



Sistem Keamanan Penyimpanan Dokumen Digital Menggunakan Algoritma AES-256 Berbasis Web

Muhammad Asril Azmi^{1*}, Rizaldy Khair²

^{1*,2}Program Studi Sistem Informasi, Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara, Medan, Indonesia
Email: ^{1*}asrilazmi25@gmail.com, ²rizaldykhair@umsu.ac.id

Abstrak

Keamanan dokumen digital menjadi aspek penting untuk melindungi informasi dari akses yang tidak sah. Penelitian ini bertujuan membangun sistem keamanan penyimpanan dokumen digital berbasis web menggunakan algoritma Advanced Encryption Standard (AES-256-CBC). Sistem dikembangkan menggunakan framework Laravel, bahasa pemrograman PHP, basis data SQLite, serta pustaka OpenSSL untuk proses enkripsi dan dekripsi dokumen. Metode pengujian yang digunakan adalah Black Box Testing untuk memastikan setiap fungsi sistem berjalan sesuai dengan kebutuhan fungsional. Fitur yang diuji meliputi login, unggah dokumen, enkripsi, dekripsi, pengunduhan, dan penghapusan dokumen. Hasil penelitian menunjukkan bahwa sistem berhasil mengimplementasikan algoritma AES-256-CBC dalam mengamankan dokumen digital. Dokumen yang diunggah dienkripsi secara otomatis sebelum disimpan, sedangkan proses dekripsi hanya dapat dilakukan menggunakan Encryption Key yang dimiliki pengguna. Selain itu, sistem menerapkan Initialization Vector (IV) yang dibangkitkan secara acak pada setiap proses enkripsi untuk meningkatkan keamanan data. Berdasarkan hasil pengujian Black Box, seluruh fitur utama sistem berjalan sesuai dengan kebutuhan fungsional yang telah dirancang. Dengan demikian, sistem yang dibangun mampu meningkatkan keamanan penyimpanan dokumen digital berbasis web.

Kata Kunci: AES-256-CBC, dokumen digital, enkripsi, kriptografi, Laravel.

PENDAHULUAN

Perkembangan teknologi informasi yang pesat telah mendorong transformasi pengelolaan dokumen dari bentuk konvensional menjadi dokumen digital. Penggunaan dokumen digital memberikan berbagai kemudahan seperti efisiensi penyimpanan, kemudahan akses, serta fleksibilitas dalam distribusi data. Namun, di balik keunggulan tersebut, muncul permasalahan baru yang berkaitan dengan aspek keamanan, khususnya dalam menjaga kerahasiaan informasi yang bersifat sensitif (Putra dkk., 2024).

Permasalahan utama yang menjadi perhatian dalam penelitian ini adalah masih banyaknya sistem penyimpanan dokumen digital yang belum menerapkan mekanisme pengamanan data secara optimal, khususnya dalam bentuk enkripsi. Dokumen sering kali disimpan dalam bentuk asli (plaintext), sehingga rentan terhadap ancaman seperti akses tidak sah, kebocoran data, dan manipulasi informasi (Widodo & Purnomo, 2020). Kondisi ini menunjukkan adanya kesenjangan antara kebutuhan akan keamanan data dengan implementasi sistem yang ada.

Program Studi Sistem Informasi Universitas Muhammadiyah Sumatera Utara (UMSU) merupakan salah satu program studi yang mengelola berbagai dokumen digital, seperti dokumen administrasi akademik, surat-menysurat, data mahasiswa, arsip kegiatan, dokumen tugas akhir, dan dokumen penting lainnya. Dokumen-dokumen tersebut memiliki nilai informasi yang penting sehingga memerlukan sistem penyimpanan yang mampu menjaga kerahasiaan, keutuhan, dan ketersediaan data. Oleh karena itu, penerapan mekanisme keamanan pada penyimpanan dokumen menjadi aspek yang penting untuk mendukung pengelolaan dokumen secara aman dan terstruktur (Kaaffah dkk., 2025; NIST, 2023).

Beberapa penelitian terdahulu telah membahas implementasi AES-256 dalam pengamanan dokumen digital. Naqiyah dkk. (2025) mengembangkan sistem keamanan dokumen digital berbasis web menggunakan AES-256 dan berhasil menunjukkan bahwa sistem mampu melindungi dokumen secara efektif. Ridho dan Ali Romli (2024) juga mengimplementasikan AES-256 untuk pengamanan dokumen dan membuktikan bahwa dokumen menjadi lebih aman dari ancaman akses tidak sah. Kaaffah dkk. (2025) mengkombinasikan AES-128 dengan SHA-256 berbasis salt dan menunjukkan peningkatan keamanan yang signifikan. Namun, penelitian-penelitian tersebut masih memiliki keterbatasan, antara lain belum

adanya integrasi sistem berbasis web secara menyeluruh, kurangnya pembahasan mengenai pengelolaan kunci (key management), serta belum adanya evaluasi fungsional sistem secara sistematis menggunakan metode Black Box Testing.

Untuk mengatasi permasalahan dan keterbatasan tersebut, penelitian ini mengimplementasikan algoritma AES-256-CBC yang telah terbukti memiliki tingkat keamanan tinggi dan banyak digunakan dalam berbagai sistem keamanan informasi modern (NIST, 2023). Kebaruan penelitian ini terletak pada integrasi menyeluruh antara algoritma AES-256-CBC dalam sistem berbasis web menggunakan framework Laravel, disertai mekanisme pengelolaan kunci yang tidak menyimpan encryption key di sisi server, serta penerapan Initialization Vector (IV) acak pada setiap enkripsi untuk meningkatkan keamanan data.

Penelitian ini bertujuan untuk merancang dan membangun sistem keamanan penyimpanan dokumen digital berbasis web menggunakan algoritma AES-256 pada Program Studi Sistem Informasi UMSU, serta menguji fungsionalitas sistem yang dibangun menggunakan metode Black Box Testing.

METODE

Penelitian ini menggunakan pendekatan penelitian pengembangan sistem (*system development*) yang meliputi beberapa tahapan, yaitu analisis kebutuhan, perancangan sistem, implementasi, dan pengujian. Sistem keamanan penyimpanan dokumen digital dikembangkan dalam bentuk aplikasi berbasis web menggunakan *framework* Laravel 12 dengan bahasa pemrograman PHP, basis data SQLite, serta pustaka OpenSSL untuk implementasi algoritma AES-256-CBC.

