ALGORITMA AES-256 UNTUK PROSES ENKRIPSI–DEKRIPSI FILE DAN ALGORITMA SHA-256 UNTUK VALIDASI INTEGRITAS FILE

SKRIPSI

DISUSUN OLEH

NABILA YUSRA
NPM. 2209010118

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
MEDAN
2026

ALGORITMA AES-256 UNTUK PROSES ENKRIPSI–DEKRIPSI FILE DAN ALGORITMA SHA-256 UNTUK VALIDASI INTEGRITAS FILE

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer (S.Kom) dalam Program Studi Sistem Informasi pada Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara

NABILA YUSRA
NPM. 2209010118

PROGRAM STUDI SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
MEDAN
2026

LEMBAR PENGESAHAN

Judul Skripsi	:	ALGORITMA AES-256 UNTUK PROSES ENKRIPSI-DEKRIPSI FILE DAN ALGORITMA SHA-256 UNTUK VALIDASI INTEGRITAS FILE
Nama Mahasiswa	:	NABILA YUSRA
NPM	:	2209010118
Program Studi	:	SISTEM INFORMASI

Menyetujui
Komisi Pembimbing

(Indah Purnama Sari, ST., M.Kom)
NIDN. 0116049001

Ketua Program Studi

(Mahardika Abdi Prawira Tanjung, S.Kom., M.Kom)
NIDN. 0117088902

Dekan

(Dr. Al-Khowarizmi, S.Kom., M.Kom.)
NIDN. 0127099201

PERNYATAAN ORISINALITAS

ALGORITMA AES-256 UNTUK PROSES ENKRIPSI–DEKRIPSI FILE DAN ALGORITMA SHA-256 UNTUK VALIDASI INTEGRITAS FILE

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, Mei 2026

Yang membuat pernyataan

Nabila Yusra

NPM. 2209010118

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama	: Nabila Yusra
NPM	: 2209010118
Program Studi	: Sistem Informasi
Karya Ilmiah	: Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bedas Royalti Non-Eksekutif *(Non-Exclusive Royalty free Right)* atas penelitian skripsi saya yang berjudul:

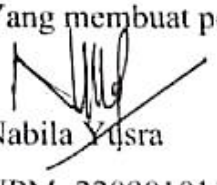
ALGORITMA AES-256 UNTUK PROSES ENKRIPSI–DEKRIPSI FILE DAN ALGORITMA SHA-256 UNTUK VALIDASI INTEGRITAS FILE

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, Mei 2026

Yang membuat pernyataan



Nabila Yusra

NPM. 2209010118

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap	: Nabila Yusra
Tempat dan Tanggal Lahir	: Martubung, 03 Oktober 2004
Alamat Rumah	: Jl. Kl Yos Sudarso Sp. Atap Lk. 7
Telepon/Faks/HP	: 082164844454
E-mail	: nabillayusrah845@gmail.com
Instansi Tempat Kerja	: -
Alamat Kantor	: -

DATA PENDIDIKAN

SD	: SD ALWASHLIYAH 29	TAMAT: 2016
SMP	: MTS YASPI LABUHANDELI	TAMAT: 2019
SMA	: SMK SINAR HUSNI	TAMAT: 2022

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT atas segala rahmat, karunia, kesehatan, serta kesempatan yang telah diberikan sehingga penulis dapat menyelesaikan skripsi yang berjudul "Algoritma AES-256 Untuk Proses Enkripsi Dekripsi File dan Algoritma SHA-256 Untuk Validasi Integritas File" dengan baik dan tepat pada waktunya. Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana pada Program Studi Sistem Informasi. Shalawat serta salam senantiasa tercurah kepada Nabi Muhammad SAW yang telah membawa umat manusia dari zaman kebodohan menuju zaman yang penuh dengan ilmu pengetahuan seperti saat ini.

Dalam proses penyusunan skripsi ini, penulis menyadari bahwa banyak hambatan dan kesulitan yang dihadapi. Namun, berkat doa, dukungan, bantuan, serta motivasi dari berbagai pihak, penulis dapat menyelesaikan skripsi ini dengan baik. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Dr. Akrim, M.Pd., Rektor Universitas Muhammadiyah Sumatera Utara (UMSU).
2. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom. selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
3. Ibu Dr. Firahmi Rizky, S.Kom., M.Kom selaku Wakil Dekan I Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.

4. Bapak Mhd. Basri, S.Si, M.Kom selaku Wakil Dekan III Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
5. Bapak Mahardika Abdi Prawira Tanjung, S.Kom., M.Kom selaku Ketua Program Studi Sistem Informasi.
6. Bapak Mulkan Azhari, S.Kom., M.Kom selaku Sekretaris Program Studi Sistem Informasi.
7. Ibu Indah Purnama Sari, ST., M.Kom selaku dosen pembimbing yang telah sabar memberikan arahan, ilmu, dan motivasi dalam penyempurnaan skripsi ini.
8. Bapak/Ibu Dosen Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara yang telah mendidik, membimbing, dan memberikan ilmu yang bermanfaat kepada penulis selama masa perkuliahan hingga dapat menyelesaikan skripsi ini dengan baik.
9. Secara khusus, penulis ingin mengucapkan terima kasih yang setulus-tulusnya kepada kedua orang tua tercinta, Ayah Muhammad Subakti dan Ibu Fauziah Hanum, yang selalu menjadi sumber kekuatan terbesar bagi penulis. Terima kasih atas segala doa yang tidak pernah putus, kasih sayang yang tulus, perhatian, pengorbanan, dukungan moral maupun materi, serta semangat yang selalu diberikan kepada penulis dalam setiap langkah kehidupan. Tanpa dukungan dan doa dari orang tua, penulis tidak akan mampu menyelesaikan pendidikan dan skripsi ini dengan baik.

10. Penulis juga mengucapkan terima kasih kepada teman-teman seperjuangan yang selalu memberikan bantuan, motivasi, dukungan, serta semangat kepada penulis selama masa perkuliahan dan penyusunan skripsi. Kehadiran teman-teman memberikan banyak pengalaman, pelajaran, dan kenangan yang berarti bagi penulis. Terima kasih karena telah menjadi tempat berbagi cerita, berdiskusi, saling membantu dalam menghadapi kesulitan, serta memberikan semangat ketika penulis merasa lelah dan kehilangan motivasi dalam menyelesaikan skripsi ini.
11. Selain itu, penulis juga ingin berterima kasih kepada diri sendiri karena telah mampu bertahan, berjuang, dan tidak menyerah selama proses penyusunan skripsi ini. Terima kasih karena tetap kuat menghadapi berbagai tekanan, rasa lelah, keraguan, dan hambatan yang ada. Perjalanan menyelesaikan skripsi ini bukanlah hal yang mudah, namun penulis berhasil melewati semua proses tersebut dengan usaha, kesabaran, dan keyakinan untuk terus maju hingga akhirnya dapat menyelesaikan penelitian ini.
12. Semua pihak yang terlibat langsung ataupun tidak langsung yang tidak dapat penulis ucapkan satu-persatu yang telah membantu penyelesaian skripsi ini.

Penulis menyadari bahwa skripsi ini masih jauh dari kata sempurna. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun demi perbaikan di masa yang akan datang. Penulis berharap skripsi ini dapat

memberikan manfaat dan menambah wawasan bagi pembaca serta dapat menjadi referensi untuk penelitian selanjutnya.

ALGORITMA AES-256 UNTUK PROSES ENKRIPSI– DEKRIPSI FILE DAN ALGORITMA SHA-256 UNTUK VALIDASI INTEGRITAS FILE

ABSTRAK

Pesatnya pertukaran data digital di era siber meningkatkan risiko akses ilegal dan modifikasi data terhadap dokumen sensitif. Penelitian ini bertujuan merancang dan mengimplementasikan aplikasi pengamanan file dokumen (*offline*) dengan menggabungkan teknik kriptografi simetris menggunakan algoritma *Advanced Encryption Standard* (AES-256) untuk menjaga kerahasiaan data dan *Secure Hash Algorithm* (SHA-256) untuk memvalidasi integritas data. Metode penelitian meliputi analisis kebutuhan, perancangan alur sistem, implementasi sistem berbasis Python menggunakan Visual Studio Code, serta pengujian sistem terhadap berbagai ukuran berkas dengan format *.txt, .pdf, dan .docx* yang bersumber dari platform Kaggle. Sistem bekerja dengan membangkitkan nilai hash awal file asli menggunakan SHA-256, mengunci file melalui 14 putaran proses enkripsi AES-256, serta melakukan dekripsi dan membandingkan kembali nilai hash akhir untuk validasi. Hasil pengujian menunjukkan bahwa kombinasi kedua algoritma ini sangat efektif meningkatkan keamanan dokumen. Algoritma AES-256 berhasil menyamarkan *plaintext* menjadi *ciphertext* tanpa merusak file asli. Sementara itu, algoritma SHA-256 mampu mendeteksi perubahan sekecil apa pun pada data melalui efek *avalanche* secara akurat. Sistem menyatakan file valid jika nilai hash sebelum enkripsi dan sesudah dekripsi bernilai identik. Pengukuran performa menunjukkan bahwa lama waktu eksekusi proses sangat dipengaruhi oleh karakteristik ukuran file yang diuji. Aplikasi prototipe ini dapat menjadi solusi praktis dalam melindungi keutuhan dan kerahasiaan data digital dokumen.

Kata Kunci: Kriptografi, Enkripsi, Dekripsi, AES-256, SHA-256, Integritas File.

AES-256 ALGORITHM FOR FILE ENCRYPTION – DECRYPTION PROCESS AND SHA-256 ALGORITHM FOR FILE INTEGRITY VALIDATION

ABSTRACT

The rapid exchange of digital data increases the risk of unauthorized access and data modification of sensitive documents. This study aims to design and implement an offline document security application by combining symmetric cryptographic techniques using the Advanced Encryption Standard (AES-256) algorithm to maintain data confidentiality and the Secure Hash Algorithm (SHA-256) to validate data integrity. The research methodologies encompass requirement analysis, system workflow design, Python-based implementation utilizing Visual Studio Code, and system testing across various file sizes with .txt, .pdf, and .docx formats sourced from the Kaggle platform. The system operates by generating an initial hash value of the original file via SHA-256, securing the file through 14 encryption rounds of AES-256, followed by decryption and comparing the final hash value for validation. The test results indicate that the combination of both algorithms significantly enhances document security. The AES-256 algorithm successfully transforms plaintext into ciphertext without damaging the original file structure. Meanwhile, the SHA-256 algorithm accurately detects even the slightest modifications through the avalanche effect. The system declares the file valid if the pre-encryption and post-decryption hash values are identical. Performance evaluation reveals that the execution runtime is heavily driven by the characteristics of the tested file sizes. This prototype application serves as a practical solution for safeguarding digital data integrity and confidentiality.

Keywords: Cryptography, Encryption, Decryption, AES-256, SHA-256, File Integrity.

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan teknologi informasi yang terus meningkat membuat pertukaran dan penyimpanan data digital menjadi kebutuhan penting di berbagai bidang, seperti pendidikan, pemerintahan, dan dunia usaha. File digital seperti dokumen teks, laporan, atau arsip penting biasanya menyimpan informasi yang bersifat rahasia dan sangat sensitif. Jika data penting tersebut bocor, maka kerahasiaannya bisa terganggu dan memberikan kesempatan bagi pihak tidak berwenang untuk menyebarkan data tersebut. Hal ini tentu bisa menimbulkan kerugian besar bagi individu atau organisasi yang terkena dampaknya. Oleh karena itu, dibutuhkan suatu cara atau metode yang bisa menjadi solusi untuk melindungi data atau informasi yang disimpan atau dikirimkan melalui platform digital agar tidak mudah diakses oleh pihak yang tidak berwenang.

Kriptografi adalah teknik yang digunakan untuk menjaga keamanan data dengan fokus pada beberapa hal penting, seperti kerahasiaan, integritas data, otentikasi, dan mencegah penyangkalan. Secara umum, cara kerja kriptografi adalah dengan mengubah data asli yang bisa dibaca (plaintext) menjadi data yang tidak terbaca (ciphertext). Enkripsi dan dekripsi data adalah langkah penting untuk menjaga rahasia informasi. Enkripsi adalah cara mengubah pesan yang asli menjadi bentuk yang tidak bisa dibaca tanpa kunci tertentu. Sementara itu, dekripsi adalah proses sebaliknya, yaitu mengembalikan pesan yang sudah dienkripsi ke bentuk aslinya agar bisa dibaca kembali (Jehian et al., 2025).

Salah satu cara yang sering digunakan untuk menjaga data tetap rahasia adalah dengan menggunakan kriptografi. Dalam penelitian ini, Advanced Encryption Standard (AES) merupakan standar algoritma kriptografi simetris yang terbaru dan mampu mengatasi tantangan dalam perkembangan teknologi komunikasi yang cepat. AES juga memiliki keunggulan dalam hal keamanan, kecepatan, dan kemudahan dalam penerapan. AES mampu mengenkripsi data dengan panjang kunci 128 bit, 192 bit, atau 256 bit. Hal ini membuat AES sangat cocok digunakan dalam sistem yang membutuhkan tingkat keamanan yang tinggi (Hasan Shadzily & Bambang Sujatmiko, 2025).

Namun, menjaga kerahasiaan saja tidak cukup. File yang sudah dienkripsi lalu didekripsi harus tetap lengkap dan tidak ada perubahan isinya. Untuk mengatasi masalah tersebut, penelitian ini menggunakan algoritma SHA-256 sebagai cara memeriksa apakah file tersebut tidak rusak atau dimodifikasi. SHA-256 menghasilkan nilai hash yang unik dan memiliki panjang 256 bit, yang bertindak sebagai sidik jari digital dari sebuah file. Jika ada perubahan sedikitpun pada isi file, maka nilai hash yang dihasilkan akan berubah secara nyata. Sebab itu, sistem akan membuat nilai hash terlebih dahulu sebelum melakukan enkripsi dan membandingkannya kembali setelah proses dekripsi selesai. Jika kedua nilai hash itu sama, maka bisa disimpulkan bahwa file tersebut tidak berubah dan integritasnya masih terjaga.

