MENINGKATKAN KEAMANAN DATA KEUANGAN DENGAN METODE AES DI STMIK KRISTEN NEUMANN INDONESIA

Mahdianta Pandia
Dosen STMIK Kristen Neumann Indonesia
mahdianta@yahoo.com

ABSTRAK

Saat ini, data yang disimpan dalam komputer sudah tidak aman lagi, oleh karena itu data harus diamankan. Untuk menjamin keamanan dan keutuhan dari suatu data, dibutuhkan suatu proses penyandian. Enkripsi dilakukan ketika data akan dikirim. Proses ini akan mengubah suatu data asli menjadi data rahasia yang tidak dapat dibaca. Sementara itu, proses dekripsi dilakukan untuk mengembalikan data yang telah disandikan menjadi data yang asli, data asli tidak akan terbaca oleh pihak yang tidak berkepentingan.

Metode AES digunakan sebagai standar algoritma kriptografi yang terbaru. algoritma sebelumnya dianggap tidak sanggup lagi untuk mengamankan data, karna panjang kunci yang sedikit. Metode AES adalah algoritma kriptografi dengan menggunakan algoritma Rijndael yang dapat mengenkripsi dan mendekripsi blok data sepanjang 256 bit.

Kata kunci : AES, enkripsi, dekripsi, kriptografi, data keuangan.

I. PENDAHULUAN

1.1. Latar belakang

Saat ini, keamanan terhadap data yang tersimpan dalam basis data sudah tidak aman lagi. Pengamanan terhadap jaringan komputer yang terhubung dengan basis data sudah tidak lagi menjamin keamanan data karena kebocoran data dapat disebabkan oleh "orang dalam" atau pihak–pihak yang langsung berhubungan dengan basis data seperti administrator.

Oleh karena itu penulis tertarik untuk membahas "keamanan data keuangan dengan metode AES di STMIK Kristen Neumann Indonesia", untuk mengamankan data keuangan supaya tidak dapat di rusak oleh orang lain. Hanya akdministrasi yang mengetahui kunci enkripsi dan dekripsi tersebut. Keamanan data sangat penting untuk mengamankan data agar tidak dirusak oleh orang lain yang mengakibatkan kerugian yang sangat besar.

1.2. Rumusan Masalah

Adapun permasalahan yang muncul adalah Belum ada sistem keamanan data keuangan yang digunakan Administrasi untuk mengamankan data-data keuangan.

1.3. Ruang Lingkup Pembahasan

Agar pembahasan lebih terarah dan sistemmatis, maka ruang lingkup pembahasan penulisan Skripsi ini dibatasi sebagai berikut:

1. Hanya mengamankan data keuangan saja, tidak berhubungan dengan data keuangan yang sebenarnya.
2. Kriptosistem ini hanya dapat mengenkripsi dan mendekripsi data berupa database.

1.4. Tujuan dan Manfaat

Tujuan dari penuliasan Skripsi ini adalah untuk mengamankan data keuangan yang di operasikan oleh Administrasi.

Adapun manfaat yang diperoleh dari pelaksanaan Penuliasan Skripsi ini adalah Data yang sudah diamankan tidak dapat diketahui oleh orang lain, kecuali anggota Administrasi.

1.5. Metode Penelitian

a. Metode Pengumpulan Data

Adapun teknik pengumpulan data yang penulis gunakan adalah sebagai berikut:

1. Studi Kepustakaan

Dengan mengumpulkan data-datayang berkaitan dengan topik yang dibahas, yang dilakukan dengan cara membaca buku-buku yang berkaitan dengan keamanan data, Analisis dan Implementasi.

2. Studi Lapangan

   Teknik pengumpulan data dilakukan dengan cara:

a. *Sampling* dan Investigasi, dilakukan dengan mengambil contoh dokumen input dan output dari kampus untuk dianalisis kelemahan dan keunggulannya.
b. Wawancara (*interview*), dilakukan dengan cara berkomunikasi atau bertanya langsung dengan Pegawai Administrasi, untuk mendapatkan informasi yang lebih lengkap.
c. Observasi (pengamatan), dilakukan dengan melakukan pengamatan secara langsung tentang data keuangan yang sudah ada.

