e-ISSN: 3110-0864 | p-ISSN: xxxx-xxxx
JURNAL ILMU KOMPUTER DAN TEKNIK INFROMATIKA
Volume 2 No 1 Januari 2026 | Page: 55-62
https://journals.raskhamedia.or.id/index.php/juikti
DOI: https://doi.org/10.64803/juikti.v2i1.100

Perancangan dan Implementasi Sistem Enkripsi Data Sensitif Menggunakan AES-256-CBC pada Aplikasi Berbasis Web Sederhana

Muhammad Randy Fachrezi[1,*], Dwiky Oldi Amsyah[2], Alwi Syahputra[3], Ibnu Rusydi[4]

[1,2,3,4]Fakultas Sains Dan Teknologi, Ilmu Komputer, Universitas Islam Negeri Sumatera Utara, Medan, Indonesia
Email: [1*]randyfachrezy6@gmail.com, [2]Dwikygg9624@gmail.com, [3]alwiisyahputra0@gmail.com, [4]ibnurusydi@dharmawangsa.ac.id
(* Email Corresponding Author: randyfachrezy6@gmail.com)
Received: January 15, 2026 | Revision: January 18, 2026 | Accepted: January 18, 2026

Abstrak

Data formulir pada aplikasi web umumnya tersimpan dalam bentuk teks biasa yang rentan dibaca ketika terjadi akses tidak sah ke sistem penyimpanan. Penelitian ini bertujuan mengimplementasikan algoritma Advanced Encryption Standard (AES-256-CBC) untuk mengenkripsi data input formulir dan membuktikan proses transformasi data dari plaintext menjadi ciphertext serta kemampuan mengembalikannya melalui dekripsi menggunakan passphrase. Metode penelitian meliputi studi literatur, identifikasi masalah, perancangan sistem, implementasi, dan pengujian pada tiga skenario data yang mencakup nama lengkap, email, nomor telepon, dan pesan. Hasil pengujian menunjukkan sistem berhasil mengenkripsi seluruh data input menjadi ciphertext yang tidak dapat dipahami tanpa kunci dekripsi. Proses dekripsi dengan passphrase yang benar menghasilkan data plaintext yang identik dengan input awal dan menampilkan status verifikasi berhasil, sedangkan passphrase yang salah menghasilkan pesan kesalahan dekripsi gagal. Penelitian ini membuktikan bahwa AES-256-CBC efektif dalam mengamankan data formulir web melalui mekanisme enkripsi-dekripsi berbasis passphrase, sehingga data sensitif tidak lagi tersimpan dalam bentuk yang mudah dibaca dan hanya dapat diakses oleh pihak yang memiliki passphrase yang valid.

Kata Kunci: AES-256-CBC, enkripsi data, dekripsi, passphrase, keamanan formulir web

Abstract

Form data in web applications are typically stored in plain text format, which is vulnerable to unauthorized reading when storage access is compromised. This research aims to implement the Advanced Encryption Standard (AES-256-CBC) algorithm to encrypt form input data and demonstrate the data transformation process from plaintext to ciphertext, as well as the ability to restore it through decryption using a passphrase. The research methodology includes literature study, problem identification, system design, implementation, and testing on three data scenarios covering full name, email, phone number, and message. Test results show that the system successfully encrypts all input data into Base64 format ciphertext that cannot be understood without a decryption key. The decryption process with the correct passphrase produces plaintext data identical to the original input and displays a successful verification status, while an incorrect passphrase generates a decryption failure error message. This research proves that AES-256-CBC is effective in securing web form data through passphrase-based encryption-decryption mechanisms, ensuring that sensitive data is no longer stored in easily readable form and can only be accessed by parties possessing the valid passphrase.

Keywords*: AES-256-CBC, data encryption, decryption, passphrase, web form security*

1. PENDAHULUAN

Masalah pada penelitian ini adalah data dari form website umumnya masih diproses sebagai teks biasa (plaintext), sehingga saat data tersebut disimpan atau ditampilkan kembali, nilainya dapat langsung dibaca. Kondisi ini membuat pengujian perlindungan data berbasis algoritma menjadi relevan, khususnya untuk membuktikan perubahan bentuk data dari plaintext menjadi bentuk tersandi. Solusi yang ditawarkan dalam penelitian ini adalah menerapkan algoritma Advanced Encryption Standard (AES) untuk mengenkripsi data form sebelum disimpan, sehingga data yang tersimpan menjadi ciphertext dan tidak dapat dipahami tanpa proses dekripsi.

