Implementasi Client-Side File Encryption Berbasis Web Menggunakan AES-GCM

Raga Fatwa Ikhwani - 13222001
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13222001@std.stei.itb.ac.id, [2]ragafatwa@gmail.com

Abstract*—Keamanan data merupakan aspek penting dalam pertukaran informasi digital, sehingga diperlukan mekanisme pengamanan yang mampu menjaga kerahasiaan dan integritas data. Percobaan ini membahas perancangan dan implementasi sistem enkripsi dan dekripsi file berbasis web menggunakan algoritma Advanced Encryption Standard (AES) dengan mode operasi Galois/Counter Mode (GCM). User dapat melakukan enkripsi dan dekripsi file menggunakan kata sandi sebagai dasar pembentukan kunci kriptografi. AES-GCM dipilih karena mampu menyediakan enkripsi dan autentikasi data secara bersamaan. Hasil pengujian menunjukkan bahwa sistem dapat melakukan proses enkripsi dan dekripsi dengan benar serta mampu mendeteksi kegagalan autentikasi ketika kata sandi yang digunakan tidak sesuai atau data terenkripsi mengalami perubahan, sementara pengukuran waktu eksekusi menunjukkan bahwa performa sistem dipengaruhi oleh ukuran file namun tetap efisien untuk aplikasi berbasis web.

***Keywords*—AES, Dekripsi, Enkripsi, File, GCM, Kriptografi**

I. Pendahuluan

Perkembangan teknologi mendorong meningkatnya pertukaran data digital, baik dalam bentuk dokumen, gambar, dan media digital lainnya. Peningkatan intensitas pertukaran data tersebut juga harus diikuti dengan keamanan dalam mengirimkan informasi untuk mencegah pencurian data, penyadapan, dan manipulasi berkas. Oleh karena itu, keamanan data harus diperhatikan khususnya dalam proses unggah dan unduh file.

Salah satu cara dalam menjaga kerahasiaan data adalah dengan menerapkan enkripsi menggunakan kriptografi. Enkripsi memungkinkan data diubah kedalam bentuk yang tidak dapat dipahami tanpa kunci yang sesuai, sehingga hanya pihak yang memiliki kunci yang dapat mengakses isi data tersebut.

Advanced Encryption Standard (AES) merupakan salah satu algoritma kriptografis simetris yang telah menjadi standar internasional dan banyak digunakan dalam berbagai sistem keamanan. Salah satu mode operasi dalam AES adalah Galois/Counter Mode (GCM). GCM menjamin integritas dan autentikasi data melalui mekanisme *authentication tag*. Karakteristik tersebut sangat sesuai dalam implementasi sistem keamanan.

Percobaan ini dibuat dengan tujuan mengimplementasikan sistem enkripsi dan dekripsi file berbasis web menggunakan algoritma AES-GCM. Sistem yang dibuat memungkinkan pengguna melakukan enkripsi dan dekripsi file secara langsung di sisi klien tanpa terhubung server atau koneksi internet. Enkripsi dan dekripsi tersebut dilakukan dengan menggunakan kata sandi sebagai dasar pembentukan kunci kriptografi yang digunakan.

II. Landasan Teori

A. Kriptografi

Kriptografi berasal dari bahasa Yunani, *cryptós* (*secret/hidden*) dan *gráphein* (*writing*), artinya *secret writing* atau *hidden writing* [1]. Menurut Schneier, kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan [1]. Kriptografi menggunakan teknik pengamanan informasi dengan mengubah pesan asli (*plaintext*) menjadi bentuk yang tidak dipahami (*ciphertext*) [1]. Tujuan utama kriptografi adalah untuk menjaga kerahasiaan (*aonfidentiality*), autentikasi (*authentication*), integritas data (*data integrity*), dan anti penyangkalan (*non-repudiation*) [1].

Kriptografi digunakan sejak zaman kuno, teknik-teknik kriptografi sederhana telah digunakan untuk menyembunyikan pesan rahasia [1]. Salah satu awal kriptografi klasik adalah *Caesar Cipher*, yang menggunakan pergeseran huruf alfabet untuk mengenkripsi pesan. Metode kriptografi klasik umumnya menggunakan teknik substitusi dan transposisi karakter.

