Perancangan Pengamanan Data Menggunakan Algoritma *AES (Advanced Encyption Standard)*

Ami Aisiah Ibrahim

Abstract- Cryptography is the study of mathematical techniques in securing information or the original message (plaintext) into a hidden text (Cipher text) and then converted into the original message back. Cryptography has three important elements are key generation, encryption and description of. In cryptography known block cipher algorithm in which there are AES (Advanced encryption Standard) is part of the Modern Symmetric Key Cipher, this algorithm uses the same key during the encryption process and descriptions so that the data we have will be difficult to understand its meaning. The algorithmic techniques used to convert the data in the form of specific codes, for the purpose so that the stored information can not be read anyone except those who are eligible. Therefore, the data security system is in need to maintain the confidentiality of information in order to stay awake.

Intisari-Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika dalam mengamankan suatu informasi atau pesan asli (*Plainteks*) menjadi sebuah teks tersembunyi (*Chiperteks*) dan kemudian di ubah menjadi pesan asli kembali. Kriptografi mempunyai tiga unsur penting yaitu pembangkitan kunci, enkripsi dan deskipsi. Dalam kriptografi dikenal algoritma block chiper yang didalamnya terdapat AES (*Anvanced Encyption Standard*) merupakan bagian dari *Modern Symmetric Key Cipher*, algoritma ini menggunakan kunci yang sama pada saat proses enkripsi dan deskripsi sehingga data yang kita miliki akan sulit dimengerti maknanya. Teknik algoritma tersebut digunakan untuk mengkonversi data dalam bentuk kode-kode tertentu, untuk tujuan agar informasi yang tersimpan tidak bisa dibaca siapa pun kecuali orang-orang yang berhak. Oleh karena itu, sistem keamanan data sangat diperlukan untuk menjaga kerahasian informasi agar tetap terjaga.

Keywords : Cryptography, Algorithms, AES, Encryption, Description

I. PENDAHULUAN

A. Latar Belakang

Seiring dengan kemajuan teknologi informasi maka sangat di perlukan sebuah keamanan data terhadap kerahasiaan informasi yang saling dipertukarkan melalui jaringan internet, apa lagi jika data tersebut dalam suatu jaringan komputer yang terhubung atau terkoneksi dengan jaringan lain. Hal tersebut tentu saja menimbulkan resiko bila informasi yang sensitif dan berharga tersebut di akses oleh orang yang tidak bertangung jawab. Yang mana jika hal tersebut sampai terjadi, kemungkinan besar akan merugikan bahkan membahayakan orang yang akan mengirim pesan, maupun organisasinya. Informasi yang terkandung di dalamnya pun bisa saja berubah sehingga menyebabkan salah penafsiran oleh penerima pesan. Selain itu data yang dibajak kemungkinan rusak atau hilang yang menimbulkan kerugian material yang besar.

Oleh karena itu untuk menghindari agar hal tersebut tidak terjadi, menggunakan algoritma kriptografi *AES* untuk proses enkripsi dan deskripsi data. Kriptografi telah menjadi suatu bagian yang tidak dapat dipisahkan dari sistem keamanan jaringan, salah satu metode enkripsi data adalah *Advanced Encryption Standard* (*AES*).

B. Tujuan Penulisan

Tujuan dari penulisan ini adalah :

1. Mengenalkan konsep keamanan data pada kriptografi *AES* serta penerapannya.
2. Dapat melakukan proses Enkripsi dan Deskripsi file ataupun teks pada algoritma *AES*.
3. Dapat membuat program aplikasi komputer yang dapat melakukan proses Enkripsi dan Deskripsi sesuai dengan algoritma *AES: Rijndael*.

C. Metode Penelitian

Studi Pustaka *(Study literature)* dilaksanakan dengan cara mengumpulkan dan mempelajari segala macam informasi yang berhubungan dengan kriptografi, algoritma *AES* dan segala hal yang berhubungan dengan model pemrogramannya, bisa melalui buku dan internet.

D. Ruang Lingkup

Sesuai dengan judul tugas akhir ini, maka pembahasan akan lebih di fokuskan pada algoritma *AES* yang merupakan bagian dari simetri. Adapun yang menjadi pembatasan masalah adalah sebagai berikut :

1. Pembahasan mengenai algoritma *AES* (*Advenced Encyption Standard*).
2. Pembahasan mengenai proses penyandian enkripsi dan deskripsi data.
3. Aplikasi pengamanan data dibuat dengan menggunakan pemprograman Microsoft Visual Studio 2010.

II. LANDASAN TEORI

A. Pengamanan Data

Secara umum data dikategorikan menjadi dua, yaitu data yang bersifat rahasia dan data yang tidak bersifat rahasia. Data yang tidak bersifat rahasia biasanya tidak akan terlalu diperhatikan. Yang sangat perlu diperhatikan adalah data yang bersifat rahasia, dimana setiap informasi yang ada didalamnya akan sangat berharga bagi pihak yang membutuhkan karena

Jurusan Teknik Informatika, Sekolah Tinggi Ilmu Manejemen Informatika dan Komputer (STMIK) Antar Bangsa, Jl. HOS Cokroaminoto Blok A5 No.29-36, Ciledug Tangerang, Banten, Indonesia (Telp.021-73453000; email : 03ami.ibrahim@gmail.com)

data tersebut dapat dengan mudah digandakan. Untuk mendapatkan informasi didalamnya, biasanya dilakukan berbagai cara yang tidak sah.

