J-SISKO TECH
Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD
P:ISSN : 2621-8976 E-ISSN : 2615-5133 ■ 138
Vol.3, No.2, Juli 2020, pp.138-150

IMPLEMENTASI SISTEM KEAMANAN DATABASE DATA PELANGGARAN HUKUM DISIPLIN PRAJURIT MENGGUNAKAN ALGORITMA ADVANCED ENCRYPTION STANDARD 128 BIT PADA PENGADILAN MILITER I-02 MEDAN

Kristovorus Zalukhu, Yohanni Syahra, Trinanda Syahputra
Program Studi Sistem Informasi, STMIK Triguna Dharma
Jl. A.H Nasution No.73 Medan, Sumatera Utara, 20142

Abstrak

Pengadilan Militer I-02 Medan memiliki masalah terkait pengamanan data. Khususnya database data pelanggaran hukum disiplin prajurit. Data pelanggaran tersebut ditransmisikan melalui jaringan komunikasi. Dari aspek keamanan menimbulkan kemungkinan adanya pihak yang tidak bertanggung jawab untuk memodifikasi data-data tersebut. Sehingga memerlukan sebuah metode yang disebut dengan kriptografi untuk melakukan pengamanan database data pelanggaran hukum disiplin prajurit.

Dari masalah tersebut diatas, pada penelitian ini akan menggunakan salah satu algoritma kriptografi yaitu: advanced encryption standard yang merupakan sebagai solusi dalam pengamanan database data pelanggaran hukum disiplin prajurit. Diharapkan dengan algoritma advanced encryption standard dapat diterapkan untuk menyelesaikan masalah diatas.

Hasil yang didapatkan dari penelitian ini adalah sebuah sistem terpadu yang mampu menyelesaikan masalah pada Pengadilan Militer I-02 Medan khususnya dalam hal pengamanan database data pelanggaran hukum disiplin prajurit. Diharapkan pengamanan database tersebut dapat ditingkatkan lagi seiring dengan perkembangan teknologi.
.

Kata kunci : Keamanan data Database advanced encryption standard

Abstract

Medan Military I-02 Court has problems related to data security. In particular the database of data on violation of soldier discipline laws. The violation data is transmitted through the communication network. From the security aspect, it raises the possibility of parties who are not responsible for modifying these data. So it requires a method called cryptography to secure data databases of violations of soldier discipline law.

From the aforementioned problems, this study will use one of the cryptographic algorithms, namely: advanced encryption standard which is a solution in securing data database violations of soldier discipline law. It is hoped that the advanced encryption standard algorithm can be applied to solve the above problems.

The results obtained from this research are an integrated system that is able to solve problems at the Military Court I-02 Medan, especially in terms of securing data databases on violations of soldier discipline law. It is hoped that the security of these databases can be improved in line with technological developments.

Keywords : Database data security advanced encryption standard

1. PENDAHULUAN

Pengadilan Militer I-02 Medan merupakan instansi pemerintahan yang menanggulangi krisis di bidang hukum bertujuan untuk pemisahan secara tegas aparatur penegak hukum dengan fungsi dan wewenang yang telah ditetapkan agar dapat tercapai profesionalitas, proporsionalitas, dan integritas yang utuh serta pemisahan yang tegas fungsi-fungsi yudikatif dan eksekutif, dengan mewujudkan hukum nasional melalui program legislasi nasional secara terpadu dan menegakkan supremasi hukum dalam kehidupan bermasyarakat, berbangsa dan bernegara.

Dalam instansi pemerintah memiliki data yang sangat penting dan perlu dilakukan pengamanan untuk menjaga kerahasiaan dan keakuratan data tersebut. Karena Kerahasiaan sebuah data merupakan aspek yang sangat penting untuk melindungi informasi supaya tidak mudah jatuh ke tangan pihak lain dan tidak merugikan pemilik informasi [1]. Selain itu instansi juga perlu melakukan perubahan sistem yang sebelumnya telah ada. Namun, perlu di tingkatkan kinerja dari sebuah sistem tersebut untuk mengikuti revolusi industri 4.0 yang terus berkembang secara dinamis.

Dalam melakukan pengamanan data dibutuhkan sebuah teknik yang disebut dengan kriptografi. Kriptografi merupakan teknik untuk mengubah atau menyamarkan sebuah informasi sehingga informasi tersebut ketika ditransmisikan menjadi tidak dikenali atau tidak berupa sebuah informasi [2]. Dalam kriptografi terdiri dari enkripsi dan dekripsi. Enkripsi adalah proses untuk pengubahan data menjadi data yang tidak berbentuk sebuah informasi dengan menggunakan algoritma tertentu. Dekripsi merupakan proses perubahan data yang telah di enkripsi ke dalam bentuk semula. Kriptografi banyak digunakan di dalam dunia digital karena tingkat keamanan datanya dapat mencegah *cyber-crime* menerobos sebuah sistem.

Algoritma *advanced encryption standard* ini akan digunakan penulis untuk melakukan pengamanan pada *record database* yang akan di lindungi. Dalam algoritma ini data yang akan dilindungi akan enkripsi, guna melindungi kerahasiaan dari data itu sendiri. Dalam algoritma ini *record database* akan di enkripsi sehingga bentuk *record database* nantinya tidak dapat dibaca karena teracak bukan seperti sebelum dilakukan enkripsi.

