INSIGHT
Volume 1 No. 1 | April 2018 : 110-115

Implementasi Algoritma Advanced Encryption Standard (AES) 128-bit Pada Aplikasi Sharing Dokumen Berbasis Android

Gentra Muchammad Akbar[1], Ichsan Taufik[2], Rahmat Jaenal Abidin[3]
Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN Sunan Gunung Djati Bandung
Jl. A.H Nasution 105 Bandung 40614
[1]Gentramuhamadakbar@gmail.com, [2]ichsan@uinsgd.ac.id, [3]rahmat.zaenal@uinsgd.ac.id

File sharing is an activity in which internet users can share data and information with other users, with how to upload data to computer server and other users can download data from computer server. At this time, the exchange of data or information is very often done, the possibility of hacking the data or information may occur in the exchange of information,so security of data or information need to get special attention. The Advanced Encryption Standard algorithm is an algorithm that does encoding of digital data, in particular is data document .doc, .PDF, .txt, .xls, and .ppt. This research Advanced Encryption Standard (AES) algorithm 128-bit implemented to increase security in the document data.

Keywords- AES, Cryptography, Decryption, Encryption

***File sharing* adalah aktifitas dimana para pengguna internet dapat berbagi data atau informasi dengan pengguna lain, dengan cara mengunggah data ke komputer server dan pengguna lainnya dapat mengunduh data tersebut dari komputer server. Pada saat ini, pertukaran data atau informasi sangat sering dilakukan, kemungkinan peretasan data atau informasi dapat terjadi dalam pertukaran informasi sehingga keamanan data atau informasi perlu mendapatkan perhatian khusus. Algoritma *Advanced Encryption Standard* merupakan algoritma yang melakukan penyandian terhadap data digital, khususnya adalah data dokumen .doc, .pdf, .txt, .xls, dan .ppt. Pada penelitian ini algoritma *Advanced Encryption Standard* (AES) 128-bit diimplementasikan untuk meningkatkan keamanan pada data dokumen.**

Kata Kunci- AES, Kriptografi, Dekripsi, Enkripsi

I. PENDAHULUAN

1.1. LATAR BELAKANG

Seiring dengan tingkat mobilitas yang sangat tinggi, beberapa tahun terakhir tengah marak perangkat bergerak. Salah satu perangkat *mobile* yang paling pesat dan berkembang adalah *smartphone* berbasis android, di mana hampir setiap orang memilikinya. Hingga saat ini Android terus berkembang baik secara sistem maupun aplikasinya. Salah satu aplikasi yang saat ini berkembang adalah aplikasi *File Sharing*.

File sharing adalah aktifitas dimana para pengguna internet dapat berbagi *file* dengan pengguna internet lainnya dengan cara penyedia *file* terlebih dahulu meng-*upload file* ke komputer server dan kemudian para pengguna internet yang lainnya dapat mendownload *file* tersebut dari komputer server.

Pada saat ini, pertukaran data atau informasi sangat sering dilakukan, kemungkinan peretasan data atau informasi dapat terjadi dalam pertukaran data atau informasi sehingga keamanan data atau informasi perlu mendapatkan perhatian khusus. Algoritma *Advanced Encryption Standard* (AES) merupakan algoritma yang melakukan penyandian terhadap data digital, khususnya adalah data dokumen .doc, .pdf, .txt, .xls, dan .ppt. Algoritma AES memiliki tiga pilihan kunci yaitu : AES 128-bit, AES 192-bit, dan AES 256-bit. Dari ketiga tipe tersebut selain memiliki tingkat keamanan yang tinggi, proses waktu enkripsi dan dekripsi AES 128-bit lebih cepat dibandingkan dengan tipe kunci yang lain [1].

Atas dasar tersebut di atas, maka dalam tugas akhir ini mengambil judul "Implementasi Algoritma *Advanced Encryption Standard* (AES) 128-bit Pada Aplikasi Sharing Dokumen Berbasis Android".

