Jurnal Komputer Antartika
Volume 4 Nomor 2 2026 Halaman 87-92
ISSN (Online) 2988-5272
DOI: https://doi.org/10.70052/jka.v4i2.1376

Pengembangan *Digital Vault Multi-Format* dan Pengujian Kerentanan Terhadap *Unauthorized Access*

Ahmad Suhaimi[1*], Muhlis Tahir[2], Anisatur Rohmah[3]

[1,2,3] Program Studi Pendidikan Informatika, Fakultas Keguruan dan Ilmu Pendidikan,
Universitas Trunojoyo Madura
Jl. Raya Telang, PO Box 2, Kamal, Bangkalan, Madura, Indonesia

e-mail korespondensi : 230631100037@student.trunojoyo.ac.id

Submit: 23-05-2026 | Revisi: 04-06-2026 | Terima: 19-06-2026 | Terbit online: 01-07-2026

Abstrak - Perkembangan teknologi pengelolaan dokumen digital menuntut mekanisme keamanan tingkat file yang andal untuk mencegah risiko kebocoran data akibat unauthorized access. Banyak perangkat lunak pengamanan saat ini hanya optimal dalam menangani dokumen teks statis, namun mengalami kegagalan berupa kerusakan struktur file (file corruption) saat memproses berkas multimedia berukuran besar. Penelitian ini merancang dan mengimplementasikan sistem Digital Vault universal multi-format berbasis Python dan Streamlit dengan mengintegrasikan algoritma Advanced Encryption Standard (AES-128) untuk enkripsi konten berkas dan Secure Hash Algorithm (SHA-256) untuk pengamanan kredensial serta pembentukan kunci kunci kriptografi yang presisi. Sistem ini dirancang untuk mendukung fungsionalitas multi-format, mencakup dokumen administratif (.pdf, .xls) hingga berkas multimedia (.jpg, .mp4) secara utuh (lossless). Pengujian keamanan dilakukan secara agresif melalui tiga parameter serangan nyata: pembukaan paksa (Bypass Test), kunci salah (Invalid Key Test), dan simulasi pencurian berkas keluar dari platform server (Data Exfiltration Test). Hasil pengujian membuktikan bahwa Digital Vault berhasil mempertahankan integritas data tepat 100%, kebal terhadap bypass aplikasi standar, secara akurat menolak kunci tidak sah, dan file cipherteks yang dicuri tetap mempertahankan enkapsulasi keamanannya sehingga tidak dapat dieksploitasi oleh peretas.

Kata Kunci : Digital Vault, Multi-Format, AES-128, SHA-256, Unauthorized Access

Abstract *- The advancement of digital document management technology demands reliable file-level security mechanisms to prevent data leakage risks due to unauthorized access. Many current security software applications are only optimized for static text documents, yet they suffer from file corruption when processing large multimedia files. This study designs and implements a universal multi-format Digital Vault system based on Python and Streamlit, integrating the Advanced Encryption Standard (AES-128) for file content encryption and the Secure Hash Algorithm (SHA-256) for credential securing and precise cryptographic key derivation. The system is engineered to support multi-format functionalities, ranging from administrative documents (.pdf, .xls) to multimedia assets (.jpg, .mp4) with lossless integrity. Security testing was aggressively conducted through three real-world attack parameters: forced opening (Bypass Test), wrong cryptographic key submission (Invalid Key Test), and data theft simulation outside the host server platform (Data Exfiltration Test). The empirical results prove that the Digital Vault successfully maintains 100% data integrity, is fully resilient against standard application bypasses, accurately rejects unauthorized keys, and the exfiltrated ciphertext completely retains its security envelope, rendering it unexploitable to intruders.*

Keywords : Digital Vault, Multi-Format, AES-128, SHA-256, Unauthorized Access

1. Pendahuluan

Perkembangan teknologi informasi saat ini mengalami kemajuan yang sangat pesat, khususnya dalam hal pengolahan dan pertukaran data digital. Berbagai aktivitas penting seperti pengiriman dokumen, penyimpanan file, hingga distribusi informasi kini dilakukan secara digital di hampir seluruh sektor administrasi operasional. Dalam lingkungan profesional modern, data kontrak hukum, laporan keuangan berbasis lembar kerja (*Excel*), hingga bukti visual berupa citra gambar dan rekaman video telah menjadi aset digital yang sangat berharga bagi jalannya roda organisasi. Namun, di balik kemudahan dan efisiensi yang ditawarkan, terdapat risiko keamanan siber yang mengancam, salah satunya adalah kebocoran data akibat lemahnya proteksi pada sistem penyimpanan [2],[6].

