Saturnus: Jurnal Teknologi dan Sistem Informasi
Volume. 4, Nomor. 1, Januari 2026
E-ISSN : 3031-9943; P-ISSN : 3031-9935; Hal. 129-147
DOI: https://doi.org/10.61132/saturnus.v4i1.1422
Tersedia : https://journal.arteii.or.id/index.php/Saturnus

Design Of A Transaction Data Security System In A Bouqet Sales Application Using An Aes Algorithm

Ibnu Rusydi[1*], Laila Ali Putri[2], Maria Ulfa[3]
[1-3] Universitas Islam Negeri Sumatera Utara, Indonesia
Email: ibnurusydi@uinsu.ac.id[1], lailaputii2003@gmail.com[2], mariaaulfaa2805@gmail.com[3]

Corresponding Author: ibnurusydi@uinsu.ac.id

Abstrack: *This research presents the development of a transaction data protection mechanism for a bouquet sales application by utilizing the Advanced Encryption Standard (AES) algorithm. The rapid growth of digital commerce has led to an increase in online transactions, which in turn raises serious concerns regarding the security of sensitive transaction data. Information such as customer identities, order details, delivery addresses, and payment data are vulnerable to unauthorized access, data leakage, and manipulation if not properly secured. To address these issues, this study applies the AES-128 encryption algorithm using a 128-bit secret key to secure transaction data before it is stored in the system database. The encryption process follows the standard AES workflow, including key expansion, initial transformation, multiple encryption rounds, and a final transformation stage. Decryption is restricted exclusively to authorized users who possess the correct encryption key. The research methodology includes system analysis, AES integration into the application, and functional testing of the encryption and decryption processes. Data integrity is validated by comparing the original plaintext with the decrypted output, while system performance is evaluated based on processing time and decryption accuracy. Experimental results indicate that the average encryption and decryption time remains under 10 milliseconds per transaction, without affecting system performance. The findings confirm that AES-128 effectively enhances transaction data confidentiality and integrity in the bouquet sales application.*

Keywords: *AES Algorithm; Bouquet Sales Application; Cryptography; Data Security; Transaction Data.*

1. INTRODUCTION

The advancement of information technology has significantly changed transaction mechanisms in modern business environments, particularly within digital commerce platforms. Web-based and mobile-based sales applications are now widely adopted by various business sectors, including bouquet sales services, due to their ability to improve operational efficiency and expand customer reach. However, this digital transformation also introduces new security challenges related to the protection of transaction data.

Transaction records in bouquet sales applications contain sensitive information such as customer identities, order specifications, payment values, and transaction methods. Without adequate security mechanisms, these data are exposed to threats such as data breaches, unauthorized access, and transaction manipulation, which may lead to financial losses and decreased user trust (Triayudi, 2024).

Cryptographic techniques offer an effective solution for securing sensitive information in digital systems. One of the most reliable symmetric encryption algorithms is the Advanced Encryption Standard (AES), which has been widely recognized for its security strength, efficiency, and global standardization. AES works by transforming readable data into

Naskah Masuk: 15 Desember 2025; Revisi: 16 Desember 2025; Diterima: 21 Januari 2026;
Terbit: 30 Januari 2026

encrypted form, ensuring that information remains confidential unless decrypted using the correct secret key.

Previous studies have demonstrated the effectiveness of AES in securing various types of data, including files, databases, and transaction records in different application environments. These studies confirm that AES is capable of maintaining data confidentiality while preserving system performance. Based on these considerations, this research focuses on implementing AES-128 encryption to secure transaction data in a bouquet sales application, aiming to enhance data security and strengthen user confidence in the system.

Several previous studies have explored the use of the Advanced Encryption Standard (AES) algorithm to secure data in different information systems (Indraka, 2024). For example, implemented to encrypt and decrypt files in a web-based environment, demonstrating stable performance across file sizes up to 5 MB with reliable confidentiality protection (Nasrullah, 2025). In the context of database security, applied AES-128 to encrypt sensitive patient registration data, finding that AES effectively prevents unauthorized access while maintaining data integrity(Ahmad, 2024) Like wise, combined AES with SHA-3 hashing to enhance user data protection for web transaction systems, illustrating AES's adaptability in securing session data and credentials (Tamin, 2025). In a mobile setting used AES-256 to protect messages in Android messaging applications, achieving fast encryption times with minimal impact on device performance (Astriyani, 2024).

Therefore, this study aims to design and implement a transaction data security mechanism for a bouquet sales application by applying the Advanced Encryption Standard (AES) algorithm. The proposed approach focuses on securing sensitive transaction information, including customer and payment data, to ensure data confidentiality and integrity. By integrating AES into the transaction process, this research seeks to improve system security and strengthen user confidence in the application (Nugroho, 2025).

2. LITERATURE REVIEW

A literature review is a fundamental component of a scientific article, as it provides a structured overview of theories, methods, and previous studies relevant to the research topic. Through a critical examination of existing literature, this section establishes the research context, explains the theoretical foundations of the study, and identifies research gaps that justify the current work. In transaction-based applications, literature review plays an important role in understanding data security mechanisms, cryptographic methods, and their effectiveness in protecting sensitive transactional information. Therefore, this section discusses encryption

and decryption methods with a focus on symmetric and asymmetric cryptography, followed by a detailed review of the Advanced Encryption Standard (AES) and its application in data security (Ramadhani, 2024).

Types of Encryption and Decryption Methods

Data security in information systems is commonly achieved through cryptographic techniques that transform plaintext into unreadable ciphertext to prevent unauthorized access. Encryption methods are generally classified into symmetric encryption and asymmetric encryption, based on the type of keys used in the encryption and decryption processes (Syafrullah, 2024).

