Pelita Tech: Journal of Computing and Information Systems
Vol. 1, Issue. 1
E.ISSN: XXXX-XXXX
March, 2026.

# Implementation of Base64-Based Data Encoding in a Web Based Sharia Cooperative Information System to Support Customer Data Security

Anita Yustivasari [1], Rifqi Rahmatika Az-Zahra[2*]
[1,2]Informatics Engineering, Universitas Muhammadiyah Ponorogo, Indonesia
[*]Corresponding author: rifqirahmatika@umpo.ac.id



**Received:** 10 February 2026
**Revised:** 23 February 2026
**Accepted:** 12 March 2026
**Available online:** 25 March 2026

***Keywords:***
*Sharia cooperative; information security; Base64 encoding; web-based information system; data protection; financial information systems*

**ABSTRACT**

The digital transformation of financial services has encouraged sharia cooperatives to adopt web-based information systems to improve efficiency, data management, and service quality. However, the increasing use of digital systems also raises concerns regarding the protection of sensitive customer information. This study aims to examine the implementation of the Base64 algorithm in a web-based sharia cooperative information system and to evaluate its role in improving customer data security. The research focused on applying Base64 encoding to data processed through the Create, Read, Update, and Delete (CRUD) functions of the system. System testing was conducted by comparing the original input data displayed on the user interface with the encoded data stored in the database. The results show that customer-related data, including member, user, savings, and financial transaction records, were successfully transformed into encoded strings before storage. This process reduced the direct readability of sensitive information in the database and provided an additional layer of protection at the application level. Furthermore, the implementation was integrated consistently across multiple system modules without disrupting the main functions of the application. These findings indicate that Base64 can support basic data protection in a sharia cooperative system by improving the handling of stored information. Nevertheless, Base64 should be regarded as a supporting data-encoding mechanism rather than a complete cryptographic solution. Future studies are therefore recommended to combine Base64 with stronger encryption methods to achieve a higher level of security for confidential financial data.

## 1. Introduction

In the current landscape of the global digital economy, Sharia financial institutions particularly Sharia cooperatives are undergoing a rapid transformation to meet the demand for efficient, community-based financial services. The integration of financial technology (Fintech) has reshaped operational business models, shifting from manual processes to automated digital systems to enhance financial inclusion and service quality [1]. However, this digital leap creates a paradoxical situation where the convenience of accessibility is often offset by heightened vulnerabilities in information security. In a digitized environment, institutional records and sensitive customer data become high-value targets for sophisticated cyber threats, including ransomware and structured data breaches [2].

Despite the clear necessity for digital adoption, a significant research gap exists regarding the practical implementation of data protection layers in small-to-medium Sharia cooperatives. While large-scale Islamic banks have the capital to implement complex cryptographic infrastructures, many web-based cooperative systems still suffer from "plain-text vulnerability," where sensitive financing records and customer identities are stored or transmitted in a directly readable format [3]. Current literature emphasizes that in Sharia-based institutions, security is not merely a technical requirement but a fundamental pillar of institutional integrity (amanah) and customer trust. A breach in data confidentiality

directly correlates to a loss of public confidence, which is much harder to recover than financial assets [4], [5].

This study addresses the aforementioned gap by proposing a pragmatic security-oriented data-handling strategy. While advanced encryption methods are standard, there is a critical need for efficient obfuscation layers that can be seamlessly integrated into the application workflow of web-based cooperative systems. The novelty of this research lies in the strategic implementation of Base64 encoding as a primary obfuscation mechanism to mitigate direct data readability during transmission and storage. By transforming sensitive financing records and transaction documents into an encoded format, the system adds a layer of protection against unauthorized casual observation and accidental data exposure [5]. Therefore, the objective of this research is to design and implement a web-based Sharia cooperative information system that utilizes Base64 to improve the security posture and protection of customer information.

