Jurnal CyberTech
Vol.x. No.x, September 2020, pp. xx~xx
P-ISSN : 1978-6603
E-ISSN : 2615-3475

APLIKASI PENGAMANAN DOKUMEN SIMPAN PINJAM UANG DIPUSAT KOPERASI KARTIKA "A" BUKIT BARISAN MENGGUNAKAN METODE ADVANCED ENCRYPTION STANDARD (AES)

Rosdoana Ito Hasibuan[*], Azanuddin, S,Kom., M.Kom [**], Muhammad Syaifuddin, S.Kom., M.Kom[**]
[*] Program Studi Sistem Informasi, STMIK Triguna Dharma
[**] Program Studi Sistem Informasi, STMIK Triguna Dharma

Article Info
Article history:
Pengamanan Dokumen Simpan Pinjam Uang Dipusat Koperasi Kartika "A" Bukit Barisan 2020

Keyword:
Algoritma Kriptografi, Advanced Encryption Standard (AES), Dokumen Simpan Pinjam Uang.

ABSTRACT

Masalah keamanan dokumen simpan pinjam uang merupakan salah satu aspek penting untuk Pusat Koperasi Kartika "A" Bukit Barisan. Namun masalah keamanan ini sering kurang mendapat perhatian dari pengelola data atau dokumen simpan pinjam uang. Jika dokumen tersebut jatuh ke pihak lain yang tidak bertanggung jawab akan dapat menimbulkan risiko dan kerugian. masalah yang menjadi titik tolak dalam pembuatan skripsi ini adalah bagaimana menyelesaikan masalah tersebut dengan menggunakan metode advanced encryption standard (AES).

Berdasarkan permasalahan diatas maka dibuatlah suatu program yang dapat membantu, yaitu Kriptografi dengan menggunakan metode Advamced Encryption Standrad (AES) program ini dibuat untuk membantu mengamankan dokumen Dipusat Koperasi Kartika "A" Bukit Barisan.

Hasil akhir dari penelitian ini adalah untuk mempermudah pengguna untuk melakukan pengamanan dokumen terhadap keamanan dokumen-dokumen penting Dipusat Koperasi Kartika "A" Bukit Barisan menggunakan metode Advanced Encryption Standard (AES), sehingga memberikan hasil dokumen yang lebih aman yang dapat menunjukkan tingkat kepercayaan sistem terhadap dokumen yang sudah di enkripsi.

.

First Author
Nama : Rosdoana Ito Hasibuan
Program Studi : Sistem Informasi
Kampus : STMIK Triguna Dharma
Email : rosdoanahasibuan@gmail.com

1. PENDAHULUAN

Lembaga keuangan adalah kegiatan usaha simpan pinjam uang kepada para anggotanya dengan memberi bungan yang lebih rendah dari simpan pinjam lainnya. Bentuk dari koperasi tersebut adalah untuk orang yang sudah tergabung dalam koperasi yang dapat menyimpan dan meminjam uang, sedangkan anggota yang tidak terdaftar tidak dapat menyimpan atau meminjam uang dari koperasi tersebut.

Masalah keamanan dokumen simpan pinjam uang merupakan salah satu aspek penting untuk Pusat Koperasi Kartika "A" Bukit Barisan. Namun Masalah keamanan ini sering kurang mendapat perhatian dari pengelola data atau dokumen simpan pinjam uang. Jika dokumen tersebut jatuh ke pihak lain yang tidak bertanggung jawab akan dapat menimbulkan risiko dan kerugian [2]. Jika selama ini dokumen hanya disimpan secara manual yang keamanannya belum pasti terjamin. maka dengan penelitian ini dibuat dalam bentuk yang berbeda menggunakan enkripsi dan dekripsi agar terjaga keamanannya.

Berdasarkan permasalahan yang telah dipaparkan maka dibutuhkan suatu sistem yang mampu mengakuisisi keilmuan dan pengetahuan peneliti agar dapat digunakan dalam suatu sistem yang dapat melakukan pengamanan dokumen. Dalam ilmu computer hal ini sangat dikaitkan dengan istilah kriptografi.

Kriptografi merupakan salah satu teknik penyandian suatu dokumen, data, informasi dan pesan disembunyikan dengan sekumpulan teknik yang digunakan untuk meningkatkan aspek keamanan suatu informasi [3].Salah satu metode dalam Kriptografi yaitu AES (Advanced Encryption Standart).

Metode Advanced Encryption Standard (AES) merupakan blok ciphertext simetris yang dapat mengenkripsikan dan mendekripsikan dokumen menggunakan kunci Kriptografi 128, 192 dan 256 bit[5] untuk mengenkripsi dan dekripsi data pada blok 128 bit.

2. KAJIAN PUSTAKA

2.1 Kriptografi

Kriptografi berasal dari bahasa asing yaitu bahasa yunani, yang dibagi menjadi dua kata, yaitu Kripto dan Graphia. Dimana Kripto adalah rahasia dan Graphia adalah tulisan. Sehingga kriptografi merupakan sebagai ilmu yang digunakan untuk menjaga keamanan dan kerahasian sebuah pesan (Privacy) [6].

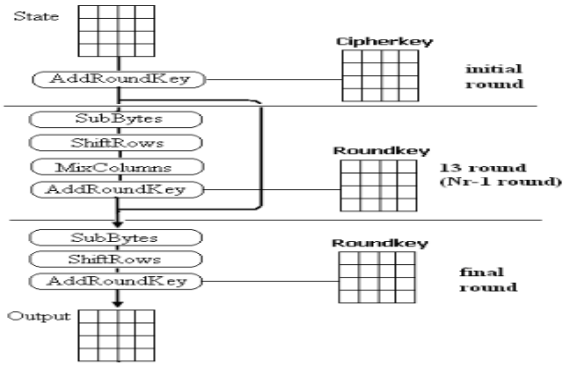
Dengan kriptografi, orang-orang tidak bisa memahami tulisan tersebut dan tidak dapat mengetahui bagaimana cara membaca serta mengartikan tulisan tersebut.

2.2 Advanced Encryption Standard (AES)

AES (Advanced Encryption Standard) adalah suatu algoritma untuk mengamankan sebuah data yang merupakan blok chipertext simetrik dan dapat mengenkripsikan serta mendekripsikan informasi. Yang menggunakan kunci 128, 192, dan 256 bit untuk mengenkripsi dan dekripsi data [10].

