & 2F \\ AF & 55 & 97 & 80 \\ 3F & 86 & F0 & A5 \\ 89 & D8 & F1 & A2 \end{bmatrix} \oplus \begin{bmatrix} 48 & 2F & 47 & 35 \\ F8 & 99 & FC & 95 \\ 6C & 00 & 6E & 1E \\ EE & 8F & E4 & 97 \end{bmatrix} = \begin{bmatrix} AD & B1 & 02 & 1A \\ 57 & CC & 6B & 15 \\ 53 & 82 & 9E & BB \\ 65 & 57 & 15 & C5 \end{bmatrix}$$

Kemudian dilanjutkan pada *SubBytes, ShiftRows, MixColumns* sampai dengan iterasi ke 13, pada Round 14 dilakukan tanpa melakukan *MixColumns* dan langsung pada *AddRoundKey, MixColumns* dilakukan sesuai dengan putaran dengan sub kunci yang sesuai. Maka akan didapat hasil enkripsi yaitu *chipertext.*

Proses Embedding/ Penyisipan

Hasil dari enkirsi yang akan disisipkan kedalam sebuah media gambar dengan panjang 8 bit.

a. Representasikan kedalam binary

Hexadecimal	*Binary*
AD	10101101
57	01010111
53	01010011
65	01100101
B1	10110001
CC	11001100
82	10000011
57	01010111
02	00000010
6B	01101011
9E	10011110
15	00010101
1A	00011010
15	00010101
BB	10111011
C5	11000101

Sesuai dengan namanya, *Least Significant Bit* artinya bit yang tidak *significant* atau tidak mempunyai pengaruh yang besar, maka metode ini mengganti nilai bit ke-8 untuk menyisipkan data.

b. Media

00000000	00000000	00000001	00000001	00000001	00000001	00000001	00000001
00000000	00000000	00000001	00000001	00000001	00000001	00000001	00000001
00000000	00000000	00000001	00000001	00000001	00000001	00000001	00000001
00000001	00000001	00000010	00000010	00000010	00000011	00000011	00000011
00000001	00000001	00000010	00000010	00000010	00000011	00000011	00000011
00000001	00000001	00000010	00000010	00000010	00000011	00000011	00000011
00000000	00000000	00000001	00000001	00000001	00000001	00000001	00000001
00000000	00000000	00000001	00000001	00000001	00000001	00000001	00000001
00000000	00000000	00000001	00000001	00000001	00000001	00000001	00000001
00000001	00000001	00000010	00000010	00000010	00000011	00000011	00000011
00000001	00000001	00000010	00000010	00000010	00000011	00000011	00000011
00000001	00000001	00000010	00000010	00000010	00000011	00000011	00000011
00000000	00000000	00000001	00000001	00000001	00000001	00000001	00000001
00000000	00000000	00000001	00000001	00000001	00000001	00000001	00000001
00000000	00000000	00000001	00000001	00000001	00000001	00000001	00000001
00000001	00000001	00000010	00000010	00000010	00000011	00000011	00000011

c. Data yang ingin disisipkan

1	0	1	0	1	1	0	1
0	1	0	1	0	1	1	1
0	1	0	1	0	0	1	1
0	1	1	0	0	1	0	1
1	0	1	1	0	0	0	1
1	1	0	0	1	1	0	0
1	0	0	0	0	0	1	1
0	1	0	1	0	1	1	1
0	0	0	0	0	0	1	0
0	1	1	0	1	0	1	1
1	0	0	1	1	1	1	0
0	0	0	1	0	1	0	1
0	0	0	1	1	0	1	0
0	0	0	1	0	1	0	1
1	0	1	1	1	0	1	1
1	1	0	0	0	1	0	1

d. Hasil Akhir Steganografi

00000001	00000000	00000001	00000000	00000001	00000001	00000000	00000001
00000000	00000001	00000000	00000001	00000000	00000001	00000001	00000001
00000000	00000001	00000000	00000001	00000000	00000000	00000001	00000001
00000000	00000001	00000011	00000010	00000010	00000011	00000010	00000011
00000001	00000000	00000011	00000011	00000010	00000010	00000010	00000011
00000001	00000001	00000010	00000010	00000011	00000011	00000010	00000010
00000001	00000000	00000000	00000000	00000000	00000000	00000001	00000001
00000000	00000001	00000000	00000001	00000000	00000001	00000001	00000001
00000000	00000000	00000000	00000000	00000000	00000000	00000001	00000000
00000000	00000001	00000011	00000010	00000011	00000010	00000011	00000011
00000001	00000000	00000010	00000011	00000011	00000011	00000011	00000010
00000000	00000000	00000010	00000011	00000010	00000011	00000010	00000011
00000000	00000000	00000000	00000001	00000001	00000000	00000001	00000000
00000000	00000000	00000000	00000001	00000000	00000001	00000000	00000001
00000001	00000000	00000001	00000001	00000001	00000000	00000001	00000001
00000001	00000001	00000010	00000010	00000010	00000011	00000010	00000011

Angka yang di *bold* menunjukan bahwa data tersebut sudah diganti sesuai dengan data yang ingin disisipkan.