## 2. Literature Review

### 2.1 Information Security and the Urgency of Cryptography

Information security constitutes a strategic pillar within modern digital infrastructure, essential for safeguarding data assets against increasingly sophisticated cyber threats. Information systems managing sensitive data entities are highly vulnerable to attacks that can directly compromise the confidentiality, integrity, and availability of such data [6]. Data breach incidents within network architectures frequently result not only in severe financial losses but also in the catastrophic degradation of institutional reputation in the public eye [2]. Consequently, the implementation of cryptographic mechanisms has been widely recognized and adopted as the most effective industrial operational standard to prevent unauthorized system access [7]. Through encryption, computing systems are capable of providing a mathematical layer of protection, ensuring that data intercepted during transmission remains encrypted and incomprehensible to malicious actors [8].

### 2.2 Symmetric and Asymmetric Cryptographic Approaches

Cryptographic mechanisms are comprehensively classified into symmetric and asymmetric approaches, each offering specific functionalities within database security architectures. Symmetric cryptography utilizes a single, identical secret key for both the encryption and decryption processes, thereby offering high computational efficiency for continuously processing large volumes of data [9]. Conversely, asymmetric cryptography employs a public and private key pair to mitigate the vulnerabilities associated with key distribution that frequently occur in open communication networks [3]. To maintain the effectiveness of this protection, framework guidelines such as the Open Web Application Security Project (OWASP) recommend rigorous key governance to ensure that private keys are never exposed within the application's source code [10]. In modern software implementation, the amalgamation of these two methods is frequently deployed as a hybrid cryptographic system to achieve an optimal balance between processing speed and the security of data exchange [11].

### 2.3 Base64 Encoding and Obfuscation Mechanisms

The Base64 algorithm is a structural data encoding technique that serves an essential function in converting raw binary data into a secure ASCII text format for transmission. Architecturally, Base64 operates by segmenting binary data sequences into 6-bit blocks, which are subsequently mapped sequentially onto 64 standard, printable ASCII alphabetical characters [5], [12]. This text encoding is highly vital in web-based computing environments as it guarantees that file integrity remains uncorrupted when traversing HTTP protocols, which generally only support plain text transmission [13], [14]. Although scientifically Base64 does not constitute an independent cryptographic encryption algorithm, its function as an obfuscation technique is highly effective in concealing the original plaintext structure from casual observation [15]. Therefore, network security research recommends integrating Base64 with genuine encryption algorithms to construct a multi-layered data protection security scheme at the database level [13].

### 2.4 Digital Transformation and Data Security in Sharia Cooperatives

Currently, Sharia cooperatives are confronted with the imperative of digital transformation to enhance operational efficiency and the quality of community-based financial services. The adoption of digital information systems enables Sharia microfinance institutions to expand financial inclusion, automate member data recording, and accelerate financing transaction workflows [15]. Nevertheless, the migration from manual record-keeping systems to web-based architectures exposes customer identity data and financial histories to the risk of destructive cyber exploitation [16]. In the context of Islamic financial institutions, the protection of customer data privacy is not merely an instrument of compliance with banking regulations; rather, it is an integral component of the amanah (trustworthiness) principle designed to preserve institutional integrity and public trust [15]. Taking these considerations into account, the implementation of compressed and efficient data security protocols, such as the integration of the Base64 method, becomes an absolutely necessary strategic element in the daily operations of Sharia cooperatives [15].

## 3. Method

### 3.1 Data Collection Methods

This research employs a systematic approach to gather the necessary data for implementing the Base64 algorithm within a Sharia cooperative information system, primarily aimed at enhancing customer data security. The data collection process consists of two main phases. First, Literature Review, this phase involves a comprehensive examination of relevant textual sources, including academic journals and technical documentation. The objective is to extract, synthesize, and construct a strong theoretical foundation regarding data security and cryptographic encoding, which supports the formulation of the research implementation.

Then interviews, to obtain empirical and technical insights, direct communications were conducted with experts in the field of algorithmic security, specifically focusing on the Base64 algorithm. These interactive sessions yielded an in-depth understanding of the algorithm's operational mechanics, performance metrics, and best practices for integration.

