Jurnal CyberTech
Vol.x. No.x, 201x, pp. xx~xx
P-ISSN:
E-ISSN :

Penerapan Algoritma AES 128 Bit Untuk Keamanan Data Peminjaman Senjata Api Pada DENPOM I/5 Medan

Yoga Arif Wibowo [*], Nurcahyo Budi Nugroho, S.Kom., M.Kom [**], Beni Andika, S.E., S.Kom., M.Kom [**]

[*] Program Studi Sistem Informasi, STMIK Triguna Dharma
[**] Program Studi Sistem Informasi, STMIK Triguna Dharma

Article Info
Article history:

Keyword:
Kriptografi,AES,Advanced Encryption Standart, Data Peminjaman, Pengamanan Data

Untuk Data adalah catatan atas kumpulan fakta. Dalam penggunaan sehari-hari data berarti suatu pernyataan yang diterima secara apa adanya. Pernyataan ini adalah hasil pengukuran atau pengamatan suatu variabel yang bentuknya dapat berupa angka, kata-kata, atau citra. Data peminjaman merupakan salah satu data yang bersifat rahasia yang hanya dapat dilihat oleh pihak-pihak tertentu yang di beri wewenang dan tanggung jawab.

Detasemen Polisi Militer I/5 Medan adalah salah satu Instansi Pemerintah yang harus menjaga data peminjaman senjata api agar tidak disalah gunakan atau dimanipulasi oleh orang-orang yang tidak bertanggung jawab dan tentu akan menimbulkan kerugian besar bagi Instansi.

Dalam hal ini diperlukan sebuah sistem dalam pengamanan data yang dapat melakukan penyandian dan pengacakan sebuah informasi yang berbasis komputer. Pengamanan ini dilakukan dengan menerapkan sebuah algoritma kriptografi yang bertujuan untuk mengenkripsi dan dekripsi sebuah pesan text. Algoritma kriptografi yang digunakan adalah algoritma AES (Advanced Encryption Standard). Hasil pengujian menunjukkan bahwa sistem keamanan data peminjaman senjata api dapat mengamankan data peminjaman dengan baik dan menghindari terjadinya penyalahgunaan atau manipulasi data oleh orang-orang yang tidak memiliki wewenang atas data tersebut.

ABSTRACT

Corresponding Author:
Nama : Yoga Arif Wibowo
Kampus : STMIK Triguna Dharma
Program Studi : Sistem Informasi
E-Mail : Yogakeren62@gmail.com

1. PENDAHULUAN

Keamanan menjadi aspek yang sangat penting saat ini dimana pertukaran data informasi menjadi tuntutan baik pekerjaan dan lainnya. Berbagai cara dilakukan untuk mengamankan data atau informasi diantaranya menggunakan Kriptografi. Beragam macam teknik digunakan untuk upaya mengamankan data atau informasi yang penting. [1]

Salah satu solusi pengamanan data yang digunakan adalah kriptografi. Kriptografi adalah ilmu dan seni untuk menjaga kerahasiaan pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi. Berdasarkan analisa keamanan data, metode Advance Encryption Standard lebih aman karena cipherteks tidak dapat dipecahkan dengan metode *Brute Force Attack*. Hal ini terjadi karena metode Advance Encryption Standard menggunakan bit/round sehingga lebih aman.

Keamanan data merupakan hal yang sangat penting dalam menjaga kerahasiaan informasi terutama informasi sensitif yang hanya boleh diketahui oleh pihak yang berhak saja. Informasi yang merupakan hasil pengolahan dari data, mempunyai nilai yang berbeda bagi setiap orang. Seringkali sebuah informasi menjadi sangat berharga dan tidak semua orang diperkenankan untuk mengetahuinya, namun selalu saja ada pihak yang berusaha untuk mengetahui informasi dengan cara-cara yang tidak semestinya bahkan bermaksud untuk merusaknya. Kesenjangan. [2]

Hasil yang di peroleh setelah melakukan beberapa percobaan dan Penerapan enkripsi dan dekripsi menggunakan AES 128 bit, maka diperoleh hasil bahwa pesan asli (plainteks) yang dienkripsi menggunakan Rijndael dapat terenkripsi dengan baik, hal ini terbukti dari pesan yang dihasilkan tidak dapat di baca oleh pengguna, kemudian setelah pesan tersebut di dekripsi, maka akan kembali seperti plainteks dan dapat di baca pesan tersebut. Selain itu untuk kapasitas dan waktu proses enkripsi dan dekripsi dimana diperoleh hasil bahwa pesan setelah dilakukan enkripsi akan lebih besar dari segi kapasitas file sedangkan untuk kecepatan enkripsi membutuhkan waktu 18 detik.

