BAB III

ANALISIS DAN DESAIN SISTEM

III.1. Analisis Masalah

Perkembangan dunia teknologi selain memberikan kemudahan-kemudahan di berbagai sektor turut juga mempengaruhi berbagai hal lain yang juga menuntut untuk menggunakan teknologi, salah satunya adalah konsep pengaman berkas penting perusahaan maupun isntitutusi lainnya.

Dahulu berkas dalam bentuk *hard copy* akan disimpan dalam satu ruangan arsip yang besar, yang tentunya semakin lama akan semakin menumpuk jumlahnya, hal yang kerap terjadi adalah kerusakan berkas dan kebutuhan ruang penyimpanan yang akan semakin besar pula mengikut banyaknya jumlah berkas yang tersimpan.

Untuk mengatasi hal tersebut diatas maka perlu diterapkan konsep baru yang dapat mengatasi masalah diatas, yaitu dengan melakukan penyimpanan berkas kedalam bentuk digital, untuk menjamin kemanan dari berkas tersebut maka perlu diterapkan konsep kriptografi agar berkas yang tersimpan tidak mudah dimanfaat oleh pihak-pihak yang tidak bertanggung jawab.

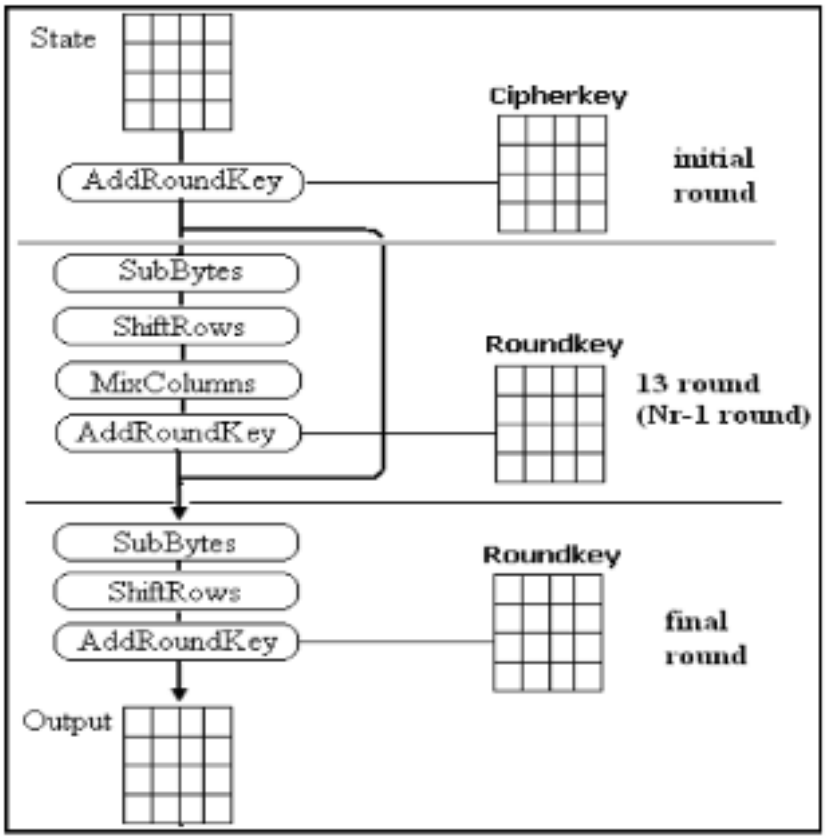
III.2. Strategi Pemecahan Masalah

Berikut adalah strategi pemecahan masalah yang akan diterapkan didalam penelitian ini, diantaranya adalah :

1. Berkas yang akan diarsipkan dapat berupa dokumen-dokumen dari Microsoft Office seperti word, excel maupun pdf .
2. Aplikasi tidak akan menggunakan database sehingga arsip yang telah di enkripsi akan disimpan langsung kedalam komputer.
3. Algoritma yang akan digunakan adalah algoritma AES256.

III.3. Proses Enkripsi *Advanced Enryption Standard*

Proses enkripsi algoritma AES terdiri dari 4 jenis transformasi bytes, yaitu *SubBytes,ShiftRows, Mixcolumns,* dan *AddRoundKey*. Pada awal proses enkripsi, *input* yang telah di*copy*kan ke dalam *state* akan mengalami transformasi byte *AddRoundKey*. Setelah itu, *state* akan mengalami transformasi *SubBytes, ShiftRows*, *MixColumns,* dan *AddRoundKey* secara berulang-ulang sebanyak Nr. Proses ini dalam algoritma AES disebut sebagai *round function Round* yang terakhir agak berbeda dengan *round-round* sebelumnya dimana pada *round* terakhir, *state* tidak mengalami transformasi *MixColumns*. Ilustrasi proses enkripsi AES dapat digambarkan seperti pada Gambar III.1 di bawah ini :



Gambar 3.1 Proses Enkripsi AES

State

32	88	31	e0
43	5a	31	37
f6	30	98	7
a8	8d	a2	34

Key

2b	28	ab	9
7e	ae	f7	cf
15	d2	15	4f
16	a6	88	3c

1. ***AddRoundKey***

   Pada proses enkripsi dan dekripsi AES proses *AddRoundKey* sama, sebuah *round key* ditambahkan pada *state* dengan operasi XOR. Setiap *round key*

terdiri dari Nb *word* dimana tiap *word* tersebut akan dijumlahkan dengan *word* atau kolom yang bersesuaian dari *state* sehingga :

$[s'_{,0}\ c\ ,\ s'_{,1}\ c\ ,\ s'_{,2}\ c\ ,\ s'_{,3}\ c\ ] = [s_{,0}\ c\ ,\ s_{,1}\ c\ ,\ s_{,2}\ c\ ,\ s_{,3}\ c\ ] \oplus [wround*Nb+c\ ]$ untuk $0 \leq c \leq Nb$

[wi] adalah *word* dari *key* yang bersesuaian dimana i = round*Nb+c. Transformasi *AddRoundKey* pada proses enkripsi pertama kali pada *round* = 0 untuk *round* selanjutnya *round* = *round* + 1, pada proses dekripsi pertama kali pada *round* = 14 untuk *round* selanjutnya *round* = *round* - 1.

32	88	31
43	5a	31
f6	30	98
a8	8d	a2

e0
37
7
34

2b
7e
15
16

28	ab	9
ae	f7	cf
d2	15	4f
a6	88	3c

Pada tabel tersebut di sebelah kiri adalah *chiper teks* dan sebelah kanan adalah round key nya. XOR dilakukan per kolom yaitu kolom-1 chiper teks di XOR dengan kolom-1 round key dan seterusnya.