Gambar 1 *Caesar Cipher* [1]

Keterbatasan kriptografi klasik mendorong berkembangnya kriptografi mekanik dan elektromekanik pada Perang Dunia II seperti mesin enigma [1]. Walaupun lebih komleks dari kriptografi klasik, sistem tersebut tetap dapat dipecahkan dengan pendekatan matematis dan komputasi yang lebih canggih.

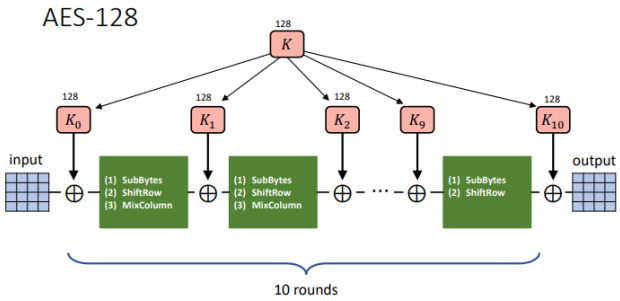
Gambar 2 Mesin Enigma [1]

Kriptografi modern berkembang pesat seiring dengan kemajuan teknologi komputer. Kriptografi modern bergantung pada kerahasiaan kunci, bukan algoritma yang digunakan [2]. Kriptografi modern terbagi menjadi tiga kategori, yaitu kriptografi simetris, kriptografi asimetris, dan fungsi hash [1]. Kriptografi simetris menggunakan satu kunci yang sama untuk enkripsi dan dekripsi. Kriptografi asimetris menggunakan pasangan kunci public dan kunci privat. Sedangkan fungsi hash mengkompresi pesan ukuran sembarang menjadi *message-digest* berukuran *fixed*.

B. Advanced Encryption Standard (AES)

Advanced Encryption Standard (AES) merupakan algoritma kriptografi simetris berbasis *block cipher* yang ditetapkan sebagai standar enkripsi oleh National Institute of Standards and Technology (NIST) pada tahun 2001 [3]. AES dirancang untuk menggantikan Data Encryption Standard (DES) yang dinilai tidak lagi aman [3]. AES dirancang oleh Vincent Rijmen dan Joan Daemen dengan nama Rijndael [3].

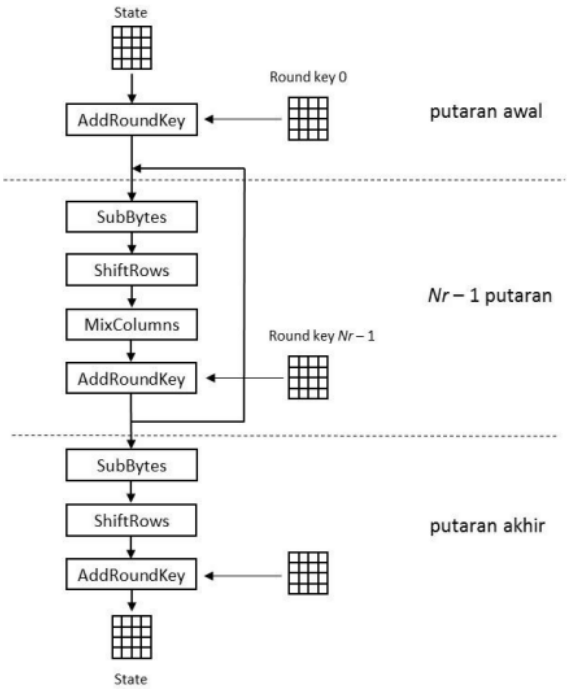
AES bekerja dengan ukuran blok tetap sebesar 128 bit dan mendukung panjang kunci 128 bit, 192 bit, dan 256 bit [3]. Panjang kunci yang berbeda menentukan jumlah putaran (*round*) dalam proses enkripsi dan dekripsi, yaitu 10 putaran untuk kunci 128 bit, 12 putaran untuk kunci 192 bit, dan 14 putaran untuk kunci 256 bit [3]. Semakin panjang kunci yang digunakan, semakin tinggi tingkat keamanan yang diperoleh, namun membutuhkan beban komputasi yang lebih tinggi.



Gambar 3 AES-128

Pada AES-128, blok plaintext direpresentasikan sebagai sebuah matriks dua dimensi berukuran 4x4 byte yang disebut sebagai *state* [3]. Setiap elemen matriks berisi 1 byte, sehingga seluruh *state* mencakup total 16 byte. Seluruh operasi enkripsi dan dekripsi dilakukan terhadap *state* ini melalui transformasi matematis tertentu pada setiap putaran [3].