Proses Ekstraksi/ *Extract Text*

a. Membaca data berupa *byte-byte* dari sebuah *file* gambar yang telah berisikan pesan rahasia.

Byte image yang berisi pesan rahasia

00000001	00000000	00000001	00000000	00000001	00000001	00000000	00000001
00000000	00000001	00000000	00000001	00000000	00000001	00000001	00000001
00000000	00000001	00000000	00000001	00000000	00000000	00000001	00000001
00000000	00000001	00000011	00000010	00000010	00000011	00000010	00000011
00000001	00000000	00000011	00000011	00000010	00000010	00000010	00000011
00000001	00000001	00000010	00000010	00000011	00000011	00000010	00000010
00000001	00000000	00000000	00000000	00000000	00000000	00000001	00000001
00000000	00000001	00000000	00000001	00000000	00000001	00000001	00000001
00000000	00000000	00000000	00000000	00000000	00000000	00000001	00000000
00000000	00000001	00000011	00000010	00000011	00000010	00000011	00000011
00000001	00000000	00000010	00000011	00000011	00000011	00000011	00000010
00000000	00000000	00000010	00000011	00000010	00000011	00000010	00000011
00000000	00000000	00000000	00000001	00000001	00000000	00000001	00000000
00000000	00000000	00000000	00000001	00000000	00000001	00000000	00000001
00000001	00000000	00000001	00000001	00000001	00000000	00000001	00000001
00000001	00000001	00000010	00000010	00000010	00000011	00000010	00000011

b. Melakukan proses ektstraksi dengan cara membaca bit terendah dari setiap byte-byte file gambar, kemudian gabungkan kembali bit-bit hasil dari proses ektraksi tersebut.

Ekstraksi								Hasil
00000001	00000000	00000001	00000000	00000001	00000001	00000000	00000001	10101101
00000000	00000001	00000000	00000001	00000000	00000001	00000001	00000001	01010111
00000000	00000001	00000000	00000001	00000000	00000000	00000001	00000001	01010011
00000000	00000001	00000011	00000010	00000010	00000011	00000010	00000011	01100101
00000001	00000000	00000011	00000011	00000010	00000010	00000010	00000011	10110001
00000001	00000001	00000010	00000010	00000011	00000011	00000010	00000010	11001100
00000001	00000000	00000000	00000000	00000000	00000000	00000001	00000001	10000011
00000000	00000001	00000000	00000001	00000000	00000001	00000001	00000001	01010111
00000000	00000000	00000000	00000000	00000000	00000000	00000001	00000000	00000010
00000000	00000001	00000011	00000010	00000011	00000010	00000011	00000011	01101011
00000001	00000000	00000010	00000011	00000011	00000011	00000011	00000010	10011110
00000000	00000000	00000010	00000011	00000010	00000011	00000010	00000011	00010101
00000000	00000000	00000000	00000001	00000001	00000000	00000001	00000000	00011010
00000000	00000000	00000000	00000001	00000000	00000001	00000000	00000001	00010101
00000001	00000000	00000001	00000001	00000001	00000000	00000001	00000001	10111011
00000001	00000001	00000010	00000010	00000010	00000011	00000010	00000011	11000101

c. Ubah ke bentuk *Hexadecimal*

Hasil	*Hexadecimal*
10101101	AD
01010111	57
01010011	53
01100101	65
10110001	B1
11001100	CC
10000011	82
01010111	57
00000010	02
01101011	6B
10011110	9E
00010101	15
00011010	1A
00010101	15
10111011	BB
11000101	C5

Proses Perhitungan Dekripsi

Pada dekripsi dilakukan pada hasil enkripsi dengan cara *AddRoundKey* pada hasil enkripsi dengan Round 14. Karena proses dekripsi ini kebalikan dari proses enkripsi. Kemudian masuk ke 4 transformasi, yaitu dengan menggunakan *invers, invers SubBytes, Invers ShiftRows, Invers MixColumns, dan AddRoundKey*, dilakukan sampai 13 putaran menggunakan sub kunci R-13 sampai R-1.

Masuk pada proses pertama yaitu dengan *AddRoundKey* hasil chipertext enkripsi dengan Round 14, kemudian masuk ke *ShiftRows*, dan masuk ke *SubBytes*, setelah didapatkan hasilnya di *AddRoundKey* dari Round 13 sampai Round 1 dengan melakukan transformasi *AddRoundKey, InversMixColumns, Invers ShiftRows, Invers SubBytes*. Dan terakhir hanya dilakukan *AddRoundKey* dengan Round 0. Maka akan dapat didengarkan kembali.

