

PENERAPAN *ADVANCED ENCRYPTION STANDARD* UNTUK SISTEM LOGIN DAN REGISTRASI PADA SISTEM INFORMASI PENJUALAN (STUDI KASUS:TOKO ILHAM BANJAR)

Imam Fauzy Muldani Rachmat*

Dosen Tetap, Universitas Insan Pembangunan

*Penulis Korespondensi: *imamfauzi43@unipi.ac.id*

ABSTRACT

Data security in sales information systems has become a critical priority in today's digital era. Ilham Banjar Store needs a system that can effectively protect user data, especially in login and registration features, which are vulnerable to security threats. The lack of encryption mechanisms in the current system results in user data, such as passwords, being transmitted in plaintext, increasing the risk of unauthorized access and data modification. Therefore, this study aims to enhance the security of the Ilham Banjar Store's sales information system by implementing the Advanced Encryption Standard (AES) 256-bit Cipher Block Chaining (CBC) method in the login and registration processes. The research method used is Research and Development (R&D), consisting of stages such as literature review, needs analysis, system design, implementation, testing, evaluation, and analysis. The implementation of the Advanced Encryption Standard (AES) involves the encryption processes of AddRoundKey, SubBytes, ShiftRows, and MixColumns, and the decryption processes are the inverse of these encryption steps. Testing includes measuring encryption and decryption times as well as verifying data integrity using HMAC (Hash-Based Message Authentication Code). The research results show that the performance testing indicates encryption and decryption times remain within acceptable limits, and the encrypted and decrypted data maintain their integrity. Thus, the new system meets both security and performance requirements.

Keywords: *Cryptography, AES, HMAC, Encryption, Decryption*

PENDAHULUAN

Keamanan data pada sistem informasi penjualan menjadi aspek yang penting untuk diperhatikan, terutama di era digital saat ini dimana ancaman terhadap data semakin meningkat. Toko Ilham Banjar membutuhkan sistem yang dapat melindungi data pengguna secara efektif. Fungsi *login* dan registrasi merupakan bagian penting dari sistem ini, karena berfungsi untuk mengidentifikasi dan mengotentikasi pengguna. Fitur *login* dan registrasi pada sistem informasi penjualan yang sedang berjalan pada Toko Ilham Banjar masih rentan terhadap berbagai ancaman keamanan karena tidak ada proses enkripsi dan dekripsi yang dirancang untuk melindungi data pengguna, sehingga data pengguna, seperti kata sandi, dikirimkan dalam bentuk *plaintext*.

Oleh karena itu, kurangnya mekanisme enkripsi, data yang dikirimkan berpotensi dapat diubah atau dicuri oleh penyerang yang melakukan komunikasi antara pengguna dan server. Selain itu, data yang dikirim dan disimpan tanpa enkripsi tidak dapat

memverifikasi keasliannya, sehingga rentan terhadap modifikasi yang tidak sah.

Penelitian sebelumnya yang relevan dilakukan oleh Laila Mustika dengan judul penelitian yaitu Implementasi Algoritma AES Untuk Pengamanan *Login* Dan Data Customer Pada E-Commerce Berbasis Web. Pada penelitian tersebut panjang kunci yang digunakan sebesar 128 bit, perancangan dibuat menggunakan bahasa pemrograman PHP dan database MySQL (Mustika 2020).

Penelitian yang dilakukan oleh Muhammad Riyan Andriyanto dan Pristi Sukmasetya berjudul Penerapan Algoritma Advanced Encryption Standard (AES) Untuk Keamanan Data Transaksi Pada Sistem E-Marketplace menghasilkan penelitian yang menerapkan algoritma dengan ukuran kunci 256-bit untuk keamanan data transaksi pengguna. Penelitian ini bertujuan untuk menciptakan sistem e-marketplace yang dapat mengelola proses transaksi jual beli bibit buah dan

tanaman di Desa Sriwedari. Hasil dari penelitian tersebut merupakan data transaksi pembayaran yang tersimpan di dalam *database* berupa data tersandi (*chipertext*) yaitu hasil dari proses enkripsi data asli yang telah di input oleh pengguna (*plaintext*) (Andriyanto and Sukmasetya 2022).