Pengujian integritas file dilakukan dengan membandingkan nilai hash SHA-256 sebelum proses enkripsi dan setelah proses dekripsi. Proses ini menunjukkan bahwa algoritma yang digunakan tidak merusak isi file dan tidak mengubah data yang ada. Selain itu, penelitian ini juga mengevaluasi tingkat keamanan sistem

yang dihasilkan dari penggabungan algoritma AES-256 dan SHA-256. AES-256 berperan untuk menjaga rahasia data dengan cara mengenkripsi data secara kuat, sedangkan SHA-256 bertugas memastikan data tidak berubah dengan menggunakan metode hash satu arah. Kombinasi kedua metode tersebut diharapkan bisa melindungi secara lengkap dari ancaman akses yang tidak berhak serta perubahan data yang tidak sah.

Selain masalah keamanan, kinerja sistem juga menjadi hal yang diperhatikan dalam penelitian ini. Sistem akan dinilai berdasarkan lama waktu yang dibutuhkan untuk melakukan proses enkripsi, dekripsi, dan memverifikasi integritas file. Pengukuran ini dilakukan agar mengetahui seberapa efisien algoritma itu dalam mengolah file yang ukurannya sudah ditentukan. Jadi, penelitian ini tidak hanya menjelaskan cara algoritma AES-256 dan SHA-256 bekerja untuk menjaga keamanan berkas, tetapi juga menilai seberapa baik sistem tersebut dalam hal kecepatan dan efisiensinya.

Berdasarkan penjelasan tersebut, penelitian ini bertujuan membuat dan menerapkan sistem perlindungan file digital dengan menggunakan AES-256 untuk proses enkripsi dan dekripsi serta SHA-256 untuk memastikan file tidak rusak atau berubah. Sistem yang dihasilkan diharapkan dapat mempertahankan kerahasiaan, menjaga keutuhan, serta memberikan performa yang baik dalam melindungi file digital. Selain itu, penelitian ini juga menganalisis performa waktu proses enkripsi dan dekripsi terhadap berbagai ukuran file.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang yang sudah dijelaskan sebelumnya, berikut beberapa permasalahan yang akan dibahas dalam penelitian ini.

1. Bagaimana proses algoritma AES-256 digunakan untuk mengenkripsi dan mendekripsi file agar data tetap aman dan rahasia?
2. Bagaimana cara menggunakan algoritma SHA-256 untuk memeriksa apakah file tetap utuh setelah mengalami proses enkripsi dan dekripsi?
3. Bagaimana hasil uji integritas file dilakukan dengan membandingkan nilai hash SHA-256 sebelum dan sesudah proses enkripsi serta dekripsi?
4. Bagaimana tingkat keamanan sistem yang dihasilkan dari penggunaan kedua algoritma AES-256 dan SHA-256 dalam melindungi file dari perubahan yang tidak sah?
5. Bagaimana performa sistem diukur berdasarkan waktu yang dibutuhkan untuk proses enkripsi, dekripsi, dan validasi integritas file?

1.3 Batasan Masalah

Agar penelitian ini lebih fokus dan tidak menyimpang dari tujuan yang telah ditetapkan, maka diperlukan batasan masalah yang mengatur cakupan penelitian ini sebagai berikut:

1. Penelitian ini hanya fokus pada cara mengamankan file digital dengan menggunakan algoritma kriptografi.
2. Algoritma yang digunakan dalam penelitian ini adalah AES-256 untuk proses mengenkripsi dan mendekripsi file, serta SHA-256 untuk memastikan integritas file tetap terjaga.

3. Jenis file yang diuji hanya terbatas pada dokumen seperti (txt, pdf, atau docx) dengan ukuran tertentu.
4. Proses mengenkripsi dan mendekripsi dilakukan secara offline, tanpa melibatkan pengiriman data melalui jaringan.
5. Penelitian ini tidak membahas tentang cara mengelola kunci atau sistem pertukaran kunci secara aman.

1.4 Tujuan Penelitian

Berdasarkan perumusan masalah yang sudah ditetapkan, tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

1. Menggunakan algoritma AES-256 untuk mengenkripsi dan mendekripsi file agar data tetap aman dan rahasia.
2. Menerapkan algoritma SHA-256 sebagai cara untuk memeriksa keaslian file, sehingga bisa dipastikan data tidak berubah secara tidak sah.
3. Menguji apakah pemeriksaan keaslian file berjalan dengan benar dengan membandingkan nilai hash sebelum dan setelah proses enkripsi serta dekripsi.
4. Melihat bagaimana sistem bekerja berdasarkan waktu yang dibutuhkan untuk mengenkripsi, mendekripsi, dan memeriksa keaslian file.
5. Menilai seberapa aman sistem yang dibuat dengan menggabungkan algoritma AES-256 dan SHA-256 dalam melindungi file digital.

1.5 Manfaat Penelitian

Penelitian ini diharapkan bisa memberikan pengetahuan tambahan serta pemahaman lebih dalam mengenai bidang keamanan informasi, terutama terkait dengan penggunaan algoritma kriptografi AES-256 untuk mengenkripsi dan

mendekripsi file, serta algoritma SHA-256 untuk memvalidasi integritas file. Selain itu, penelitian ini juga bisa digunakan sebagai bahan referensi untuk penelitian lainnya yang membahas tentang perlindungan data digital.

Secara praktis, penelitian ini diharapkan memberikan gambaran serta contoh penerapan sistem keamanan file digital yang mudah digunakan namun tetap efektif. Sistem yang telah dibuat bisa dijadikan solusi pertama untuk melindungi file dari akses yang tidak diperbolehkan serta mendeteksi perubahan data. Dengan demikian, sistem ini dapat membantu orang pribadi maupun kelompok organisasi dalam menjaga keamanan informasi digital.

Bagi peneliti, penelitian ini menjadi cara untuk menerapkan teori kriptografi yang telah dipelajari ke dalam bentuk sistem nyata. Selain itu, penelitian ini juga bisa membantu peneliti meningkatkan kemampuannya dalam menganalisis, merencanakan, dan membangun sistem keamanan informasi.

BAB II
LANDASAN TEORI

2.1 Keamanan Informasi

2.1.1. Definisi Keamanan Informasi

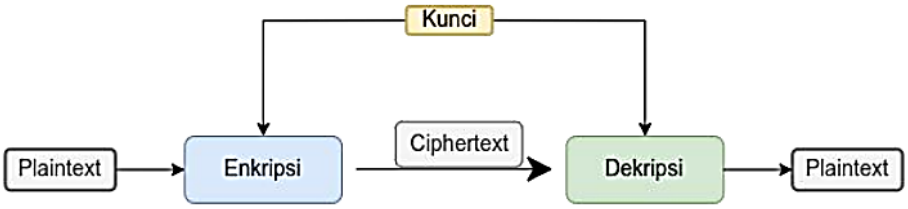
Perlindungan informasi adalah cara untuk menjaga keamanan data dan informasi dari berbagai risiko yang bisa terjadi, baik disengaja maupun tidak. Khususnya saat penggunaan Teknologi Informasi dan Komunikasi (TIK) semakin berkembang, ancaman ini bisa mengganggu kerahasiaan, keutuhan, serta kelancaran layanan. Oleh karena itu, perlindungan informasi semakin penting, karena dampak dari kebocoran atau gangguan informasi bisa sangat serius bagi suatu organisasi (Herman et al., 2021).

2.2 Kriptografi

Istilah kriptografi berasal dari bahasa Yunani, yaitu "cryptos" yang berarti "rahasia" atau "tersembunyi" dan "graphien" yang berarti "tulisan" atau "menulis". Jadi, kriptografi artinya "tulisan tersembunyi" atau "tulisan rahasia". Secara umum, kriptografi adalah ilmu yang mempelajari cara melindungi informasi. Perlindungan ini dilakukan dengan mengubah informasi tersebut menggunakan kunci khusus. Kriptografi adalah cabang ilmu matematika yang berkaitan dengan aspek keamanan informasi, seperti integritas data, identitas entitas, dan keaslian data. Dalam kriptografi terdapat dua proses utama dalam penyandian yaitu enkripsi dan dekripsi. Enkripsi dilakukan saat pengiriman pesan atau informasi dengan cara mengubah data asli menjadi bentuk kode yang membuatnya menjadi data rahasia. Sementara itu, dekripsi dilakukan saat penerima menerima data

tersebut dengan cara mengubah kembali kode rahasia tersebut menjadi bentuk data asli yang mudah dipahami(Fathurrozi, 2021).

Algoritma kriptografi dibagi menjadi dua jenis utama, yaitu algoritma simetris dan algoritma asimetris. Pada algoritma simetris, proses enkripsi dan dekripsi menggunakan satu kunci yang sama, sehingga prosesnya lebih cepat dibandingkan algoritma asimetris. Salah satu keuntungan utama algoritma simetris adalah kemampuannya dalam memproses data secara efisien, terutama untuk data yang besar seperti database akademik (Jefry G G Saragih, 2025).



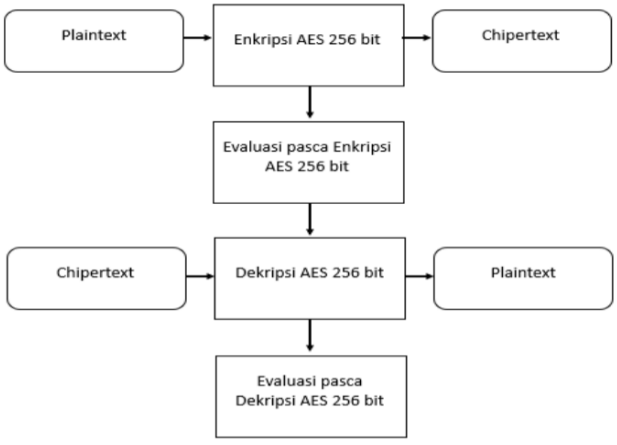
Gambar 2. 1 Algoritma Simetris

2.3 Algoritma AES 256 (Advanced Encryption Standard)

AES (Advanced Encryption Standard) adalah algoritma kriptografi yang menggunakan metode cipher blok dengan ukuran blok tetap sebesar 128 bit. Standar ini dibuat sebagai pengganti DES (Data Encryption Standard) karena lebih aman dan lebih kuat menghadapi berbagai jenis serangan. Proses enkripsi dan dekripsi pada AES terdiri dari beberapa putaran berturut-turut, yang mencakup tahapan seperti substitusi byte, pergeseran baris, pencampuran kolom, serta penerapan operasi XOR dengan kunci enkripsi (Dwi Insani & Dwi Anggraeni, 2025).

Algoritma AES mendukung beberapa ukuran kunci yang berbeda. Algoritma AES terdiri dari tiga jenis ukuran kunci, yaitu AES-128, AES-192, dan AES-256. Perbedaan antara masing-masing jenis ini terletak pada ukuran kunci dan ukuran blok yang digunakan. Perbedaan ukuran kunci ini memengaruhi jumlah proses yang dilakukan saat melakukan enkripsi dan dekripsi. Dengan kata lain, perbedaan terdapat pada jumlah putaran atau round yang digunakan dalam proses enkripsi dan dekripsi (Fathurrozi, 2021).

Penelitian ini menggunakan metode enkripsi dengan Advance Encryption Standard (AES) 256 bit sebagai cara untuk melindungi file. AES memiliki tiga variasi, yaitu 128 bit, 192 bit, dan 256 bit. Pemilihan AES 256 bit diharapkan memberikan perlindungan yang lebih baik dibandingkan dengan AES 128 bit dan AES 192 bit. AES 256 bit merupakan tingkatan tertinggi dalam sistem AES, dan dianggap paling aman. Jika AES 128 bit memiliki 10 putaran enkripsi, AES 192 bit memiliki 12 putaran, sedangkan AES 256 bit memiliki 14 putaran enkripsi. Setelah proses enkripsi selesai, file sampel tersebut dievaluasi dari beberapa aspek. Selain itu, file hasil enkripsi juga diuji proses dekripsinya sebelum dilakukan evaluasi (Indrayani et al., 2025).

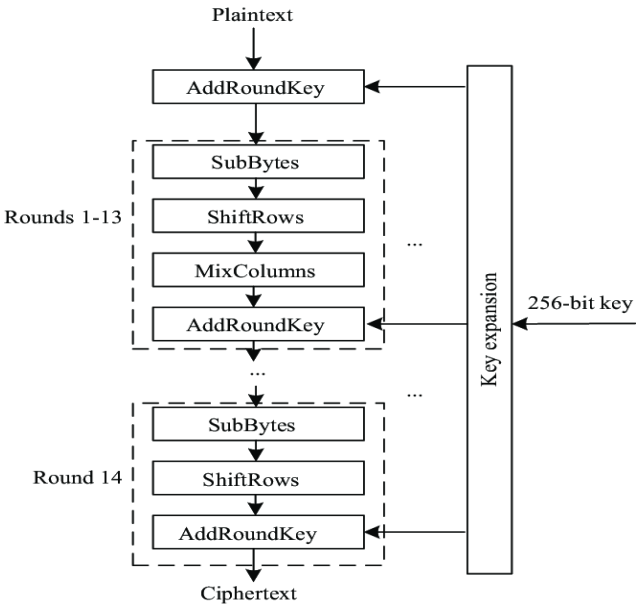


Gambar 2. 2 Alur Enkripsi dan Dekripsi

Dua dasar proses dalam penelitian ini adalah enkripsi dan dekripsi yang menggunakan metode kriptografi simetris. Kunci yang digunakan untuk melakukan proses enkripsi sama dengan kunci yang digunakan untuk melakukan proses dekripsi.

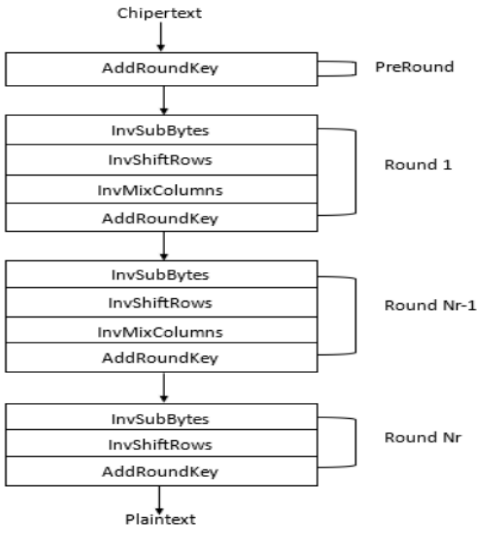
2.4 Enkripsi dan Dekripsi File Menggunakan AES-256

Enkripsi AES 256 bit melakukan proses penyandian selama 14 putaran yang terdiri dari beberapa tahapan, yaitu Addroundkey, Bytesub, Shiftrows, dan Mixcolumn. Sebelum memasuki putaran pertama, password dan plaintext terlebih dahulu diolah dalam proses penjadwalan kunci AES yang disebut sebagai tahap PreRound. Setelah selesai dengan penjadwalan kunci, proses enkripsi dilanjutkan hingga putaran ke-1 sampai putaran ke-13 dengan tahapan SubBytes, ShiftRows, MixColumn, dan AddRoundKey. Dalam putaran terakhir, yaitu putaran ke-14, tidak dilakukan proses MixColumn lagi, tetapi hanya melalui tahapan SubBytes, ShiftRows, dan AddRoundKey. Hasil akhir dari seluruh proses enkripsi disebut ciphertext (Indrayani et al., 2025).