Analisis Kebutuhan Sistem

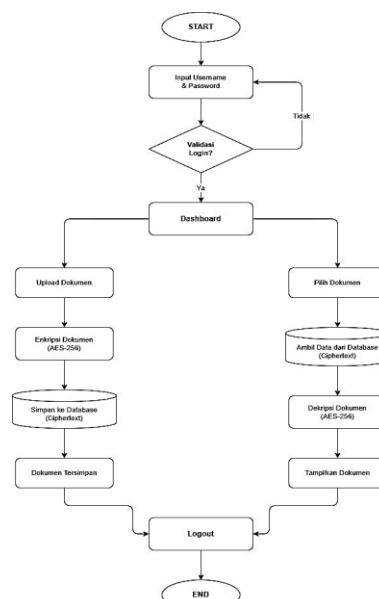
Tahap analisis kebutuhan dilakukan untuk mengidentifikasi kebutuhan fungsional dan non-fungsional sistem. Kebutuhan fungsional sistem meliputi: (1) proses autentikasi pengguna (login); (2) unggah dokumen; (3) enkripsi dokumen menggunakan algoritma AES-256-CBC secara otomatis; (4) penyimpanan dokumen dalam bentuk ciphertext; (5) penampilan daftar dokumen; (6) dekripsi dokumen menggunakan Encryption Key; serta (7) pengunduhan dan penghapusan dokumen.

Sementara itu, kebutuhan non-fungsional sistem mencakup aspek keamanan, aksesibilitas, kinerja sistem, serta kemudahan penggunaan (*user friendly*).

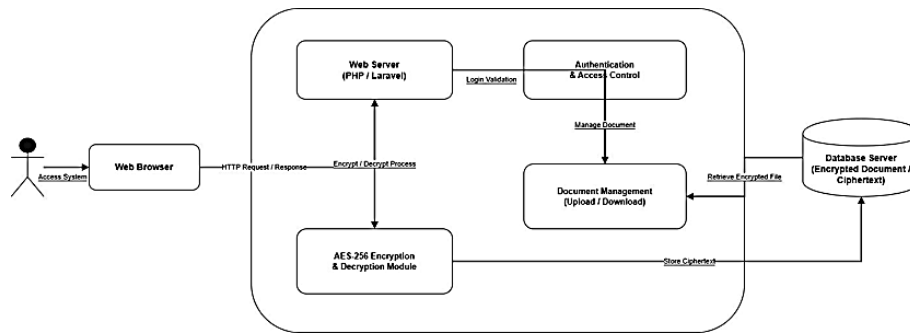
Perancangan Sistem

Perancangan sistem dilakukan menggunakan pendekatan *Unified Modeling Language* (UML), yang mencakup *Use Case Diagram* dan *Activity Diagram*. *Use Case Diagram* digunakan untuk menggambarkan interaksi antara pengguna sebagai aktor dengan fitur-fitur yang tersedia dalam sistem. Sementara itu, *Activity Diagram* digunakan untuk menggambarkan alur aktivitas sistem secara berurutan, mulai dari proses *login* hingga pengelolaan dokumen. Selain menggunakan UML, *flowchart* sistem digunakan untuk menggambarkan alur kerja proses enkripsi dan dekripsi dokumen secara terstruktur.

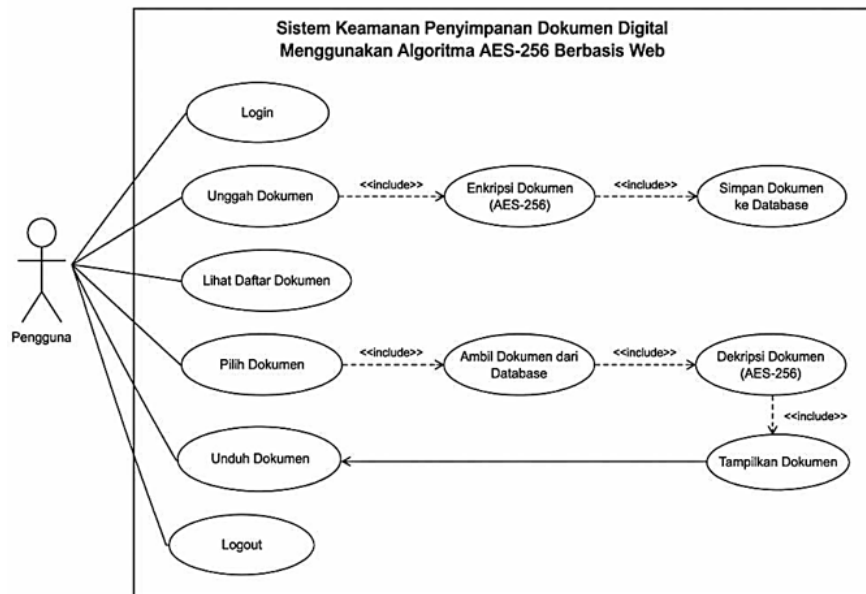
Arsitektur sistem menggunakan model *client-server*, di mana pengguna mengakses sistem melalui *web browser*. Permintaan dari pengguna kemudian dikirimkan ke *web server* untuk diproses. *Web server* menangani proses autentikasi pengguna, pengelolaan dokumen, serta proses enkripsi dan dekripsi menggunakan algoritma AES-256. Hasil proses selanjutnya disimpan pada basis data dalam bentuk terenkripsi.



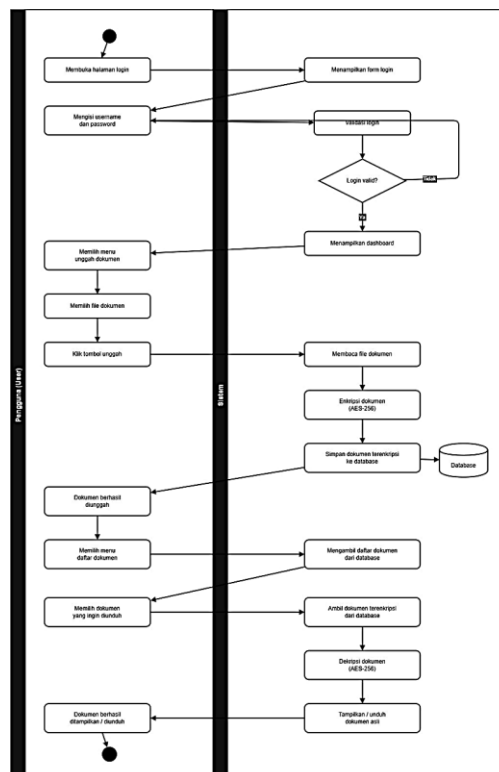
Gambar 1. Flowchart Sistem



Gambar 2. Arsitektur Sistem



Gambar 3. Use Case Diagram



Gambar 4. Activity Diagram

Implementasi Algoritma AES-256-CBC

Algoritma AES-256-CBC diimplementasikan menggunakan pustaka OpenSSL yang terintegrasi dengan *framework* Laravel. Mode operasi *Cipher Block Chaining* (CBC) dipilih karena setiap blok data dalam proses enkripsi dipengaruhi oleh blok sebelumnya, sehingga menghasilkan *ciphertext* yang tidak deterministik dan lebih aman dibandingkan dengan mode ECB (*Electronic Codebook*).

