RANCANG BANGUN PROGRAM KRIPTOGRAFI ADVANCED ENCRYPTION STANDARD

Sumi Khairani[1]
Fhery Agustin[2]
Ananda Fahmi[3]

sumi@potensi-utama.ac.id, fhery@potensi-utama.ac.id, fahmi@potensi-utama.ac.id

ABSTRAKSI

Untuk menjamin keamanan dan keutuhan dari suatu data, dibutuhkan suatu proses penyandian. Enkripsi dilakukan ketika data akan dikirim. Proses ini akan mengubah suatu data asal menjadi data rahasia yang tidak dapat dibaca. Sementara itu, proses dekripsi dilakukan oleh penerima data yang dikirim tersebut. Data rahasia yang diterima akan diubah menjadi data asal. Dengan cara penyandian, data asli tidak akan terbaca oleh pihak yang tidak berkepentingan, melainkan hanya oleh penerima yang memiliki kunci dekripsi. AES digunakan sebagai standard algoritma kriptograpi yang terbaru. Algoritma sebelumnya dianggap tidak mampu lagi untuk menjawab tantangan perkembangan teknologi komunikasi yang sangat cepat. AES sendiri adalah algoritma kriptografi dengan menggunakan algoritma Rijndael yang dapat mengenkripsikan block data sepanjang 128 bit dengan panjang kunci 128 bit, 192 bit dan 256 bit.

Kata kunci : AES, Algoritma Rijndael, Dekripsi, Enkripsi, Kriptografi

PENDAHULUAN

Kriptografi berbasis pada algoritma pengkodean data informasi yang mendukung kebutuhan dari dua aspek keamanan informasi, yaitu *secrecy* (perlindungan terhadap kerahasiaan data informasi) dan *authenticity* (perlindungan terhadap pemalsuan dan pengubahan informasi yang tidak diinginkan.

Keamanan dan kerahasiaan data merupakan salah satu aspek yang sangat penting pada sistem informasi saat ini. Hal ini disebabkan pesatnya perkembangan ilmu pengetahuan dan teknologi yang memungkinkan munculnya suatu teknik-teknik yang baru yang disalahgunakan oleh pihak-pihak tertentu yang mengancam keamanan dari sistem informasi tersebut. Ironisnya, teknik yang digunakan untuk mengancam keamanan data selalu setingkat lebih maju daripada teknik yang digunakan untuk mengamankan

1. **Dosen Program Studi Teknik Informatika, STMIK Potensi Utama**
   Jl. K.L. Yos Sudarso Km. 6,5 No. 3A Medan, Telp. (061) 6640525
2. **Dosen Program Studi Sistem Informasi, STMIK Potensi Utama**
   Jl. K.L. Yos Sudarso Km. 6,5 No. 3A Medan, Telp. (061) 6640525
3. **Mahasiswa Program Studi Teknik Informatika, STMIK Potensi Utama**
   Jl. K.L. Yos Sudarso Km. 6,5 No. 3A Medan, Telp. (061) 6640525

data. Karena itu timbul suatu gagasan yang mengacu kepada permasalahan tersebut, yakni untuk membuat suatu sistem keamanan yang dapat melindungi data yang dianggap penting dengan cara menyandikan data sehingga sulit untuk dideteksi oleh pihak yang tidak berhak. Untuk keperluan tersebut, maka diperlukan teknik kriptografi dengan metode enkripsi dan dekripsi.

Salah satu faktor yang menyebabkan suatu teknik kriptografi menjadi lebih sering digunakan adalah tingkat kerahasiaan untuk mengamankan data yang tinggi disertai dengan kemudahan penggunaannya.

ANALISA

AES didesain berdasarkan *wide trail strategy* yang dicetuskan pendesain Rijndael dan Daemen, dalam disertasinya. Strategi ini mengusulkan agar cipher terdiri dari tiga komponen utama yaitu pencampuran kunci, transformasi tidak linear dan transformasi linear. Pencampuran kunci bertujuan agar keamanan algoritma tidak terletak pada dirahasiakannya algoritma, melainkan pada kerahasiaan kunci. Transformasi nonlinear bertujuan agar bila diketahui keluaran, maka tidak dapat diketahui masukannya. Hal ini dapat dilakukan dengan *S-Box*. Transformasi linear bertujuan agar sebanyak mungkin transfomasi nonlinear yang aktif. Dengan memisahkan transformasi linear dengan non linear, diharapkan kita dapat mendesain transformasi non linear terbebas dari transformasi linear dan sebaliknya. Daemen menekankan betapa perlunya desain transformasi linear yang baik. Analisis sandi pada DES menunjukkan betapa buruknya transformasi linear pada DES (permutasi) sehingga tidak dapat menahan analisis sandi linear (ASL) dan analisis sandi diferensial (ASD). Pada awal dan akhir cipher, diberikan operasi *Pre-whitening* dan *post-whitening*, yang berupa XOR plaintext /ciphertext dengan subkey. Operasi ini bertujuan meningkatkan keamanan seperti halnya pada DES-X. Sedangkan pada awal dan akhir DES hanya terdapat permutasi yang tidak meningkatkan keamanan.