Gambar 2. 3 Proses Enkripsi AES 256

Pada proses dekripsi menggunakan metode AES 256 bit, tahapan dekripsi berlangsung secara terbalik dari proses enkripsi. Chipertext diproses selama 14 putaran untuk kembali mendapatkan plaintext. Password dan chipertext diproses terlebih dahulu dalam tahap PreRound, kemudian diikuti dengan putaran 1 sampai putaran 13 dengan tahapan bernama InvSubBytes, InvShiftRows, InvMixColumns, dan AddRoundKey. Pada putaran terakhir, yaitu putaran ke-14, hanya melalui tahapan InvSubBytes, InvShiftRows, dan AddRoundKey tanpa InvMixColumns. Hasil akhir dari seluruh proses dekripsi adalah plaintext yang sama seperti sebelumnya saat proses enkripsi (Indrayani et al., 2025).



Gambar 2. 4 Proses Dekripsi AES 256

2.5 Algoritma SHA-256 (Secure Hash Algorithm)

SHA-256 (Secure Hash Algorithm 256) adalah algoritma hash kriptografi yang mengubah data yang dimasukkan menjadi nilai hash unik dengan ukuran 256 bit. Dalam sistem ini, SHA-256 digunakan untuk memberi identitas unik pada setiap file arsip yang diunggah, sehingga perubahan kecil pada isi file akan menghasilkan nilai hash yang berbeda dan dapat terdeteksi sebagai tanda adanya

perubahan. Pengujian algoritma SHA-256 dilakukan untuk mengevaluasi kemampuannya dalam menjaga keaslian arsip melalui efek avalanche. Efek avalanche didefinisikan sebagai persentase jumlah bit yang berubah pada nilai hash karena adanya perubahan kecil pada data masukan dibandingkan dengan total jumlah bit keseluruhan (Muhsin & Wiria Nugraha, 2025).

Integritas data dijaga dengan menggunakan algoritma SHA-256 yang membuat kode hash unik untuk setiap dokumen. Jika ada perubahan kecil pada file, maka kode hashnya akan berbeda, sehingga bisa memastikan apakah dokumen asli atau telah dimodifikasi, serta mengenali adanya kecurangan dalam data. Cara kerja ini menjaga keamanan dan kepercayaan terhadap arsip digital. Keberhasilan implementasi sistem tidak hanya ditentukan oleh aspek teknis, tetapi juga oleh tingkat penerimaan pengguna (Muhsin & Wiria Nugraha, 2025).

2.6 Fungsi Hash Kriptografis

Fungsi hash kriptografis adalah algoritma yang mengubah data masukan berukuran apa saja menjadi nilai hash tetap, secara pasti. Artinya, jika data yang dimasukkan sama, maka hasil hash yang dihasilkan juga akan sama. Beberapa ciri penting dari fungsi hash kriptografis adalah sifat one-way, yaitu tidak bisa kembali ke data asli dari hash; collision-resistant, yaitu sangat sulit menemukan dua data berbeda yang menghasilkan hash yang sama; dan avalanche effect, artinya perubahan kecil pada data masukan akan menghasilkan perubahan besar pada hash yang dihasilkan. Algoritma SHA-256 adalah bagian dari keluarga fungsi hash SHA-2 yang menghasilkan nilai hash sepanjang 256 bit (atau 32

byte), sehingga menjadi cara yang kuat dan sering digunakan untuk memastikan integritas data dalam sistem keamanan informasi (Sinaga et al., 2024).

2.7 Python

Salah satu bahasa pemrograman yang saat ini mulai banyak digunakan untuk pengembangan berbagai jenis aplikasi berbasis desktop, web, dan mobile adalah bahasa pemrograman Python. Popularitas penggunaan Python membuatnya menjadi bahasa pemrograman yang mulai banyak dipelajari oleh mahasiswa, terutama mahasiswa di kampus yang berbasis IT, untuk menyelesaikan tugas kuliah, tugas akhir, maupun tugas penelitian(Romzi & Kurniawan, 2020).

Python menjadi salah satu bahasa pemrograman yang direkomendasikan bagi pemula. Sintaksnya mudah dipahami, struktur programnya jelas, dan memiliki komunitas yang luas, sehingga belajar bahasa ini terasa nyaman dan efektif. Berbeda dengan bahasa seperti C++ atau Java yang membutuhkan pemahaman mendalam mengenai konsep teknis seperti pengelolaan memori atau struktur kelas, Python memungkinkan pemula fokus pada logika dan konsep dasar pemrograman tanpa terganggu oleh kesulitan teknis lainnya. Pada tahun 2000, versi Python 2.0 dirilis, yang membawa beberapa fitur baru seperti sistem pengumpulan sampah berbasis deteksi siklus. Lalu, pada tahun 2008, versi Python 3.0 dilepaskan sebagai versi besar, tetapi tidak sepenuhnya kompatibel dengan versi sebelumnya (Python 2.x). Meskipun hal ini menyulitkan sebagian pengembang, Python 3.0 membawa banyak perbaikan agar tetap relevan sebagai bahasa pemrograman modern.

Python merupakan bahasa yang fleksibel dan bisa digunakan dalam berbagai paradigma pemrograman, seperti pemrograman prosedural dan berorientasi objek.

Selain itu, Python memiliki pustaka standar yang lengkap, sehingga pengguna dapat memanfaatkan berbagai fungsi dengan mudah tanpa perlu menulis kode dari awal. Kemudahan ini membuat Python tidak hanya cocok bagi pemula, tetapi juga dapat digunakan dalam proyek-proyek berskala besar.

2.8 Visual Studio Code

Salah satu IDE (Integrated Development Environment) untuk memprogram bahasa Python adalah VSCode (Visual Studio Code), yang bisa dijalankan di berbagai sistem operasi seperti Windows, Linux, dan Mac. Tulisan ini bertujuan untuk membantu pembelajaran dasar pemrograman Python menggunakan VSCode, mulai dari instalasi Python, instalasi VSCode di Windows, serta penggunaan Python pada VSCode berdasarkan algoritma pemrograman. Metode yang digunakan adalah studi pustaka dan eksperimen. Studi pustaka dilakukan untuk memahami konsep-konsep dasar pemrograman yang terkandung dalam algoritma, yaitu algoritma runtunan, algoritma percabangan, dan algoritma perulangan. Adapun eksperimen dilakukan untuk mencoba kode algoritma tersebut dalam pemrograman python yang diimplementasikan menggunakan VSCode (Romzi & Kurniawan, 2020).

2.9 Penelitian Terdahulu

Dalam mendukung topik penelitian ini, penelitian terdahulu digunakan sebagai acuan dan pembanding untuk menganalisis hasil serta kesimpulan yang telah ada, sehingga penelitian ini memiliki perbedaan dan nilai kebaruan.

Tabel 2. 1 Penelitian Terdahulu

No	Peneliti dan Tahun	Judul Penelitian	Hasil Penelitian

No	Peneliti dan Tahun	Judul Penelitian	Hasil Penelitian
1	Dian Sri Purwanti, Muhammad Fadli, Muhammad Surono, Erliyan Redy Susanto (2025)	Perancangan Penerapan Algoritma Kriptografi AES 256 Untuk Keamanan Database Aplikasi Manajemen Siswa	Hasil penelitian menunjukkan bahwa penerapan algoritma AES 256 pada basis data aplikasi manajemen siswa berbasis web di SMKN XYZ Bandar Lampung berhasil meningkatkan keamanan data siswa. Data yang disimpan dalam basis data telah dienkripsi sehingga tidak dapat dibaca oleh pihak yang tidak berwenang.
2	Rini Indrayani, Pramudhita Ferdiansyah, Muhammad Koprawi (2024)	Analisis Penggunaan Kriptografi Metode AES 256 Bit pada Pengamanan File dengan Berbagai Format	Hasil penelitian menunjukkan bahwa penerapan algoritma AES 256 bit menyebabkan perubahan yang signifikan pada objek kriptografi. Evaluasi metadata dan visual memperlihatkan bahwa file hasil enkripsi mengalami perubahan menyeluruh sehingga tidak dapat dikenali

No	Peneliti dan Tahun	Judul Penelitian	Hasil Penelitian
			atau dibaca secara langsung.
3	Ahmad Fathurrozi, Selviyani (2021)	Penerapan Algoritma Advanced Encryption Standard (AES256) Dengan Mode CBC DanSecure Hash Algorithm (SHA-256) Untuk Pengamanan Data File	Hasil penelitian menunjukkan bahwa penerapan algoritma Advanced Encryption Standard (AES) dan Secure Hash Algorithm (SHA) berhasil meningkatkan keamanan data Universitas Bhayangkara Jakarta Raya. Data yang dienkripsi tidak dapat dibaca oleh pihak yang tidak berwenang, sementara proses dekripsi mampu mengembalikan data ke bentuk semula tanpa perubahan isi.
4	Herman, Robby Wijaya, Kenner Farandi, Satriya Miharja, Wilson (2021)	Implementasi Algoritma AES-128 Dan SHA-256 Dalam Perancangan Aplikasi Pengamanan File	Hasil penelitian menunjukkan bahwa penerapan kombinasi algoritma AES-128 dan SHA-256 berhasil meningkatkan keamanan file dokumen dari ancaman akses

No	Peneliti dan Tahun	Judul Penelitian	Hasil Penelitian
		Dokumen	tidak sah. File yang telah dienkripsi tidak dapat dibaca tanpa proses dekripsi yang sesuai, sehingga lebih aman dibandingkan dengan proteksi kata sandi saja.
5	Pazrian Nurul Latip (2025)	IMPLEMENTASI ALGORITMA KRIPTOGRAFI AES DALAM PENGAMANAN FILE TEKS	Hasil penelitian menunjukkan bahwa penerapan algoritma Advanced Encryption Standard (AES) pada enkripsi file teks mampu memberikan tingkat keamanan yang lebih tinggi serta proses yang lebih efisien dibandingkan dengan algoritma Vigenère. Pengujian pada prototipe aplikasi menunjukkan bahwa AES memiliki kecepatan enkripsi dan dekripsi yang lebih baik serta lebih tahan terhadap serangan kriptografis, sehingga lebih

No	Peneliti dan Tahun	Judul Penelitian	Hasil Penelitian
			efektif digunakan untuk melindungi file teks yang berisi informasi sensitif.

BAB III
ANALISA DAN PERANCANGAN SISTEM

3.1 Analisis Permasalahan

Dokumen merupakan salah satu jenis file data yang sangat penting karena berfungsi sebagai sumber informasi yang diperlukan oleh Instansi atau perusahaan di era digital yang terus berubah. Menjaga kerahasiaan data menjadi sangat krusial untuk memastikan bahwa informasi yang tersimpan tidak dapat diakses atau dibaca oleh pihak yang tidak berwenang. Dengan semakin tingginya ancaman keamanan siber, seperti pencurian data dan peretasan, data yang tidak dilindungi bisa dimanfaatkan dengan cara yang tidak benar, sehingga bisa merugikan bukan hanya orang pribadi, tetapi juga nama baik lembaga tersebut. Oleh karena itu, institusi harus menggunakan teknologi yang mampu memberikan perlindungan sebaik mungkin terhadap data yang dimiliki. Keamanan data bukan hanya tanggung jawab IT, tetapi juga melibatkan Seluruh bagian dalam organisasi (Maryo & Widagdo, 2025).

Salah satu cara yang bisa dilakukan adalah penggunaan teknologi Kriptografi. Kriptografi adalah seni dan ilmu untuk menjaga kerahasiaan informasi melalui metode enkripsi yang kompleks. Dengan kriptografi, pesan asli (plaintext) dapat diubah menjadi pesan terenkripsi (ciphertext) yang aman, dan kemudian dapat dikembalikan ke bentuk aslinya melalui proses dekripsi. Ini sangat membantu dalam melindungi data sensitif dari akses yang tidak sah, sehingga memberikan rasa aman bagi pengguna. Implementasi kriptografi dapat membantu meningkatkan kepercayaan antara pihak pihak yang terlibat (Maryo & Widagdo, 2025).

Permasalahan tersebut memerlukan pembuatan sebuah aplikasi untuk melindungi informasi dokumen dengan menerapkan algoritma kriptografi. Proses penyamaran plaintext ke ciphertext disebut enkripsi, sedangkan proses mengembalikan ciphertext menjadi teks biasa kembali disebut dekripsi. Pada penelitian ini akan menggabungkan dua algoritma kriptografi, yaitu Advanced Encryption Standard (AES) dengan panjang kunci 256 bit dan algoritma hash aman (SHA) dengan nilai 256 bit yang digunakan pada aplikasi perlindungan informasi dokumen yang akan dibangun. Algoritma AES-256 akan mengenkripsi file dokumen dan algoritma SHA-256 akan digunakan untuk memastikan keamanan data, mengunci informasi catatan penting mengenai file tersebut agar tidak bisa diubah (Herman et al., 2021).

Oleh karena itu, penelitian ini menggunakan algoritma kriptografi AES-256 untuk mengenkripsi dan mendekripsi file, serta algoritma SHA-256 untuk memverifikasi integritas file dengan membandingkan nilai hash sebelum dan setelah proses enkripsi atau dekripsi.

Dalam penelitian ini, uji coba sistem dilakukan dengan menggunakan berbagai file sampel yang memiliki format dan ukuran berbeda. Menggunakan berbagai jenis format file bertujuan untuk memeriksa sejauh mana algoritma AES-256 dan SHA-256 mampu bekerja secara konsisten dalam mengolah berbagai jenis data digital. Dataset yang digunakan didapatkan dari platform Kaggle karena platform tersebut menyediakan data publik yang mudah dibuka dan sering digunakan dalam berbagai penelitian serta pengujian sistem.

