JIKOMSI [Jurnal Ilmu Komputer dan Sistem Informasi] E-ISSN : 2723-6129
Vol.4 No.2 September 2021, pp 75-85

Implementasi Metode *Advanced Encryption Standard* (AES 128 Bit) Untuk Mengamankan Data Keuangan

Niolinda Cristy[1], Fristi Riandari[2]

[a] Program Studi Teknik Informatika, STMIK Pelita Nusantara, Jl. St. Iskandar Muda No. 1 Medan
Email : [1]niolindacristy03@gmail.com, [2]fristy.rianda@ymail.com

INFORMASI ARTIKEL

Kata Kunci:
AES
Dekipsi
Enkripsi
Kriptgrafi
Keamanan Data

ABSTRAK

Masalah keamanan data dan informasi merupakan salah satu aspek penting dari sebuah informasi computer. Salah satu contoh masalah keamanan data yaitu kemanan data uang SPP disekolah. Data uang SPP merupakan kumpulan data yang berifat sensitive bagi pihak sekolah. Data yang ada di dalamnya berupa rangkuman atau catatan pembayaran administrasi sekolah. Permasalahan yang terjadi pada data uang SPP yaitu masalah pencurian data dan informasi, hingga pencuri dapat memanipulasi data. Maka diperlukan sebuah Teknik untuk mengamankan data yang sering di sebut dengan kriptografi. Salah satu algoritma atau metode dalam kriptografi adalah *Advanced Encryption Standard* (AES). AES memiliki putaran kunci untuk proses enkripsi dan dekripsi. AES digunakan karena memberikan tingkat kemanan yang tinggi berdasarkan kunci rahasia yang kompleks sehingga dapat merahasiakan data yang akan diamankan. AES melakukan Teknik enkripsi-dekripsi pada data uang SPP sekolah agar tidak dapat dibaca, dicuri, dimanipulasi, dan dibocorkan oleh orang yang tidak bertanggung jawab. Teknik enkripsi membuat isi dari data berubah menjadi kode-kode tertentu yang tidak dapat dibaca isinya. Untuk itu, fungsi keberadaan kriptografi AES diperlukan sebagai cara untuk mengamankan isi dari data uang SPP pada sekolah SMK Harapan Bangsa tersebut agar aman dari pencurian data.

Keywords:
AES
Decryption
Encryption
Cryptography
Data Security

ABSTRACT

The problem of data and information security is one of the important aspects of computer information. One example of a data security problem is the data security of tuition fees in schools. SPP money data is a collection of sensitive data for schools. The data in it is in the form of a summary or record of school administration payments. The problem that occurs in the SPP money data is the problem of data and information theft, so that the thief can manipulate the data. So we need a technique to secure data which is often called cryptography. One of the algorithms or methods in cryptography is the Advanced Encryption Standard (AES). AES has a key loop for the encryption and decryption process. AES is used because it provides a high level of security based on a complex secret key so that it can keep the data to be secured secret. AES performs encryption-decryption techniques on school tuition money data so that it cannot be read, stolen, manipulated, and leaked by irresponsible people. Encryption techniques make the contents of the data turn into certain codes that cannot be read. For this reason, the existence of AES cryptography is needed as a way to secure the contents of the SPP money data at the Harapan Bangsa Vocational School to be safe from data theft.

I. Pendahuluan

Teknologi komputer dan telekomunikasi saat ini telah mengalami kemajuan dan sudah menjadi suatu kebutuhan yang penting bagi setiap orang, karena banyaknya pekerjaan yang dapat diselesaikan dengan cepat. Salah satu dampak negatif di dalam perkembangan teknologi adalah adanya pencurian data. Dengan adanya pencurian data maka sisi keamanan dalam pertukaran informasi dan penyimpanan data dianggap penting. Salah satu data yang penting yaitu masalah keuangan, data keuangan merupakan sebuah data yang berisifat sensitif. Sehingga banyak kejahatan yang terjadi. Untuk menjaga keamanan data ini di perlukan sistem yang mampu untuk mengamankan data keuangan. Terdapat beberapa algoritma dalam kriptografi, salah satunya yaitu AES.