Algoritma Rijndael inilah yang kemudian dikenal dengan *Advanced Encryption Standard* (AES). Setelah mengalami beberapa proses standarisasi oleh NIST, setelah Rijndael diadopsi menjadi standard algoritma kriptografi secara resmi pada 22 Mei 2002. Pada 2006, AES merupakan salah satu algoritma terpopuler yang digunakan dalam kriptografi kunci simetrik.[3]

Berdasarkan latar belakang diatas keamanan data peminjaman senjata api yang menggunakanmetode AES 128 bit telah di uraikan, maka dibuatlah penalitian (skripsi) dengan judul **"Penerapan Algoritma AES 128 bit Untuk Keamanan Data Peminjaman Senjata Api Pada DENPOM I/5 Medan".**

2. KAJIAN PUSTAKA

2.1 Data Peminjaman

Data adalah catatan atas kumpulan fakta. Dalam penggunaan sehari-hari data berarti suatu pernyataan yang diterima secara apa adanya. Pernyataan ini adalah hasil pengukuran atau pengamatan suatu variabel yang bentuknya dapat berupa angka, kata-kata, atau citra.

2.2 Kriptografi

Kriptografi merupakan salah satu teknik untuk menjamin kerahasiaan informasi yang dikomunikasikan. Informasi ini terlindung karena pesan asli akan diubah menjadi pesan cipher (pesan sandi) dengan menggunakan kunci tertentu sehingga pesan ini tidak dapat diketahui pihak yang tidak berkepentingan. Seiring dengan perkembangannya kriptografi ternyata dapat dimanfaatkan untuk mendukung aspek kemanan informasi lainnya. Aspek keamanan informasi yang dapat didukung oleh kriptografi adalah kerahasiaan.

2.3 *Advanced Encryption Standart*

AES (*Advanced Encryption Standart*) merupakan sistem penyandian blok yang berkarakter non-Faistel, karna AES memakai komponen yang slelau mempunyai invers dengan panjang blok 128, 192 dan 256 bit. Penyandian AES menggunakan proses yang iteratif atau disebut juga ronde.

Pada tahun 90-an, setelah beberapa tahun standart penyandian simetris DES (*Data Encryption Standart*) dianggap tidak aman lagi, lembaga standart Amerika Serikat National Institute of Standart and Technology (NIST) membuat sayembara untuk menggantikan DES dengan sebuah sistem penyandian *Advanced Encryption Standart* pada tanggal 12 September 1997. Kemudian NIST memberi beberapa spesifikasi untuk AES, yakni memiliki panjang blok 128 bit, serta mampu men support panjang kunci 128, 192 dan 256

2.3.1 Proses Ekspansi Kunci

Proses Ekspansi Kunci Algoritma AES mengambil kunci *cipher* dan melakukan rutin ekspansi kunci untuk membentuk *key schedule*. Ekspansi kunci menghasilkan total Nb (Nr+1) word. Algoritma ini membutuhkan set awal *key* yang terdiri dari Nb *word*, dan setiap *RoundKey* Nr membutuhkan data kunci sebanyak Nb *word*. Hasil *key schedule* terdiri dari array 4 byte word linear yang dinotasikan dengan [wi].

2.3.2 Proses Enkripsi

Proses enkripsi di dalam algoritma AES terdiri dari 4 jenis Transformasi byte, yaitu *SubBytes*, *Shiftrows*, *MixColumn*, dan Add*RoundKey*. Pada awal proses enkripsi, masukkan yang telah disalin ke dalam state mengalami Transformasi byte Add*RoundKe*. Setelah itu, state akan mengalami Transformasi *SubBytes*, *Shiftrows*, *MixColumns*, dan Add*RoundKey* secara berulang-ulang sebanyak Nr. Proses ini dalam algoritma AES disebut sebagai *RoundKey* function. Sedangkan *RoundKey* yang terakhir state tidak menjalani Transformasi *MixColumns*.