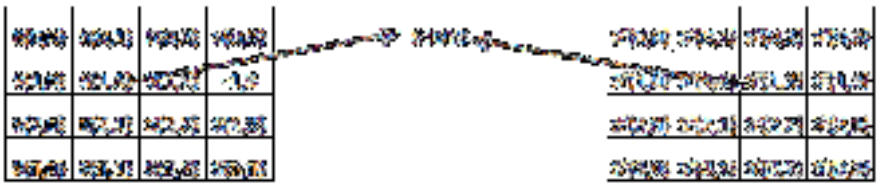
2. ***Sub Bytes***

SubBytes merupakan transformasi *byte* dimana setiap elemen pada *state* akan dipetakan dengan menggunakan sebuah tabel substitusi (S-Box). tabel substitusi S-Box akan dipaparkan dalam Tabel III.1

Tabel 3.1. Tabel Subtitusi S-Box
(*Sumber :Herdaya Adiyasa, et-al, 2014*)

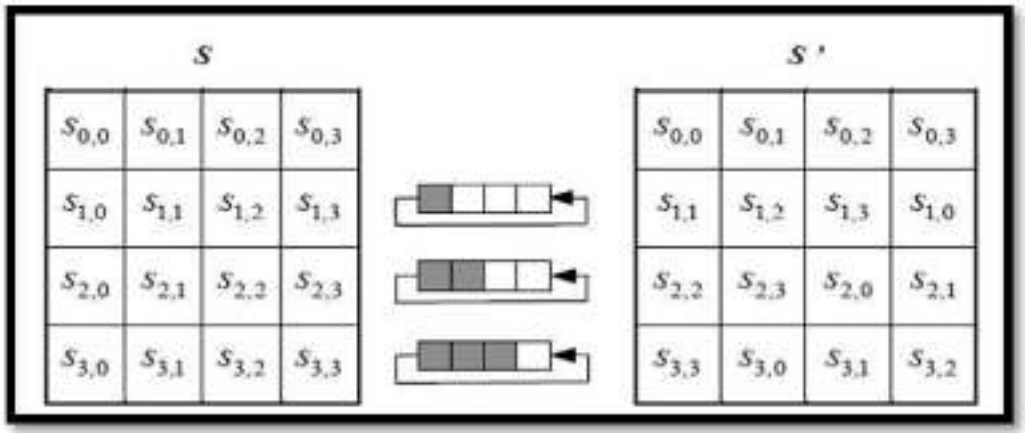
		y														
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	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

Untuk setiap *byte* pada *array state*, misalkan S[r, c] = xy, yang dalam hal ini xy adalah digit heksadesimal dari nilai S[r, c], maka nilai substitusinya, dinyatakan dengan S'[r, c], adalah elemen di dalam tabel subtitusi yang merupakan perpotongan baris x dengan kolom y. Gambar 3 mengilustrasikan pengaruh pemetaan byte pada setiap byte dalam state.



3. ***Shiftrows***

Transformasi *Shiftrows* pada dasarnya adalah proses pergeseran bit dimana bit paling kiri akan dipindahkan menjadi bit paling kanan (rotasi bit). Proses pergeseran *Shiftrow* ditunjukkan dalam Gambar 4 berikut:



Gambar 3.2. Transformasi *ShiftRows*

4. ***MixColumns***

MixColumns mengoperasikan setiap elemen yang berada dalam satu kolom pada state. Secara lebih jelas, transformasi mixcolumns dapat dilihat pada perkalian matriks berikut ini:

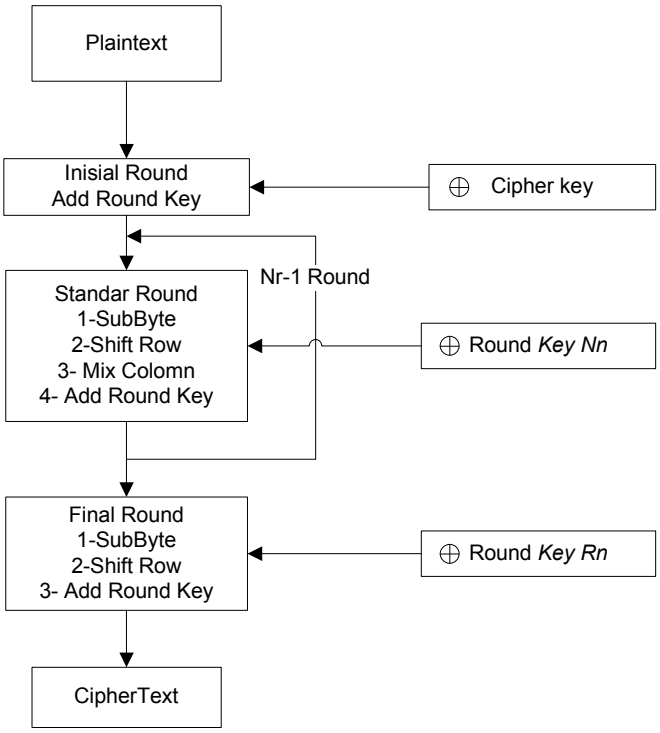
$$\begin{bmatrix} S'_{0,C} \\ S'_{1,C} \\ S'_{2,C} \\ S'_{3,C} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,C} \\ S_{1,C} \\ S_{2,C} \\ S_{3,C} \end{bmatrix}$$

Hasil dari perkalian matriks diatas dapat dianggap seperti perkalian yang ada di bawah ini :

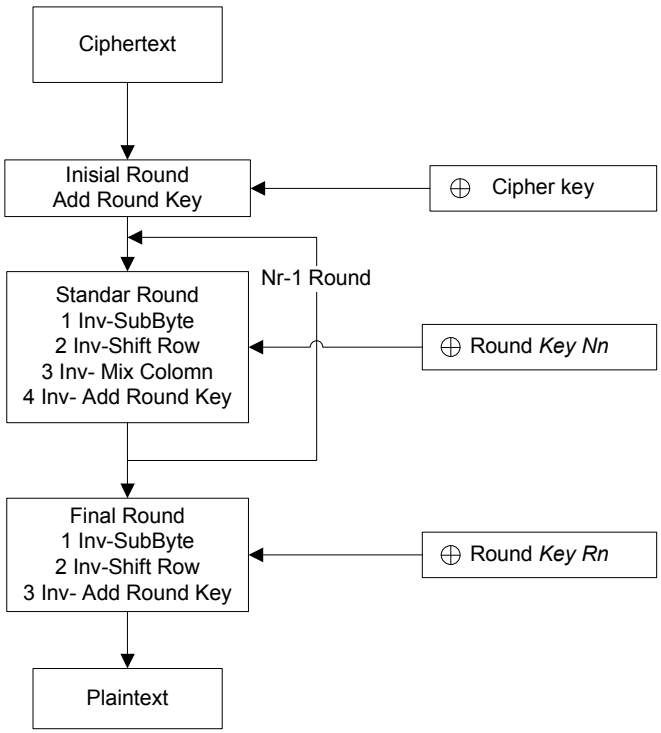
$$s'_{0,c} = (\{02\} \bullet s_{0,c}) \oplus (\{03\} \bullet s_{1,c}) \oplus s_{2,c} \oplus s_{3,c}$$
$$s'_{1,c} = s_{0,c} \oplus (\{02\} \bullet s_{1,c}) \oplus (\{03\} \bullet s_{2,c}) \oplus s_{3,c}$$
$$s'_{2,c} = s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{2,c}) \oplus (\{03\} \bullet s_{3,c})$$
$$s'_{3,c} = (\{03\} \bullet s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (\{02\} \bullet s_{3,c})$$

III.3.1. Flowchart AES256

Berikut adalah flowchart dari proses enkripsi pada algoritma AES256 :



Gambar 3.3. Flowchart Enkripsi AES256



Gambar 3.4. Flowchart Proses Dekripsi Aes256

III.3.2. Studi Kasus

Berikut adalah merupakan studi kasus perhitungan secara manual untuk algoritma AES 256.