B. Algoritma

"Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematik dan logis"[5]. Kata logis merupakan kata kunci dalam algoritma. Langka-langka dalam algoritma harus logis dan harus dapat ditentukan bernilai salah atau benar.

C. Kriptografi

Kriptografi merupakan ilmu sekaligus seni untuk menjaga keamanan pesan (*Cryptography is the art and science of keeping messages secure*) selain itu ada pengertian tentang kriptografi yaitu kriptografi merupakan ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, serta otentikasi. Kata "seni" di dalam definisi di atas maksudnya adalah mempunyai cara yang unik untuk merahasiakan pesan. Kata "*graphy*" di dalam "*cryptography*" itu sendiri sudah menyiratkan sebuah seni.

Untuk dapat menjalakan dengan baik pada proses kriptografi haruslah terdapat emapat elemen utama didalamnya, yang paling berkait satu sama lain, yaitu :

1. *Plain Text*
   Merupakan sebagai pesan awal atau pesan asli yang dikirim pada proses komunikasi. *Plain Text* inilah yang kemudian dienkripsi dan dideskripsi.
2. *Cipher Text*
   Merupakan pesan yang tersembunyi, yaitu pesan asli (*Plain Text*) yang telah dienkripsi pada proses kriptografi. *Cipher Text* ini dapat diubah kembali kebentuk aslinya (*Plain Text*) memanfaatkan *Key* yang telah disediakan.
3. *Cryptography Key*
   Merupakan kunci yang di gunakan untuk melakukan enkripsi dan deskripsi pada proses kriptografi. Tanpa adanya kunci (*key*) yang sama maka proses enkripsi dan deskripsi tidaka dapat dilakukan dengan baik. Kunci (*key*) merupakan informasi yang padat menjadi kendali terhadap proses terjadinya kriptografi.
4. *Encryption Decryption Algorithm*
   Komponen terakhir yang juga sama pentingnya dalam proses kriptografi adalah algoritma yang di gunakan untuk enkripsi dan dekripsi.

D. Advanced Encryption Standard (AES)

Advanced Encryption Standard (AES) merupakan algoritma *cryptographic* yang dapat digunkan untuk mengamankan data. Algoritma AES adalah blok *chipertext* simetrik yang dapat mengenkripsi (*encipher*) dan dekripsi (*decipher*) informasi. Enkripsi merubah data yang tidak dapat lagi dibaca disebut *ciphertext*, sebaliknya dekripsi adalah merubah *ciphertext* data menjadi bentuk semula yang kita kenal sebagai *plaintext*. Algoritma AES menggunakan kunci kriptografi 128, 192, dan 256 bits untuk mengenkrip dan dekripsi data.

III. HASIL DAN PEMBAHASAN

Seiring dengan kemajuan teknologi informasi yang semakin pesat maka sangat diperlukan sebuah keamanan data terhadap kerahasiaan informasi. Sangat banyak informasi yang sensitif dan berharga tersebut diakses oleh orang yang tidak bertangung jawab, kemungkinan besar akan merugikan bahkan membahayakan orang yang akan mengirim pesan, maupun penerimanya. Informasi yang terkandung di dalamnya pun bisa saja berubah sehingga menyebabkan salah penafsiran oleh penerima pesan. Selain itu data yang dibajak kemungkinan rusak atau hilang yang menimbulkan kerugian material yang besar. Oleh karena itu untuk menghindari agar hal tersebut tidak terjadi, penulis menggunakan algoritma kriptografi *AES (Advanced Encryption Standard)* untuk proses enkripsi dan deskripsi data.

A. Analisa Kebutuhan

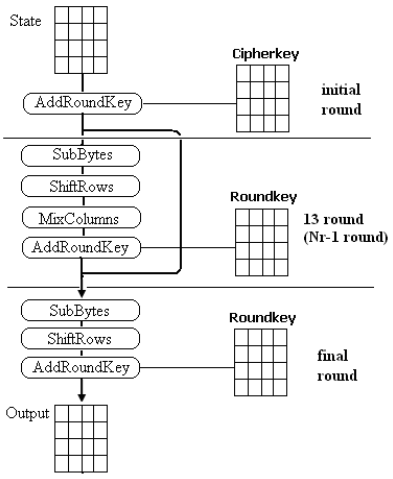
Dalam kriptografi dikenal algoritma *block chiper* yang didalamnya terdapat AES (*Anvanced Encyption Standard*) merupakan bagian dari *Modern Symmetric Key Cipher*, algoritma ini menggunakan kunci yang sama pada saat proses enkripsi dan deskripsi sehingga data yang kita miliki akan sulit dimengerti maknanya. Teknik algoritma tersebut digunakan untuk mengkonversi data dalam bentuk kode-kode tertentu, untuk tujuan agar informasi yang tersimpan tidak bisa di baca siapa pun kecuali orang-orang yang berhak. Oleh karena itu, sistem keamanan data sangat di perlukan untuk menjaga kerahasian informasi agar tetap terjaga.