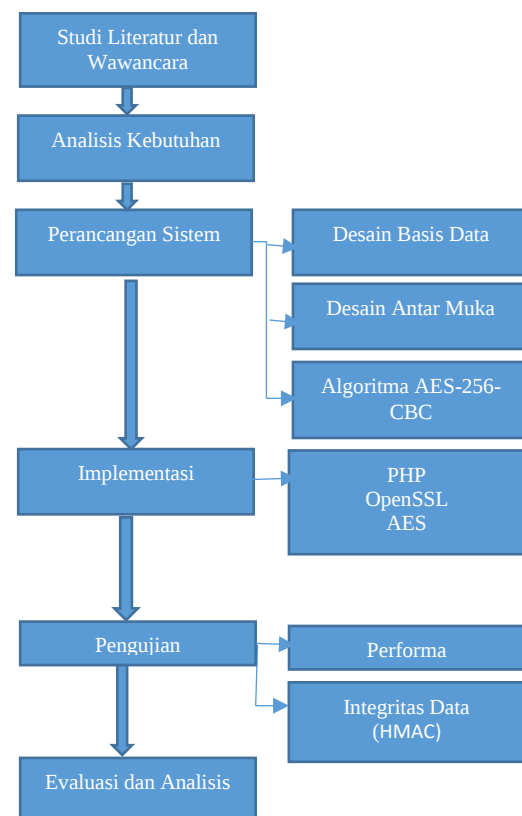
Penelitian yang dilakukan oleh Yendi Putra yang berjudul Meningkatkan Keamanan Web Menggunakan Algoritma *Advanced Encryption Standard (AES)* terhadap Serangan *Cross Site Scripting* menghasilkan penelitian yang berfokus pada penerapan algoritma AES dan pengujian kerentanan menggunakan Aplikasi Acunetix pada website *elearning* SMK maritim nusantara. Penelitian tersebut algoritma AES diterapkan pada *Secret Key Token* yang dibuat menggunakan JWT (*Json Web Token*) (Putra, Yuhandri, and Sumijan 2021)

Berdasarkan penelitian sebelumnya maka penelitian ini bertujuan untuk mengatasi masalah-masalah pada sistem berjalan dengan menerapkan metode *Advanced Encryption Standard* dalam proses *login* dan registrasi pada sistem informasi penjualan Toko Ilham Banjar. Selain itu, penelitian ini akan melibatkan pengembangan sistem menggunakan metode *Research and Development* yang mencakup implementasi enkripsi dan dekripsi data menggunakan AES proses enkripsi yang terdiri dari proses *Subbytes*, *ShiftRows*, *MixColumns*, dan *AddRoundKey* (Permana and Jaelani 2020). Ukuran kunci yang digunakan 256 bit serta menggunakan CBC (*Cipher Block Chaining*) yang bekerja efektif untuk meningkatkan keamanan dalam menyediakan kerahasiaan data yang tinggi dan otentikasi (Afsari et al. 2022). Selain itu penelitian ini membandingkan sistem berjalan dengan sistem baru setelah menerapkan AES-256-CBC. Sebelum diintegrasikan fitur *login* dan registrasi dibuat dalam bentuk *prototype* menggunakan kode PHP dengan fungsi *OpenSSL (Open Source Secure Sockets Layer)* yang merupakan toolkit kriptografi *open-source* pada fungsi PHP yang menyediakan implementasi dari protokol *Transport Layer Security (TLS)*. Selain itu, pengembangan fitur *login* dan registrasi diikuti dengan pengujian performa untuk memastikan bahwa sistem *login* dan registrasi yang baru tetap responsif dan dapat diandalkan. Setelah

menerapkan AES-256-CBC Pengujian performa tersebut berdasarkan panjang data dan iterasi *login*, sedangkan pengujian integritas data dengan menerapkan HMAC (*Hash-Based Message Authentication Code*)

METODOLOGI PENELITIAN

Penelitian ini dilakukan pada sistem informasi penjualan toko Ilham Banjar dengan metode *Research and Development (R&D)* yang digunakan untuk menghasilkan produk tertentu, dan menguji keefektifan produk tersebut (Sati et al. 2023). Metode penelitian tersebut mencakup beberapa tahapan penting dalam pengembangan dan pengujian sistem *login* dan registrasi menggunakan algoritma *Advanced Encryption Standard (AES)* dengan ukuran kunci 256 bit dalam mode *Cipher Block Chaining (CBC)*. Berikut adalah tahapan *Research and Development*.



Gambar 1. Tahapan Penelitian *Research and Development*

1. Studi Literatur

Tahap ini melibatkan pengumpulan dan kajian literatur yang relevan dengan topik penelitian. Studi literatur ini bertujuan untuk

memahami konsep dasar, algoritma enkripsi AES, dan mode operasinya, serta untuk meninjau penelitian-penelitian sebelumnya yang terkait dengan penerapan enkripsi pada sistem *login* dan registrasi.