PlainText : 32 43 f6 a8 | 88 5a 30 8d | 31 31 98 a2 | - - - -

PKCS 5 : 32 43 f6 a8 | 88 5a 30 8d | 31 31 98 a2 | 04 04 04 04

Key : 2b 7e 15 16 | 28 ae d2 a6 | - - - - | - - - -

PKCS 5 : 2b 7e 15 16 | 28 ae d2 a6 | 08 08 08 08 | 08 08 08 08

Tabel 3.2. Tabel RCon

Rcon(0) = 01000000	Rcon(1) = 02000000
Rcon(2) = 04000000	Rcon(3) = 08000000
Rcon(4) = 10000000	Rcon(5) = 20000000
Rcon(6) = 40000000	Rcon(7) = 80000000
Rcon(8) = 1B000000	Rcon(9) = 36000000
Rcon(10) = 6C000000	Rcon(11) = D8000000
Rcon(12) = AB000000	Rcon(13) = 4D000000
Rcon(14) = 9A000000	

Tabel 3.3. Tabel S-Box AES

S-Box Values																
S(xy)		**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

Pertama hitung dulu key schedule. Hasil dari perhitungan akan digunakan pada proses selanjutnya untuk peoses enkripsi.

Key Schedule

Cipher Key : 2b 7e 15 16 | 28 ae d2 a6 | 08 08 08 08 | 08 08 08 08

2b	28	08	08
7e	ae	08	08
15	d2	08	08
16	a6	08	08

RotWord : 08 08 08 08

S-Box : 30 30 30 30

2b	28	08	08
7e	ae	08	08
15	d2	08	08
16	a6	08	08

2b		30		01		1a	32	3a	32
7e	⊕	30	⊕	00	=	4e	e0	e8	e0
15		30		00		25	f7	ff	f7
16		30		00		26	80	88	80
				rcon					

Round Key 1 : 1a 4e 25 26 | 32 e0 f7 80 | 3a e8 ff 88 | 32 e0 f7 80

Round Key 2 : f9 26 e8 05 | cb c6 1f 85 | f1 2e e0 0d | c3 ce 17 8d

Round Key 3 : 76 d6 b5 2b | bd 10 aa ae | 4c 3e 4a a3 | 8f f0 5d 2e

Round Key 4 : f2 9a 84 58 | 4f 8a 2e f6 | 03 b4 64 55 | 8c 44 39 7b

Round Key 5 : f9 88 a5 3c | b6 02 8b ca | b5 b6 ef 9f | 39 f2 d6 e4

Round Key 6 : 50 7e cc 2e | e6 7c 47 e4 | 53 ca a8 7b | 6a 38 7e 9f

Round Key 7 : 17 8d 17 2c | f1 f1 50 c8 | a2 3b f8 b3 | c8 03 86 2c

Round Key 8 : ec c9 66 c4 | 1d 38 36 0c | bf 03 ce bf | 77 00 48 93

Round Key 9 : 94 9b ba 31 | 89 a3 8c 3d | 36 a0 42 82 | 41 a0 0a 11

Round Key 10 : 42 fc 38 b2 | cb 5f b4 8f | fd ff f6 0d | bc 5f fc 1c

Initial Round

Pertama dilakukan proses inisialisasi dengan operasi XOR antara *State* dan *Key*. Jika *state* atau *key* kurang dari 16 *byte* maka isi *byte* yang kosong dengan aturan *padding* PKCS5.

Initial State : 32 43 f6 a8 | 88 5a 30 8d | 31 31 98 a2 | 04 04 04 04

Cipher Key : 2b 7e 15 16 | 28 ae d2 a6 | 08 08 08 08 | 08 08 08 08

After AddRoundKey : 19 3d e3 be | a0 f4 e2 2b | 39 39 90 aa | 0c 0c 0c 0c

Round 1

State : 19 3d e3 be | a0 f4 e2 2b | 39 39 90 aa | 0c 0c 0c 0c

Round Key : 1a 4e 25 26 | 32 e0 f7 80 | 3a e8 ff 88 | 32 e0 f7 80

Setelah tahap inisialisasi maka dimulai round 1, tahap pertama dalam setiap round atau putaran adalah SubBytes, yaitu subtitusi State menggunakan SBox. Seperti yang terlihat pada tabel berikut ini , Cara mencarinya sebagai berikut:

State Round 1 memiliki matriks { 19 3d e3 be | a0 f4 e2 2b | 39 39 90 aa | 0c 0c 0c 0c }, jika diketahui byte pertama 19 y = 1 , x = 9, maka :

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0y	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

Begitu seterusnya sampai semua matriks tersubtitusi,

After SubBytes : d4 27 11 ae | e0 bf 98 f1 | 12 12 60 ac | fe fe fe fe

Tahap selanjutnya adalah *ShiftRows,*

d4	e0	12	fe	→	d4	e0	12	fe	→	d4	e0	12	fe
27	bf	12	fe		bf	12	fe	27		bf	12	fe	27
11	98	60	fe		**11**	**98**	60	fe		60	fe	11	98
ae	f1	ac	fe		ae	f1	ac	fe		**ae**	**f1**	**ac**	fe

→	d4	e0	12	fe
	bf	12	fe	27
	60	fe	11	98
	fe	**ae**	**f1**	**ac**

After ShiftRows : d4 bf 60 fe | e0 12 fe ae | 12 fe 11 f1 | fe 27 98 ac

Selanjutnya adalah tahap MixColumns yaitu perkalian matriks antara hasil ShiftRows dan GF(8) atau matriks gaulios field yang telah ditentukan,

					Matriks GF(8)			
d4	e0	12	fe	x	02	01	01	03
bf	12	fe	27		03	02	01	01
60	fe	11	98		01	03	02	01
fe	ae	f1	ac		01	01	02	03