1. Proses Enkripsi

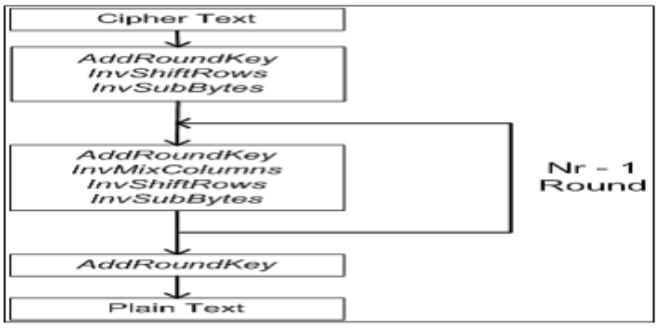
Proses enkripsi pada algoritma AES terdiri dari 4 jenis transformasi bytes, yaitu SubBytes, ShiftRows, Mixcolumns dan AddRoundKey. Ilustrasi proses enkripsi AES dapat digambarkan seperti pada gambar dibawah ini.



Gambar 2.1 Proses Enkripsi AES

2. Proses Dekripsi

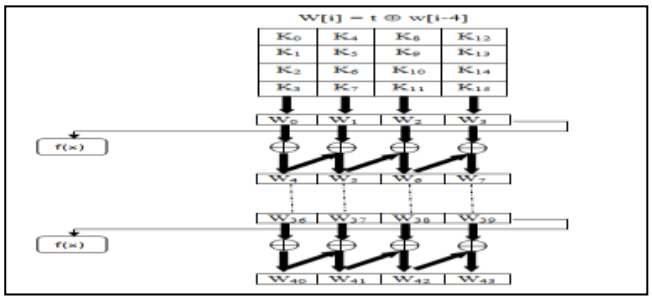
Transformasi cipher dapat dibalikkan dan di implementasikan dalam arah yang berlawanan untuk menghasilkan inverse cipher yang mudah dipahami untuk algoritma AES. Transformasi byte yang digunakan pada invers cipher adalah InvShiftRows, InvSubBytes, InvMixColumns, dan AddRoundKey. Algoritma dekripsi dapat dilihat pada gambar berikut.



Gambar 2.2 Diagram Dekripsi AES

3. Enkspansi Kunci AES

Penyandian AES membutuhkan kunci ronde untuk setiap ronde transformasi. Kunci ronde ini dibangkitkan (diekspansi) dari kunci AES. Pada bagian ini dibahas bagaimana kunci ronde dibangkitkan oleh kunci AES dengan panjang 128 bit. Ekspansi kunci ronde dengan panjang 192 bit dan 256 bit mirip dengan ekspansi kunci AES 128 bit dengan sedikit modifikasi.



Gambar 2.3 Ekspansi Kunci AES

2.3 Simpan Pinjam Uang

Istilah simpan pinjam uang adalah jasa yang menyediakan peminjaman dan penyimpanan dana kepada anggotanya dengan modal dan bunga yang lebih rendah [15].

Pada kegiatan meminjam dan menyimpan uang dengan modalnya yang diperoleh dari simpanan-simpanan sebagai berikut [16]

1. Simpanan menjadi bentuk permanen yang diberikan anggota pada awal setoran yang disebut dengan simpanan pokok.
2. Simpanan yang bisa diambil dengan waktu tertentu yang disebut dengan simpanan wajib.
3. Simpanan yang sudah diterima tetapi tidak dari anggota koperasi itu sendiri yang disebut dengan simpanan sukarela.

Dari ilustrasi tersebut, simpan pinjam uang adalah simpanan yang dikumpulkan secara bersama-sama dan diberikan pinjaman untuk anggota yang memerlukan dengan mengajukan surat permohonan tertulis dengan membuat jumlah uang yang ingin dipinjam [17].

3. METODOLOGI PENELITIAN

3.1 Analisisa Dan Hasil

Berikuti ini merupakakn hasil dari analisis algoritma perhitungan metode AES, dimana dimulai dari proses ekpansi kunci, enkripsi dan dekripsi.

3.1.1 Ekspansi Kunci

Kunci ronde round key diperlukan untuk proses enkripsi dan dekripsi Advanced Encryption Standard. Maximal panjang kunci adalah sebnayak 10 digit dan jumlah kunci ronde yang dibutuhkan yaitu 10 kunci yang akan diperoleh dari proses ekspansi kunci. Pada kasus ini, kunci yang akan digunakan yaitu “DATASIMPANPINJAM”. Berikut ini adalah proses ekspansi kunci pada algoritma Advanced Encryption Standard.

1. Urutkan kunci kedalam blok berukuran 128 bit (16 kode ASCII). Lalu ubah kunci kedalam bentuk *hexadecimal*.

D	A	T	A	S	I	M	P	A	N	P	I	N	J	A	M
44	41	54	41	53	49	4D	50	41	4E	50	49	4E	4A	41	4D

2. Langkah selanjutnya yaitu susun kunci yang telah diubah ke dalam bentuk *hexadecimal* ke dalam *state* berukuran 4 × 4 seperti dibawah ini :

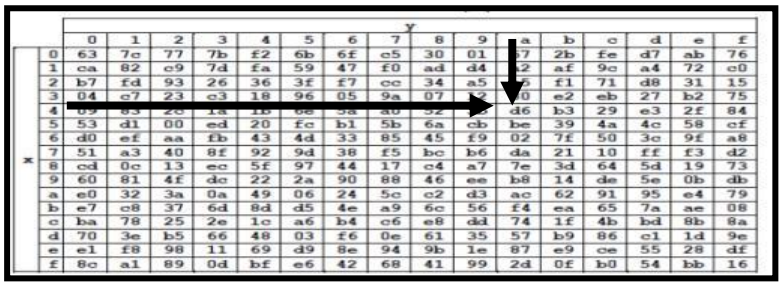
44	53	41	4E
41	49	4E	4A
54	4D	50	41
41	50	49	4D

Cipherkey/ kunci ronde ke - 0

3. Setelah itu, untuk mendapatkan hasil kolom pertama pada sub kunci, langkah pertama yaitu dilakukan fungsi . *RotWord* yaitu dengan menggeser setiap bit pada kolom ke-4 ke atas 1 kali dari kunci ronde ke-0.

