

BAB III

METODOLOGI PENELITIAN

3.1 Analisis Sistem

Analisis sistem adalah sebuah teknik atau metode pemecahan masalah dengan cara menguraikan sistem yang sudah ada ke dalam komponen-komponen tertentu dengan tujuan untuk mengetahui bagaimana komponen-komponen tersebut bekerja dan saling berinteraksi satu sama lain untuk mencapai tujuan sistem. Pada tahap ini menjelaskan kebutuhan-kebutuhan untuk merancang bangun sistem atau perangkat lunak yang akan dibuat yang meliputi pemodelan kebutuhan sistem.

3.2 Analisis Masalah

Perkembangan teknologi informasi sangat pesat dan semakin memudahkan penggunaanya dalam beraktifitas, berkomunikasi atau berinteraksi melalui media, perlu diketahui bahwa dalam berkomunikasi akan melibatkan pengirim dan penerima data file. Semakin pesat perkembangan teknologi semakin pesat pula pelaku kejahatan yang terjadi, sehingga keamanan pada data atau informasi perlu mendapat perhatian khusus. Salah satunya dalam kasus defacing, pembobolan website dan pencurian data pada server yang sering kali terjadi, maka untuk itu perlu adanya peningkatan keamanan data guna menghindari perilaku kejahatan tersebut. Untuk menghindari kejahatan peretasan dan pencurian data tersebut, data-data yang dikirimkan ke server melalui aplikasi akan diamankan dan file yang terenkripsi dengan membutuhkan sebuah key untuk mengetahui isi informasi dari data tersebut.

3.3 Analisis Kebutuhan

3.3.1 Pemodelan Kebutuhan Sistem

Pemodelan kebutuhan sistem pada sistem kriptografi penyandian file dibutuhkan analisis terlebih dahulu, maka diperlukan pengetahuan terhadap kebutuhan-kebutuhan yang harus didapat dan diperlukan untuk membangun suatu sistem agar sebuah sistem tercapai. Pada pemodelan sistem ini terdapat kebutuhan fungsional sistem.

3.3.2 Kebutuhan Perangkat Keras (Hardware)

Adapun untuk kebutuhan spesifikasi perangkat keras yang digunakan dalam perancangan dan pembuatan aplikasi ini diantaranya :

A. Minimum Hardware

- a. Laptop/PC
- b. Monitor, Mouse, Keyboard
- c. Processor Core i3 2.2 GHz
- d. Memory 4GB DDR3
- e. Graphics 1.5 GHz

3.3.3 Kebutuhan Perangkat Lunak (Software)

Adapun perangkat lunak (*software*) yang digunakan untuk melakukan perancangan atau pemodelan dan pembuatan aplikasi diantaranya :

A. Minimum Software

- a. Sistem Operasi (Linux, Windows, Mac OS)
- b. PHP v8.0, v7.4
- c. MYSQL v5.7
- d. PHP Storm IDE v2022.2
- e. Docker v4.8
- f. Start UML v4
- g. Balsamiq v3
- h. Sybase Power Designer v15
- i. Google chrome v103.0

3.3.4 Pengguna (Brainware)

Adapun pengguna yang akan menjalankan aplikasi kriptografi terbagi dua yaitu:

1. Admin

Yakni super user yang bisa melakukan upload file dan hapus file.

2. Karyawan

Yakni user yang hanya bisa melakukan upload file.

3.3.5 Analisis Algoritma AES (Advanced Encryption Standard)

Plaintext (P) = PIRMANABDUROHMAN

Kunci (K) = INFORMATIKA12345

Input *plaintext* dan *key* dalam matriks 4x4

Tabel 3.1 Input plaintext dan key kedalam matriks

<i>PLAINTEXT</i>			
P	A	D	H
I	N	U	M
R	A	R	A
M	B	O	N
<i>KEY</i>			
I	R	I	2
N	M	K	3
F	A	A	4
O	T	1	5



Ubah *plaintext* dan *key* diatas menjadi hexadecimal

Tabel 3.2 Ubah plaintext dan key state 0 menjadi hexadecimal

<i>PLAINTEXT</i>			
50	41	44	48
49	4E	55	4D
52	41	52	41
4D	42	4F	4E
<i>KEY</i>			
49	52	49	32
4E	4D	4B	33
46	41	41	34
4F	54	31	35

Initial Round :

1. Transformasi *AddRoundKey*

Pada *AddRoundKey* ini dilakukan XOR antara *plaintext* dan *key state 0* :

Tabel 3.3 Operasi XOR antara plaintext dengan key state 0

PLAINTEXT				XOR	KEY STATE 0				=	AddRoundKey			
50	41	44	48		49	52	49	32		19	13	0D	7A
49	4E	55	4D		4E	4D	4B	33		07	03	1E	7E
52	41	52	41		46	41	41	34		14	00	13	75
4D	42	4F	4E		4F	54	31	35		02	16	7E	7B

a. Key Expansion atau Key Schedule

Pada step ini *key* yang tadi sudah di ubah kedalam hexadecimal, kemudian dilakukan substitusi dengan table S-box (Tabel 3.4), sehingga didapatkan hasil seperti tabel 3.3 yang merupakan tabel hasil *key expansion*, dimana tabel tersebut dipakai untuk mencari *key expansion* selanjutnya. Dalam ekspansi kunci ini terdapat tahapan khusus yang dikenal sebagai *AES key schedule* atau *Rijndael's key schedule*. Untuk tahapan *key schedule* bisa dilihat pada tabel 3.5 sampai dengan tabel 3.19.

Tabel 3.4 S-Box

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
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

Tabel 3.5 Hasil substitusi key dengan tabel S-Box

KEY				→	KEY STATE 0				→	Sub S-Box			
I	R	I	2		49	52	49	32		3B	00	3B	23
N	M	K	3		4E	4D	4B	33		2F	E3	B3	C3
F	A	A	4		46	41	41	34		5A	83	83	18
O	T	1	5		4F	54	31	35		54	20	C7	96

Pada table 3.5 diatas dilakukan *konversi* untuk key dari *plaintext* ke *hexadecimal*, setelah hasil konversi didapatkan maka substitusi hasil tersebut dengan tabel s-box yang ada pada tabel 3.4.

Tabel 3.6 RCON table atau key schedule

Round	1	2	3	4	5	6	7	8	9	10
Rcon[]	01	02	04	08	10	20	40	80	1b	36
	00	00	00	00	00	00	00	00	00	00
	00	00	00	00	00	00	00	00	00	00
	00	00	00	00	00	00	00	00	00	00

Tabel 3.6 diatas adalah RCON table yang digunakan untuk perhitungan mencari *key state*. Setelah *key state* ditemukan, maka *key state* tersebut digunakan untuk proses *addRoundKey* dengan melakukan XOR bersama hasil *mixColumn*.

Key Schedule :

Pada tahap ini akan dilakukan perhitungan untuk menemukan *key state* yang akan digunakan untuk mencari hasil *addRoundkey* nantinya. Gambaran perhitungannya sebagai berikut :

Tabel 3.7 RotWord dan Sub-Byte column terakhir Key State 0

KEY STATE 0				→	RotWord			→	Sub-Bytes	
49	52	49	32		32	→	33		C3	
4E	4D	4B	33		33		34		18	
46	41	41	34		34		35		96	
4F	54	31	35		35		32		23	

Tabel 3.7 diatas adalah proses *RotWord* atau pegeseran 1 baris ke bawah dengan cara mengambil kolom terakhir dari *key state 0* selanjutnya row pertama dipindahkan menjadi baris terakhir, row kedua menjadi baris pertama dst. Selanjutnya hasil *RotWord* dilakukan substitusi dengan tabel s-box yang ada pada tabel 3.4

Tabel 3.8 Perhitungan Key State 0 Row 1

<i>Key State 0 Column 1</i>					
49	XOR	C3	XOR	01	8B
42		18		00	5A
4E		96		00	D8
55		23		00	76

Tabel 3.8 diatas adalah perhitungan untuk mencari key state 1 dengan cara mengambil kolom pertama dari key state 0, kemudian lakukan proses xor dengan hasil dari tabel 3.7, selanjutnya hasil dari xor tersebut dilakukan proses xor kembali dengan RCON tabel yang ada pada tabel 3.6.

