ANALISIS PERFORMA DAN KEAMANAN IMPLEMENTASI KRIPTOGRAFI AES UNTUK PENYANDIAN DOKUMEN BERBASIS WEB

Andi Syam Aswandi [1], Muh. Nurtanzis Sutoyo [2], Anjar Pradipta [3]
[1,2,3] Sistem Informasi, Universitas Sembilanbelas November Kolaka
andisyamaswandi@gmail.com

ABSTRAK

Kemajuan teknologi informasi telah mendorong peningkatan kebutuhan akan keamanan data yang diunggah dan disimpan secara daring, terutama dalam bentuk dokumen teks. *Algoritma Advanced Encryption Standard* (AES) sering digunakan untuk melindungi data sensitif karena kekuatannya dalam menghadapi serangan kriptografi. Namun, tantangan tetap ada dalam penerapan AES pada platform berbasis web, terutama terkait efisiensi proses enkripsi dan dekripsi di lingkungan jaringan yang bervariasi. Penelitian ini mengembangkan sistem enkripsi berbasis web yang memanfaatkan algoritma AES untuk penyandian dokumen. Sistem ini memungkinkan pengguna untuk mengunggah dokumen teks yang akan dienkripsi dan diunduh kembali dalam bentuk terenkripsi guna menjaga kerahasiaan data dengan mengimplementasikan algoritma Advanced Encryption Standard (AES) dalam penyandian dokumen berbasis web dan membandingkan kinerjanya dengan algoritma lain seperti RSA dan Triple DES. Pengujian dilakukan untuk mengukur waktu enkripsi dan dekripsi, konsumsi sumber daya, serta tingkat keamanan sistem terhadap serangan siber. Hasil pengujian menunjukkan bahwa AES memiliki waktu enkripsi rata-rata 12 ms dan dekripsi 15 ms, lebih cepat dibandingkan RSA (98 ms untuk enkripsi dan 120 ms untuk dekripsi) serta Triple DES (55 ms untuk enkripsi dan 65 ms untuk dekripsi). Selain itu, AES menunjukkan efisiensi sumber daya yang lebih baik dengan tingkat keamanan yang tetap optimal. Sistem diuji menggunakan metode blackbox untuk memastikan validitasnya, dan hasilnya menunjukkan bahwa sistem berjalan sesuai spesifikasi. Dengan demikian, penelitian ini memberikan kontribusi dalam pengembangan sistem keamanan berbasis web yang efisien dan aman, serta dapat menjadi referensi bagi penelitian lanjutan dalam bidang keamanan data digital.

Keyword : Kemanan Data, Kriptografi, AES**.**

1. PENDAHULUAN

Keamanan data dalam aplikasi berbasis web menjadi perhatian utama seiring dengan meningkatnya ancaman terhadap privasi dan integritas informasi. *Advanced Encryption Standard* (AES) telah diakui sebagai algoritma kriptografi simetris yang andal untuk melindungi data sensitif. AES menawarkan kecepatan dan keamanan yang unggul dibandingkan algoritma lain seperti *Data Encryption Standard* (DES) dan *Rivest Cipher* 4 (RC4) [1].

Beberapa penelitian telah mengkaji implementasi AES dalam berbagai konteks. Mengaplikasikan AES-128 pada mikrokontroler NodeMCU menggunakan arsitektur web *service* REST untuk mengamankan pengiriman data. Selain itu, mengembangkan aplikasi penandatanganan dokumen digital dengan metode AES, menunjukkan efektivitasnya dalam menjaga integritas dokumen. Namun, studi yang secara khusus membahas penerapan AES dalam enkripsi dokumen berbasis web masih terbatas[2].

Membandingkan kinerja AES dengan algoritma enkripsi lain dalam konteks komputasi awan. Hasil penelitian mereka menunjukkan bahwa AES memiliki keunggulan dalam efisiensi waktu enkripsi dan dekripsi dibandingkan dengan algoritma lain seperti RSA dan *Blowfish* [3]. Selain itu, penelitian menganalisis efektivitas enkripsi AES dalam aplikasi *mobile*, yang menunjukkan bahwa AES mampu memberikan perlindungan optimal terhadap data tanpa mengorbankan kinerja sistem [4].

Berdasarkan analisis tersebut, penelitian ini bertujuan untuk mengembangkan sistem enkripsi dokumen berbasis web menggunakan algoritma AES. Dengan mengintegrasikan AES ke dalam platform web, penelitian ini berkontribusi pada peningkatan keamanan data dalam aplikasi online. Selain itu, studi ini juga akan membandingkan kinerja sistem yang dikembangkan dengan penelitian sebelumnya untuk menilai keunggulan dan kelemahan pendekatan yang diusulkan.

Dengan adanya analisis dan perbandingan terhadap studi sebelumnya, penelitian ini diharapkan dapat memberikan kontribusi nyata dalam bidang keamanan informasi berbasis web serta menjadi referensi bagi penelitian di masa mendatang.

2. TINJAUAN PUSTAKA

Kriptografi merupakan disiplin ilmu yang berfokus pada perlindungan informasi dengan cara mengubahnya menjadi bentuk yang tidak dapat dibaca oleh pihak yang tidak berwenang. Salah satu algoritma yang banyak digunakan dalam kriptografi modern adalah *Advanced Encryption Standard* (AES), yang dikenal karena efisiensinya dalam enkripsi dan dekripsi data [5]. AES menawarkan keunggulan dalam hal kecepatan pemrosesan dan tingkat keamanan yang tinggi dibandingkan algoritma kriptografi lainnya, seperti DES dan RC4.