4E	4A
4A	41
41	4D
4D	4E

4. Kemudian hasil dari *RotWord* disubtitusikan dengan nilai pada tabel S-*Box* (*SubBytes*).



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

5. Tahap akhir untuk mendapatkan kolom pertama yaitu proses *XOR* antara kolom pertama dari kunci ronde ke-0 dan hasil dari S*ubBytes* lalu di XOR-kan lagi dengan Rcon.

1	2	3	4	5	6	7	8	9	10
01	02	04	08	10	20	40	80	1B	3C
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

kolom rcon

44		D6		01	=	93	kolom pertama (*wi*) pada kunci
41	⊕	83	⊕	00		C2	
54		E3		00		B7	
41		2F		00		6E	

6. Untuk mendapatkan kolom kedua, diperoleh dengan *XOR* antara *wi* dengan kolom kedua dari kunci ronde ke-0. Untuk mendapatkan kolom ketiga dan keempat kunci ronde ke-1, dilakukan proses seperti memperoleh kolom kedua

53		93		C0		41		C0		81		4E		81		CF	
49	⊕	C2	=	8B		4E	⊕	8B	=	C5		4A	⊕	C5	=	8F	
4D		B7		FA		50		FA		AA		41		AA		EB	
50		6E		3E		9		3E		77		4D		77		3A	
				↑						↑						↑	
				Kolom Kedua						Kolom Ketiga						Kolom Keempat	

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

93	C0	81	CF
C2	8B	C5	BF
B7	FA	AA	EB
6E	3E	77	3A

Algoritma Advanced Encryption Standard 128 bit ini memiliki 10 ronde sehingga diperlukan 10 kunci ronde (round key). Untuk mendapatkan kunci ronde ke-2 sampai ke-10, proses diatas diulang sampai 10 kali. Kunci masing-masing ronde akan digunakan saat proses enkripsi dan dekripsi. Di bawah ini adalah hasil ekspansi key hingga ronde ke-10 yang nantinya akan digunakan dalam proses enkripsi dan dekripsi :

93	C0	81	CF
C2	8B	C5	BF
B7	FA	AA	EB
6E	3E	77	3A

Kunci Ronde Ke-1

E2	22	A3	6C
2B	A0	65	EA
37	CD	67	8C
E4	DA	AD	97

Kunci Ronde Ke-2

61	43	E0	8C
4F	EF	8A	60
BF	72	15	99
B4	6E	C3	54

Kunci Ronde Ke-3

B9	FA	1A	96
A1	4E	C4	A4
9F	ED	F8	61
D0	BE	7D	29

Kunci Ronde Ke-4

E0	1A	00	96
4E	00	C4	60
3A	D7	2F	4E
40	FE	83	AA

Kunci Ronde Ke-5

10	0A	0A	9C
61	61	A5	C5
96	41	6E	20
D0	2E	AD	07

Kunci Ronde Ke-6

F6	FC	F6	6A
D6	B7	12	D7
53	12	7C	5C
0E	20	8D	8A

Kunci Ronde Ke-7

78	84	72	18
9C	2B	39	EE
2D	3F	43	1F
0C	2C	A1	2B

Kunci Ronde Ke-8

4B	CF	BD	A5
5C	77	4E	A0
DC	E3	A0	BF
A1	8D	2C	07

Kunci Ronde Ke-9

97	58	E5	40
54	23	6D	CD
19	FA	5A	E5
A7	2A	06	01

Kunci Ronde Ke-10

3.1.2 Enkripsi

Penjelasan mengenai proses enkripsi algoritma Advanced Encryption Standard telah diuraikan pada bab sebelumnya. Pada kasus ini, plaintext yang akan di enkripsi berupa dokumen simpan pinjam uang dipusat koperasi kartika "A" bukit barisan. Adapun contoh simpan pinjam uang yaitu "PUSKOPKARTIKA BB". Berikut adalah proses enkripsi dari plaintext tersebut :

1. Urutkan *plaintext* ke dalam blok dan ubah ke bilangan *hexadecimal*.

P	U	S	K	O	P	K	A	R	T	I	K	A		B	B
50	55	53	4B	4F	50	4B	41	52	54	49	4B	41	20	42	42

2. Lalu, susun 16 *byte* pertama dari *plaintext* yang telah diubah ke bentuk *hexadecimal* ke dalam *state* 4 × 4.

50	4F	52	41
55	50	54	20
53	4B	49	42
4B	41	4B	42

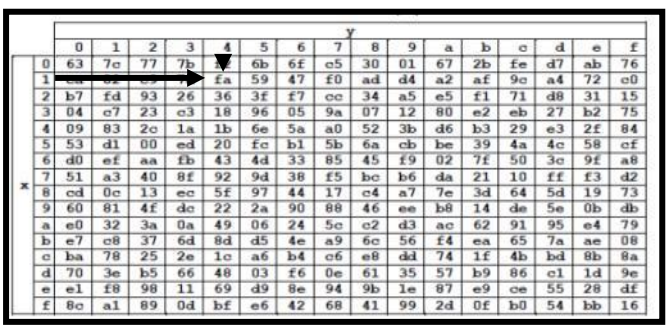
3. *Plaintext* di atas di *XOR*-kan dengan *cipherkey* atau kunci ronde ke-0. Proses ini dinamakan *AddRoundKey*.

50	4F	52	41	⊕	44	53	41	4E	=	14	1C	13	0F
55	50	54	20		41	49	4E	4A		14	19	1A	6A
53	4B	49	42		54	4D	50	41		07	06	19	03
4B	41	4B	42		41	50	49	4D		0A	11	02	0F

4. Proses di atas masih dalam *pra-round*. Hasil dari *AddRoundKey* di atas akan menjadi masukan untuk ronde ke-1 yang akan diproses dengan 4 transformasi, yaitu *SubBytes*, *ShiftRows*, *MixColumns* dan *AddRoundKey*.

