Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th February 2026. Vol. 5. No. 2; e-ISSN: 2808-4519

Digital Image Data Security Using MATLAB -Based AES-128 Algorithm

Mazayah Tsaqofah [1] *, Juwita Sari [2] , Nurhidayati [3] , Adinda Tarisyah Hsb [4] , Rizky Abdillah [5] , Mrs. Rusdi [6]

[123456] *Computer Science , Faculty of Science and Technology , State Islamic University of North Sumatra*
tsaqofahmazayah@gmail.com[1*] , juwita0701221028@uinsu.ac.id[2] , nurhidayati0701222122@uincu.ac.id[3] ,
adinda0701222116@uinsu.ac.id [4] , kiabdillah90@gmil.com [5] , ibnurusdi@darmawangsa.ac.id [6]

Abstract

Digital images are widely used for storing and transmitting sensitive information , so that aspect data security is becoming very important. This research discuss digital image data security using the Advanced Encryption Standard (AES) algorithm with a long key 128 bit implemented in MATLAB environment. The method used is cryptography . symmetric , where the encryption and decryption processes use a key the same secret . The encryption process done to randomize the pixel value of the original image so that visual information cannot be recognized , while the decryption process aim return image encrypted to condition back to the beginning without data changes . System evaluation done through histogram analysis , process time measurement , and evaluation quality image using the Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) parameters. The test results show that The encrypted image has a random histogram pattern and does not resemble original image . In addition that , the decryption process produce MSE value of zero and PSNR value not finite , which indicates the absence of lost information . Thus , it can concluded that the MATLAB -based AES-128 algorithm is effective and reliable in improving the security of digital image data .

Keywords *: AES-128, Digital Image, Encryption , Decryption , Matlab*

1. Introduction

The rapid progress of information and communication technologies has led to a growing reliance on digital images as an essential medium for representing and exchanging information. Digital images are extensively applied in many sensitive and critical domains, such as medical imaging, biometric identification, military operations, satellite observation, surveillance systems, and digital forensic analysis. Given their critical role and sensitive content, digital images often store confidential information that requires protection against unauthorized access and misuse [1] . During storage and transmission processes, digital image data faces numerous security threats, including interception, unauthorized copying, data manipulation, and illegal distribution. Compared to textual data, digital images generally have larger file sizes and higher levels of redundancy, making them more vulnerable to security breaches if appropriate protection mechanisms are not implemented. As a result, maintaining the confidentiality, integrity, and authenticity of digital image data has become a major concern in contemporary information systems [2] .

Cryptography serves as a fundamental approach for safeguarding digital data by converting original information into an unreadable format that can only be interpreted by authorized parties. Among various cryptographic techniques, the Advanced Encryption Standard (AES) is widely acknowledged as a robust and efficient symmetric encryption algorithm. AES-128 employs a 128-bit secret key, offering a high level of security while ensuring efficient processing speed and minimal computational cost, which makes it well suited for real-time systems and digital image encryption [3] . The effective application of cryptographic algorithms depends on a reliable computational platform that supports accuracy and performance. MATLAB is a high-level programming environment commonly used for numerical analysis, matrix-based computation, and digital image processing. Its comprehensive built-in functions and strong visualization features make MATLAB an appropriate platform for implementing and evaluating encryption and decryption techniques on digital images [4] .

This research concentrates on the application of the AES-128 algorithm for securing digital image data using MATLAB. The encryption process transforms the original image into an encrypted form that appears random and unintelligible, whereas the decryption process reconstructs the encrypted image back to its original state using the correct se [1] cret key. The performance of the proposed approach is assessed through visual analysis and by verifying that the decrypted image preserves the quality and informational content of the original image. This study is expected to demonstrate that AES-128 is an effective and reliable method for enhancing digital image security [5] .

Before the encryption process can be applied, digital image data must be properly represented in a format that is compatible with the AES-128 algorithm. A digital image is initially read as a two-dimensional matrix consisting of pixel intensity values. These pixel values are then converted into an 8-bit byte format and arranged into 128-bit data blocks, which correspond to the block size required by the AES

encryption standard. This data preparation stage is essential to ensure that every part of the image can be processed accurately and securely during encryption and decryption [6] .

The security strength of the AES-128 algorithm is highly dependent on the secrecy and randomness of the encryption key. The key used in AES-128 must be exactly 128 bits in length and should be generated using a method that produces high randomness to prevent predictability. Proper key management is therefore a critical aspect of the encryption system, including secure key storage, controlled key distribution, and correct usage during both encryption and decryption processes. Any compromise in key handling may significantly reduce the overall security of the image encryption system [7] .

