SAGA: Journal of Technology and Information Systems
Vol 2, Issue 1, February 2024, Page 175-189
ISSN: 2985-8933 (Media Online)
DOI: 10.58905/SAGA.vol2i1.250

Combination of AES (Advanced Encryption Standard) and SHA256 Algorithms for Data Security in Bill Payment Applications

Muhamad Rais Rabtsani[1], Agung Triayudi[2*], Gatot Soepriyono[3]

Informatics Department, Faculty of Communication and Information Technology, Universitas Nasional, Jakarta, Indonesia

Author Email: m.raisrabtsani@gmail.com[1], agungtriayudi@civitas.unas.ac.id[2], gatot@civitas.unas.ac.id[3]

Abstract. In the era of information technology development, digital payments and e-payments are becoming a dominant trend, supported by the crucial role of payment gateways such as Midtrans. Midtrans uses APIs to facilitate various online transactions, including debit cards. In this research, two cryptographic algorithms will be combined, namely Advance Encryption Standard (AES) with 256 bits and Secure Hash Algorithm (SHA) with 256 bits. The importance of data security in e-payments is recognized, with the application of cryptographic algorithms to protect sensitive transaction information. Yayasan Antero Prosesi Edukasi (YAPE) as an educational marketing consultant faced the challenge of time-consuming and inefficient manual payments. In an effort towards efficiency and security, YAPE plans to develop an online payment application with Midtrans Payment Gateway and the use of AES-256 and SHA-256 cryptographic algorithms. This step is expected to help the foundation keep up with technological developments, provide ease of payment, and achieve computerized efficiency. The results showed that the use of a combination of AES 256 bit and SHA256 encryption significantly increased security, making hacking attempts ineffective because encrypted data could not be accessed, with quite complicated calculation stages.

Keywords: Cryptography, Advance Encryption Standard, Secure Hash Algorithm, Payment Gateway, Midtrans, Payment System.

1 Introduction

In today's era, the development of information technology is transforming payment patterns to digital with the advent of e-payments. Payment gateways, such as those provided by Midtrans in Indonesia, play a crucial role in connecting customers, businesses, and banking institutions [1]. With the payment gateway, making payments online makes the transaction process faster, easier and more practical [2]. E-payment, or electronic payment, is a cashless payment system that uses an internet connection to connect a payment website with a third party's computer system, verifying the transaction [3]

Midtrans uses APIs to facilitate various payment transactions, including debit cards, credit cards, and cash withdrawals and remittances [4]. This change is driven by the need for a faster, easier, and more practical transaction process, with many millennials turning to digital payments. This change results in new payment patterns, more and more people especially millennials, are turning to digital payments. Year after year, technology continues to evolve, and organizations or companies utilize it to improve their performance in order to achieve their set goals [5].

In its application, e-payment requires data security considering that many cyber crimes target websites with high user traffic. Cyber crime refers to criminal acts that use computer technology, especially the internet, as the main tool to launch their criminal activities. One important aspect of securing information is protecting confidential and sensitive user transaction data.

The combination of Advanced Encryption Standard (AES) and SHA256 cryptographic algorithms is used to convert original data into encrypted data (encryption) and vice versa (decryption), allowing the processing of information or data in encrypted form and returning it to its original form [6][7]. Encryption processes such as Advanced Encryption Standard (AES) and SHA256 are implemented to protect data in e-payment systems, with AES operating as a symmetric algorithm that allows encryption of data into an unreadable ciphertext form and then decrypted back to plaintext for data recovery purposes [8][9].

SAGA: Journal of Technology and Information Systems
Vol 2, Issue 1, February 2024, Page 175-189
ISSN: 2985-8933 (Media Online)
DOI: 10.58905/SAGA.vol2i1.250

Yayasan Antero Prosesi Edukasi (YAPE) as an educational marketing consultant, management of independent classes (employees) and the Foundation's brand commonly called AEC-ACADEMY in further education both Undergraduate, Postgraduate, and Doctoral Programs. The foundation has collaborated with private universities in Jabodetabek, West Java, Banten, Central Java, East Java and international universities. Building a web-based payment application as a student bill payment process.

Yayasan Antero Prosesi Edukasi (YAPE), an educational marketing consultancy, offers independent class management services (employees) in advanced education for undergraduate, postgraduate, and doctoral programs. The foundation has collaborated with Private Universities (PTS) in Jabodetabek, West Java, Banten, Central Java, East Java and International Universities. Currently, the payment system is still manual, with students making transfers through the bank and sending proof of transfer via WhatsApp. Apart from the time and efficiency aspects, this manual approach also faces security issues, as the proof of transfer sent via messaging apps may be vulnerable to data interception or errors in recording due to its unencrypted nature.

Yayasan Antero Prosesi Edukasi does not currently have an application to facilitate online tuition payments. In their efforts to develop such an application, they want to implement a more flexible payment method, utilizing the Midtrans Payment Gateway which is capable of handling various types of common payments in Indonesia. The use of AES (Advanced Encryption Standard) and SHA256 algorithms in the payment system at the Antero Prosesi Edukasi Foundation is expected to help reduce the risk of crime related to the security of payment data.

Based on some of the problems above, to keep up with the development of information technology and meet the expectations of the Antero Prosesi Edukasi Foundation, the application of a combination of AES-256 and SHA-256 algorithms to the YAPE SPP payment system has been proven effective in protecting data, with AES-256 chosen as the main encryption option, and the use of hashed keys using SHA256. It shows a significant improvement in data security due to the high complexity of the calculations. This application is designed to assist students in an easy and secure payment process, by applying a combination of two cryptographic algorithms, namely Advance Encryption Standard (AES) with a key length of 256 bits and Secure Hash Algorithm (SHA) with a hash length of 256 bits with the aim of developing a computerized foundation and can contribute to improving the security of database systems.