Pada setiap proses enkripsi, sistem membangkitkan *Initialization Vector* (IV) secara acak menggunakan fungsi `random_bytes()`. Penggunaan IV secara acak menyebabkan dokumen dengan isi yang identik dapat menghasilkan *ciphertext* yang berbeda pada setiap proses enkripsi. Proses enkripsi dinyatakan dengan persamaan berikut:

$$C = E(P, K) \quad (1)$$

Keterangan:

C = Ciphertext

P = Plaintext

K = Key (kunci enkripsi)

Sedangkan proses dekripsi dinyatakan dengan persamaan:

$$P = D(C, K) \quad (2)$$

Keterangan:

C = Ciphertext

P = Plaintext

K = Key (kunci enkripsi)

Pada algoritma AES-256, proses enkripsi terdiri atas 14 ronde. Setiap ronde mencakup tahapan *AddRoundKey*, *SubBytes*, *ShiftRows*, *MixColumns*, dan *AddRoundKey*, dengan pengecualian pada ronde terakhir, yaitu tahapan *MixColumns* tidak dilakukan (NIST, 2023).

Kunci enkripsi (*Encryption Key*) yang dihasilkan sistem hanya ditampilkan satu kali kepada pengguna setelah proses enkripsi selesai dan tidak disimpan oleh sistem. Dengan mekanisme tersebut, proses dekripsi hanya dapat dilakukan oleh pengguna yang memiliki *Encryption Key* yang benar.

Metode Pengujian

Pengujian sistem menggunakan metode *Black Box Testing*, yaitu metode pengujian perangkat lunak yang berfokus pada pengujian fungsi-fungsi sistem berdasarkan masukan (*input*) dan keluaran (*output*) yang dihasilkan, tanpa memperhatikan struktur kode program secara internal.

Pengujian dilakukan terhadap lima fitur utama sistem, yaitu: (1) proses login; (2) unggah dokumen; (3) enkripsi dokumen; (4) dekripsi dokumen; dan (5) penghapusan dokumen.

Setiap fitur diuji menggunakan berbagai skenario masukan, baik masukan yang valid maupun tidak valid. Pengujian tersebut dilakukan untuk memastikan bahwa setiap fungsi sistem dapat menghasilkan keluaran yang sesuai dengan kebutuhan fungsional yang telah dirancang.

HASIL DAN PEMBAHASAN

Implementasi Sistem

Penelitian ini menghasilkan sebuah sistem keamanan penyimpanan dokumen digital berbasis web yang diberi nama **DocuSafe**. Sistem dikembangkan menggunakan *framework* Laravel 12 dengan bahasa pemrograman PHP, basis data SQLite, serta pustaka OpenSSL untuk mendukung proses kriptografi. Implementasi sistem dirancang untuk memberikan mekanisme perlindungan terhadap dokumen digital melalui proses enkripsi sebelum dokumen disimpan pada *storage* aplikasi.

Laravel menyediakan mekanisme enkripsi berbasis OpenSSL dan mendukung penggunaan algoritma AES-256-CBC. Dalam implementasinya, Laravel menggunakan *encryption key* yang dikonfigurasi melalui `APP_KEY`, sedangkan proses pembangkitan kunci aplikasi memanfaatkan generator bilangan acak yang aman secara kriptografis.

Lingkungan perangkat lunak yang digunakan dalam proses implementasi sistem ditunjukkan pada Tabel 1 berikut ini.

Tabel 1. Lingkungan Implementasi Sistem

Komponen	Spesifikasi
Sistem Operasi	Windows 11
<i>Framework</i>	Laravel 12
Bahasa Pemrograman	PHP
Basis Data	SQLite
Pustaka Kriptografi	OpenSSL
Algoritma Kriptografi	AES-256-CBC
<i>Text Editor</i>	Visual Studio Code

Sistem DocuSafe terdiri atas beberapa komponen antarmuka utama, yaitu halaman autentikasi, *dashboard*, halaman unggah dokumen, *modal* untuk menampilkan *encryption key*, dan *modal* untuk proses dekripsi dokumen. Setiap komponen dirancang untuk mendukung tahapan pengamanan dokumen mulai dari autentikasi pengguna, pengunggahan dokumen, enkripsi, penyimpanan, hingga pemulihan dokumen melalui proses dekripsi.

Struktur Basis Data

Basis data DocuSafe terdiri atas dua tabel utama, yaitu *users* dan *documents*. Tabel *users* digunakan untuk menyimpan informasi akun pengguna yang memiliki hak akses terhadap sistem, sedangkan tabel *documents* digunakan untuk menyimpan metadata dokumen yang telah diproses melalui mekanisme enkripsi.

Tabel *users* menyimpan informasi identitas akun, termasuk alamat *email* dan kata sandi pengguna. Kata sandi tidak disimpan dalam bentuk teks asli, tetapi dalam bentuk *hash* sesuai dengan mekanisme autentikasi aplikasi.

Sementara itu, tabel *documents* menyimpan informasi yang berkaitan dengan dokumen pengguna, antara lain *user_id* sebagai relasi terhadap pengguna, *encrypted_path* sebagai lokasi penyimpanan berkas terenkripsi, serta *iv* (*Initialization Vector*) yang diperlukan dalam proses dekripsi.

column_name	data_type	is_nullable	column_default	foreign_key
id	INTEGER	NO	NULL	EMPTY →
name	varchar	NO	NULL	EMPTY →
email	varchar	NO	NULL	EMPTY →
email_verified_at	datetime	YES	NULL	EMPTY →
password	varchar	NO	NULL	EMPTY →
remember_token	varchar	YES	NULL	EMPTY →
created_at	datetime	YES	NULL	EMPTY →
updated_at	datetime	YES	NULL	EMPTY →

Gambar 5. Struktur Tabel *users*

column_name	data_type	is_nullable	column_default	foreign_key
id	INTEGER	NO	NULL	EMPTY →
user_id	INTEGER	NO	NULL	users(id) →
original_name	varchar	NO	NULL	EMPTY →
encrypted_path	varchar	NO	NULL	EMPTY →
file_type	varchar	NO	NULL	EMPTY →
file_size	INTEGER	NO	NULL	EMPTY →
iv	varchar	NO	NULL	EMPTY →
created_at	datetime	YES	NULL	EMPTY →
updated_at	datetime	YES	NULL	EMPTY →

Gambar 6. Struktur Tabel *documents*

Arsitektur penyimpanan tersebut memisahkan antara data metadata dengan isi dokumen. Isi dokumen tidak disimpan secara langsung sebagai *binary data* di dalam basis data, tetapi ditempatkan pada *storage* dalam bentuk ciphertext. Basis data hanya menyimpan informasi yang diperlukan untuk mengidentifikasi dan memproses dokumen, termasuk lokasi berkas terenkripsi dan nilai IV.

