Blantika: Multidisciplinary Jornal
Volume 3 Number 6, Mei, 2025
p- ISSN 2987-758X e-ISSN 2985-4199

Perancangan Sistem Enkripsi dan Dekripsi *File* Dokumen Berbasis *Web* Pada *Google Drive* Menggunakan Algoritma *Advanced Encryption Standard* (AES)

Ignasius Kosat Jessica Anastasia A.Tanesib, Maria O.Bona, Falentina Sallu, Mayela K. Sifa
Fakultas Sains dan Teknik, Universitas Timor, Indonesia
Email: ignaskosat01@gmail.com, jessicatanesib7@gmail.com, mayelasifa@gmail.com, moyle3846@gmail.com, falentinasallu@gmail.com

ABSTRAK

Keamanan data merupakan aspek yang sangat penting, terutama saat menggunakan layanan penyimpanan seperti Google Drive. Meskipun Google Drive dilengkapi dengan sistem enkripsi, pengguna masih memerlukan perlindungan tambahan untuk memastikan keamanan file mereka. Penelitian ini bertujuan untuk merancang dan mengembangkan sistem berbasis web yang dapat mengenkripsi dan mendekripsi dokumen menggunakan algoritma Advanced Encryption Standard (AES). File yang didukung oleh sistem ini meliputi file teks (.txt), dokumen Word (.docx), PowerPoint (.pptx), dan Excel (.xlsx). Sistem yang dikembangkan menyediakan antarmuka yang intuitif, termasuk fitur unggah otomatis yang memungkinkan pengguna untuk langsung mengunggah file hasil enkripsi ke Google Drive hanya dengan mengklik tombol yang disediakan. Integrasi ini bertujuan untuk meningkatkan pengalaman pengguna dalam mengamankan data, serta mencegah potensi akses tidak sah dari pihak lain yang mungkin memiliki akses ke tautan file di Google Drive tanpa izin pengguna. Sistem ini berfungsi dengan cara mengenkripsi file sebelum diunggah ke Google Drive dan mendekripsi file setelah diunduh, sehingga hanya pengguna yang memiliki akses yang dapat membuka file tersebut. Proses pembuatan sistem dilakukan melalui beberapa tahap, mulai dari analisis kebutuhan, perancangan antarmuka, integrasi dengan Google Drive, hingga pengujian. Hasil pengujian menunjukkan bahwa sistem berfungsi dengan baik dan mampu menjaga keamanan file pengguna. Sistem ini dapat menjadi solusi tambahan untuk melindungi file yang disimpan di Google Drive.

Kata kunci: Enkripsi, Dekripsi, AES, Google Drive, Keamanan File, Web.

ABSTRACT

Data security is a very important aspect, especially when using storage services such as Google Drive. Although Google Drive is equipped with an encryption system, users still need additional protection to ensure the security of their files. This study aims to design and develop a web-based system that can encrypt and decrypt documents using the AES (Advanced Encryption Standard) algorithm. Files supported by this system include text files (.txt), Word documents (.docx), PowerPoint (.pptx), and Excel (.xlsx). The developed system provides an intuitive interface, including an auto-upload feature that allows users to directly upload encrypted files to Google Drive by simply clicking the button provided. This integration aims to improve the user experience in securing data, as well as prevent potential unauthorized access from other parties who may have access to file links on Google Drive without user permission. This system works by encrypting files before they are uploaded to Google Drive and decrypting files after they are downloaded, so that only users who have access can open the files. The system creation

process is carried out through several stages, starting from needs analysis, interface design, integration with Google Drive, to testing. The test results show that the system functions well and is able to maintain the security of user files. This system can be an additional solution to protect files stored on Google Drive.
Keywords: *Encryption, Decryption, AES, Google Drive, File Security, Web.*

Manuscript accepted:27 May 2025 Revised: 27 May 2025 Date of publication: 2 June 2025

PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi (TIK) global telah merevolusi berbagai aspek kehidupan manusia, termasuk dalam hal penyimpanan dan pengelolaan data. Data digital kini menjadi komoditas penting yang mendasari banyak aktivitas, mulai dari pendidikan, bisnis, hingga pemerintahan. Penyimpanan data berbasis cloud telah menjadi standar baru karena kemampuannya menyediakan akses fleksibel, efisiensi biaya, dan skalabilitas tinggi (Kaufman et al., 2009). Namun, seiring dengan meningkatnya penggunaan layanan cloud, ancaman terhadap keamanan dan privasi data turut meningkat. Laporan oleh IBM (2022) menunjukkan bahwa rata-rata kerugian akibat pelanggaran data global mencapai USD 4,35 juta per insiden. Selain itu, studi dari Alasmary dan Alhaidari (2021) mencatat bahwa penyimpanan berbasis cloud menjadi target utama serangan siber karena menyimpan data sensitif dalam jumlah besar..

Tabel 1. Jumlah Sekolah, Guru, dan Siswa Berdasarkan Jenjang Pendidikan

No	Jenjang Pendidikan	Jumlah Sekolah	Jumlah Guru	Jumlah Siswa
1	SD	147.726	1.190.412	25.015.893
2	SMP	40.421	598.632	10.565.793
3	SMA	13.689	344.583	4.842.746
4	SMK	14.678	379.255	5.024.562

Di Indonesia, isu keamanan data dalam layanan cloud menjadi semakin relevan seiring meningkatnya adopsi teknologi digital di sektor pendidikan. Berdasarkan data Kemendikbudristek, terdapat 147.726 sekolah dasar (SD), 40.421 sekolah menengah pertama (SMP), 13.689 sekolah menengah atas (SMA), dan 14.678 sekolah menengah kejuruan (SMK), dengan total lebih dari 2,5 juta guru dan 45 juta siswa. Penggunaan Google Drive sebagai platform penyimpanan digital telah meluas di kalangan pendidik dan pelajar karena kemudahannya dalam berbagi dan mengakses file pembelajaran (Kementerian Pendidikan dan Kebudayaan, 2023). Namun, belum semua institusi menerapkan standar pengamanan data yang memadai, sehingga risiko kebocoran dan penyalahgunaan data peserta didik menjadi perhatian utama (Setiawan & Nugroho, 2022; Faridah et al., 2023).