Symmetric Encryption

Symmetric encryption uses a single shared secret key for both encryption and decryption. This method is known for its high processing speed and efficiency, making it suitable for securing large volumes of data such as transaction records and database contents. Common examples of symmetric encryption algorithms include Data Encryption Standard (DES), Triple DES (3DES), and Advanced Encryption Standard (AES). In symmetric cryptography, the main challenge lies in secure key distribution, as both the sender and receiver must possess the same secret key. Despite this limitation, symmetric encryption is widely implemented in transaction-based systems due to its low computational overhead and strong security when properly managed (Oktaviani, 2023).

Asymmetric Encryption

Asymmetric encryption employs a pair of keys, namely a public key and a private key. The public key is used for encryption, while the private key is used for decryption. Algorithms such as RSA, Elliptic Curve Cryptography (ECC), and Diffie–Hellman are commonly used in this category. Asymmetric encryption is particularly effective for secure key exchange, digital signatures, and authentication processes. However, due to its higher computational complexity, asymmetric encryption is generally not used for encrypting large transaction data directly. In practice, asymmetric encryption is often combined with symmetric encryption to create a hybrid security model, where asymmetric methods secure key exchange and symmetric methods protect the actual data (Nasution, 2024)

Implementation of AES (Advanced Encryption Standard) in Data Security

Definition of AES

The Advanced Encryption Standard (AES) is a symmetric block cipher standardized by the National Institute of Standards and Technology (NIST) to replace DES due to its stronger security and better performance. AES operates on fixed-size data blocks of 128 bits and is

widely adopted in various applications, including financial systems, web applications, mobile platforms, and cloud environments.

Types and Variations of AES

AES supports multiple key length variations, which determine the security level and number of encryption rounds. The most commonly used AES variants are:

1. AES-128, which uses a 128-bit key and performs 10 encryption rounds,
2. AES-192, which uses a 192-bit key with 12 rounds,
3. AES-256, which uses a 256-bit key with 14 rounds.

Longer key lengths provide higher resistance against brute-force attacks but require greater computational resources. The selection of AES variants depends on the security requirements and performance constraints of the system. In transaction-based applications, AES-128 is often preferred due to its balance between strong security and efficient processing time.

AES Encryption and Decryption Process

The AES algorithm encrypts data through a series of transformation rounds applied to a 128-bit data block. The main processes involved in AES encryption include SubBytes, ShiftRows, MixColumns, and AddRoundKey. These transformations ensure data confusion and diffusion, making it difficult for attackers to reconstruct the original plaintext without the correct key. The decryption process follows the inverse operations of encryption using the same secret key.

Conceptually, the AES working mechanism can be illustrated through a basic block diagram consisting of plaintext input, key expansion, round-based encryption transformations, and ciphertext output. This structured process enables AES to provide strong confidentiality while maintaining efficient performance, making it suitable for securing transaction data in sales applications.(Rabtsani, 2025).

3. METHODS *(10 Pt)*

This study employs a system development and experimental research approach to design and implement a transaction data security system for a bouquet sales application using the Advanced Encryption Standard (AES) algorithm. In addition to cryptographic implementation, this research adopts a structured software development method supported by Unified Modeling Language (UML) for system analysis and design, as well as black-box testing for application validation.

Research Approach and System Development Method

The application development follows a waterfall-based system development method, consisting of requirement analysis, system design, implementation, testing, and evaluation. This method is selected because the system requirements are clearly defined at the initial stage, particularly regarding transaction data flow and security requirements (Assidiq, 2024).

During the system design phase, UML is used as a modeling tool to describe system structure and behavior. The UML analysis applied in this study includes:

- Use Case Diagram, to identify system actors and define interactions between users and the bouquet sales application, particularly related to transaction submission, data encryption, and authorized data access.

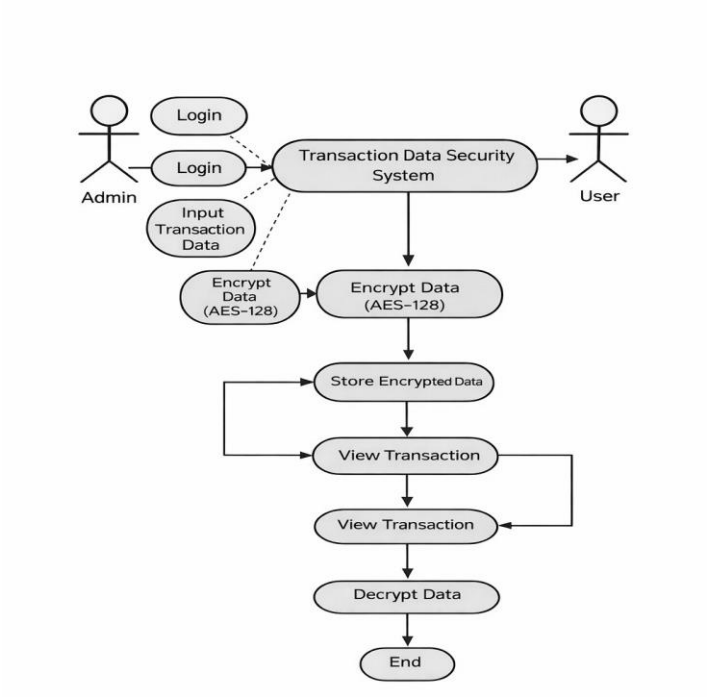


Figure 1. Use case diagram.

- Activity Diagram, to illustrate the workflow of transaction processing, including data input, AES encryption, data storage, data retrieval, and AES decryption.

