IMPLEMENTASI ALGORITMA *ADVANCED ENCRYPTION STANDARD* 128 UNTUK PENGAMANAN DATABASE SISTEM REGISTRASI PASIEN

Tarisa Auliya Ramadhani[1], Adi Fajaryanto Cobantoro[2], Sugianti[3]

[1,2,3] Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Ponorogo, Indonesia
[1]tarisaauliyaramadhani@gmail.com, [2] adifajaryanto@umpo.ac.id, [3] sugianti@umpo.ac.id

Abstrak

Kehidupan masyarakat Indonesia sangat terkena dampak kemajuan ilmu pengetahuan dan teknologi, khususnya di bidang keamanan siber. Keamanan siber sangat penting untuk melindungi jaringan komputer, perangkat, dan data dari akses tidak sah dan bahaya lainnya. Penelitian ini mengkaji pelayanan kesehatan Arif Merbabu Care di Ponorogo sebagai studi kasus untuk lebih memahami bagaimana algoritma *Advanced Encryption Standard* (AES-128) dapat digunakan untuk mengamankan data pasien di Sistem Registrasi Pasien. Masalah utama yang diidentifikasi adalah tingginya risiko kebocoran data pribadi pasien, yang mengancam privasi dan keamanan informasi sensitif. Untuk mengatasi masalah ini, penelitian ini menerapkan metode algoritma AES-128. Pada tahap implementasi, algoritma AES-128 digunakan untuk mengenkripsi data pasien sebelum disimpan ke dalam database, meliputi informasi seperti NIK, nama, alamat, data kelahiran, jenis kelamin, umur, dan nomor telepon. Data yang dianalisis dalam penelitian ini mencakup catatan medis dan literatur ilmiah terkait enkripsi dan keamanan data. Hasil penelitian ini menunjukkan bahwa mengenkripsi data menggunakan AES-128 dapat menurunkan kemungkinan kebocoran data, melindungi privasi pasien, dan menghentikan akses ilegal. Keunggulan AES-128 dibandingkan algoritma lainnya adalah efisiensinya dalam proses enkripsi dan dekripsi serta keakuratan pengembalian data ke bentuk asal tanpa kehilangan informasi. Pengujian sistem menunjukkan bahwa enkripsi AES-128 memberikan perlindungan kuat terhadap peretasan, meningkatkan kepercayaan pasien dan keamanan operasional lembaga kesehatan. Implementasi algoritma AES-128 pada sistem registrasi pasien terbukti efektif dalam meningkatkan keamanan database pasien, sehingga penting bagi lembaga kesehatan lainnya untuk melindungi data sensitif dari ancaman kebocoran dan akses ilegal.

Kata kunci: *advanced encryption standard* (AES-128), database, enkripsi, keamanan data

1. Pendahuluan

Keamanan siber memiliki peran krusial dalam konteks perlindungan sistem komputer, jaringan, perangkat, dan data dari berbagai ancaman dan upaya akses illegal (Prabowo et al., 2020;Rahmawati, 2019). Selain itu, keamanan siber menjadi landasan bagi kemajuan teknologi yang semakin kompleks dan meningkatkan kesadaran masyarakat akan pentingnya menjaga keamanan informasi pribadi (Purweni et al., 2022;Kautsar, 2022). Indonesia sering kali mengalami kasus kebocoran data (Firdaus, 2022). Menurut data dari perusahaan keamanan siber Surfshark, Indonesia bahkan menempati peringkat ketiga dalam jumlah kasus kebocoran data terbanyak di dunia pada tahun 2022, dengan lebih dari 12 juta akun yang terkena dampak (CNN Indonesia, 2022). Bahkan pada tahun 2023, telah terjadi 94 insiden kebocoran data di Indonesia, dimana 35 diantaranya terjadi pada tahun tersebut. Salah satu insiden yang mencolok adalah aksi peretasan oleh kelompok LockBit yang berhasil mengeksploitasi data internal Bank Syariah Indonesia (BSI) hingga sekitar 1,5 terabyte. Tidak hanya kasus peretasan terhadap BSI, tetapi juga banyak kasus kebocoran data lainnya yang telah mencuat, seperti kebocoran data BPJS Ketenagakerjaan, data paspor, data dari Dukcapil, dan bahkan data kartu SIM (CNN Indonesia, 2023). Semua peristiwa ini menunjukan kerentanan yang serius terhadap privasi dan keamanan data individu serta lembaga-lembaga yang terkena dampak.