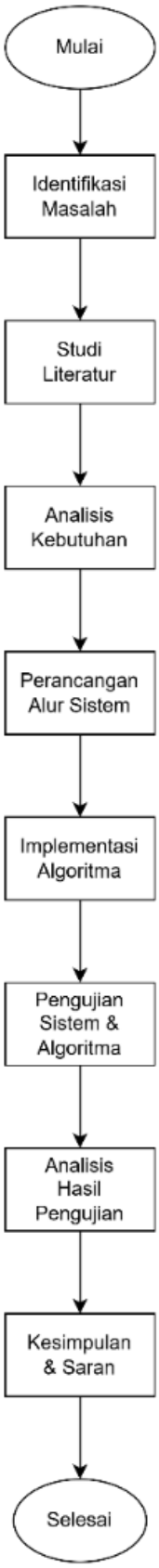
Tabel 3. 1 Dataset Sampel Pengujian Sistem

Sumber: Kaggle

No	Nama File	Format	Ukuran File (KB)	Kategori Data
1	bbc-news-data.txt	TXT	4.962	Teks Berita
2	dokumen 1.doc	DOCX	16.656	Dokumen Teks
3	file 1.pdf	PDF	7.886	Dokumen Laporan
4	emails.txt	TXT	30.511	Pesan Email
5	File R.pdf	PDF	5.037	Dokumen Laporan
…	……	…	…	……

Dataset tersebut menggunakan berbagai ukuran dan format file agar bisa melihat pengaruh karakteristik data terhadap waktu yang dibutuhkan untuk proses enkripsi, dekripsi, dan menghitung nilai hash, sehingga hasil pengujian yang diperoleh lebih representatif.

Untuk memudahkan pemahaman tentang langkah-langkah penelitian yang dilakukan, diperlukan penjelasan mengenai proses kerja penelitian secara teratur dan terstruktur. Alur penelitian ini menjelaskan langkah-langkah penelitian secara berurutan, mulai dari mengenali masalah, mengumpulkan data, merancang sistem, menerapkan algoritma AES-256 dan SHA-256, hingga tahap pengujian dan menganalisis hasilnya. Oleh karena itu, alur penelitian ditampilkan dalam bentuk flowchart agar proses penelitian menjadi lebih mudah dipahami dan terstruktur dengan baik.



Gambar 3. 1 Flowchart Alur Penelitian

3.2 Algoritma Sistem

Algoritma sistem dalam penelitian ini dibuat untuk melindungi file dokumen dengan menggunakan metode enkripsi-dekripsi berupa AES-256 dan SHA-256 untuk memastikan file tetap utuh dan tidak rusak. Tahapan algoritma sistem dijelaskan sebagai berikut:

1. Pemilihan File

   Pengguna memilih file dokumen yang ingin dienkripsi, seperti file berformat .txt, .pdf, atau .docx, menggunakan aplikasi tersebut.

2. Input Kunci (Password)

   Pengguna memasukkan password yang berfungsi sebagai kunci rahasia untuk mengenkripsi dan mendekripsi data dengan menggunakan algoritma AES-256.

3. Pembangkitan Nilai Hash Awal (SHA-256)

   Sebelum proses enkripsi dimulai, sistem membuat nilai hash SHA-256 dari file asli sebagai acuan untuk memeriksa apakah data tetap utuh dan tidak berubah.

4. Proses Enkripsi File (AES-256)

   File asli dalam bentuk teks biasa dienkripsi dengan algoritma AES-256 yang memiliki 14 putaran, terdiri dari beberapa tahapan yaitu SubBytes, ShiftRows, MixColumns, dan AddRoundKey, sehingga menghasilkan file yang sudah dienkripsi.

5. Penyimpanan File Terenkripsi

File hasil enkripsi disimpan di dalam perangkat penyimpanan dan tidak bisa dibuka atau dibaca kecuali melalui proses dekripsi dengan menggunakan kunci yang tepat.

6. Proses Dekripsi File (AES-256)

   Jika pengguna ingin membuka kembali file tersebut, sistem akan melakukan proses dekripsi dengan menggunakan kunci yang sama untuk mengubah ciphertext kembali menjadi file asli dalam bentuk plaintext.

7. Pembangkitan Nilai Hash Akhir (SHA-256)

   Setelah proses dekripsi selesai, sistem kembali membuat nilai hash SHA-256 dari file yang sudah didekripsi.

8. Validasi Integritas File

   Sistem memeriksa nilai hash sebelum file dienkripsi dan setelah file didekripsi. Jika hasil hash sama, berarti file tersebut tidak rusak dan tidak mengalami perubahan. Jika berbeda, berarti file telah mengalami modifikasi.

9. Output Sistem

   Sistem menampilkan hasil berupa file yang sudah terenkripsi atau didekripsi serta status validasi integritas file, apakah valid atau tidak valid.

3.3 Pemodelan dan Perancangan Sistem

Pemodelan dan perancangan sistem dalam penelitian ini bertujuan untuk menunjukkan bagaimana proses kerja sistem pengamanan file berjalan secara teratur dan mudah dimengerti. Sistem yang dibuat adalah aplikasi sederhana (prototype) yang beroperasi tanpa koneksi internet dan menggunakan bahasa pemrograman Python.

1. Perancangan Alur Sistem

   Alur sistem dimulai dengan pengguna yang memasukkan file beserta kunci enkripsi. Sistem kemudian memproses file dengan menggunakan algoritma AES-256 untuk mengenkripsi dan mendekripsi data, serta memakai algoritma SHA-256 untuk membuat dan membandingkan nilai hash sebagai cara memastikan file tidak rusak atau diubah.

2. Perancangan Proses Enkripsi dan Dekripsi

   Proses mengenkripsi dan mendekripsi dibuat dengan cara menggunakan kriptografi simetris, di mana kunci yang digunakan untuk mengenkripsi sama dengan kunci yang digunakan untuk mendekripsi. AES-256 dipilih karena menawarkan perlindungan yang sangat aman dengan proses enkripsi yang terdiri dari 14 putaran.

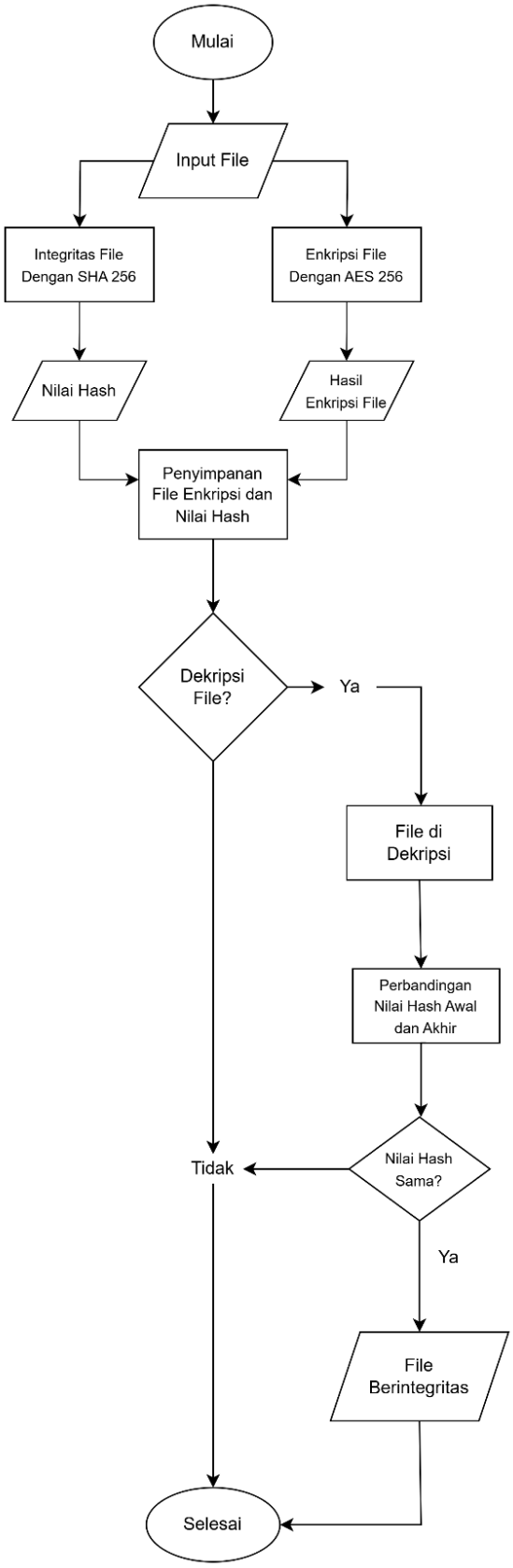
3. Perancangan Validasi Integritas File

   Validasi integritas file dilakukan dengan membandingkan nilai hash SHA-256 dari file sebelum dan setelah proses enkripsi serta dekripsi. Perbandingan ini dilakukan agar isi file tetap sama dan tidak berubah.

4. Perancangan Sistem Secara Umum

   Sistem dibuat dengan antarmuka yang mudah dipahami agar pengguna bisa menggunakannya dengan nyaman. Fokus utama dalam merancang sistem adalah pada fungsi keamanan file, bukan pada tampilan antarmuka. Sistem ini berfungsi sebagai sarana untuk mengevaluasi seberapa efektif algoritma AES-256 dan SHA-256 digunakan dalam menjaga kerahasiaan serta keutuhan file digital.

Untuk memudahkan pemahaman mengenai cara kerja sistem yang dibuat, disajikan flowchart alur sistem yang menunjukkan langkah-langkah proses yang dilakukan aplikasi, mulai dari input file oleh pengguna, proses perhitungan hash untuk memvalidasi integritas file, proses enkripsi dan dekripsi menggunakan algoritma AES-256, hingga proses verifikasi keutuhan file berdasarkan hasil perbandingan nilai hash. Flowchart ini menjelaskan seluruh proses kerja sistem secara terperinci, sehingga memudahkan pemahaman tentang bagaimana sistem tersebut berfungsi.



Gambar 3. 2 Flowchart Alur Sistem

Flowchart sistem di atas menjelaskan cara kerja aplikasi secara menyeluruh, sehingga dapat dipahami bahwa setiap tahap saling terkait dan dirancang agar file tetap aman, baik dalam hal kerahasiaannya maupun keutuhan datanya.

3.3.1 Perancangan Sistem

Perancangan sistem merupakan tahapan penting dalam penelitian yang bertujuan untuk menggambarkan bagaimana sistem akan dibangun dan bekerja secara keseluruhan. Pada penelitian ini, sistem dirancang untuk mengamankan file digital dengan menggunakan algoritma AES-256 untuk proses enkripsi dan dekripsi, serta algoritma SHA-256 untuk validasi integritas file. Sistem yang dirancang bersifat sederhana dan berjalan secara offline, dengan fokus utama pada fungsi keamanan data. Proses perancangan sistem dimulai dari analisis kebutuhan pengguna, kemudian dilanjutkan dengan perancangan alur kerja sistem yang menggambarkan langkah-langkah proses mulai dari input file, proses enkripsi, dekripsi, hingga validasi integritas file.

Secara umum, alur kerja sistem adalah sebagai berikut: pengguna mengunggah file yang ingin diamankan, kemudian memasukkan password sebagai kunci enkripsi. Sistem akan menghasilkan nilai hash awal menggunakan SHA-256 sebelum proses enkripsi dilakukan. Selanjutnya, file dienkripsi menggunakan algoritma AES-256 dan disimpan dalam bentuk terenkripsi. Untuk membuka kembali file, pengguna melakukan proses dekripsi dengan menggunakan password yang sama. Setelah proses dekripsi selesai, sistem akan menghasilkan nilai hash akhir dan membandingkannya dengan hash awal untuk memastikan bahwa file tidak mengalami perubahan. Perancangan sistem ini juga memperhatikan kemudahan penggunaan (user friendly), sehingga pengguna dapat

dengan mudah melakukan proses enkripsi dan dekripsi tanpa memerlukan pemahaman teknis yang mendalam.

3.3.2 Mockup Sistem

Mockup sistem merupakan gambaran awal tampilan antarmuka (user interface) yang dirancang untuk menunjukkan bagaimana sistem akan terlihat dan digunakan oleh pengguna. Mockup ini bertujuan untuk memberikan visualisasi sebelum sistem benar-benar diimplementasikan. Pada penelitian ini, mockup sistem dirancang dalam bentuk tampilan sederhana yang terdiri dari beberapa komponen utama, yaitu:

1. Halaman Utama

   Menampilkan judul aplikasi serta pilihan fitur yang tersedia, seperti enkripsi dan dekripsi file.

2. Form Upload File

   Pengguna dapat memilih file yang akan diproses, baik untuk enkripsi maupun dekripsi. File yang didukung meliputi format .txt, .pdf, dan .docx.

3. Input Password

   Tersedia kolom untuk memasukkan password yang akan digunakan sebagai kunci dalam proses enkripsi dan dekripsi. Password ini bersifat rahasia dan harus sama saat proses dekripsi.

4. Tombol Proses

   1) Tombol Encrypt untuk melakukan enkripsi file
   2) Tombol Decrypt untuk melakukan dekripsi file

5. Tampilan Nilai Hash

   Sistem menampilkan nilai hash SHA-256 dari file, baik sebelum enkripsi maupun setelah dekripsi, sebagai indikator validasi integritas file.

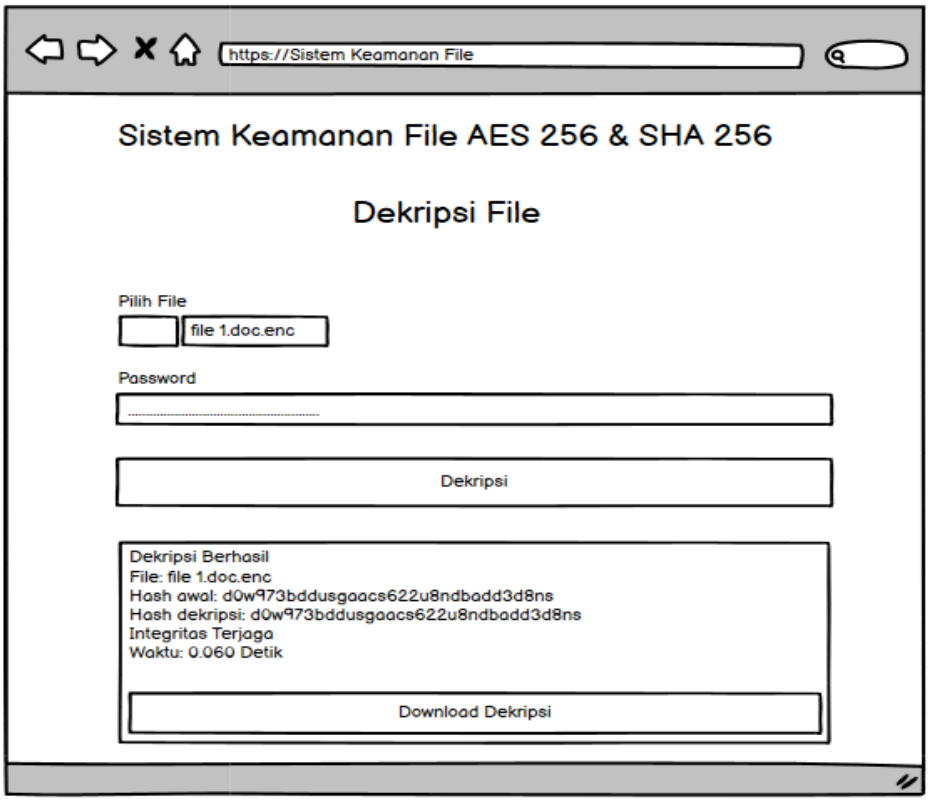
6. Status Validasi

   Menampilkan informasi apakah file dalam kondisi valid (tidak berubah) atau tidak valid (mengalami perubahan).