Pengadilan Militer I-02 Medan memiliki jaringan komputer untuk memperlancar arus informasi di dalam instansi. Jaringan tersebut memiliki sebuah server dan memiliki client kurang lebih 25. Informasi dari instansi tersebut di *share* melalui jaringan komputer. Namun kemudahan tersebut tidak memiliki pengamanan yang baik. Sehingga perlu dilakukan peningkatan keamanan data untuk menghindari hal-hal yang tidak diinginkan seperti perusakan data, manipulasi dan pencurian data khususnya pada *record database* data pelanggaran hukum disiplin prajurit. Untuk itu peneliti berupaya untuk mewujudkan implementasi sistem keamanan *database* menggunakan metode *advanced encryption standard* dalam skripsi yang berjudul **"Implementasi Sistem Keamanan Database Data Pelanggaran Hukum Disiplin Prajurit Menggunakan Algoritma Advanced Encryption Standard 128 Bit Pada Pengadilan Militer I-02 Medan"**.

1. METODE PENELITIAN

2.1 Kriptografi

Teknologi kriptografi sangat berperan penting dalam komponen komunikasi, untuk melakukan enkripsi/penyandian (pengacakan) data yang ditransmisikan selama pengiriman dari sumber ke tujuan dan juga melakukan dekripsi (pengembalian ke awal) data yang telah teracak tersebut [5]. Sebuah data perlu dilakukan pengamanan untuk menjaga kerahasiaan data dan informasi penting. Yang sering digunakan untuk melakukan pengamanan data adalah kriptografi. Kriptografi adalah salah satu teknik yang berkaitan dengan keamanan data seperti kerahasiaan sebuah data, integritas data, serta otentikasi [6].

2.2 Algoritma Advanced Encryption Standard

Advanced encryption standard dikerjakan dalam mode penyandian blok (*block cipher*) diproses dengan *block* data 128-bit dengan panjang kunci 128-bit, 192-bit, dan 256-bit. Algoritma kriptografi yang digunakan dalam penyandian *block* pada *advanced encryption standard* meliputi *Electronic Code Book* (ECB), *Cipher Block Chaining* (CBC), *Cipher Feedback* (CFB), dan *Output Feedback* (OFB). ECB, CBC, CFB, dan OFB memiliki tingkat keamanan data

yang berbeda-beda dan memiliki kelebihan dan kekurangan dari masing-masing penyandian *block* tersebut. Algoritma kriptografi *advanced encryption standard* awalnya diterbitkan sebagai Rijndael yang dikembangkan oleh Vincent Rijmen dan John Daemen asal Belgia yang keluar sebagai pemenang kontes algoritma kriptografi pengganti DES pada 26 November 2001 yang diselenggarakan oleh NIST (*National Institutes of Standards and Technology*) milik pemerintah Amerika Serikat. Algoritma Rijndael inilah yang kemudian dikenal dengan *advanced encryption standard* (AES). Setelah mengalami proses standarisasi selama 5 tahun, Rijndael kemudian disahkan pada 22 Mei 2002 menjadi standard algoritma kriptografi secara resmi. Pada tahun 2006, kriptografi simetrik menjadi algoritma terpopuler dan *advanced encryption standard* banyak digunakan dalam pengamanan data hingga saat ini [10].

2.2.1 Proses Ekspansi Kunci AES 128 bit

Pada tahap *AddRoundKey*, dilakukan proses pembangkitan kunci yang dilakukan berulang sebanyak Nr. Untuk mendapatkan ekspansi kunci diperlukan Nb(Nr+1) word [13]. agar bisa menggunakan *advanced encryption standard* 128 bit maka 4(10+1) = 40 *word* = 44 x 32 bit = 1408-bit *subkey*. Ekspansi kunci dari 128 menjadi 1408-bit *subkey*. Proses ini biasanya disebut dengan *key schedule*. *Subkey* ini diperlukan karena setiap round merupakan suatu nilai inisial dari Nb word untuk Nr = 0 dan 2 untuk Nr = 1,3 untuk Nr = 2…., yang berisi *array linier* empat *byte* word (w), 0=1 Nb (Nr + 1).

2.2.2 Proses Enkripsi AES 128 bit

Enkripsi adalah mengubah pesan menjadi pesan yang tidak mudah di ketahui oleh pihak lain dalam bentuk sandi [14]. Algoritma *advanced encryption standard* memiliki panjang kunci 128-bit terdiri dari *AddRoundKey*, *SubBytes*, *ShiftRows* dan *MixColumms*. Berikut adalah langkah-langkah enkripsi *advanced encryption standard* 128-bit:

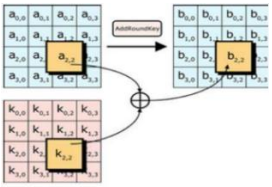
1. *Key Schedule*

Key schedule dilakukan untuk mendapatkan *subkey* dari kunci utama untuk melakukan proses enkripsi dan dekripsi. Operasi ini terdiri dari:

a. *Rotate*, merupakan proses perputaran 8-bit menjadi 32-bit kunci.
b. *SubBytes*, merupakan proses substitusi dari 8 bit *subkey* dengan nilai dari S-Box.
c. *Rcon*, operasi ini menggunakan nilai dalam *galois field* kemudian di XORkan dengan hasil operasi *subbytes* sesuai dengan nilai yang diinginkan user yang dipangkatkan 2.
d. Operasi pada XOR dengan w[i-Nk] yaitu word yang berada pada Nk sebelumnya.