Permasalahan kebocoran data ini sering kali terjadi pada sistem informasi berbasis web yang belum memiliki mekanisme keamanan terdesentralisasi yang memadai, sehingga data menjadi rentan terhadap tindakan pencurian, penyadapan, dan penyalahgunaan oleh pihak luar. Data yang tidak terlindungi dengan baik di dalam

direktori penyimpanan server dapat dengan mudah diakses oleh pihak yang tidak berwenang (*Unauthorized Access*). Banyak instansi yang sebenarnya telah menyadari pentingnya sistem enkripsi, namun perangkat lunak keamanan yang beredar di pasar sering kali hanya berfokus pada pengamanan dokumen statis berbasis teks. Ketika sistem tersebut dipaksakan untuk memproses berkas multimedia yang memiliki karakteristik struktur data, *header*, dan metadata yang berbeda, sistem gagal mengenali pola biner tersebut sehingga hasil dekripsi sering kali berujung pada kerusakan file (*file corruption*) [3],[5].

Limitasi operasional ini terlihat jelas pada penelitian-penelitian terdahulu. Sebagai contoh, sistem pengamanan kombinasi AES-128 dan SHA-256 berbasis web yang dikembangkan oleh Kaaffah dkk. mencatatkan batasan teknis di mana kapasitas berkas yang diunggah dibatasi maksimal sebesar 5 MB dan hanya mendukung file teks atau PDF. Kegagalan algoritma dekripsi dalam merekonstruksi data awal multimedia menandakan hilangnya integritas data (*data integrity*) yang mutlak dibutuhkan dalam sebuah sistem keamanan informasi [1],[4].

Untuk mengatasi rentetan permasalahan tersebut, diperlukan suatu arsitektur keamanan tingkat berkas (*file-level encryption*) yang tidak hanya membatasi hak akses pengguna pada menu halaman *login*, melainkan melekat langsung untuk melindungi struktur internal berkas. Salah satu solusi canggih yang diajukan dalam penelitian ini adalah pengembangan sistem *Digital Vault* universal multi-format berbasis bahasa pemrograman Python dan kerangka kerja (*framework*) Streamlit. Sistem ini dirancang secara khusus untuk mampu melakukan proses konversi bolak-balik dari dokumen asli menjadi teks sandi (*cipherteks*) terenkripsi dan mengembalikannya secara presisi 100% ke bentuk semula tanpa merusak integritas data (*lossless*), baik pada resolusi video, kualitas citra gambar, maupun format baris data pada berkas akuntansi [8],[9].

Inovasi utama dari penelitian ini terletak pada fleksibilitas penanganan karakteristik berkas multi-format skala besar yang diintegrasikan dengan fase pengujian penetrasi keamanan tingkat lanjut (*vulnerability assessment*). Pengujian sistematis dijalankan untuk mengevaluasi ketahanan file sandi dari berbagai skenario serangan *unauthorized access* seperti upaya pembukaan paksa (*bypass test*), manipulasi kunci (*invalid key test*), dan skenario pengrusakan perimeter melalui eksfiltrasi data (*data exfiltration*) keluar dari lingkungan aplikasi.[12],[10]

Kebaruan dari penelitian ini terletak pada tiga aspek utama yang membedakannya dari penelitian sebelumnya. Pertama, sistem yang dikembangkan mendukung enkripsi dan dekripsi berkas multi-format secara universal, mencakup dokumen administratif maupun berkas multimedia berukuran besar tanpa mengalami kerusakan struktur file (*file corruption*), mengatasi keterbatasan mendasar pada sistem terdahulu yang hanya mampu menangani dokumen teks statis berukuran kecil. Kedua, sistem ini mengintegrasikan dua algoritma kriptografi sekaligus, yaitu AES-128 untuk enkripsi konten dan SHA-256 untuk pengamanan kredensial dan derivasi kunci, sehingga menghasilkan lapisan perlindungan berlapis yang lebih andal. Ketiga, penelitian ini melengkapi pengembangan sistem dengan pengujian penetrasi terstruktur menggunakan tiga skenario serangan nyata (*bypass test*, *invalid key test*, dan *data exfiltration test*) untuk memvalidasi ketahanan sistem terhadap ancaman *unauthorized access* di dunia nyata, sesuatu yang belum dilakukan secara komprehensif pada penelitian-penelitian terdahulu di bidang ini.