Untuk mengatasi masalah kebocoran data yang telah dijelaskan sebelumnya, salah satu langkah yang diambil adalah mengimplementasikan enkripsi pada tingkat database, termasuk proses pengenkripsian seluruh tabel yang ada dalam database tersebut. Dengan menerapkan enkripsi di level database, tujuan utamanya adalah memastikan bahwa keamanan data tetap terjaga dengan baik (Sholihin et al., 2022). Terlepas dari upaya akses ilegal yang terjadi di berbagai lembaga, termasuk layanan kesehatan seperti Arif Merbabu Care Ponorogo. Dalam konteks penelitian ini, digunakan algoritma enkripsi yang sudah mempunyai lisensi berstandar internasional dan telah diadopsi secara luas oleh perusahaan maupun lembaga keuangan di seluruh dunia yaitu Advanced Encryption Standard (AES-128) (Sitorus et al., 2020).

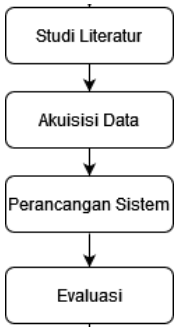
Pada penelitian terdahulu, Asri Prameshwari dan Nyoman Putra Sastra menggunakan algoritma AES-128 untuk enkripsi dan dekripsi data berupa file dokumen (PDF, DOC, TXT) (Prameshwari & Sastra, 2018). Selain itu, Azanuddin, Suardi Yakub, dan Jaka Prayudha juga melakukan penelitian dengan algoritma AES-128 yang diterapkan pada enkripsi dan dekripsi citra digital (Azanuddin et al., 2022). Penelitian oleh Ahmad Galih Pramudito dan Dewi Kusumaningsih memanfaatkan algoritma AES-128 untuk enkripsi dan dekripsi isi pesan yang dikirimkan melalui email (Pramudito & Kusumaningsih, 2018). Dari ketiga penelitian tersebut, algoritma AES-128 telah diterapkan untuk mengenkripsi dokumen, citra digital, maupun isi pesan email dengan pengaman yang baik. Oleh karena itu, pada penelitian kali ini algoritma AES-128 akan diterapkan pada database pasien.

Algoritma AES-128 adalah sistem enkripsi blok non-Feistel dengan panjang blok 128 bit. Proses enkripsinya melibatkan serangkaian ronde yang mengubah state teks asli menjadi teks terenkripsi (Sadikin, 2021).Setiap blok 128-bit dari data input dienkripsi menjadi blok 128-bit dari data keluaran. Jika panjang data tidak kelipatan 128 bit, dilakukan padding. Proses enkripsi dimulai dengan AddRoundKey, di mana blok teks asli di-XOR dengan kunci ronde awal. Setiap ronde berikutnya (ronde 1 hingga Nr-1) melibatkan empat transformasi utama: SubBytes (substitusi byte menggunakan S-Box), ShiftRows (permutasi sirkuler byte di setiap baris), MixColumns (pengacakan kolom menggunakan matriks khusus), dan AddRoundKey. Pada ronde terakhir (ronde ke-Nr), semua transformasi dilakukan kecuali MixColumns (Fitriani & Utomo, 2020). Kunci ronde dihasilkan melalui proses ekspansi kunci yang penting untuk keamanan AES. Dalam mode operasi Electronic Codebook (ECB), data plaintext dibagi menjadi blok-blok 128-bit dan setiap blok dienkripsi secara terpisah dengan kunci yang sama, menghasilkan blok ciphertext yang identik untuk plaintext yang sama, membuat proses enkripsi ini bersifat deterministic (Simangunsong et al., 2022).