One important indicator of successful image encryption is the change in the statistical characteristics of the image, particularly the pixel value distribution. In an original image, pixel values typically follow a structured distribution that reflects the visual content of the image. After encryption, the encrypted image should exhibit a nearly uniform and random histogram distribution. This transformation indicates that the visual information and inherent patterns of the original image have been effectively concealed, making statistical analysis attacks more difficult [8] .

In addition to visual and statistical analysis, the robustness of an image encryption system must be evaluated against common cryptographic attacks. AES-128 is designed to resist various attack methods, including brute force attacks, statistical attacks, and known-plaintext attacks. With a key space of 2^{128} possible combinations, AES-128 provides an extremely high level of security, making brute force attacks computationally infeasible using current technology. Furthermore, the strong diffusion and confusion properties of AES ensure that small changes in the input data result in significant differences in the encrypted output [9] .

Performance efficiency is another important consideration in digital image encryption, especially when dealing with large image sizes. The computational complexity of the encryption and decryption processes affects the overall system performance. AES-128 offers a favorable balance between security and speed, allowing image data to be encrypted and decrypted efficiently without excessive computational overhead. This makes the algorithm suitable for practical applications that require both high security and fast processing. Despite its advantages, the implementation of AES-128 for digital image security also presents certain challenges. Large image sizes may increase processing time and memory usage, particularly in software-based implementations. Additionally, any errors in data conversion, block arrangement, or key usage can result in unsuccessful decryption or loss of image information. Therefore, careful implementation and thorough testing are required to ensure that the encrypted image can be accurately restored to its original form [10] .

2. Methodology

System digital image data security using the AES-128 algorithm built with language programming MATLAB **.** Security process image done through two stages main , namely encryption and decryption , using the same key because AES is a cryptographic algorithm symmetric . In this study, the AES-128 algorithm was used with a length of key 128 bit . Encryption process aims to change the original image (*plain image*) into a plain image encrypted (*cipher image*) so that visual information cannot be recognized . Next , the decryption process done to restore image encrypted to the original image form without lost information . The selection of AES-128 is based on the level of high security as well as efficiency good computing , so it is suitable for application in MATLAB- based digital image data security .

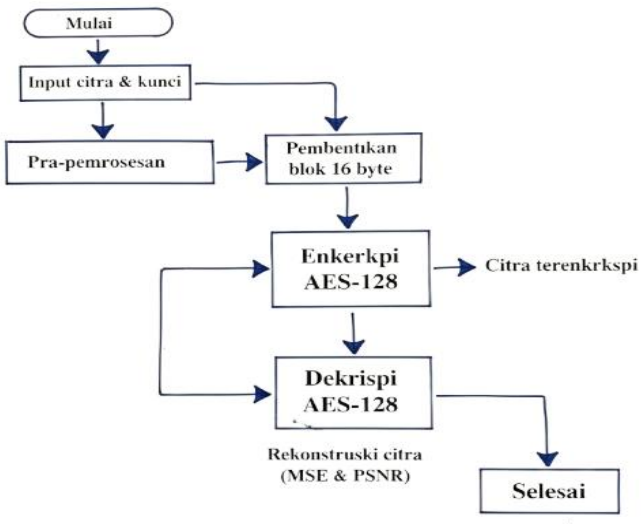


Fig. 1:. Application Flowchart

The Rijndael algorithm works by applying the processes of substitution , permutation , and a number of rounds on each data blocks to be encrypted and decrypted . On each round , used different keys known as *round key* . Rijndael has a fixed block size of 128 bits or equivalent to 16 bytes. This algorithm support long key starts from 128 bit to 256 bit in 32 bit increments . However, deep Advanced

Encryption Standard (AES) standard , length The keys used are limited to 128 bits, 192 bits, and 256 bits, so known variants of AES-128, AES-192, and AES-256 [11] . The Rijndael algorithm has three main parameters , namely :

1. *Plaintext* , in the form of an array of size 16 working bytes as input data
2. *Ciphertext* , which is an array of size 16 bytes that store the results of the encryption process
3. Key , in the form of an array of size 16 bytes used as key password or *cipher key* in the encryption and decryption process .

