

BAB II

TINJAUAN PUSTAKA

2.1 Kriptografi

2.1.1 Pengertian Kriptografi

Kriptografi berasal dari dua kata dalam bahasa Yunani, yaitu "kryptos" yang berarti tersembunyi atau rahasia, dan "graphein" yang berarti menulis. Sehingga, secara sederhana, kriptografi dapat diartikan sebagai seni atau teknik menulis sesuatu dengan cara yang disembunyikan. (Stallings, 2017) mengemukakan bahwa kriptografi adalah metode yang digunakan untuk melindungi komunikasi dengan cara menyamarkan pesan yang dikirim agar tidak bisa dibaca oleh pihak yang tidak diinginkan. Sementara itu, menurut (Menezes, 1996) kriptografi adalah ilmu yang menggunakan pendekatan matematika untuk memastikan keamanan, keaslian, dan kerahasiaan suatu informasi. Dari beberapa definisi ini, kriptografi bisa disimpulkan sebagai ilmu yang mempelajari teknik-teknik khusus untuk mengamankan data atau pesan dengan cara menyembunyikan isinya. Tujuannya agar pesan tersebut hanya dapat dipahami oleh orang yang berhak menerimanya, menjaga kerahasiaan, dan memastikan pesan tidak diubah atau dimanipulasi selama pengiriman.

2.1.2 Sejarah Kriptografi

Kriptografi sudah ada sejak zaman dahulu ketika orang-orang mulai mencoba menyembunyikan pesan rahasia menggunakan teknik yang sederhana. (Singh, 1999) dalam bukunya *The Code Book* menjelaskan bahwa pada awalnya, orang menggunakan metode klasik seperti substitusi, di mana huruf-huruf dalam pesan diganti dengan huruf lain sesuai aturan tertentu, atau transposisi, di mana urutan huruf-huruf dalam pesan diubah. Pada abad ke-20, perkembangan teknologi membawa perubahan besar dalam kriptografi. Salah satu tonggak penting dalam sejarah kriptografi modern adalah pengenalan sistem kunci publik oleh (Diffie, 1976) yang memungkinkan

dua pihak berkomunikasi dengan aman tanpa harus berbagi kunci rahasia sebelumnya. Saat ini, algoritma seperti AES (*Advanced Encryption Standard*) yang disahkan oleh NIST pada tahun 2001 telah menjadi salah satu standar enkripsi yang paling banyak digunakan.

2.1.3 Tujuan Kriptografi

Adapun beberapa tujuan dari kriptografi ialah:

1. Kerahasiaan, yaitu untuk memastikan bahwa informasi atau pesan hanya bisa dibaca oleh orang yang berhak.
2. Integritas Data, yaitu untuk memastikan bahwa pesan yang dikirimkan tidak mengalami perubahan atau modifikasi selama proses pengiriman.
3. Otentikasi, yaitu untuk memverifikasi identitas pengirim dan penerima pesan.
4. *Non-Repudiation* (Anti-Penyangkalan), yaitu untuk mencegah pengirim atau penerima pesan menyangkal telah melakukan komunikasi. Misalnya seperti tanda tangan digital yang merupakan mekanisme untuk memastikan bahwa pihak-pihak tidak dapat menyangkal tindakan atau perjanjian mereka di kemudian hari.

2.1.4 Elemen Kriptografi

Adapun komponen pada kriptografi ialah:

1. *Plaintext*, yaitu merupakan pesan asli yang belum dienkripsi. Pesan ini bisa dalam bentuk teks, gambar, atau data lainnya yang dapat dibaca dan dipahami oleh pengguna.
2. *Ciphertext*, yaitu merupakan pesan yang telah dienkripsi dari *plaintext* menjadi bentuk yang tidak dapat dibaca tanpa kunci dekripsi. *Ciphertext* ini dihasilkan melalui algoritma enkripsi.
3. Algoritma Kriptografi, yaitu merupakan prosedur matematis yang digunakan untuk mengenkripsi dan mendekripsi data
4. Kunci (*Key*), yaitu merupakan kunci informasi rahasia yang digunakan untuk proses enkripsi dan dekripsi. Ada dua jenis kunci utama dalam kriptografi, yaitu kunci simetris (kunci yang

sama digunakan untuk mengenkripsi atau mendekripsi) dan kunci asimetris (kunci enkripsi dan dekripsinya berbeda).

2.2 Enkripsi dan Dekripsi

Enkripsi adalah suatu proses yang digunakan untuk mengubah data atau informasi asli (*plaintext*) menjadi format yang tidak bisa dibaca atau dimengerti (*ciphertext*). Hal ini dilakukan dengan memanfaatkan algoritma tertentu dan kunci rahasia. Tujuan utama dari enkripsi adalah untuk melindungi kerahasiaan informasi, sehingga hanya individu atau pihak yang memiliki kunci yang dapat mengakses dan membaca data tersebut. (Kusumo, 2019) menyatakan bahwa enkripsi sangat penting dalam menjaga keamanan data saat dikirim melalui jaringan, sehingga informasi tersebut tidak bisa diakses oleh orang yang tidak berwenang. Kemudian dekripsi adalah proses yang berfungsi untuk mengembalikan *ciphertext* menjadi *plaintext* dengan menggunakan kunci yang sesuai. Proses ini memungkinkan penerima untuk membaca dan memahami informasi yang telah dienkripsi. Menurut (Mulia, 2020) dekripsi adalah langkah yang vital (penting) karena menjamin bahwa hanya pihak yang berwenang yang dapat mengakses dan membaca data yang dikirim.

2.3 Advanced Encryption Standard (AES)

Algoritma *Advanced Encryption Standard* adalah algoritma untuk mengenkripsi dan mendekripsi file dokumen. Kekuatan keamanan AES yang telah terbukti menjadikannya salah satu algoritma kriptografi simetris yang paling banyak digunakan di dunia. Dalam hal mengamankan file dokumen, penerapan AES 128 sangat penting karena memberikan lapisan keamanan tambahan yang diperlukan untuk melindungi informasi sensitif yang disimpan dalam file ini. Penelitian ini merinci bagaimana AES 128 digunakan dalam praktiknya untuk melindungi kerahasiaan data dalam file dokumen (Asri, 2018).