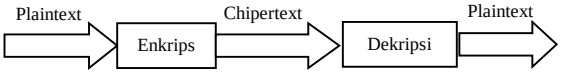
II. TINJAUAN PUSTAKA

2.1. Kriptografi

Kriptografi merupakan seni dan ilmu menyembunyikan informasi dari penerima yang tidak berhak.Kata cryptographi berasal dari kata Yunani *kryptos* (tersembunyi) dan *graphein* (menulis).*Cryptanalysis* adalah aksi untuk memecahkan mekanisme kriptografi dengan cara mendapatkan plaintext atau kunci dari ciphertext

yang digunakan untuk mendapatkan informasi berharga kemudian mengubah atau memalsukan pesan dengan tujuan untuk menipu penerima yang sesungguhnya. *Encryption* adalah mentransformasi data kedalam bentuk yang tidak dapat terbaca tanpa sebuah kunci tertentu.Tujuannya adalah untuk meyakinkan privasi dengan menyembunyikan informasi dari orang-orang yang tidak ditujukan, bahkan mereka mereka yang memiliki akses ke data terenkripsi.Dekripsi merupakan kebalikan dari enkripsi, yaitu transformasi data terenkripsi kembali ke bentuknya semula.

Enkripsi dan dekripsi pada umumnya membutuhkan penggunaan sejumlah informasi rahasia, disebut sebagai kunci. Untuk beberapa mekanisme enkripsi, kunci yang sama digunakan baik untuk enkripsi dan dekripsi; untuk mekanisme yang lain, kunci yang digunakan untuk enkripsi dan dekripsi berbeda. Dua tipe dasar dari teknologi kriptografi adalah *symmetric key (secret/private key)cryptography*dan*asymmetric (public key) cryptography*.Pada *symmetric key cryptography*, baik pengirim maupun penerima memiliki kunci rahasia yang umum.Pada *asymmetric key cryptography*, pengirim dan penerima masing-masing berbagi kunci publik dan privat.



Gambar 1.Kriptografi berbasis kunci

2.2. Sejarah AES (Advance Encryption Standard)

Pada hari pertama tercatat bahwa pertemuan tersebut dihadiri oleh 180 partisipan (termasuk NIST) yang berasal dari 23 negara.Sekitar 28 paper yang diterima dan 21 paper dipresentasikan. Hampir seluruh kandidat algoritma saling mengungkapkan kelemahan satu sama lain namun hal yang cukup mengejutkan adalah Rijndael, tidak terdapat satupun komentar negatif yang ditujukan pada algoritma ini. Kemudian komentar datang dari Eli Biham mengenai estimasi jumlah round minimum yang harus dipenuhi supaya algoritma tersebut dinyatakan "*secure*", dalam artian bahwa attack yang dilakukan memiliki kompleksitas yang lebih kecil daripada "*exhaustive* key search".

2.3. Algoritma AES

Input dan output dari algoritma AES terdiri dari urutan data sebesar 128 bit. Urutan data yang sudah terbentuk dalam satu kelompok 128 bit tersebut disebut juga sebagai blok data atau *plaintext* yang nantinya akan dienkripsi menjadi *ciphertext*. *Cipher key* dari AES terdiri dari *key* dengan panjang 128 bit, 192 bit, atau 256 bit.

2.4. Pengantar Matematis

a. Penjumlahan

Penjumlahan dari dua elemen dalam suatu *finite field* dilakukan dengan menjumlahkan koefisien dari pangkat polinom yang bersesuaian dari dua elemen tersebut.Penjumlahan dilakukan dengan operasi XOR dan dinotasikan dengan $\oplus$. Dengan operasi ini, maka $1 \oplus 1 = 0$, $1 \oplus 0 = 1$, $0 \oplus 1 = 1$, dan $0 \oplus 0 = 1$. Pengurangan dari polinomial identik dengan penjumlahan polinomial.Sebagai alternatif, penjumlahan elemen-elemen pada finite field dapat dijelaskan sebagai penjumlahan *modulo* 2 dari bit yang bersesuaian dalam byte. Untuk 2 byte $\{a_7a_6a_5a_4a_3a_2a_1a_0\}$ dan $\{b_7b_6b_5b_4b_3b_2b_1b_0\}$, hasil penjumlahannya adalah $\{c_7c_6c_5c_4c_3c_2c_1c_0\}$ dimana setiap $c_i = a_i \oplus b_i$. Contoh dari operasi penjumlahan adalah sebagai berikut :

