JURNAL INFORMATIKA
Volume 13 No. 1 April 2026, pp. 62-70
ISSN (Online) 2528-2247 | ISSN (Print) 2355-6579
DOI: https://doi.org/10.31294/informatika.v13i1.12372

Implementation of Image Data Security Using the AES-256 Algorithm in the Work Accident Recording System

Dzulfiqar Alang Setiawan[1*], Bambang Agus Herlambang [1], Ramadhan Renaldy[1]

[1]Universitas PGRI Semarang
Jl. Lontar No.24, Karangtempel, Kota Semarang, Indonesia

Correspondence e-mail: 22670127@upgris.ac.id

Submission:	Revision:	Acceptance:	Available Online:
05-03-2026	11-03-2026	25-03-2026	04-04-2026

Abstract - This study aims to implement an image data security mechanism in a web-based work accident recording system in an industrial environment by applying the Advanced Encryption Standard (AES) 256-bit cryptographic algorithm in Cipher Block Chaining (CBC) mode. The problem faced is the lack of an adequate protection mechanism for sensitive work accident photo files so that they have the potential to be accessed, copied, or modified by unauthorized parties. This study uses the Software-Oriented Prototyping method which allows system development to be carried out iteratively based on user needs and evaluation results at each development stage. The encryption process is carried out when the image file is uploaded into the system by generating a random Initialization Vector (IV) of 16 bytes, then the image data is encrypted using the AES-256-CBC algorithm and stored in ciphertext form with the file extension .enc. The decryption process is carried out when the file will be displayed again using the appropriate secret key and IV without storing the file in plaintext form on the server. The test results show that the encryption process has an average execution time of around 0.002–0.008 seconds, while the decryption process takes around 0.00004–0.001 seconds. In addition, the size of the encrypted file relatively follows the size of the original image file with an additional size of around 16–32 bytes due to the padding process and the use of initialization vectors. The results of the study indicate that the application of the AES-256-CBC algorithm is able to maintain the confidentiality and integrity of image data without having a significant impact on system performance. Thus, the developed system can improve the security of digital file storage and support a more structured, secure, and efficient management of work accident data.

Keywords: AES-256-CBC; Image encryption; Data security; Workplace accident system

1. Introduction

Occupational safety can be defined as the protection or avoidance of individuals from accidents or hazards, whether those that could cause physical harm, while they are working. Occupational safety is crucial for maintaining the well-being of workers and reducing the risk of occupational diseases or accidents. Despite the fact that many companies have implemented Occupational Safety Management Systems to improve employee safety and health, several obstacles remain (Febriyanti et al., 2024).

Advances in information technology offer an opportunity to address these issues through the implementation of web-based information systems capable of digitally managing workplace accident data. Such systems enable fast and efficient data storage, retrieval, and reporting. However, data security remains crucial, particularly since these systems store sensitive information such as technician identities, incident locations, and potentially private photographic evidence of accidents.

To ensure the security and confidentiality of the data, a protection mechanism through encryption is required. In this study, the Advanced Encryption Standard (AES-256) method was applied, AES is a symmetric cryptographic algorithm that is capable of encrypting data at high speed and a high level of security, so it is very suitable for protecting sensitive data (Set et al., 2025). AES is considered as one of the efficient and high-performance cryptographic techniques (Indrayani et al., 2024). AES has advantages such as high data processing speed and efficiency and a high level of security, which makes it difficult to be cracked by unauthorized parties (Dewantara, 2025).

In the article "Cryptography for Double Encryption on Images Using the AES (Advanced Encryption Standard) and RC5 (Rivest Code 5) Algorithms" explains that the AES algorithm, especially with a key length of 256 bits, has a high level of security and is effective in protecting digital image data through the encryption process, so that it is able to maintain the confidentiality of information from the threat of unauthorized access and cryptographic attacks (Hadiana & Sabrina, 2022).

Another study by (Marsiani et al., 2021) entitled "Implementation of AES 256-Bit GCM Security System to Secure Personal Data" focuses on the implementation of AES 256-bit with Galois/Counter Mode (GCM)

operation mode as a method for securing sensitive data. The results of this study show that AES-256 in GCM mode not only provides strong encryption, but also provides data authentication that ensures the integrity and authenticity of information. Thus, the implementation of this algorithm is very effective in protecting personal data from potential unauthorized modification and access without significantly sacrificing system performance.