### 3.2 System Development Methodology

The implementation of the Base64 algorithm into the Sharia cooperative system follows the Waterfall methodology. This model was selected because it provides a clear, sequential, and highly structured framework for software development. As shown in Figure 1 The research follows a sequential methodology beginning with the Analysis phase, where initial vulnerabilities are identified and specific requirements for integrating the Base64 algorithm to secure customer data are established.

Following this, the Design stage translates these requirements into a structural blueprint, encompassing the architectural design of the Base64 integration and the creation of user interface mockups.

The subsequent Implementation (Coding) phase transforms these conceptual designs into an executable software program by embedding the Base64 encoding logic into the Sharia cooperative's web-based architecture using appropriate programming languages.

Once implemented, the system enters the Testing phase, undergoing rigorous evaluation to identify and resolve any operational errors or bugs, ensuring alignment with initial security requirements and successful customer data encryption without loss.

Finally, the Maintenance phase commences after deployment, involving the continuous monitoring of system performance and the rectification of any latent defects not discovered during testing.

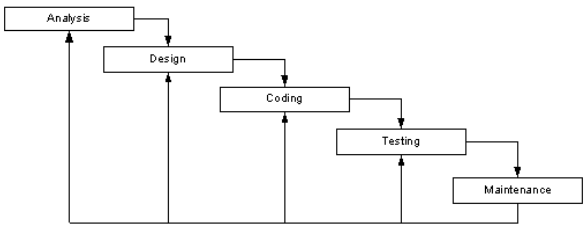


Figure 1. Waterfall Method Diagram

### 3.3 Algorithmic Process and System Analysis

Base64 is a data encoding and decoding algorithm that converts binary data into a printable ASCII format based on a radix-64 representation. While technically an encoding scheme, in the context of this system, it acts as an obfuscation layer to secure binary data during transmission and storage. The conversion process relies on two

primary reference tables as shown in Table 2, the standard ASCII table and the Base64 Index table.

Table 2. ASCII Code Table and the Base64 Index

| Binary | Oct | Dec | Hex | Glyph |
|---|---|---|---|---|
| 010 0000 | 040 | 32 | 20 | sp |
| 010 0001 | 041 | 33 | 21 | ! |
| 010 0010 | 042 | 34 | 22 | " |
| 010 0011 | 043 | 35 | 23 | # |
| 010 0100 | 044 | 36 | 24 | $ |
| 010 0101 | 045 | 37 | 25 | % |
| 010 0110 | 046 | 38 | 26 | & |
| 010 0111 | 047 | 39 | 27 | ' |
| 010 1000 | 050 | 40 | 28 | ( |
| 010 1001 | 051 | 41 | 29 | ) |
| 010 1010 | 052 | 42 | 2A | * |
| 010 1011 | 053 | 43 | 2B | + |
| 010 1100 | 054 | 44 | 2C | , |
| 010 1101 | 055 | 45 | 2D | - |
| 010 1110 | 056 | 46 | 2E | . |
| 010 1111 | 057 | 47 | 2F | / |
| 011 0000 | 060 | 48 | 30 | 0 |
| 011 0001 | 061 | 49 | 31 | 1 |
| 011 0010 | 062 | 50 | 32 | 2 |
| 011 0011 | 063 | 51 | 33 | 3 |
| 011 0100 | 064 | 52 | 34 | 4 |
| 011 0101 | 065 | 53 | 35 | 5 |
| 011 0110 | 066 | 54 | 36 | 6 |
| 011 0111 | 067 | 55 | 37 | 7 |
| 011 1000 | 070 | 56 | 38 | 8 |
| 011 1001 | 071 | 57 | 39 | 9 |
| 011 1010 | 072 | 58 | 3A | : |
| 011 1011 | 073 | 59 | 3B | ; |
| 011 1100 | 074 | 60 | 3C | < |
| 011 1101 | 075 | 61 | 3D | = |
| 011 1110 | 076 | 62 | 3E | > |
| 011 1111 | 077 | 63 | 3F | ? |