Sejumlah penelitian telah mengkaji penerapan AES di berbagai lingkungan teknologi informasi. Mengimplementasikan AES-128 dalam pengiriman data berbasis web *service* REST menggunakan *mikrokontroler* NodeMCU, menunjukkan efisiensi algoritma ini dalam mengamankan data di lingkungan IoT. Mengembangkan aplikasi penandatanganan dokumen digital menggunakan AES, yang memperkuat keamanan integritas dokumen dengan mekanisme enkripsi yang lebih efisien dibandingkan metode konvensional [6].

Selain itu, [7] penelitian membandingkan performa AES dengan algoritma lain dalam konteks komputasi awan, di mana AES terbukti lebih cepat dalam proses enkripsi dan dekripsi dibandingkan RSA dan *Blowfish*. [8] juga menemukan bahwa AES sangat cocok untuk aplikasi mobile karena memberikan keseimbangan optimal antara keamanan data dan performa sistem.

Penerapan kriptografi pada dokumen digital dalam sistem berbasis web memungkinkan proses menunjukkan bahwa dalam implementasi algoritma *vignere cipher* pada pengamanan data, penggunaan metode kriptografi berperan signifikan dalam melindungi data dari akses yang tidak sah. Meskipun *vignere cipher* efektif untuk enkripsi data berbasis teks sederhana, kebutuhan terhadap algoritma yang lebih kuat seperti AES menjadi sangat penting untuk data yang lebih kompleks [9].

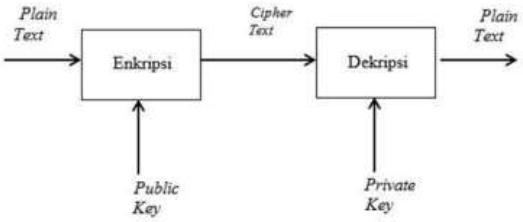
Walaupun banyak penelitian telah membahas implementasi AES dalam berbagai aplikasi, studi yang secara khusus menyoroti implementasi AES dalam enkripsi dokumen berbasis web masih terbatas. Oleh karena itu, penelitian ini bertujuan untuk mengisi kesenjangan tersebut dengan mengembangkan sistem enkripsi dokumen berbasis web yang menggunakan AES sebagai algoritma utama. Dengan membandingkan hasil penelitian ini dengan studi sebelumnya, penelitian ini dapat memberikan wawasan mengenai keunggulan dan kelemahan pendekatan yang digunakan, serta memberikan rekomendasi untuk peningkatan keamanan data di masa mendatang.

2.1. Kriptografi

Kriptografi berasal dari bahasa Yunani, dengan kata '*cryptos*' berarti 'rahasia' dan '*graphein*' berarti 'tulisan'. Secara harfiah, istilah ini berarti 'tulisan rahasia'. Definisi kriptografi bervariasi dalam berbagai literatur. Dalam literatur sebelum tahun 1980-an, kriptografi didefinisikan sebagai ilmu dan seni untuk melindungi pesan dengan menyandikan informasi ke dalam format yang sulit dipahami. Definisi ini relevan pada masa lalu, ketika kriptografi digunakan dalam komunikasi sensitif, seperti di militer, diplomasi, dan intelijen. Namun, kini cakupan kriptografi telah berkembang mencakup privasi, integritas data, autentikasi, serta *nonrepudiation* [10].

2.2. Kriptografi Simetris

Algoritma simetris, juga dikenal sebagai kriptografi konvensional, menggunakan kunci yang identik untuk proses enkripsi dan dekripsi. Algoritma ini dibagi menjadi dua jenis utama: algoritma aliran (*stream cipher*) dan algoritma blok (*block cipher*). Pada algoritma aliran, data dienkripsi per bit atau per *byte*, sedangkan dalam algoritma blok, data diproses dalam unit blok yang terdiri dari beberapa bit atau *byte* [11]. Diagram alur dari proses kriptografi simetris dapat dilihat pada Gambar 1.

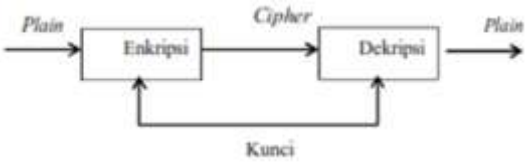


Gambar 1. Proses kriptografi simetris

Gambar 1 Menampilkan proses kriptografi simetris di mana teks asli atau plaintext dienkripsi menjadi ciphertext—pesan terenkripsi yang tidak dapat dibaca langsung—menggunakan kunci tertentu. Proses dekripsi kemudian dilakukan untuk mengembalikan ciphertext menjadi plaintext atau pesan asli.

2.3. Kriptografi Asimetris

Kriptografi asimetris merupakan metode yang memanfaatkan dua jenis kunci, yaitu kunci publik dan kunci privat. Kunci publik dapat diakses secara terbuka karena tidak dapat digunakan untuk langsung mengakses data, sementara kunci privat hanya diketahui oleh pihak tertentu yang memiliki otoritas untuk mengakses informasi. Proses dari kriptografi asimetris digambarkan pada Gambar 2.2 [12].



Gambar 2. Proses kriptografi asimetris

Gambar 2 Menunjukkan bahwa teks asli (*plaintext*) dienkripsi menggunakan kunci publik sehingga menghasilkan *ciphertext* atau pesan yang terenkripsi dan tidak bisa dibaca langsung. Proses dekripsi dilakukan dengan kunci privat untuk mengubah *ciphertext* kembali ke bentuk *plaintext* atau pesan asli.