Tabel 3.9 Perhitungan Key State 0 Column 2 dan 3

Key State 0 Column 2				
8B	XOR	52	→	D9
5A		4D		17
D8		41		99
76		54		22
Key State 0 Column 3				
D9	XOR	49	→	90
17		4B		58
99		41		D8
22		31		13

Tabel 3.9 diatas adalah lanjutan untuk mencari key state 1 pada kolom 2 dan 3 yang diambil dari key state 0. Proses perhitungannya dengan cara melakukan proses xor antara hasil dari tabel 3.8 dengan kolom ke-2 key state 0, kemudian hasil dari perhitungan tersebut digunakan untuk proses xor dengan kolom ke-3 key state 0

Tabel 3.10 Perhitungan Key State 0 Column 4 dan hasil Key State 1

Key State 0 Column 4						KEY STATE 1			
90	XOR	32	→	A2	→	8B	D9	90	A2
58		33		6B		5A	17	58	6B
D8		34		EC		D8	99	D8	EC
13		35		26		76	22	13	26

2. Sub-Bytes

Tabel 3.11 Proses hasil addRoundKey subBytes

<i>Hasil AddRoundKey</i>					<i>Sub S-Box</i>			
19	13	0D	7A	→	D4	7D	D7	DA
07	03	1E	7E		C5	7B	72	F3
14	00	13	75		FA	63	7D	9D
02	16	7E	7B		77	47	F3	21

Table 3.11 diatas adalah proses *sub-bytes*, cara perhitungannya ambil *addRoundKey* yang ada pada tabel 3.3 dan lakukan proses substitusi dengan tabel s-box yang terdapat pada tabel 3.4

3. *ShiftRows*

Tabel 3.12 Pergeseran baris pertama

D4	7D	D7	DA	0 byte →	D4	7D	D7	DA
C5	7B	72	F3		C5	7B	72	F3
FA	63	7D	9D		FA	63	7D	9D
77	47	F3	21		77	47	F3	21

Tabel 3.12 baris pertama melakukan pergeseran ke kiri sebesar *0 byte*. Disebabkan nilai pergeserannya *0 byte*, maka tidak ada pergeseran dan state yang lain pun tidak mengalami pergeseran.

Tabel 3.13 Pergeseran baris kedua

	D4	7D	D7	DA	1 byte →	D4	7D	D7	DA
C5	7B	72	F3			7B	72	F3	C5
	FA	63	7D	9D		FA	63	7D	9D
	77	47	F3	21		77	47	F3	21

Tabel 3.13 baris kedua melakukan pergeseran ke kiri sebesar *1 byte*. Maka *state* akan mengalami perubahan. Pada baris kedua kolom pertama bergeser menjadi baris kedua kolom ke empat.

Tabel 3.14 Pergeseran baris ketiga

	D4	7D	D7	DA	2 byte →	D4	7D	D7	DA
C5	7B	72	F3			7B	72	F3	C5
FA	63	7D	9D			7D	9D	FA	63

77	47	F3	21		77	47	F3	21
----	----	----	----	--	----	----	----	----

Tabel 3.14 baris ketiga kolom pertama dan kolom kedua bergeser menjadi baris ketiga kolom ketiga dan kolom keempat. Pada kolom pertama baris ketiga bergeser menjadi kolom ketiga baris ketiga, sedangkan kolom kedua baris ketiga bergeser menjadi kolom keempat baris ketiga.

Tabel 3.15 Pergeseran baris keempat

				D4	7D	D7	DA	3 byte →	D4	7D	D7	DA
				C5	7B	72	F3		7B	72	F3	C5
				FA	63	7D	9D		7D	9D	FA	63
77	47	F3	21						21	77	47	F3

Tabel 3.15 baris keempat kolom keempat akan bergeser menjadi kolom pertama dan diikuti dengan kolom pertama yang menjadi kolom kedua, kolom kedua menjadi kolom ketiga, dan kolom ketiga menjadi kolom keempat.

4. MixColumn

Proses *mix column* menggunakan bilangan *polynomial* seperti yang di tampilkan oleh (Tabel 3.16)

UNIVERSITAS ISLAM NEGERI
SUNAN GUNUNG DJATI
BANDUNG

Tabel 3.16 Polynomial

02	03	01	01
01	02	03	01
01	01	02	03
03	01	01	02

Berikut merupakan proses *mix column* :

02	03	01	01	•	D4	=
----	----	----	----	---	----	---

01	02	03	01		7B	
01	01	02	03		7D	
03	01	01	02		21	

$$\begin{aligned}
&= (02 \cdot D4) \text{ XOR } (03 \cdot 7B) \text{ XOR } (01 \cdot 7D) \text{ XOR } (01 \cdot 21) \\
&= (01 \cdot D4) \text{ XOR } (02 \cdot 7B) \text{ XOR } (03 \cdot 7D) \text{ XOR } (01 \cdot 21) \\
&= (01 \cdot D4) \text{ XOR } (01 \cdot 7B) \text{ XOR } (02 \cdot 7D) \text{ XOR } (03 \cdot 21) \\
&= (03 \cdot D4) \text{ XOR } (01 \cdot 7B) \text{ XOR } (01 \cdot 7D) \text{ XOR } (02 \cdot 21) \\
&= C6 \text{ XOR } 19 \text{ XOR } 7D \text{ XOR } C5 = 67 \\
&= 63 \text{ XOR } E7 \text{ XOR } 93 \text{ XOR } C5 = D2 \\
&= 63 \text{ XOR } FE \text{ XOR } FA \text{ XOR } 54 = 33 \\
&= A5 \text{ XOR } FE \text{ XOR } 7D \text{ XOR } 91 = B7
\end{aligned}$$

02	03	01	01	•	7D	=
01	02	03	01		72	
01	01	02	03		9D	
03	01	01	02		77	

$$\begin{aligned}
&= (02 \cdot 7D) \text{ XOR } (03 \cdot 72) \text{ XOR } (01 \cdot 9D) \text{ XOR } (01 \cdot 77) \\
&= (01 \cdot 7D) \text{ XOR } (02 \cdot 72) \text{ XOR } (03 \cdot 9D) \text{ XOR } (01 \cdot 77) \\
&= (01 \cdot 7D) \text{ XOR } (01 \cdot 72) \text{ XOR } (02 \cdot 9D) \text{ XOR } (03 \cdot 77) \\
&= (03 \cdot 7D) \text{ XOR } (01 \cdot 72) \text{ XOR } (01 \cdot 9D) \text{ XOR } (02 \cdot 77) \\
&= 80 \text{ XOR } 81 \text{ XOR } 6B \text{ XOR } 43 = 29 \\
&= 40 \text{ XOR } FE \text{ XOR } BD \text{ XOR } 43 = 40 \\
&= 40 \text{ XOR } 7F \text{ XOR } D6 \text{ XOR } C1 = 29 \\
&= C0 \text{ XOR } 7F \text{ XOR } 6B \text{ XOR } 86 = 52
\end{aligned}$$

02	03	01	01	•	D7	=
01	02	03	01		F3	
01	01	02	03		FA	
03	01	01	02		47	

$$\begin{aligned}
&= (02 \cdot D7) \text{ XOR } (03 \cdot F3) \text{ XOR } (01 \cdot FA) \text{ XOR } (01 \cdot 47) \\
&= (01 \cdot D7) \text{ XOR } (02 \cdot F3) \text{ XOR } (03 \cdot FA) \text{ XOR } (01 \cdot 47) \\
&= (01 \cdot D7) \text{ XOR } (01 \cdot F3) \text{ XOR } (02 \cdot FA) \text{ XOR } (03 \cdot 47)
\end{aligned}$$

$$\begin{aligned}
&= (03 \cdot D7) \text{ XOR } (01 \cdot F3) \text{ XOR } (01 \cdot FA) \text{ XOR } (02 \cdot 47) \\
&= FA \text{ XOR } 20 \text{ XOR } 30 \text{ XOR } 6F = 85 \\
&= 7D \text{ XOR } C4 \text{ XOR } 50 \text{ XOR } 6F = 86 \\
&= 7D \text{ XOR } EF \text{ XOR } 60 \text{ XOR } B1 = 43 \\
&= 93 \text{ XOR } EF \text{ XOR } 30 \text{ XOR } DE = 92
\end{aligned}$$

02	03	01	01	•	DA	=
01	02	03	01		C5	
01	01	02	03		63	
03	01	01	02		F3	

$$\begin{aligned}
&= (02 \cdot DA) \text{ XOR } (03 \cdot C5) \text{ XOR } (01 \cdot 63) \text{ XOR } (01 \cdot F3) \\
&= (01 \cdot DA) \text{ XOR } (02 \cdot C5) \text{ XOR } (03 \cdot 63) \text{ XOR } (01 \cdot F3) \\
&= (01 \cdot DA) \text{ XOR } (01 \cdot C5) \text{ XOR } (02 \cdot 63) \text{ XOR } (03 \cdot F3) \\
&= (03 \cdot DA) \text{ XOR } (01 \cdot C5) \text{ XOR } (01 \cdot 63) \text{ XOR } (02 \cdot F3) \\
&= BB \text{ XOR } 19 \text{ XOR } AF \text{ XOR } A2 = AF \\
&= D0 \text{ XOR } E7 \text{ XOR } E0 \text{ XOR } A2 = 75 \\
&= D0 \text{ XOR } FE \text{ XOR } 45 \text{ XOR } FD = 96 \\
&= 6B \text{ XOR } FE \text{ XOR } AF \text{ XOR } 57 = 6D
\end{aligned}$$

Tabel 3.17 Hasil Mix Column

Hasil <i>Mix Column</i>			
67	29	85	AF
D2	40	86	75
33	29	43	96
B7	52	92	6D

Proses diatas adalah proses mix column dengan tabel *polynomial* (tabel 3.16). Perhitungan nya dengan cara menggunakan metode perhitungan *dot product* untuk tabel *polynomial* dan tiap-tiap kolom dari hasil shiftRow (tabel 3.15). Setelah hasil *dot product* tersebut ditemukan, selanjutnya dilakukan proses *xor* untuk setiap hasil *dot product*.

