JURNAL LOCUS: Penelitian & Pengabdian
Volume 4 No. 8 Agustus 2025
E-ISSN 2829-7334| P-ISSN 2829-5439
Hompage: https://locus.rivierapublishing.id/index.php/jl

Implementasi Algoritma Enkripsi Blowfish dan Rijndael dalam Pengamanan File Text

Desnantia Eka Ramadhani[1*], Puspita Nurul Sabrina[2*], Herdi Ashaury[3*]
[1,2,3]Universitas Jenderal Achmad Yani, Indonesia
ramacahbarat@gmail.com[1]*,puspita.sabrina@lecture.unjani.ac.id[2*],
herdi.ashaury@lecture.unjani.ac.id[3*]

Abstrak:
Kemajuan teknologi informasi telah meningkatkan kebutuhan akan pengamanan data digital, terutama untuk file teks yang rentan terhadap ancaman seperti pencurian dan manipulasi. Penelitian ini bertujuan untuk menganalisis dan membandingkan performa dua algoritma kriptografi simetris, yaitu Blowfish dan Rijndael (AES), dalam mengamankan file teks. Implementasi dilakukan menggunakan bahasa pemrograman Python dengan fokus pada evaluasi kecepatan enkripsi-dekripsi, perubahan ukuran file, serta akurasi hasil dekripsi. Metode penelitian yang digunakan adalah eksperimen kuantitatif, di mana kedua algoritma diuji pada file teks dengan berbagai ukuran untuk mengukur efisiensi dan keandalannya. Hasil penelitian menunjukkan bahwa Blowfish lebih unggul dalam kecepatan pemrosesan file berukuran kecil, sedangkan Rijndael (AES) lebih konsisten dan efisien untuk file berukuran besar. Selain itu, Rijndael juga lebih tahan terhadap serangan modern seperti Sweet32, yang menjadi kelemahan Blowfish. Penelitian ini memberikan rekomendasi praktis dalam pemilihan algoritma enkripsi berdasarkan kebutuhan sistem dan karakteristik data. Dengan demikian, temuan ini berkontribusi pada peningkatan keamanan data digital dan dapat menjadi acuan dalam pengembangan sistem enkripsi yang lebih efektif.

Kata Kunci: enkripsi; Blowfish; rijindael; kriptografi simetris; keamanan data

Abstract:
Advances in information technology have heightened the need for securing digital data, particularly text files vulnerable to threats like theft and manipulation. This study aims to analyze and compare the performance of two symmetric cryptographic algorithms, Blowfish and Rijndael (AES), in securing text files. The implementation was conducted using the Python programming language, focusing on evaluating encryption-decryption speed, file size changes, and decryption accuracy. A quantitative experimental method was employed, testing both algorithms on text files of varying sizes to measure their efficiency and reliability. The results indicate that Blowfish excels in processing speed for small files, while Rijndael (AES) demonstrates more consistent efficiency for larger files. Additionally, Rijndael shows greater resistance to modern attacks like Sweet32, a known vulnerability of Blowfish. This study provides practical recommendations for selecting encryption algorithms based on system requirements and data characteristics. These findings contribute to enhancing digital data security and serve as a reference for developing more effective encryption systems.

Keywords: *Encryption; Blowfish; Rijindael; Symmetric Cryptography; Data Security*

Corresponding: Desnantia Eka Ramadhani
E-mail: ramacahbarat@gmail.com

PENDAHULUAN

Di era digital yang terus berkembang pesat, informasi telah menjadi aset yang sangat berharga, baik bagi individu, organisasi, maupun bisnis (Harto et al., 2023; Joesyiana, Basriani, & Susanti, 2024). Ketergantungan yang semakin tinggi pada data digital, mulai dari komunikasi pribadi, transaksi finansial, hingga penyimpanan dokumen penting, secara signifikan meningkatkan risiko terhadap berbagai ancaman siber (Chic & Bilqisthi, 2024; Nova, Hamzah, & Unsong, 2024). Ancaman seperti pencurian data, manipulasi informasi, dan akses tidak sah menjadi perhatian serius yang menuntut adanya langkah-langkah perlindungan data yang kuat untuk menjaga privasi, integritas, dan kerahasiaan informasi sensitif, termasuk file teks (Gamaliel et al., 2024).