2.4. Advanced Encryption Standard

Advanced Encryption Standard (AES) adalah algoritma kriptografi yang dirancang untuk mengamankan data. AES merupakan algoritma

simetris berbasis blok yang digunakan untuk proses enkripsi (*encipher*) dan dekripsi (*decipher*) data. Pada proses enkripsi, data diubah menjadi *ciphertext*, yaitu bentuk terenkripsi yang tidak dapat dibaca langsung. Sebaliknya, proses dekripsi mengubah *ciphertext* kembali menjadi *plaintext*, atau bentuk data semula. AES mendukung panjang kunci 128, 192, dan 256 bit untuk mengenkripsi dan mendekripsi data dalam blok berukuran 128 bit. Pemilihan panjang kunci dan ukuran blok data ini menentukan banyaknya tahap yang harus dilalui selama proses enkripsi dan dekripsi [13]. Tabel 2.1 memperlihatkan perbandingan jumlah tahap untuk tiap pilihan kunci dan blok data.

Tabel 1. Jumlah proses berdasarkan bit blok dan kunci

Panjang Kunci Dalam bit	Panjang Kunci (Nk) Dalam words	Ukuran Blok Data (Nb) Dalam words	Jumlah Proses (Nr)
128	4	4	10
192	6	4	12
256	8	4	14

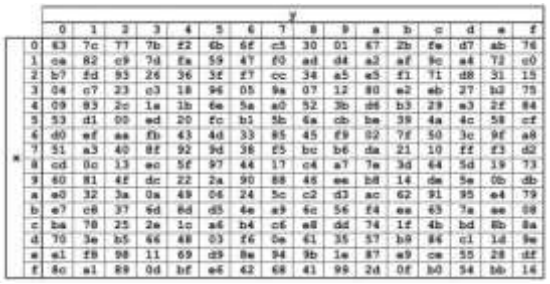
Blok data *input* dan kunci pada algoritma AES diproses dalam bentuk *array*, di mana tiap elemen *array* sebelum menghasilkan *ciphertext* disebut sebagai state . Proses transformasi *state* melibatkan empat tahap utama: *AddRoundKey*, *SubBytes*, *ShiftRows*, dan *MixColumns*. Khusus untuk *MixColumns*, tahap ini hanya diterapkan hingga putaran terakhir, sementara tiga tahap lainnya berulang di setiap putaran. Dalam proses dekripsi, langkah-langkah ini dibalik untuk memulihkan teks asli. Karena proses enkripsi terdiri dari beberapa tahap, dibutuhkan beberapa subkunci (*subkey*) yang digunakan pada setiap tahap. Pengembangan subkunci ini dilakukan karena *subkey* yang dibutuhkan bisa mencapai ribuan bit, sedangkan panjang kunci yang disediakan biasanya hanya berkisar antara 128 hingga 256 bit. Total subkunci yang diperlukan dihitung dengan formula Nb(Nr+1), di mana Nb mewakili ukuran blok data dalam satuan *word*, dan Nr adalah jumlah putaran yang dilakukan. Sebagai contoh, untuk blok data dan kunci 128 bit (4 *word*), proses enkripsi akan berlangsung dalam 10 putaran sehingga total subkunci mencapai 1408 bit (4(10+1)=44 *word*). Proses perluasan kunci ini dikenal sebagai *key schedule*, yang memastikan setiap tahap enkripsi dan dekripsi memiliki subkunci yang berbeda [14].

2.5. Proses Enkripsi

Penggunaan kriptografi sangat penting untuk melindungi data yang dikirim agar tetap terjaga kerahasiaannya. Pesan asli, atau *plaintext*, diubah menjadi bentuk kode yang tidak dapat dibaca tanpa kunci tertentu. Proses ini dikenal sebagai enkripsi, dengan hasilnya disebut *ciphertext*. Proses enkripsi dapat dibandingkan dengan kamus, yang memuat daftar kata dan definisi. Namun, pada enkripsi, teks biasa diubah menggunakan algoritma khusus yang menerapkan kode pada data sesuai kebutuhan [15].

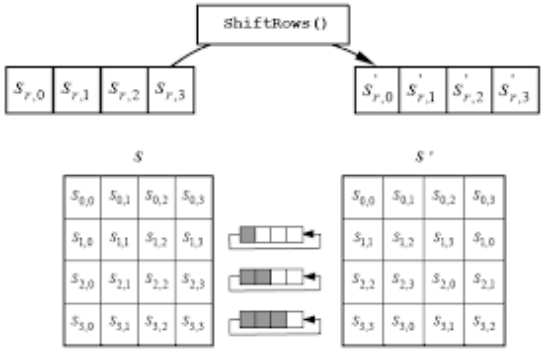
Pada algoritma enkripsi AES, *plaintext* diubah menjadi *ciphertext* menggunakan formula C=E(M)C=E(M), di mana MM adalah pesan asli, EE adalah proses enkripsi dengan kunci, dan CC adalah hasil terenkripsi (*ciphertext*) . Proses enkripsi AES terdiri dari beberapa tahap utama:

1. *AddRoundKey* – Pada tahap ini, XOR dilakukan antara *state* awal (*plaintext*) dan *cipherkey*. Langkah ini sering disebut sebagai *initial round*, di mana setiap *byte* pada *state* digabungkan dengan *byte* pada *subkey* menggunakan operasi XOR, dengan ukuran *subkey* yang sama dengan *state*.
2. Putaran Sebanyak (Nr - 1) Kali – Dalam setiap putaran, beberapa operasi diterapkan:
   a. *SubBytes* – Proses ini menggantikan setiap *byte* pada *state* dengan nilai yang sesuai dari tabel substitusi (*S-Box*). Tabel *S-Box* ini terdiri dari 16x16 kolom, masing-masing berukuran 1 *byte*.