2. *AddRoundKey*

Pada tahap ini dilakukan kombinasi *chipertext* yang sudah ada dengan *chiperkey* yang dihubungkan dengan XOR.



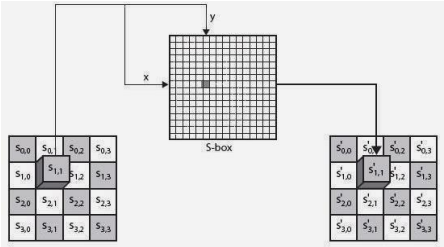
Gambar 1. Proses *AddRoundKey* [6]

3. *SubBytes*

Pada tahap ini dilakukan penukaran isi matriks yang ada dengan matriks lain yang disebut dengan S-Box.

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

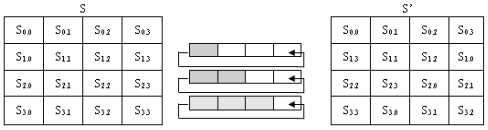
Gambar 2. S-Box [6].



Gambar 3. Proses *subbytes* [6].

4. *ShiftRows*

Pada tahap ini dilakukan pergeseran tiap baris dari tabel *state*. Pada baris pertama tidak dilakukan pergeseran, pada baris kedua dilakukan pergeseran 1-*byte*, pada baris ketiga dilakukan pergeseran 2-*byte* dan pada baris keempat dilakukan pergeseran 3-*byte*.



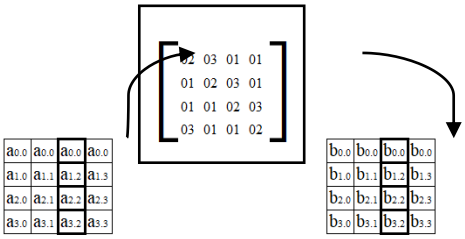
Gambar 4. Proses *Shift Rows* [6].

5. *MixColumns*

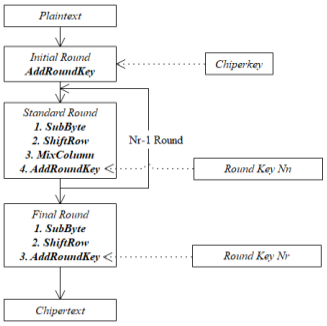
Proses ini dilakukan perkalian tiap elemen dari *block chipper* dengan matriks. Pengalian dilakukan seperti perkalian biasa menggunakan *dot product* kemudian perkalian keduanya dimasukkan ke dalam sebuah *block chipper* baru. Persamaan *mixcolumns* dapat dilihat pada gambar dibawah ini.

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} x \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Gambar 5. Persamaan *mixcolumns.*



Gambar 6. Proses Perkalian *mixcolumns.*



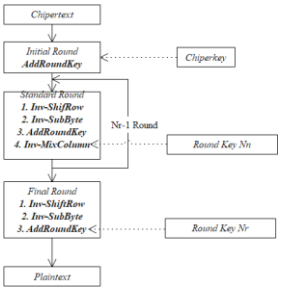
Gambar 7. Proses enkripsi *advanced encryption standard* [13].

Waktu yang dibutuhkan dalam mengubah *plaintext* menjadi *ciphertext*. Ukuran kunci menjadi faktor utama dalam waktu enkripsi. Waktu enkripsi berdampak pada kinerja enkripsi. Waktu enkripsi yang lebih sedikit adalah bukti algoritma yang efektif dan efisien [15].

2.2.3 Proses Dekripsi AES 128 bit

Dekripsi merupakan suatu proses pengembalian data ke bentuk semula yang telah terenkripsi (*chipertext*) dan dapat dibaca kembali seperti *plaintext*. Transformasi pada dekripsi sama dengan proses pada transformasi enkripsi. Namun, pada transformasi dekripsi dilakukan proses *InvSubBytes*, pada tahap ini state ditukar nilainya dengan table Invers S-Box. Pada *InvShiftRows* dilakukan pergeseran ke kanan 3 baris terakhir pada state dan selanjutnya adalah

InvMixColumns, pada tahap ini dilakukan perkalian matriks yang telah ada dengan state hasil *AddRoundKey*.

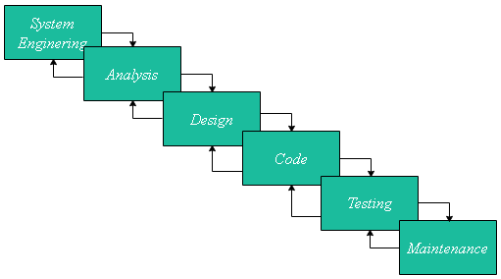


Gambar 8. Dekripsi proses *advanced encryption standard* [13].