Different from previous research, this study focuses on the application of the AES-256 cryptographic algorithm with Cipher Block Chaining (CBC) mode on a web-based work accident recording system at PT. Telkom Akses Witel Kudus. This system not only functions to store data digitally, but also applies encryption to image files (photos of accident evidence) to protect sensitive information from unauthorized access. The encryption process is carried out when the user uploads a photo, while decryption only occurs when the data is displayed again using a secret key and an Initialization Vector (IV) that is unique for each file. By using a Software-Oriented Prototyping approach, the system was developed iteratively with the user so that the implementation results show that AES-256-CBC is effective in maintaining data security and integrity without reducing system performance. This research provides a real contribution to the application of cryptography in industrial environments by combining aspects of digital data security and information system efficiency, while expanding the application of AES-256 to web-based information systems that are relevant in the digital era.

The main objective of this system is to control work-related risks to create a safe, efficient, and productive work environment. In addition, the OHS Management System aims to ensure that all processes related to occupational safety and health are well-managed (Maimunah et al., 2024).

2. Research Methods

This research began with an analysis of the existing work accident recording system at PT. Telkom Akses Witel Kudus through observation and interviews with Health, Safety, and Environment (HSE) personnel. The analysis revealed that the work accident recording process is still carried out manually using written forms, with separate and unintegrated data storage, potentially leading to delays in reporting, difficulties in data retrieval, and the risk of errors and information loss.

This situation has led to several problems, including the risk of data loss, difficulty tracking accident history, and a lack of protection for sensitive accident photo evidence files. The analysis determined that the new system needed to have the following capabilities record work accident data digitally and in an integrated manner, store data with a high level of security, encrypt photo evidence of work accidents to prevent unauthorized access.

To meet these needs, the AES-256 algorithm is used as a digital data security solution, because this algorithm has proven to be efficient and strong in protecting multimedia files, as shown in research by (Hernandi & Chandra, 2024) which proves that AES-256 is able to maintain the integrity and confidentiality of digital image files.

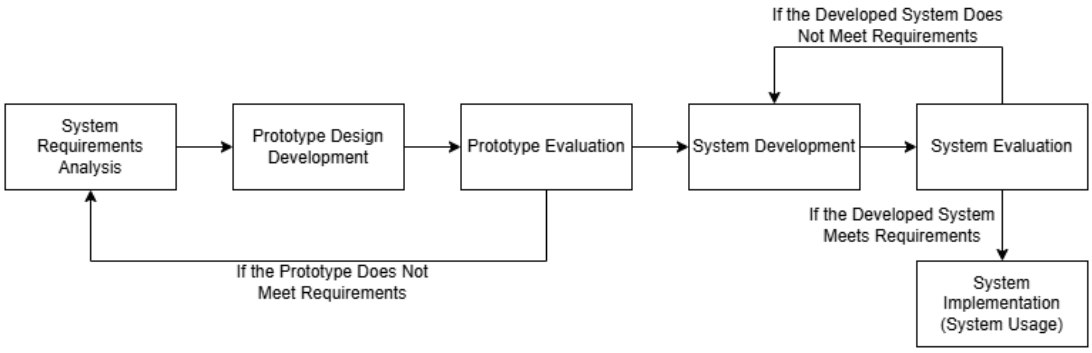

Figure 1. Methods of Study

This research applies the Software-Oriented Prototyping method as an approach in system development, where the system is developed in stages through the creation of an initial prototype which is then evaluated and refined repeatedly (Siswidiyanto et al., 2020). In each stage of development, users are directly involved to provide input and feedback on the functionality, appearance, and suitability of the system to the expected needs, so that the final system results can be more optimal, appropriate, and in accordance with user needs (Arsad & Muare, 2024).

This method was chosen because the system being developed requires iterative adjustments to meet real-world security needs and workflows. The stages of this method include:

1) Needs Analysis
   This stage aims to determine user requirements for the system to be developed. The analysis is conducted through interviews and observations with HSE personnel to identify required features, such as incident data management, uploading photo evidence, report verification, and generating work accident reports (Senubekti et al., 2024).

