



Department of Digital Business

Journal of Artificial Intelligence and Digital Business (RIGGS)

Homepage: <https://journal.ilmudata.co.id/index.php/RIGGS>

Vol. 5 No. 2 (2026) pp: 8341-8363

P-ISSN: 2963-9298, e-ISSN: 2963-914X

Implementasi Algoritma AES-128 Untuk Keamanan Dokumen Soal Ujian di SMA Negeri 1 Kabawo

Irwan*, Alfiansyah Hasibuan, Kristofel Santa

Jurusan Teknik Informatika, Fakultas Teknik, Universitas Negeri Manado

*21210070@unima.ac.id, alfiansyahhasibuan@unima.ac.id, kristofelsanta@unima.ac.id

Abstrak

Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem pengamanan dokumen soal ujian berbasis web menggunakan algoritma Advanced Encryption Standard (AES)-128 di SMA Negeri 1 Kabawo. Permasalahan utama yang ditemukan adalah pengelolaan dokumen soal ujian masih dilakukan secara manual menggunakan media penyimpanan lokal dan aplikasi pesan instan tanpa perlindungan keamanan yang memadai. Kondisi tersebut menyebabkan tingginya risiko kebocoran, kehilangan, dan penyalahgunaan dokumen oleh pihak yang tidak berwenang. Penelitian ini menggunakan metode pengembangan sistem Waterfall yang meliputi analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan. Teknik pengumpulan data dilakukan melalui observasi, wawancara, studi dokumentasi, dan studi literatur. Sistem dikembangkan menggunakan framework CodeIgniter dengan basis data MySQL serta menerapkan algoritma AES-128 untuk proses enkripsi dan dekripsi dokumen. Hasil penelitian menunjukkan bahwa sistem mampu mengenkripsi dokumen menjadi ciphertext sehingga tidak dapat dibaca tanpa kunci dekripsi yang sesuai. Proses dekripsi juga berhasil mengembalikan dokumen ke bentuk asli tanpa kehilangan data. Pengujian menggunakan metode Blackbox Testing, Unit Testing, Integration Testing, dan User Acceptance Testing menunjukkan seluruh fitur sistem berjalan dengan baik, stabil, aman, serta mudah digunakan oleh pengguna sekolah. Sistem ini juga mampu membantu pihak sekolah dalam mengelola dokumen secara lebih terstruktur, efisien, dan terkontrol sehingga dapat meminimalkan potensi kebocoran soal ujian. Dengan demikian, implementasi AES-128 terbukti efektif dalam meningkatkan keamanan dan kerahasiaan dokumen soal ujian di SMA Negeri 1 Kabawo.

Kata kunci: AES-128, Kriptografi, Keamanan Dokumen, Soal Ujian, Sistem Berbasis Web.

1. Latar Belakang

Di era digital saat ini, keamanan data menjadi salah satu isu yang sangat penting, terutama dalam konteks pendidikan. Soal ujian merupakan dokumen yang bersifat rahasia karena berpengaruh langsung pada integritas proses penilaian siswa [1]. Perkembangan teknologi sangat penting bagi kehidupan manusia dalam semua aspeknya. Perkembangan ilmu teknologi khususnya dalam bidang komputer terkait dengan semakin banyaknya penggunaan sistem komputer [2].

Keamanan dokumen merupakan salah satu aspek penting dalam era digitalisasi, di mana semakin banyak informasi yang disimpan dan dibagikan secara elektronik. Oleh karena itu, perlindungan data dan keamanan informasi telah menjadi prioritas utama bagi organisasi di seluruh dunia [3]. Dengan adanya pencurian data maka sisi keamanan dalam pertukaran informasi dan penyimpanan data dianggap penting [4].

Isu kebocoran data semakin nyata seiring dengan meningkatnya ancaman terhadap keamanan informasi. Salah satu kasus adalah data pemilih Indonesia 2,3 juta yang bocor, yang berukuran 2,36 GB dan berisi data dalam format file PDF. Jika kebocoran seperti ini terjadi di lingkungan sekolah, misalnya kebocoran soal ujian, dampaknya bisa sangat merusak. Selain dapat menyebabkan kecurangan, hal ini juga bisa menurunkan integritas dan kepercayaan terhadap sistem pendidikan [5].

Ujian dapat dimaknai sebagai suatu evaluasi dari apa yang sudah di pelajari atau dilaksanakan sebelumnya, dalam dunia pendidikan ujian bertujuan sebagai tolak ukur mengetahui tingkat kemampuan siswa dari hasil proses belajar yang telah dilaksanakan [6]

Pendidikan di Indonesia pada masa ini telah masuk era teknologi, yang dimana hampir setiap sesuatunya dilakukan menggunakan gadget dan komputasi. Pada SMA Negeri 1 Kabawo data-data penting yang ada dalam sekolah tersebut masih berupa teks asli, yang dapat dibaca tanpa adanya pengamanan. Terkait dengan data penting yang dimana hanya pihak sekolah yang memiliki wewenang untuk mengetahui file-file tersebut karena file tersebut bersifat rahasia.

SMA Negeri 1 Kabawo merupakan sekolah yang berlokasi di kecamatan Kabawo, kabupaten Muna. Layaknya seperti sekolah lain, SMA Negeri 1 Kabawo setiap semester akan dilaksanakan ujian, karna file soal ujian bersifat internal dimana hanya pihak sekolah yang terkait saja yang memiliki wewenang untuk mengetahui file soal ujian tersebut.

Berdasarkan hasil observasi sementara di SMA Negeri 1 Kabawo, saat ini pengelolaan dokumen soal ujian masih bersifat manual. Pembuatan soal dilakukan oleh guru mata pelajaran menggunakan aplikasi Microsoft Word atau program pengolah dokumen lainnya di komputer pribadi. Setelah disusun, file disimpan di perangkat lokal tanpa sistem otorisasi akses atau perlindungan khusus. Pengumpulan file biasanya dilakukan melalui USB flashdisk atau pengiriman lewat aplikasi pesan instan seperti WhatsApp atau Telegram. Dokumen tersebut kemudian dikompilasi dan dicetak untuk digunakan dalam ujian. Pengelolaan yang masih bergantung pada penyimpanan fisik dan pengiriman manual ini sangat rentan terhadap kehilangan, kerusakan file, maupun penyalahgunaan akses.

Selain itu, SMA Negeri 1 Kabawo juga belum memiliki mekanisme kontrol terhadap akses file. Hal ini menyulitkan dalam memastikan siapa saja yang dapat melihat atau mengubah dokumen penting seperti soal ujian. Ketiadaan sistem otorisasi pengguna menyebabkan dokumen dapat dengan mudah disalin atau diakses oleh pihak yang tidak berwenang, baik secara sengaja maupun tidak disengaja. Dengan tidak adanya sistem keamanan yang efektif, potensi kebocoran atau manipulasi soal menjadi sangat besar, yang pada akhirnya dapat merusak keadilan dan kepercayaan dalam proses penilaian siswa.

Permasalahan-permasalahan tersebut mencerminkan bahwa pengelolaan dokumen di SMA Negeri 1 Kabawo belum sepenuhnya memenuhi standar keamanan dan efisiensi yang diperlukan dalam menjaga kerahasiaan soal ujian. Ketergantungan pada metode manual, tidak adanya kontrol akses yang memadai, serta lemahnya sistem perlindungan terhadap dokumen penting menandakan perlunya pendekatan baru yang lebih terstruktur dan terintegrasi. Salah satu solusi yang dapat diimplementasikan adalah pengembangan sistem pengamanan dokumen soal ujian berbasis web, yang tidak hanya memungkinkan penyimpanan file secara digital, tetapi juga menyediakan mekanisme otorisasi pengguna untuk mengatur hak akses secara jelas. Dengan adanya sistem ini, guru dan pihak sekolah dapat mengelola dokumen dengan lebih fleksibel dan aman, tanpa bergantung pada media penyimpanan fisik yang rentan terhadap kehilangan atau penyalahgunaan.

Namun, sistem web saja belum cukup menjamin kerahasiaan isi dokumen. Oleh karena itu, diperlukan lapisan keamanan tambahan berupa enkripsi data, agar isi dokumen tidak dapat dibaca meskipun file berhasil diakses oleh pihak tidak berwenang. Salah satu metode enkripsi yang terbukti efektif adalah algoritma *Advanced Encryption Standard (AES)*, khususnya versi *AES-128*. Algoritma ini mampu menyamarkan isi file sehingga hanya dapat dibuka oleh pihak yang memiliki kunci dekripsi. Selain ringan dan cepat, *AES-128* juga mendukung berbagai format file seperti *.docx*, *.txt*, *.xlsx*, *.pdf*, dan *.pptx* [7].

Kriptografi merupakan bidang ilmu yang mempelajari tentang metode untuk mengirim pesan secara rahasia (di-enkripsi atau disamarkan) sehingga hanya penerima pesan yang dituju yang dapat menghapus penyamaran dan membaca pesan [8]. Kriptografi menggunakan banyak algoritma untuk menyamarkan data. *Advanced Encryption Standard* atau AES merupakan salah satu jenis kriptografi. AES merupakan *cipher* blok simetris yang dapat mengenkripsi data atau bisa disebut *encipher*, dan dekripsinya disebut *decipher*. [9]

Algoritma *Advanced Encryption Standard (AES)* telah terbukti efektif dalam mengamankan data dan informasi penting. Penggunaan AES sebagai metode enkripsi dapat memberikan lapisan perlindungan tambahan bagi dokumen soal ujian, sehingga hanya pihak-pihak yang berwenang yang dapat mengakses dan menggunakan dokumen tersebut. Melalui penerapan sistem pengamanan dokumen soal ujian menggunakan AES, diharapkan dapat meningkatkan kepercayaan stakeholders, termasuk siswa, guru, dan orang tua, terhadap proses penilaian yang lebih adil dan transparan.

Penelitian mengenai kriptografi *AES-128* pernah dilakukan oleh [10], Menurutnya kriptografi dengan metode *Advanced Encryption Standard (AES-128)* dapat meningkatkan keamanan data karena isi data telah diacak dengan menggunakan huruf, angka, simbol, atau kombinasi ketiganya.

Sedangkan pada penelitian lain mengemukakan bahwa Algoritma *Advanced Encryption Standard* dapat digunakan dalam melakukan pengamanan data pemesanan produk dengan melakukan enkripsi setiap data pemesanan produk yang diinput oleh customer menjadi bentuk kode-kode yang tidak dapat diketahui [11],

pada penelitian sebelumnya juga dilakukan oleh [12] menunjukkan bahwa algoritma *AES-128* memiliki performa lebih unggul dibandingkan *RSA* dan *Triple DES*. Data pengujian menunjukkan bahwa *AES* memiliki waktu enkripsi 12 ms dan dekripsi 15 ms dengan konsumsi sumber daya rendah, sedangkan *RSA* membutuhkan waktu enkripsi 98 ms dan dekripsi 120 ms dengan konsumsi sumber daya tinggi, serta *Triple DES* memerlukan waktu enkripsi 55 ms dan dekripsi 65 ms dengan konsumsi sumber daya sedang. Berdasarkan perbandingan tersebut, *AES-128* lebih cepat dan efisien sehingga lebih sesuai digunakan untuk pengamanan dokumen.

2. Metode Penelitian

2.1 Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini menggunakan Algoritma *AES-128*. Penelitian ini bertujuan untuk merancang dan membangun sebuah system yang dapat mengamankan dokumen soal ujian melalui mekanisme enkripsi dan dekripsi menggunakan algoritma kriptografi *AES-128*.