1.2. RUMUSAN MASALAH

Beradasarkan pada latar belakang di atas, maka masalah yang dapat dirumuskan dalam proses pembangunan sistem ini adalah:

1. Bagaimana membuat *Aplikasi Sharing Dokumen* berbasis Android.
2. Bagaimana menerapkan Algoritma *Advanced Encryption Standard (AES) 128-bit* pada Aplikasi *Sharing* Dokumen*?*
3. Bagaimana kinerja Algoritma *Advanced Encryption Standard (AES) 128-bit?*

1.3. MAKSUD DAN TUJUAN

Tujuan yang hendak dicapai dalam proses pembuatan aplikasi ini adalah :

1. Membuat *Aplikasi Sharing Dokumen* berbasis Android.
2. Menerapkan Algoritma *Advanced Encryption Standar (AES) 128-bit* pada *Aplikasi Sharing Dokumen*.
3. Mengetahui kinerja Algoritma *Advanced Encryption Standar (AES) 128-bit* untuk keamanan data.

1.4. BATASAN MASALAH

Dalam proses pengerjaan aplikasi yang dirancang akan dibatasi. Adapun batasan masalah yang melingkupi kinerja sistem ini yaitu:

1. *Aplikasi Sharing Dokumen* hanya akan berjalan pada smartphone bersistem operasi Android.
2. Proses enkripsi dan dekripsi hanya digunakan untuk data .doc, .ppt, .xls, .txt, dan .pdf.
3. Proses enkripsi dan dekripsi yang digunakan yaitu dengan metode AES (*Advanced Encryption Standard*) *128-bit.*
4. Batas maksimal dokumen 20Mb.
5. Aplikasi ini dirancang menggunakan bahasa pemograman *Java.*
6. Menggunakan *Firebase Realtime Database* untuk penyimpanan data dokumen.
7. Fitur yang terdapat dalam aplikasi ini adalah:
    a. Sistem *login* yang menggunakan akun Google.
    b. Pengguna dapat merubah/mengedit dokumen.
    c. Pengguna dapat menambahkan pertemanan.

II. METODE PENELITIAN

2.1. KRIPTOGRAFI

Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan. Sebagai pembanding, selain definisi tersebut, terdapat pula definisi yang dikemukakan yaitu kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data serta otentikasi [2].

dalam kriptografi kita akan sering menemukan berbagai istilah atau terminology [3]. Beberapa istilah penting untuk diketahui dibawah ini:

a. Pesan, Plainteks, dan Chiperteks
   Pesan adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Nama lain untuk pesan adalah plainteks atau teks jelas. Bentuk pesan yang tersandi disebut chiperteks atau kriptogram agar tidak dapat dipahami maknanya oleh pihak yang tidak berkewenangaa.
b. Enkripsi dan Dekripsi
   Proses menyandikan plainteks menjadi cipherteks disebut enkripsi, sedangkan proses mengembalikan chiperteks menjadi plainteks semula dinamakan dekripsi.

2.2. ALGORITMA AES

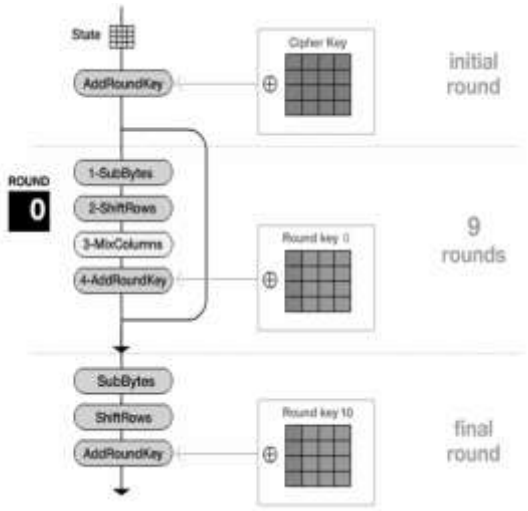
Algoritma AES merupakan algoritma simetris yaitu mengunakan kunci yang sama untuk proses enkripsi dan dekripsi [4]. Algoritma AES memiliki tiga pilihan kunci yaitu tipe: AES-128, AES-192 dan AES-256. Masing-masing tipe menggunakan kunci internal yang berbeda yaitu round key untuk setiap proses putaran. Proses putaran enkripsi AES-128 dikerjakan sebanyak 10 kali (a=10), yaitu sebagai berikut:

1. Addroundkey
2. Putaran sebanyak a-1 kali, proses yang dilakukan pada setiap putaran adalah: SubBytes, ShiftRows, MixColumns, dan AddRoundKey.
3. Final round, adalah proses untuk putaran terakhir yang meliputi SubBytes, ShiftRows, dan AddRoundKey.