2) Prototype Design Creation
   Implementing the design into an initial web-based system that can be tested by users (Meisak & Agustini, 2022).
3) Prototype Evaluation
   Involving users to provide input on the system's function, appearance, and security, then making improvements until the system meets operational needs (Fridayanthie et al., 2021).
4) System Development
   After the prototype design evaluation phase is completed and approved by the user, the next step is to translate the design results into appropriate programming code, namely PHP, with a MySQL database for data storage (Sari et al., 2023). During this implementation phase, the Advanced Encryption Standard (AES) 256-bit algorithm is also implemented as a security mechanism to protect sensitive data, particularly photo files containing evidence of work accidents. Encryption occurs when data is stored in the system, while decryption occurs when the data is displayed to authorized users (Bhaudhayana & Widiartha, 2015).
5) System Evaluation
   After the initial design phase or prototype creation of the Workplace Accident Recording Information System was completed, an application evaluation was conducted to obtain user feedback regarding the system's functionality and appearance. This evaluation process aimed to ensure that the developed system met user needs and effectively supported the management of workplace accident data. The system prototype was then provided to potential users, namely the Health, Safety, and Environment (HSE) Department, for direct testing (Pradipta et al., 2021). Users tested the system's features and provided input, criticism, and suggestions for improvement as a basis for system refinement. Furthermore, system performance was tested by measuring the encryption and decryption times of image files using the AES-256-CBC algorithm implemented through the OpenSSL library in the PHP Laravel framework. The test results showed that the encryption process had an average execution time of approximately 0.002–0.008 seconds, while the decryption process took approximately 0.00004–0.001 seconds. An analysis was also performed on the file sizes before and after the encryption process, where the results showed that the encrypted file size relatively followed the size of the original image file with an additional 16–32 bytes due to the use of padding and initialization vectors. These test results indicate that the implemented encryption mechanism is capable of maintaining image data security without significantly impacting system performance.
6) System Usage
   The system usage stage is the process of officially implementing and utilizing the system in an operational environment. At this stage, the system, which has undergone testing and evaluation, begins to be used by users according to their respective functions and access rights (Ichwani et al., 2021).

The AES CBC mode cryptographic algorithm uses the Initialization Vector (IV) value in the cipher block. The IV value is based on the size of each input plaintext block, and each bit of the plaintext sequence will be divided into equal blocks until they have the same size. To create the ciphertext, the block chaining mode is used (Irvai & Efranda, 2024).

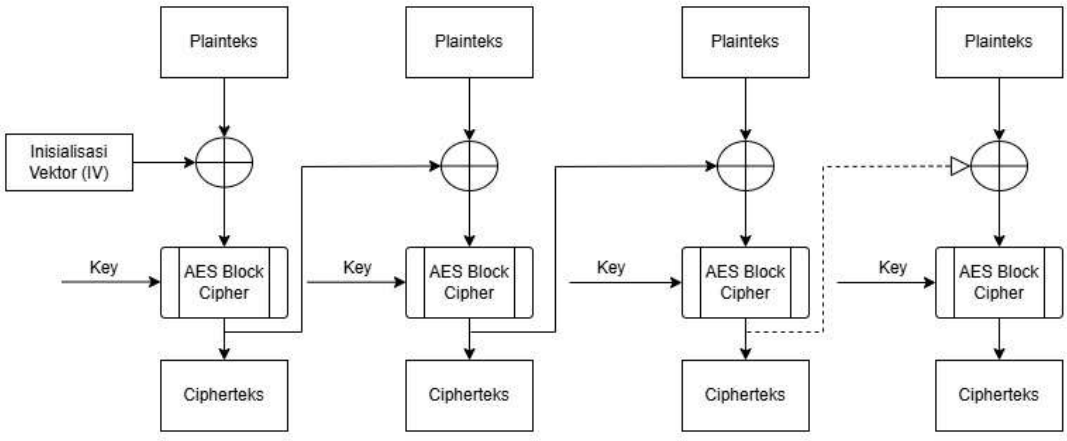

Figure 2. AES CBC mode encryption algorithm

Figure 2 explains that the modified AES with CBC operates sequentially, where the result of the first data block affects the result of the second block, and so on. In the initial operation, the researcher uses the IV data value in the block with the initial output of the AES Block Cipher. The performance of the block uses AES with 128 bits, which corresponds to a 16-character key (128 bits), a 24-character key (192 bits) and a 32-character key (256 bits) (Nugrahantoro et al., 2020).