Langkah kerja enkripsi adalah sebagai berikut:

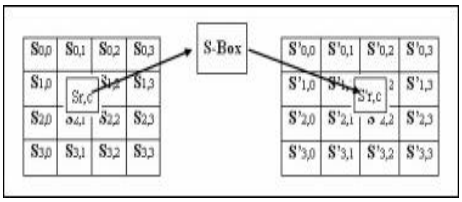
a. Transformasi *SubBytes*

SubBytes merupakan Transformasi *byte* dimana setiap elemen pada state akan dipetakan dengan menggunakan sebuah tabel substitusi (S-Box). Tabel substitusi S-Box akan dipaparkan dalam Gambar 1.

x \ y	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 1 Tabel Subtitusi Untuk Transformasi *SubBytes*

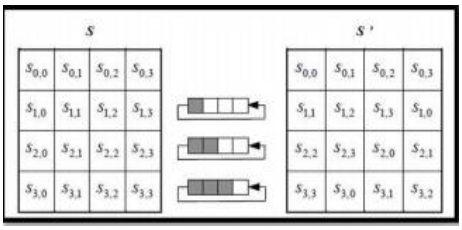
Untuk setiap byte pada array state, misalkan S[r, c] = xy, yang dalam hal ini xy adalah digit heksadesimal dari nilai S[r, c], maka nilai substitusinya, dinyatakan dengan S'[r, c], adalah elemen di dalam tabel subtitusi yang merupakan perpotongan baris x dengan kolom y. Gambar 2 mengilustrasikan pengaruh pemetaan byte pada setiap byte dalam state.



Gambar 2 Pengaruh Pemetaan pada Setiap *Byte* dalam *State*

b. *Shiftrows*

Transformasi *Shiftrows* pada dasarnya adalah proses pergeseran bit dimana bit paling kiri akan dipindahkan menjadi bit paling kanan (rotasi bit). Proses pergeseran *Shiftrows* ditunjukkan dalam Gambar 3 berikut:



Gambar 3 Transformasi *Shiftrows*

c. *MixColumns*

MixColumn mengoperasikan setiap elemen yang berada dalam satu kolom pada *state*. Secara lebih jelas, Transformasi *MixColumns* dapat dilihat pada perkalian matriks berikut ini:

$$\begin{bmatrix} S'_{0,C} \\ S'_{1,C} \\ S'_{2,C} \\ S'_{3,C} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,C} \\ S_{1,C} \\ S_{2,C} \\ S_{3,C} \end{bmatrix}$$

Gambar 4 Persamaan Transformasi *MixColumns*

Keluaran dari hasil perkalian matriks diatas bisa dianggap seperti perkalian pada gambar berikut:

$$
\begin{aligned}
s'_{0,c} &= (\{02\} \bullet s_{o,c}) \oplus (\{03\} \bullet s_{1,c}) \oplus s_{2,c} \oplus s_{3,c} \\
s'_{1,c} &= s_{0,c} \oplus (\{02\} \bullet s_{1,c}) \oplus (\{03\} \bullet s_{2,c}) \oplus s_{3,c} \\
s'_{2,c} &= s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{2,c}) \oplus (\{03\} \bullet s_{3,c}) \\
s'_{3,c} &= (\{03\} \bullet s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (\{02\} \bullet s_{3,c})
\end{aligned}
$$

Gambar 5 Perkalian matriks *MixColumns*

d. *AddRoundKey Key*

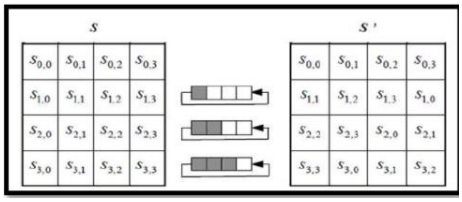
Add*RoundKey Key* melakukan XOR antara state sekarang dengan *RoundKey* key.

2.3.3 Proses Dekripsi

Transformasi *cipher* dapat dibalikkan dalam arah yang untuk menghasilkan inverse cipher yang mudah dimengerti untuk algoritma AES. Transformasi *byte* yang digunakan pada *invers cipher* adalah *InvShiftrows*, Inv*SubBytes*, *InvMixColumns*, dan *AddRoundKey*.

a. *InvShiftrows*

Transformasi *invers* terhadap *Shiftrows* disebut *InvShiftrows* yang merupakan Transformasi *byte* yang berkebalikan dengan Transformasi *SubRows*. Ilustrasi transformasi *InvShiftrows* terdapat pada Gambar 8 sebagai berikut



Gambar 6 Transformasi *InvShiftrows*

b. *InvSubBytes*

InvSubBytes juga merupakan transformasi *byte* yang berkebalikan dengan Transformasi *SubBytes*. Pada *InvSubByte* tiap elemen pada state dipetakan dengan menggunakan tabel *Inverse* S-Box

