Jurnal INSAN (Journal of Information Systems Management Innovation)
Volume 6 No. 1 Juni 2026
E-ISSN: 2777-1385

Analisis Keamanan Data Pada Sistem Informasi Menggunakan AES-128 Berbasis Transformasi Kunci ASCII Ke Heksadesimal

Sri Mulyan Farhatan Nurazizah[1], Ai Ilah Warnilah[2*], Izma Milien[3] , Deddy Supriadi[4] ,Ratningsih[5]

Program Studi Sistem Informasi, Universitas Bina Sarana Informatika Kampus Tasikmalaya
Jalan tanuwijaya No. 4, Empangsari, Tawang, Tasikmalaya.

email : [1] 19241324@bsi.ac.id, [2*] ai.aiw@bsi.ac.id, [3] 19242269@bsi.ac.id,
[4] deddy.dys@bsi.ac.id, [5] ratningsih.rnn@bsi.ac.id,

Artikel Info : Diterima : 18-05-2026 | Direvisi : 24-06-2026 | Disetujui : 26-06-2026

Abstrak - Perkembangan sistem informasi yang pesat meningkatkan kebutuhan terhadap mekanisme pengamanan data yang efektif. Data yang dikirimkan melalui jaringan tanpa perlindungan kriptografi berpotensi mengalami penyadapan, manipulasi, dan kebocoran informasi. Algoritma Advanced Encryption Standard (AES) merupakan salah satu metode kriptografi modern yang banyak digunakan karena tingkat keamanan dan efisiensinya. Penelitian ini bertujuan untuk menganalisis keamanan data pada sistem informasi menggunakan algoritma AES-128 berbasis transformasi kunci ASCII ke heksadesimal. Transformasi kunci dilakukan untuk memastikan kesesuaian representasi kunci terhadap panjang bit yang dibutuhkan dalam proses enkripsi AES-128. Metode penelitian meliputi implementasi proses enkripsi dan dekripsi serta analisis perubahan *ciphertext* akibat variasi kunci. Hasil pengujian menunjukkan bahwa perubahan kecil pada kunci menghasilkan perbedaan *ciphertext* yang signifikan, sehingga menunjukkan sensitivitas dan ketahanan algoritma terhadap serangan *cryptanalysis*. Dengan demikian, penerapan AES-128 berbasis transformasi kunci ASCII ke heksadesimal dapat meningkatkan keamanan data pada sistem informasi

Kata Kunci : AES-128, Kriptografi, Keamanan Data, Transformasi Kunci, Sistem Informasi

Abstracts *- The rapid development of information systems has increased the need for effective data security mechanisms. Data transmitted over networks without cryptographic protection is vulnerable to interception, manipulation, and information leakage. The Advanced Encryption Standard (AES) algorithm is one of the modern cryptographic methods widely used due to its security level and computational efficiency. This study aims to analyze data security in information systems using the AES-128 algorithm based on ASCII to hexadecimal key transformation. The key transformation is performed to ensure compatibility with the required bit length in the AES-128 encryption process. The research method includes the implementation of encryption and decryption processes as well as analysis of ciphertext changes caused by key variations. The results indicate that minor changes in the key produce significant differences in ciphertext, demonstrating the algorithm's sensitivity and resistance to cryptanalytic attacks. Therefore, the implementation of AES-128 based on ASCII to hexadecimal key transformation can enhance data security in information systems.*

Keywords : AES-128, Cryptography, Data Security, Key Transformation, Information Systems

PENDAHULUAN

Perkembangan teknologi informasi telah meningkatkan intensitas pertukaran data digital dalam berbagai sistem informasi. Namun, peningkatan tersebut juga diikuti oleh meningkatnya risiko ancaman keamanan data, seperti penyadapan dan manipulasi informasi oleh pihak yang tidak berwenang. Keamanan data menjadi aspek penting untuk menjamin kerahasiaan, integritas, dan ketersediaan informasi dalam sistem digital (Ananda & Lukman, 2022).