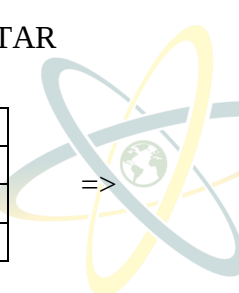
Terdapat tahapan dalam algoritma AES yaitu:

1. *SubBytes* merupakan penggantian *byte* menggunakan *S-box*
2. *ShiftRows* merupakan penggeseran baris matriks
3. *MixColumns* merupakan perkalian setiap kolom dengan matriks
4. *AddRoundKey* merupakan XOR antara *state* dengan kunci ronde

Contoh

Plaintext: AYOSEMANGATAFTAR

Kunci: TETAPSEMANGATTAR



A	E	G	F
Y	M	A	T
O	A	T	A
S	N	A	R

 $\Rightarrow$

41	45	47	46
59	4D	41	54
4F	41	54	41
53	4E	41	52

Gambar 2.1 Contoh *Plaintext*



T	P	A	T
E	S	N	T
T	E	G	A
A	M	A	R

 $\Rightarrow$

54	50	41	54
45	53	4E	54
54	45	47	41
41	4D	41	52

Gambar 2.2 Contoh Kunci

Hasil XOR antara *Plaintext* dan Kunci:

XOR Result:

15	15	06	12
1C	1E	0F	00
1B	04	13	00
12	03	00	00

Gambar 2.3 Hasil XOR *Plaintext* dan Kunci

Tahap Awal: *AddRoundKey*

State setelah *AddRoundKey*:

15	15	06	12
1C	1E	0F	00
1B	04	13	00
12	03	00	00

Gambar 2.4 Tahap Awal *AddRoundKey*

Round 1**Tahap SubBytes:****State setelah SubBytes:**

59	59	6F	C9
9C	72	76	63
AF	F2	7D	63
C9	7B	63	63

Gambar 2.5 Tahap SubBytes 1

Tahap ShiftRows:**State setelah ShiftRows:**

59	59	6F	C9
72	76	63	9C
7D	63	AF	F2
63	C9	7B	63

Gambar 2.6 Tahap ShiftRows 1

Tahap MixColumns:**State setelah MixColumns:**

3A	82	AF	A7
59	D9	38	84
74	A9	C4	0F
22	77	8B	E8

Gambar 2.7 Tahap MixColumns 1

Tahap AddRoundKey:**State setelah AddRoundKey:**

4F	A7	CB	97
9F	4C	E3	0B
20	B8	92	18
43	5B	E6	D7

Gambar 2.8 Tahap AddRoundKey 1

Round 2**Tahap SubBytes:****State setelah SubBytes:**

84	5C	1F	88
DB	29	11	2B
B7	6C	4F	AD
1A	39	8E	0E

Gambar 2.9 Tahap SubBytes 2

Tahap ShiftRows:**State setelah ShiftRows:**

84	5C	1F	88
29	11	2B	DB
4F	AD	B7	6C
0E	1A	39	8E

Gambar 2.10 Tahap ShiftRows 2

Tahap MixColumns:**State setelah MixColumns:**

29	3C	CD	9F
09	88	B2	1F
21	22	0A	02
ED	6C	CF	33

Gambar 2.11 Tahap MixColumns 2

Tahap AddRoundKey:**State setelah AddRoundKey:**

2D	1D	88	EA
3F	2B	CA	E8
00	12	6C	73
88	25	EB	28

Gambar 2.12 Tahap AddRoundKey 2

Round 3**Tahap SubBytes:****State setelah SubBytes:**

D8	A4	C4	87
75	F1`	74	9B
63	C9	50	8F
C4	3F	E9	34

Gambar 2.13 Tahap SubBytes 3**Tahap ShiftRows:****State setelah ShiftRows:**

D8	A4	C4	87
F1	74`	9B	75
50	8F	63	C9
34	C4	3F	E9

Gambar 2.14 Tahap ShiftRowss 3**Tahap MixColumns:****State setelah MixColumns:**

C7	84	79	AA
E5	02`	73	C4
D5	82	D8	5B
BA	9F	D1	E7

Gambar 2.15 Tahap ShiftRows 3**Tahap AddRoundKey:****State setelah AddRoundKey:**

AF	CD	75	D3
70	34`	3D	7D
5B	3C	00	F2
42	2E	44	69

Gambar 2.16 Tahap ShiftRows 3**Round 4****Tahap SubBytes:****State setelah SubBytes:**

79	BD	9D	66
51	18	27	FF
39	EB	63	89
2C	31	1B	F9

Gambar 2.17 Tahap SubBytes 4**Tahap ShiftRows:****State setelah ShiftRows:**

79	BD	9D	66
18	27	FF	51
63	89	39	EB
F9	2C	31	1B

Gambar 2.18 Tahap ShiftRows 4**Tahap MixColumns:****State setelah MixColumns:**

40	AD	33	CF
15	5F	02	F9
B7	E7	43	D7
19	2A	18	26

Gambar 2.19 Tahap MixColumns 4**Tahap AddRoundKey:****State setelah AddRoundKey:**

76	D2	40	C5
53	2F	3C	7E
20	CE	B2	8F
57	D5	72	C2

Gambar 2.20 Tahap AddRoundKey 4

Round 5**Tahap SubBytes:****State setelah SubBytes:**

38	B2	09	A6
ED	15	EB	F3
B7	8B	37	73
5B	03	40	25

Gambar 2.21 Tahap SubBytes 5**Tahap ShiftRows:****State setelah ShiftRows:**

38	B2	09	A6
15	EB	F3	ED
37	73	B7	8D
25	5B	03	40

Gambar 2.22 Tahap ShiftRows 5**Tahap MixColumns:****State setelah MixColumns:**

5D	7F	A8	B0
6E	B6	35	A1
2C	55	8A	86
20	EA	59	17

Gambar 2.23 Tahap MixColumn 5**Tahap AddRoundKey:****State setelah AddRoundKey:**

6C	31	95	87
42	EA	57	44
D2	82	AC	F8
09	3C	E5	4F

Gambar 2.24 Tahap AddRoundKey 5**Round 6****Tahap SubBytes:****State setelah SubBytes:**

50	C7	2A	17
2C	87	5B	1B
B5	13	91	41
01	EB	D9	84

Gambar 2.25 Tahap SubBytes 6**Tahap ShiftRows:****State setelah ShiftRows:**

50	C7	2A	17
87	5B	1B	2C
91	41	B5	13
84	01	EB	D9

Gambar 2.26 Tahap ShiftRows 6**Tahap MixColumns:****State setelah MixColumns:**

27	38	27	90
69	B3	33	A3
79	1D	66	6D
F5	4A	1D	AF

Gambar 2.27 Tahap MixColumn 6**Tahap AddRoundKey:****State setelah AddRoundKey:**