2 Methodology

In this research, using the agile method, which is a software development method that allows improvements to be made while the system is active in short-term use [10]. The main reason for its use is its simplicity and ease of implementation because it is flexible and does not interfere with the performance of the running system. The following are the steps of system development regarding the Agile Method workflow as follows:

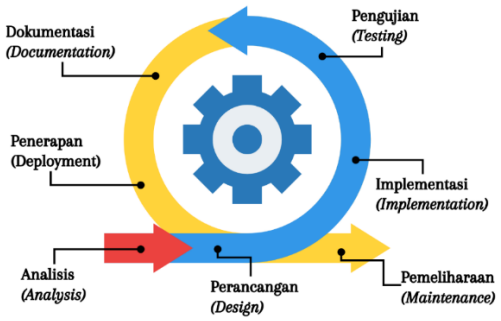

Figure 1. Agile Method

SAGA: Journal of Technology and Information Systems
Vol 2, Issue 1, February 2024, Page 175-189
ISSN: 2985-8933 (Media Online)
DOI: 10.58905/SAGA.vol2i1.250

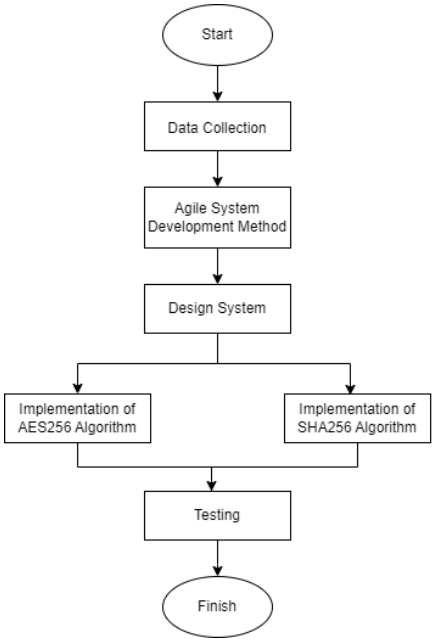

Figure 2. Research Stages

a. Data Collection Stage: this stage involves searching and collecting data directly from the object under study. The data will be used as a foundation for developing research applications.
b. Agile System Development Method: the selection and application of an adaptive and structured system development method, such as the Agile method. This step enables iterative development, allowing adjustments along the way.
c. Design System: the payment application with AES and SHA-256 involves requirements analysis, cryptographic algorithm selection, database design, user interface development, and cryptographic algorithm implementation. This ensures the security and integrity of transaction data, as well as optimal functionality for users.
d. Implementation phase: development using the Laravel framework as well as HTML, CSS, PHP, and JavaScript programming languages with the implementation of Advanced Encryption Standard (AES) and SHA256 algorithms to increase security by encrypting data. MySQL was chosen as the database management system (DBMS). In addition, the integration of this system requires midtrans payment gateway API.
e. Testing phase: involves manually validating the AES-256 and SHA-256 algorithms, by calculating the encryption and hashing results. Furthermore, the encryption speed was tested by measuring the time required for the data encryption process. The system satisfaction aspect was evaluated through user feedback regarding the usability, interface, and navigation of the application. Thus, the testing aimed to ensure the security, speed, and user satisfaction of the payment application.

2.1 Data Collection

2.1.1 Primary Data

Collecting data directly from the object under study through the following methods:

a. Observation Method: Involving direct observation of the payment data management process by conducting direct observation at the Antero Prosesi Edukasi Foundation.
b. Interview Method: Through interview techniques or direct dialog with the chairman of the Antero Prosesi Edukasi Foundation to obtain information related to payment methods that are still carried out manually.

2.1.1 Secondary Data

The information used in this research includes knowledge obtained by the author from various sources such as lecture materials, references such as journals, articles, books, and internet search results related to the research topic. The aim is to explore the theoretical principles that support this research.

SAGA: Journal of Technology and Information Systems
Vol 2, Issue 1, February 2024, Page 175-189
ISSN: 2985-8933 (Media Online)
DOI: 10.58905/SAGA.vol2i1.250

2.2 Advanced Encryption Standard-256 Algorithm

AES (Advanced Encryption Standard) is a cipher algorithm used to protect confidential data [6]. With its security advantages, AES replaced Data Encryption Standard (DES). The key size of AES can be 128 bits, 192 bits, or 256 bits, which affects the number of rounds in the encryption and decryption process. A comparison of the number of rounds and key length is given in Table 1, where AES-128 has 10 rounds, AES-192 has 12 rounds, and AES-256 has 14 rounds.

Tabel 1. Comparison of number of keys and rounds of AES

Type	Key Length	Block Length	Number of Rounds
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

The AES encryption process begins with the input of data into a "state", involving steps such as AddRoundKey, SubBytes, ShiftRows, MixColumns, and AddRoundKey that are repeated a certain number of rounds [11]. The encryption process diagram, as seen in Figure 3, provides a visual illustration of these steps. Encryption is the process of converting the original message into an encrypted message (ciphertext) [7].

$$C = E\ (M)$$

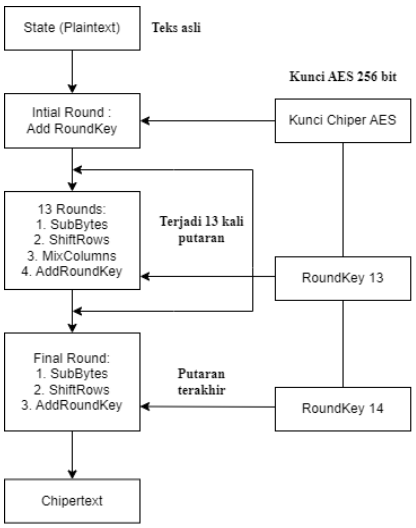

Figure 3. AES256 Algorithm Encryption Flowchart

The description process in the AES Algorithm aims to transform the encrypted message (ciphertext) back into the original message (plaintext) by using the description function M=D(C). The cipher transformation can be reversed, forming a process known as inverse cipher. The decryption process involves steps such as InvShiftRows, InvSubBytes, InvMixColumns, and AddRoundKey. The description process diagram in Figure 4 provides a visual illustration of these steps. The inversion of the encryption steps allows for easy and comprehensible recovery of the original data.

