Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

June 2026. Vol. 5. No. 3; e-ISSN: 2808-4519

Development and Evaluation of a Desktop Academic Data Encryption Application Using AES with Password-Based Key Management

Muhammad Wahyu Rizqi Pratama[1]*, Haris Yuana[2], Fatikhatul Trisna Ardinansyah[3]

[1,2,3]*Universitas Islam Balitar*

wahyurizqi363@gmail.com[1],harisyuana2010@gmail.com[2], ftardiansyah.net@gmail.com[3]*

Abstract

This study evaluates a desktop-based academic data security application integrating the Advanced Encryption Standard (AES) and Password-Based Key Derivation Function 2 (PBKDF2) at Pondok Pesantren Nur Rohman. Using an experimental approach, three AES key lengths (128-bit, 192-bit, and 256-bit) were tested against 10 authentic academic files to assess computational performance and cryptographic strength. The results show that AES-128 achieves the fastest speed with a throughput of 57.84–65.00 MB/s, while AES-256 requires the longest processing time due to its 14 internal rounds. Regarding security, all variations exhibit ideal mathematical resilience: ciphertext entropy ranges tightly between 7.9988 and 7.9999 (near absolute randomness), and avalanche effect percentages remain stable between 49.85% and 50.19%. Furthermore, PBKDF2 successfully mitigates brute-force vulnerabilities by mapping passwords into precise hexadecimal keys. The process introduces a minimal, constant file size overhead of exactly 64 bytes (Salt, IV, and padding). Hardware utilization is exceptionally efficient, recording low CPU usage (0.1%–0.2%) and stable RAM allocation (53%–54%). In conclusion, the system delivers an optimal equilibrium between high-level data protection and local hardware efficiency.

Keywords: *AES, PBKDF2, Academic Data Security, Cryptographic Performance, Pondok Pesantren Nur Rohman*

1. Introduction

The development of information technology has brought significant changes to data management systems across various educational institutions, including Islamic boarding schools (pondok pesantren) [1]. The digitization of academic data—such as student biodata, grades, administrative archives, financial reports, and learning evaluation documents—facilitates easy storage and distribution of information [2]. However, on the other hand, digital transformation also escalates the risk of data security threats in the form of data theft, information manipulation, unauthorized access, and leaks of sensitive documents due to weak protection mechanisms for digital data [3], [4]. Information security has become a critical aspect of academic data management, as the stored records contain sensitive personal and administrative information [5]. Therefore, a data security mechanism capable of maintaining confidentiality, integrity, and availability of information in an optimal manner is highly required [6].

In recent years, the increasing cases of data breaches in educational institutions indicate that data security systems remain a primary challenge in the implementation of information technology [7]. Many educational institutions still rely on conventional data storage methods without adequate encryption systems, rendering them vulnerable to cyberattacks [8]. Weak authentication and encryption systems are among the main causes of security breaches in the education sector [4]. Furthermore, the protection of academic data must be carried out comprehensively, spanning storage, distribution, and user access, to minimize the risks of digital attacks [9].

One of the most widely used methods to protect digital data is cryptography [10]. Cryptography is a data protection technique that transforms the original information (plaintext) into an encrypted form (ciphertext) so that it cannot be read by unauthorized parties [11]. Modern cryptographic algorithms have advanced rapidly and are implemented in various information security systems because they provide a high level of protection for digital data [12]. The Advanced Encryption Standard (AES) is one of the most widely utilized symmetric cryptographic algorithms due to its high level of security, good computational efficiency, and its status as an international standard for digital data protection [13],[14].

AES offers three variations of key lengths, namely AES-128, AES-192, and AES-256, each offering different levels of security and computational complexity [15]. A longer key size provides a higher level of security, but it also impacts computational time and system resource utilization [16]. Theoretically, AES-256 offers higher security compared to AES-128 and AES-192, but it demands more time for the encryption and decryption processes [17]. Therefore, the selection of the key length must consider the equilibrium between system performance and the required security level [18].

In addition to key length, key management is a vital factor in cryptographic systems [19]. Many encryption system implementations fail to provide optimal protection due to the use of weak passwords or insecure key generation processes [20]. Passwords used directly as encryption keys possess a high level of vulnerability against brute-force and dictionary attacks [21]. Hence, a key derivation method capable of producing stronger and unpredictable encryption keys is required. One commonly used method is the Password-Based Key Derivation Function 2 (PBKDF2) [22]. PBKDF2 operates by transforming a password into a cryptographic key through a repetitive hashing process using a specific salt and iteration count [23]. This method enhances password security by increasing the complexity of the key generation process [24] and has been proven to improve data security compared to using direct passwords without a derivation process [25].

Various prior studies have discussed the implementation of AES in data security systems. The utilization of AES-128 in document security systems demonstrates that the algorithm can secure data effectively within desktop environments [26]. Furthermore, a performance comparison between AES-128 and AES-256 across different file sizes indicates that AES-256 requires longer computational time but delivers a higher level of security [15]. The implementation of AES in web-based academic document archiving systems also shows that encryption effectively maintains document confidentiality [9].

Nevertheless, most previous studies have focused strictly on the implementation of the AES algorithm without conducting a comprehensive evaluation of all AES key length variations alongside the integration of PBKDF2-based key management within a single desktop application system [27]. In addition, crucial aspects such as the expansion of file size after encryption due to the insertion of Salt, Initialization Vector (IV), and padding are frequently overlooked, despite their impact on digital storage efficiency. Prior research generally evaluates encryption performance solely from the aspect of processing time without analyzing mathematical security parameters and its operational impacts on the user's computer [28]. Therefore, more in-depth research is still required regarding the performance evaluation of AES with various key length variations combined with PBKDF2 in a desktop application environment.

This study was conducted to develop and evaluate a desktop-based academic data encryption application using the AES algorithm with password-based key management via PBKDF2. An experimental approach was applied by testing three AES key length variations (128, 192, and 256 bits) against 10 authentic academic data files from Pondok Pesantren Nur Rohman, Sumberdadi Village, Sumbergempol District, Tulungagung Regency.

To guarantee a comprehensive and objective evaluation, the test variables in this study were expanded into two main aspects: computational performance and security strength. The performance aspect was evaluated through the parameters of computational time (encryption/decryption speed), ciphertext size expansion analysis, and hardware resource monitoring (CPU and RAM usage) to ensure the application runs optimally on standard-specification computers. Meanwhile, the protection quality of the ciphertext was evaluated mathematically through avalanche effect testing to measure the algorithm's diffusion characteristics, as well as entropy value testing to analyze the randomness of the ciphertext against statistical cyber-attacks.

The selection of Pondok Pesantren Nur Rohman as the research object was based on the institution's urgent need for a practical, secure, and efficient local academic data protection system that does not corrupt the original data structure. As an institution newly transitioning to digital, system stability within a local desktop ecosystem is the primary choice as it does not rely on an internet connection. The novelty of this study lies in the comparative integration of three AES key length variations combined with PBKDF2 key strengthening, along with an in-depth analysis of ciphertext randomness and system memory efficiency [27], [28]. The results of this study are expected to provide concrete recommendations regarding the most ideal AES variant in delivering an optimal equilibrium between academic data protection strength and computer hardware performance efficiency.

2. Research Method

This study utilizes an experimental research method to develop and evaluate the performance of a desktop-based academic data encryption application integrating the Advanced Encryption Standard (AES) algorithm with Password-Based Key Derivation Function 2 (PBKDF2) key management. The experimental method was selected because the research explicitly focuses on testing the performance and security of multiple AES algorithm variations through direct implementation and the empirical measurement of defined parameters.