Proses enkripsi dilakukan pada data pasien sebelum data tersebut disimpan ke dalam database. Selama proses enkripsi, informasi sensitif seperti NIK, nama, alamat, data kelahiran, jenis kelamin, umur, nomor telepon, dan riwayat penyakit, keluhan, password, email diubah menjadi format yang hanya dapat dibaca dengan menggunakan kunci enkripsi yang benar. Dengan menerapkan pendekatan ini, penelitian ini bertujuan untuk mengurangi risiko terhadap kemungkinan bocornya data pasien. Kebocoran data termasuk dalam masalah serius yang dapat mengancam privasi pasien, dan dengan memasukkan lapisan keamanan ini dalam sistem database, upaya dilakukan untuk memberikan solusi praktis guna menjaga keamanan data pasien, memastikan privasi tetap terjaga, dan mengurangi potensi akses yang tidak sah.

2. Metode

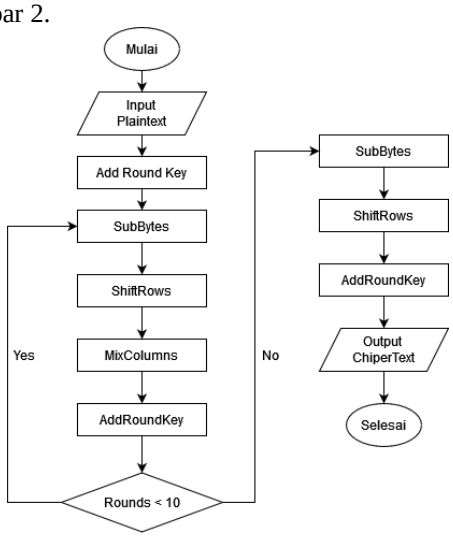
Alur kerja dalam penelitian ini meliputi beberapa langkah, berikut langkah-langkah ditunjukkan pada Gambar 1.



Gambar 1. Alur Kerja Penelitian

a. Studi Literatur: Tujuan dari studi literatur yang dilakukan untuk proyek ini adalah untuk mengumpulkan dan memeriksa sumber informasi terkait mengenai penggunaan algoritma Advanced Encryption Standard (AES-128) untuk mengamankan database sistem registrasi pasien.
b. Akuisisi Data: Penulis melakukan akuisisi data dengan menganalisis contoh atau kasus tertentu untuk memperoleh informasi yang diperlukan dalam implementasi Algoritma *Advanced Encryption Standard* (AES-128) untuk pengamanan database pada Sistem Registrasi Pasien.
c. Perancangan Sistem: Ini adalah proses merancang proses sistem. Perancangan ini menggunakan alat diagram alur, yang dibuat melalui pembuatan diagram.
d. Evaluasi: Evaluasi dilakukan dengan membandingkan hasil implementasi terhadap pengujian yang dilakukan.

Flowchart algoritma AES-128 ditunjukkan pada Gambar 2.



Gambar 2. Flowchart algoritma AES-128

Gambar 2 menjelaskan langkah-langkah enkripsi AES 128 bit dalam sistem sebagai berikut:

a. Sistem mengambil data plaintext dari input data.
b. Plaintext atau pesan masuk diproses dengan panjang tetap n, dan jika ukuran data terlalu panjang, data akan dipecah menjadi blok-blok yang lebih kecil.
c. Plaintext menjadi blok-blok data.
d. Plaintext menggunakan vektor inisialisasi untuk blok pertama dan melakukan XOR setiap blok dengan enkripsi blok sebelumnya.
e. Dilakukan enkripsi AES 128 bit dengan langkah-langkah sebagai berikut:
    1) AddRoundKey: Mengkombinasikan ciphertext dengan cipherkey menggunakan operasi XOR.
    2) SubBytes: Mengganti isi matriks dengan Rijndael S-Box.
    3) ShiftRows: Memindahkan atau menggeser setiap elemen blok atau tabel baris demi baris.
    4) Mix Column: Mengalikan setiap elemen cipher blok dengan matriks.
f. Proses AddRoundKey diulang sebanyak 10 putaran. Babak terakhir sedikit berbeda, di mana keadaan babak terakhir belum mengalami konversi MixColumns.
g. Periksa apakah semua blok telah dienkripsi. Jika tidak, proses kembali ke langkah d . Ketika semua blok selesai, ciphertext enkripsi AES 128-bit diambil.