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

Gambar 7 Tabel Subtitusi Untuk Transformasi *InvSubBytes*

c. *InvMixColumns*

Setiap kolom dalam state dikalikan dengan matrik perkalian dalam AES. Perkalian dalam matrik dapat ditulis dengan persamaan seperti pada Gambar 10 sebagai berikut

$$
\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}
$$

Gambar 8 Persamaan Matrik InvMixColums.

d. *InvAddRoundKey*

Transformasi *Inverse AddRoundKey Key* tidak berbeda dengan Transformasi *AddRoundKey Key* karna dalam Transformasi ini hanya dilakukan operasi penambahan sederhana dengan operasi *bitwise* XOR

3. ANALISA DAN HASIL

3.1 Algoritma Sistem

Advance Encryption Standard merupakan satu dari banyak algoritma kriptografi. Advance Encryption Standard menggunakan 4 tahapan dalam proses penyandian data, antara lain adalah *SubBytes, ShiftRows, MixColumns* dan *AddRoundKey*. Proses penyandian dilakukan berulang atau bisa disebut dengan ronde. Banyaknya ronde pada Advance Encryption Standard dengan panjang kunci 128 bit adalah 10 ronde. Plaintext yang akan disandikan akan diurutkan dan dimasukkan ke dalam state 4 x 4. Plaintext yang telah dimasukkan ke dalam state akan diproses 4 kali transformasi dan akan di XOR-kan dengan masing-masing kunci yang berbeda setiap rondenya (*RoundKey*).

3.1.1 Proses Enkripsi AES

Dalam proses enkripsi algoritma AES, ada dua tahapan yaitu ekspansi kunci dan enkripsi

1. Proses Ekspansi Kunci

Kunci ronde *(RoundKey key*) dibutuhkan untuk proses enkripsi dan dekripsi pada algoritma Advanced Encryption Standard. Maksimal panjang kunci adalah sebanyak 16 digit dan jumlah kunci ronde yang diperlukankan adalah 10 kunci yang akan diperoleh dari proses ekspansi kunci. Pada kasus ini, kunci yang akan digunakan yaitu "yoga arif wibowo".

a. Urutkan kunci kedalam blok berukuran 128 bit (16 kode ASCII). Lalu ubah kunci kedalam bentuk heksadecimal

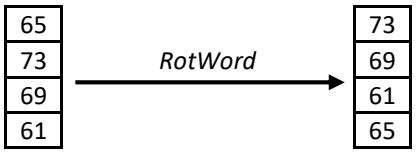
y	o	g	a		a	r	i	f		w	i	b	o	w	o
79	6F	67	61	20	61	72	69	66	20	77	69	62	6F	77	6F

b. Langkah selanjutnya yaitu susun kunci ke dalam state berukuran 4 × 4 seperti dibawah ini :

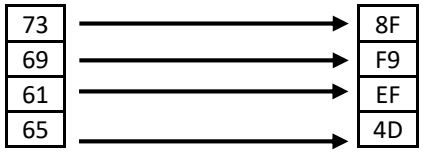
RoundKey Key Ke-0

54	72	69	67
75	6E	61	49
6E	64	6F	6E
65	73	69	61

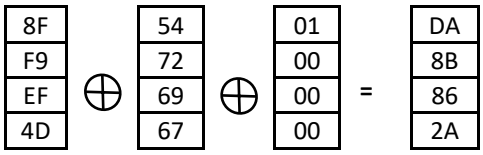
c. Kemudian, lakukan fungsi RotWord, yaitu dengan menggeser setiap bit pada kolom ke-4 ke atas 1 kali dari kunci ronde ke-0.



d. Hasil dari RotWord disubtitusikan dengan nilai pada tabel S-Box (*SubBytes*).



e. Tahap akhir yaitu lakukan proses XOR antara kolom pertama dari kunci ronde ke-0, hasil dari *SubBytes* lalu di-XOR-kan lagi dengan RCon.



f. Untuk mendapatkan kolom kedua, diperoleh dengan proses XOR antara Wi dengan kolom kedua dari kunci ronde ke-0. Sedangkan untuk mendapatkan kolom ketiga dan keempat kunci ronde ke-1, dilakukan proses seperti memperoleh kolom kedua.