$(x^6+x^4+x^2+x+1) \oplus (x^7+x+1) = x^7+x^6+x^4+x^2$ (notasi polinomial)

$\{01010111\} \oplus \{10000011\} = \{11010100\}$ (notasi biner)

$\{57\} \oplus \{83\} = \{d4\}$ (notasi hexadesimal)

b. Perkalian

Dalam representasi polinomial, perkalian dalam **GF**(2^8) yang dinotasikan dengan mengacu pada perkalian modulo polinomial sebuah *irreduciblepolynomial* yang berderajat 8. Sebuah polinom bersifat *irreducible* jika satu-satunya pembagi adalah dirinya sendiri dan 1. Untuk algoritma AES, *irreducible polynomial* ini adalah :$m(x) = x^8 + x^4 + x^3 + x + 1$

atau dalam notasi hexadesimal adalah {01}{1b}. Sebagai contoh,

{57}•{83} = {c1}

, karena

$(x^6+ x^4 + x^2 + x + 1) \bullet (x^7+ x + 1) = x^{13} + x^{11} + x^9 + x^8 + x^7 +$

$x^7 + x^5 + x^3 + x^2 + x +$

$x^6 + x^4 + x^2 + 1$

$= x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 +$

$x^4 + x^3 + 1$

dan

$x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + x + 1$ modulo $(x^8 + x^4 + x^3 + x + 1)$

$= x^7 + x^6 + 1$

2.5. Algoritma AES– *Rijndael*

Dengan blok input atau blok data sebesar 128 bit, *key* yang digunakan pada algoritma AES tidak

harus mempunyai besar yang sama dengan blok input. *Cipher key* pada algoritma AES bisa menggunakan kunci dengan panjang 128 bit, 192 bit, atau 256 bit. Perbedaan panjang kunci akan mempengaruhi jumlah *round* yang akan diimplementasikan pada algoritma AES ini. Di bawah ini adalah Tabel yang memperlihatkan jumlah *round* (*Nr*) yang harus diimplementasikan pada masing-masing panjang kunci.

Tabel 1. Perbandingan jumlah Round dan Key

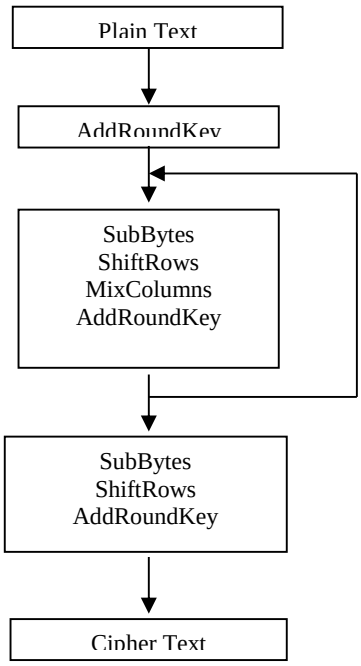
	Panjang Kunci (*Nk words*)	Ukuran Blok (*Nb words*)	Jumlah Putaran (*Nr*)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