3. Hasil dan Pembahasan

Implementasi Algoritma AES-128 untuk mengamankan data pasien dilakukan dengan mengubah data *plaintext* menjadi *ciphertext* yang aman. Proses algoritma ini berjalan pada sistem registrasi pasien, di mana hasil dari enkripsi data pasien akan disimpan dalam database dalam bentuk string heksadesimal.

Langkah – Langkah Implementasi AES-128:

a. Mendapatkan Data Pasien yang Diinputkan
   Data pasien yang diinputkan oleh user dan form registrasi disimpan dalam variabel seperti halnya NIK, nama, Alamat dan lain sebagainya.
b. Menyimpan Data tersebut dalam Variabel
   Data yang diinputkan oleh user disimpan dalam variabel untuk diproses lebih lanjut. Sebagai contoh “data_pasien” disimpan dalam variabel “$data”.
c. Memanggil Method Enkripsi
   Memanggil method *encrypt()* dan mem-passing variabel “$data” yang masih berbentuk *plaintext* ke dalam method ini untuk dienkripsi. Hasil enkripsi akan berupa string heksadesimal yang aman untuk disimpan di database
d. Memanggil Method Dekripsi
   Memanggil method *decryption()* dan mem-passing variabel “$data” yang sudah berbentuk *chipertext* ke dalam method ini untuk didekripsi. Hasil dekripsi akan berupa hasil data asli yang akan ditampilkan di dalam aplikasi si-periksa.

Proses enkripsi dilakukan untuk mengubah data asli (*plaintext*) menjadi data terenkripsi (*ciphertext*) menggunakan kunci enkripsi. Dalam proses enkripsi menggunakan AES-128, langkah awal yaitu mengubah *state* dan *key* ke bentuk hexadecimal dengan berpedoman Tabel ASCII yang ditampilkan pada Gambar 3. Hal ini berguna untuk menentukan initial round atau putaran pertama dalam proses enkripsi.

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(	72	48	110	H	104	68	150	h
9	9	11		41	29	51	)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[	123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	|
29	1D	35		61	3D	75	=	93	5D	135	]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

Gambar 3. Tabel ASCII

Pada Tabel 1 dan Tabel 2 menunjukkan hasil konversi *state* dan *key* ke dalam bentuk hexadesimal berpedoman tabel ASCII yang ditampilkan pada Gambar 3.

Tabel 1. *State*

State	A	R	I	A	N	D	O	H	A	R	E	D	I	S	O	N
Hexa	41	52	49	41	4E	44	4F	48	41	52	45	44	49	53	4F	4E

Tabel 2. *Chipper key*

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

Untuk mendapatkan hasil putaran pertama, caranya melakukan XOR antara *state* dengan *chipper key*. Berikut adalah tampilan hasil XOR antara *state* dengan *chipper key* ditunjukkan pada Tabel 3.

Tabel 3. Hasil XOR *state* dengan *chipper key*

State	41	52	49	41	4E	44	4F	48	41	52	45	44	49	53	4F	4E
Chiper key	4B	52	49	50	54	4F	47	52	41	46	49	41	45	53	4B	55
Hasil XOR	0A	0	0	11	1A	0B	8	1A	0	14	0C	5	0C	0	4	1B

Pembangkitan kunci (*key expansion*) adalah proses menghasilkan serangkaian kunci ronde dari kunci utama yang diberikan sebelumnya pada algoritma kriptografi (Widodo & Purnomo, 2020). Dalam konteks AES (*Advanced Encryption Standard*), pembangkitan kunci adalah tahap penting dalam proses enkripsi (Paramarta et al., 2018). Berikut hasil pembangkitan kunci atau *key expansion* ditampilkan pada Tabel 4 sampai Gambar 7.