Kotak-S *(S-Box)*

Kotak-S pada AES didesain dengan rumusan matematika untuk menghilangkan kecurigaan akan ditanamnya backdoor pada kotak-S. Penggunaan inversi x-1 pada GF(28) dikarenakan ketahanan operasi ini terhadap analisis sandi linear dan diferensial. Maksimum peluang propagasi diferensialnya DPmaks = 2-6 dan korelasi linear maksimumnya LPmaks = 2-3. Meskipun inversi ini sudah mengamankan AES dari analisis sandi diferensial (ASD) dan linear (ASL), namun karena kesederhanaannya, maka dikuatirkan kotak-S ini mudah diserang dengan *algebra attack*, khususnya *interpolation attack*. Karena itulah ditambahkan transformasi linear yang memiliki sifat tidak mempengaruhi ketahanan terhadap ASD dan ASL, namun menghilangkan kesempatan

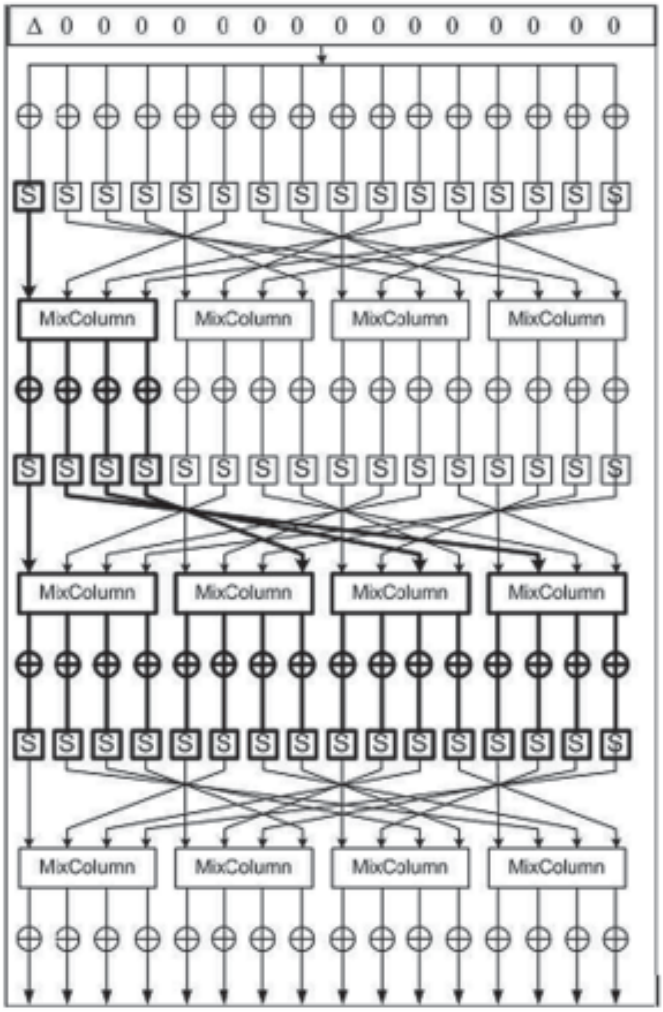
penyerang untuk mengeksploitasi kesederhanaan aljabar AES. Karena itu dikalikanlah dengan matrik L dalam GF(2). Ketidakkompatibelan operasi antara GF(28) dan GF(2) ini mempersulit serangan aljabar terhadap AES. Kemudian, untuk menghindari pemetaan 0 ke 0, maka ditambahkan konstanta c. Pemetaan 0 ke 0 dalam kotak-S pada DES membuat DES rentan terhadap serangan ASD dan ASL. Dalam ASD, pemetaan semacam ini memudahkan penyerang mendapatkan karakteristik iteratif sehingga DES dapat dipecahkan. Kotak-S juga didesain agar memungkinkan operasi berjalan secara paralel penuh. Pada setiap ronde, ke-16 kotak-S AES dapat dioperasikan bersamaan, khususnya pada perangkat keras, sehingga operasi dapat berjalan dengan sangat cepat. AES hanya memiliki satu macam kotak-S (*S-Box)*, sehingga menghemat jumlah gerbang yang diperlukan pada perangkat keras. Kotak-S ini juga invertible sehingga inversinya dapat digunakan pada proses dekripsi.

Berikut ini adalah ringkasan kriteria kotak-S :

a. Invertibility (untuk dekripsi)
b. Minimisasi korelasi antara kombinasi linear bit-bit masukan dan kombinasi linear bit-bit keluaran (menahan ASL)
c. Minimisasi nilai terbesar pada tabel XOR (menahan ASD)
d. Kompleksitas ekspresi aljabar pada GF(28)
e. Kesederhanaan deskripsi (mudah analisisnya)

Operasi Mixcolumns

Bersama dengan operasi *ShiftRows*, *MixColumn* merupakan transformasi linear yang bertujuan untuk menyebarkan pengaruh transformasi nonlinear ke sebanyak mungkin komponen nonlinear di ronde selanjutnya. Bila shiftRows bertujuan menyebarkan pada arah baris, maka *MixColumn* bertujuan menyebarkan ke arah kolom. Dengan perpaduan dua operasi ini, diperolehlah difusi yang sangat baik. Dalam MixColumn diperkenalkan konsep jumlah cabang (yang dicetuskan Daemen dalam disertasinya) â untuk matrik M. Bila jumlah koefisien tidak-nol dalam vektor a dinyatakan dengan wb(a), maka untuk a ‘“ b â = min { wb (a Å b) + wb (Ma Å Mb) }Matrik MixColumn M memiliki â = 5. Artinya bahwa beda tidak nol dalam satu byte akan disebarkan paling sedikit ke beda tidak nol dalam empat byte, seperti terlihat pada gambar 1, terlihat bahwa ASD dan ASL untuk 4 ronde ke atas akan melibatkan sedikitnya 21 kotak-S aktif, sehingga DPmaks = (2-6)21 = 2-126 dan LPmaks = 2(21-1)(2-3)21 = 2202-63 = 2-43.