Kriptografi merupakan metode yang digunakan untuk mengamankan data dengan cara mengubah plaintext menjadi ciphertext menggunakan algoritma dan kunci tertentu. Algoritma kriptografi modern dirancang dengan kompleksitas matematis tinggi sehingga sulit dipecahkan tanpa mengetahui kunci yang digunakan (Setiawan & Mulyati, 2024). Salah satu algoritma kriptografi simetris yang широко digunakan adalah Advanced Encryption Standard (AES). AES bekerja dengan mekanisme block cipher melalui beberapa tahapan transformasi seperti SubBytes, ShiftRows, MixColumns, dan AddRoundKey (Khoirunnisa et al., 2025). Beberapa penelitian sebelumnya menunjukkan bahwa AES-128 memiliki performa yang baik dalam mengamankan dokumen digital

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

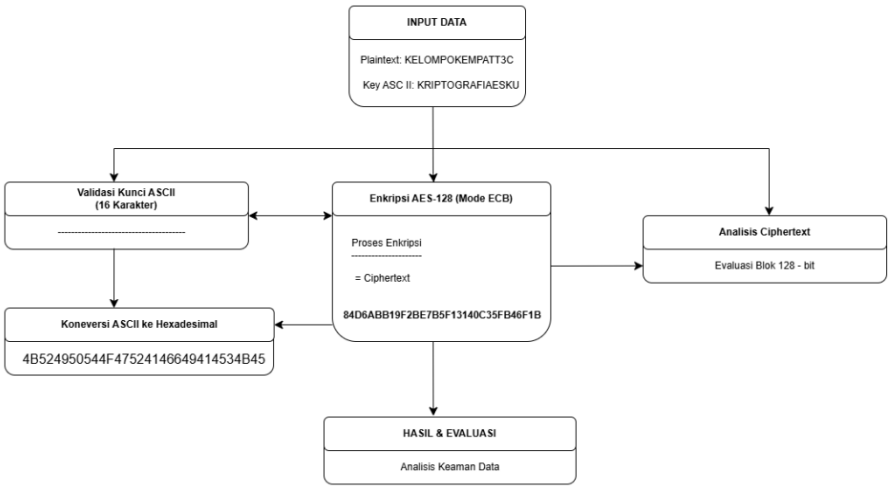
dan data berbasis web karena efisiensi proses komputasi serta tingkat keamanan yang tinggi (Sukiman et al., 2026). Selain itu, pengujian avalanche effect menunjukkan bahwa perubahan kecil pada plaintext atau kunci dapat menghasilkan perubahan signifikan pada ciphertext, yang menjadi indikator kuatnya algoritma kriptografi (Sholikhatin et al., 2023).

Meskipun demikian, salah satu aspek penting dalam implementasi AES adalah pengelolaan dan format kunci. Representasi kunci harus sesuai dengan kebutuhan panjang bit yang ditentukan, khususnya pada AES-128 yang menggunakan panjang kunci 128-bit. Dalam implementasinya, kunci direpresentasikan dalam bentuk byte dan umumnya ditampilkan dalam format heksadesimal untuk mempermudah proses komputasi serta menjaga konsistensi struktur data selama tahapan transformasi (Journal & Bodipudi, 2023). Selain itu, pengolahan dan penyesuaian format kunci sebelum proses enkripsi juga menjadi bagian penting dalam memastikan tidak terjadi kesalahan interpretasi data pada tahap pembangkitan dan ekspansi kunci (Patty et al., 2025). Oleh karena itu, transformasi kunci dari ASCII ke heksadesimal dapat diposisikan sebagai langkah praproses untuk menyesuaikan representasi kunci dengan kebutuhan struktur bit dalam algoritma AES-128.

Meskipun algoritma AES-128 telah banyak diterapkan untuk mengamankan data pada berbagai sistem informasi, sebagian besar penelitian sebelumnya hanya berfokus pada implementasi proses enkripsi dan dekripsi tanpa membahas representasi kunci sebelum digunakan pada algoritma AES. Penelitian ini menawarkan pendekatan berupa transformasi kunci dari format ASCII ke heksadesimal sebagai tahap praproses sebelum proses enkripsi AES-128. Transformasi tersebut bertujuan memastikan representasi kunci sesuai dengan struktur 128-bit yang digunakan algoritma AES sehingga proses pembangkitan kunci menjadi lebih konsisten. Kebaruan penelitian ini terletak pada analisis penerapan transformasi kunci ASCII ke heksadesimal terhadap proses enkripsi AES-128 serta pengaruhnya terhadap keamanan data pada sistem informasi.