2.1 Research Stages

The research stages were conducted systematically, ranging from problem identification to the analysis of experimental results. The research workflow is illustrated in Figure (insert figure number for research workflow), which consists of the following phases:

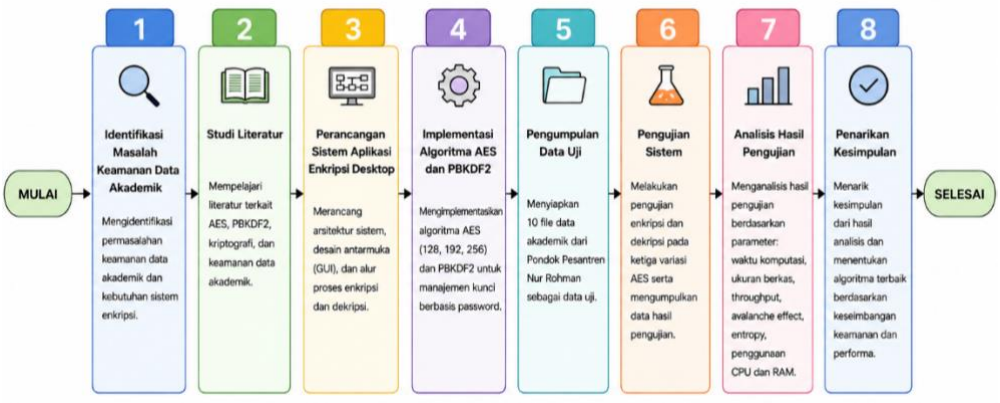

Fig. 1: Research Stages

2.2 Research Object

The research object utilizes the academic data of Pondok Pesantren Nur Rohman, located in Sumberdadi Village, Sumbergempol District, Tulungagung Regency. The evaluation dataset consists of 10 academic files of varying sizes and file formats to determine the impact of data volume on the performance of the encryption algorithms.

2.3 Data Collection Techniques

The study utilizes the following approaches:

a) Literature Study

The literature study was conducted by reviewing journals, books, and prior research related to cryptography, the AES algorithm, PBKDF2, and academic data security.

b) Observation

Observations were carried out on the academic data management system at Pondok Pesantren Nur Rohman to identify data security requirements and the processing workflows of academic documents.

c) Experimentation

Experiments were executed by testing the encryption application using three AES variations, namely AES-128, AES-192, and AES-256, across 10 academic data files.

2.4 System Design

The application is developed as a desktop-based system featuring a Graphical User Interface (GUI) that enables users to perform file encryption and decryption processes seamlessly. The system implements PBKDF2 to generate cryptographic keys from user passwords prior to their utilization within the AES-128, AES-192, and AES-256 encryption processes.

a) Encryption Flowchart

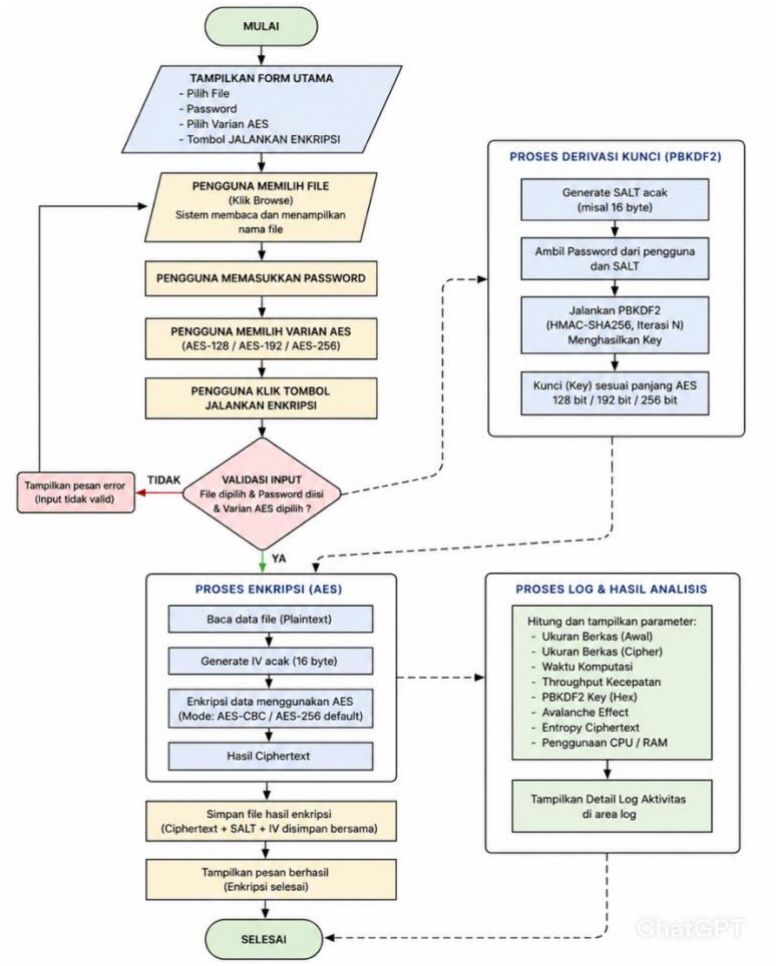

Fig. 2: Encryption Flowchart

b) Decryption Flowchart

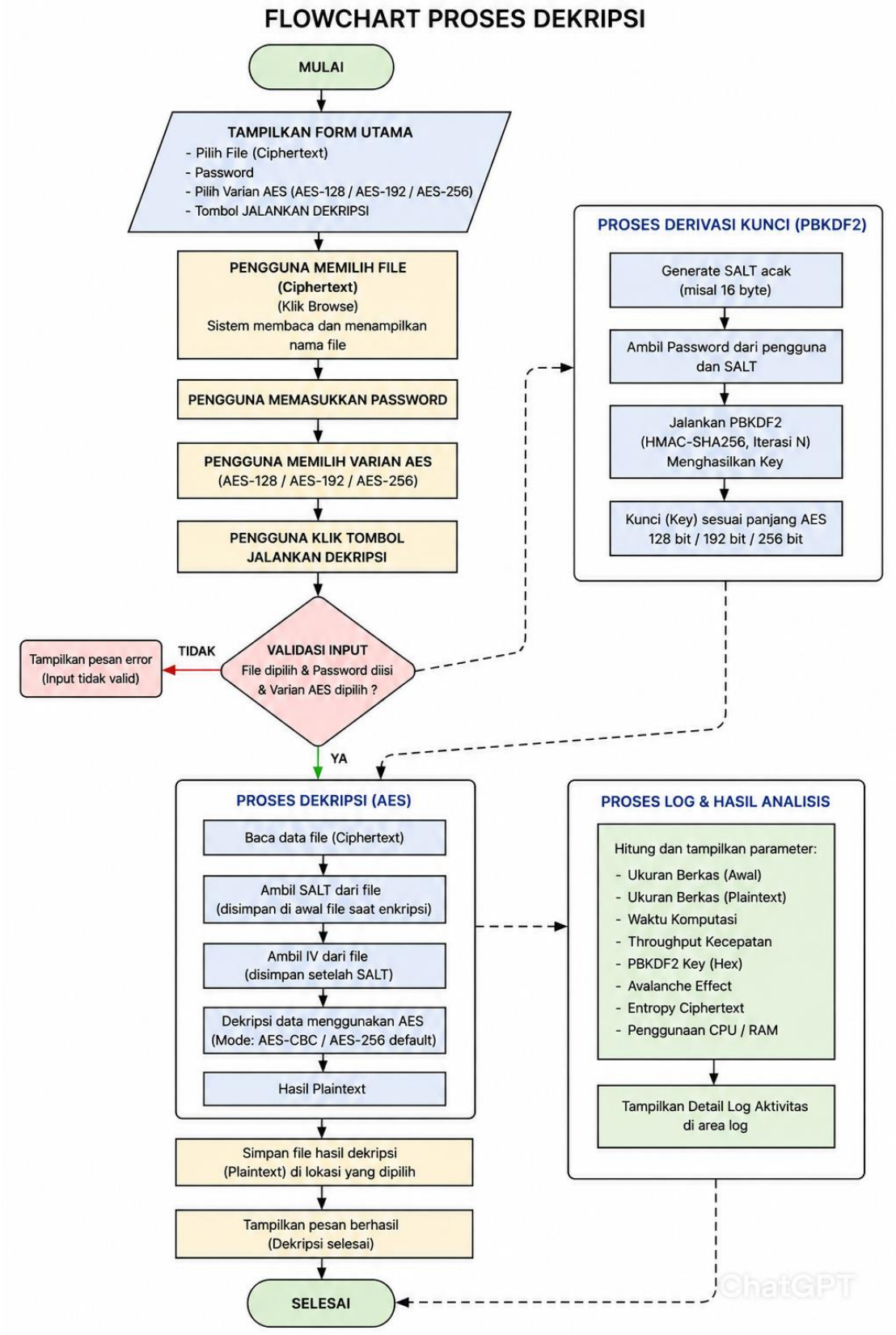

Fig. 3: Decryption Flowchart

2.5 Research Variables

This study utilizes the following variables:

a) Independent Variables

The independent variables in this research are the variations of the AES key lengths, which consist of:

- AES-128
- AES-192
- AES-256

b) Dependent Variables
The measured dependent variables include:
- Computational time for encryption and decryption
- PBKDF2 key generation size
- Ciphertext size
- Processing throughput speed
- Avalanche effect
- Ciphertext entropy
- CPU and RAM utilization

2.6 Testing Methods

The testing is conducted by encrypting and decrypting 10 academic data files using each variation of AES. Each test scenario is executed multiple times to ensure the consistency and reliability of the empirical results.

a) Computational Time Testing
Computational time testing is conducted to determine the system performance during the data encryption and decryption processes. The measurement is carried out by calculating the time difference between the initiation and the completion of the execution process. A lower computational time indicates better and more efficient algorithm performance in processing data.