Pembuatan software aplikasi sistem keamanan data menggunakan Microsoft Visual Studio 2010. Microsoft Visual Studio 2010 mengatasi semua masalah pengembangan apliakasi berbasis Windows dan memiliki fasilitas penanganan Bug yang hebat dan Real Time Backroud Compiler. Hal ini akan memberi kontribusi yang kuat dalam sistem enkripsi dan deskripsi.

B. Desain

1. Proses Enkripsi

Proses enkripsi algoritma AES terdiri dari 4 jenis transformasi bytes, yaitu *SubBytes, ShiftRows, Mixcolumns,* dan *AddRoundKey*. Pada awal proses enkripsi, *input* yang telah di*copy*kan ke dalam *state* akan mengalami transformasi byte *AddRoundKey*. Setelah itu, *state* akan mengalami transformasi *SubBytes, ShiftRows, MixColumns,* dan *AddRoundKey* secara berulang-ulang sebanyak Nr. Proses ini dalam algoritma AES disebut sebagai *round function*. *Round* yang terakhir agak berbeda dengan *round-round* sebelumnya dimana pada *round* terakhir, *state* tidak mengalami transformasi *MixColumns*. Ilustrasi proses enkripsi AES dapat digambarkan seperti pada gambar 1



Gambar 1. Diagram enkripsi AES [5]

Garis besar algoritma AES Rijndael yang beroprasi pada blok 128-bit dengan kunci 128-bit (diluar proses pembangkitan *roundkey*) adalah sebagai berikut :

1. *AddRoundKey*, melakukan XOR antara awal (*plantext*) dengan *cipher key*.
2. Putaran sebanyak Nr-1 kali. Proses yang dilakukan pada setiap putaran adalah :
   a. *SubBytes* adalah subsitusi *byte* menggunakan table subsitusi (*S-Box*).
   b. *ShiftRows* adalah pergeseran baris-baris *array state* secara *wrapping.*
   c. *MixColumns* adalah mengacak data di masing-masing kolom *array state.*
   d. *AddRoundKey* adalah melakukan XOR antara state sekarang *round key.*
3. *Final round*, proses untuk putaran terakhir :
   a. *SubBytes*
   b. *ShiftRows*
   c. *AddRoundKey*

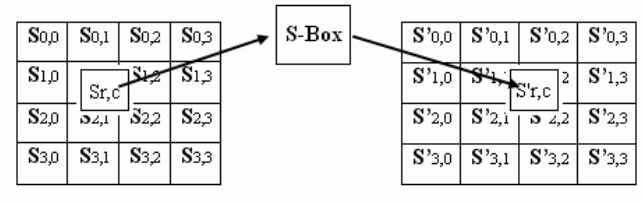
Langkah kerja enkripsi adalah sebagai berikut :

a. Transpormasi *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 Tabel 1.

TABEL 1.
S-BOX RIJNDAEL [5]

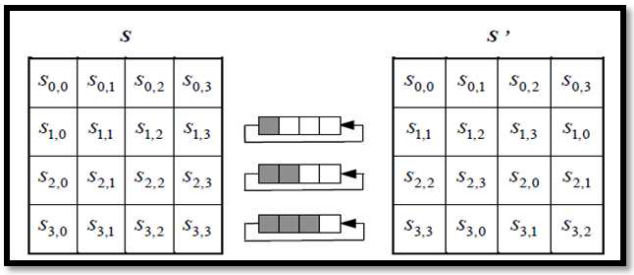
x \ y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	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

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*.



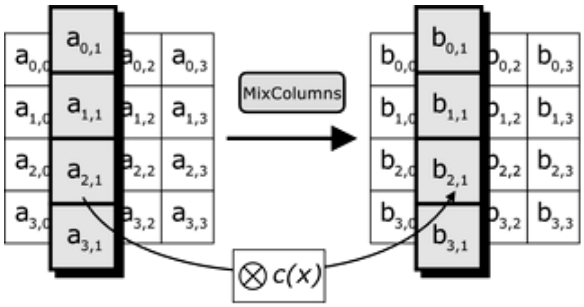
Gambar 2. Pengaruh Pemetaan pada Setiap *Byte* dalam *State* [5]

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



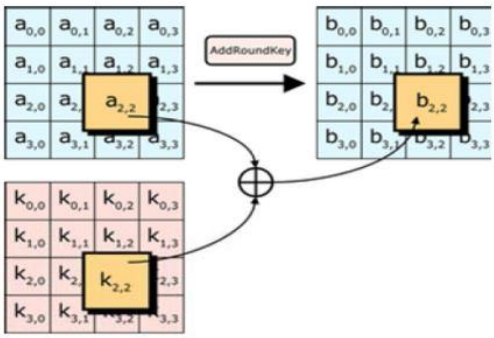
Gambar.3 Proses *ShifRows* [5]

c. MixColumns
MixColumns mengoperasikan setiap elemen yang berada dalam satu kolom pada state. Secara lebih jelas, transformasi *mixcolumns* dapat dilihat pada perkalian matriks pada gambar 4



