KOMPUTASI, Vol ,5 No. 9. 2007, 36 - 37

RANCANGAN DAN IMPLEMENTASI SISTEM PENGAMANAN DATA DENGAN ALGORITMA RIJNDAEL

Sri Setyaningsih [1)], Sena Ramadona Cakrawijaya [1)]

1) Program Studi Ilmu Komputer

I. PENDAHULUAN

1.1. Latar Belakang

Dalam komunikasi data, suatu metode pengamanan data dikenal dengan kriptografi (*Cryptography*). Kriptografi adalah seni dan ilmu untuk menjaga kemanan pesan (Yusuf Kurniawan,2004). Kriptografi terdiri dari berbagai macam sistem sandi (*Cryptosystem*) yang memiliki algoritma, tujuan penggunaan dan tingkat kerahasiaan berbeda. Dalam prakteknya, menentukan algoritma kriptografi yang digunakan menjadi suatu masalah tersendiri, di sisi lain user menginginkan kemudahan baik itu dari sisi kerahasiaan, ketepatan, kecepatan maupun biaya yang murah.

Kenyataan di lapangan, proses penanganan data dengan menggunakan metode kriptografi seringkali membutuhkan waktu yang relatif lebih lama dibandingkan tanpa proses kriptografi. Untuk itu perlu diciptakan suatu sistem sandi yang relatif cepat dalam proses penanganan data tanpa mengabaikan kaidah kerahasiaan yang ingin dicapai.

Pengamanan data tidak hanya sebatas mengupayakan agar data tersebut tidak dibaca oleh pihak yang tidak berkepentingan, tetapi juga bagaimana agar data tersebut tidak dapat dimanipulasi atau dimodifikasi, sehingga dibutuhkan suatu cara agar diperoleh otentikasi yang meyakinkan terhadap data yang dikirimkan/disimpan. Pemilihan teknik kriptografi yang sesuai dengan kebutuhan menjadi hal penting yang harus dipertimbangkan.

Pengamanan Pengamanan file (data) menggunakan metode algoritma Rijndael dilakukan dengan memberikan masukan (input) file yang akan dienkripsi. Kemudian hasilnya (output) adalah file dengan jenis data (ekstensi) yang berbeda karena file tersebut telah terenkripsi tidak dapat ditulis atau dibaca sebelum diubah kembali (dekripsi) dengan metode algoritma yang sama.Dengan adanya proses enkripsi-dekripsi file, sebuah file diubah kebentuk chipertext yang tidak dapat dibaca sebelum diubah bentuknya kembali ke dalam bentuk plaintext. Proses enkripsi dilakukan dengan algoritma Rijndael. Penggunaan algoritma Rijndael karena memiliki fleksibilitas platform dan tingkat keamanan yang sangat baik.

Sistem pengamanan data menggunakan metode Rijndael ini dapat dikembangkan menggunakan bahasa pemrograman Visual Basic 6.0 karena kemampuan kompabilitasnya yang baik dengan sistem operasi Windows dan juga sangat efisien dalam perancangan kode-kode pemrograman.

1.2. Tujuan Penelitian

Tujuan penelitian ini adalah untuk

merancang dan mengimplementasikan Sistem Pengamanan Data Dengan Algoritma Rijndael.

1.3. Ruang Lingkup

Sistem yang dibuat memiliki ruang lingkup sebagai berikut :

1. Berjalan pada sistem operasi windows xp.
2. Menggunakan ukuran blok data 128 bit.
3. Menggunakan ukuran blok kunci 128 bit.
4. Data yang dienkripsi/diamankan adalah file dokumen, file gambar, file audio dan file video.