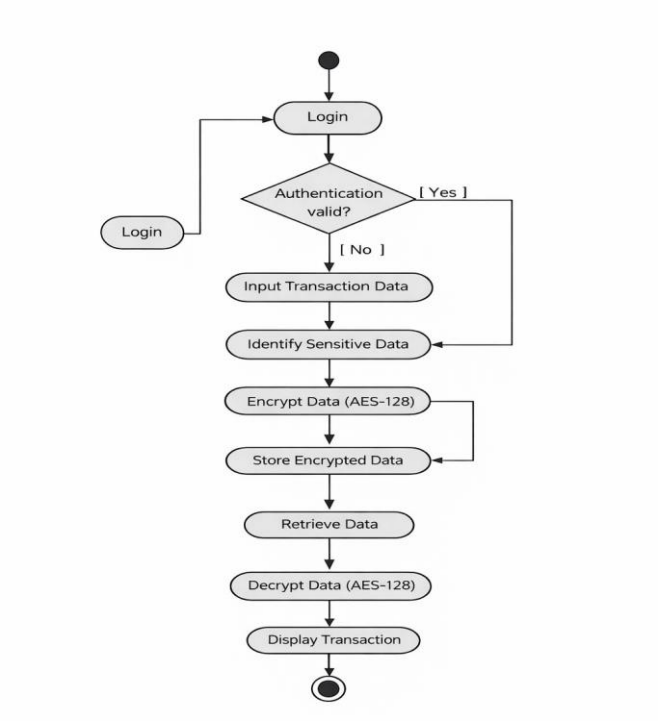


Figure 2. Activity for diagram Aes

- Sequence Diagram, to describe the sequential interaction between system components (user interface, encryption module, database, and decryption module) during transaction processing.

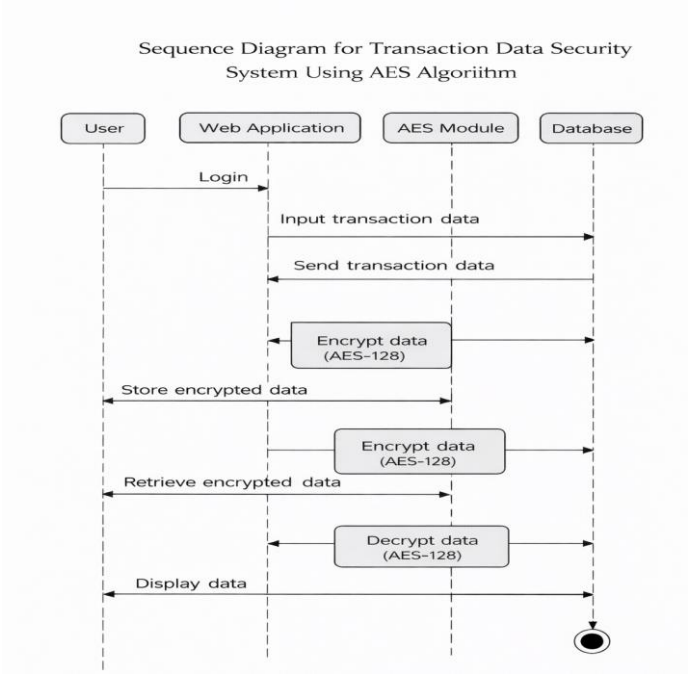


Figure 3. Sequence diagram.

These UML diagrams ensure that the integration of AES into the transaction process is clearly defined and systematically implemented.

AES Encryption and Decryption Design

The AES algorithm is implemented using AES-128 with a 128-bit secret key. The encryption process follows the standard AES procedure, including key expansion, initial AddRoundKey, nine main rounds (SubBytes, ShiftRows, MixColumns, and AddRoundKey), and a final round without MixColumns. Decryption is performed using the inverse AES transformations with the same secret key.

Sensitive transaction fields customer identity, order details, delivery address, and payment amount are encrypted at the field level before being stored in the relational database. Non-sensitive fields, such as transaction ID and transaction timestamp, are not encrypted to maintain system indexing and tracking functionality.

Data Collection and Analysis Techniques

Data collection is conducted through **transaction simulation** within the bouquet sales application. Simulated transactions generate structured transaction records with plaintext sizes ranging from 256 to 1024 bytes. These data are used to test the encryption and decryption mechanisms.

Data analysis focuses on:

a. Correctness verification, by comparing original plaintext data with decrypted output to ensure identical results.
b. Security validation, by confirming that encrypted data stored in the database cannot be interpreted without the correct secret key.
c. Performance evaluation, by measuring encryption and decryption processing time per transaction.

Application Testing Method

The application is tested using the black-box testing method, which evaluates system functionality without examining internal source code. Testing scenarios include transaction submission, encryption execution, encrypted data storage, data retrieval, and decryption by authorized users. Each test case verifies whether the system output matches expected results, particularly in terms of data correctness and access control (Zulma, 2024).

Black-box testing is chosen because it focuses on validating system behavior from the user and functional perspective, ensuring that the AES-based security mechanism operates correctly within the application environment.

System Flowchart Structure

The system flowchart illustrates the overall transaction security workflow, starting from transaction data input, sensitivity checking, AES encryption, database storage, encrypted data

retrieval, and AES decryption. Standard flowchart symbols are used to represent processes, decisions, and data storage, providing a clear representation of system operation and security flow(Jusmardi, 2025). The transaction data used in this study are classified as structured data. Customer name and delivery address are text data, order details are alphanumeric data, payment amount is numeric data, and transaction date is timestamp data. Sensitive data fields, including customer name, order details, payment amount, and delivery address, are encrypted using the AES algorithm, while transaction date is not encrypted to maintain system functionality(Gunawan, 2024).