Tabel 4. Key expansion round 0-10

Round ke-	Key expansion
0	4b52495054544f475241464945534b55
1	A7e1b53ef3aef26cb2e8bb2df7bbf078

2	4f6d0956bcc3fb3a0e2b4017f990b06f
3	2b8aa1cf97495af599621ae260f2aa8d
4	Aa26fc1f3d6fa6eaa40dbc08c4ff1685
5	Ac616b03910ecde9035371e1f1fc6764
6	3ce428a2adeae54b98e994aa6915f3ce
7	25e9a35b8803461010ead2ba79ff2174
8	B31431ed3b1777fd2bfda54752028433
9	Df4bf2ede45c8510cfa120579da3a464
10	E302b1b3075e34a3c8ff14f4555cb090

Untuk mendapatkan hasil pada Gambar 6 harus melalui beberapa cara seperti berikut :

a) *Round* 0 = *Chipper Key*

b) Kolom pertama setiap *round* dihasilkan dengan cara melakukan Rotword (Rotasi byte) pada kolom pertama round sebelumnya. Hasil rotword disubtitusi setiap byte-nya dengan berpedoman tabel S-Box (*Subbytes*). Setelah itu dilakukan XOR antara hasil *subbytes* dengan kolom pertama *round* sebelumnya dan juga dengan tabel RCON yang ditunjukkan pada Tabel 5.

Tabel 5. 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

c) Kolom kedua sampai kolom keempat dihasilkan dengan cara melakukan XOR antara kolom keempat dengan kolom pertama di *round* sebelumnya.

Proses enkripsi *round* 1:

a. *Subbytes*

Untuk *Round* 1 dihasilkan dengan cara hasil XOR antara *plaintext* dengan *chipper key* diubah senilai dengan berpedoman tabel S-Box. Sedangkan untuk *round* 2 sampai round 10 dihasilkan dengan cara merubah hasil proses enkripsi pada round sebelumnya senilai dengan berpedoman tabel S-Box yang ditunjukkan pada Gambar 4. Berikut merupakan contoh tahap *subbytes* pada *round* 1 yang ditunjukkan pada Gambar 5.

	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

Gambar 4. Tabel S-Box

11	13	04	02	15	6f	13	00	08	15	07	00	12	12	1f	1c
Tranformasi menggunakan tabel S-Box															
82	7d	f2	77	59	a8	7d	63	30	59	c5	63	c9	c9	c0	9c

Gambar 5. *Subbytes*

b. *Shiftrows*

Tahap *ShiftRows* pada AES melibatkan penggeseran byte pada setiap baris state. Dalam tahap ini Baris pertama tetap tidak mengalami penggeseran (geser ke kiri sebanyak 0 kali). Baris kedua mengalami penggeseran ke kiri sebanyak 1 kali. Baris ketiga mengalami penggeseran ke kiri sebanyak 2 kali. Baris keempat mengalami penggeseran ke liri sebanyak 3 kali. Contoh penerapan *shiftrows* seperti ditunjukkan pada Gambar 6.

82	7d	f2	77	59	a8	7d	63	30	59	c5	63	c9	c9	c0	9c
Pergeseran byte pada setiap baris *state*															
82	a8	c5	9c	59	59	c0	77	30	c9	f2	63	c9	7d	7d	63

Gambar 6. *Shiftrows*

Tahap ini hanya berlaku pada proses enkripsi *round* 1 sampai *round* 9.

c. *Mix Column*

Melakukan perkalian antara matrix yang sudah ditentukan dengan hasil *shiftrows*. Berikut merupakan proses *Shiftrows* ditunjukkan pada Gambar 7.