| Binary | Oct | Dec | Hex | Glyph |
|---|---|---|---|---|
| 100 0000 | 100 | 64 | 40 | @ |
| 100 0001 | 101 | 65 | 41 | A |
| 100 0010 | 102 | 66 | 42 | B |
| 100 0011 | 103 | 67 | 43 | C |
| 100 0100 | 104 | 68 | 44 | D |
| 100 0101 | 105 | 69 | 45 | E |
| 100 0110 | 106 | 70 | 46 | F |
| 100 0111 | 107 | 71 | 47 | G |
| 100 1000 | 110 | 72 | 48 | H |
| 100 1001 | 111 | 73 | 49 | I |
| 100 1010 | 112 | 74 | 4A | J |
| 100 1011 | 113 | 75 | 4B | K |
| 100 1100 | 114 | 76 | 4C | L |
| 100 1101 | 115 | 77 | 4D | M |
| 100 1110 | 116 | 78 | 4E | N |
| 100 1111 | 117 | 79 | 4F | O |
| 101 0000 | 120 | 80 | 50 | P |
| 101 0001 | 121 | 81 | 51 | Q |
| 101 0010 | 122 | 82 | 52 | R |
| 101 0011 | 123 | 83 | 53 | S |
| 101 0100 | 124 | 84 | 54 | T |
| 101 0101 | 125 | 85 | 55 | U |
| 101 0110 | 126 | 86 | 56 | V |
| 101 0111 | 127 | 87 | 57 | W |
| 101 1000 | 130 | 88 | 58 | X |
| 101 1001 | 131 | 89 | 59 | Y |
| 101 1010 | 132 | 90 | 5A | Z |
| 101 1011 | 133 | 91 | 5B | [ |
| 101 1100 | 134 | 92 | 5C | \ |
| 101 1101 | 135 | 93 | 5D | ] |
| 101 1110 | 136 | 94 | 5E | ^ |
| 101 1111 | 137 | 95 | 5F | _ |

| Binary | Oct | Dec | Hex | Glyph |
|---|---|---|---|---|
| 110 0000 | 140 | 96 | 60 | ` |
| 110 0001 | 141 | 97 | 61 | a |
| 110 0010 | 142 | 98 | 62 | b |
| 110 0011 | 143 | 99 | 63 | c |
| 110 0100 | 144 | 100 | 64 | d |
| 110 0101 | 145 | 101 | 65 | e |
| 110 0110 | 146 | 102 | 66 | f |
| 110 0111 | 147 | 103 | 67 | g |
| 110 1000 | 150 | 104 | 68 | h |
| 110 1001 | 151 | 105 | 69 | i |
| 110 1010 | 152 | 106 | 6A | j |
| 110 1011 | 153 | 107 | 6B | k |
| 110 1100 | 154 | 108 | 6C | l |
| 110 1101 | 155 | 109 | 6D | m |
| 110 1110 | 156 | 110 | 6E | n |
| 110 1111 | 157 | 111 | 6F | o |
| 111 0000 | 160 | 112 | 70 | p |
| 111 0001 | 161 | 113 | 71 | q |
| 111 0010 | 162 | 114 | 72 | r |
| 111 0011 | 163 | 115 | 73 | s |
| 111 0100 | 164 | 116 | 74 | t |
| 111 0101 | 165 | 117 | 75 | u |
| 111 0110 | 166 | 118 | 76 | v |
| 111 0111 | 167 | 119 | 77 | w |
| 111 1000 | 170 | 120 | 78 | x |
| 111 1001 | 171 | 121 | 79 | y |
| 111 1010 | 172 | 122 | 7A | z |
| 111 1011 | 173 | 123 | 7B | { |
| 111 1100 | 174 | 124 | 7C | \| |
| 111 1101 | 175 | 125 | 7D | } |
| 111 1110 | 176 | 126 | 7E | ~ |

The algorithmic mechanism operates through two main procedures. Encryption (Encoding) Process, to demonstrate the encoding mechanism, the string "ITUMPO" is used as a simulation. The first step involves converting each character into its corresponding ASCII decimal value, which is then translated into an 8-bit binary format. These 8-bit sequences are concatenated and subsequently divided into new blocks of 6-bit binary sequences. The new 6-bit blocks are converted back into decimal values, which act as index pointers for the Base64 Index table. Mapping these indices yields the encoded string "SVRVTVBP". If the final binary sequence does not form a complete 6-bit block, zero-padding is applied, and the padding operation is represented by the equals sign ("=") in the final Base64 output.