2. Analisis Kebutuhan

Tahap ini mencakup analisis terhadap sistem informasi penjualan yang sedang berjalan di Toko Ilham Banjar. Analisis ini bertujuan untuk mengidentifikasi kelemahan-kelemahan sistem yang ada, terutama dalam hal keamanan data pengguna.

3. Perancangan Sistem

Pada tahap ini, perancangan sistem *login* dan registrasi yang menggunakan enkripsi AES-256-CBC dilakukan. Perancangan ini mencakup perancangan basis data yaitu merancang tabel users untuk menyimpan data pengguna yang terenkripsi. Merancang halaman *login* dan registrasi yang *user-friendly* dan aman. Kemudian perancangan Proses Enkripsi dan Dekripsi yaitu menentukan alur proses enkripsi dan dekripsi yang akan diimplementasikan dalam kode PHP.

4. Implementasi

Tahap implementasi melibatkan pengkodean sistem berdasarkan desain yang telah dibuat. Implementasi dilakukan menggunakan bahasa pemrograman PHP dan *database MySQL*. Kode untuk enkripsi dan dekripsi dengan AES-256-CBC diimplementasikan pada fungsi *login* dan registrasi.

5. Pengujian

Tahap pengujian dilakukan untuk memastikan bahwa sistem yang dikembangkan memenuhi kebutuhan keamanan dan performa yang diharapkan. Pengujian tersebut terdiri dari pengujian integritas data dan pengujian performa. Pengujian integritas data bertujuan untuk memastikan data yang didekripsi sama dengan data asli yang dienkripsi, menggunakan HMAC (*Hash-Based Message Authentication Code*) untuk verifikasi. Oleh karena itu, HMAC (*Hash-Based Message Authentication Code*) dapat digunakan sebagai metode untuk memverifikasi integritas data yang dikirim antar sistem. HMAC menggunakan kunci kriptografi dan fungsi hash untuk membangun kode autentikasi pesan, yang kemudian ditambahkan ke akhir pesan yang ditujukan untuk penerima. (Dalimunthe, Reza, and Marzuki 2022). Pengujian performa bertujuan untuk mengukur waktu yang dibutuhkan untuk

proses enkripsi dan dekripsi pada saat registrasi dan *login*, dinyatakan dalam milidetik.

6. Evaluasi dan Analisis

Hasil pengujian dievaluasi untuk memastikan bahwa sistem yang baru memenuhi kebutuhan keamanan dan performa. Analisis dilakukan untuk membandingkan sistem sebelum dan sistem baru setelah menerapkan enkripsi AES-256-CBC, untuk melihat peningkatan keamanan dan kinerja yang dicapai.

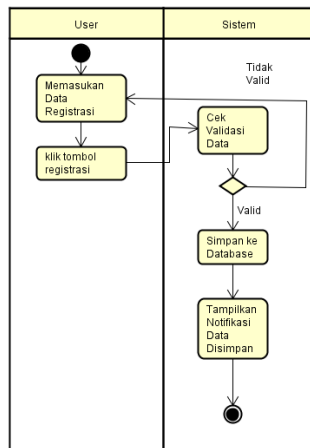
HASIL DAN PEMBAHASAN

A. Analisis Kebutuhan

Analisis Kebutuhan Sistem yang berjalan di Toko Ilham Banjar bertujuan untuk mengidentifikasi kelemahan-kelemahan sistem yang ada, terutama dalam hal keamanan data pengguna. Pada tahapan ini untuk mengetahui proses yang sedang berjalan maka dibuatkan model menggunakan *activity diagram*. *Activity diagram* tersebut menunjukkan aktivitas sistem dan aktor yang berinteraksi mulai dari aktivitas dimulai, keputusan apa saja, dan hingga aktivitas selesai (Rachmat 2022). Berikut adalah poin-poin utama dari analisis kebutuhan:

1. Proses Registrasi:

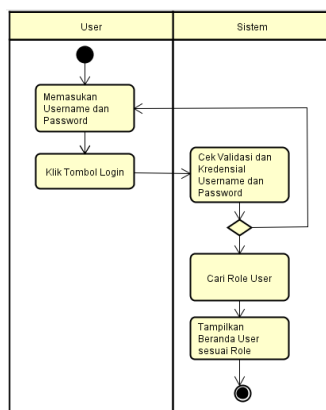
Pengguna baru memasukkan data registrasi berupa informasi pribadi (nama, alamat email, nomor telepon) dan membuat kata sandi. Data yang dimasukkan divalidasi apakah data yang dimasukan valid atau tidak, jika valid maka dikirim ke server dalam bentuk *plaintext*. Pada aktivitas validasi seperti *field* yang wajib diisi, ketentuan jumlah panjang *password* minimal 6 karakter, dan format *email*. Kemudian server menyimpan data pengguna pada *database* tanpa enkripsi sedangkan apabila tidak *valid* maka *user* dapat menginput data registrasi kembali.



