IMPLEMENTATION OF A ONE-TIME TOKEN FOR PDF DOCUMENT SECURITY BASED ON AES-128 ENCRYPTION

Desyderius Minggu, S.T.,M.T.,[1],Hermansyah[2] dan Asep Suryanta , S.T.,[3]

[1] Program Studi Telekomunikasi, Politeknik Angkatan Darat, Malang, Jawa Timur, Indonesia
[2] Program Studi Telekomunikasi, Politeknik Angkatan Darat, Malang, Jawa Timur, Indonesia
[3] Program Studi Telekomunikasi, Politeknik Angkatan Darat, Malang, Jawa Timur, Indonesia

E - mail : desydariusminggu@gmail.com,hermansyah231096@gmail.com[2] dan zenilybaz@gmail.com

IMPLEMENTATION OF A ONE-TIME TOKEN FOR PDF DOCUMENT SECURITY BASED ON AES-128 ENCRYPTION

Abstract: *Digital document security has become a crucial aspect in preventing unauthorized access to sensitive information. This study proposes and implements a hybrid security model for PDF documents that integrates AES-128 encryption with a One-Time Token (OTT) mechanism as a dynamic access key operating in a local environment. The research method employed is software engineering using a prototyping approach combined with quantitative experimental testing. The evaluation results indicate that the system consistently achieves an access denial rate exceeding 96.7% across all unauthorized access scenarios, including replay attacks and expired token usage. In terms of performance, the system demonstrates high efficiency, with an average encryption time of 2.61 seconds for a 10 MB file. Although performance limitations were observed for extremely large files (~1 GB), this study concludes that the proposed model is an effective, reliable, and practical solution for enhancing the security of digital document distribution without relying on cloud infrastructure.*

Keywords: *AES-128 Encryption, Document Access, Document Security, Digital Security, One-Time Token, Single-Use Token.*

Abstrak: *Digital document security has become a crucial aspect in preventing unauthorized access to sensitive information. This research proposes and implements a hybrid security model for PDF documents that integrates AES-128 encryption with a One-Time Token (OTT) mechanism as a dynamic access key operating in a local environment. The research method employed is software engineering with a prototyping approach and quantitative experimental testing. The results demonstrate that the system consistently achieves an effectiveness rate of over 96.7% in rejecting all unauthorized access scenarios, including replay attacks and the use of expired tokens. In terms of performance, the system exhibits high efficiency with an average encryption time of 2.61 seconds for a 10 MB file. Although certain performance limitations were identified for very large files (~1 GB), this study concludes that the proposed model is an effective, reliable, and practical solution for enhancing the security of digital document distribution without relying on cloud infrastructure.*

Keywords*: AES-128 Encryption, Document Access, Document Security, Digital Security, One-Time Token, Single-Use Token.*

PENDAHULUAN

In recent decades, digital transformation has fundamentally shifted the way humans store, manage, and disseminate information. The presence of digital documents has gradually replaced physical formats and has now become the primary standard across various sectors, including government, military, education, and public services (Asyura Binti Sofian et al., 2024). However, the massive adoption of electronic documents has introduced new challenges related to information security. Issues concerning data confidentiality, information integrity, and access control mechanisms are crucial factors that must be safeguarded to prevent data breaches or unauthorized access that could potentially harm multiple stakeholders (Asyura Binti Sofian et al., 2024).

The increasing threats to digital assets have driven the development of various security approaches. Among the most widely applied methods, encryption remains essential. The Advanced Encryption Standard (AES) is one of the most popular algorithms, as it strikes a balance between processing speed and a high level of security (Bharat et al., 2024). Nevertheless, encryption alone is insufficient. Once a document has been decrypted, the risk of redistribution or unauthorized sharing remains. This highlights the need for multilayered security systems that not only protect data confidentiality but also regulate and control user access (Aldaoud et al., 2024).

In addition to encryption, dynamic authentication mechanisms have been widely adopted as an additional layer to strengthen security systems. One common form of such authentication is the One-Time Password (OTP) or one-time token. This technology has proven effective since it is valid for a single use or a limited time period, reducing the risk of replay attacks and credential theft (El-Booz et al., 2016). Due to its advantages, OTP is widely used in two-factor authentication (2FA), digital banking transactions, and web-based applications requiring stronger protection of user identity and sensitive data (Chee Lee Chong & Nur Ziadah Harun, 2025).

Previous academic studies have examined the integration of AES encryption with OTP systems. Some research even demonstrated that this combination can significantly enhance information system security (Bartlomiejczyk & El Fray, 2024).