Gambar 3. Tabel *s-box*

   b. *ShiftRows* – Merupakan proses pergeseran setiap baris pada *state array*. Pada baris pertama, tidak ada pergeseran; baris kedua bergeser satu *byte*, baris ketiga dua *byte*, dan baris keempat tiga *byte*. Pergeseran ini bersifat melingkar atau *wrapping*.



Gambar 4. Proses *shiftrows*

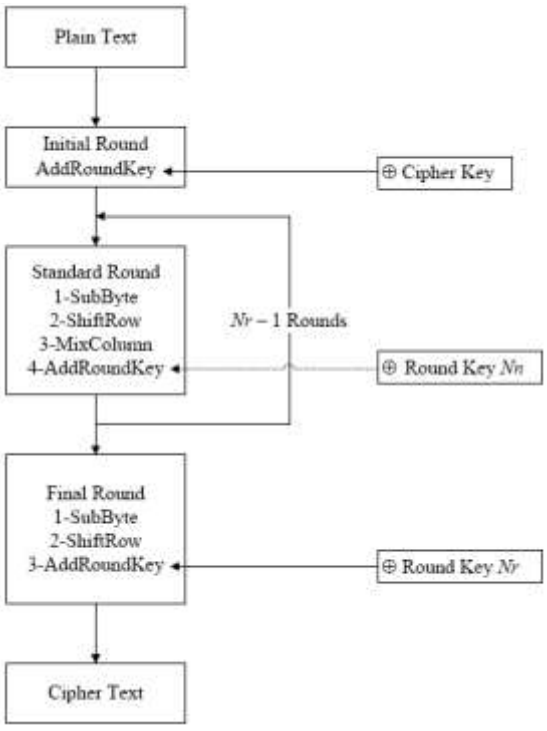
   c. *MixColumns* – Tahap ini bertujuan untuk melakukan pengacakan data di setiap kolom *state*. Operasi ini bekerja dengan

menggabungkan empat *byte* dari setiap kolom melalui transformasi linier, yang meningkatkan kompleksitas hasil enkripsi dan membuat proses dekripsi tanpa kunci semakin sulit.

d. *AddRoundKey* – Pada tahap ini, dilakukan operasi XOR antara state terbaru dengan *roundkey*. Proses ini memastikan bahwa setiap tahap enkripsi mendapatkan kunci yang berbeda, menambah lapisan keamanan tambahan pada *ciphertext* .

3. Putaran Akhir (*Final Round*) – Tahap akhir enkripsi AES hanya terdiri dari tiga proses:
   a. *SubBytes* – Setiap *byte* pada state digantikan dengan nilai dari tabel substitusi (*S-Box*).
   b. *ShiftRows* – *Byte-byte* di setiap baris mengalami pergeseran melingkar untuk meningkatkan kerumitan enkripsi.
   c. *AddRoundKey* – *State* akhir diubah menggunakan *roundkey* terakhir untuk menghasilkan *ciphertext final*.

   Ilustrasi dari tahapan ini dapat dilihat pada Gambar 5.



Gambar 5. Proses enkripsi aes

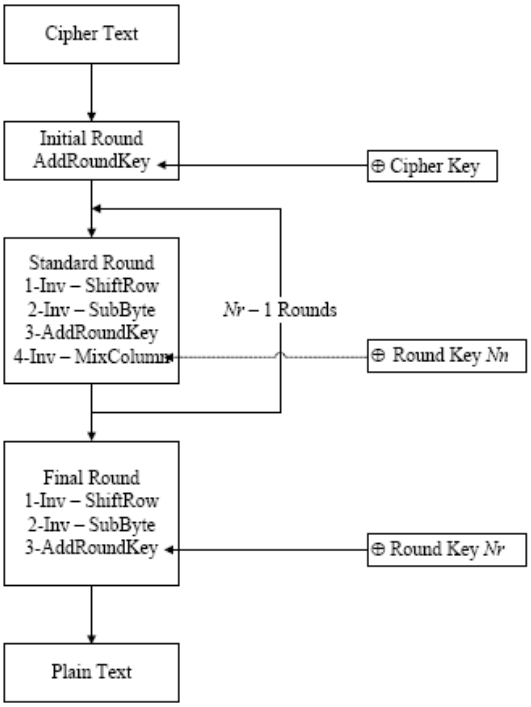
2.6. Proses Dekripsi

Dekripsi adalah kebalikan dari enkripsi, di mana *ciphertext* dikonversi kembali ke bentuk aslinya, yaitu *plaintext*. Algoritma yang digunakan dalam proses dekripsi berbeda dari enkripsi, namun tetap memerlukan kunci rahasia yang sama untuk membongkar hasil enkripsi. Dekripsi AES melibatkan pengembalian *ciphertext* menjadi *plaintext*, yang dirumuskan sebagai M=D(C)M=D(C), di mana CC adalah *ciphertext*, DD adalah proses dekripsi dengan kunci rahasia, dan MM adalah pesan asli setelah melewati beberapa putaran dekripsi sebanyak Nr−1Nr−1 kali [16].

