Performance Analysis of RC4 Symmetric and RSA Asymmetric Cryptographic Algorithms In Securing Normal Text Messages

Renaldi
Department of Information System, Universitas Muhammadiyah Sumatera Utara, Indonesia

ABSTRACT

Information security plays a crucial role in the digital age, particularly in safeguarding text messages from unauthorized access. Cryptography serves as a means of data security by employing encryption algorithms that transform the original message into a format that is hard to comprehend. This thesis examines the performance evaluation of two cryptographic algorithms: RC4, a symmetric algorithm, and RSA, an asymmetric algorithm, in the protection of regular text messages. This research centers on comparing the two algorithms regarding their encryption and decryption speeds, along with their effectiveness in masking the character frequency pattern present in the original text message. Testing was conducted with various short, medium, and long text documents. The test outcomes indicate that RC4 excels in speed for both encryption and decryption processes, particularly when handling large texts. Nonetheless, RSA excels in security due to its capacity to generate a more random and unpredictable character frequency distribution.

Keyword : Cryptography, RC4, RSA, Encryption, Decryption, Message Security

Corresponding Author:
Renaldi,
Department of Information System,
Universitas Muhammadiyah Sumatera Utara,
Jalan Kapten Muktar Basri No 3 Medan 20238, Indonesia.
Email: renaldi@gmail.com

1. INTRODUCTION

In the digital age, information security is a vital component in the process of data sharing. The protection of messages is necessary to prevent unauthorized parties from accessing confidential information (Sina et al., 2022). One of the main challenges in ensuring data security is the risk of data leakage and misuse. Not all information is intended for the public; therefore, confidentiality must be maintained to avoid the exposure of sensitive data to the wrong hands.

Data confidentiality and message security are crucial for both individuals and organizations. Senders of important information require a high level of protection to ensure their data remains secure (Rasudin et al., 2022). With technological advancement, data encryption can now be achieved using cryptographic algorithms (Fitriana & Djuniadi, 2022). Cryptography is generally divided into two main types: symmetric and asymmetric cryptography (Rasudin et al., 2022).

Symmetric cryptography uses the same private key for both encryption and decryption, offering simplicity and speed. On the other hand, asymmetric cryptography involves a pair of keys—public and private—to secure the data. Well-known asymmetric algorithms include RSA (Rivest–Shamir–Adleman), El Gamal, Elliptic Curve, Hill Cipher, and Diffie-Hellman, while popular symmetric methods include RC4, Blowfish, and DES (Suhandinata et al., 2019).

This study aims to analyze how the symmetric RC4 algorithm and the asymmetric RSA algorithm protect textual messages, as each cryptographic method has its own advantages and disadvantages. It compares both algorithms based on encryption and decryption speed, and the resulting ciphertext size for text messages of equal length. Frequency analysis will be applied to evaluate their security performance.

The study is limited to the implementation of RC4 and RSA algorithms in a desktop-based Java NetBeans application. This research is expected to provide deeper insights into the strengths and limitations of both algorithms and offer recommendations on their optimal usage for secure message encryption. Additionally, it aims to contribute to educational resources and future research in the field of information security.

2. RESEARCH METHOD

A. Data Analysis Techniques

In this research, the approach used is a literature review, involving analysis of relevant previous studies. The analysis is divided into two main stages: analysis using symmetric cryptography methods and analysis using asymmetric cryptography methods.

a. Analysis Using Symmetric Methods

The first stage of this study involves an analysis of symmetric cryptography methods. Previous research relevant to this context will be examined in-depth. During this stage, various aspects of symmetric cryptography will be evaluated, including its basic principles, the algorithms used, as well as their respective strengths and weaknesses. Data and findings from this analysis will be recorded for further comparison.

b. Analysis Using Asymmetric Methods

The second stage of this study focuses on analyzing asymmetric cryptography methods. Relevant previous research in the context of asymmetric cryptography will also be studied in-depth. This analysis includes the basic principles of asymmetric cryptography, the algorithms used, their advantages, and disadvantages. The information obtained will be used to compare with the results of the symmetric cryptographic method analysis.

c. Frequency Analysis

In a simple substitution cipher, each letter in the plaintext is replaced by another letter, and a specific letter in the plaintext will always be replaced by the same letter in the ciphertext. For example, if every occurrence of the letter 'e' is replaced by 'X', then a ciphertext containing many 'X's may indicate that 'X' represents 'e'.

Basic frequency analysis is conducted by calculating the frequency of letter appearances in the ciphertext and attempting to match them with predicted letters in the plaintext. If the number of 'X's exceeds other letters in the ciphertext, it might suggest that 'X' represents 'e' in the plaintext. However, this is not always accurate, as letters like 't' and 'a' are also common in English, meaning 'X' might actually represent one of those. Conversely, rarely used letters in English such as 'z' or 'q' are less likely to be involved. Therefore, a cryptanalyst may need to try different mapping combinations between the ciphertext and plaintext letters (F. Riza, 2018).

d. Data Security Level Evaluation

After analyzing both types of cryptography, an evaluation will be conducted on the level of data security achievable with each method. This evaluation process will cover factors such as confidentiality, integrity, authentication, as well as potential risks or attacks that may compromise data security. The results will be used to conclude and compare the security levels between symmetric and asymmetric cryptographic methods.

e. Implementation of Symmetric Cryptography Method Using RC4 Algorithm

The first stage of implementation involves the symmetric cryptography method using the RC4 algorithm. This process includes the user creating a text to be encrypted, and after encryption, decrypting it within the system.

i. Input Text to Be Encrypted
First, the user fills in a form with the text that is to be encrypted.

ii. Input Key
Next, the user must create a key to encrypt the text.

iii. Encryption Process
The RC4 method is used to encrypt the prepared text. The original plaintext is transformed into ciphertext.

iv. Encryption Result
After the encryption process, the plaintext becomes ciphertext, appearing different from the original text.

v. Input Encrypted Text
After generating the ciphertext, the user can convert it back into plaintext by inputting the ciphertext into the form designated for decryption.

vi. Input Key
The decryption process is done by entering the same key that was used for encryption.

vii. Decryption Result
The decryption result returns the original plaintext created by the user. This process ensures that the original text is properly secured and can only be read by parties with the correct encryption key.

f. Implementation of Asymmetric Cryptography Method: Use of Public and Private Keys with RSA Algorithm

After implementing both symmetric and asymmetric encryption and decryption methods, it is time to analyze and compare them

B. System Design

a. RC4 Symmetric Algorithm Flowchart: Begins with key input and array initialization. It generates a pseudo-random byte stream using KSA (Key Scheduling Algorithm) and PRGA (Pseudo-Random Generation Algorithm). Plaintext is encrypted or decrypted using XOR with the generated stream.

b. RSA Asymmetric Algorithm Flowchart: Starts by selecting two prime numbers (p and q), calculates `n = p * q`, and generates a public key (PK) and private key (SK). Encryption uses the public key, while decryption uses the private key with modular arithmetic.

