Analisis Perbandingan Enkripsi File Teks Berformat .txt dan .docx Menggunakan Algoritma AES

Kennardy Andrew Limartha[a1], I Gede Arta Wibawa[a2]

Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia
[1]andrewkennardy@gmail.com
[2]gede.arta@unud.ac.id

Abstract

In this research, the Advanced Encryption Standard (AES) algorithm is used to encrypt text files. Data stored in various formats has the potential to cause the size of the data on the data storage device to become large and does not necessarily guarantee the security of the data. One method of data security is to encrypt data. This research utilizes data in text file format with data types in the form of ".txt" and ".docx" which refer to the file size for testing. Based on the test results, it shows that the AES encryption method provides good results for text file formats. The time required for the encryption process using the AES algorithm is influenced by each different file size, the larger the file size used, the longer the computing time required and vice versa.

Keywords: *Advanced Encryption Standard (AES), Encrypt, Data Security, Text Files, .TXT, .DOCX*

1. Pendahuluan

Dunia sudah memasuki era digitalisasi yang semakin berkembang pesat. Komputer merupakan hal yang sangat berpengaruh bagi sarana distribusi data dan informasi. Pastinya di dalam setiap komputer terdapat file atau dokumen yang berisi data-data rahasia. Data yang disimpan dalam berbagai macam format berpotensi menimbulkan data yang memiliki ukuran besar pada perangkat penyimpanan data dan keamanannya belum tentu terjamin. Untuk itu diperlukan suatu pengamanan data yang baik sehingga data dan informasi dapat disimpan dengan keamanan yang terjamin [1]. Keamanan ini diperlukan agar data tersebut hanya dapat dibuka dan diakses oleh si penerima atau orang-orang yang memiliki kewenangan atas data tersebut sehingga pihak lain yang tidak memiliki hak akses tidak dapat mengetahui makna atau isi dari data tersebut [2]. Kriptografi merupakan bentuk solusi yang dapat ditawarkan di dalam keamanan komputer. Teknik yang digunakan dalam kriptografi adalah menyamarkan atau mengubah informasi sehingga ketika dikirim menjadi sesuatu yang tidak bermakna. Misalkan seseorang mengirimkan data yang berisikan informasi rahasia berupa “keuangan”, maka pada proses pengiriman kata akan diganti menjadi sesuatu yang tidak dapat dibaca yaitu “#%! @^!&”. Kata yang tidak bermakna tersebut dinamakan dengan *ciphertext* sedangkan kata atau pesan yang aslinya disebut *plaintext* [2]. Dalam kriptografi ini terdapat berbagai metode untuk mengatasi keamanan data. Salah satunya metode yang kita perlukan adalah enkripsi. Enkripsi merupakan proses mengubah data menjadi format yang tidak dapat dibaca atau sulit dimengerti tanpa memiliki kunci enkripsi yang sesuai, sehingga melindungi kerahasiaan dan integritas data selama transmisi. Enkripsi terjadi pada saat proses pengiriman berlangsung sedangkan dekripsi terjadi saat proses penyampaian pesan ke tujuan yang dituju dengan cara merubah data rahasia pada proses enkripsi kembali lagi menjadi data aslinya. Enkripsi juga sering diartikan sebagai kode atau (*chipper).* Pengkodean dilakukan dengan algoritma tertentu untuk mengkodekan semua aliran data bit dari suatu pesan asli (*plaintext*) menjadi pesan rahasia (*chipertext*). Proses enkripsi menjadikan suatu informasi akan lebih sulit untuk diketahui oleh orang yang tidak berhak. Kemananan tersebut sangatlah diperlukan untuk menghindari terjadinya penyadapan maupun pembajakan terhadap file dengan informasi yang penting, maka dari itu dibutuhkan sebuah algoritma yang dapat memproteksi file penting yaitu Algoritma *Advanced Standard Encryption* (AES). AES adalah algoritma simetris 2