Jenis metode ini dipilih karena penelitian difokuskan pada proses perancangan dan implementasi sistem informasi yang mampu memberikan solusi terhadap permasalahan keamanan data. Proses pengembangan sistem dilakukan menggunakan model *Waterfall*, yang terdiri dari tahapan-tahapan sistematis, yaitu: analisis kebutuhan, perancangan sistem, implementasi, pengujian, penerapan, dan pemeliharaan sistem.

Algoritma *AES-128* digunakan sebagai metode enkripsi untuk menjaga kerahasiaan dokumen soal ujian. Pada saat dokumen diunggah ke sistem oleh admin atau guru, dokumen akan melalui proses enkripsi. Sebaliknya, saat dokumen akan diakses oleh pengguna yang memiliki otorisasi, dokumen akan melalui proses dekripsi. Dengan menerapkan metode ini, diharapkan dokumen soal ujian dapat terlindungi dari ancaman kebocoran atau penyalahgunaan data.

2.2. Teknik Pengumpulan Data

Teknik pengumpulan data dalam penelitian ini dilakukan melalui empat pendekatan utama, yaitu observasi, wawancara, studi dokumentasi, dan studi literatur.

1. Observasi

Observasi dilakukan dengan mengamati langsung proses pengelolaan dokumen ujian di SMA Negeri 1 Kabawo. Tujuannya adalah untuk memahami alur kerja manual yang digunakan, termasuk cara penyimpanan, distribusi, dan akses dokumen, serta mengidentifikasi kelemahan atau celah keamanan yang ada.

2. Wawancara

Wawancara dilakukan dengan guru dan staf administrasi untuk mendapatkan informasi tentang kebutuhan sistem pengamanan dokumen, tantangan dalam pengelolaan dokumen, dan harapan mereka terhadap pengembangan sistem.

3. Studi Dokumentasi

Studi dokumentasi melibatkan pengumpulan dokumen soal ujian sebagai bahan uji coba sistem, dengan format yang mencakup *.docx*, *.pdf*, dan *.xlsx*. Dokumentasi lain, seperti prosedur pengelolaan dokumen di sekolah, juga dikumpulkan untuk mendukung analisis kebutuhan sistem.

4. Studi Literatur

Studi literatur dilakukan untuk memperkuat landasan teoretis dan teknis terkait algoritma *Advanced Encryption Standard* (AES) dan sistem keamanan data. Literatur yang digunakan meliputi jurnal ilmiah, buku, dan artikel dari sumber terpercaya.

2.3. Landasan Teori

2.3.1. Pengamanan Dokumen

Pengamanan dokumen mencakup teknik enkripsi untuk melindungi data, serta autentikasi untuk memastikan bahwa hanya pengguna yang sah dapat mengakses dokumen. Sistem pengamanan ini juga melibatkan deteksi intrusi dan pengendalian akses yang tepat untuk memastikan bahwa dokumen tidak jatuh ke tangan yang salah [13].

2.3.2. Kriptografi

a. Definisi Kriptografi

Menurut Ariyus (2008) kriptografi merupakan langkah-langkah logis bagaimana menyembunyikan pesan dari orang-orang yang tidak berhak atas pesan tersebut [14].

Menurut Menez kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, serta otentikasi [15].

Dalam ilmu kriptografi, terdapat dua buah proses yaitu melakukan enkripsi dan dekripsi. Enkripsi berarti proses penyembunyian data pesan, mengubah *plaintext* menjadi *chipertext*. Sedangkan dekripsi merupakan kebalikan dari enkripsi, bertujuan untuk memahami pesan yang ada, dan kunci adalah teknik yang digunakan untuk enkripsi maupun dekripsi [16].

b. Sejarah Kriptografi

Catatan sejarah kuno menunjukkan bahwa kriptografi pertama (yang ditemukan) digunakan oleh orang Mesir kuno pada awal tahun 3000 SM. Ini dimulai pada masa kejayaan Yunani oleh Spartan sekitar 400 SM. Pada saat itu, alat yang digunakan untuk membuat pesan tersembunyi disebut *Scytale*. *Scytale* berbentuk batang silinder dengan delapan belas huruf dan pita panjang yang terbuat dari bahan papirus, dan menggulung pita untuk membaca pesan. Pada abad ke-15 M, orang Cina dan Jepang mulai mengenali kriptografi [17].

c. Fungsi Kriptografi

Cara baru untuk mencegah serangan seperti penyadapan dan perubahan data yang dikirimkan terus dikembangkan dalam teknologi informasi. Kriptografi, yang menggunakan transformasi data, adalah cara untuk mengatasi masalah ini. Ini memastikan bahwa pihak yang tidak berhak mengakses tidak dapat memahami data yang dihasilkan.

Transformasi ini menyelesaikan dua jenis masalah keamanan data: privasi (*privacy*) dan keautentikan. Privasi berarti bahwa informasi yang dikirim hanya dapat dipahami oleh penerima yang sah atau berhak. Namun, keotentikan mencegah pihak ketiga untuk menyampaikan informasi yang salah atau mengubahnya.

Oleh karena itu, diharapkan bahwa kriptografi dapat memenuhi syarat-syarat berikut [18]:

- Kerahasiaan (*confidentiality*), yang dapat dicapai melalui adanya enkripsi (penyandian).
- Keutuhan (*integrity*) atas data-data pembayaran, yang dapat terjamin dengan penggunaan fungsi hash. Fungsi hash adalah fungsi yang secara efisien mengubah string input dengan panjang berhingga menjadi string output dengan panjang tetap yang disebut nilai hash.
- Jaminan atas identitas dan keabsahan (*authenticity*) pihak-pihak yang melakukan transaksi, yang terjamin dengan digunakannya password atau sertifikat digital. Sedangkan keautentikan data transaksi dapat terjamin dengan menggunakan tanda tangan digital.

- d. Dengan menggunakan tanda tangan digital dan sertifikat digital, transaksi dapat digunakan sebagai barang bukti yang tidak bisa disangkal (*non-repudiation*).

d. Jenis Kriptografi

Algoritma kriptografi dibagi menjadi dua yaitu [19]:

a. Kriptografi Simetris

Kriptografi simetris menggunakan kunci yang sama untuk proses enkripsi dan dekripsi. Metode ini disebut juga *secret key cryptography*.

Cara kerja kriptografi simetris dimulai ketika pihak sender mengirim data dengan format *plaintext* (file normal) kemudian dilakukan proses enkripsi dengan suatu kunci. File yang telah dienkripsi tersebut berubah menjadi file *ciphertext* (file yang diacak). Setelah *ciphertext* sampai di pihak *recipient*, *recipient* mendekripsi file *ciphertext* tersebut menggunakan kunci yang sama yang digunakan oleh pihak sender menjadi *plaintext*.

b. Non Simetris

Enkripsi non simetris menggunakan dua kunci yang berbeda untuk proses enkripsi dan dekripsi. Proses enkripsi menggunakan *public key*, proses dekripsi menggunakan *private key*.

Cara kerja kriptografi non simetris dimulai ketika pihak sender mengirim data dengan format *plaintext* (file normal) kemudian dilakukan proses enkripsi dengan suatu kunci A (*recipient public key*). File yang telah dienkripsi tersebut berubah menjadi file *ciphertext* (file yang diacak). Setelah *ciphertext* sampai di pihak *recipient*, *recipient* mendekripsi file *ciphertext* tersebut menggunakan kunci B (*recipient private key*) menjadi *plaintext*. Kunci B dibuat menggunakan algoritma tertentu yang dibangkitkan berdasarkan kunci A.

e. Unsur Kriptografi

Ilmu kriptografi terdiri dari beberapa unsur penting, [20] yaitu :

a. Pesan (*Plaintext* dan *Ciphertext*)

Data atau informasi yang dapat dibaca dan dimengerti disebut pesan. Pesan asli dikenal sebagai *plainteks* (*plaintext*) atau teks-jelas (*cleartext*). Pesan yang sudah disandikan dikenal sebagai *cipherteks* (*chipertext*).

b. Pengirim dan Penerima

Komunikasi data terjadi ketika dua entitas bertukar pesan satu sama lain. Penerima adalah entitas yang menerima pesan, dan pengirim adalah entitas yang mengirimkannya.

c. Penyadap (*eavesdropper*)

orang yang mencoba menangkap pesan selama ditransmisikan

d. Kriptanalisis dan Kriptologi

Kriptanalisis (*cryptanalysis*) adalah ilmu dan seni untuk memecahkan *chiperteks* menjadi *plainteks* tanpa mengetahui kunci yang digunakan. Pelakunya disebut kriptanalisis. Kriptologi (*cryptology*) adalah studi mengenai kriptografi dan kriptanalisis.

e. Enkripsi dan Dekripsi

Proses menyandikan *plainteks* menjadi *cipherteks* disebut enkripsi, atau enkripsi. Proses mengembalikan *cipherteks* menjadi *plainteks* semula disebut dekripsi, atau *deciphering*.

f. Cipher dan Kunci

Aturan untuk enkripsi dan dekripsi, atau fungsi matematika yang digunakan untuk enkripsi dan dekripsi, adalah nama lain algoritma kriptografi. Parameter yang digunakan untuk *enciphering* dan *dechipering* adalah kunci. Kunci biasanya terdiri dari string atau deretan angka.

f. Enkripsi dan Dekripsi

Enkripsi merupakan sebuah metode penyandian sebuah pesan atau informasi menjadi sebuah teks yang tidak dapat dibaca [14].

Dekripsi adalah proses mengembalikan cipherteks menjadi plainteks semula [21]. Proses yang dilakukan untuk mengubah plainteks ke dalam cipherteks disebut enkripsi (*encryption* atau *encipherment*). Sedangkan proses untuk mengubah ciphertext kembali ke plaintext disebut dekripsi (*decryption* atau *decipherment*). Secara sederhana istilah-istilah di atas dapat digambarkan sebagai berikut [18]:

3. Hasil dan Diskusi

3.1. Hasil Analisis Kebutuhan

Tahapan analisis kebutuhan bertujuan untuk mengidentifikasi kondisi nyata di SMA Negeri 1 Kabawo terkait pengelolaan dokumen soal ujian, serta menentukan kebutuhan sistem keamanan dokumen yang akan dikembangkan. Analisis ini dilakukan dengan dua pendekatan, yaitu observasi langsung di lapangan dan wawancara dengan pihak sekolah.

3.1.1. Hasil Observasi

Metode Observasi dilakukan secara langsung di lingkungan SMA Negeri 1 Kabawo untuk mengetahui bagaimana proses pembuatan, penyimpanan, dan distribusi dokumen soal ujian dilakukan sebelum adanya sistem pengamanan berbasis enkripsi. Dari hasil observasi, diperoleh beberapa temuan berikut:

1. Proses pengelolaan dokumen soal ujian masih manual.
Guru membuat soal ujian menggunakan *Microsoft Word* dan menyimpannya di komputer pribadi atau flashdisk tanpa perlindungan enkripsi atau kata sandi.
2. Distribusi dokumen menggunakan media tidak aman.
Pengiriman file dilakukan melalui *WhatsApp* atau *flashdisk*, sehingga berpotensi tinggi terhadap kebocoran data jika file jatuh ke tangan yang tidak berwenang.
3. Tidak terdapat sistem otorisasi pengguna.
File soal yang telah disimpan dapat dibuka oleh siapa saja yang memiliki salinannya, tanpa mekanisme *login* atau autentikasi untuk membatasi hak akses.
4. Tidak ada sistem cadangan (backup) dokumen.
File soal hanya disimpan di satu perangkat, sehingga apabila terjadi kerusakan perangkat, dokumen berisiko hilang.
5. Keamanan data rendah.
Dokumen masih berbentuk *plaintext* dan belum terlindungi oleh sistem kriptografi, sehingga mudah dibaca dan dimodifikasi tanpa izin.