Computational Time Formula:

$$T = t_{end} - t_{start} \quad (1)$$

Where:
- T = Computational time
- t_{start} = Process start time
- t_{end} = Process completion time

b) PBKDF2 Key Generation Size Testing
This evaluation is conducted to determine the key sizes generated by PBKDF2 for each AES variation. The user's password is processed using PBKDF2 along with a Salt and an iteration count to produce the exact cryptographic keys required for AES-128, AES-192, and AES-256.

The testing parameters include:

- Derived key length
- Key size consistency
- Key compliance with AES standards

The standard key sizes utilized are:

- AES-128 = 128-bit
- AES-192 = 192-bit
- AES-256 = 256-bit

The evaluation is performed across all academic data files using different passwords to ensure that the key generation process operates consistently and complies fully with the standard cryptographic specifications of the AES algorithm.

c) Ciphertext Size Testing
Ciphertext size testing is conducted to determine the changes in file size after undergoing the encryption process using the AES algorithm. This evaluation aims to analyze the impact of the encryption process on the resulting file sizes.

The testing parameters include:

- File size before encryption
- Ciphertext size after encryption
- File size variance (overhead)

The evaluation is performed across all academic data files using AES-128, AES-192, and AES-256. The empirical results are utilized to assess data storage efficiency following the encryption process, as well as to determine the operational impact of embedding the Salt, Initialization Vector (IV), and block padding into the final ciphertext structure.

d) Throughput Testing

Throughput testing is conducted to determine the system's capacity to process data during the encryption and decryption procedures. The throughput value is obtained by comparing the size of the processed data against the total time required by the system. A higher throughput value indicates superior and more efficient algorithm performance in handling large-volume data.

Throughput Formula:

$$\text{Throughput} = \frac{Data\ Size}{Processing\ Time} \tag{2}$$

Where:

- Throughput = Speed of the encryption/decryption process
- Data Size = The volume of the file or data being processed
- Processing Time = The duration required to complete the encryption/decryption process

e) Avalanche Effect Testing

The Avalanche Effect testing is conducted to measure the sensitivity of the algorithm to minor alterations in the plaintext or the encryption key. The evaluation is performed by comparing two ciphertexts generated from plaintexts that possess a single-character or single-bit difference. The Avalanche Effect value is calculated based on the percentage of the number of flipped (changed) ciphertext bits. An algorithm is considered to exhibit optimal performance if its Avalanche Effect value approaches 50%.

Avalanche Effect Formula:

$$\text{AE} = \frac{Number\ of\ Flipped\ Bits}{Total\ Ciphertext\ Bits}\ X\ 100\% \tag{3}$$

Where:

- AE = Avalanche Effect value
- Number of Flipped Bits = The total number of differing bits between the two ciphertexts
- Total Ciphertext Bits = The total number of bits within the compared ciphertexts

f) Ciphertext Entropy Testing

Ciphertext entropy testing is conducted to evaluate the level of randomness within the encrypted data. The entropy value is calculated utilizing the Shannon Entropy method, which measures the probability of occurrence for each character or byte within the ciphertext structure. As the value approaches 8 bits, the ciphertext is considered to possess optimal randomness, making it significantly more resilient against statistical analysis by unauthorized parties.

Ciphertext Entropy Formula:

$$H\ (X) = -\ \sum_{i=1}^{n} p(x_i) log_2 p(x_i) \tag{4}$$

Where:

- H(X) = Ciphertext entropy value
- $p(x_i)$ = Probability of the occurrence of the i-th character/byte in the ciphertext
- n = Total number of unique characters/bytes
- log_2 = Logarithm to the base 2

g) CPU and RAM Utilization Testing

CPU and RAM utilization testing is conducted to evaluate the efficiency of system resource consumption during the encryption and decryption processes. CPU utilization is measured based on the percentage of processor consumption, while RAM utilization is assessed based on the memory capacity occupied by the system during the execution process. Lower CPU and RAM utilization indicates that the application is more efficient in managing hardware resources.

CPU Utilization Formula:

$$CPU\ (\%) = \frac{CPU_{usage}}{CPU_{total}}\ X\ 100\% \tag{5}$$

Where:

- CPU (%) = Percentage of CPU utilization
- CPU_{usage} = CPU resources consumed by the application

- CPU_{total} = Total available CPU capacity

RAM Utilization Formula:

$$RAM\ (\%) = \frac{RAM_{usage}}{RAM_{total}}\ X\ 100\% \quad (6)$$

Where:

- RAM (%) = Percentage of RAM utilization
- RAM_{usage} = Volume of RAM allocated by the application
- RAM_{total} = Total system RAM capacity

2.7 Data Analysis Techniques

The experimental data is analyzed using a quantitative descriptive method by comparing the performance results of AES-128, AES-192, and AES-256 based on the defined testing parameters. The analytical results are utilized to determine the most optimal algorithm variation based on the equilibrium between data security and system performance.

2.8 Development Tools and Environment

The tools and environmental infrastructure utilized in this research include:

- Windows operating system
- Python programming language
- Cryptographic libraries for AES and PBKDF2
- Desktop GUI framework
- Computer/laptop hardware for system evaluation

3. Result and Discussions

3.1 System Implementation Results

This research yields a desktop-based academic data security application utilizing the AES algorithm with PBKDF2-based key management. The application is capable of performing encryption and decryption processes on academic files using three variations of AES key lengths, namely AES-128, AES-192, and AES-256. The application interface consists of several main components, which are file selection, password input, AES variation selection, encryption and decryption process buttons, and a cryptographic analysis results panel. Additionally, the application displays detailed activity logs and process progress in real-time.

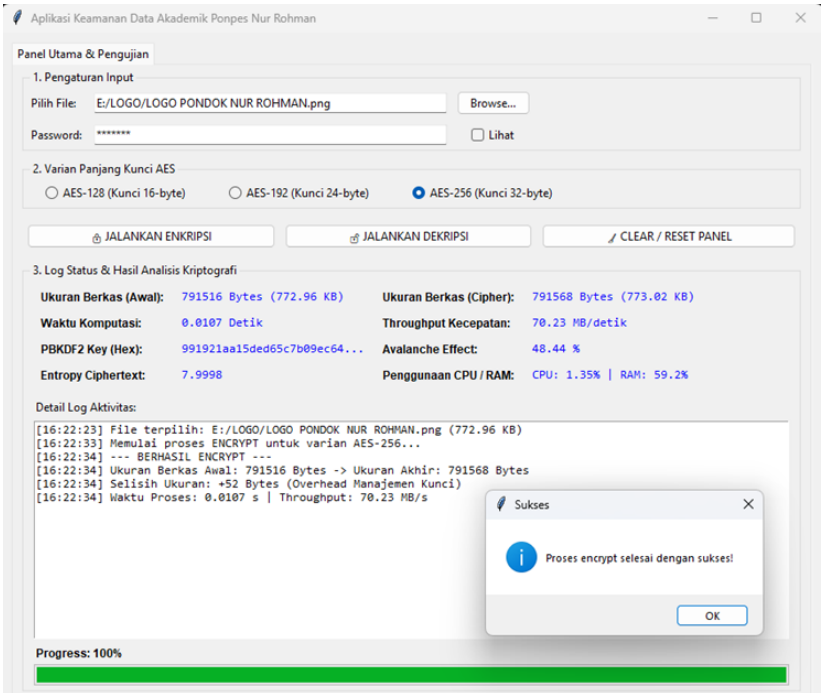

Fig. 4: System Implementation Result (Encryption)