Waktu untuk mengembalikan *chipertext* ke *plaintext* disebut waktu dekripsi. Waktu dekripsi diharapkan mirip dengan waktu enkripsi untuk membuat algoritma responsif dan cepat. Waktu dekripsi berdampak pada kinerja sistem [15].

2.3 Metode Perancangan Sistem

Dalam perancangan sistem dibutuhkan sebuah metode yang digunakan dalam perancangan perangkat lunak. Metode yang peneliti gunakan dalam penyusunan ini adalah algoritma *waterfall*.



Gambar 9. *Waterfall Model*

2.4 Algoritma Sistem

Algoritma *advanced encryption standard* merupakan satu dari banyak algoritma kriptografi. Algoritma ini menggunakan 4 tahapan dalam proses penyandian data, antara lain *SubBytes, ShiftRows, MixColumns* dan *AddRoundKey*. Proses penyandian dilakukan berulang atau sering disebut putaran. Banyaknya putaran algoritma *advanced encryption standard* dengan panjang kunci 128 bit adalah 10 putaran. *Plaintext* yang dienkripsi akan dimasukkan kedalam *state* 4 x 4.

3.3.1 Ekspansi Kunci

Ekspansi kunci digunakan untuk proses *AddRoundKey*. Jumlah kunci algoritma *advanced encryption standard* 128 bit yang dibutuhkan dalam ekspansi kunci adalah 10 kunci. Kunci yang digunakan pada kasus ini adalah “PERADILANMILITER”. Berikut adalah proses ekspansi kunci *advanced encryption standard*:

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

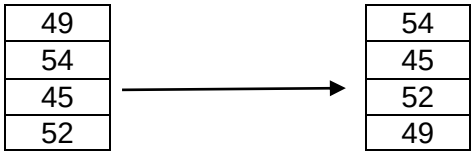
P	E	R	A	D	I	L	A	N	M	I	L	I	T	E	R
50	45	52	41	44	49	4C	41	4E	4D	49	4C	49	54	45	52

2. Proses selanjutnya adalah mengubah kunci ke dalam *state* 4 x 4 seperti berikut:

50	44	4E	49
45	49	4D	54
52	4C	49	45
41	41	4C	52

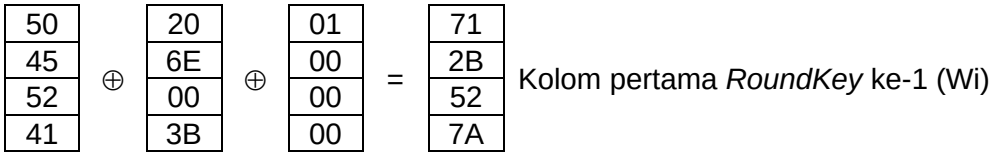
3. Kemudian melakukan fungsi *RotWord* untuk menghasilkan kunci ke-1, yaitu dengan menggeser setiap bit pada kolom 4 ke atas 1 kali yang dimulai dari baris ke 2 menggunakan kunci putaran ke-0.

⟶ *RoundKey* ke-0

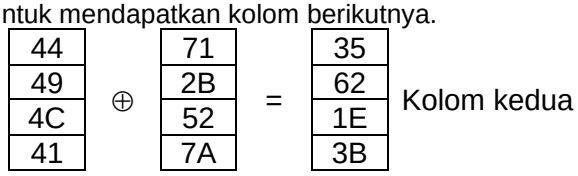


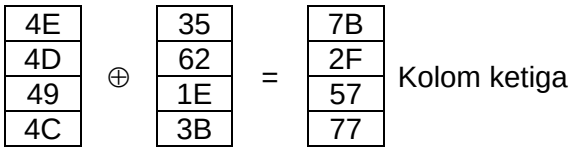
4. Setelah itu, melakukan substitusi hasil dari *RotWord* dengan nilai yang ada pada tabel *S-Box* (*SubBytes*).

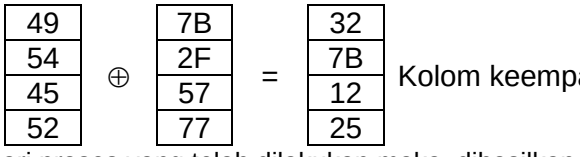
5. Selanjutnya, untuk mendapatkan kolom pertama dari *RoundKey* ke-1 adalah proses XOR antara kolom pertama dari *RoundKey* ke-0 dan hasil dari *SubBytes* di XOR-kan dengan *Rcon*.



6. Untuk mendapatkan nilai kolom selanjutnya dilakukan XOR antara kolom pertama (Wi) dengan kolom kedua dari *RoundKey* ke-0, setelah itu lakukan proses seperti kolom kedua untuk mendapatkan kolom berikutnya.







7. Dari proses yang telah dilakukan maka, dihasilkan *RoundKey* ke-1, yaitu:

71	35	7B	32
2B	62	2F	7B
52	1E	57	12
7A	3B	77	25
7A	3B	77	25

Untuk mendapatkan *RoundKey* ke-2 sampai *RoundKey* ke-10 maka, proses diatas diulang sebanyak 10 kali. Kunci dari setiap *round* akan digunakan untuk proses enkripsi dan dekripsi, berikut hasil seluruh ekspansi *key*:

71	35	7B	32
2B	62	2F	7B
52	1E	57	12
7A	3B	77	25

RoundKey ke-1

52	67	1C	2E
E2	80	AF	D4
6D	73	24	36
59	62	15	30

RoundKey ke-2

…..