02	03	01	01		82	59	30	c9		b6	ge	b1	10
01	02	03	01	x	a8	59	c9	7d	=	4f	ff	22	5f
01	01	02	03		c5	c0	f2	7d		e4	99	a0	dd
03	01	01	02		9c	77	63	63		d3	9c	ad	9d

Gambar 7. Mix Column

d. *Add Round Key*

Melakukan XOR antara hasil *mixcolumn* dengan round yang sudah didapat pada proses pembangkitan kunci. Berikut proses *add round key* pada proses *round* 1 ditunjukkan pada Gambar 8.

b6	ge	b1	10		a7	f3	b2	f7		11	1d	03	e7
4f	ff	22	5f	⊕	e1	ae	e8	bb	=	4e	51	ca	e4
e4	99	a0	dd		b5	f2	bb	f0		51	6b	1b	2d
d3	9c	ad	9d		3e	6c	2d	78		ed	f0	80	e5

Gambar 8. *Add Round Key*

Pada proses *round* 2 sampai *round* 9 sama dengan *round* 1, sedangkan untuk *round* 10 proses XOR-nya dilakukan dengan hasil *shiftrows*.

Hasil akhir enkripsi :

Plaintext: ZAMRA TRISNAWATI

Chipper Key : KRIPTOGRAFIAESKU

Chipper Text :

1fff63449e4c49998b5ea2543abfc90f

Hasil enkripsi kemudian didekripsi, proses dekripsi merupakan proses pengembalian dari proses enkripsi. Berikut merupakan langkah – langkah dekripsi:

Proses *round* 0 :

Proses *round* 0 ini hasilnya akan menjadi block yang akan digunakan pada *round* 1. Proses ditunjukkan pada Gambar 9.

1f	ff	63	44		e3	07	c8	55		fc	f8	ab	11
9e	4c	49	99	⊕	02	5e	ff	5c	=	9c	12	b6	c5
8b	5e	a2	54		b1	34	14	b0		3a	6a	b6	e4
3a	bf	c9	0f		b3	a3	f4	90		89	1c	3d	9f

Gambar 9. *Round* 0 dekripsi

Proses *round* 1 :

1. Invers *Shiftrows*

fc	f8	ab	11
9c	12	b6	c5
3a	6a	b6	e4
89	1c	3d	9f

→

fc	f8	ab	11
12	b6	c5	9c
b6	e4	3a	6a
9f	89	1c	3d

Gambar 10. Invers *Shiftrows*

2. Invers *Subbytes*

fc	f8	ab	11
12	b6	c5	9c
b6	e4	3a	6a
9f	89	1c	3d

→

55	e1	0e	e3
39	79	07	1c
79	ae	a2	58
6e	f2	c4	8b

Gambar 11. Invers *Subbytes*

3. Invers *Add Round Key*

55	e1	0e	e3
39	79	07	1c
79	ae	a2	58
6e	f2	c4	8b

⊕

df	e4	cf	9d
4b	5c	a1	a3
f2	85	20	a4
ed	10	57	64

=

8a	05	c1	7e
72	25	a6	bf
8b	2b	82	fc
83	e2	93	ef

Gambar 12. Invers *Add Round Key*

4. Invers *Mix Column*

0e	0b	0d	09
09	0e	0b	0d
0d	09	0e	0b
0b	0d	09	0e

x

8a	05	c1	7e
72	25	a6	bf
8b	2b	82	fc
83	e2	93	ef

=

50	ee	64	9e
e6	ac	a5	dc
b5	7a	8a	76
49	d8	bb	ac

Gambar 13. Invers *Mix Column*

Tahap ini hanya berlaku pada proses dekripsi *round* 1 sampai *round* 9.

Pada proses *round* 2 sampai *round* 9 sama dengan *round* 1, sedangkan untuk *round* 10 proses XOR-nya dilakukan dengan hasil *shiftrows* tanpa proses *mix column*.