II. METODOLOGI PENELITIAN

2.1. Kerangka Pemikiran

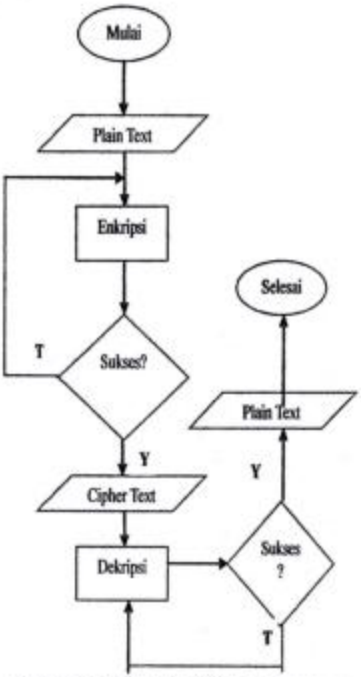
Ada beberapa faktor yang sering menjadi pertimbangan dalam memilih suatu metode enkripsi yang tepat, yaitu kecepatan enkripsi, sumber daya yang dibutuhkan (memori, kecepatan PC), ukuran file hasil enkripsi, besarnya dan kompleksitas algoritma. Alasan digunakannya Algoritma Rijndael Penggunaan algoritma Rijndael karena memiliki fleksibilitas platform dan tingkat keamanan yang sangat baik. Rijndael adalah algoritma yang kuat terhadap berbagai serangan yang umum diketahui seperti serangan kriptanalisis (Aulia Rahman, 2006).

2.3. Tahapan Pelaksanaan Penelitian

Tahapan pembuatan sistem ini berdasarkan metode SDLC (System Development Life Cycle), metode ini memiliki lima fase (Gambar 1).

Gambar 1. Diagram System Depelovment Life Cycle

Penelitian diawali dengan analisis, kemudian tahapan perancangan secara umum mengenai sistem yang akan dibuat, menggunakan metode Data Flow Diagram (DFD), Flowchart Sistem, perancangan ini menggambarkan sistem yang aka dibuat. Tahapan secara rinci mengenai sistem yang akan dibuat, menggunakan metode rancangan Form (*UserInterface*).



Gambar 2. Diagram Alir Perancangan Program

Sebagai contoh, file dokumen dengan ekstensi txt :
Plaintext : 32 43 f6 a8 88 5a 30 8d 31 31 98 a2 e0 37 07 34
Kunci : 2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c

Maka akan dihasilkan chipertext atau file yang telah terenkripsi dengan ekstensi file berbeda.

Ciphertext : 39 25 84 1d 02 dc 09 fb dc 11 85 97 19 6a 0b 32

File terenkripsi (*chipertext*) didapat setelah melalui proses operasi : **SubBytes, ShiftRows, MixColums, AddRoundkey.**

Setelah melakukan tahapan perancangan atau desain, sistem dapat dibuat dengan menggunakan Visual Basic 6.0 IDE sebagai editor untuk membuat antar muka dan juga sebagai editor untuk kode program. Kegiatan penelitian dilanjutkan dengan uji coba sistem dan penggunaan sistem.

III. PERANCANGAN DAN IMPLEMENTASI

3.1 Analisis Masalah

Pengembangan Sistem Pengamanan Data sangat dibutuhkan bagi berbagai kalangan terutama instansi dan perusahaan yang sangat peduli akan keamanan data mereka. Oleh karena itu pemilihan algoritma Rijndael sangat tepat karena memiliki keseimbangan antar keamanan dan kemudahan dalam penggunaannya. Proses kerja dalam pengamanan data secara umum terlihat pada Gambar 3.