Figure 3. AES 256 Encryption Process Diagram

The implementation of the AES-256 algorithm in the work accident recording system is carried out to maintain the security of image data (photo evidence of accidents) uploaded by users into the system. The encryption process begins when the user uploads a photo via the add accident history form. The system then reads the image file and automatically generates a random 16-byte Initialization Vector (IV) for each encryption process. Next, the AES-256 algorithm in Cipher Block Chaining (CBC) mode is used together with a secret key to convert the original data (plaintext) into ciphertext. The result of the encryption process is a file with the extension .enc, which contains a combination of IV and ciphertext, then saved to the system storage directory (uploads/). Thus, the file stored on the server is already encrypted and cannot be opened directly without going through the decryption process.

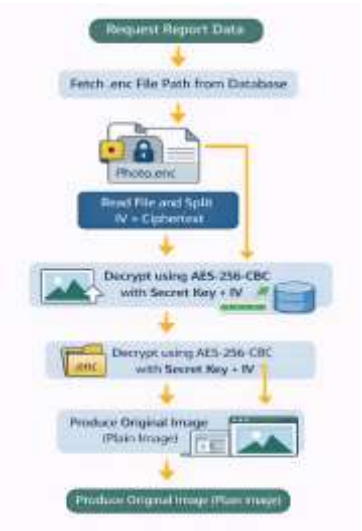

Figure 4. AES 256 Process Description Diagram

The decryption process is performed when a user or administrator accesses the work accident history page or displays specific data. The system retrieves the encrypted file based on the path stored in the database, then separates the IV and ciphertext from the file. Next, the AES-256-CBC algorithm is reused to perform the decryption process, using the same secret key and IV used during encryption. The result of this process is a plain image that can be displayed on the system interface for verification and reporting purposes. After the image is displayed, the decrypted file is temporarily deleted from storage to maintain data security.

With this mechanism, the system is able to guarantee the confidentiality, integrity, and security of image files from possible unauthorized access or data manipulation. The implementation of the AES-256 algorithm in this system not only ensures information security, but also demonstrates the real application of the concept of

symmetric cryptography in the context of web-based information systems in industrial environments (Purwanti et al., 2025).

3. Result and Discussion

This research produces a web-based work accident data management information system that implements the Advanced Encryption Standard (AES) cryptographic algorithm with a 256-bit key length in Cipher Block Chaining (CBC) mode to secure user-uploaded photo files. The implementation uses the PHP programming language with the Laravel framework and utilizes the OpenSSL library for encryption and decryption.