5. AddRoundKey

Tabel 3.18 Operasi XOR Hasil Mix Column dengan Round Key 1

Hasil <i>Mix Column</i>					<i>Round Key 1</i>					Hasil <i>AddRoundKey</i>			
67	29	85	AF	XOR	8B	D9	90	A2		EC	F0	15	D

D2	40	86	75		5A	17	58	6B		88	57	DE	1E
33	29	43	96		D8	99	D8	EC		EB	B0	9B	7A
B7	52	92	6D		76	22	13	26		C1	70	81	4B

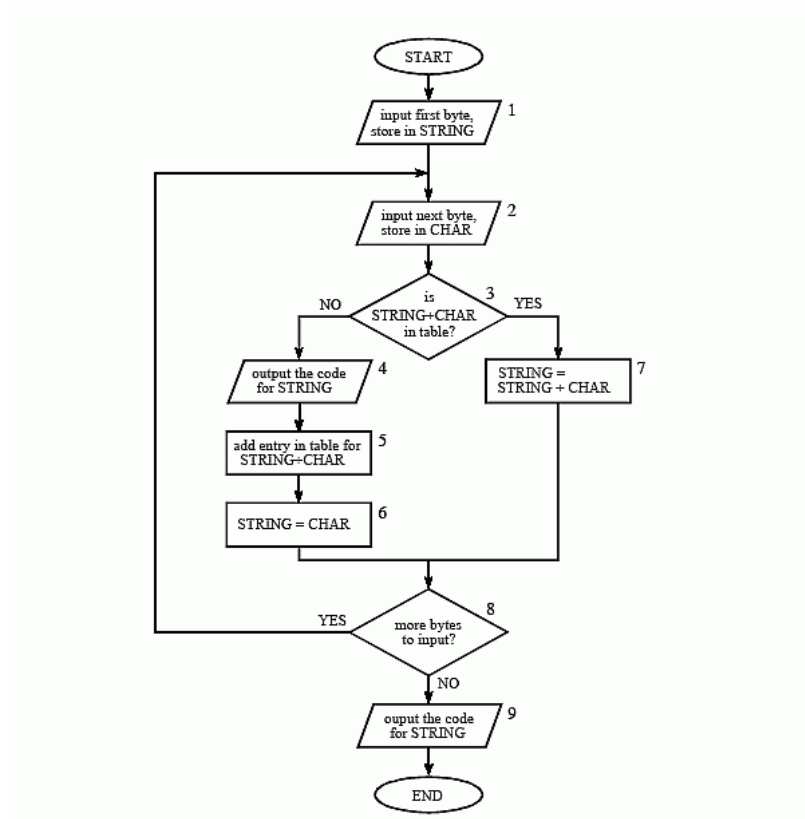
Pada transformasi *addRoundKey*, maka akan dilakukan operasi XOR antara hasil state 128 bit dengan kunci ronde hasil ekspansi kunci (*cipher key*). Awal proses enkripsi, untuk 128 bit *plaintext* akan dilakukan operasi XOR dengan 128 bit kunci yang asli. Setelah itu pada 128 bit *plaintext* akan dilakukan 3 transformasi lainnya *Sub Bytes*, *Shift Rows* dan *Mix Columns*. Adapun pada transformasi *addRoundKey*, maka untuk 128 bit yang sudah melewati tiga transformasi tersebut akan dilakukan operasi XOR terhadap kunci ronde hasil ekspansi kunci pertama. Kemudian untuk hasil *addRoundKey* merupakan state untuk ronde 1. Kemudian pada state 1 ini akan mengalami kembali ketiga transformasi tersebut. Pada transformasi *addRoundKey* yang berikutnya, maka 128 bit yang telah mengalami perubahan pada ketiga transformasi tersebut akan dilakukan operasi XOR kembali dengan kunci hasil ekspansi kunci kedua dan seterusnya, sampai sebanyak 10 kali namun pada ronde terakhir atau final Round (ronde 10) tidak dilakukan transformasi *MixColumns* [27].

3.3.6 Analisis Algoritma LZW (Lempel-Ziv-Welch)

Algoritma LZW (*Lempel-Ziv-Welch*) ini melakukan kompresi dengan menggunakan kode table 256 hingga 4095 untuk mengkodekan pasangan byte atau string yang dapat dikodekan dengan mengacu pada string yang telah muncul sebelumnya dalam teks. Algoritma kompresi LZW secara lengkap [28]:

1. Kamus (*Dictionary*) diinisialisasi dengan semua karakter dasar yang ada: {A..Z, a..z, 0..9}
2. W = karakter pertama dalam stream karakter.
3. K = karakter berikutnya dalam stream karakter.
4. Lakukan pengecekan apakah (W+K) terdapat dalam Kamus:
 - a. Jika ya, maka W = W+K (gabungan W & K menjadi string baru)
 - b. Jika tidak, maka:
 - Output sebuah kode untuk menggantikan string W

- Tambahkan string (W+K) ke dalam Kamus dan berikan nomor/kode berikutnya yang belum digunakan dalam Kamus untuk string tersebut
 - W -> K
- c. Lakukan pengecekan apakah masih ada karakter berikutnya dalam stream karakter
- Jika ya, maka Kembali ke langkah 2.
 - Jika tidak, maka output kode yang menggantikan string W, lalu terminasi proses (stop), keseluruhan string biner diproses [29]



Gambar 3.1 Flowchart proses kompresi [29]

Sebagai contoh, string “ABBABABAC” akan dikompresi dengan LZW. Isi dictionary pada diset dengan tiga karakter dasar yang ada: “A”, “B”, dan “C”. Tahapan proses kompresi ditunjukkan pada gambar 3.2 berikut:

Posisi,[Kode] Huruf	Gabungan Posisi	Huruf	[Kode] dictionary	Output
inisialisasi		A	[1] A	
		B	[2] B	
		C	[3] C	
		D	[4] D	
1,[1] A	1+2	AB	[5] AB	[1]
2,[2] B	2+3	BB	[6] BB	[2]
3,[2] B	3+4	BA	[7] BA	[2]
4,[1] A	4+5	AB	Ada	
,[5] AB	4+5+6	ABA	[8] ABA	[5]
6,[1] A	6+7	AB	Ada	
,[5] AB 6	6+7+8	ABA	Ada	
,[8] ABA	6+7+8+9	ABAC	[9] ABAC	[8]
9,[3] C	9+10	CA	[10] CA	[3]
10,[1] A	10+12	AA	[11] AA	[1]
11,[1] A	11+12	AC	[12] AC	[1]
12,[3] C	12+13	CD	[13] CD	[3]
13,[4] D	13+14	DD	[14] DD	[4]
14,[4] D	14+15	DD	Ada	
,[14] DD	14+15+	Habis		[14]

Gambar 3.2 Tahapan kompresi LZW [29]

Dari sini diperoleh output berupa kode dan dictionary berikut:

Kode = [1] [2] [2] [5] [8] [3] [1] [1] [3] [4] [14]

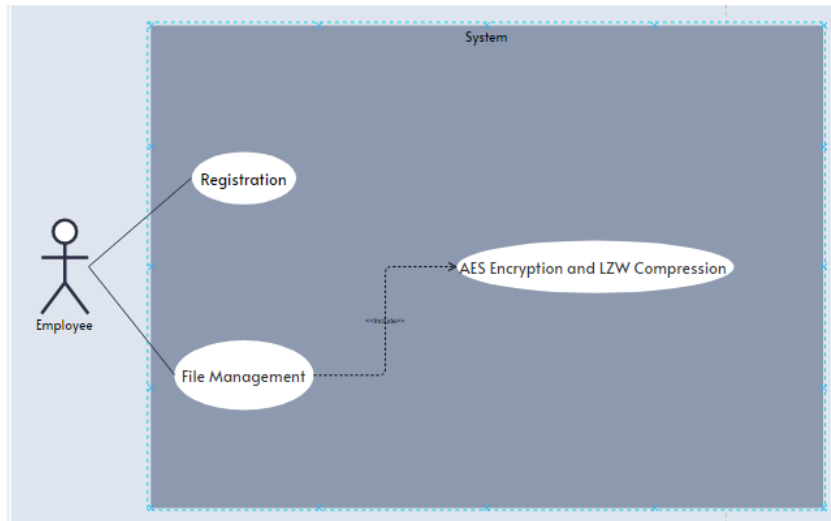
Dictionary / Kamus = [1] A [2] B [3] C [4] D [5] AB [6] BB [7] BA [8] ABA
[9] ABAC [10] CA [11] AA [12] AC [13] CD [14] DD

3.4 Desain Sistem

Desain sistem merupakan gambaran dari keseluruhan rincian bagaimana sistem akan berjalan. Tujuan desain sistem untuk mengetahui bagaimana sistem akan memenuhi kebutuhan dan tujuan yang telah diciptakan atau dibuat. Dalam penelitian yang dilakukan ini, desain sistem terdiri dari perancangan use case diagram, activity diagram, dan class diagram

3.4.1 Use Case Diagram

Use Case Diagram menggambarkan kebutuhan sistem secara fungsional dengan mengidentifikasi aktor-aktor yang terlibat dan berinteraksi dengan fungsi dasar pada sistem. *Use Case Diagram* akan menjelaskan fungsi apa saja yang dikerjakan oleh sistem. Aktor yang terlibat pada penelitian ini adalah pengirim dan penerima namun didefinisikan dengan satu aktor karena aktor tersebut dapat melakukan pengiriman dan penerimaan data file sekaligus. Adapun use case dalam program ini ditujukan pada gambar berikut.