Proses enkripsi AES diawali dengan operasi AddRoundKey, yaitu penggabungan state awal dengan kunci putaran pertama menggunakan operasi XOR [3]. Setelah itu, dilakukan serangkaian putaran enkripsi dan pada setiap putaran terdiri dari empat transformasi utama, yaitu SubBytes, ShiftRows, MixColumns, dan AddRoundKey [3]. Pada putaran terakhir, transformasi MixColumns tidak lagi diterapkan.



Gambar 4 Garis Besar Algoritma AES [3]

C. Galois/Counter Mode (GCM)

Galois/Counter Mode (GCM) merupakan salah satu mode operasi *block cipher* yang mengkombinasikan mekanisme enkripsi dan autentikasi data dalam satu skema yang terintegrasi [4]. GCM dirancang untuk digunakan dengan algoritma kriptografi simetris seperti AES.

GCM termasuk dalam kategori Authenticated Encryption with Associated Data (AEAD), yaitu skema enkripsi yang mampu mendeteksi setiap perubahan atau manipulasi pada ciphertext. GCM menggabungkan dua mekanisme utama, yaitu *counter* (CTR) untuk enkripsi dan Galois Message Authenticatoin Code (GMAC) untuk proses autentikasi [4].

Pada mekanisme CTR, ciphertext didapatkan dengan cara mengenkripsi nilai *counter* yang kemudian dilakukan operasi XOR dengan plaintext. Mekanisme ini memungkinkan proses enkripsi dilakukan secara paralel, sehingga meningkatkan efisiensi dan kecepatan sistem.

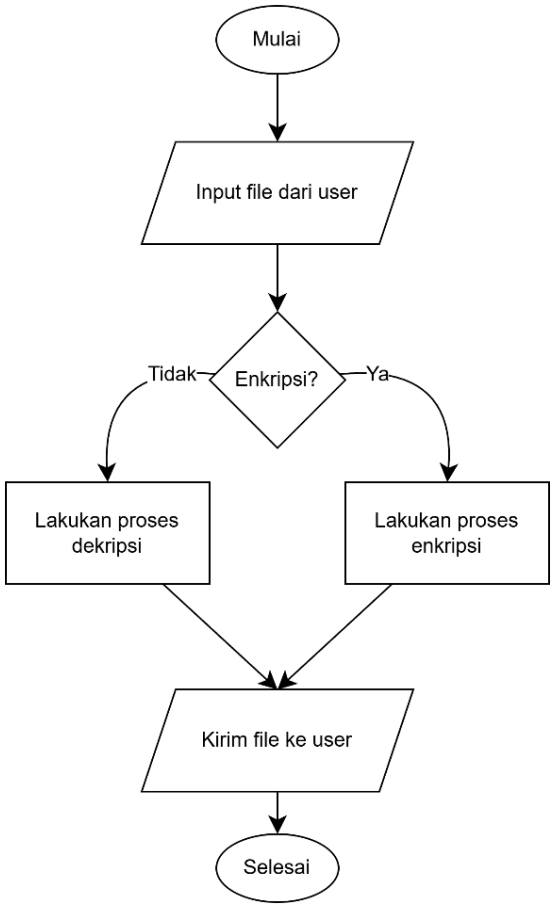
Proses GMAC dilakukan untuk menghasilkan nilai *authentication tag* yang berfungsi sebagai penjamin integritas dan keaslian data. *Authentication tag* ini dihitung berdasarkan ciphertext dan *additional authenticated data* (AAD) jika ada, sehingga setiap perubahan kecil pada data akan menghasilkan tag yang berbeda dan menyebabkan proses dekripsi gagal. Dengan mekanisme ini, GCM mampu mendeteksi kesalahan akibat manipulasi data maupun penggunaan kunci yang tidak sesuai.

III. IMPLEMENTASI

Pada percobaan ini, dilakukan implementasi sistem enkripsi dan dekripsi file berbasis web menggunakan algoritma AES dengan mode operasi GCM. Sistem dirancang agar pengguna dapat melakukan proses enkripsi dan dekripsi file secara langsung melalui antarmuka web dengan memanfaatkan kata sandi sebagai dasar pembentukan kunci kriptografi.