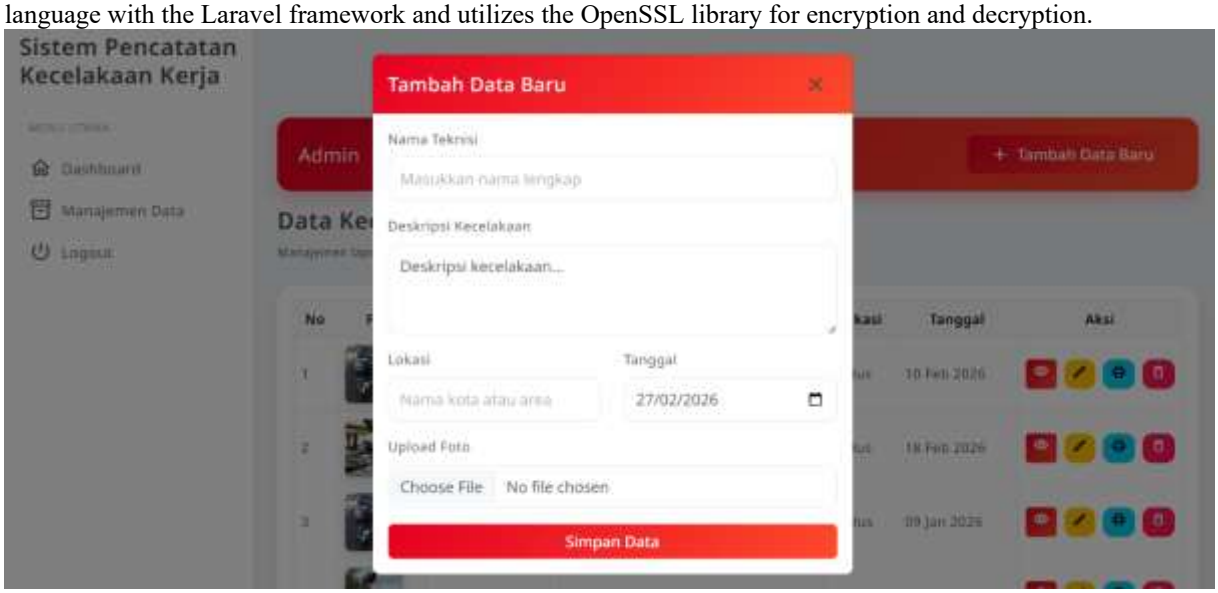

Figure 5. File upload and encryption process using AES-256-CBC

Security mechanisms are implemented during the file storage process. Each image file uploaded by a user is first validated based on its type and size. After successful validation, the file is read in binary form and encrypted using the AES-256-CBC algorithm. The encryption key is derived from a secret value stored in the system configuration file and then processed using the SHA-256 hash function to generate a 256-bit key according to the AES standard.

The system generates a random 16-byte Initialization Vector (IV) for each encryption process. This IV is combined with the ciphertext and stored as a .enc file in the server's storage directory. This approach ensures that the original file is never stored in plaintext, minimizing the risk of unauthorized access.

Figure 6. The encrypted file is saved in ciphertext format (.enc)

During image access, the system performs decryption by separating the IV from the ciphertext and then executing the decryption process using the same key. The decrypted file is not saved back to the server but is instead sent directly as an HTTP response, enhancing data storage security.

Figure 7. AES-256-CBC encryption process flow on the system

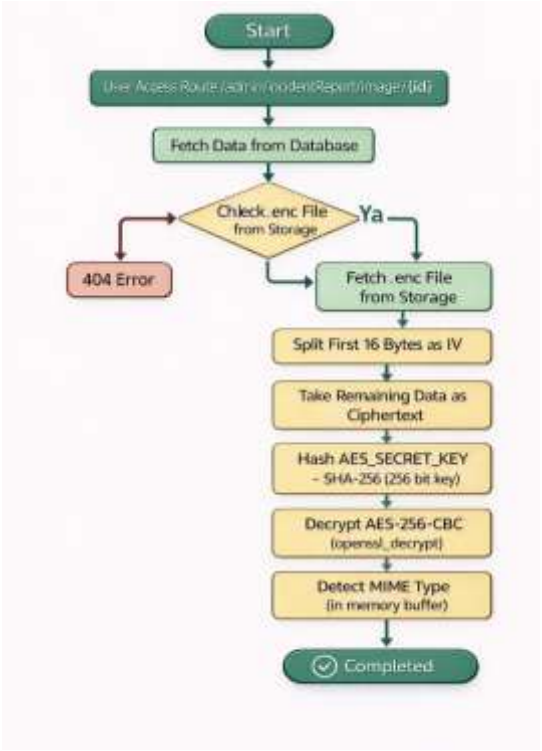

Figure 8. AES-256-CBC decryption process flow on the system

The encryption flowchart illustrates the process when an image is first uploaded to the system. After a user submits a file, the system doesn't immediately store it as is. The file is first read as raw data, then the system generates a random value (IV) that serves as an additional scrambler. The secret key is then processed to create a strong key, and the image is altered using the AES-256-CBC cryptographic algorithm so that the file's contents are no longer recognizable. The resulting scramble is combined with the IV value and saved to server storage in the form of a .enc file. This means that what is stored on the server is not the original image, but rather a locked version.

The decryption flowchart illustrates the reverse, when an image is to be displayed again. When a user opens a detail page and the system is prompted to display a photo, it retrieves the encrypted file from storage. The system extracts the initial portion of the file as the IV value, then reuses the same secret key to decrypt the image. Once the decryption process is complete, the data is restored to its original image state. The system then recognizes the file type and sends it to the browser, allowing the user to view the photo as usual.

```
private function encryptFile($file)
{
$key = hash('sha256', env('AES_SECRET_KEY'));
$iv = openssl_random_pseudo_bytes(16);

$fileContent = file_get_contents($file->getRealPath());

$encrypted = openssl_encrypt(
$fileContent,
'AES-256-CBC',
$key,
OPENSSL_RAW_DATA,
$iv
);

return $iv . $encrypted;
}
```