Then, as shown in Table 3, the decryption process reverses the encoding workflow. Using the encoded string "SVRVTVBP", each character is mapped to its numerical value using the Base64 Index table. These decimals are converted into 6-bit binary blocks. The 6-bit blocks are concatenated and regrouped into standard 8-bit binary sequences. Finally, the 8-bit binaries are converted to decimal values and mapped against the ASCII table to reconstruct the original plaintext string, "ITUMPO".

Table 3. Encryption Base64 Process

| Base64 | S | V | R | V | T | V | B | P |
|---|---|---|---|---|---|---|---|---|
| Index | 18 | 21 | 17 | 21 | 19 | 21 | 1 | 15 |
| Binner 6 | 010010 | 010101 | 010001 | 010101 | 010011 | 010101 | 000001 | 001111 |
| Binner 8 | 01001001 | 01010100 | 01010101 | 01001101 | 01010000 | 01001111 | | |
| ASCII | 73 | 84 | 85 | 77 | 80 | 79 | | |
| Word | I | T | U | M | P | O | | |

## 3.4 User Interface Architecture

The interface design translates the operational logic into an accessible web-based platform tailored for a Sharia Cooperative (BMT). The system architecture is divided into multiple access levels, accommodating administrators, cooperative members, and regular customers. The core modules include security access (Login), a centralized Dashboard, Master Data management (users, members, products), Savings and Financing records, and Cash Flow management.

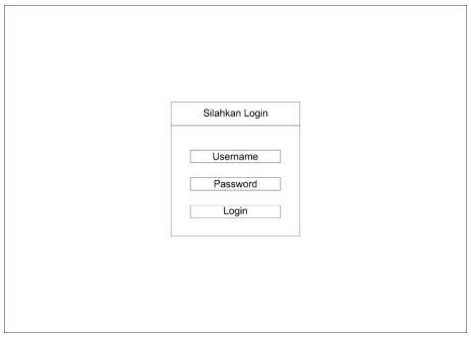


Figure 2. Login Page

Figure 2, illustrates the Login page interface, which serves as the primary security gateway within the Sharia Cooperative (BMT) system architecture. In this interface, an authentication process is enforced to validate user credentials before the system grants access rights. This account verification mechanism is crucial as the first line of defense to ensure that only authorized entities or administrators can interact with the system. This access control constitutes a fundamental preventive measure to protect operational data entities from the risks of modification or unauthorized access by external parties.

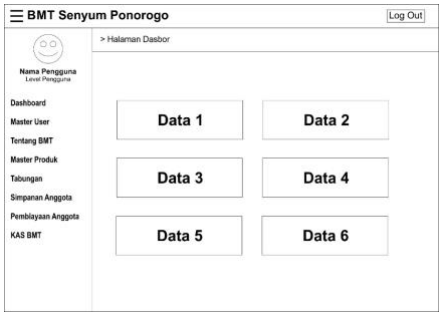

Figure 3. Dashboard

Upon successfully bypassing the authentication layer, the system directs the user to the Dashboard page, as presented in Figure 3. This page represents the main structural architecture of the cooperative's information system. The dashboard acts as a centralized navigation hub that integrates all BMT operational modules into a single, structured interface. These modules encompass master data management (users, members, customers), savings records, member financing, and cash flow management. This centralized interface design approach aims to enhance operational efficiency while abstracting the complexity of the backend logic including the encoding and decoding processes of customer data thereby enabling users to execute system functionalities intuitively.

## 4. Results and Discussion

### 4.1 Implementation of Base64 Encoding in System Architecture

The implementation of the Base64 algorithm was directly integrated into the functional CRUD (Create, Read, Update, Delete) architecture of the Sharia cooperative's information system. This functional approach was selected to ensure that data protection is inextricably linked to the data lifecycle, from the moment of entry to its eventual deletion. The encoding logic was strategically embedded within several primary source code files responsible for managing member information and system operational flows, including csrf.php, data.php, form_action.php, and get_data.php. The fundamental objective of this integration is to guarantee that all sensitive customer and member information automatically undergoes an obfuscation phase immediately before being transmitted and recorded into the database. This method establishes a zero-trust security architecture at the application level, where the database never receives or stores information in its original, vulnerable state.