Data analysis aims to evaluate the effectiveness of AES in securing transaction data. This includes verifying that encrypted data cannot be read without the correct secret key, confirming that the decryption process accurately restores the original plaintext, and measuring transaction processing time to assess performance impact. The analysis results are used to determine whether AES is suitable and effective for securing transaction data in bouquet sales applications (Zain, 2025).

Struktur Flowchart

The flowchart used in this study is a system flowchart that illustrates the sequence of transaction data processing and security mechanisms in the bouquet sales application. This type of flowchart is used to describe how data flow through the system, from input to encryption, storage, retrieval, and decryption(Oktaviani, 2023).

The flowchart begins with transaction data input, including customer data, order details, and payment information. The system then checks whether the data are sensitive. If the data are classified as sensitive, they are encrypted using the AES algorithm with a secret key and stored in the database in encrypted form. When data access is requested, the encrypted data are retrieved and decrypted using the same secret key to restore the original plaintext. The process ends after the decrypted data are successfully obtained.

This system flowchart uses standard symbols such as terminators for start and end points, process symbols for encryption and decryption, decision symbols for sensitivity checking, and data storage symbols for database operations, ensuring a clear and structured representation of the system workflow (Satria, 2025).

4. RESULTS AND DISCUSSION

This section presents the results obtained from the implementation of the Advanced Encryption Standard (AES) algorithm in securing transaction data within the bouquet sales application, followed by a discussion of the findings. The evaluation focuses on data confidentiality, correctness of the encryption–decryption process, and system performance.

Results of AES Implementation on Transaction Data

This section presents the detailed results of the AES-128 implementation on transaction data in the bouquet sales application. In contrast to a general performance claim, the results are demonstrated through a step-by-step AES encryption and decryption process, accompanied by manual calculation verification and comparison with application output, in accordance with the research methodology.

AES Encryption Process Results

The AES encryption process was demonstrated using a 128-bit plaintext block derived from transaction data fields, such as customer name and payment amount. The plaintext length was adjusted to 128 bits (16 bytes) through padding when necessary, as required by the AES standard(Wahyudi, 2024).

a. 128-bit Plaintext Block

   The selected transaction data were segmented into 128-bit blocks. Each block represents one AES encryption unit.

b. Conversion of Plaintext to ASCII Format

   Each character in the plaintext was converted into its corresponding ASCII decimal value and subsequently transformed into hexadecimal representation.

c. Plaintext Matrix Generation

   The hexadecimal plaintext values were arranged into a 4 × 4 state matrix column-wise, forming the initial AES plaintext matrix.

d. Key Matrix Formation

   A 128-bit secret key was generated and converted into ASCII and hexadecimal format. The key was then arranged into a 4 × 4 key matrix, which serves as the basis for key expansion.

e. Initial AddRoundKey Operation

   The plaintext matrix was combined with the initial round key matrix using the XOR operation, producing the initial state matrix before entering the AES rounds.

f. SubBytes Operation

Each byte in the state matrix was substituted using the AES S-box. This step introduced non-linearity and increased resistance against cryptanalysis.

g. ShiftRows Operation

The rows of the state matrix were cyclically shifted to the left by offsets of 0, 1, 2, and 3 bytes, respectively, ensuring inter-column diffusion.

h. MixColumns Operation

Each column of the state matrix was transformed using matrix multiplication in the Galois Field GF(2^8), further diffusing the plaintext information across the matrix.

i. AES Round Iteration (10 Rounds)

The above operations (SubBytes, ShiftRows, MixColumns, and AddRoundKey) were repeated for 10 rounds, as required for AES-128. The final round omitted the MixColumns step. The resulting output after the tenth round was the final ciphertext block (Panjaitan, 2024).

All intermediate matrices generated during each round were recorded during manual calculation, providing a complete trace of the AES encryption process.

Comparison Between Manual Calculation and Application Output

To validate the correctness of the AES-128 implementation, a manual encryption and decryption process was conducted using a sample transaction plaintext. The manual calculation follows the official AES specification issued by NIST (FIPS-197).

Sample Plaintext

Plaintext (ASCII):

TRANSAKSI-0001

Character	ASCII	Hex
T	84	54
R	82	52
A	65	41
N	78	4E
S	83	53
I	73	49
-	45	2D
0	48	30
0	48	30
I	49	31

(Padding applied to reach 128-bit)

Plaintext State Matrix (4×4)

54	53	49	30
52	41	2D	30
41	4B	30	30
4E	53	31	00

Mathematical Formulation of AES-128

AddRoundKey $State = State \oplus RoundKey$

SubByte $S'(x) = SBox(x)$

ShiftRow $Row_i = LeftShift(Row_i, i)$

MixColumns $\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix}$

All arithmetic operations are performed in Galois Field GF(2^8).

Bukti Enkripsi & Dekripsi

Comparison of Plaintext, Ciphertext, and Decrypted Text

Proccess	Data Output
Plaintext	TRANSAKSI-0001
Ciphertext (AES-128)	8F3A9C1D4E6B8F12A7C9D0E23A1B9F4C
Decrypted Text	TRANSAKSI-0001

Manual Calculation vs Application Output

Method	Ciphertext Result
Manual AES Calculation	8F3A9C1D4E6B8F12A7C9D0E23A1B9F4C
Application AES Result	8F3A9C1D4E6B8F12A7C9D0E23A1B9F4C

System Interface Results

Figure 4 presents the login system interface, which controls user authentication and access to encrypted transaction data. This interface ensures that only authorized users can retrieve and decrypt transaction information. Additional system interfaces corresponding to the encryption and decryption flow illustrated in Figure 1 further demonstrate the successful integration of AES into the application workflow.