Gambar 3.3 Use Case Diagram Sistem

3.4.2 Definisi Aktor

Deskripsi pendefinisian aktor pada penggunaan aplikasi keamanan data file sebagai berikut:

Tabel 3.19 Definisi Aktor Use Case

No.	Aktor	Deskripsi
1	Employee	User adalah orang yang menggunakan aplikasi keamanan data file untuk melakukan proses upload dan download data file.

3.4.3 Definisi Use Case

Berikut adalah deskripsi pendefinisian *use case* pada penggunaan aplikasi keamanan data file seperti yang ditampilkan tabel 3.20 berikut:

Tabel 3.20 Definisi Use Case

No	Use Case	Deskripsi
1	<i>Registration</i>	Merupakan halaman yang digunakan untuk mendaftar menjadi user aplikasi
2	<i>File Management</i>	Merupakan halaman yang digunakan untuk menampilkan mengatur semua <i>file</i> pada aplikasi
3	<i>Download File</i>	Merupakan proses untuk mendownload file yang sudah pernah di unggah ke sistem
4	Delete File	Merupakan proses untuk menghapus file dari sistem
5	Upload File	Merupakan halaman yang digunakan oleh user untuk

		mem-upload file
6	<i>AES Encryption & LZW Compression</i>	Merupakan proses enkripsi data dengan algoritma AES dan kompresi data dengan algoritma LZW yang dijalankan ketika meng- <i>upload/download file</i>

3.4.4 Skenario Use Case

Berikut adalah deskripsi pendefinisian skenario *use case* pada penggunaan aplikasi keamanan data file sebagai berikut :

a. Skenario use case registration

Tahap ini adalah proses *ragistration* dengan skenario seperti pada gambar 3.21 berikut:

Tabel 3.21 Skenario Use Case Registration

Nama Use Case	<i>Registration</i>
Deskripsi	Digunakan untuk mendaftar user ke sistem
Aktor	<i>User</i>
Kondisi Awal	User belum memiliki akun untuk mengakes sistem
Skenario	
Aktor	Sistem
1. Mengklik tombol registrasi	2. Sistem menampilkan halaman registrasi
3. Mengisi data nama, email, dan password	4. Sistem melakukan verifikasi email, jika belum ada di database maka user akan didaftarkan sebagai user baru
Kondisi Akhir	Akun user berhasil dibuat dan bisa digunakan untuk proses login

b. Skenario *File Management*

Tahap ini adalah proses *file management* dengan skenario seperti pada gambar 3.22 berikut:

Tabel 3.22 File Management

Nama Use Case	<i>File Management</i>
Deskripsi	Digunakan untuk melihat data file yang telah diupload yang tersimpan di cloud storage.
Aktor	<i>User</i>
Kondisi Awal	<i>User</i> diharuskan untuk login
Skenario	
Aktor	Sistem
1. memilih menu List File	2. Menampilkan semua file yang pernah diupload

Kondisi Akhir	Sistem menunjukkan semua data yang terdapat pada sistem
----------------------	---

c. Skenario Download File

Tahap ini adalah proses *download file* dengan skenario seperti pada gambar 3.23 berikut:

Tabel 3.23 Download File

Nama Use Case	<i>Download File</i>
Deskripsi	Digunakan untuk menghapus data file yang terenkripsi ke cloud storage
Aktor	<i>User</i>
Kondisi Awal	<i>User</i> diharuskan untuk login
Skenario	
Aktor	Sistem
1. memilih menu List File	2. Menampilkan semua file yang pernah diupload
3. Mengklik tombol Download	4. Menampilkan form input password
4. Mengisi password	5. Jika password benar, maka sistem memulai deskripsi dan men-download file
Kondisi Akhir	File di download

d. Skenario Delete File

Tahap ini adalah proses *delete file* dengan skenario seperti pada gambar 3.24 berikut:

Tabel 3.24 Delete File

Nama Use Case	<i>Delete File</i>
Deskripsi	Digunakan untuk menghapus data file yang terenkripsi ke cloud storage
Aktor	<i>User</i>
Kondisi Awal	<i>User</i> diharuskan untuk login
Skenario	
Aktor	Sistem
1. memilih menu List File	2. Menampilkan semua file yang pernah diupload
3. Mengklik tombol Hapus	4. Menampilkan form input password
4. Mengisi password	5. Jika password benar, maka sistem memulai menghapus file
Kondisi Akhir	Data kehapus dari sistem dan cloud storage

e. Skenario *Use Case Upload File*

Tahap ini adalah proses *upload file* dengan skenario seperti pada gambar 3.25 berikut:

Tabel 3.25 Skenario Use Case Upload File

Nama Use Case	Unggah Data file
Deskripsi	Digunakan untuk mengirim data file yang terenkripsi ke cloud storage
Aktor	<i>User</i>
Kondisi Awal	<i>User</i> diharuskan untuk login
Skenario	
Aktor	Sistem
1. memilih menu kirim data file	2. Menampilkan form halaman untuk unggah data file.
3. memasukan file dan password. Kemudian mengklik tombol upload	4. Sistem melakukan verifikasi password,
	5. Jika password benar, maka sistem memulai enkripsi isi data file dengan menggunakan EAS dan Bluefish
	6. File yang sudah dienkripsi kemudian di unggah ke cloud storage Amazon S3
Kondisi Akhir	Sistem mengirim data file terenkripsi dan menyimpan data ke dalam database serta di cloud storage

f. Skenario *Use Case AES Encryption and LZW Compression*

Tahap ini adalah proses *AES Encryption and LZW Compression* dengan skenario seperti pada gambar 3.26 berikut:

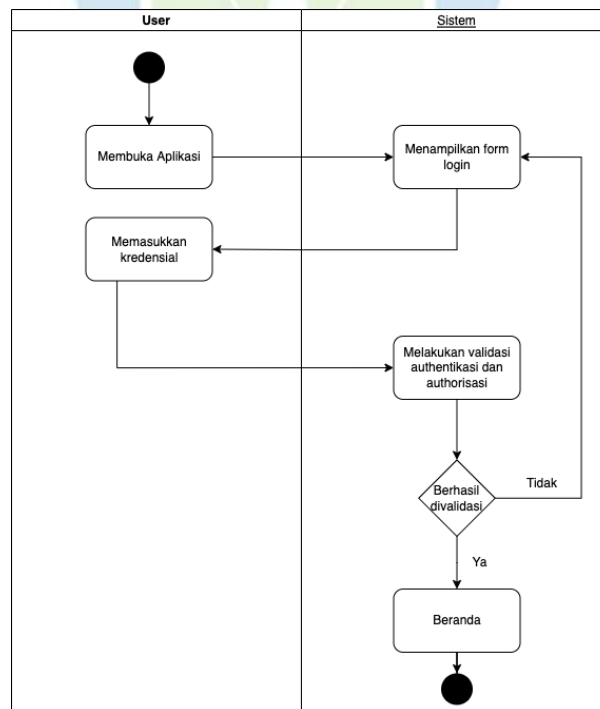
Tabel 3.26 Skenario Use AES Encryption and LZW Compression

Nama Use Case	<i>AES Encryption and LZW Compression</i>
Deskripsi	Dijalankan ketika proses <i>upload</i> atau <i>download</i> data, sistem akan meng-enkripsi file dengan algoritma <i>AES</i> dan meng-kompresi dengan algoritma <i>LZW</i>
Aktor	<i>Sistem</i>
Kondisi Awal	Aksi <i>Upload / Download</i> dijalankan
Skenario	
	Sistem

	Menjalankan proses enkripsi dan kompresi
Kondisi Akhir	Sistem mengirim data file terenkripsi dan menyimpan data ke dalam database serta di cloud storage