Kolom ke-2 Kolom ke-3 Kolom ke-4

DA		75		AF	AF		6E		C1	C1		65		A4
8B	⊕	6E	=	E5	E5	⊕	64	=	81	81	⊕	73	=	F2
86		61		E7	E7		6F		88	88		69		E1
2A		49		63	63		6E		0D	0D		61		6C

g. Dari seluruh proses diatas, maka diperoleh kunci untuk ronde ke-1 yaitu :

DA	AF	C1	A4
8B	E5	81	F2
86	E7	88	E1
2A	63	0D	6C

Untuk mendapatkan kunci ronde ke-2 sampai ke-10, proses diatas diulang 10 kali. Dibawah ini adalah hasil ekpansi kunci hingga ronde ke 10:

RoundKey Key Ke-1 *RoundKey Key* Ke-2 *RoundKey Key* Ke-10

DA	AF	C1	A4	51	FE	3F	9B		8C	72	4D	D6
8B	E5	81	F2	73	96	17	E5		19	8F	98	7D
86	E7	88	E1	D6	31	B9	58	.....	39	08	B1	E9
2A	63	D	6C	63	0	D	61		77	77	7A	1B

2. Proses Enkripsi

Plaintext yang akan digunakan yaitu “Suhendra_0655776”. Kemudian urutkan kedalam blok lalu ubah kedalam bilangan heksadesimal.

S	u	h	e	n	d	r	a	_	0	6	5	5	7	7	6
53	75	68	65	6E	64	72	61	5F	30	36	35	35	37	37	36

Susun 16 byte pertama dari *plaintext* yang telah diubah kedalam state 4x4:

44	4B	64	46
2E	3C	3C	20
3C	73	5F	20
A2	50	46	20

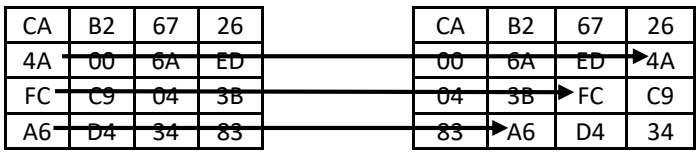
Lakukan XOR antara *plaintext* dengan *RoundKey Key* 0. Proses ini dinamakan Add*RoundKey* Key.

44	4B	64	46		54	75	6E	65		10	3E	A	23
2E	3C	3C	20		72	6E	64	73		5C	52	58	53
3C	73	5F	20	⊕	69	61	6F	69	=	55	12	30	49
A2	50	46	20		67	49	6E	61		C5	19	28	41

Proses Add*RoundKey Key* diatas masih sebagai pra-*RoundKey* dan akan menjadi masukan untuk ronde ke1 yang akan diproses dengan 4 Transformasi yaitu *SubBytes*, *Shiftrows*, *MixColumns* dan Add*RoundKey* Key. Hasil dari pra-*RoundKey* disubtitusikan dengan nilai pada tabel S-Box (*SubBytes*).

10	3E	0A	23	→	CA	B2	67	26
5C	52	58	53	→	4A	00	6A	ED
55	12	30	49	→	FC	C9	04	3B
C5	19	28	41	→	A6	D4	34	83

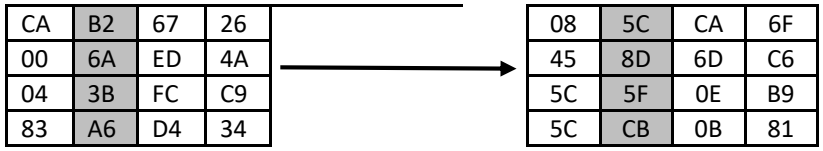
1. Lakukan *Shiftrows* pada hasil dari subtitusi *SubBytes* yang dieksekusi lewat pergeseran siklik secara memutar dengan geseran yang acak pada tiga baris terakhir state (baris pertama, r = 0, tidak digeser). Baris ke dua digeser secara siklik ke kiri sekali, baris ke tiga dua kali, dan baris ke empat tiga kali.



CA	B2	67	26
4A	00	6A	ED
FC	C9	04	3B
A6	D4	34	83

CA	B2	67	26
00	6A	ED	4A
04	3B	FC	C9
83	A6	D4	34

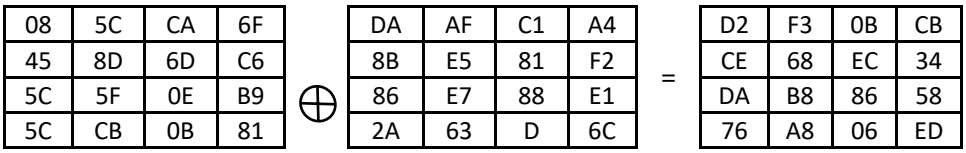
2. Transformasi *MixColumns* dengan mengoperasikan state kolom demi kolom pada state kolom, dengan mengkoversikan setiap kolom sebagai polinomial.