Perkembangan teknologi informasi mendorong organisasi memanfaatkan aplikasi berbasis web untuk mempercepat layanan dan pertukaran informasi, tetapi hal ini juga memperbesar paparan risiko karena aplikasi web sering menangani data sensitif. OWASP Top 10:2021 memetakan risiko yang sering muncul pada aplikasi web, termasuk Broken Access Control dan Cryptographic Failures yang berkaitan dengan kegagalan pembatasan akses dan perlindungan kriptografi pada data. Literatur jurnal juga menegaskan bahwa kriptografi berperan fundamental untuk melindungi data digital dari ancaman siber dan menjaga aspek keamanan seperti kerahasiaan/integritas melalui teknik enkripsi, enkripsi kunci-publik, dan hashing [1][2].

Pada konteks aplikasi web, penerapan enkripsi seperti AES penting karena data sensitif tidak hanya dikirim lewat jaringan, tetapi juga disimpan di sisi server (database/penyimpanan data). Dengan enkripsi, data yang disimpan tidak lagi dalam bentuk yang mudah dibaca, sehingga ketika terjadi akses tidak sah, informasi

e-ISSN: 3110-0864 | p-ISSN: xxxx-xxxx
JURNAL ILMU KOMPUTER DAN TEKNIK INFROMATIKA
Volume 2 No 1 Januari 2026 | Page: 55-62
https://journals.raskhamedia.or.id/index.php/juikti
DOI: https://doi.org/10.64803/juikti.v2i1.100

tetap lebih terlindungi. Penelitian pengembangan keamanan data berbasis web juga menunjukkan bahwa penggunaan algoritma enkripsi modern dapat membantu menjaga kerahasiaan data, baik dilakukan di sisi klien maupun sisi server [3][4].

Penelitian pertama dilakukan oleh Tania (2024) yang berjudul "Algoritma AES untuk Keamanan Data Digital". Penelitian ini mengimplementasikan algoritma AES 128 bit untuk mengamankan dokumen surat penting di Kantor Desa Aman Damai melalui aplikasi berbasis web. Hasilnya menunjukkan bahwa AES mampu mengenkripsi dokumen digital dengan waktu proses yang cepat dan tingkat keamanan yang tinggi, sehingga dokumen penting tetap terjaga kerahasiaannya meskipun tersimpan dalam sistem digital yang terhubung dengan jaringan [5].

Penelitian kedua oleh Nasrullah (2025) berjudul "Secure Web-Based File Encryption Using AES-128". Studi ini mengembangkan sistem enkripsi dan dekripsi file berbasis web dengan menggunakan AES-128 bit mode CBC (Cipher Block Chaining). Penelitian ini menunjukkan bahwa file dengan berbagai format seperti dokumen teks, gambar, dan PDF dapat dienkripsi dengan aman menggunakan AES, dan hanya pengguna yang memiliki kunci yang tepat yang dapat mendekripsi kembali file tersebut. Sistem yang dikembangkan juga dilengkapi dengan antarmuka web yang user-friendly sehingga memudahkan pengguna awam dalam melakukan proses enkripsi [6].

Penelitian ketiga dilakukan oleh Hermawan et al. (2021) dengan judul "Implementasi Algoritma Kriptografi AES untuk Peningkatan Keamanan Proses Login pada Website". Penelitian ini fokus pada pengamanan password pengguna saat proses login dengan menerapkan AES untuk mengenkripsi password sebelum disimpan ke database. Metode yang digunakan adalah AES 256 bit dengan mode operasi ECB (Electronic Code Book). Hasil penelitian menunjukkan bahwa password yang tersimpan dalam database tidak lagi dapat dibaca secara langsung oleh penyerang meskipun berhasil mengakses tabel database, sehingga risiko pencurian akun pengguna dapat diminimalisir secara signifikan [7].

Penelitian keempat yang dilakukan oleh Liauren et al. (2025) dengan judul "Implementasi Algoritma AES dan Bcrypt untuk Pengamanan Data Pengguna pada Website Jahitku". Penelitian ini menerapkan kombinasi dua algoritma yaitu AES 256 CBC untuk mengenkripsi data pribadi pengguna seperti nama, nomor telepon, dan jenis kelamin, serta algoritma Bcrypt untuk hashing password. Pengujian dilakukan dengan mengukur waktu enkripsi dan dekripsi data, hasilnya menunjukkan bahwa AES 256 CBC memiliki waktu enkripsi rata-rata 0,567 ms dan dekripsi 0,419 ms, sehingga tidak menimbulkan keterlambatan yang signifikan pada sistem meskipun diakses oleh banyak pengguna secara bersamaan. Penelitian ini juga menegaskan bahwa data yang tersimpan di database MySQL dalam bentuk terenkripsi dapat melindungi privasi pengguna dengan baik [8].