2. Metode Penelitian

Penelitian ini menerapkan metodologi rekayasa perangkat lunak terstruktur yang dikombinasikan dengan analisis ketahanan kriptografi siber. Alur tahapan perancangan hingga evaluasi sistem mengadopsi kerangka kronologis ilmiah yang terdiri atas beberapa fase utama.

a. Identifikasi Masalah dan Analisis Kebutuhan

Pada tahapan awal, dilakukan pemetaan masalah terhadap kerentanan sistem penyimpanan dokumen konvensional. Masalah utama yang ditemukan adalah tingginya risiko pembacaan biner secara langsung oleh peretas saat berhasil menembus direktori server, serta ketidakmampuan platform *existing* dalam mengeksekusi enkripsi multi-format secara aman. Berdasarkan analisis tersebut, ditentukan kebutuhan fungsional sistem, antara lain: (1) menyediakan form interaktif untuk unggahan dokumen multi-format, (2) memproses enkripsi blok berbasis AES-128 tanpa merusak *header* berkas asli, (3) menghasilkan kunci kriptografi unik berbasis turunan SHA-256, dan (4) menyediakan mekanisme verifikasi kunci autentikasi saat pengunduhan berkas hasil dekripsi.

b. Perancangan Sistem dan Implementasi Python-Streamlit

Sistem *Digital Vault* diwujudkan dalam bentuk aplikasi berbasis web interaktif memanfaatkan efisiensi bahasa pemrograman Python dan pustaka Streamlit. Python dipilih karena keandalannya yang superior dalam menangani operasi komputasi matematis tingkat tinggi serta memanipulasi susunan *byte* pada berkas biner multimedia. Antarmuka Streamlit mempermudah pengguna untuk berinteraksi dengan sistem keamanan melalui form enkripsi yang ringkas dan praktis.

c. Mekanisme Kriptografi Multi-Format Bolak-Balik

Aplikasi mengamankan aset digital menggunakan algoritma Advanced Encryption Standard (AES) dengan panjang kunci 128 bit. Berkas multimedia yang diunggah akan diurai menjadi representasi blok biner tetap berukuran 128 bit (16 byte). Setiap blok data dimasukkan ke dalam matriks *state* untuk melewati 10 putaran (*round*) transformasi kriptografi standar, meliputi *SubBytes*, *ShiftRows*, *MixColumns*, dan *AddRoundKey*.

Konversi matematika bolak-balik untuk enkripsi dan dekripsi mengikuti model formal pada persamaan (1) dan (2).

$$C = E\_\{K\}(P) \quad (1)$$
$$P = D\_\{K\}(C) \quad (2)$$

Keterangan untuk persamaan (1) dan (2) adalah C merupakan struktur berkas hasil enkripsi (*ciphertext*), P menggambarkan berkas asli digital (*plaintext*) baik berupa PDF, XLS, JPG, maupun MP4 , E melambangkan fungsi transformasi enkripsi biner AES-128, D melambangkan fungsi transformasi dekripsi biner AES-128, dan K melambangkan kunci rahasia (*secret key*) yang diturunkan melalui ekstraksi hash SHA-256.

d. Perancangan Skenario Pengujian Unauthorized Access

Untuk membuktikan ketahanan sistem dari ancaman intrusi siber di mana individu berusaha mendapatkan data tanpa izin yang sah , dirancang tiga skenario pengujian penetrasi destruktif:

1. Pengujian Pembukaan Paksa (*Bypass Test*): Melakukan eksperimen akses langsung pada berkas *ciphertext* menggunakan aplikasi pemutar media atau dokumen eksternal tanpa melewati sistem *Digital Vault*.
2. Pengujian Kunci Salah (*Invalid Key Test*): Menguji ketahanan respons matematis aplikasi ketika penyerang mencoba menebak atau memasukkan kata sandi dekripsi yang salah.
3. Simulasi Eksfiltrasi Data (*Data Exfiltration Test*): Mensimulasikan kondisi di mana berkas sandi berhasil dicuri dari repositori utama server ke perangkat penyimpanan eksternal milik peretas (*flashdisk*) untuk dianalisis struktur internalnya.