yang menggunakan kunci yang sama untuk proses enkripsi dan dekripsi dengan panjang kunci yang bervariasi, yakni: AES 128-bit, AES 192-bit, AES 256-bit [3]. AES merupakan *chipper* yang berorientasi pada bit, sehingga memungkinkan dalam pengimplementasian algoritma yang efisien ke dalam software dan hardware. One Time Pad (OTP) yang dipilih untuk mengkombinasikan masing-masing karakter pada *plaintext* dengan satu karakter pada satu kunci dienkripsi dengan satu algoritma kemudian diteruskan dengan algoritma yang lainnya [4]. Algoritma AES ini memiliki kelebihan dari segi jenis kunci simetri yang digunakan. Dengan panjang kunci paling sedikit yaitu 128 setidaknya terdapat 2^{128} kemungkinan kunci. Jika komputer tercepat dapat mencoba 1 juta kunci tiap detik maka akan dibutuhkan waktu $5{,}45{,}4 \times 10^{24}$ tahun untuk mencoba seluruh kunci [5]. Tujuan dilakukannya penelitian ini yaitu untuk menganalisa perbandingan hasil terhadap keamanan proses enkripsi dengan menggunakan algoritma AES pada file text berupa txt dan docx. Kedua file ini seringkali dikirimkan ataupun dipulikasikan melalui jaringan publik sehingga sangat rentan terhadap pencurian karena dapat dibaca dan dimengerti oleh semua pihak manusia [6]. Maka dari itu, dilakukanlah analisis ini untuk mengetahui ukuran file yang diperoleh setelah dilakukannya proses enkripsi melalui perbandingan besar atau kecilnya ukuran file dengan waktu yang dibutuhkan dalam proses komputasi. Sehingga diketahui manakah dari kedua file ini yang lebih efektif menggunakan algoritma AES.

2. Metode Penelitian

2.1 Data Penelitian

Data yang digunakan untuk melakukan enkripsi dengan algoritma AES adalah dengan menggunakan format file teks. Akan digunakan 2 tipe data yang berbeda pada file teks untuk melakukan perbandingan kinerja algoritma AES dalam melakukan enkripsi pada 2 buah format file. Tipe data yang akan digunakan yakni, ".txt", dan untuk setiap data akan disamakan ukurannya dengan ukuran yang akan diuji yakni berukuran 100 KB, 200 KB, 500 KB dan 1000 KB. Evaluasi perbandingan didasarkan pada kecepatan atau waktu komputasi dan ukuran setelah dilakukannya proses enkripsi pada data tersebut.

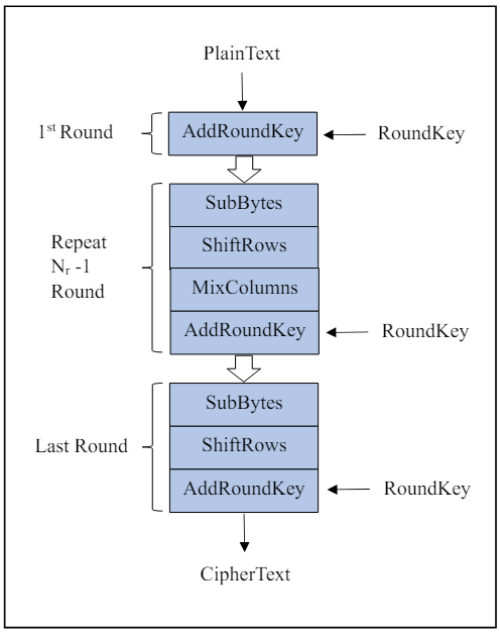
2.2 Algoritma AES

Advanced Encryption Standard (AES) memiliki nama lain yaitu Algoritma Rijndael yang merupakan suatu algoritma enkripsi tipe simetrik *block cipher* dengan sistem permutasi dan subtitusi. Ada tiga jenis algoritmeAES, yaitu AES-128, AES-192, dan AES-256. Pengelompokan ini berdasarkan panjang kunci yang digunakan pada algoritma AES. Selain itu ada beberapa hal lain yang membedakan antar jenis algoritma AES, yaitu *round* yang digunakan. AES-128 menggunakan 10 *round*, AES-192 menggunakan 12 *round*, dan AES-256 menggunakan 14 *round* [7].

2.3 Proses Enkripsi Algoritma AES

Dalam enkripsi terjadi proses pengubahan pesan asli *(plaintext)* menjadi pesan bersandi *(ciphertext).* Secara garis besar proses enkripsi AES terdiri dari 4 jenis transformasi byte, yaitu *SubBytes, ShiftRows, MixColumns dan AddRoundKey*. Proses enkripsi dimulai dari input yang telah dimasukkan ke dalam *array state* akan mengalami perubahan byte *AddRoundKey*. Setelah itu, *array state* akan mengalami perubahan *SubBytes*, *ShiftRows*, *MixColumns*, dan *AddRoundKey* secara berulang-ulang sebanyak putaran. Proses ini dalam algoritma AES disebut sebagai *round function*. *Round* yang terakhir agak berbeda dengan *round round* sebelumnya dimana pada *round* terakhir, *array state* tidak mengalami perubahan *MixColumns* [8]. Dibawah ini merupakan gambar diagram proses enkripsi seperti yang ditunjukkan pada Gambar 1.