Berdasarkan uraian tersebut, penelitian ini berfokus pada analisis keamanan data pada sistem informasi menggunakan algoritma AES-128 berbasis transformasi kunci ASCII ke heksadesimal. Penelitian ini bertujuan untuk mengevaluasi proses enkripsi dan dekripsi serta menganalisis sensitivitas perubahan ciphertext terhadap variasi kunci sebagai indikator tingkat keamanan sistem.

METODE PENELITIAN



Gambar 1. Proses Analisis Keamanan Data Sistem Informasi dengan AES 128
Sumber: (Artikel & Wiharto, 2022)

Penelitian ini menggunakan pendekatan eksperimen kuantitatif berbasis implementasi kriptografi untuk menganalisis keamanan data pada sistem informasi menggunakan algoritma Advanced Encryption Standard (AES-128). Metodologi dirancang secara sistematis untuk mengevaluasi validitas kunci, dan konsistensi blok. Berikut Langkah-langkah yang harus dilakukan Adalah:

1. Penentuan parameter eksperimen, yaitu plaintext "KELOMPOKEMPATT3C" sebagai representasi data sistem informasi dan kunci ASCII "KRIPTOGRAFIAESKU". Kunci yang digunakan terdiri dari 16 karakter ASCII. Mengingat satu karakter ASCII bernilai 1 byte (8-bit), maka total panjang kunci adalah 16 byte atau 128-bit, sesuai dengan standar AES-128.
2. Validasi panjang kunci. Pada tahap ini dilakukan verifikasi bahwa jumlah karakter kunci tepat 16 karakter sehingga memenuhi persyaratan algoritma AES-128. Validasi ini penting untuk memastikan kesesuaian struktur internal algoritma terhadap ukuran blok dan key schedule.
3. Transformasi kunci dari ASCII ke representasi heksadesimal. Setiap karakter ASCII dikonversi menjadi nilai

heksadesimal dua digit, sehingga diperoleh 32 digit heksadesimal yang sebanding dengan 128-bit. Transformasi ini dilakukan karena proses internal AES bekerja dalam representasi biner/heksadesimal, bukan dalam bentuk teks langsung.

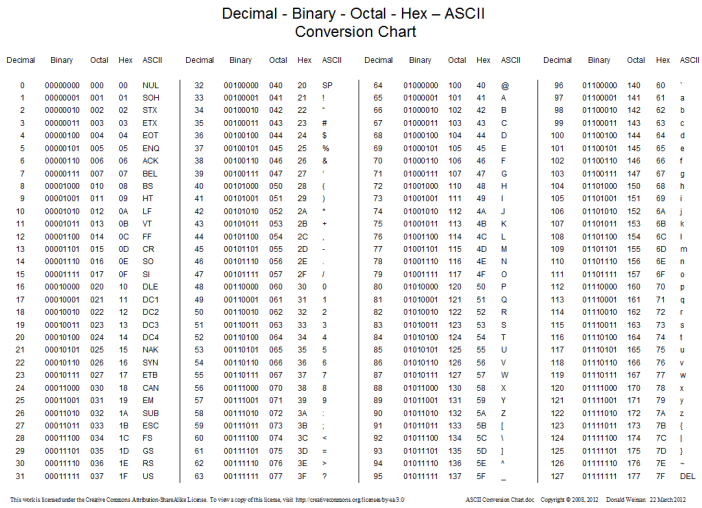
Proses enkripsi menggunakan AES-128 mode Electronic Code Book (ECB). Algoritma AES-128 bekerja melalui 10 ronde transformasi yang meliputi SubBytes, ShiftRows, MixColumns, dan AddRoundKey. Plaintext yang telah dikonversi ke dalam blok 128-bit dienkripsi menggunakan kunci hasil transformasi sebelumnya.

HASIL DAN PEMBAHASAN

1. Enkripsi

K	R	I	P	T	O	G	R	A	F	I	A	E	S	K	U
4B	52	49	50	54	4F	47	52	41	46	49	41	45	53	4B	55

Ekspansi kunci dibutuhkan untuk proses enkripsi dan deskripsi pada algoritma Advanced Encryption Standard (AES). Maksimal panjang kunci pada algoritma Advanced Encryption Standard 128 bit Adalah 16 digit yang membutuhkan 10 kunci ronde (RoundKey) dalam ekspansi kunci. Kunci yang digunakan pada kasus ini Adalah " KRIPTOGRAFIAESKU".