Gambar 4. Proses *MixColumns* [5]

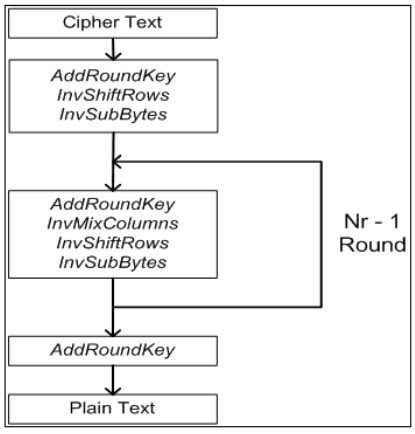
d. AddRoundKey
AddRoundKey: melakukan *XOR* antara *state* sekarang dengan *round key*.



Gambar 5. Proses *AddRoundKey* [5]

2. Proses Dekripsi

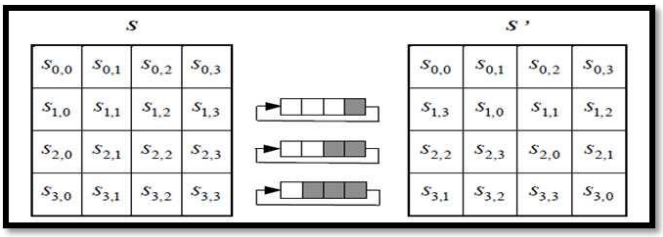
Transformasi *cipher* dapat dibalikkan dan diimplementasikan dalam arah yang berlawanan untuk menghasilkan *inverse cipher* yang mudah dipahami untuk algoritma AES. Transformasi byte yang digunakan pada *invers cipher* adalah *InvShiftRows*, *InvSubBytes*, *InvMixColumns*, dan *AddRoundKey*. Algoritma dekripsi dapat dilihat pada gambar 6



Gambar 6. Proses dekripsi [5]

1) *InvShiftRows*

InvShiftRows adalah transformasi byte yang berkebalikan dengan transformasi *ShiftRows*. Pada transformasi *InvShiftRows*, dilakukan pergeseran *bit* ke kanan sedangkan pada *ShiftRows* dilakukan pergeseran *bit* ke kiri. Ilustrasi transformasi *InvShiftRows* terdapat pada Gambar 7 :



Gambar 7. Proses *InvShiftRows* [5]

2) *InvSubBytes*

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

TABEL II
PROSES *INVSUBBYTES* [5]

Hex		Y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
X	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	17	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

3) *InvMixColumns*

Setiap kolom dalam *state* dikalikan dengan matrik perkalian dalam AES. Perkalian dalam matrik dapat dituliskan :

$$\begin{bmatrix} S'_{0,C} \\ S'_{1,C} \\ S'_{2,C} \\ S'_{3,C} \end{bmatrix} = \begin{bmatrix} 0E\ 0B\ 0D\ \theta \\ \theta\ 0E\ 0B\ 0D \\ 0D\ \theta\ 0E\ 0B \\ 0B\ 0D\ \theta\ 0E \end{bmatrix} \begin{bmatrix} S_{0,C} \\ S_{1,C} \\ S_{2,C} \\ S_{3,C} \end{bmatrix}$$

Hasil dari perkalian matriks adalah

$$S'_{0.C} = (\{0E\} \bullet S_{0,C}) \oplus (\{0B\} \bullet S_{1.C}) \oplus (\{0D\} \bullet S_{2.C}) \oplus (\{\theta\} \bullet S_{3.C})$$

$$S'_{1.C} = (\{\theta\} \bullet S_{0,C}) \oplus (\{0E\} \bullet S_{1.C}) \oplus (\{0B\} \bullet S_{2.C}) \oplus (\{0D\} \bullet S_{3.C})$$

$$S'_{2.C} = (\{0D\} \bullet S_{0,C}) \oplus (\{\theta\} \bullet S_{1.C}) \oplus (\{0E\} \bullet S_{2.C}) \oplus (\{0B\} \bullet S_{3.C})$$

$$S'_{3.C} = (\{0B\} \bullet S_{0,C}) \oplus (\{0D\} \bullet S_{1.C}) \oplus (\{\theta\} \bullet S_{2.C}) \oplus (\{0E\} \bullet S_{3.C})$$

4) *Inverse AddRounKey*

Transormasi *Inverse AddRounKey* tidak berbeda dengan transformasi *AddRoundkey* karna dalam transformasi ini hanya dilakukan oprasi penambahan sederhana dengan oprasi *bitwise* XOR.