Use Case and Activity Diagrams

a. Activity Diagrams (RC4)
- Encryption: User selects RC4, inputs text and key, clicks encrypt, and gets the ciphertext.
- Decryption: User selects RC4, inputs ciphertext and key, clicks decrypt, and gets the plaintext.

b. Activity Diagrams (RSA)
- Encryption: User selects RSA, generates keys, inputs text and public key, clicks encrypt, and receives ciphertext.
- Decryption: User selects RSA, inputs ciphertext and previously generated private key, clicks decrypt, and retrieves the plaintext.

Sequence Diagrams

Show the interaction flow between user and system during encryption and decryption for both RC4 and RSA methods.

System Specifications
- Software: Apache NetBeans IDE 21, JDK 22.0.1.
- Hardware: Minimum laptop specs include Intel Celeron N4020 processor and 8GB RAM.

Interface Design
- Main Page: Contains buttons for selecting encryption/decryption methods.
- RC4 Pages: Separate interfaces for encryption and decryption with text and key input forms.
- RSA Pages: Similar layout but includes key generation features.

3. RESULTS AND DISCUSSION

A. General Description

This research aims to evaluate and compare the performance of two different cryptographic algorithms: RC4 as an example of a symmetric algorithm and RSA as an example of an asymmetric algorithm, in the context of securing plain text messages. In a world increasingly reliant on digital information exchange, data security has become crucial—especially in ensuring that messages sent and received remain confidential and inaccessible to unauthorized parties. Both algorithms are analyzed with a focus on the frequency analysis of characters generated after the encryption process. These frequency results are used to assess how well patterns in the original text can be disguised by each algorithm, providing insights into their strengths and weaknesses in securing text messages. This study is expected to contribute to the selection of the most appropriate cryptographic algorithm based on specific needs in terms of speed, efficiency, and security.

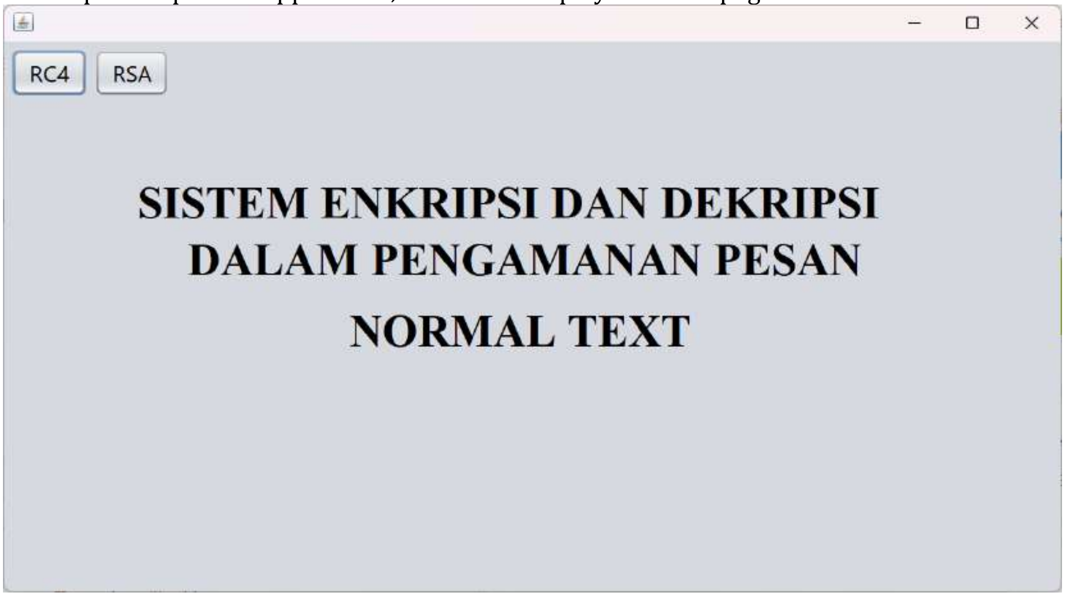
In this research, the symmetric method—RC4 algorithm—is known for its speed and efficiency in data encryption, but it has vulnerabilities related to security if not implemented correctly. On the other hand, the asymmetric RSA method is known for its higher level of security due to the use of public and private keys but is often criticized for its slower encryption and decryption processes, especially when dealing with large data sets. This study will analyze the performance of both algorithms through frequency analysis.

B. Implementation of Rivest Code (RC4)

Based on the implementation of the method proposed in the previous chapter, the following is the result of the implementation in a desktop application developed using the RC4 method.