p-ISSN: 2986-3929
e-ISSN: 3032-1948

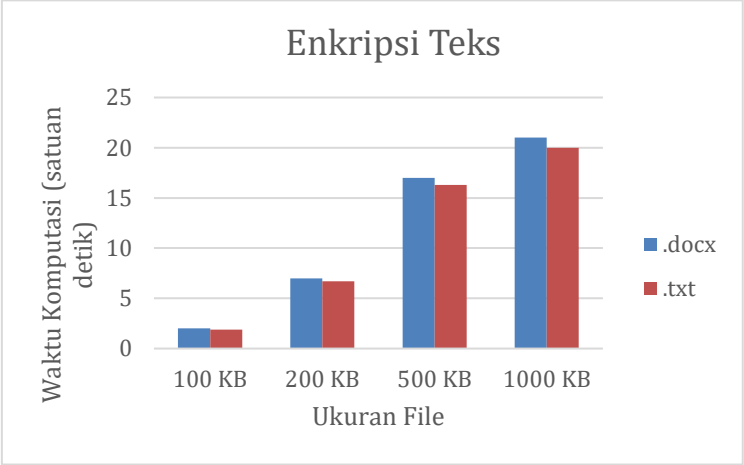


Gambar 1. Diagram Proses Enkripsi AES

a. *AddRoundKey*, melakukan XOR antara pesan asli dengan *cipherkey*
b. *Repeat N_r -1 Round*, memilki proses yang akan dilakukan setiap putaran yaitu:
   - *SubBytes:* Mensubstitusi byte dengan menggunakan tabel substitusi (S-box).
   - *ShiftRows*: Pergeseran baris-baris *array state* secara *wrapping* (perubahan permutasi).
   - *MixColumns*: Mengalikan data di kolom-kolom *array state* (perubahan pengacakan).
   - *AddRoundKey:* Melakukan XOR antara *state* sekarang dengan *round key* (perubahan penambahan kunci).
c. *Last Round*, proses round terakhir sama dengan *round* sebelumnya namun tanpa perubahan *MixColumns* yang meliputi:
   - *SubBytes* (perubahan subtitusi).
   - *ShiftRows* (perubahan permutasi).
   - *AddRoundKey* (perubahan pengacakan).

3. Hasil dan Diskusi

3.1 Perbandingan Waktu Komputasi Tipe Data “.txt” dan “.docx”

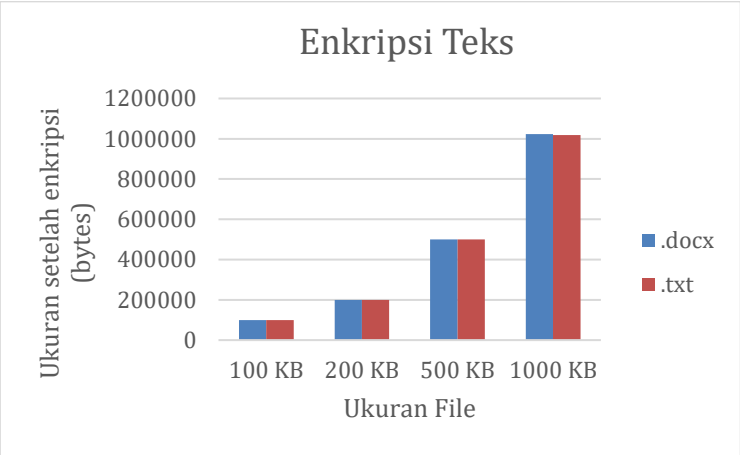


Gambar 2. Perbandingan Waktu Komputasi Pada File Teks

p-ISSN: 2986-3929
e-ISSN: 3032-1948

Berdasarkan pada grafik diatas didapatkan hubungan antara ukuran file dengan waktu komputasi yang dibutuhkan dalam satuan detik. Semakin besar ukuran file yang digunakan maka semakin besar pula waktu yang dibutuhkan untuk komputasi. Grafik diatas menyajikan ukuran file yang sama terhadap kedua tipe data namun terdapat perbedaan yang tergolong tidak terlalu signifikan. Pada ukuran file 1000 KB dengan tipe data ".txt dibutuhkan waktu komputasi 20 detik. Untuk tipe data ".docx" dibutuhkan waktu 21 detik. Sehingga dapat disimpulkan bahwa tipe data ".txt" membutuhkan waktu komputasi yang lebih cepat dibandingkan dengan tipe data ".docx" dalam proses enkripsinya. Hal tersebut disebabkan oleh struktur tipe data ".txt" yang lebih sederhana. Meskipun struktur yang dimiliki oleh tipe data ".docx" lebih kompleks perbedaan yang terlihat tidak terlalu besar. Kemungkinan disebabkan oleh ukuran file yang sama dan efisiensi dari algoritma AES.