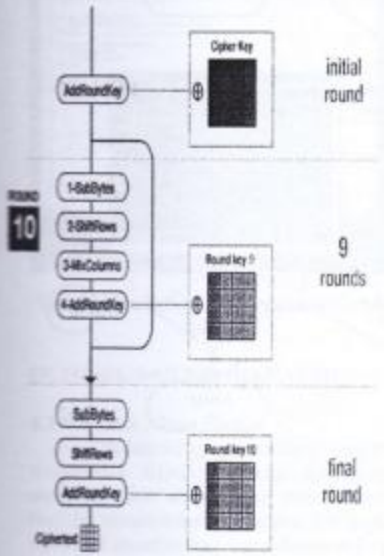
Gambar 3. Sistem Pengamanan Data

3.2 Perancangan Sistem

3.2.1 Pengacakan File dengan Algoritma Rijndael (Enkripsi)

Secara umum, proses enkripsi dilakukan dengan initial round yaitu

melakukan XOR antara state awal yang masih berupa plain text dengan cipher key. Kemudian melakukan keempat proses diatas sebanyak 9 kali putaran, dan terakhir adalah final round yang melibatkan proses sub bytes, shift rows, dan add round key. Adapun proses enkripsi Algoritma Rijndael ditampilkan pada Gambar 4.



Gambar 4. Proses Enkripsi Algoritma Rijndael

Proses Enkripsi file dilakukan melalui tahapan transformasi SubBytes(), ShiftRows(), MixColumns(), dan AddRoundKey(). Untuk lebih jelas dapat dilihat pseudo code pada Gambar 5.

```
Cipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
begin
byte state[4,Nb]
state = in
AddRoundKey(state, w[0, Nb-1])
for round = 1 step 1 to Nr-1
SubBytes(state)
ShiftRows(state)
MixColumns(state)
AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
end for
SubBytes(state)
ShiftRows(state)
AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])
out = state
end
```

Gambar 5. Pseudo code Enkripsi Rijndael

3.2.2 Pengembalian File kebentuk Normal (Dekripsi)

Untuk mengubah file cipher text ke bentuk semula yaitu plain text, dilakukan transformasi Inverse yaitu : InvShiftRows(), InvSubBytes(), InvMixColumns(), dan AddRoundKey().

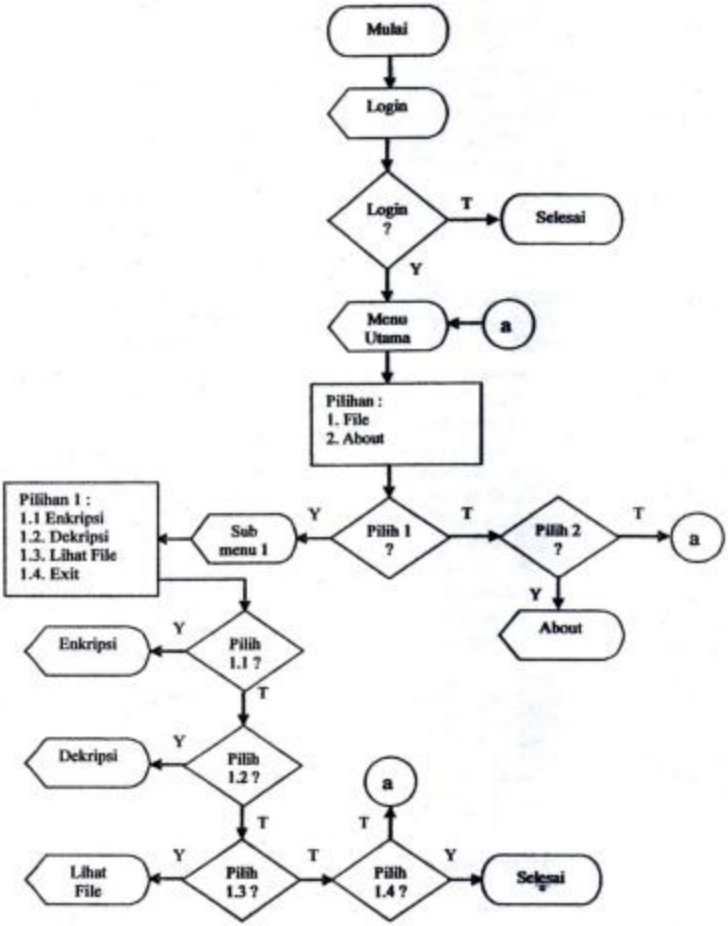
Untuk lebih jelas dapat dilihat pseudo code Gambar 6.

```
InvCipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
begin
byte state[4,Nb]
state = in
AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])
for round = Nr-1 step -1 downto 1
InvShiftRows(state)
InvSubBytes(state)
AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
InvMixColumns(state)
end for
InvShiftRows(state)
InvSubBytes(state)
AddRoundKey(state, w[0, Nb-1])
out = state
end
```

Gambar 6. Pseudo code Dekripsi Rijndael

3.2.3 Perancangan Sistem Secara Umum

Rancangan sistem secara umum dapat diwakili melalui Flowchart Sistem seperti disajikan pada Gambar 7.