C. *Software Architecture*

1. Proses Enkripsi

```
case"Encryption":
    {
        try
        {
            string password=textBox3.Text;
            UnicodeEncoding UE= new UnicodeEncoding();
            byte[]key=UE.GetBytes(password);
                string cryptedfile= textBox2.Text;
            FileStream FsCrypt= new FileStream(cryptedfile, FileMode.Create);
            RijndaelManaged RMCrypt= new RijndaelManaged();
            CryptoStream CS = new CryptoStream(FsCrypt,RMCrypt.CreateEncryptor(key,key),CryptoStreamMode.Write);
            FileStream FSIN = new FileStream(textBox1.Text,FileMode.Open);
            int data;
            while ((data=FSIN.ReadByte())!=-1)
                CS.WriteByte((byte)data);
            FSIN.Close();
            CS.Close();
            FsCrypt.Close();

            label2.Text = "Done";
            MessageBox.Show("the encryption operation is done correctly");


        }
        catch(Exception ex)
        {
            label2.Text="the encryption opration failed";
            MessageBox.Show("Error" + ex);
        }
        break;
```

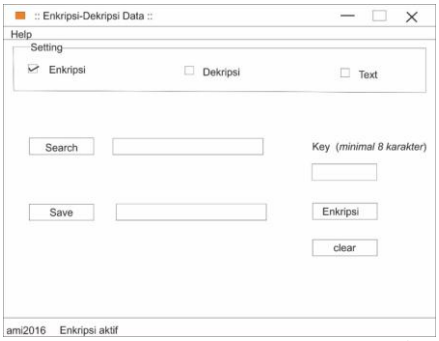
2. Proses Dekripsi

```
}
    case "Decryption":
            {
                try
                {
                    string password=textBox3.Text;
                UnicodeEncoding UE= new UnicodeEncoding();
                byte[]key=UE.GetBytes(password);
                    string cryptedfile= textBox1.Text;
                FileStream FsCrypt= new FileStream(cryptedfile, FileMode.Open);
                RijndaelManaged RMCrypt= new RijndaelManaged();
                CryptoStream CS = new CryptoStream(FsCrypt,RMCrypt.CreateDecryptor(key,key),CryptoStreamMode.Read);
                FileStream FSout = new FileStream(textBox2.Text,FileMode.Create);
                int data;
                while ((data=CS.ReadByte()) !=-1)
                    FSout.WriteByte((byte)data);
                FSout.Close();
                CS.Close();
                FsCrypt.Close();
                label2.Text = "done";
                MessageBox.Show("the decryption operation is done correctly");

                }
                catch(Exception ex)
                {
                    label2.Text="operation failed";
                    MessageBox.Show("Error"+ex);
                }

                break;
            }
```

3. *User Interface*

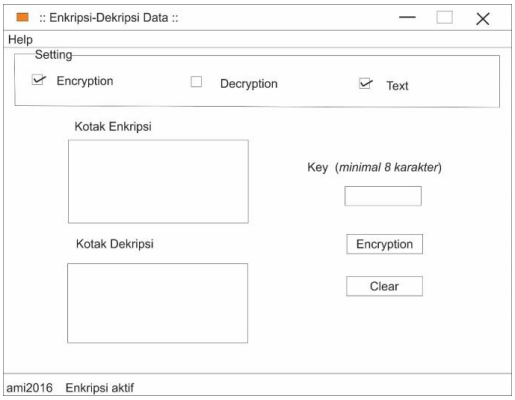
Pada Perancangan aplikasi ini terdapat dua keguanan, yaitu untuk enkripsi dan dekripsi file serta untuk enkripsi dan dekripsi text. Tampialan awal aplikasi yang dibuat seperti ini :

a) Perancangan Enkripsi dan Dekripsi File



Gambar 8. Perancangan enkripsi dan dekripsi file

b) Perancangan Enkripsi dan Dekripsi Data



Gambar 9. Perancangan enkripsi dekripsi teks

4. *Black Box Testing*

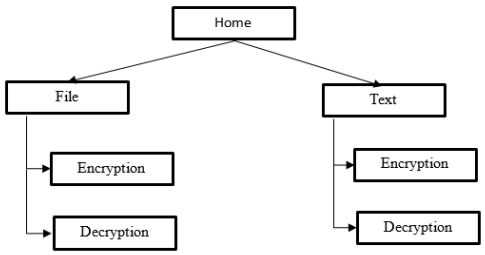
TABEL.III
PENGUJIAN ENKRIPSI FILE DAN DEKRIPSI FILE TEKS

No	Skenario Pengjian	*Tase Case*	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Mengklik *CheckBox* Enkripsi dan mengkosongkan *TexsBox searching, save* dan *key*		Sistem akan menolak dan menampilkan pesan "Diharap mengisi semua konten"		Valid
2	Mengklik *CheckBox* Dekripsi dan mengkosongkan *TexsBox searching, save* dan *key*		Sistem akan menolak enkripsi dan menampilkan pesan "Diharap mengisi semua konten"		Valid
3	*Searching* file notepad (txt), dan *save* dengan format txt, mengisi *key* 8 digit. Lalu klik *Button Encryption.*		Sistem akan mengenkripsi dan otomatis file hasil tersimpan difolder yang sudah ditentukan. Kemudian muncul pesan "Berhasil".		Valid
4	Tidak mengisi kotak save		Sistem menolak dan muncul pesan "Diharap mengisi kotak save"		Valid