Gambar 1. Analisis Sandi Diferensial dan Linier AES

Ini mengindikasikan bahwa jumlah pasang plaintext yang diperlukan untuk ASD minimal sebanyak 2126 dan untuk ASL sebanyak (½-43)2 = 286. Sedangkan jumlah plaintext yang mungkin sebanyak 2-128. Artinya, 4 ronde AES
akan kebal terhadap ASD dan ASL, sedangkan telah diketahui bahwa jumlah minimal ronde AES adalah 10 ronde. Bila diperhitungkan pula *differential* dan *linear hull* diperkirakan AES 5 ronde akan mampu menghadapinya.

Kriteria Ekspansi Kunci

Subkey pada tiap ronde dapat diperoleh dari rumus : W[i] = W[i-6] Å W[i-1], W[6i] = W[6i-6] Å f(W[6i-1]) di mana f() merupakan fungsi penggunaan kotak-S *(S-Box)* dan penambahan konstanta ronde. Penjadwalan kunci dapat diimplementasikan tanpa eksplisit menggunakan array W. Jika jumlah yang tersedia kecil, maka kunci per ronde dapat dihitung *on-the-fly* dan hanya membutuhkan buffer sebesar 64 byte. Rekursi ekspansi kunci bersifat invertible. Ini berarti bahwa mengetahui 4 word (64 byte) berurutan dari kunci terekspansi akan dapat membangkitkan seluruh tabel subkey. Meskipun demikian, mengetahui sebagian bit subkey (kurang dari 4 word) tidak akan mengijinkan mengetahui bit-bit subkey atau *key* yang lain. Ekspansi kunci sederhana dan efisien untuk banyak prosesor. Konstanta per ronde menghilangkan sifat simetri. Substitusi pada ekspansi kunci memberikan ketidaklinearan sehingga dapat menghindari *related-key attack*

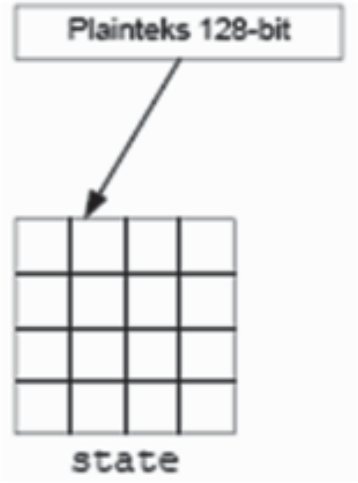
Analisis Desain AES

Pengelompokkan jenis AES ini adalah berdasarkan panjang kunci yang digunakan. Angka-angka di belakang kata AES menggambarkan panjang kunci yang digunakan pada tipa-tiap AES. Selain itu, hal yang membedakan dari masing-masing AES ini adalah banyaknya round yang dipakai. AES-128 menggunakan 10 round, AES-192 sebanyak 12 round, dan AES-256 sebanyak 14 round.AES memiliki ukuran block yang tetap sepanjang 128 bit dan ukuran kunci sepanjang 128, 192, atau 256 bit.

Algoritma Rijndael mempunyai 3 parameter yaitu

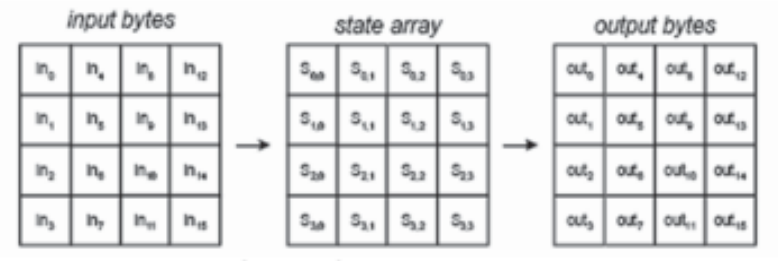
1. Plaintext : *array* yang berukuran 16-*byte*, yang berisi data masukan.
2. Ciphertext : *array* yang berukuran 16-*byte*, yang berisi hasil enkripsi.
3. Key : *array* yang berukuran 16-*byte*, yang berisi kunciciphering (disebut juga *cipher key*).

Dengan 16 *byte*, maka baik blok data dan kunci yang berukuran 128-bit dapat disimpan di dalam ketiga *array* tersebut (128 = 16 x 8). Selama kalkulasi plainteks menjadi cipherteks, status sekarang dari data disimpan di dalam *array of bytes* dua dimensi, state, yang berukuran NROWS ´ NCOLS. Untuk blok data 128-bit, ukuran state adalah 4 ´ 4. Elemen *array* state diacu sebagai $S[r,c]$, dengan $0 £ r < 4$ dan $0 £ c < Nb$ (*Nb* adalah panjang blok dibagi 32. Pada *AES*- 128, Nb = 128/32 = 4).