Pada penelitian sebelumnya dalam jurnal [1] dengan judul Implementasi Algoritma *Advanced Encryption Standard* (AES) 128 Untuk Enkripsi dan Dekripsi file Dokumen, proses enkripsi dan dekripsi yang dilakukan dalam perlindungan data informasi berbasiskan algoritma kriptografi Algoritma AES-128 dengan Kunci Simetri. Penerapan algoritma ini akan dilakukan pada pengamanan data berjenis dokumen dengan tipe pdf, doc, txt. Pada SMK Harapan bangsa masih sangat kecil keamanan dalam keamanan dalam kasus keuangan, contohnya pada data keuangan SPP. Sehingga pernah terjadi ketidaksamaan data atau pencurian data keuangan SPP.

Maka dengan adanya metode AES ini, dapat mempermudah pekerjaan untuk mengamankan dokumen sekolah pada SMK Harapan Bangsa. Khususnya untuk pengamanan data uang SPP. Didalam penelitian ini terdapat proses enkripsi dan dekripsi yang dilakukan dalam menggunakan algoritma AES 128 bit. Dengan menggunakan sistem berbasis desktop, menggunakan *software Microsoft Visual Studio* 2015, *Microsoft Excel*, dan *Microsoft Accsess* untuk proses mengamankan data uang SPP.

II. Metode

A. Kriptografi

Kriptografi berasal dari bahasa Yunani, *crypto* dan *graphia*. *Crypto* berarti *secret* (rahasia) dan *graphia* berarti *writing* (tulisan). Menurut terminologinya, Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan atau data dikirim dari suatu tempat ke tempat yang lain, [2].

B. Algoritma Advanced Encryption Standard (AES)

Algoritma *Advanced Encryption Standard* (AES) adalah suatu algoritma *block chiper* dan mempunyai sifat simetri yang menggunakan kunci simetri pada waktu proses enkripsi dan dekripsi. Pada tahun 2001, AES digunakan sebagai standar algoritma kriptografi terbaru yang dipublikasikan oleh NIST (*National Institute of Standard and Technology*) sebagai pengganti algoritma DES (*Data Encryption Standard*) yang sudah berakhir masa penggunaannya, [1].

Proses enkripsi yang dilakukan menggunakan algoritma AES yaitu:

1. *SubBytes*
   SubBytes merupakan transformasi *byte* dimana setiap elemen pada state akan dipetakan dengan menggunakan sebuah tabel substitusi (S-Box). Untuk setiap *byte* pada state dinyatakan dengan S'[r, c]. S'[r, c] adalah elemen di dalam tabel subtitusi yang merupakan perpotongan baris (x) dengan kolom (y).
2. *Shiftrows*
   *Shiftrow*s pada dasarnya adalah proses pergeseran bit dimana bit paling kiri akan dipindahkan menjadi bit paling kanan (rotasi bit). Namun jumlah pergeseran yang dilakukan berbeda, tergantung untuk setiap barisnya. Baris pertama tidak terjadi pergeseran. Setiap *byte* dari baris kedua pada matriks state digeser satu *byte* ke kiri. Selanjutnya baris ketiga digeser ke kiri sebanyak dua *byte* dan pada baris keempat digeser ke kiri sebanyak tiga *byte*. Proses ini bertujuan untuk menghasilkan *diffusion* yakni dengan menyebarkan pengaruh transformasi nonlinear pada baris-baris matriks state untuk putaran selanjutnya[3].
3. *MixColumns*
   Pada proses *MixColumns*, tiap kolom dari matriks state dilakukan operasi perkalian. Hal ini bertujuan untuk menyebarkan pengaruh setiap bit plaintext dan *cipherkey* terhadap *ciphertext* yang dihasilkan, pada arah kolom matriks state. Setiap kolom matriks state diperlakukan sebagai polinomial empat suku dalam *Galois field*, kemudian dikalikan dengan modulo ($X^8 + X^4 + X^3 + X + 1$). Operasi *MixColumns* juga dapat

dipandang sebagai perkalian matriks, dengan mengalikan empat bilangan di dalam *Galois field* *MixColumn*s juga disebut sebagai proses mengalikan setiap kolom dengan matriks berikut:

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}$$

Atau : $A(x) = \{03\}x^2 + \{01\}x^2 + \{01\}x + \{02\}$

4. *AddRound*Key

Dalam tahap *AddRoundKey* ini, *cipherkey* yang telah ada di ekspansikan terlebih dahulu maka akan di dapat *roundkey* yang akan digunakan untuk proses selanjutnya. Kemudian setiap *byte* dari matriks state keluaran proses *MixColumns* dilakukan operasi XOR dengan setiap *byte* dari *roundkey*. Proses *round* atau proses *SubBytes, ShiftRows, MixColumns, dan AddRoundKey* dilakukan hingga putaran ke-n dengan cara yang sama. Sedangkan utuk putaran terakhir atau disebut juga *final round proses SubBytes, ShiftRows, dan AddRoundKey* tetap dilakukan tetapi proses *MixColumns* tidak dilakukan, [2][4].