CA	B2	67	26
00	6A	ED	4A
04	3B	FC	C9
83	A6	D4	34

08	5C	CA	6F
45	8D	6D	C6
5C	5F	0E	B9
5C	CB	0B	81

3. Langkah terakhir untuk mendapatkan enkripsi putaran pertama, lakukan XOR antara hasil *MixColumns* dengan *RoundKey Key* Ke-1, proses ini disebut *AddRoundKey Key*.



08	5C	CA	6F
45	8D	6D	C6
5C	5F	0E	B9
5C	CB	0B	81

⊕

DA	AF	C1	A4
8B	E5	81	F2
86	E7	88	E1
2A	63	D	6C

=

D2	F3	0B	CB
CE	68	EC	34
DA	B8	86	58
76	A8	06	ED

Lakukan proses diatas sampai 10 putaran (*RoundKey*). Berikut adalah hasil enkripsi hingga *RoundKey* ke 10:

RoundKey Ke-1

D2	F3	0B	CB
CE	68	EC	34
DA	B8	86	58
76	A8	06	ED

RoundKey Ke-2

FE	FF	D4	20
D5	9A	37	2C
51	6E	79	A5
0C	C3	A0	79

...

RoundKey Ke-10

6E	D6	7D	47
62	C2	DD	9A
B9	B9	DD	42
A7	74	EE	5E

Dan hasil dari enkripsi AES menghasilkan *cipherteks* “A — h : Î [] i × E g Õ @ [] [] _ ‰”.

3.1.2 Proses Dekripsi AES

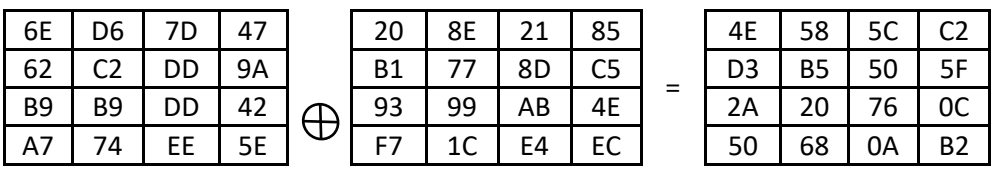
Proses-proses Transformasi pada dekripsi dalam metode *Advanced Encryption Standart* yaitu *InvSubBytes, InvShiftrows, InvMixColumns dan AddRoundKey Key.* Add*RoundKey Key* merupakan Transformasi yang bersifat self-invers. Kunci yang digunakan sama dengan yang digunakan pada proses enkripsi. Berikut adalah proses dekripsi dari hasil *ciphertext* yang telah diperoleh dari proses enkripsi sebelumnya.

A	—	h	:	Î	[]	i	×	E	g	Õ	@	[]	[]	_	‰
41	97	68	3A	CE	11	69	D7	45	67	D5	40	B	7	5F	89

Kemudian susun 16 byte pertama dari *ciphertext* yang telah diubah ke bentuk heksadesimal kedalam state 4x4:

6E	D6	7D	47
62	C2	DD	9A
B9	B9	DD	42
A7	74	EE	5E

Lakukan XOR antara *cipherteks* dengan *RoundKey Key* Ke-10. Proses ini dinamakan *AddInvRoundKey Key*.



6E	D6	7D	47
62	C2	DD	9A
B9	B9	DD	42
A7	74	EE	5E

⊕

20	8E	21	85
B1	77	8D	C5
93	99	AB	4E
F7	1C	E4	EC

=

4E	58	5C	C2
D3	B5	50	5F
2A	20	76	0C
50	68	0A	B2

1. Lakukan *InvShiftrows* pada hasil initial-*RoundKey* dari AddInv*RoundKey Key* yang dieksekusi lewat pergeseran siklik secara memutar. Baris ke dua digeser secara siklik ke kiri tiga kali, baris ke tiga dua kali, baris ke empat sekali.