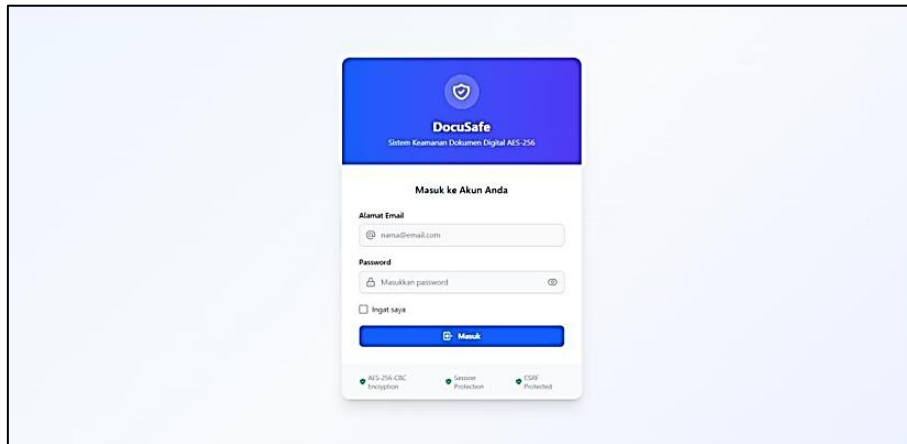
Penyimpanan IV bersama metadata tidak dimaksudkan sebagai penyimpanan kunci rahasia. IV merupakan parameter yang diperlukan dalam proses enkripsi/dekripsi AES-CBC dan dapat disimpan bersama ciphertext. Dengan demikian, kerahasiaan dokumen tetap bergantung pada kerahasiaan kunci enkripsi.

Antarmuka Sistem

Halaman Login

Halaman login merupakan komponen autentikasi yang berfungsi sebagai pintu masuk pengguna ke dalam sistem DocuSafe. Pengguna harus memasukkan alamat *email* dan kata sandi yang telah terdaftar sebelum memperoleh akses terhadap dokumen yang tersimpan.

Pada tahap autentikasi, sistem menerapkan mekanisme perlindungan sesi dan *Cross-Site Request Forgery* (CSRF) untuk mengurangi risiko penyalahgunaan sesi maupun pengiriman permintaan yang tidak sah. Setelah kredensial berhasil diverifikasi, pengguna diarahkan ke halaman *dashboard*.



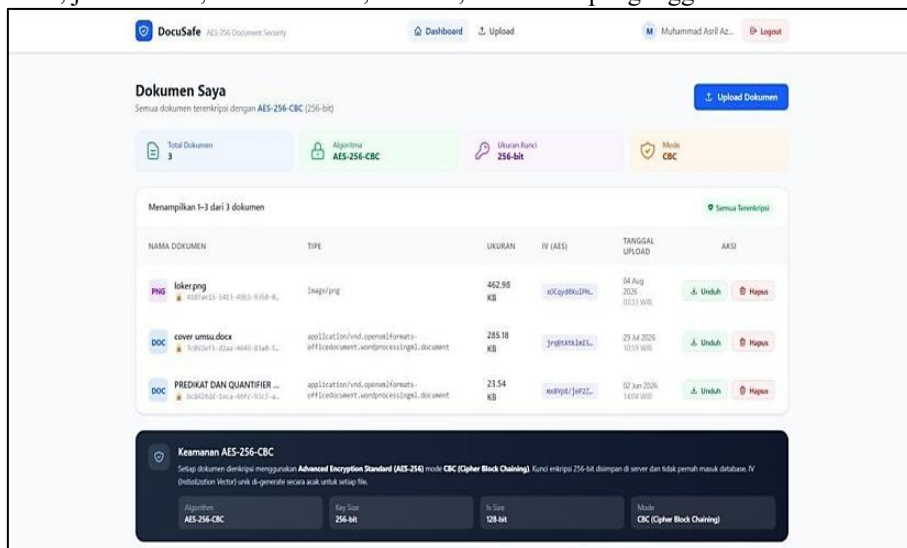
Gambar 7. Tampilan Halaman Login

Hasil implementasi menunjukkan bahwa halaman login dapat membedakan kondisi kredensial valid dan tidak valid serta memberikan validasi ketika pengguna tidak mengisi data yang dipersyaratkan. Mekanisme tersebut menjadi lapisan awal dalam membatasi akses terhadap dokumen yang tersimpan pada sistem.

Halaman Dashboard

Halaman *dashboard* merupakan halaman utama yang ditampilkan setelah pengguna berhasil melakukan autentikasi. Halaman ini memberikan ringkasan kondisi penyimpanan dokumen sekaligus menyediakan akses terhadap dokumen yang dimiliki oleh pengguna.

Informasi yang ditampilkan mencakup jumlah dokumen yang tersimpan serta informasi mengenai konfigurasi algoritma kriptografi yang digunakan. Daftar dokumen juga menampilkan informasi berupa nama dokumen, jenis berkas, ukuran berkas, nilai IV, dan waktu pengunggahan.



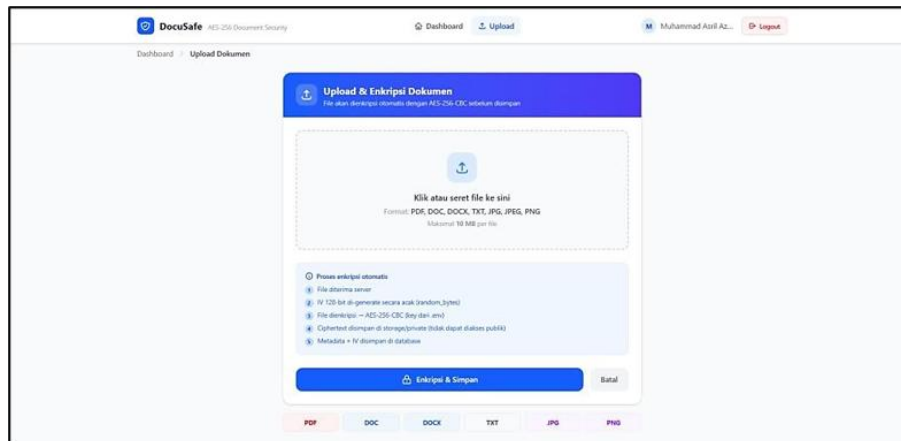
Gambar 8. Tampilan Halaman Dashboard

Pemisahan antara informasi dokumen dan isi dokumen terenkripsi memungkinkan pengguna tetap memperoleh informasi metadata tanpa harus menampilkan isi dokumen secara langsung. Dengan demikian, dokumen yang tersimpan pada *storage* tetap berada dalam bentuk ciphertext sampai proses dekripsi dilakukan.