Sedangkan pada proses dekripsi AES-128, proses putaran juga dikerjakan sebanyak 10 kali (a=10), yaitu sebagai berikut:

1. Addroundkey
2. Putaran sebanyak a-1 kali, dimana pada setiap putaran dilakukan proses: InverseShiftRows, InverseSubBytes, AddRoundKey, dan InverseMixColumns.
3. Final round, adalah proses untuk putaran terakhir yang meliputi InverseShiftRows, InverseSubBytes, dan AddRoundKey.

Pada enkripsi dan dekripsi AES-192 proses putaran dikerjakan 12 kali (a=12), sedangkan untuk AES-256 proses putaran dikerjakan 14 kali (a=14). Algoritma AES digunakan untuk mengenkripsi dan mendekripsi file dokumen digital khususnya adalah file dokumen PDF, DOC, XLS, PPT, dan TXT. Pada Gambar 2.1 dapat dilihat diagram proses enkripsi.

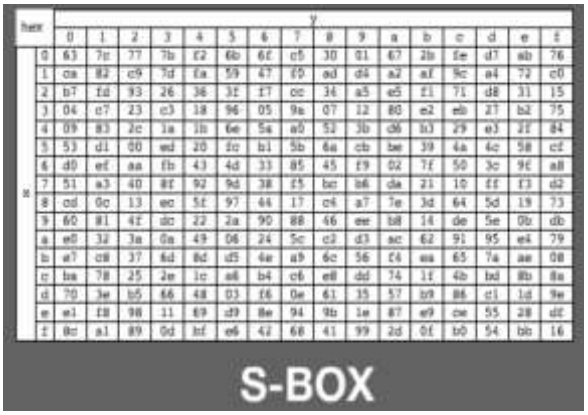


Gambar 2.1 Diagram Proses Enkripsi [8]

Pada gambar 2.1 dapat dijelaskan sebagai berikut:

1. AddRoundKey pada dasarnya adalah mengkombinasikan chiperteks yang sudah ada dengan chiper key dengan hubungan XOR
2. Transformasi SubBytes() memetakan setiap byte dari array state dengan menggunakan tabel substiusi S-box. Tidak seperti DES yang mempunyai S-box berbeda pada setiap putaran, AES hanya mempunyai satu buah S-box. Tabel S-box yang digunakan bisa dilihat pada Tabel 2.1.

Tabel 2.1 Rijndael S-Box[8]



Cara pensubstitusian adalah 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. Contoh S[0, 0] = 19, maka S'[0, 0] = d4.

3. Shift Rows seperti namanya adalah sebuah proses yang melakukan shift atau pergeseran pada setiap elemen blok/tabel yang dilakukan per barisnya. Yaitu baris pertama tidak dilakukan pergeseran, baris kedua dilakukan pergeseran 1 byte, baris ketiga dilakukan pergeseran 2 byte, dan baris keempat dilakukan pergeseran 3 byte. Pergeseran tersebut terlihat dalam sebuah blok adalah sebuah pergeseran tiap elemen ke kiri tergantung berapa byte tergesernya, tiap pergeseran 1 byte berarti bergeser ke kiri sebanyak satu kali.
4. Mix Column adalah mengalikan tiap elemen dari blok chiper dengan matriks. Tabel sudah ditentukan dan siap pakai. Pengalian dilakukan seperti perkalian matriks biasa yaitu menggunakan dot product lalu perkalian keduanya dimasukkan ke dalam sebuah blok chiper baru.