3. Hasil dan Pembahasan

Bagian ini memaparkan realisasi antarmuka aplikasi *Digital Vault Multi-Format* yang berhasil dibangun, diikuti oleh analisis mendalam dari hasil pengujian penetrasi terhadap skenario serangan *unauthorized access*.

3.1. Implementasi Antarmuka Sistem Digital Vault

Sistem keamanan *Digital Vault Multi-Format* berhasil dibangun dengan antarmuka pengguna berbasis Streamlit yang interaktif dan responsif. Melalui komponen form di sisi antarmuka, pengguna dapat dengan mudah mengunggah aset digital berharga mereka, baik berupa dokumen administratif maupun berkas multimedia berukuran besar.

Gambar 1 Desain antarmuka

Penjelasan dari alur visualisasi Gambar 1 menyediakan antarmuka form pengamanan dokumen yang menyediakan tombol interaktif *Choose File* yang secara adaptif dapat memproses berbagai ekstensi biner tanpa melakukan pembatasan format (*universal multi-format support*). Setelah pengguna memasukkan teks sandi pada kolom password kunci dan menekan tombol eksekusi, sistem secara otomatis membaca pola *byte* dari file, mengikatnya dengan algoritma kriptografi, dan menyediakan tautan unduhan langsung untuk file *ciphertext* yang aman.

3.2. Hasil Pengujian Performa Berkas Multi-Format

Untuk menguji keandalan sistem dalam memproses tipe berkas yang bervariasi, dilakukan serangkaian uji coba komputasi terhadap berkas administratif dan multimedia berskala besar. Hasil pengujian performa dicatat secara komprehensif pada Tabel 1.

Analisis data dari Tabel 1 memperlihatkan bahwa pengembangan aplikasi ini berhasil memecahkan limitasi kapasitas yang ada pada sistem terdahulu. File multimedia berupa rekaman video berukuran lebih dari 6 MB dapat diproses secara kilat dengan waktu enkripsi hanya sebesar 208 milidetik dan waktu dekripsi sebesar 228 milidetik. Paling penting, status hasil rekonstruksi akhir menunjukkan tingkat keberhasilan mutlak 100% (*lossless decryption*). Hal ini membuktikan bahwa algoritma penanganan *file stream* Python yang diimplementasikan

mampu mengenali karakteristik unik dari *header* dan metadata berkas multimedia, sehingga mengeliminasi risiko kerusakan file (*file corruption*) setelah pemrosesan kriptografi bolak-balik.

Tabel 1. Hasil Pengujian Performa Konversi Berkas Digital Vault

No	Nama File Uji	Format Ekstensi	Ukuran Asli (KB)	Waktu Enkripsi (ms)	Waktu Dekripsi (ms)	Hasil Rekonstruksi Akhir
1	Beswan Djarum	.pdf	256	120	135	100% Utuh Sesuai Asli
2	laporan-keuangan	.xlsx	512	126	140	Format Baris Akurat
3	Dokumentasi_Aset	.jpg	2,048	155	170	Resolusi Citra Tetap
4	Rekaman_Observasi	.mp4	6,144	208	228	Video Lancar Tanpa Korup

3.3 Hasil Pengujian Skenario Serangan Unauthorized Access

Pengujian ketahanan terhadap *unauthorized access* dilakukan dengan memosisikan file sandi di bawah skenario ancaman dunia nyata ketika aset digital tersebut jatuh ke tangan pihak agresor atau entitas luar yang tidak memiliki otoritas sah.

3.3.1 Pengujian Pembukaan Paksa (Bypass Test)

Skenario pertama dilakukan dengan mencoba membuka berkas *ciphertext* hasil enkripsi secara langsung menggunakan perangkat lunak pembaca multimedia bawaan sistem operasi (seperti *Windows Media Player Legacy* atau *PDF Viewer*) tanpa mengeksekusi modul dekripsi aplikasi.