Gambar 2. Registrasi Sistem berjalan

2. Proses Login:

Pengguna memasukkan alamat email dan kata sandi. Kemudian data *login* dikirim ke server dalam bentuk *plaintext*. Server memvalidasi dan memverifikasi kredensial dengan mencocokkan data yang disimpan di *database*.



Gambar 3. Login Sistem berjalan

Kelemahan Sistem Berjalan:

Pengiriman data *login* seperti *password* masih dalam bentuk *plaintext*. Data pribadi dan kata sandi pengguna dikirim melalui jaringan tanpa enkripsi, sehingga rentan terhadap ancaman *man-in-the-middle* (MitM) *attack*. Selain itu, data yang dikirim dan disimpan tidak dapat diverifikasi keasliannya, sehingga rentan terhadap modifikasi oleh pihak yang tidak sah.

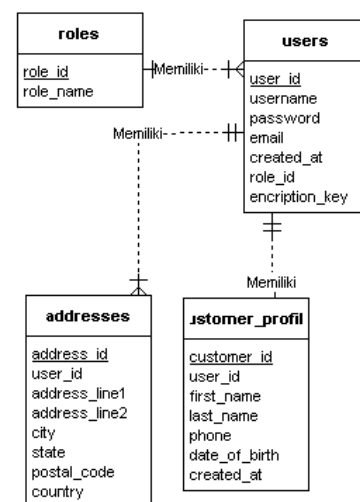
B. Perancangan Sistem

Pada tahap ini, dilakukan perancangan sistem *login* dan registrasi menggunakan

enkripsi AES-256-CBC. Perancangan ini mencakup beberapa aspek penting:

1. Desain Basis Data:

Pada perancangan basis data, diagram yang digunakan pada tahapan ini adalah menggunakan *class diagram* yang digunakan untuk menggambarkan struktur dan hubungan antara objek-objek yang terlibat dalam sistem (Rachmat 2023). Berikut adalah *class diagram* untuk *form login* dan registrasi yang akan diusulkan dengan menerapkan AES-256-CBC.

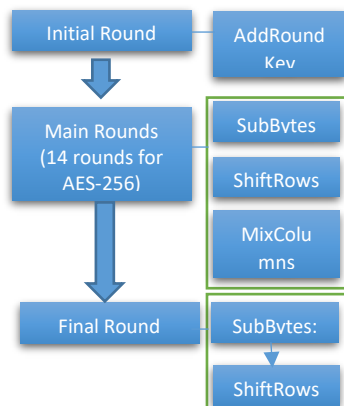


Gambar 4. Class Diagram Form Login dan Registrasi yang diusulkan

Pada perancangan class diagram untuk sistem baru, terdapat penambahan atribut *encryption_key* pada tabel *users*. Tujuan dari penambahan ini adalah untuk memastikan setiap pengguna memiliki kunci enkripsi yang unik guna melindungi data sensitif mereka. Proses enkripsi dan dekripsi menggunakan kunci ini dengan algoritma AES-256-CBC, yang memberikan tingkat keamanan tinggi untuk sistem login dan registrasi. Proses enkripsi dan dekripsi menggunakan kunci tersebut dengan algoritma AES-256-CBC, yang menyediakan tingkat keamanan yang tinggi untuk sistem *login* dan registrasi.

2. Desain Proses Enkripsi dan Dekripsi:

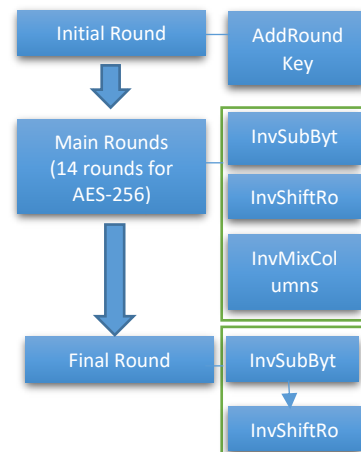
Tahap ini membuat desain proses enkripsi dan dekripsi yang akan diimplementasikan dalam kode PHP pada sistem login dan registrasi di Toko Ilham Banjar. Langkah-langkah utama meliputi pembuatan kunci enkripsi unik untuk setiap pengguna, enkripsi kata sandi sebelum disimpan dalam basis data, dan dekripsi kata sandi saat proses login.