3.4.5 Perancangan Activity Diagram

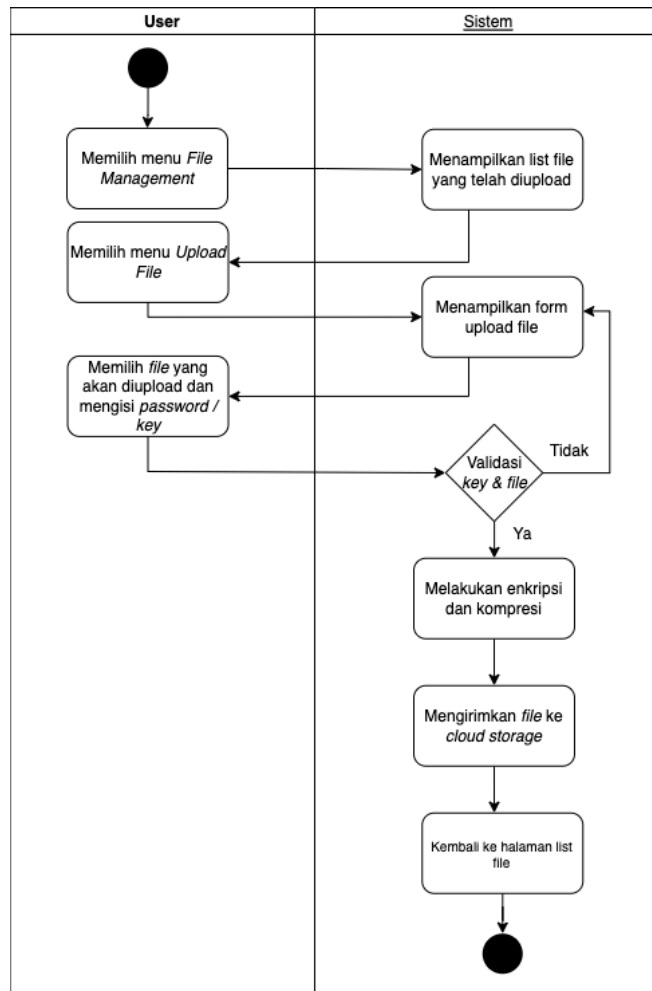
Activity Diagram merupakan bentuk khusus dari *state machine* yang bertujuan memodelkan komputasi-komputasi dan aliran-aliran kerja yang terjadi dalam sistem/perangkat lunak yang sedang dikembangkan. *Activity Diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi dan bagaimana keseluruhan aktivitas berakhir. Berikut skema diagram aktifitas yang telah dirancang dan akan dijelaskan oleh gambar-gambar diagram activity



Gambar 3.4 Aktivitas autentikasi dan authorisasi user

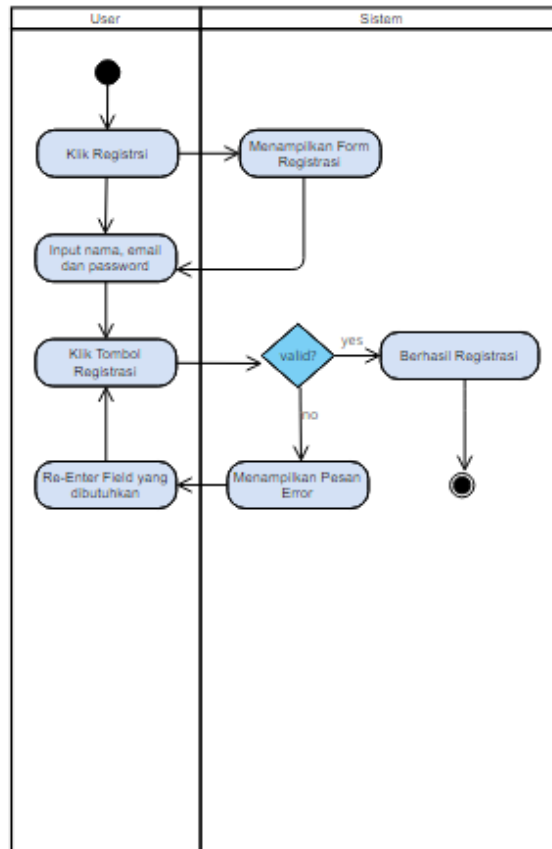
Pada gambar 3.4 diatas menggambarkan proses yang terjadi pada login di sistem user. Terdapat beberapa komponen yang perlu diisi oleh user, yaitu email dan password. Proses ini merupakan gambaran dari aktivitas aktor melakukan *login* untuk dapat masuk ke dalam halaman utama sistem. Pada gambar diatas menunjukkan bahwa aktivitas pertama, *user* harus masukkan *username* dan

password dengan benar. Proses selanjutnya adalah proses autentifikasi untuk mengecek keabsahan data yang diinput oleh user. Jika *username* dan *password* yang dimasukkan salah, maka *user* akan kembali mengisi hingga *username* dan *password* benar dan tepat penulisannya.



Gambar 3.5 Aktivitas Upload File

Pada gambar 3.5 diatas menggambarkan proses *upload file* yang bertujuan untuk mengamankan file yang akan disimpan di *cloud storage*. Proses ini diawali dengan memilih menu *File Management* yang kemudian sistem akan menampilkan daftar berkas yang berada di *cloud storage*, lalu kemudian memilih menu *upload file* lalu kemudian sistem akan menampilkan *form* yang akan *user* isi inputan berupa *file*, dan password atau *key*. Pada proses ini terdapat validasi data, apabila data berhasil tervalidasi maka sistem akan melakukan proses enkripsi dan kompresi lalu setelah berhasil maka *file* tersebut akan terkirim ke *cloud storage*.

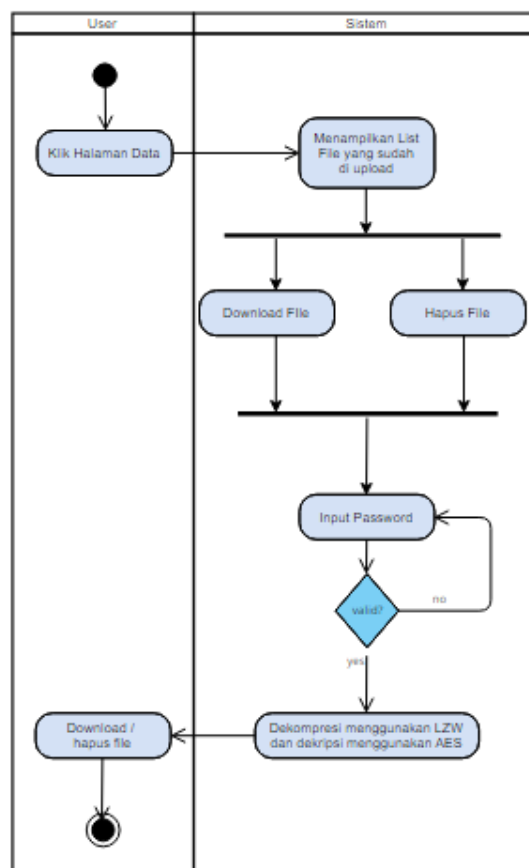


Gambar 3.6 Diagram aktivitas Registrasi

Pada gambar 3.6 diatas menggambarkan proses *registrasi* yang bertujuan untuk membuat akun pengguna.

a. Activity Diagram View, Download dan Hapus Data file

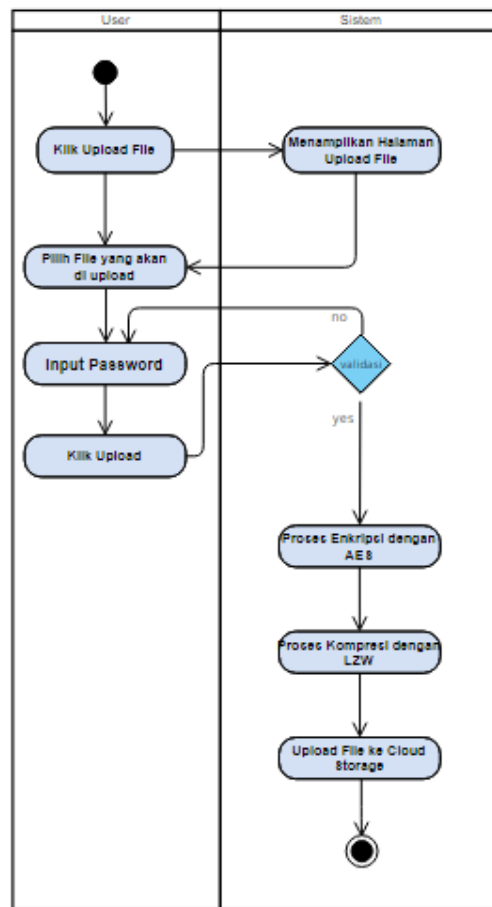
Pada aktivitas ini menunjukan alur dalam melihat file yang sudah diunggah, melakukan download file dan proses untuk menghapus data. Dalam proses men-download dan menghapus file yang sudah di upload, memerlukan password untuk pengecekan keabsahan user. Pada gambar 3.7 berikut ini adalah aktivitas diagram pada proses view data, download, dan hapus.