$$\begin{bmatrix} AD & B1 & 02 & 1A \\ 57 & CC & 6B & 15 \\ 53 & 82 & 9E & BB \\ 65 & 57 & 15 & C5 \end{bmatrix} \oplus \begin{bmatrix} A3 & 2E & 60 & 27 \\ 24 & 97 & 9F & 5C \\ 66 & 95 & 1D & 48 \\ 0E & F7 & CA & 36 \end{bmatrix} = \begin{bmatrix} 0E & 9F & 62 & 3D \\ 73 & 5B & F4 & 49 \\ 35 & 17 & 83 & F3 \\ 6B & A0 & DF & F3 \end{bmatrix}$$

Masuk ke transformasi *Invers ShiftRows* :

$$\begin{bmatrix} 0E & 9F & 62 & 3D \\ 73 & 5B & F4 & 49 \\ 35 & 17 & 83 & F3 \\ 6B & A0 & DF & F3 \end{bmatrix} \longrightarrow \begin{bmatrix} 0E & 9F & 62 & 3D \\ 5B & F4 & 49 & 73 \\ 83 & F3 & 35 & 17 \\ F3 & 6B & A0 & DF \end{bmatrix}$$

Masuk ke Transformasi *Invers SubBytes* :

$$\begin{bmatrix} 0E & 9F & 62 & 3D \\ 5B & F4 & 49 & 73 \\ 83 & F3 & 35 & 17 \\ F3 & 6B & A0 & DF \end{bmatrix} \longrightarrow \begin{bmatrix} AB & DB & AA & 27 \\ 39 & BF & 3B & 8F \\ EC & 0D & 96 & F0 \\ 0D & 7F & E0 & 9E \end{bmatrix}$$

Kemudian melakukan *AddRoundKey* dari Round 13 sampai Round 1 dengan tahapan *AddRoundKey, Invers MixColumns, Invers ShiftRows, dan Invers SubBytes* sampai putaran 13 kali.

$$\begin{bmatrix} AB & DB & AA & 27 \\ 39 & BF & 3B & 8F \\ EC & 0D & 96 & F0 \\ 0D & 7F & E0 & 9E \end{bmatrix} \oplus \begin{bmatrix} DF & F7 & 6A & 4E \\ 05 & 4D & 11 & 0F \\ 94 & 88 & 87 & 15 \\ 1B & 4F & 62 & 74 \end{bmatrix} = \begin{bmatrix} 74 & 2C & C0 & 69 \\ 0C & F2 & 2A & 80 \\ 78 & 85 & 11 & E5 \\ 16 & 30 & 82 & EA \end{bmatrix}$$

Invers MixColumns :

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \oplus \begin{bmatrix} 74 & 2C & C0 & 69 \\ 0C & F2 & 2A & 80 \\ 78 & 85 & 11 & E5 \\ 16 & 30 & 82 & EA \end{bmatrix} = \begin{bmatrix} 12 & 07 & FA & 27 \\ 2A & 27 & 56 & 6E \\ 3A & 78 & DB & 62 \\ 1C & 33 & 0C & D3 \end{bmatrix}$$

Invers ShiftRows : $\begin{bmatrix} 12 & 07 & FA & 27 \\ 27 & 56 & 6e & 2a \\ DB & 62 & 3A & 78 \\ D3 & 1C & 33 & 0C \end{bmatrix}$

Invers SubBytes : $\begin{bmatrix} C9 & C5 & 2D & CC \\ CC & B1 & 9F & E5 \\ B9 & AA & 80 & BC \\ 66 & 9C & C3 & FE \end{bmatrix}$

Kemudian dilakukan AddRoundKey kembali dari Round-12 dan transformasi *Invers MixColumns, Invers ShiftRows, Invers SubBytes* sampai dengan Round – 1 sebanyak 13 kali putaran. Dan pada putaran terakhir atau Round 14 hanya dilakukan AddRoundkKey saja dengan Round 0. Maka hasil dekripsi adalah :

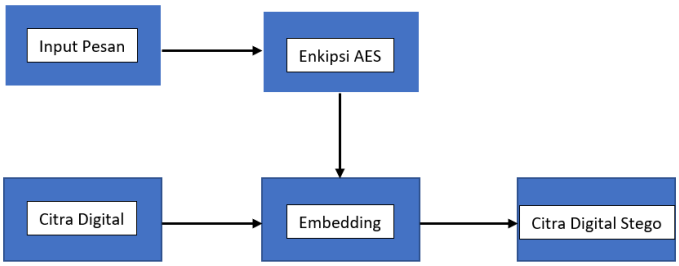
61	64	68	70
6b	61	6b	74
75	69	72	6f
61	61	69	67

3.4. Arsitektur Sistem

Aplikasi yang dibuat merupakan aplikasi yang memperoses penyisipan (*embedding*) dengan enkripsi, dekripsi dan pengekstrakan sebuah pesan atau teks kedalam sebuah objek citra dengan format JPG, PNG dan BMP.