Halaman Upload Dokumen

Halaman unggah dokumen digunakan untuk memasukkan dokumen digital ke dalam sistem sebelum dilakukan proses pengamanan. Sistem mendukung beberapa format berkas, yaitu PDF, DOC, DOCX, TXT, JPG, JPEG, dan PNG, dengan ukuran maksimum berkas sebesar 10 MB.

Setelah pengguna memilih dokumen dan menjalankan proses unggah, sistem melakukan serangkaian tahapan pengamanan. Tahapan tersebut meliputi: (1) membaca isi dokumen yang diunggah; (2) membangkitkan Initialization Vector (IV); (3) melakukan proses enkripsi menggunakan AES-256-CBC melalui OpenSSL; (4) menyimpan hasil enkripsi dalam bentuk ciphertext pada storage aplikasi; dan (5) menyimpan metadata dokumen serta IV pada basis data. AES merupakan algoritma symmetric block cipher yang mendukung ukuran kunci 128, 192, dan 256 bit. Pada penelitian ini digunakan konfigurasi AES-256-CBC untuk proses perlindungan isi dokumen.

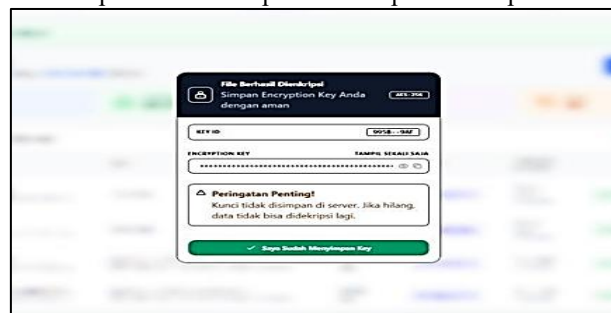


Gambar 9. Tampilan Halaman Upload Dokumen

Tahapan tersebut memastikan bahwa berkas yang telah berhasil disimpan pada *storage* tidak berada dalam bentuk dokumen asli, melainkan telah dikonversi menjadi ciphertext. Dengan demikian, akses langsung terhadap berkas pada *storage* tidak secara otomatis memberikan akses terhadap isi dokumen.

Modal Box Encryption Key

Setelah proses enkripsi berhasil, sistem menampilkan *modal* yang berisi *encryption key* yang digunakan untuk mengakses kembali dokumen melalui proses dekripsi. Kunci tersebut ditampilkan kepada pengguna setelah proses enkripsi dan tidak disimpan sebagai data dokumen pada basis data. Oleh karena itu, pengguna memiliki tanggung jawab untuk menyimpan kunci secara aman. Kehilangan kunci dapat menyebabkan pengguna tidak dapat melakukan proses dekripsi terhadap dokumen yang bersangkutan.

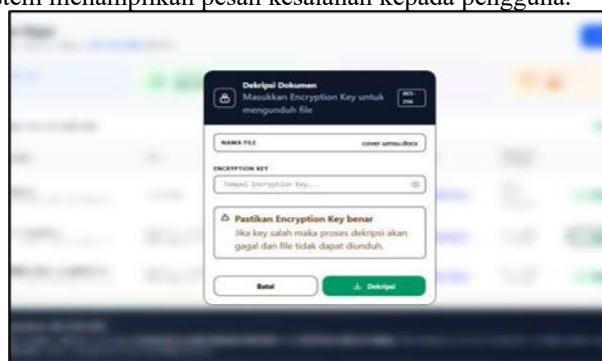


Gambar 10. Tampilan Modal Box Encryption Key

Mekanisme ini memberikan pemisahan antara penyimpanan ciphertext dengan informasi kunci. Dengan tidak menyimpan kunci sebagai metadata dokumen, sistem mengurangi kemungkinan kunci diperoleh secara langsung melalui akses terhadap basis data.

Modal Box Dekripsi Dokumen

Modal dekripsi ditampilkan ketika pengguna memilih dokumen untuk diakses kembali. Antarmuka tersebut menampilkan identitas dokumen dan menyediakan kolom untuk memasukkan *encryption key*. Setelah kunci dimasukkan, sistem melakukan proses validasi dan dekripsi. Apabila kunci yang diberikan sesuai dengan mekanisme enkripsi yang digunakan, ciphertext diproses menggunakan AES-256-CBC dengan IV yang tersimpan pada metadata dokumen sehingga dokumen dapat dikembalikan ke bentuk semula. Sebaliknya, apabila kunci tidak sesuai, proses dekripsi tidak menghasilkan dokumen yang dapat dipulihkan dan sistem menampilkan pesan kesalahan kepada pengguna.



Gambar 11. Tampilan Modal Box Input Encryption Key untuk Dekripsi

Mekanisme tersebut menunjukkan bahwa proses pengambilan kembali dokumen tidak hanya bergantung pada hak akses akun, tetapi juga memerlukan informasi kriptografis yang diperlukan untuk proses dekripsi.

Implementasi Algoritma AES-256-CBC

Implementasi AES-256-CBC pada DocuSafe ditujukan untuk memberikan perlindungan terhadap kerahasiaan dokumen digital selama dokumen berada dalam penyimpanan. AES merupakan standar kriptografi simetris yang digunakan untuk mengenkripsi dan mendekripsi data menggunakan kunci kriptografis. NIST menetapkan AES sebagai algoritma kriptografi yang dapat digunakan untuk melindungi data elektronik, dengan dukungan ukuran kunci 128, 192, dan 256 bit.

Pada proses enkripsi, sistem terlebih dahulu membaca isi dokumen yang diunggah. Selanjutnya, sistem menyiapkan IV sebagai parameter yang digunakan bersama kunci dalam mode CBC. Data dokumen kemudian diproses menggunakan AES-256-CBC sehingga menghasilkan ciphertext. Ciphertext disimpan pada *storage*, sedangkan informasi metadata dan IV disimpan pada basis data.