Round 1

1. Pada transformasi *SubBytes*, setiap *byte* akan ditukar dengan tabel S-*Box*.



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

14	1C	13	0F
14	19	1A	6A
07	06	19	03
0A	11	02	0F

SubBytes →

FA	9C	7D	76
FA	D4	A2	02
C5	6F	D4	7B
67	82	77	76

2. Kemudian, dilanjutkan dengan melakukan proses *ShiftRows*, yaitu menggeser setiap baris pada *state*.

Tetap →	FA	9C	7D	76	=	FA	9C	7D	76
Digeser 1 byte ke kiri →	FA	D4	A2	02		D4	A2	02	FA
Digeser 2 byte ke kiri →	C5	6F	D4	7B		D4	7B	C5	6F
Digeser 3 byte ke kiri →	67	82	77	76		76	67	82	77

3. Proses selanjutnya yaitu *MixColumns*. Pada proses ini, dilakukan proses perkalian antara suatu polinominal tetap dengan *state* hasil *ShiftRows*.

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} FA & 9C & 7D & 76 \\ D4 & A2 & 02 & FA \\ D4 & 7B & C5 & 6F \\ 76 & 67 & 82 & 77 \end{bmatrix} = \begin{bmatrix} 2A & C2 & BB & E1 \\ 58 & 29 & AF & 5F \\ FA & 61 & 73 & CB \\ F9 & A8 & 5F & E1 \end{bmatrix}$$

Proses perhitungan untuk mencari baris pertama menggunakan operator polinominal GF(2^8) dimana jika dikali 01 maka hasilnya tetap, jika dikali 02 maka dileftshit 1 dan jika hasil leftshift 3 byte maka di XOR dengan 11B dan jika dikali 03 maka dilakukan operasi dikali 02 dan XOR dengan bilangan itu sendiri, dibawah ini merupakan uraian perkalian MixColumns.

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

02 × FA = (10) × (11111010)
= 11110100
= F4

03 × D4 = (11) × (11010100)
= 01111100
= 7C

01 × D4 = (1) × (11010100)
= 11010100
= D4

01 × 76 = (1) × (01110110)
= 01110110
= 76

Untuk mendapatkan $S'_{0,0}$, semua hasil dari proses perkalian di atas di-*XOR*-kan seperti di bawah ini.

$S'_{0,0}$ = F4 $\oplus$ 7C $\oplus$ D4 $\oplus$ 76
= 2A

4. Setelah hasil dari proses *MixColumns* diperoleh, langkah terakhir dari ronde ke-1 yaitu *AddRoundKey*. Proses *AddRoundKey* ini sama dengan sebelumnya, namun *state* hasil dari proses *MixColumns* di-*XOR*-kan dengan kunci ronde ke-1. Dibawah ini adalah proses *AddRoundKey* ronde ke-1.

2A	C2	BB	E1
58	29	AF	5F
07	61	73	CB
F9	A8	5F	E1

93	CO	81	CF
C2	8B	C5	BF
B7	FA	AA	EB
6E	3E	77	3A

B9	02	3A	2E
9A	A2	6A	E0
B0	9B	D9	20
97	96	28	DB

Hasil proses enkripsi dari ronde ke-1 akan menjadi masukkan untuk ronde ke-2 begitu juga untuk ronde selanjutnya. Proses di atas akan diulangi sampai ronde ke-10. Hasil dari transformasi proses enkripsi untuk ronde ke-2 sampai ke-10 dapat dilihat di bawah ini.

Round 2

Text	*SubBytes*	*ShiftRows*	*MixColumns*	*RoundKey 2*	*AddRoundKey*
B9 02 3A 2E	56 77 80 31	56 77 80 31	6E D7 FC 91	E2 22 A3 6C	8C F5 5F FD
9A A2 6A E0	B8 3A 02 70	3A 02 70 B8	C4 39 C2 52	2B A0 65 EA	EF 99 A7 B8
B0 9B D9 20	E7 14 35 B7	35 B7 E7 14	D6 83 8E FD	37 CD 67 8C	E1 4E E9 71
97 96 28 DB	88 90 34 B9	B9 88 90 34	9C 27 37 97	E4 DA AD 97	78 FD 9A 00

Round 3

Text	*SubBytes*	*ShiftRows*	*MixColumns*	*RoundKey 3*	*AddRoundKey*
8C F5 5F FD	64 E6 CF 54	64 E6 CF 54	9C 2C 9D 45	61 43 E0 8C	FD 2C 7D C9
EF 99 A7 B8	DF EE 5C 6C	EE 5C 6C DF	E2 1C 50 38	4F EF 8A 60	AD F3 DA 58
E1 4E E9 71	F8 2F 1E A3	1E A3 F8 2F	13 38 B4 06	BF 72 15 99	AC 4A A1 9F
78 FD 9A 00	BC 54 B8 63	63 BC 54 B8	9A AD 76 67	B4 6E C3 54	2E C3 B5 33

Round 4

Text	*SubBytes*	*ShiftRows*	*MixColumns*	*RoundKey 4*	*AddRoundKey*
FD 2C 7D C9	54 A8 FF DD	54 A8 FF DD	4E 58 E4 06	B9 FA 1A 96	F7 A2 FE 90
AD F3 DA 58	95 0D 57 6A	0D 57 6A 95	DB 41 AD 58	A1 4E C4 A4	7A 0F 69 FC
AC 4A A1 9F	91 D6 32 D8	32 D8 91 D6	63 01 DE 9B	9F ED F8 61	FC EC 26 FA
2E C3 B5 33	31 2E D5 C3	C3 31 2E D5	5E 0D BD 8E	D0 BE 7D 29	8E B3 C0 A7

Round 5

Text	*SubBytes*	*ShiftRows*	*MixColumns*	*RoundKey 5*	*AddRoundKey*
F7 A2 FE 90	68 3A BB 60	68 3A BB 60	E1 50 7B C1	E0 1A 00 96	01 4A 7B 57
7A 0F 69 FC	DA 76 F9 B0	76 F9 B0 DA	DA BD 66 3C	4E 00 C4 60	94 BD A2 5C
FC EC 26 FA	B0 CE F7 2D	F7 2D B0 CE	0F B2 C7 E8	3A D7 2F 4E	35 65 E8 A6
8E B3 C0 A7	19 6D BA 5C	5C 19 6D BA	81 A8 0C DB	40 FE 83 AA	C1 56 8F 71