Gambar 7. *Flowchart* Sistem Pengamanan Data Dengan

3.3 Implementasi

Sistem Pengamanan Data Dengan Algoritma Rijndael diimplementasikan dengan menggunakan bahasa pemrograman Visual Basic 6.0 dengan salah satu hasil implementasi disajikan pada Gambar 8.

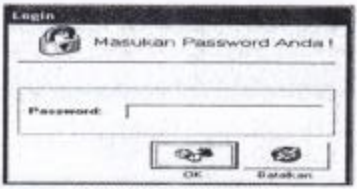
Gambar 8. Tampilan Pembuatan *Form Splash*

IV. HASIL DAN PEMBAHASAN

4.1. Hak Akses Sistem

Sistem Pengamanan Data Dengan Algoritma Rijndael yang dibangun memiliki *form* utama yang memberikan fasilitas utama dalam mengamankan data. *Form* ini memberikan kemudahan dalam pengoperasiannya, sedangkan untuk memasuki *form* utama harus melewati bagian *form login*, pengguna diwajibkan mengisi *password* pada form login untuk memasuki form *utama*.

Form Login digunakan untuk mengamankan sistem pengamanan data karena diwajibkan mengisi *password* pengguna untuk dapat melanjutkan ke *form* utama. Form Login ditampilkan pada Gambar 9.



Gambar 9. Form Login

Pada saat memasukan password ke *form login*, akan mencocokan input *password* ke dalam kode sumber yang digunakan yaitu :

```
.....
Private Sub cmdBatal_Click()
Unload frmLogin
End Sub
Private Sub cmdOK_Click()

InputPassword = txt_pass
Password = "xxxx"
If InputPassword = Password Then
fRijndael.Show
  Unload frmLogin
Else
MsgBox ("Password Anda Salah!")
End If
.....
```

4.2 Proses Enkripsi Data

Proses enkripsi data terdapat pada *form* utama yang berfungsi untuk mengambil *input* data dari *user*/pengguna. *Form* ini terdiri dari *form* input "kata kunci", tombol enkripsi untuk melalukan proses pengamanan data dan tombol dekripsi untuk melalukan pengembalian file ke bentuk normal. Selain itu terdapat form *about* yang berisi tentang informasi Sistem Keamanan Data.

4.2.1 *Input* Data

Sebelum melakukan enkripsi, terlebih dahulu memasukan kunci enkripsi

agar dapat melakukan proses algoritma enkripsi. Setelah itu masukan data penting yang akan dienkripsi pada menu pilihan yang tersedia.

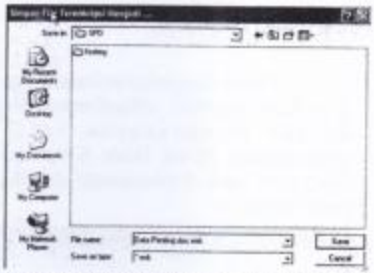
Gambar 9. *Form* Utama

Pada saat tombol Enkripsi ditekan, maka akan terbuka jendela baru untuk melakukan pemilihan file data yang akan diamankan seperti diperlihatkan pada Gambar 10.