Proses dekripsi yang dilakukan menggunakan algoritma *Advanced Encryption Standard* (AES) yaitu:

1. *AddRoundKey*
   Inverse atau kebalikan dari tahap *AddRoundKey* adalah operasi XOR antara *byte-byte* matriks *state* yang disusun dari *ciphertext* dengan *byte-byte roundkey* yang dibangkitkan sebelumnya. *Roundkey* yang digunakandi setiap iterasinya berkebalikan dengan *roundkey* yang ada pada proses enkripsi. *Inverse* dari transformasiini digunakan untuk proses dekripsi [5].
2. *Inverse SubBytes*
   Inverse SubBytes juga merupakan transformasi *bytes* yang berkebalikan dengan transformasi *SubBytes*. Pada *Inverse SubBytes*, tiap elemen pada state dipetakan dengan menggunakan tabel *Inverse S-Box*.
3. *InverseShiftRow*
   Untuk proses dekripsinya dilakukan proses *Inverse* dari transformasi *ShiftRows. InverseShiftRow* sadalah transformasi *byte* yang berkebalikan dengan transformasi *ShiftRows.* Pada transformasi *InverseShiftRows,* dilakukan pergeseran bit ke kanan sedangkan pada *ShiftRows* dilakukan pergeseran bit ke kiri.
4. *Inverse MixColumns*
   Untuk melakukan dekripsi pesan, dilakukan *inverse* dari transformasi *MixColumns* yakni mengalikan setiap kolom hasil dari *Inverse AddRoundKey* dengan matriks berikut, [2].

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0\mathrm{E} \end{bmatrix}$$

III. Hasil dan Pembahasan

A. Analisa

Analisis data adalah cara untuk menganalisa permasalahan berdasarkan data yang telah diperoleh dari hasil studi lapangan baik observasi maupun dengan wawancara. Data yang dianalisis penulis pada penelitian ini seperti data nama siswa, nilai uang SPP/bulan, dan jumlah keseluruhan uang SPP dalam 1 tahun.

Tabel 1. Sampel Data Keuangan SPP Unit XII TKJ SMK Harapan Bangsa

No	Nama Siswa	Bulan				
		Juli	Agustus	September	Oktober	November
1	CINDY TRI ANTIKA	Rp.140.000	Rp.140.000	Rp.140.000	Rp.140.000	Rp.140.000

1. Ekspansi Kunci

Ekspansi kunci dibutuhkan untuk proses enkripsi dan deskripsi pada algoritma *Advanced Encryption Standard* (AES) [6]. 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 "SMKHARAPANBANGSA". Berikut adalah proses ekspansi kunci dengan metode AES 128 bit.

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

S	M	K	H	A	R	A	P	A	N	B	A	N	G	S	A
53	4D	4B	48	41	52	41	50	41	4E	42	41	4E	47	53	41

2. Selanjutnya adalah mengubah kunci yang telah diubah ke dalam *state* 4 x 4 seperti berikut:

53	41	41	4E
4D	52	4E	47
4B	41	42	53
48	50	41	41

⟶ *RoundKey* ke-0

3. 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.

4E			47
47	⟶		53
53			41
41			4E

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

47	⟶	A0
53	⟶	ED
41	⟶	83
4E	⟶	2F

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*.

53		A0		01		F2	
4D	⊕	ED	⊕	00	=	A0	Kolom ke-1
4B		83		00		C8	
48		2F		00		67	

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

41		F2	=	B3
52	⊕	A0		F2

Kolom ke-2

41		C8		89	
50		67		37	
41	⊕	B3	=	F2	
4e		F2		BC	Kolom ke-3
42		89		CB	
41		37		76	
4e		F2		BC	
47		BC		FB	
53	⊕	CB	=	98	Kolom ke-4
41		76		37	

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

F2	B3	F2	BC
A0	F2	BC	FB
C8	89	CB	98
67	37	76	37

Untuk mendapatkan *RoundKey* ke-2 sampai dengan *RoundKey* ke-10, proses di atas diulang sebanyak 10 kali.