Penelitian kelima yang dilakukan oleh Joshi et al. (2026) dengan judul "Enhancing Cloud Data Security Through AES Encryption and SHA-256". Penelitian ini mengusulkan kerangka kerja kriptografi hibrida yang menggabungkan AES-256 untuk enkripsi data dengan SHA-256 untuk menjaga integritas data pada penyimpanan cloud seperti Amazon S3 dan Google Cloud Storage. Metode yang digunakan melibatkan enkripsi data sebelum diunggah ke cloud dan validasi integritas data menggunakan hash SHA-256. Hasil penelitian menunjukkan bahwa kerangka kerja ini mampu melindungi data dalam jumlah besar dengan efisien, serta mencegah serangan seperti brute-force, man-in-the-middle, dan manipulasi data. Penelitian ini juga menekankan pentingnya manajemen kunci yang aman dalam implementasi AES untuk mencegah kebocoran kunci enkripsi [9].

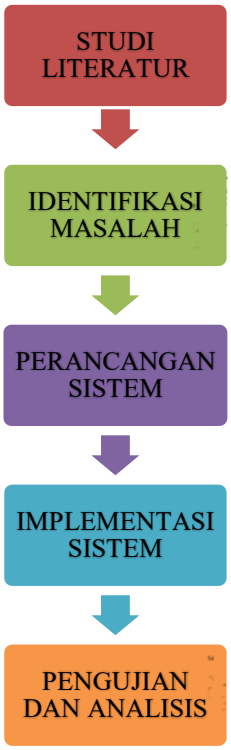
Tujuan penelitian ini adalah mengimplementasikan algoritma Advanced Encryption Standard (AES) pada data yang berasal dari formulir untuk membuktikan alur kerja enkripsi–dekripsi secara nyata, yaitu mengubah data formulir dari plaintext menjadi ciphertext dan mengembalikannya kembali menjadi plaintext melalui proses dekripsi. Penelitian ini juga bertujuan menguji penggunaan passphrase sebagai dasar pembentukan kunci (key) yang dipakai pada proses enkripsi dan dekripsi, sehingga dapat diamati bahwa ciphertext yang tersimpan tidak dapat dipahami tanpa passphrase yang benar, dan hasil dekripsi akan kembali sesuai dengan data input awal ketika passphrase yang digunakan valid. Harapan yang ingin dicapai adalah menghasilkan pemahaman implementatif yang jelas mengenai bagaimana AES bekerja pada skenario data formulir—termasuk peran passphrase dalam proses dekripsi—serta menyediakan rancangan pengujian sederhana yang dapat dijadikan acuan untuk penerapan enkripsi data input pada sistem sejenis.

2. METODOLOGI PENELITIAN

Dalam pelaksanaan penelitian ini, digunakan pendekatan yang sistematis agar setiap tahapan penelitian dapat berjalan dengan rapi, mudah dipahami, dan dapat dipertanggungjawabkan. Pendekatan ini dipilih karena penelitian tidak hanya berfokus pada pembuatan sistem, tetapi juga pada pembuktian bahwa algoritma AES mampu melakukan proses enkripsi dan dekripsi data formulir secara konsisten dengan menggunakan passphrase yang sama. Melalui pendekatan ini, hasil penelitian tidak hanya menunjukkan bahwa sistem dapat berjalan dengan baik, tetapi juga mampu menggambarkan secara jelas proses perubahan data dari

e-ISSN: 3110-0864 | p-ISSN: xxxx-xxxx
JURNAL ILMU KOMPUTER DAN TEKNIK INFROMATIKA
Volume 2 No 1 Januari 2026 | Page: 55-62
https://journals.raskhamedia.or.id/index.php/juikti
DOI: https://doi.org/10.64803/juikti.v2i1.100

plaintext menjadi ciphertext, serta proses pengembalian ciphertext ke bentuk plaintext. Metode penelitian yang diterapkan meliputi studi literatur, identifikasi masalah, perancangan sistem, implementasi sistem, dan pengujian sistem. Setiap tahapan dirancang untuk saling berkaitan sehingga menghasilkan sistem enkripsi data yang efektif dan dapat diandalkan. Berikut flowchart penelitiannya:



Gambar 1. Struktur Penelitian

2.1 Algoritma AES (Advanced Encryption Standard)

AES merupakan algoritma enkripsi kunci simetris yang menggunakan kunci yang sama untuk mengenkripsi dan mendekripsi data, sehingga kunci tersebut harus dijaga kerahasiaannya dengan baik. Algoritma ini telah terbukti efektif dalam melindungi data dengan tingkat keamanan yang tinggi dan efisiensi dalam penggunaan daya komputasi, serta digunakan secara luas di berbagai bidang seperti keamanan jaringan, perlindungan file, komunikasi digital, dan sistem perbankan. AES menggunakan ukuran blok 128 bit dengan tiga pilihan panjang kunci yaitu 128, 192, atau 256 bit, dimana semakin panjang kunci yang digunakan maka semakin tinggi tingkat keamanannya karena jumlah kombinasi kunci yang harus dicoba penyerang akan semakin besar [10].

Cara kerja AES dimulai dengan mengubah data asli (plaintext) menjadi matriks byte berukuran 4x4 yang disebut state, kemudian kunci enkripsi diperluas untuk menghasilkan kunci-kunci ronde yang akan digunakan pada setiap putaran proses enkripsi. Selanjutnya dilakukan transformasi berulang sebanyak 10 ronde untuk AES-128, 12 ronde untuk AES-192, atau 14 ronde untuk AES-256, dimana setiap ronde melakukan empat operasi utama yaitu SubBytes untuk mengganti byte menggunakan tabel substitusi, ShiftRows untuk menggeser baris pada matriks, MixColumns untuk mencampur kolom, dan AddRoundKey untuk menambahkan kunci ronde. Pada ronde terakhir proses yang sama dilakukan namun tanpa operasi MixColumns, dan state akhir kemudian diubah kembali menjadi ciphertext yang merupakan data terenkripsi yang tidak dapat dibaca tanpa kunci dekripsi yang tepat.

2.2 Studi Literatur

Tahapan pertama adalah studi literatur, di mana peneliti melakukan kajian terhadap berbagai sumber referensi seperti jurnal ilmiah, artikel, dan dokumentasi teknis terkait algoritma AES dan keamanan aplikasi berbasis web. Studi literatur dilakukan untuk memahami konsep dasar enkripsi AES, serta implementasi AES pada aplikasi web yang telah dilakukan oleh peneliti terdahulu. Selain itu, peneliti juga mempelajari bahasa

e-ISSN: 3110-0864 | p-ISSN: xxxx-xxxx
JURNAL ILMU KOMPUTER DAN TEKNIK INFROMATIKA
Volume 2 No 1 Januari 2026 | Page: 55-62
https://journals.raskhamedia.or.id/index.php/juikti
DOI: https://doi.org/10.64803/juikti.v2i1.100

pemrograman dan framework yang akan digunakan dalam pengembangan sistem untuk memastikan implementasi dapat berjalan dengan baik.

2.3 Identifikasi Masalah

Tahapan kedua adalah identifikasi masalah, yaitu mengidentifikasi dan merumuskan permasalahan keamanan data yang terjadi pada aplikasi berbasis web, khususnya terkait penyimpanan data input formulir seperti nama, email, nomor telepon, dan pesan pengguna yang biasanya disimpan dalam bentuk teks biasa (plaintext) di database. Plaintext adalah data atau informasi yang dapat dibaca secara langsung tanpa memerlukan proses dekripsi. Penyimpanan data dalam bentuk plaintext sangat berisiko karena jika terjadi kebocoran database, penyerang dapat dengan mudah membaca seluruh informasi sensitif pengguna [11]. Untuk mengatasi permasalahan tersebut, peneliti merumuskan solusi berupa implementasi enkripsi AES yang akan mengubah seluruh data input formulir menjadi ciphertext sebelum disimpan ke database. Ciphertext adalah hasil dari proses enkripsi yang mengubah data asli menjadi bentuk terenkripsi yang tidak dapat dipahami tanpa kunci dekripsi. Dengan diubah menjadi ciphertext, data yang tersimpan tidak dapat dibaca meskipun database berhasil diakses oleh pihak yang tidak berwenang, karena diperlukan kunci dekripsi yang tepat untuk mengembalikan data ke bentuk aslinya [12].

