LINIER
LITERATUR INFORMATIKA & KOMPUTER
e-ISSN: 3063-2218
DOI : 10.33096/linier.v2i3.3143
Literatur Informatika & Komputer
Vol 2, No 3 (2025), pp 317-328

Implementasi Algoritma Kriptografi AES untuk Peningkatan Keamanan Proses Login Website

Nur Azizah Khoirunnisa[a], Ramdan Satra[b], Dewi Widyawati [c]
Universitas Muslim Indonesia, Makassar, Indonesia
Email: [a]13020200242@umi.ac.id; [b]ramdan@umi.ac.id; [c]dewiwidyawati@umi.ac.id

Received: 23-02-2025 | Revised: 02-08-2025 | Accepted: 12-09-2025 | Published: 29-09-2025

Abstrak

Keamanan dalam proses *login* pada *website* merupakan aspek yang sangat penting untuk mencegah akses tidak sah dan pencurian data pengguna. Salah satu metode yang dapat digunakan untuk meningkatkan keamanan adalah dengan menerapkan algoritma *Advanced Encryption Standard* (AES) dalam proses penyimpanan dan verifikasi *password*. AES merupakan algoritma enkripsi simetris yang dikenal memiliki tingkat keamanan tinggi dan efisiensi dalam proses enkripsi serta dekripsi data. Penelitian ini membahas implementasi AES untuk enkripsi *password* pada sistem *login* berbasis *PHP Native*. Sistem ini dirancang untuk menyimpan *password* dalam bentuk terenkripsi guna mencegah akses tidak sah terhadap informasi pengguna. Selain itu, pengujian dilakukan untuk memastikan integritas proses enkripsi serta efektivitasnya dalam melindungi data pengguna. Hasil implementasi menunjukkan bahwa penggunaan AES dapat meningkatkan keamanan penyimpanan *password* di *database* dengan mencegah pencurian data melalui eksploitasi langsung terhadap *database*. Kesimpulan dari penelitian ini adalah bahwa metode enkripsi AES dapat diterapkan secara efektif dalam sistem *login* berbasis web guna meningkatkan keamanan autentikasi pengguna.

Kata kunci: Keamanan *Website,* Kriptografi, *Advanced Encryption Standard* (AES), *Login*.

Pendahuluan

Dalam era digital yang terus berkembang, keamanan sistem autentikasi pada *website* menjadi aspek yang sangat krusial untuk melindungi data sensitif pengguna dari berbagai ancaman siber. *Form login* sebagai pintu utama akses ke dalam sistem sering kali menjadi target serangan akibat kelemahan dalam mekanisme autentikasi dan enkripsi data. Beberapa potensi kerentanan yang umum terjadi meliputi lemahnya pengelolaan kata sandi, tidak digunakannya algoritma enkripsi yang kuat, serta kurangnya perlindungan terhadap transmisi data sensitif. Celah keamanan ini dapat dimanfaatkan oleh pihak yang tidak bertanggung jawab untuk melakukan berbagai jenis serangan yang berpotensi mengancam integritas serta kerahasiaan data pengguna. Oleh karena itu, diperlukan solusi yang efektif untuk meningkatkan keamanan proses *login* pada *website.*

Salah satu metode yang dapat diterapkan adalah penggunaan algoritma kriptografi *Advanced Encryption Standard* (AES) dalam proses autentikasi. AES dikenal sebagai algoritma enkripsi yang kuat, efisien, dan telah digunakan secara luas untuk melindungi data digital. Dengan mengimplementasikan AES, informasi sensitif seperti kata sandi dapat dienkripsi sebelum disimpan atau dikirim, sehingga mengurangi risiko pencurian data oleh pihak yang tidak berwenang. Penelitian ini bertujuan untuk mengimplementasikan algoritma AES dalam proses *login website* guna meningkatkan keamanan autentikasi pengguna serta mencegah berbagai potensi serangan siber [1].

Penelitian terkait keamanan *website* telah banyak dilakukan oleh para peneliti sebelumnya untuk menghadapi ancaman. Misalnya, penelitian oleh Hermawan et al. (2021) membahas implementasi kombinasi algoritma AES dan RSA untuk meningkatkan keamanan data pada sistem informasi, yang menunjukkan efektivitas enkripsi dalam melindungi informasi dari akses tidak sah [2]. Penelitian lain oleh Fadlullah et al. (2023) menerapkan algoritma AES pada autentikasi *login* sistem informasi, yang berhasil meningkatkan ketahanan dengan penerapan enkripsi kunci secara aman [3]. Selain itu, Khoirudin dan Windarto (2024) menyoroti penerapan algoritma AES-512 untuk pengamanan data berbasis web, di mana AES terbukti mampu menjaga integritas data selama proses transmisi [4].

Meskipun berbagai metode keamanan telah diterapkan, masih terdapat banyak sistem yang belum mengimplementasikan enkripsi yang optimal untuk melindungi data *login* pengguna. Oleh karena itu, penelitian ini bertujuan untuk meningkatkan keamanan proses *login website* dengan menerapkan algoritma

Advanced Encryption Standard (AES). Algoritma AES dipilih karena memiliki tingkat keamanan yang tinggi, efisiensi dalam proses enkripsi dan dekripsi, serta telah menjadi standar dalam berbagai sistem keamanan data.