2. Enkripsi (*Encryption*)

Proses enkripsi akan dilakukan pada *record database* data keuangan SPP unit XII TKJ T.A 2020/2021 di SMK Harapan Bangsa. *Plaintext* yang dienkripsi adalah berdasarkan data sampel diatas yaitu "CINDY TRI ANTIKA", dengan proses enkripsi seperti berikut ini:

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

C	I	N	D	Y		T	R	I		A	N	T	I	K	A
43	49	4E	44	59	20	54	52	49	20	41	4E	54	49	4B	41

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

43	59	49	54
49	20	20	49
4E	54	41	4B
44	52	4E	41

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

43	59	49	54
49	20	20	49
4E	54	41	4B
44	52	4E	41

⊕

53	41	41	4E
4D	52	4E	47
4B	41	42	53
48	50	41	41

→

10	18	08	1A
04	72	6E	0E
05	15	03	18
0C	02	0F	00

4. Hasil dari *AddRoundKey* diatas akan menjadi *round* ke-1 untuk diproses dengan 4 transformasi, yaitu *SubBytes, ShiftRows, MixColumns* dan *AddRoundKey*.
   a. Transformasi pertama yaitu *SubBytes*, pada tahap ini setiap *byte* akan ditukar dengan nilai pada tabel S-*Box*.

10	18	08	1A
04	72	6E	0E
05	15	03	18
0C	02	0F	00

→

CA	AD	30	A2
F2	40	9F	AB
6B	59	7B	AD
FE	77	76	63

   b. Transformasi berikutnya adalah ShiftRows, baris pertama tidak ada 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

CA	AD	30	A2
F2	40	9F	AB
6B	59	7B	AD
FE	77	76	63

→

CA	AD	30	A2
40	9F	AB	F2
7B	AD	6B	59
63	FE	77	76

   c. Selanjutnya adalah proses *MixColumns*, dimana proses ini akan melakukan 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

CA	AD	30	A2
40	9F	AB	F2
7B	AD	6B	59
63	FE	77	76

Byte baris 1 kolom 1 ($S^1_{0,0}$)

= ({02}{CA}) ⊕ ({03}{40}) ⊕ ({01}{7B}) ⊕ ({01}{63})

= ({10}{1001010}) ⊕ ({11}{01000000}) ⊕ 01111011 □ 01100011

= 10001111 ⊕ 11000000 ⊕ 01111011 ⊕ 01100011

= 01010111

= 57

   d. Tranformasi akhir dari *round* ke-1 adalah *AddRoundKey*, hasil dari *MixColumns* akan di XOR-kan dengan *RoundKey* ke-1, seperti dibawah ini.

57	C3	25	AF
53	1B	BC	85
4E	F7	F1	65
E2	03	16	71

⊕

10	18	08	1A
04	72	6E	0E
05	15	03	18
0C	02	0F	00

=

A5	**1B**	**68**	**C1**
04	**68**	**0B**	**3B**
11	**E3**	**1F**	**E0**
DF	**0E**	**08**	**8D**

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-10

SubBytes

C7	B6	1E	72
19	1A	14	38
5D	9D	63	26
33	97	3F	37

ShiftRows

C7	B6	1E	72
1A	14	38	19
63	26	5D	9D
37	33	97	3F

RoundKey ke-10

6D	AA	A6	7F
97	5E	A8	DF
FF	02	05	6D
5E	77	F1	CA

AddRoundKey

AA	1C	B8	0D
8D	4A	90	C6
9C	24	58	F0
69	44	66	F5

Hasil dari proses enkripsi yaitu : AA8D9C691C4A2444B89058660DC6F0F5

3. Dekripsi (*Decryption*)

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* [7]. Berikut adalah proses dekripsi *chipertext* "AA8D9C691C4A2444B89058660DC6F0F5", yaitu:

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

2.