2.4 Perancangan Sistem

Dalam membangun sistem aplikasi berbasis web ini, terdapat tiga komponen teknologi utama yang digunakan, yaitu:

a. python

Python adalah bahasa pemrograman tingkat tinggi yang bersifat open-source, mudah dipelajari, dan memiliki sintaks yang sederhana sehingga sangat cocok untuk pengembangan aplikasi web. Python mendukung berbagai paradigma pemrograman seperti pemrograman berorientasi objek, fungsional, dan prosedural, serta memiliki ekosistem library yang sangat luas [13]. Pada kasus ini digunakan sebagai bahasa pemrograman utama untuk menulis skrip algoritma enkripsi AES dan mengembangkan backend sistem. Python dipilih karena memiliki library kriptografi yang lengkap seperti cryptography atau pycryptodome yang mendukung implementasi AES dengan berbagai mode operasi, serta mampu menangani logika bisnis, proses enkripsi dan dekripsi data, serta komunikasi dengan database.

b. react

Digunakan untuk mengembangkan frontend aplikasi sebagai library JavaScript yang memungkinkan pembuatan antarmuka pengguna yang interaktif dan responsif. React memudahkan pengembangan komponen form input data pengguna dan menampilkan data yang telah didekripsi dengan cara yang dinamis dan efisien [14].

c. html/css

Digunakan untuk membuat struktur halaman web dan styling tampilan antarmuka pengguna. HTML berfungsi untuk mendefinisikan elemen-elemen halaman seperti form input, tabel, dan layout, sedangkan CSS digunakan untuk mengatur tampilan visual seperti warna, ukuran font, spacing, dan responsivitas halaman [15].

Desain database mencakup pembuatan struktur data yang akan menyimpan ciphertext hasil enkripsi, dengan field yang sesuai untuk menyimpan nama, email, nomor telepon, dan pesan pengguna dalam bentuk terenkripsi. Dalam hal ini tampilan data dari hasil pengisian formulir akan muncul setelah *user* menginputkan suatu data.

2.5 Implementasi Sistem

Tahapan keempat adalah implementasi sistem, yaitu proses pengembangan aplikasi web berbasis kode program sesuai dengan rancangan yang telah dibuat. Implementasi dimulai dengan menyiapkan lingkungan pengembangan yang meliputi instalasi Python versi terbaru beserta package manager seperti pip untuk

e-ISSN: 3110-0864 | p-ISSN: xxxx-xxxx
JURNAL ILMU KOMPUTER DAN TEKNIK INFROMATIKA
Volume 2 No 1 Januari 2026 | Page: 55-62
https://journals.raskhamedia.or.id/index.php/juikti
DOI: https://doi.org/10.64803/juikti.v2i1.100

mengelola library yang dibutuhkan. Library kriptografi seperti cryptography atau pycryptodome diinstal untuk mendukung implementasi algoritma AES. Pada sisi frontend, dilakukan instalasi Node.js dan npm (Node Package Manager) sebagai prasyarat untuk menggunakan React.

2.6 Pengujian Dan Analisis

Tahapan terakhir adalah pengujian sistem, yang bertujuan untuk memastikan bahwa sistem enkripsi berfungsi dengan baik dan data dapat dienkripsi serta didekripsi kembali dengan benar. Pengujian dilakukan dengan menguji proses enkripsi dan dekripsi data pada aplikasi yang telah diimplementasikan. Pengujian enkripsi dilakukan dengan cara menginputkan data berupa nama, email, nomor telepon, dan pesan melalui form yang telah dibuat, kemudian sistem akan mengenkripsi data tersebut menggunakan algoritma AES-256 dengan mode CBC. Setelah proses enkripsi selesai, dilakukan pengecekan pada penyimpanan data untuk memastikan bahwa data yang tersimpan benar-benar dalam bentuk ciphertext yang tidak dapat dibaca secara langsung. Hasil enkripsi didokumentasikan dengan mencatat data asli (plaintext) dan hasil enkripsinya (ciphertext) untuk dibandingkan.