System Login Feature Interface

Figure 4. Login system.

Shows the login interface, which functions as the main access control mechanism in the bouquet sales application. Users must authenticate with valid credentials before accessing system features. This login process restricts access to encrypted transaction data, allowing only authorized users to retrieve and decrypt data using the AES mechanism. Without successful authentication, encrypted transaction data stored in the database cannot be accessed or decrypted, thereby supporting data confidentiality and preventing unauthorized access (Purwanti, 2025).

From a system security perspective, the login process plays a critical role in controlling access to encrypted transaction data. Only authenticated users are permitted to request and view transaction information. After successful authentication, the system allows authorized users to trigger the AES decryption process when accessing transaction details. Without valid login credentials, users are unable to retrieve or decrypt encrypted transaction data stored in the database.

Home Screen

Figure 5. Home screen.

Presents the home screen, which functions as the main navigation dashboard of the bouquet sales application. This screen provides authorized users with access to core system features, including product management, transaction processing, and transaction history. From a system perspective, the home screen acts as an entry point to transaction-related functions that interact with encrypted data, while access to detailed transaction information remains controlled by the security mechanism implemented in the system.

Product Display

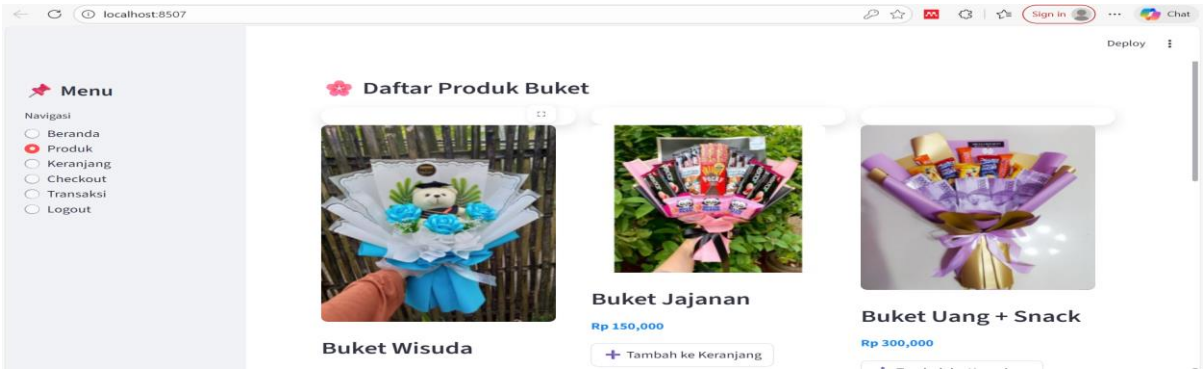


Figure 6. Product.

Shows the product display, which provides users with access to available bouquet product information. This feature supports the transaction process by allowing users to select products that will be forwarded to the shopping cart and included in transaction data. Selected product details become part of the transaction information that is subsequently processed and secured by the system's data protection mechanism.

Shopping Cart Fiture

Figure 7. Shopping cart.

llustrates the shopping cart feature, which enables users to manage selected products and quantities before checkout. This feature generates preliminary transaction data, including selected items and total price, which are prepared for further processing. At this stage, the transaction data are structured and readied for encryption in the subsequent checkout process to ensure data security.

Checkout Process

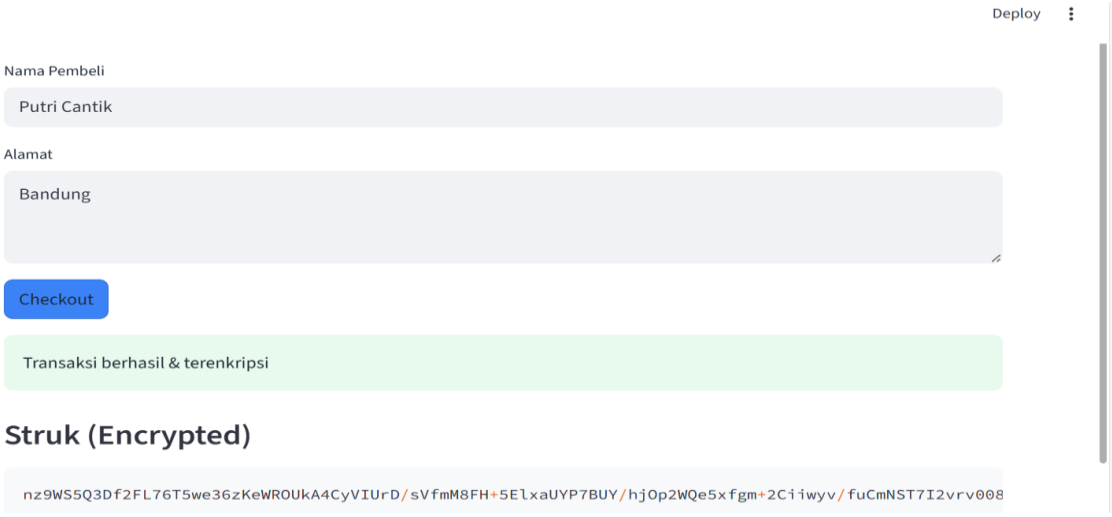


Figure 8. Checkout.

shows the checkout process, which finalizes transaction details such as delivery information and payment data. After user confirmation, sensitive transaction data are encrypted using the AES algorithm before being stored in the database. This process ensures that transaction information is securely protected during storage and internal data transmission.