7. Tombol Download

   Pengguna dapat mengunduh file hasil enkripsi atau dekripsi melalui tombol yang tersedia.

Gambar 3. 3 Halaman Utama Sistem Keamanan File



Gambar 3. 4 Halaman Utama Sistem Keamanan File

Tampilan ini dibuat dengan cara yang sederhana dan ramah pengguna untuk membantu pengguna dalam menjalankan sistem tanpa perlu keahlian teknis yang kompleks. Dengan adanya informasi hasil enkripsi dan dekripsi yang ditampilkan, pengguna juga dapat memahami apakah file yang sedang diproses tetap memiliki integritas yang terjaga atau tidak.

Validasi nilai hash merupakan proses yang dilakukan untuk memastikan keutuhan dan keaslian data setelah melalui proses enkripsi dan dekripsi. Proses ini dilakukan dengan cara membandingkan nilai hash sebelum enkripsi dengan nilai hash setelah dekripsi. Apabila kedua nilai hash tersebut sama, maka dapat disimpulkan bahwa data tidak mengalami perubahan dan integritasnya terjaga.

Sebaliknya, jika nilai hash berbeda, maka data dianggap telah mengalami perubahan atau kerusakan selama proses berlangsung.

Dalam kondisi nilai hash yang berbeda, sistem akan melakukan beberapa tindakan sebagai berikut:

1. Menampilkan notifikasi kesalahan kepada pengguna bahwa integritas file tidak terjaga atau terjadi kegagalan dalam proses dekripsi.
2. Menolak hasil dekripsi sebagai file yang valid, sehingga file tersebut tidak direkomendasikan untuk digunakan lebih lanjut.
3. Mengidentifikasi kemungkinan penyebab kesalahan, seperti kesalahan dalam memasukkan kata sandi, file yang telah mengalami perubahan, atau kerusakan data.
4. Menyarankan pengguna untuk mengulangi proses, baik dengan menggunakan kata sandi yang benar maupun dengan menggunakan file asli yang belum mengalami perubahan.
5. Menjaga keamanan sistem, dengan tidak memproses lebih lanjut file yang terindikasi tidak valid guna mencegah potensi kesalahan atau risiko keamanan.

BAB IV
IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Kebutuhan Sistem

Kebutuhan sistem adalah langkah pertama dalam proses penerapan yang bertujuan untuk mengetahui berbagai komponen yang dibutuhkan agar sistem bisa berjalan dengan lancar. Dalam penelitian ini, sistem membutuhkan komponen seperti perangkat keras, perangkat lunak, serta kebutuhan fungsional dan non-fungsional yang mendukung proses mengenkripsi, mendekripsi, serta memvalidasi integritas file dengan menggunakan algoritma AES-256 dan SHA-256. Dengan melakukan analisis kebutuhan sistem, diharapkan aplikasi yang dikembangkan bisa berjalan dengan baik dan mampu mencapai tujuan penelitian yang sudah ditentukan. Adapun kebutuhan sistem dalam penelitian ini adalah sebagai berikut:

1. Kebutuhan Perangkat Keras (Hardware)

   Berikut merupakan kebutuhan perangkat keras yang digunakan untuk menjalankan sistem yang telah dibangun:

   1) Laptop/PC
   2) Processor minimal Intel i3 atau setara
   3) RAM minimal 4 GB
   4) Media penyimpanan (Harddisk/SSD)

2. Kebutuhan Perangkat Lunak (Software)

   Selain perangkat keras, sistem ini juga memerlukan perangkat lunak sebagai pendukung dalam proses pengembangan dan pengoperasian sistem, yaitu sebagai berikut:

1) Sistem Operasi: Windows 10 atau Windows 11
2) Bahasa Pemrograman: Python
3) IDE: Visual Studio Code
4) Library Python:
    a. pycryptodome (untuk proses enkripsi AES-256)
    b. hashlib (untuk proses hashing SHA-256)
    c. Flask (untuk pengembangan aplikasi berbasis web)
5) Browser (digunakan untuk menjalankan aplikasi berbasis web)

3. Kebutuhan Fungsional Sistem

Kebutuhan fungsional merupakan kebutuhan yang berkaitan langsung dengan fitur atau fungsi utama yang harus dimiliki oleh sistem agar dapat berjalan sesuai dengan tujuan penelitian. Adapun kebutuhan fungsional sistem adalah sebagai berikut:

1) Sistem dapat mengunggah file dengan format .txt, .pdf, dan .docx
2) Sistem dapat melakukan proses enkripsi menggunakan algoritma AES-256
3) Sistem dapat melakukan proses dekripsi menggunakan algoritma AES-256
4) Sistem dapat menghasilkan nilai hash menggunakan algoritma SHA-256
5) Sistem dapat membandingkan nilai hash untuk validasi integritas file
6) Sistem menyediakan fitur untuk mengunduh file hasil enkripsi maupun dekripsi

4. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan kebutuhan yang berkaitan dengan kualitas sistem, seperti kinerja, keamanan, dan kemudahan penggunaan. Adapun kebutuhan non-fungsional dalam sistem ini adalah sebagai berikut:

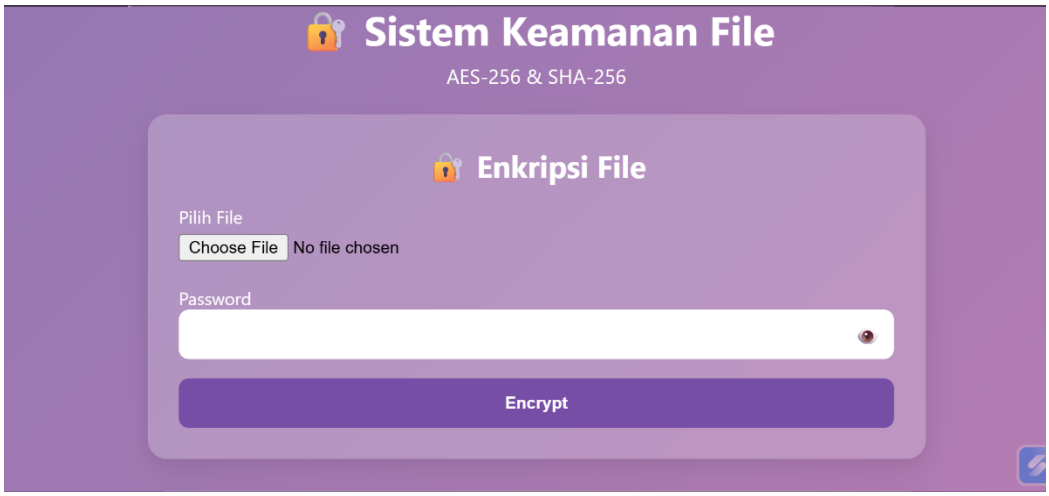
1) Sistem dapat berjalan secara offline tanpa koneksi internet
2) Sistem memiliki waktu proses yang cepat dan efisien
3) Sistem memiliki tampilan antarmuka yang sederhana dan mudah digunakan

4.2 Implementasi Sistem

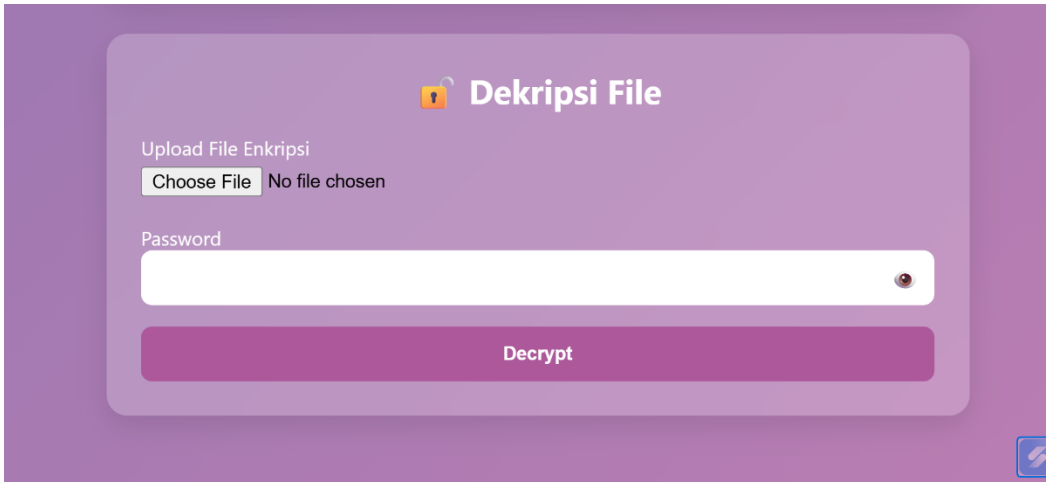
Implementasi sistem merupakan tahap lanjutan dari proses perancangan yang bertujuan untuk merealisasikan sistem yang telah dirancang menjadi sebuah aplikasi yang dapat digunakan. Pada tahap ini, seluruh konsep, alur kerja, serta desain sistem yang telah dibuat pada tahap sebelumnya diimplementasikan ke dalam bentuk program menggunakan bahasa pemrograman Python. Sistem yang dibangun dalam penelitian ini merupakan aplikasi pengamanan file digital yang menggunakan algoritma AES-256 untuk proses enkripsi dan dekripsi, serta algoritma SHA-256 untuk melakukan validasi integritas file.

Implementasi sistem dilakukan secara offline dengan memanfaatkan framework Flask sebagai penghubung antara proses backend dan tampilan antarmuka berbasis web. Dalam proses implementasi, sistem dirancang agar dapat menerima input berupa file dan password dari pengguna, kemudian memproses file tersebut melalui tahapan enkripsi atau dekripsi sesuai dengan pilihan

pengguna. Selain itu, sistem juga secara otomatis menghasilkan nilai hash sebelum dan sesudah proses dilakukan, yang digunakan untuk memastikan bahwa file tidak mengalami perubahan.



Gambar 4. 1 Halaman Utama Sistem



Gambar 4. 2 Halaman Utama Sistem

Gambar di atas memperlihatkan tampilan muka dari sistem yang dikembangkan dalam penelitian ini. Halaman ini adalah tampilan pertama (homepage) yang muncul ketika pengguna membuka aplikasi melalui browser untuk pertama kalinya. Di halaman muka ini, pengguna disuguhkan dengan berbagai elemen utama yang berfungsi sebagai panduan dalam menggunakan

sistem, contohnya judul aplikasi, deskripsi singkat, dan menu atau tombol yang mengarah ke fitur-fitur penting yang tersedia. Fitur-fitur tersebut mencakup proses enkripsi yang berfungsi untuk menjaga keamanan file dan proses dekripsi yang digunakan untuk mengembalikan file ke bentuk semula.

Desain antarmuka pada halaman utama diatur secara sederhana, teratur, dan sistematis agar pengguna dapat dengan mudah memahami alur penggunaan sistem tanpa menghadapi kesulitan. Penempatan elemen visual seperti tombol, form input, dan informasi tambahan juga dirancang secara intuitif yang dapat meningkatkan kenyamanan pengguna dalam berinteraksi dengan sistem. Dengan desain yang ramah pengguna ini, diharapkan bahwa pengguna dapat menjalankan fungsi sistem dengan efisien dan efektif sesuai dengan tujuan penelitian, yaitu untuk melindungi file digital menggunakan algoritma AES-256 dan memastikan integritas data melalui SHA-256.

4.3 Pengujian Sistem

Pengujian sistem merupakan tahap yang dilakukan untuk memastikan bahwa sistem yang telah diimplementasikan dapat berjalan dengan baik sesuai dengan tujuan yang telah ditetapkan. Tahap ini bertujuan untuk menguji fungsi-fungsi yang terdapat dalam sistem, serta memastikan bahwa proses input, proses, dan output berjalan secara benar dan menghasilkan hasil yang sesuai dengan yang diharapkan.

Pada penelitian ini, pengujian sistem dilakukan terhadap aplikasi pengamanan file digital yang menggunakan algoritma AES-256 untuk proses enkripsi dan dekripsi, serta algoritma SHA-256 untuk validasi integritas file. Pengujian dilakukan dengan menggunakan beberapa file sampel dengan format dan ukuran yang

berbeda, seperti file .txt, .pdf, dan .docx, untuk mengetahui kinerja sistem dalam berbagai kondisi.

Proses pengujian dimulai dari tahap pengunggahan file oleh pengguna, kemudian dilanjutkan dengan proses enkripsi menggunakan password yang telah ditentukan. Setelah itu, file yang telah dienkripsi diuji kembali melalui proses dekripsi untuk memastikan bahwa file dapat dikembalikan ke bentuk semula tanpa mengalami perubahan. Selanjutnya, sistem akan membandingkan nilai hash sebelum proses enkripsi dan setelah proses dekripsi untuk memastikan bahwa integritas file tetap terjaga.

Selain itu, pengujian juga dilakukan untuk mengukur performa sistem, yaitu dengan menghitung waktu yang dibutuhkan dalam proses enkripsi, dekripsi, dan validasi integritas file. Hasil pengujian ini digunakan untuk mengetahui tingkat efisiensi sistem dalam mengolah data. Dengan adanya pengujian sistem ini, dapat diketahui apakah sistem yang dibangun telah berfungsi dengan baik, mampu menjaga kerahasiaan data melalui proses enkripsi, serta memastikan keutuhan data melalui validasi hash. Hasil dari pengujian ini akan menjadi dasar dalam menarik kesimpulan mengenai keberhasilan sistem yang telah dikembangkan.

Berikut ini merupakan tampilan isi dari sampel file yang akan digunakan untuk enkripsi dan dekripsi.