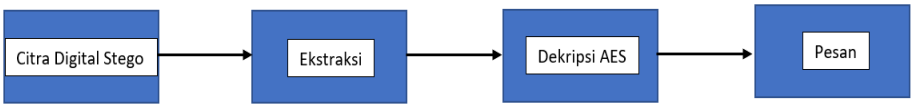
2. Proses Penyisipan (*embedding*) Pesan

   Ada dua tahap yang dilakukan dalam proses embedding, yaitu enkripsi aes yang mentransformasikan pesan asli (*plaintext*) menjadi teks acak (*ciphertext*), selanjutnya dilakukan penyisipan (*embedding*) dalam citra digital. Proses embedding dapat dilihat pada gambar 3.3



3. Proses Ekstraksi Pesan

   Ada dua tahap yang dilakukan dalam proses ekstraksi, yaitu ekstraksi yang memisahkan antara citra digital stego dengan ciphertext dan dekripsi aes mentransformasikan ciphertext menjadi plaintext.



Gambar 3. 4 Proses Ekstraksi

3.5. Pemodelan Sistem

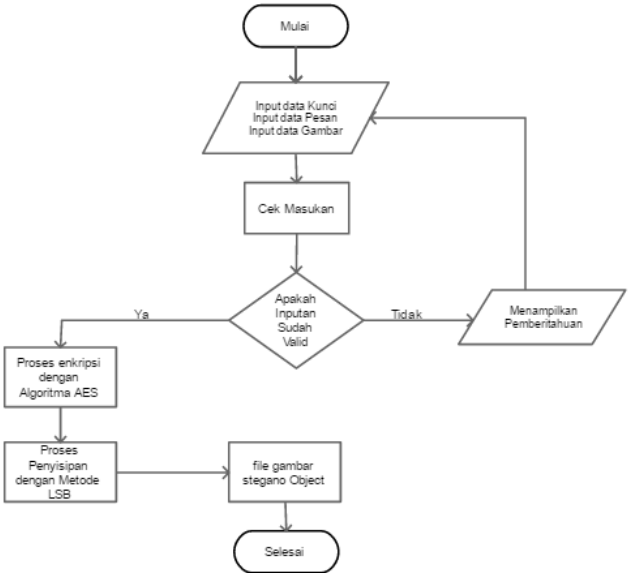
Pemodelan merupakan proses dari perancangan perangkat lunak yang akan dibuat sebelum melakukan pengkodean. Pemodelan yang digunakan ada dua metode, yaitu metode Prosedural dengan menggunakan *flowchart* dan metode berbasis objek dengan menggunakan metode UML (*Unified Modeling Language*).

3.4.1. *Flowchart*

Flowchart menggambarkan proses yang terjadi pada aplikasi secara keseluruhan. Proses ini digambarkan secara runtun dari awal proses perancangan.

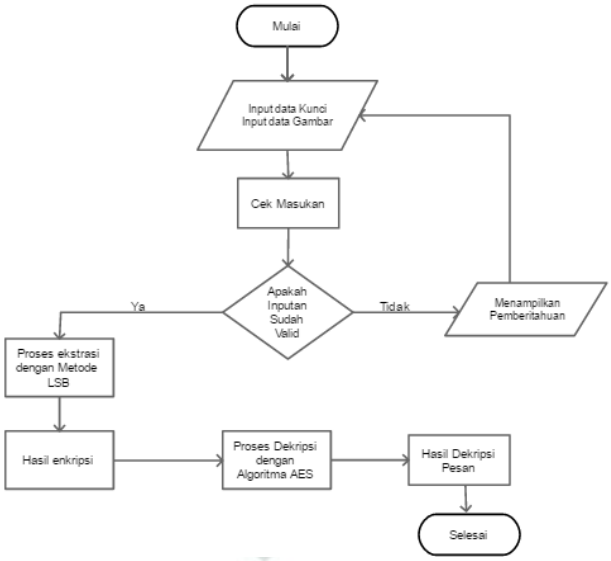
4. *Flowchart* Penyisipan (*encoding*)

Proses penyisipan dilakukan oleh user, dengan cara memasukan pesan teks, gambar, kunci (*key*) berupa teks dan lokasi penyimpanan file hasil penyisipan. Setalah itu sistem akan melakukan proses penyisipan yang telah dimasukkan oleh user. Adapun flowchart penyisipan (*encoding*) dapat dilihat pada gambar 3.5



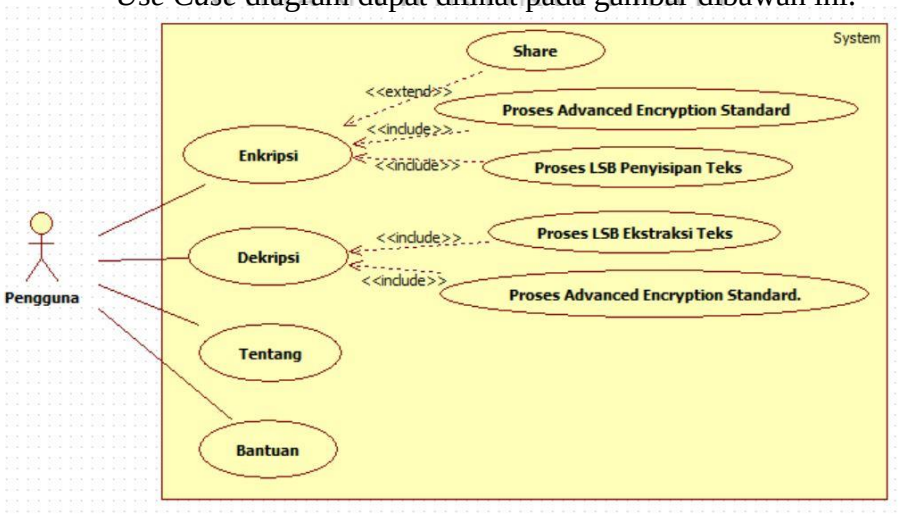
5. *Flowchart* Pengekstrakan (*decoding*)