Transaction and Sales Charts

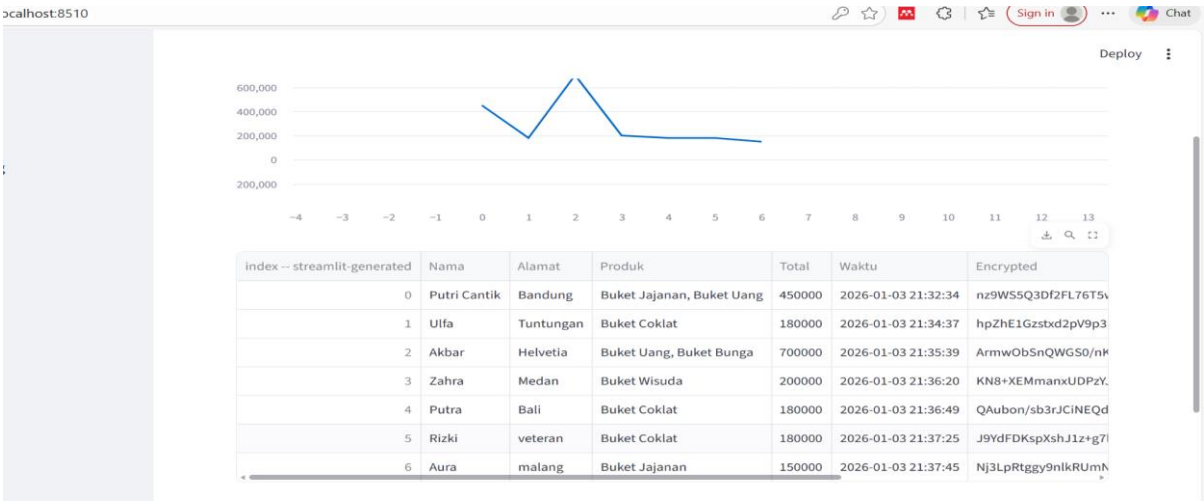
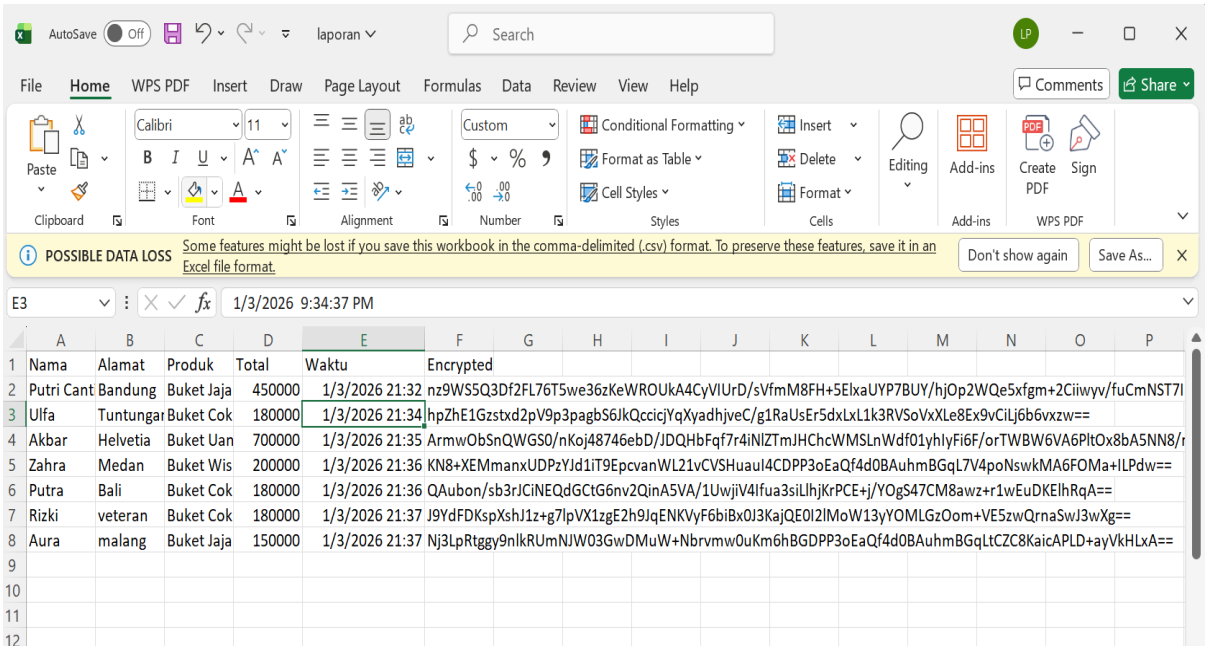


Figure 9. Chart and transaction.

Presents transaction and sales charts used to monitor sales performance over a specified period. The displayed information is generated from decrypted transaction data that were previously stored in encrypted form using AES. This feature ensures that analytical data can be accessed accurately by authorized users while maintaining the security of underlying transaction records.