Berbagai penelitian sebelumnya telah membahas integrasi kriptografi dalam penyimpanan cloud. Misalnya, Kumar et al. (2021) meneliti penggunaan algoritma AES dalam sistem penyimpanan berbasis cloud dan menemukan bahwa metode ini mampu menjaga integritas dan kerahasiaan data secara efektif. Studi lain oleh Zhao dan Liu (2020) menyarankan kombinasi antara algoritma enkripsi dan otentikasi ganda sebagai pendekatan komprehensif terhadap perlindungan data pengguna. Sementara itu, Prasetya dan Wibowo (2022) mengevaluasi sistem enkripsi file pada platform cloud lokal dan

menyimpulkan perlunya pendekatan yang lebih terintegrasi dengan platform global seperti Google Drive agar dapat meningkatkan adopsi pengguna.

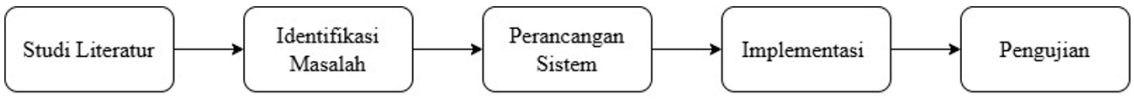
Novelty atau kebaruan dari penelitian ini terletak pada pengembangan sistem enkripsi file berbasis algoritma AES yang secara langsung terintegrasi dengan layanan Google Drive, sehingga pengguna tidak perlu mengenkripsi file secara manual sebelum mengunggah. Sebagian besar aplikasi sejenis masih bersifat terpisah dan memerlukan proses yang kompleks (Rahim et al., 2020). Selain itu, pendekatan penelitian ini lebih menekankan pada usability dan kompatibilitas dengan ekosistem pendidikan digital di Indonesia, yang belum banyak dikaji sebelumnya (Fitriansyah & Hidayat, 2021; Alam et al., 2023). Dengan pendekatan ini, diharapkan dapat mengurangi hambatan teknis pengguna akhir dan meningkatkan adopsi sistem keamanan berbasis kriptografi.

Tujuan utama dari penelitian ini adalah untuk mengembangkan sistem enkripsi file menggunakan algoritma Advanced Encryption Standard (AES) yang dapat terintegrasi secara langsung dengan layanan Google Drive. Sistem ini dirancang untuk memberikan perlindungan data yang optimal dengan proses enkripsi dan dekripsi yang efisien dan user-friendly. Selain itu, sistem ini bertujuan untuk menjawab kebutuhan pengguna dalam mengamankan dokumen digital penting, khususnya dalam konteks pendidikan dan lembaga yang mengandalkan penyimpanan cloud.

Manfaat dari penelitian ini adalah tersedianya solusi keamanan data yang praktis dan dapat diadopsi secara luas oleh institusi pendidikan, organisasi, maupun individu. Implementasi sistem enkripsi ini diharapkan dapat meningkatkan kesadaran terhadap pentingnya perlindungan data digital, sekaligus menekan risiko penyalahgunaan data oleh pihak tidak bertanggung jawab. Selain itu, penelitian ini juga dapat menjadi acuan bagi pengembangan sistem keamanan lainnya di masa depan, baik di ranah akademik maupun industri teknologi informasi.

METODE PENELITIAN

Tahapan yang dilakukan pada penelitian ini dimulai dari studi literatur untuk mengumpulkan informasi, teori dan hasil penelitian sebelumnya yang berkaitan dengan konsep dasar dan teknologi yang digunakan yaitu, algoritma *Advenced Encryption Standard* (AES) dan Integrasi dengan Google Drive API. Setelah mengumpulkan informasi dari studi literatur, tahap selanjutnnya adalah identifikasi masalah utama yang akan diselesaikan melalui penelitian ini. Kemudian dilanjutkan dengan perancangan sistem untuk mengatasi masalah yang telah diidentifikasi. Pada tahap ini, perancangan mencakup pembuatan *Flowchart* yang menggambarkan alur kerja sistem secara keseluruhan. Tahap berikutnya adalah Implementasi yaitau membangun sistem yang sesuai dengan rancangan yang telah dibuat dan tahap terakhir adah pengujian untuk memastikan bahwa sistem yang dibangun berjalan sesuai dengan kebutuhan. Pada tahap ini, metode *Black Box Testing* digunankan untuk menguji fungsionalitas sistem.



Gambar 1. Metode Penelitian

A. Algoritma Advanced Encryption Standard (AES)

Advanced Encryption Standard (AES) merupakan algoritma kriptografi yang dapat digunakan untuk mengamankan data. Algoritma AES adalah blok *ciphertext*

simetrik yang dapat mengenkripsi (*encipher*) dan mendekripsi (*decipher*) informasi. Enkripsi mengubah data asli menjadi bentuk yang tidak terbaca, disebut *ciphertext,* sedangkan dekripsi adalah proses mengembalikan *ciphertext* menjadi bentuk semula yang dikenal sebagai *plaintext*. AES menggunakan kunci kriptografi berukuran 128, 192, dan 256 bit untuk proses enkripsi dan dekripsi data (Daemen & Rijmen, 2002; Stallings, 2017). Keunggulan AES terletak pada efisiensi dan kekuatannya terhadap berbagai serangan kriptografi, sehingga menjadikannya standar global untuk keamanan data di berbagai sistem informasi dan komunikasi (Menezes et al., 1996; Singh et al., 2013). Algoritma AES mengenkripsi setiap blok dalam sejumlah putaran tertentu, Berikut Pada Tabel 1 terdapat jumlah putaran kunci yang digunakan dalam algoritma AES.