However, most of these studies focus on applications in login authentication or network access. Meanwhile, research addressing the protection of individual documents remains limited. This research gap provides an opportunity to explore the use of one-time tokens as unique keys for opening documents in a single session, thereby ensuring stronger access control.

This study aims to fill this gap by designing a PDF document security system that integrates AES-128 encryption with one-time tokens. The proposed system not only protects data confidentiality through encryption but also regulates user access rights via unique tokens that expire immediately after use. Another advantage of the proposed model is its independence—it can operate locally without relying on cloud infrastructure. This makes it particularly suitable for environments with high sensitivity to privacy and data sovereignty (Bartlomiejczyk & El Fray, 2024).

For token storage and management, a lightweight database is employed that can be executed locally, ensuring efficiency while maintaining security. This mechanism is designed to avoid unnecessary complexity while still providing robust data protection. Such an approach is also relevant for organizations with limited resources yet demanding high levels of security in digital document management (Kalaikavitha et al., 2013).

The research methodology adopts an experimental approach with a quantitative orientation. Empirical testing is conducted to measure the system's effectiveness in rejecting access attempts using previously used or expired tokens. In addition, the study evaluates system performance in terms of document encryption–decryption efficiency, token validation speed, and resilience against various unauthorized access scenarios.

The study is expected to yield a document security system that not only relies on encryption for data protection but also provides advanced access control through the use of one-time tokens. Thus, the proposed solution has the potential to contribute significantly to digital document security, particularly for organizations and individuals requiring high levels of protection without relying on third-party services. In the future, this design could be further enhanced by incorporating additional layers of security, such as biometric validation or location-based authentication, to develop a more comprehensive document protection system.

METODE PENELITIAN

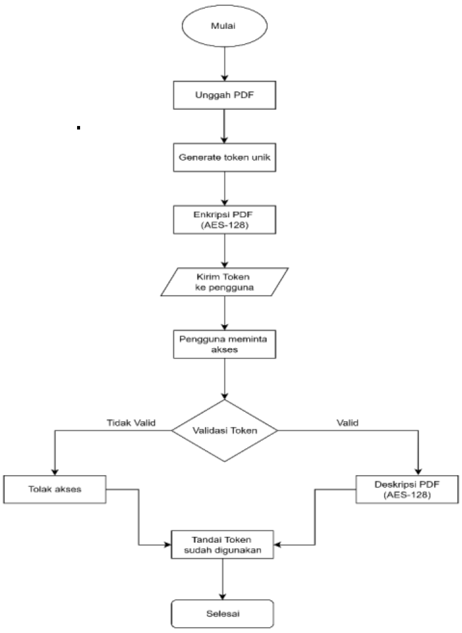
This study adopts a quantitative methodology implemented through an experimental approach. Its primary focus is to empirically validate the effectiveness of a document authentication system that integrates AES-128 encryption with a one-time token mechanism to enhance security in document distribution.

The research design was realized by developing a functional prototype. The prototype was implemented using Python as the main programming language, Flask as the web framework, SQLite as a lightweight database, and the Cryptography library for encryption functions.

The experimental process followed a systematic sequence, beginning with the design of the system architecture and workflow. This stage was followed by code development, which included the implementation of the AES-128 encryption module, the creation of a unique token generator, and the logic for validating one-time tokens integrated into the Flask-based web application. Once the prototype was completed, a series of functional tests were conducted under multiple scenarios, such as attempts to access documents using valid, invalid, and expired tokens, as well as performance tests with documents of varying sizes.

The final phase involved performance evaluation, in which key metrics were measured, including encryption/decryption latency, token validation success rate, and the system's resilience against replay attacks.

Figure 1 illustrates the process flow diagram for the implementation of the one-time token system based on AES-128 for securing PDF documents.



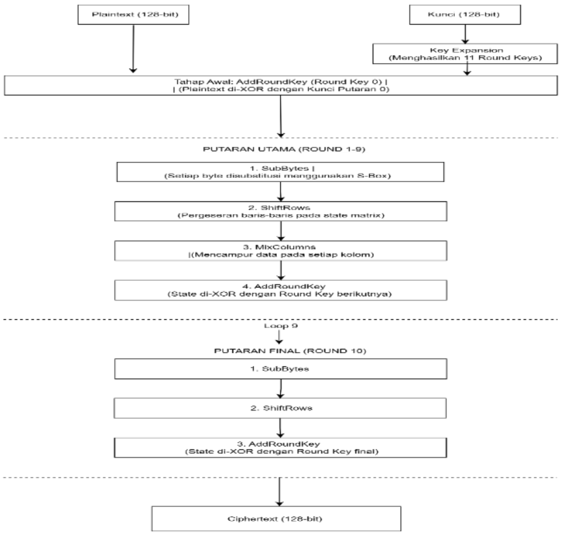
Gambar 1. Diagram Alir Proses Implementasi Sistem One-Time Token Berbasis AES 128 untuk Keamanan Dokumen PDF