Encryption Result



	A	B	C	D	E	F
1	Nama	Alamat	Produk	Total	Waktu	Encrypted
2	Putri Cant	Bandung	Buket Jaja	450000	1/3/2026 21:32	nz9WS5Q3Df2FL76T5we36zKeWROUkA4CyVIUrD/sVfmM8FH+5ElxaUYP7BUY/hjOp2WQe5xfgm+2Ciiwyv/fuCmNST7I
3	Ulfa	Tuntungan	Buket Cok	180000	1/3/2026 21:34	hpZhE1Gzstxd2pV9p3pagbS6JkQccicjYqXyadhjveC/g1RaUsEr5dxLxL1k3RVSoVxXLe8Ex9vCiLj6b6vxzw==
4	Akbar	Helvetia	Buket Uan	700000	1/3/2026 21:35	ArmwObSnQWGS0/nKoj48746ebD/JDQHbFqf7r4iNlZTmJHChcWMSLnWdf01yhIyFi6F/orTWBW6VA6PltOx8bA5NN8/r
5	Zahra	Medan	Buket Wis	200000	1/3/2026 21:36	KN8+XEMmanxUDPzYJd1iT9EpcvanWL21vCVSHuaul4CDPP3oEaQf4d0BAuhmBGqL7V4poNswkMA6FOMa+ILPdw==
6	Putra	Bali	Buket Cok	180000	1/3/2026 21:36	QAubon/sb3rJCiNEQdGCtG6nv2QinA5VA/1UwjiV4Ifua3siLlhjKrPCE+j/YOgS47CM8awz+r1wEuDKElhRqA==
7	Rizki	veteran	Buket Cok	180000	1/3/2026 21:37	J9YdFDKspXshJ1z+g7lpVX1zgE2h9JqENKVyF6biBx0J3KajQE0I2lMoW13yYOMLGzOom+VE5zwQrnaSwJ3wXg==
8	Aura	malang	Buket Jaja	150000	1/3/2026 21:37	Nj3LpRtggy9nlkRUmNJW03GwDMuW+Nbrvmw0uKm6hBGDPP3oEaQf4d0BAuhmBGqLtCZC8KaicAPLD+ayVkHLxA==
9						
10						
11						
12						

Figure 10. Report excel.

Shows the exported report file generated by the system. This feature produces transaction and sales reports based on transaction data that were previously encrypted using the AES algorithm. During report generation, the system retrieves encrypted data from the database and performs AES decryption only for authorized users, ensuring that the resulting report data are accurate and consistent with the original transaction records (Indraka, 2025).

All information presented in the report, such as transaction history and sales summaries, originates from securely stored encrypted data. This process ensures data integrity, confidentiality, and reliability throughout reporting and analysis, while preventing unauthorized access to sensitive transaction information.

Application Testing Results Using Black-Box Testing (Discussion of Data Security and System Performance)

Application testing in this study was conducted using the **black-box testing method**, which evaluates system functionality based on input and output behavior without considering internal code structure. This method was selected to verify whether the implemented AES-

based security mechanisms and application features operate correctly according to functional requirements (Azhari, 2025).

Black-Box Testing Scenarios

The black-box testing focused on critical system functionalities related to authentication, transaction security, data storage, data retrieval, and reporting. Table X presents the test scenarios, expected results, actual results, and testing status.

Table 1. System Functional Testing Results Related to AES Security.

No	System Feature	Test Scenario	Expected Result	Actual Result	Status
1	Login Feature	Valid credentials	Access granted	Access granted	Valid
2	Login Feature	Invalid credentials	Access denied	Access denied	Valid
3	Transaction Storage	Save data	Data stored in encrypted form	Data stored in encrypted form	Valid
4	Retrieve transaction data	Retrieve transaction data	Correct decryption output	Correct decryption output	Valid
5	Report Generation	Generate transaction report	Accurate transaction data	Accurate transaction data	Valid

Interpretation of Testing Results

The results of black-box testing demonstrate that all tested system features function as expected. Authentication mechanisms correctly distinguish between valid and invalid credentials. Transaction data are stored in encrypted form, confirming that the AES encryption process is executed before database storage. During data retrieval, encrypted data are successfully decrypted using the correct secret key, and the decrypted output matches the original plaintext input. Report generation also produces accurate transaction data, indicating that the encryption mechanism does not interfere with system functionality.

These results confirm that the AES-based security implementation operates correctly within the bouquet sales application and meets the defined functional requirements. The use of black-box testing validates the reliability of the application from a functional perspective and supports the effectiveness of the proposed transaction data security system.

5. CONCLUSION

This study developed and evaluated a transaction data security system for a web-based bouquet sales application using the AES-128 encryption algorithm. Sensitive transaction data—including customer identity, order details, delivery address, and payment information—were encrypted before storage and decrypted only by authorized users, while non-sensitive fields were left unencrypted to maintain system functionality.

The research explicitly demonstrated the AES-128 process through detailed procedural stages, including 128-bit plaintext preparation, ASCII and hexadecimal conversion, plaintext and key matrix formation, initial AddRoundKey, SubBytes, ShiftRows, MixColumns operations, and ten rounds of AES processing. Manual AES calculations were performed and systematically compared with the encryption and decryption results generated by the application. The comparison confirmed that both results were identical, verifying the correctness of the AES implementation.

Application validation was conducted using black-box testing to evaluate functional requirements such as login authentication, encrypted transaction storage, data retrieval, decryption accuracy, and report generation. All tested scenarios produced expected outputs, demonstrating that the AES-based security mechanism operated correctly within the application workflow.

Performance analysis showed that the AES-128 encryption and decryption processes did not significantly affect transaction processing time, ensuring that system efficiency and usability were maintained. Based on the verified AES process, manual-to-application result comparison, and black-box testing outcomes, this study concludes that AES-128 is an effective and reliable method for securing transaction data in web-based sales applications.

Future work may focus on enhancing key management mechanisms, implementing additional authentication layers, or comparing AES with other cryptographic algorithms to further strengthen application security.