I. *Pseudo code Algoritma AES*

```
Program AES {Menyembunyikan atau merahasiakan (enkripsi) dan mengembalikan (dekripsi) file teks}
Cipher(byte in[], byte out[], word W[])
  /*Nama fungsi*/
Begin
In = 4 * Nb
Out = 4 * Nb
W = Nb * (Nr +1)
Byte state[4,Nb]
State = In /* Memasukkan Input ke state*/
AddRoundKey(state,W) For round=1 step 1 to Nr-1
/*proses yang berlaku untuk semua ronde kecuali ronde terakir*/
SubBytes(state)
shiftRows(state)
MixColumns(state)
AddRoundKey(state, w + round * Nb)
End for
SubBytes(state) /*proses yang berlaku khusus untuk ronde terakhir*/
ShiftRows(state)
AddRoundKey(state, w+round * Nb) /*
Mengirimkan keluaran ke out */
Out = state End
```

pseudocode AES dimana program AES ini bertujuan untuk menyembunyikan atau merahasiakan (enkripsi) dan mengembalikan (dekripsi) *file* teks. Dimana Cipher(byte in[], byte out[], word W[])untuk menentukan byte input, byte output dan kata yang akan dienkripsi.

2.3. FIREBASE

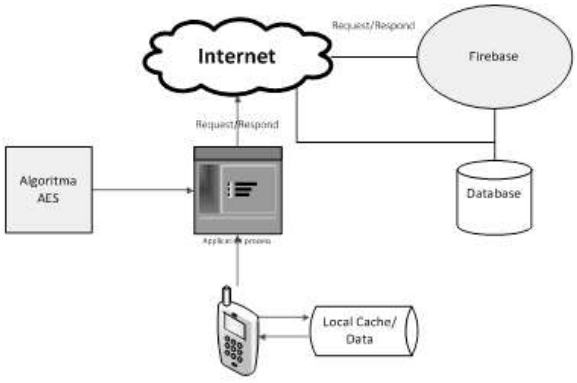
Firebase adalah *cloud services provider* dengan *backend as a service* (BaaS) yang berbasis di San Fransisco, California. Perusahaan ini membuat sejumlah produk untuk pengembangan aplikasi *mobile* ataupun *web*. Firebase di dirikan oleh Andrew Lee dan James Tamplin pada tahun 2011 dan diluncurkan dengan *cloud database* secara *realtime* di tahun 2012. Produk utama firebase yakni suatu *database* yang menyediakan *API* untuk memungkinkan pengembang menyimpan dan mensinkronisasi data lewat *multiple client*. Perusahaan ini diakuisisi oleh Google pada Oktober 2014 [5].

Firebase sendiri sebernarnya lebih merujuk kepada produk yang mereka namakan dengan nama perusahaan, firebase menyediakan *realtime database* dan *backend* sebagai layanan. Suatu aplikasi layanan yang memungkinkan pengembang membuat API untuk disinkronisasikan untuk *client* yang berbeda-beda dan disimpan pada *cloud*-nya firebase. Firebase memiliki banyak *library* yang memungkinkan untuk mengintegrasikan layanan ini dengan Android, iOS, Javascript, Java, Objective-C dan Node.js. *Database* firebase juga bersifat bisa diakses lewat *REST API* dan data *binding* untuk beberapa *framework* javascript seperti halnya AngularJS, ReactJS, Ember.JS, dan Backbone.JS. *REST API* tersebut menggunakan protokol *Server-Sent event* dengan membuat koneksi HTTP untuk menerima *push notification* dari *server*.

Firebase dikembangkan dengan menggunakan *database* MongoDB sehingga Firebase menggunakan tipe database NoSQL. Karena memakai tipe *database* NoSQL maka struktur *database* dari Firebase bersifat fleksibel dan cepat sehingga cocok untuk digunakan pada aplikasi berbasis *mobile*.

III. HASIL DAN PEMBAHASAN

3.1 GAMBARAN UMUM SISTEM



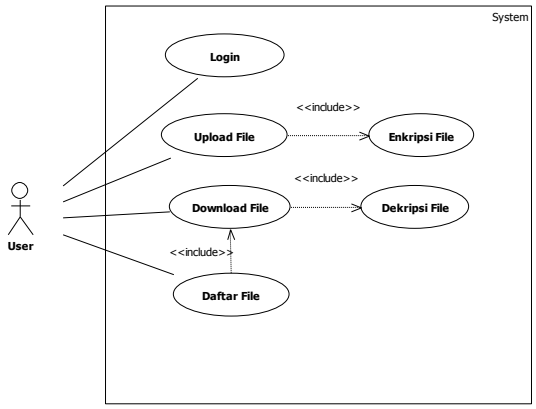
Gambar 3.1 Arsitektur Sistem

Gambar 3.1 menjelaskan proses pengirim mengenkripsi pesan dengan aplikasi yang akan dibuat, dimana aplikasi ini menggunakan bahasa pemrograman *java* dan algoritma AES 128-bit.