3. HASIL DAN PEMBAHASAN

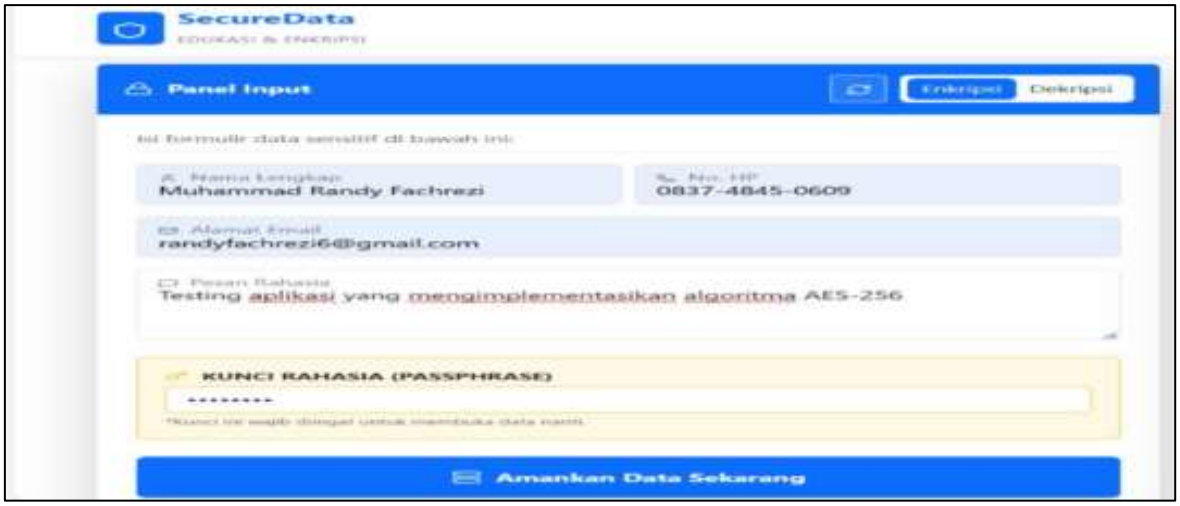
3.1 Skenario Dan Data Uji

Tabel 1. Data Uji Enkripsi Untuk Algoritma AES

No	Nama	Email	Nomor HP	Pesan Rahasia	PassPhrase
1	Muhammad Randy Fachrezi	randyfachrezy6@gmail.com	083748450609	Testing aplikasi yang mengimplementasikan algoritma AES-256	Randy123
2	Alwi Syahputra	alwiisyahputra0@gmail.com	085837775771	Pengujian algoritma AES-256	Alwi334
3	Dwiki Oldi Amsyah	Dwikygg9624@gmail.com	081265840038	Uji Proses enkripsi ke deskripsi	Dikigg

Tabel data uji ini merangkum tiga set data yang digunakan sebagai sampel untuk menguji proses enkripsi dan dekripsi menggunakan algoritma AES pada formulir. Setiap baris merepresentasikan satu skenario pengujian lengkap yang terdiri dari nama, alamat email, nomor HP, serta pesan rahasia yang diinput ke dalam form, kemudian dipasangkan dengan sebuah passphrase yang berbeda untuk setiap pengguna.

3.2 Hasil Enkripsi Formulir



Gambar 2. Halaman Input Data

Pada tampilan ini, pengguna mengisi beberapa field seperti nama lengkap, nomor HP, email, dan pesan rahasia, lalu memasukkan passphrase sebagai kunci rahasia yang nantinya dipakai dalam proses enkripsi dan juga diperlukan kembali saat melakukan dekripsi. Bagian ini penting karena merepresentasikan kondisi awal data masih berupa teks yang bisa dibaca (plaintext) sebelum diproses oleh algoritma, sekaligus

e-ISSN: 3110-0864 | p-ISSN: xxxx-xxxx
JURNAL ILMU KOMPUTER DAN TEKNIK INFROMATIKA
Volume 2 No 1 Januari 2026 | Page: 55-62
https://journals.raskhamedia.or.id/index.php/juikti
DOI: https://doi.org/10.64803/juikti.v2i1.100

memperlihatkan bahwa sistem memang meminta kunci dari pengguna sebagai bagian dari mekanisme pengamanan yang diuji.

Gambar 3. Halaman Hasil Enkripsi Data

Sementara itu, gambar output memperlihatkan hasil setelah tombol enkripsi dijalankan, di mana sistem menampilkan status proses yang berhasil, serta nilai IV yang digenerate dalam format heksadesimal dan hasil enkripsi dalam bentuk ciphertext yang ditampilkan pada gambar. Output ini menggambarkan perubahan yang diharapkan dalam penelitian pada data formulir yang sebelumnya terbaca berubah menjadi rangkaian karakter acak yang tidak bermakna.