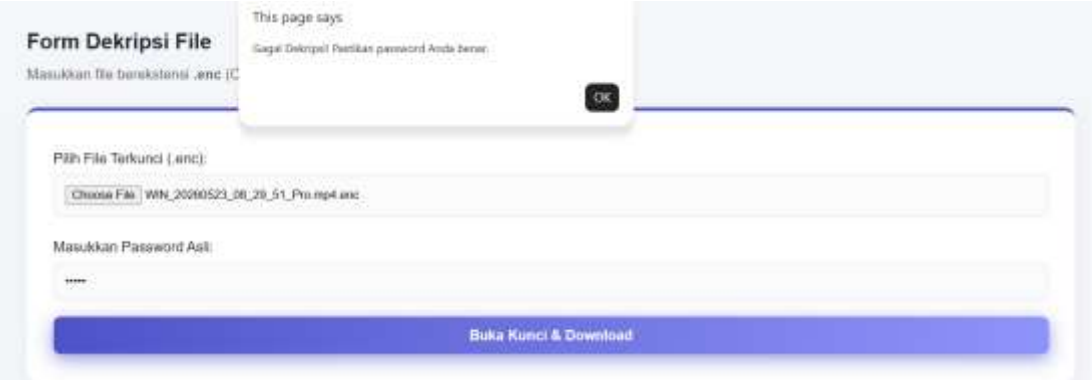
Gambar 2. Pengujian Pembukaan Paksa

Hasil eksperimen yang diilustrasikan oleh dialog sistem pada Gambar 2 membuktikan bahwa *Digital Vault* memiliki kekebalan total terhadap upaya pembukaan paksa (*bypass*). Perangkat lunak bawaan komputer secara instan memunculkan pesan kegagalan membaca *codec* berkas. Fenomena ini terjadi karena keseluruhan struktur internal biner dan penanda identitas awal (*header*) berkas asli telah diacak secara menyeluruh oleh transformasi enkripsi AES-128. Dengan demikian, penyerang tidak akan bisa mendapatkan bocoran informasi visual maupun tekstual apa pun dari dokumen tersebut.

3.3.2 Pengujian Autentikasi Kunci Salah (Invalid Key Test)

Skenario kedua mensimulasikan kondisi di mana penyerang berhasil mendapatkan akses masuk ke halaman aplikasi *Digital Vault*, namun mencoba membongkar proteksi berkas dengan memasukkan kata sandi (kunci dekripsi) yang palsu atau salah.

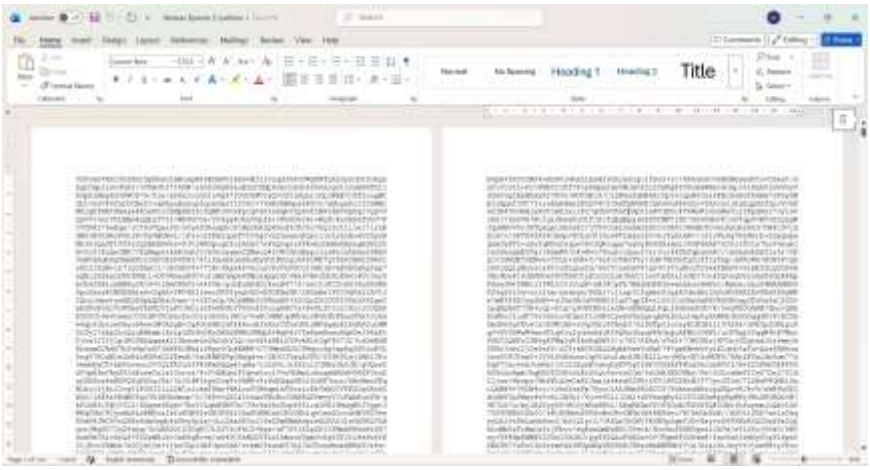
Pada Gambar 3, ketika teks sandi yang tidak valid dimasukkan, aplikasi secara akurat mengidentifikasi adanya ketidaksesuaian nilai matematis kunci. Sistem langsung memblokir alur rekonstruksi data biner dan memunculkan notifikasi peringatan kegagalan seperti yang ditunjukkan pada Gambar 3 Karena proses dekripsi AES simetris menuntut tingkat presisi parameter perhitungan yang mutlak, perbedaan satu karakter saja pada sandi input akan menggagalkan pencocokan *hash* SHA-256, sehingga mengunci rapat akses file dari pengguna ilegal.