Berdasarkan hasil observasi ini, dapat disimpulkan bahwa pengelolaan dokumen soal ujian di SMA Negeri 1 Kabawo masih memiliki celah keamanan yang besar dan sangat membutuhkan sistem berbasis kriptografi untuk menjaga kerahasiaan, integritas, dan otentikasi data.

3.1.2. Hasil Wawancara

Wawancara dilakukan dengan pihak-pihak yang terlibat langsung dalam pengelolaan soal ujian di sekolah, seperti kepala sekolah, guru mata pelajaran, dan staf tata usaha. Tujuannya adalah untuk mengetahui pandangan mereka tentang sistem pengamanan dokumen serta kebutuhan yang diharapkan.

Kesimpulan Wawancara:

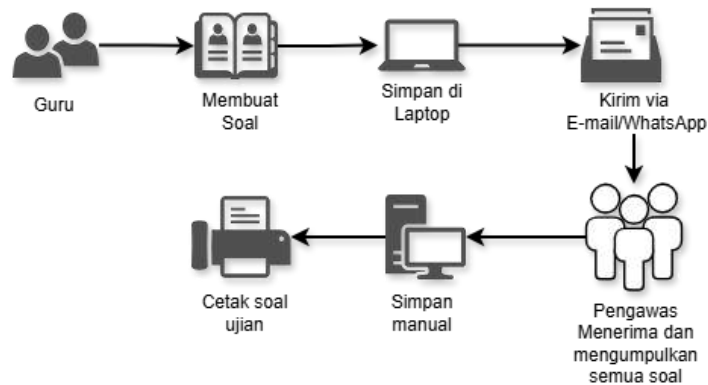
Dari hasil wawancara diperoleh informasi bahwa pihak sekolah mengharapkan adanya sistem pengamanan dokumen berbasis komputer yang mampu:

- Membatasi akses hanya kepada pengguna berwenang.
- Menyediakan proses enkripsi dan dekripsi untuk menjaga kerahasiaan dokumen.
- Mempermudah penyimpanan serta pengiriman file secara digital dengan aman.

3.1.3. Analisis Alur Proses

a. Alur Proses Sebelum Ada Sistem

Berikut merupakan gambar alur proses sebelum adanya system di sekolah SMA Negeri 1 Kabawo

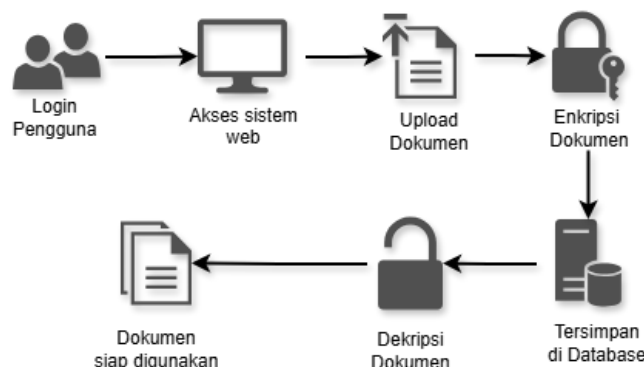


Gambar 1. Alur Proses Sebelum Adanya Sistem

Pada gambar 1, menunjukan bahwa guru terlebih dahulu membuat soal menggunakan aplikasi pengolah kata seperti Microsoft Word. Setelah selesai, dokumen soal disimpan di laptop atau perangkat pribadi lainnya tanpa perlindungan keamanan seperti enkripsi atau kata sandi. Selanjutnya, file soal dikirim kepada pengawas atau panitia ujian melalui media digital yang tidak aman, seperti email atau WhatsApp. Pengawas kemudian menerima seluruh file soal dari masing-masing guru dan mengumpulkannya secara manual. Dokumen yang diterima disimpan kembali secara manual tanpa adanya kontrol akses maupun mekanisme pembatasan hak pengguna, sehingga siapa pun yang memiliki file tersebut dapat membuka dan mengubahnya. Pada tahap akhir, dokumen-dokumen tersebut dicetak untuk kebutuhan pelaksanaan ujian.

b. Alur Proses Sesudah Ada Sistem

Berikut merupakan gambar alur proses sebelum adanya system di sekolah SMA Negeri 1 Kabawo



Gambar 2 Alur Proses Sesudah Adanya Sistem

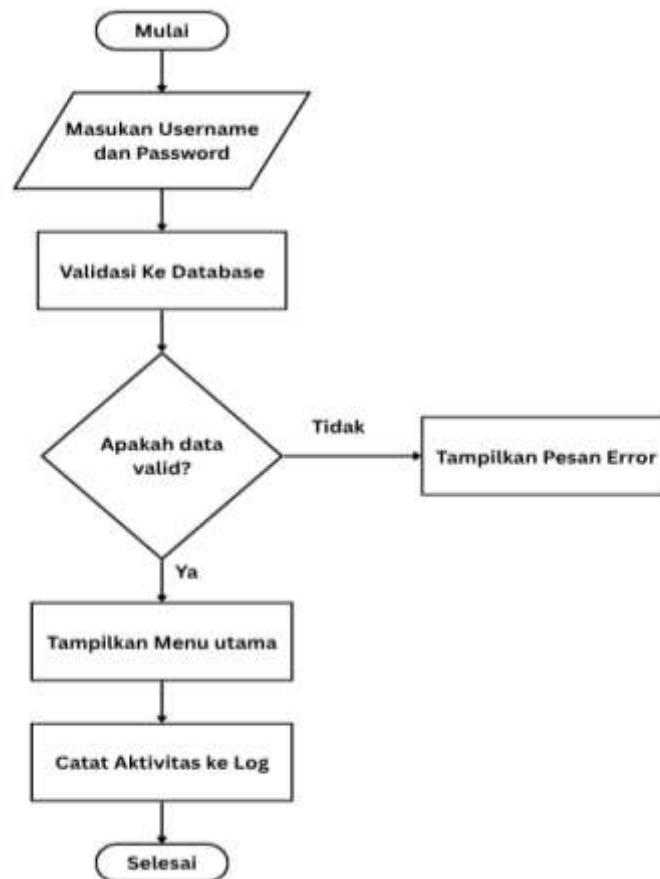
Pada gambar 2, menunjukan bahwa pengguna terlebih dahulu melakukan login menggunakan akun yang telah terdaftar untuk memastikan bahwa hanya pihak berwenang yang dapat mengakses aplikasi. Setelah berhasil masuk, pengguna dapat mengakses sistem web dan memilih menu untuk mengunggah dokumen soal ujian. Setiap dokumen yang di-upload akan diproses oleh sistem melalui mekanisme enkripsi, sehingga file yang tersimpan tidak lagi dalam bentuk asli, melainkan telah terlindungi. Hasil enkripsi kemudian disimpan secara aman di dalam database, sehingga hanya dapat diakses melalui sistem. Ketika dokumen diperlukan, pengguna dapat melakukan proses dekripsi melalui menu yang telah disediakan, sesuai hak akses dan kunci keamanan yang dimiliki. Setelah proses dekripsi berhasil, dokumen kembali ke bentuk asli dan siap digunakan untuk kebutuhan pencetakan atau pemeriksaan.

3.2. Perancangan Sistem

Tahapan ini bertujuan untuk merancang struktur sistem keamanan dokumen soal ujian agar sesuai dengan hasil analisis kebutuhan.

1. Diagram Alir Sistem (*Flowchart*)

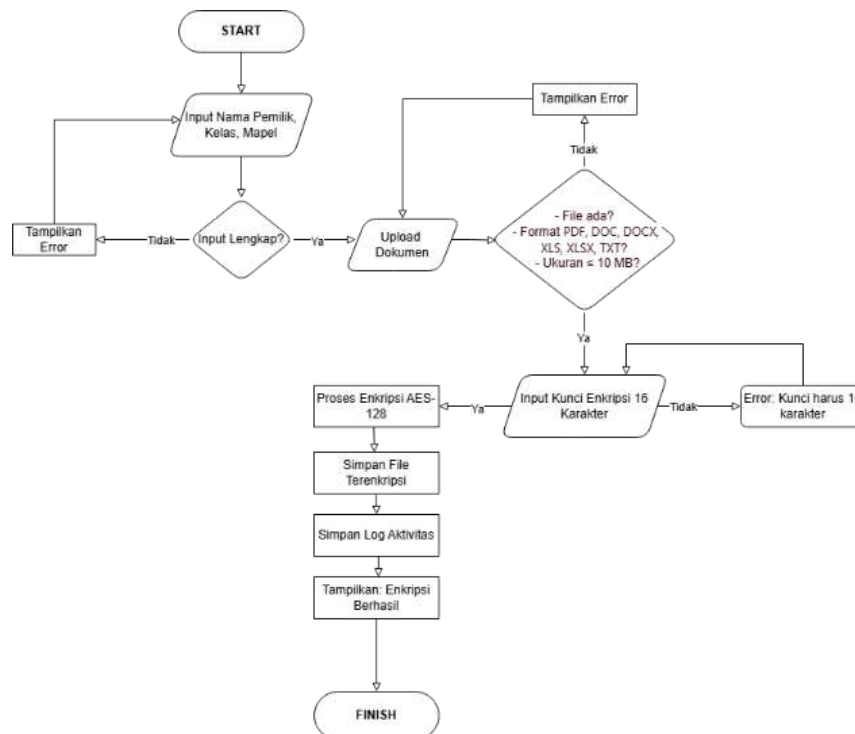
a) *Flowchart* Proses Login



Gambar 3. *Flowchart* Proses Login

Flowchart login pada Gambar 4.3 menggambarkan proses autentikasi pengguna. Pengguna (admin/guru) memasukkan *username* dan *password*, lalu sistem melakukan validasi data ke database. Jika data valid, pengguna diarahkan ke menu utama, dan aktivitas *login* dicatat dalam log aktivitas. Jika tidak valid, sistem menampilkan pesan kesalahan.

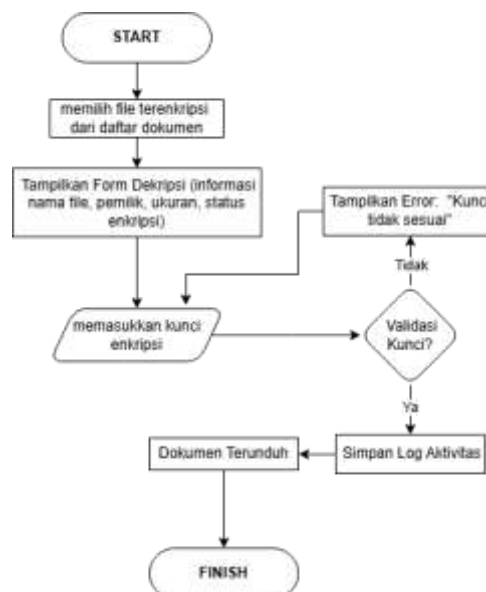
b) *Flowchart Form Enkripsi*



Gambar 4. *Flowchart Form Enkripsi*

Flowchart Form Enkripsi pada gambar 4.4 menjelaskan alur ketika pengguna mengirimkan dokumen untuk dienkripsi. Pengguna terlebih dahulu mengisi data wajib seperti nama pemilik, kelas, dan mata pelajaran, kemudian mengunggah file yang akan diverifikasi format dan ukurannya. Selanjutnya pengguna memasukkan kunci enkripsi sepanjang 16 karakter serta keterangan opsional. Setelah seluruh data dinyatakan valid, sistem menjalankan proses enkripsi dokumen menggunakan algoritma *AES-128*, menyimpan hasil enkripsi bersama metadata, serta mencatat aktivitas ke log. Terakhir, sistem menampilkan pemberitahuan bahwa proses enkripsi berhasil.