$$M = D(C)$$

SAGA: Journal of Technology and Information Systems
Vol 2, Issue 1, February 2024, Page 175-189
ISSN: 2985-8933 (Media Online)
DOI: 10.58905/SAGA.vol2i1.250

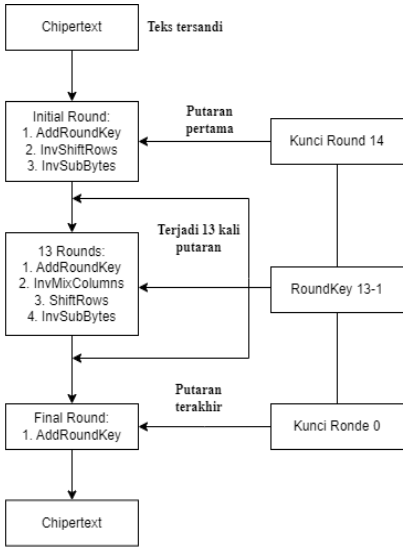

Figure 4. AES256 Algorithm Decryption Flowchart

2.3 CBC Mode of Operation

Cipher Block Chaining (CBC) is an encryption technique that involves processing the current bit block by combining the encrypted results of previous blocks. In CBC, a string of bits in the original text is divided into bit blocks of uniform length [12]. Cipher Block Chaining (CBC) mode involves the use of an Initialization Vector (IV) that is combined with the first plaintext and the previous ciphertext block, becoming the IV of the next block. This process includes an XOR operation between the current plaintext block and the previous encrypted ciphertext block, with the result being used as input for the encryption function. In Cipher Block Chaining (CBC) mode, each ciphertext block depends on the current plaintext block and all previous plaintext blocks. In the decryption process, the current ciphertext block becomes the input for the decryption function and is XORed with the previous ciphertext block. The previous ciphertext block acts as feedback in the final stage of decryption, providing an additional level of security.

CBC encryption: $C_i = E_k(p_i \oplus C_{i-1}), C_0 = IV$
CBC decryption: $P_i = D_k(C_i) \oplus C_{i-1}, C_0 = IV$

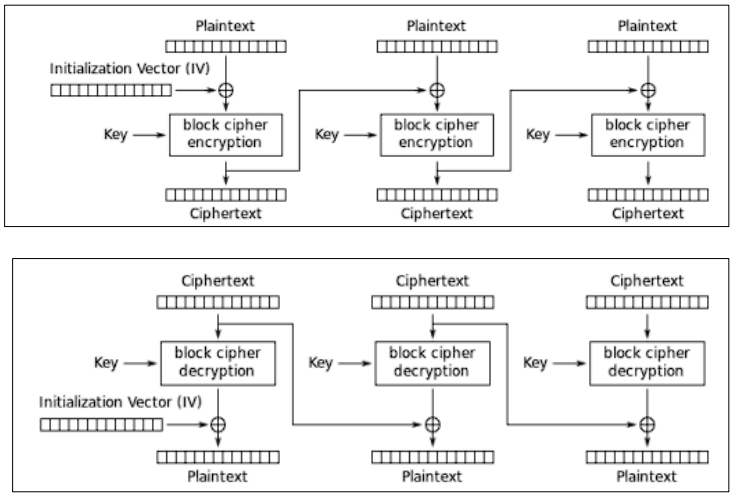

Figure 5. Cipher Block Chining (CBC) Mode Encryption-Decryption

2.4 SHA256 Algorithm

The Secure Hash Algorithm (SHA), developed by the National Institute of Standards and Technology (NIST), follows similar principles to MD4 and was formalized as a Federal Information Processing Standard

SAGA: Journal of Technology and Information Systems
Vol 2, Issue 1, February 2024, Page 175-189
ISSN: 2985-8933 (Media Online)
DOI: 10.58905/SAGA.vol2i1.250

(FIPS 180) in 1993 [13]. SHA is a one-way hash function that compresses messages of any size into a fixed message summary. The second generation, SHA256, is a highly secure one-way hash function that makes it impossible to recover the original message from the hash value. Hash functions, such as SHA-256, play a crucial role in cryptography by ensuring the security integrity of messages. To solve a SHA256 hash value requires 2^256 attempts, performed over 64 rounds, with non-linear functions, cyclic rotations, and round constants. The SHA256 architecture describes the steps of message processing, including message input, padding, expansion, scheduling, compression, to produce a 256-bit hash value.

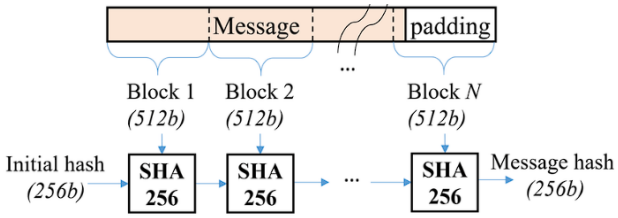

Figure 6. SHA256 Hashing Process

Figure 6 of the SHA256 architecture shows several critical stages, including Message Padding for message length adjustment, Message Expansion to determine the original message length before padding, Message Scheduler to schedule message block processing, and Message Compression to generate a 256 bit message digest after 64 rounds. During the processing cycle, variables and initial values change dynamically, providing high security against hash value recalculation attempts [14].

2.5 Design System

2.5.1 Architecture Design System

The tuition fee payment system architecture has a web browser software that functions as a client, the Laravel framework acts as a server, midtrans is used as a payment gateway, and MySQL functions as a database management. The role of APIs is to mediate between different applications, either within the same platform or across platforms [15]. Ilustrasi arsitektur sistem pembayaran biaya pendidikan ini dapat dilihat dalam Gambar 7.