Gambar 2. State Awal

Pada awal enkripsi, 16-*byte* data masukan, *in*0, *in*1, ..., *in*15 disalin ke dalam *array* state (direalisasikan oleh fungsi CopyPlaintextToState(state, plaintext)) seperti diilustrasikan sebagai berikut:



Gambar 3. Alur State Array

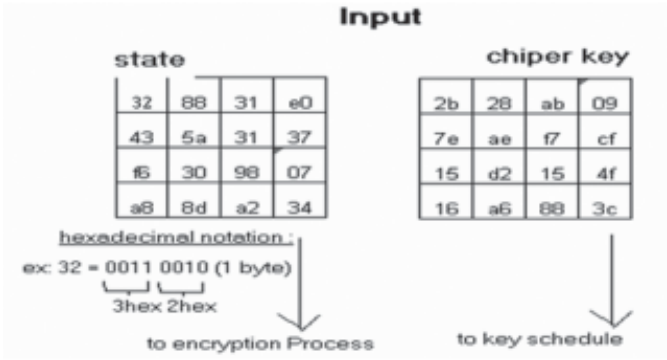
Operasi enkripsi/dekripsi dilakukan terhadap *array S*, dan keluarannya ditampung didalam array *out*. Skema penyalinan array masukan *in* ke *array S*: *S*[*r*, *c*] ¬ *in*[*r* + 4*c*] untuk 0 £ *r* < 4 dan 0 £ *c* < *Nb* Skema penyalinan *array S* ke *array* keluaran *out*: *out*[*r*+4*c*] ¬ *S*[*r*, *c*] untuk 0 £ *r* < 4 dan 0 £ *c* < *Nb.*

Berdasarkan ukuran block yang tetap, AES bekerja pada matriks berukuran 4x4 di mana tiap-tiap sel matriks terdiri atas 1 byte (8 bit). Berikut adalah pemahaman algoritma enkripsi AES-128 menurut Rinaldi Munir (2004 : 5), berdasarkan garis besar Algoritma *Rijndael* yang beroperasi pada blok 128-bit dengan kunci 128-bit adalah sebagai berikut:

1. Siapkan array berukuran 4x4 bernama kunci.
2. Siapkan array berukuran 4x4 bernama state
3. Cetak :"Masukkan 16 bilangan heksadesimal sebagai kunci."
4. Simpan 16 nilai tesebut sebagai nilai dari masing-masing elemen array kunci.
5. Cetak : "Masukkan teks yang akan dienkripsikan."
6. Konversikan teks tersebut kedalam bentuk bit menggunakan kode ASCII.
7. Konversikan kode ASCII tersebut dalam heksadesimal.
8. Kelompokkan bit-bit teks tersebut menjadi 128 bit tiap bagiannya.
9. Ambil 128 bit pertama untuk diproses.
10. Kelompokkan bit teks tersebut menjadi 16 bagian dengan 8 bit tiap bagiannya.
11. Masukkan tiap-tiap bagian eks tersebut ke dalam tiap-tiap sel pada mariks berukuran 4x4.
12. konversikan bit kedalam heksadesimal.
13. Lakukan langkah AddRoundKey
14. Lakukan langkah Sub Bytes
15. Lakukan langkah ShiftRows
16. Lakukan langkah MixColumns
17. Lakukan langkah AddRoundKey
18. Ulangi langkah 13 -16 sebanyak 9 kali.
19. Jika langkah 17 sudah dilakukan, maka lakukan langkah SubBytes
20. Lakukan langkah ShiftRows
21. Lakukan langkah AddRoundKey.
22. Selesai.

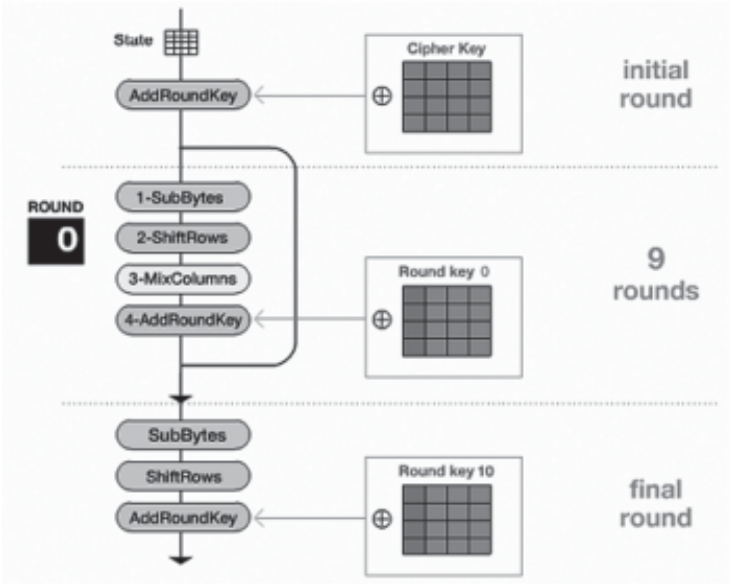
Penjelasannya :

Berikut ini adalah contoh penerapan proses enkripsi AES lihat. Pada rounds pertama *plaintext* dan *key* yang bernotasi *hexadecimal* di XOR adala sebagai berikut:



Gambar 4. Elemen state dan kunci dalam notasi HEX

Misal, teks sudah diolah sedemikian rupa sehingga matriks (array) berukuran 4 x 4 dan begitu pula kata kunci.