A. Perancangan Arsitektur Aplikasi

Sistem enkripsi file berbasis web ini terdiri dari dua komponen utama, yaitu antarmuka pengguna dan pemrosesan kriptografi. Antarmuka pengguna bertugas menerima masukan berupa file dan kata sandi, serta menampilkan hasil keluaran berupa file terenkripsi atau terdekripsi. Sementara itu, kriptografi bertugas dalam proses pembentukan kunci, enkripsi, dekripsi.



Gambar 5 Flowchart Sistem Level 0

Arsitektur sistem dibangun menggunakan bahasa pemrograman Python dengan bantuan *framework* web Flask dan *library* kriptografi yang menyediakan implementasi AES-GCM.

B. Pembentukan Kunci Kriptografi

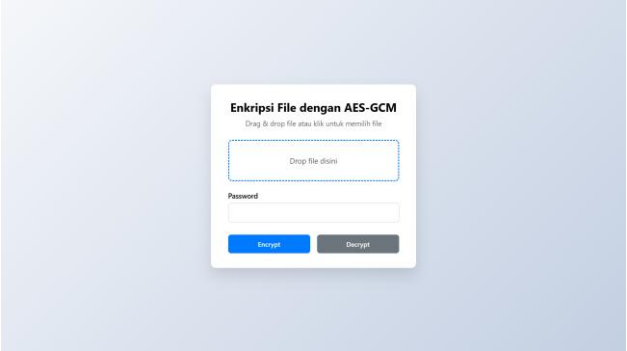
Kunci kriptografi tidak dimasukkan secara langsung oleh pengguna, melainkan dibangkitkan dari kata sandi yang diberikan. Kata sandi tersebut diproses menggunakan fungsi hash untuk menghasilkan nilai kunci dengan panjang tetap yang sesuai dengan kebutuhan algoritma AES. Kunci yang dihasilkan kemudian digunakan sebagai kunci simetris pada proses enkripsi dan dekripsi AES-GCM.

C. Enkripsi dan Dekripsi

Proses enkripsi dan dekripsi akan dimulai jika user sudah memasukkan kata sandi (tanpa minimal panjang). Pada proses enkripsi, output yang akan dikirimkan ke user yaitu sebuah file yang sudah dienkripsi dengan tipe file .enc. Sedangkan pada proses dekripsi, outputnya disesuaikan dengan file yang terenkripsi dari user. Proses dekripsi akan gagal jika kata sandi yang digunakan tidak sesuai dengan kata sandi enkripsi.

IV. HASIL DAN ANALISIS

A. Tampilan Utama



Gambar 6 Tampilan Sistem

Gambar 6 di atas merupakan tampilan utama dari sistem yang dibuat. Sistem hanya memiliki satu halaman tampilan saja untuk melakukan enkripsi dan dekripsi. Pada halaman ini, terdapat kolom untuk *drop* file yang akan dilakukan proses enkripsi atau dekripsi. Selanjutnya terdapat kolom untuk mengetik kata sandi yang akan digunakan sebagai kunci enkripsi atau dekripsi file.

B. Percobaan Enkripsi File

Pada percobaan ini dilakukan proses enkripsi file PNG dengan ukuran yang relatif kecil yaitu 200 KB dan file MP4 dengan ukuran 1.37 GB. Masing-masing file tersebut dihitung waktu yang dibutuhkan untuk selesai melakukan enkripsi.

Tabel 1 Hasil Percobaan Enkripsi File

File	Waktu
PNG	Sangat cepat (instan)
MP4	16 detik

Hasilnya, sistem berhasil melakukan enkripsi dengan waktu yang sangat cepat (instan) untuk file PNG. Sedangkan untuk file MP4, sistem membutuhkan waktu sekitar 16 detik.

C. Percobaan Dekripsi File

Pada percobaan ini dilakukan proses dekripsi file dengan file hasil enkripsi pada percobaan sebelumnya. Masing-masing file tersebut dihitung waktu yang dibutuhkan untuk selesai melakukan dekripsi.