Gambar 5.Desain Proses Enkripsi

Berdasarkan tahapan proses enkripsi, sebelum melakukan enkripsi, langkah pertama adalah penggunaan *initial* vektor untuk *generate key*. *Initial* vektor digunakan sebagai input beserta kunci enkripsi, untuk menghasilkan kunci yang akan digunakan dalam proses enkripsi. Setelah itu, proses dilanjutkan dengan tahapan *Initial Round* di mana menggunakan *AddRoundKey* dalam fungsi *openssl_encrypt*. Pada tahap ini, setiap *byte* dalam *plaintext* di-XOR dengan kunci enkripsi yang dihasilkan dari *initial* vektor. Kemudian, dilanjutkan dengan tahapan *Main Rounds*, dimana fungsi *openssl_encrypt* secara otomatis menangani urutan langkah-langkah enkripsi (*SubBytes*, *ShiftRows*, *MixColumns*, dan *AddRoundKey*) sesuai dengan standar AES-256-CBC. *SubBytes* mengubah *byte* dalam blok menggunakan S-box, *ShiftRows* melakukan pergeseran baris dalam blok ke kiri sesuai aturan AES, dan *MixColumns* memodifikasi kolom-kolom dalam blok menggunakan transformasi linier. Pada tahap *AddRoundKey*, blok *plaintext* di-XOR dengan bagian dari kunci enkripsi untuk setiap ronde. Terakhir, pada tahap *Final*

Round, proses *MixColumns* tidak dilakukan. Fungsi *openssl_encrypt* menangani hal tersebut dengan mengimplementasikan *SubBytes*, *ShiftRows*, dan *AddRoundKey*. Pada *Final Round* dimana *MixColumns* tidak dilakukan dengan tujuan untuk menjaga efisiensi tanpa mengorbankan keamanan.



Gambar 6. Desain Proses Dekripsi

Sebelum melakukan tindakan dekripsi, langkah awalnya adalah menggunakan *initial* vektor yang digunakan sebagai input dengan kunci enkripsi untuk menghasilkan kunci yang dibutuhkan dalam proses dekripsi. Langkah berikutnya adalah tahap *Initial Round*, di mana *AddRoundKey* digunakan dalam fungsi *openssl_decrypt*. Pada tahap ini, setiap *byte* dalam ciphertext di-XOR dengan kunci enkripsi yang berasal dari *initial* vektor. Selanjutnya, proses dilanjutkan dengan tahapan *Main Rounds*, di mana urutan langkah-langkah dekripsi (*InvShiftRows*, *InvSubBytes*, dan *AddRoundKey*) ditangani secara otomatis oleh fungsi *openssl_decrypt* sesuai dengan standar AES-256-CBC. *InvShiftRows* melakukan pergeseran baris dalam blok ke arah kanan sesuai dengan aturan AES, *InvSubBytes* mengubah kembali *byte* dalam blok menggunakan tabel substitusi inversi (Inverse S-box), dan *AddRoundKey* melakukan operasi XOR dengan kunci enkripsi yang berbeda untuk setiap ronde. Pada tahap *Final Round*, hanya *InvShiftRows*, *InvSubBytes*, dan

AddRoundKey yang dilakukan, tanpa melibatkan langkah *MixColumns*.

C. Implementasi

Tahap implementasi melibatkan pengkodean sistem berdasarkan desain yang telah dibuat. Implementasi dilakukan menggunakan bahasa pemrograman PHP dan database *MySQL*. Kode untuk enkripsi dan dekripsi dengan *AES-256-CBC* diimplementasikan pada fungsi *login* dan registrasi. Berikut adalah cuplikan penerapan AES pada PHP.

Tabel 1. Fungsi Enkripsi dan Dekripsi

Nama Fungsi	Cuplikan Kode
Enkripsi	<pre>function encryptData(\$data, \$key) { \$iv = openssl_random_pseudo _bytes(openssl_cipher_iv _length('aes-256-cbc')); \$encrypted = openssl_encrypt(\$data, 'aes-256-cbc', \$key, 0, \$iv); return base64_encode(\$encrypt ed . '::' . \$iv); } // Saat registrasi, menghasilkan kunci enkripsi dan menyimpan data terenkripsi \$username = \$_POST['username']; \$password = \$_POST['password']; \$email = \$_POST['email']; \$encryption_key = bin2hex(openssl_random _pseudo_bytes(32)); \$encrypted_password = encryptData(\$password, \$encryption_key);</pre>