1) Encryption Implementation

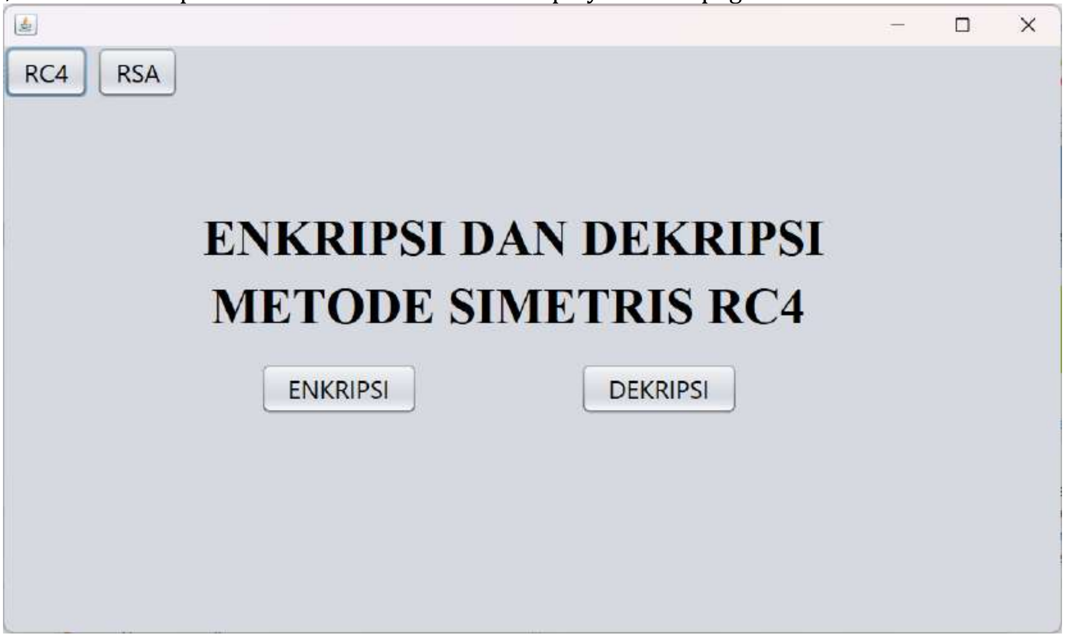
The first step is to open the application, which will display the main page.



SISTEM ENKRIPSI DAN DEKRIPSI
DALAM PENGAMANAN PESAN
NORMAL TEXT

Figure 1. Main Page

Then, the second step is to click the RC4 button to display the RC4 page.



ENKRIPSI DAN DEKRIPSI
METODE SIMETRIS RC4

Figure 2. RC4 Page

Then, the third step is click the encryption button to implement the encryption

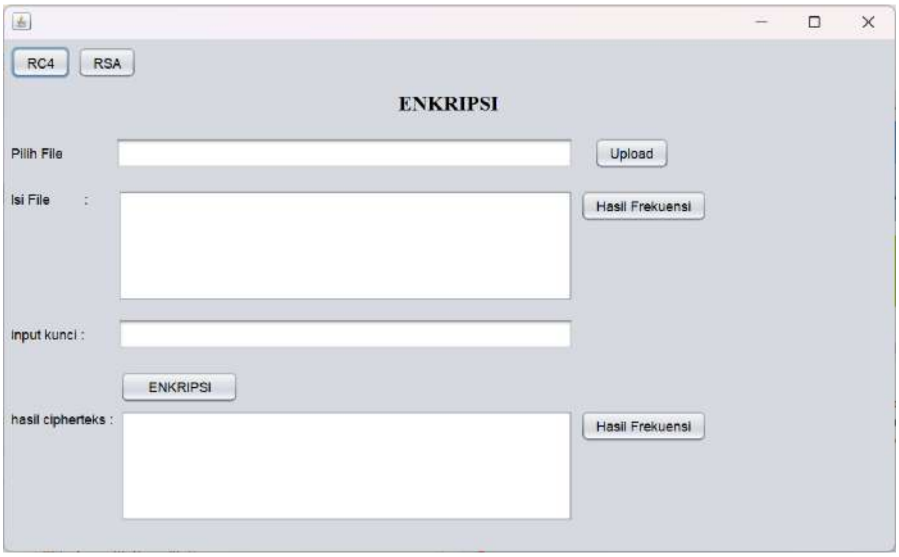
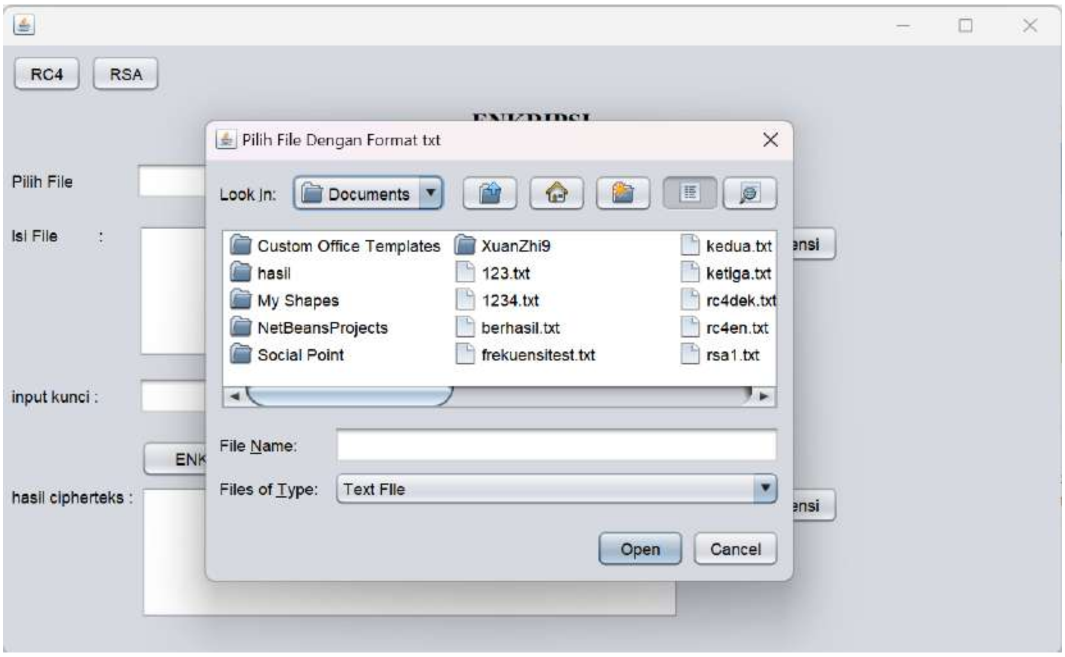


Figure 3. Encription Page

Then, the fourth step is to input the .txt file that you want to encrypt by clicking the Upload File button. After that, a file selection frame will appear, allowing you to choose the file to be encrypted, as shown in Figure 4.