No	Skenario Pengjian	*Tase Case*	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Mengklik *CheckBox Text, CheckBox* Enkripsi dan mengkosongkan semua konten		Sistem akan menolak dan muncul pesan "Diharap mengisi semua konten"		Valid
2	Mengisi kotak enkripsi dan key 8 karakter		Sistem berjalan dan muncul hasil serta pesan "Berhasil".		Valid
3	Mengisi kotak enkripsi tapi menginput key kurang 8 karakter		Sistem akan menolak dan muncul pesan "Key lemah, isi 8 karakter atau lebih"		Valid
4	Mengklik *CheckBox Text, CheckBox* Dekripsi dan tidak mengisi		Sistem akan menolak dan muncul pesan "diharap mengisi		Valid

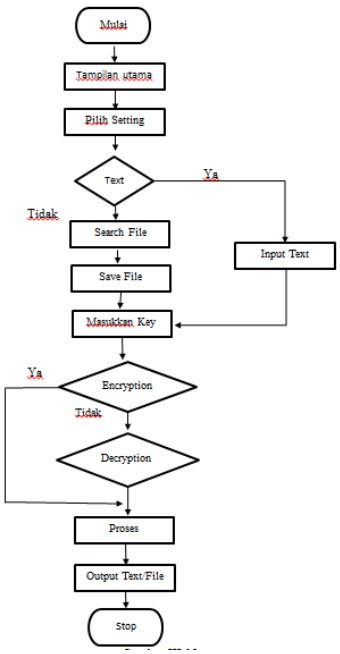
D. Perancangan Struktur Program

Adapun struktur program dalam aplikasi keaman data enkripsi dan dekripsi adalah sebagai berikut :



Gambar 10. Struktur program menu utama

E. Perancangan FlowChat Program

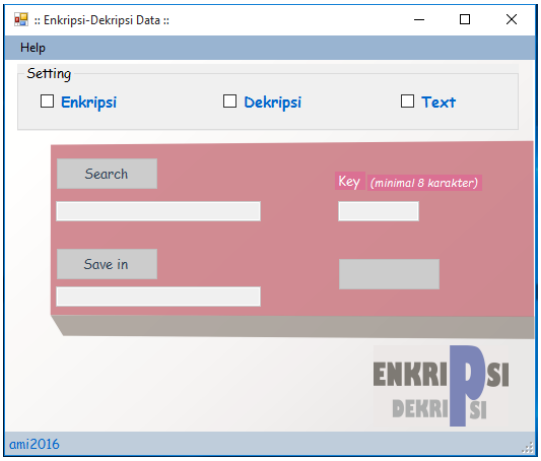


Gambar 11 FlowChat Aplikasi Enkripsi dekripsi

F. Implementasi

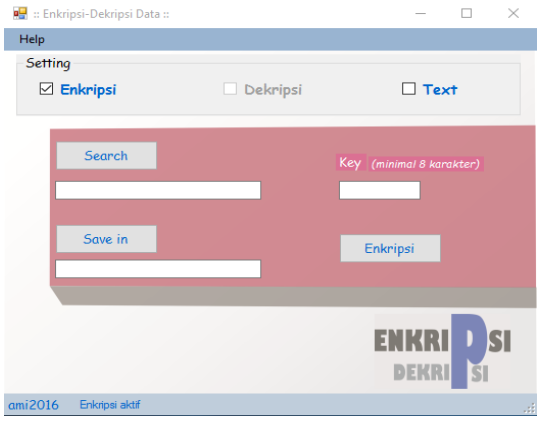
Implementasi algorotma AES (*Advanced Encryption Standard*) Rijdael dibuat Menggunakan Microsoft Visual Studio 2010. Aplikasi yang terdiri dari enkripsi dekripsi text dan enkripsi dekripsi file.

1) Implementasi *Encryption File*

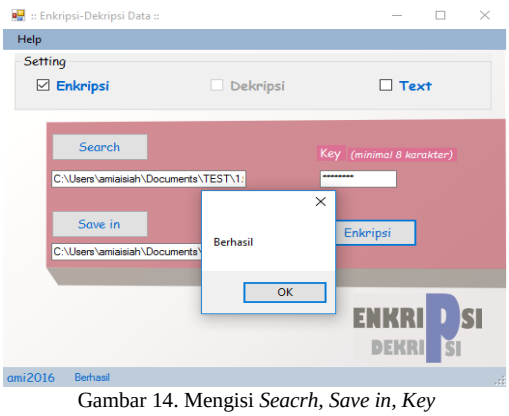


Gambar 12. Tampialan utama aplikasi enkripsi dekripsi

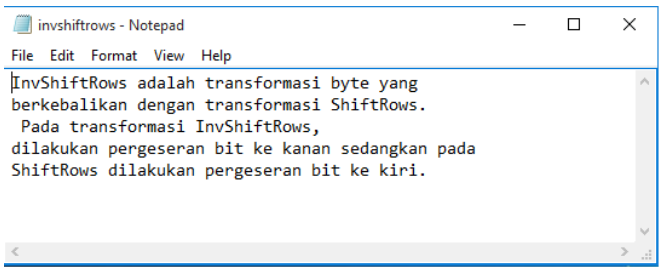
Pada tampilan utama aplikasi terdapat *setting* yang terdiri dari *encryption*, *decryption*, dan *text.* Serta difungsikan langsung untuk enkripsi atau dekripsi file.