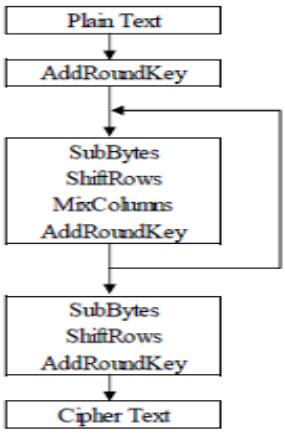


Fig. 2: Encryption Process Diagram

Encryption process steps :

1. Plain Text
   The process begins with input data in the form of processed plain *text* in block sized 128 bit **.**
2. AddRoundKey (Initial Round)
   Plain text combined with the initial key using the XOR operation to enter influence key into the data before transformation main done .
3. SubBytes
   Each byte of data is replaced using a non-linear Substitution Box (S-Box) to increase the level of *confusion* .
4. ShiftRows
   Rows in the matrix *state* shifted cyclically to spread the relationship between bytes (*diffusion*).
5. MixColumns
   Each column in *the state* transformed using linear algebra operations to strengthen the *diffusion* process .
6. AddRoundKey (Main Round)
   Transformation result combined with *round key* through the XOR operation.
7. Round
   Repetition Steps 3 to 6 are repeated as much as nine times on the AES-128 algorithm.
8. Final Round
   In the final *round* , the MixColumns process is not performed , so that it only involves SubBytes , ShiftRows , and AddRoundKey .
9. Cipher Text
   The final result of all over the process series is cipher text , which is data that has been encrypted and cannot be read read without appropriate key .

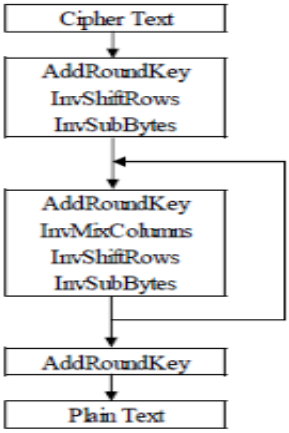


Fig. 3: Decryption Process Diagram

Here are some steps in the process of decryption as follows:

1. Cipher Text
   Decryption process started with input data in the form of *cipher text* , namely results from the AES encryption process that is not can read in a way direct .
2. AddRoundKey (Initial Round)
   Cipher text combined with *round key* final use XOR operation . In the decryption process , the sequence use key done in a way backwards compared to with the encryption process .
3. InvShiftRows
   Rows in the matrix *state* shifted in a way cyclic to opposite direction from the ShiftRows process on encryption For return original byte position .
4. InvSubBytes
   Each byte in *the state* restored using the Inverse S-Box, which is opposite from the substitution process at the stage SubBytes .
5. Main Round Process
   Stage furthermore done in a way repeatedly (as many as nine rounds on AES-128) with order transformation :
   a. AddRoundKey
   b. InvMixColumns
   c. InvShiftRows
   d. InvSubBytes
   Transformation InvMixColumns functioning reverse effect diffusion generated by MixColumns in the encryption process .
6. Final AddRoundKey
   After all over *round* finished , the AddRoundKey process is carried out final For combine results transformation with key beginning .
7. Plain Text
   Final result from the decryption process is plain text, namely the original data before encrypted .

3. Implementation and Testing

3.1.Testing Application

Testing application explain about method program operation and necessary steps done by the user to run application image file encryption and decryption .

3.2. Main Menu View

Main menu view The application is a display where you can start using the application as seen in Figure 7.