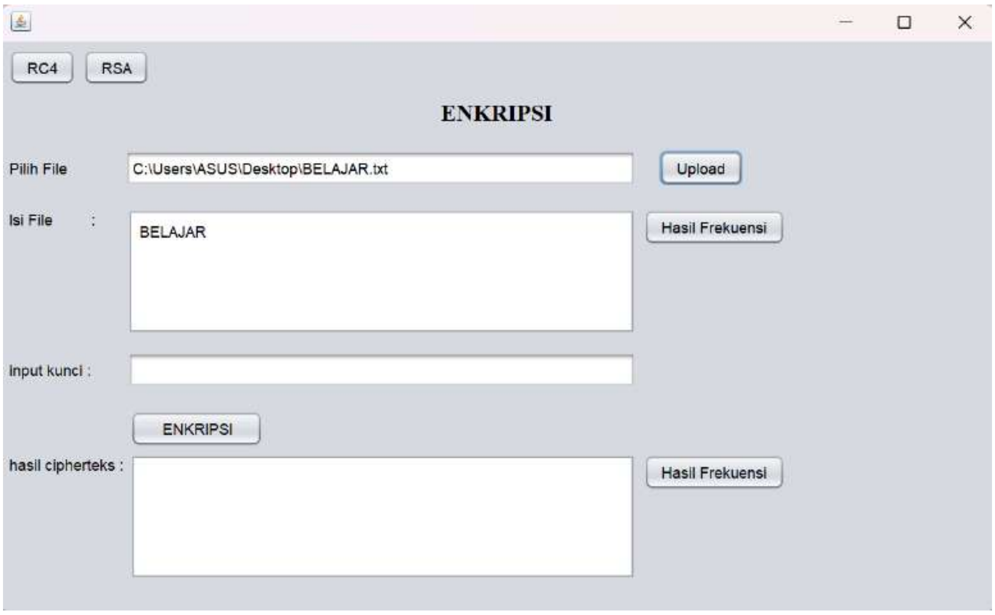

Pilih File Dengan Format txt

Look In: Documents

Files of Type: Text File

Figure 4. Frame File Display to Select File

After selecting the file, the application will display the file name and the file content in the Choose File form and the File Content form, as shown in Figure 5.



ENKRIPSI

Pilih File C:\Users\ASUS\Desktop\BELAJAR.txt

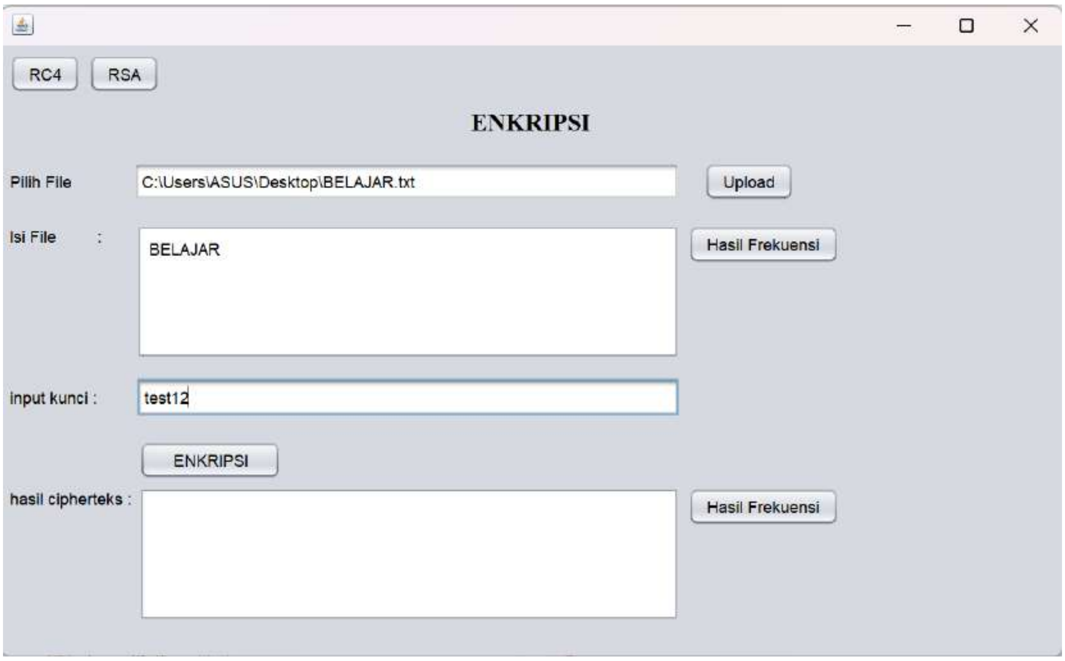
Isi File : BELAJAR

Input kunci :

hasil cipherteks :

Figure 5. File Selection View

The fifth step is to enter the password or key into the key input form, which only the user knows.



ENKRIPSI

Pilih File C:\Users\ASUS\Desktop\BELAJAR.txt

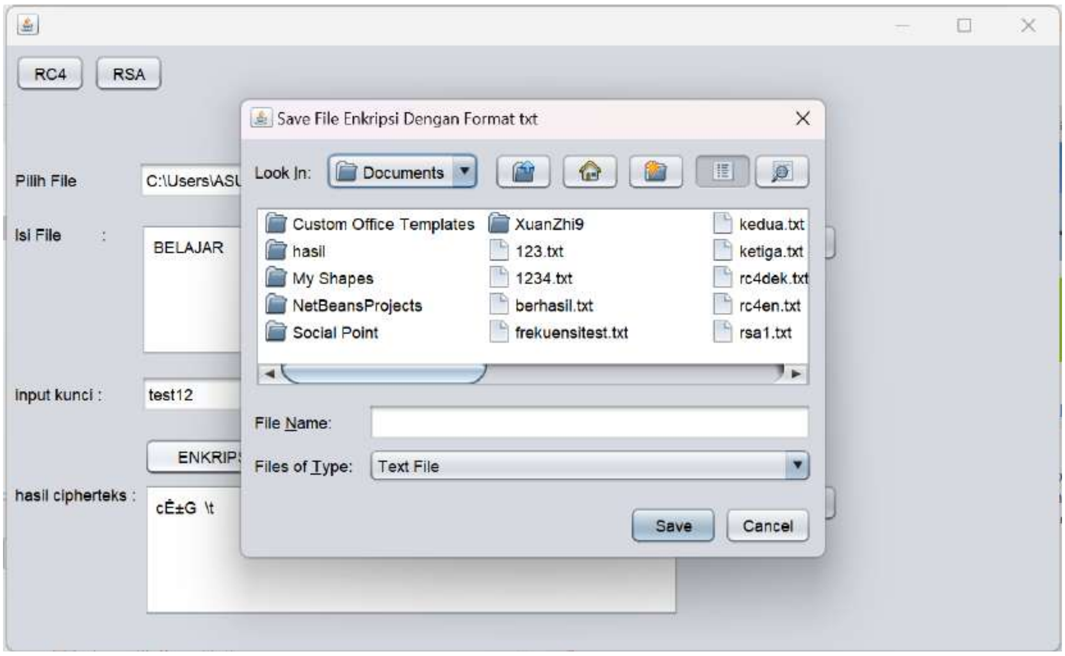
Isi File : BELAJAR

Input kunci : test12

hasil cipherteks :

Figure 6. Key Input Content Display

Next, in the sixth step, click the Encrypt button to proceed with the encryption process for the file. Once the process is complete, the application will display the result in the Ciphertext Result section and show a frame to save the encrypted file, as illustrated in Figure 7.