United States Senate

COMMITTEE ON
HOMELAND SECURITY AND GOVERNMENTAL AFFAIRS
WASHINGTON, DC 20510–6250

February 28, 2012

The Honorable Mark G. Pearce
Chairman
National Labor Relations Board
1099 14th Street NW
Washington, DC 20570

Dear Mr. Chairman:

As part of the Subcommittee's ongoing oversight of public relations and advertising contracts, we are writing to ask for information regarding the National Labor Relations Board's contracts for the acquisition of public relations, publicity, advertising, communications, or similar services. Please provide a list of all the Board's contracts for these types of services awarded on or after October 1, 2008. Please submit your response using the spreadsheet provided with the following information for each contract:

(1) The contract number;

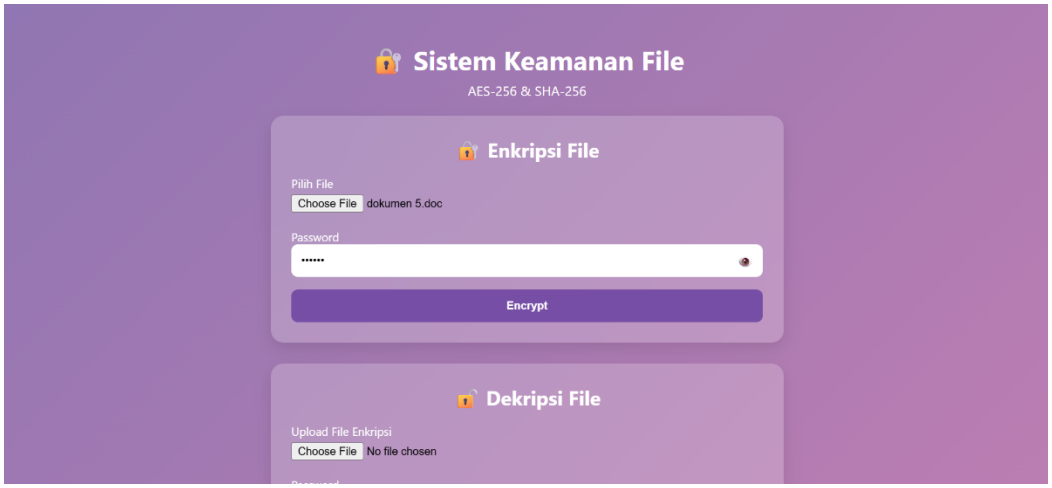
(2) The date of contract award;

Gambar 4. 3 Sampel File

Gambar 4.3 menampilkan contoh file yang akan digunakan dalam penelitian ini sebagai data uji dalam pengujian sistem. Contoh file ini berfungsi sebagai subjek eksperimen yang akan diproses oleh sistem, terutama pada bagian enkripsi dan dekripsi. File yang dipilih mencakup berbagai format, seperti txt, pdf, dan docx, guna menguji kemampuan sistem dalam mengelola berbagai tipe file. Selama tahap enkripsi, contoh file tersebut akan diubah menjadi format terenkripsi dengan menggunakan algoritma AES-256, sementara pada tahap dekripsi, file itu akan dikembalikan ke keadaan semula.

Selain itu, sistem juga akan menghasilkan nilai hash dengan algoritma SHA-256 untuk memeriksa integritas file sebelum dan sesudah proses berlangsung. Penggunaan contoh file ini bertujuan untuk menilai kinerja serta efektivitas algoritma yang diterapkan, baik dalam hal kecepatan proses maupun ketepatan dalam menjaga keaslian data. Dengan begitu, hasil dari pengujian ini dapat menjadi acuan dalam menilai keberhasilan sistem yang telah dibangun.

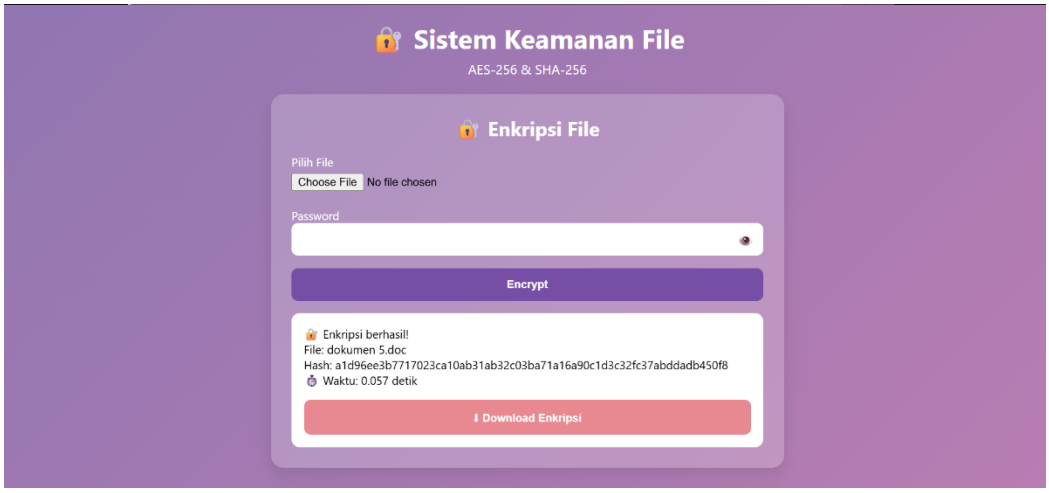
Berikut adalah tampilan sistem untuk mengunggah file dan memasukkan kata sandi.



Gambar 4. 4 Tampilan Upload File dan Input Password

Gambar 4.4 menampilkan tampilan form upload file dan input password yang digunakan oleh pengguna untuk memilih file yang akan diproses. Sistem memiliki fitur yang memungkinkan Anda mengunggah file berformat .txt, .pdf, atau .docx sebagai masukan awal sebelum melakukan proses enkripsi atau dekripsi. Pengguna harus memasukkan password yang rahasia agar file tetap aman saat proses enkripsi dan dekripsi dilakukan.

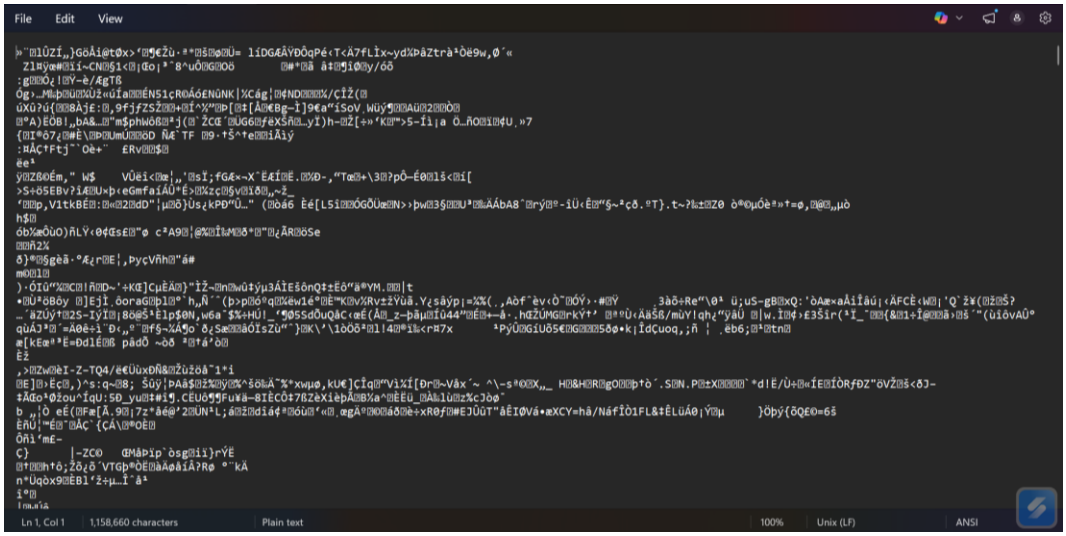
Berikut ini merupakan tampilan halaman proses enkripsi dan tampilan nilai hash.



Gambar 4. 5 Proses Enkripsi dan Tampilan Nilai Hash

Gambar 4.5 menampilkan proses enkripsi yang dijalankan sistem setelah pengguna mengunggah file dan memasukkan kata sandi. Sistem akan mengolah file dengan algoritma AES-256, sehingga file tersebut berubah menjadi bentuk yang terenkripsi dan tidak bisa dibaca langsung. Selain itu, sistem juga menampilkan nilai hash yang dihasilkan dengan menggunakan algoritma SHA-256. Nilai hash ini berfungsi sebagai identitas khusus file dan digunakan untuk memastikan data tidak rusak sebelum dan setelah proses dilakukan.

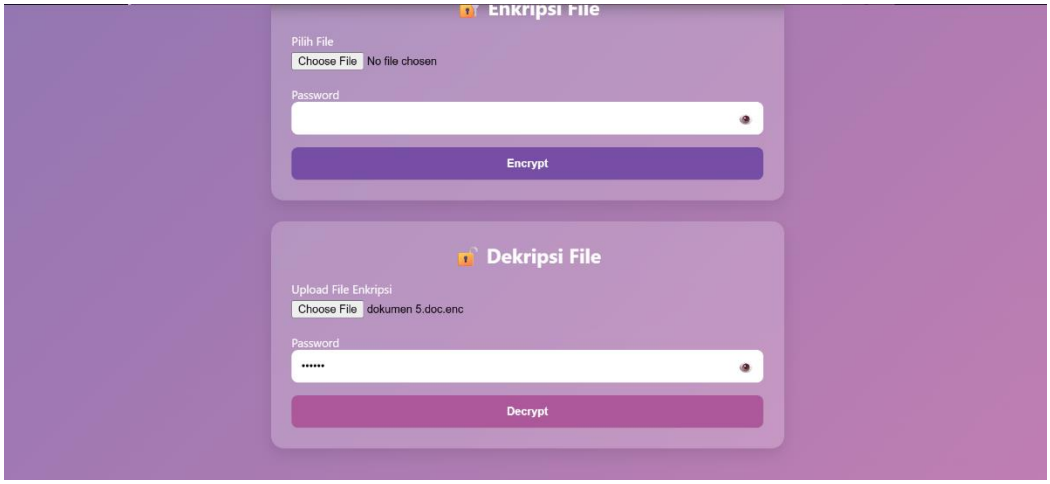
Berikut ini merupakan tampilan file yang telah melalui proses enkripsi pada sistem.



Gambar 4. 6 Hasil Enkripsi

Gambar 4.6 menampilkan hasil dari proses perlindungan file melalui langkah enkripsi yang telah dilakukan oleh sistem. Dalam fase ini, file yang sebelumnya dalam bentuk asli (plaintext) telah berhasil diubah menjadi bentuk terenkripsi (ciphertext) dengan menggunakan algoritma AES-256, sehingga isi file tidak bisa dibaca secara langsung tanpa melalui tahap dekripsi. Hasil enkripsi tersebut kemudian disimpan secara otomatis dalam sistem atau folder penyimpanan yang telah ditentukan. Selain itu, sistem juga menyediakan tombol atau fitur untuk mengunduh file hasil enkripsi tersebut, sehingga pengguna dapat menyimpan file yang telah dilindungi ke dalam perangkat mereka. Dengan fitur ini, pengguna dapat dengan mudah mengakses kembali file terenkripsi untuk keperluan penyimpanan atau proses dekripsi di masa mendatang. Hasil ini menunjukkan bahwa sistem telah berhasil melaksanakan fungsi perlindungan data sesuai dengan tujuan penelitian, yaitu menjaga kerahasiaan isi file dari akses yang tidak sah.

Berikut adalah tampilan sistem untuk mengunggah file dan memasukkan kata sandi.

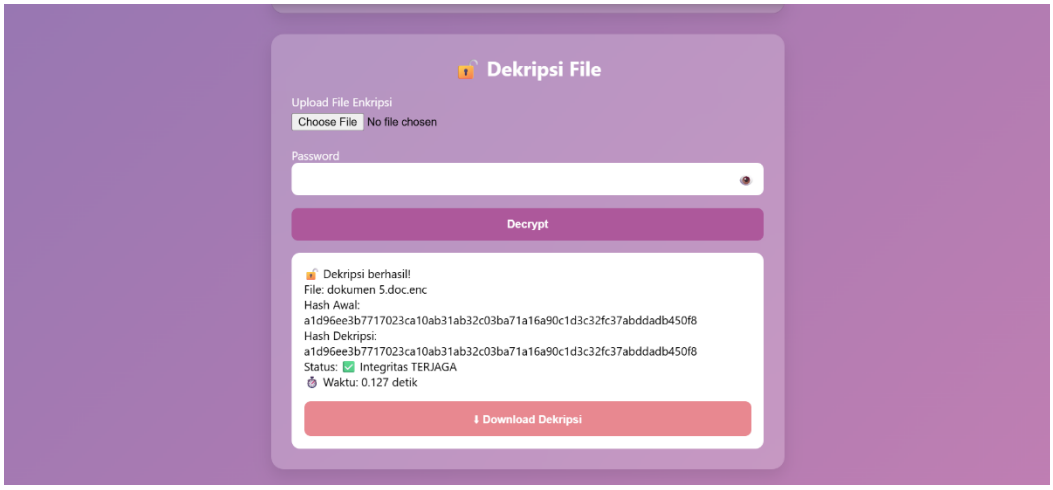


Gambar 4. 7 Tampilan Upload File dan Input Password

Gambar 4.7 menampilkan tahap-tahap yang dilakukan dalam proses pengembalian file yang terenkripsi dalam sistem. Proses dimulai ketika pengguna meng-upload file yang telah menjalani tahap enkripsi sebelumnya. Setelah file berhasil di-upload, pengguna diminta untuk memasukkan kata kunci yang sama dengan yang digunakan saat enkripsi, kata kunci ini bertindak sebagai kunci untuk membuka file yang telah dilindungi.

Selanjutnya, sistem akan memproses file tersebut dengan memanfaatkan algoritma AES-256 untuk melaksanakan dekripsi. Melalui tahapan ini, file yang semulanya berada dalam bentuk ciphertext akan diubah kembali ke bentuk aslinya (plaintext) sehingga dapat diakses dan dipergunakan kembali oleh pengguna. Proses dekripsi ini mengindikasikan bahwa sistem mampu mengembalikan file dengan efektif selama kata kunci yang dimasukkan sesuai, serta memastikan bahwa sistem keamanan yang diterapkan berfungsi sebagaimana mestinya.