AA	1C	B8	0D
8D	4A	90	C6
9C	24	58	F0
69	44	66	F5

⊕

6D	AA	A6	7F
97	5E	A8	DF
FF	02	05	6D
5E	77	F1	CA

=

C7	B6	1E	72
1A	14	38	19
63	26	5D	9D
37	33	97	3F

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

C7	B6	1E	72
1A	14	38	19
63	26	5D	9D
37	33	97	3F

C7	B6	1E	72
19	1A	14	38
5D	9D	63	26
33	97	3F	37

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

31	79	E9	1E
8E	43	9B	76
8D	75	00	23
66	85	25	B2

5. Selanjutnya, lakukan operasi XOR antara *InvSubBytes* dengan *RoundKey* ke- 9 untuk transformasi *AddRoundKey*.

	31	79	E9	1E
a.	8E	43	9B	76
	8D	75	00	b. 23
	66	85	25	c. B2

⊕

AE	C7	0C	D9
D2	C9	F6	77
1D	FD	07	68
6B	29	86	3B

=

9F	BE	E5	C7
5C	8A	6D	01
90	88	07	4B
0D	AC	A3	89

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

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

X

9F	BE	E5	C7
5C	8A	6D	01
90	88	07	4B
0D	AC	A3	89

Baris 1 kolom 1

(0E).(9F)

= (1110).(10000011)

= (x3 + x2 + x) (x7 + x + 1)

= (x10 + x4 + x3) + (x9 + x3 + x2) + (x8 + x2 + x)

= x10 + x4 + x9 + x8 + x

= (x2 . x8) + (x . x8) + x8 + x4 + x

= x2 (x4 + x3 + x + 1) + x (x4 + x3 + x + 1) + x4 + x3 + x + 1 + x4 + x

= x6 + x5 + x3 + x2 + x5 + x4 + x2 + x + x4 + x3 + x + 1 + x4 + x

= x6 + x4 + x + 1

= 01010011

= 53

(0B).(5C)

= (1011).(01101101)

= (x3 + x + 1) (x6 + x5 + x3 + x2 + 1)

= (x9 + x8 + x6 + x5 + x3) + (x7 + x6 + x4 + x3 + x) + (x6 + x5 + x3 + x2 + 1)

= x9 + x8 + x7 + x4 + x + x6 + x3 + x2 + 1

= (x . x8) + x8 + x7 + x6 + x4 + x3 + x2 + x + 1

= x (x4 + x3 + x + 1) + (x4 + x3 + x + 1) + x7 + x6 + x4 + x3 + x2 + x + 1

= x5 + x4 + x2 + x + x4 + x3 + x + 1+ x7 + x6 + x4 + x3 + x2 + x + 1

= x7 + x6 + x5 + x4 + x

= 11110010

= F2

(0D).(90)

= (1101).(01011101)

= (x3 + x2 + 1) (x6 + x4 + x3 + x2 + 1)

= (x9 + x7 + x6 + x5 + x3) + (x8 + x6 + x5 + x4 + x2) + (x6 + x4 + x3 + x2 + 1)

= x9 + x7 + x8 + x6 + 1

= (x . x8) + x7 + x8 + x6 + 1

= x (x4 + x3 + x + 1) + x7 + (x4 + x3 + x + 1) + x6 + 1

= x5 + x4 + x2 + x + x7 + x4 + x3 + x + 1 + x6 + 1

= x7 + x6 + x5 + x3 + x2

= 11101100

= EC

(09).(0D)

= (1001).(11011010)

= (x3 + 1) (x7 + x6 + x4 + x3 + x)

= (x10 + x9 + x7 + x6 + x4) + (x7 + x6 + x4 + x3 + x)

= x10 + x9 + x3 + x

= (x2 . x8) + (x . x8) + x + 1

= x2 (x4 + x3 + x + 1) + x (x4 + x3 + x + 1) + x3 + x

= x6 + x5 + x3 + x2 + x5 + x4 + x2 + x + x3 + x

= x6 + x4

= 01010000

= 50

Setelah proses di atas, masing-masing hasil di-XOR, seperti berikut ini :

= 01010001 ⊕ 11110010 ⊕ 11101100 ⊕ 01010000

= 00011101

= A6

Setelah itu, lakukan proses yang sama seperti pada baris 1 kolom 1. Di bawah ini adalah hasil dalam bentuk *state* 4x4:

A6	9E	1C	72
18	86	FB	A7
15	88	A6	F6
F5	80	6D	27

43	59	49	54
49	20	20	49
4E	54	41	4B
44	52	4E	41

Proses di atas akan diulangi untuk mendapatkan hasil transformasi *round* ke-2 sampai dengan *round* ke-10.