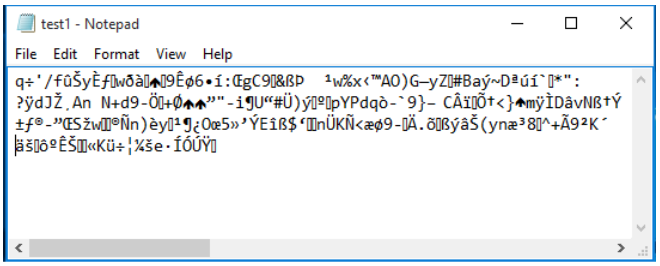
Gambar 13. Pilih *Setting Encryption*



Gambar 14. Mengisi *Seacrh, Save in, Key*



Gambar 15. File yang akan dienkripsi

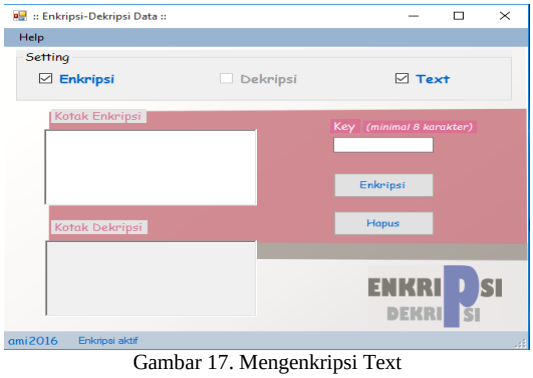


Gambar 16. Hasil Enkripsi

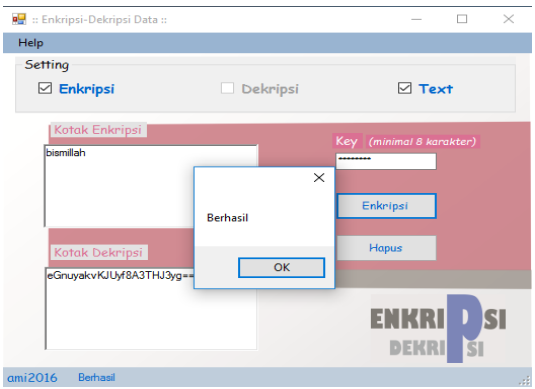
Pada enkripsi dekripsi file, file yang akan dienkripsi ialah file-file yang terdapat dalam sistem komputer seperti file Microsoft Office, file MP3, file PDF, dan file gambar. File yang telah dienkripsi akan berekstensi *aes* dan untuk melihat hasil dari enkripsi file tersebut dapat dilihat menggunakan *notepad.*

2) Implementasi Text

a. *Encryption Text*

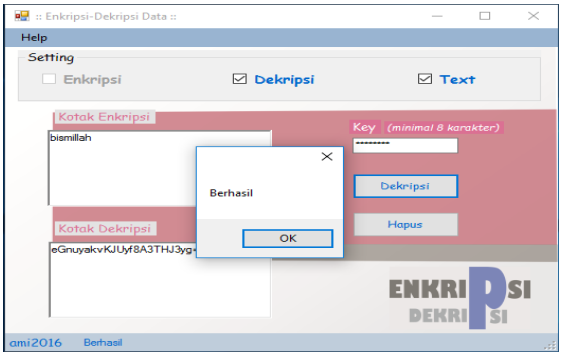


Gambar 17. Mengenkripsi Text



Gambar 18. Enkripsi Text berhasil

b. *Decryption Text*



Gambar 19. Dekripsi Text

Pada enkripsi file dan teks jika terjadi perubahan kunci, maka hasilnya tidak akan terbaca dan dingggap file tersebut rusak. Jika terjadi perbedaan, maka disimpulkan ada modifikasi terhadap isi pesan. Aplikasi ini didasarkan pada kenyataan bahwa perubahan kunci akanmenyebabkan perubahan data pada saat akan di kembalikan pada bentuk semula atau didekripsi dan hasilnya akan berbeda dengan aslinya (plainteks). Dengan kata lain, *Rijndael* sangat peka terhadap perubahan sekecil apa pun pada data masukan.