Gambar 3 Pengujian Autentikasi Kunci Salah

3.3.3 Simulasi Eksfiltrasi Data (Data Exfiltration Test)

Skenario pengujian terakhir dirancang untuk mensimulasikan ancaman internal (*insider threat*) atau peretasan server tingkat tinggi, di mana berkas sandi berhasil disusupi, dicuri dari repositori utama server, dan dipindahkan ke dalam media penyimpanan eksternal milik peretas. Berkas yang berhasil diculik tersebut kemudian dianalisis menggunakan editor teks tingkat rendah (*low-level text editor*) seperti *Microsoft Word* atau *Notepad++* di luar ekosistem aplikasi kriptografi untuk dicari pola datanya.



Gambar 4 Simulasi eksfiltrasi data

Hasil pengamatan struktur internal berkas pada Gambar 4 menunjukkan bahwa berkas yang dieksfiltrasi keluar dari jaringan utama server tetap mempertahankan integritas keamanannya secara kuat sebagai *cipherteks* sejati. Tanpa adanya modul dekripsi spesifik Python yang memegang kunci rahasia yang cocok, teks sandi tersebut hanya berupa tumpukan karakter acak yang rusak, hancur, dan tidak dapat difungsikan sama sekali oleh pihak penyerang. Hasil ini mengonfirmasi bahwa enkripsi tingkat berkas (*file-level encryption*) yang diimplementasikan pada *Digital Vault* sukses memberikan perlindungan data yang tangguh meskipun perimeter keamanan jaringan server telah ditembus oleh peretas.

4. Kesimpulan

Berdasarkan seluruh rangkaian proses perancangan, implementasi antarmuka, hingga pengujian penetrasi siber yang telah dilaksanakan pada sistem Digital Vault Multi-Format, maka dapat ditarik kesimpulan sebagai berikut,Sistem keamanan *Digital Vault* universal berbasis Python dan Streamlit telah berhasil dikembangkan secara fungsional dan terbukti andal dalam mengamankan ragam aset digital lintas platform, mencakup dokumen administratif (.pdf, .xls) hingga berkas multimedia berukuran besar (.jpg, .mp4). Sistem terbukti mampu menjaga integritas data secara sempurna 100% (*lossless*) saat berkas dikonversi kembali ke bentuk aslinya tanpa mengalami degradasi kualitas visual maupun kerusakan struktural berkas (*file corruption*). Sistem memiliki tingkat ketahanan dan proteksi yang sangat baik dalam menangkal berbagai teknik serangan *Unauthorized Access*. Berkas sandi (*cipherteks*) terbukti kebal dari pembukaan paksa aplikasi eksternal, andal menolak manipulasi kunci, dan tetap terjaga kerahasiaannya meskipun berhasil dicuri keluar dari server lokal. Meskipun sistem telah berjalan dengan sukses memenuhi target fungsionalitasnya, beberapa ruang penyempurnaan masih terbuka lebar. Sebagai saran untuk pengembangan sistem selanjutnya, disarankan untuk mengintegrasikan basis data sistem dengan fitur

pencatatan log aktivitas otomatis (*audit trail / secure logging*) yang terhubung langsung ke basis data server. Fitur ini sangat penting bagi instansi untuk merekam waktu eksekusi biner secara *real-time* serta melacak alamat digital pihak-pihak yang berulang kali gagal melewati verifikasi kunci, sehingga tindakan mitigasi penyusupan siber dapat dilakukan secara lebih dini dan komprehensif.

Referensi

[1] F. M. Kaaffah, N. Nugroho, D. Nurnaningsih, and Harriansyah, "Sistem Enkripsi Dokumen Digital Melalui Kombinasi AES-128 dan Hashing SHA-256 Berbasis Salt," *Jurnal Ilmiah FIFO*, vol. 17, no. 1, pp. 78-90, 2025. https://doi.org/10.22441/fifo.2025.v17i1.007

[2] N. Aprizaldi, M. A. Hasan, and T. Parmonangan, "Aplikasi keamanan data berbasis web menggunakan algoritma AES 128 untuk enkripsi dan dekripsi data," *ZONAsi: Jurnal Sistem Informasi*, vol. 4, no. 1, pp. 12-21, 2022. https://doi.org/10.31849/zn.v4i1.8211