Pilih File C:\Users\ASU

Isi File : BELAJAR

input kunci : test12

hasil cipherteks : cÊ±G \t

Figure 7. Encryption Result Display and File Save Frame

Subsequently, the file should be saved with a name that corresponds to the user's preferences. In this encryption section, a file has been successfully encrypted.

2) Decryption Implementation

Following encryption, the subsequent stage is to initiate the decryption implementation process for the encrypted file.

Subsequent to encryption, the RC4 button on the encryption display must be selected, thus initiating the RC4 page.

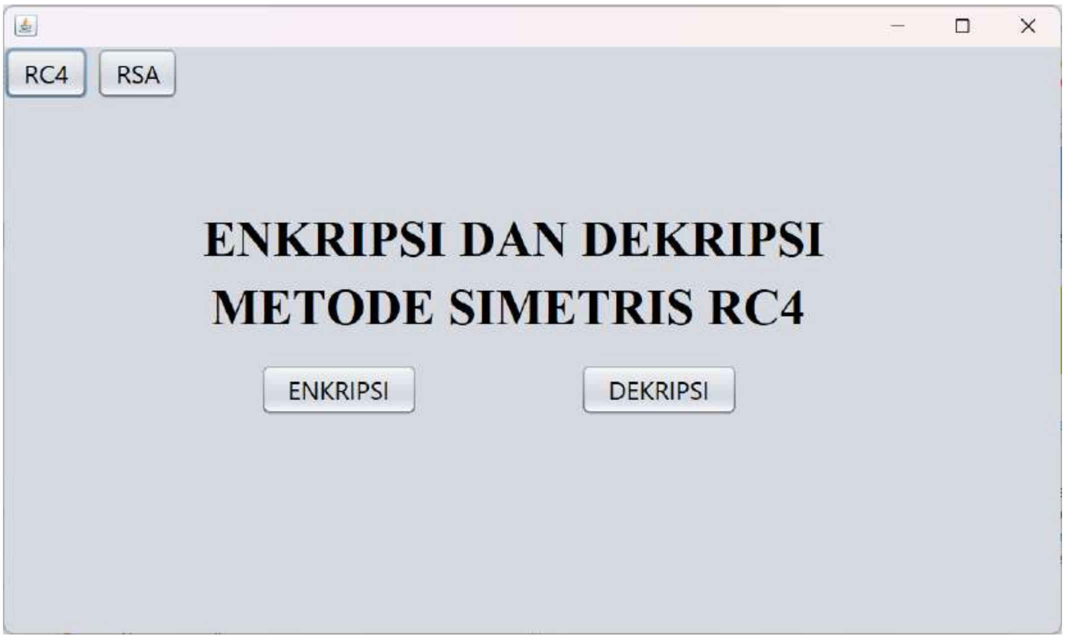


Figure 8. RC4 Display

The subsequent step in the process is to initiate the decryption process by selecting the relevant option. Subsequent to the clicking of the aforementioned button, the decryption page will appear.

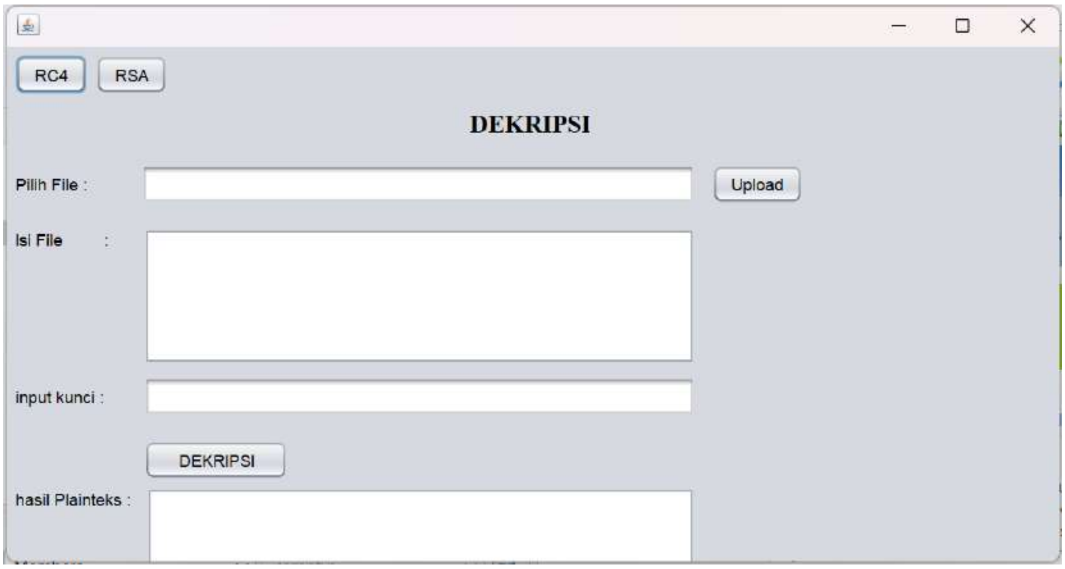
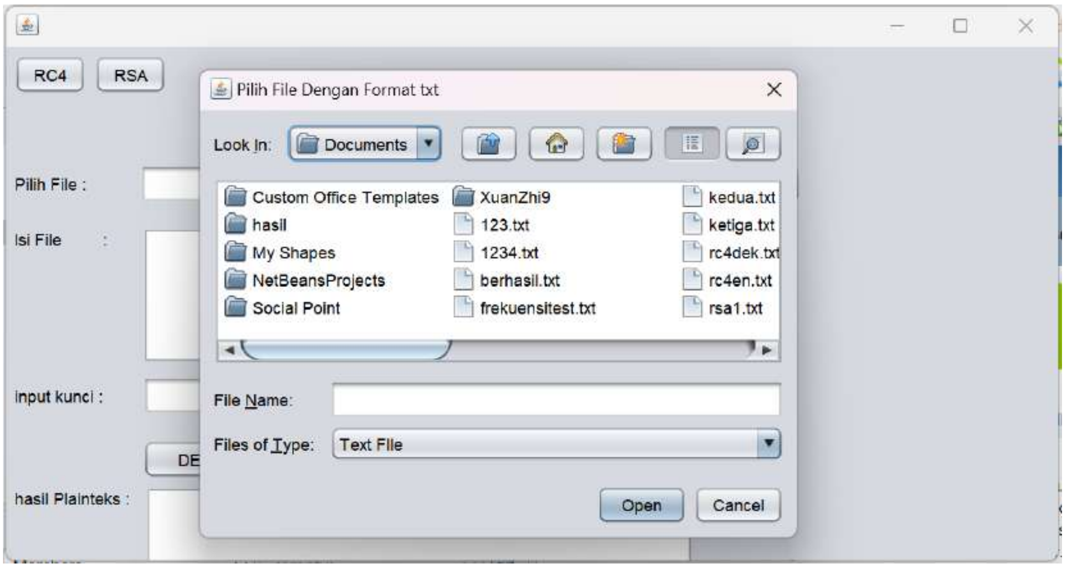