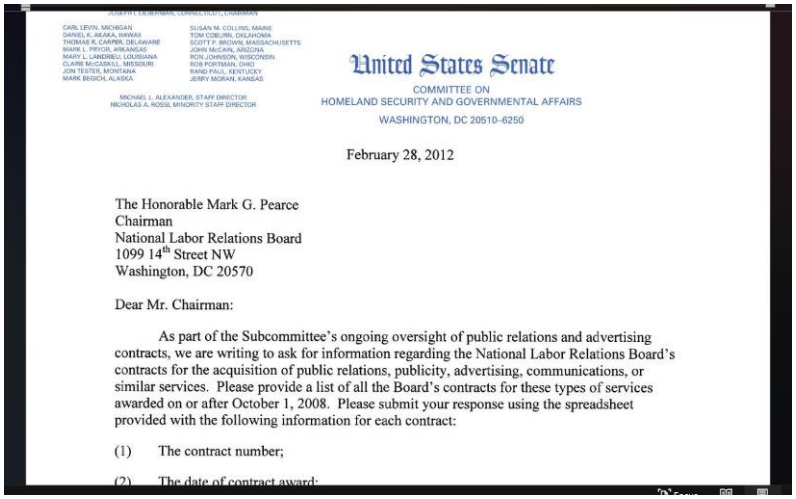
Berikut ini merupakan tampilan halaman proses dekripsi dan tampilan validasi integritas file.



Gambar 4. 8 Proses Dekripsi dan Validasi Integritas File

Gambar 4.8 menampilkan langkah-langkah dalam mendekripsi file yang dilakukan dengan cara meng-upload file yang telah dienkripsi serta memasukkan kata sandi yang sama seperti kata sandi saat melakukan enkripsi. Sistem ini akan mengembalikan berkas ke kondisi awalnya menggunakan algoritma AES-256 dan menampilkan proses pengecekan integritas berkas dengan membandingkan nilai hash sebelum enkripsi dan sesudah dekripsi. Apabila kedua nilai hash tersebut identik, maka berkas tersebut dianggap sah dan tidak mengalami modifikasi.

Berikut ini merupakan tampilan file yang telah melalui proses dekripsi pada sistem.



United States Senate

COMMITTEE ON
HOMELAND SECURITY AND GOVERNMENTAL AFFAIRS
WASHINGTON, DC 20510-6250

February 28, 2012

The Honorable Mark G. Pearce
Chairman
National Labor Relations Board
1099 14th Street NW
Washington, DC 20570

Dear Mr. Chairman:

As part of the Subcommittee's ongoing oversight of public relations and advertising contracts, we are writing to ask for information regarding the National Labor Relations Board's contracts for the acquisition of public relations, publicity, advertising, communications, or similar services. Please provide a list of all the Board's contracts for these types of services awarded on or after October 1, 2008. Please submit your response using the spreadsheet provided with the following information for each contract:

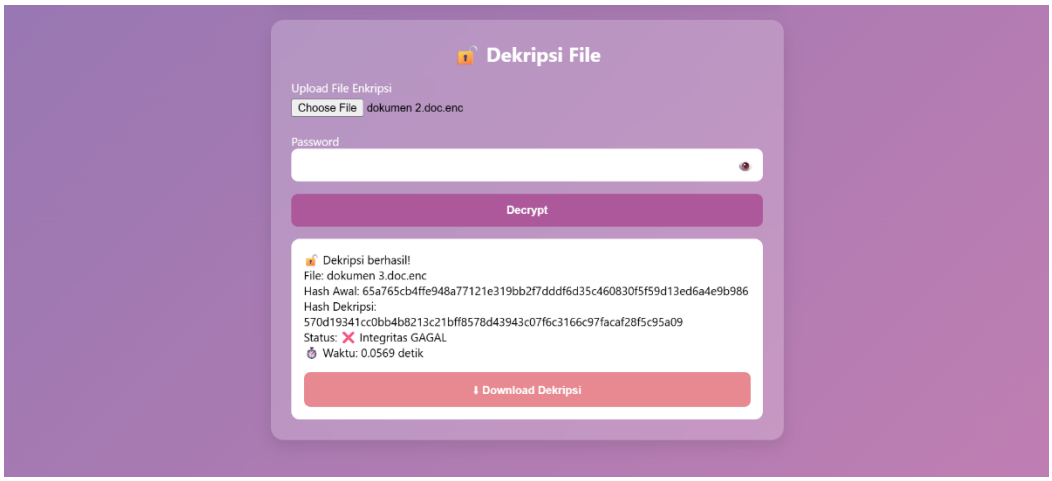
(1) The contract number;

(2) The date of contract award;

Gambar 4. 9 Hasil Dekripsi

Gambar 4.9 menampilkan hasil dari proses pembalikan enkripsi yang sudah dilakukan oleh sistem. Pada fase ini, dokumen yang sebelumnya dalam format

terenkripsi (ciphertext) telah berhasil dipulihkan ke bentuk semula (plaintext) menggunakan algoritma AES-256 dengan kata sandi yang tepat. Hasil dari proses pembalikan ini menunjukkan kemampuan sistem untuk mengembalikan file tanpa mengalami perubahan data. Selain itu, file yang sudah didekripsi itu dapat diunduh kembali oleh pengguna melalui fitur yang tersedia dalam sistem, sehingga pengguna bisa mendapatkan kembali file dalam keadaan aslinya. Proses ini juga menunjukkan bahwa mekanisme pembalikan enkripsi dalam sistem berfungsi dengan baik dan sesuai dengan desain yang telah dibuat.



Gambar 4. 10 Hasil Dekripsi dengan Status Integritas Gagal

Gambar 4.10 menunjukkan tampilan hasil proses dekripsi file pada sistem keamanan file yang dibangun. Pada gambar tersebut terlihat bahwa pengguna telah mengunggah file terenkripsi dan memasukkan kata sandi untuk melakukan proses dekripsi. Sistem kemudian menampilkan informasi hasil dekripsi, seperti nama file, nilai hash awal, nilai hash setelah dekripsi, status integritas, serta waktu yang dibutuhkan dalam proses dekripsi. Berdasarkan hasil yang ditampilkan, nilai hash awal dan hash setelah dekripsi memiliki nilai yang berbeda, sehingga sistem menyatakan bahwa integritas file gagal.

Hal ini menunjukkan bahwa file yang didekripsi tidak sama dengan file asli sebelum proses enkripsi. Perbedaan nilai hash tersebut dapat disebabkan oleh beberapa faktor, seperti kesalahan dalam memasukkan kata sandi, file yang telah mengalami perubahan, atau kerusakan data selama proses berlangsung. Oleh karena itu, sistem memberikan indikasi bahwa file hasil dekripsi tidak dapat dianggap valid dan tidak disarankan untuk digunakan lebih lanjut. Meskipun proses dekripsi berhasil dilakukan, hasil yang diperoleh tidak menjamin keaslian data, sehingga pengguna perlu memastikan kembali file dan kata sandi yang digunakan agar integritas data tetap terjaga.

Adapun solusi yang dapat dilakukan ketika terjadi kegagalan integritas file adalah sebagai berikut:

1. Memastikan kembali kata sandi yang digunakan

   Pengguna perlu memeriksa dan memasukkan kembali kata sandi yang benar sesuai dengan yang digunakan saat proses enkripsi.

2. Menggunakan file terenkripsi yang asli

   Pastikan file yang digunakan belum mengalami perubahan atau kerusakan sebelum dilakukan proses dekripsi.

3. Mengulangi proses dekripsi

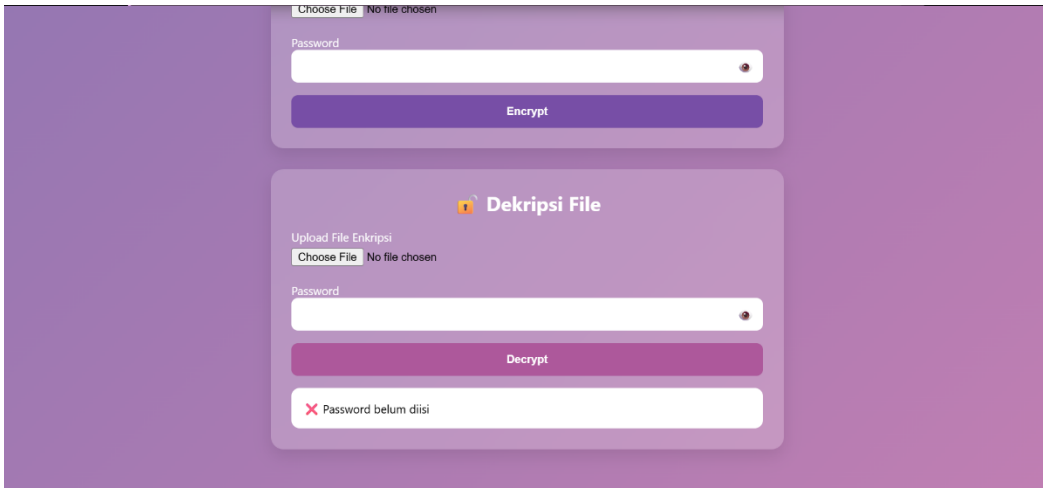
   Lakukan kembali proses dekripsi dengan input yang benar untuk memastikan hasil yang sesuai.

4. Melakukan enkripsi ulang pada file asli

   Jika file asli masih tersedia, pengguna dapat melakukan proses enkripsi ulang untuk menghasilkan file terenkripsi yang valid.

5. Menghindari penggunaan file yang gagal integritas

File yang telah dinyatakan gagal integritas sebaiknya tidak digunakan lebih lanjut karena berpotensi mengandung data yang tidak sesuai atau rusak.



Gambar 4. 11 Hasil Dekripsi dengan Status Password Belum Diisi

Gambar 4.11 menunjukkan situasi sistem ketika pengguna sudah mengunggah file yang akan didekripsi, tetapi belum memasukkan kata sandi. Saat ini, sistem tidak bisa melanjutkan proses dekripsi karena kata sandi adalah bagian penting yang dibutuhkan untuk membuka file tersebut. Sebab itu, sistem memberi tahu pengguna bahwa kolom kata sandi masih kosong dan harus diisi dulu sebelum proses dekripsi bisa dilakukan. Tujuan dari hal ini adalah mencegah terjadinya kesalahan selama proses dan memastikan hanya pengguna yang mempunyai kata sandi yang benar yang bisa mengakses isi file tersebut. Dengan adanya mekanisme ini, sistem bisa meningkatkan tingkat keamanannya sekaligus memberi arahan kepada pengguna untuk mengikuti langkah-langkah yang benar saat mengdekripsikan file.

4.4 Hasil Pengujian Sistem

Pengujian sistem dilakukan dengan beberapa contoh file yang bervariasi dalam ukuran dan format, seperti . txt, . pdf, dan . docx. Sasaran dari pengujian ini adalah untuk mengevaluasi performa sistem dalam melakukan proses enkripsi dan dekripsi menggunakan algoritma AES-256, serta untuk menjamin keutuhan data dengan menggunakan algoritma SHA-256. Hasil dari pengujian tersebut ditampilkan dalam bentuk tabel yang berisi informasi terkait nama file, ukuran file, durasi proses enkripsi dan dekripsi, nilai hash sebelum serta sesudah proses, serta status validasi integritas file.

Tabel berikut menampilkan hasil dari pengujian sistem yang dilakukan dengan menggunakan sampel file dengan ukuran spesifik.

Tabel 4. 1 Hasil Pengujian Sistem

No	Nama File	Ukuran	Waktu Enkripsi	Waktu Dekripsi
1	dokumen 5.doc	1.132 KB	0.057 detik	0.127 detik
2	pdf 1.pdf	10 KB	0.2957 detik	0.115 detik
3	file 3.txt	30.511 KB	1.305 detik	0.8843 detik

Tabel 4. 2 Hasil Pengujian Sistem

No	Nama File	Hash Awal	Hash Akhir	Status
1	dokumen 5.doc	a1d96ee3b7717023c a10ab31ab32c03ba 71a16a90c1d3c 32fc37abddadb450f8	a1d96ee3b7717023c a10ab31ab32c03ba 71a16a90c1d3c 32fc37abddadb450f8	Integritas Terjaga
2	pdf 1.pdf	4fff6f211c2c86da577	4fff6f211c2c86da577	Integritas

No	Nama File	Hash Awal	Hash Akhir	Status
		c6a922c836b7797a2 e4170a084ecf121dd 73b1bf6e089	c6a922c836b7797a2 e4170a084ecf121dd 73b1bf6e089	Terjaga
3	file 3.txt	3e5727ecfef33928 5d996d8841c7fa82 d8c5d155c1ed9256 39d87197d09f9cf1	3e5727ecfef33928 5d996d8841c7fa82 d8c5d155c1ed9256 39d87197d09f9cf1	Integritas Terjaga

Berdasarkan hasil pengujian yang tertera di Tabel 4.1 dan Tabel 4.2, sistem yang dibangun telah sukses melakukan enkripsi dan dekripsi dengan efisien serta dapat mempertahankan integritas data. Ini dibuktikan dengan waktu pemrosesan yang cukup cepat serta kecocokan nilai hash sebelum dan sesudah pemrosesan. Oleh karena itu, dapat disimpulkan bahwa algoritma AES-256 dan SHA-256 yang diterapkan mampu menyediakan tingkat perlindungan dan keutuhan data dengan efektif. Selanjutnya, hasil yang terlihat pada Tabel 4.2 menunjukkan bahwa nilai hash yang dihasilkan sebelum proses enkripsi dan setelah dekripsi adalah sama. Persamaan nilai hash ini membuktikan bahwa tidak terdapat perubahan ataupun kerusakan data selama berlangsungnya proses enkripsi dan dekripsi. Oleh karena itu, integritas file dapat dikatakan tetap terpelihara dengan baik.

Dengan mempertimbangkan hasil tersebut, bisa disimpulkan bahwa algoritma AES-256 yang diimplementasikan mampu melindungi data dengan mengubah file menjadi format yang tidak dapat dibaca tanpa kunci yang tepat, sementara algoritma SHA-256 berfungsi untuk memastikan keutuhan data melalui sistem

verifikasi nilai hash. Penggabungan kedua algoritma ini terbukti efektif dalam memberikan tingkat keamanan dan keandalan tinggi dalam perlindungan file, sehingga sistem yang dikembangkan bisa dimanfaatkan secara optimal untuk menjaga data agar tidak diakses secara tidak sah dan tetap menjaga keutuhan informasi yang tersimpan.