EF	BE	9C	1C
B6	30	D2	A7
ED	5E	03	76
46	2F	C4	2E

Gambar 2.28 Tahap AddRoundKey 6

Round 7**Tahap SubBytes:****State setelah SubBytes:**

DF	AE	DE	9C
4E	04	B5	5C
55	58	7B	38
5A	15	1C	31

Gambar 2.29 Tahap SubBytes 7**Tahap ShiftRows:****State setelah ShiftRows:**

DF	AE	DE	9C
04	B5	5C	4E
7B	38	55	58
31	5A	15	1C

Gambar 2.30 Tahap ShiftRows 7**Tahap MixColumns:****State setelah MixColumns:**

E3	E1	03	B5
6B	CD	8C	F4
7E	85	17	46
67	D0	5A	91

Gambar 2.31 Tahap MixColumns 7**Tahap AddRoundKey:****State setelah AddRoundKey:**

99	1D	44	7E
1B	3E	9E	E2
E6	5E	A9	E3
B0	62	31	7B

Gambar 2.32 Tahap AddRoundKey 7**Round 8****Tahap SubBytes:****State setelah SubBytes:**

EE	A4	1B	F3
AF	B2	0B	98
8E	58	D3	11
E7	AA	C7	21

Gambar 2.33 Tahap SubBytes8**Tahap ShiftRows:****State setelah ShiftRows:**

EE	A4	1B	F3
B2	0B	98	AF
D3	11	8E	58
21	E7	AA	C7

Gambar 2.34 Tahap ShiftRows 8**Tahap MixColumns:****State setelah MixColumns:**

F8	B9	A1	88
DE	66	13	99
82	BF	61	BE
0A	38	74	6C

Gambar 2.35 Tahap MixColumns 8**Tahap AddRoundKey:****State setelah AddRoundKey:**

45	F9	A7	45
A8	E3	84	18
9D	7B	1B	61
C2	42	65	97

Gambar 2.36 Tahap AddRoundKey 8

*Round 9*Tahap *SubBytes*:State setelah *SubBytes*:

6E	99	5C	6E
C2	11	5F	AD
5E	21	AF	EF
25	2C	4D	88

Gambar 2.37 Tahap *SubBytes* 9Tahap *ShiftRows*:State setelah *ShiftRows*:

6E	99	5C	6E
11	5F	AD	C2
AF	EF	5E	21
88	25	2C	4D

Gambar 2.38 Tahap *ShiftRows* 9Tahap *MixColumns*:State setelah *MixColumns*:

C8	02	26	ED
2E	28	D3	DF
B9	6C	39	39
07	4A	4F	CB

Gambar 2.39 Tahap *MixColumns* 9Tahap *AddRoundKey*:State setelah *AddRoundKey*:

62	E9	CB	CD
C6	45	29	A4
A9	B8	97	48
72	45	51	2E

Gambar 2.40 Tahap *AddRoundKey* 9*Final Round*Tahap *SubBytes*:State setelah *SubBytes*:

AA	1E	1F	BD
B4	6E	A5	49
D3	6C	88	52
40	6E	D1	31

Gambar 2.41 Hasil *SubBytes*Tahap *ShiftRows*:State setelah *ShiftRows*:

AA	1E	1F	BD
6E	A5	49	B4
88	52	D3	6C
31	40	6E	D1

Gambar 2.42 Hasil *ShiftRows*Tahap *AddRoundKey*:State setelah *AddRoundKey*
(Ciphertext):

17	48	A4	26
25	83	95	13
41	4F	60	AE
F3	8D	BD	E7

Gambar 2.43 Hasil *AddRoundKey*

Ciphertext (Hex): 172541F348834F8DA49560BD2613AEE7

Ciphertext (Chars): .%A.H.O...`.&...

Tahap Awal Decrypt: *AddRoundKey*

State setelah *AddRoundKey*:

AA	1E	1F	BD
6E	A5	49	B4
88	52	D3	6C
31	40	6E	D1

Gambar 2.44 Tahap Awal Decrypt

Round 9

Tahap *InvShiftRows*:

State setelah *InvShiftRows*:

AA	1E	1F	BD
B4	6E	A5	49
D3	6C	88	52
40	6E	D1	31

Gambar 2.45 Tahap *InvShiftRows* 9

Tahap *InvSubBytes*:

State setelah *InvSubBytes*:

62	E9	CB	CD
C6	45	29	A4
A9	B8	97	48
72	45	51	2E

Gambar 2.46 Tahap *InvSubByte* 9

Tahap *AddRoundKey*:

State setelah *AddRoundKey*:

C8	02	26	ED
2E	28	D3	DF
B9	6C	39	39
07	4A	4F	CB

Gambar 2.47 Tahap *AddRoundKey* 9

Tahap *InvMixColumns*:

State setelah *InvMixColumns*:

6E	99	5C	6E
11	5F	AD	C2
AF	EF	5E	21
88	25	2C	4D

Gambar 2.48 Tahap *InvMixColumns* 9

Round 8

Tahap *InvShiftRows*:

State setelah *InvShiftRows*:

6E	99	5C	6E
C2	11	5F	AD
5E	21	AF	EF
25	2C	4D	88

Gambar 2.49 Tahap *InvShiftRows* 8

Tahap *InvSubBytes*:

State setelah *InvSubBytes*:

45	F9	A7	45
A8	E3	84	18
9D	7B	1B	61
C2	42	65	97

Gambar 2.50 Tahap *InvSubBytes* 8

Tahap *AddRoundKey*:

State setelah *AddRoundKey*:

F8	B8	A1	88
DE	66	13	99
82	BF	61	BE
0A	38	74	6C

Gambar 2.51 Tahap *AddRoundKey* 8

Tahap *InvMixColumns*:

State setelah *InvMixColumns*:

EE	A4	1B	F3
B2	0B	98	AF
D3	11	8E	58
21	E7	AA	C7

Gambar 2.52 Tahap *InvMixColumns* 8

*Round 7*Tahap *InvShiftRows*:State setelah *InvShiftRows*:

EE	A4	1B	F3
AF	B2	0B	98
8E	58	D3	11
E7	AA	C7	21

Gambar 2.53 Tahap *InvShiftRows* 7Tahap *InvSubBytes*:State setelah *InvSubBytes*:

99	1D	44	7E
1B	3E	9E	E2
E6	5E	A9	E3
B0	62	31	7B

Gambar 2.54 Tahap *InvSubBytes* 7Tahap *AddRoundKey*:State setelah *AddRoundKey*:

E3	E1	03	B5
6B	CD	8C	F4
7E	85	17	46
67	D0	5A	91

Gambar 2.55 Tahap *AddRoundKey* 7Tahap *InvMixColumns*:State setelah *InvMixColumns*:

DF	AE	DE	9C
04	B5	5C	4E
7B	38	55	58
31	5A	15	1C

Gambar 2.56 Tahap *InvMixColumns* 7*Round 6*Tahap *InvShiftRows*:State setelah *InvShiftRows*:

DF	AE	DE	9C
4E	04	B5	5C
55	58	7B	38
5A	15	1C	31

Gambar 2.57 Tahap *InvShiftRows* 6Tahap *InvSubBytes*:State setelah *InvSubBytes*:

EF	BE	9C	1C
B6	30	D2	A7
ED	5E	03	76
46	2F	C4	2E

Gambar 2.58 Tahap *InvSubBytes* 6Tahap *AddRoundKey*:State setelah *AddRoundKey*:

27	38	27	90
69	B3	33	A3
79	1D	66	6D
F5	4A	1D	AF

Gambar 2.59 Tahap *AddRoundKeys* 6Tahap *InvMixColumns*:State setelah *InvMixColumns*:

50	C7	2A	17
87	5B	1B	2C
91	41	B5	13
84	01	EB	D9

Gambar 2.60 Tahap *InvMixColumns* 6

*Round 5*Tahap *InvShiftRows*:State setelah *InvShiftRows*:

50	C7	2A	17
2C	87	5B	1B
B5	13	91	41
01	EB	D9	84

Gambar 2.61 Tahap *InvShiftRows* 5Tahap *InvSubBytes*:State setelah *InvSubBytes*:

6C	31	95	87
42	EA	57	44
D2	82	AC	F8
09	3C	E5	4F

Gambar 2.62 Tahap *InvSubBytes* 5Tahap *AddRoundKey*:State setelah *AddRoundKey*:

5D	7F	A8	B0
6E	B6	35	A1
2C	55	8A	86
20	EA	59	17

Gambar 2.63 Tahap *AddRoundKey* 5Tahap *InvMixColumns*:State setelah *InvMixColumns*:

38	B5	09	A6
15	EB	F3	ED
37	73	B7	8B
25	5B	03	40

Gambar 2.64 Tahap *InvMixColumns* 5*Round 4*Tahap *InvShiftRows*:State setelah *InvShiftRows*:

38	B5	09	A6
ED	15	EB	F3
B7	8B	37	73
5B	03	40	25

Gambar 2.65 Tahap *InvShiftRows* 4Tahap *InvSubBytes*:State setelah *InvSubBytes*:

76	D2	40	C5
53	2F	3C	7E
20	CE	B2	8F
57	D5	72	C2

Gambar 2.66 Tahap *InvSubBytes* 4Tahap *AddRoundKey*:State setelah *AddRoundKey*:

40	AD	33	CF
15	5F	02	F9
B7	E7	43	D7
19	2A	18	26

Gambar 2.67 Tahap *AddRoundKey* 4Tahap *InvMixColumns*:State setelah *InvMixColumns*:

79	BD	9D	66
18	27	FF	51
63	89	39	EB
F9	2C	31	1B

Gambar 2.68 Tahap *InvMixColumns* 4

*Round 3*Tahap *InvShiftRows*:State setelah *InvShiftRows*:

79	BD	9D	66
51	18	27	FF
39	EB	63	89
2C	31	1B	F9

Gambar 2.69 Tahap *InvShiftRows* 3Tahap *InvSubBytes*:State setelah *InvSubBytes*:

AF	CD	75	D3
70	34	3D	7D
5B	3C	00	F2
42	2E	44	69

Gambar 2.70 Tahap *InvSubBytes* 3Tahap *AddRoundKey*:State setelah *AddRoundKey*:

C7	84	79	AA
E5	02	73	C4
D5	82	D8	5B
BA	9F	D1	E7

Gambar 2.71 Tahap *AddRoundKey* 3Tahap *InvMixColumns*:State setelah *InvMixColumns*:

D8	A4	C4	87
F1	74	9B	75
50	8F	63	C9
34	C4	3F	E9

Gambar 2.72 Tahap *InvMixColumns* 3*Round 2*Tahap *InvShiftRows*:State setelah *InvShiftRows*:

D8	A4	C4	87
75	F1	74	9B
63	C9	50	8F
C4	3F	E9	34

Gambar 2.73 Tahap *InvShiftRows* 2Tahap *InvSubBytes*:State setelah *InvSubBytes*:

2D	1D	88	EA
3F	2B	CA	E8
00	12	6C	73
88	25	EB	28

Gambar 2.74 Tahap *InvSubBytes* 2Tahap *AddRoundKey*:State setelah *AddRoundKey*:

29	3C	CD	9F
09	88	B2	1F
21	22	0A	02
ED	6C	CF	33

Gambar 2.75 Tahap *AddRoundKey* 2Tahap *InvMixColumns*:State setelah *InvMixColumns*:

84	5C	1F	88
29	11	2B	DB
4F	AD	B7	6C
0E	1A	39	8E

Gambar 2.76 Tahap *InvMixColumns* 2

*Round 1*Tahap *InvShiftRows*:State setelah *InvShiftRows*:

84	5C	1F	88
DB	29	11	2B
B7	6C	4F	AD
1A	39	8E	0E

Gambar 2.77 Tahap *InvShiftRows* 1Tahap *InvSubBytes*:State setelah *InvSubBytes*:

4F	A7	CB	97
9F	4C	E3	0B
20	B8	92	18
43	5B	92	D7

Gambar 2.78 Tahap *InvSubBytes* 1Tahap *AddRoundKey*:State setelah *AddRoundKey*:

3A	82	AF	A7
59	D9	38	84
74	A9	C4	0F
22	77	8B	E8

Gambar 2.79 Tahap *AddRoundKey* 1Tahap *InvMixColumns*:State setelah *InvMixColumns*:

59	59	6F	C9
72	76	63	9C
7D	63	AF	F2
63	C9	7B	63

Gambar 2.80 Tahap *InvMixColumns* 1*Final Round Decryp*Tahap *InvShiftRows*:State setelah *InvShiftRows*:

59	59	6F	C9
9C	72	76	63
AF	F2	7D	63
C9	7B	63	63

Gambar 2.81 Hasil *InvShiftRows*Tahap *InvSubBytes*:State setelah *InvSubBytes*:

15	15	06	12
1C	1E	0F	00
1B	04	13	00
12	03	00	00

Gambar 2.82 Hasil *InvSubBytes*

Tahap *AddRoundKey*:

State setelah *AddRoundKey* (Plaintext):

41	45	47	46	=>	A	E	G	F
59	4D	41	54		Y	M	A	T
4F	41	54	41		O	A	T	A
53	4E	41	52		S	N	A	R

Gambar 2.83 Hasil *AddRoundKey*

=== Hasil ===

Plaintext: AYOSEMANGATAFTAR

Ciphertext (Hex): 172541F348834F8DA49560BD2613AEE7

Ciphertext (Chars): .%A.H.O...`.&...