Figure 9. Description Page

The third step in the process is to input the file by clicking on the upload button. This action will result in the appearance of a file selection frame, as illustrated in Figure 10.



Look In: Documents

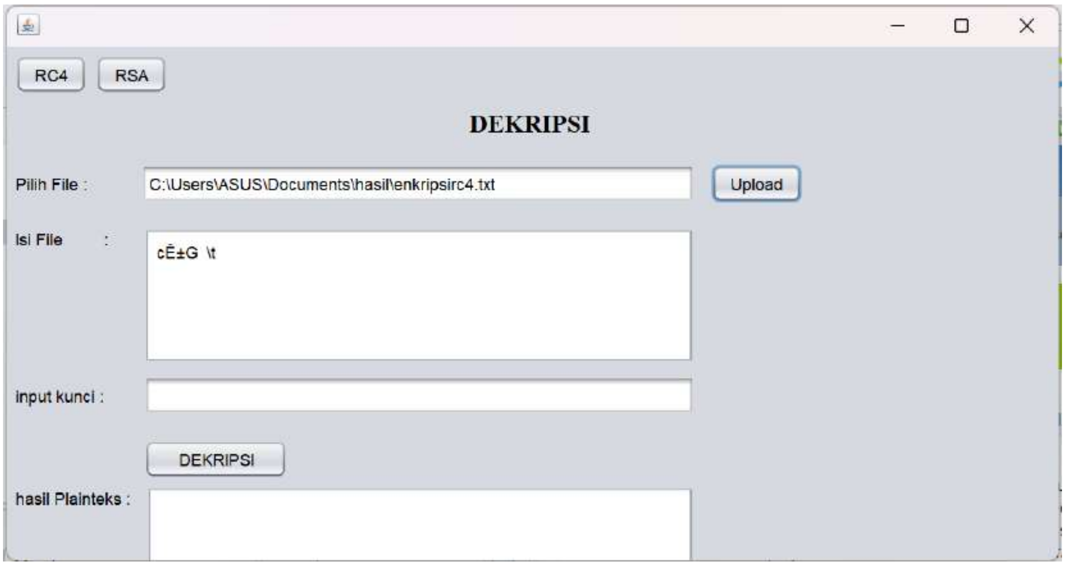
Custom Office Templates
hasil
My Shapes
NetBeansProjects
Social Point
XuanZhi9
123.txt
1234.txt
berhasil.txt
frekuensitest.txt
kedua.txt
ketiga.txt
rc4dek.txt
rc4en.txt
rsa1.txt

File Name:

Files of Type: Text File

Figure 10. Select File Frame Display

Following the selection of the file, the file name and contents will be displayed in the 'Select file and file contents' form, as illustrated in Figure 11.



Pilih File : C:\Users\ASUS\Documents\hasil\enkripsirc4.txt

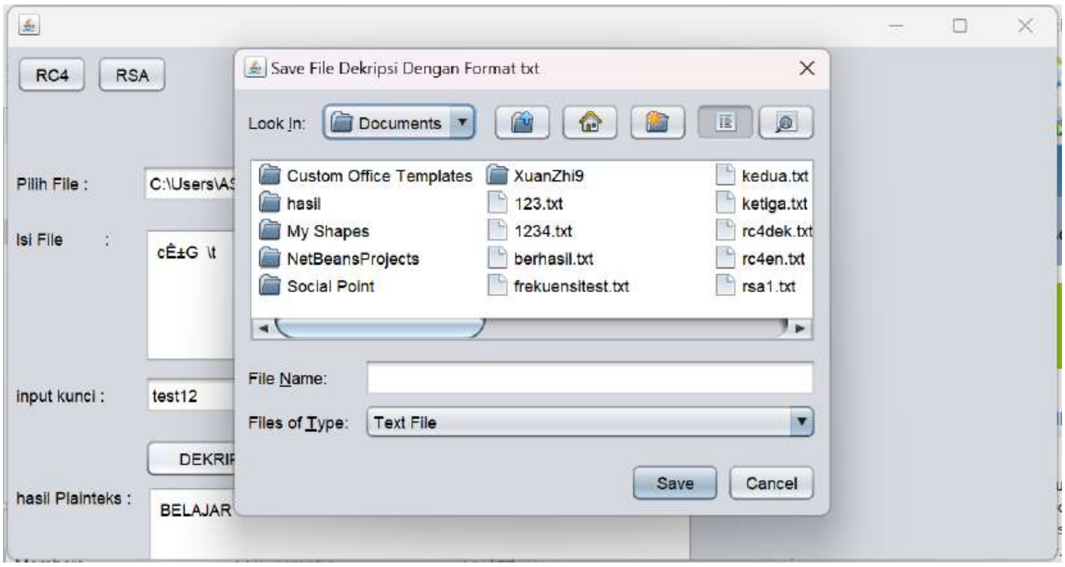
Isi File : cÊ±G \t

input kunci :

hasil Plainteks :

Figure 11. After Selecting File Display

The fourth step in the process is to enter the key that has been created in the key encryption implementation, a step which is only known to the user. This action will result in the appearance of a file save frame. In the event of the key being correct, the decryption result or plaintext will appear according to the correct file contents, as illustrated in Figure 12. Conversely, if the key is incorrect, the plaintext result will not match the correct file contents, as demonstrated in Figure 13.