Dekripsi	<pre>function decryptData(\$data, \$key) { list(\$encrypted_data, \$iv) = explode(':', base64_decode(\$data), 2); return openssl_decrypt(\$encryp ted_data, 'aes-256-cbc', \$key, 0, \$iv); } //pada saat login if (\$result->num_rows > 0) { \$user = \$result- >fetch_assoc(); \$encrypted_password = \$user['password']; \$encryption_key = \$user['encryption_key']; \$decrypted_password = decryptData(\$encrypted_ password, \$encryption_key); if (\$password === \$decrypted_password) { echo "Login successful!"; //masukke baranda user } else { echo "Incorrect password!"; } } else { echo "User not found!"; } }</pre>
----------	---

Fungsi *encryptData* dirancang untuk mengenkripsi data menggunakan algoritma *AES-256-CBC*. Pertama, fungsi ini menghasilkan sebuah inisialisasi vektor (IV) yang dihasilkan secara acak menggunakan *openssl_random_pseudo_bytes*, yang ukurannya sesuai dengan panjang IV untuk *AES-256-CBC*. Kemudian, data yang akan dienkripsi dienkripsi menggunakan *openssl_encrypt* dengan algoritma *AES-*

256-CBC, kunci enkripsi yang diberikan, dan IV yang baru dibuat. Data terenkripsi dan IV tersebut digabungkan dan dikodekan menggunakan *base64_encode* untuk memastikan bahwa data yang dihasilkan aman untuk disimpan atau ditransmisikan dalam bentuk string.

Saat proses registrasi pengguna, sistem menerima input dari form registrasi, yaitu username, password, dan email. Kunci enkripsi yang unik untuk setiap pengguna dihasilkan menggunakan *bin2hex* dan *openssl_random_pseudo_bytes* untuk membuat string acak sepanjang 32 byte (256 bit), yang merupakan panjang kunci untuk AES-256. Password yang dimasukkan oleh pengguna dienkripsi dengan menggunakan fungsi *encryptData*, dengan kunci enkripsi yang baru dihasilkan. Hasil dari proses enkripsi (*password* terenkripsi) kemudian disimpan dalam *database* bersama dengan username, email, kunci enkripsi, dan *role_id* melalui *query SQL*. Kunci enkripsi ini juga disimpan dalam *database* agar dapat digunakan kembali saat proses dekripsi data.

Fungsi *decryptData* bertujuan untuk mendekripsi data yang telah dienkripsi menggunakan algoritma AES-256-CBC. Fungsi ini pertama-tama mengambil data yang dikodekan dalam *base64* dan memisahnya menjadi dua bagian: data terenkripsi dan inisialisasi vektor (IV). Setelah dipisahkan, *openssl_decrypt* digunakan untuk mendekripsi data dengan algoritma AES-256-CBC menggunakan kunci enkripsi dan IV yang sama seperti pada saat proses enkripsi. Selama proses *login*, sistem menerima input dari form *login* dan memeriksa keberadaan pengguna dalam *database* berdasarkan *username* yang diberikan. Jika pengguna ditemukan, data terenkripsi (*password* terenkripsi) dan kunci enkripsi diambil dari *database*. Data *password* yang terenkripsi kemudian didekripsi menggunakan fungsi *decryptData* dengan kunci enkripsi yang sesuai. Setelah mendapatkan *password* yang didekripsi, sistem membandingkannya dengan password yang dimasukkan oleh pengguna. Jika password yang didekripsi cocok dengan password yang diberikan, *login* berhasil dan pengguna diarahkan ke halaman beranda. Jika tidak, pesan kesalahan "*Incorrect password!*" akan ditampilkan. Jika pengguna tidak ditemukan

dalam *database*, pesan "*User not found!*" akan ditampilkan.

Berikut adalah tampilan registrasi dapat disajikan pada gambar 7 dan untuk tampilan *login* pada gambar 8..

Gambar 7. Form Registrasi

Gambar 8. Form Login

D. Pengujian

Tahap pengujian dilakukan untuk memastikan bahwa sistem yang dikembangkan memenuhi kebutuhan keamanan dan performa yang diharapkan. Pengujian meliputi:

1. Pengujian performa bertujuan untuk mengukur waktu yang dibutuhkan untuk proses enkripsi dan dekripsi pada saat registrasi dan *login*. Pengukuran dilakukan dalam milidetik untuk memastikan sistem responsif dan efisien. Pada pengujian performa bagian yang diukur berdasarkan panjang data. Oleh karena itu, mengukur performa sistem dengan variasi panjang data membantu untuk mengetahui bagaimana sistem merespon terhadap berbagai ukuran data yang berbeda. Kemudian iterasi *login*, dengan mengukur performa sistem

dengan variasi jumlah iterasi *login*, maka dapat memahami bagaimana kinerja sistem berubah seiring dengan meningkatnya jumlah percobaan *login* yang gagal. Berikut adalah tabel pengujian performa dengan panjang kunci 256 bit