Decrypted Plaintext: AYOSEMANGATAFTAR

Enkripsi dan dekripsi berhasil! Plaintext sama.

2.4 Steganografi

Steganografi adalah teknik yang digunakan untuk menyembunyikan informasi atau pesan di dalam media lain, sehingga keberadaan pesan tersebut tidak terdeteksi oleh orang yang tidak berwenang. Berbeda dengan kriptografi, yang hanya fokus pada pengamanan isi pesan, steganografi berupaya untuk menyembunyikan fakta bahwa sebuah pesan sedang dikirim (Tarigan, 2022). Terdapat beberapa jenis steganografi berdasarkan media yang digunakan, antara lain:

1. **Steganografi Berbasis Citra:** Teknik ini menyembunyikan pesan di dalam citra digital. Dengan memanfaatkan variasi warna pada piksel, data tersembunyi dapat dimasukkan tanpa mengubah tampilan visual citra secara signifikan. Metode ini sering kali menggunakan algoritma seperti *Least Significant Bit* (LSB) untuk menyisipkan data (Kusnadi, 2021).
2. **Steganografi Berbasis Audio:** Dalam metode ini, informasi disisipkan ke dalam file audio. Teknik yang umum digunakan termasuk modifikasi LSB dari sampel audio, di mana data disembunyikan di bagian yang tidak terdengar oleh telinga manusia (Husni, 2022).
3. **Steganografi Berbasis Teks:** Teknik ini menyisipkan pesan rahasia dalam teks biasa. Metode yang digunakan bisa berupa pengaturan spasi, pemilihan kata, atau penggunaan karakter khusus yang tidak terlihat.

Steganografi berbasis teks memungkinkan pesan untuk disembunyikan dengan cara yang sulit terdeteksi (Ziliwu, 2022).

Keunggulan dari steganografi terletak pada kemampuannya untuk menyembunyikan keberadaan pesan, menjadikannya metode yang efektif untuk komunikasi aman. Namun, tantangan yang dihadapi adalah memastikan bahwa teknik yang digunakan tidak memengaruhi kualitas media asli dan bahwa pesan yang disembunyikan dapat diambil kembali dengan akurat oleh penerima yang berwenang.

2.5 Bit-Plane Complexity Segmentation (BPCS)

Bit-Plane Complexity Segmentation (BPCS) adalah teknik steganografi yang diperkenalkan oleh Eiji Kawaguchi dan R. O. Eason pada tahun 1997. Menurut (Solichin, 2015), BPCS bekerja dengan membagi citra digital menjadi segmen-segmen berukuran 8x8 piksel. Setiap segmen kemudian dipecah menjadi delapan *bit-plane*, yang merepresentasikan piksel dari setiap bit dalam citra tersebut. Proses ini disebut *bit slicing*, di mana setiap *bit-plane* memiliki karakteristik yang berbeda, memungkinkan penyisipan data dilakukan dengan lebih efektif. Terdapat tahapan dalam BPCS yaitu:

1. Pemisahan Citra: Citra digital dibagi menjadi segmen-segmen 8x8 piksel. Setiap segmen ini akan memiliki 8 *bit-plane*, yang masing-masingnya merepresentasikan nilai piksel dalam format biner (Kawaguchi, 1998).
2. Penilaian Kompleksitas: Setiap *bit-plane* dinilai kompleksitasnya. Segmen dengan kompleksitas tinggi, yang dikenal sebagai *noise-like regions*, dipilih sebagai tempat penyisipan data. Pada tahap ini, penyisipan tidak hanya dilakukan pada *least significant bit* (LSB), tetapi juga pada seluruh *bit-plane*, termasuk *most significant bit* (MSB) (Hidayat, 2020).
3. Proses Penyisipan: Data rahasia disisipkan ke dalam *bit-plane* yang telah ditentukan sebelumnya berdasarkan kompleksitasnya. (Solichin, 2015).
4. Penentuan Nilai *Threshold*: Dalam implementasi awal BPCS, penentuan nilai *threshold* dilakukan secara manual oleh pengguna, dengan nilai inisial biasanya sebesar 0,3. Oleh karena itu, pendekatan digunakan untuk menentukan nilai *threshold* berdasarkan dua *input*: perbandingan antara

pesan dan gambar serta rata-rata kompleksitas dari *bit-plane* (Hidayat, 2020).

Contoh

Diketahui sebuah citra ukuran 8x8 piksel dengan nilai sebagai berikut

Nilai Piksel 8x8:

145	150	160	175	120	100	110	95
140	135	130	160	155	125	115	90
150	145	135	170	180	140	125	100
155	160	170	190	200	145	135	105
130	140	150	160	140	135	125	110
110	120	130	140	120	115	100	95
140	150	160	175	120	100	110	95
120	125	130	150	140	130	115	90

Gambar 2.84 Contoh Sebuah Citra 8x8 Piksel

Konversi ke Biner

Tabel 2.1 Konversi Nilai Biner

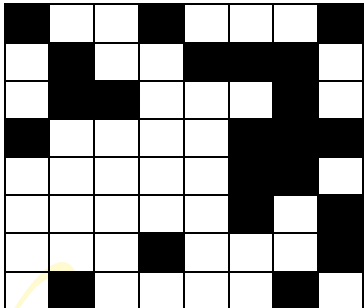
10010001	10010110	10100000	10101111	01111000	01100100	01101110	01011111
10001100	10000111	10000010	10100000	10011011	01111101	01110011	01011010
10010110	10010001	10000111	10101010	10110100	10001100	01111101	01100100
10011011	10100000	10101010	10111110	11001000	10010001	10000111	01101001
10000010	10001100	10010110	10100000	10001100	10000111	01111101	01101110
01101110	01111000	10000010	10001100	01111000	01110011	01100100	01011111
10001100	10010110	10100000	10101111	01111000	01100100	01101110	01011111
01111000	01111101	10000010	10010110	10001100	10000010	01110011	01011010