Decimal - Binary - Octal - Hex – ASCII
Conversion Chart

Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII
0	00000000	000	00	NUL	32	00100000	040	20	SP	64	01000000	100	40	@	96	01100000	140	60	`
1	00000001	001	01	SOH	33	00100001	041	21	!	65	01000001	101	41	A	97	01100001	141	61	a
2	00000010	002	02	STX	34	00100010	042	22	"	66	01000010	102	42	B	98	01100010	142	62	b
3	00000011	003	03	ETX	35	00100011	043	23	#	67	01000011	103	43	C	99	01100011	143	63	c
4	00000100	004	04	EOT	36	00100100	044	24	$	68	01000100	104	44	D	100	01100100	144	64	d
5	00000101	005	05	ENQ	37	00100101	045	25	%	69	01000101	105	45	E	101	01100101	145	65	e
6	00000110	006	06	ACK	38	00100110	046	26	&	70	01000110	106	46	F	102	01100110	146	66	f
7	00000111	007	07	BEL	39	00100111	047	27	'	71	01000111	107	47	G	103	01100111	147	67	g
8	00001000	010	08	BS	40	00101000	050	28	(	72	01001000	110	48	H	104	01101000	150	68	h
9	00001001	011	09	HT	41	00101001	051	29	)	73	01001001	111	49	I	105	01101001	151	69	i
10	00001010	012	0A	LF	42	00101010	052	2A	*	74	01001010	112	4A	J	106	01101010	152	6A	j
11	00001011	013	0B	VT	43	00101011	053	2B	+	75	01001011	113	4B	K	107	01101011	153	6B	k
12	00001100	014	0C	FF	44	00101100	054	2C	,	76	01001100	114	4C	L	108	01101100	154	6C	l
13	00001101	015	0D	CR	45	00101101	055	2D	-	77	01001101	115	4D	M	109	01101101	155	6D	m
14	00001110	016	0E	SO	46	00101110	056	2E	.	78	01001110	116	4E	N	110	01101110	156	6E	n
15	00001111	017	0F	SI	47	00101111	057	2F	/	79	01001111	117	4F	O	111	01101111	157	6F	o
16	00010000	020	10	DLE	48	00110000	060	30	0	80	01010000	120	50	P	112	01110000	160	70	p
17	00010001	021	11	DC1	49	00110001	061	31	1	81	01010001	121	51	Q	113	01110001	161	71	q
18	00010010	022	12	DC2	50	00110010	062	32	2	82	01010010	122	52	R	114	01110010	162	72	r
19	00010011	023	13	DC3	51	00110011	063	33	3	83	01010011	123	53	S	115	01110011	163	73	s
20	00010100	024	14	DC4	52	00110100	064	34	4	84	01010100	124	54	T	116	01110100	164	74	t
21	00010101	025	15	NAK	53	00110101	065	35	5	85	01010101	125	55	U	117	01110101	165	75	u
22	00010110	026	16	SYN	54	00110110	066	36	6	86	01010110	126	56	V	118	01110110	166	76	v
23	00010111	027	17	ETB	55	00110111	067	37	7	87	01010111	127	57	W	119	01110111	167	77	w
24	00011000	030	18	CAN	56	00111000	070	38	8	88	01011000	130	58	X	120	01111000	170	78	x
25	00011001	031	19	EM	57	00111001	071	39	9	89	01011001	131	59	Y	121	01111001	171	79	y
26	00011010	032	1A	SUB	58	00111010	072	3A	:	90	01011010	132	5A	Z	122	01111010	172	7A	z
27	00011011	033	1B	ESC	59	00111011	073	3B	;	91	01011011	133	5B	[	123	01111011	173	7B	{
28	00011100	034	1C	FS	60	00111100	074	3C	<	92	01011100	134	5C	\	124	01111100	174	7C	|
29	00011101	035	1D	GS	61	00111101	075	3D	=	93	01011101	135	5D	]	125	01111101	175	7D	}
30	00011110	036	1E	RS	62	00111110	076	3E	>	94	01011110	136	5E	^	126	01111110	176	7E	~
31	00011111	037	1F	US	63	00111111	077	3F	?	95	01011111	137	5F	_	127	01111111	177	7F	DEL