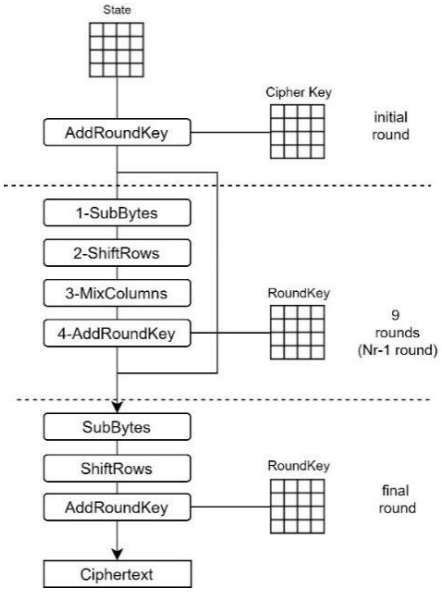
Tabel 1. Putaran Kunci Algoritma AES

AES (Bits)	Panjang Kunci (Nk Words)	Ukuran Blok (Nb Words)	Jumlah Putaran (Nr)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Langkah dekripsi AES, yang juga dikenal sebagai *Inverse Cipher* dari algoritma Rijndael dengan blok 128-bit dan kunci 128-bit, dimulai dengan ciphertext sebagai input awal, yaitu data yang telah dienkripsi. Proses ini diawali dengan *Initial Round* yang terdiri dari tiga tahap: *AddRoundKey*, yaitu melakukan operasi XOR antara ciphertext dengan round key terakhir; *InvShiftRows*, yaitu pergeseran baris ke kanan sebagai kebalikan dari proses enkripsi; dan *InvSubBytes*, yaitu pengembalian setiap byte ke nilai semula menggunakan *Inverse S-Box*. Selanjutnya, dilakukan *Nr-1 Rounds* (jumlah ronde dikurangi satu), di mana setiap ronde melibatkan empat langkah utama: *AddRoundKey*, *InvMixColumns* untuk membalik transformasi kolom, *InvShiftRows*, dan *InvSubBytes*. Pada *Final Round*, hanya dilakukan *AddRoundKey* tanpa *InvMixColumns*, sebagaimana pada ronde akhir saat proses enkripsi. Setelah semua ronde selesai dijalankan, hasil akhirnya adalah plaintext, yaitu data asli yang berhasil dikembalikan dari bentuk terenkripsi.

B. Proses Enkripsi AES

Proses enkripsi terdiri dari 4 jenis transformasi bytes, yaitu *SubBytes, ShiftRows, Mixcolumns,* dan *AddRoundKey.* Pada awal proses enkripsi, input yang telah dicopikan ke dalam state akan mengalami transformasi byte *AddRoundKey*. Setelah itu, state akan mengalami transformasi *SubBytes, ShiftRows, MixColumns,* dan *AddRoundKey* secara berulang- ulang sebanyak Nr. Proses ini dalam algoritma AES disebut sebagai *round function*. Round yang terakhir agak berbeda dengan round-round sebelumnya dimana pada round terakhir, state tidak mengalami transformasi *MixColumns* (8).

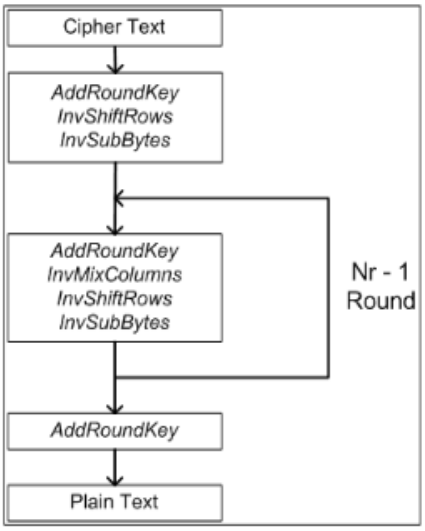


Gambar 2. Proses Enkripsi AES (9).

Garis besar algoritma AES Rijndael yang beroperasi pada blok 128-bit dengan kunci 128-bit (di luar proses pembangkitan *round key*) terdiri dari tiga tahapan utama. Tahap pertama adalah *Initial Round*, yang dilakukan satu kali dan mencakup operasi *AddRoundKey*, yaitu menggabungkan data awal (*state*) dengan kunci enkripsi utama. Selanjutnya, dilakukan 9 ronde utama atau *Nr-1 Rounds*, di mana setiap ronde terdiri dari empat langkah berurutan: *SubBytes* yang melakukan substitusi byte menggunakan tabel S-Box, *ShiftRows* yang menggeser baris data secara sistematis, *MixColumns* untuk mencampur kolom-kolom data guna meningkatkan difusi, serta *AddRoundKey* untuk menambahkan kunci ronde yang dihasilkan dari proses ekspansi kunci. Terakhir, pada *Final Round* yang juga dilakukan sekali, algoritma menjalankan tiga langkah tanpa menyertakan *MixColumns*, yaitu *SubBytes*, *ShiftRows*, dan *AddRoundKey*, yang menutup keseluruhan proses enkripsi AES dengan menjaga keamanan dan kompleksitas transformasi data.