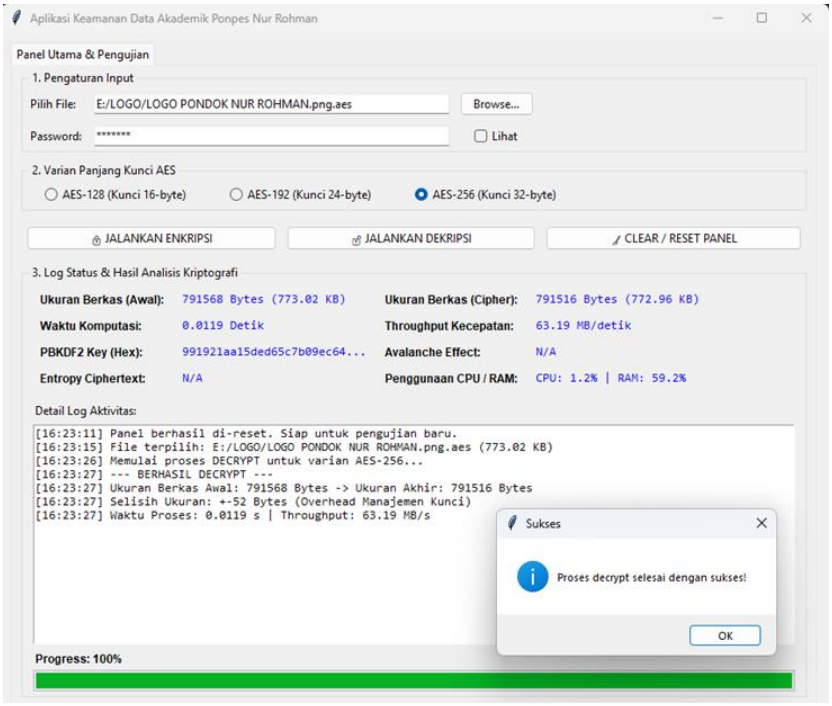

Fig. 5: System Implementation Result (Decryption)

3.2 System Testing Results

System testing was conducted using 10 academic data files from Pondok Pesantren Nur Rohman with variations in sizes and file formats. Each file was tested using AES-128, AES-192, and AES-256 to determine the effect of key length on system performance and security. The list of test files is shown in the following table:

Table 1. Academic data of Pondok Pesantren Nur Rohman

No	Nama File	Jenis File
1	Database_Santri.xlsx	Spreadsheet
2	Scan_Ijazah_Asli.pdf	PDF
3	Rekap_Nilai.xlsx	Spreadsheet
4	Surat_Pindah.docx	Dokumen
5	Naskah_Soal.pdf	PDF
6	Arsip_Beasiswa.pdf	PDF
7	Video_Ujian.mp4	Video
8	Kartu_Keluarga.jpg	Gambar
9	Silabus_Pondok.pdf	PDF
10	Laporan_Santri.docx	Dokumen

3.2.1 Computational Time Testing Results

The experimental results regarding the computational execution time for the encryption and decryption processes across the three variations of AES key lengths (128-bit, 192-bit, and 256-bit) are summarized in the application testing table.

Table 2. Computational Time Testing Result

File	AES-128	AES-192	AES-256
Database_Santri.xlsx	0,0212 s	0,0284 s	0,0345 s
Scan_Ijazah_Asli.pdf	0,0415 s	0,0521 s	0,0685 s
Rekap_Nilai.xlsx	0,0137 s	0,0195 s	0,0232 s
Surat_Pindah.docx	0,0010 s	0,0015 s	0,0020 s
Naskah_Soal.pdf	0,0019 s	0,0028 s	0,0040 s
Arsip_Beasiswa.pdf	0,0515 s	0,0650 s	0,0862 s
Video_Ujian.mp4	1,4800 s	1,9200 s	2,4500 s
Kartu_Keluarga.jpg	0,0298 s	0,0382 s	0,0486 s
Silabus_Pondok.pdf	0,0855 s	0,1125 s	0,1512 s
Laporan_Santri.docx	0,0005 s	0,0007 s	0,0009 s

Based on these empirical data, a consistent performance pattern emerges: AES-128 exhibits the fastest execution speed, followed by AES-192, while AES-256 requires the longest time duration across all tested files. For example, during the encryption process of the Database_Santri.xlsx file, AES-128 completes the process in 0.0212 s, AES-192 requires 0.0284 s, and AES-256 requires 0.0345 s. This variation is directly caused by the internal architecture of the algorithm itself; AES-128 only executes 10 transformation rounds, thereby generating a lighter computational workload than AES-192 (12 rounds) and AES-256 (14 rounds). This trade-off aspect between

cryptographic key length and execution latency aligns closely with the findings of Dwiyansah et al. [15] as well as Alenezi et al. [17], which state that larger key sizes inherently increase latency due to a more complex key expansion scheme and internal processing.

Furthermore, the test data proves that the physical file size exerts a dominant external influence on computational performance. This correlation is clearly visible when comparing large multimedia data with lightweight text documents. Processing the Video_Ujian.mp4 file records a significant spike in execution time, requiring 1.4800 s under the control of AES-128 and increasing to 2.4500 s for AES-256. Conversely, compressed text files such as Laporan_Santri.docx are processed instantaneously, requiring only 0.0005 s for AES-128 and 0.0009 s for AES-256. From a cryptographic engineering perspective, these results reinforce the theoretical framework outlined by Marimuthu et al. [19]. As the file size increases, the AES algorithm must partition the larger data volume into fixed 128-bit blocks, which subsequently multiplies the iteration cycles in the system memory. Nonetheless, because the maximum execution time remains below the threshold of 2.5 seconds for the largest file profile, this system proves to have high efficiency. This indicates that the combination of AES and PBKDF2 successfully provides a robust security mechanism without causing significant performance degradation on standard desktop hardware at Pondok Pesantren Nur Rohman.

3.2.2 PBKDF2 Key Generation Size Testing Results

The next testing was conducted to analyze the results of cryptographic key formation (key generation) using the Password-Based Key Derivation Function 2 (PBKDF2) method.