This work is licensed under the Creative Commons Attribution-ShareAlike License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/ ASCII Conversion Chart.doc Copyright © 2003, 2012 Donald Weiman 22 March 2012

Gambar 2. Tabel ASC II
Sumber: (Information & Standards, 2023)

a. Urutkan plaintext kunci kedalam blok berukuran 128 bit (16 Kode ASCII), kemudian kunci diubah kedalam bentuk Hexadecimal.

K	E	L	O	M	P	O	K	E	M	P	A	T	T	3	C
4B	45	4C	4F	4D	5D	4F	4B	45	4D	50	41	54	54	33	43

b. Selanjutnya Adalah mengubah kunci yang telah diubah ke dalam state 4 x 4 seperti berikut:

4B	45	4C	4F
4D	50	4F	4B
45	4D	50	41
52	54	33	43

Roundkey ke – 0

c. Setelah itu, melakukan fungsi RotWord, yaitu dengan menggeser setiap bit pada kolom 4 ke atas 1 kali menggunakan RoundKey ke-0 untuk menghasilkan RoundKey ke 1.

74

32
33
43

32
33
43
74

d. Setelah itu, melakukan substitusi hasil dari RotWord dengan nilai yang ada pada tabel S-Box (SubBytes)

		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

Figure 7. S-box: substitution values for the byte xy (in hexadecimal format).

Gambar 1.3 Tabel S-Box

Sumber: (Artikel & Wiharto, 2022)

Cara mengambil nilai S-Box

32
33
43
74

=➔

23
C3
1A
92

Untuk mendapatkan nilai S-Box dari input 32, kita memisahkan angka heksadesimalnya menjadi 3 sebagai baris dan 2 sebagai kolom, kemudian melihat tabel S-Box pada perpotongan baris ke-3 dan kolom ke-2. Nilai yang diperoleh dari posisi tersebut adalah 23.

1) 32
   - 3 = baris 3
   - 2 = kolom 2
   - → S-Box[3][2] = 23
2) 33
   - 3 = baris 3
   - 3 = kolom 3
   - → S-Box[3][3] = C3
3) 43
   - 4 = baris 4
   - 3 = kolom 3
   - → S-Box[4][3] = 1A
4) 74
   - 7 = baris 7
   - 4 = kolom 4
   - → S-Box[7][4] = 92

e. Selanjutnya, untuk mendapatkan kolom pertama dari RoundKey ke-1 adalah proses XOR antara kolom pertama dari RoundKey ke-0 dan hasil dari SubBytes di XOR-kan dengan Rcon.

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

Gambar 4. Tabel Rcon

Sumber: (Information & Standards, 2023)

Note: kolom ke 1

4B		23		01		69
65		C3		00		A6
6C	⊕	1A	⊕	00	=	76
6F		92		00		FD

Baris 1
$4B \oplus 23 \oplus 01$

$4B \oplus 23 = 68$
$68 \oplus 01 = 69$
Hasil baris 1 = 69
Lakukan ke baris berikutnya seperti Langkah bariss pertama sampai dengan selesai.

f. Untuk mendapatkan nilai kolom selanjutnya dilakukan XOR antara kolom pertama (Wi) dengan kolom kedua dari RoundKey ke-0, kemudian untuk mendapatkan kolom

Note: kolom ke 2

6D		69		04
70		A6		D6
6F	⊕	76	=	19
6B		FD		96

Note: kolom ke 3

65		04		61
6D		D6		BB
70	⊕	19	=	69
61		96		F7

Note: kolom ke 4

74		61		15
32		BB		89
33	⊕	69	=	5A
43		F7		B4

g. Dari seluruh proses yang telah dilakukan seperti di atas, maka didapatlah RoundKey ke-1, yaitu:

69	04	62	15
A6	D9	BB	89
76	19	69	5A
FD	96	F7	B4

Untuk mendapatkan RoundKey ke-2 sampai dengan RoundKey ke-10, proses di atas.

h. Selanjutnya masuk ke Langkah enkripsi dimulai dengan proses Subbyte dengan mengsubtitusikan ke dalam tabel S-Box.