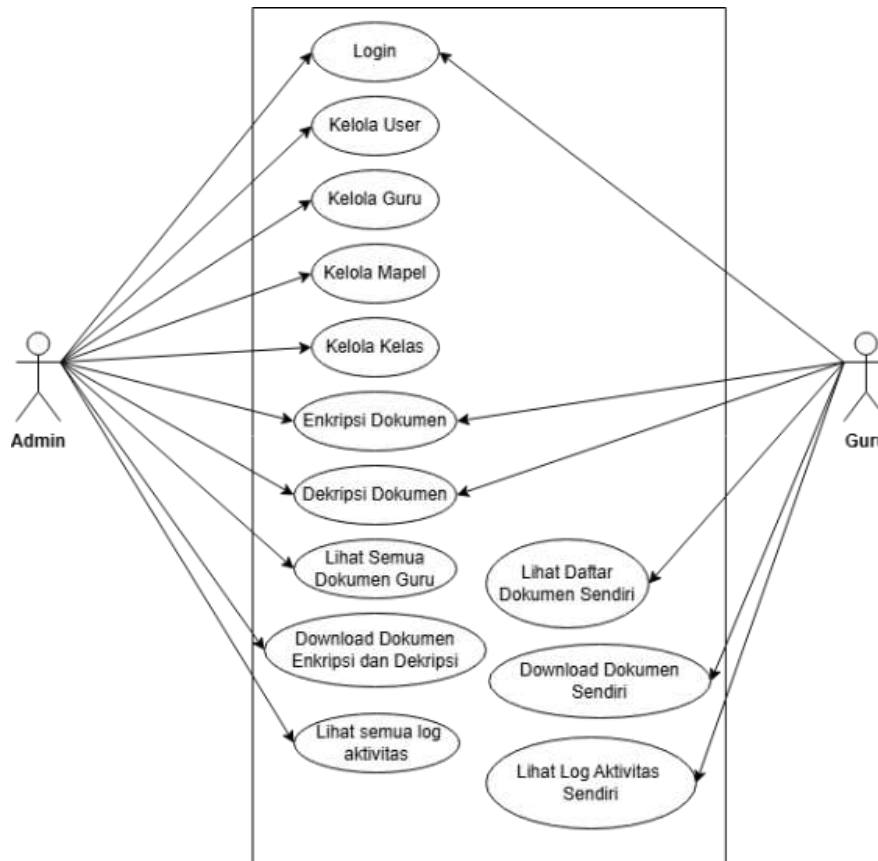
c) *Flowchart Form Dekripsi*



Gambar 4. *Flowchart Form Dekripsi*

Flowchart pada gambar 5. menggambarkan proses ketika pengguna mendekripsi dokumen. Pengguna memilih file yang masih terenkripsi, lalu sistem menampilkan form dekripsi beserta informasi dokumen. Setelah itu pengguna memasukkan kunci enkripsi dan sistem memvalidasinya. Jika kunci salah, sistem menampilkan pesan error. Jika benar, sistem melakukan dekripsi menggunakan AES-128 dan mencatat aktivitas dekripsi ke dalam log sistem. Setelah proses berhasil, sistem menyediakan file hasil dekripsi untuk diunduh oleh pengguna.

2. Use Case Diagram



Gambar 5. Use Case Diagram Admin dan Guru

Dari gambar 6 merupakan diagram use case dari sistem aplikasi pengamanan dokumen soal ujian yang penulis kembangkan. Pada diagram ini menggambarkan interaksi antara dua aktor (admin dan guru). Fitur-fitur utama yang dimiliki Admin dan Guru sebagai berikut.

a. Admin memiliki fitur yang lebih lengkap karena bertindak sebagai pengelola sistem. Berikut fitur-fiturnya:

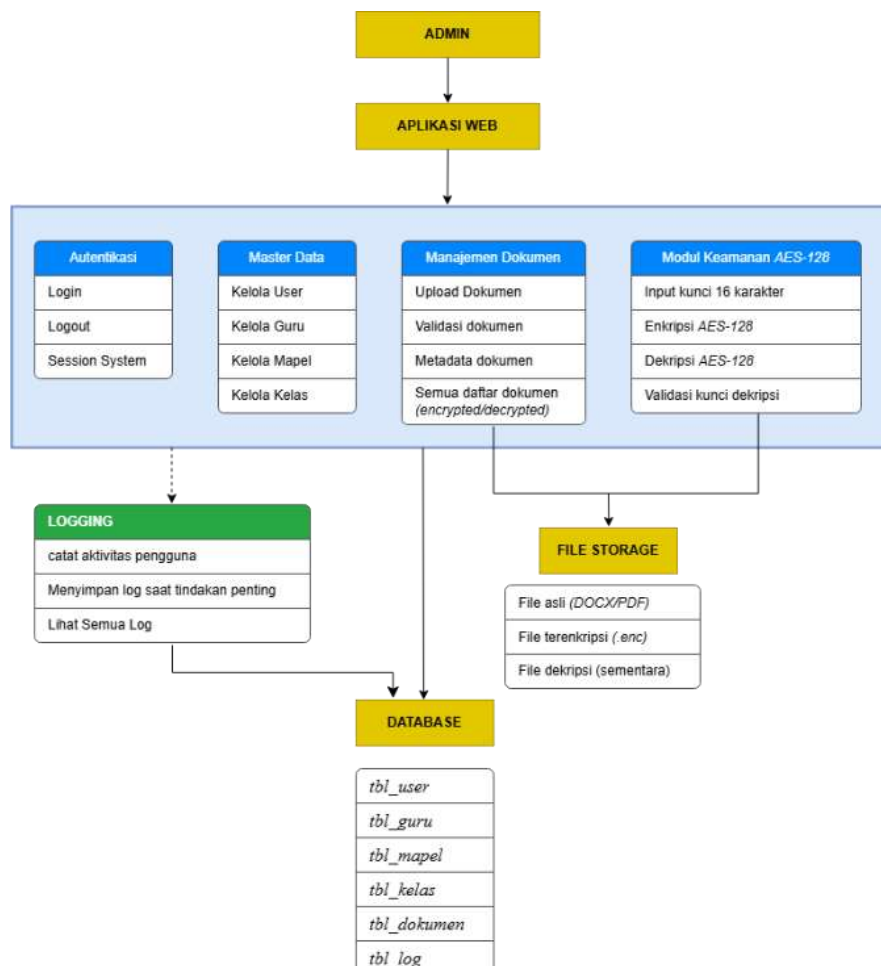
1. Admin masuk ke sistem menggunakan username dan password.
2. Mengelola user seperti tambah akun user, edit data akun user, dan menghapus akun user.
3. Mengelola data guru seperti Menambahkan data guru, Edit data guru, dan Menghapus data guru.
4. Mengelola data mata Pelajaran seperti tambah mata Pelajaran, edit mata Pelajaran, dan hapus mata Pelajaran.
5. Mengelola data kelas seperti tambah kelas, edit data kelas, dan hapus data kelas.
6. Enkripsi dokumen
7. Dekripsi dokumen
8. Melihat semua dokumen soal guru dalam bentuk terenkripsi dan terdekripsi.
9. Download dokumen dalam bentuk terenkripsi dan terdekripsi.

10. Melihat semua log aktivitas yang ada pada system seperti pengguna yang login dan pengguna yang melakukan enkripsi dan dekripsi.
- b. Guru memiliki fitur yang terbatas karena bertindak sebagai pengguna utama sistem yang memiliki fokus pada proses pengelolaan soal, Berikut fitur-fiturnya:
 1. Guru masuk ke sistem menggunakan username dan password.
 2. Enkripsi dokumen
 3. Dekripsi dokumen
 4. Melihat daftar dokumen sendiri dalam bentuk terenkripsi dan terdekripsi.
 5. Download dokumen sendiri dalam bentuk terenkripsi dan terdekripsi.
 6. Melihat log aktivitas sendiri yang ada pada system seperti pengguna yang login dan pengguna yang melakukan enkripsi dan dekripsi.

3. Arsitektur Sistem

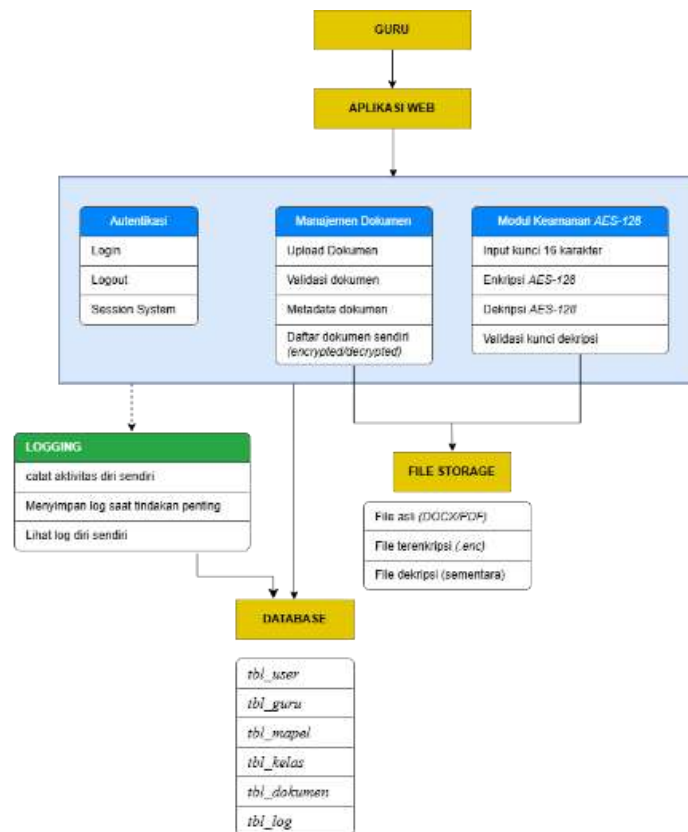
Perancangan dilakukan berdasarkan kebutuhan yang telah diidentifikasi. Sistem dirancang menggunakan bahasa pemrograman PHP (CodeIgniter Framework) dengan basis data MySQL dan dijalankan pada server XAMPP.

a. Arsitektur Sistem Admin



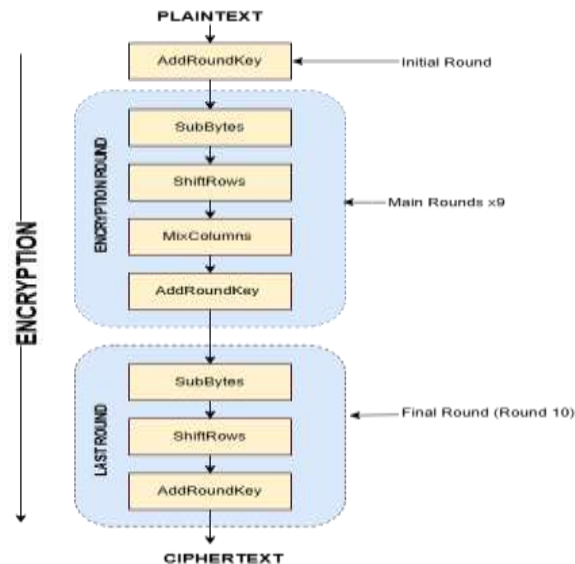
Gambar 6. Arsitektur Sistem Admin

b. Arsitektur Sistem Guru



Gambar 7. Arsitektur Sistem Guru

c. Arsitektur Algoritma AES-128



Gambar 8. Arsitektur Algoritma AES-128

Pada gambar 9. menunjukkan bahwa proses enkripsi AES-128 dimulai dari *plaintext*, yang terlebih dahulu melewati *Initial Round* dengan operasi *AddRoundKey* untuk menggabungkan data dengan kunci awal. Selanjutnya, data masuk ke 9 *Main Rounds*, masing-masing terdiri dari *SubBytes* (*substitusi byte*), *ShiftRows* (*pergeseran baris*), *MixColumns* (*pencampuran kolom*), dan *AddRoundKey* (*penggabungan dengan round key*). Setelah itu, data melalui *Final Round* yang hanya melakukan *SubBytes*, *ShiftRows*, dan *AddRoundKey* tanpa *MixColumns*. Hasil akhirnya adalah *ciphertext 128-bit*, yaitu data terenkripsi yang siap disimpan atau dikirim dengan aman.