Pilih File : C:\Users\AS

Isi File : cÊ±G \t

input kunci : test12

hasil Plainteks : BELAJAR

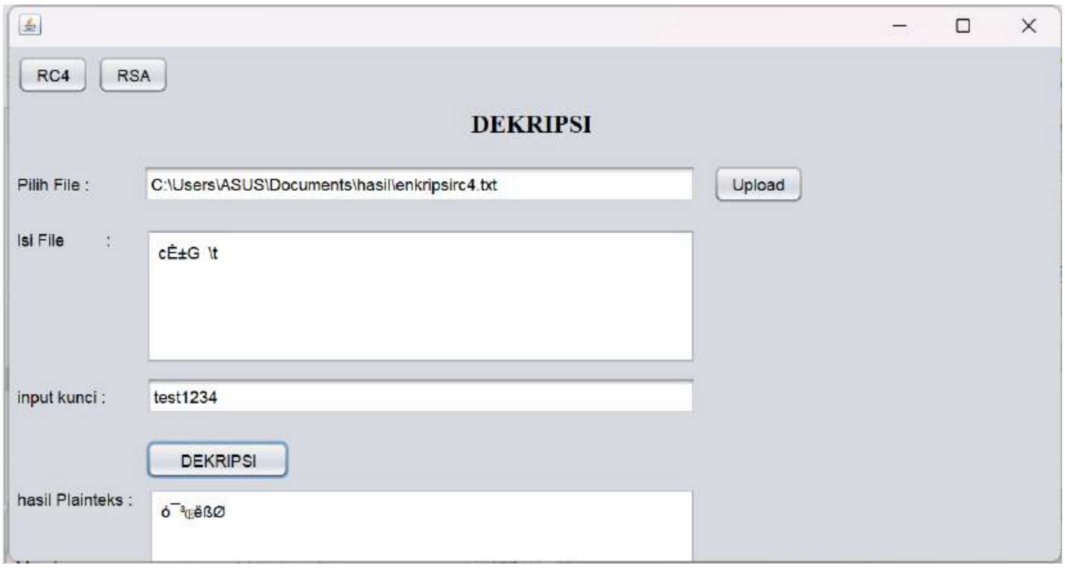
Look In: Documents

Custom Office Templates, hasil, My Shapes, NetBeansProjects, Social Point, XuanZhi9, 123.txt, 1234.txt, berhasil.txt, frekuensitest.txt, kedua.txt, ketiga.txt, rc4dek.txt, rc4en.txt, rsa1.txt

File Name:

Files of Type: Text File

Figure 12. Decryption result with correct key



DEKRIPSI

Pilih File : C:\Users\ASUS\Documents\hasil\enkripsirc4.txt

Isi File : cÊ±G \t

input kunci : test1234

hasil Plainteks : ó¯³⃞ëßØ

Figure 13. Results with wrong key

C. RC4 Calculation

The RC4 algorithm, a symmetric encryption method, is demonstrated through a manual calculation and implementation.

Key Steps in RC4:

S-Box Initialization (S):

A 256-byte array initialized from 0 to 255.

Key Expansion (K):

A key (test12) is converted into ASCII values (e.g., t = 116, e = 101, etc.), then repeated until 256 bytes in length.

KSA (Key Scheduling Algorithm):

S is shuffled based on K using the formula:

j = (j + S[i] + K[i]) mod 256

Each S[i] is swapped with S[j].

After 256 iterations, a new permuted array S is generated.

Example (First few iterations):

- Iteration 1: i = 0, j = 116, swap S[0] and S[116]
- Iteration 2: i = 1, j = 218, swap S[1] and S[218]

(Continues up to iteration 255)

PRGA (Pseudo-Random Generation Algorithm):

For 7 characters of plaintext, PRGA is performed 7 times to generate the keystream.

- Iteration 1: i = 1, j = 84, generate key byte K1 = 249
- Iteration 2: i = 2, j = 212, generate K2 = 143
- Iteration 3: i = 3, j = 154, generate K3 = ...

(Continues for the remaining characters)

Each generated key byte is XOR-ed with the corresponding character in plaintext to produce the ciphertext.

Result: The RC4 encryption and decryption process was successfully calculated and matched expectations based on the symmetric key principle.

4. CONCLUSION

The analysis of "Performance Analysis of Symmetric RC4 and Asymmetric RSA Cryptography Algorithms in Securing Normal Text Messages" reveals the relative strengths of RC4 and RSA in terms of speed and security. RC4 demonstrates superior speed, while RSA exhibits superior security, particularly in the generation of random ciphertext. Furthermore, RSA is resistant to analysis using the character frequency analysis method. This evaluation furnishes a comprehensive depiction of the strengths and weaknesses of each algorithm in the context of speed and security level.

REFERENCES

[1] Ainafatul Nur Muslikah, Riswanto, H. R., Safinah, K., & Holle, K. F. H. (2020). Implementasi Teknik Kriptografi Rsa Untuk Pengamanan Data Pengiriman Sms. Jurnal Ilmiah Informatika, 5(1), 61–66. https://doi.org/10.35316/jimi.v5i1.749

[2] Arianti, T., Fa'izi, A., Adam, S., & Mira Wulandari. (2022). Perancangan Sistem Informasi Perpustakaan Menggunakan Diagram Uml (Unified Modelling Language). Jurnal Ilmiah Komputer ..., 1(1), 19–25. https://journal.polita.ac.id/index.php/politati/article/view/110/88

[3] Arif, Z., & Nurokhman, A. (2023). Analisis Perbandingan Algoritma Kriptografi Simetris Dan Asimetris Dalam Meningkatkan Keamanan Sistem Informasi Comparative Analysis of Symmetric and Asymmetric Cryptographic Algorithms in Improving Information System Security. In JTSI (Vol. 4, Issue 2).