Tabel 2. Pengujian Performa

Panjang Data	Iterasi Login	Waktu Enkripsi (ms)	Waktu Dekripsi (ms)
8	1	0,026	0,008
16	1	0,005	0,005
32	1	0,004	0,004
64	1	0,004	0,006
128	1	0,005	0,007
8	2	0,008	0,007
16	2	0,008	0,009
32	2	0,007	0,01
64	2	0,008	0,01
128	2	0,008	0,01
8	3	0,011	0,009
16	3	0,011	0,011
32	3	0,011	0,012
64	3	0,011	0,012
128	3	0,011	0,009
8	4	0,014	0,011
16	4	0,014	0,012
32	4	0,014	0,012
64	4	0,016	0,012
128	4	0,015	0,012

Berdasarkan hasil pengujian performa, dapat disimpulkan waktu enkripsi dan dekripsi cenderung meningkat seiring dengan bertambahnya panjang data. Terdapat fluktuasi dalam waktu enkripsi dan dekripsi pada setiap iterasi *login*, tetapi fluktuasi tersebut relatif kecil. Peningkatan iterasi *login* cenderung meningkatkan waktu enkripsi dan dekripsi secara proporsional. Walaupun terjadi fluktuasi, waktu enkripsi dan dekripsi pada umumnya tetap dalam rentang yang wajar dan dapat diterima. Dengan demikian, berdasarkan hasil pengujian tersebut, AES tetap menunjukkan performa yang baik dan dapat diandalkan untuk penggunaan dalam enkripsi dan dekripsi data.

2. Pengujian Integritas Data yaitu untuk memastikan bahwa data yang didekripsi sama dengan data asli yang dienkripsi. Salah satu metode yang untuk verifikasi integritas data yaitu HMAC (*Hash-Based Message Authentication Code*). HMAC yang merupakan kunci kriptografi dan fungsi hash untuk membangun kode autentikasi. Pada pengujian integritas data data dienkripsi menggunakan AES-256-CBC dan kemudian HMAC dihasilkan dari data terenkripsi. Data bersama dengan HMAC dikirimkan ke penerima. Penerima kemudian menerima data dan melakukan verifikasi integritas dengan membandingkan HMAC yang diterima dengan HMAC yang dihasilkan dari data terenkripsi. Jika kedua HMAC cocok, integritas data dianggap terjamin. Berikut adalah cuplikan kode PHP penambahan HMAC.

Tabel 3. Cuplikan Kode PHP Pengujian Integritas Data

Kode PHP	Ket
function generateHMAC(\$data, \$key) { return hash_hmac('sha256', \$data, \$key); }	Fungsi untuk menghasilkan HMAC
\$expectedHMAC = generateHMAC(\$receivedEncryptedData, \$encryptionKey); if (\$receivedHMAC === \$expectedHMAC) { return "Integritas data terjamin. Data diterima dengan aman."; } else { return "Integritas data tidak terjamin. Data mungkin dimanipulasi atau rusak selama transmisi."; }	Verifikasi Integritas Data

Setelah *script* dijalankan maka server akan bertindak sebagai penerima yang memverifikasi integritas data yang dikirimkan oleh klien. Dengan demikian,

server memisahkan *HMAC* dari data terenkripsi, menghasilkan *HMAC* baru dari data terenkripsi menggunakan kunci yang sama, dan membandingkan dua *HMAC* tersebut untuk memastikan integritas data. Berdasarkan pengujian tersebut output yang dihasilkan berupa pesan “Integritas data terjamin. Data diterima dengan aman”. Hasil tersebut berarti menunjukkan integritas data terjaga selama transmisi antara klien dan server. Hal ini menunjukkan bahwa data yang dienkripsi oleh klien dan dikirim ke server tetap utuh tanpa mengalami perubahan atau kerusakan.

E. Evaluasi dan Analisis

Hasil pengujian menunjukkan bahwa sistem yang baru memenuhi kebutuhan keamanan dan performa. Dengan penerapan enkripsi AES-256-CBC, data yang dienkripsi dan didekripsi tetap terjaga integritasnya, dibuktikan melalui penggunaan *HMAC*. Pengujian performa menunjukkan bahwa waktu enkripsi dan dekripsi tetap dalam batas yang dapat diterima meskipun ada variasi panjang data dan iterasi *login*.