Tabel 2 Hasil Percobaan Dekripsi File

File	Waktu
PNG yang terenkripsi	Sangat cepat (instan)
MP4 yang terenkripsi	9 detik

Hasilnya, sistem berhasil melakukan dekripsi dengan waktu yang sangat cepat (instan) untuk file PNG. Sedangkan untuk file MP4, sistem membutuhkan waktu sekitar 9 detik (lebih cepat dari proses enkripsi).

D. Percobaan Mengubah File yang Terenkripsi

Pada percobaan ini dilakukan perubahan pada file yang telah terenkripsi. File yang digunakan sama seperti file yang digunakan dalam proses dekripsi (file PNG dan MP4). Perubahan file dilakukan dengan mengubah ekstensi file dari .enc menjadi ekstensi file masing-masing (PNG dan MP4).

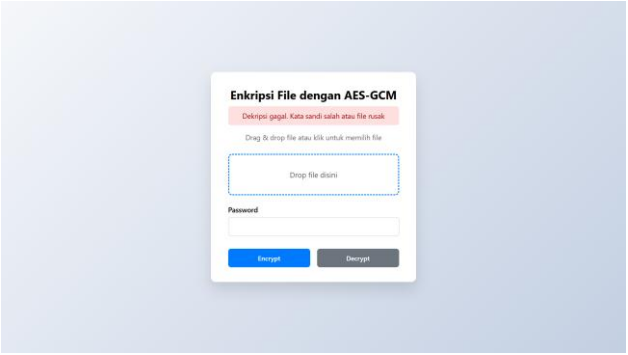
Tabel 3 Hasil Percobaan Mengubah File yang Terenkripsi

File	Hasil
PNG yang terenkripsi	Tidak dapat dibuka
MP4 yang terenkripsi	Tidak dapat dibuka

Hasilnya, kedua file yang terenkripsi tersebut tidak dapat dibuka meskipun ekstensi file sudah sesuai dengan file yang belum terenkripsi. Hal ini membuktikan bahwa proses enkripsi berjalan dengan baik dan sistem mampu menjaga integritas data dari manipulasi.

E. Percobaan Dekripsi dengan Kata Sandi Berbeda

Pada percobaan ini dilakukan proses dekripsi, namun dengan menggunakan kata sandi yang berbeda dengan kata sandi saat proses enkripsi.



Gambar 7 Hasil Percobaan Dekripsi dengan Kata Sandi Berbeda

Hasilnya, sistem akan menampilkan pesan kesalahan sesuai dengan Gambar 7. Hal ini membuktikan bahwa dekripsi tidak dapat dilakukan dengan kata sandi yang berbeda dan sistem tidak akan mendekripsi file dengan kata sandi yang salah.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

- Sistem dapat menampilkan antarmuka untuk user yang akan melakukan enkripsi atau dekripsi file
- Sistem dapat melakukan enkripsi dan dekripsi sesuai dengan file dan kata sandi yang diberikan oleh user
- Sistem membutuhkan waktu lebih untuk melakukan enkripsi dan dekripsi file dengan ukuran yang relatif besar
- Sistem tidak akan melakukan dekripsi jika kata sandi yang digunakan salah atau tidak sesuai dengan kata sandi enkripsi

- File yang sudah terenkripsi tidak dapat dilakukan perubahan/manipulasi data

B. Saran

Untuk pengembangan percobaan serupa di masa mendatang, terdapat beberapa saran yang dapat diterapkan, khususnya untuk melakukan analisis.

- Sistem dapat menampilkan waktu yang dibutuhkan dalam proses enkripsi atau dekripsi
- Sistem dapat menerima beberapa file sekaligus untuk dilakukan proses enkripsi atau dekripsi

PRANALA GITHUB

https://github.com/ragafatwa/Web_Encrypt

REFERENCES

[1] Munir, Rinaldi. (2025). "Pengantar Kriptografi". https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2025-2026/01-Pengantar-Kriptografi-(2025).pdf.

[2] Munir, Rinaldi. (2025). "Kriptografi Modern". https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2025-2026/12-Kripto-modern-2025.pdf.

[3] Munir, Rinaldi. (2025). "Beberapa Block Cipher (Bagian 2)". https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2025-2026/17-Beberapa-block-cipher-bagian2-2025.pdf.

[4] https://www.cincopa.com/learn/introduction-to-aes-gcm-mode-and-its-advantages, diakses pada 22.34 WIB, 25 Desember 2025.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 26 Desember 2025

Raga Fatwa Ikhwani
13222001