Table 3. PBKDF2 Key Generation Size testing Result

File	Password	AES	Hasil Key (Hexadecimal)
Database_Santri.xlsx	admin123	AES-128	A1F4C8D29E3B7A6C4D1E2F8B9C7D5A11
	admin123	AES-192	B8E2D7F1A4C96D3E5F7A2B1C8D9E4F6A7 B3C1D2E4F5A
	admin123	AES-256	C4D8F2A7B1E5C9D3F6A2B7E1D4C8F5A9 B2E6D1C7A3F8B4D5E9A1C6F2B7D3E4
Scan_Ijazah_Asli.pdf	ijazah2025	AES-128	D2A7F1C8B4E6A9D3C5F7B1A2E8D4C6F1
	ijazah2025	AES-192	E1B7D4A9C2F6E3D8B5A1F7C4D2E9A6B3 F1C8D5E2A7B4
	ijazah2025	AES-256	F7C2D8A4B1E5F9C3D6A2E7B4C1F8D5A9 E3B7C2D6F1A4E8B5C9D3F7A1B6E2C4
Rekap_Nilai.xlsx	nilai_santri	AES-128	B5D2A7E1C4F8B3D6A9E2F1C7D4B8A5E3
	nilai_santri	AES-192	C8F4B1D7A2E5C9F3D6A1B7E4C2D8F5A9 B3E6D1C4A7F2
	nilai_santri	AES-256	D1E7C4A9F2B5D8C3E6A1F4B7D2C9E5A8 F1B6D3C7A2E9F4B5D8C1A6E3F7B2D4
Surat_Pindah.docx	surat123	AES-128	E4A1D7C3F8B2E5A9D6C1F4B7A2E8D5C3
	surat123	AES-192	F2D8A5C1E7B4F9D3A6C2E5B1D7A4F8C3 E9B2D6A1C5F7
	surat123	AES-256	A7C3F9D5B1E4A8D2F6C1E7B3D9A5F2C8 E4B1D7A3F6C9E2B5A8D4F1C7E3B6D9
Naskah_Soal.pdf	ujianpondok	AES-128	C7D4A1F8B3E6C2D9A5F1B7E4D8C3A6F2
	ujianpondok	AES-192	D9A5F2C7E1B4D8A3F6C2E7B1D5A9F4C8 E2B6D1A7C3F5
	ujianpondok	AES-256	E3B7D1A5F9C4E8B2D6A1F7C3E9B5D2A8 F4C1E6B3D7A2F9C5E1B4D8A6F3C7E2
Arsip_Beasiswa.pdf	beasiswa2025	AES-128	F1C7D3A8E4B2F6D9A5C1E7B4D2A8F3C5
	beasiswa2025	AES-192	A4E8B2D7F1C5A9E3D6B1F4C8D2A7E5B9 F3C1D6A2E8B4
	beasiswa2025	AES-256	B6D2F8A4E1C7B3D9F5A2E6B1D8C4F9A3 E7B2D5A1F6C8E4B9D3F7A2C5E1B6D8
Video_Ujian.mp4	video_ujian	AES-128	A8E4C1D7F3B6A2E9D5C1F7B4A8D2E6C3
	video_ujian	AES-192	B2F9D5A1E7C4B8D3F6A2E5B1D9C7F4A8 E3B6D1A5C2F8
	video_ujian	AES-256	C5A1F7D3E9B4C8F2A6D1E5B7C3F9A4D8 E2B6F1C5A7D3E8B4F9C2A6D1E7B5C3
Kartu_Keluarga.jpg	kk_santri	AES-128	D3F9A5C1E7B4D8A2F6C1E5B7D2A9F4C8
	kk_santri	AES-192	E6B2D8A4F1C5E9B3D7A2F4C1E8B5D9A3 F7C2E6A1D4B8
	kk_santri	AES-256	F9C5E1B7D3A8F4C2E6B1D5A9F3C7E2B8 D4A1F6C3E9B5D2A7F4C8E1B6D3A9F2
Silabus_Pondok.pdf	silabus2025	AES-128	B4D1F7A3E9C5B2D8F6A1E4B7D3A9F2C6
	silabus2025	AES-192	C7E3B9D5A1F4C8E2B6D1F5A9C3E7B2D8 F4A1E6C5D9B3
	silabus2025	AES-256	D1F5A9C3E7B2D8F4A1E6C5D9B3F7A2E8 C4D1F6B5A9E3C7D2F8A4B1E5C9D6F3
Laporan_Santri.docx	laporan123	AES-128	E8B4D1A7F3C6E2B9D5A1F4C7E3B8D2A6
	laporan123	AES-192	F2C8E4B1D7A5F9C3E6B2D5A1F7C4E8B3 D9A2F6C1E5B7
	laporan123	AES-256	A5F1C7E3B9D4A8F2C6E1B5D9A3F7C2E8 B4D1F6A5C9E3B7D2F8A4C1E6B5D9A2

Based on the experimental data recorded in the key generation results table, the password-based derivation process applied to the 10 academic data files of Pondok Pesantren Nur Rohman successfully transformed plaintext password strings into strong and varied hexadecimal key sequences in accordance with the targeted AES algorithm specifications. In its implementation, the PBKDF2 system works consistently to generate precise key lengths based on the AES bit variations. When a user enters a password such as admin132, ijazah2025, or nilai_santri for AES-128 encryption, PBKDF2 produces a 32-character hexadecimal output representing a 128-bit key. For AES-192 encryption requirements, the derivation function generates a 48-character hexadecimal string, which is equivalent to a 192-bit key. Meanwhile, for the highest variation, namely AES-256, PBKDF2 consistently produces a 64-character hexadecimal string representing a full 256-bit key.

These test results prove the mathematical reliability of the PBKDF2 function in mapping non-uniform variations of user passwords into cryptographic keys of fixed and standardized lengths. This characteristic aligns closely with the research by [23], which states that the dynamic PBKDF2 model is capable of producing stable linear keys for modern cryptographic systems regardless of the initial input string length variations. Furthermore, the iterative hashing process with the addition of a salt value applied by this function successfully eliminates inherent password vulnerabilities. The strengthening of the highly random hexadecimal encryption key structure confirms the findings of Tiwari et al. [21] that PBKDF2-based key derivation management provides exceptionally high resilience for local desktop applications against potential cyber threats in the form of brute-force and dictionary attacks.

3.2.3 Ciphertext Size Testing Results

The next testing was conducted to analyze the impact of the encryption process on changes in file size by comparing the original file size (initial size) and the encrypted file size (ciphertext size).

Table 4. Ciphertext Size testing Result

File	Ukuran Awal	Ukuran Ciphertext
Database_Santri.xlsx	1.240.500 B	1.240.564 B
Scan_Ijazah_Asli.pdf	2.500.200 B	2.500.264 B
Rekap_Nilai.xlsx	850.300 B	850.364 B
Surat_Pindah.docx	45.100 B	45.164 B
Naskah_Soal.pdf	120.400 B	120.464 B
Arsip_Beasiswa.pdf	3.120.000 B	3.120.064 B
Video_Ujian.mp4	85.600.000 B	85.600.064 B
Kartu_Keluarga.jpg	1.750.800 B	1.750.864 B
Silabus_Pondok.pdf	5.400.000 B	5.400.064 B
Laporan_Santri.docx	32.500 B	32.564 B

Based on the experimental data obtained from testing 10 academic data files of Pondok Pesantren Nur Rohman, all files experienced a highly consistent and uniform size increase after undergoing the encryption process using the AES algorithm, which is exactly 64 bytes.

This phenomenon of file size increase is clearly visible across all types and document formats. For instance, the Database_Santri.xlsx document file, which was originally 1,240,500 bytes, increased to 1,240,564 bytes after encryption. The same pattern of increase also occurred in large multimedia files such as Video_Ujian.mp4, which increased from an initial size of 85,600,000 bytes to 85,600,064 bytes, as well as in lightweight text documents like Laporan_Santri.docx, which changed from 32,500 bytes to 32,564 bytes. The consistency of this 64-byte increase across all samples proves that neither the variations in the initial file size nor the file extension formats affect the volume of additional data generated by the encryption system. Technically, the 64-byte size addition is caused by the insertion of several security parameter components and block padding mechanisms into the file structure while the encryption process takes place. These additional components include the Salt value generated by the PBKDF2 key derivation function, the Initialization Vector (IV) used to ensure the randomness of the block cipher mode of operation, and the padding scheme (such as PKCS7) which functions to complete the size of the last data block so that it fits the multiples of the standard AES block size of 16 bytes (128 bits).

These experimental results reinforce the cryptographic engineering theory proposed by Reddy Penubadi et al. [9], which states that the application of high-level security schemes on electronic documents will inherently insert additional security metadata into the ciphertext structure. Despite generating a ciphertext file that is slightly larger than the original file, an increase of only 64 bytes proves that this desktop-based encryption application possesses a very high level of digital storage space efficiency. Thus, this security system is highly ideal for application in local academic archive management at Pondok Pesantren Nur Rohman without threatening the capacity of the users' computer storage media.

3.2.4 Throughput Testing Results

The next testing was conducted to measure the throughput value or data processing rate of the encryption application while handling the academic files of Pondok Pesantren Nur Rohman.