Gambar 5. Diagram Proses Enkripsi

Kemudian akan dilakukan 4 transformasi sebagai berikut sebanyak 9 kali :

1. SubBytes.
2. ShiftRows.
3. MixColumns.
4. AddRoundKey

Setelah itu, untuk round ke 10 dilakukan 3 transformasi sebagai berikut :

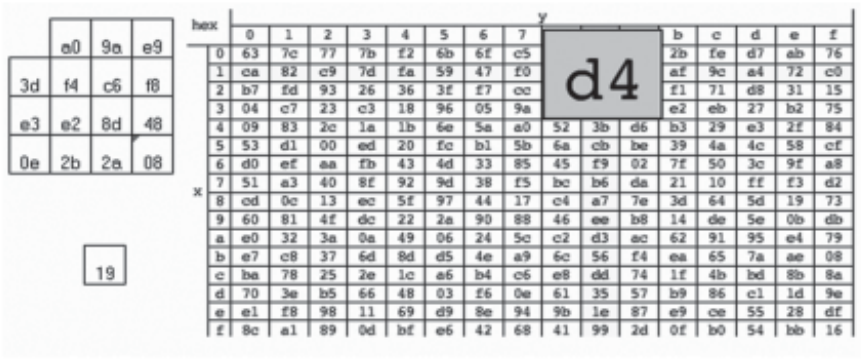
1. SubBytes.
2. ShiftRows.
3. AddRoundKey.

Berikut adalah penjelasan dari Proses SubBytes, ShiftRows, MixColumns dan AddRoundKey adalah :

1. Transformasi *SubBytes*

Transformasi *SubBytes* memetakan setiap byte dari *array state* dengan menggunakan tabel substiusi *S-box*. *AES* hanya mempunyai satu buah *S-box*. Tabel *S-box* yang digunakan adalah:

Tabel III.1 Substitusi (*S-Box*)



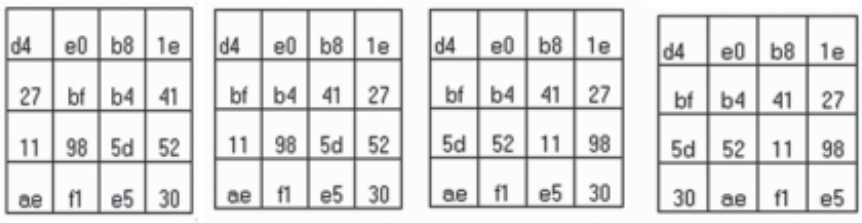
Gambar 6. Transformasi SubBytes

Cara pensubstitusian adalah sebagai berikut: 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 *S-box* yang merupakan perpotongan baris *x* dengan kolom *y*. Misalnya *S*[0, 0] = 19, maka S'[0, 0] = d4.

2. Transformasi *ShiftRows*

ShiftRows bertujuan menyebarkan pada arah baris Transformasi *ShiftRows* melakukan pergeseran secara *wrapping* (siklik) pada 3 baris terakhir dari array state. Jumlah pergeseran bergantung pada nilai baris (r). Baris *r* = 1 digeser sejauh 1 *byte*, baris r = 2 digeser sejauh 2 *byte*, dan baris *r* = 3 digeser sejauh 3 *byte*. Baris *r* = 0 tidak digeser.

1. Transformasi *ShiftRows()* melakukan pergeseran secara *wrapping* (siklik) pada 3 baris terakhir dari *array state*.
2. Jumlah pergeseran bergantung pada nilai baris (*r*). Baris *r* = 1 digeser sejauh 1 *byte*, baris *r* = 2 digeser sejauh 2 *byte*, dan baris *r* = 3 digeser sejauh 3 *byte*. Baris *r* = 0 tidak digeser. (lihat Gambar 6)



Gambar 7. State awal ,rotate 1, rotate2 dan rotate 3

3. Transformasi *MixColumn*

MixColumn merupakan transformasi linear yang bertujuan untuk menyebarkan pengaruh transformasi nonlinear ke sebanyak mungkin komponen nonlinear di ronde selanjutnya, dan menyebarkan ke arah kolom.

1. Transformasi *MixColumns()* mengalikan setiap kolom dari *array state* dengan polinom *a*(*x*) mod (*x*4 + 1).
2. Setiap kolom diperlakukan sebagai polinom 4-suku pada GF(28).
3. *a*(x) yang ditetapkan adalah:
   a(*x*) = {03}*x*3 + {01}*x*2 + {01}*x* + {02}
   Transformasi ini dinyatakan sebagai perkalian matriks:

$$s'(x) = a(x) \otimes s(x)$$

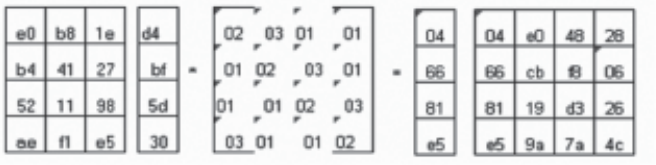
$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