00	19	04	0C
17	1F	0B	07
05	08	19	78
1F	19	00	16

63	D4	F2	FE
F0	C0	2B	C5
6B	30	D4	BC
C0	D4	63	47

i. Setelah tahap subtitusi kemudian lakukan ShiftRows.

63	D4	F2	FE
F0	C0	2B	C5
6B	30	D4	BC
C0	D4	63	47

63	D4	F2	FE
C0	2B	C5	F0
D4	BC	6B	30
47	C0	D4	63

j. Setelah melakukan tahap ShiftRows kemudian masuk ke proses MixColumns.

63	D4	F2	FE
C0	2B	C5	F0
D4	BC	6B	30
47	C0	D4	63

x

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

8E	4D	A1	BC
4D	7D	3F	D1
A1	3F	DE	BC
BC	D1	BC	0E

k. Transformasi akhir setelah melakukan beberapa tahapan yaitu AddRoundkey.

8E	4D	A1	BC
4D	7D	3F	D1
A1	3F	DE	BC
BC	D1	BC	0E

⊕

A7	F3	B2	F7
E1	AE	E8	BB
B5	F2	BB	FD
3E	6C	2D	78

29	AC	14	82
BE	DE	CD	BD
13	D7	B5	91
4B	6A	41	76

Lakukan Langkah tersebut 10 round sampai menemukan kunci akhir.

l. Berikut hasil tranasformasi proses enkripsi round ke – 2 sampai dengan round ke-10.

SubByte

67	98	39	0A
E8	D4	75	CE
54	DF	1A	03
16	37	8B	02

ShiftRows

67	98	39	0A
D4	75	CE	E8
1A	D3	54	DF
02	16	37	8B

RoundKey Ke-10

E3	07	C8	55
02	5E	FF	5C
B1	34	14	B0
B3	A3	F4	90

AddRoundKey

84	9F	F1	5F
D6	2B	31	B4
AB	E7	40	6F
B1	B5	C3	1B

Hasil dari proses enkripsi yaitu: **84D6ABB19F2BE7B5F13140C35FB46F1B**

2. Deskripsi

Proses ini dilakukan untuk mengembalikan record yang telah dienkripsi menjadi plaintext kembali. Transformasi deskripsi pada algoritma advanced encryption standard (AES) 128 bit adalah InvSubBytes, InvShiftRows, InvMixColumns, dan AddRoundKey . Berikut adalah proses dekripsi chipertext "84D6ABB19F2BE7B5F13140C35FB46F1B", yaitu:

a. Langkah awal yang harus dilakukan untuk proses deskripsi Adalah melakukan proses XOR anatara chipertext dengan Roundkey ke- 10.

84	9F	F1	5F
D6	2B	31	B4
AB	E7	40	6F
B1	B5	C3	1B

⊕

E3	07	C8	55
02	5E	FF	5C
81	34	14	B0
B3	A3	F4	90

67	98	39	0A
D4	75	CE	E8
1A	D3	54	DF
02	16	37	8B

b. Selanjutnya, Pada round ke-1 sampai round ke-9 proses deskripsi dilakukan transformasi InvShiftRows, InvSubBytes, InvMixColumns dan AddRoundKey.

Round ke-1

67	98	39	0A
D4	75	CE	E8
1A	D3	54	DF
02	16	37	8B

67	98	39	0A
E8	D4	75	CE
54	DF	1A	D3
16	37	8B	02

c. Kemudian, lakukan proses InvSubBytes dengan mengsubtitusikan ke dalam Tabel Inverse S-Box.

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

Gambar 1.5 Tabel Inverse S-Box
Sumber: (Artikel & Wiharto, 2022)

67	98	39	0A
E8	D4	75	CE
54	DF	1A	D3
16	37	8B	02

00	E2	5B	A3
C8	19	3F	EC
FD	EF	43	A9
FF	B2	CE	6A

d. Hasil dari Inverse SubByte kemudian di XOR kan dengan Roundkey (Kunci Putaran) ke-9 pada tabel Roundkey yang telah ditentutkan pada saat awal.