Berdasarkan urgensi tersebut, penelitian ini mengangkat judul "Implementasi Algoritma Kriptografi AES untuk Peningkatan Keamanan Proses *Login Website*". Fokus utama penelitian ini adalah mengoptimalkan sistem *login* melalui penerapan algoritma AES guna meningkatkan perlindungan data pengguna. Dengan implementasi ini, diharapkan keamanan proses autentikasi pengguna dapat ditingkatkan, sehingga risiko kebocoran data dapat diminimalkan dan kepercayaan pengguna terhadap sistem semakin terjaga.

Metode

A. Keamanan *Website*

Dalam era perkembangan teknologi yang pesat, sistem *online* dan *website* memiliki peran penting dalam berbagai aspek kehidupan. Namun, hal ini juga meningkatkan kebutuhan akan kualitas keamanan *website* yang lebih baik untuk melindungi data dan informasi. Kebocoran informasi pada *website* kepada pihak yang tidak berkepentingan dapat menyebabkan kerugian besar bagi pemilik data, baik secara finansial maupun reputasi. Saat ini, pengetahuan tentang teknik peretasan, seperti *cracking* dan *hacking*, semakin mudah diakses oleh publik. Ditambah lagi, banyak tersedia alat atau tools yang dapat digunakan untuk melancarkan serangan terhadap server. Kondisi ini menuntut penerapan langkah-langkah keamanan yang lebih kuat pada sistem *website*, seperti penggunaan algoritma enkripsi yang handal, pengelolaan akses yang ketat, dan penerapan protokol keamanan untuk mencegah terjadinya kebocoran data atau peretasan [5].

B. Kriptografi

Kriptografi berasal dari dua kata dalam bahasa Yunani, yaitu "*crypto*" yang berarti rahasia (*secret*) dan "*grapho*" yang berarti menulis (*writing*). Dengan demikian, kriptografi merupakan ilmu dan seni yang digunakan untuk melindungi keaslian serta keabsahan pesan, di mana pesan dijaga kerahasiaannya dengan cara disandikan ke dalam bentuk yang tidak dapat dimengerti oleh pihak yang tidak berwenang [6][7]. Lebih lanjut, kriptografi mempelajari teknik-teknik matematika yang berkaitan dengan aspek keamanan informasi, seperti kerahasiaan, keabsahan, integritas, dan autentikasi data [8][9]. Prinsip dasar kriptografi terletak pada penggunaan metode pengiriman pesan secara diam-diam, sehingga hanya penerima pesan yang dituju yang dapat mengembalikan penyamaran tersebut dan membaca pesan aslinya. Dalam kerangka tersebut, terdapat dua proses utama, yaitu enkripsi dan dekripsi. Enkripsi adalah proses mengubah pesan asli (*plain text)* menjadi pesan bahasa kode (*cipher text*), sedangkan dekripsi adalah proses mengembalikan pesan yang telah disandikan kembali ke bentuk aslinya [10].

C. *Advanced Encryption Standard* (AES)

AES (*Advanced Encryption Standard*) adalah algoritma kriptografi simetris yang berfungsi untuk mengenkripsi dan mendekripsi data. Algoritma ini diperkenalkan oleh *National Institute of Standards and Technology* (NIST) pada tahun 2001 sebagai pengganti *Data Encryption Standard* (DES) yang sudah berakhir masa penggunaannya [11]. Sejak saat itu, AES telah menjadi standar utama dalam dunia kriptografi dan banyak diterapkan dalam berbagai aplikasi serta protokol keamanan. AES menggunakan block cipher dengan ukuran blok tetap sebesar 128 bit, serta mendukung panjang kunci 128, 192, atau 256 bit. Dalam proses kerjanya, algoritma ini mengonversi teks terang (*plaintext*) menjadi teks terenkripsi (*ciphertext*) menggunakan kunci yang sama, sehingga menghasilkan data yang sulit diretas tanpa kunci yang tepat. Proses dekripsi dilakukan dengan kunci yang sama untuk mengembalikan teks terenkripsi menjadi teks terang seperti semula [12].

Perbedaan ketiga jenis kunci 128, 192, 256-bit terletak pada panjang kunci yang mempengaruhi jumlah perputaran (*round*) yang bisa digambarkan pada gambar berikut.

Tabel 1. Perbedaan 3 Jenis Algoritma AES

	Panjang Kunci	Panjang Blok	Jumlah Putaran
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Proses enkripsi dan deskripsi algoritma AES akan melakukan 4 tahapan proses, yaitu *SubBytes, ShiftRows, MixColumn dan addArroundKey.* Proses tahapan yang dilakukan seperti:

1. *SubBytes* Berfungsi untuk menukar isi dari *byte* dengan menggunakan tabel substitusi (S-BOX).
2. *ShiftRows* Proses pergeseran blok per baris pada *state array.*
3. *MixColumn* Proses mengalikan blok data (pengacakan) pada setiap *state array.*
4. *array* dan *round key* dengan hubungan XOR.

Untuk proses dekripsi menggunakan tahap berikut:

1. *InvShiftRows* Melakukan pergeseran bit ke kanan pada setiap blok baris.
2. *InvSubBytes* Setiap elemen pada state dipetakan dengan tabel *Inverse S-Box.*
3. *InvMixColumn* Setiap kolom dalam *state* dikalikan dengan matriks AES.
4. *AddRoundKey* Menggabungkan *state array* dan *round key* dengan hubungan XOR [13][14][15].