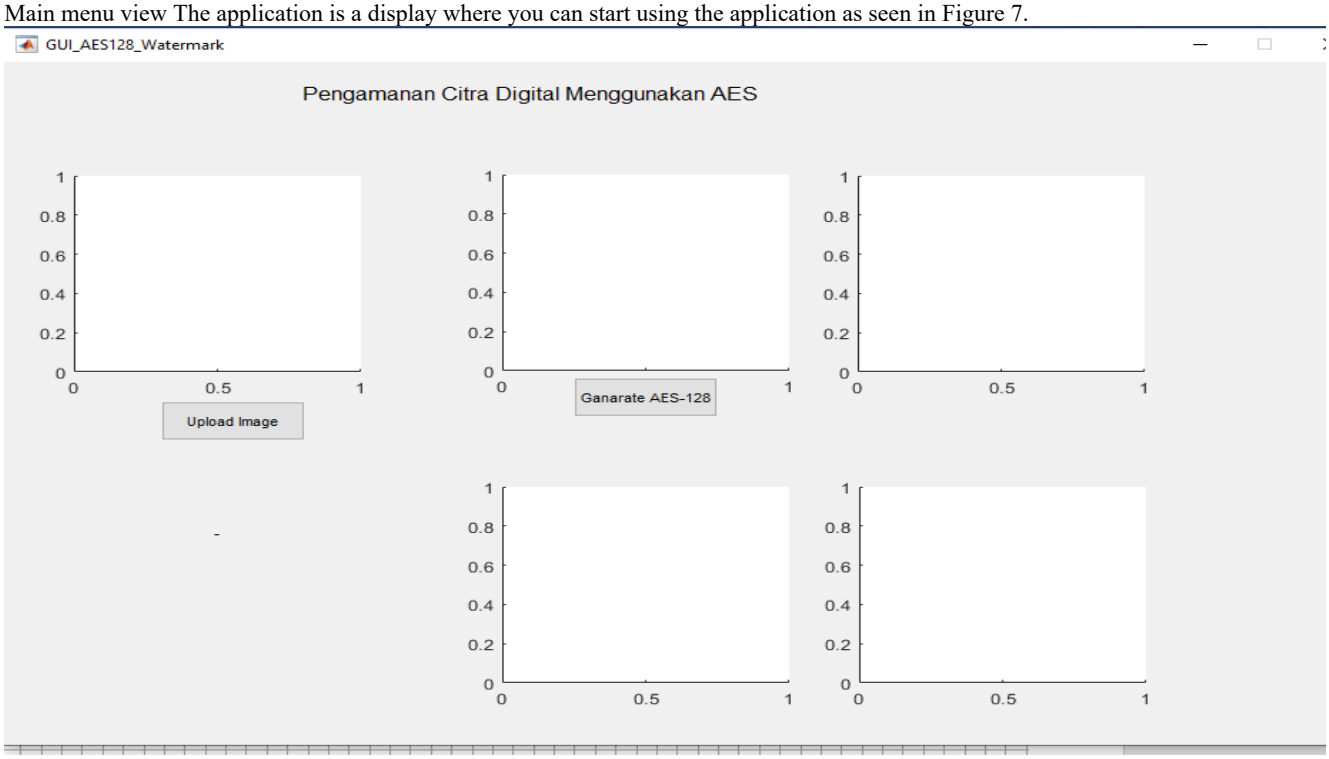


Fig. 4: Application view

To do encryption and decryption using the AES algorithm using the same plain image , the user can choose the type of AES to be used by pressing one of the existing push button . For example , the user selects type AES-128, then system will display encryption and decryption results along with time and histogram .

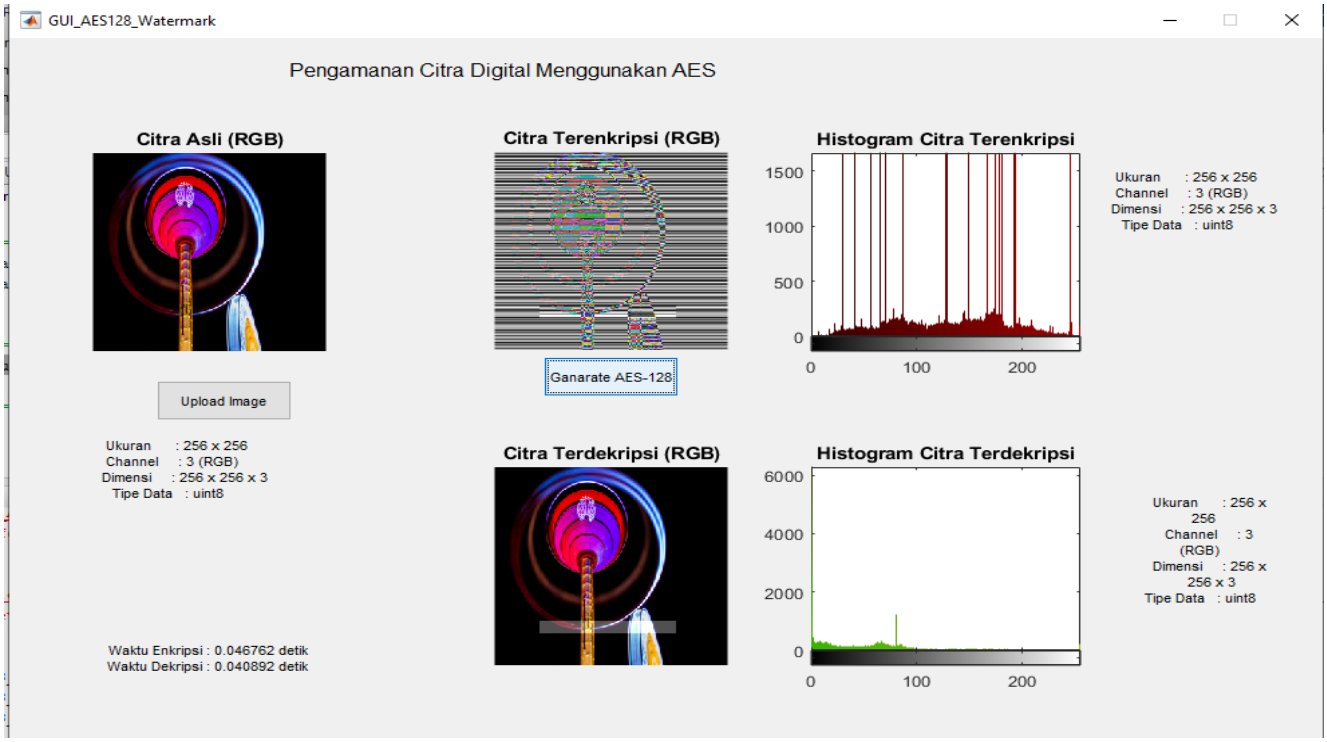
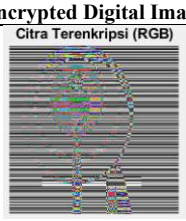
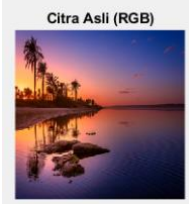
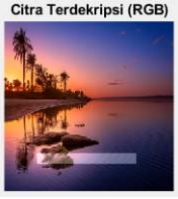