After MixColumns : f7 ef b2 5f | bd 73 fc 90 | dd 37 c6 20 | ba af 1d e5

After AddRoundKey : ed a1 97 79 | 8f 93 0b 10 | e7 df 39 a8 | 88 4f ea 65

c. Round 2

After SubBytes : 55 32 88 b6 | 73 dc 2b ca | 94 9e 12 c2 | c4 84 87 4d

After ShiftRows : 55 dc 12 4d | 73 9e 87 b6 | 94 84 88 ca | c4 32 2b c2

After MixColumns : 8a 8d 7a ab | 6e 70 39 fb | e6 ce 5e 24 | 2c 1f fd d1

After AddRoundKey : 73 ab 92 ae | a5 b6 26 7e | 17 e0 be 29 | ef d1 ea 5c

d. Round 3

After SubBytes : 8f 62 4f e4 | 06 4e f7 f3 | f0 e1 ae a5 | df 3e 87 4a

After ShiftRows : 8f 4e ae 4a | 06 e1 87 e4 | f0 3e 4f f3 | df 62 f7 a5

After MixColumns : 33 b0 58 fe | 57 a9 c5 bf | 05 ae 5e 87 | 51 bc bc be

After AddRoundKey : 45 66 ed d5 | ea b9 6f 11 | 49 90 14 24 | de 4c e1 90

e. Round 4

After SubBytes : 6e 33 55 03 | 87 56 a8 82 | 3b 60 fa 36 | 1d 29 f8 60

After ShiftRows : 6e 56 fa 60 | 87 60 f8 03 | 3b 29 55 82 | 1d 33 a8 36

After MixColumns : bc b7 77 de | 4e 57 09 0c | da 14 25 2e | f1 ae 3f d0

After AddRoundKey : 4e 2d d3 86 | 01 dd 27 fa | d9 a0 41 7b | 7d ea 06 ab

f. Round 5

After SubBytes : 2f d8 0d 44 | 7c c1 cc 2d | 35 e0 83 21 | ff 87 6f 62

After ShiftRows : 2f c1 83 62 | 7c e0 6f 44 | 35 87 0d 2d | ff d8 cc 21

After MixColumns : e7 e8 d8 7b | e8 52 8e 83 | d8 1a df 8f | 7b 3a c7 4c

After AddRoundKey : 1e c2 f0 cb | 5e 50 05 49 | 6d ac 30 10 | 42 c8 11 a8

g. Round 6

After SubBytes : 72 25 8c 1f | 58 53 6b 3b | 3c 91 04 ca | 2c e8 82 c2

After ShiftRows : 72 53 04 c2 | 58 91 82 1f | 3c e8 8c 3b | 2c 25 6b ca

After MixColumns : d7 1a 74 5e | 85 e3 f7 c5 | ec 43 9a 56 | 96 11 9a b5

After AddRoundKey : 87 64 b8 70 | 63 9f b0 21 | bf 89 32 2d | fc 29 e4 2a

h. Round 7

After SubBytes : 17 43 6c 51 | fb db f7 fd | 08 a7 23 d8 | b0 a5 69 e5

After ShiftRows : 17 db 23 e5 | fb a7 69 51 | 08 a5 6c fd | b0 43 e7 d8

After MixColumns : 9e 3a be 10 | 27 44 7d 7a | 75 10 69 30 | 81 dc 55 c4

After AddRoundKey : 89 b7 a9 3c | d6 b5 2d b2 | d7 2b 91 83 | 49 df d3 e8

i. Round 8

After SubBytes : a7 a9 d3 eb | f6 d5 d8 37 | 0e f1 81 ec | 3b 9e 66 9b

After ShiftRows : a7 d5 81 9b | f6 f1 66 eb | 0e 9e d3 37 | 3b a9 d8 ec

After MixColumns : 2b 15 dd 8b | 72 4e ed 5b | 41 70 74 31 | a2 ed 16 ff

After AddRoundKey : c7 dc bb 4f | 6f 76 db 57 | fe 73 ba 80 | d5 ed 5e 6c

j. Round 9

After SubBytes : c6 86 ea 84 | a8 38 b9 5b | bb 8f f4 19 | 03 55 58 50

After ShiftRows : c6 38 f4 50 | a8 8f 58 84 | bb 55 ea 5b | 03 86 b9 19

After MixColumns : 7b e1 fd 3d | 1d c1 00 27 | 23 6f cc df | 37 dd c7 08

After AddRoundKey : ef 7a 47 0c | 94 62 8c 1a | 15 cf 8e 5d | 76 7d cd 19

k. Final Round

After SubBytes : df da a0 fe | 22 aa 64 a2 | 59 8a 19 4c | 38 ff bd d4

After ShiftRows : df aa 19 d4 | 22 8a bd fe | 59 ff a0 a2 | 38 da 64 4c

After AddRoundKey : 9d 56 21 66 | e9 d5 09 71 | a4 00 56 af | 84 85 98 50

Hasil CipherText adalah 9d 56 21 66 | e9 d5 09 71 | a4 00 56 af | 84 85 98 50

Analisis Dekripsi

CipherText

Block : 9d 56 21 66 | e9 d5 09 71 | a4 00 56 af | 84 85 98 50

Key : 2b 7e 15 16 | 28 ae d2 a6 | 08 08 08 08 | 08 08 08 08

Key Schedule

Round Key 1 : 7b e1 fd 3d | 1d c1 00 27 | 23 6f cc df | 37 dd c7 08

Round Key 2 : 2b 15 dd 8b | 72 4e ed 5b | 4170 74 31 | a2 ed 16 ff

Round Key 3 : 9e 3a be 10 27 44 7d 7a 75 10 69 30 81 dc 55 c4

Round Key 4 : d7 1a 74 5e 85 e3 f7 c5 ec 43 9a 56 96 11 9a b5

Round Key 5 : e7 4a 55 f7 e8 52 8e 83 d8 1a df 8f 7b 3a c7 4c

Round Key 6 : bcb7 77 de 4e 57 09 0c da 14 25 2e f1 ae 3f d0