Kemudian pisahkan setiap bit dari biner tersebut ke dalam tabel untuk masing-masing bit-plane:

Bit-Plane 0:

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

=>

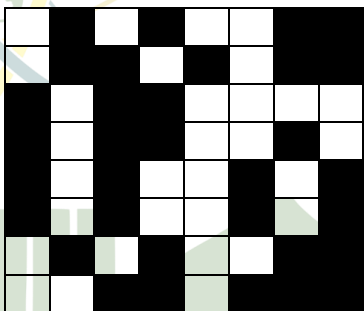


Gambar 2.85 Konversi Nilai Bit-Plane 0

Bit-Plane 1:

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

=>

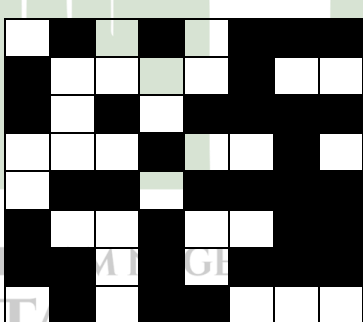


Gambar 2.86 Konversi Nilai Bit-Plane 1

Bit-Plane 2:

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

=>

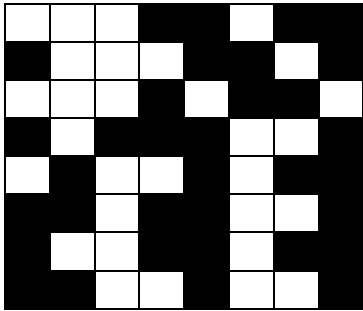


Gambar 2.87 Konversi Nilai Bit-Plane 2

Bit-Plane 3:

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

=>

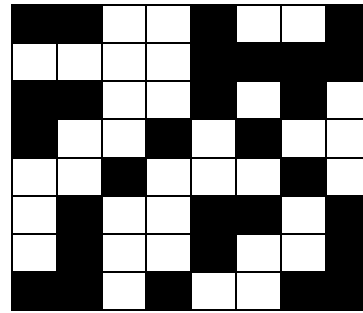


Gambar 2.88 Konversi Nilai Bit-Plane 3

Bit-Plane 4:

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

=>

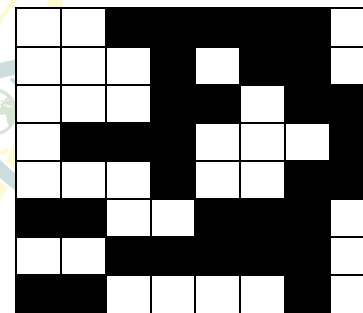


Gambar 2.89 Konversi Nilai Bit-Plane 4

Bit-Plane 5:

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

=>

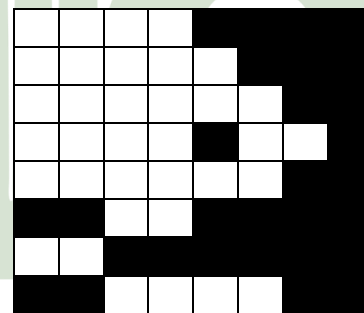


Gambar 2.90 Konversi Nilai Bit-Plane 5

Bit-Plane 6:

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

=>

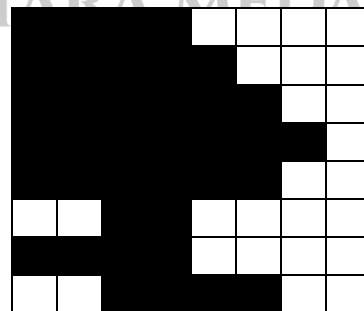


Gambar 2.91 Konversi Nilai Bit-Plane 6

Bit-Plane 7:

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

=>



Gambar 2.92 Konversi Nilai Bit-Plane 7

Kemudian menghitung kompleksitas bit-plane:

Rumus:

$$\text{Kompleksitas} = \frac{\text{Jumlah Transisi}}{\text{Jumlah Elemen}}$$
 Ketentuan:
 Jumlah Elemen: 112
 Threshold: 0,3

Bit-plane 0: 51 transisi (Horizontal: 26, Vertikal: 25)

$$\begin{aligned} \text{Kompleksitas} &= \frac{\text{Jumlah Transisi}}{\text{Jumlah Elemen}} \\ &= \frac{51}{112} \end{aligned}$$

Kompleksitas: 0.4554, Kategori: Kompleks

Bit-plane 1: 61 transisi (Horizontal: 38, Vertikal: 23)

$$\begin{aligned} \text{Kompleksitas} &= \frac{\text{Jumlah Transisi}}{\text{Jumlah Elemen}} \\ &= \frac{61}{112} \end{aligned}$$

Kompleksitas: 0.5446, Kategori: Kompleks

Bit-plane 2: 65 transisi (Horizontal: 31, Vertikal: 34)

$$\begin{aligned} \text{Kompleksitas} &= \frac{\text{Jumlah Transisi}}{\text{Jumlah Elemen}} \\ &= \frac{65}{112} \end{aligned}$$

Kompleksitas: 0.5804, Kategori: Kompleks

Bit-plane 3: 60 transisi (Horizontal: 32, Vertikal: 28)

$$\begin{aligned} \text{Kompleksitas} &= \frac{\text{Jumlah Transisi}}{\text{Jumlah Elemen}} \\ &= \frac{60}{112} \end{aligned}$$

Kompleksitas: 0.5357, Kategori: Kompleks

Bit-plane 4: 62 transisi (Horizontal: 33, Vertikal: 29)

$$\begin{aligned} \text{Kompleksitas} &= \frac{\text{Jumlah Transisi}}{\text{Jumlah Elemen}} \\ &= \frac{62}{112} \end{aligned}$$

Kompleksitas: 0.5536, Kategori: Kompleks

Bit-plane 5: 51 transisi (Horizontal: 23, Vertikal: 28)

$$\begin{aligned} \text{Kompleksitas} &= \frac{\text{Jumlah Transisi}}{\text{Jumlah Elemen}} \end{aligned}$$

$$= \frac{51}{112}$$

Kompleksitas: 0.4554, Kategori: Kompleks

Bit-plane 6: 28 transisi (Horizontal: 12, Vertikal: 16)

$$\text{Kompleksitas} = \frac{\text{Jumlah Transisi}}{\text{Jumlah Elemen}}$$

$$= \frac{28}{112}$$

Kompleksitas: 0.2500, Kategori: Tidak Kompleks

Bit-plane 7: 24 transisi (Horizontal: 10, Vertikal: 14)

$$\text{Kompleksitas} = \frac{\text{Jumlah Transisi}}{\text{Jumlah Elemen}}$$

$$= \frac{24}{112}$$

Kompleksitas: 0.2143, Kategori: Tidak Kompleks

Kemudian menyisipkan pesan, setelah menemukan bagian gambar yang cukup kompleks (seperti *noise*), kita bisa menyisipkan pesan rahasia di sana, pesan ini akan dimasukkan dalam bentuk kode biner (0 dan 1). Penyisipan dilakukan dengan mengganti bit-bit pada bagian yang kompleks tadi dengan bit-bit dari pesan rahasia.