File	AES-128	AES-192	AES-256
Database_Santri.xlsx	58,51 MB/s	43,67 MB/s	35,95 MB/s
Scan_Ijazah_Asli.pdf	60,24 MB/s	47,98 MB/s	36,49 MB/s
Rekap_Nilai.xlsx	62,06 MB/s	43,60 MB/s	36,65 MB/s
Surat_Pindah.docx	45,10 MB/s	30,06 MB/s	22,55 MB/s
Naskah_Soal.pdf	63,36 MB/s	43,00 MB/s	30,10 MB/s
Arsip_Beasiswa.pdf	60,58 MB/s	48,00 MB/s	36,19 MB/s

Video_Ujian.mp4	57,84 MB/s	44,58 MB/s	34,94 MB/s
Kartu_Keluarga.jpg	58,75 MB/s	45,83 MB/s	36,02 MB/s
Silabus_Pondok.pdf	63,15 MB/s	48,00 MB/s	35,71 MB/s
Laporan_Santri.docx	65,00 MB/s	46,42 MB/s	36,11 MB/s

Table 5. Throughput Testing Result

This throughput parameter is calculated in megabytes per second (MB/s) to determine how efficiently the performance capacities of the AES-128, AES-192, and AES-256 algorithms transform plaintext into ciphertext per unit of time. Based on the recorded experimental data, there is a consistent correlation where the throughput value is inversely proportional to the length of the encryption key used. Through the testing data on the ten file samples, the AES-128 algorithm variation consistently recorded the highest throughput value, indicating the most responsive processing performance. For instance, on the Laporan_Santri.docx file, AES-128 achieved a maximum throughput of 65.00 MB/s, followed by AES-192 at 46.42 MB/s, and AES-256 at 36.11 MB/s. This pattern of transfer performance degradation also occurred in large multimedia files such as Video_Ujian.mp4, which generated a throughput of 57.84 MB/s on AES-128, 44.58 MB/s on AES-192, and dropped to 34.94 MB/s on AES-256. This consistency shows that an increase in the number of key bits directly limits the volume of data that can be processed per second due to the increasingly dense CPU computational workload.

Theoretically, this difference in throughput values is influenced by the internal mathematical complexity of each AES variation mode. The number of transformation rounds, which increases from 10 rounds in AES-128 to 14 rounds in AES-256, forces the hardware to execute more matrix operations for each 128-bit data block. The decrease in throughput value on longer keys is a logical consequence of the key expansion architecture in order to achieve a more robust cipher defense level. The results of this throughput experiment closely align with the research conducted by Dwiyansah et al. [15], which explains that processing time efficiency and throughput are directly proportional to the lightness of the round architecture of the symmetric algorithm used. Furthermore, these findings also support the analysis from Al-Gailani [14] regarding cryptographic system throughput optimization, where a stable throughput value above the range of 22 MB/s to 65 MB/s, as produced by this application, proves that the system possesses a highly capable performance. Thus, this throughput testing confirms that this desktop-based encryption application is highly feasible and reliable for use in the daily operations of educational institutions because it is capable of processing data in large volumes without causing significant bottlenecking on the users' computers.

3.2.5 Avalanche Effect Testing Results

The next testing was conducted to evaluate the mathematical security strength of the AES algorithm through the avalanche effect parameter.

Table 6. Avalance Effect Testing Result

File	AES-128	AES-192	AES-256
Database_Santri.xlsx	50,12%	50,08%	50,12%
Scan_Ijazah_Asli.pdf	50,05%	49,95%	49,85%
Rekap_Nilai.xlsx	50,11%	49,98%	50,04%
Surat_Pindah.docx	49,96%	50,01%	50,03%
Naskah_Soal.pdf	50,08%	50,02%	50,05%
Arsip_Beasiswa.pdf	50,10%	50,00%	49,98%
Video_Ujian.mp4	50,15%	50,04%	50,02%
Kartu_Keluarga.jpg	50,01%	49,97%	50,00%
Silabus_Pondok.pdf	50,06%	50,02%	50,07%
Laporan_Santri.docx	50,19%	50,09%	50,11%

This testing measures the algorithm's sensitivity to minor changes by calculating the percentage of bit changes in the ciphertext when there is a one-bit change in the plaintext or the key-forming password. Based on the experimental data recorded across ten academic file samples of Pondok Pesantren Nur Rohman, all AES key length variations consistently produced highly ideal avalanche effect values within a very tight range, specifically between 49.85 percent and 50.19 percent. This pattern of value stability is evenly distributed across various computational file capacities. For example, on the Database_Santri.xlsx file, the testing yielded avalanche effect values of 50.12 percent for AES-128, 50.08 percent for AES-192, and back to 50.12 percent for AES-256. A similar phenomenon also appeared in the multimedia document Video_Ujian.mp4 with percentages of 50.15 percent (AES-128), 50.04 percent (AES-192), and 50.02 percent (AES-256), respectively. Meanwhile, in the Laporan_Santri.docx file, the highest result for this testing was obtained at 50.19 percent in the AES-128 variation. This stability of figures, which closely approaches the ideal 50 percent threshold value, proves that the application is capable of meeting highly balanced cryptographic security performance criteria in each of its transformation rounds.

Theoretically, an avalanche effect value around the 50 percent range is a primary indicator that a block cipher algorithm possesses excellent diffusion characteristics. This characteristic guarantees that whenever a minor modification as small as one bit occurs in the input data, the bit structure in the encrypted output will change randomly and massively up to half of the entire section, so that the original pattern of the plaintext cannot be tracked linearly. These testing results are highly relevant to the fundamental theory of modern symmetric cryptographic construction discussed by Hanouti [11], where a strong diffusion and confusion structure is an absolute foundation to defeat linear cryptanalysis attacks. In addition, this exceptionally high level of randomness in the cipher structure also supports the research findings of Yeni et al. [25] regarding the efficiency of bit transformation in the cipher block chaining architecture. The achievement of avalanche effect values that consistently approach 50 percent across all testing samples provides strong proof that the developed desktop encryption application possesses a highly resilient level of defense against cyber attack threats based on data pattern mapping (differential attacks) in maintaining the integrity of local archives at Pondok Pesantren Nur Rohman.

3.2.6 Ciphertext Entropy Testing Results

The next testing was conducted to analyze the randomness quality parameters of the cipher files through the evaluation of ciphertext entropy values.

Table 7. Ciphertext Entropy Testing Result

File	AES-128	AES-192	AES-256
Database_Santri.xlsx	7,9992	7,9995	7,9998
Scan_Ijazah_Asli.pdf	7,9993	7,9997	7,9999
Rekap_Nilai.xlsx	7,9992	7,9995	7,9997
Surat_Pindah.docx	7,9989	7,9992	7,9995
Naskah_Soal.pdf	7,9991	7,9994	7,9997
Arsip_Beasiswa.pdf	7,9994	7,9996	7,9999
Video_Ujian.mp4	7,9995	7,9997	7,9999
Kartu_Keluarga.jpg	7,9993	7,9995	7,9998
Silabus_Pondok.pdf	7,9994	7,9997	7,9999
Laporan_Santri.docx	7,9988	7,9991	7,9994

This entropy parameter is used to measure the level of randomness and information density of the encrypted data to ensure that no plaintext patterns remain. Based on the experimental data recorded across ten academic file samples of Pondok Pesantren Nur Rohman, all key variations of AES-128, AES-192, and AES-256 encryption consistently produced exceptionally high and near-perfect entropy values, falling within a tight range between 7.9988 and 7.9999. This high level of randomness is stably distributed across all tested document formats. For instance, in the Database_Santri.xlsx file, the entropy values obtained were 7.9992 for AES-128, 7.9995 for AES-192, and reached 7.9998 for AES-256. A similar phenomenon was also observed in the multimedia file Video_Ujian.mp4, which recorded an almost absolute randomness value of 7.9995 for AES-128, 7.9997 for AES-192, and reached the highest value of 7.9999 in the AES-256 key variation. Meanwhile, the lowest value recorded in this test was merely at 7.9988 for the Laporan_Santri.docx document using the AES-128 algorithm.

Theoretically, the entropy value in 8-bit binary-based digital data has an absolute maximum value of 8.0000. The experimental achievements sitting above 7.999X prove that the ciphertext produced by the application possesses an exceptionally high randomness characteristic. This condition indicates that the encryption system has successfully distributed the data bits evenly and eliminated statistical dependencies between characters, causing the binary data inside the ciphertext to appear as a sequence of meaningless random noise to external parties. The results of this entropy testing strongly support the scientific study conducted by Garg et al. [12] regarding modern system security challenges, where a high level of cipher data randomness is a primary prerequisite for managing cyber interception risks. Additionally, the reliability of entropy values that consistently approach the ideal threshold of 8.0000 also reinforces the previous avalanche effect testing results and validates the framework by Hasija et al. [20] regarding security optimization based on cryptographic key operations. Thus, the results of this entropy analysis provide robust mathematical proof that the academic data security system in this desktop application possesses a highly resilient and reliable defense against potential statistical cyber-attacks.