Round 6

Text	*SubBytes*	*ShiftRows*	*MixColumns*	*RoundKey 6*	*AddRoundKey*
01 4A 7B 57	7C D6 21 5B	7C D6 21 5B	4E A5 BB EE	10 0A 0A 9C	5E A1 B1 72
94 BD A2 5C	22 7A 3A 4A	7A 3A 4A 22	9D B6 A5 BB	61 61 A5 C5	FC D7 00 7E
35 65 E8 A6	96 4D 9B 24	9B 24 96 4D	D5 2C 94 76	96 41 6E 20	43 6D FA 56
C1 56 8F 71	78 B1 73 A3	A3 78 B1 73	38 8F C6 64	D0 2E AD 07	E8 A1 6B 63

Round 7

Text	*SubBytes*	*ShiftRows*	*MixColumns*	*RoundKey 7*	*AddRoundKey*
5E A1 B1 72	58 79 C8 40	58 79 C8 40	74 7D AD 08	F6 FC F6 6A	82 81 5B 62
FC D7 00 7E	B0 0E 63 F3	0E 63 F3 B0	C8 EC 29 00	D6 B7 12 D7	1E 5B 3B D7
43 6D FA 56	1A 3C 2D B1	2D B1 1A 3C	1A D5 59 09	53 12 7C 5C	49 C7 25 55
E8 A1 6B 63	9B 32 7F FB	FB 9B 32 7F	26 74 C3 B2	0E 20 8D 8A	28 54 4E 38

Round 8

Text	*SubBytes*	*ShiftRows*	*MixColumns*	*RoundKey 8*	*AddRoundKey*
82 81 5B 62	13 0C 39 AA	13 0C 39 AA	55 ED 7B 05	78 84 72 18	2D 69 09 1D
1E 5B 3B D7	72 39 E2 0E	39 E2 0E 72	27 F8 48 05	9C 2B 39 EE	BB D3 71 EB
49 C7 25 55	3B C6 3F FC	3F FC 3B C6	5D 51 21 61	2D 3F 43 1F	70 6E 62 7E
28 54 4E 38	34 20 1A 07	07 34 20 1A	3D 62 3E 65	0C 2C A1 2B	31 5C 9F 4E

Round 9

Text	*SubBytes*	*ShiftRows*	*MixColumns*	*RoundKey 9*	*AddRoundKey*
2D 69 09 1D	D8 F9 01 A4	D8 F9 01 A4	84 23 5C 32	4B CF BD A5	CF EC E1 97
BB D3 71 EB	EA 66 A3 E9	66 A3 E9 EA	DE 6D 14 0A	5C 77 4E A0	82 1A 5A 00
70 6E 62 7E	51 9F AA F3	AA F3 51 9F	80 F5 3B 1D	DC E3 A0 BF	5C 16 9B A2
31 5C 9F 4E	C7 2F DB 2F	2F C7 2F DB	E1 D5 E5 2F	A1 8D 2C 07	40 58 C9 28

Round 10

Text				*SubBytes*				*ShiftRows*				*RoundKey 10*				*AddRoundKey*				*CipherText*			
CF	EC	E1	97	8A	CE	F8	88	8A	CE	F8	88	97	58	E5	40	1D	96	1D	C8	1D	96	1D	C8
82	1A	5A	00	13	A2	BE	63	A2	BE	63	13	54	23	6D	CD	F6	9D	C1	DE	F6	9D	C1	DE
5C	16	9B	A2	4A	47	14	3A	14	3A	4A	47	19	FA	5A	E5	0D	C0	10	A2	0D	C0	10	A2
40	58	C9	28	09	6A	DD	34	34	09	6A	DD	A7	2A	06	01	93	23	6C	DC	93	23	6C	DC

Hasil dari proses *AddRoundKey* pada ronde ke-10 merupakan hasil akhir proses enkripsi yaitu : 1DF60D93969DC0231DC1106CC8DEA2DC.

3.1.2 Dekripsi

Proses transformasi pada dekripsi dalam metode Advanced Encryption Standard yaitu InvSubBytes, InvShiftRows, InvMixColumns dan AddRoundKey. AddRoundKey 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.

Round 1

1. Lakukan proses *AddRoundKey* antara *ciphertext* yang telah diperoleh dari proses enkripsi dengan *roundkey* ke-10.

1D	96	1D	C8
F6	9D	C1	DE
0D	C0	10	A2
93	23	6C	DC

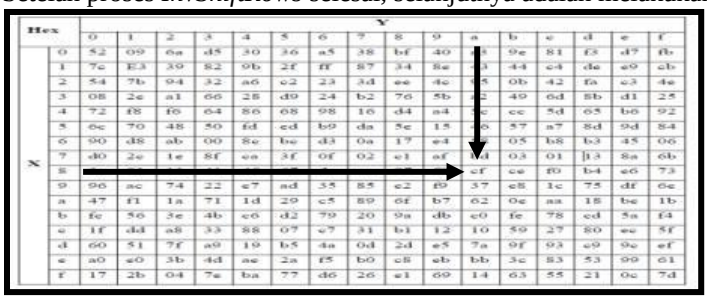
97	58	E5	40
54	23	6D	CD
19	FA	5A	E5
A7	2A	06	01

8A	CE	F8	88
A2	BE	63	13
14	3A	4A	47
34	09	6A	DD

2. Karena pada ronde ke-1 dalam proses dekripsi, tidak melakukan proses *InvMixColumns*. Maka proses selanjutnya adalah melakukan transformasi *InvShiftRows*.

Tetap →	8A	CE	F8	88	=	8A	CE	F8	88
Digeser 1 byte ke kanan →	A2	BE	63	13		13	A2	BE	63
Digeser 2 byte ke kanan →	14	3A	4A	47		4A	47	14	3A
Digeser 3 byte ke kanan →	34	09	6A	DD		09	6A	DD	34

3. Setelah proses *InvShiftRows* selesai, selanjutnya adalah melakukan proses transformasi *SubBytes*.



8A	CE	F8	88
13	A2	BE	63
4A	47	14	3A
09	6A	DD	34

InvSubBytes →

CF	EC	E1	97
82	1A	5A	00
5C	16	9B	A2
40	58	C9	28

4. Setelah proses *InvSubBytes* kemudian melakukan operasi *XOR* antara hasil *InvSubBytes* dengan Round Key 9 untuk melakukan transformasi putaran ke 2.