Fig. 5: Display of AES Algorithm Process Results

4. Testing and Analysis of Test Results

This trial , it will is done with 2 scenarios for each algorithm. In the AES algorithm, the first scenario is done testing with various pairs public key and private key private to the same plain image , and the second scenario is done testing with various image dimensions against pairs same key .

Table 1: AES Encryption and Decryption Results

No	Original Digital Image	Encrypted Digital Image	Encryption Time	Encrypted Digital Image	Time Description
1			0.046762s		0.0408893s
2			0.041507s		0.036305s
3			0.039967s		0.036079s
4			0.062302s		0.042058s

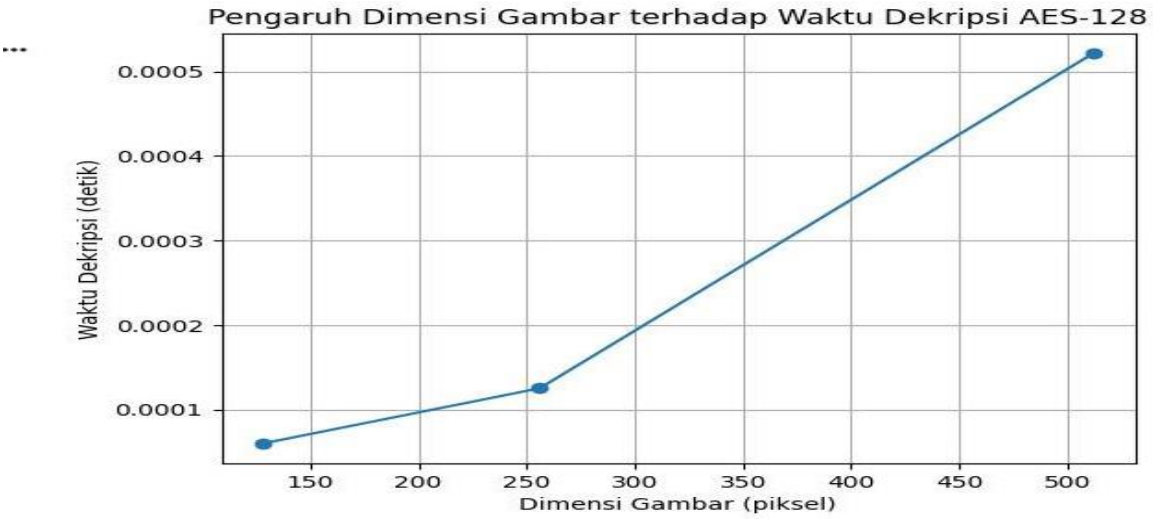


Fig. 6: Influence Image Dimensions Against Encryption Process

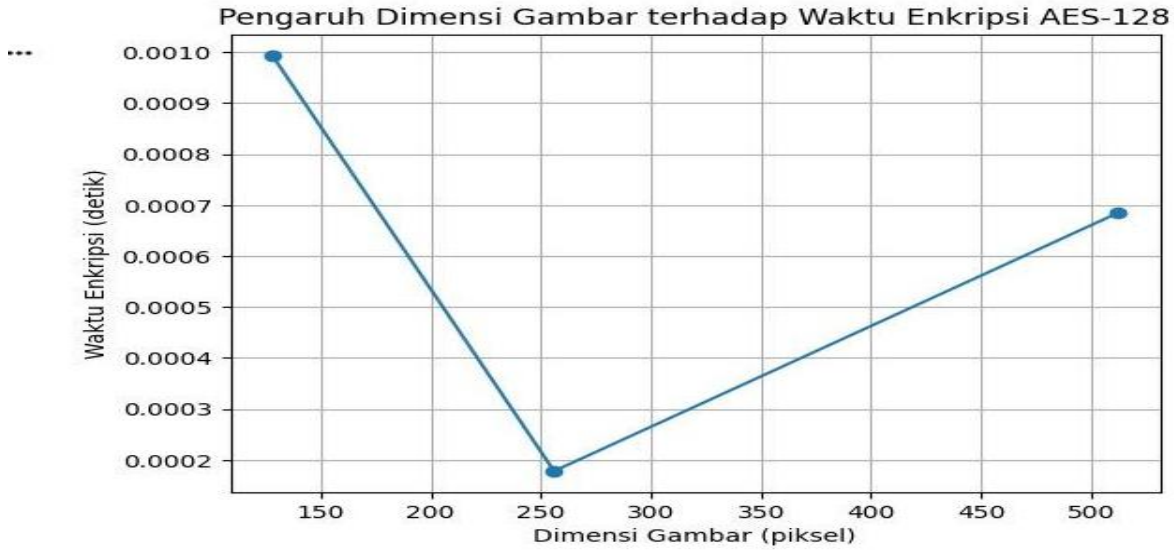