a. Ekspansi Kunci

Algoritma AES mengambil kunci *cipher*, K, dan melakukan rutin ekspansi kunci (*key expansion*) untuk membentuk *key schedule*. Ekspansi kunci menghasilkan total Nb(Nr+1) *word*. Algoritma ini membutuhkan set awal *key* yang terdiri dari *Nb word*, dan setiap *round Nr* membutuhkan data kunci sebanyak *Nb word*. Hasil *key schedule* terdiri dari array 4 byte word linear yang dinotasikan dengan [w_i]. *SubWord* adalah fungsi yang mengambil 4 byte *word* input dan mengaplikasikan S-Box ke tiap-tiap data 4 byte untuk menghasilkan *word* output. Fungsi *RotWord* mengambil *word* [a_0, a_1, a_2, a_3] sebagai input, melakukan permutasi siklik, dan mengembalikan *word* [a_1, a_2, a_3, a_0]. Rcon[i] terdiri dari nilai-nilai yang diberikan oleh [x^{i-1}, {00}, {00}, {00}], dengan x^{i-1} sebagai pangkat dari x (x dinotasikan sebagai {02} dalam *field GF*(2^8). W*ord* ke *Nk* pertama pada ekspansi kunci berisi kunci *cipher*. Setiap word berikutnya, w[i], sama dengan XOR dari *word* sebelumnya, w[i-1] dan *wordNk* yang ada pada posisi sebelumnya, w[i-Nk]. Untuk *word* pada posisi yang merupakan kelipatan *Nk*, sebuah transformasi diaplikasikan pada w[i-1] sebelum XOR, lalu dilanjutkan oleh XOR dengan konstanta *round*, Rcon[i]. Transformasi ini terdiri dari pergeseran siklik dari byte data dalam suatu word *RotWord*, lalu diikuti aplikasi dari *lookup Tabel* untuk semua 4 byte data dari word *SubWord*.

b. Enkripsi

Proses enkripsi pada algoritma AES terdiri dari 4 jenis transformasi bytes, yaitu SubBytes, ShiftRows, Mixcolumns, dan AddRoundKey. Pada awal proses enkripsi, input yang telah dikopikan 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.



Gambar 3. Diagram Alir Proses Enkripsi

c. SubBytes

SubBytes merupakan transformasi byte dimana setiap elemen pada state akan dipetakan dengan menggunakan sebuah Tabel substitusi (*S-Box*). Hasil yang didapat dari pemetaan dengan menggunakan Tabel *S-Box* (Tabel 8 : substitusi (*S-Box*))ini sebenarnya adalah hasil dari dua proses transformasi bytes, yaitu :

1. *Invers* perkalian dalam GF(2^8) adalah fungsi yang memetakan 8 bit ke 8 bit yang merupakan *invers* dari elemen *finite field* tersebut. Suatu byte a merupakan *invers* perkalian dari byte b bila a•b = 1, kecuali {00} dipetakan ke dirinya sendiri. Setiap elemen pada *state*akan dipetakan pada Tabel *invers*. Sebagai contoh, elemen “01010011” atau {53} akan dipetakan ke {CA} atau “11001010”.
2. Transformasi *affine* pada *state* yang telah dipetakan. Transformasi *affine* ini apabila dipetakan dalam bentuk matriks adalah sebagai berikut :

$$\begin{bmatrix} b_0' \\ b_1' \\ b_2' \\ b_3' \\ b_4' \\ b_5' \\ b_6' \\ b_7' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

$b_7\ b_6\ b_5\ b_4\ b_3\ b_2\ b_1\ b_0$ adalah urutan bit dalam elemen state atau array byte dimana $b_7$adalah *most significant bit* atau bit dengan posisi paling kiri.

d. ShiftRows

Transformasi Shiftrows pada dasarnya adalah proses pergeseran bit dimana bit paling kiri akan dipindahkan menjadi bit paling kanan (rotasi bit). Transformasi ini diterapkan pada baris 2, baris 3, dan baris 4. Baris 2 akan mengalami pergeseran bit sebanyak satu kali, sedangkan baris 3 dan baris 4 masing-masing mengalami pergeseran bit sebanyak dua kali dan tiga kali.

e. MixColumns

MixColumns mengoperasikan setiap elemen yang berada dalam satu kolom pada *state*(Gambar 15 : MixColumns()). Elemen pada kolom dikalikan dengan suatu polinomial tetap a(x) = {03}x^3 + {01}x^2 + {01}x + {02}. Secara lebih jelas, transformasi mixcolumns dapat dilihat pada perkalian matriks berikut ini :

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

Melakukan proses penambahan pada operasi ini berarti melakukan operasi *bitwise* XOR. Maka hasil dari perkalian matriks diatas dapat dianggap seperti perkalian yang ada di bawah ini :