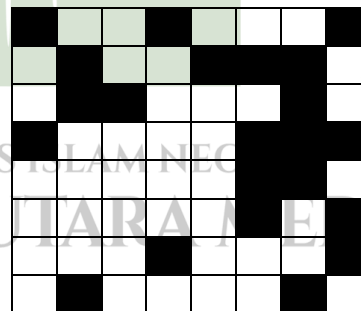
Pesan yang akan disisipkan: AYOAFATAR

Memilih bit-plane yang akan disisipkan: bit-plane 0

Bit-plane 0 sebelum penyisipan

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

=>



Gambar 2.93 Bit-plane 0 Sebelum Penyisipan

Proses penyisipan

1. Sisipkan 0 di Bit-plane 0, Baris 1, Kolom 1, Karakter A
2. Sisipkan 1 di Bit-plane 0, Baris 1, Kolom 2, Karakter A
3. Sisipkan 0 di Bit-plane 0, Baris 1, Kolom 3, Karakter A
4. Sisipkan 0 di Bit-plane 0, Baris 1, Kolom 4, Karakter A
5. Sisipkan 0 di Bit-plane 0, Baris 1, Kolom 5, Karakter A
6. Sisipkan 0 di Bit-plane 0, Baris 1, Kolom 6, Karakter A

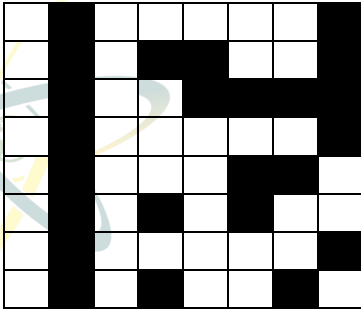
7. Sisipkan 0 di Bit-plane 0, Baris 1, Kolom 7, Karakter A
8. Sisipkan 1 di Bit-plane 0, Baris 1, Kolom 8, Karakter A
9. Sisipkan 0 di Bit-plane 0, Baris 2, Kolom 1, Karakter Y
10. Sisipkan 1 di Bit-plane 0, Baris 2, Kolom 2, Karakter Y
11. Sisipkan 0 di Bit-plane 0, Baris 2, Kolom 3, Karakter Y
12. Sisipkan 1 di Bit-plane 0, Baris 2, Kolom 4, Karakter Y
13. Sisipkan 1 di Bit-plane 0, Baris 2, Kolom 5, Karakter Y
14. Sisipkan 0 di Bit-plane 0, Baris 2, Kolom 6, Karakter Y
15. Sisipkan 0 di Bit-plane 0, Baris 2, Kolom 7, Karakter Y
16. Sisipkan 1 di Bit-plane 0, Baris 2, Kolom 8, Karakter Y
17. Sisipkan 0 di Bit-plane 0, Baris 3, Kolom 1, Karakter O
18. Sisipkan 1 di Bit-plane 0, Baris 3, Kolom 2, Karakter O
19. Sisipkan 0 di Bit-plane 0, Baris 3, Kolom 3, Karakter O
20. Sisipkan 0 di Bit-plane 0, Baris 3, Kolom 4, Karakter O
21. Sisipkan 1 di Bit-plane 0, Baris 3, Kolom 5, Karakter O
22. Sisipkan 1 di Bit-plane 0, Baris 3, Kolom 6, Karakter O
23. Sisipkan 1 di Bit-plane 0, Baris 3, Kolom 7, Karakter O
24. Sisipkan 1 di Bit-plane 0, Baris 3, Kolom 8, Karakter O
25. Sisipkan 0 di Bit-plane 0, Baris 4, Kolom 1, Karakter A
26. Sisipkan 1 di Bit-plane 0, Baris 4, Kolom 2, Karakter A
27. Sisipkan 0 di Bit-plane 0, Baris 4, Kolom 3, Karakter A
28. Sisipkan 0 di Bit-plane 0, Baris 4, Kolom 4, Karakter A
29. Sisipkan 0 di Bit-plane 0, Baris 4, Kolom 5, Karakter A
30. Sisipkan 0 di Bit-plane 0, Baris 4, Kolom 6, Karakter A
31. Sisipkan 0 di Bit-plane 0, Baris 4, Kolom 7, Karakter A
32. Sisipkan 1 di Bit-plane 0, Baris 4, Kolom 8, Karakter A
33. Sisipkan 0 di Bit-plane 0, Baris 5, Kolom 1, Karakter F
34. Sisipkan 1 di Bit-plane 0, Baris 5, Kolom 2, Karakter F
35. Sisipkan 0 di Bit-plane 0, Baris 5, Kolom 3, Karakter F
36. Sisipkan 0 di Bit-plane 0, Baris 5, Kolom 4, Karakter F
37. Sisipkan 0 di Bit-plane 0, Baris 5, Kolom 5, Karakter F
38. Sisipkan 1 di Bit-plane 0, Baris 5, Kolom 6, Karakter F
39. Sisipkan 1 di Bit-plane 0, Baris 5, Kolom 7, Karakter F
40. Sisipkan 0 di Bit-plane 0, Baris 5, Kolom 8, Karakter F
41. Sisipkan 0 di Bit-plane 0, Baris 6, Kolom 1, Karakter T
42. Sisipkan 1 di Bit-plane 0, Baris 6, Kolom 2, Karakter T
43. Sisipkan 0 di Bit-plane 0, Baris 6, Kolom 3, Karakter T
44. Sisipkan 1 di Bit-plane 0, Baris 6, Kolom 4, Karakter T
45. Sisipkan 0 di Bit-plane 0, Baris 6, Kolom 5, Karakter T
46. Sisipkan 1 di Bit-plane 0, Baris 6, Kolom 6, Karakter T
47. Sisipkan 0 di Bit-plane 0, Baris 6, Kolom 7, Karakter T
48. Sisipkan 0 di Bit-plane 0, Baris 6, Kolom 8, Karakter T
49. Sisipkan 0 di Bit-plane 0, Baris 7, Kolom 1, Karakter A
50. Sisipkan 1 di Bit-plane 0, Baris 7, Kolom 2, Karakter A
51. Sisipkan 0 di Bit-plane 0, Baris 7, Kolom 3, Karakter A
52. Sisipkan 0 di Bit-plane 0, Baris 7, Kolom 4, Karakter A
53. Sisipkan 0 di Bit-plane 0, Baris 7, Kolom 5, Karakter A
54. Sisipkan 0 di Bit-plane 0, Baris 7, Kolom 6, Karakter A