1. Algoritma Enkripsi AES-128

In the proposed document security system architecture, the selection of cryptographic algorithms serves as a fundamental pillar that determines the level of confidentiality and data integrity. The Advanced Encryption Standard (AES) with a 128-bit key length (AES-128) was selected as the primary encryption mechanism. This decision is based on an analysis of the optimal balance it provides between proven cryptographic strength and high computational efficiency. These factors represent crucial criteria in designing a secure and responsive application (Balasta et al., 2022). The main function of this algorithm within the system is to perform cryptographic transformations on PDF documents, converting the original data (plaintext) into an encrypted format (ciphertext) that cannot be interpreted without the appropriate decryption key.

AES is a symmetric-key encryption algorithm of the block cipher type, officially adopted as a federal standard by the United States government and published by the National Institute of Standards and Technology (NIST) (Balasta et al., 2022). As a block cipher, AES operates on fixed-size blocks of data, specifically 128 bits.

AES-128 was chosen because it offers a balance between security level and computational efficiency. Encryption is applied to PDF documents in the form of a byte stream, producing an encrypted document that cannot be accessed without the correct key. The AES-128 key is secret and is also required during the decryption process once a valid token is received. Figure 2 presents the flow diagram illustrating the steps performed on a 128-bit block of data during the encryption process using AES-128.


Plaintext (128-bit)
Kunci (128-bit)
Key Expansion
(Menghasilkan 11 Round Keys)
Tahap Awal: AddRoundKey (Round Key 0) |
| (Plaintext di-XOR dengan Kunci Putaran 0)
PUTARAN UTAMA (ROUND 1-9)
1. SubBytes |
(Setiap byte disubstitusi menggunakan S-Box)
2. ShiftRows
(Pergeseran baris-baris pada state matrix)
3. MixColumns
|(Mencampur data pada setiap kolom)
4. AddRoundKey
(State di-XOR dengan Round Key berikutnya)
Loop 9
PUTARAN FINAL (ROUND 10)
1. SubBytes
2. ShiftRows
3. AddRoundKey
(State di-XOR dengan Round Key final)
Ciphertext (128-bit)

Gambar 2. Diagram Alir ini mengilustrasikan langkah-langkah yang terjadi pada satu blok data berukuran 128-bit selama proses enkripsi menggunakan AES-128.

2. One-Time Token (OTT)

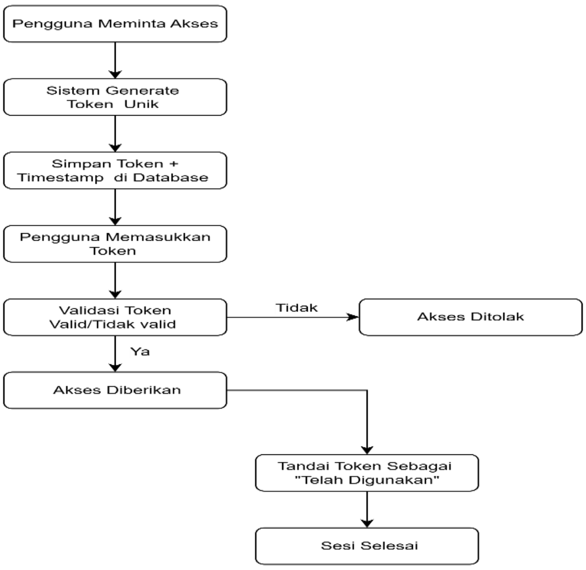
The One-Time Token (OTT), also commonly referred to as the One-Time Password (OTP), is a fundamental authentication security mechanism in modern cybersecurity architectures. This concept was designed to address the inherent weaknesses of traditional static password–based authentication methods, which are highly vulnerable to theft and reuse (Bharat et al., 2024).

A One-Time Token is a dynamic credential consisting of a sequence of characters (alphanumeric or numeric) that is randomly generated and valid only for a single login session or transaction within a very limited time frame. Once it is used or its validity period has expired, the token automatically becomes invalid and cannot be reused.

The key distinction between a static password and a One-Time Token lies in their usage properties:

A. Static Passwords: These credentials are permanent and can be reused multiple times until the user manually changes them. The main weakness lies in the fact that if these credentials are leaked or stolen (for example, through phishing or keylogging attacks), attackers can exploit them at any time to gain unauthorized access.