ISSN (Online) 2528-2247 | ISSN (Print) 2355-6579

The code snippet is an implementation of the image file encryption process using the AES-256-CBC algorithm before saving it to the server. The system first generates a 256-bit key from an environment variable hashed with SHA-256, then generates a random Initialization Vector (IV) of 16 bytes according to the AES block size. The uploaded file is read in binary form and encrypted using the openssl_encrypt() function with the raw data option to produce ciphertext in its original format. The final result is a combination of the IV and ciphertext that is returned by the function for storage, so that the file is stored in an encrypted state and can only be opened again through the decryption process with the same key.

```
private function decryptFile($encryptedData)
{
    $key = hash('sha256', env('AES_SECRET_KEY'));

    $iv = substr($encryptedData, 0, 16);
    $ciphertext = substr($encryptedData, 16);

    return openssl_decrypt(
        $ciphertext,
        'AES-256-CBC',
        $key,
        OPENSSL_RAW_DATA,
        $iv
    );
}
```

This code snippet implements the decryption process for a file previously encrypted using AES-256-CBC. The system first regenerates a 256-bit key from an environment variable hashed with SHA-256 to ensure consistency with the encryption process. Next, the encrypted data is split into two parts: the first 16 bytes as the Initialization Vector (IV) and the remainder as ciphertext. The decryption process is performed using the openssl_decrypt() function with the same algorithm and key, and the OPENSSL_RAW_DATA option to read the data in binary format. If the key and IV match, the function returns the original data (plaintext), allowing the file to be displayed again without permanently storing its decrypted form on the server.

```
[2026-03-11 03:10:01] local.INFO: Waktu Dekripsi AES-256-CBC: 8.6069107055664E-5 detik
[2026-03-11 03:10:02] local.INFO: Waktu Dekripsi AES-256-CBC: 5.2928924560547E-5 detik
[2026-03-11 03:10:03] local.INFO: Waktu Dekripsi AES-256-CBC: 6.2942504882812E-5 detik
[2026-03-11 03:10:03] local.INFO: Waktu Dekripsi AES-256-CBC: 5.7220458984375E-5 detik
[2026-03-11 03:10:04] local.INFO: Waktu Dekripsi AES-256-CBC: 6.103515625E-5 detik
[2026-03-11 03:10:23] local.INFO: Waktu Enkripsi AES-256-CBC: 0.0028681755065918 detik
[2026-03-11 03:10:25] local.INFO: Waktu Dekripsi AES-256-CBC: 0.0016679763793945 detik
[2026-03-11 03:10:26] local.INFO: Waktu Dekripsi AES-256-CBC: 8.702278137207E-5 detik
[2026-03-11 03:10:26] local.INFO: Waktu Dekripsi AES-256-CBC: 5.793571472168E-5 detik
[2026-03-11 03:10:27] local.INFO: Waktu Dekripsi AES-256-CBC: 6.2942504882812E-5 detik
[2026-03-11 03:10:28] local.INFO: Waktu Dekripsi AES-256-CBC: 4.9114227294922E-5 detik
[2026-03-11 03:10:28] local.INFO: Waktu Dekripsi AES-256-CBC: 5.793571472168E-5 detik
[2026-03-11 03:18:54] local.INFO: Waktu Enkripsi AES-256-CBC: 8.2969665527344E-5 detik
[2026-03-11 03:18:56] local.INFO: Waktu Dekripsi AES-256-CBC: 6.103515625E-5 detik
[2026-03-11 03:18:56] local.INFO: Waktu Dekripsi AES-256-CBC: 0.0011730194091797 detik
```

The AES-256-CBC algorithm was successfully implemented to secure image files uploaded to the work accident recording system. Performance testing was conducted by measuring the encryption and decryption process times using the timer function in the PHP Laravel framework that utilizes the OpenSSL cryptography library. The test results show that the encryption process execution time is in the range of approximately 0.002 to 0.008 seconds, while the decryption process takes approximately 0.00004 to 0.001 seconds. These values indicate that the image data security process can be carried out very quickly so that it does not place a significant burden on system performance.