Analisis perbandingan antara sistem sebelum dan setelah penerapan enkripsi AES-256-CBC menunjukkan peningkatan signifikan dalam aspek keamanan. Data yang dikirim dan diterima menjadi lebih aman dari potensi penyusupan dan modifikasi. Waktu proses yang sedikit bertambah masih dalam batas toleransi dan tidak mempengaruhi pengalaman pengguna secara signifikan.

KESIMPULAN

Penerapan enkripsi AES-256-CBC pada sistem registrasi dan *login* di Toko Ilham Banjar berhasil meningkatkan keamanan data pengguna dengan mengatasi kelemahan pengiriman dan penyimpanan data dalam bentuk *plaintext*. Pengujian menunjukkan peningkatan waktu enkripsi dan dekripsi yang tetap dalam batas yang dapat diterima, sementara penggunaan *HMAC* memastikan integritas data selama transmisi. Sistem baru setelah menerapkan AES-256-CBC efektif melindungi data sensitif pengguna tanpa mengorbankan performa yang signifikan. Penelitian berikutnya dapat membandingkan AES-256-CBC dengan metode enkripsi lain seperti *RSA*, *ChaCha20*, *Twofish*, dan *Blowfish* untuk menentukan algoritma terbaik

berdasarkan efisiensi dan keamanan. Selain itu, pengujian kinerja pada berbagai lingkungan komputasi dapat memberikan wawasan tambahan tentang dampaknya terhadap performa dan keamanan sistem enkripsi.

DAFTAR PUSTAKA

- Afsari, Melani, Dadang Iskandar Mulyana, Alfiani Damaiyanti, and Naini Sa'adah. 2022. “Implementasi Mode Operasi Kombinasi Cipher Block Chaining Dan Metode LSB-1 Pada Pengamanan Data Text.” *Jurnal Pendidikan Sains dan Komputer* 2(01): 70–82.
- Andriyanto, Muhammad Riyan, and Pristi Sukmasetya. 2022. “Penerapan Algoritma Advanced Encryption Standard (AES) Untuk Keamanan Data Transaksi Pada Sistem E-Marketplace.” *Journal of Computer System and Informatics (JoSYC)* 4(1): 179–87.
- Dalimunthe, Syabdan, Joeahrsyah Reza, and Asep Marzuki. 2022. “The Model for Storing Tokens in Local Storage (Cookies) Using Json Web Token (Jwt) With Hmac (Hash-Based Message Authentication Code) in E-Learning Systems.” *Journal of Applied Engineering and Technological Science* 3(2): 149–55.
- Mustika, Laila. 2020. “Implementasi Algoritma AES Untuk Pengamanan Login Dan Data Customer Pada E-Commerce Berbasis Web.” *JURIKOM (Jurnal Riset Komputer)* 7(1): 148.
- Permana, Aji, and Elan Jaelani. 2020. “Implementasi Algoritma AES 128 Bit Sebagai Pengaman Teks Di Aplikasi Note Berbasis Android.” *JEJARING: Jurnal Teknologi dan Manajemen Informatika* 5(2): 9–17.
- Putra, Yendi, Y Yuhandri, and S Sumijan. 2021. “Meningkatkan Keamanan Web Menggunakan Algoritma Advanced Encryption Standard (AES) Terhadap Serangan Cross Site Scripting.” *Jurnal*

- Sistim Informasi dan Teknologi* 3: 56–63.
- Rachmat, Imam Fauzy Muldani. 2022. “Perancangan Mobile Commerce Berbasis Android Menggunakan Metode Rational Unified Process (Studi Kasus:TOKO ILHAM BANJAR).” *IPSIKOM* 10(2): 34–43.
- . 2023. “IMPLEMENTASI METODE RAD PADA PERANCANGAN SISTEM INFORMASI PELAYANANJASA CUCI MOTOR DAN MOBIL BERBASIS WEB(STUDI KASUS:PRIMA WASH).” *IPSIKOM* 11(2).
- Sati, Ardiansyah Tria, Dimas Tri Aditya, Nabila Latifah Azzahra, and Roeslan Djutalov. 2023. “Perancangan Sistem Informasi Keuangan Peninggaran Raya (OPERA) Berbasis Dekstop Dengan Java SE & Mysql Menggunakan Metode Research and Development (RND).” *JORAPI: Journal of Research and Publication Innovation* 1(2): 196–200. <https://jurnal.portalpublikasi.id/index.php/JORAPI/index>.