B. One-Time Token (Dynamic): These credentials are temporary and valid for a single use only. Even if an attacker manages to intercept the token, its value becomes invalid almost immediately due to its short lifespan or because it has already been used by the legitimate user. This significantly reduces the attacker's *window of opportunity* and effectively mitigates the risk of replay attacks, in which stolen credentials are reused to attempt unauthorized access (Aldaoud et al., 2024). Figure 3 illustrates the operational workflow of the One-Time Token mechanism.


Pengguna Meminta Akses
Sistem Generate Token Unik
Simpan Token + Timestamp di Database
Pengguna Memasukkan Token
Validasi Token Valid/Tidak valid
Tidak
Akses Ditolak
Ya
Akses Diberikan
Tandai Token Sebagai "Telah Digunakan"
Sesi Selesai

Gambar 3. Diagram alir Mekanisme operasional *One-Time Token*.

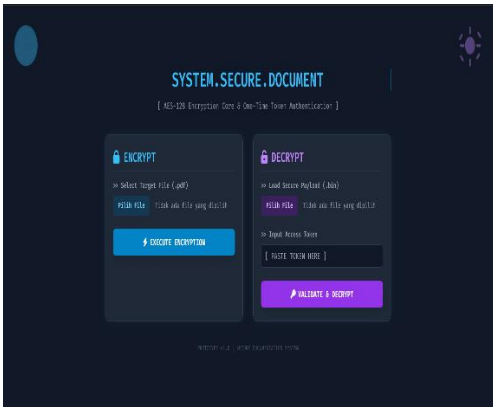
3. Arsitektur Sistem

The prototype system interface was designed using a user-centered design approach, emphasizing ease of interaction and clarity of core functions for end users. Figure 4 presents the web-based prototype interface of the system:

A. "Encrypt" Panel (left): Users can select a PDF file for encryption. Once uploaded, the system applies AES-128 encryption (in CBC mode) and generates a unique One-Time Token (OTT) that can only be used once.

B. "Decrypt" Panel (right): Users enter the received token along with the encrypted file to restore the original PDF document. If the token is invalid, already used, or expired, the system denies access and displays an appropriate error message.

The design of two separate panels clarifies the workflow and reduces the user’s cognitive load, thereby enhancing interaction efficiency and security (Chee Lee Chong & Nur Ziadah Harun, 2025). This approach also aligns with the principles of user-centered design, which emphasize iterative improvements tailored to real user needs.



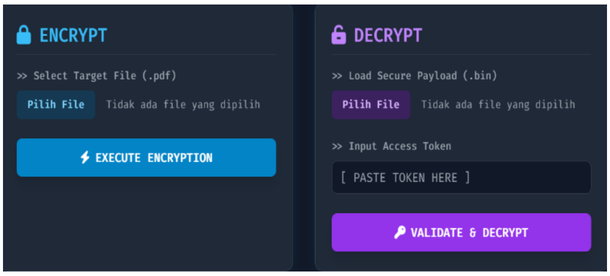
Tidak ada file yang dipilih

Tidak ada file yang dipilih

Gambar 4. Tampilan antarmuka prototipe sistem keamanan dokumen berbasis AES-128

4. Core System Components

The User Interface (Frontend) was developed using HTML and Tailwind CSS, serving as the primary medium for user interaction in uploading, encrypting, and decrypting PDF documents. Tailwind CSS was selected due to its utility-first approach, which enables the rapid development of responsive prototypes without the need for extensive manual CSS coding. This significantly accelerates the design and iteration process (Kurniawan et al., 2021). Figure 5 illustrates the User Interface (Frontend) built using HTML and Tailwind CSS.

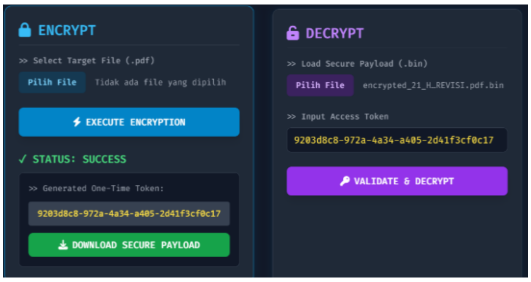


Tidak ada file yang dipilih

Tidak ada file yang dipilih

Gambar 5. Antarmuka Pengguna (Frontend) dibangun menggunakan HTML dan Tailwind CSS

The web application backend was developed using Flask (Python). This application is responsible for handling HTTP requests from the frontend, managing the overall workflow of the system, and connecting the user interface with the core processing logic and data storage. The Flask framework was chosen because it supports the development of lightweight and modular web prototypes, making it highly suitable for experimental system design. Figure 6 presents the program implementation of the web application backend.