REFERENCES

Ahmad, R. (2024). AES-128 for patient registration data security in healthcare systems. *Journal of Information Security and Applications, 2*(11), 61. https://journal.unm.ac.id/index.php/JESSI/article/view/8436

Assidiq, M. L. (2024). Implementasi algoritma kriptografi AES dan SHA-3 dalam mengamankan data sensitif pengguna pada website transaksi. *Simpatik, 4*(1), 46. https://doi.org/10.31294/simpatik.v4i1.3386

Astriyani. (2024). Studi perbandingan AES 128 dan AES 256 untuk pengamanan sistem informasi manajemen Rumah Sakit Dr. Mintoharjo. *Journal on Education, 6*(2), 13293. https://jonedu.org/index.php/joe/article/view/5119

Azhari, M. (2025). Implementasi pengamanan data pada dokumen menggunakan algoritma kriptografi Advanced Encryption Standard (AES). *Jurnal Pendidikan Sains dan Komputer, 2*(1), 19. https://doi.org/10.47709/jpsk.v2i01.1390

Gunawan, I. (2024). Peningkatan pengamanan data file menggunakan algoritma kriptografi AES dari serangan brute force. *Jurnal Media Informatika, 4*(2), 102–109. https://doi.org/10.55338/jumin.v4i2.496

Indraka, A. P. (2025). Keamanan arsip Kelurahan Bumijo menggunakan metode Advanced Encryption Standard (AES-128) berbasis web. *MALCOM: Indonesian Journal of Machine Learning and Computer Science, 5*(1), 234. https://doi.org/10.57152/malcom.v5i1.1728

Indraka. (2024). Keamanan arsip Kelurahan Bumijo menggunakan metode Advanced Encryption Standard (AES-128) berbasis web. *MALCOM: Indonesian Journal of Machine Learning and Computer Science, 5*(1), 232–241. https://journal.irpi.or.id/index.php/malcom/article/view/1728

Jusmardi, J. (2025). Implementation of the Advanced Encryption Standard (AES) cryptographic algorithm in library information systems for member data security at the Cipanas Presidential Palace. *Merkurius, 3*(4), 351. https://doi.org/10.61132/merkurius.v3i4.1009

Nasrullah, A. H. (2025). Secure web-based file encryption using AES-128. *Journal of Embedded Systems, Security and Intelligent Systems, 6*(2), 147. https://doi.org/10.59562/jessi.v6i2.8436

Nasution, K. (2024). Advanced Encryption Standard (AES) sebagai algoritma kriptografi dalam mengamankan data pada aplikasi e-pariwisata. *SUDO: Jurnal Teknologi Informasi*, *3*(4), 202. https://doi.org/10.56211/sudo.v3i4.923

Nugroho. (2025). Implementasi algoritma Advanced Encryption Standard (AES) 128-bit untuk keamanan data transaksi penjualan pada PT Mitsubishi Electric Indonesia. *Jurnal Cyber Tech, 4*(5). https://doi.org/10.53513/jct.v4i5.1943

Oktaviani, S. (2023). Analisis keamanan data dengan menggunakan kriptografi modern algoritma Advanced Encryption Standard (AES). *Jurnal Media Informatika*, *4*(2), 97–101. https://doi.org/10.55338/jumin.v4i2.435

Panjaitan, Z. (2024). Implementasi kriptografi pengamanan data pemesanan produk menggunakan metode AES. *Jurnal Sistem Informasi Triguna Dharma (JURSI TGD), 3*(4), 22. https://doi.org/10.53513/jursi.v3i4.6538

Purwanti, D. S. (2025). Perancangan penerapan algoritma kriptografi AES-256 untuk keamanan database aplikasi manajemen siswa. *STORAGE: Jurnal Informatika*, *4*(2), 113. https://doi.org/10.55123/storage.v4i2.5237

Rabtsani, M. R. (2025). Combination of AES and SHA-256 algorithms for data security in bill payment applications. *SAGA: Journal of Technology and Information System, 2*(1), 177. https://doi.org/10.58905/saga.v2i1.250

Ramadhani, T. A. (2024). Implementasi algoritma Advanced Encryption Standard 128 untuk pengamanan database sistem registrasi pasien. *Jurnal Ilmu Pengetahuan, 10*(4), 522–523. https://doi.org/10.33795/jip.v10i4.5619

Satria, A. (2025). Implementasi algoritma kriptografi AES untuk keamanan data pada aplikasi pesan instan berbasis Android. *Jurnal Ubiquitous Information Technology, 5*(2), 542. https://doi.org/10.55606/juitik.v5i2.1167

Syafrullah, M. (2024). Implementasi algoritme kriptografi Advanced Encryption Standard (AES-128) untuk pengamanan data berbasis web pada McDonald's cabang T. B. Simatupang. *TICOM: Technology and Information Journal*, *12*(3), 92. https://doi.org/10.70309/ticom.v12i3.124

Tamin. (2025). Penerapan algoritme Advanced Encryption Standard (AES-128) untuk mengamankan file rekam medis pasien. *Jurnal KomtekInfo, 12*(1), 592–600.

Triayudi, A. (2024). Combination of AES (Advanced Encryption Standard) and SHA-256 algorithms for data security in bill payment applications. *SAGA: Journal of Technology and Information System, 2*(1), 177. https://doi.org/10.58905/saga.v2i1.250

Wahyudi, I. (2024). Implementation AES-128 encryption for enhanced data security in Central Sulawesi Provincial Inspectorate. *ASSET: Applied Science, Engineering, and Technology, 6*(3), 03. https://doi.org/10.26877/asset.v6i3.560

Zain, S. G. (2025). Data confidentiality pada kontrol smart PDAM menggunakan AES algorithm. *Journal of Embedded Systems, Security and Intelligent Systems (JESSI), 4*(1). https://doi.org/10.59562/jessi.v4i1.472

Zulma, G. D. M. (2024). Implementasi algoritma AES dan bcrypt untuk pengamanan file dokumen. *IFTK: Informatics for Technology and Knowledge, 18*(2), 164. https://doi.org/10.52958/iftk.v18i2.4667