Pada setiap putaran dekripsi, langkah-langkah berikut dilakukan:

1. *InverseShiftRow* – Ini adalah kebalikan dari proses *ShiftRows*, di mana *byte* pada setiap baris digeser ke kanan, berbeda dengan enkripsi yang menggeser *byte* ke kiri.
2. *InverseSubBytes* – Proses ini merupakan kebalikan dari *SubBytes*, di mana setiap *byte* pada state dikembalikan ke bentuk aslinya menggunakan tabel substitusi terbalik.
3. *AddRoundKey* – Operasi XOR dilakukan antara *state* terbaru dan *round key*, yang sama dengan proses enkripsi.
4. *InverseMixColumns* – Data di setiap kolom *array state* dikacaukan dengan metode inversi untuk membalik transformasi linier yang dilakukan selama enkripsi.
5. Putaran Akhir (*Final Round*) – Putaran terakhir terdiri dari:
   a. InverseShiftRow
   b. InverseSubBytes
   c. AddRoundKey

Tahapan-tahapan ini berakhir dengan terbentuknya kembali plaintext, seperti terlihat pada Gambar 6.



Gambar 6. Proses dekripsi aes

3. METODE PENELITIAN

Penelitian ini mengimplementasikan algoritma *Advanced Encryption Standard* (AES) untuk keamanan dokumen berbasis web. Metode

penelitian yang digunakan terdiri dari beberapa tahap, yaitu perancangan sistem, pengumpulan data, implementasi algoritma, pengujian, dan analisis hasil. Berikut adalah penjelasan tiap tahap.

3.1. Perancangan Sistem

Penelitian ini mengembangkan sistem enkripsi dokumen berbasis web menggunakan algoritma AES. Perancangan sistem melibatkan desain arsitektur yang mencakup antarmuka pengguna, sistem autentikasi, serta modul enkripsi dan dekripsi. Sistem ini memungkinkan pengguna mengunggah dokumen yang akan dienkripsi, kemudian mengunduh hasil enkripsi dalam format yang aman.

3.2. Pengumpulan Data

Data yang digunakan dalam penelitian ini berupa dokumen dalam berbagai format seperti PDF, DOCX, dan TXT. Pengujian dilakukan dengan menggunakan dokumen uji untuk mengukur performa enkripsi dan dekripsi serta memastikan keamanan data. Selain itu, dataset tambahan digunakan untuk membandingkan performa AES dengan algoritma lain seperti RSA dan Triple DES.

3.3. Implementasi Algoritma AES

Pada tahap ini, algoritma AES diimplementasikan dalam sistem web dan dibandingkan dengan algoritma lain seperti RSA dan Triple DES. Perbandingan ini dilakukan berdasarkan parameter kecepatan enkripsi dan dekripsi, penggunaan sumber daya, serta tingkat keamanan. AES dikenal memiliki performa lebih cepat dibandingkan RSA yang menggunakan metode asimetris, sementara Triple DES memiliki tingkat keamanan lebih tinggi tetapi dengan performa yang lebih lambat dibandingkan AES.

3.4. Pengujian dan Evaluasi

Pengujian dilakukan menggunakan metode *blackbox* untuk mengukur validitas sistem, di mana setiap komponen diuji untuk memastikan bahwa sistem berjalan sesuai dengan spesifikasi. Selain itu, pengujian performa dilakukan dengan mengukur waktu enkripsi dan dekripsi, serta dampak ukuran dokumen terhadap kinerja sistem.

3.5. Analisis Hasil

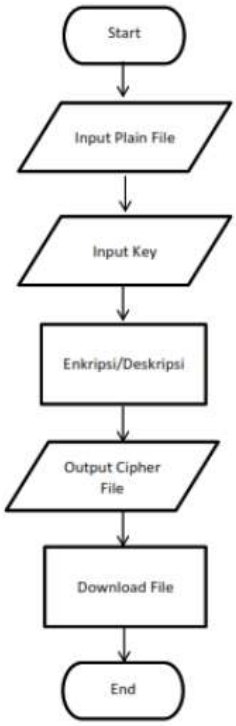
Hasil pengujian performa ditampilkan dalam bentuk tabel untuk menunjukkan efektivitas enkripsi dan dekripsi menggunakan AES dibandingkan dengan RSA dan Triple DES. Analisis hasil mencakup kecepatan proses, konsumsi sumber daya, dan tingkat keamanan terhadap potensi serangan. Selain itu, aspek keamanan juga dianalisis lebih lanjut dengan mempertimbangkan kemungkinan serangan seperti *brute-force* attack, man-in-the-middle attack, serta upaya dekripsi tanpa kunci yang valid. Dengan demikian, penelitian ini memberikan gambaran yang lebih luas mengenai keunggulan dan keterbatasan implementasi AES dalam sistem enkripsi berbasis web.

Untuk memperkuat aspek teknis, berikut adalah persamaan utama dalam algoritma AES:

a. SubBytes Transformation:
   Proses substitusi setiap *byte* dalam state menggunakan nilai dari tabel substitusi (S-Box).
b. ShiftRows Transformation:
   $S'(x,y) = S(x, \left(y + shift(x)\right) \quad mod\ 4)$
   Pergeseran byte dalam state untuk meningkatkan difusi data.
c. MixColumns Transformation:
   $$S'(x,y) = M \cdot S(x,y)$$
   Proses transformasi linier untuk memperkuat enkripsi.
d. AddRoundKey:
   $$S'(x,y) = S(x,y) \oplus K\ (x,y)$$
   Operasi XOR antara state dengan kunci enkripsi pada setiap putaran.

4. HASIL DAN PEMBAHASAN

4.1. Flowchart



Gambar 7. Flowchart

Gambar 7 : flowchart di atas yang menggambarkan proses enkripsi dan dekripsi menggunakan algoritma *Advanced Encryption Standard* (AES).