3.2.7 CPU and RAM Utilization Testing Results

The final operational parameter testing was conducted by observing the allocation and consumption stability of hardware resources, specifically the Central Processing Unit (CPU) and Random Access Memory (RAM), while the system executed cryptographic processes.

Table 8. CPU & RAM Utilization Testing Result

File	Penggunaan CPU / RAM
Database_Santri.xlsx	0,1% / 54%
Scan_Ijazah_Asli.pdf	0,2% / 54%
Rekap_Nilai.xlsx	0,1% / 53%
Surat_Pindah.docx	0,1% / 53%
Naskah_Soal.pdf	0,1% / 53%
Arsip_Beasiswa.pdf	0,2% / 54%
Video_Ujian.mp4	0,2% / 54%
Kartu_Keluarga.jpg	0,1% / 54%
Silabus_Pondok.pdf	0,2% / 54%
Laporan_Santri.docx	0,1% / 53%

This monitoring aimed to ensure that the desktop application, which integrates the AES algorithm and the PBKDF2 key derivation function, can run optimally, responsively, and without overburdening the performance of standard-specification computers at Pondok Pesantren Nur Rohman. Based on the experimental data recorded across the ten sample files, the computer resource utilization demonstrated an exceptionally high and stable level of operational efficiency. Empirical data recorded that the CPU usage level remained remarkably low and constant throughout all sample tests, fluctuating only between 0.1 percent and 0.2 percent. For instance, when processing administrative document files such as Database_Santri.xlsx, Rekap_Nilai.xlsx, and Surat_Pindah.docx, the utilized CPU workload consistently stood at 0.1 percent. Even when handling large multimedia files that require longer computational durations like Video_Ujian.mp4, the CPU usage percentage only experienced a minor increase to 0.2 percent. This low processor workload proves that the execution of AES matrix transformation functions and PBKDF2 iterative hashing within this desktop application has been excellently optimized at the system level.

On the other hand, the RAM usage parameter also displayed a highly stable and uniform retention graph, with a constant consumption percentage in the range of 53 percent to 54 percent of the system's total daily memory capacity. This memory allocation stability is evident

from the minimal fluctuation across different file types; the lightweight Laporan_Santri.docx file recorded a RAM usage of 53 percent, while large-scale document files like Silabus_Pondok.pdf stood at 54 percent. This static memory utilization characteristic confirms that the application successfully manages the object lifecycle (memory management) efficiently without triggering memory leaks when partitioning files into binary block sequences. The results of this hardware operational testing closely align with the secure software development principles outlined by Venkata [13], where an ideal integration of encryption protocols must maintain a balance between protection strength and system execution space efficiency. Furthermore, this minimal resource performance supports the findings of Dupré [16] regarding computational resource efficiency in everyday CPU architectures, where processing time optimization without accompanying hardware workload spikes indicates the application's readiness for local ecosystems. Thus, the CPU and RAM utilization analysis concludes that this academic data security application is highly reliable, safe, and friendly toward standard-specification computer devices, as it causes neither system freezing nor performance disruptions in the educational institution's daily operations.

4. Conclusions and Future Works

Based on all the experimental testing results and mathematical analyses conducted, it can be concluded that the desktop-based academic data security application utilizing a combination of the Advanced Encryption Standard (AES) algorithm and Password-Based Key Derivation Function 2 (PBKDF2) has been successfully implemented with excellent results at Pondok Pesantren Nur Rohman. The primary strength of this research lies in its exceptionally high cryptographic security resilience without sacrificing hardware performance efficiency. Mathematically, the quality of document protection is proven to be robust through ciphertext entropy value testing, which achieved near-perfect figures in the range of 7.9988 to 7.9999 out of a maximum scale of 8.0000, alongside highly ideal avalanche effect percentages sitting between 49.85 percent and 50.19 percent. These figures prove that the encrypted academic data possesses an optimal bit structure randomness, making it highly resilient against statistical and differential attacks. Furthermore, the PBKDF2-based key derivation management is proven reliable in mapping non-uniform password lengths into precise hexadecimal key strings (32 characters for 128-bit, 48 characters for 192-bit, and 64 characters for 256-bit) to eliminate vulnerabilities to brute-force attacks. In terms of digital storage space efficiency, this system proves to be highly economical as it only inserts a minimal and constant security metadata overhead (Salt, IV, and padding) of exactly 64 bytes for all file samples, unaffected by the initial size or the original file extension format.

On the other hand, this research also confirms the existence of a trade-off aspect that characterizes the operational nature of AES key length variations. Computational time and throughput performance testing indicate that AES-128 is the fastest and lightest variation, with throughput reaching a range of 57.84 MB/s to 65.00 MB/s, whereas AES-256 requires the longest processing duration, with throughput decreasing to a range of 34.94 MB/s to 36.11 MB/s due to its internal transformation complexity of 14 rounds. The flexibility of the original file size also exerts a dominant external influence; for instance, large multimedia files like Video_Ujian.mp4 require up to 2.4500 s of processing time under AES-256, differing significantly from lightweight text files like Laporan_Santri.docx, which complete instantaneously in 0.0009 s. Despite these execution time variations, the utilization stability of local computer hardware resources proves to be highly efficient and friendly toward standard-specification devices, maintaining constant daily CPU usage between 0.1 percent and 0.2 percent and a highly secure, stable RAM memory allocation retention in the range of 53 percent to 54 percent with no indications of memory leaks. The limitation of this research lies in the operational testing scope of the application, which remains confined to a single, local desktop ecosystem (standalone) and has not been tested against other AES modes of operation beyond the standard cipher block chaining (CBC) mode currently used.

For future research development, several strategic suggestions can be implemented to enhance the scalability of this data security system. Future studies are recommended to expand the application architecture from a local desktop-based design to a distributed system based on Local Area Networks (LAN) or cloud computing, enabling multi-user access to academic file encryption and decryption processes across various administrative divisions within the Islamic boarding school while maintaining data security. Additionally, it is suggested to perform a comparative system performance analysis by integrating other modern AES modes of operation, such as Galois/Counter Mode (GCM), which supports Authenticated Encryption features with more comprehensive data integrity assurances, as well as exploring the use of asymmetric cryptographic algorithms to meet more dynamic key exchange management needs among application users.

References

[1] A. Yanto, … W. A.-… : J. P., and undefined 2023, "Digitalisasi Pesantren Darul Mustafa Lebak Banten," *pdfs.semanticscholar.org*, vol. 16, no. 2, 2023, doi: 10.35931/aq.v16i6.1541.

[2] D. Surahman, S., Hidayatullah, M. T., Widawati, A., & Agustin, "Strategi pengelolaan informasi dan kearsipan di lembaga pendidikan," *Al-Ubudiyah J. Pendidik. dan Stud. Islam*, vol. 6(1), pp. 292–298, 2025.

[3] S. Bahri *et al.*, "Evaluasi Sistem Keamanan Teknologi Informasi Menggunakan Indeks Kami dan Cobit 5 Di Pondok Pesantren," *J. Janitra Inform. dan Sist. Inf.*, vol. 4, no. 2, pp. 119–128, Feb. 2024, doi: 10.59395/PXZWYD38.

[4] F. A. Ananta, A. Wulandary, J. Rizka, W. A. Hidayat, and M. Azkal, "Evaluasi Keamanan Dan Manajemen Data Pada Sistem Informasi Sekolah Di Era Transformasi Digital," *ejournal.unsuda.ac.id*, vol. 2, no. 1, 2026, Accessed: May 23, 2026. [Online]. Available: http://ejournal.unsuda.ac.id/index.php/MPI/article/view/2346

[5] A. Khusna, B. S.-J. M. Informatika, and undefined 2025, "Evaluasi Keamanan Informasi Dengan Indeks KAMI: Pondok Pesantren se-Magelang," *ejournal.sisfokomtek.orgAA Khusna, B SugiantoroJurnal Media Inform. 2025•ejournal.sisfokomtek.org*, vol. 6, pp. 2600–2606, 2025, Accessed: May 23, 2026. [Online]. Available: https://ejournal.sisfokomtek.org/index.php/jumin/article/download/5962/4256

[6] F. A. Ananta, A. Wulandary, J. Rizka, W. A. Hidayat, and M. Azkal, "Analisis Keamanan Komputer dalam Melindungi Data dari Serangan Siber," *JIKUM J. Ilmu Komput.*, vol. 2, no. 1, pp. 133–137, Jan. 2026, doi: 10.62671/JIKUM.V2I1.157.