Perancangan

A. Struktur Sistem

Tabel 2. Struktur Direktori

Direktori/File	Deskripsi
/project-aes-login	Direktori utama sistem
/css/style.css	File CSS untuk desain tampilan
/config/db.php	File konfigurasi koneksi database
/config/config.php	File konfigurasi AES
/auth/register.php	Proses registrasi pengguna
/auth/login.php	Proses login pengguna
register_form.php	Halaman form registrasi
login_form.php	Halaman form login
index.php	Halaman utama setelah login
logout.php	Proses logout pengguna

B. Alur Flowchart

Pada penerapan AES, proses enkripsi dan dekripsi digunakan sesuai dengan kebutuhan untuk menjaga kerahasiaan data sensitif. Maka penerapannya dalam konteks proses *login* terbagi menjadi dua skema yaitu penerapan AES dengan hanya menggunakan enkripsi dan penerapan AES dengan menggunakan enkripsi-dekripsi.

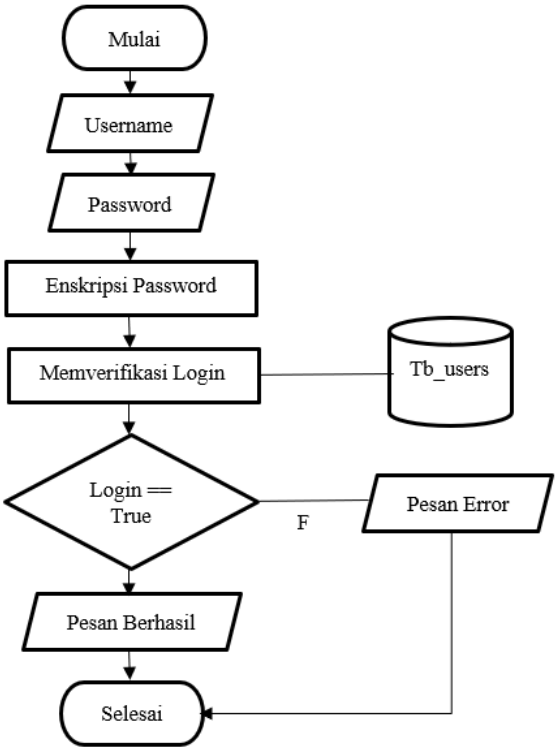
1. Dalam Proses Enkripsi
   a. Enkripsi di proses daftar *user*/*create user* (Registrasi)
      Ketika pengguna mendaftar, *password* yang dimasukkan akan dienkripsi menggunakan AES sebelum disimpan ke dalam *database*. Dengan begitu, *password* tidak disimpan dalam bentuk teks asli (*plain text*), melainkan dalam bentuk terenkripsi.

      Flowchart pada Gambar 1 dijelaskan sebagai berikut:

      1. Mulai: Proses *login* dimulai ketika pengguna mengakses halaman *login.*
      2. *Username:* Pengguna memasukkan *username* yang telah terdaftar di sistem.
      3. *Password*: Pengguna memasukkan *password* sebagai kredensial autentikasi.
      4. Enkripsi *Password*: *Password* yang dimasukkan dienkripsi menggunakan algoritma AES sebelum dibandingkan dengan data yang tersimpan di database.
      5. Memverifikasi *Login*: Sistem mengambil data pengguna dari *table users* dan membandingkan *username* serta *password* terenkripsi dengan yang tersimpan di *database*.
      6. Pengecekan *Login*

1) Jika *login* == *true* (*username* dan *password* sesuai), sistem menampilkan Pesan Berhasil dan pengguna diberikan akses ke sistem.
2) Jika *login* == *false* (*username* atau *password* tidak sesuai), sistem menampilkan Pesan *Error*, memberi tahu pengguna bahwa *login* gagal.

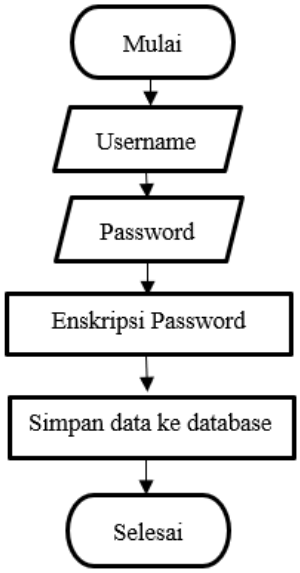
7. Selesai: Proses *login* berakhir baik berhasil maupun gagal.



Gambar 1. Flowchart Daftar *User/create user*

b. Enkripsi di proses *login*

Ketika pengguna mencoba *login, password* yang diinputkan juga dienkripsi dengan algoritma dan kunci yang sama. Hasil enkripsi ini kemudian dibandingkan dengan nilai terenkripsi yang ada di *database*. Jika keduanya cocok, maka proses autentikasi dianggap berhasil.