### 4.2 Data Input Security and Obfuscation Outcomes

To empirically evaluate the efficacy and consistency of the encoding process, system testing was conducted using sample member data in a real-time environment. The input parameters during the form submission stage included various high-sensitivity identifiers according to data privacy regulations, such as Member ID (615-011-3101), National Identification Number or NIK (1234567890123456), Full Name (Teknik Informatika UMPO), Gender (Male), and Phone Number (987252354).

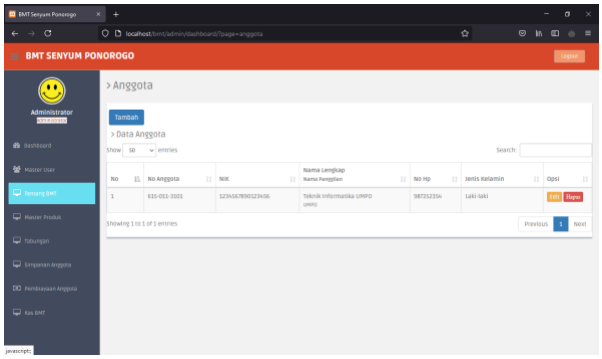


Figure 1. Input form

As illustrated in detail in Figure 1, the initial data entry process by the system administrator is performed in plaintext format. The presence of this input interface is vital to demonstrate that the security mechanism does not disrupt the user experience (UX). Users do not require a specialized understanding of cryptography to operate the system; they continue to input data normally and intuitively.

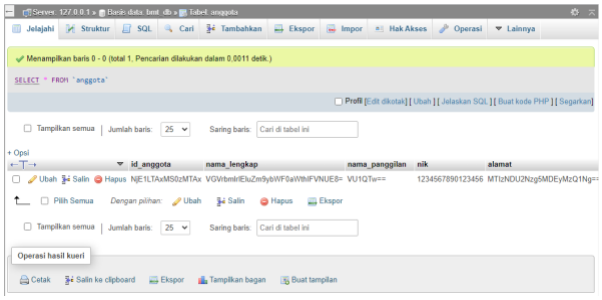

Figure 2. Database

However, immediately after the form is submitted to the server, the form_action.php module processes these parameters using the Base64 algorithm prior to executing the storage query. The result of this background transformation is visually evidenced in Figure 2, which displays a direct screenshot of the database table after the encoding process. The figure clearly illustrates how sensitive data, such as NIK and phone numbers that were previously easily readable, have now been encapsulated into randomized alphanumeric strings. This structural obfuscation dramatically reduces the direct readability of the data. In the context of cyber threat modeling, the representation in Figure 2 confirms that if a malicious actor were to breach

the server through SQL Injection exploitation, they would only extract meaningless, randomized data without the appropriate decoding algorithm.

### 4.3 Database Integrity Testing on Administrative Modules

Beyond securing public identities, comprehensive functional testing was also implemented on administrative modules. These modules store the credentials of internal users who possess administrative modification rights; thus, their protection is a top priority to prevent insider threats. Figure 3 presents a review of the admin and user data login page before the security algorithm is activated. In this pre-encoding condition, usernames, passwords, and other administrative access parameters are transparently displayed. This represents a fatal vulnerability in standard software security.