49	6A	94	51
D5	8B	F2	7F
CD	19	72	30
BF	16	7F	79

RoundKey ke-10

3.3.2 Enkripsi

Proses enkripsi akan dilakukan pada *record database* data pelanggaran hukum disiplin prajurit Pengadilan Militer I-02 Medan. *Plaintext* yang dienkripsi adalah “PANJI TRIYANTORO” berikut ini adalah proses enkripsinya:

1. *Plaintext* diurutkan kedalam blok dan diubah kedalam bentuk bilangan *hexadecimal*.

P	A	N	J	I		T	R	I	Y	A	N	T	O	R	O
50	41	4E	4A	49	20	54	52	49	59	41	4E	54	4F	52	4F

2. *Plaintext* yang diubah ke *hexadecimal* yang telah disusun 16 *byte* pertama dibentuk kedalam *state* 4 x 4.

50	49	49	54
41	20	59	4F
4E	54	41	52
4A	52	4E	4F

3. Selanjutnya proses *AddRoundKey*, pada proses ini XOR-kan *plaintext* dengan *RoundKey* ke-0.

50	49	49	54
41	20	59	4F
4E	54	41	52
4A	52	4E	4F

⊕

50	44	4E	49
45	49	4D	54
52	4C	49	45
41	41	4C	52

=

00	0D	07	1D
04	69	14	1B
1C	18	08	17
0B	13	02	1D

4. Hasil dari *AddRoundKey* diatas akan menjadi *round* ke-1 untuk diproses dengan 4 transformasi, yaitu *SubBytes*, *ShiftRows*, *MixColumns* dan *AddRoundKey*.

Round ke-1

a. Transformasi pertama yaitu *SubBytes*, pada tahap ini setiap *byte* akan ditukar dengan nilai pada tabel *S-Box*.

00	0D	07	1D
04	69	14	1B
1C	18	08	17
0B	13	02	1D

SubBytes ⟶

63	D7	C5	A4
F2	F9	FA	AF
9C	AD	30	F0
2B	7D	77	A4

b. Proses selanjutnya adalah *ShiftRows*, pada baris pertama tidak dilakukan pergeseran, baris kedua dilakukan pergeseran 1 *byte* ke kiri, pada baris ketiga digeser 2 *byte* ke kiri dan baris keempat digeser 3 *byte* ke kiri, berikut adalah penjelasannya:

		63	D7	C5	A4
	F2	F9	FA	AF	
9C	AD	30	F0		
2B	7D	77	A4		

=

63	D7	C5	A4
F9	FA	AF	F2
30	F0	9C	AD
A4	2B	7D	77

c. Setelah itu melakukan proses *MixColumns*, proses ini dilakukan perkalian antara *polynomial* tetap dengan *state* hasil dari *ShiftRows*.

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

x

63	D7	C5	A4
F9	FA	AF	F2
30	F0	9C	AD
A4	2B	7D	77

Aturan dalam operator *polynomial* adalah jika dikali 01 maka hasilnya tetap, jika dikali 02 maka *bitshift* 1x ke kiri jika MSB = 0 dan *bitshift* 1x ke kiri diikuti operasi XOR dengan 1B (0001 1011) jika MSB = 1, dan jika dikali 03 maka dilakukan operasi dikali 02 dan XOR dengan bilangan *hexadecimal* pada hasil bilangan *ShiftRows* itu sendiri. Berikut adalah uraian perkalian antara *polynomial* dengan hasil *ShiftRows*.

d. Tahap terakhir dari *round* ke-1 adalah *AddRoundKey*, hasil dari *MixColumns* akan di XOR-kan dengan *RoundKey* ke-1, berikut adalah prosesnya:

42	7B	9A	84
7E	18	42	C0
0D	AB	CE	8E
3F	3E	9D	46

⊕

71	35	7B	32
2B	62	2F	7B
52	1E	57	12
7A	3B	77	25

=

33	4E	E1	B6
55	7A	6D	BB
5F	B5	99	9C
45	05	EA	63

Proses diatas akan diulangi untuk *round* ke-2 sampai dengan *round* ke-10. Namun, pada *round* ke 10 transformasi *MixColumns* tidak lagi dilakukan. Berikut hasil transformasi proses enkripsi *round* ke-2 sampai dengan *round* ke-10:

Round ke-2

SubBytes

C	2F	F8	4E
F	DA	3C	EA
C	D5	EE	DE
6	6B	87	FB

ShiftRows

C3	2F	F8	4E
DA	3C	EA	FC
EE	DE	CF	D5
FB	6E	6B	87

MixColumns

FD	AA	6A	D1
BE	40	16	4E
C8	06	2A	91
87	4F	E0	EE

RoundKey ke-2

5	67	1C	2E
E	80	AF	D4
6	73	24	36
5	62	15	30

AddRoundKey

AF	CD	76	FF
5C	C0	B9	9A
A5	75	0E	A7
DE	2D	F5	DE

……..