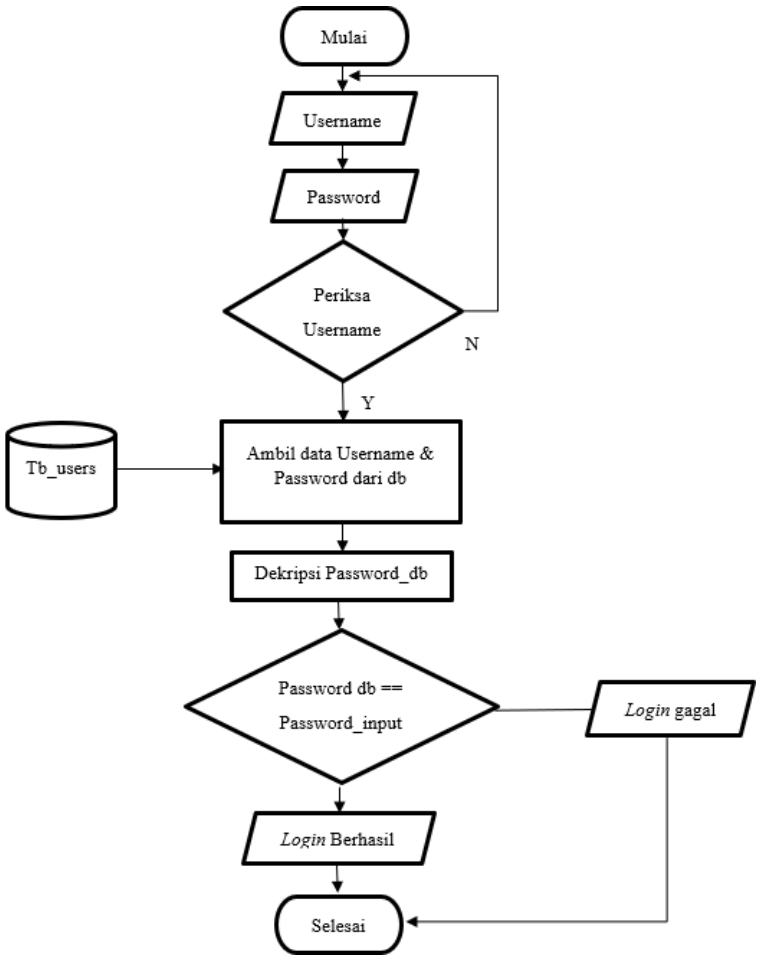


Gambar 2. Flowchart Proses *Login*

Flowchart pada Gambar 2 dijelaskan sebagai berikut:

1. Mulai: Proses dimulai ketika pengguna mengakses form pendaftaran atau *login* pada *website*.
2. *Username*: Pengguna memasukkan *username* yang akan digunakan untuk proses autentikasi.
3. *Password*: Pengguna memasukkan *password* yang akan digunakan sebagai kredensial *login.*
4. Enkripsi *Password*: Sebelum disimpan ke dalam *database, password* dienkripsi menggunakan algoritma AES untuk meningkatkan keamanan dan mencegah akses tidak sah.
5. Simpan Data ke *Database*: Setelah dienkripsi, *password* bersama *username* disimpan ke dalam *database.*
6. Selesai: Proses penyimpanan data selesai, dan pengguna dapat melanjutkan ke tahap *login*.

2. Dalam Proses Dekrispi
   a. Dekripsi saat *Login* (Pengambilan Data Asli)
   Proses dekripsi digunakan untuk mengembalikan data terenkripsi ke bentuk aslinya ketika diperlukan. Misalnya, jika ada data sensitif selain *password* yang dienkripsi dan perlu ditampilkan atau diproses lebih lanjut dalam bentuk asli, maka proses dekripsi akan diterapkan



Gambar 3. Flowchart Dekripsi

Flowchart pada Gambar 3 dijelaskan sebagai berikut:

1. Mulai: Proses *login* dimulai ketika pengguna mengakses halaman *login.*
2. *Username:* Pengguna memasukkan *username* pada halaman *login.*
3. *Password*: Pengguna memasukkan *password* sebagai kredensial autentikasi.
4. Periksa *Username*: Sistem memeriksa apakah *username* yang dimasukkan tersedia dalam *database.*

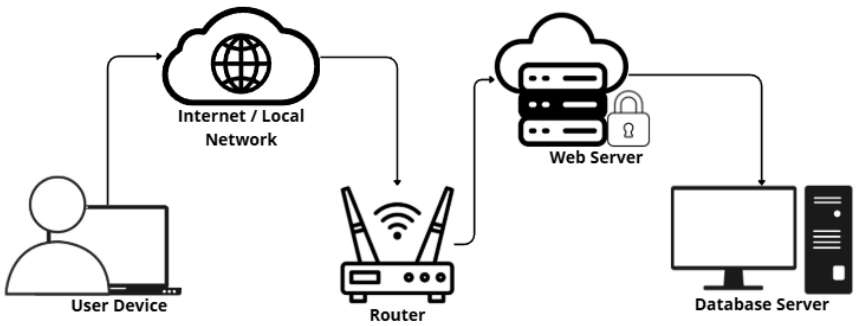
1) Jika *username* tidak ditemukan, maka proses dikembalikan ke langkah 2 untuk meminta pengguna memasukkan kembali *username* yang benar.
2) Jika *username* ditemukan, lanjut ke langkah berikutnya.