Round Key 7 : 33 ba 58 fe 57 a9 c5 bf 05 ae 5e 87 51 bc bc be

Round Key 8 : 8a 8d 7a ab 6e 70 39 fb e6 ce 5e 24 2c1f fd d1

Round Key 9 : f7 ef b2 5f bd 73 fc 90 dd 37 c6 20 ba af 1d e5

a. Initial Round

After AddRoundKey : df aa 19 d4 | 22 8a bd fe | 59 ff a0 a2 | 38 da 64 4c

b. Round 1

After InvShiftRow : df da a0 fe 22 aa 64 a2 59 8a 19 4c 38 ff bd d4

After InvSubByte : ef 7a 47 0c 94 62 8c 1a 15 cf 8e 5d 76 7d cd 19

After InvMixColumn : 94 9b ba 31 89 a3 8c 3d 36 a0 42 82 41 a0 0a 11

After AddRoundKey : c6 38 f4 50 a8 8f 58 84 bb 55 ea 5b 03 86 b9 19

c. Round 2

After InvShiftRow : c6 86 ea 84 a8 38 b9 5b bb 8f f4 19 03 55 58 50

After InvSubByte : c7 dc bb 4f 6f 76 db 57 fe 73 ba 8e d5 ed 5e 6c

After InvMixColumn : ec c9 66 c4 1d 38 36 0c bf 03 ce bf 77 00 48 93

After AddRoundKey : a7 d5 81 9b f6 f1 66 eb 0e 9e d3 37 3b a9 d8 ec

d. Round 3

After InvShiftRow : a7 a9 d3 eb f6 f5 d8 37 0e f1 81 ec 0e f1 81 ec 3b 9e 66 9b

After InvSubByte : c7 dc bb 4f 6f 76 db 57 fe 73 ba 8e d5 ed 5e 6c

After InvMixColumn : ec c9 66 c4 1d 38 36 0c bf 03 ce bf 77 00 48 93

After AddRoundKey : 17 db 23 e5 fb a7 69 51 08 a5 6c fd b0 43 e7 d8

e. Round 4

After InvShiftRow : 17 43 6c 51 fb db e7 fd 08 a7 23 d8 b0 a5 69 e5

After InvSubByte : 87 64 b8 70 63 9f b0 21 bf 89 32 2d fc 29 e4 2a

After InvMixColumn : 50 7e cc 2e e6 7c 47 e4 53 ca a8 7b 6a 38 7e 9f

After AddRoundKey : 72 53 04 c2 58 91 82 1f 3c e8 8c 3b 2c 25 6b ca

f. Round 5

After InvShiftRow : 72 25 8c 1f 58 53 6b 3b 3c 91 04 ca 2c e8 82 c2

After InvSubByte : 1e c2 f0 cb 5e 50 05 49 6d ac 30 10 42 c8 11 a8

After InvMixColumn : f9 88 a5 3c b6 02 8b ca b5 b6 ef 9f 39 f2 d6 e4

After AddRoundKey : 2f c1 83 62 7c e0 6f 44 35 87 0d 2d ff f8 cc 21

g. Round 6

After InvShiftRow : 2f d8 0d 44 7c c1 cc 2d 35 e0 8e 21 ff 87 6f 62

After InvSubByte : 4e 2d f3 86 01 dd 27 fa d9 a0 41 7b 7d ea 06 ab

After InvMixColumn : f2 9a 84 58 4f 8a 2e f6 03 b4 64 55 8c 44 39 7b

After AddRoundKey : 6e 56 fa 60 87 60 f8 03 3b 29 55 92 1d 33 a8 36

h. Round 7

After InvShiftRow : 6e 33 55 03 87 56 a8 82 3b 60 fa 36 1d 29 f8 60

After InvSubByte : 45 66 ed d5 ea b9 6f 11 49 90 14 24 de 4c e1 90

After InvMixColumn : 76 d6 b5 2b bd 10 aa ae 4c 3e 4a a3 8f f0 5d 2e

After AddRoundKey : 8f 4e ae 4a 06 e1 87 e4 f0 3e 4f f3 df 62 f7 a5

i. Round 8

After InvShiftRow : 8f 62 4f e4 06 4e ff7 f3 f0 e1 ae a5 df 3e 87 4a

After InvSubByte : 73 ab 92 ae a5 b6 26 7e 17 e0 be 29 ef d1 ea 5c

After InvMixColumn : f9 26 e8 05 cb c6 1f 85 f1 2e e0 0d c3 ce 17 8d

After AddRoundKey : 55 dc 12 4d 73 9e 87 b6 94 84 88 ca c4 32 2b c2

j. Round 9

After InvShiftRow : 55 32 88 b6 73 dc 2b ca 94 9e 12 c2 c4 84 87 4d

After InvSubByte : ed a1 97 79 8f 93 0b 10 e7 df 39 a8 88 4f ea 65

After InvMixColumn : 1a 4e 25 26 32 e0 f7 80 3a e8 ff 88 32 e0 f7 80

After AddRoundKey : d4 bf 60 fe e0 12 fe ae 12 fe 11 f1 fe 27 98 ac

k. Round 10

After InvShiftRow : d4 27 11 ae e0 bf 98 f1 2 12 60 ac fe fe fe fe

After InvSubByte : 19 3d e3 be a0 f4 e2 2b 39 39 90 aa 0c 0c 0c 0c

After AddRoundKey : 32 43 f6 a8 | 88 5a 30 8d | 31 31 98 a2 |04 04 04 04

Berdasarkan perhitungan maka hasil dekripsi yang dihasilkan adalah :

32 43 f6 a8 | 88 5a 30 8d | 31 31 98 a2 | 04 04 04 04.

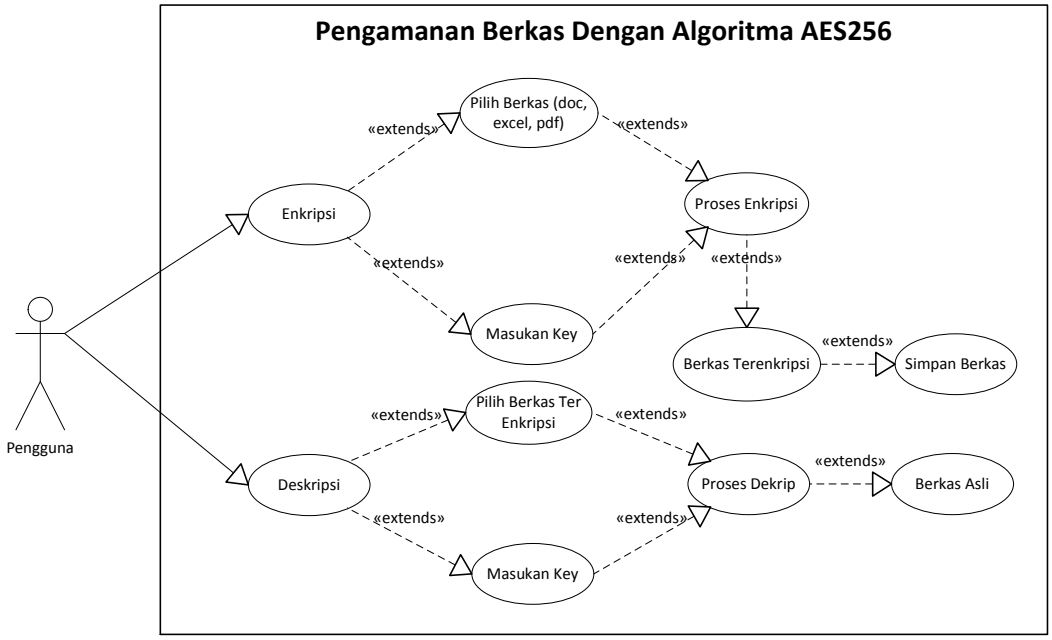
III.4. Desain Sistem

Berikut adalah rancangan dari alur kerja dari aplikasi yang akan dirancang, yang divisualisasikan dengan menggunakan UML, seperti yang terlihat pada gambar-gambar berikut :