4E	58	5C	C2
D3	B5	50	5F
2A	20	76	0C
50	68	0A	B2

→

4E	58	5C	C2
5F	D3	B5	50
76	0C	2A	20
68	0A	B2	50

2. Hasil dari *InvShiftrows* disubtitusikan dengan nilai pada tabel 〖S-Box〗^(-1) (Inv*SubBytes*).

4E	58	5C	C2
5F	D3	B5	50
76	0C	2A	20
68	0A	B2	50

→

B6	5E	A7	A8
84	A9	D2	6C
0F	81	95	54
F7	A3	3E	6C

3. XOR kan hasil dari *InvSubBytes* dengan *RoundKey Key* Ke-9. Proses ini disebut *AddInvRoundKey Key*.

B6	5E	A7	A8
84	A9	D2	6C
0F	81	95	54
F7	A3	3E	6C

⊕

44	AE	AF	A4
68	C6	FA	48
A3	0A	32	E5
BE	EB	F8	08

=

F2	F0	08	0C
EC	6F	28	24
AC	8B	A7	B1
49	48	C6	64

4. Hasil dari *AddInvRoundKey Key* diTransformasi kan oleh *InvMixColumns* dengan mengoperasikan state kolom demi kolom. Operasi ini dilakukan pada state kolom, dengan mengkoversikan setiap kolom sebagai polinomial.

F2	F0	08	0C
EC	6F	28	24
AC	8B	A7	B1
49	48	C6	64

→

CF	A8	BD	CA
AC	AA	C7	5A
7C	6B	90	96
E4	35	AB	FB

Proses diatas diulang sampai 10 kali putaran (*RoundKey*). Berikut adalah hasil dari dekripsi hingga *RoundKey* ke 10:

RoundKey Ke-1

CF	A8	BD	CA
AC	AA	C7	5A
7C	6B	90	96
E4	35	AB	FB

RoundKey Ke-2

A4	B8	53	8D
11	08	71	CC
F5	2D	0B	6C
A2	DB	F2	39

....

RoundKey Ke-10

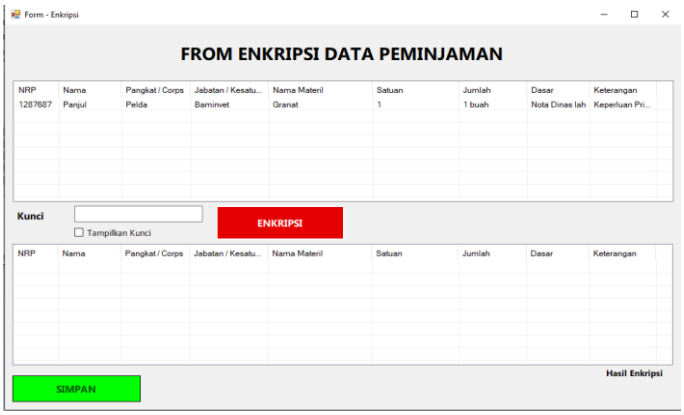
44	4B	64	46
2E	3C	3C	20
3C	73	5F	20
A2	50	46	20

Dan hasil dekripsi dari AES menghasilkan *plaintext* "S u h e n d r a _ 0 6 5 5 7 7 6^".

4. PENGUJIAN DAN IMPLEMENTASI

4.1 *Form* Enkripsi

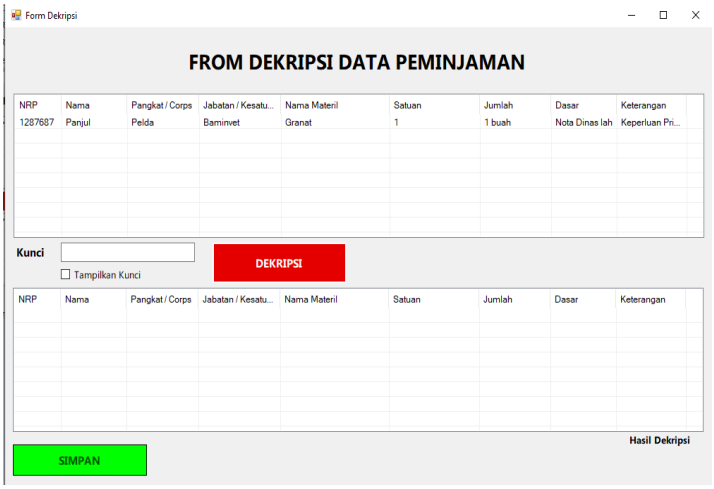
Berikut ini merupakan tampilan dari *Form* enkripsi yang berfungsi untuk melakukan proses enkripsi data:



Gambar 9 Tampilan *Form* Enkripsi

4.2 *Form* Dekripsi

Berikut ini merupakan tampilan dari *Form* dekripsi yang berfungsi untuk melakukan proses dekripsi data:



Gambar 10 Tampilan *Form* Dekripsi

5. KESIMPULAN

Berdasarakan perumusan dan pembahasan bab-bab sebelumnya dapat diambil beberapa kesimpulan dan beberapa saran.