Fig. 7: Influence Image Dimensions versus Decryption Process

Diagrams 2 and 3 show How big small influence dimensions an image on the encryption and decryption process . The size of the image greatly influences the encryption and decryption process . Because, the larger the image, the larger the image . size image dimensions , the more Lots *pixels* that will processed **.**

Fig. 8: Effect of Key Length on the Encryption Process

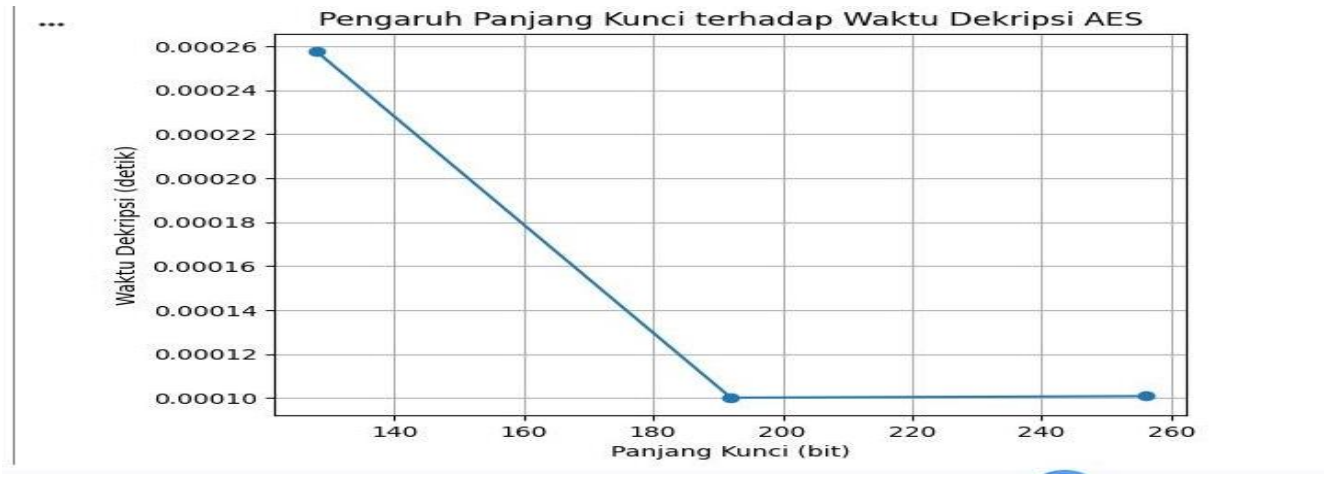
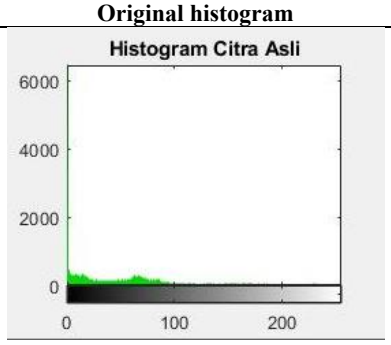
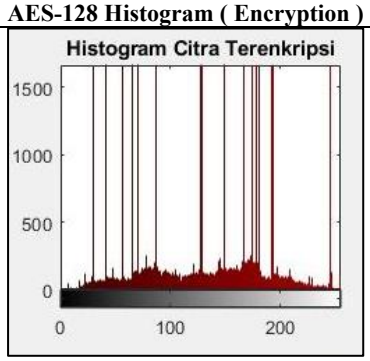
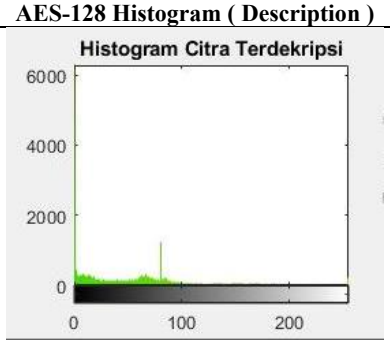
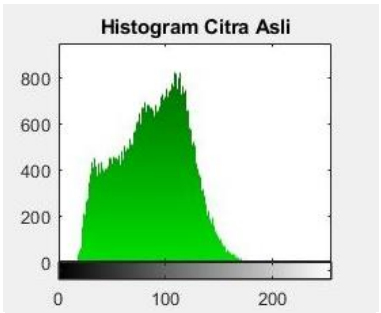
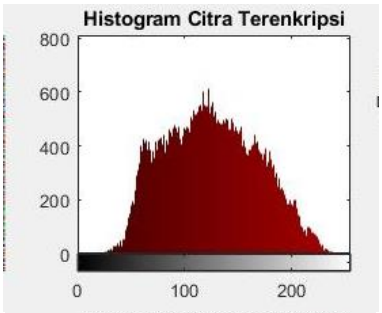
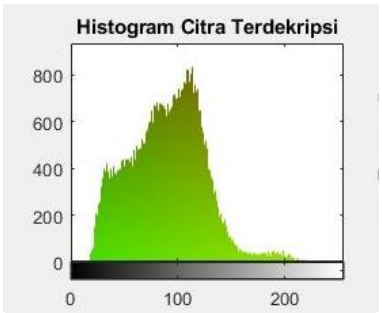
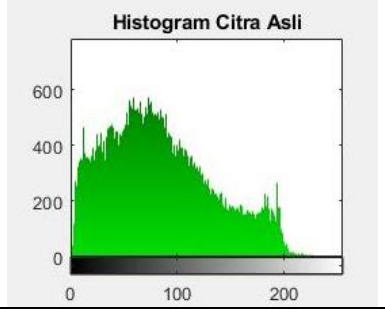
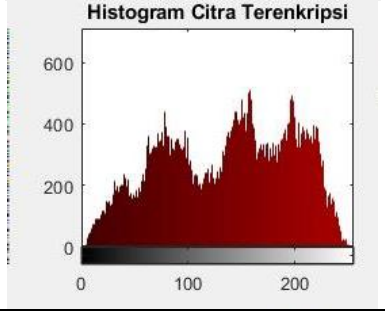
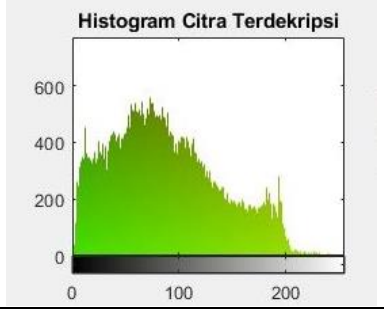