Round ke-10

SubBytes

8	EC	11	67
1	82	6C	EE
F	D0	5E	BC
4	2D	B1	A0

ShiftRows

8C	EC	11	67
82	6C	EE	17
5E	BC	FE	D0
A0	49	2D	B1

RoundKey ke-10

49	6A	94	51
D5	8B	F2	7F
CD	19	72	30
BF	16	7F	79

AddRoundKey

C	86	85	36
5	E7	1C	68
9	A5	8C	E0
1	5F	52	C8

Hasil dari proses enkripsi yaitu: C557931F86E7A55F851C8C523668E0C8

3.3.3 Dekripsi

Dekripsi merupakan proses untuk mengembalikan *record* yang telah dienkripsi menjadi *plaintext* kembali. Transformasi deskripsi pada algoritma *advanced encryption standard* adalah *InvSubBytes*, *InvShiftRows*, *InvMixColumns*, dan *AddRoundKey*. Berikut adalah proses dekripsi *chipertext* "C557931F86E7A55F851C8C523668E0C8":

1. Melakukan proses XOR antara *chipertext* dengan *RoundKey* ke-10.

C5	86	85	36
57	E7	1C	68
93	A5	8C	E0
1F	5F	52	C8

49	6A	94	51
D5	8B	F2	7F
CD	19	72	30
BF	16	7F	79

=

8C	EC	11	67
82	6C	EE	17
5E	BC	FE	D0
A0	49	2D	B1

2. Selanjutnya, Pada *round* ke-1 sampai *round* ke-9 proses dekripsi dilakukan transformasi *InvShiftRows*, *InvSubBytes*, *InvMixColumns* dan *AddRoundKey*.

Round ke-1

InvShiftRows

8C	EC	11	67	
	82	6C	EE	17
		5E	BC	FE	D0	
			A0	49	2D	B1

=

8C	EC	11	67
17	82	6C	EE
FE	D0	5E	BC
49	2D	B1	A0

3. Kemudian, lakukan proses *InvSubBytes*. Untuk S-Box *InvSubBytes* berbeda dengan S-BOX *SubBytes* karena telah dilakukan *invers* namun, cara kerjanya sama.

8C	EC	11	67	*InvSubBytes* →	F0	83	E3	0A
17	82	6C	EE		87	11	B8	99

FE	D0	5E	BC		0C	60	9D	78
49	2D	B1	A0		A4	FA	56	47

4. Setelah itu, lakukan operasi XOR antara *InvSubBytes* dengan *RoundKey* ke-9 untuk transformasi *AddRoundKey*.

F0	83	E3	0A	⊕	22	23	FE	C5	=	D2	A0	1D	CF
87	11	B8	99		F9	5E	79	8D		7E	4F	C1	14
0C	60	9D	78		A2	D4	6B	42		AE	B4	F6	3A
A4	FA	56	47		19	A9	69	06		BD	53	3F	41

5. Selanjutnya, melakukan proses transformasi antara hasil *AddRoundKey* dengan aturan *irreducible polynomial*.

0E	0B	0D	09	x	D2	A0	1D	CF
09	0E	0B	0D		7E	4F	C1	14
0D	09	0E	0B		AE	B4	F6	3A
0B	0D	09	0E		BD	53	3F	41

Proses perhitungan dilakukan dengan mengubah bilangan *hexadecimal* ke bilangan biner, kemudian diubah kedalam bilangan *polynomial* Proses diatas akan diulangi untuk *round* ke-2 sampai dengan *round* ke-10, hasil transformasi proses dekripsi *round* ke-2 sampai dengan *round* ke-10 dapat dilihat dibawah ini:

Round ke-2

InvShiftRows

4A	7D	82	4C
D0	D4	9A	C7
43	E2	53	79
96	13	DE	72

InvSubBytes

5C	13	11	5D
60	19	37	31
64	3B	50	AF
35	82	9C	1E

RoundKey ke-8

86	01	DD	3B
5C	A7	27	F4
0A	76	BF	29
FB	B0	C0	6F

AddRoundKey

DA	12	CC	66
3C	BE	10	C5
6E	4D	EF	86
CE	32	5C	71

InvMixColumns

AB	27	D9	54
3D	54	E4	67
93	C5	24	1C
43	65	76	7B

……..