Secara konseptual, proses enkripsi dapat digambarkan sebagai:

Plaintext + Encryption Key + IV → AES-256-CBC → Ciphertext

Sebaliknya, proses pemulihan dokumen dilakukan melalui tahapan:

Ciphertext + Encryption Key + IV → AES-256-CBC → Plaintext

Dengan mekanisme tersebut, dokumen yang tersimpan dalam *storage* tidak berada dalam bentuk plaintext. Dokumen baru dapat dikembalikan ke bentuk semula apabila parameter yang diperlukan dalam proses dekripsi tersedia dan sesuai.

Pada aspek pengelolaan kunci, implementasi sistem membedakan antara kunci aplikasi dan kunci yang digunakan pada mekanisme akses dokumen. Laravel sendiri menggunakan konfigurasi APP_KEY sebagai kunci enkripsi aplikasi dan menyediakan layanan enkripsi berbasis OpenSSL dengan AES-256-CBC. Laravel juga memberikan autentikasi terhadap nilai yang dienkripsi melalui *message authentication code* (MAC), sehingga perubahan terhadap ciphertext atau nilai terenkripsi dapat dideteksi dalam mekanisme enkripsi Laravel.

Dalam penelitian ini, IV diperlakukan sebagai parameter non-rahasia yang diperlukan untuk proses dekripsi dan disimpan bersama metadata dokumen. Penggunaan IV yang berbeda pada setiap proses enkripsi penting karena membantu mencegah terbentuknya ciphertext identik ketika plaintext yang sama dienkripsi dengan kunci yang sama. Dengan demikian, penggunaan IV tidak hanya menjadi bagian dari kebutuhan teknis mode CBC, tetapi juga berkontribusi terhadap pengurangan pola yang dapat diamati dari hasil ciphertext.

Dari perspektif keamanan informasi, mekanisme tersebut terutama berkaitan dengan aspek *confidentiality*, yaitu memastikan bahwa informasi tidak dapat diakses atau diungkapkan kepada pihak yang tidak berwenang. NIST menempatkan *confidentiality*, *integrity*, dan *availability* sebagai tiga tujuan utama dalam *information security*.

Hasil Pengujian Black Box

Pengujian sistem dilakukan menggunakan metode *Black Box Testing* untuk memverifikasi kesesuaian fungsi sistem terhadap kebutuhan yang telah ditetapkan. Pengujian difokuskan pada lima kelompok fungsi utama, yaitu autentikasi pengguna, unggah dokumen, enkripsi dokumen, dekripsi dokumen, dan penghapusan dokumen.

Tabel 2. Hasil Pengujian Fungsi Login

No.	Skenario Pengujian	Input	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Login dengan kredensial valid	Email dan kata sandi valid	Sistem mengarahkan pengguna ke halaman dashboard	Halaman dashboard berhasil ditampilkan	Berhasil
2	Login dengan kata sandi salah	Email valid dan kata sandi salah	Sistem menolak dan menampilkan pesan kesalahan	Sistem menampilkan pesan kesalahan	Berhasil
3	Login dengan email tidak terdaftar	Email tidak terdaftar	Sistem menolak autentikasi	Sistem menolak autentikasi	Berhasil
4	Login tanpa mengisi data	Kolom email dan kata sandi kosong	Sistem menampilkan validasi	Validasi ditampilkan	Berhasil

Hasil pengujian menunjukkan bahwa fungsi autentikasi dapat membedakan kondisi kredensial yang valid dan tidak valid. Validasi terhadap input kosong juga berjalan sesuai dengan skenario pengujian.

Tabel 3. Hasil Pengujian Fungsi Upload Dokumen

No.	Skenario Pengujian	Input	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Mengunggah dokumen dengan format yang didukung	PDF, DOC, DOCX, TXT, JPG, JPEG, atau PNG	Dokumen diterima dan diproses sistem	Dokumen berhasil diunggah dan diproses	Berhasil
2	Mengunggah dokumen melebihi batas ukuran	Berkas > 10 MB	Sistem menolak berkas	Sistem menolak dan menampilkan pesan kesalahan	Berhasil
3	Mengunggah format yang tidak didukung	Berkas EXE	Sistem menolak berkas	Sistem menolak berkas	Berhasil
4	Mengunggah tanpa memilih berkas	Tidak ada berkas	Sistem menampilkan validasi	Validasi ditampilkan	Berhasil

Hasil pengujian menunjukkan bahwa mekanisme validasi unggah dapat membatasi format dan ukuran berkas sesuai dengan spesifikasi sistem.

Tabel 4. Hasil Pengujian Fungsi Enkripsi Dokumen

No.	Skenario Pengujian	Input	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Enkripsi setelah unggah berhasil	Berkas valid	Sistem melakukan enkripsi AES-256-CBC	Dokumen berhasil dienkripsi	Berhasil
2	Menampilkan Encryption Key	Proses enkripsi berhasil	Kunci ditampilkan kepada pengguna	Kunci berhasil ditampilkan	Berhasil
3	Menyimpan ciphertext	Proses enkripsi berhasil	Ciphertext tersimpan pada storage	Ciphertext berhasil disimpan	Berhasil
4	Menyimpan metadata	Proses enkripsi berhasil	Metadata dan IV tersimpan pada basis data	Metadata berhasil disimpan	Berhasil

Hasil pengujian menunjukkan bahwa proses unggah terintegrasi dengan tahapan enkripsi dan penyimpanan. Setelah enkripsi selesai, ciphertext dapat disimpan pada *storage* dan informasi pendukung proses dekripsi dapat disimpan pada basis data.

Tabel 5. Hasil Pengujian Fungsi Dekripsi Dokumen

No.	Skenario Pengujian	Input	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Dekripsi dengan kunci yang benar	Encryption Key valid	Dokumen berhasil didekripsi	Dokumen berhasil didekripsi dan diunduh	Berhasil
2	Dekripsi dengan kunci yang salah	Encryption Key tidak valid	Sistem menolak proses dekripsi	Sistem menampilkan pesan kesalahan	Berhasil
3	Dekripsi tanpa kunci	Kolom kunci kosong	Sistem meminta pengguna mengisi kunci	Validasi ditampilkan	Berhasil
4	Verifikasi hasil dekripsi	Encryption Key valid	Dokumen hasil dekripsi sesuai dengan dokumen asli	Dokumen berhasil dipulihkan sesuai dokumen asli	Berhasil

Pengujian dekripsi menunjukkan bahwa sistem dapat mengembalikan dokumen terenkripsi ke bentuk semula ketika parameter dekripsi yang diberikan sesuai. Sebaliknya, input kunci yang tidak sesuai menyebabkan proses akses terhadap dokumen gagal.