Fig. 9: Effect of Key Length on the Decryption Process

5. Testing Quality Encryption

Table 2: Histogram Comparison

No	Original histogram	AES-128 Histogram (Encryption)	AES-128 Histogram (Description)
1			
2			
3			

4

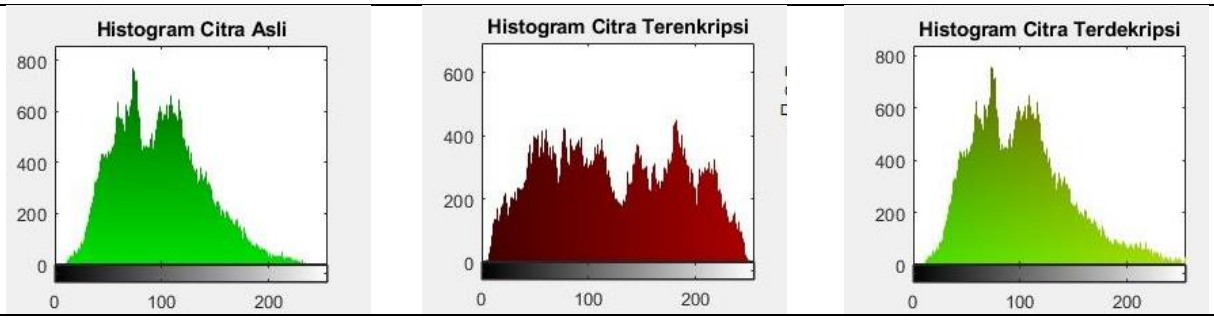


Table 3: MSE and PSNR Values of AES Algorithm Decryption Results

No	Citra's Name	MSE(R)	MSE(G)	MSE(B)	PSNR(R)	PSNR(G)	PSNR(B)
1	Figure 1	0	0	0	inf	inf	inf
2	Figure 2	0	0	0	inf	inf	inf
3	Figure 3	0	0	0	inf	inf	inf
4	Figure 4	0	0	0	inf	inf	inf

Quality decrypted image can seen from PSNR (Peak Signal Noise Ratio) and MSE (Mean Squared Error) values . MSE is mean square error value between original image with the original image and decryption . While PSNR is comparison between mark maximum from quality original image and image that has been undergoing a decryption process . In theory, the PSNR value can calculated using the formula .