3.3. Implementasi Sistem

Implementasi sistem dilakukan menggunakan bahasa pemrograman *PHP* dengan *framework CodeIgniter* sebagai *backend*, sedangkan *MySQL* digunakan sebagai basis data untuk menyimpan informasi pengguna, metadata dokumen, serta log aktivitas. Sistem ini mengimplementasikan algoritma *Advanced Encryption Standard (AES-128)* sebagai metode enkripsi utama. Implementasi dilakukan pada sisi *backend*, basis data, antarmuka pengguna, serta pemrosesan file.

3.3.1. Implementasi Enkripsi Algoritma AES-128

Implementasi enkripsi pada sistem dimulai dari pendefinisian kelas *AESEncryption*, yang menjadi pusat seluruh proses enkripsi dan manajemen kunci. Pada bagian awal kode, kelas ini menetapkan jenis cipher yang akan digunakan, yaitu *"aes-128-cbc"*. Penetapan ini sangat penting karena menentukan skema blok, panjang kunci, serta mekanisme chaining yang digunakan dalam proses enkripsi. Secara khusus, mode *CBC* dipilih karena mampu memutus pola data sehingga tiap blok *ciphertext* bergantung pada blok sebelumnya, meningkatkan keamanan terhadap analisis pola.

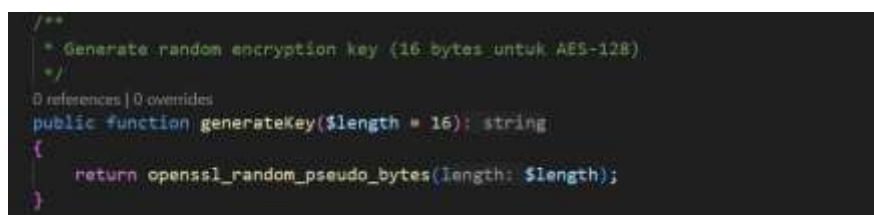


```
4 references
private $cipher = "aes-128-cbc";
```

Gambar 10. Potongan kode deklarasi kelas dan properti *\$cipher*

Pada potongan kode di gambar 10 tersebut terlihat bahwa properti *\$cipher* berfungsi sebagai konfigurasi utama yang akan dipanggil oleh seluruh fungsi dalam kelas. Penempatan konfigurasi cipher sebagai properti terpisah juga membuat kode lebih fleksibel dan memudahkan perubahan algoritma jika suatu saat dibutuhkan. Ini menunjukkan bahwa desain library telah memperhatikan aspek modularitas serta pemeliharaan jangka panjang.

Selanjutnya, proses enkripsi membutuhkan sebuah kunci acak. Fungsi *generateKey()* digunakan untuk menghasilkan kunci sepanjang 16 byte, sesuai standar AES-128. Fungsi ini memanfaatkan *openssl_random_pseudo_bytes()* yang mampu menghasilkan data acak berkualitas kriptografis sehingga tidak mudah diprediksi. Kunci ini nantinya digunakan dalam proses enkripsi maupun dekripsi agar data dapat dikelola secara simetris dengan aman.



```
/**
 * Generate random encryption key (16 bytes untuk AES-128)
 */
0 references | 0 overrides
public function generateKey($length = 16): string
{
    return openssl_random_pseudo_bytes($length);
}
```

Gambar 11. Potongan kode fungsi *generateKey()*

Jika diperhatikan pada gambar 11 tersebut, fungsi menerima parameter panjang kunci dengan nilai default 16 byte. Hal ini disesuaikan dengan standar AES-128. Penggunaan fungsi bawaan *OpenSSL* memastikan bahwa keamanan kunci lebih terjamin dibandingkan pembuatan kunci manual. Dengan demikian, proses pembentukan kunci telah memenuhi standar keamanan kriptografi modern.

Tahapan berikutnya adalah pembentukan *Initialization Vector (IV)* melalui fungsi *privat generateIV()*. *IV* berfungsi menambahkan randomisasi pada blok pertama enkripsi sehingga meskipun plaintext sama dienkripsi berulang kali, *ciphertext* tetap berbeda. Fungsi ini menghitung panjang *IV* berdasarkan *cipher* yang digunakan sehingga panjangnya selalu valid dan sesuai konfigurasi algoritma *AES-128-CBC*.

```
/**
 * Generate random IV (Initialization Vector)
 */
// reference
private function generateIV(): string
{
    return openssl_random_pseudo_bytes(length: openssl_cipher_iv_length(cipher_algo: $this->cipher));
}
```

Gambar 12. Potongan kode fungsi *generateIV()*

Pada gambar 4.19 terlihat bahwa sistem secara otomatis menyesuaikan panjang *IV* menggunakan *openssl_cipher_iv_length()*. Pendekatan ini menghindari kesalahan manual dalam menentukan ukuran *IV* serta memastikan nilai *IV* selalu aman digunakan. Karena *IV* bersifat unik dan tidak harus dirahasiakan, nilai ini nantinya digabungkan bersama *ciphertext* sebelum disimpan di database.

Proses enkripsi utama dilakukan oleh fungsi *encrypt()*, yang menerima data *plaintext* beserta *key* yang digunakan. Saat fungsi dipanggil, sistem terlebih dahulu membuat *IV* baru, kemudian memproses *plaintext* menggunakan *openssl_encrypt()* dengan metode *AES-128-CBC*. *Output* enkripsi kemudian digabungkan dengan *IV* dan di-*encode* menggunakan *Base64* agar hasilnya dapat disimpan pada database atau ditransmisikan melalui jaringan, perhatikan gambar berikut.

```
public function encrypt($data, $key): bool|string
{
    try {
        if (strlen($key) != 16) {
            throw new \Exception('Key harus 16 bytes untuk AES-128');
        }
        $iv = $this->generateIV();
        // Encrypt data
        $encrypted = openssl_encrypt(
            $data,
            $this->cipher,
            $key,
            OPENSSL_RAW_DATA,
            $iv
        );
        if ($encrypted == false) {
            throw new \Exception('Enkripsi gagal');
        }
        $result = $iv . $encrypted;
        // Return dalam format base64 untuk penyimpanan
        return base64_encode($result);
    } catch (\Exception $e) {
        log_message('error', 'AES Encryption Error: ' . $e->getMessage());
        return false;
    }
}
```

Gambar 13. Potongan kode fungsi *encrypt()*

Potongan kode pada gambar 13 tersebut menunjukkan secara jelas bagaimana *IV* diletakkan di awal *ciphertext* sebelum di-*encode*. Teknik ini penting karena *IV* harus tersedia saat proses dekripsi. Selain itu, penggunaan *Base64* memastikan *ciphertext* tidak mengandung karakter yang tidak aman untuk penyimpanan berbasis teks. Dengan demikian, seluruh proses enkripsi memenuhi tahapan *AES* yang benar: pembentukan kunci → pembentukan *IV* → pembangkitan *ciphertext* → pengemasan data terenkripsi.

3.3.2. Implementasi Dekripsi Algoritma AES-128

Proses dekripsi dilakukan menggunakan fungsi *decrypt()*, yang bertujuan mengembalikan *ciphertext* menjadi *plaintext* asli. Tahapan pertama dalam proses ini adalah melakukan *decode* dari *Base64* sehingga sistem memperoleh kembali struktur asli *ciphertext* yang berisi *IV* di bagian awal dan data terenkripsi pada bagian berikutnya.

```
public function decrypt($encryptedData, $key): bool|string
{
    try {
        // Pastikan key 16 bytes untuk AES-128
        if (strlen($key) != 16) {
            throw new \Exception(message: 'Key harus 16 bytes untuk AES-128');
        }

        $data = base64_decode(string: $encryptedData);
        if ($data === false) {
            throw new \Exception(message: 'Data tidak valid');
        }

        $ivLength = openssl_cipher_iv_length($cipher_algo: $this->cipher);
        $iv = substr($data, offset: 0, length: $ivLength);
        $encrypted = substr($data, offset: $ivLength);

        // Dekripsi
        $decrypted = openssl_decrypt(
            data: $encrypted, cipher_algo: $this->cipher, passphrase: $key,
            options: OPENSSL_RAW_DATA, iv: $iv );
        if ($decrypted === false) {
            throw new \Exception(message: 'Dekripsi gagal');
        }

        return $decrypted;
    } catch (\Exception $e) {
        log_message(level: 'error', message: 'AES Decryption Error: ' . $e->getMessage());
        return false;
    }
}
```

Gambar 14 Potongan kode fungsi *decrypt()*

Pada potongan kode di Gambar 14 tersebut terlihat bahwa *ciphertext* yang telah di-*decode* kemudian dipisahkan berdasarkan ukuran *IV*. Sistem mengambil panjang *IV* menggunakan *openssl_cipher_iv_length()* agar pemisahan dilakukan secara akurat dan sesuai standar AES-128. Setelah itu, bagian awal data dijadikan *IV*, sementara sisa data dianggap sebagai *ciphertext* yang siap didekripsi. Teknik ini memastikan konsistensi dan integritas proses dekripsi.

Tahapan berikutnya adalah memanggil *openssl_decrypt()* dengan parameter *cipher*, *key*, *ciphertext*, dan *IV* yang telah dipisahkan. Jika seluruh parameter sesuai, fungsi ini menghasilkan plaintext asli tanpa perubahan sedikit pun. Hal ini menunjukkan bahwa enkripsi dan dekripsi berjalan simetris sesuai sifat dasar AES.

```
// Dekripsi
$decrypted = openssl_decrypt(
    data: $encrypted,
    cipher_algo: $this->cipher,
    passphrase: $key,
    options: OPENSSL_RAW_DATA,
    iv: $iv
);
```

Gambar 15. Potongan kode pemanggilan *openssl_decrypt()*

Potongan kode pada gambar 15. tersebut menjelaskan bahwa fungsi dekripsi hanya dapat berhasil jika *ciphertext*, *key*, dan *IV* cocok satu sama lain. Inilah yang menjamin keamanan sistem: tanpa *key* yang benar, data tidak dapat dikembalikan ke bentuk awal. Dengan demikian, implementasi dekripsi telah memenuhi prinsip kriptografi simetris, konsistensi blok AES-128, serta standar keamanan data yang diperlukan dalam penyimpanan soal ujian.