Tidak ada file yang dipilih

✓ STATUS: SUCCESS

9203d8c8-972a-4a34-a405-2d41f3cf0c17

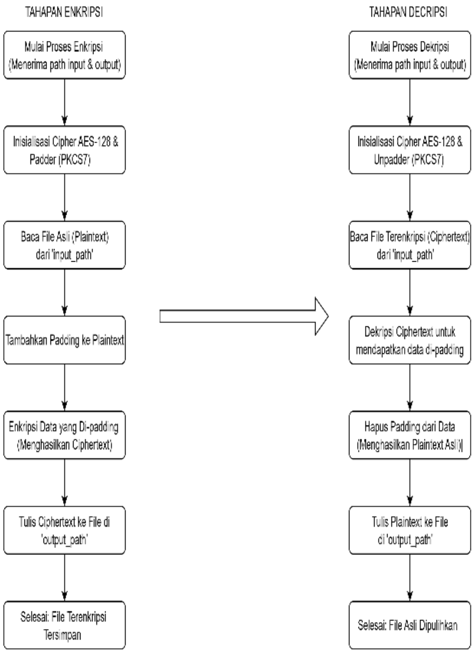
encrypted_21_H_REVISI.pdf.bin

9203d8c8-972a-4a34-a405-2d41f3cf0c17

Gambar 6. Menunjukan program dari Aplikasi Web

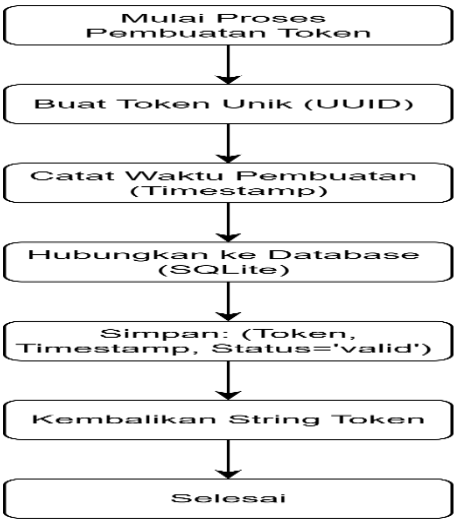
The core logic module (crypto_handler.py) is responsible for managing the encryption and decryption processes of files using the AES-128 algorithm in CBC mode, implemented through the cryptography library. This method has been proven effective in enhancing data security within digital document systems (Kurniawan et al., 2021). Figure 7 illustrates the encryption and decryption workflow of files using the

AES-128 algorithm in CBC mode with the cryptography library.


TAHAPAN ENKRIPSI
Mulai Proses Enkripsi (Menerima path input & output)
Inisialisasi Cipher AES-128 & Padder (PKCS7)
Baca File Asli (Plaintext) dari 'input_path'
Tambahkan Padding ke Plaintext
Enkripsi Data yang Di-padding (Menghasilkan Ciphertext)
Tulis Ciphertext ke File di 'output_path'
Selesai: File Terenkripsi Tersimpan
TAHAPAN DECRIPSI
Mulai Proses Dekripsi (Menerima path input & output)
Inisialisasi Cipher AES-128 & Unpadder (PKCS7)
Baca File Terenkripsi (Ciphertext) dari 'input_path'
Dekripsi Ciphertext untuk mendapatkan data di-padding
Hapus Padding dari Data (Menghasilkan Plaintext Asli)
Tulis Plaintext ke File di 'output_path'
Selesai: File Asli Dipulihkan

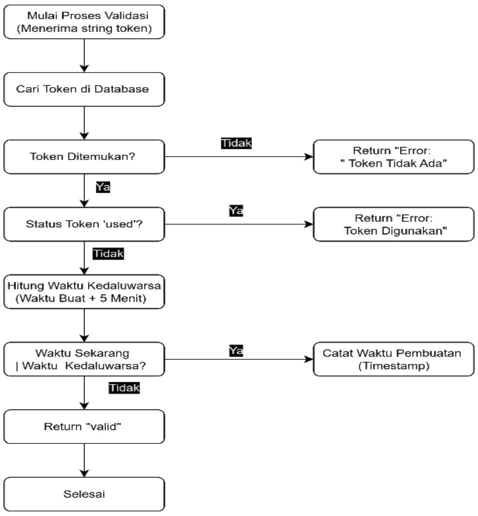
Gambar 7. proses proses enkripsi dan dekripsi file menggunakan algoritma AES-128

The token_manager.py module is responsible for the generation, storage, validation, and deactivation of One-Time Tokens (OTT) to safeguard decryption access. This mechanism ensures that tokens are securely managed and cannot be reused after expiration or successful validation. Figure 8 illustrates the workflow of token generation and storage functions.