Gambar 10. *Form* Pemilihan Data *plain text*

4.2.2 *Output* Data

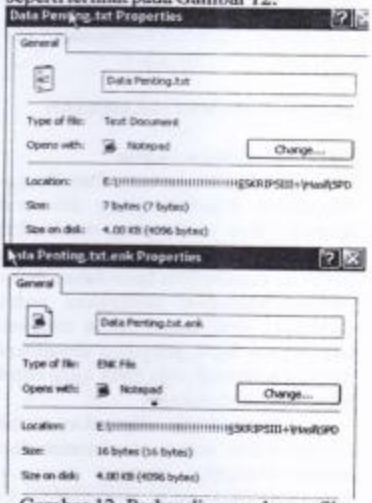
Setelah itu tentukan nama file yang baru terhadap file data yang telah diamankan tersebut. Pada saat tombol *save* ditekan, maka akan dilakukan proses enkripsi dan menyimpan data yang telah terenkripsi sesuai nama file yang telah kita



Gambar 11. *Form* Pemberian nama data *cipher text*

4.2.3 Melihat Informasi File

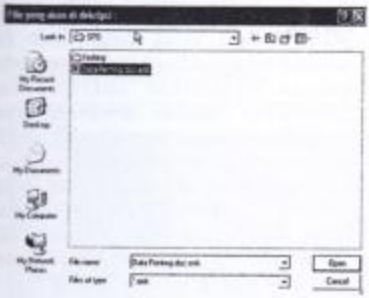
File yang telah dienkripsi telah menjadi chiper text, bila dibandingakan dengan file awal yang berupa plain text, ternyata ukuran kedua file tidak berubah. Ukuran file dilihat dari "size on disk", seperti terlihat pada Gambar 12.



Gambar 12. Perbandingan ukuran file *plain text* dan *chiper text*

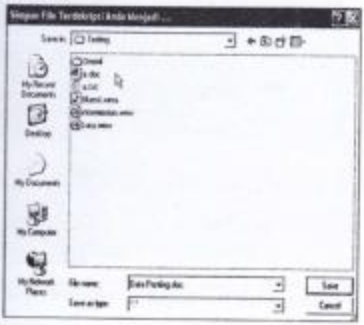
4.3 Proses Dekripsi Data

Untuk mengembalikan file data kebentuk semula digunakan tombol dekripsi, kemudian dipilih file data yang terenkripsi yang akan dikembalikan ke kondisi *plain text* seperti pada Gambar 13.



Gambar 13. *Form* pemilihan file data *chiper text*

Setelah itu tentukan nama file yang baru terhadap file data yang telah dideksripsi dan telah menjadi *plain text* kembali.

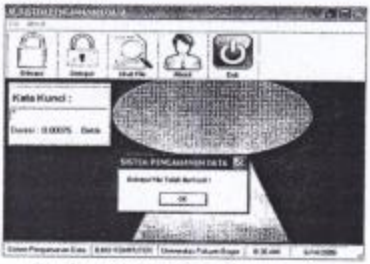


Gambar 14. *Form* Pemberian nama data *plain text*

Pada saat tombol *save* ditekan, maka akan dilakukan proses dekripsi dan menyimpan data yang telah terdekripsi sesuai nama file yang telah kita tentukan.

4.4 *Running Time*

Pada Sistem Pengamanan Data Dengan Algoritma Rijndael dapat dilihat waktu proses enkripsi maupun dekripsi file. Informasi ini ditampilkan untuk mengetahui pengaruh perbedaan besarnya file terhadap waktu proses yang dibutuhkan.



Gambar 15. Tampilan *Running Time*

4.5 Uji Coba Sistem

Uji coba sistem dilakukan setelah pembuatan modul-modul sistem selesai dengan percobaan pada komputer user *interface*. Dengan melakukan uji coba ini dapat diketahui kekurangan sistem yang telah dibuat sebelumnya, diantaranya : apakah sistem yang dibuat sesuai dengan perancangan sistem yang dirancang, apakah penanganan berfungsi dengan baik. Berikut ini adalah beberapa pengujian yang dilakukan terhadap sistem yang dibuat.