3.3.3. Implementasi Antar Muka Sistem

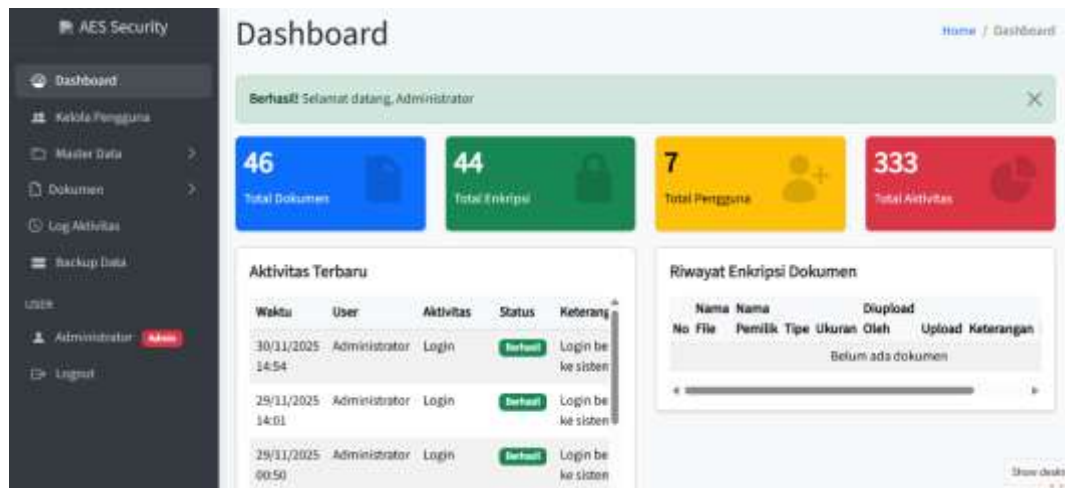
a) Halaman Login



Gambar 16. Tampilan Halaman Login

Halaman login merupakan pintu utama untuk mengakses sistem. Pengguna harus memasukkan username dan password yang valid untuk masuk. Sistem menyediakan validasi untuk memastikan hanya pengguna terdaftar yang dapat melakukan autentikasi. Jika data yang dimasukkan benar, pengguna diarahkan ke dashboard sesuai levelnya (Admin atau Guru). Jika salah, akan muncul pesan error. Halaman ini juga menjadi kontrol keamanan awal agar setiap aktivitas tercatat berdasarkan identitas pengguna masing-masing.

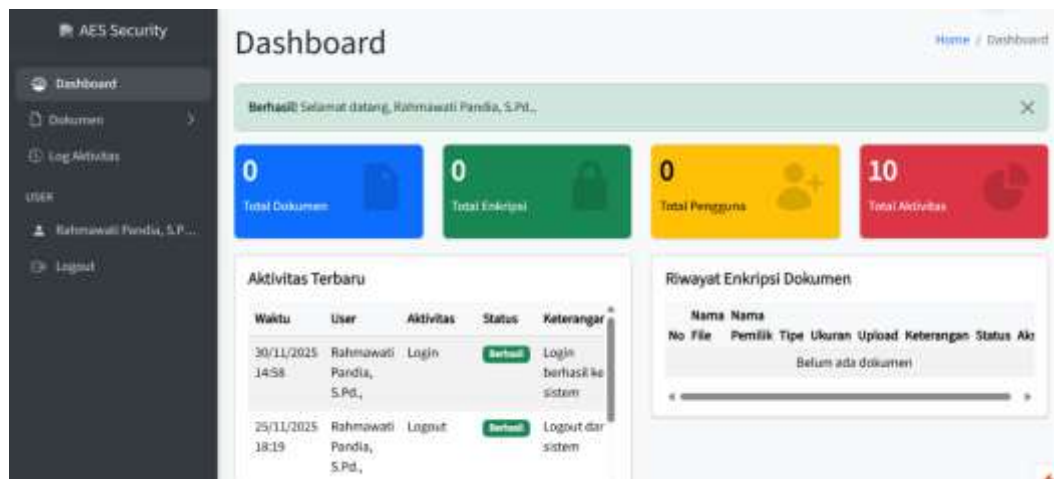
b) Halaman Dashboard Admin



Gambar 17. Tampilan Halaman Dashboard

Dashboard Admin menampilkan ringkasan informasi sistem seperti jumlah pengguna, jumlah guru, jumlah mapel, jumlah dokumen, dan log aktivitas terakhir. Halaman ini berfungsi sebagai pusat navigasi, sehingga Admin dapat mengakses fitur penting seperti Kelola Pengguna, Data Guru, Data Mapel, Data Kelas, Pengelolaan Dokumen, serta Log Aktivitas. Dashboard ini dirancang agar Admin memperoleh gambaran umum kondisi sistem secara cepat.

c) Halaman Dashboard Guru



Gambar 18. Tampilan Halaman Dashboard Guru

Dashboard Guru hanya menampilkan fitur yang sesuai dengan hak akses guru, seperti daftar dokumen, menu unggah dokumen, enkripsi, dan dekripsi. Guru dapat dengan mudah melihat dokumen yang telah diunggah, melakukan enkripsi soal, atau mengunduh dokumen soal yang sudah diproses. Tampilan dibuat sederhana agar mudah digunakan oleh pengguna non-teknis.

d) Halaman Kelola Pengguna Pada Admin



Gambar 19. Tampilan Halaman Kelola Pengguna

Halaman ini hanya dapat diakses oleh Admin. Fungsinya adalah untuk menambahkan pengguna baru, mengedit data pengguna, atau menghapus pengguna. Data yang dikelola meliputi username, nama lengkap, email, level akses, dan status akun. Setiap perubahan yang dilakukan Admin akan tercatat di log aktivitas sehingga dapat ditelusuri kembali jika diperlukan.

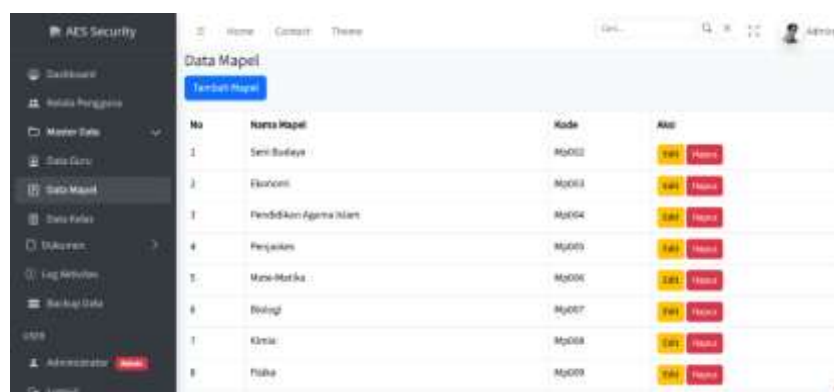
e) Halaman Data Guru Pada Admin



Gambar 20. Tampilan Halaman Kelola Data Guru

Halaman ini digunakan Admin untuk mengelola data para guru yang terdaftar dalam sistem. Admin dapat menambahkan guru baru, memperbarui data guru, atau menghapus guru. Data yang diatur meliputi nama lengkap, NIP, jabatan, dan jenis kelamin. Data guru terhubung dengan tabel pengguna sehingga integritas informasi tetap terjaga.

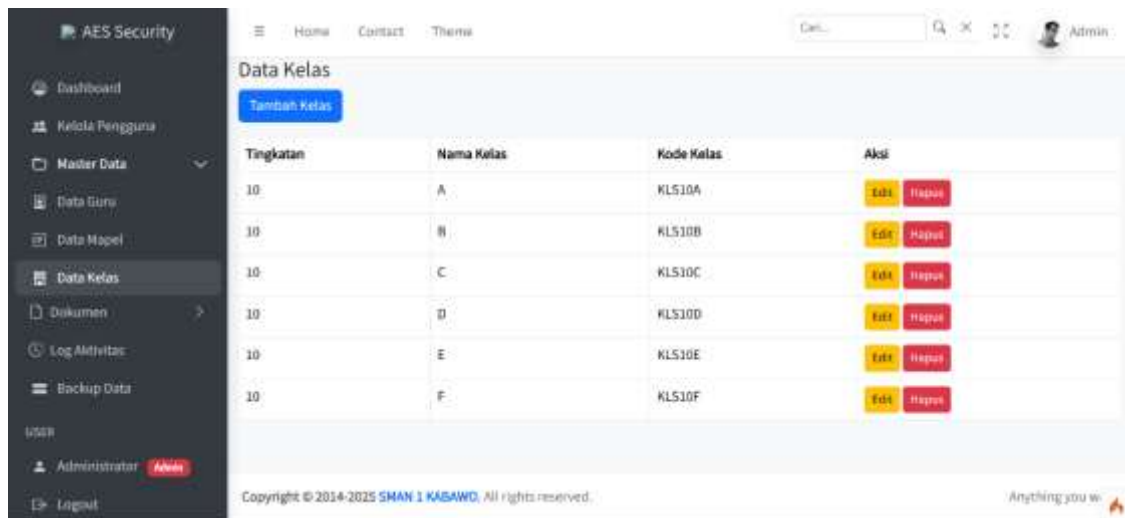
f) Halaman Data Mapel



Gambar 21. Tampilan Halaman kelola Data Mapel

Pada halaman ini Admin dapat mengelola data mata pelajaran yang ada di SMA Negeri 1 Kabawo. Admin dapat menambahkan mapel baru, mengedit mapel, serta menghapus mapel yang tidak digunakan lagi. Informasi mapel diperlukan untuk mengelompokkan dokumen soal berdasarkan pelajaran masing-masing.