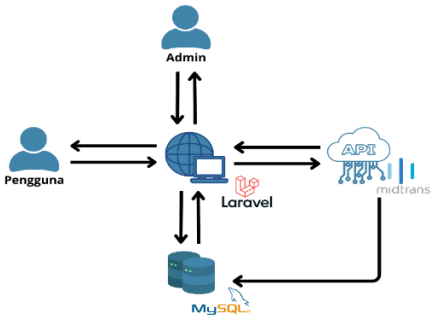

Figure 7. Architecture Design System

2.5.2 Use Case Diagram

Use cases are used to describe the functional requirements of the system in the form of interaction diagrams that show how various actors interact with the system being developed. In use cases, we can identify the functions that will exist in the application being created. This system allows Admins to login, create expense reports, edit billing data, and use AES 256 and SHA256 encryption on billing data, while Users can login, make payments, and log out of the system.

SAGA: Journal of Technology and Information Systems
Vol 2, Issue 1, February 2024, Page 175-189
ISSN: 2985-8933 (Media Online)
DOI: 10.58905/SAGA.vol2i1.250

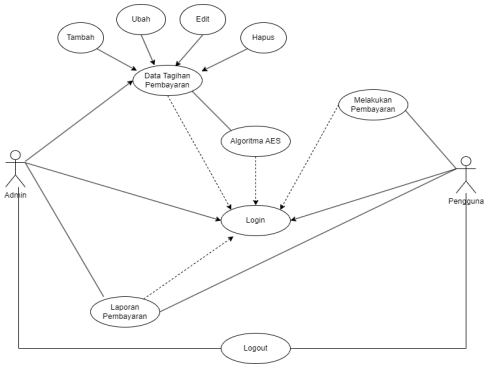

Figure 8. Use Case Diagram

2.5.3 Activity Diagram1

Activity Diagram is a visual representation of a series of tasks or activities that occur in a running system. This diagram helps describe the workflow of the system so that users or program developers can understand how the processes in the system run and are interconnected [16]. The user visits the main page, selects the bill, chooses the payment method, views the payment details, presses "pay now," the system connects with Midtrans, the user follows the payment instructions on the Midtrans page, and after the payment is completed, a successful payment notification is received.

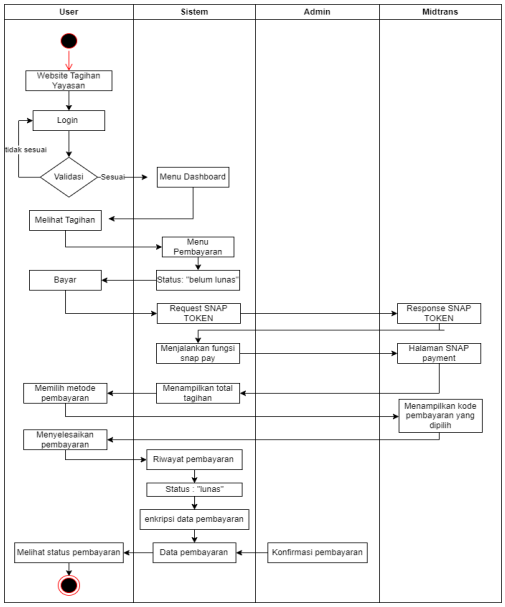

Figure 9. Activity Diagram

2.5.4 Class Diagram

The following in Figure 10 is a class diagram of the fee payment system at the Antero Prosesi Edukasi Foundation. The class model in a system has attributes and behaviors that are responsible for the administrative management of the system. Meanwhile, the controller class has attributes and behaviors that are responsible for running certain methods or functions in the system.

SAGA: Journal of Technology and Information Systems
Vol 2, Issue 1, February 2024, Page 175-189
ISSN: 2985-8933 (Media Online)
DOI: 10.58905/SAGA.vol2i1.250

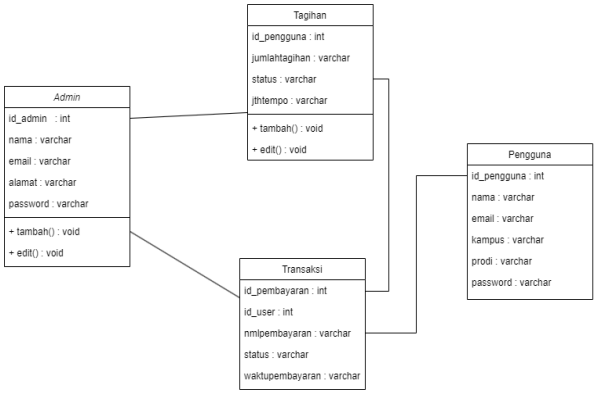

Figure 10. Class Diagram

3 Results and Discussion

In this step, researchers will analyze the steps that the product will run during the encryption and decryption process. This analysis aims to compile the steps that the program will run based on the initial analysis. Based on the theories contained in the Advanced Encryption Standard and SHA256 cryptographic algorithms, the following algorithm is obtained:

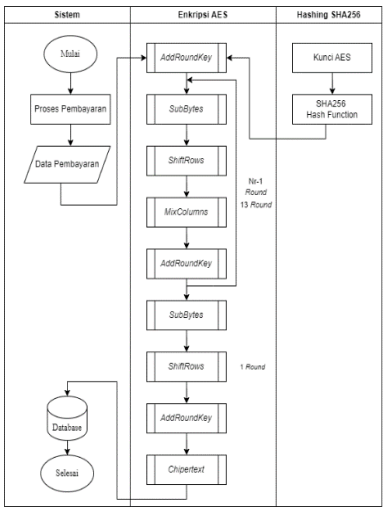

Figure 11. Flowchart of AES and SHA256 Algorithm Encryption Process

3.1 Simulation of Encryption Process of AES-256 and SHA256

In this research, Advanced Encryption Standard (AES) is applied to protect the payment database in the bill payment application. When a bill is entered, the system automatically encrypts the data. After the encryption process is successful and the data is verified, access to the website is allowed. To start the encryption step, two sets of 8x4 matrix arrays called Plaintext and Key are created.