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

f. AddRoundKey

Pada proses AddRoundKey, sebuah *round key* ditambahkan pada *state* dengan operasi bitwise XOR. Setiap *round key* terdiri dari *Nbword* dimana tiap *word* tersebut akan dijumlahkan dengan *word* atau kolom yang bersesuaian dari *state* sehingga:

$$[s'_{0,c}, s'_{1,c}, s'_{2,c}, s'_{3,c}] = [s_{0,c}, s_{1,c}, s_{2,c}, s_{3,c}] \oplus [w_{round*Nb+c}] \text{ untuk } 0 \le c \le Nb$$

[w_i] adalah *word* dari key yang bersesuaian dimana $i = round*Nb+c$. Transformasi AddRoundKey diimplementasikan pertama kali pada *round* = 0, dimana *key* yang digunakan adalah *initial key* (*key* yang dimasukkan oleh kriptografer dan belum mengalami proses *key expansion*).

g. 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.

h. 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. Pada baris kedua, pergeseran bit dilakukan sebanyak 3 kali, sedangkan pada baris ketiga dan baris keempat, dilakukan pergeseran bit sebanyak dua kali dan satu kali.

i. 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 ini berbeda dengan Tabel *S-Box* dimana hasil yang didapat dari Tabel ini adalah hasil dari dua proses yang berbeda urutannyayaitu transformasi *affine* terlebih dahulu, baru kemudian perkalian *invers* dalam GF(2^8).

$$\begin{bmatrix} b_7' \\ b_6' \\ b_5' \\ b_4' \\ b_3' \\ b_2' \\ b_1' \\ b_0' \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} b_7 \\ b_6 \\ b_5 \\ b_4 \\ b_3 \\ b_2 \\ b_1 \\ b_0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

Perkalian *invers* yang dilakukan pada transformasi InvSubBytes ini sama dengan erkalian *invers* yang dilakukan pada transformasi SubBytes.

j. InvMixColumns

Pada InvMixColumns, kolom-kolom pada tiap *state* (*word*) akan dipandang sebagai polinom atas GF(2^8) dan mengalikan modulo $x^4 + 1$ dengan polinom tetap $a^{-1}(x)$ yang diperoleh dari :

$$a^{-1}(x) = \{0B\}x^3 + \{0D\}x^2 + \{09\}x + \{0E\}.$$

Atau dalam matriks :

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

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

Hasil dari perkalian diatas adalah :

$$s'_{0,c} = (\{0E\} \bullet s_{o,c}) \oplus (\{0B\} \bullet s_{1,c}) \oplus (\{0D\} \bullet s_{2,c}) \oplus (\{09\} \bullet s_{3,c})$$

$$s'_{1,c} = (\{09\} \bullet s_{o,c}) \oplus (\{0E\} \bullet s_{1,c}) \oplus (\{0B\} \bullet s_{2,c}) \oplus (\{0D\} \bullet s_{3,c})$$

$$s'_{2,c} = (\{0D\} \bullet s_{o,c}) \oplus (\{09\} \bullet s_{1,c}) \oplus (\{0E\} \bullet s_{2,c}) \oplus (\{0B\} \bullet s_{3,c})$$

$$s'_{3,c} = (\{0B\} \bullet s_{o,c}) \oplus (\{0D\} \bullet s_{1,c}) \oplus (\{09\} \bullet s_{2,c}) \oplus (\{0E\} \bullet s_{3,c})$$

k. Inverse AddRoundKey

Transformasi Inverse AddRoundKey tidak mempunyai perbedaan dengan transformasi AddRoundKey karena pada transformasi ini hanya dilakukan operasi penambahan sederhana dengan menggunakan operasi bitwise XOR.

2.6. Microsoft Access

Microsoft Access adalah program pengolah database yang canggih yang biasanya digunakan untuk mengolah berbagai jenis data dengan pengoperasian yang mudah yang misalnya, untuk menampung daftar pelanggan, pendataan data karyawan, dan lain sebagainnya.