PSNR often stated in scale logarithmic in decibels (dB). I max is the value intensity highest . This indicates that the greater the PSNR value , the the better the decryption results , and the more similar image decryption with images original . When the measured MSE is worth 0, then can it is said that second compared images similar One each other. If MSE is 0 then PSNR is automatically valued not finite (infinity). And if MSE is 0 and PSNR is not finite , then can it says there is no lost data .

6. Conclusion

Based on results design , implementation , and testing system security digital image using MATLAB- based AES-128 algorithm , can taken a number of conclusion as following :

Advanced Encryption Standard (AES) algorithm with long key 128 bit succeed implemented for encryption and decryption processes digital image using MATLAB as a programming medium . Encryption process capable change image original become image encrypted that is not can recognized visually , as indicated by the difference histogram distribution between image original and image results encryption . Decryption process use same key with the encryption process can return image encrypted to form back to the beginning without lost information , which is proven with Mean Squared Error (MSE) value is 0 and Peak Signal-to-Noise Ratio (PSNR) value is not infinity . Dimension digital images influence encryption and decryption process time , where the more big size image so the longer it takes Because increasing amount processed pixels . Based on results testing quality image and analysis performance system , AES-128 algorithm is proven effective in increase digital image data security .

References

[1] D. Pramudia, A. Fitriansyah, and R. Ramliyana, "A Hybrid Application of EOF Steganography and AES-128 Encryption for PDF File Security on Android," *J. Ris. and Apl. Mhs. Inform.* , vol. 2, no. 03, pp. 437–444, 2021, doi: 10.30998/jrami.v2i03.1144.

[2] S. Wijaya, "Analysis of Late Payments in Industry," vol. 11, no. 2, pp. 166–177, 2024.

[3] R. Rahmawati and D. Rahardjo, "Data Security Application Using Discrete Cosine Transform Steganography Algorithm and AES 128 BIT Cryptography at SMK PGRI 15 Jakarta," *J. Tech. Inform. and Sist. Inf.* , vol. 2, no. April, pp. 67–74, 2023.

[4] DC Anggarini and C. Kurniawan, "DESIGN OF AES RIJNDAEL ALGORITHM APPLICATION FOR 128 BIT JPEG FILE DIGITAL IMAGE ENCRYPTION," 2024.

[5] Ismai, "Textbook of Basic Concepts of Website Programming with PHP," vol. 7, pp. 51–61, 2024.

[6] ADIBDI Journal, MTIMM Dr. Ir. Untung Rahardja, K. Zelina, and ADI PUBLISHER, *ADI Interdisciplinary Digital Business Journal (ABDI Journal) First Edition Vol 1. No 1. June 2020.* ADI Publisher, 2021.

[7] R. Rahman *et al.* , *Computer Network Security Textbook* . Jambi: PT. Sonpedia Publishing Indonesia, 2024.

[8] Galih Yuga Pangestu, Asep Id Hadiana, and Puspita Nurul Sabrina, "Cryptography for Double Encryption of Images Using AES (Advanced Encryption Standard) and RC5 (Rivest Code 5) Algorithms," *Informatics Digit. Expert* , vol. 4, no. 1, pp. 25–32, 2022.

[9] F. Nuraeni, S. Tinggi, T. Garut, and D. Kurniadi, "Implementation of QR-Code and Digital Signature Schemes using a Combination of RSA and AES Algorithms for Electronic Certificate Data Security Academic Information System View project Educational Data Mining View project," pp. 43–52, 2020.

[10] A. Sapaatullah and M. Darip, "Web-Based Digital Document Security with the AES-128 Algorithm in Kindergarten Educational Institutions," *Betrik* , vol. 16, no. 01, pp. 37–51, 2025, doi: 10.36050/fjb35559.

[11] GG Putri, W. Styorini, and RD Rahayani, "Introduction With the rapid development of technology, the internet has become the main platform in the Theoretical Basis of Cryptography Cryptography is the science and art of maintaining the confidentiality of messages by means of Cryptographic Algorithms Cryptographic algorithms are also called ciphers, namely rules for enchiphering and Symmetric Algorithms These algorithms are also often called classical algorithms, because they use keys that Non-Symmetric Algorithms Non-symmetric algorithms are also often called public key algorithms, with the meaning," pp. 197–207.