CF	EC	E1	97
82	1A	5A	00
5C	16	9B	A2
40	58	C9	28

⊕

4B	CF	BD	A5
5C	77	4E	A0
DC	E3	A0	BF
A1	8D	2C	07

=

84	23	5C	32
DE	6D	14	0A
80	F5	3B	1D
E1	D5	E5	2F

Hasil *InvSubByte* *Round Key 9* Hasil *AddRound Key*

5. Kemudian hasil *AddRound Key* tersebut akan melakukan proses transformasi *InvMixColumns* dengan aturan *irreducible polynomial*.

84	23	5C	32
DE	6D	14	0A
80	F5	3B	1D
E1	D5	E5	2F

⊕

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

=

6F	F9	01	A4
66	A3	E9	EA
AA	F5	51	9F
2F	C7	2F	DB

Hasil *AddRound Key* Nilai Matriks *InvMixColumns* Hasil Matriks *InvMixColumns*

Berikut uraian perhitungan transformasi *InvMixColumns* yang sesuai dengan perhitungan diatas.

$S_{0,0}$ = ($S_{0,0}$ * 0E) ⊕ ($S_{1,0}$ * 0B) ⊕ ($S_{2,0}$ * 0D) ⊕ ($S_{3,0}$ * 09)
= 84 * 0E ⊕ DE * 0B ⊕80 * 0D ⊕ E1 * 09
= 79 ⊕ D3 ⊕ 6D ⊕ A8
= 6F

Proses uraian perhitungan di atas dapat dirincikan dengan mengubah bilangan heksadesimal ke bilangan biner, kemudian di aplikasikan dengan *irreducible polynomial* sebagai berikut :

a. Representasi dari $S_{0,0}$ yaitu 84 (10000100) dalam *polynomial* ($x^7 + x^2$) dan bilangan 0E (00001110) dalam *polynomial* ($x^3 + x^2 + x$).

$S_{0,0}$ = 84 * 0E
= (10000100) * (00001110)
= ($x^7 + x^2$) * ($x^3 + x^2 + x$)
= ($x^{10} + x^9 + x^8 + x^5 + x^4 + x^3$)
= (($x^2.x^8$) + $x^9 + x^8 + x^5 + x^4 + x^3$)
= ((x^2 ($x^4 + x^3 + x + 1$)) + $x^9 + x^8 + x^5 + x^4 + x^3$)
= ((x^6 + ~~x^5~~ + ~~x^3~~ + x^2)) $x^9 + x^8$ + ~~x^5~~ + x^4 + ~~x^3~~)
= ($x^9 + x^8 + x^6 + x^4 + x^2$)
= ((x ($x^4 + x^3 + x + 1$)) + $x^8 + x^6 + x^4 + x^2$)
= ((x^5 + ~~x^4~~ + ~~x^2~~ + x) + $x^8 + x^6$ + ~~x^4~~ + ~~x^2~~)
= ($x^8 + x^6 + x^5 + x$)
= ((1 ($x^4 + x^3 + x + 1$)) + $x^6 + x^5 + x$)
= (($x^4 + x^3$+ ~~x~~ + 1) + $x^6 + x^5$ + ~~x~~)
= $x^6 + x^5 + x^4 + x^3 + 1$
= 01111001
= 79

Untuk proses *round* selanjutnya hanya akan ditampilkan hasil dari masing-masing transformasi yang dapat dilihat di bawah ini.

Round 2

Text	*Round Key 9*	*AddRoundKey*	*InvMixColumns*	*InvShiftRows*	*InvSubBytes*
CF EC E1 97	4B CF BD A5	84 23 5C 32	D8 F9 01 A4	D8 F9 01 A4	2D 69 09 1D
82 1A 5A 00	5C 77 4E A0	DE 6D 14 0A	66 A3 E9 EA	EA 66 A3 E9	BB D3 71 EB
5C 16 9B A2	DC E3 A0 BF	80 F5 3B 1D	AA F3 51 9F	51 9F AA F3	70 6E 62 7E
40 58 C9 28	A1 8D 2C 07	E1 D5 E5 2F	2F C7 2F DB	C7 2F DB 2F	31 5C 9F 4E

Round 3

Text	*Round Key 8*	*AddRoundKey*	*InvMixColumns*	*InvShiftRows*	*InvSubBytes*
2D 69 09 1D	78 84 72 18	55 ED 7B 05	13 0C 39 AA	13 0C 39 AA	82 81 5B 62
BB D3 71 EB	9C 2B 39 EE	27 F8 48 05	39 E2 0E 72	72 39 E2 0E	1E 5B 3B D7
70 6E 62 7E	2D 3F 43 1F	5D 51 21 61	3F FC 3B C6	3B C6 3F FC	49 C7 25 55
31 5C 9F 4E	0C 2C A1 2B	3D 62 3E 65	07 34 20 1A	34 20 1A 07	28 54 4E 38

Round 4

Text	*Round Key 7*	*AddRoundKey*	*InvMixColumns*	*InvShiftRows*	*InvSubBytes*
82 81 5B 62	F6 FC F6 6A	74 7D AD 08	58 79 C8 40	58 79 C8 40	5E A1 B1 72
1E 5B 3B D7	D6 B7 12 D7	C8 EC 29 00	0E 63 F3 B0	B0 0E 63 F3	FC D7 00 7E
49 C7 25 55	53 12 7C 5C	1A D5 59 09	2D B1 1A 3C	1A 3C 2D B1	43 6D FA 56
28 54 4E 38	0E 20 8D 8A	26 74 C3 B2	FB 9B 32 7F	9B 32 7F FB	E8 A1 6B 63

Round 5

Text	*Round Key 6*	*AddRoundKey*	*InvMixColumns*	*InvShiftRows*	*InvSubBytes*
5E A1 B1 72	10 0A 0A 9C	4E A5 BB EE	7C D6 21 5B	7C D6 21 5B	01 4A 7B 57
FC D7 00 7E	61 61 A5 C5	9D B6 A5 BB	7A 3A 4A 22	22 7A 3A 4A	94 BD A2 5C
43 6D FA 56	96 41 6E 20	D5 2C 94 76	9B 24 96 4D	96 4D 9B 24	35 65 E8 A6
E8 A1 6B 63	D0 2E AD 07	38 8F C6 64	A3 78 B1 73	78 B1 73 A3	C1 56 8F 71