3.1.1 Determine the Plaintext, AES Key, and Initial Vector (IV)

Table 2. Plaintext, AES Key, dan IV

Plaintext: muhamadrais
AES Key: passwordsppaes256private12345678 IV : 000000000000000

SAGA: Journal of Technology and Information Systems
Vol 2, Issue 1, February 2024, Page 175-189
ISSN: 2985-8933 (Media Online)
DOI: 10.58905/SAGA.vol2i1.250

3.1.2 Perform Hash Function with SHA256 Algorithm on a 256 Bit AES Key

The next stage will be a hash process with an aes key of 256 bits through the hash function in the SHA256 algorithm so that it cannot be read and is much more secure. The following is an example of the hash result on the key:

AES Key: passwordsppaes256private12345678

Convert the message into a binner:
[112, 97, 115, 115, 119, 111, 114, 100, 115, 112, 112, 97, 101, 115, 50, 53, 54, 112, 114, 105, 118, 97, 116, 101, 49, 50, 51, 52, 53, 54, 55, 56]
01110000011000010111001101110011011101110110111011100100110010001110011011100000111000001
10000101100101011100110011001000110101001101100111000001110010011010010111011001100001011
01000110010100110001001100100011001100110100001101010011011000110111001110000

Add 1 number at the end and padding the code with zeros until the data is 512bit: (512 bits)
01110000011000010111001101110011011101110110111011100100110010001110011011100000111000001
10000101100101011100110011001000110101001101100111000001110010011010010111011001100001011
01000110010100110001001100100011001100110100001101010011011000110111001110000 1
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000100000000

The padding process resulted in 64 words remaining in the message schedule (w):
W47: 10000111011110001001101101000001
W48: 00001001100101111000001111100010
W49: 00100101111000110011010001001010
W50: 01110000011100000101110000010010
W51: 00111010110100010001000111101011
W52: 10000011100011010011000111111111
W53: 10111101011111111100110111100101

From the message above, get the result of the final hash
a = 11011011011101011010000001001010 = db75a04a
b = 11111110110110111100000000001011 = fedbc00b
c = 00110110000101010101011101011011 = 3615575b
d = 01100001111111000011100110001010 = 61fc398a
e = 00100011000110110001010110111101 = 231b15bd
f = 00110011010110110010101110011110 = 335b2b9e
g = 00110110111001111001001001111110 = 36e7927e
h = 00110110110011010001000111100110 = 36cd11e6

SHA256: db75a04afedbc00b3615575b61fc398a

3.1.3 AES256 encryption process

The Plaintext and Key are converted into bit representations using ASCII codes. The bit representation is converted into hexadecimal format. Next, each hexadecimal part is placed into each cell of the 4x4 matrix and the key is 8x4. This results in an end result where each cell in the matrix is filled with the hexadecimal value derived from the ASCII conversion of the text and key.

Table 3. Plaintext dan AES Key

Plaintext				AES Key							
6D	6D	61	61	64	61	66	63	33	35	36	33
75	61	69	62	62	30	65	30	36	37	31	39
68	64	73	74	37	34	64	30	31	35	66	38
61	72	72	73	35	61	62	62	35	62	63	61

SAGA: Journal of Technology and Information Systems
Vol 2, Issue 1, February 2024, Page 175-189
ISSN: 2985-8933 (Media Online)
DOI: 10.58905/SAGA.vol2i1.250

Before moving on to the next stage, the process of creating a key schedule is done by performing an XOR operation. This process generates a series of round keys called Roundkeys at the encryption stage. After getting 14 Roundkeys from 1 to 14, the encryption process continues by using these keys in each round of encryption. The steps that need to be executed are as follows:

In the AddRoundKey (Initial State) process, the key is combined with the PlainText. This merging process is done by XOR operation on each byte of the key with each byte of the original text (PlainText). So the result is as follows:

Table 4. Process Result AddRoundKey

Plaintext					AES Key					AddRoundKey Result			
6D	6D	61	61	XOR	64	61	66	63		09	0C	07	02
75	61	69	62	⊕	62	30	65	30	=	17	51	0C	52
68	64	73	74		37	34	64	30		5F	50	17	44
61	72	72	73		35	61	62	62		54	13	10	11

Plaintext XOR AESKey
$6D = 0110\ 1101$
$64 = 0110\ 0100$
$0000\ 1001$
$= 09\ (1x1)$

The SubBytes process is a process in the AES algorithm where the byte values from the AddRoundKey result (initial state) are replaced with the byte values in the SBox table. For example, if we want to replace element 09 in the initial state. Then, the corresponding element in SBox is in the 0th row and 9th column, which in this case results in the value 01. The output can be observed as follows:

Table 5. Process SubBytes

09	0C	07	02		01	FE	C5	77
17	51	0C	52		F0	D1	FE	00
5F	50	17	44	→	CF	53	F0	1B
54	13	10	11		20	7D	CA	82

The ShiftRows stage involves shifting the bytes in each row of the Plaintext table. In this process, byte rotation occurs in the last rows, namely rows 1, 2, and 3, with a different number of rotations. Row 1 will experience one rotation, row 2 two rotations, and row 3 three rotations. Meanwhile, row 0 remains unchanged. The result of this step reflects the change in the order of bytes in each row, creating a new layout.

Table 6. ShiftRows Process

01	FE	C5	77		01	FE	C5	77
F0	D1	FE	00		D1	FE	00	F0
CF	53	F0	1B	→	F0	1B	CF	53
20	7D	CA	82		82	20	7D	CA

The MixColumns process involves a matrix multiplication operation that uses vector coordinate multiplication. Where the results after transformation are r0, r1, r2, and r3. After the data has gone through the substitution process in Sbox, a0-a3 can be obtained from the matrix.