Round ke-10

InvShiftRows

CA	AD	30	A2
F2	40	9F	AB
6B	59	7B	AD
FE	77	76	63

InvSubBytes

10	18	08	1A
04	72	6E	0E
05	15	03	18
0C	02	0F	00

RoundKey ke-0

53	41	41	4E
4D	52	4E	47
4B	41	42	53
48	50	41	41

AddRoundKey

C	I	N	D	Y		T	R	I		A	N	T	I	K	A
43	49	4E	44	59	20	54	52	49	20	41	4E	54	49	4B	41

Hasil dari proses dekripsi yaitu : “43494E44592054524920414E54494B41”

IV. Kesimpulan

Berdasarkan penelitian dan uraian yang tertera, maka didapatkan kesimpulannya sebagai berikut :

1. Kriptografi dengan metode AES 128 bit dapat melakukan proses enkripsi dan dekripsi data uang SPP SMK Harapan Bangsa dalam bentuk *file excel.*
2. Penelitian ini menghasilkan satu kemajuan teknologi yang dapat mengamankan data yang bersifat sensitive.

Ucapan Terima Kasih

Dalam penyusunan dan penerbitan jurnal ini penulis mengucapkan terimakasih kepada Tuhan Yang Maha Esa, kepada kedua orang tua yaitu Bapak Mujiono dan Ibu Sumiaty, kepada Dosen Pembimbing saya yaitu Ibu Fristi Riandari, M.Kom, kepada teman terbaik saya yaitu Aditya Pratama, kepada teman seperjuangan saya yaitu Rizky Alexander Manulang, dan kepada pihak JIKOMSI.

Daftar Pustaka

[1] A. Prameshwari and N. P. Sastra, "Implementasi Algoritma Advanced Encryption Standard (AES) 128 Untuk Enkripsi dan Dekripsi File Dokumen," *Eksplora Inform.*, vol. 8, no. 1, p. 52, 2018, doi: 10.30864/eksplora.v8i1.139.

[2] J. N. Sitompul, "Implementasi Metode Kriptografi Advanced Encryption Standard (AES) untuk Proteksi Pesan Audio," vol. 4, no. 1, pp. 37–45, 2019.

[3] C. Irawan, A. Winarno, P. Studi, S. Informasi, F. I. Komputer, and U. D. Nuswantoro, "Kombinasi Algoritma Kriptogarafi Aes Dan Des Untuk Enkripsi," *Proceeding SENDIU*, pp. 28–35, 2020.

[4] Fricles Ariwisanto Sianturi, "Perancangan Aplikasi Pengamanan Data Dengan Kriptografi Advanced Encryption Standard (AES)," *Pelita Inform. Budi Darma*, vol. 4, no. 1, pp. 42–46, 2013, [Online]. Available: http://ejurnal.stmik-budidarma.ac.id/index.php/pelita/article/view/208.

[5] S. Suhandinata, R. A. Rizal, D. O. Wijaya, P. Warren, and S. Srinjiwi, "Analisis Performa Kriptografi Hybrid Algoritma Blowfish Dan Algoritma Rsa," *JURTEKSI (Jurnal Teknol. dan Sist. Informasi)*, vol. 6, no. 1, pp. 1–10, 2019, doi: 10.33330/jurteksi.v6i1.395.

[6] A. Y. Mulyadi, E. P. Nugroho, and R. R. J. P, "Implementasi Algoritma AES 128 dan SHA – 256 Dalam Pengkodean pada Sebagian Frame Video CCTV MPEG-2," *JATIKOM J. Teor. dan Apl. Ilmu Komput.*, vol. 1, no. 1, pp. 33–39, 2018.

[7] D. Novianto and Y. Setiawan, "Aplikasi Pengamanan Informasi Menggunakan Metode Least Significant Bit (Lsb) dan Algoritma Kriptografi Advanced Encryption Standard (AES)," *J. Ilm. Inform. Glob.*, vol. 9, no. 2, pp. 83–89, 2019, doi: 10.36982/jig.v9i2.561.