III.4.1. *Use Case Diagram*

Use case mendiskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi

yang ada didalam sistem informasi tersebut. Berikut adalah *use case diagram* dari sistem yang dirancang :



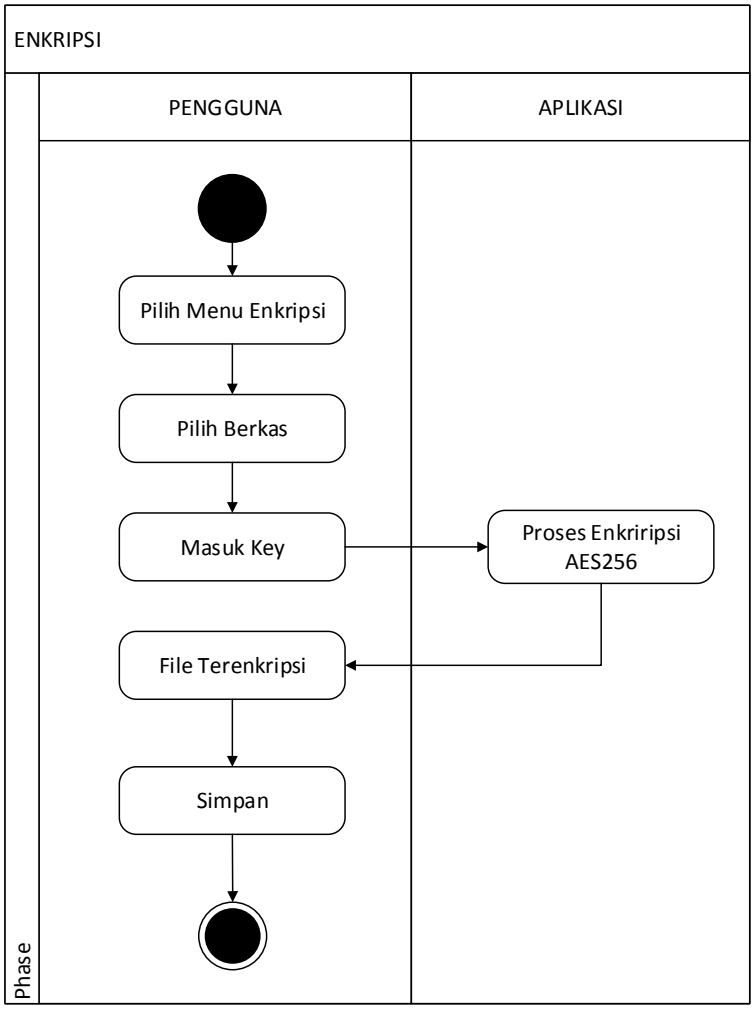
Gambar 3.5. *Use Case Diagram* Pengamanan Berkas dengan Algoritma AES256

III.4.2. *Activity Diagram*

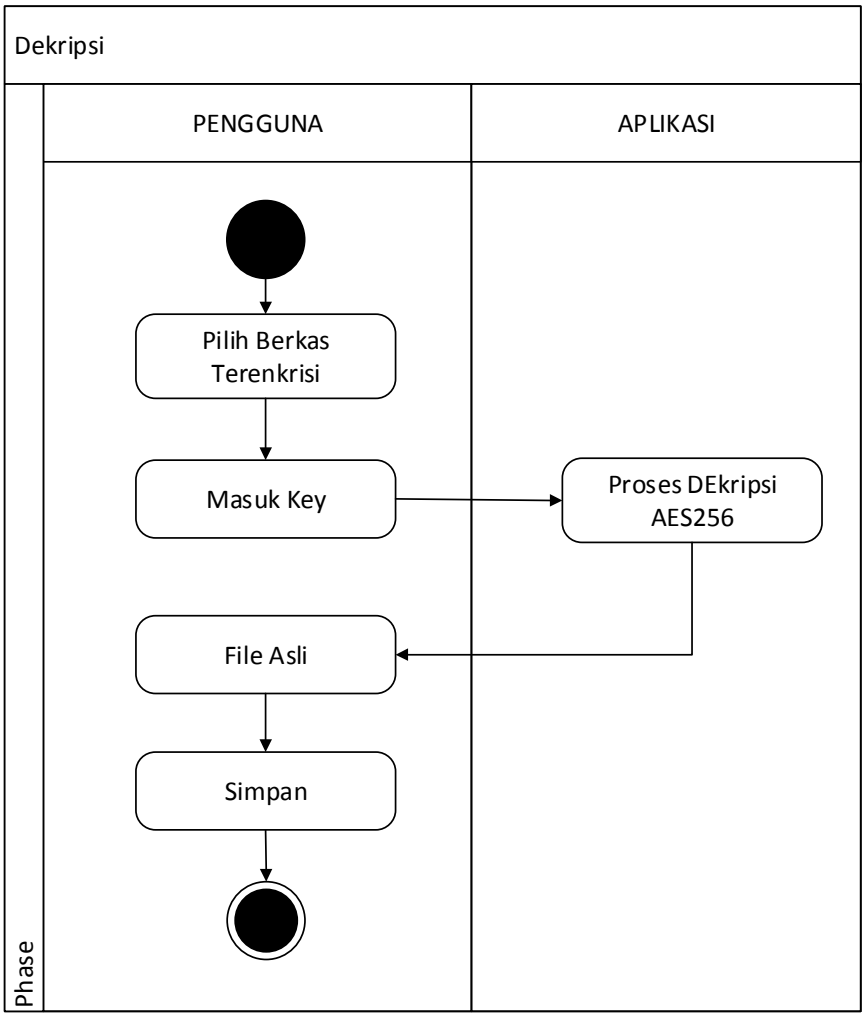
Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* yang terdapat pada aplikasi yaitu sebagai berikut :

1. *Activity Diagram Enkripsi*

Activity diagram berikut menggambarkan proses yang akan berjalan pada aplikasi saat melakukan proses enkripsi terhadap berkas arsip dokumen digital.