Uji coba ini bertujuan untuk membandingkan antara data yang telah dienkripsi dengan data sebelum dienkripsi, dengan contoh input : "a" dan kata kunci

yang sama yaitu : "a" pada format file *txt*.Salah satu tampilan File sebelum dan setelah dienkripsi disajikan pada Gambar 16. Adapun Gambar 17 menunjukkan file bersangkutan setelah dideskripsi. Ilustrasi tersebut menunjukkan bahwa Sistem Pengamanan Data dengan Algoritma Rijndael yang dibangun sukses digunakan, dengan kondisi file setelah dideskripsi sama seperti file awal.

Gambar 16. Isi file txt sebelum (*plain text*) dan sesudah dienkripsi (*cipher text*)

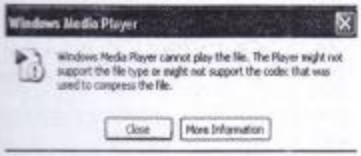
File setelah didekripsi kembali :

Gambar 17. Isi file txt sesetelah didekripsi kembali

Uji coba dilakukan juga pada file audio, video dan gambar. Contoh input file berupa file audio dipilih secara acak adalah "Blues1.wma" dan kata kunci yaitu : "penting" pada format file *wma*. Contoh input file berupa file video, dan file yang dipilih : "Lucu.wmv dan kata kunci yaitu : "penting" pada format file *wmv*. Contoh input file gambar: "Gambar.jpg dan kata kunci yaitu : "penting" pada format file *jpg*. Uji coba untuk ketiga jenis file tersebut menunjukkan bahwa sistem telah sukses melakukan enkripsi dan dekripsi.File sebelum dienkripsi dapat dimainkan dengan normal oleh media player untuk file audio dan video, berturut-turut seperti terlihat pada Gambar 18 dan Gambar 21, kondisi terkunci setelah dienkripsi (Gambar 19 dan Gambarn 22), dan kondisi file normal kembali setelah didekripsi (Gambar 20 dan Gambar 23).

Gambar 18. File Blues1.wma dapat dimainkan dengan baik

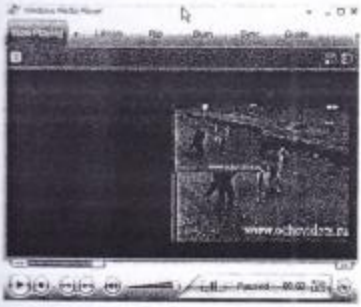
File setelah dienkripsi :



Gambar 19. File Blues1.wma tidak dapat dimainkan

File setelah didekripsi kembali :

Gambar 20. File Blues1.wma dapat kembali dimainkan



Gambar 21. File Lucu.wmv dapat dimainkan dengan baik

File setelah dienkripsi :

Gambar 22. File Lucu.wmv tidak dapat dimainkan

File setelah didekripsi kembali

Gambar 23. File Lucu.wmv dapat kembali dimainkan:

File sebelum dienkripsi file dapat dilihat dengan normal oleh media gambar (Gambar 24)

Gambar 24. File Gambar.jpg dapat dilihat

File setelah dienkripsi :

Gambar 25. File Gambar.jpg tidak dapat dilihat

File setelah didekripsi kembali :

Gambar 26. File Gambar.jpg dapat kembali dilihat

4.5.1 Uji Coba Struktural

Uji coba struktural adalah uji coba yang dilakukan pada saat pembuatan sistem dan memastikan kinerja dari sistem yang dibuat. Uji coba ini dilakukan dengan cara menjalankan setiap *form* atau menu yang telah dirancang. Jika terjadi kesalahan atau tidak berfungsi, maka proses akan kembali ke tahap implementasi. Hal ini dilakukan berulang, sampai didapat hasil yang diinginkan. Hasil uji coba struktural ditampilkan pada Tabel 1.