3.2 Perbandingan Ukuran File Enkripsi Tipe Data ".txt" dan ".docx"



Gambar 3. Perbandingan Ukuran Enkripsi File Teks

Berdasarkan grafik pada gambar 3 menunjukkan bahwa ukuran file setelah dienkripsi antara kedua tipe data hampir tidak berbeda. Pada ukuran file 100 KB dengan tipe data ".txt ukuran file setelah di enkripsi menjadi 100062 bytes. Untuk tipe data ".docx" ukuran file setelah dienkripsi menjadi 100093 bytes. Pada ukuran file 200 KB dengan tipe data ".txt ukuran file setelah di enkripsi menjadi 200087 bytes. Untuk tipe data ".docx" ukuran file setelah dienkripsi menjadi 20121 bytes. Pada ukuran file 500 KB dengan tipe data ".txt ukuran file setelah di enkripsi menjadi 500173 bytes. Untuk tipe data ".docx" ukuran file setelah dienkripsi menjadi 500385 bytes. Pada ukuran file 1000 KB dengan tipe data ".txt ukuran file setelah di enkripsi menjadi 1018 KB. Untuk tipe data ".docx" ukuran file setelah dienkripsi menjadi 1023 KB. Dari data yang diperoleh menunjukkan bahwa ukuran tipe data ".txt" dan ".docx" terenkripsi menjadi lebih besar dibandingkan dengan ukuran aslinya sebelum dienkripsi. Hasil dari enkripsi menunjukkan bahwa ukuran tipe data ".txt" menjadi lebih kecil dibandingkan dengan ukuran tipe data ".docx".

4. Kesimpulan

Berdasarkan hasil pembahasan yang telah dilakukan di atas dapat disimpulkan bahwa:

a. Algoritma AES merupakan salah satu cara yang terbaik dan efisien untuk meningkatkan keamanan data dalam hal enkripsi.
b. Ukuran file menjadi salah satu faktor yang berpengaruh dalam waktu proses enkripsi dan waktu enkripsi menjadi salah satu tolak ukur dari cepat atau lambatnya suatu proses enkripsi terhadap file.
c. Pada ukuran file 1000 KB dengan tipe data ".txt dibutuhkan waktu komputasi 20 detik dan ukuran file setelah di enkripsi menjadi 1018 KB. Untuk tipe data ".docx" dibutuhkan waktu 21 detik dan ukuran file setelah dienkripsi menjadi 1023 KB.
d. Analisis perbandingan file teks dengan tipe data ".txt" dan ".docx" menggunakan algoritma

Advanced Encryption Standard (AES) untuk mendapatkan hasil perbandingan berupa waktu komputasi dan ukuran file setelah dienkripsi. Hasil keduanya tidak terlalu jauh perbedaannya, untuk waktu komputasi tipe data ".txt" lebih cepat dibandingkan ".docx" karena tipe data ".txt" memiliki struktur yang lebih sederhana.

Daftar Pustaka

[1] Chandra, R, V, H., Ari, K., dan Mahendra, D. Analisis Performa Proses Ekripsi dan Dekripsi Menggunakan Algoritma AES-128 Pada Berbagai Format File. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer,* 3(1): 481-486.

[2] Indrayani, L, A., dan I Made, S. 2019. Implementasi Kriptografi dengan Modifikasi Algoritma Advanced Encryption Standard (AES) untuk Pengamanan File Document. Journal of Informatics and Computer Science, 01(01): 42-47.

[3] Handoyo, J., dan Yulieo, M, S. 2020. Keamanan Dokumen Menggunakan Algoritma Advanced Encryption Standard (AES). *Jurnal Sistem Informasi dan Teknologi,* 3(2): 144-150.

[4] Abdullah, I, N., Dewi, K., dan Mohammad, A. 2018. Aplikasi Enkripsi File Dokumen Menggunakan Metode Algoritma AES (Advanced Encryption Standard) dan OTP (One Time Pad) Berbasis Web Pada PT. MNC Sky Vision. *Jurnal Telematika MKOM,* 10(1):11-16

[5] Lana, F, N. 2020. *Pengamanan File MP3 Menggunakan Algoritma Advanced Encryption Standard (AES).* Skripsi. Universitas Airlangga

[6] Sancaka, T, M, P., dan Veronica, L. 2020. Penerapan Metode Playfair Cipher Dalam Aplikasi Enkripsi Dekripsi File Teks. *Jurnal Ilmiah Elektronika dan Komputer*, 15(02): 260-270.

[7] Kromodimoeljo, S. 2009. *Teori dan Aplikasi Kriptografi.* SPK IT Consulting

[8] Tulloh, A, R. 2016. *Kriptografi Advanced Enciption Standard (AES) Untuk Penyandian File Dokumen ,* Bandung :Universitas Islam.

p-ISSN: 2986-3929
e-ISSN: 3032-1948

Halaman ini sengaja dibiarkan kosong