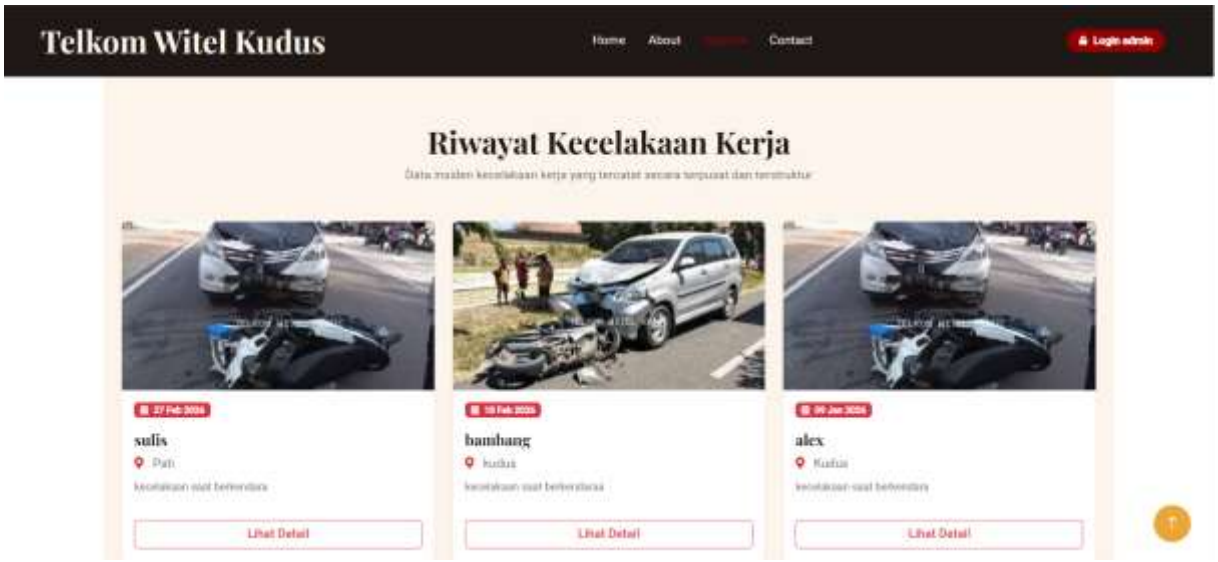

Figure 9. Reports page display

On the report page, the system does not allow users to right-click and save images directly (right-click, Save As). Furthermore, each displayed image is watermarked with a system identifier and data ownership marker permanently placed on the image. This watermark serves as a visual protection layer to prevent misuse, unauthorized distribution, or manipulation of the content by unauthorized parties. This restriction is part of the data security mechanisms implemented in the system to maintain the confidentiality, integrity, and control access to workplace accident documentation.

4. Conclusion

This research successfully designed and implemented a web-based occupational accident recording system equipped with an image data security mechanism using the Advanced Encryption Standard (AES) 256-bit algorithm in Cipher Block Chaining (CBC) mode. The implementation of the encryption mechanism during the file upload process and decryption during the data display process was able to maintain the confidentiality and integrity of the occupational accident evidence photos without storing the data in its original form on the server. The test results showed that the encryption and decryption processes had a relatively fast execution time so that they did not provide a significant computational burden on system performance. In addition, the size of the encrypted file relatively followed the size of the original image file with minimal size addition due to the padding process and the use of initialization vectors. These findings indicate that the application of the AES-256-CBC algorithm is effective in improving the security of image data storage in a web-based occupational accident recording system. Further research can develop further security approaches through the application of encryption methods that support data authentication, such as AES-GCM, as well as the integration of asymmetric cryptography-based key management mechanisms to improve the level of system security more comprehensively.

References

Arsad, R., & Muare, M. S. (2024). Perancangan Sistem Informasi Jdih Berbasis Web Dengan Metode Prototype. *Seminar Nasional Teknologi & Sains*, *3*(1), 67–75.

Bhaudhayana, G. W., & Widiartha, I. M. (2015). Implementasi algoritma kriptografi aes 256 dan metode steganografi lsb pada gambar bitmap. *Jurnal Ilmu Komputer Universitas Udayana*, *8*(2), 15–25.

Dewantara, D. O. (2025). *Kombinasi perlindungan data material SAP menggunakan Algoritma AES 128 Bit dan Reverse Cipher di PT Indonesia Comnet Plus*. Universitas Islam Negeri Maulana Malik Ibrahim.

Febriyanti, A. D., Yulinar, R. D., Samudra, S. F., & Radianto, D. O. (2024). Peningkatan Keselamatan Kerja Melalui Implementasi Sistem Manajemen Keselamatan dan Kesehatan Kerja (SMK3). *Journal of Educational Innovation and Public Health*, *2*(2), 72–85.

Fridayanthie, E. W., Haryanto, H., & Tsabitah, T. (2021). Penerapan metode prototype pada perancangan sistem informasi penggajian karyawan (persis gawan) berbasis web. *Jurnal Khatulistiwa Informatika*, *23*(2), 472897.