Microsoft Access merupakan salah satu software pengolah database yang berjalan dibawah sistem windows. Microsoft Access merupakan program aplikasi perangkat manajemen yang luwes yang bisa digunakan untuk mengurutkan, menyeleksi dan mengatur informasi penting yang diperlukan. (Jim Rohn, 2009)

III. ANALISA DAN PERANCANGAN

3.1. Kriptografi Metoda AES

Tahapan ini berisi teori-teori mengenai kriptografi metoda AES. Modul ini dibagi menjadi 2 bagian, yaitu :

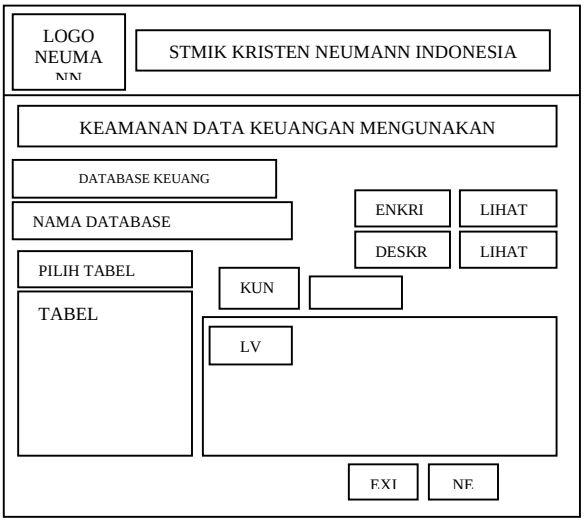
1. Teori Proses Enkripsi.
   Pada modul ini dijelaskan mengenai proses kerja dari algoritma enkripsi, S-Box dan urutan kunci yang digunakan dalam proses enkripsi pada metode AES.
2. Teori Proses Dekripsi.
   Pada modul ini dijelaskan mengenai proses kerja dari algoritma dekripsi dan urutan kunci yang digunakan dalam proses dekripsi pada metoda AES.

3.2. Perancangan Sistem Keamanan

Perancangan program aplikasi kriptografi dengan menggunakan algoritma *Advanced Encryption Standard*. Rancangan ini digunakan untuk meningkatkan keamanan dalam proses pengiriman atau pertukaran data. Rancangan ini dilakukan dalam beberapa tahap yaitu dimulai dari pembuatan diagram konteks dilanjutkan lagi dengan perancangan antar muka program.

1. Form KriptografiAES

Desain form kriptografiAES seperti yang terlihat pada gambar 3.



Gambar 3. Desain Form KriptografiAES

Form KriptografiAES terdiri dari atas Logo Neumann, Tulisan STMIK Kristen Neumann Indonesia, Judul, Keamanan Data Keuangan Mengunakan Metode AES. Juga terdapat butten database keuangan, nama database pilih tabel, nama tabel, terdapat juga butten kata kunci, proses enkripsi, proses deskripsi dan lihat tabel. Juga terdapat ListView tempat melihat hasil enkripsi dan deskripsi. Dan butten next untuk melanjut ke form berikutnya dan butten exit untuk menghentikan sistem.

3.3. Perancangn Basisdata

Untuk merancang aplikasi ini perlu dipikirkan rancangan database untuk keperluan sistem. Setiap fields yang akan di isi kedalam tabel harus menggunakan tipe data text dan memo, karena didalam proses enkripsi hanya bisa mengenkripsi yang tipe datanya berupa text dan memo. Aplikasi ini hanya mengunakn satu tabel yaitu tabel penggajian dosen, dan dapat diuraikan seperti berikut:

Tabel 2. Penggajian

Nama Fields	Tipe Data
NID	Memo
NamaDosen	Memo
GajiPokok	Memo
Tunjangan	Memo
GajiPokok	Memo

Primary Key : NID

IV. HASIL DAN PEMBAHASAN

4.1. Hasil Pembahasan

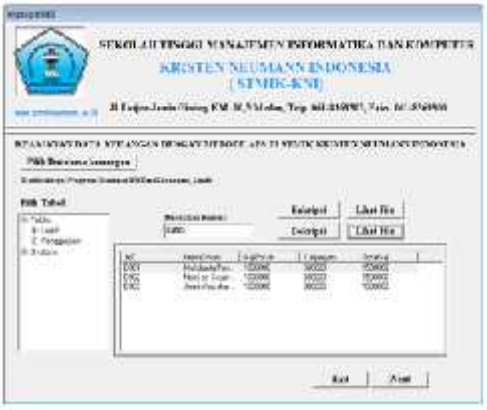
Setelah program aplikasi dirancang, maka tahap selanjutnya adalah tahap hasil dari perancangan. Hasil perancangan ini dilakukan dengan tujuan untuk mengetahui apakah program

berhasil ataw tidak dan sesuai dengan yang dirancang. Aplikasi yang digunakan adalah hanya mengamankan Data Keuangan berupa database. Tidak bergantung pada database keuangan yang sebenarnya.