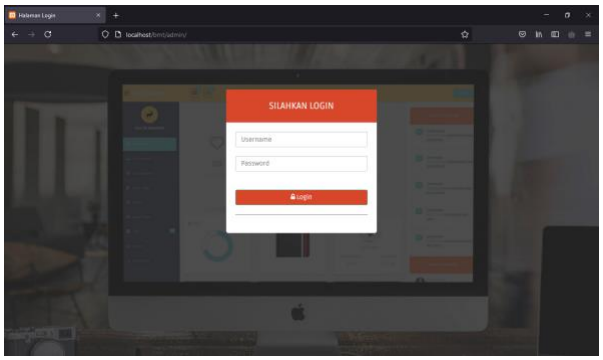


Figure 3. Login page

In contrast, Figure 4.8 provides empirical evidence regarding the state of the database during the post-encoding phase. The figure convincingly demonstrates that administrative credential information within the database has been fully converted and secured by Base64. This transformation is crucial; it guarantees that bad actors cannot manually analyze the users table to exploit administrative data for privilege escalation.

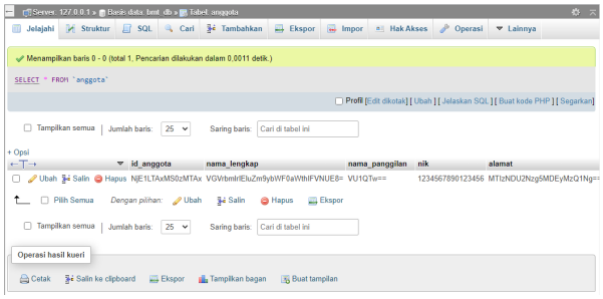

Figure 4. After encoding process

### 4.4 Scalability of Protection for Member and Customer Entities

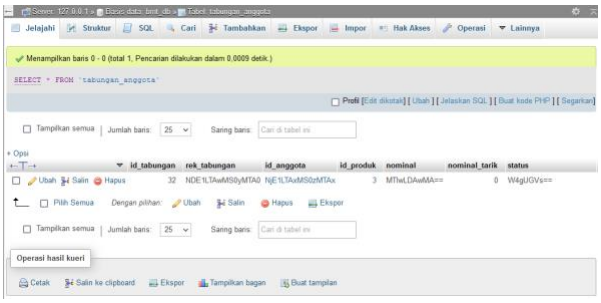

Figure 5. Raw Data Cashflow in database

The technical reliability of this algorithm was further validated on modules handling a more massive volume of data, specifically registered member and general customer entities. Figure 5 shows the raw data of member profiles in their entirety before protection is applied. This image represents the actual data load managed by the Sharia Cooperative daily. Conversely, Figure 4.10 displays the same data records after undergoing the Base64 encoding mechanism, proving that the system is capable of simultaneous encoding across multiple data columns without encountering errors.

Identical privacy protection patterns were also executed on the customer database. By demonstrating successful transformations across different data entities (members vs. customers), this research proves that this Base64-based protection mechanism can be implemented comprehensively across the entire database architecture, aligning with the principle of amanah (trustworthiness) in maintaining public integrity and confidentiality within Sharia financial institutions.

### 4.5 Security of Financial Transaction Trails

In financial systems, data leaks threaten not only personal identities but also the historical records of monetary activities. Therefore, the Base64 encoding in this system was extended to secure functional transaction data. This representation is a significant finding as it shows that the system secures nominal numerical transaction values. By concealing nominal details and financial flows, the cooperative successfully minimizes the risk of external parties performing unauthorized wealth profiling of customers, thereby meeting Information Security Management standards in the micro-finance sector.

### 4.6 Evaluation of Interface and System Performance

Overall, the integration of the Base64 algorithm has proven to provide a significant layer of data protection without degrading system efficiency. Testing of the User Interface (UI) confirmed that the addition of this lightweight cryptographic process does not cause processing latency. All supporting menu elements, such as the operational home page and interface configurations, remain responsive. The strategic decision to utilize Base64 as opposed to heavier, more resource-intensive algorithms is proven to be highly relevant for maintaining the stability of the Sharia Cooperative's web architecture while ensuring that customer data privacy remains the highest priority.

## 5. Conclusion

This study demonstrates that the implementation of the Base64 algorithm in a web-based sharia cooperative information system can improve the protection of customer data at the application level. The results show that sensitive data entered through system forms, such as member, customer, administrative, savings, and cash records, were successfully transformed into encoded strings before being stored in the database. As a result, the original data were no longer directly readable in the storage layer, which reduced the risk of immediate exposure to unauthorized parties.