Mulai Proses Pembuatan Token
Buat Token Unik (UUID)
Catat Waktu Pembuatan (Timestamp)
Hubungkan ke Database (SQLite)
Simpan: (Token, Timestamp, Status='valid')
Kembalikan String Token
Selesai

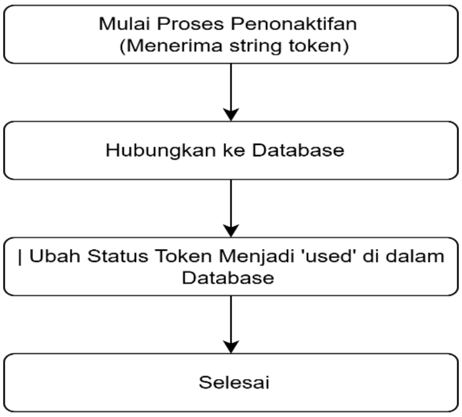
Gambar 8. Alur fungsi generate dan penyimpanan token cryptography

The validate_token function operates as a meticulous gatekeeper within the system's security architecture. Its primary objective is to verify whether a token provided by the user is valid, active, and authorized for use. This process ensures that only legitimate tokens can grant access to the decryption service, thereby preventing unauthorized or replay attacks. Figure 9 illustrates the workflow of the token validation function.


Mulai Proses Validasi (Menerima string token)
Cari Token di Database
Token Ditemukan?
Tidak
Return "Error: " Token Tidak Ada"
Ya
Status Token 'used'?
Ya
Return "Error: Token Digunakan"
Tidak
Hitung Waktu Kedaluwarsa (Waktu Buat + 5 Menit)
Waktu Sekarang | Waktu Kedaluwarsa?
Ya
Catat Waktu Pembuatan (Timestamp)
Tidak
Return "valid"
Selesai

Gambar 9. Alur fungsi alur fungsi validasi token

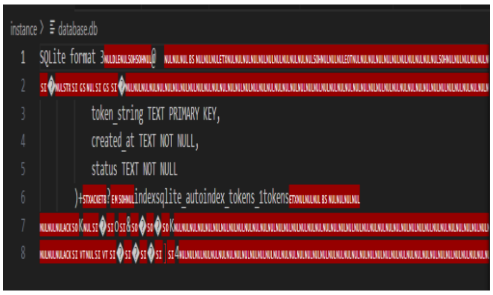
The invalidate_token function is designed with a more straightforward purpose compared to the validation process. Its single and critical task is to ensure that a token can no longer be reused once it has fulfilled its role. By invalidating tokens after successful usage or upon expiration, the system effectively prevents replay attacks and unauthorized reuse. Figure 10 depicts the workflow of the token invalidation function.


Mulai Proses Penonaktifan
(Menerima string token)
Hubungkan ke Database
| Ubah Status Token Menjadi 'used' di dalam Database
Selesai

Gambar 10. Alur fungsi generate dan penyimpanan token cryptography

5. Implementasi SQLite

This system implements SQLite as the database engine. SQLite was chosen due to its characteristics as a *serverless, self-contained, zero-configuration, transactional SQL database engine*. In other words, SQLite does not require a separate server process and stores the entire database within a single file, making it a highly efficient solution for embedded applications and rapid prototyping (Kurniawan et al., 2021). In this implementation, all token-related data is stored in the file instance/database.db. Figure 11 illustrates the *instance* folder containing the database.db file.



```
SQLite format 3
    token_string TEXT PRIMARY KEY,
    created_at TEXT NOT NULL,
    status TEXT NOT NULL
  )+ ? indexsqlite_autoindex_tokens_1tokens
```

Gambar 11. Folder instance yang berisakan database.db.

Interaksi The interaction between the application logic (*token_manager.py* module) and the SQLite database can be described through the following four fundamental operations:

A. Database Schema Initialization (init_db)
This function ensures that the tokens table schema is ready to use by executing the SQL command CREATE TABLE IF NOT EXISTS. This guarantees the integrity of the data structure (consisting of *token_string*, *created_at*, and *status*) before any read/write operations are performed.

B. Data Insertion Operation (generate_and_save_token)
After the encryption process, this function executes an SQL INSERT command to store the newly generated unique token, timestamp, and initial status ('valid') into the tokens table. This ensures that every token has a permanent record for future validation.

C. Data Retrieval Operation (validate_token)
This function performs a SELECT ... WHERE query to search for a matching token. If found, the *created_at* and *status* fields are returned to the application logic for business rule validation (e.g., expiration or prior use). This follows

standard practices in software design (Aldaoud et al., 2024).