Hasil Akhir Dekripsi

Chipper Key : KRIPTOGRAFIAESKU

Chipper Text : 1fff63449e4c49998b5ea2543abfc90f

Plaintext Hexadesimal : 54414d524120545249534e4157415449

Plaintext String : ZAMRA TRISNAWATI

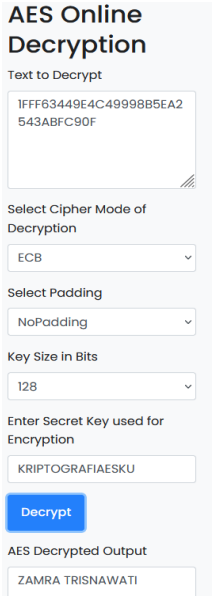
Proses enkripsi dan dekripsi algoritma AES-128 diatas akan diterapkan pada aplikasi resgitrasi pasien di Arif Merbabu Care. Proses implementasi dilakukan dengan memanfaatkan fungsi *openssl_encrypt()* dan *openssl_decrypt()* dari library *OpenSSL* pada class "Aes_encrypt". *Class* tersebut akan ditambahkan pada setiap proses data penting yang akan masuk pada database. Pada saat pasien sudah melakukan pendaftaran, data antrian yang muncul pada sistem seperti pada Gambar 14 di bawah :

Gambar 14. Tampilan data pasien pada aplikasi

Pada Gambar 14 menampilkan data pasien yang masih bisa terbaca secara jelas. Sedangkan pada database, data pasien sudah tidak terbaca secara jelas seperti pada Gambar 15 di bawah :

Gambar 15. Tampilan pada database system registrasi pasien

Pada Gambar 15 terdapat kolom nama yang terenkripsi "1fff63449e4c49998b5ea2543abfc90f". Ketika dilakukan dekripsi secara manual hasil dari dekripsi menunjukkan hasil hexadesimal "54414d524120545249534e4157415449" yang jika di convert ke dalam text biasa memiliki arti "ZAMRA TRISNAWATI". Kemudian jika kode enkripsi tersebut dimasukkan ke situs https://www.devglan.com/online-tools/aes-encryption-decryption#google_vignette untuk di dekripsi maka akan muncul hasil "ZAMRA TRISNAWATI". Hal ini menunjukkan bahwa enkripsi Algoritma *Advanced Encryption Standard-128* telah berhasil diterapkan dengan hasil pengujian secara manual maupun dengan cara dekripsi otomatis seperti Gambar 16.



Gambar 16. Gambar hasil pengujian

4. Kesimpulan

Hasil pengujian dan penerapan algoritma AES-128 pada aplikasi registrasi pasien "Si-Periksa" dilakukan perumusan kesimpulan bahwa bahasa pemrograman PHP dan paket OpenSSL digunakan untuk berhasil mengimplementasikan algoritma AES-128. Proses enkripsi dan dekripsi berjalan sesuai dengan yang diharapkan, dengan data yang terenkripsi dapat diubah kembali menjadi bentuk aslinya tanpa kehilangan informasi. Hasil dekripsi yang sesuai dengan data asli menegaskan bahwa implementasi algoritma ini telah dilakukan dengan benar dan efektif.

Daftar Pustaka:

Azanuddin, A., Yakub, S., & Prayudha, J. (2022). Implementasi Keamanan Citra Menggunakan Algoritma AES-128 Dengan Aplikasi Client-Server. *Jurasik (Jurnal Riset Sistem Informasi*

Dan Teknik Informatika), *7*(1), 51. https://doi.org/10.30645/jurasik.v7i1.415

CNN Indonesia. (2022). *Bjorka Bikin RI Raih Juara Ketiga Dunia Kebocoran Data*. CNN Indonesia. https://www.cnnindonesia.com/teknologi/20221115073902-192-873785/bjorka-bikin-ri-raih-juara-ketiga-dunia-kebocoran-data

CNN Indonesia. (2023). *4 Kasus Kebocoran Data di Semester I 2023, Mayoritas Dibantah*. CNN Indonesia. https://www.cnnindonesia.com/teknologi/20230720060802-192-975421/4-kasus-kebocoran-data-di-semester-i-2023-mayoritas-dibantah