Table 7. MixColumns Result

Hasil *MixColumns*			
18	C5	23	7C
31	14	F2	B3
B6	56	C7	64
3D	BC	61	B5

SAGA: Journal of Technology and Information Systems
Vol 2, Issue 1, February 2024, Page 175-189
ISSN: 2985-8933 (Media Online)
DOI: 10.58905/SAGA.vol2i1.250

In the AddRoundKey (Round 1) process, the key (Key) is merged with the original text (plaintext) which is the result of the MixColumns calculation. In this AddRoundKey process, the key used is taken from the key schedule (round 1) that has been prepared in the previous stage, namely step 3. The key combination is done in a similar way as in step 4, namely through an XOR operation on each bit of the key with each bit of the MixColumns result.

Table 8. Proses AddRoundKey (Round 1)

MixColumns Result					Key Schedule (Round 1)					AddRoundKey Result (Round 1)			
18	C5	23	7C	XOR	61	00	66	05	=	79	C5	45	79
31	14	F2	B3	⊕	66	56	33	03		57	42	C1	B0
B6	56	C7	64		9D	A9	CD	FD		2B	FF	0A	99
3D	BC	61	B5		CE	AF	CD	AF		F3	13	AC	1A

Re-do the SubBytes, ShiftRows, MixColumns, and AddRoundKey process 13 times to generate the output from round 2 to round 13. Then, to get the result of the 14th round, repeat the SubBytes, ShiftRows, and AddRoundKey steps (final state) without involving the MixColumns step. The ciphertext generated from the AES algorithm encryption process can be observed as follows:

Table 9. Hasil AddRoundKey 14 (Final State)

AddRoundKey Result 14 (Final State)			
98	f9	3c	8a
e3	7a	00	b9
32	ad	91	7a
fd	ee	22	a8

After the encryption process is complete, the results are then converted back into text using the ASCII table. The conversion resulted in the text as seen below:

Table 10. Chipertext Conversion Result

Chipertext AddRoundKey 14 (Final State)			
98	f9	3c	8a
e3	7a	00	b9
32	ad	91	7a
fd	ee	22	a8

So the ciphertext resulting from the encryption process using AES 256 CBC is 98e332fdf97aadee3c0091228ab97aa8. The ciphertext is encrypted again using Base 64.
Chipertext (Base 64): mOMy/fl6re48AJEiirl6qA==

3.2 Implementation of AES-256 and SHA256

3.2.1 Encryption

The following is an implementation of the coding script to encrypt for data security using the AES-256-CBC and SHA256 algorithms.

SAGA: Journal of Technology and Information Systems
Vol 2, Issue 1, February 2024, Page 175-189
ISSN: 2985-8933 (Media Online)
DOI: 10.58905/SAGA.vol2i1.250

```php
$payment = Payment::create([
    'payment_id' => $this->encrypt('PYM' . time()),
    'user_id' => $this->encrypt($request->user_id),
    'academic_year_id' => $this->encrypt($request->academic_year_id),
    'academic_year_status' => $this->encrypt($request->academic_year_status),
    'semester' => $this->encrypt($request->semester),
    'amount' => $this->encrypt($request->amount),
    'fine' => $request->fine ? $this->encrypt($request->fine) : null,
    'discount' => $request->discount ? $this->encrypt($request->discount) : null,
    'type' => $this->encrypt($request->type),
    'description' => $request->description ? $this->encrypt($request->description) : null,
    'status' => $this->encrypt('unpaid'),
]);
```

Figure 12. Data Encryption Code in Paymentcontroller

```php
public function encrypt($text)
{
    $method = 'AES-256-CBC';
    $aesKey = 'passwordsppaes256private12345678';
    $option = 0;
    $hashKey = substr(hash('sha256', $aesKey, true), 0, 16);
    $keyEncode = bin2hex($hashKey);
    $iv = str_repeat("0", openssl_cipher_iv_length($method));

    return openssl_encrypt($text, $method, $keyEncode, $option, $iv);
}
```

Figure 13. Code Encryption Algorithms AES256 and SHA256

3.2.2 Decryption

The following is an implementation of the coding script to decrypt the data to become the original text using the AES-256-CBC and SHA256 algorithms.

```php
103 references | 0 overrides
public function decrypt($encryptedHex)
{
    $method = 'AES-256-CBC';
    $aesKey = 'passwordsppaes256private12345678';
    $option = 0;
    $hashKey = substr(hash('sha256', $aesKey, true), 0, 16);
    $keyEncode = bin2hex($hashKey);
    $iv = str_repeat("0", openssl_cipher_iv_length($method));

    return openssl_decrypt($encryptedHex, $method, $keyEncode, $option, $iv);
}
```

Figure 14. Code decryption algorithm AES256 and SHA256

3.3 Review of AES-256-CBC and SHA256 Algorithms on the System

In this study, researchers used a combination method of the Advanced Encryption Standard cryptographic algorithm with a key length of 256 bits and SHA256. This method aims to secure data in the YAPE SPP payment application database. The data security in question is payment data security. The following is a display of the original data on one of the menus contained in the payment application with data that has been secured in the database.

3.3.1 Admin Payment Menu View dan Database

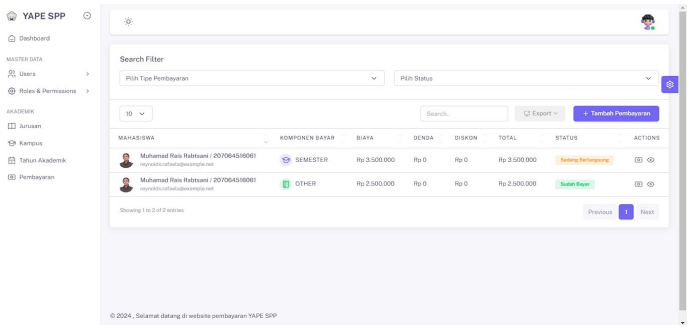

Figure 14. Admin Payment Menu View

SAGA: Journal of Technology and Information Systems
Vol 2, Issue 1, February 2024, Page 175-189
ISSN: 2985-8933 (Media Online)
DOI: 10.58905/SAGA.vol2i1.250

On the admin payment menu in this YAPE SPP application which displays a list of all payments that have been successfully validated by the admin. The menu has several columns, namely payment_id, user_id, academic year, semester, amount, discount, type_payment, payment date and status. Payment data contained in the payment list menu is in the form of plaintext or readable words. In contrast to what is contained in the database, payment data is in the form of ciphertext or words that cannot be read, this is because the payment data contained in the database has been encrypted using the AES-256-CBC and SHA256 algorithms. The following is an image of the database.