Round ke-10

InvShiftRows

63	D7	C5	A4
F2	F9	FA	AF
9C	AD	30	F0
2B	7D	77	A4

InvSubBytes

00	0D	07	1D
04	69	14	1B
1C	18	08	17
0B	13	02	1D

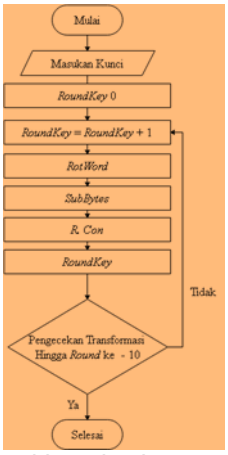
RoundKey ke-0

50	44	4E	49
45	49	4D	54
52	4C	49	45
41	41	4C	52

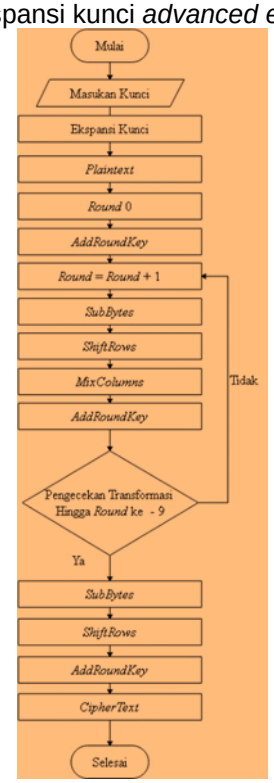
AddRoundKey

50	49	49	54
41	20	59	4F
4E	54	41	52
4A	52	4E	4F

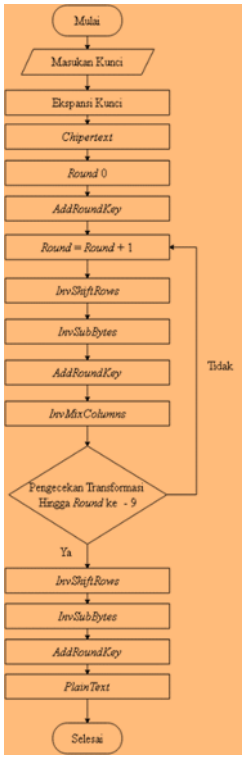
Dari hasil proses diatas didapatkan hasil “50 41 4E 4A 49 20 54 52 49 59 41 4E 54 4F 52 4F”. Jika dikonversikan ke karakter didapatkan *plaintext* “PANJI TRIYANTORO”.



Gambar 10. *Flowchart* ekspansi kunci *advanced encryption standard* 128 bit.



Gambar 11. *Flowchart* enkripsi *advanced encryption standard* 128 bit.



Gambar 12. *Flowchart* dekripsi *advanced encryption standard* 128 bit

1. Kelebihan Sistem
   a. Sistem yang dibangun dapat diakses secara *offline* dan *online* melalui jaringan komputer berbasis *server*.
   b. Keamanan terhadap *database* yang dienkripsi sangat tinggi karena menggunakan algoritma *advanced encryption standard* dengan panjang kunci 128 bit.
   c. Sistem yang dibangun memiliki *user interface* yang baik.
2. Kelemahan Sistem
   a. Sistem menggunakan algoritma *advanced encryption standard* dengan panjang kunci 128 bit.
   b. Sistem hanya bisa digunakan oleh pimpinan dan IT, Pelaporan dan Perencanaan serta pihak lain yang diberi wewenang untuk mengakses sistem.

2. KESIMPULAN

6.1 Kesimpulan

Berdasarkan uraian dari bab-bab sebelumnya, maka dapat diambil kesimpulan dari penelitian ini sebagai berikut:

1. Berdasarkan hasil penerapan algoritma *advanced encryption standard*. Maka, algoritma tersebut dapat diterapkan dalam penyelesaian masalah pada pengamanan *database* data pelanggaran hukum disiplin prajurit.
2. Berdasarkan hasil rancangan, maka Sistem Keamanan *database* data pelanggaran hukum disiplin prajurit dapat menjadi solusi dalam mengamankan *record database*.
3. Berdasarkan hasil implementasi, sistem yang dibangun dapat membantu instansi Pengadilan Militer I-02 Medan dalam mengamankan *database* data pelanggaran hukum disiplin prajurit menggunakan algoritma *advanced encryption standard*.

DAFTAR PUSTAKA

[1] E. D. Saragih, N. A. Hasibuan, and E. Bu'ulolo, "Implemantasi Algoritma Triple DES Dan Algoritma Advanced Encryption Standard dalam Penyandian File," *Inti*, vol. 13, no. September, pp. 263–269, 2018.

[2] L. A. Indrayani and I. M. Suartana, “Implementasi Kriptografi dengan Modifikasi Algoritma Advanced Encryption Standard (AES) untuk Pengamanan File Document,” *Jinacs*, vol. 01, no. November 2001, pp. 42–47, 2019.

[3] I. Suryanto, C. Suhery, and Y. Brianorman, “Pengembangan Aplikasi Chat Messenger dengan Metode Advanced Encryption Standard (AES) pada Smartphone,” *J. Coding Sist. Komput. Untan*, vol. 03, no. 2, pp. 1–10, 2017.

[4] D. Novianto and Y. Setiawan, “Aplikasi Pengamanan Informasi Menggunakan Metode Least Significant Bit (Lsb) dan Algoritma Kriptografi Advanced Encryption Standard (AES),” vol. 09, no. 2, pp. 83–89, 2018.

[5] S. Waluyo, “Sistem Keamanana Management File Menggunakan Algoritma Advanced Encryption Standard (AES-128) Studi Kasus : Tabitha Indonesia,” pp. 639–644, 2018.

[6] D. Nurnaningsih and A. A. Permana, “Rancangan Aplikasi Pengamanan Data Dengan Algoritma Advanced Encyption Standard (Aes),” *J. Tek. Inform.*, vol. 11, no. 2, pp. 177–186, 2018.