D. Data Update Operation (invalidate_token)
To prevent replay attacks, this function executes an SQL UPDATE command after successful decryption. The value of the *status* field is updated to ‘used’, thereby permanently deactivating the token within the database.

HASIL PENELITIAN

System testing was conducted to evaluate the effectiveness of the prototype. The testing approach covered functional, security, and performance aspects. The testing scenarios applied in this study are detailed in Table 1 below.

Tabel 1. Skenario Pengujian Validasi Keamanan

Katagori Pengujian	**Skenario**	**Jumlah percobaan (n)**	**Keberhasilan (x)**	**Persentase (%)**	**Standar deviasi (SD)**
Validasi Keamanan (n=30/percobaan)	Validasi Token salah	30	30	100	0.00
	Simulasi Repplay attack	30	30	100	0.00
	Validasi kadaluarsa Token	30	29	96,7	4,3

After the security validation process was completed, the next stage was the system performance evaluation. This testing aimed to measure the efficiency of the encryption and decryption processes, particularly in terms of computation time under various workloads. Performance evaluation is crucial since cryptographic algorithms, although secure, may impose a significant processing overhead if not properly optimized (Bharat et al., 2024).

The testing was conducted based on three main scenarios: small workload files (~100 KB), large workload files (~10 MB), and boundary testing (stress test) to measure system stability with extreme file sizes (~1 GB). The selection of these file size variations refers to the performance testing methodology recommended by recent studies in cryptographic performance analysis (Kurniawan et al., 2021).

The metrics measured included the average computation time for encryption and decryption processes, as well as observations regarding system stability and resource utilization. The results of this evaluation are presented in Table 2 below.

Tabel 2. Evaluasi Kinerja Sistem

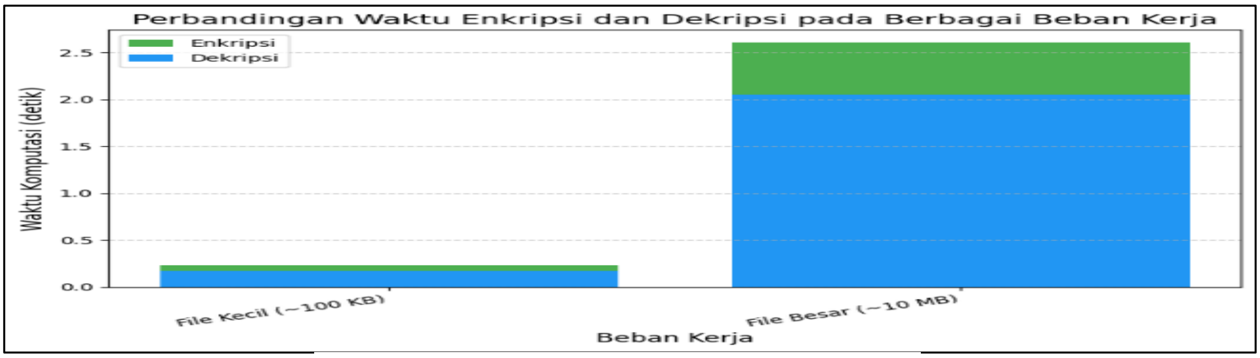
Beban kerja	**Metrik yang di ukur**	**Rata-rata(detik)**
File kecil (100 KB)	Enkripsi	0.23
	Dekripsi	0.17
File besar (100 MB)	Enkripsi	2.61
	Dekripsi	2.05
Uji Batas (1 GB)	**-**	**-**

The computational performance of the implemented document security system was evaluated through a series of empirical tests. The primary metrics measured were the average time required for cryptographic operations (AES-128 encryption and decryption) across three levels of data volume: ~100 KB (small), ~10 MB (large), and ~1 GB (stress test). Quantitative results (Figure 2) confirmed a linear relationship between data volume and latency. For the small workload, encryption and decryption times were recorded at 0.23 and 0.17 seconds, respectively. For the large workload, these times increased to 2.61 and 2.05 seconds. The stress test scenario revealed a performance bottleneck, where processing a ~1 GB file led to inefficient memory resource utilization and significant delays, making precise time measurement infeasible.

These findings indicate that the current prototype architecture is optimal for small- to medium-scale documents but requires optimization—such as the implementation of streaming techniques—to handle large-scale data. This conclusion is consistent with previous research that highlights the impact of system resource limitations on cryptographic algorithm performance (Bharat et al., 2024).