Proses pengekstrakan dilakukan oleh pengguna, dengan cara memasukan gambar dan kunci (key) berupa teks. Setelah itu sistem akan melakukan proses pengekstarkan yang telah dimasukan oleh pengguna.Adapun *Flowchart* pengekstarkan dapat dilihat pada Gambar 3.6



3.4.2. *Use Case* Diagram

Use Case diagram dapat dilihat pada gambar dibawah ini.



Gambar 3. 7 Use case diagram

Sekenario Usecase Diagram diatas dapat didekripsikan sebagai berikut.

1) Definisi Aktor

Tabel 3. 1 Definisi Aktor

No	Aktor	Deskripsi
1	Pengguna	Pengguna bisa berfungsi sebagai pengirim ataupun penerima pesan tergantung situasi.

2) **Spesifikasi *Use Case***

a. Skenario *Use Case* Proses Enkripsi

Tabel 3. 2 Skenario Use Case Proses Enkripsi

Nomor	UCD-1
Use case	Proses Enkripsi
Aktor	Pengguna
Deskripsi	Merupakan proses masukan dimana pengguna yakni dianalogikan sebagai pengirim menginputkan kunci dan pesan untuk untuk melakukan proses enkripsi dengan algoritma AES. Pengguna memilih gambar untuk melakukan proses *Embedd/*penyisipan pesan yang sudah terenkripsi kedalam gambar menggunakan metode *LSB* Hasil gambar tersebut dapat dishare oleh pengguna.
Kondisi Awal	Tampil *Button* Enkripsi

b. Skenario *Use case* Proses Dekripsi

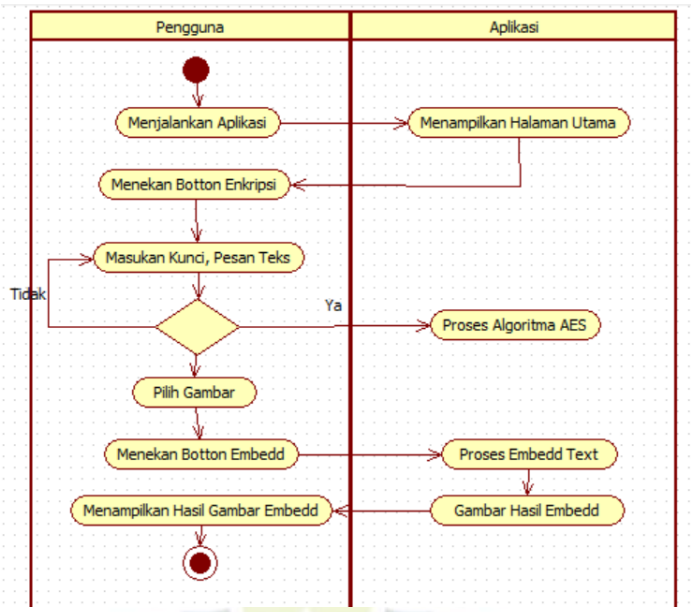
Tabel 3. 3 Skenario Use case Proses Dekripsi.

Nomor	UCD-2
Use case	Proses Dekripsi
Aktor	Pengguna
Deskripsi	Dimana dekripsi merupakan proses mengembalikan cipherteks menjadi plainteks semula, yaitu dengan proses Extract Text Metode LSB untuk mengeluarkan pesan dari gambar, setelah pesan keluar kemudian dilakukan proses dekripsi dengan algoritma AES untuk mendapatkan pesan asli dari hasil semula.
Kondisi Awal	Tampil *Button* Dekripsi

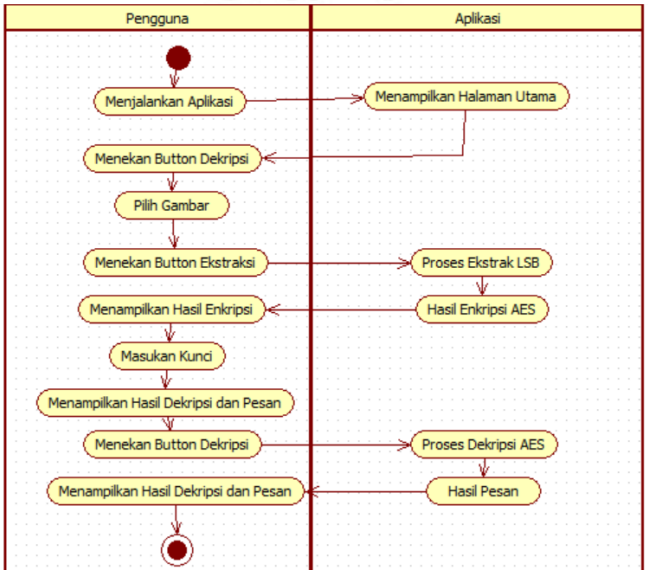
3.4.3. Activity Diagram

Activity diagram adalah proses alur bisnis atau kerja antara aktor dan sistem/aplikasi.