[4] Azhari, M., Perwitosari, J., & Ali, F. (2022). Implementasi Pengamanan Data pada Dokumen Menggunakan Algoritma Kriptografi Advanced Encryption Standard (AES). Jurnal Pendidikan Sains Dan Komputer, 2(1), 2809–476. https://doi.org/10.47709/jpsk.v2i1.1390

[5] Basri, H., Teknik Elektro, P., Teknik Universitas Muhammadiyah Lampung, F., Lingkar Selatan, J., Bantul, K., Zainal Abidin Pagar Alam No, J. H., & Lampung, B. (2020). Pembuatan Aplikasi Penjualan Buku Berbasis Java Desktop dengan Netbeans (Creating a Java Desktop based Book Sales Application with Netbeans). Jurnal IlmiahTeknik Elektro UML, 1(1).

[6] Dakhi, O., Masril, M., Novalinda, R., Jufrinaldi, J., & Ambiyar, A. (2020). Analisis Sistem Kriptografi dalam Mengamankan Data Pesan Dengan Metode One Time Pad Cipher. INVOTEK: Jurnal Inovasi Vokasional Dan Teknologi, 20(1), 27–36. https://doi.org/10.24036/invotek.v20i1.647

[7] Dhika, H., Isnain, N., & Tofan, M. (2019). Manajemen Villa Menggunakan Java Netbeans Dan Mysql. IKRA-ITH INFORMATIKA : Jurnal Komputer Dan Informatika, 3(2), 104–110.

[8] Fitriana, R. N., & Djuniadi, D. (2022). Analisis Perbandingan Algoritma AES Dan RC4 Pada Enkripsi dan Dekripsi Data Teks Berbasis CrypTool 2. Systemic: Information System and Informatics Journal, 7(2), 1–7. https://doi.org/10.29080/systemic.v7i2.1263

[9] Maulana, M. M., Azanuddin, & Suharsil. (2020). Pengamanan Data Surat-Surat Berharga di Kantor Notaris Susanto, S.H., M.Kn., Menggunakan Kriptografi Dengan Metode Rives Code 4 (R4). Jurnal CyberTech, September, 1–10.

[10] Prasetya, A. F., Sintia, & Putri, U. L. D. (2022). Perancangan Aplikasi Rental Mobil Menggunakan Diagram UML (Unified Modelling Language). Jurnal Ilmiah Komputer Terapan Dan Informasi, 1(1), 14–18.

[11] Rasudin, R., Zulfan, Z., & Rizki, P. (2022). Analisis Perbandingan Keamanan Kriptografi Klasik Pada Algoritma Secure Hill Cipher Berbasis Kode Ascii Dan Monoalphabetic. Jurnal Teknologi Terapan and Sains 4.0, 3(1), 729. https://doi.org/10.29103/tts.v3i1.9411

[12] Riza, F., Sridewi, N., Husein, A. M., & Harahap, M. K. (2018). Analisa Frekuensi Hasil Enkripsi Pada Algoritma Kriptografi Blowfish Terhadap Keamanan Informasi. Jurnal Teknologi Dan Ilmu Komputer Prima (JUTIKOMP), 1(1), 11–15. https://doi.org/10.34012/jutikomp.v1i1.233

[13] Rizky Dzullian, M. (2022). Perancangan Sistem Informasi Penjualan Berbasis Java Netbeans. Blend Sains Jurnal Teknik, 1(2), 76–87. https://doi.org/10.56211/blendsains.v1i2.112

[14] Rosaly, R., & Prasetyo, A. (2020). Flowchart Beserta Fungsi dan Simbol-Simbol. Journal of Chemical Information and Modeling, 2(3), 5–7.

[15] Sary, Y., & Al-iksan, F. (2022). Cryptography Generator for Prevention SQL Injection Attack In Big Data. Journal of Computer Science, Information Technology and Telecommunication Engineering, 3(2), 292–298. https://doi.org/10.30596/jcositte.v3i2.11566

[16] Sina, D. R., Kiu, G. A., Djahi, B. S., & Pandie, E. S. Y. (2022). Aplikasi Keamanan Pesan (.Txt) Menggunakan Metode Triple Des Dan Metode Kombinasi Lsb Dan Blum-Blum-Shub. Jurnal Komputer Dan Informatika, 10(2), 204–209. https://doi.org/10.35508/jicon.v10i2.8465

[17] Siregar, S. J., Nugroho, N. B., & Sigalingging, H. (2023). Implementasi Algoritma Kriptografi RSA (Rivest Shamir Adleman) Dalam Pengamanan Data Gaji Karyawan Di Kantor BSPJI. Jurnal SAINTIKOM (Jurnal Sains Manajemen Informatika Dan Komputer), 22(2), 528. https://doi.org/10.53513/jis.v22i2.9409

[18] Suhandinata, S., Rizal, R. A., Wijaya, D. O., Warren, P., & Srinjiwi, S. (2019). ANALISIS PERFORMA KRIPTOGRAFI HYBRID ALGORITMA BLOWFISH DAN ALGORITMA RSA. JURTEKSI (Jurnal Teknologi Dan Sistem Informasi), 6(1), 1–10. https://doi.org/10.33330/jurteksi.v6i1.395

[19] Widyastuti, R., Amelia, R., Gea, Y. Y., Murlena, M., & Syahindra, W. (2022). Perancangan Aplikasi Administrasi Pada Fakultas Keguruan dan Ilmu Pendidikan Menggunakan Java Netbeans Ide 8.1 dan Mysql. Arcitech: Journal of Computer Science and Artificial Intelligence, 2(2), 103. https://doi.org/10.29240/arcitech.v2i2.6498

[20] Widyawan, D., & Imelda, I. (2021). Pengamanan File Menggunakan Kriptografi Dengan Metode Aes-128 Berbasis Web Di Komite Nasional Keselamatan Transportasi. Skanika, 4(1), 15–22. https://doi.org/10.36080/skanika.v4i1.2216

[21] Yanto, M., Bima, P. E., Bahron, M., & Ikasari, I. H. (2023). Pemrograman Menggunakan Java NetBeans. BIIKMA : Buletin Ilmiah Ilmu Komputer Dan Multimedia, 1(3), 367–377. https://jurnalmahasiswa.com/index.php/biikma/article/view/555