Gambar 3.7 Diagram Aktivitas view data, download, dan hapus data

b. Activity Diagram Unggah Data file

Proses unggah merupakan proses yang harus dilakukan oleh user ketika ingin menyimpan data ke cloud storage. Dalam proses ini user harus memasukan file dan password. Pada bagian backend process terdapat fungsi untuk melakukan enkripsi data file. Enkripsi file dilakukan dengan dua algoritma, yaitu algoritma blowfish kemudian dilanjutkan dengan algoritma blowfish. Adapun Activity diagram Kirim data file adalah gambaran dari saat user akan mengirimkan data file yang terenkripsi. Untuk proses unggah data file diperlihatkan seperti gambar 3.8 berikut.



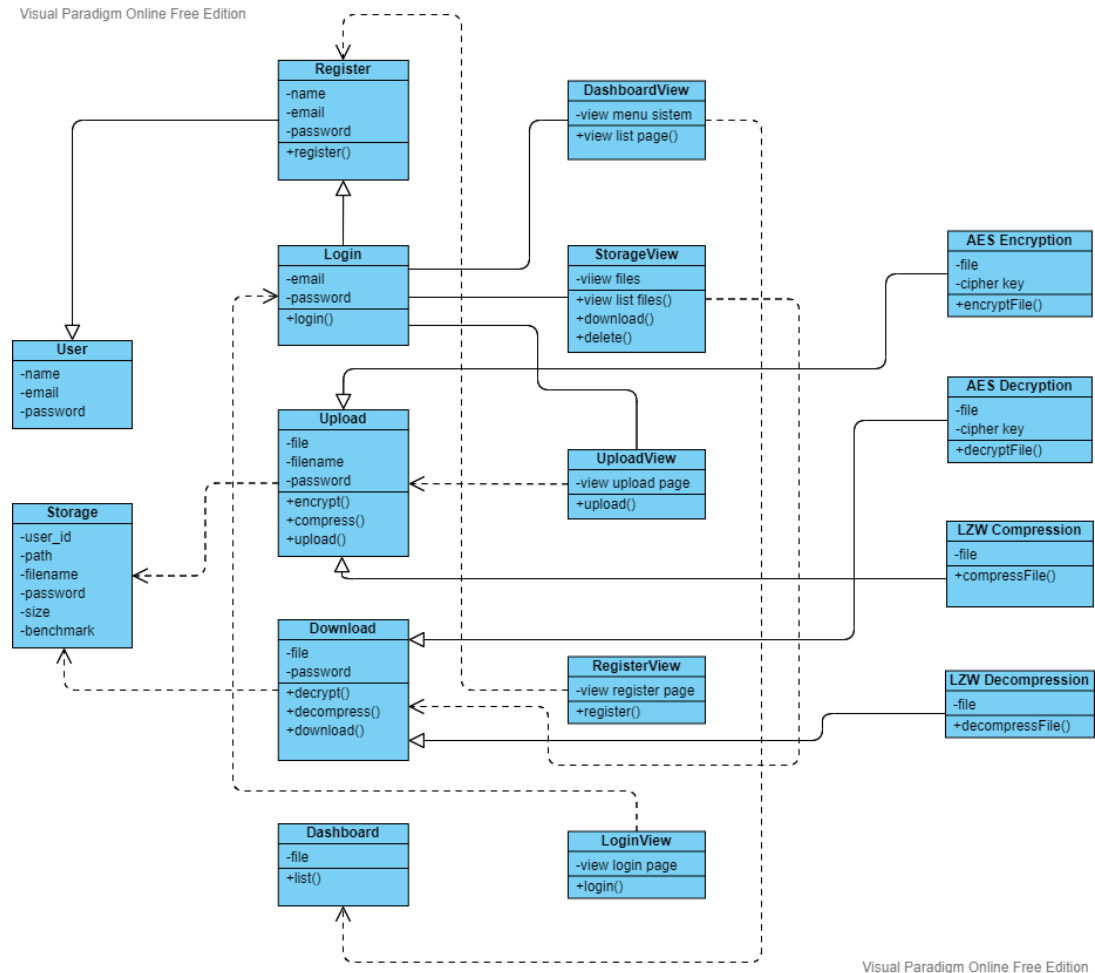
Gambar 3.8 Activity Diagram Upload Data file

Pada gambar activity diagram kirim data file, *user* perlu memilih menu upload data file jika ingin mengirimkan data file terenkripsi. Sebelum mengirimkan data file terenkripsi, *user* perlu menginput data file dan password yang valid. Setelah nya *user* meng-klik *button* upload data file, untuk memulai proses pengenkripsian sekaligus pengiriman data file/file yang sudah diinputkan. Hasil dari proses ini akan akan mengubah data file/file yang dikirim menjadi *ciphertext*. Data ini kemudian akan diupload ke cloud storage Amazon S3.

3.4.6 Perancangan *Class Diagram*

Class Diagram merupakan salah satu dari diagram UML. *Class diagram* menggambarkan struktur objek sistem dan menunjukkan kelas objek yang menyusun sistem juga hubungan antara kelas objek tersebut. *Class Diagram* menampilkan kelas – kelas yang digunakan didalam aplikasi enkripsi, dekripsi,

kompresi dan dekompresi file dengan menggunakan AES (*Advanced Encryption Standard*) dan LZW (*Lempel-Ziv-Welch*). Hal ini memberikan gambaran tentang sistem dan relasi apa yang terjadi didalamnya. Adapun *class diagram* pada sistem yang dibangun adalah seperti yang ditampilkan pada gambar 3.9 berikut.



Gambar 3.9 Class Diagram

3.4.7 Perancangan *Sequence Diagram*

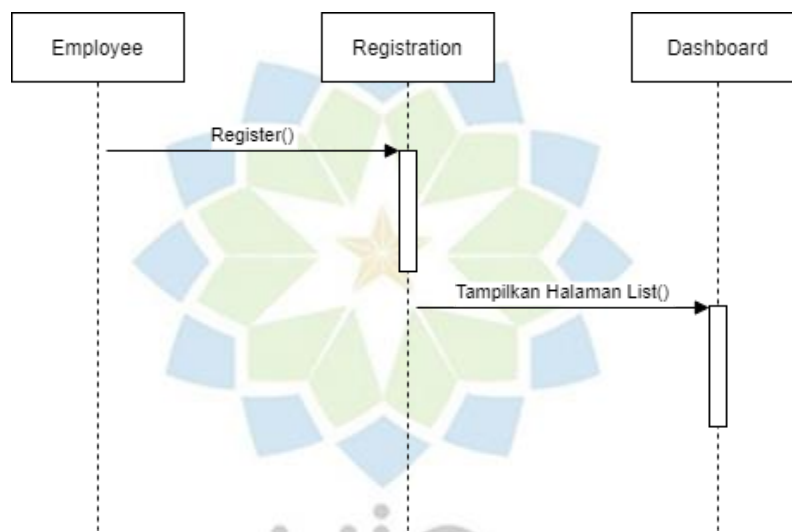
Sequence diagram menggambarkan interaksi antar masing-masing objek pada setiap use case dalam urutan waktu. Interaksi ini berupa pengiriman serangkaian data antar objek-objek yang saling berinteraksi. Berikut ini adalah diagram-diagram *sequence* untuk sistem yang dibangun.

a. *Sequence Diagram Registration*

Proses registrasi dimulai dengan mengisi nama user, email, password dan konfirmasi password. Sistem akan melakukan pengecekan email yang diinput

apakah sudah terdaftar dalam sistem atau tidak. Jika data email yang dimasukan sudah terdaftar dalam sistem, maka sistem akan meminta user untuk mengganti dengan email lain yang belum terdaftar pada sistem. Selanjutnya sistem juga akan mengecek kecocokan password dengan konfirmasi password yang ada. Jika tidak sesuai atau cocok, maka sistem akan meminta user untuk mengecek kembali inputan password tersebut.

Gambar 3.10 dibawah adalah gambar untuk squence diagram untuk proses *registration*.