5. Ambil Data dari *Database*: Sistem mengambil data *username* dan *password* terenkripsi dari tabel *users* dalam *database.*
6. Dekripsi *Password*: *Password* yang tersimpan dalam database akan didekripsi menggunakan algoritma AES.
7. Bandingkan *Password*: Sistem membandingkan *password* hasil dekripsi dengan *password* yang diinputkan pengguna.
   1) Jika *login* sesuai, lanjut ke langkah berikutnya
   2) Jika tidak sesuai, pengguna akan mendapatkan notifikasi *Login* Gagal dan harus kembali ke langkah 2 untuk mencoba *login* lagi.
8. *Login* Berhasil: Jika *password* cocok, pengguna berhasil *login* dan diarahkan ke halaman utama.
9. Selesai: Proses login selesai dan pengguna dapat mengakses sistem.

Pada proses *login* dengan *password,* dekripsi umumnya tidak diperlukan karena yang dibandingkan adalah hasil enkripsi dari *password* yang dimasukkan pengguna dengan nilai terenkripsi yang tersimpan di *database.* Dengan pendekatan ini, *password* asli tidak pernah perlu didekripsi atau diungkapkan, sehingga meningkatkan tingkat keamanan sistem. Juga, skema yang diterapkan menggunakan AES tanpa dekripsi terbukti lebih efisien karena jumlah tahapan yang diperlukan lebih sedikit. Selain itu, penyimpanan *password* dalam bentuk *cipher text* memastikan bahwa data sensitif tidak tersimpan dalam format *plaintext,* yang tentunya sejalan dengan prinsip keamanan data. Menurut penulis, pendekatan ini tidak hanya menyederhanakan proses autentikasi tetapi juga mengurangi risiko kebocoran informasi, karena tidak ada tahap dekripsi yang dapat membuka celah keamanan. Dengan demikian, penulis hanya menerapkan skema pertama yaitu AES *login* dengan enkripsi saja, sehingga solusi yang diambil memberikan keseimbangan optimal antara efisiensi dan keamanan yang menjadi prioritas utama.

Pemodelan

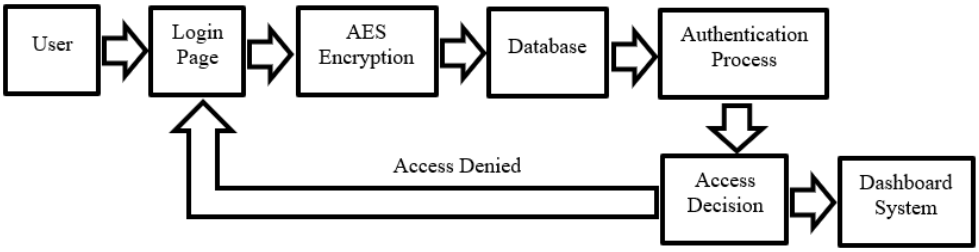
A. Topologi Jaringan



Gambar 4. Tolpologi *Client Server*

Topologi jaringan yang digunakan dalam sistem ini adalah topologi *client-server* berbasis web. Dalam topologi ini, *User Device* (laptop, PC, atau *smartphone*) mengakses layanan melalui Internet atau Jaringan Lokal (LAN/Wi-Fi). Koneksi ini diarahkan oleh Router, yang meneruskan permintaan ke Web Server. Web Server bertanggung jawab untuk memproses autentikasi pengguna dengan metode kriptografi AES, sebelum mengakses *Database Server* untuk validasi data *login*.

B. *Block Diagram System*

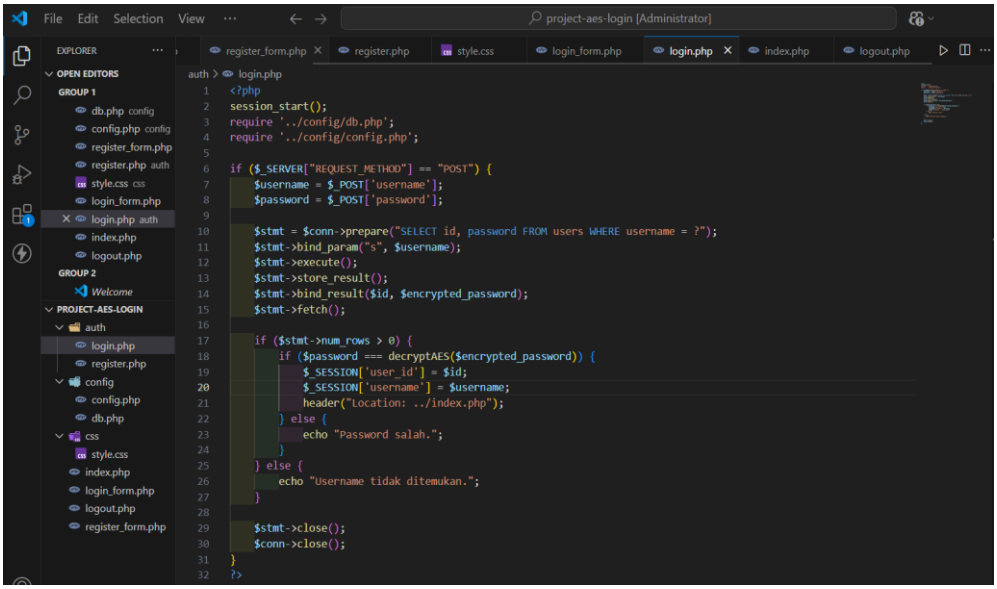


Gambar 5. *Block Diagram System*

Diagram ini menggambarkan alur autentikasi *login* dengan enkripsi AES untuk meningkatkan keamanan sistem. Proses dimulai ketika pengguna memasukkan kredensial pada halaman *login*, yang kemudian dienkripsi menggunakan algoritma AES sebelum dikirim ke *database.* Sistem kemudian melakukan autentikasi dengan mencocokkan data yang diberikan dengan informasi yang tersimpan dalam *database*. Jika kredensial valid, pengguna diberikan akses ke *Dashboard System*, sedangkan jika tidak valid, sistem menampilkan pesan *Access Denied* dan mengarahkan pengguna kembali ke halaman *login* untuk mencoba kembali. Dengan implementasi enkripsi AES, keamanan data pengguna lebih terjamin, terutama dalam melindungi informasi sensitif dari potensi ancaman siber.