1. Dalam mengatasi masalah yang terjadi pada Detasemen Polisi Militer I/5 Medan untuk pengamanan data peminjaman senjata api yaitu dengan melihat begitu pentingnya data peminjaman senjata api di Detasemen Polisi Militer I/5 Medan sehingga data tersebut harus dirahasikan dengan menggunakan algoritma AES (*Advanced Encryption Standart*).
2. Dalam merancang aplikasi menggunakan algoritma AES (*Advanced Encryption Standart*) yang dapat digunakan dalam pengamanan data peminjaman senjata api di DENPOM (Detasemen Polisi Militer) I/5 Medan, yaitu dengan membuat pemodelan sistem seperti *use case* diagram, *activity* diagram dan *class* diagram, kemudian membuat *flowchart* dari algortima sistem, selanjutnya membangun *database* untuk menampung dan menyimpan data, terakhir melakukan pengkodingan dengan pemrograman VB.Net.
3. Sistem yang telah dirancang selanjutnya diuji dan diimplementasikan dengan memasukkan data-data sampel sesuai dengan yang ada pada bab-bab sebelumnya, kemudaian jika hasil *output*nya sesuai dengan data manual maka dalam pengujian ini sistem berjalan dengan baik.

UCAPAN TERIMA KASIH

Puji syukur kehadirat Allah SWT atas izin-Nya yang telah melimpahkan rahmat dan karunia-Nya sehingga dapat menyelesaikan jurnal ilmiah ini. Pada kesempatan ini diucapkan terima kasih yang sebesar-besarnya kepada kedua Orang Tua tercinta yang selama ini memberikan do'a dan dorongan baik secara moril

maupun materi sehingga dapat terselesaikan pendidikan dari tingkat dasar sampai bangku perkuliahan dan terselesaikannya jurnal ini.

REFERENSI

1. F. N. Pabokory, I. F. Astuti, and A. H. Kridalaksana, "Implementasi Kriptografi Pengamanan Data Pada Pesan Teks, Isi File Dokumen, Dan File Dokumen Menggunakan Algoritma Advanced Encryption Standard," In*Form*. Mulawarman J. Ilm. Ilmu Komput., vol. 10, no. 1, p. 20, 2016.
2. E. R. Agustina and A. Kurniati, "Pemanfaatan Kriptografi dalam Mewujudkan Keamanan Informasi pada e-Voting di Indonesia," Agustina, Esti Rahmawati Kurniati, Agus, vol. 2009, no. semnasIF, pp. 22–28, 2009.
3. A. F. Marisman and A. Hidayati, "Pembangunan Aplikasi Pembanding Kriptografi Den-Gan Caesar Cipher Dan Advance Encryption Standard (AES) Untuk File Teks," pp. 213–222, 2015..
4. A. P. Widyassari, "Aplikasi Sistem Pendukung Keputusan Penilaian Kinerja Karyawan untuk Kenaikan Gaji pada PT AAA," Intensif, vol. 1, no. 2, pp. 92–101, 2017..
5. Ilhamsyah, "Jurnal Coding Sistem Komputer Untan Jurnal Coding Sistem Komputer Untan ISSN : 2338-493X," vol. 05, no. 1, pp. 68–79, 2017.
6. R. Sadikin, Kriptografi untuk Keamanan Jaringan dan Implementasinya dalam Bahasa Java, I. ANDI Yogyakarta, 2018.
7. A. Arif and P. Mandarani, "Rekayasa Perangkat Lunak Kriptografi Menggunakan Algoritma Advanced Encryption Standard (AES) 128 Bit Pada Sistem Keamanan Short Message Service (SMS) Berbasis Android," Teknoif, vol. 4, no. 1, pp. 1–10, 2016..

BIOGRAFI PENULIS

	**Yoga Arif Wibowo,** Laki – laki kelahiran Payakumbuh, 30 April 1996, anak pertama dari empat bersaudara ini merupakan seorang mahasiswa STMIK Triguna Dharma yang sedang dalam proses menyelesaikan skripsi.
	**Nurcahyo Budi Nugroho, S.Kom., M.Kom**, Beliau merupakan dosen tetap STMIK Triguna Dharma Medan dan aktif sebagai pengajar pada bidang ilmu Sistem Komputer.
	**Beni Andika, ST., S.Kom.,M.Kom**, Beliau merupakan dosen tetap STMIK Triguna Dharma Medan dan aktif sebagai pengajar pada bidang ilmu Sistem Informasi.