Gambar 3.6. *Activity Diagram* Enkripsi berkas



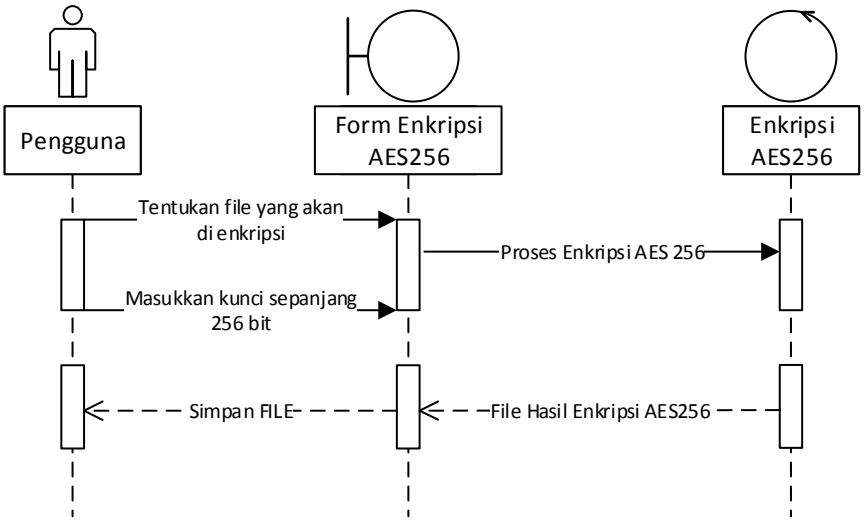
Gambar 3.7 *Activity Diagram* berkas *Dekripsi AES*

III.5. Sequence Diagram

Sequence Diagram adalah salah satu dari diagram - diagram yang ada pada UML, sequence diagram ini adalah diagram yang menggambarkan kolaborasi dinamis antara sejumlah object. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara object juga interaksi antara object. Sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem, berikut adalah sequence

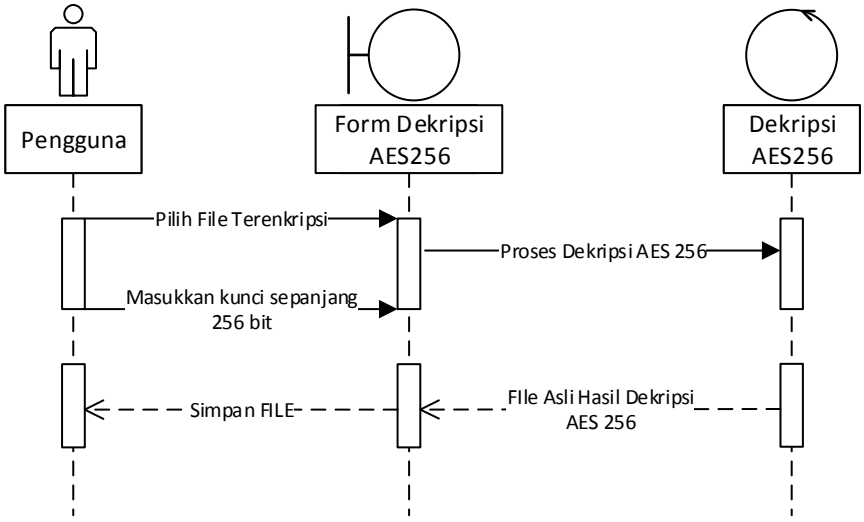
diagram untuk aplikasi penyandian berkas dengan menggunakan algoritma AES256.