Tabel 1. Uji Coba Struktural Menu Utama

No.	Sistem	Hasil	Keterangan
1.	Menu *Login*	Tampil	Dijalankan dari *file .exe*
2.	Menu Utama	Tampil	Tampil setelah *Login* berhasil
3.	Enkripsi	Tampil	Dijalankan dari menu utama
4.	Dekripsi	Tampil	Dijalankan dari menu utama
5.	Lihat File	Tampil	Dijalankan dari menu utama
6.	About	Tampil	Dijalankan dari menu utama

4.5.2 Uji Coba Fungsional

Setelah memasukan password yang benar pada menu **login**, maka akan memasuki *form* utama yang berfungsi mengamankan file data dengan proses enkripsi (diamankan) dengan cara menekan tombol *Enkripsi* kemudian mengembalikan data kebentuk semula dengan menekan tombol dekripsi.

Tabel 10. Uji Coba Fungsional Menu Utama

No.	Form	Button	Sub Menu	Hasil
1.	Menu Login	OK	-	Berfungsi
2.	Menu Login	Batalkan	-	Berfungsi
3.	Menu Utama	File	Enkripsi	Berfungsi
4.	Menu Utama	File	Dekripsi	Berfungsi
5.	Menu Utama	File	Lihat File	Berfungsi
6.	Menu Utama	File	Exit	Berfungsi
7.	Menu Utama	About	-	Berfungsi
8.	Menu Utama	Enkripsi	-	Berfungsi
9.	Menu Utama	Dekripsi	-	Berfungsi

4.5.3 Validasi

Uji coba validasi adalah uji coba yang dimaksudkan untuk menguji kebenaran dari aplikasi yang telah dirancang dengan menggunakan Algoritma Rijndael. Uji coba ini akan melakukan pengecekan terhadap file file yang akan dienkripsi dan didekripsi kembali.

V. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Pemilihan penggunaan algoritma dalam sistem pengamanan data yang tepat dapat meningkatkan tingkat keamanan. Pemilihan algoritma Rijndael tepat karena sangat sulit untuk dipecahkan tanpa mengetahui kunci yang benar. Selain itu algoritma Rijndael juga mudah diimplementasikan kedalam berbagai perangkat, baik *software* maupun *hardware*.

Sistem Pengamanan Data Dengan Algoritma Rijndael dapat digunakan sebagai sarana pengamanan data atau file yang handal dikarenakan memakai algoritma yang kerahasiaan kuncinya belum terpecahkan. Kerahasiaan kunci menjadi faktor penting keamanan data karena algoritma termasuk kedalam kriptographi kunci simetri.

Hasil serangkaian uji coba, menunjukkan file yang tidak berubah dalam ukuran data, sehingga tidak menimbulkan masalah tempat penyimpanan data. Waktu proses enkripsi dan dekripsi sangat singkat sehingga tidak penyulitkan pengguna Sistem Pengamanan Data Dengan Algoritma Rijndael.

5.2. Saran

Sistem Pengamanan Data Dengan Algoritma Rijndael ini masih dapat disempurnakan dengan melakukan penyembunyian kunci atau disebut juga *data hiding*, dengan begitu pengguna tidak perlu memasukan kata kunci atau *password* berulang kali pada saat proses dekripsi. Algoritma Rijndael juga dapat digunakan dalam bidang lain, misalnya keamanan jaringan dan keamanan data berbasis *hardware* atau mikrokontroler.

DAFTAR PUSTAKA

Hendra, ST, 4 Maret 2006. *Dasar Pemrograman Visual Basic*, http://www.indoprog.com

Islab, Mei 2008. Rijndael. Http://islab.oregonstate.edu

Kurniawan, Yusuf, 2004, Kriptografi Keamanan Internet dan Jaringan Telekomunikasi. Informatika, Bandung.

Kusumo, 2003 *Visual Basic 6.0*, Maxikom, Palembang

Rahman, Aulia. 15 April 2006, Studi Blok Chiper Serpent dan Rijndael, http://www.informatika.org

Wikipedia, 20 Mei 2008. *Algoritma*. Http://en.wikipedia.org/wiki/algoritma.htm.

Wikipedia, 20 Mei 2008. *AES*. http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

Wikipedia, 20 Mei 2008. *Encrypt*. Http://en.wikipedia.org/wiki/Encr