Tabel 6. Hasil Pengujian Fungsi Penghapusan Dokumen

No.	Skenario Pengujian	Input	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Menghapus dokumen	Pengguna memilih tombol Hapus	Dokumen dihapus dari sistem, basis data, dan storage	Dokumen berhasil dihapus	Berhasil
2	Membatalkan penghapusan	Pengguna memilih Batal	Dokumen tetap tersedia	Dokumen tetap tersedia	Berhasil

Berdasarkan hasil pengujian pada Tabel 2 hingga Tabel 6, seluruh skenario pengujian memperoleh status **Berhasil**. Temuan tersebut menunjukkan bahwa fungsi utama yang diuji telah berjalan sesuai dengan kebutuhan fungsional yang ditetapkan, baik pada kondisi input valid maupun pada beberapa kondisi input tidak valid.

Namun demikian, hasil *Black Box Testing* perlu dipahami sebagai bukti kesesuaian fungsi berdasarkan skenario yang diuji, bukan sebagai bukti bahwa sistem telah bebas dari seluruh kerentanan keamanan. Pengujian keamanan yang lebih komprehensif masih dapat dilakukan melalui *penetration testing*, *vulnerability assessment*, analisis manajemen kunci, serta pengujian terhadap kemungkinan serangan pada lapisan aplikasi.

Pembahasan

Hasil implementasi menunjukkan bahwa DocuSafe telah mengintegrasikan autentikasi pengguna, validasi unggah, enkripsi, penyimpanan ciphertext, pengelolaan metadata, dan dekripsi ke dalam satu alur pengamanan dokumen. Integrasi tersebut membentuk mekanisme perlindungan berlapis, di mana autentikasi digunakan untuk membatasi akses aplikasi, sedangkan enkripsi digunakan untuk melindungi isi dokumen pada saat disimpan.

Penggunaan AES-256-CBC memberikan mekanisme enkripsi simetris terhadap isi dokumen. Secara standar, AES mendukung ukuran kunci hingga 256 bit dan dirancang untuk melindungi data elektronik. Dalam konteks penelitian ini, penggunaan AES-256-CBC terutama diarahkan untuk menjaga kerahasiaan (*confidentiality*) dokumen yang disimpan.

Aspek lain yang penting adalah pemisahan antara ciphertext dan kunci. Dokumen terenkripsi disimpan pada *storage*, sedangkan kunci tidak disimpan sebagai metadata dokumen. Pendekatan tersebut membatasi kemungkinan diperolehnya kunci secara langsung melalui pembacaan tabel dokumen. Meskipun demikian, keamanan keseluruhan tetap bergantung pada implementasi pengelolaan kunci, keamanan lingkungan server, keamanan akun pengguna, dan perlindungan terhadap akses tidak sah.

Penggunaan IV yang berbeda pada setiap proses enkripsi juga menjadi bagian penting dalam implementasi AES-CBC. IV berfungsi sebagai parameter awal dalam proses CBC dan tidak diperlakukan sebagai kunci rahasia. Oleh karena itu, penyimpanannya bersama metadata dokumen tidak dengan sendirinya menghilangkan kerahasiaan kunci.

Dari perspektif keamanan informasi, hasil penelitian ini terutama menunjukkan dukungan terhadap tujuan *confidentiality*. NIST mendefinisikan *confidentiality* sebagai perlindungan terhadap akses dan pengungkapan informasi yang tidak sah, sedangkan *integrity* berkaitan dengan perlindungan terhadap perubahan atau penghancuran yang tidak semestinya dan *availability* berkaitan dengan ketersediaan informasi ketika diperlukan.

Dengan demikian, keberhasilan pengujian fungsional tidak secara otomatis dapat diinterpretasikan sebagai pembuktian keamanan menyeluruh pada ketiga aspek CIA Triad. Temuan penelitian lebih tepat diposisikan sebagai bukti bahwa mekanisme fungsional pengamanan dokumen, khususnya autentikasi, enkripsi, dekripsi, dan pengelolaan penyimpanan ciphertext, telah berjalan sesuai rancangan.

Hasil tersebut juga memberikan ruang pengembangan untuk penelitian berikutnya. Pengujian dapat diperluas dengan mengukur waktu proses enkripsi dan dekripsi berdasarkan ukuran berkas, penggunaan sumber daya sistem, pengujian terhadap berbagai jenis dokumen, serta evaluasi keamanan melalui *penetration testing* dan *vulnerability assessment*. Pengembangan selanjutnya juga dapat mempertimbangkan mekanisme *authenticated encryption*, seperti AES-GCM, apabila kebutuhan sistem mencakup perlindungan kerahasiaan sekaligus autentikasi integritas ciphertext secara lebih terintegrasi.

KESIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian, penelitian ini berhasil menghasilkan sistem keamanan penyimpanan dokumen digital berbasis web yang mengintegrasikan autentikasi pengguna, pengunggahan dokumen, enkripsi, dekripsi, pengunduhan, dan penghapusan dokumen. Sistem dikembangkan menggunakan *framework* Laravel dengan penerapan algoritma AES-256-CBC sebagai mekanisme perlindungan terhadap isi dokumen yang disimpan.

Implementasi AES-256-CBC menunjukkan bahwa dokumen yang diunggah dapat diproses menjadi ciphertext sebelum disimpan pada *storage* aplikasi, sehingga dokumen tidak tersimpan dalam bentuk *plaintext*. Penggunaan *Initialization Vector* (IV) sebagai bagian dari parameter enkripsi memungkinkan setiap proses enkripsi menggunakan nilai IV yang berbeda, sedangkan nilai tersebut disimpan sebagai metadata yang diperlukan dalam proses dekripsi.

Penelitian ini juga menerapkan mekanisme pengelolaan *encryption key* dengan tidak menyimpan kunci sebagai metadata dokumen pada basis data. Kunci ditampilkan kepada pengguna setelah proses enkripsi dan digunakan kembali ketika dokumen akan didekripsi. Mekanisme tersebut memberikan pemisahan antara penyimpanan ciphertext dan informasi kunci, meskipun keamanan keseluruhan sistem tetap bergantung pada pengamanan kunci oleh pengguna serta keamanan lingkungan aplikasi dan server.

Hasil *Black Box Testing* terhadap fungsi utama sistem menunjukkan bahwa seluruh skenario pengujian yang dirancang memperoleh status berhasil. Pengujian mencakup autentikasi, validasi unggah dokumen, proses enkripsi, dekripsi, pemulihan dokumen, dan penghapusan dokumen. Dengan demikian, berdasarkan ruang lingkup pengujian yang dilakukan, sistem telah memenuhi kebutuhan fungsional yang ditetapkan dalam penelitian.