IV. PENUTUP

A. Kesimpulan

Pembuatan aplikasi sistem pengamanan data enkripsi dekripsi file dan teks menggunakan Microsoft Visual Studio 2010 mendukung perkembangan zaman yang semakin canggih. Pemilihan algoritma harus yang tepat salah satunya adalah algoritma AES: Rijndael, karena algoritma ini merupakan algoritma yang cukup sulit dipecahkan saat ini, sebab belum ada serangan atau pemecahan yang belum mampu secara analisis matematika dengan efektif dan efesien dengan alasan pola yang dibentuk cukup acak.

AES: Rijndael memiliki keunggulan karena memiliki daya memori dan kecepatan komputasi dalam pengoprasian. Pengoprasian yang tidak memakan memori yang telalu besar ini banyak diminati pasar karena kebutuhan efesiensi waktu yang relatif cepat.

B. Saran

1. Pembaca yang ingin melakukan enkripsi dan dekripsi baik file ataupun teks perlu mempertimbangkan factor kelebihan maupun kekurangan dari suatu algiritma yang akan digunakan agar file dan teks terjaga keamanan dan kerahasiannya.
2. Diharapkan kepada pembaca yang melakukan proses enkripsi dan dekripsi file ataupun teks untuk menggunkan kunci yang lebih variatif atara huruf dan anggka ataupun karakter yang akan lebih menyulitkan pemecahan cipherteks oleh kriptanalis.
3. Dalam mengimplementasinya algoritma Rijndael ini hanya dalam cakupan kecil dan sangat mendasar menggunakan Bahasa pemprograman Microsoft Visual Studio 2010. Penyusun berharap agar pembaca dapat mengembangkannya.

REFERENSI

[1] Anhar. 2010. Cara Mudah Mengamankan Data Komputer dan Laptop. Jakarta: Media Kita.

[2] Fachrurozi, Muhammad Farid. 2006. Enkripsi Pesan Rahasia Menggunakan Algoritma (*Advandes Encryption Standard*) AES: Rijndael. Jakarta: Universitas Islam Negri Syarif Hidayatullah Jakarta.

[3] Hasanaji, Mohammad. 2016. Pengertian dan Sejarah Microsoft Visual Studio 2010. Diambil dari : http://www.dasarpendidikan.co.id/2014/07/pengertian-keistimewaan-dan-sejarah-microsoft-visual-studio-2010.html

[4] Ibrahim, Rohmat Nur. 2012. Kriptografi Algoritma DES, AES/Rijndael, Blowfish Untuk Keamanan Citra DigitalDengan Menggunakan Metode Discrete Wavelete Transformasion (DWT). ISSN: 2442-4943. Bandung: Jurnal Computech and Bisnis, Vol. 6, No.2, Desember 2012, 82-95.

[5] Munir, Rinaldi. 2006. *Kriptografi*. Bandung: Informatika Bandung.

[6] Pratama, I Putu Agus Eka. 2014. Handbook Jaringan Komputer. Bandung: Informatika.

[7] Primartha, Rifkie. 2013. Penerapan Enkripsi dan Dekripsi File Menggunakan Algoritma *Advens Encryption Standard* (AES). ISSN: 2301-8488. Palembang: *Journal of Research in Computer Science and Applications* – Vol. 2, No. 1, Januari 2013: 13-18.

[8] Rahmayunita, Isnawaty, Sutardi. 2015. Penyadapan SMS dan GPS Berbasis Android Menggunakan Algoritma *Advanced Encryption Standard* (AES). ISSN: 2460-1446. Kendari. SemanTIK, Vol. 1, No.2, Juli-Desember 2015, pp. 11-22.

[9] Riadi, Muchlisin. 2014. Pengertian, Sejarah dan Jenis Kriptografi. Diambil dari: http://www.kajianpustaka.com/2014/01/pengertian-sejarah-dan-jenis-kriptografi.html?m=1

[10] Satria, Eko. 2009. Studi Algoritma Rijndael dalam Sistem Keamanan Data. Sumatra Utara: USU Repository.

[11] Setyaningsih, Emy. 2015. Kriptografi dan Implementasinya Menggunakan MATLAB. Yogyakarta: Andi.

[12] Sianturi, Fricles Ariwisanto. 2013. Perancangan Aplikasi Pengamanan Data Dengan Kriptografi *Advances Encryption Standard* (AES). ISSN: 2301-9425. Medan: Pelita Informatika Budi Darma, Vol. 4, No. 1, Agustus 2013: 42-46.

[13] Sutejo, Budi dan Michael. 2004. Algoritma dan Teknik Pemprograman. Yogyakarta: Andi.

[14] Tjandra, Suhatati dan C Pickerling. 2015. Aplikasi Metode-Metode *Software Testing* Pada *Configuration, Compatibility* dan *Usability* Perangkat Lunak. ISSN : 2089-1121. Surabaya: Seminar Nasional "Inovasi dalam Desain dan Teknologi", 267-273.

Ami Aisiah Ibrahim, lahir di Serang, 12 Mei 1993, Lulus S1 Sistem Informasi tahun 2016 di STMIK Antar Bangsa, Ciledug Tangerang. Saat ini aktif sebagai Staf di Daarul Qur"an Full Day.