BAB V
PENUTUP

5.1 KESIMPULAN

Berdasarkan hasil perancangan, penerapan, dan pengujian sistem perlindungan file yang telah dilakukan, beberapa kesimpulan dapat ditarik sebagai berikut:

1. Sistem yang dibangun telah berhasil menerapkan algoritma AES-256 dalam proses enkripsi serta dekripsi file, sehingga file yang dilindungi tidak bisa dibaca tanpa adanya kata sandi yang tepat.
2. Algoritma SHA-256 yang diimplementasikan dalam sistem ini dapat digunakan untuk menghasilkan nilai hash sebagai bentuk konfirmasi integritas data, sehingga dapat memastikan bahwa file tidak mengalami perubahan selama enkripsi dan dekripsi berlangsung.
3. Berdasarkan pengujian yang dilakukan, sistem ini mampu melakukan enkripsi dan dekripsi dengan baik pada berbagai tipe dan ukuran file, dengan waktu pemrosesan yang terbilang cepat dan masih dalam batas yang dapat diterima.
4. Hasil dari pengujian integritas menunjukkan bahwa nilai hash sebelum enkripsi dan setelah dekripsi adalah sama, sehingga dapat disimpulkan bahwa integritas data terjaga dengan baik.

5. Sistem ini dirancang dengan antarmuka yang sederhana dan user-friendly, sehingga memudahkan pengguna dalam mengamankan file tanpa memerlukan pengetahuan teknis yang mendalam.

5.2 SARAN

Berikut adalah beberapa saran untuk pengembangan sistem di masa mendatang:

1. Sistem ini dapat dikembangkan menjadi platform berbasis web atau cloud agar dapat diakses dengan lebih luas dan tidak terbatas pada pengguna lokal.
2. Pengembangan berikutnya bisa menambahkan fitur manajemen pengguna (multi-user) untuk meningkatkan aspek keamanan dan kontrol akses terhadap file.
3. Sistem ini bisa diperluas dengan dukungan terhadap lebih banyak tipe file serta ukuran file yang lebih besar.
4. Untuk meningkatkan tingkat keamanan, dapat ditambahkan metode pengaman tambahan seperti tanda tangan digital atau kombinasi algoritma kriptografi lainnya dapat dipertimbangkan.
5. Penelitian ke depan bisa melakukan perbandingan kinerja dengan algoritma yang lain untuk memahami kelebihan dan kekurangan dari metode yang telah digunakan.

DAFTAR PUSTAKA

Dwi Insani, F., & Dwi Anggraeni, M. (2025). "Jurnal TRANSFORMASI (Informasi & Pengembangan Iptek)" (STMIK BINA PATRIA) ANALISIS KINERJA ALGORITMA ADVANCED ENCRYPTION STANDARD (AES) ENCRYPTION DAN ALGORITMA BLOWFISH PADA PROSES ENKRIPSI DAN DEKRIPSI. *Jurnal TRANSFORMASI*, *21*(2), 147–156.

Fathurrozi, A. (2021). Penerapan Algoritma Advanced Encryption Standard (AES-256) Dengan Mode CBC DanSecure Hash Algorithm (SHA-256) Untuk Pengamanan Data File. *Journal of Information and Information Security (JIFORTY)*, *2*(2), 227–238. http://ejurnal.ubharajaya.ac.id/index.php/jiforty

Hasan Shadzily, & Bambang Sujatmiko. (2025). Analisis Tingkat Keamanan File Dokumen Menggunakan Algoritma Advanced Encryption Standard (AES). In *Inovate : Jurnal Ilmiah Inovasi Teknologi Informasi* (Vol. 10, Number 1). https://doi.org/10.33752/inovate.v10i1.9251

Herman, W. R., Farandi Kenner, , & Miharja Satriya, W. (2021). *IMPLEMENTASI ALGORITMA AES-128 DAN SHA-256 DALAM PERANCANGAN APLIKASI PENGAMANAN FILE DOKUMEN* (Number 2).

Indrayani, R., Ferdiansyah, P., & Koprawi, M. (2025). Analisis Penggunaan Kriptografi Metode AES 256 Bit pada Pengamanan File dengan Berbagai Format. *Digital Transformation Technology*, *4*(2), 1245–1251. https://doi.org/10.47709/digitech.v4i2.5457

Iwan Setiadi, Santi Widianti, & I Putu Prachanda Kayuan. (2025). Implementasi Kriptografi Pengamanan Data Soal Ujian di Lingkungan Perguruan Tinggi Menggunakan Algoritma AES-256 dan SHA-256. *Jurnal Penelitian Rumpun Ilmu Teknik*, *4*(3), 65–90. https://doi.org/10.55606/juprit.v3i4.4569

Jefry G G Saragih. (2025). Penerapan Kriptografi untuk Pengamanan Data Nilai Siswa dengan Algoritma Super Enkripsi. *ADA Journal of Information System Research*, *2*(2), 77–85. https://doi.org/10.64366/adajisr.v2i2.78

Jehian, N. T., Kiswanto, D., Fitra, M. R. A., & Evanthe, H. V. (2025). PENGEMBANGAN SISTEM KEAMANAN DATA BERBASIS WEB MENGGUNAKAN KOMBINASI ALGORITMA CHACHA20-POLY1305 DAN ARGON2. *Jurnal Informatika Dan Teknik Elektro Terapan*, *13*(3S1). https://doi.org/10.23960/jitet.v13i3S1.8151

Maryo, M. A., & Widagdo, B. W. (2025). *IMPLEMENTASI KRIPTOGRAFI ADVANCED ENCRYPTION STANDARD (AES-128) UNTUK PENGAMANAN FILE SOAL BERBASIS WEB PADA SMP GUNUNG JATI KOTA TANGERANG* (Vol. 3, Number 2). https://mypublikasi.com/

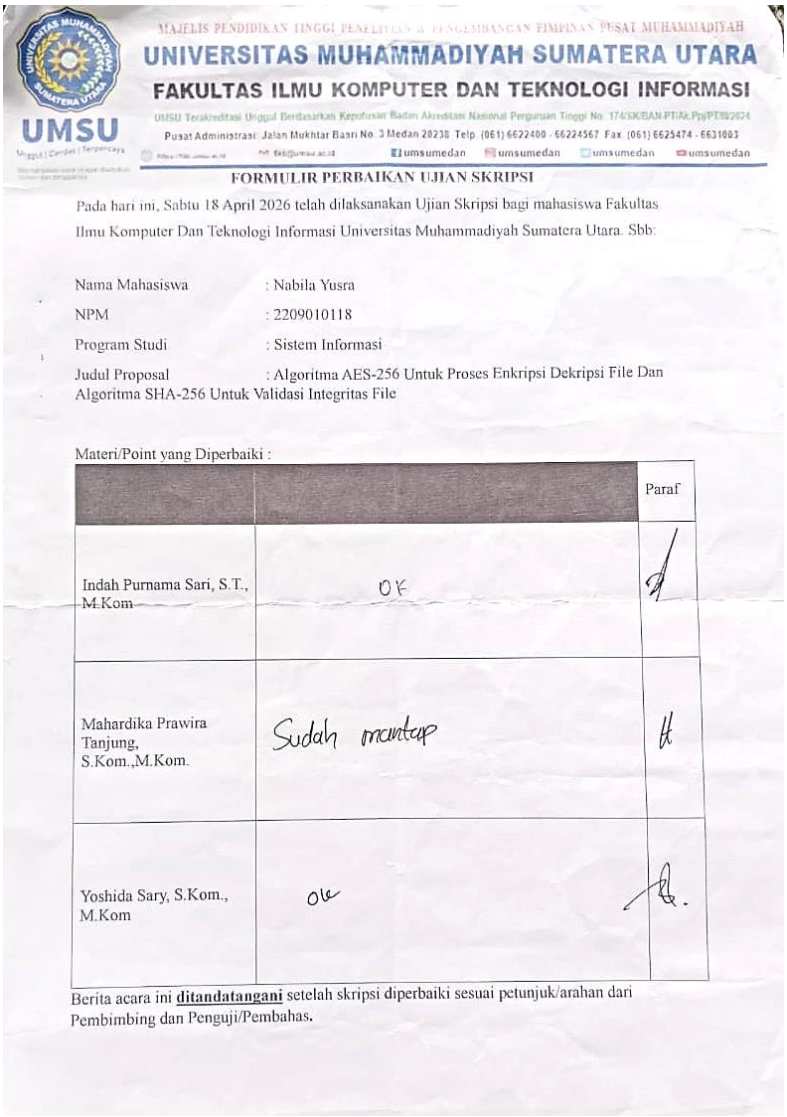
Muhsin, A., & Wiria Nugraha, D. (2025). Implementasi Algoritma Levenshtein Distance dan SHA-256 Pada Sistem Pengelolaan Arsip Dengan Evaluasi TAM. *Jurnal Algoritma*, *22*(2). https://doi.org/10.33364/algoritma/v.22-2.3085

Romzi, M., & Kurniawan, B. (2020). Implementasi Pemrograman Python Menggunakan Visual Studio Code. In *JIK: XI* (Number 2). www.python.org

Sinaga, J. S. G., Nehemia Sitorus, & Steven Lukas Samosir. (2024). Analisis Kinerja Algoritma Hash pada Keamanan Data: Perbandingan Antara SHA-256, SHA-3, dan Blake2. *JURNAL QUANCOM: QUANTUM COMPUTER JURNAL*, *2*(2), 9–16. https://doi.org/10.62375/jqc.v2i2.432

LAMPIRAN

A. Form Revisi Sidang



MAJELIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PUSAT MUHAMMADIYAH

UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UMSU Terakreditasi Unggul Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 174/SK/BAN-PT/Ak.Ppj/PT/III/2024

Pusat Administrasi: Jalan Mukhtar Basri No. 3 Medan 20238 Telp. (061) 6622400 - 66224567 Fax. (061) 6625474 - 6631003

umsumedan umsumedan umsumedan umsumedan

FORMULIR PERBAIKAN UJIAN SKRIPSI

Pada hari ini, Sabtu 18 April 2026 telah dilaksanakan Ujian Skripsi bagi mahasiswa Fakultas Ilmu Komputer Dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara. Sbb:

Nama Mahasiswa : Nabila Yusra

NPM : 2209010118

Program Studi : Sistem Informasi

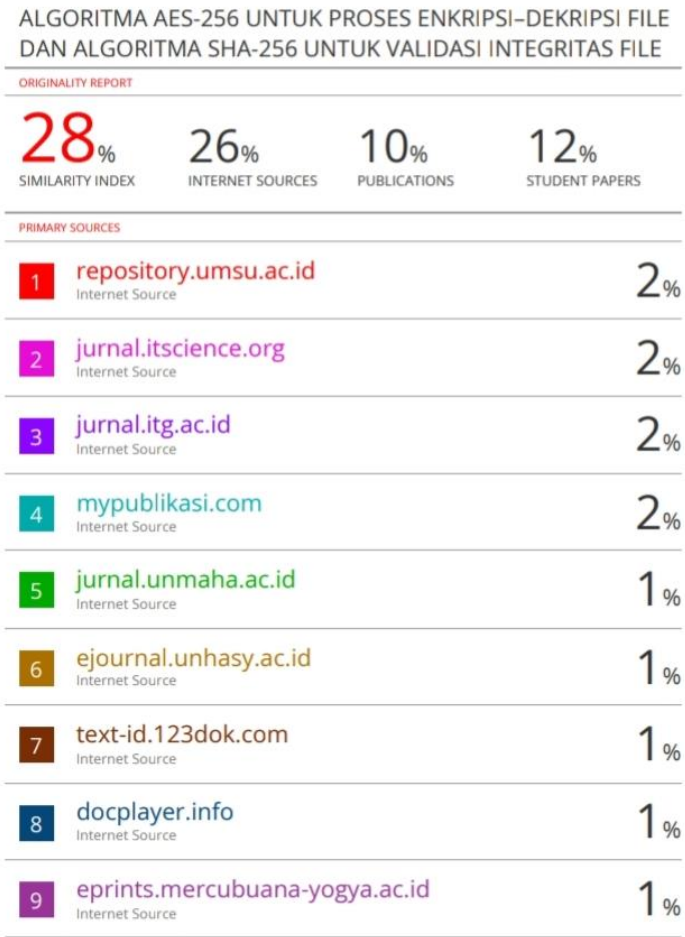
Judul Proposal : Algoritma AES-256 Untuk Proses Enkripsi Dekripsi File Dan Algoritma SHA-256 Untuk Validasi Integritas File

Materi/Point yang Diperbaiki :

		Paraf
Indah Purnama Sari, S.T., M.Kom	OK	
Mahardika Prawira Tanjung, S.Kom.,M.Kom.	Sudah mantap	
Yoshida Sary, S.Kom., M.Kom	ok	

Berita acara ini **ditandatangani** setelah skripsi diperbaiki sesuai petunjuk/arahan dari Pembimbing dan Penguji/Pembahas.

B. Turnitin



ALGORITMA AES-256 UNTUK PROSES ENKRIPSI–DEKRIPSI FILE DAN ALGORITMA SHA-256 UNTUK VALIDASI INTEGRITAS FILE

ORIGINALITY REPORT

28%	26%	10%	12%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	repository.umsu.ac.id Internet Source	2%
2	jurnal.itscience.org Internet Source	2%
3	jurnal.itg.ac.id Internet Source	2%
4	mypublikasi.com Internet Source	2%
5	jurnal.unmaha.ac.id Internet Source	1%
6	ejournal.unhasy.ac.id Internet Source	1%
7	text-id.123dok.com Internet Source	1%
8	docplayer.info Internet Source	1%
9	eprints.mercubuana-yogya.ac.id Internet Source	1%

C. LoA

Blend Sains
Jurnal Teknik
https://jurnal.ilmubersama.com/index.php/blendsains

LETTER OF ACCEPTANCE (LoA)

Kepada Yth Bpk/Ibu/Sdr

Nabila Yusra, Indah Purnama Sari

Di

Tempat

Dengan ini kami sampaikan bahwa naskah dengan rincian berikut dinyatakan diterima untuk diterbitkan di dalam Blend Sains Jurnal Teknik, ISSN 2964-7347 (Online) pada terbitan Volume 5 Nomor 1 Edisi Juli 2026

Judul	Algoritma AES-256 Untuk Proses Enkripsi Dekripsi File dan Algoritma SHA-256 Untuk Validasi Integritas File
Penulis	Nabila Yusra, Indah Purnama Sari
Correspondent Email	indahpurnama@umsu.ac.id

Demikianlah surat keterangan ini kami buat untuk dapat digunakan seperlunya.

Medan, 29 Mei 2026

Editor in Chief

Sony Bahagia Sinaga, M.Kom

D. Bukti Submit