Figure 11. Comparison of Computation Time by File Size
The bar chart above visually compares the average time (in seconds) required for encryption and decryption processes on two different file sizes. It is evident that computation time increases significantly

with larger file sizes; however, both operations remain within an efficient range for practical use.



Gambar 11. Perbandingan Waktu Komputasi Berdasarkan Ukuran File

PEMBAHASAN

This study highlights how the proposed system successfully addresses fundamental challenges in digital document security. At its core, the solution integrates One-Time Tokens as a dynamic access gateway to safeguard AES-128 encrypted PDF documents.

Practically, the experimental results demonstrate that the system is highly reliable. Every attempt to access documents using invalid or expired tokens was effectively blocked, proving the robustness of the validation mechanism with no exploitable gaps. This directly mitigates common attacks such as replay attacks. On the other hand, AES-128 encryption acts as a strong barrier to preserve document confidentiality, a method still widely recognized for its cryptographic strength.

The advantages of this system become even more apparent when compared to traditional password-based methods, where the combination of encryption and one-time tokens provides a significant security enhancement.

This implies that the developed model holds substantial potential for application in various data-sensitive domains. In the future, the system can be further enriched with technologies such as Public Key Infrastructure (PKI) to reinforce identity verification, or even blockchain to create immutable distribution records.

PENUTUP

Based on the findings of this study, it can be concluded that the security model integrating AES-128 symmetric encryption with a One-Time Token (OTT) mechanism significantly enhances the security posture in PDF document distribution.

The observed relationship among variables confirms that the system’s effectiveness is determined by the combination of data confidentiality ensured by encryption and dynamic access control enabled by OTT. This implementation has proven effective in mitigating common attack vectors such as replay attacks by ensuring that each token is valid for only a single session.

For future research, it is recommended to explore the use of streaming encryption methods to optimize the processing of large-scale files, as well as the potential integration with distributed ledger technologies (such as blockchain) to improve the reliability and transparency of the system’s audit trail.

DAFTAR PUSTAKA

Aldaoud, M., Al-Abri, D., Kausar, F., & Awadalla, M. (2024). NDNOTA: NDN One-Time Authentication. *Information (Switzerland)*, *15*(5). https://doi.org/10.3390/info15050289

Asyura Binti Sofian, Ayu Fitri Alafiah Binti Peradus, Fidel Yong, Irvine Shearer, Nurrul Nazwa Binti Ismail, Yugendran A/L Mahendran, & Muhammad Faisal. (2024). Enhancing Authentication Security: Analyzing Time-Based One-Time Password Systems. *International Journal of Computer Technology and Science*, *1*(3), 56–70. https://doi.org/10.62951/ijcts.v1i3.25

Balasta, D. U., Marie Pelito, S. C., Christopher Blanco, M. R., Alipio, A. J., Mata, K. E., & Michael Cortez, D. A. (2022). Enhancement of Time-Based One-Time Password for 2-Factor Authentication. In *International Journal of Innovative Science and Research Technology* (Vol. 7, Nomor 6).

Bartlomiejczyk, M., & El Fray, I. (2024). Device Risk Analysis Protocol for SMS-Based OTP Authentication. *IEEE Access*, *12*, 123177–123192. https://doi.org/10.1109/ACCESS.2024.3445931

Bharat, M., Dash, R., Reddy, K. J., Murty, A. S. R., C., D., & Muyeen, S. M. (2024). Secure and efficient prediction of electric vehicle charging demand using α2-LSTM and AES-128 cryptography. *Energy and AI*, *16*. https://doi.org/10.1016/j.egyai.2023.100307

Chee Lee Chong, & Nur Ziadah Harun. (2025). Secure File Sharing System with Strong Password and One Time Password Authentication. *Journal of Computing Research and Innovation*, *10*(1), 98–107. https://doi.org/10.24191/jcrinn.v10i1.500

El-Booz, S. A., Attiya, G., & El-Fishawy, N. (2016). A secure cloud storage system combining time-based one-time password and automatic blocker protocol. *Eurasip Journal on Information Security*, *2016*(1). https://doi.org/10.1186/s13635-016-0037-0

Kalaikavitha, M. E. C. A., Phil, M., Juliana, M., Sc, M., & Professor, A. (2013). Secure Login Using Encrypted One Time Password (Otp) and Mobile Based Login Methodology. In *Research Inventy: International Journal Of Engineering And Science* (Vol. 2).

Kurniawan, D. E., Iqbal, M., Friadi, J., Hidayat, F., & Permatasari, R. D. (2021). Login Security Using One Time Password (OTP) Application with Encryption Algorithm Performance. *Journal of Physics: Conference Series*, *1783*(1). https://doi.org/10.1088/1742-6596/1783/1/012041