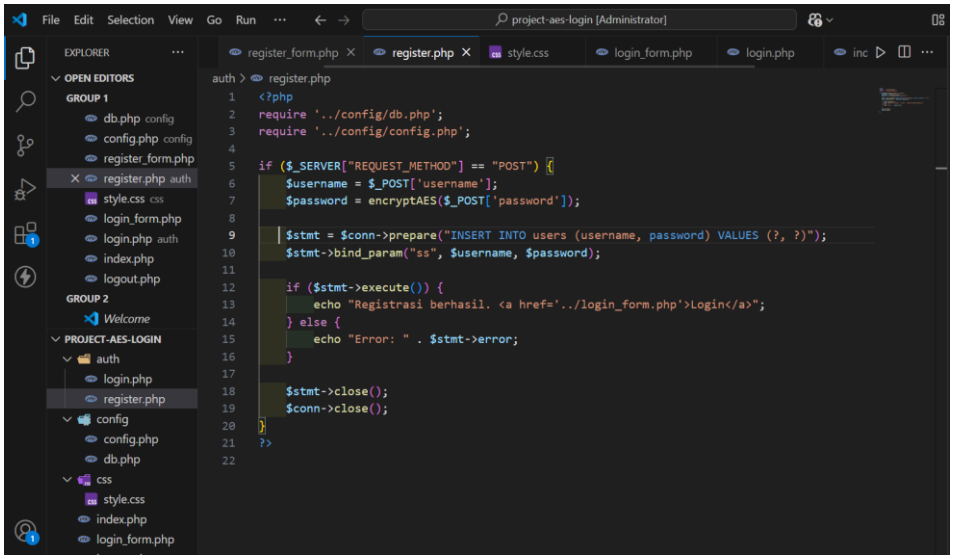
C. *Advanced Encryption Standard* (AES)

Berikut adalah *source code* yang digunakan untuk enkripsi dan dekripsi *password* pada *form login* dan *register:*



Gambar 6. *Source code Login*

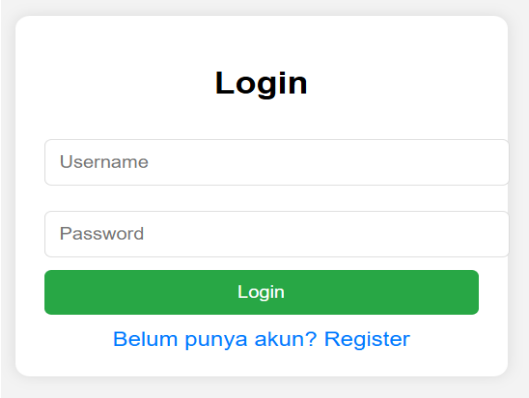
Gambar 6 menunjukkan implementasi algoritma AES dalam proses autentikasi pengguna pada fitur login. Ketika pengguna memasukkan *username* dan *password* pada *form login,* sistem akan mengenkripsi *password* menggunakan algoritma AES sebelum dibandingkan dengan data yang tersimpan di *database.*



Gambar 7. *Source code Register*

Gambar 7 menunjukkan implementasi algoritma AES untuk mengenkripsi *password* pada fitur registrasi. Ketika pengguna memasukkan *username* dan *password* pada form registrasi, sistem akan secara otomatis mengenkripsi *password* menggunakan algoritma AES sebelum disimpan ke dalam *database.* Dengan metode ini, *password* tidak akan disimpan dalam bentuk *plaintext,* melainkan dalam bentuk terenkripsi yang lebih aman.

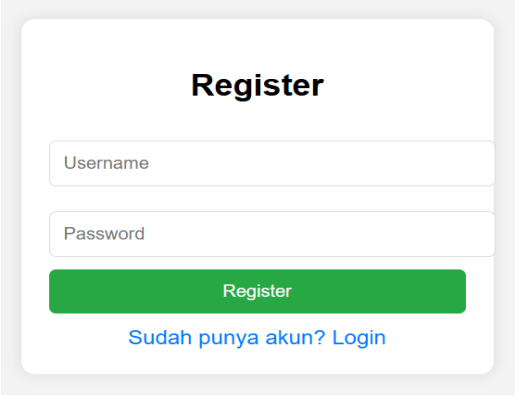
1. Halaman Login



Gambar 8. *Form Login*

Gambar 8 menampilkan fitur *login* pada *website.* Untuk dapat mengakses halaman utama, pengguna diwajibkan terlebih dahulu melakukan *login* dengan memasukkan *username* dan *password.*