[7] A. Pudoli and D. Kusumaningsih, “Penggunaan Hybrid Cryptosystem Untuk Enkripsi dan Dekripsi Pesan Messanger Menggunaka Algoritma Rivest Shamir Adleman (RSA) dan Advanced Encryption Standard (AES) dengan Firebase pada Android,” vol. 9, no. 3, pp. 125–131, 2017.

[8] P. Pradinata and M. Syafrullah, “Keamanan Algoritma Kriptografi Database menggunakan Metode Advanced Encryption Standard (AES-128) Berbasis Desktop,” *Skanika*, vol. 1, no. 2, pp. 732–738, 2018.

[9] Y. Yanti, “Teknik Pengamanan File Dokumen Berbasis Text Menggunakan Metode Advanced Encryption Standard (AES),” *Semin. Nas. II USM 2017 Eksplor. Kekayaan Marit. Aceh di Era Glob. dalam Mewujudkan Indones. sebagai Poros Marit. Dunia*, vol. 1, pp. 87–90, 2017.

[10] J. Prayudha, “Implementasi Keamanan Data Gaji Karyawan Pada PT . Capella Medan Menggunakan Metode Advanced Encryption Standard (AES),” vol. 18, no. 2, 2019.

[11] W. K. Hadi and S. M. M. Kom, “Pengamanan Aplikasi Chatting Pada Perangkat Android Menggunakan Kriptografi Dengan Metode Advanced Encryption Standard (Aes) 128 Pada Pt . Salam Medina Indonesia Issn : 1693-9166,” vol. 14, no. 2, pp. 62–69, 2017.

[12] R. J. A. Gentra Muchammad Akbar, Ichsan Taufik, “Implementasi algoritma Advanced Encryption Standard (AES) 128-bit pada aplikasi sharing dokumen berbasis android,” *Insight*, vol. 1, no. 1, pp. 110–115, 2018.

[13] J. Putri and A. Vatresia, “Pada Citra Digital Menggunakan Advanced Encyption Standard 128 dan Least Significant bit -1,” vol. 7, no. 2, 2019.

[14] T. Erlangga, D. Kusumaningsih, F. T. Informasi, U. B. Luhur, P. Utara, and K. Lama, “Implementasi Algoritma Advanced Encryption Standard -128 (Aes-128) Untuk Pengamanan Database,” vol. 1, no. 2, pp. 565–569, 2018.

[15] R. Perangin-angin, I. K. Jaya, B. Rumahorbo, and B. J. R. Marpaung, “Analisa Alokasi Memori dan Kecepatan Kriptograpi Simetris Dalam Enkripsi dan Dekripsi,” vol. 4, no. 1, 2019.

[16] D. Rubiyanto, D. Diaty, and Allwar, “Crude clove bud oil (CBO) quality improvement by bentonite adsorption process in flow system,” *AIP Conf. Proc.*, vol. 1823, no. 1, pp. 24–44, 2017.

[17] Suendri, “Implementasi Diagram UML (Unified Modelling Language) Pada Perancangan Sistem Informasi Remunerasi Dosen Dengan Database Oracle (Studi Kasus: UIN Sumatera Utara Medan),” *J. Ilmu Komput. dan Inform.*, vol. 3, no. 1, pp. 1–9, 2018.

[18] S. Rosa A and M. Shalahuddin., *Rekayasa Perangkat Lunak*, Ed.Rev. Bandung: Informatika Bandung, 2018.

[19] D. W. T. Putra and R. Andriani, “Unified Modelling Language (UML) dalam Perancangan SistemInformasi Permohonan Pembayaran Restitusi SPPD,” *J. TEKNOIF (Teknik Inform.*, vol. 7, no. 1, pp. 32–39, 2019.

[20] E. F. Wati and A. A. Kusumo, “Penerapan Metode Unified Modeling Language (UML) Berbasis Desktop Pada Sistem Pengolahan Kas Kecil Studi Kasus Pada PT Indo Mada Yasa Tangerang,” *J. Inform.*, vol. 5, no. 1, pp. 24–36, 2016.

[21] M. D. Irawan, “Implementasi Kriptografi Vigenere Cipher Dengan Php,” *J. Teknol. Inf.*, vol. 1, no. 1, p. 11, 2017.

[22] “Simbol Flowchart - Pengertian, Fungsi, Tujuan, Jenis, Contoh.” [Online]. Available: https://www.dosenpendidikan.co.id/simbol-flowchart/. [Accessed: 13-Jan-2020].

[23] Y. Trimarsiah and M. Arafat, “Analisis dan Perancangan Website Sebagai Sarana,” *Anal. dan Peranc. Website Sebagai Sarana*, pp. 1–10, 2017.

[24] M. Fungsi, I. Pada, and M. Kuliah, “Aplikasi website berbasis HTML dan JavaScript untuk menyelesaikan fungsi integral pada mata kuliah kalkulus,” *J. Inov. Teknol. Pendidik.*, vol. 6, no. 1, pp. 80–91, 2019.

[25] E. Orlando, “Aplikasi Pengajuan Cuti Pada Human Resource Management Menggunakan PHP dan MYSQL (Studi Kasus Pada PT. INTILOKA),” *J. Ilm. KOMPUTASI*, vol. 16, no. 3, pp. 275–284, 2017.