00	E2	5B	A3
C8	19	3F	EC
FD	EF	43	A9
FF	B2	CE	6A

$\oplus$

DF	E4	CF	9D
4B	5C	A1	A3
F2	85	20	A4
ED	10	57	64

DF	06	94	3E
83	45	9E	4F
0F	6A	63	0D
12	A2	99	0E

e. Setelah melakukan proses Inverse SubByte kemudian masuk ke proses Inverse MixColumn

DF	06	94	3E
83	45	9E	4F
0F	6A	63	0D
12	A2	99	0E

x

0E	0B	0D	09
09	0E	0B	0D
0D	09	0E	0B
0B	0D	09	0E

00	E2	5B	A3
C8	19	3F	3C
FD	EF	43	A9
FF	B2	CE	6A

Lakukan langkah tersebut sampai dengan RoundKey ke – 0. Sehingga mendapatkan hasil akhir atau chipertext sebenarmya.

f. Transformasi akhir dari langkah deskripsi yaitu

Inverse ShiftRows

63	D4	F2	82
D4	F0	C0	30
D4	63	6B	30
D4	63	47	C0

Inverse SubByte

00	19	04	11
17	1F	0B	07
05	08	19	78
1F	19	00	16

RoundKey ke-0

4B	54	41	43
52	4F	46	53
49	47	49	4B
50	52	41	55

Hasil akhir dari deskripsi

4B	45	4C	4F
4D	50	4F	4B
45	4D	50	41
52	54	33	43

K	E	L	O	M	P	O	K	E	M	P	A	T	T	3	C
4B	45	4C	4F	4D	5D	4F	4B	45	4D	50	41	54	54	33	43

KESIMPULAN

Berdasarkan tujuan penelitian yang telah dirumuskan pada bagian pendahuluan, yaitu untuk menganalisis keamanan data pada sistem informasi menggunakan algoritma AES-128 berbasis transformasi kunci ASCII ke heksadesimal, hasil penelitian menunjukkan bahwa proses transformasi kunci berhasil memastikan kesesuaian representasi kunci dengan struktur 128-bit yang dipersyaratkan oleh algoritma AES-128. Implementasi enkripsi dan dekripsi yang dilakukan membuktikan bahwa algoritma bekerja secara konsisten melalui tahapan transformasi kriptografi hingga mampu menghasilkan ciphertext yang berbeda secara signifikan ketika terjadi perubahan pada kunci, serta mampu mengembalikan data ke bentuk awal melalui proses dekripsi. Hal ini menunjukkan bahwa mekanisme yang diterapkan kompatibel dengan prinsip keamanan kriptografi dalam menjaga kerahasiaan dan integritas data pada sistem informasi.

Secara umum, penerapan AES-128 dengan praproses transformasi kunci memberikan kontribusi dalam meningkatkan validitas format kunci serta mendukung kestabilan proses ekspansi kunci pada sistem. Prospek pengembangan penelitian ini dapat diarahkan pada pengujian implementasi AES dalam mode operasi yang berbeda, seperti CBC atau GCM, untuk meningkatkan aspek keamanan terhadap pola blok data. Selain itu, penelitian lanjutan dapat mengkaji integrasi algoritma AES dengan mekanisme manajemen kunci atau kombinasi dengan algoritma kriptografi asimetris guna memperkuat sistem keamanan secara menyeluruh sesuai dengan perkembangan kebutuhan keamanan informasi modern.

REFERENSI

Afthar Kautsar;Muhammad Ikhsan. (2025). *Implementasi Algoritma Advanced Encryption Standard (AES) dan Teknik Steganografi Bit Plane Complexity Segmentation (BPCS) dalam Eskalasi Keamanan File Teks Implementation of the Advanced Encryption Standard (AES) Algorithm and Bit Plane Complexity*. *14*, 956–968.

Ananda, S. P., & Lukman, S. (2022). *Analisa Metode Kriptogra Modern Advance Encryption Standar (AES) 128 Bit dalam Mengenkripsi dan Mendekripsi File Dokumen Digital Pendahuluan*. *21*(September), 333–344.