1. Sequence Diagram Enkripsi



Gambar 3.8 *Sequence Diagram* berkas Enkripsi AES256

2. Sequence Diagram Dekripsi

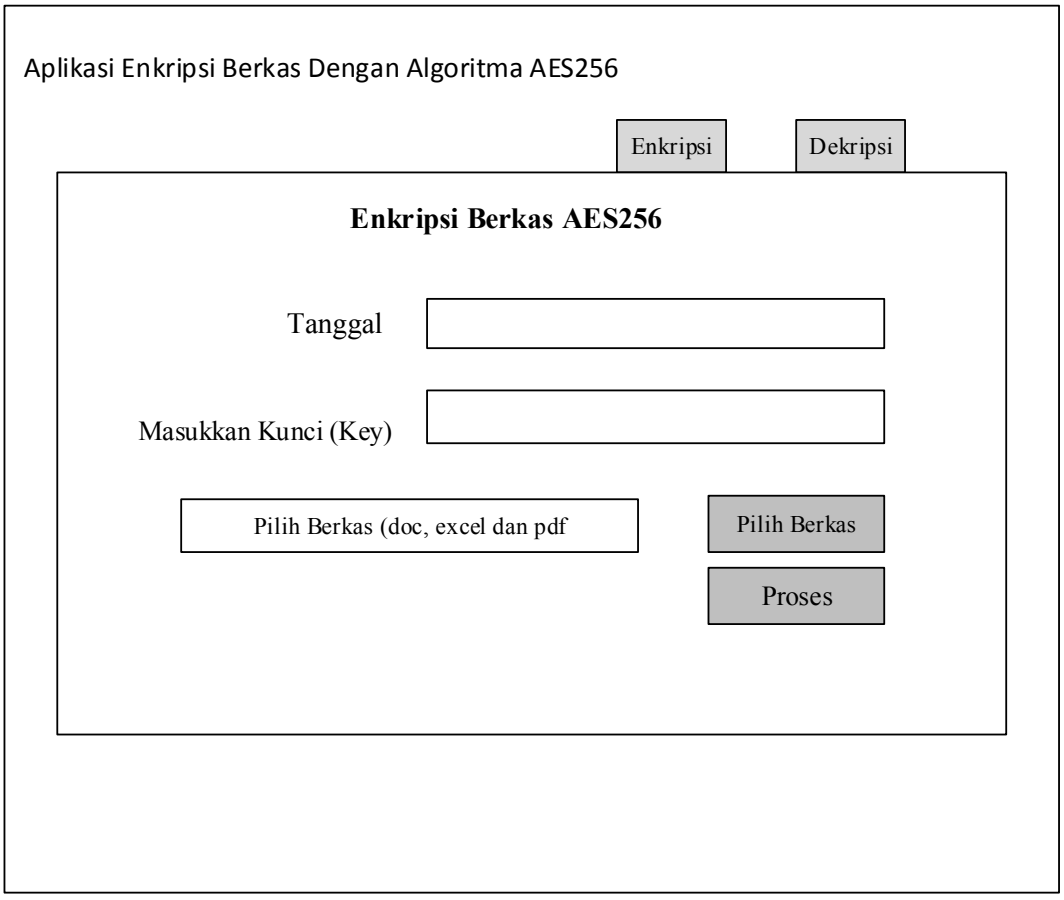


Gambar 3.9 *Sequence Diagram* berkas Dekripsi AES256

III.6. Desain *User Interface*

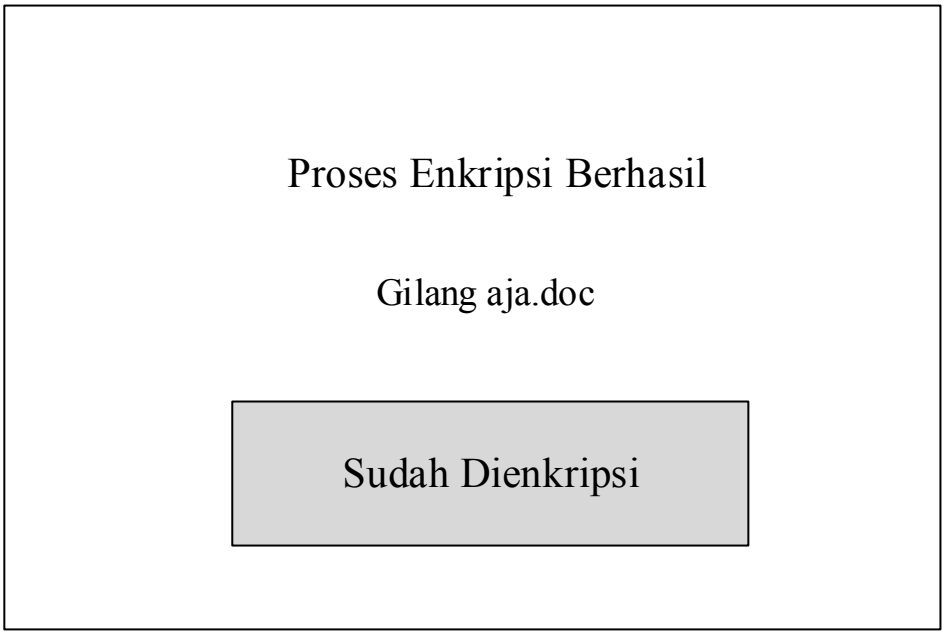
Antarmuka pemakai (*user interface*) adalah tampilan program yang dapat dilihat, didengar atau dipersepsikan oleh pengguna dan perintah-perintah atau mekanisme yang digunakan pemakai untuk mengendalikan operasi dan memasukkan data.

1. Desain Form Enkripsi



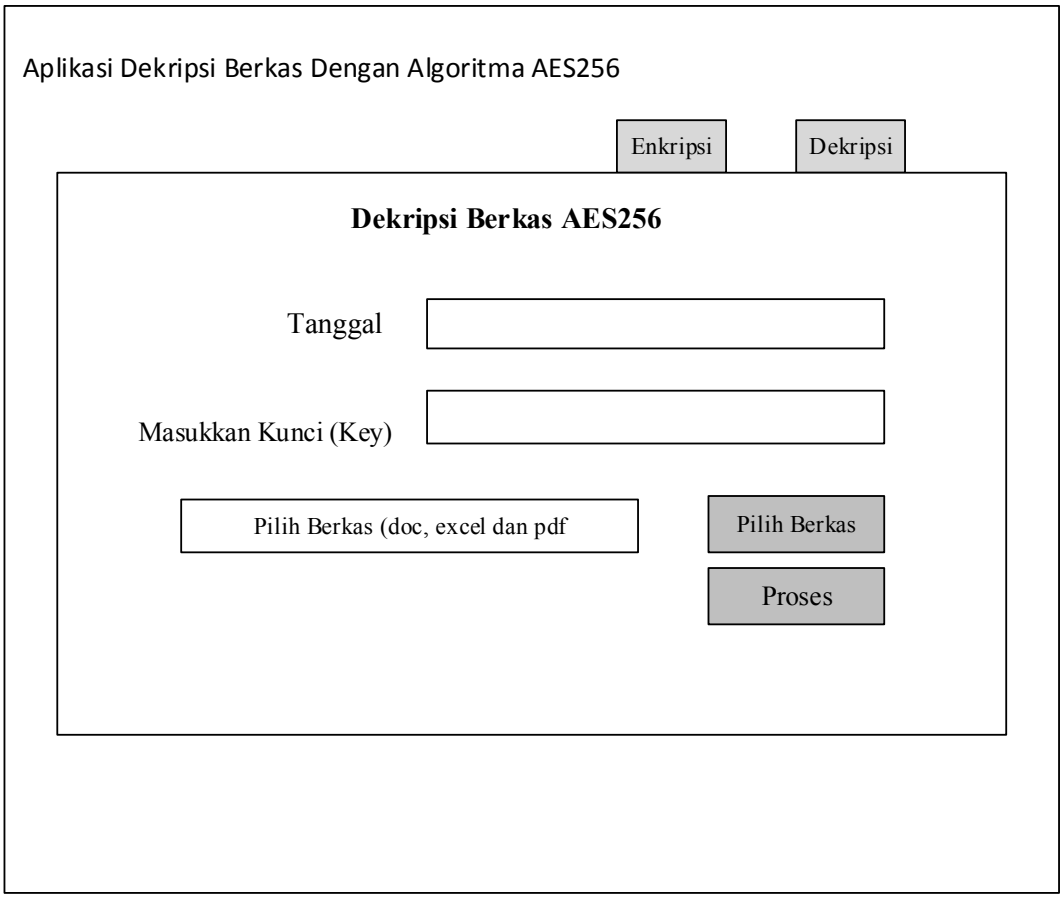
Gambar 3.10 ***Desain Form Encoding***

2. Desain Form Hasil enkripsi



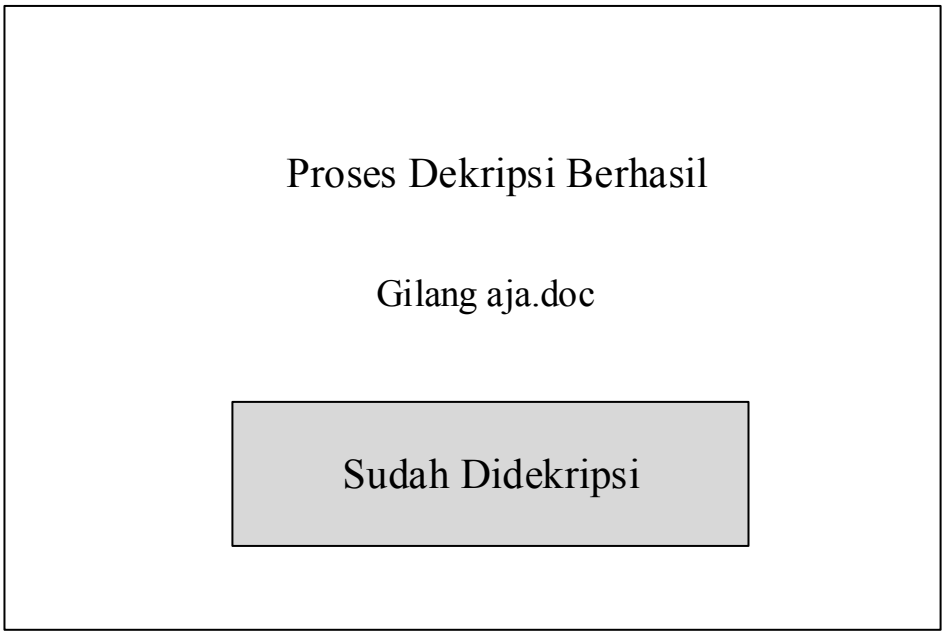
Gambar 3.11 *Desain Form* Hasil *Encoding*

3. Desain Form Dekripsi



Gambar 3.12 ***Desain Form Decoding***

4. Desain Form Hasil Dekripsi



Gambar 3.13 ***Desain Form*** **Hasil** ***Decoding***