Untuk pengembangan selanjutnya, penelitian dapat diarahkan pada penerapan *Two-Factor Authentication* (2FA), pengembangan mekanisme manajemen kunci yang lebih kuat, penggunaan skema kriptografi yang sekaligus menyediakan autentikasi integritas, serta integrasi dengan *cloud storage*. Selain itu, penelitian berikutnya dapat melakukan evaluasi performa berdasarkan variasi ukuran dan jenis dokumen serta melengkapi pengujian fungsional dengan *vulnerability assessment* dan *penetration testing* untuk memperoleh evaluasi keamanan yang lebih komprehensif.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Bapak Rizaldy Khair, S.Kom., M.Kom. selaku dosen pembimbing yang telah memberikan bimbingan dan arahan dalam penyelesaian penelitian ini, serta kepada Program Studi Sistem Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara yang telah memberikan dukungan dalam pelaksanaan penelitian ini.

DAFTAR PUSTAKA

- Aditya, F., & Romli, M. A. (2025). Implementasi metode kriptografi Advanced Encryption Standard 256 bit berbasis web dan mobile pada pengamanan dokumen notaris. *TIN: Terapan Informatika Nusantara*, 6(6), 707–714. <https://doi.org/10.47065/tin.v6i6.8637>
- Amanullah, S. (2025). Implementasi keamanan data berbasis web menggunakan kombinasi algoritma Secure Hash Algorithm 256 dan Advanced Encryption Standard Cipher Block Chaining 256 dengan manajemen database Firestore pada akun website [Skripsi, Universitas Teknologi Nusantara].
- Ayu Pramesthi Wardhani, T. D., & Asriningtias, Y. (2024). Aplikasi pengamanan dokumen digital perusahaan berbasis Android menggunakan algoritma AES-256. *Jurnal Sistem dan Teknologi Informasi (JUSTIN)*, 12(1), 143–154. <https://doi.org/10.26418/justin.v12i1.71449>
- Baso, F., & Anriani L, N. (2024). Implementasi teknik kriptografi dengan metode AES 256 untuk keamanan file. *Information Technology Education Journal*, 3(3), 84–87.
- Fachrezi, M. R., Amsyah, D. O., Syahputra, A., & Rusydi, I. (2026). Perancangan dan implementasi sistem enkripsi data sensitif menggunakan AES-256-CBC pada aplikasi berbasis web sederhana. *Jurnal Ilmu Komputer dan Teknik Informatika*, 2(1), 55–62. <https://doi.org/10.64803/juikti.v2i1.100>
- Hayat Arka Putri, Z., Yustraini, Y. A., Ariansyah, R., Nisa, N. C., Wahyuni, E. D., & Putra, A. B. (2025). Pengamanan data akademik berbasis web dengan enkripsi AES-256 (Studi kasus pada pendataan digital SMA XYZ). *Jurnal Nasional Teknologi Informasi dan Aplikasinya*, 3(3), 707–716. <https://doi.org/10.24843/JNATIA.2025.v03.i03.p26>
- Indriyawati, H., Winarti, T., & Vydia, V. (2021). Web-based document certification system with Advanced Encryption Standard digital signature. *Indonesian Journal of Electrical Engineering and Computer Science*, 22(1), 516–521. <https://doi.org/10.11591/ijeecs.v22.i1.pp516-521>
- Irvai, M., & Efranda, N. (2024). Optimalisasi enkripsi file menggunakan algoritma AES-256 berbasis web dengan integrasi kompresi adaptif. *Jurnal Ilmiah Betrik*, 15(3), 1–9.
- Kaaffah, F. M., Nugroho, N., Nurnaningsih, D., & Harriansyah. (2025). Sistem enkripsi dokumen digital melalui kombinasi AES-128 dan hashing SHA256 berbasis salt. *Jurnal Ilmiah FIFO*, 17(1), 78–90. <https://doi.org/10.22441/fifo.2025.v17i1.009>
- Naqiyah, A., Hayati, L. N., & Faradibah, A. (2025). Implementasi algoritma AES-256 berbasis web untuk pengamanan dokumen digital di CV. Karya Eksklusif Nusantara. *Jurnal Ilmiah Informatika Komputer*, 30(3), 275–286. <https://doi.org/10.35760/ik.2025.v30i3.60>
- National Institute of Standards and Technology. (2023). *Advanced Encryption Standard (AES)* (FIPS PUB 197, updated). U.S. Department of Commerce. <https://doi.org/10.6028/NIST.FIPS.197-upd1>
- Permana, A. A., Maulindar, J., & Hartanti, D. (2025). Implementasi sistem kriptografi algoritma AES (256-bit) berbasis web API untuk mengamankan data pribadi di CV. Elang Cahaya Sukses Surakarta. *Jurnal Teknik Informatika*, 4(2), 1.
- Putra, J. S., Ardianto, R., & Purwono. (2024). Tinjauan terhadap implementasi Advanced Encryption Standard 256 dalam keamanan data. *Journal Computer Science and Information Technology*, 5(2), 335–355.
- Reddy Penubadi, H., Shah, P., Sekhar, R., Alrasheedy, M. N., Niu, Y., Dheyaa Radhi, A., Tharwat, M., Fadhil Tawfeq, J., Muwafaq Ghani, H., & Shawkat Abdulbaqi, A. (2023). Sustainable electronic document security: A comprehensive framework integrating encryption, digital signature and

- watermarking algorithms. *Health Security and Defence*, 5(2), 391–404. <https://doi.org/10.37868/hsd.v4i1.359>
- Ridho, A., & Romli, M. A. (2024). Sistem pengamanan dokumen menggunakan algoritma kriptografi Advanced Encryption Standard (AES-256). *Jurnal Informatika Teknologi dan Sains (JINTEKS)*, 6(4), 1044–1052.
- Surya, J., Louis, A., Rini, F., Mulyati, S., & Elzas. (2025). Cryptographic framework for cloud-based document storage using AES-256 and SHA-256 hybrid systems. *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, 11(2), 535–549. <https://doi.org/10.33480/jitk.v11i2.7132>
- Widodo, B. E., & Purnomo, A. S. (2020). Implementasi Advanced Encryption Standard pada enkripsi dan dekripsi dokumen rahasia Ditintelkam Polda DIY. *Jurnal Teknik Informatika (JUTIF)*, 1(2), 69–77. <https://doi.org/10.20884/1.jutif.2020.1.2.21>