From the system testing, the integration of Base64 into CRUD processes was implemented consistently without disrupting the main functional features of the cooperative system. This indicates that the method is technically feasible and easy to integrate into an existing web-based information system. However, although Base64 contributes to basic data obfuscation, it should be understood as a supporting protection mechanism rather than a complete cryptographic solution. Therefore, future studies are recommended to combine Base64 with stronger encryption techniques to provide a higher level of confidentiality and security for sensitive financial data in sharia cooperative systems.

## References

[1] M. K. Hassan, A. Muneeza, and B. S. Sairally, "Fintech and Islamic Finance: Literature Review and Research Agenda," *International Journal of Islamic and Middle Eastern Finance and Management*, vol. 14, no. 4, pp. 669–688, 2021.

[2] S. Ali, M. Ali, and N. Ahmed, "Cybersecurity Challenges in the Fintech Era: A Systematic Literature Review," *Comput. Secur.*, vol. 97, p. 101960, 2020.

[3] A. Firmansyah, N. Respati, and M. Mahrus, "Information Technology Governance and Security in Indonesian Micro-Finance Institutions," *Heliyon*, vol. 8, no. 10, p. e11025, 2022.

[4] T. Khan and M. Ahmed, "The Role of Data Obfuscation in Enhancing Privacy for Web Applications," *IEEE Access*, vol. 9, pp. 12450–12465, 2021.

[5] H. Mohamad, N. Ibrahim, and D. A. Razak, "Trust and Risk in Sharia Fintech Adoption: Evidence from Indonesia," *Journal of Islamic Marketing*, vol. 13, no. 2, pp. 345–362, 2022.

[6] A. A. Alani, "Securing data in transit and at rest using advanced cryptography," *IEEE Access*, vol. 9, pp. 45612–45625, 2021.

[7] M. A. Al-Shabi, "A survey on symmetric and asymmetric cryptography algorithms in information security," *International Journal of Scientific and Technology Research*, vol. 8, no. 9, pp. 576–589, 2019.

[8] M. Z. A. Bhuiyan, "Data obfuscation techniques in web applications: A robust framework," *Future Generation Computer Systems*, vol. 108, pp. 234–245, 2020.

[9] X. Chen and Y. Huang, "Public key encryption with efficient key management," *Inf. Sci. (N. Y).*, vol. 512, pp. 1205–1215, 2020.

[10] M. K. Hassan, "Fintech and Islamic Finance: Literature Review and Research Agenda," *International Journal of Islamic and Middle Eastern Finance and Management*, vol. 14, no. 4, pp. 669–688, 2021.

[11] T. Khan and M. Ahmed, "The Role of Data Obfuscation in Enhancing Privacy for Web Applications," *IEEE Access*, vol. 9, pp. 12450–12465, 2021.

[12] T. Lovian and I. Fitri, "Implementasi Algoritma Base64 Sebagai Tingkat Keamanan Data Pada Website Sistem

Informasi Pencatat Barang," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 6, no. 1, p. 692, Jan. 2022, doi: 10.30865/mib.v6i1.3513.

[13] S. Sharma, "Comparative analysis of symmetric cryptography algorithms for secure data transmission," *Journal of Information Security and Applications*, vol. 58, p. 102745, 2021.

[14] A. F. Cobantoro, M. B. Setyawan, and H. Oktavianto, "Rekayasa Aplikasi Eposal Menggunakan Algoritma Base64 Untuk Menyimpan Data Pengguna," *Jurnal Komtika (Komputasi dan Informatika)*, vol. 7, no. 1, pp. 31–38, May 2023, doi: 10.31603/komtika.v7i1.8711.

[15] H. Mohamad, "Trust and Risk in Sharia Fintech Adoption: Evidence from Indonesia," *Journal of Islamic Marketing*, vol. 13, no. 2, pp. 345–362, 2022.

[16] S. R. Tariq, "Lightweight cryptography for resource-constrained devices in smart systems," *Computers & Electrical Engineering*, vol. 92, p. 107123, 2021.