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

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

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

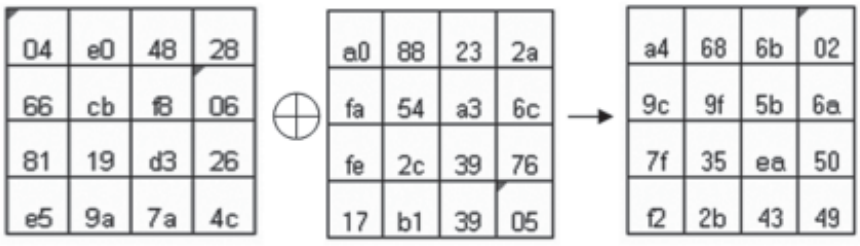
$$s'_{3,c} = (\{03\} \bullet s_{0,c}) \oplus s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{3,c})$$



Gambar 8. Proses mix columns

4. Transformasi *AddRoundKey*

Pencampuran kunci (*AddRoundKey*) bertujuan agar keamanan algoritma tidak terletak pada dirahasiakannya algoritma, melainkan pada kerahasiaan kunci. Transformasi ini melakukan operasi XOR terhadap sebuah *round key* dengan *array state*, dan hasilnya disimpan di *array state*. Contoh AddRoundKey adalah :



Gambar 9. Proses add round keys

Kolom ke-1 mariks state ang telah melalui proses SubBytes dan ShiftRows di – XOR kan dengan kolom ke-1 RoundKey Begitu pula dengan kolom ke-2, ke-3 dan ke-4, kemudian akan dilanjutkan sampai RoundKey 10. Berikut adalah hasil dari AddRoundKey Ke-1 :

a4	68	6b	02
9c	9f	5b	6a
7f	35	ea	50
f2	2b	43	49

Gambar 10. Hasil AddRoundKey ke-1

Berikut ini adalah tampilan *rounds* selanjutnya setelah penjelasan dari round pertama,dimulai dari round ke dua sampai kesepuluh untuk satu buah mariks state, teks 128 bit. Round terakhir tidak melibatkan pengubahan dengan cara mix columns.

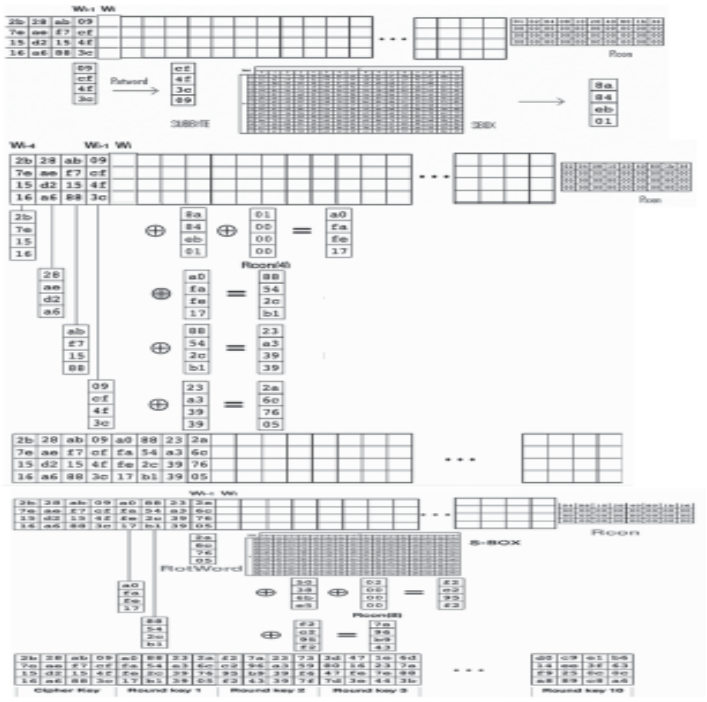
Gambar 11. Proses round kedua sampai kesepuluh

5. Ekspansi Kunci

Algoritma AES melaksanakan *cipher key* dan membuat suatu ekspansi kunci untuk menghasilkan suatu *key schedule*. Ekspansi kunci yang diperlukan AES Nb(Nr+1), word, sehingga bias digunakan AES 128 bit maka,4(10+1) = 40 word =44 x 32 bit = 1408 bit sub key. Ekspansi dari 128 menjadi 1408 bit sub key. Proses ini disebut dengan key schedule. Subkey ini diperlukan karena setiap round merupakan suatu inisial dari Nb word untuk Nr = 0 dan 2 Nb untuk Nr = 1,3 untuk Nr = 2,....., yang berisi array linier empat byte word (Wi),0 = i Nb (Nr+1). Contoh :

a. RotWord () mengambil input empat-byte word [a0, a1,a2, a3]dan membentuk cyclic permutasi seperti [a1,a2,a,3,a0].
b. Sub word () mengambil input empat-byte word dan menggunakan S-Box sehingga didapat empat byte procedure output word.
c. Rcon () menghasilkan round yang tetap dari word array dan berisi nilai yang diberikan oleh [xid1,{00},{00},{00}] dengan xid1 dari i ke 1.