[3] A. Fauzi and D. Rahmawati, "Integrasi teknologi keamanan enkripsi dan autentikasi pengguna pada sistem informasi berbasis web," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 9, no. 3, pp. 455-462, 2022. https://doi.org/10.25126/jtiik.2022939999

[4] F. Kurniawan and N. O. Saputri, "Analisis fleksibilitas dan kemudahan akses pada pengembangan aplikasi pencatatan berbasis web," *Jurnal Sistem Informasi Teknologi Informatika dan Komputer*, vol. 10, no. 2, pp. 159-168, 2020. https://doi.org/10.30998/jsitik.v10i2.6754

[5] D. S. Purwanti, R. A. Putri, and T. Hidayat, "Implementasi algoritma Advanced Encryption Standard (AES-128) untuk peningkatan keamanan data digital," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 8, no. 1, pp. 112-118, 2024. https://doi.org/10.29207/resti.v8i1.5291

[6] A. Ridho and A. Romli, "Sistem Pengamanan Dokumen Menggunakan Algoritma Kriptografi Advanced Encryption Standard (AES-256)," *Jurnal Informatika Teknologi dan Sains (JINTEKS)*, vol. 6, no. 1, pp. 44-51, 2024. https://doi.org/10.36767/jinteks.v6i1.1982

[7] R. Saputra and A. Firmansyah, "Penerapan Algoritma Kriptografi AES 128 Pada Aplikasi Manajemen Pengarsipan Berbasis Web," *Jurnal Keamanan Siber dan Sandi*, vol. 5, no. 2, pp. 34-41, 2023. https://doi.org/10.55229/jkss.v5i2.337

[8] R. Setiani, E. T. Imananda, W. E. Wicaksono, M. A. Baihaqi, and J. Kuswanto, "Perbandingan algoritma AES128 dengan SHA256 dalam kecepatan enkripsi pengiriman data," *JOINS (Journal of Information System)*, vol. 9, no. 1, pp. 50-59, 2024. https://doi.org/10.33633/joins.v9i1.9185

[9] M. Tania, T. S. Alasi, and R. Yap, "Algoritma AES untuk Keamanan Data Digital Berbasis Web di Kantor Desa Aman Damai," *Jurnal TIMES*, vol. 13, no. 2, pp. 142-149, 2024. https://doi.org/10.52698/times.v13i2.4421

[10] I. Priambudi and M. Mufti, "Implementasi Kriptografi Dengan Metode AES-128 Untuk Pengamanan File Berbasis Web Pada SMP Yapipa," *SKANIKA Sistem Komputer dan Tek. Inform.*, vol. 6, no. 1, pp. 22-31, 2023. https://doi.org/10.36080/skanika.v6i1.3267

[11] I. Saputra and S. D. Nasution, "Perbandingan Performa Algoritma Md5 Dan Sha-256 Dalam Membangkitkan Identitas File," *J. Sains Komput. Inform.*, vol. 6, no. 1, pp. 172-187, 2022. https://doi.org/10.30645/j-sakti.v6i1.425

[12] J. Hutagalung, P. S. Ramadhan, and S. J. Sihombing, "Keamanan Data Menggunakan Secure Hashing Algorithm (SHA)-256 dan Rivest Shamir Adleman (RSA) pada Digital Signature," *J. Teknologi Informasi dan Ilmu Komputer*, vol. 10, no. 6, pp. 1213-1222, 2023. https://doi.org/10.25126/jtiik.2023106897

[13] H. Herman, R. Wijaya, S. Miharja, and Wilson, "Implementasi Algoritma AES-128 dan SHA-256 dalam Perancangan Aplikasi Pengamanan File Dokumen," *Jurnal TIMES*, vol. 10, no. 2, pp. 80-87, 2022. https://doi.org/10.52698/times.v10i2.2991

[14] M. Muhammed *et al.*, "Comparative Analysis of AES, Blowfish, Twofish, Salsa20, and ChaCha20 for Image Encryption," *Kurdistan J. Appl. Res.*, vol. 9, no. 1, pp. 52-65, 2024. https://doi.org/10.24017/science.2024.9.1.5

[15] K. A. Pikatan, "Perancangan Aplikasi Keamanan Data Kriptografi Modern AES-128 (Advanced Encryption Standard) Berbasis Website," S1 thesis, Universitas Mercu Buana, Bekasi, 2022. https://doi.org/10.26905/thesis.2022