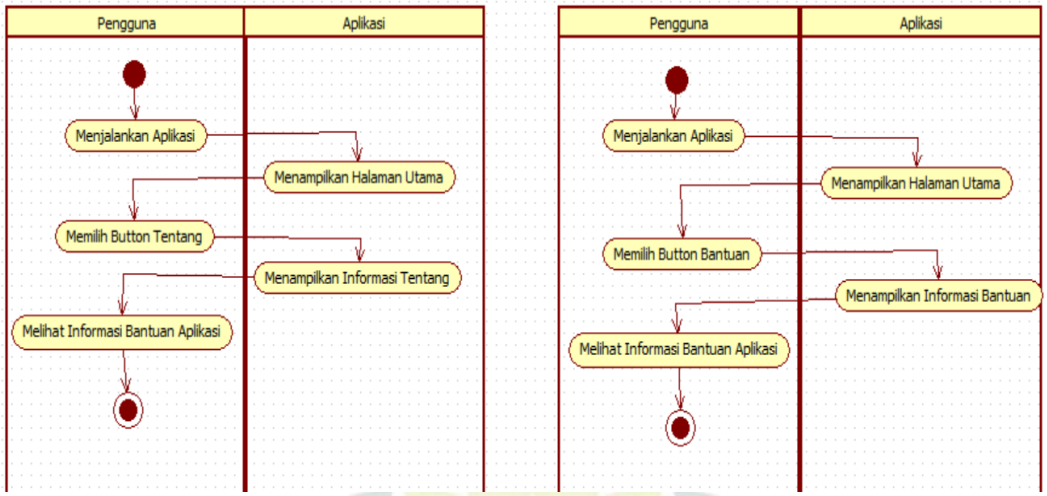
2. Activity diagram Enkripsi



3. Activity diagram Dekripsi

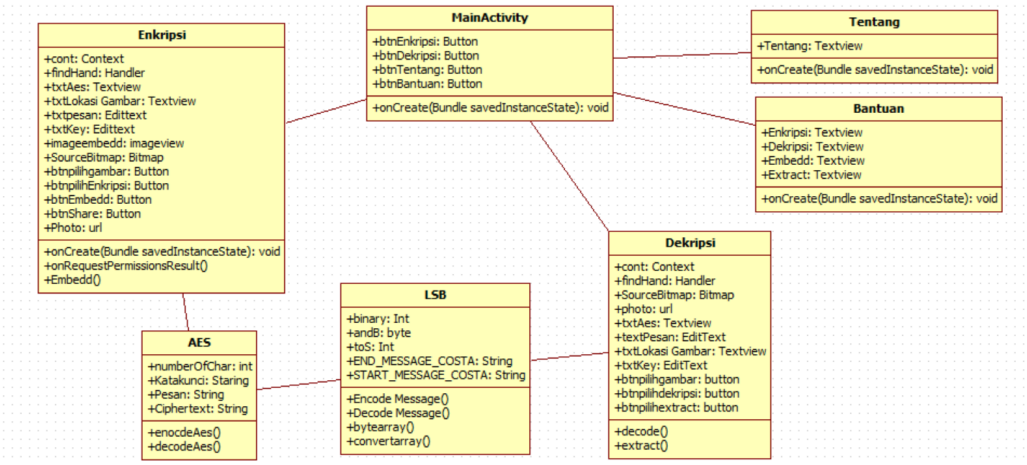


4. Activity diagram Tentang dan Bantuan



3.4.4. Class Diagram

Class diagram adalah diagram yang digunakan untuk menampilkan kelas-kelas serta paket-paket yang terdapat dalam sistem atau aplikasi yang akan dibangun. *Class diagram* dapat dilihat pada Gambar 3.12