C. Proses Dekripsi AES

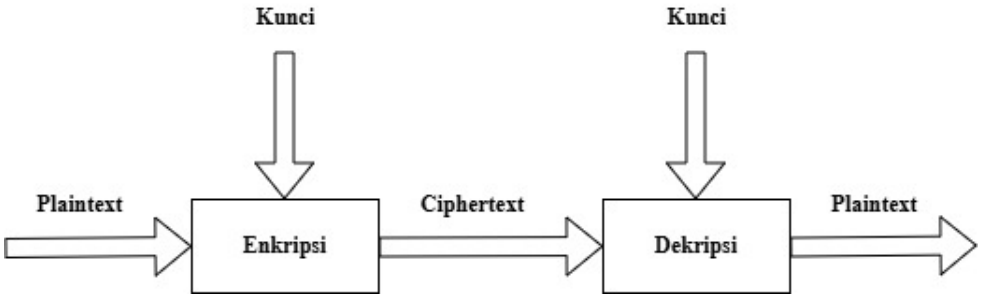
Transformasi *cipher* dapat dilakukan secara invers dan diaplikasikan dalam tujuan yang berlawanan guna untuk menciptakan cipher invers yang gampang dimengerti untuk algoritma AES. Byte yang diubah untuk keperluan invers. cipher merupakan *InvShiftRows, InvSubBytes, InvMixColumns* dan *AddRoundKey.* Algoritma dekripsi dapat diamati pada skema di bawah ini:



Gambar 3. Proses Dekripsi AES (9).

Langkah dekripsi AES, yang juga dikenal sebagai Inverse Cipher dari algoritma Rijndael dengan blok 128-bit dan kunci 128-bit, dimulai dengan ciphertext sebagai input awal, yaitu data yang telah dienkripsi. Proses ini diawali dengan Initial Round yang terdiri dari tiga tahap: AddRoundKey, yaitu melakukan operasi XOR antara ciphertext dengan round key terakhir; InvShiftRows, yaitu pergeseran baris ke kanan sebagai kebalikan dari proses enkripsi; dan InvSubBytes, yaitu pengembalian setiap byte ke nilai semula menggunakan Inverse S-Box. Selanjutnya, dilakukan Nr-1 Rounds (jumlah ronde dikurangi satu), di mana setiap ronde melibatkan empat langkah utama: AddRoundKey, InvMixColumns untuk membalik transformasi kolom, InvShiftRows, dan InvSubBytes. Pada Final Round, hanya dilakukan AddRoundKey tanpa InvMixColumns, sebagaimana pada ronde akhir saat proses enkripsi. Setelah semua ronde selesai dijalankan, hasil akhirnya adalah plaintext, yaitu data asli yang berhasil dikembalikan dari bentuk terenkripsi.

D. Skema Enkripsi dan Dekripsi



Gambar 4. Skema Enkripsi dan Dekripsi AES-128

Gambar tersebut menggambarkan proses enkripsi dan dekripsi dalam kriptografi simetris. Plaintext diubah menjadi ciphertext menggunakan sebuah kunci melalui proses enkripsi. Setelah itu, ciphertext dapat dikirimkan atau disimpan dengan aman.

Untuk mendapatkan kembali plaintext, dilakukan proses dekripsi dengan menggunakan kunci yang sama seperti saat enkripsi.

E. Pengertian Google Drive API

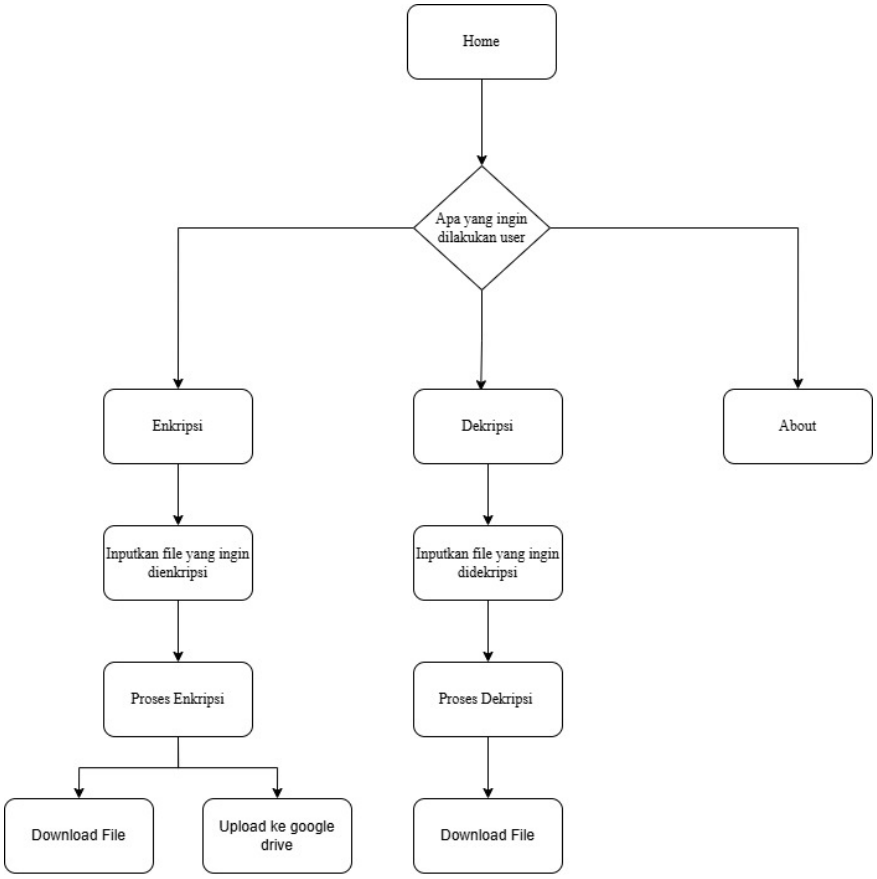
Google Drive API adalah antarmuka pemrograman aplikasi (API) yang memungkinkan aplikasi pihak ketiga untuk mengakses, mengunggah, mengedit, dan mengelola file yang disimpan di Google Drive secara terprogram (10). API ini menyediakan berbagai fungsi yang memudahkan pengembang dalam membuat file baru, menghapus file yang sudah ada, serta mengunggah file ke Google Drive secara otomatis. Dengan memanfaatkan Google Drive API, sistem yang dibangun dapat terhubung langsung dengan layanan penyimpanan cloud Google Drive, sehingga memungkinkan proses upload dan download file berjalan lebih aman, efisien, dan terpusat.