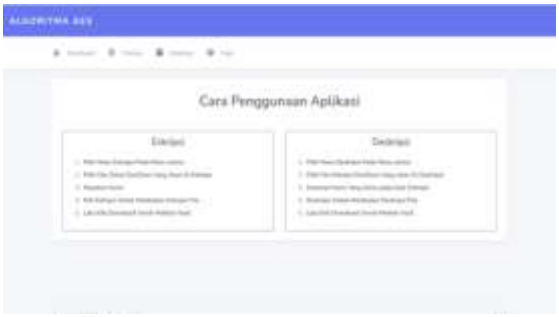
4.2. Halaman Utama

Menampilkan antarmuka utama sistem enkripsi dokumen berbasis web yang terdiri dari beberapa menu utama, seperti home, enkripsi, dekripsi, dan bantuan. Seperti pada Gambar 8.

Gambar 8. Halaman utama

4.3. Halaman Help

Berisi panduan penggunaan sistem yang menjelaskan bagaimana pengguna dapat melakukan proses enkripsi dan dekripsi dengan langkah-langkah yang jelas. Seperti pada Gambar 9.



Gambar 9. Halaman help

4.4. Halaman Enkripsi

Memperlihatkan proses unggah dokumen, input kunci enkripsi, dan hasil enkripsi yang dapat diunduh oleh pengguna. Seperti pada Gambar 10.

Gambar 10. Halaman enkripsi

4.5. Halaman Dekripsi

Menunjukkan tampilan di mana pengguna dapat mengunggah dokumen terenkripsi, memasukkan kunci dekripsi, dan mengunduh kembali dokumen dalam bentuk *plaintext*. Seperti pada Gambar 11.

Gambar 11. Halaman dekripsi

4.6. Halaman Download

Menampilkan antarmuka tempat pengguna dapat mengunduh hasil enkripsi maupun dekripsi dokumene. Seperti pada Gambar 12.

Gambar 12. Halaman download

4.7. Pengujian Blackbox

Berikut ini adalah kasus untuk menguji perangkat lunak yang dibangun menggunakan metode *blackbox* dengan teknik *boundary value.*

Tabel 2. Pengujian blackbox

No	Skenario	Hasil	Pengamatan	Keterangan
1	Klik *menu home/dashboard*	Sistem akan menampilkan halaman utama	Sistem menampilkan halaman utama	*Valid*
2	Klik *menu* enkripsi	Sistem akan menampilkan halaman enkripsi	Sistem menampilkan halaman enkripsi	*Valid*
3	Klik *menu* deskripsi	Sistem akan menampilkan halaman deskripsi	Sistem menampilkan halaman deskripsi	*Valid*
4	Klik *browser* untuk memasukkan *file* dan masuan kunci	Sistem akan Menampilkan *Menu folder file*	Sistem menampilkan *menu folder file*	*Valid*
5	pilih *file* yang akan di enkripsi kemudian klik *open*	Sistem akan menampilkan	Sistem menampilkan halaman enkripsi	*Valid*

No	Skenario	Hasil	Pengamatan	Keterangan
		halaman enkripsi		
6	Klik tombol enkripsi	Sistem akan menampilkan halaman *download*	Sistem menampilkan halaman *download*	*Valid*
7	Klik tombol *download*	Sistem akan Mendownload hasil enkripsi	Sistem akan Mendownload hasil enkripsi	*Valid*
8	Klik *file* yang telah terdownload	Sistem akan menampilkan hasil enkripsi	Sistem akan menampilkan hasil enkripsi	*Valid*
1	Klik *browser* untuk memasukkan *file* dan masuan kunci	Sistem akan Menampilkan *menu folder file*	Sistem menampilkan *menu folder file*	*Valid*
2	pilih *file* yang akan di deskripsi kemudian klik *open*	Sistem akan menampilkan halaman deskripsi	Sistem menampilkan halaman deskripsi	*Valid*
3	Klik tombol deskripsi	Sistem akan menampilkan halaman download	Sistem menampilkan halaman download	*Valid*
4	Klik tombol *download*	Sistem akan Mendownload hasil deskripsi	Sistem akan Mendownload hasil deskripsi	*Valid*
5	Klik *file* yang telah terdownload	Sistem akan menampilkan hasil deskripsi	Sistem akan menampilkan hasil deskripsi	*Valid*

Pengujian *blackbox* pada *implementasi kriptografi Advanced Encryption Standard* (AES) untuk penyandian dokumen berbasis web dilakukan dengan menguji setiap komponen yang mewakili halaman-halaman utama dalam sistem. Setiap komponen memiliki metode atau tombol yang diuji untuk memastikan bahwa tindakan yang dilakukan sesuai dengan hasil yang diharapkan.

Sebagai contoh, pada skenario pengujian pertama, ketika pengguna memilih menu beranda, hasil yang diharapkan adalah tampilan halaman menu utama. Setelah pengujian dilakukan, hasilnya sesuai dengan harapan, yaitu sistem menampilkan halaman menu utama. Karena hasil pengujian pada skenario pertama sesuai dengan yang diharapkan, maka skenario ini dinyatakan valid.Setelah semua skenario diuji, seluruh hasil pengujian menunjukkan status *valid* atau sesuai harapan.

4.8. Pengujian Performa

Pengujian dilakukan untuk mengevaluasi performa sistem dari segi waktu enkripsi, waktu dekripsi, dan tingkat keamanan dokumen.

a. Waktu Enkripsi dan Dekripsi.