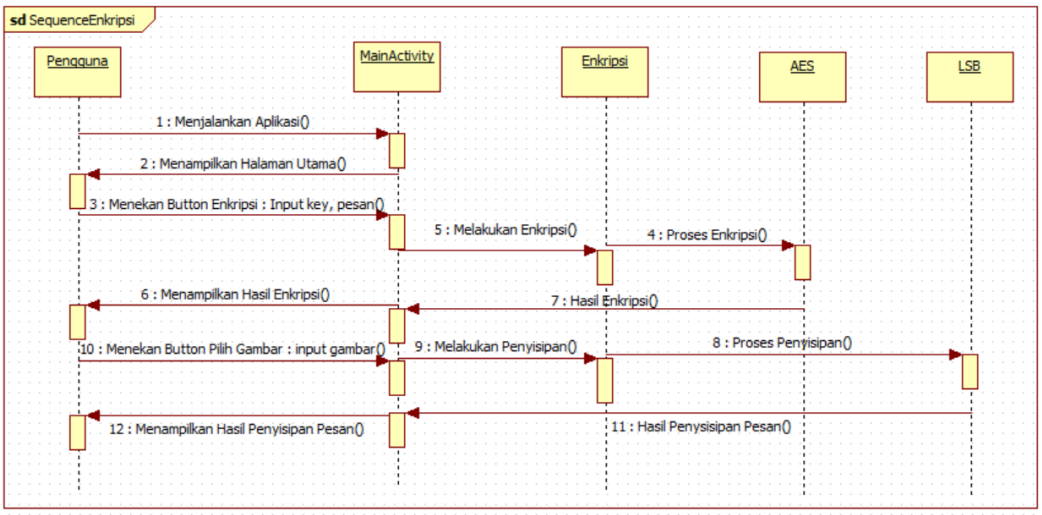


Gambar 3. 11 Class diagram Aplikasi

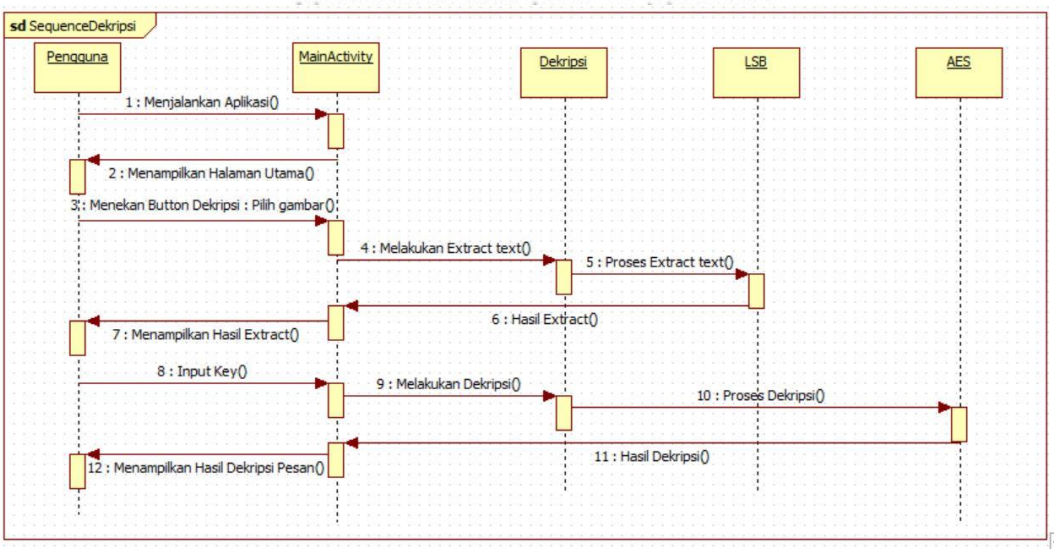
3.4.5. Sequence Diagram

Sequence diagram adalah sebuah diagram yang menggunakan interaksi antara objek di dalam sebuah sistem. Sequence diagram terdiri dari dimensi *horizontal* (Objek-objek) dan dimensi *Vertical* (waktu).