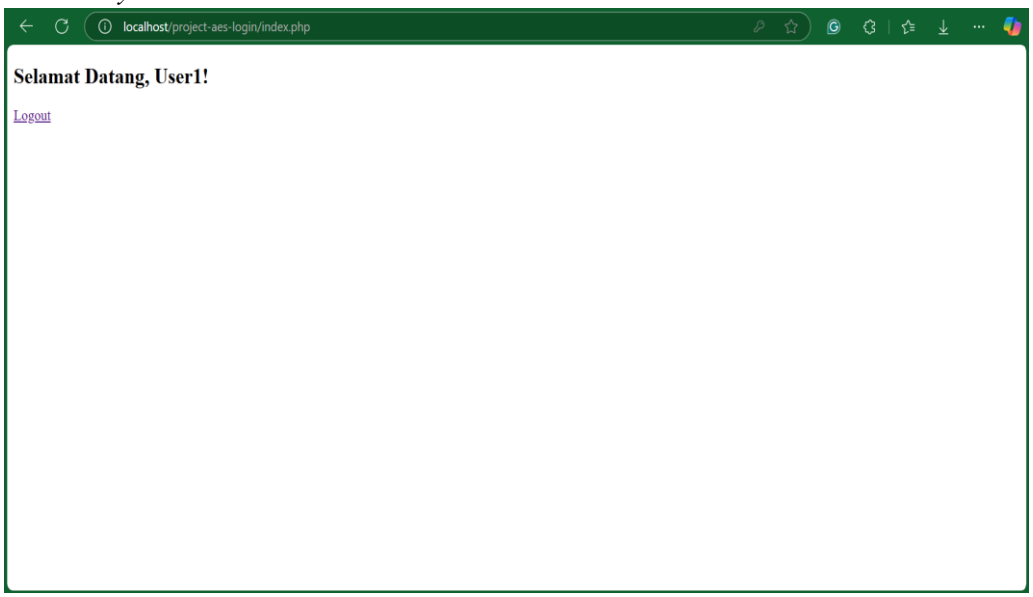
2. Halaman Registrasi



Gambar 9. *Form Register*

Gambar 9 di atas menunjukkan tampilan fitur pendaftaran (*register*) pada *website.* Jika pengguna belum memiliki akun, maka harus melakukan registrasi terlebih dahulu. Dalam proses ini, pengguna diminta untuk memasukkan data seperti *username* dan *password,* kemudian dapat melanjutkan dengan mengklik tombol *register*.

3. *Dashboard System*

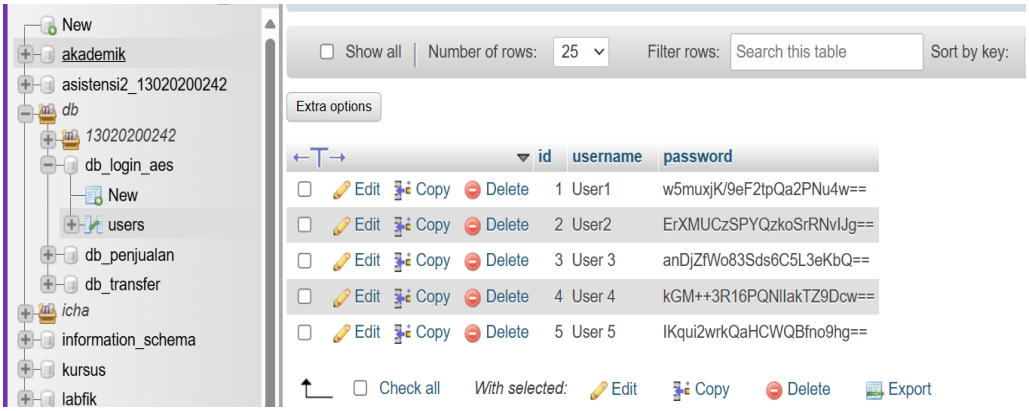


Gambar 10. *Dashboard System*

Gambar 10 menunjukkan tampilan *dashboard system* pada *website.* Setelah proses registrasi berhasil, pengguna akan diarahkan ke halaman *Dashboard system.*

4. *Database Server*

Pada Gambar 11 menunjukkan hasil implementasi enkripsi *password* menggunakan algoritma *Advanced Encryption Standard* (AES) pada tabel *users* dalam database *db_login_aes*. Setiap *password* yang dimasukkan oleh pengguna telah dienkripsi sebelum disimpan ke dalam *database*, sehingga meningkatkan keamanan data dan melindungi informasi pengguna dari akses yang tidak sah. Dengan metode ini, meskipun data dalam *database* diakses oleh pihak yang tidak berwenang, *password* tetap dalam bentuk terenkripsi dan tidak dapat dibaca secara langsung



Gambar 11. *Database Server*

.

Kesimpulan

Implementasi AES dalam proses *login website* berhasil meningkatkan keamanan dengan mencegah penyimpanan *password* dalam bentuk teks biasa. Pengujian menunjukkan bahwa enkripsi AES membuat *password* lebih sulit untuk diakses oleh pihak yang tidak berwenang. Dengan sistem ini, keamanan autentikasi pengguna dapat ditingkatkan secara signifikan. Hasil implementasi menunjukkan bahwa penggunaan AES dapat meningkatkan keamanan penyimpanan *password* di *database* dengan mencegah pencurian data melalui eksploitasi langsung terhadap *database*. Kesimpulan dari penelitian ini adalah bahwa metode enkripsi AES dapat diterapkan secara efektif dalam sistem *login* berbasis web guna meningkatkan keamanan autentikasi pengguna. Selain itu, kombinasi AES dengan metode keamanan tambahan seperti *hashing* dan tokenisasi dapat memberikan perlindungan yang lebih optimal terhadap ancaman keamanan digital. Untuk pengembangan lebih lanjut, dapat ditambahkan fitur seperti *multi-factor authentication* (MFA) serta sistem monitoring keamanan berbasis kecerdasan buatan guna mendeteksi ancaman siber secara lebih proaktif.