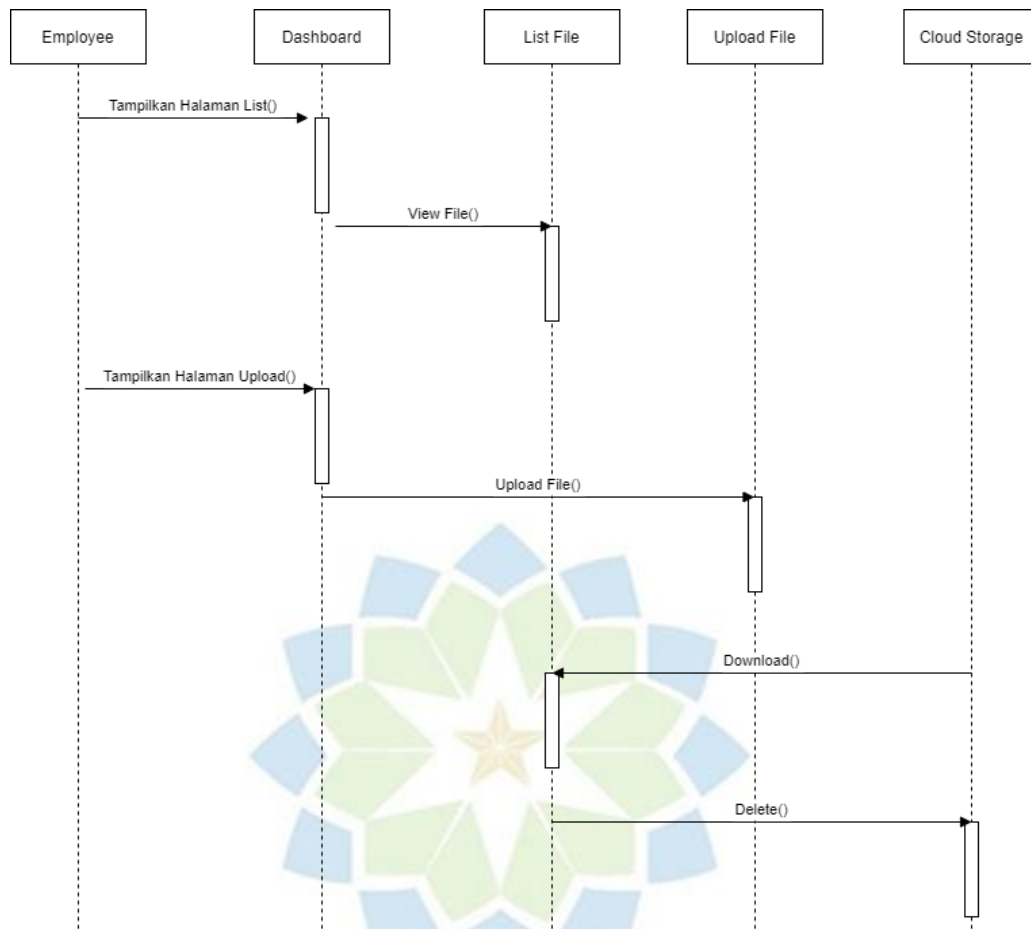


Gambar 3.10 Sequence Diagram Registration

b. Sequence Diagram *File Management*

Pada proses *file management*, terdapat tombol *upload* untuk mengirim file ke *cloud storag* dan data file yang pernah di *upload* akan ditampilkan dalam bentuk list atau daftar yang diurutkan berdasarkan tanggal terakhir upload. Pada masing-masing file terdapat dua tombol untuk download dan untuk menghapus file. Sebelum proses download ataupun melakukan proses penghapusan data, user diminta untuk mengisi password yang valid agar proses dapat dijalankan oleh sistem.

Gambar 3.11 dibawah adalah gambar untuk squence diagram untuk proses *file management*.

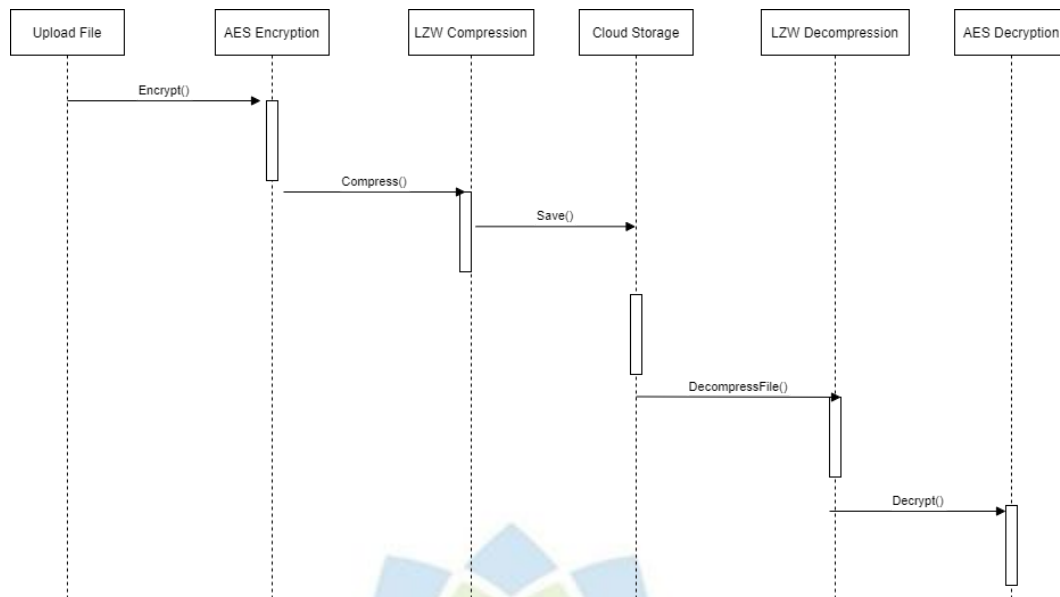


Gambar 3.11 Sequence Diagram File Management

c. Sequence Diagram *AES Encryption* dan *LZW Compression*

Proses ini merupakan proses untuk enkripsi dan dekripsi yang dilakukan oleh sistem. Pada proses ini, file yang akan akan diunggah terlebih dahulu dilakukan proses enkripsi dengan algoritma AES. Hasil dari enkripsi ini akan dikompresi oleh algoritma LZW. Setelah proses enkripsi berhasil maka file bisa diunggah ke *cloud storage*. Pada bagian ini juga terdapat komponen untuk mengisi password yang valid. Proses dekripsi dan dekompresi dijalankan ketika file akan diunduh oleh *actor*. Berikut ini adalah diagram sequence proses upload file.

Gambar 3.12 dibawah adalah gambar untuk squence diagram untuk proses *AES Encryption* dan *LZW Compression*.



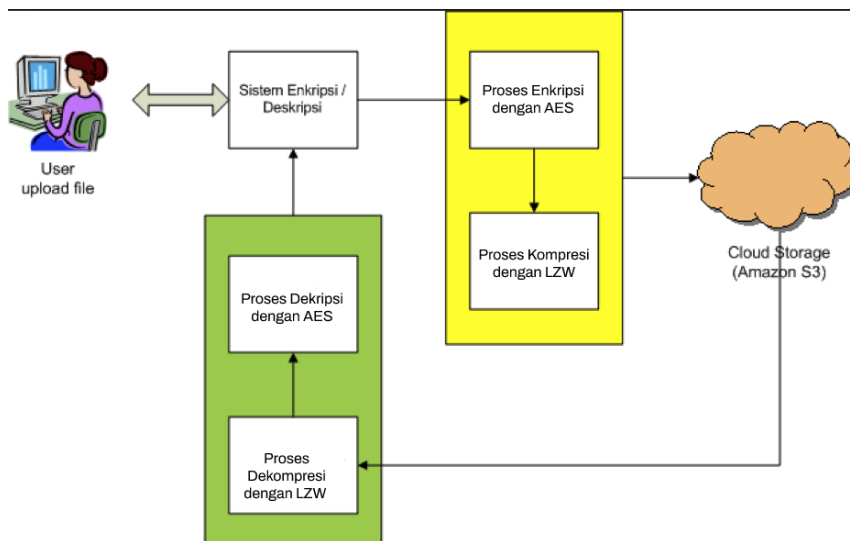
Gambar 3.12 Sequence Diagram AES Encryption dan LZW Compression

3.5 Prototype Sistem

Prototype sistem merupakan pembuatan rancangan sementara yang difokuskan untuk memberikan gambaran kepada client tentang sistem yang sedang dikembangkan.

3.5.1 Arsitektur Sistem Usulan

Pada sistem ini dibangun sistem enkripsi dan dekripsi pada file yang diupload ke cloud storage. Penggunaan sistem ini dapat memberikan keamanan data dari para user yang tidak diinginkan (*digital attacker*). Dalam membangun sistem ini menerapkan algoritma kriptografi untuk proses enkripsi dan dekripsi, dan algoritma LZW untuk proses kompresi dan dekompresi. Pada saat proses enkripsi, file upload akan di enkripsi dengan algoritma AES yang kemudian dilanjutkan oleh proses kompresi dengan algoritma LZW. Sedangkan pada proses dekripsi dilakukan pertama oleh algoritma AES, hasilnya kemudian dilanjutkan lagi oleh algoritma LZW untuk proses dekompresi. Adapun arsitektur sistem yang akan dibangun ditunjukkan pada gambar 3.13 berikut ini.



Gambar 3. 13 Arsitektur Sistem Pengamanan File

Sistem ini dibangun berbasis website dengan framework laravel yang diakses menggunakan *web browser* melalui jaringan internet. Sistem yang berbasis website memiliki kelebihan dalam proses akses bagi user.

Pada gambar 3.13 ditunjukkan bahwa dalam sistem keamanan file ini terdapat dua proses yang harus ada yaitu proses enkripsi, deskripsi dan proses kompresi, dekompresi. Proses enkripsi merupakan sebuah proses yang melakukan perubahan sebuah kode dari yang terbaca menjadi sebuah kode yang tidak bisa terbaca. Dalam sistem keamanan yang dibangun, proses enkripsi dilakukan oleh algoritma AES dan proses kompresi dilakukan oleh algoritma LZW.