Artikel, R., & Wiharto, Y. (2022). *Implementasi Advanced Encryption Standard 128 Sebagai Pengamanan Basis Data Obat- o batan Apotek Implementation of Advanced Encryption Standard 128 to Secure Pharmacy Drug Database*. *8*, 335–350.

Chete, F. O., & Okpako, A. E. (2024). *Text Encryption Using Advanced Encryption Standard (AES) Algorithm*. *6*(2), 214–228.

Djami, D., Ninu, I., Naitkakin, P. T., Rosana, N., Apriani, A., Blaang, D., Pertanian, F., Kesehatan, S. D., Informasi, T., & Timor, U. (2025). *IMPLEMENTASI ALGORITMA AES-128 UNTUK PENGAMANAN TEXT DALAM SISTEM KEAMANAN DATA*. *3*(6), 815–824.

Dokumen, E., Geulis, G., & Abadi, E. (2022). *Penerapan Algoritma AES (Advance Encryption Standart) 128 untuk*. *5*, 1–10.

Hasan, M. A. (2022). *Aplikasi Keamanan Data Berbasis Web Menggunakan Algoritma AES 128 Untuk Enkripsi Dan Dekripsi Data*. *2*(2).

Info, A., Encryption, A., & Encryption, S. (2025). *Enhancing Better Security for Encryption Decryption of Data using AES Algorithm*. *11*(09), 26–33.

Information, F., & Standards, P. (2023). *Advanced Encryption Standard (AES)*.

Irwanto, D. (2024). *File Encryption and Decryption Using Algorithm Aes-128 Bit Based Website*. *4*(April), 670–677.

Journal, E., & Bodipudi, A. (2023). *Development of New Cryptographic Protocols with Optimized Algorithms and Encryption*. *10*(9), 94–103.

Khoirunnisa, N. A., Satra, R., Widyawati, D., & Indonesia, U. M. (2025). *Implementasi Algoritma Kriptografi AES untuk Peningkatan Keamanan Proses Login Website*. *2*(3), 317–326.

Naimnule, F. A., Hanoe, F. A. L., Banusu, M. N., Mano, M. O., Studi, P., Informasi, T., & Timor, U. (2025). Implementation of AES Encryption for Data Security on Web-Based Information Systems in Fafinesu A Village. *Sistem Kendali & Jaringan) E-ISSN*, *4*(3), 2808–3520. https://doi.org/10.58982/krisnadana.v4i3.836https://ejournal.sidyanusa.org/index.php/jkdn/index

Nasrullah, A. H. (2025). *Secure Web-Based File Encryption Using AES-128*. *6*(2), 146–155.

Patty, D., Kololu, S. M., Dahoklory, N., & Leleury, Z. A. (2025). *Implementation of the Advanced Encryption Standard (AES) Algorithm to Protect Children Personal Data*. *04*(2), 261–274.

Rasidin, A., & Nugroho, A. J. (2022). *Analisa Penggunaan Teknik Advanced Encryption (AES) dalam Kriptografi Algoritma Rijndael memiliki spesifikasi : • Rijndael Dapat mendukung fleksibilitas dengan Panjang pengunci 128-byte hingga 256- • Setiap blok terenkripsi pada nomor bilangan bulat tentu saja sama dengan DES . April*, 1–3.

Sagala, H. A. (2023). *Perancangan Aplikasi Audit Internal Dengan Menerapkan Algoritma AES 128 Bit Untuk Pengamanan Data*. *2*(2), 75–86.

Setiawan, Y. R., & Mulyati, S. (2024). *IMPLEMENTASI KRIPTOGRAFI ALGORITME ADVANCED ENCRYPTION STANDARD (AES-128) UNTUK PENGAMANAN DOKUMEN BERBASIS WEB PADA PT . XYZ*. *3*(April), 30–39.

Sholikhatin, S. A., Kuncoro, A. P., Munawaroh, A. L., Setiawan, G. A., & Artikel, I. (2023). *Comparative Study of RSA Asymmetric Algorithm and AES Algorithm for Data Security*. *9*(127), 60–67.

Sukiman, T. S. A., Meiyanti, R., Lika, C., Sandy, M., Rizal, R. A., & Fadlan, S. (2026). *Gameology And Multimedia Expert Implementation Of AES Algorithm for Text Message Encryption and Decryption Process*. *3*(1), 25–30.