F. Penerapan Google Drive API dalam Sistem

Dalam penelitian ini, Google Drive API diterapkan untuk mendukung proses penyimpanan dan pengambilan file hasil enkripsi dan dekripsi. Setelah file teks berhasil dienkripsi menggunakan algoritma Advanced Encryption Standard (AES), file hasil enkripsi secara otomatis diunggah ke akun Google Drive pengguna, sementara dalam proses dekripsi, pengguna dapat mengunduh file terenkripsi dari Google Drive untuk dikembalikan ke bentuk plaintext. Pemanfaatan Google Drive API dalam sistem ini mencakup beberapa fungsi penting, seperti mengunggah file hasil enkripsi, mengunduh file terenkripsi untuk proses dekripsi, serta mengelola metadata file, termasuk nama file, ukuran, dan lokasi penyimpanan di dalam folder Drive. Selain itu, sistem menjamin keamanan dengan membatasi akses aplikasi hanya pada folder tertentu, sehingga privasi data pengguna tetap terlindungi. Melalui penerapan ini, sistem menawarkan solusi penyimpanan berbasis cloud yang aman, fleksibel diakses dari berbagai perangkat, dan mampu mengurangi risiko kehilangan data akibat kerusakan atau kehilangan perangkat lokal.

G. Integrasi Google Drive API

Integrasi Google Drive API dalam sistem ini dilakukan melalui protokol autentikasi OAuth 2.0, yang berfungsi untuk memastikan bahwa akses ke akun Google Drive pengguna dilakukan secara aman dan sesuai dengan izin yang diberikan. Dalam proses integrasi ini, beberapa endpoint utama yang digunakan meliputi *Upload File*, yang memungkinkan sistem mengirim file hasil enkripsi secara otomatis ke Google Drive pengguna; *Download File*, yang berfungsi untuk mengambil file terenkripsi dari Google Drive guna diproses dalam tahap dekripsi; serta *Delete File*, yang memungkinkan pengguna menghapus file dari Google Drive apabila sudah tidak diperlukan. Dengan adanya integrasi ini, sistem tidak hanya dapat mengelola file secara efisien di cloud, tetapi juga menjaga aspek keamanan, keteraturan, dan kemudahan akses terhadap file yang tersimpan, sehingga meningkatkan pengalaman pengguna dalam pengelolaan data digital secara terproteksi.



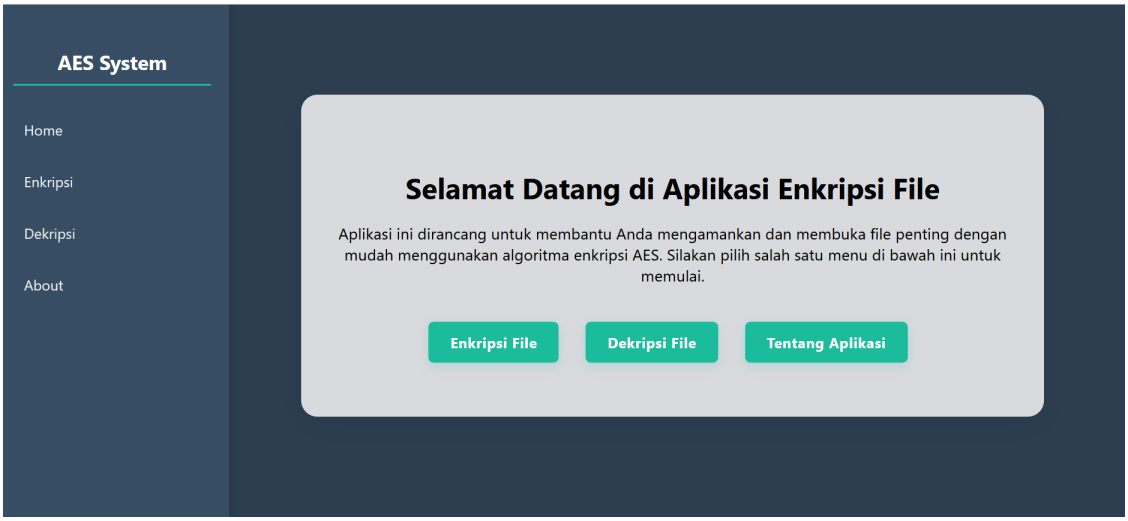
Gambar 5. Flowchart Sistem

Diagram alur di atas menggambarkan proses utama dalam sistem enkripsi dan dekripsi file yang terintegrasi dengan Google Drive. Alur dimulai dari halaman utama (Home), tempat pengguna mengakses seluruh fungsi sistem. Di halaman ini, pengguna disajikan tiga pilihan aksi, yaitu Enkripsi, Dekripsi, dan About. Jika memilih Enkripsi, sistem akan meminta pengguna untuk mengunggah file yang ingin diamankan. File tersebut kemudian diproses menggunakan algoritma AES hingga menghasilkan file terenkripsi. Setelah proses selesai, pengguna dapat memilih untuk mengunduh file ke perangkat lokal atau mengunggahnya langsung ke Google Drive melalui integrasi API. Sebaliknya, jika memilih Dekripsi, pengguna akan diminta untuk mengunggah file terenkripsi, yang kemudian akan didekripsi oleh sistem agar kembali ke bentuk aslinya (plaintext), dan dapat diunduh kembali. Sementara itu, pada menu About, sistem menampilkan informasi tentang tujuan sistem, metode enkripsi yang digunakan, serta deskripsi fitur-fitur utama yang tersedia.