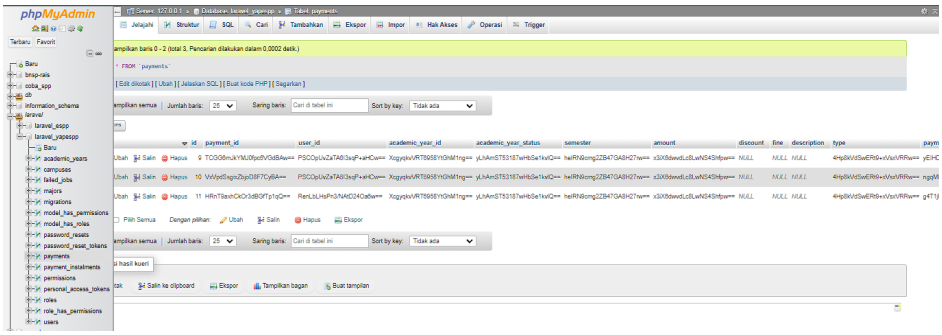

Figure 15. Encrypted payment database

3.3.2 User Payment Menu View

On the user payment menu in this YAPE SPP application which displays a list of all payments. The menu has several columns, namely name, component, cost, discount, total, status, and payment method. Payment data contained in the payment list menu is in the form of plaintext or readable words that have been decrypted.

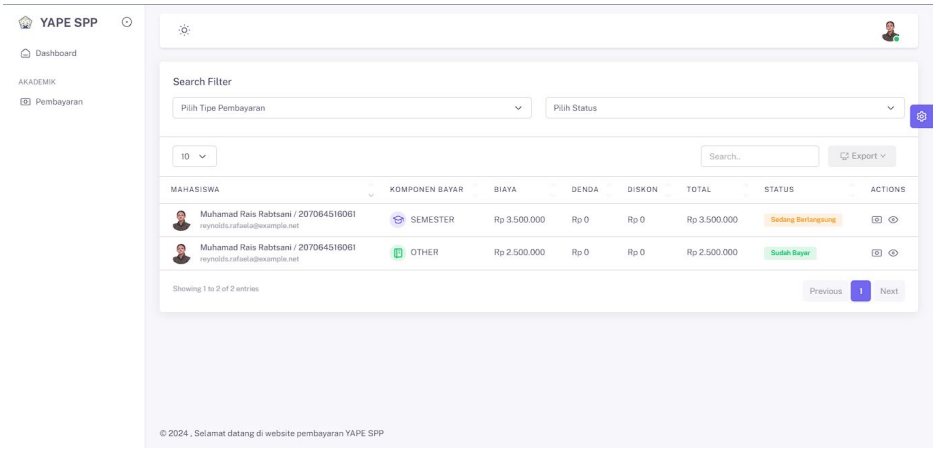

Figure 16. User Payment Menu View

3.4 Testing the Results of Encryption and Decryption of Payment Data

After the application goes through the validation stage and is declared feasible / good to use as security in the payment application, then the application is tested by implementing the algorithm in the YAPE SPP payment application to get the results of the suitability between the algorithm and the system. The results of the Advanced Encryption Standard and SHA256 algorithm trials are as follows:

Table 11. Testing the Results of Encryption and Decryption of Payment Data

Data	Decryption Result	Encryption Result	Description
payment_id	PYM1707063094	HRnT9axhCkOr3dBGfTp1qQ==	Success
user_id	13 (Muhamad Rais Rabtsani)	hy6g31rxi68E7EFGJ1qYZeMij6 WC1IdwByDeDal1UXg=	Success
year_academic	2023/2024	kem3n48m85rYXKcUjaYzdA==	Success

SAGA: Journal of Technology and Information Systems
Vol 2, Issue 1, February 2024, Page 175-189
ISSN: 2985-8933 (Media Online)
DOI: 10.58905/SAGA.vol2i1.250

status_academic	ganjil	yLhAmST53187wHbSe1kvlQ==	Success
semester	3	heIRN9cmg2ZB47GA8H27rw==	Success
amount	3500000	x3iX6dwvdLc8LwNS4Shfpw==	Success
discount	*NULL*	*NULL*	*NULL*
type_payment	semester	4Hp8kVdSwERt9+xVsxVRRw==	Success
date_payment	2024-02-04 16:11:43	g4T1jMfHYr/5GirTTzQVs6iCBh+J7mWD1B1g6EuB9zs=	Success
payment_method	instalment	YBztLHizJTvka5Nkk1ToNQ==	Success
status	in_progress (Sedang Berlangsung)	GmCTmIzBwif4CTDVI80d/A==	Success

Based on the test results, the algorithm runs as expected so that the algorithm can be implemented on the system without any problems. Thus, the slightest change in the plaintext or key has a significant impact on the chipertext result of the encryption process. Therefore, the AES algorithm proved to be very effective in maintaining data security in this payment bill application.

3.5 System Feasibility Test

Based on the results of the system feasibility test conducted by students and officers, a percentage of 92% was obtained. With this percentage, the system is feasible to use.