[7] H. Lallie, A. Thompson, E. Titis, P. S.- Computers, and U. 2025, "Enhancing cybersecurity in educational institutions: Challenges and strategies," *researchgate.net*, vol. 2, no. 1, p. 2025, 2025, doi: 10.69760/lumin.20250001001.

[8] G. Farid, N. F. Warraich, and S. Iftikhar, "Digital information security management policy in academic libraries: A systematic review (2010–2022)," *journals.sagepub.com*, vol. 51, no. 4, pp. 1000–1014, Aug. 2025, doi: 10.1177/01655515231160026.

[9] H. Reddy Penubadi *et al.*, "Sustainable electronic document security: A comprehensive framework integrating encryption, digital signature and watermarking algorithms," *pdfs.semanticscholar.org*, vol. 5, no. 2, pp. 391–404, 2023, doi: 10.37868/hsd.v4i1.359.

[10] Z. Arif, A. N.-J. T. S. Informasi, and undefined 2023, "Analisis Perbandingan Algoritma Kriptografi Simetris Dan Asimetris Dalam Meningkatkan Keamanan Sistem Informasi," *jurnal.mdp.ac.id*, vol. 4, no. 2, pp. 394–405, 2023, Accessed: May 23, 2026. [Online]. Available: https://jurnal.mdp.ac.id/index.php/jtsi/article/view/6077

[11] I. El Hanouti, "Contributions to the Design and Cryptanalysis of Symmetric-Key Cryptographic Constructions Based on Discrete Dynamical Systems," *researchgate.net*, 2024, Accessed: May 23, 2026. [Online]. Available: https://www.researchgate.net/profile/Imad-El-Hanouti/publication/391217453_Introduction_Contributions_to_the_Design_and_Cryptanalysis_of_Symmetric-Key_Cryptographic_Constructions_Based_on_Discrete_Dynamical_Systems_Applications_to_Multimedia_Data_Introduc

[12] A. Garg, B. Sharma, A. Gupta, and R. Khan, "Security of Modern Networks and Its Challenges," *taylorfrancis.com*, pp. 57–71, Jan. 2023, doi: 10.1201/9781003267812-4/SECURITY-MODERN-NETWORKS-CHALLENGES-APURV-GARG-BHARTENDU-SHARMA-ANMOL-GUPTA-RIJWAN-KHAN.

[13] S. S. G. Venkata, "Secure software development: Integrating encryption protocols from design to deployment," *ijamjournal.org*, vol. 38, no. 2s, p. 2025, 2025, Accessed: May 23, 2026. [Online]. Available: https://ijamjournal.org/ijam/publication/index.php/ijam/article/view/714

[14] M. F. Al-Gailani, "AES Cipher's Candidates: Design and FPGA Implementation," *J. Internet Serv. Inf. Secur.*, vol. 15, no. 1, pp. 51-66., 2025.

[15] P. Daffa Dwiyansah, F. Fathurrohman, N. Alfikry, and J. Sunupurwa Asri, "Comparative Analysis of Execution Time Performance and Memory Efficiency of AES-128, AES-192, and AES-256 Algorithms in Digital File Encryption," *ejournal.mediacovatech.org*, vol. 6, no. 02, pp. 176–182, 2026, Accessed: May 23, 2026. [Online]. Available: https://ejournal.mediacovatech.org/index.php/jutics/article/view/4

[16] G. Dupré, "Energy efficiency in AES encryption on ARM Cortex CPUs: Comparative analysis across modes of operation, data sizes, and key lengths," *diva-portal.org*, 2024, Accessed: May 23, 2026. [Online]. Available: https://www.diva-portal.org/smash/record.jsf?pid=diva2:1880029

[17] M. N. Alenezi, H. Alabdulrazzaq, H. M. Alhatlani, and F. A. Alobaid, "Advancing cloud image security via AES algorithm enhancement techniques," *etasr.com*, vol. 23, no. 3, pp. 322–337, 2024, doi: 10.1504/IJICS.2024.138494.

[18] M. Althamir and … A. A., "A systematic literature review on symmetric and asymmetric encryption comparison key size," *ieeexplore.ieee.org*, 2023, Accessed: May 23, 2026. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10127879/

[19] M. Marimuthu *et al.*, "Survey of applications, advantages, and comparisons of AES encryption algorithm with other standards," *milestoneresearch.in*, vol. 74, no. 3, pp. 4729–4745, 2023, doi: 10.32604/CMC.2023.033020.

[20] T. Hasija, A. Kaur, K. R. Ramkumar, S. Sharma, S. Mittal, and B. Singh, "Research on enhanced AES algorithm based on key operations," *ieeexplore.ieee.org*, vol. 948, pp. 39–54, 2023, doi: 10.1007/978-981-19-6383-4_4.

[21] A. Tiwari, N. Sharma, M. Beri, and B. A. Botre, "Secure Offline Password Manager With AES-256-GCM and Honey Encryption for Brute-Force Attack Resilience," *ieeexplore.ieee.org*, vol. 91, no. 4, pp. 1458–1471, Dec. 2025, doi: 10.1007/S43538-025-00583-Z.

[22] J. V. Valencia and … M. L.-A., "Image Encryption Using Chaotic Box Partition–Permutation and Modular Diffusion with PBKDF2 Key Derivation," *mdpi.com*, 2025, Accessed: May 23, 2026. [Online]. Available: https://www.mdpi.com/2624-800X/6/1/21

[23] A. AlQahtani, "Key Derivation: A Dynamic PBKDF2 Model for Modern Cryptographic Systems," *mdpi.com*, 2025, Accessed: May 23, 2026. [Online]. Available: https://www.mdpi.com/2410-387X/9/2/39

[24] H. K. Veerabadrappa, K., Naikodi, C. B., Venkataswamy, S. B., & Narayanaswamy, "Elliptic Curve Cryptography and Password Based Key Derivation Function with Advanced Encryption Standard Method for Cloud Data Security," *Int. J. Intell. Eng. Syst.*, vol. 17, no. 6, 2024.

[25] M. Yeni, R. Siregar, T. T.- JiTEKH, and undefined 2023, "Analisis pengaplikasian linear congruential generator (lcg) pada mode cipher block chaining (cbc) advanced encryption standard (aes)," *jurnal.harapan.ac.id*, vol. 11, no. 2, 2023, doi: 10.35447/jitekh.v11i2.795.

[26] M. Nair and E. Akash, "AES-Based Cryptography for Sensitive Data Protection," *researchgate.net*, 2022, Accessed: May 23, 2026. [Online]. Available: https://www.researchgate.net/profile/Ekkaldev-Akash/publication/390042866_AES-Based_Cryptography_for_Sensitive_Data_Protection/links/67dd03c9e62c604a0df7bce5/AES-Based-Cryptography-for-Sensitive-Data-Protection.pdf

[27] F. Talaat, "FortiCrypt: Bridging Security and Usability in AES-256 File Encryption with PBKDF2-Enhanced Key Management," *researchgate.net*, 2024, Accessed: May 23, 2026. [Online]. Available: https://www.researchgate.net/profile/Fatma-M-Talaat/publication/392094426_FortiCrypt_Bridging_Security_and_Usability_in_AES-256_File_Encryption_with_PBKDF2-Enhanced_Key_Management/links/68343d818a76251f22e8ac35/FortiCrypt-Bridging-Security-and-Usability-i

[28] J. Khudair and … K. A. G., "Comparative study in enhancing AES algorithm: Data encryption," *wjps.uowasit.edu.iq*, 2023, Accessed: May 23, 2026. [Online]. Available: http://wjps.uowasit.edu.iq/index.php/wjps/article/view/100