HASIL DAN PEMBAHASAN

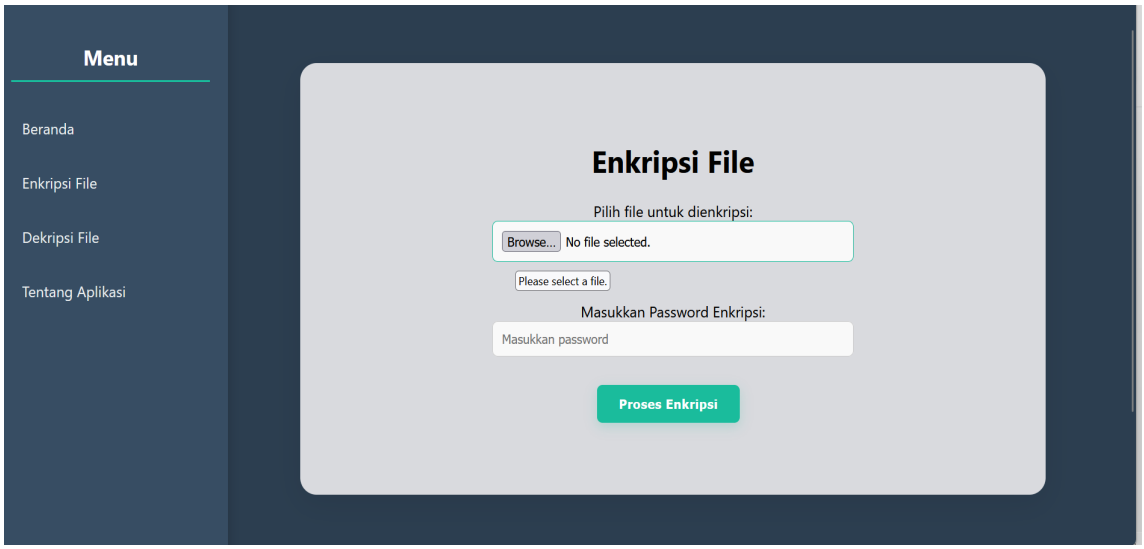
Setelah perancangan sistem dilakukan pada bagian sebelumnya. Selanjutnya akan diuraikan tentang hasil dari penerapan sistem dengan AES-128 dan pengujian sistem melalui Black Box Testing dengan cara berikut:

A. Implementasi Sistem



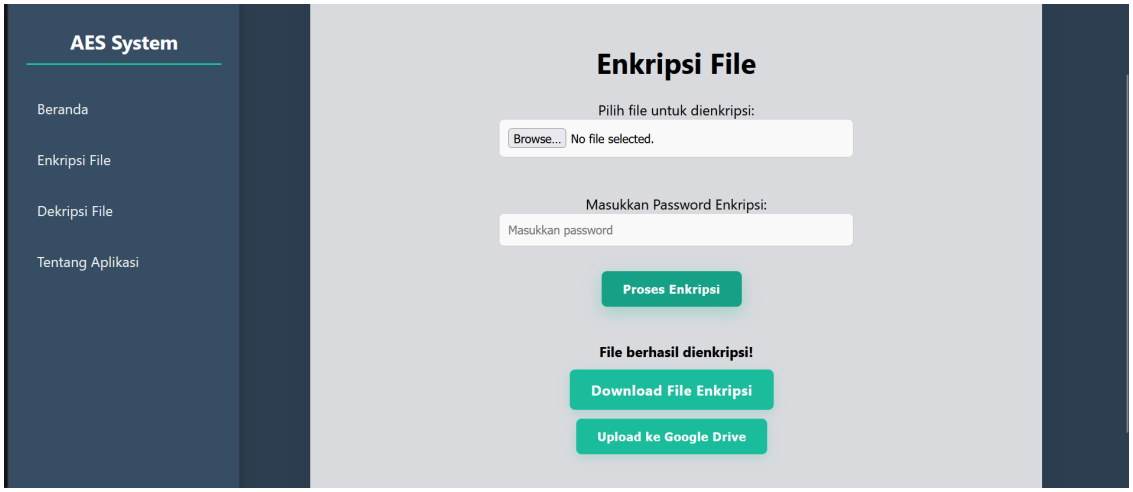
Gambar 6. Tampilan Utama

Gambar ini memperlihatkan tampilan utama dari aplikasi enkripsi file yang berbasis web. Aplikasi ini dibuat untuk membantu pengguna dalam melindungi dan mengakses file-file penting dengan menggunakan algoritma enkripsi AES (Advanced Encryption Standard). Pada tampilan ini, terdapat tiga tombol utama yang mengarahkan pengguna ke fitur Enkripsi File, Dekripsi File, dan Informasi Aplikasi. Navigasi yang berada di sisi kiri memudahkan pengguna untuk beralih antar halaman dalam aplikasi.