3.2. PERANCANGAN SISTEM

Use Case digunakan untuk memodelkan sistem dan menjelaskan bentuk khusus dari fungsi yang diinginkan oleh sistem tersebut [6]. *Use Case* bekerja dengan cara mendeskripsikan interaksi antara pengguna dan sistem [7]. Secara umum *Use Case* terbagi menjadi tiga bagian yaitu definisi *actor*, definisi *use case* dan *scenario use case*.

Use Case digambarkan melalui dua buah kolom, kolom pertama selaku *actor* dan kolom kedua menggambarkan perilaku sistem. Pada gambar 3.4 dibawah ini adalah *use case* dari sistem aplikasi sharing dokumen berbasis android yang mengimplementasikan algoritma AES (*Advanced Encryption Standard*) 128-bit untuk keamanan data



Gambar 3.2 *Use Case Diagram*

Gambar 3.2 menjelaskan pengirim melakukan enkripsi, dan penerima melakukan dekrispi, dimana dekripsi tidak akan dilakukan apabila enkripsi tidak dilakukan.

3.3. IMPLEMENTASI SISTEM ANTARMUKA

1. *Form Login*

Gambar 3.3 Hasil Implementasi Tampilan Login

Gambar 3.3 merupakan tampilan yang muncul pada awal aplikasi, dimana pengguna harus memasukan username dan password dengan menggunakan akun Google untuk dapat mengakses aplikasi yang telah dibuat.

2. *Form* Daftar *File*

Gambar 3.4 Hasil Implementasi Tampilan Daftar File

Gambar 3.4 merupakan tampilan Daftar File. Di halaman ini kita bisa memilih salah satu dokumen. Pada halaman ini proses dekripsi dilakukan saat membuka dokumen.

3. *Form Upload File*

Gambar 3.5 Hasil Implementasi Tampilan Upload File

Gambar 3.5 merupakan tampilan hasil implementasi dari *Upload File*. Pada halaman ini terdapat tombol *browse* untuk memilih *file* yang akan dienkripsi, *key* untuk memasukan kunci, deskripsi file, memilih kepada siapa kita akan mengirim file, dan tombol upload dimana file tersebut akan dienkripsi dan disimpan ke database.

4. *Form Friendship*

Gambar 3.6 Hasil Implementasi Tampilan Friend

Gambar 3.6 merupakan tampilan *Friendship*. Di halaman ini kita dapat melihat *Friendlist* pada akun Google.

3.4. PENGUJIAN HASIL ANALISIS

Untuk mencoba keberhasilan perangkat lunak yang dibuat maka akan dilakukan uji coba perangkat lunak. Uji coba perangkat lunak enkripsi dan dekripsi bertujuan untuk menguji keberhasilan dalam proses enkripsi dan dekripsi file teks.

Dalam proses enkripsi, akan dilakukan uji coba proses enkripsi dokumen dengan format *.txt, *.docx, *.pdf, *.xls, dan *.ppt dengan kunci “KRIPTOGRAFI”.

Proses waktu, isi teks document dan hasil dari enkripsi dan dekripsi dapat dilihat pada gambar 3.7, gambar 3.8, dan tabel 3.1.