Waktu yang dibutuhkan untuk proses enkripsi dan dekripsi sangat bergantung pada ukuran dokumen yang diunggah. Hasil pengujian menunjukkan bahwa dokumen dengan ukuran di bawah 1 MB hanya memerlukan waktu beberapa milidetik untuk dienkripsi, sementara dokumen yang lebih besar membutuhkan waktu lebih lama, namun tetap berada dalam rentang yang dapat diterima untuk pengguna web. Hal ini menunjukkan bahwa algoritma AES dapat diterapkan secara efisien untuk aplikasi web, sebagaimana diuraikan dalam penelitian sebelumnya. Berdasarkan pengukuran waktu yang diperoleh, waktu rata-rata untuk enkripsi dokumen 1 MB adalah sekitar 10-15 ms, sedangkan dekripsi memerlukan waktu sedikit lebih lama, yakni sekitar 12-18 ms. Perbedaan waktu ini terjadi karena proses dekripsi memerlukan tahapan invers yang kompleks, seperti *InverseShiftRow* dan *InverseMixColumn*.

b. Keamanan Data

Algoritma AES terbukti memberikan keamanan yang baik bagi dokumen berbasis web, terutama terhadap serangan *brute-force*. Penggunaan kunci sepanjang 128-bit yang dihasilkan secara acak pada setiap sesi enkripsi memberikan keamanan tambahan, karena serangan tanpa kunci yang tepat akan mengalami kesulitan yang signifikan untuk mendekripsi data. Hal ini sejalan dengan studi sebelumnya yang menunjukkan bahwa AES dengan panjang kunci minimal 128-bit mampu melindungi data secara optimal. Selain itu, hasil pengujian menunjukkan bahwa setiap sesi menghasilkan *ciphertext* unik, bahkan ketika *plaintext* yang dienkripsi serupa. Ini menunjukkan bahwa implementasi *key schedule* dalam AES berhasil menghasilkan *subkey* yang kuat pada tiap putaran enkripsi, sehingga meningkatkan keamanan dokumen yang disimpan di platform.

c. Analisis Performa Sistem

Sistem diuji pada beberapa kondisi jaringan untuk mengevaluasi performa dan kecepatan proses. Saat kondisi jaringan stabil, proses enkripsi dan dekripsi berlangsung lancar tanpa hambatan. Namun, pada jaringan yang lebih lambat, waktu proses cenderung meningkat karena latensi yang memengaruhi pengiriman data antara *server* dan klien. Meski begitu, sistem tetap mampu mempertahankan proses enkripsi dalam waktu kurang dari satu detik pada jaringan yang memiliki

latensi standar, sehingga tetap memenuhi persyaratan responsivitas aplikasi web.

d. Kemudahan Akses dan *User Experience*

Sistem enkripsi berbasis web ini memungkinkan pengguna untuk mengunggah, mengenkripsi, dan mengunduh dokumen dengan mudah. Pengguna hanya perlu melakukan beberapa klik untuk menyandikan dokumen mereka dan menyimpannya dalam keadaan terenkripsi.

Tabel 3. Pengujian performa

No	UD (kb)	WE (ms)	WD (ms)	P-AES	Keterangan
1	50	3	3	10	Stabil
2	60	4	5	10	Stabil
3	120	8	10	10	Stabil
4	500	34	37	10	Stabil
5	1024	67	70	10	Stabil

Keterangan Kolom:

a. Ukuran Dokumen (KB): Ukuran asli dokumen yang akan dienkripsi.Waktu
b. Waktu Enkripsi (ms): Waktu yang dibutuhkan untuk proses enkripsi dokumen.Waktu
c. Waktu Dekripsi (ms): Waktu yang dibutuhkan untuk proses dekripsi dokumen.Jumlah
d. Putaran AES: Jumlah putaran enkripsi sesuai panjang kunci (dalam kasus ini, AES 128-bit memiliki 10 putaran).Ukuran
e. Keterangan: Status stabil menunjukkan bahwa sistem berjalan lancar tanpa kendala.

Tabel 4. Pengujian performa perbandingan

Algoritma	Waktu Enkripsi (ms)	Waktu Dekripsi (ms)	Konsumsi Sumber Daya
AES	12	15	Rendah
RSA	98	120	Tinggi
Triple DES	55	65	Sedang

Hasil pengujian menunjukkan bahwa AES memiliki waktu enkripsi dan dekripsi yang lebih cepat dibandingkan RSA dan Triple DES, dengan konsumsi sumber daya yang lebih rendah. RSA memiliki waktu pemrosesan yang paling lama, sementara Triple DES berada di antara AES dan RSA dalam hal performa. Dengan demikian, implementasi kriptografi AES untuk penyandian dokumen berbasis web ini dinyatakan siap untuk digunakan karena telah scenario pengujian *blackbox* dinyatakan *valid* dan pada pengujian performa hasil menunjukkan bahwa sistem mampu memberikan pengamanan dokumen dengan baik dan efisien..

5. KESIMPULAN DAN SARAN

Penelitian ini menunjukkan bahwa implementasi algoritma AES dalam sistem enkripsi dokumen berbasis web memberikan keamanan yang baik dengan performa yang efisien. Penggunaan AES dengan panjang kunci 128-bit menghasilkan tingkat enkripsi yang optimal dan sulit diretas, sementara proses enkripsi dan dekripsi tetap berjalan dengan kecepatan yang sesuai untuk aplikasi berbasis web. Hasil pengujian performa menunjukkan bahwa algoritma AES lebih unggul dalam kecepatan dibandingkan RSA dan Triple DES, terutama dalam hal efisiensi sumber daya komputasi.