Hadiana, A. I., & Sabrina, P. N. (2022). Kriptografi Untuk Enkripsi Ganda Pada Gambar Menggunakan Algoritma AES (Advanced Encryption Standard) Dan RC5 (Rivest Code 5). *Informatics and Digital Expert (INDEX)*, *4*(1), 25–32.

Hernandi, R. M. H., & Chandra, J. C. (2024). Implementasi Algoritme AES-256 dan AES-GCM untuk Mengamankan Dokumen Pada Sistem Data Rekam Medis Klinik Mulya. *KRESNA: Jurnal Riset Dan*

Pengabdian Masyarakat, *4*(1), 12–22.
Ichwani, A., Anwar, N., Karsono, K., & Alrifqi, M. (2021). Sistem Informasi Penjualan Berbasis Website dengan Pendekatan Metode Prototype. *Prosiding Sisfotek*, *5*(1), 1–6.
Indrayani, R., Ferdiansyah, P., & Koprawi, M. (2024). Analisis Penggunaan Kriptografi Metode AES 256 Bit pada Pengamanan File dengan Berbagai Format. *Digital Transformation Technology*, *4*(2), 1245–1251.
Irvai, M., & Efranda, N. (2024). Optimalisasi Enkripsi File Menggunakan Algoritma Aes-256 Berbasis Web Dengan Integrasi Kompresi Adaptif. *Betrik*, *15*(03), 528–536.
Maimunah, P., Munthe, S., Mahendra, A. F. R., Haridani, H., & Purba, S. H. (2024). Penerapan Sistem Manajemen Keselamatan dan Kesehatan Kerja (SMK3) di Perusahaan Pertambangan: Review Literatur. *Journal of Educational Innovation and Public Health*, *2*(3), 115–125.
Marsiani, E. S., Setiadi, I., & Cahyo, A. (2021). Implementasi Sistem Keamanan AES 256-Bit GCM Guna Mengamankan Data Pribadi. *Jurnal Rekayasa Komputasi Terapan*, *1*(02).
Meisak, D., & Agustini, S. R. (2022). Penerapan metode prototype pada perancangan sistem informasi penjualan mediatama solusindo jambi. *STORAGE: Jurnal Ilmiah Teknik Dan Ilmu Komputer*, *1*(4), 1–11.
Nugrahantoro, A., Fadlil, A., & Riadi, I. (2020). Optimasi Keamanan Informasi Menggunakan Algoritma Advanced Encryption Standard (AES) Mode Chiper Block Chaining (CBC). *Jurnal Ilmiah FIFO*, *12*(1), 12.
Pradipta, A. A., Prasetyo, Y. A., & Ambarsari, N. (2021). Pengembangan Web E-Commerce Bojana Sari Menggunakan Metode Prototype. *EProceedings of Engineering*, *2*(1).
Purwanti, D. S., Fadli, M., Surono, M., & Susanto, E. R. (2025). Perancangan Penerapan Algoritma Kriptografi Aes 256 Untuk Keamanan Database Aplikasi Manajemen Siswa. *Storage: Jurnal Ilmiah Teknik Dan Ilmu Komputer*, *4*(2), 111–119.
Sari, I. P., Sulaiman, O. K., Al-Khowarizmi, A.-K., & Azhari, M. (2023). Perancangan Sistem Informasi Pelayanan Masyarakat pada Kelurahan Sipagimbar dengan Metode Prototype Berbasis Web. *Blend Sains Jurnal Teknik*, *2*(2), 125–134.
Senubekti, M. A., Dajoreyta, G. L., & Anggraini, N. (2024). Pembuatan desain UI/UX dengan metode prototyping pada aplikasi layanan Pengadilan Negeri Bale Bandung menggunakan Figma. *Jurnal Informatika Terpadu*, *10*(1), 1–10.
Set, F. M. C. B., Bana, C. M. N., Anunut, M. A., Da Costa, D., & Niis, Y. (2025). Penerapan Steganografi LSB dan Enkripsi AES untuk Keamanan Data Rahasia pada Gambar Digital. *Blantika: Multidisciplinary Journal*, *3*(7), 1040–1047.
Siswidiyanto, S., Wijayanti, D., & Haryadi, E. (2020). Sistem Informasi Penyewaan Rumah Kontrakan Berbasis Web Dengan Menggunakan Metode Prototype. *Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi Dan Komunikasi*, *15*(1), 16–23.