Pada AES memiliki sistem penjadwalan kunci sehingga pada setiap perputaran atau rounds kunci selalu berubah- ubah, menurut Rinaldi Munir (2004 : 16) dalam *www.ilmukomputer.org* seperti pada gambar 12. Penjadwalan kunci sebagai berikut :



Gambar 12. Penjadwalan kunci

Dalam algoritma AES ukuran teks yang dapat dienkripsikan senilai 8192 karakter. Karena dalam AES 128 memiliki panjang kunci (*Nr word*) = 4. dengan catatan 1 *word* = 32 bit atau 4 word = 128 bit, dan maksimal block adalah 512 block. Maka 512 block x 128 bit = 65536 bit. Sehingga dapat dituliskan 8192 karakter = 8192 byte = 65536 bit, maka dalam batasan masalah algoritma program untuk ukuran teks maka hanya dapat memproses atau merahasiakan data sebesar 8192 karakter. Berikut adalah penggalan program untuk membatasi ukuran teks adalah sebagai berikut :

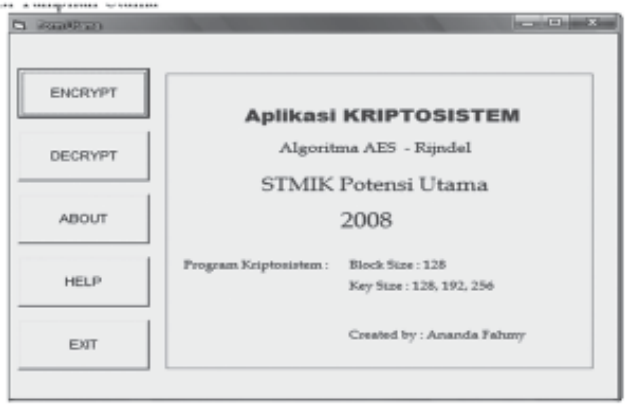
```
Private Sub DisplayString(TheTextBox As TextBox, ByVal TheString
As String)
    If Len(TheString) < 65536 Then
        TheTextBox.Text = TheString
    Else
         MsgBox "Can not assign a String larger than 64k " &
vbCrLf & _
                        "to the Text property of a TextBox
control." & vbCrLf & _
                      "If you need to support Strings longer
than 64k," & vbCrLf & _
                        "you can use a RichTextBox control
instead.", vbInformation
        MsgBox "File Size Error - ciphertext file not a
multiple of block size"
```

Hal yang membedakan dari masing-masing AES ini adalah banyaknya round yang dipakai. AES-128 menggunakan 10 round, AES-192 sebanyak 12 round, dan AES-256 sebanyak 14 round.AES memiliki ukuran block yang tetap sepanjang 128 bit dan ukuran kunci sepanjang 128, 192, atau 256 bit.

IMPLEMENTASI

Aplikasi yang telah selesai dirancang selanjutnya akan diteruskan kepada tahap pengimplementasian. Pada tahap ini difokuskan kepada penerapan aplikasi yang telah diciptakan dengan bahasa pemrograman yang sesuai, sehingga pada akhirnya diperoleh hasil yang sesuai dengan yang diinginkan. Adapun implementasi dari hasil perancangan program adalah sebagai berikut:

1. Modul Tampilan Utama



Gambar 13. Tampilan Utama Program

2. Modul Enkripsi

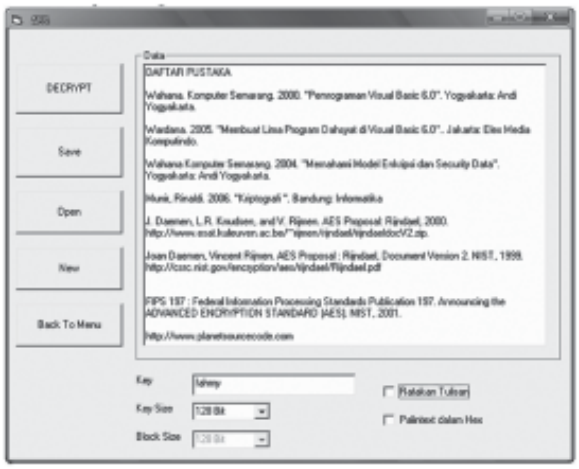
Modul enkripsi yang berfungsi sebagai pengacak data atau informasi sehingga menjadi bentuk yang tidak dapat terbaca oleh orang lain dan menjadi suatu data yang rahasia. Fungsi modul enkripsi pada program ini berjalan sesuai dengan spesfikasi rancangan. Tampilan modul enkripsi dapat dilihat pada gambar dibawah ini.

Gambar 14. Modul Enkripsi

2. Modul Dekripsi

Modul Dekripsi yang berfungsi untuk menerjemahkan pesan yang telah diacak sehingga dapat dibaca oleh si penerima pesan atau oleh pihak – pihak yang berhak menerima

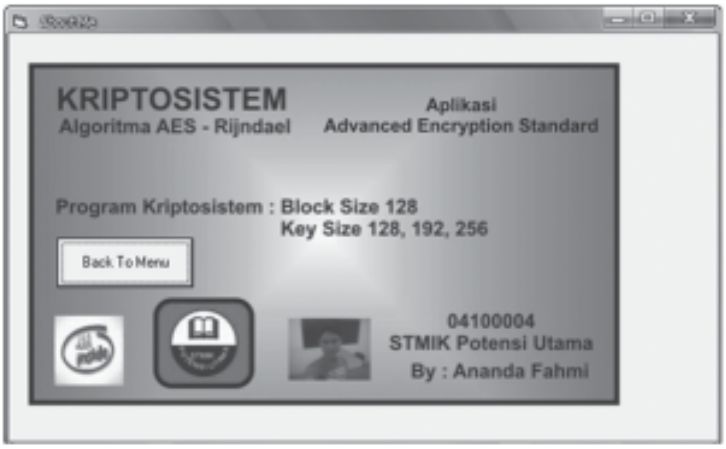
data tersebut. Fungsi modul dekripsi pada program ini berjalan sesuai dengan spesfikasi rancangan. Tampilan modul dekripsi sebagai berikut :