3.3 Hasil Deskripsi Formulir

Gambar 4. Halaman Dekripsi Data

Pada tahap ini, pengguna menempelkan kode ciphertext ke dalam kolom yang disediakan, kemudian memasukkan passphrase pada bagian “Kunci Rahasia”, yaitu kunci yang sebelumnya digunakan saat proses enkripsi. Tampilan tombol “Buka Kunci Data” menandai bahwa sistem baru akan mencoba melakukan proses dekripsi setelah kedua input tersebut diberikan, sehingga langkah ini menjadi pintu masuk utama untuk menguji apakah kombinasi ciphertext dan passphrase yang dimasukkan benar.

e-ISSN: 3110-0864 | p-ISSN: xxxx-xxxx
JURNAL ILMU KOMPUTER DAN TEKNIK INFROMATIKA
Volume 2 No 1 Januari 2026 | Page: 55-62
https://journals.raskhamedia.or.id/index.php/juikti
DOI: https://doi.org/10.64803/juikti.v2i1.100

Gambar 5. Halaman Hasil Dekripsi Data

Gambar output diatas memperlihatkan hasil setelah proses dekripsi berhasil dilakukan dengan passphrase yang tepat. Pada bagian atas tercantum status “Sukses” dan algoritma yang digunakan, yaitu AES-256-CBC, diikuti panel berisi hasil dekripsi dalam bentuk plaintext yang kembali menampilkan nama, email, nomor HP, dan pesan persis seperti saat data pertama kali diinput. Pesan “Verifikasi Kunci Berhasil” di bagian bawah menegaskan bahwa passphrase yang dimasukkan cocok dengan kunci yang digunakan saat enkripsi, sehingga sistem mampu mengembalikan ciphertext menjadi data asli secara utuh

3.4 Pengujian PassPhrase

Gambar 6. Pengujian PassPhrase

Gambar ini menunjukkan hasil percobaan dekripsi yang gagal pada, di mana pengguna telah memasukkan ciphertext dan sebuah passphrase, namun sistem menampilkan pesan kesalahan. Pesan ini muncul karena passphrase yang dimasukkan tidak cocok dengan passphrase yang digunakan saat data tersebut dienkripsi, sehingga algoritma AES tidak dapat menghasilkan kunci derivasi yang tepat untuk membuka ciphertext dan mengembalikannya ke bentuk plaintext.

Sebaliknya, jika passphrase yang dimasukkan benar saat proses dekripsi sistem akan menampilkan output terminal berstatus "Sukses" disertai informasi algoritma AES-256-CBC, kemudian memunculkan seluruh data asli (nama lengkap, email, nomor HP, dan pesan rahasia) dalam bentuk plaintext yang dapat dibaca kembali, serta menampilkan pesan konfirmasi "Verifikasi Kunci Berhasil" seperti yang terlihat pada hasil pengujian sebelumnya.

4. KESIMPULAN

Hasil penelitian ini menunjukkan bahwa algoritma AES-256-CBC dapat diterapkan secara efektif untuk mengamankan data formulir pada aplikasi web. Algoritma ini bekerja dengan mengubah informasi sensitif yang semula mudah dibaca seperti nama, email, nomor HP, dan pesan menjadi data tersandi (ciphertext) yang tidak dapat dipahami tanpa kunci yang sesuai. Pengujian dilakukan menggunakan tiga set data uji, dan hasilnya menunjukkan bahwa seluruh data input berhasil dienkripsi dengan baik, sehingga isi data tidak lagi dapat dikenali secara langsung. Temuan penting dari pengujian ini terletak pada proses dekripsi. Ketika passphrase yang dimasukkan sama dengan passphrase yang digunakan saat proses enkripsi, sistem mampu mengembalikan seluruh data ke bentuk aslinya secara utuh. Sebaliknya, apabila passphrase yang dimasukkan tidak sesuai, sistem secara otomatis menolak proses dekripsi dan menampilkan pesan kesalahan yang jelas. Hal ini menunjukkan bahwa data yang telah dienkripsi tidak dapat diakses oleh pihak yang tidak memiliki kunci yang benar. Berdasarkan hasil tersebut, dapat disimpulkan bahwa tujuan penelitian untuk membuktikan mekanisme kerja algoritma AES dalam proses enkripsi dan dekripsi data formulir telah tercapai. Passphrase terbukti berperan sebagai elemen kunci yang menentukan keberhasilan akses terhadap data terenkripsi. Penelitian ini diharapkan dapat menjadi referensi praktis bagi pengembang sistem web dalam menerapkan mekanisme enkripsi sederhana namun efektif, khususnya untuk melindungi data input pengguna agar tidak tersimpan dalam bentuk teks biasa yang rentan terhadap akses tidak sah.