Daftar Pustaka

[1] A. Almadira, Y. Pratama, and Fenny Purwani, "Melindungi Data Di Dunia Digital: Peran Strategis Enkripsi Dalam Keamanan Data," *J. Sci. Res. Dev.*, vol. 6, no. 2, pp. 540–549, 2024, [Online]. Available: https://idm.or.id/JSCR/index.php/JSCR

[2] A. Hermawan and H. I. E. Ujianto, "Implementasi Enkripsi Data Menggunakan Kombinasi AES dan RSA," *J. Nas. Inform. dan Teknol.*, vol. 5, no. 2, pp. 325–330, 2021.

[3] Fadlullah Fadlullah *et al.*, "Implementasi Algoritma AES pada Autentikasi Login Sistem Informasi," *J. Bintang Pendidik. Indones.*, vol. 1, no. 2, pp. 251–263, 2023, doi: 10.55606/jubpi.v1i2.1420.

[4] N. H. Khoirudin, "Penerapan Algoritme Advanced Encryption Standard (AES-512) untuk Pengamanan File Berbasis Web Application of Advanced Encryption Standard (AES-512) Algorithm for Web-Based File Security," vol. 4, pp. 62–71, 2024.

[5] M. Satriawan and H. S. Y, "Pendeteksi Serangan Brute Force Pada Keamanan website Berbasi Mobile," *J. Juara*, vol. 2, no. 2, pp. 2798–3315, 2022.

[6] S. Rediansyah, R. T. Shita, and ..., "Pengamanan File Berbasis Web Dengan Menerapkan Algoritme Aes-128 Pada Pt. Samudra Katulistiwa Nusantara," *Pros. Semin. ...*, vol. 2, no. April, pp. 166–174, 2023, [Online]. Available: https://senafti.budiluhur.ac.id/index.php/senafti/article/view/666%0Ahttps://senafti.budiluhur.ac.id/index.php/senafti/article/download/666/312

[7] R. Suriadi, R. Satra, and F. Fattah, "Peningkatan Keamanan Data dengan Menggunakan Equation pada Metode Playfair Cipher," *Bul. Sist. Inf. dan Teknol. Islam*, vol. 1, no. 4, pp. 266–269, 2020, doi: 10.33096/busiti.v1i4.685.

[8] L. Firdaus *et al.*, "Jurnal Sustainable : Jurnal Hasil Penelitian dan Industri Terapan Penerapan Kombinasi Algoritma Advanced Encryption Standard (AES) dan Elgamal Dengan Fungsi Secure Hash Algorithm (SHA) Dalam Penyandian File Dokumen," vol. 11, no. 01, 2022.

[9] D. A. A. Reski Pratiwi, F. Fattah, "Implementasi Kriptografi RSA Pada Aplikasi Chat Berbasis Android," *Pros. 4th Semin. Nas. Penelit. Pengabdi. Kpd. Masy. 2020*, pp. 124–129, 2020.

[10] R. D. Putranto, "Analisa dan Perancangan Sistem Keamanan File Dengan Advanced Encryption Standard (AES) Berbasis Website," vol. 1, no. 6, 2023.

[11] F. Muharram, H. Azis, and A. R. Manga, "Algorithm Analysis of File Encryption and Decryption Process Using Advanced Encryption Standard (AES)," *Pros. Semin. Nas. Ilmu Komput. dan Teknol. Inf.*, vol. 3, no. 2, pp. 112–115, 2018.

[12] N. F. Aprilia, D. Mafa, A. R. Muchtar, K. A. A. Rohim, and R. N. I. Firdaus, "Penerapan Algoritma AES untuk Enkripsi pada Halaman Register serta Penerapan AES untuk Deskripsi pada Halaman Login Website," *J. Informatics Dev.*, vol. 1, no. 2, pp. 75–82, 2023, doi: 10.30741/jid.v1i2.1041.

[13] I. G. Indra, "Peningkatan Pengamanan Data File Menggunakan Algoritma Kriptografi AES Dari Serangan Brute Force," *J. Media Inform.*, vol. 4, no. 2, pp. 102–109, 2023, doi: 10.55338/jumin.v4i2.496.

[14] F. S. Anam and T. Fatimah, "Pengamanan File Berbasis Web Dengan Menerapkan Algoritma Advanced Encryption Standard (Aes-128) Web-Based File Security Using Advanced Encryption Standard (Aes-128) Algorithm on Cv Mitra Kurir Express," vol. 2, no. September, pp. 332–340, 2023.

[15] P. Suranta Surbakti, "Implementasi Algoritme AES-128 untuk Enkripsi dan Dekripsi File Dokumen Berbasis Web Pada Law Office Erdi Surbakti, S.H & Rekan," 2024.