Firdaus, I. (2022). Upaya Perlindungan Hukum Hak Privasi Terhadap Data Pribadi dari Kejahatan Peretasan. *Jurnal Rechten : Riset Hukum Dan Hak Asasi Manusia*, *4*(2), 23–31. https://doi.org/10.52005/rechten.v4i2.98

Fitriani, I., & Utomo, A. B. (2020). Implementasi Algoritma Advanced Encryption Standard (AES) pada Layanan SMS Desa. *JISKA (Jurnal Informatika Sunan Kalijaga)*, *5*(3), 153–163. https://doi.org/10.14421/jiska.2020.53-03

Kautsar, T. R. (2022). *Kajian Literatur Terstruktur Terhadap Kebocoran Data Pribadi dan Regulasi Perlindungan Data Pribadi*.

Paramarta, D. Q. P. A., Kusyanti, A., & Data, M. (2018). Implementasi Algoritme Advance Encryption Standard (AES) pada Enkripsi dan Dekripsi QR-Code. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer (J-PTIIK) Universitas Brawijaya*, *2*(12), 6729–6736. http://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/3732

Prabowo, W., Wibawa, S., & Azmi, F. (2020). Perlindungan Data Personal Siber di Indonesia. *Padjadjaran Journal of International Relations*, *1*(3), 218. https://doi.org/10.24198/padjir.v1i3.26194

Prameshwari, A., & Sastra, N. P. (2018). Implementasi Algoritma Advanced Encryption Standard (AES) 128 Untuk Enkripsi dan Dekripsi File Dokumen. *Eksplora Informatika*, *8*(1), 52. https://doi.org/10.30864/eksplora.v8i1.139

Pramudito, A. G., & Kusumaningsih, D. (2018). Implementasi Algoritma Aes 128 Dan Rc4 Untuk Pengamanan Email Pada Pt. Dinamika Hydro Engineering. *SKANIKA*, *1*(3), 869–876.

Purweni, M., Hariyadi, D., Nastiti, F. E., & Fazlurrahman, F. (2022). Model Inspeksi Keamanan Jaringan Nirkabel Dengan Teknik Wardrving Berbasis ChatBot. *Jurnal Komtika (Komputasi Dan Informatika)*, *6*(2), 83–90. https://doi.org/10.31603/komtika.v6i2.7943

Rahmawati, C. (2019). Tantangan dan Ancaman Keamanan Siber Indonesia di Era Revolusi Industri 4.0. *Seminar Nasional Sains Teknologi Dan Inovasi Indonesia (SENASTINDO AAU)*, *1*(1), 299–306. https://aau.e-journal.id/senastindo/article/view/116

Sadikin, R. (2021). *Kriptografi Untuk Keamanan Jaringan*. ANDI Yogyakarta.

Sholihin, H., Sari, H. L., & Aspriyono, H. (2022). Implementasi Kriptografi Klasik Untuk Pengamanan Database Berbasis Web. *Jurnal Media Infotama*, *18*(1), 87–93. https://doi.org/https://doi.org/10.37676/jmi.v18i1.2122

Simangunsong, H., Agung Raharja, M., & Raya Kampus Unud, J. (2022). Penerapan Algoritma Advanced Encryption Standard (AES-128) Dengan Mode ECB Dalam Pengamanan File. *Jurnal Nasional Teknologi Informasi Dan Aplikasinya*, *1*(1), 743–748.

Sitorus, F. A., Nugroho, N. B., & Pane, U. F. S. S. (2020). Implementasi Algoritma Advanced Encryption Standard (AES) 128 Bit untuk Keamanan Data Transaksi Penjualan Pada PT. Mitsubishi Electric Indonesia. *Jurnal CyberTech*, *x*, 1–15. https://ojs.trigunadharma.ac.id/

Widodo, B. E., & Purnomo, A. S. (2020). Implementasi Advanced Encryption Standard Pada Enkripsi Dan the Implementation of Advanced Encryption Standard on the Encryption and Decryption of the Confidential Documents At. *Jurnal Teknik Informatika (Jutif)*, *1*(2), 69–77.