1. Hasil Pembahasan pada Form KriptografiAES
   Pada form ini berfungsi untuk mengenkripsi dan mengdeskripsi database yang telah disediakan. Untuk melihat hasil enkripsi dari sistem keamanan inipertama-tama memilih database yang akan dienkripsi, setelah itu memilih tabelnya, setelah tabel terpilih masukkan kata kunci untuk mengenkripsi tabel, setelah itu klik batten Enkripsi yang disediakan, tabel yang dipilih langsung memasuki proses enkripsi, setelah itu hasil enkripsi dapat dilihat dengan memilih batten Lihat File, maka hasil enkripsi akan muncul pada lestview. Seperti yang terlihat pada gambar 4.

Gambar 4. Hasil Enkripsi

Setelah dilakukan hasil enkripsi secara otomatis data yang ada didalam tabel sudah disimpan dan diubah atau di rusak, untuk mengembalikan data seperti semula maka perlu dilakukan proses dekripsi dengan kata kunci yang sama, maka dapat memilih batten Dekripsi, untuk melihat hasil enkripsi dapat dilihat pada batten Lihat File, data yang di enkripsi berubah menjadi data seperti semula seperti pada gambar



Gambar 5. Hasil Dekripsi

V. KESIMPULAN DAN SARAN

4.2. Kesimpulan

Dari hasil perancangan aplikasisi Keamanan Data Keuangan dengan Metode *Advanced Encryption Standar*d (AES) ini, dapat diambil kesimpulan sebagai berikut :

1. Enkripsi Data Keuangan dengan Metode AES menambah keamanan Data Keuangan karena data sudah disandikan dari data sebelumnya.
2. Ukuran file sebelum enkripsi bertambah setelah di enkripsi kurang lebih 25%
3. Program aplikasi ini menjaga kerahasiaan Data atau Tabel yang penting yang ada dalam sebuah komputer.

4.3. Saran

Saran yang berguna untuk pengembangan lebih lanjut terhadap program aplikasi ini adalah jadi agar data-data keuangan yang digunakan administrasi lebih aman sebaiknya keungan administrasi menggunakan perancangan aplikasi keamanan data keuangan dengan metode Advaced Encryption Standard (AES).

DAFTAR PUSTAKA

[1.] Bobby Gustiono, http://downloads.ziddu.com/downloadfile/12613952/ Skripsi4.rar.html, tanggal akses 13 Desember 2011

[2.] Dony Ariyus, 2008, *Pengantar Ilmu Kriptografi*, Penerbit Andi, Yogyakarta.

[3.] Eli Biham, *Design Tradeoffs of The AES Candidates*, Oktober 1998.

[4.] FIPS 197 : Federal Information Processing Standards Publication 197. Announcing the ADVANCED ENCRYPTION STANDARD (AES). NIST, 2001

[5.]Hartono,http://downloads.ziddu.com/downloadfiles/12613337/Skripsi20.zip, tanggal akses 13 Desember 2011

[6.] Jim Rohn,http://www2.ukdw.ac.id/kuliah/info/TI2023/Modul07C.pdf, tanggal akses 10 Agustus 2012

[7.] Joan Daemen, Vincent Rijmen, http://csrc.nist.gov/encryption/aes/rijndael/Rijndael.pdf, tanggal akses 10 Desember 2011

[8.] R.Johannes Buchman A, 2004, Pengantar Kriptografi, Publisher: Springer; 2 edisi.

[9.]Wahana Computer, 2003, Memahami Model Enkripsi dan Dekripsi metode AES, Penerbit Andi, Yogyakarta.