Table 12. System Feasibility Test Results

Eligibility Aspects						
No	Respondent Name	System Performance	System Efficiency	System Information	System Control	System Services
		Angka Penilaian				
1	Dewi Febrianti	4	5	5	4	4
2	Dewi Cahya Novita Sari	5	4	5	5	4
3	Arif Yogi	5	5	4	5	5
Total		14	14	14	14	13
Maximum Number		15	15	15	15	15
Percentage		93,33%	93,33%	93,33%	93,33%	86,67%
Average Percentage				92,00%		

4 Conclusion

Based on the test results of the encryption system implemented in the YAPE SPP payment system application using the Advanced Encryption Standard (AES) 256 and SHA256 algorithms, it can be concluded that the AES algorithm is suitable for application in securing data contained in the system/database. The encryption process is carried out to produce data that is not easy to read and not easily hacked by others unless they have the key to decrypt the data. The keys used to encrypt and decrypt are encrypted with SHA256 implementation, so they are not easy to read. The feasibility test of the system involved students and officers with a percentage of 92%, indicating that the system passed the feasibility test and was considered feasible to use. The test results also show that the use of a combination of 256 bit AES and SHA256 encryption significantly increases security, making hacking attempts ineffective because encrypted data cannot be accessed, with quite complicated calculation stages.

References

[1] Alfian, P. Sokibi, and L. Magdalena, “Penerapan Payment Gateway pada Aplikasi Marketplace Waroeng Mahasiswa Menggunakan Midtrans,” *Jurnal Informatika Universitas Pamulang*, vol. 5, no. 3, p. 387, Sep. 2020, doi: 10.32493/informatika.v5i3.6719.

[2] T. R. Saputro and J. Sutopo, “PENERAPAN PAYMENT GATEWAY SEBAGAI SISTEM VERIFIKASI PEMBAYARAN PADA WEBSITE PEMESANAN PAKET WISATA,” 2020.

[3] I. Fauzi and I. H. Ikasari, "Rancang Bangun Penerapan Teknologi Aplikasi Payment Gateway pada Sistem Pembayaran Berbasis Web (Studi Kasus : Toko Bandar Aki)," *Jurnal Informatika MULTI*, vol. 1, no. 3, 2023.

[4] Y. Wiharto and Mufti, "Implementasi Advanced Encryption Standard 128 Sebagai Pengamanan Basis Data Obat-obatan Apotek," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 8, no. 2, Aug. 2022, doi: 10.28932/jutisi.v8i2.4817.

[5] L. G. I. Maharani, W. H. N. Putra, and W. Purnomo, "Pengembangan Aplikasi Penjualan Toko Wulan berbasis Web menggunakan Api Midtrans sebagai Payment Gateway," 2022. [Online]. Available: http://j-ptiik.ub.ac.id

[6] M. R. Andriyanto and P. Sukmasetya, "Penerapan Algoritma Advanced Encryption Standard (AES) Untuk Keamanan Data Transaksi Pada Sistem E-Marketplace," *Journal of Computer System and Informatics (JoSYC)*, vol. 4, no. 1, pp. 179–187, Dec. 2022, doi: 10.47065/josyc.v4i1.2451.

[7] L. Sodikin and T. Hidayat, "Analisa Keamanan E-Commerce Menggunakan Metode AES Algoritma," *TEKNOKOM*, pp. 8–13, 2020, doi: https://doi.org/10.31943/teknokom.v3i2.46.

[8] A. Dharmawan and H. Munandar, "PENERAPAN ALGORITME KRIPTOGRAFI SHA-256 DAN AES-256 UNTUK PENGAMANAN FILE PADA PT PELANGI SENTRAL KREASI," 2023.

[9] H. Wulandari, J. Karman, and Elmayati, "IMPLEMENTASI ALGORITMA AES (ADVANCED ENCRYPTION STANDARD) PADA PENJUALAN ALAT-ALAT ELEKTRONIK BERBASIS WEB (STUDI KASUS TOKO SONIA ELEKTRONIK)," 2023.

[10] Y. Fatman, N. K. Nafisah, and P. B. J. Pambudi, "Implementasi Payment Gateway dengan Menggunakan Midtrans pada Website UMKM Geberco," *Jurnal KomtekInfo*, pp. 64–72, Jun. 2023, doi: 10.35134/komtekinfo.v10i2.364.

[11] I. Fitriani and A. B. Utomo, "Implementasi Algoritma Advanced Encryption Standard (AES) pada Layanan SMS Desa," *JISKA (Jurnal Informatika Sunan Kalijaga)*, vol. 5, no. 3, pp. 153–163, Nov. 2020, doi: 10.14421/jiska.2020.53-03.

[12] A. Fathurrozi, "Penerapan Algoritma Advanced Encryption Standard (AES-256) Dengan Mode CBC DanSecure Hash Algorithm (SHA-256) Untuk Pengamanan Data File," *Journal of Information and Information Security (JIFORTY)*, vol. 2, no. 2, pp. 227–238, 2021, [Online]. Available: http://ejurnal.ubharajaya.ac.id/index.php/jiforty

[13] T. Abdurrachman and B. R. Suteja, "Pengembangan Sistem Informasi Asosiasi Jasa Konstruksi dengan Menerapkan Tanda Tangan Digital," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 7, no. 1, Apr. 2021, doi: 10.28932/jutisi.v7i1.3431.

[14] R. Oktafiani, E. I. H. Ujianto, and R. Rianto, "Kombinasi Algoritma Kriptografi Vigenere Cipher dan SHA256 untuk Keamanan Basis Data," *Jurnal Sistem Komputer dan Informatika (JSON)*, vol. 4, no. 3, p. 433, Mar. 2023, doi: 10.30865/json.v4i3.5583.

[15] V. H. Pranatawijaya and H. Yulianto, "Penerapan API (Application Programming Interface) MIDTRANS Sebagai Payment Gateway Pada Indekos Berbasis Website," 2022.

[16] M. M. A. F. Prawiranegara and I. G. L. P. E. Prismana, "Rancang Bangun Aplikasi Equity Crowdfunding Syariah untuk Usaha Mikro Kecil Menengah berbasis Website menggunakan Payment Gateway Midtrans dengan Framework Laravel," *JEISBI*, vol. 02, pp. 102–110, 2021.