Gambar 15. Modul Dekripsi

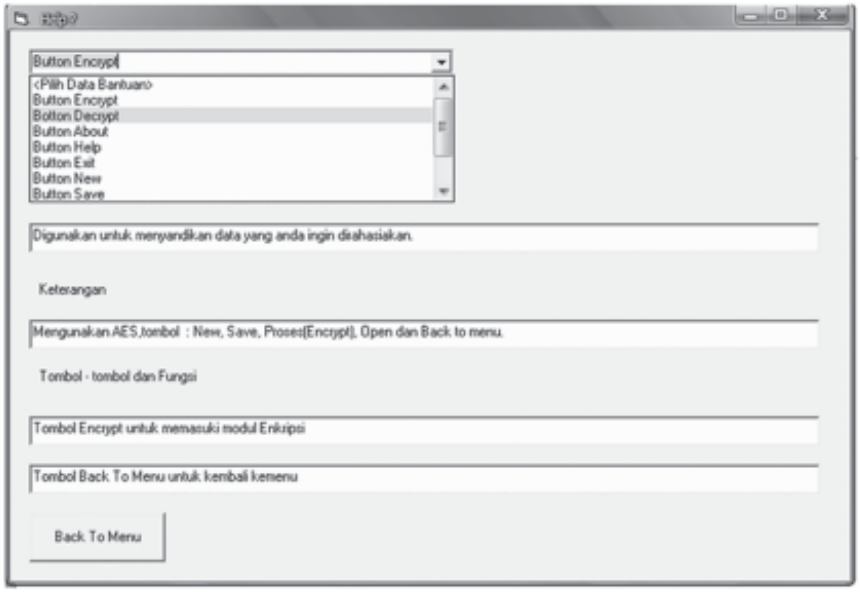
4. Modul About

Modul ini menampilkan penjelasan singkat mengenai program aplikasi dalam pembuatan program aplikasi ini. Fungsi modul About pada program aplikasi ini berjalan sesuai dengan spesfikasi rancangan. Modul ini berjalan dengan baik, dapat dilihat pada gambar dibawah ini :



Gambar 16. Modul About

5. Modul Help.
Modul Help dapat digunakan sebagai panduan untuk menjalankan program aplikasi ini. Fungsi modul ini berjalan sesuai dengan spesfikasi rancangan. Modul ini berjalan dengan baik, dapat dilihat pada gambar 17.



Gambar 17. Modul Help

KESIMPULAN

Dari hasil perancangan dan pembuatan program aplikasi kriptosistem menggunakan algoritma Advanced Encryption Standard (*AES*) ini, dapat diambil kesimpulan sebagai berikut :

1. Program aplikasi kriptosistem ini menjaga kerahasiaan data atau informasi yang ada dalam sebuah komputer dan akan membatasi orang yang tidak berhak atas data yang dimiliki oleh si-pengirim untuk dibaca karena data sudah dienkripsi.
2. Kriptografi *Advanced Encryption Standard* (AES) adalah seni dan ilmu menyembunyikan data dengan menggunakan algoritma AES yang memiliki panjang kunci 128 bit, 192 bit dan 256 bit.
3. Program aplikasi kriptosistem ini dapat mengenkripsikan dan mendekripsikan data yang berupa text yang tersedia pada keyboard, bukan suara maupun gambar.

DAFTAR PUSTAKA

1. Ariyus Dony, 2006, *Kriptografi Keamanan Data Dan Komunikasi,* Graha Ilmu, Yogyakarta.

2. Fahmi Ananda, 2008, *Perancangan Aplikasi Kriptography Advanced Encryption Standard*, Penelitian Tugas Akhir Mahasiswa Strata 1 Program Studi Teknik Informatika, STMIK Potensi Utama, Medan

3. J. Daemen, L.R. Knudsen, and V. Rijmen. AES Proposal: Rijndael, 2000, Diakses pada tanggal 15 Januari 2008 dari http://www.esat.kuleuven.ac.be/rijmen/rijndael/ rijndaeldocV2.zip.

4. Joan Daemen, Vincent Rijmen. *AES Proposal : Rijndael, Document Version 2*. NIST, 1999, Diakses pada tanggal 15 Januari 2008 dari http://csrc.nist.gov/encryption aes/rijndael/ Rijndael.pdf

5. Munir Rinaldi, 2006. **"*Kriptografi*"**. Bandung, Informatika

6. Wahana. Komputer Semarang. 2000. ***"Pemrograman Visual Basic 6.0,.*** Penerbit Andi Yogyakarta.

7. Wardana. 2005. ***"Membuat Lima Program Dahsyat di Visual Basic 6.0".,*** Elex Media Komputindo, Jakarta.