e-ISSN: 3110-0864 | p-ISSN: xxxx-xxxx
JURNAL ILMU KOMPUTER DAN TEKNIK INFROMATIKA
Volume 2 No 1 Januari 2026 | Page: 55-62
https://journals.raskhamedia.or.id/index.php/juikti
DOI: https://doi.org/10.64803/juikti.v2i1.100

REFERENCES

[1] OWASP Foundation, "OWASP Top 10:2021 - The Ten Most Critical Web Application Security Risks," OWASP Official Website. [Online]. Available: https://owasp.org/Top10/2021/

[2] A. A. Adesola, "A review of the cryptographic approaches to data security: The impact of quantum computing, evolving challenges and future solutions," *World J. Adv. Res. Rev.*, vol. 25, no. 2, pp. 1916–1924, 2025, doi: 10.30574/wjarr.2025.25.2.0434.

[3] P. Alifia Rizky and S. Soim, "RESISTOR Journal | 71 Implementasi Algoritma Kriptografi AES CBC Untuk Keamanan Komunikasi Data Pada Hardware," *Resistor*, vol. 3, no. 1, pp. 71–78, 2024, [Online]. Available: https://s.id/jurnalresistor

[4] N. T. Jehian *et al.*, "Pengembangan sistem Keamanan Data Berbasis Web Menggunakan ALGORITMA CHACHA20-POLY1305 DAN ARGON2," *JITET (Jurnal Inform. dan Tek. Elektro Ter.*, vol. 13, no. 3, 2025.

[5] M. Tania, T. S. Alasi, and R. Yap, "Algoritma Aes Untuk Keamanan Data Digital Berbasis Web Di Kantor Desa Aman Damai," *J. TIMES*, vol. 13, no. 2, pp. 142–149, 2024, doi: 10.51351/jtm.13.2.2024781.

[6] A. H. Nasrullah, "Secure Web-Based File Encryption Using AES-128," *J. Embed. Syst. Secur. Intell. Syst.*, vol. 6, no. 2, pp. 146–155, 2025, doi: 10.59562/jessi.v6i2.8436.

[7] N. A. Khoirunnisa and R. Satra, "Implementasi Algoritma Kriptografi AES untuk Peningkatan Keamanan Proses Login Website," *LINIER Lit. Inform. dan Komput.*, vol. 2, no. 3, pp. 317–328, 2025.

[8] R. M. Liauren, B. Zaman, and S. Bahri, "Implementasi Algortima Aes Dan Bcrypt Untuk Pengamanan Data Pengguna Pada Website Jahitku," *KHARISMA Tech*, vol. 20, no. 1, pp. 57–71, 2025, doi: 10.55645/kharismatech.v20i1.535.

[9] M. Joshi, "Enhancing Cloud Data Security Through AES Encryption and SHA-256 Integrity Verification," *Int. J. Multidiscip. Innov. Res.*, vol. 6, no. 1, pp. 28–37, 2025.

[10] J. Pierre, "Advanced encryption standard," Gaithersburg, 2023. doi: 10.1201/b17707.

[11] H. Listia, S. Wardaniah, and Z. Indra, "Aplikasi Enkripsi dan Deskripsi Teks Menggunakan Vernam Chiper Berbasis Web," *J. Educ. Transp.*, vol. 1, no. 2, pp. 596–604, 2024, doi: 10.57235/jetbus.v1i2.4164.

[12] F. Nuraeni, "Implementasi Superenkripsi DSA dan Aes 128 Bit Dalam Pengamanan File Surat Digital," *J. Algoritm.*, vol. 22, pp. 601–613, 2025, doi: 10.33364/algoritma/v.22-1.1832.

[13] P. D. Pitroda, B. C. Donga, D. H. B. Domadiya, and D. D. H. Domadiya, "Beyond The Basics: A Detailed Survey of Advanced Python Applications and Innovations," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 12, no. 10, pp. 94–97, 2024, doi: 10.22214/ijraset.2024.64457.

[14] F. Akmal, "Journal of Digital Business and Technology Innovation (DBESTI) PENERAPAN REACTJS DALAM PENGEMBANGAN FRONT-END APLIKASI," *J. Digit. Bus. Technol. Innov.*, vol. 2, no. 2, pp. 198–205, 2025.

[15] R. Jain, V. Shrivastava, A. Pandey, and A. Sharma, "Modern Web Development using CSS & HTML," *Int. J. Emerg. Sci. Eng.*, vol. 12, no. 6, pp. 13–16, 2024, doi: 10.35940/ijese.g2574.12060524.