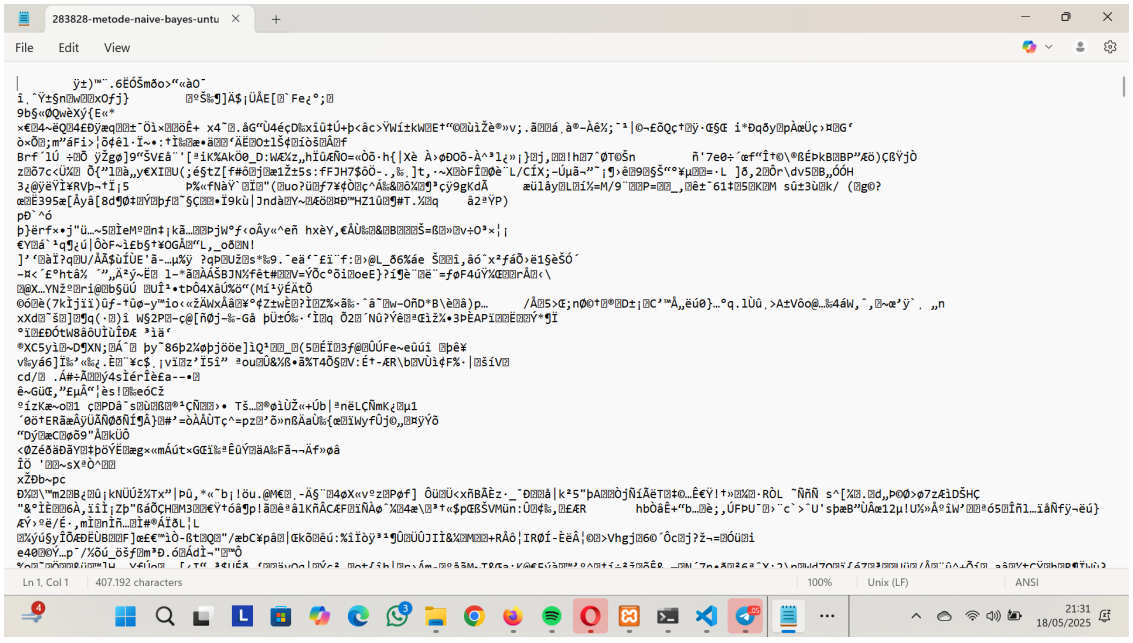
Gambar 7. Tampilan Halaman Enkripsi File

Gambar ini menampilkan halaman enkripsi file pada aplikasi. Pengguna dapat memilih file yang ingin dienkripsi melalui tombol *Browse*, kemudian memasukkan kata sandi (*password*) untuk proses enkripsi. Setelah data diisi, pengguna dapat menekan tombol *Proses Enkripsi* untuk mengenkripsi file menggunakan algoritma AES.

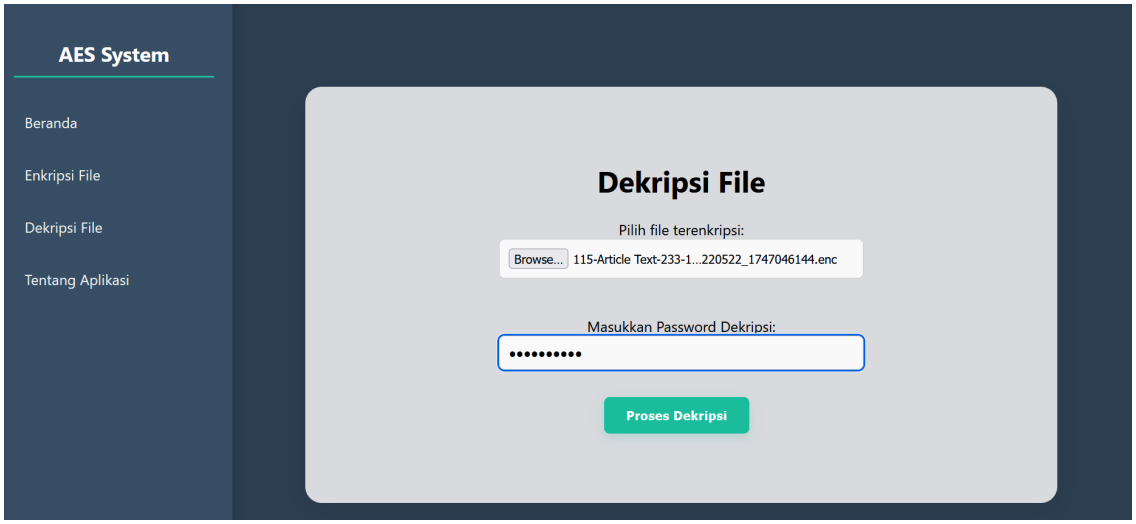


Gambar 8. Tampilan Hasil Enkripsi File

Gambar ini menunjukkan tampilan setelah proses enkripsi file berhasil dilakukan. Pengguna diberikan dua opsi lanjutan, yaitu *Download File Enkripsi* untuk mengunduh file hasil enkripsi ke perangkat lokal, dan *Upload ke Google Drive* untuk menyimpan file tersebut secara aman di cloud.

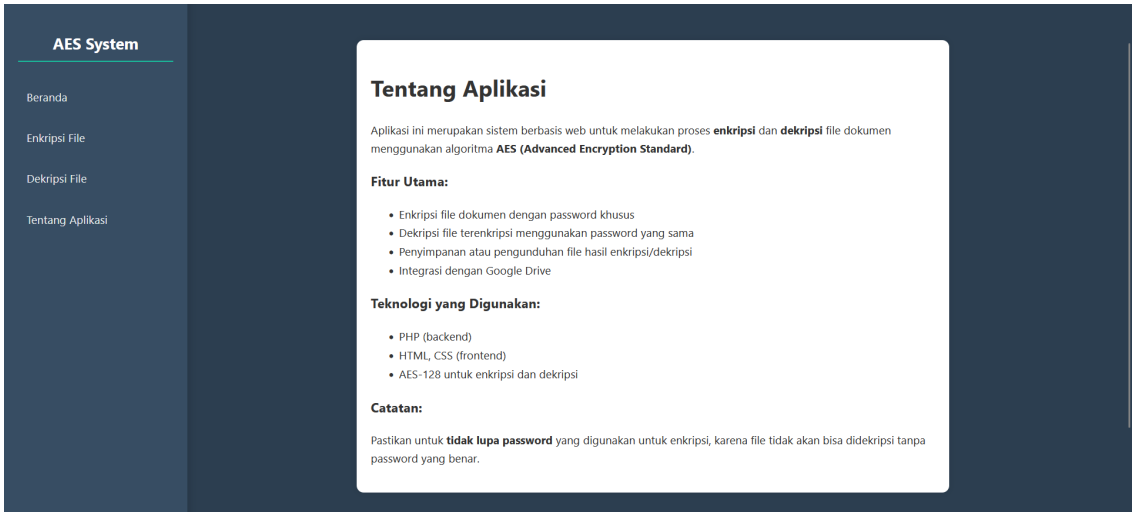


Gambar 9. File Hasil Enkripsi



Gambar 10. Tampilan Halaman Dekripsi File

Gambar ini menunjukkan antarmuka halaman dekripsi file pada aplikasi. Pengguna dapat memilih file yang sebelumnya telah dienkripsi (ditandai dengan ekstensi .enc), kemudian memasukkan password yang digunakan saat proses enkripsi. Setelah itu, pengguna dapat menekan tombol *Proses Dekripsi* untuk mengembalikan file ke bentuk aslinya.