Proses kedua adalah proses dekripsi, yaitu proses mengubah *ciphertext* kembali ke dalam bentuk *plaintext*. Pada sistem yang dibangun, proses dekripsi menerapkan algoritma AES dan proses dekompresi dilakukan menggunakan algoritma LZW.

3.5.2 Perancangan User Interface

Pada bagian perancangan ini, digambarkan antarmuka yang terdapat dalam sistem yang akan dibangun. Perancangan antarmuka disesuaikan dengan kebutuhan dari pengguna berdasarkan dari hasil analisis pada bab sebelumnya. Berikut ini adalah rancangan *user interface* (UI) masing-masing bagian pada sistem.

a. Perancangan UI Login

Halaman login adalah halaman pertama kali muncul ketika aplikasi diakses. Pada halaman ini terdapat bagian untuk inputan email, password dan satu tombol untuk melakukan proses login. Berikut ini adalah perancangan UI untuk halaman login. Untuk perancangan UI login di tampilkan pada gambar 3.14 berikut.

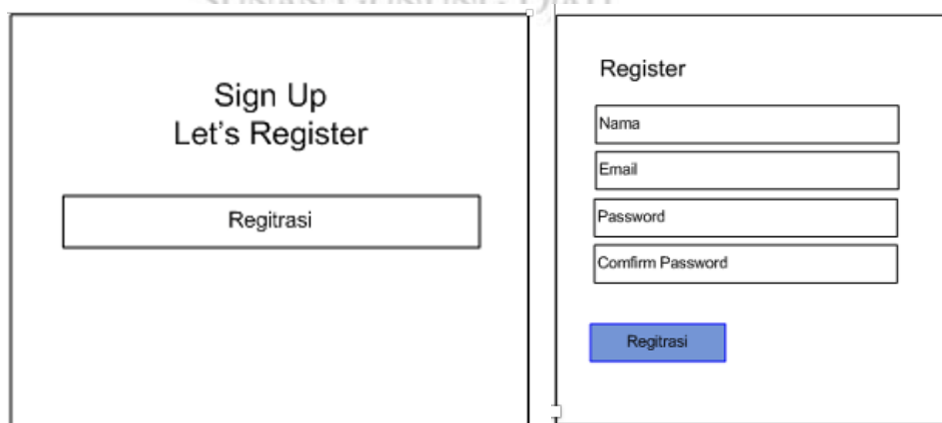


The image shows a login form with a title 'Login' at the top. Below the title are two input fields: 'Email' and 'Password'. At the bottom of the form is a blue button labeled 'Login'.

Gambar 3.14 Perancangan UI Login

b. Perancangan UI Registrasi

Perancangan UI registrasi merupakan halaman yang akan digunakan *user* untuk melakukan proses registrasi akun baru. Pada halaman ini terdiri dari komponen inputan nama pengguna, email pengguna, inputan password, inputan konfirmasi password, serta satu tombol registrasi. Adapun rancangan dari user interface halaman registrasi ditunjukkan pada gambar 3.15 berikut ini.

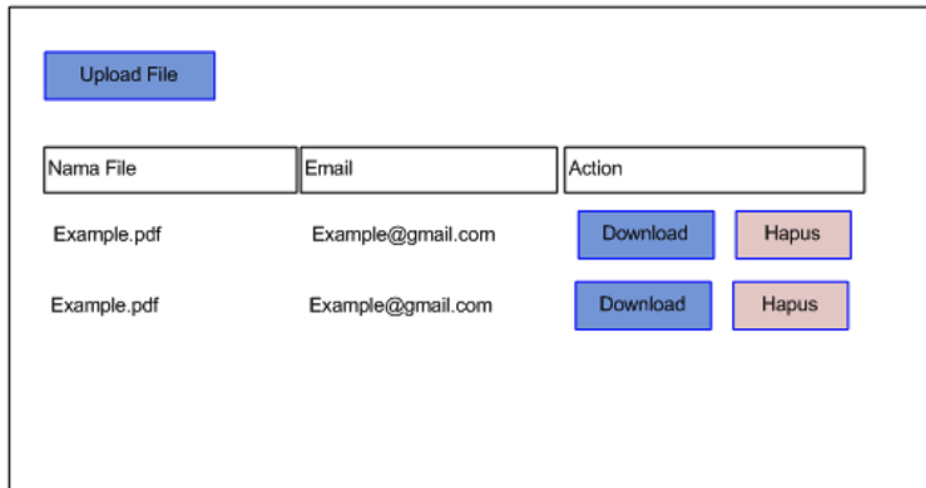


The image shows two versions of a registration form. The left version has a title 'Sign Up' and a subtitle 'Let's Register', followed by a single 'Registrasi' button. The right version has a title 'Register' and four input fields: 'Nama', 'Email', 'Password', and 'Confirm Password'. At the bottom of the right version is a blue button labeled 'Registrasi'.

Gambar 3.15 Perancangan UI Registrasi

c. Perancangan UI List File

Halaman list file merupakan halaman yang menampilkan semua file yang diupload ke sistem dan tersimpan pada cloud storage. Pada halaman ini terdapat tombol untuk membuka halaman upload file, daftar file yang sudah diupload, serta dua tombol download dan tombol hapus. Adapun rancangan dari UI List File ditunjukkan seperti gambar 3.16 berikut.



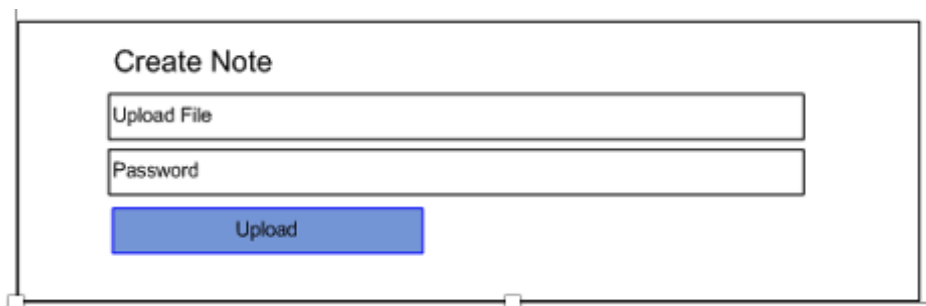
The UI design for the List File page includes an 'Upload File' button at the top left. Below it is a table with three columns: 'Nama File', 'Email', and 'Action'. The table contains two rows of data, each with a file name 'Example.pdf' and an email 'Example@gmail.com'. To the right of each row are two buttons: 'Download' (blue) and 'Hapus' (pink).

Nama File	Email	Action
Example.pdf	Example@gmail.com	<button>Download</button> <button>Hapus</button>
Example.pdf	Example@gmail.com	<button>Download</button> <button>Hapus</button>

Gambar 3.16 Perancangan UI List File

d. Perancangan UI Upload File

Halaman upload file berguna untuk mem-upload file ke sistem dan disimpan pada cloud storage. Pada halaman ini terdapat beberapa komponen yang memudahkan pengguna untuk melakukan proses upload file, yaitu komponen browse file yang akan diupload, komponen inputan password, serta satu tombol untuk menjalankan proses upload. Berikut ini adalah perancangan user interface upload file pada sistem yang dibuat seperti gambar 3.17 dibawah.



The UI design for the Upload File page features a 'Create Note' section. It contains two input fields: 'Upload File' and 'Password'. Below these fields is a blue 'Upload' button.

Create Note

Upload

Gambar 3.17 Perancangan UI Upload File

e. Pseudocode

1) Enkripsi AES

```
Encryption(byte in[16], byte out[16], word w[44])
begin
    byte state[16]
    state = in
    AddRoundKey(state, w[0, 3])           // Section. 3.4.4
    for round = 1 step 1 to 9
        SubBytes(state)                   // Section 3.4.1
        ShiftRows(state)                  // Section 3.4.2
        MixColumns(state)                 // Section 3.4.3
        AddRoundKey(state, w[round*4, (round+1)*3])
    end for
    SubBytes(state)
    ShiftRows(state)
    AddRoundKey(state, w[40, 43])
    out = state
end
```

2) Dekripsi AES

```
Aes Decrypt (byte in[4*Nb], byte out [4*Nb], word w [Nb*
(Nr+1)]).
Begin
    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
```

3) Kompresi LZW

```
string s;  
char ch;  
...  
  
s = empty string;  
while (there is still data to be read)  
{  
    ch = read a character;  
    if (dictionary contains s+ch)  
    {  
        s = s+ch;  
    }  
    else  
    {  
        encode s to output file;  
        add s+ch to dictionary;  
        s = ch;  
    }  
}  
encode s to output file;
```

4) Dekompresi LZW

```
string entry;  
char ch;  
int prevcode, currcode;  
...  
  
prevcode = read in a code;  
decode/output prevcode;  
while (there is still data to read)  
{  
    currcode = read in a code;  
    entry = translation of currcode from dictionary;  
    output entry;  
    ch = first char of entry;  
    add ((translation of prevcode)+ch) to dictionary;  
    prevcode = currcode;  
}
```