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Gambar 3.7 Hasil Pengujian Dekripsi

Tabel 3.1 Proses Waktu Enkripsi dan Dekripsi

No	Format	Ukuran File	Enkripsi (ms)	Dekripsi (ms)	Ket.
1	*.txt	228 b	3.1670 ms	2.0396 ms	Sukses
2	*.docx	39.32 kb	9.7821 ms	9.3784 ms	Sukses
3	*.pdf	422.58 kb	4.9608 ms	5.8196 ms	Sukses
4	*.xls	118.24 kb	1.5573 ms	2.3994 ms	Sukses
5	*.ppt	322.06 kb	5.9236 ms	5.9014 ms	Sukses
6	*.docx	23 mb	-	-	Gagal
7	*pdf	19.7 mb	2.0796 ms	2.2780 ms	Sukses

IV. PENUTUP

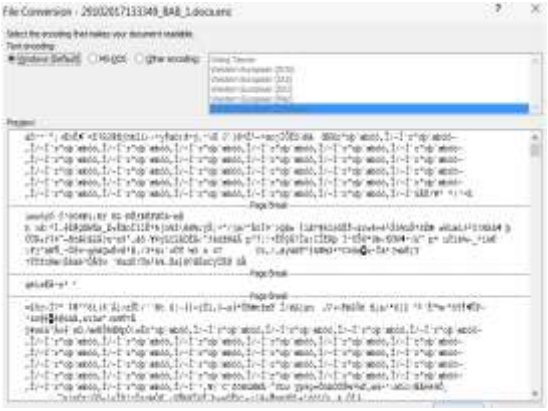
4.1. KESIMPULAN

Berdasarkan dari pengamatan selama perancangan, implementasi dan uji coba, dapat diambil kesimpulan sebagai berikut:

1. Aplikasi ini mampu memberikan kemudahan kepada pengguna untuk berbagi file dokumen dengan pengguna lain dengan keamanan dokumen yang tetap terjaga.
2. Penerapan algoritma *Advanced Encryption Standard* (AES) pada aplikasi *sharing* dokumen untuk mengamankan isi dari dokumen yang *diupload* oleh *user*.
3. Ukuran file dari hasil proses enkripsi berbanding lurus dengan file aslinya. Dan

4.2. SARAN

Aplikasi *Sharing* Dokumen ini tidak terlepas dari kekurangan. Oleh karena itu, untuk kebaikan pengembangan sistem lebih lanjut, adapun saran agar aplikasi ini bisa berfungsi dengan lebih optimal adalah :



Gambar 3.8 Hasil Pengujian Enkripsi

Proses waktu enkripsi tidak jauh berbeda dengan proses waktu dekripsi.

1. Penambahan berbagai macam jenis *file*, tidak hanya untuk jenis *file* dokumen.
2. Menggunakan metode lain seperti *Rivest Code, Serpent* dan lain-lain.
3. Penambahan fitur seperti pemberitahuan ,kolom komentar, dan lain-lain.
4. Pengoprasian aplikasi dapat dilakukan pada platform lain seperti iPhone, Blacberry dan Windows Phone.

V. REFERENSI

[1] N. B. Tampubolon, R. R. Isnanto, and E. W. Sinuraya, "Implementasi Dan Analisis Algoritma Advanced Encryption Standard (Aes) Pada Tiga Variasi Panjang Kunci Untuk Berkas Multimedia," *TRANSIENT*, vol. 4, no. 4, 2015.

[2] R. Munir, "Advanced Encryption Standard (AES)," *Kriptografi*, 2008. .

[3] S. T. C. Kurniawan, Supriyadi, and Dedih, "Implementasi Kriptografi Algoritma Rivest Shamir Adleman dengan Playfair Cipher pada Pesan Teks Berbasis Android," *JOIN (Jurnal Online Inform.*, vol. 2, no. 2, pp. 102–109, 2017.

[4] J. T. Purwanto and E. Utami, "Implementasi Algoritma Kriptografi AES dan Wateramark dengan Metode LSB pada Data Citra," 2013.

[5] "Firebase Realtime Database," *Google Firebase*, 2017. .

[6] M. A. Ramdhani, *Metodologi Penelitian untuk Riset Teknologi Informasi*. Bandung: UIN Sunan Gunung Djati Bandung, 2013.

[7] R. Ramadhan, I. F. Astuti, and D. Cahyadi, "Sistem Pakar Diagnosis Penyakit Kulit Pada Kucing Persia Menggunakan Metode," *Pros. Semin. Ilmu Komput. dan Teknol. Inf.*, vol. 2, no. 1, 2017.