1. *Sequence diagram* enkripsi

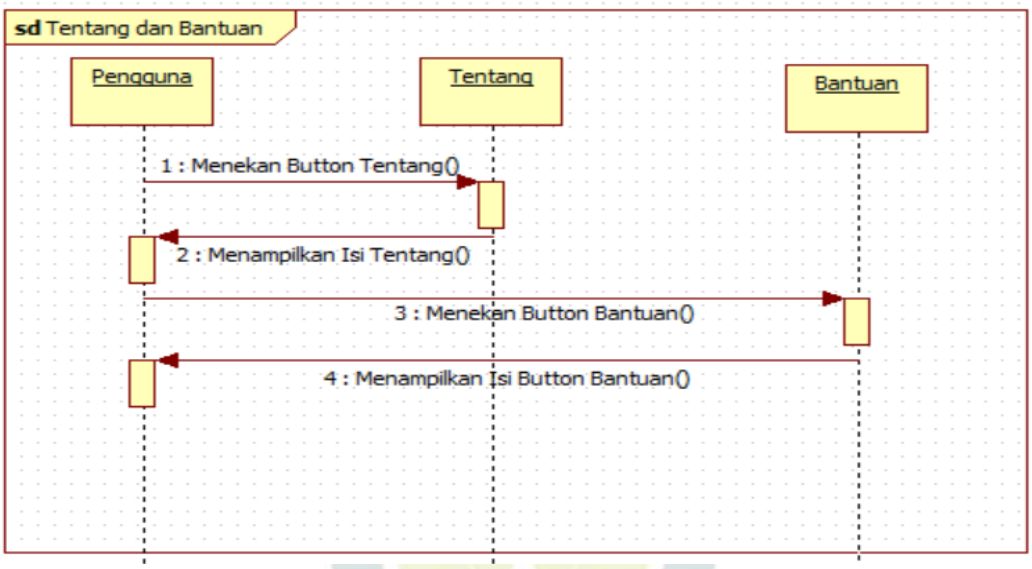


2. *Sequence diagram* dekripsi



Gambar 3. 13 Sequence diagram dekripsi

3. *Sequence diagram* Tentang dan Bantuan



3.6. Perancangan Antarmuka

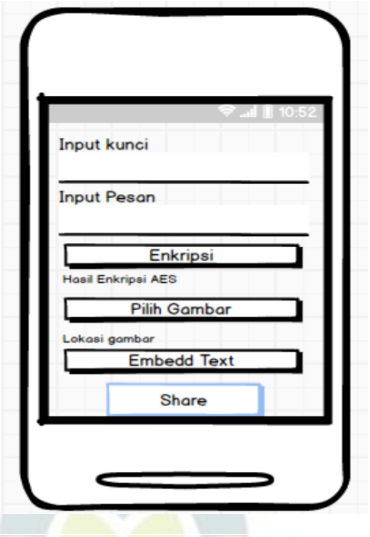
Perancangan antarmuka ini bertujuan untuk memberikan gambaran desain aplikasi yang akan dibuat. Dibawah ini disajikan beberapa tampilan aplikasi yang akan dibuat.

3.5.1. Tampilan Utama

Berikut adalah gambar perancangan antarmuka Tampilan Utama

Pada Gambar 3.15 Merupakan perancangan tampilan utama dari aplikasi, tampilan utama ini dapat mengakses Enkripsi, Dekripsi, Tentang dan Bantuan.

3.5.2. Tampilan Enkripsi



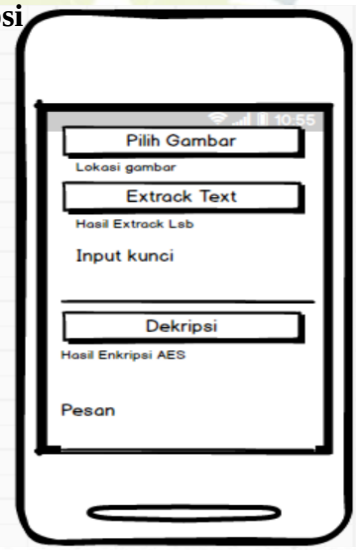
Pada Gambar 3.16 Merupakan perancangan antarmuka ini menampilkan untuk *Input text* yang ingin di sisipkan ke dalam gambar, *user input text*, *key* lalu klik button enkripsi terlebih dahulu untuk mengenkripsi pesan, upload gambar yang akan kita sisipkan chiperteks dan setelah itu klik *button embedded* untuk penyisipan pesan ke dalam gambar. Berikut pseude code proses enkripsi AES 256 bit :

```
//Algoritma enkripsi Kunci AES
Input : byte state (byte in[4*Nb)
Output: out
Prosedur
byte state[4,Nb]
state = in
        AddRoundKey(state, w[0, Nb-1])
        for round = 1 step 1 to Nr–1
        SubBytes(state)
        ShiftRows(state)
        MixColumns(state)
        AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
end for
        SubBytes(state)
        ShiftRows(state)
        AddRoundKey(state, w[Nr*Nb,
        (Nr+1)*Nb-1])
```

```
out = state
end

//Algoritma penyisipan (encoding) LSB
function Encode (img, msg, K, Offset)
        for i ← 1 to msg.length do
                msg[i] ← msg[i] + K
                for j ← 7 to 0 do
                B ← (msg[i] bit shift right b) AND 1
                img[offset] = (img[offset] AND 0xFE) OR b
                offset++
                end
        end
end
```

3.5.3. Tampilan Dekripsi



Pada Gambar 3.17 Merupakan perancangan antarmuka ini menampilkan untuk hasil dari enkripsi, upload gambar dan *input key* untuk mengekstraks pesan dari gambar dengan klik *button Exstract*, lalu klik button dekrips untuk mengembalikan plainteks dari chiperteks. Berikut pseude code proses dekripsi AES 256 bit :

```
//Algoritma dekripsi Kunci AES}
        Input : chipertext=(byte in[4*Nb], word w[Nb*(Nr+1)])
        Output : out
Prosedur  byte state[4,Nb]  state = in
        AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])
        for round = Nr-1 step -1 downto 1
        InvShiftRows(state)
        InvSubBytes(state)
        AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
        InvMixColumns(state)  end for
        InvShiftRows(state)
        InvSubBytes(state)
        AddRoundKey(state, w[0, Nb-1])
out = state , end

//Algoritma ekstraksi (decoding) LSB
function Decode (img, K, Offset)
        for i ← 1 to msg.length do
                for j ← 0 to 7 do
                msg[i] = (msg[i] bit shift left 1) OR (img[offset] AND 1)
                offset++
        end
        msg[i] ← msg[i] - K
        end
        return msg
end
```