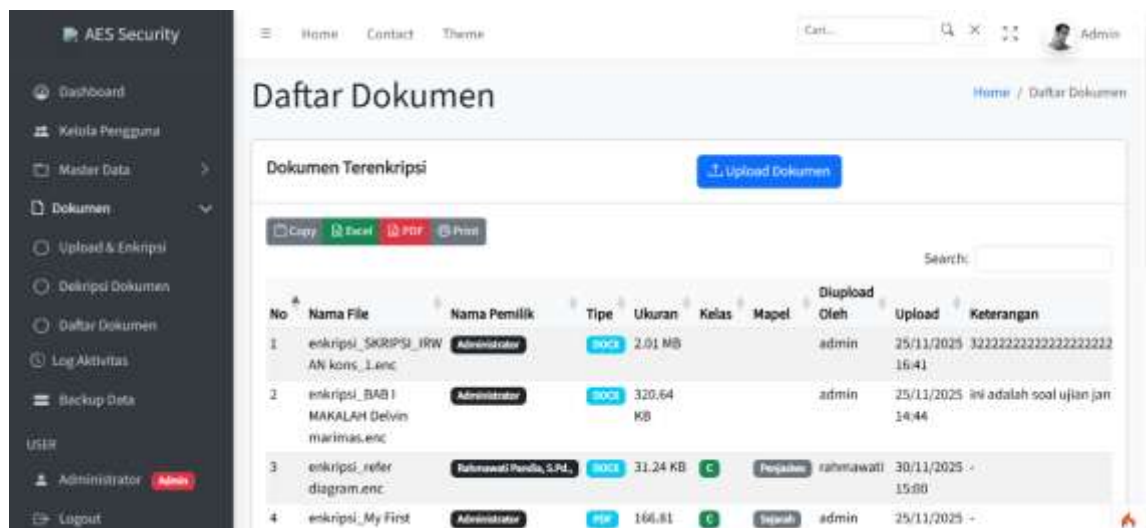
g) Halaman Data Kelas



Gambar 22. Tampilan Halaman kelola data Kelas

Halaman Data Kelas digunakan untuk mengelola informasi kelas seperti nama kelas dan tingkatannya. Data kelas akan digunakan pada proses penamaan dan pengelompokan dokumen soal sehingga memudahkan dalam manajemen dokumen berdasarkan kelas.

h) Halaman Daftar Dokumen



Gambar 23. Tampilan Halaman Daftar Dokumen

Halaman ini menampilkan daftar seluruh dokumen yang telah diunggah pengguna. Informasi yang ditampilkan mencakup nama dokumen, jenis file, pemilik, tanggal unggah, status enkripsi, serta opsi untuk melihat, mengunduh, mengenkripsi, atau mendekripsi dokumen. Guru hanya dapat melihat dokumen miliknya sendiri, sedangkan Admin dapat melihat semua dokumen.

i) Halaman Form Enkripsi



Gambar 24. Tampilan Halaman Form enkripsi

Halaman Form Enkripsi digunakan untuk melakukan proses enkripsi dokumen menggunakan algoritma AES-128. Pengguna memilih dokumen, sistem menghasilkan kunci enkripsi, dan file diproses menjadi ciphertext (file .enc). Setelah proses selesai, kunci harus disimpan oleh pengguna karena akan digunakan pada saat dekripsi. Halaman ini juga memberikan informasi bahwa file terenkripsi tidak dapat dibaca tanpa kunci.

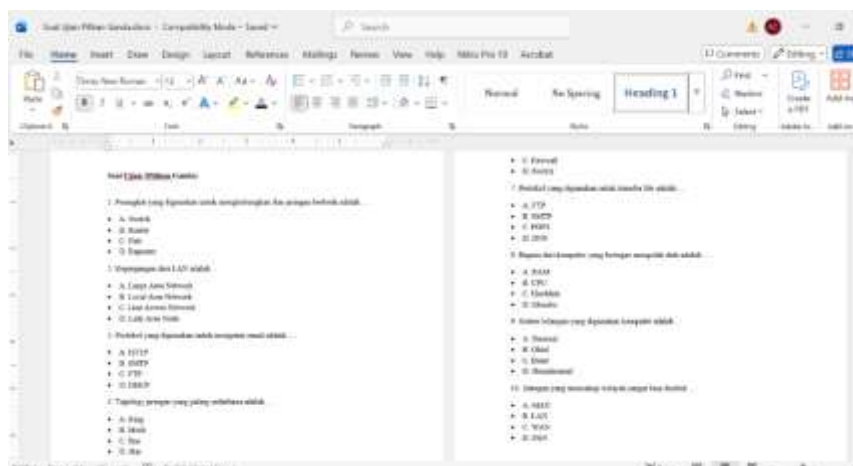
3.4. Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa seluruh fungsi yang dikembangkan telah berjalan sesuai dengan kebutuhan. Metode pengujian yang digunakan adalah *Blackbox Testing*, yaitu metode pengujian yang difokuskan pada kesesuaian *input* dan *output* tanpa melihat struktur internal kode.

Pengujian dilakukan pada seluruh fitur sistem, termasuk proses autentikasi, manajemen data, pengolahan dokumen, serta proses enkripsi dan dekripsi *AES-128*.

3.4.1. Pengujian Hasil Algoritma AES-128

Pada tahap pengujian algoritma *AES-128*, dilakukan proses enkripsi terhadap sebuah dokumen untuk melihat perubahan isi file sebelum dan sesudah diproses oleh sistem. Pengguna terlebih dahulu mengunggah file soal ujian, misalnya *Soal_Ujian.docx*. Pada tampilan awal, isi file masih berbentuk teks normal yang dapat dibaca dengan jelas. Contoh isi dokumen sebelum enkripsi adalah sebagai berikut:

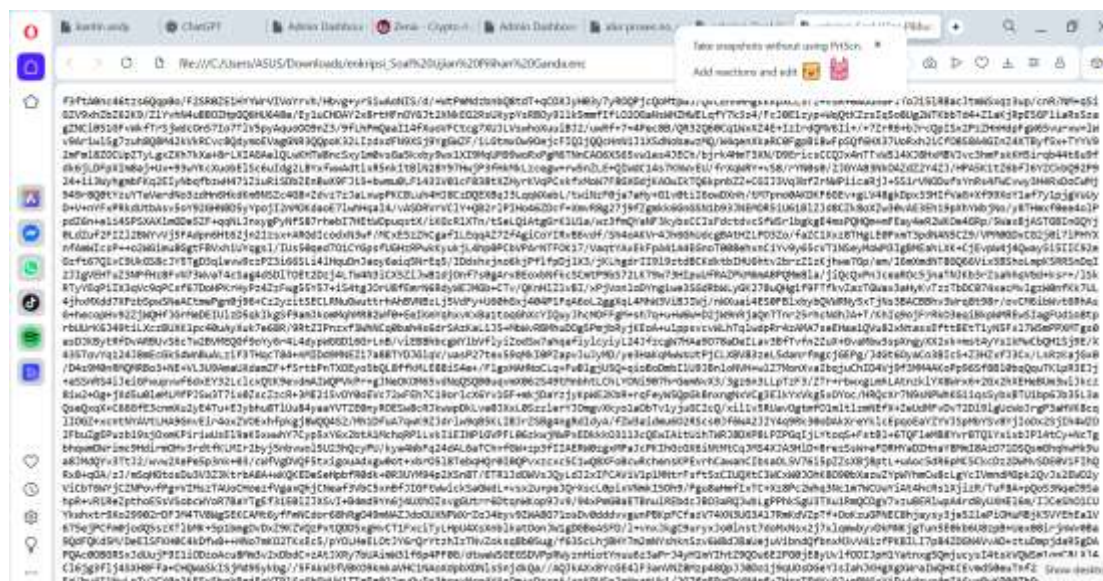


Gambar 25. Isi Dokumen asli

Isi dokumen terlihat pada gambar 25. dalam format normal, misalnya memuat daftar soal dan teks paragraf sesuai file aslinya. Dokumen pada tahap ini masih dapat dibuka menggunakan *Microsoft Word* karena belum melalui proses pengamanan. Kondisi ini menunjukkan bahwa file masih dalam bentuk plaintext dan dapat dibaca tanpa batasan.

Selanjutnya, pengguna memasukkan kunci enkripsi. Pada pengujian ini, kunci yang digunakan memiliki bentuk seperti “1234567890123456”. Kunci ini memiliki 16 karakter dan berfungsi sebagai parameter utama algoritma *AES-128* untuk mengubah data asli menjadi ciphertext. Setelah kunci dimasukkan, pengguna menekan tombol Enkripsi dan sistem memulai proses pemrosesan data secara blok menggunakan mode operasi *AES-128-CBC*.

Proses enkripsi melibatkan pembangkitan *Initialization Vector (IV)* secara acak, melakukan padding terhadap data file, kemudian mengolah setiap blok plaintext menjadi blok ciphertext yang tidak dapat dibaca. Hasilnya kemudian ditulis sebagai file baru dengan ekstensi *.enc*. Sebagai contoh, file asli *Soal_Ujian.docx* akan berubah menjadi:



Gambar 26. Isi Dokumen Setelah di Enkripsi

Sesudah dilakukan enkripsi, dokumen yang tadinya masih bisa di baca, kemudian menjadi tulisan yang sulit dibaca.

Berikut ini merupakan tabel hasil pengujian dengan jenis dokumen yang berbeda, perhatikan tabel 1. berikut.

Tabel 1. Hasil Pengujian Algoritma AES-128

No	Nama Dokumen	Proses Enkripsi	Proses Dekripsi	Kecocokan Isi Setelah Dekripsi	Status
1	Soal_Ujian.docx	Berhasil	Berhasil	100% Sama	Valid
2	Soal_Biologi.pdf	Berhasil	Berhasil	100% Sama	Valid
3	Kisi_Kisi_Ujian.xlsx	Berhasil	Berhasil	100% Sama	Valid
4	Soal_Sosiologi.docx	Berhasil	Berhasil	100% Sama	Valid

Pada tabel 1. dapat dilihat bahwa algoritma *AES-128* telah berfungsi dengan baik. Semua dokumen berhasil dienkripsi menjadi ciphertext dan dapat dikembalikan sepenuhnya ke bentuk asli tanpa kehilangan informasi.

Unit Testing dilakukan untuk menguji fungsi-fungsi utama sistem secara individual, khususnya fungsi-fungsi yang berhubungan dengan keamanan data.

Tabel 2. Hasil Unit Testing

No	Fungsi yang Diuji	Input	Output yang Diharapkan	Hasil Pengujian	Keterangan
1	<i>generateKey()</i>	–	Kunci 16 byte	Berhasil	Valid
2	<i>encryptFile()</i>	File .docx	File .enc	Berhasil	Valid
3	<i>decryptFile()</i>	File .enc	File asli	Berhasil	Valid
4	<i>uploadDocument()</i>	Dokumen	Tersimpan	Berhasil	Valid
5	<i>downloadDocument()</i>	ID file	File diunduh	Berhasil	Valid
6	<i>validateKey()</i>	Kunci 16 karakter	Kunci diterima	Berhasil	Valid
7	<i>login()</i>	Username + Password	Akses diberikan	Berhasil	Valid
8	<i>saveLog()</i>	Aktivitas	Tercatat	Berhasil	Valid

Integration Testing memastikan bahwa hubungan antar modul bekerja tanpa error, seperti modul login, enkripsi, database, dan log aktivitas. Perhatikan tabel Integration Testing berikut

Tabel 3. Hasil Integration Testing

No	Skenario Pengujian	Modul Terlibat	Hasil Diharapkan	Hasil Pengujian	Status
1	Upload → Enkripsi → Simpan DB	Upload, AES, DB	File terenkripsi tersimpan	Berhasil	Valid
2	Login → Dashboard	Auth, Menu	Menu tampil sesuai role	Berhasil	Valid
3	Enkripsi → Simpan Log	AES, Log	Aktivitas tersimpan	Berhasil	Valid
4	Dekripsi → Download	AES, Download	File asli diunduh	Berhasil	Valid
5	Admin Kelola User	User, DB	Data tersimpan	Berhasil	Valid
6	Guru Upload Dokumen	Guru, Dokumen	Dokumen tampil sesuai user	Berhasil	Valid
7	Logout → Login Page	Session	Session terhapus	Berhasil	Valid

UAT dilakukan dengan pengguna sebenarnya (guru dan admin SMA Negeri 1 Kabawo) untuk menilai kenyamanan dan kesesuaian sistem.

Tabel 4. Hasil User Acceptance Testing (UAT)

No	Fitur yang Diuji	Pengguna	Hasil Diharapkan	Hasil Pengujian	Kepuasan
1	Login	Admin & Guru	Akses sesuai role	Berhasil	100%
2	Upload Dokumen	Guru	File tersimpan	Berhasil	95%
3	Enkripsi	Admin & Guru	File menjadi ciphertext	Berhasil	98%
4	Dekripsi	Admin & Guru	File kembali ke asli	Berhasil	97%
5	Download	Semua	File dapat diunduh	Berhasil	100%
6	Log Aktivitas	Admin	Aktivitas tampil	Berhasil	96%
7	Navigasi Sistem	Semua	Mudah digunakan	Berhasil	94%

3.4.2. Perbandingan File Asli, Terenkripsi, dan Terdekripsi

Bagian ini menunjukkan bagaimana file berubah setelah proses enkripsi, serta bagaimana file kembali sempurna ke bentuk awal setelah dekripsi.