55. Sisipkan 0 di Bit-plane 0, Baris 7, Kolom 7, Karakter A
56. Sisipkan 1 di Bit-plane 0, Baris 7, Kolom 8, Karakter A
57. Sisipkan 0 di Bit-plane 0, Baris 8, Kolom 1, Karakter R
58. Sisipkan 1 di Bit-plane 0, Baris 8, Kolom 2, Karakter R
59. Sisipkan 0 di Bit-plane 0, Baris 8, Kolom 3, Karakter R
60. Sisipkan 1 di Bit-plane 0, Baris 8, Kolom 4, Karakter R
61. Sisipkan 0 di Bit-plane 0, Baris 8, Kolom 5, Karakter R
62. Sisipkan 0 di Bit-plane 0, Baris 8, Kolom 6, Karakter R
63. Sisipkan 1 di Bit-plane 0, Baris 8, Kolom 7, Karakter R
64. Sisipkan 0 di Bit-plane 0, Baris 8, Kolom 8, Karakter R

Bit-plane 0 setelah penyisipan

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

=>



Gambar 2.94 Bit-plane 0 Setelah Penyisipan

2.6 Citra digital

Citra digital adalah representasi dari gambar dua dimensi yang terdiri dari elemen-elemen gambar yang dikenal sebagai piksel. Secara umum, citra digital dapat direpresentasikan dalam bentuk matriks $f(x,y)$, di mana terdapat M kolom dan N baris. Pertemuan antara kolom dan baris tersebut membentuk piksel, atau yang biasa disebut sebagai elemen gambar (*picture element*) (Wahyu, 2022). Citra digital dapat dikelompokkan menjadi tiga kategori utama:

1. Citra Biner: Jenis citra ini hanya memiliki dua warna pada setiap piksel, yaitu hitam dan putih. Oleh karena itu, setiap piksel pada citra biner cukup diwakili oleh 1 bit, yang hanya dapat berisi nilai 0 atau 1.
2. Citra *Grayscale*: Citra ini memperlihatkan gradasi warna dari putih ke hitam. Setiap piksel dalam citra grayscale diwakili oleh 8 bit (*1 byte*), yang memungkinkan adanya 256 tingkat intensitas warna abu-abu.
3. Citra RGB (*Red, Green, Blue*): Citra ini memiliki kemampuan untuk menampilkan berbagai warna. Setiap piksel dari citra RGB terdiri dari tiga komponen warna merah, hijau, dan biru. Setiap komponen memiliki rentang nilai antara 0 hingga 255 (Arief, 2021).

2.7 Eskalasi Keamanan File

Secara umum, eskalasi keamanan merujuk pada proses meningkatkan tingkat perlindungan terhadap suatu sistem, data, atau informasi dari ancaman atau risiko yang lebih besar. Dalam konteks ini, eskalasi keamanan file teks berarti peningkatan upaya perlindungan terhadap file teks untuk memastikan bahwa informasi yang terkandung di dalamnya aman dan terlindungi dari akses yang tidak sah, pencurian data, atau kerusakan. Eskalasi keamanan dapat mencakup berbagai tindakan atau teknologi yang digunakan untuk memperkuat pertahanan file, seperti:

1. Enkripsi untuk menyembunyikan isi file.
2. Steganografi untuk menyembunyikan keberadaan file atau informasi di dalamnya.
3. Kontrol akses untuk membatasi siapa yang bisa melihat atau memodifikasi file.
4. Penggunaan kata sandi atau otentikasi ganda untuk mencegah akses oleh pihak yang tidak berwenang.

Dengan eskalasi keamanan, tujuan utamanya adalah untuk mengurangi risiko dan ancaman yang mungkin merusak atau mengekspos informasi yang terkandung dalam file teks, serta memastikan data tetap aman meskipun terjadi potensi kebocoran atau pencurian (Salim, 2022).

Adapun beberapa contoh jenis file teks yang bersifat rahasia:

1. Surat Wasiat: yaitu surat wasiat yang berisi instruksi atau keputusan seseorang mengenai pembagian harta setelah kematiannya.
2. Kontrak atau Perjanjian Kerja: yaitu dokumen yang berisi ketentuan mengenai pekerjaan, tanggung jawab, dan hak-hak dari kedua belah pihak mencakup informasi sensitif seperti gaji, hak atas kekayaan intelektual, dan ketentuan lainnya yang perlu dilindungi.
3. Dokumen Keuangan Pribadi: yaitu laporan keuangan, rekening bank, kartu kredit, atau dokumen lainnya yang mengandung informasi finansial pribadi.

4. Data Medis Pribadi: yaitu data medis atau hasil tes kesehatan yang mengandung informasi pribadi tentang kondisi fisik atau mental seseorang.
5. Dokumen Identitas Pribadi: yaitu dokumen yang mengandung informasi identitas seperti KTP, paspor, SIM, atau nomor identitas lainnya. Data ini sangat penting dan bisa disalahgunakan jika jatuh ke tangan yang salah.
6. Dokumen Hukum: yaitu bukti atau dokumen terkait dengan proses hukum persidangan yang menyangkut seseorang atau perusahaan.
7. Surat Pengunduran Diri: yaitu surat yang berisi pengunduran diri atau penghentian kerja seseorang mencakup informasi terkait alasan pribadi atau profesional yang harus dijaga kerahasiaannya.
8. Proposal Bisnis: yaitu proposal yang berisi informasi bisnis yang belum dipublikasikan seperti strategi pemasaran, pengembangan produk, atau data pasar yang bisa merugikan perusahaan jika bocor.
9. Dokumen Rahasia Pemerintah: yaitu laporan atau dokumen yang mengandung informasi sensitif atau strategis terkait kebijakan, perencanaan, atau keamanan negara.
10. Kekayaan Intelektual: yaitu data atau hasil penelitian dan pengembangan yang belum dipatenkan atau dipublikasikan, yang melibatkan penemuan baru, algoritma, atau teknologi yang memiliki nilai ekonomi.