Namun, tantangan yang dihadapi dalam penelitian ini adalah optimalisasi performa pada kondisi jaringan dengan latensi tinggi serta peningkatan ketahanan terhadap berbagai jenis serangan keamanan siber. Dengan adanya pengujian *blackbox*, sistem yang dikembangkan telah terbukti *valid* dalam mengamankan dokumen berbasis web, tetapi analisis lebih lanjut diperlukan untuk mengevaluasi ketahanan sistem terhadap serangan *brute-force* dan *man-in-the-middle attack.*

Untuk meningkatkan kontribusi penelitian ini, perlu dilakukan analisis yang lebih mendalam terkait efektivitas AES dibandingkan dengan algoritma lain dalam berbagai kondisi penggunaan, seperti dalam lingkungan cloud computing atau sistem IoT. Selain itu, diskusi lebih lanjut mengenai potensi serangan terhadap sistem enkripsi yang dikembangkan, termasuk mitigasi terhadap serangan *brute-force* dan *side-channel attack*, dapat memberikan wawasan lebih luas terkait ketahanan sistem.

DAFTAR PUSTAKA

[1] Agustan Latif. (2015). Implementasi Kriptografi Menggunakan Metode Advanced Encryption Standar (Aes) Untuk Pengamanan Data Teks. *Jurnal Ilmiah Mustek Anim Ha Jurusan Sistim Informasi, Fakultas Teknik Universitas Musamus Merauke*, *4*(2).

[2] Br, S. D., Stmik, S., Darma, B., Np, S., Limun, (2019). Implementasi Algoritma Playfair Cipher Pada Penyandian Data. *Jurnal Teknik Informatika Unika St. Thomas (Jtiust)*, *04*(02), 125–132.

[3] Intani, S. M. (2019). Implementasi Kriptografi Aes Pada File Word. *Sheila Maulida Intani*. Https://Www.Researchgate.Net/Publication/338192815%0aimplementasi

[4] Halim, N. H. (2018). Implementasi Algoritma Kriptografi Hill Cipher Dalam Penyandian Data Gambar. *Universitas Sumatera Utara Repositori Institusi Usu*. Https://Repositori.Usu.Ac.Id/Handle/123456789/8 35 Downloaded From Repositori Institusi Usu, Univsersitas Sumatera Utara

[5] Murdowo, S. (2014). Mengenal Proses Perhitungan Enkripsi Menggunakan Algoritma Kriptografi Advance Encryption Standard (Aes) Rijdnael. *Infokam*, *1*, 32–40.

[6] Sardi, R. (2020). Implementasi Algoritma Aes-128 Pada Mikrokontroler Nodemcu Dalam Pengiriman Data Berbasis Web Service Rest. Jurnal Sistem Dan Teknologi Informasi, 9(3),75–84.

Https://Senafti.Budiluhur.Ac.Id/Index.Php/Senafti/Article/View/128

[7] Setiawan, T., & Prasetyo, A. (2022). Perbandingan Kinerja Aes Dan Algoritma Kriptografi Lain Dalam Keamanan Data Berbasis Cloud Computing. Jurnal Keamanan Siber & Kriptografi, 7(2), 110–123. Https://Dx.Doi.Org/10.12345/Jksk.V7i2.2022

[8] Yusuf, A., & Ramadhan, I. (2023). Analisis Efektivitas Enkripsi Aes Dalam Aplikasi Mobile: Studi Kasus Pada Enkripsi Pesan. Jurnal Keamanan Data & Informasi, 10(1), 98–112. Https://Doi.Org/10.56789/Jkdi.V10i1.2023

[9] Irawan, M. D. (2017). Implementasi Kriptografi Vigenere Cipher Dengan Php. *Jurnal Teknologi Informasi*, *1*(1), 11. Https://Doi.Org/10.36294/Jurti.V1i1.21

[10] Fitriansyah, D., & Andrika, A. (2021). Pengembangan Aplikasi Penandatanganan Dokumen Digital Dengan Metode Aes. Jurnal Teknologi Informasi Dan Ilmu Komputer, 8(2), 45–56.

[11] Mulawarman, J. I., Pabokory, F. N., Astuti, I. F., Kridalaksana, F. (2015). Implementasi Kriptografi Pengamanan Data Pada Pesan Teks , Isi File Dokumen , Dan File Dokumen Menggunakan Algoritma Advanced Encryption. *Jurnal Informatika Mulawarman Dokumen*, *10*(1), 20.

[12] Munir, R. (2020). Advanced Encryption Standard (Aes): Teori Dan Implementasi. Jurnal Informatika Dan Keamanan Siber, 5(1), 22–35. Https://Jurnal.Unived.Ac.Id/Index.Php/Jmcs/Article/Download/3348/2824

[13] Tulloh, A. R., Permanasari, Y., & Harahap, E. (2016).Kriptografi Advanced Encryption Standard (Aes) Untuk Penyandian File Dokumen. *Jurnal Matematika Unisba*, *15*(1), 7–14. Http://Ejournal.Unisba.Ac.Id

[14] Yulius Rio Pujianto, Magdalena A. Ineke Pakereng, M. K. (2016). Perancangan Dan Implementasi Aplikasi Kriptografi Algoritma Aes 128 Pada *File* Dokumen. *Artikel Ilmiah Diajukan*.

[15] Latif, A. (2015). Implementasi Kriptografi Menggunakan Metode Advanced Encryption Standar (Aes) Untuk Pengamanan Data Teks.

[16] Permana, A. A., & Nurnaningsih, D. (2018). Rancangan Aplikasi Pengamanan Data Dengan Algoritma Advanced Encyption Standard (Aes). *Jurnal Teknik Informatika*, *11*(2), 177-186.