Tabel 5. Perbandingan File Sebelum & Sesudah Enkripsi

No	Nama Dokumen	Ukuran Asli	Ukuran Enkripsi	Ekstensi Enkripsi	Dapat Dibaca?	Keterangan
1	Soal_Ujian.docx	84 KB	88 KB	.enc	Tidak	Berhasil dienkripsi
2	Soal_Biologi.pdf	120 KB	123 KB	.enc	Tidak	Berhasil dienkripsi
3	Kisi_Kisi.xlsx	45 KB	47 KB	.enc	Tidak	Berhasil dienkripsi

Tabel 6. Perbandingan File Terenkripsi & Setelah Dekripsi

No	File Terenkripsi	File Hasil Dekripsi	Ekstensi Asli	Kecocokan Isi	Status
1	Soal_Ujian.enc	Soal_Ujian.docx	.docx	100% Sama	Valid
2	Soal_Biologi.enc	Soal_Biologi.pdf	.pdf	100% Sama	Valid
3	Kisi_Kisi.enc	Kisi_Kisi.xlsx	.xlsx	100% Sama	Valid

AES-128 berhasil bekerja dengan sangat baik dalam menjaga kerahasiaan dokumen. File yang terenkripsi berubah sepenuhnya menjadi ciphertext acak dan tidak dapat dibuka. Setelah didekripsi, file kembali ke bentuk asli tanpa kehilangan isi, sehingga menunjukkan integritas data terjaga 100%.

3.5. Pemeliharaan Sistem

Pemeliharaan sistem dilakukan untuk memastikan aplikasi tetap berjalan dengan baik, aman, dan sesuai kebutuhan pengguna dari waktu ke waktu. Pemeliharaan yang dilakukan meliputi pemeriksaan rutin, perbaikan bug, dan pembaruan fitur ketika diperlukan. Kegiatan pemeliharaan dilakukan melalui beberapa tahap, yaitu:

1. Pemantauan Kinerja Sistem
Sistem diperiksa secara berkala untuk memastikan proses enkripsi, dekripsi, dan pengolahan data berjalan tanpa kendala. Log aktivitas digunakan sebagai acuan untuk memantau penggunaan dan mendeteksi gangguan.
2. Perbaikan dan Penanganan Bug
Jika ditemukan kesalahan pada fitur seperti enkripsi, dekripsi, atau pengelolaan data, developer melakukan perbaikan segera agar tidak mengganggu pengguna. Perbaikan dilakukan tanpa mengubah data yang sudah ada.
3. Pembaruan Sistem (Update)
Pembaruan dilakukan apabila terdapat kebutuhan untuk meningkatkan keamanan, menambah fitur baru, atau menyesuaikan sistem dengan kebijakan sekolah. Pembaruan dilakukan secara terencana agar tidak mengganggu aktivitas pengguna.
4. Keamanan Data dan Backup
Backup data dilakukan secara periodic untuk menghindari kehilangan data penting seperti dokumen terenkripsi, akun pengguna, dan log aktivitas. Pengamanan juga terus ditingkatkan dengan memastikan algoritma AES-128 diterapkan secara benar tanpa celah keamanan.
5. Dukungan Pengguna (User Support)
Pemeliharaan juga mencakup penanganan keluhan atau kendala pengguna, seperti kesalahan kunci enkripsi, kesulitan mengunggah dokumen, atau masalah akses akun. Setiap laporan pengguna digunakan sebagai bahan evaluasi untuk meningkatkan sistem.

Melalui kegiatan pemeliharaan tersebut, sistem diharapkan tetap stabil, aman, dan dapat digunakan secara jangka panjang untuk mengamankan dokumen soal ujian di SMA Negeri 1 Kabawo

4. Kesimpulan

Berdasarkan hasil penelitian dan implementasi, sistem pengamanan dokumen soal ujian berhasil dibangun menggunakan algoritma AES-128 dengan mekanisme enkripsi dan dekripsi yang berjalan baik. Implementasi AES-128 mampu mengubah file asli menjadi *ciphertext* sepenuhnya sehingga tidak dapat dibaca tanpa kunci dekripsi, dan proses dekripsi mampu mengembalikan dokumen ke bentuk semula secara utuh. Hasil pengujian menunjukkan bahwa metode ini efektif dalam menjaga kerahasiaan dokumen, sehingga sistem yang dikembangkan dapat digunakan sebagai solusi pengamanan dokumen soal ujian di SMA Negeri 1 Kabawo. Hasil implementasi sistem menunjukkan bahwa fitur-fitur utama seperti autentikasi pengguna, manajemen hak akses (admin dan guru), pengelolaan data pengguna, pengelolaan dokumen soal, proses *enkripsi-dekripsi* dokumen, serta pencatatan log aktivitas mampu membatasi akses hanya kepada pihak berwenang. Hasil pengujian *Unit Testing*, *Integration Testing*, dan *User Acceptance Testing* membuktikan bahwa fitur-fitur tersebut berfungsi dengan baik dan mendukung keamanan sistem secara keseluruhan. Dengan demikian, fitur yang dirancang telah memenuhi kebutuhan pengamanan dokumen sebagaimana ditetapkan dalam penelitian ini.

Referensi

- [1] B. Cerlia Ashari and S. Waluyo, "Pengamanan File Ujian Menggunakan Algoritma Advanced Encryption Standard 128 Di Smp Negeri 22," *Semin. Nas. Mhs. Fak. Teknol. Inf. Jakarta-Indonesia*, no. September, pp. 240–247, 2022.
- [2] B. Wicaksana and M. Setiawan, "Penerapan Algoritma Advanced Encryption Standard (AES) Untuk Pengamanan Berkas Soal Ujian," *Teknois J. Ilm. Teknol. Inf. dan Sains*, vol. 10, no. 1, pp. 25–34, 2020, doi: 10.36350/jbs.v10i1.74.

- [3] Dola Ramalinda, Jayadi, and Agung Rachmat Raharja, "Strategi Perlindungan Data Menggunakan Sistem Kriptografi Dalam Keamanan Informasi," *J. Int. Multidiscip. Res.*, vol. 2, no. 6, pp. 665–671, 2024, doi: 10.62504/jimr679.
- [4] N. Cristy and F. Riandari, "Implementasi Metode Advanced Encryption Standard (AES 128 Bit) Untuk Mengamankan Data Keuangan," *J. Ilmu Komput. dan Sist. Inf.*, vol. 4, no. 2, pp. 75–85, 2021.
- [5] J. Handoyo and Y. M. Subakti, "Keamanan Dokumen Menggunakan Algoritma Advanced Encryption Standard (Aes)," *J. SITECH Sist. Inf. dan Teknol.*, vol. 3, no. 2, pp. 143–152, 2020, doi: 10.24176/sitech.v3i2.5865.
- [6] L. Hakim, A. T. Martadinata, A. Ujian, O. Dan, P. Algoritma, and L. C. G. Untuk, "Aplikasi ujian online dan penerapan algoritma lcg untuk proses pengacakan soal ujian di smk negeri tugumulyo," vol. 7, no. 2, pp. 99–108, 2022.
- [7] R. Ramadhan and H. Soetanto, "Penerapan Kriptografi Menggunakan Advanced Encryption Standard 128 Untuk Pengamanan File Pada SMK Muhammadiyah 4," *Semin. Nas. Mhs. Fak. Teknol. Inf.*, no. September, pp. 29–38, 2022.
- [8] R. D. Putranto, "Analisa dan Perancangan Sistem Keamanan File Dengan Advanced Encryption Standard (AES) Berbasis Website," *J. Inform. MULTI*, vol. 1, no. 6, 2023.
- [9] R. R. D. Andika and S. Mulyati, "Penerapan Algoritma Aes-128 Untuk Aplikasi Pengarsipan Dokumen Berbasis Web Pada Pt Studio Inovasi Teknologi Application of Aes-128 Algorithm for Web-Based Document Archiving Application At Pt Studio Inovasi Teknologi," *Semin. Nas. Mhs. Fak. Teknol. Inf. Jakarta-Indonesia*, no. September, pp. 411–420, 2022.
- [10] A. Ramadan and Painem, "Pengamanan Data Keuangan Menggunakan Algoritma Advanced Encryption Standard 128 Pada Pt. Charise Deo Indonesia," *Semin. Nas. Mhs. Fak. Teknol. Inf. Jakarta-Indonesia*, no. September, pp. 49–57, 2022.
- [11] L. Sidabutar, M. Ramadhan, and Z. Panjaitan, "Implementasi Kriptografi Pengamanan Data Pemesanan Produk Menggunakan Metode AES," vol. 3, pp. 440–449, 2024.
- [12] A. S. Aswandi, A. Pradipta, S. Informasi, U. Sembilanbelas, and N. Kolaka, "Analisis performa dan keamanan implementasi kriptografi aes untuk penyandian dokumen berbasis web," vol. 8, no. 1, pp. 24–32, 2025.
- [13] M. Stamp, *Information security: principles and practice*. John Wiley & Sons, 2011.
- [14] E. Setyawati, C. E. Widjayanti, R. R. Siraiz, and H. Wijoyo, "Pengujian keamanan komputer kriptografi pada surat elektronik berbasis website dengan enkripsi metode MD5," *J. Manajemen Inform. Jayakarta*, vol. 1, no. 1, p. 56, 2021, doi: 10.52362/jmijayakarta.v1i1.367.
- [15] R. Munir, "Kriptografi Edisi Kedua," *Bandung. Penerbit Inform.*, 2019.
- [16] H. D. Novianti and A. T. Hidayat, "IMPLEMENTASI KRIPTOGRAFI ADVANCED ENCRYPTION STANDARD 128 BIT DALAM PENGAMANAN DATA KEUANGAN KAS:(Studi Kasus: Masjid Al-Ikhlas Trini ...," *J. Komput. dan Teknol.*, 2023.
- [17] N. A. Nanda, S. M. S. Silalahi, D. Patricia Nasution, M. Sari, and I. Gunawan, "Kriptografi dan Penerapannya Dalam Sistem Keamanan Data," *J. Media Inform.*, vol. 4, no. 2, pp. 90–93, 2023, doi: 10.55338/jumin.v4i2.428.
- [18] N. Fithria, "Jenis-Jenis Serangan terhadap Kriptografi," *ITB J.*, no. 13506036, p. 1, 2016.
- [19] I. Gunawan, S. Tinggi, and T. Ronggolawe, "KEAMANAN DATA :," no. January, 2021.
- [20] I. Febriana and G. Aji, "Penerapan Teknik Kriptografi Pada Keamanan SMS Android," *JOEICT (Jurnal Educ. Inf. Commun. Technol.*, vol. 1, no. 1, pp. 29–36, 2017.
- [21] Yusfrizal, "Rancang Bangun Aplikasi Kriptografi Pada Teks Menggunakan Metode Reverse Chiper Dan Rsa Berbasis Android," *JTIK (Jurnal Tek. Inform. Kaputama)*, vol. 3, no. 2, pp. 29–37, 2019.