Gambar 11. Halaman Tentang

Pengujian Black Box Testing

Tabel 2. Pengujian *Black Box Testing*

Tampilan	Pengujian	Hasil yang Diharapkan	Hasil
Halaman Enkripsi	Klik tombol “Pilih File”	Menampilkan dialog pemilihan file untuk dienkripsi	Berhasil
	Masukkan password dan klik tombol “Proses Enkripsi”	Menampilkan notifikasi bahwa file berhasil dienkripsi	Berhasil

	Klik tombol “Download File Enkripsi”	File hasil enkripsi berhasil diunduh	Berhasil
	Klik tombol “Upload ke Google Drive”	File hasil enkripsi berhasil diunggah ke akun Google Drive pengguna	Berhasil
Halaman Dekripsi	Klik tombol “Pilih File”	Menampilkan dialog pemilihan file berekstensi .enc	Berhasil
	Masukkan password dan klik tombol “Proses Dekripsi”	Menampilkan notifikasi bahwa file berhasil didekripsi	Berhasil
	File hasil dekripsi dapat dibuka sesuai format aslinya	Isi file sesuai dengan konten sebelum dienkripsi	Berhasil

KESIMPULAN

Hasil implementasi algoritma kriptografi AES-128 pada sistem ini menunjukkan performa yang efektif dalam mengenkripsi dan mendekripsi file digital. Penggunaan metode Black Box Testing membuktikan bahwa seluruh fitur utama aplikasi, seperti unggah file, proses enkripsi-dekripsi, serta integrasi dengan Google Drive, berjalan dengan baik dan sesuai harapan. Pengguna dapat melakukan enkripsi dokumen penting dengan mudah menggunakan password sebagai kunci, kemudian menyimpan hasilnya secara lokal maupun di cloud (Google Drive). Proses enkripsi menggunakan algoritma AES-128 dengan keamanan tinggi yang melalui empat tahap utama dan memberikan perlindungan kuat terhadap akses tidak sah. Dengan keberhasilan pengujian ini, sistem direkomendasikan untuk pengembangan lebih lanjut demi peningkatan keamanan informasi dan pengalaman pengguna.

DAFTAR PUSTAKA

Alam, M. R., Hasan, M. M., & Sultana, S. (2023). Usability evaluation of cloud encryption systems: A case study on educational platforms. *International Journal of Information Security, 22*(1), 15–29. https://doi.org/10.1007/s10207-022-00635-x

Alasmary, W., & Alhaidari, F. (2021). Security challenges in cloud computing environments: A survey. *Journal of Cloud Computing: Advances, Systems and Applications, 10*(1), 1–18. https://doi.org/10.1186/s13677-021-00239-3

Daemen, J., & Rijmen, V. (2002). *The design of Rijndael: AES - The Advanced Encryption Standard*. Springer. https://doi.org/10.1007/978-3-662-04722-4

Faridah, S., Rahayu, I., & Sutrisno, A. (2023). Analisis keamanan data peserta didik pada layanan cloud storage di lingkungan pendidikan. *Jurnal Teknologi dan Keamanan Informasi, 11*(2), 107–114.

Fitriansyah, R., & Hidayat, D. (2021). Penggunaan algoritma AES untuk sistem keamanan data pada layanan cloud storage. *Jurnal Teknologi Informasi, 15*(1), 1–8.

IBM. (2022). *Cost of a Data Breach Report 2022*. IBM Security. https://www.ibm.com/reports/data-breach

Kaufman, L. M., Dorai, G., & Varadarajan, R. (2009). Data security in the world of cloud computing. *IEEE Security & Privacy, 7*(4), 61–64. https://doi.org/10.1109/MSP.2009.87

Kementerian Pendidikan dan Kebudayaan. (2023). *Data Pokok Pendidikan*. https://dapo.kemdikbud.go.id

Kumar, S., Singh, A., & Sharma, S. (2021). Secure and efficient cloud storage using hybrid AES encryption. *Journal of Cyber Security Technology, 5*(2), 91–105. https://doi.org/10.1080/23742917.2020.1841372

Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (1996). *Handbook of applied cryptography*. CRC Press.

Prasetya, A., & Wibowo, H. (2022). Evaluasi implementasi sistem enkripsi file lokal berbasis cloud storage. *Jurnal Informatika dan Komputer, 18*(3), 145–153.

Rahim, A., Nurhayati, S., & Zainal, M. (2020). Rancang bangun aplikasi enkripsi berbasis cloud menggunakan AES. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK), 7*(4), 681–688.

Setiawan, A., & Nugroho, H. (2022). Kajian keamanan data cloud storage dalam sistem pendidikan daring. *Jurnal Teknologi dan Sistem Komputer, 10*(3), 187–194.

Singh, S., Singh, S., & Sharma, S. (2013). Security analysis of AES and RSA algorithms. *International Journal of IT, Engineering and Applied Sciences Research, 2*(2), 70–73.

Stallings, W. (2017). *Cryptography and network security: Principles and practice* (7th ed.). Pearson Education.

Zhao, Y., & Liu, Z. (2020). A secure cloud storage model combining AES and double authentication. *IEEE Access, 8*, 191523–191534. https://doi.org/10.1109/ACCESS.2020.3032156