Round 6

Text	*Round Key 5*	*AddRoundKey*	*InvMixColumns*	*InvShiftRows*	*InvSubBytes*
01 4A 7B 57	E0 1A 00 96	E1 50 7B C1	68 3A BB 60	68 3A BB 60	F7 A2 FE 90
94 BD A2 5C	4E 00 C4 60	DA BD 66 3C	76 F9 B0 DA	DA 76 F9 B0	7A 0F 69 FC
35 65 E8 A6	3A D7 2F 4E	0F B2 C7 E8	F7 2D B0 CE	B0 CE F7 2D	FC EC 26 FA
C1 56 8F 71	40 FE 83 AA	81 A8 0C DB	5C 19 6D BA	19 6D BA 5C	8E B3 C0 A7

Round 7

Text	*Round Key 4*	*AddRoundKey*	*InvMixColumns*	*InvShiftRows*	*InvSubBytes*
F7 A2 FE 90	B9 FA 1A 96	4E 58 E4 06	54 A8 FF DD	54 A8 FF DD	FD 2C 7D C9
7A 0F 69 FC	A1 4E C4 A4	DB 41 AD 58	0D 57 6A 95	95 0D 57 6A	AD F3 DA 58
FC EC 26 FA	9F ED F8 61	63 01 DE 9B	32 D8 91 D6	91 D6 32 D8	AC 4A A1 9F
8E B3 C0 A7	D0 BE 7D 29	5E 0D BD 8E	C3 31 2E D5	31 2E D5 C3	2E C3 B5 33

Round 8

Text	*Round Key 3*	*AddRoundKey*	*InvMixColumns*	*InvShiftRows*	*InvSubBytes*
FD 2C 7D C9	61 43 E0 8C	9C 2C 9D 45	64 E6 CF 54	64 E6 CF 54	8C F5 5F FD
AD F3 DA 58	4F EF 8A 60	E2 1C 50 38	EE 5C 6C DF	DF EE 5C 6C	EF 99 A7 B8
AC 4A A1 9F	BF 72 15 99	13 38 B4 06	1E A3 F8 2F	F8 2F 1E A3	E1 4E E9 71
2E C3 B5 33	B4 6E C3 54	9A AD 76 67	63 BC 54 B8	BC 54 B8 63	78 FD 9A 00

Round 9

Text	*Round Key 2*	*AddRoundKey*	*InvMixColumns*	*InvShiftRows*	*InvSubBytes*
8C F5 5F FD	E2 22 A3 6C	6E D7 FC 91	56 77 80 31	56 77 80 31	B9 02 3A 2E
EF 99 A7 B8	2B A0 65 EA	C4 39 C2 52	3A 02 70 B8	B8 3A 02 70	9A A2 6A E0
E1 4E E9 71	37 CD 67 8C	D6 83 8E FD	35 B7 E7 14	E7 14 35 B7	B0 9B D9 20
78 FD 9A 00	E4 DA AD 97	9C 27 37 97	B9 88 90 34	88 90 34 B9	97 96 28 DB

Round 10

Text				*Round Key 1*				*AddRoundKey*				*InvMixColumns*				*InvShiftRows*				*InvSubBytes*			
B9	02	3A	2E	93	C0	81	CF	2A	C2	BB	E1	FA	9C	7D	76	FA	9C	7D	76	14	1C	13	0F
9A	A2	6A	E0	C2	8B	C5	BF	58	29	AF	5F	D4	A2	02	FA	FA	D4	A2	02	14	19	1A	6A
B0	9B	D9	20	B7	FA	AA	EB	07	61	73	CB	D4	7B	C5	6F	C5	6F	D4	7B	07	06	19	03
97	96	28	DB	6E	3E	77	3A	F9	A8	5F	E1	76	67	82	77	67	82	77	76	0A	11	02	0F

Setelah proses ronde ke-10 selesai, hasil dari *InvSubBytes* ronde ke-10 di-*XOR*-kan dengan *cipherkey* atau kunci ronde ke-0.

14	1C	13	0F	44	53	41	4E	50	4F	52	41
14	19	1A	6A	41	49	4E	4A	55	50	54	20
07	06	19	03	54	4D	50	41	53	4B	49	42
0A	11	02	0F	41	50	49	4D	4B	41	4B	42

Langkah selanjutnya adalah mengubah hasil dari *InvSubBytes* ronde ke-10 di-*XOR*-kan dengan *cipherkey* ke dalam bentuk bilangan desimal kemudian diubah lagi ke dalam bentuk text berdasarkan kode ASCII.

50	55	53	4B	4F	50	4B	41	52	54	49	4B	41	20	42	42

PlainText

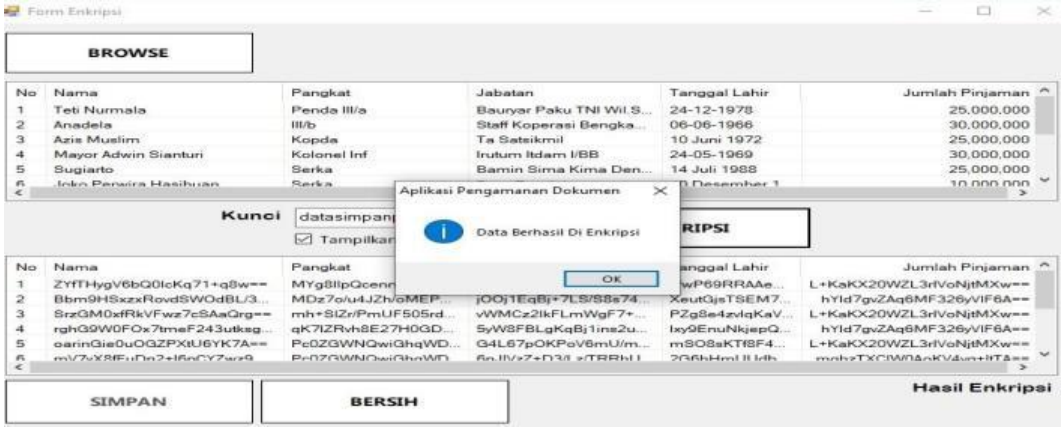
P	U	S	K	O	P	K	A	R	T	I	K	A		B	B

3.2 Implementasi dan Pengujian

Berikut ini merupakan tampilan dari implementasi sistem pengamanan dokumen simpan pinjam uang Dipusat Koperasi Kartika "A" Bukit Barisan.

1. Proses Enkripsi

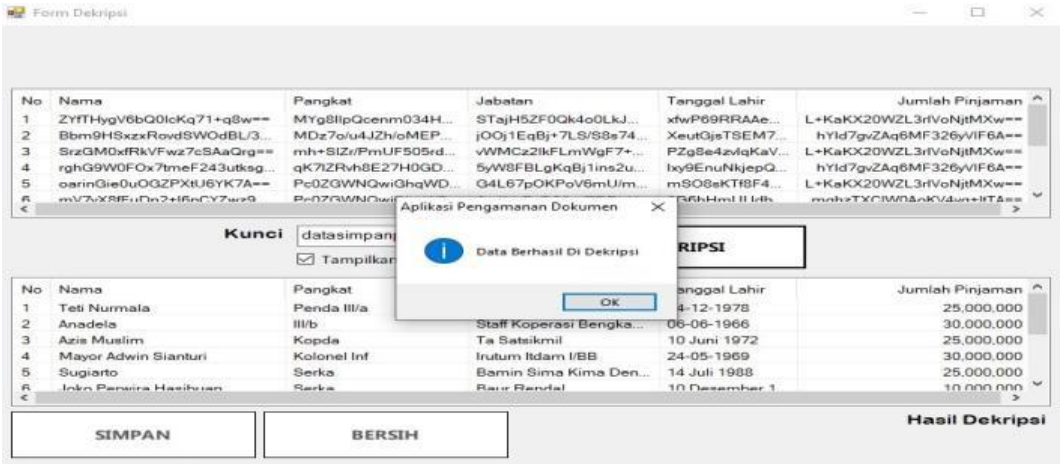
Form enkripsi berfungsi untuk mengenkripsi dokumen simpan pinjam uang dipusat koperasi kartika "A" bukit barisan. Dibawah ini merupakan tampilan *form* enkripsi, adalah sebagai berikut:



Gambar 3.1 *Form* Enkripsi

2. Proses Dekripsi

Form dekripsi berfungsi untuk mendekripsi dokumen simpan pinjam uang dipusat koperasi kartika "A" bukit barisan. Dibawah ini merupakan tampilan *form* dekripsi, adalah sebagai berikut:



Gambar 3.2 *Form* Dekripsi

4. KESIMPULAN

Adapun kesimpulan yang diperoleh dari penelitian ini adalah sebagai berikut :

1. Dengan adanya sistem pengamanan dokumen simpan pinjam uang menggunakan metode Advanced Encrryption Standart pada Pusat Koperasi Kartika "A" Bukit Barisan dapat mempermudah dan mempercepat dalam pembuatan dokumen simpan pinjam uang dan juga memberikan keamanan dalam hal penyimpanan dokumen simpan pinjam uang di database.
2. Dengan merancang sistem pengamanan dokumen simpan pinjam uang menggunakan metode Advanced Encrryption Standart pada Pusat Koperasi Kartika "A" Bukit Barisan berbasis visual basic, dapat mempermudah admin dalam melakukan pengamanan dokumen simpan pinjam uang.
3. pengamanan dokumen dalam simpan pinjam uang menggunakan metode Advanced Encrryption Standart sangat tepat dalam penerapannya sehingga dapat mengamankan dokumen dengan valid.
4. Tampilan aplikasi pengamanan dokumen dalam simpan pinjam uang menggunakan metode Advanced Encrryption Standart sangat sederhana, sehingga pengguna aplikasi dapat memahami yang menggunakannya dengan mudah.

REFERENSI

[2] A. Krisna Prastyo, "Pengamanan Data Dengan Metode Advanced Encryption Standard Dan Metode Least Significant Bit (Krisna Prastyo, Adetya, 2011)," vol. 1, p. 9, 2011.

[3] S. Informasi, U. Gunadarma, T. Informatika, F. T. Industri, and U. Gunadarma, "Implementasi Algoritma Advanced Encryption Standard (AES) Untuk Enkripsi Dan Dekripsi Pada Dokumen Teks Ana Kurniawati 1 , Muhammad Dwiky Darmawan 2 1)," Tesla, vol. 8, no. 2, pp. 1–13, 2010.

[5] 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.

[6] M. M. Amin, "Implementasi Kriptografi Klasik Pada Komunikasi Berbasis Teks," Pseudocode, vol. 3, no. 2, pp. 129–136, 2017, doi: 10.33369/pseudocode.3.2.129-136.

[10] 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," Inform. Mulawarman J. Ilm. Ilmu Komput., vol. 10, no. 1, p. 20, 2016, doi: 10.30872/jim.v10i1.23

BIOGRAFI PENULIS

Data Diri

Nama	: Rosdoana Ito Hasibuan
Tempat/Tanggal Lahir	: Situmbaga, 09 Desember 1997
Jenis Kelamin	: Perempuan
Agama	: Islam
Status	: Belum Menikah
Pendidikan Terakhir	: Sekolah Menengah Atas
Kewarganegaraan	: Indonesia
E-mail	: rosdoanahasibuan@gmail.com

Pendidikan Formal

1. Tahun 2004 - 20010 : SDN 100790 Situmbaga
2. Tahun 2010-2013 : MTS Darul Falah Langga Payung
3. Tahun 2013-2016 : SMK Swasta Raja Mas Langga Payung

Azanuddin, S.Kom., M.Kom, Beliau merupakan dosen tetap STMIK Triguna Dharma, serta aktif sebagai dosen pengajar khusus pada bidang ilmu Sistem Informasi.

Muhammad Syaifuddin, S.Kom., M.Kom, Beliau merupakan dosen tetap STMIK Triguna Dharma, serta aktif sebagai dosen pengajar khusus pada bidang ilmu Sistem Informasi.