Dalam menghadapi tantangan keamanan siber, kriptografi diakui sebagai teknik fundamental untuk mengamankan data melalui fungsi matematis yang mengubah data menjadi ciphertext yang tidak bisa dibaca tanpa kunci yang tepat (ScienceDirect Topics, 2025; Rana, 2023). Di antara kategori kriptografi, kriptografi kunci simetris menonjol karena efisiensi dan kecepatannya—menjadikannya pilihan unggul untuk mengenkripsi volume data besar seperti file (Balli, 2020; tunmise Adewale, 2025). AES, algoritma simetris yang paling umum digunakan, secara konsisten mengungguli DES dan RSA dalam hal kecepatan enkripsi dan penggunaan sumber daya pada berbagai lingkungan komputasi (Adewale, 2025). Implementasi yang dioptimalkan—seperti GPU-accelerated AES—telah terbukti sangat efektif untuk pipeline data besar di lingkungan cloud (Yang et al., 2025). Benchmarking juga menunjukkan bahwa kunci simetris seperti AES menyediakan throughput tinggi dengan konsumsi memori yang relatif kecil, membuatnya sesuai untuk sistem real-time (Balli, 2020; Istifan & Makovac, 2022). Analisis terbaru tentang manajemen kunci kriptografis juga menyoroti pentingnya protokol efisien untuk menjaga keamanan dan kinerja sistem simetris (Rana, 2023). Karena satu kunci rahasia sama dipakai untuk enkripsi dan dekripsi, metode ini sangat cepat dan hemat sumber daya dibandingkan metode asimetris (ScienceDirect Topics, 2025; Balli, 2020; Adewale, 2025).

Penelitian ini secara spesifik berfokus pada implementasi dan perbandingan dua algoritma *block cipher* kunci simetris yang telah dikenal luas: Blowfish dan Rijndael (*Advanced Encryption Standard/AES*). Blowfish (Saputra & Widyanto, 2023), yang dirancang oleh Bruce Schneier pada tahun 1993, sempat populer karena kecepatan dan sifatnya yang bebas royalti, beroperasi pada blok 64-bit dengan panjang kunci variabel (Haryono, 2020). Di sisi lain, Rijndael, yang kemudian distandardisasi oleh National Institute of Standards and Technology (NIST) pada tahun 2001 sebagai AES, telah menjadi "standar emas" global untuk perlindungan data. AES beroperasi pada blok 128-bit dan dikenal karena keamanan, efisiensi tinggi, serta dukungan akselerasi perangkat keras modern (AES-NI).

Meskipun keduanya merupakan algoritma enkripsi simetris yang kuat, terdapat perbedaan signifikan yang memengaruhi kecocokannya dalam konteks pengamanan file teks modern. Blowfish, dengan ukuran blok 64-bitnya, telah diidentifikasi rentan terhadap serangan seperti Sweet32 ketika mengenkripsi data dalam jumlah besar (lebih dari 4GB), bahkan perancangnya sendiri merekomendasikan untuk beralih ke algoritma yang lebih baru seperti Twofish (Riza et al., 2018).

Penelitian sebelumnya oleh Saputra dan Widyanto (2023) mengimplementasikan algoritma Blowfish untuk enkripsi file teks dan menemukan bahwa algoritma ini efisien untuk file berukuran kecil, namun tidak membahas performanya pada file besar atau kerentanannya terhadap serangan modern. Di sisi lain, penelitian oleh Anand Kumar dan Karthikeyan (2012) membandingkan Blowfish dan AES dalam konteks kecepatan enkripsi, tetapi tidak mengeksplorasi dampak perubahan ukuran file atau manajemen kunci yang optimal. Kedua penelitian ini memiliki gap dalam hal analisis komprehensif terhadap performa kedua algoritma dalam berbagai skenario pengamanan file teks, termasuk kecepatan, perubahan ukuran file, dan kerentanan terhadap serangan. Penelitian ini bertujuan untuk mengisi gap tersebut dengan melakukan analisis mendalam terhadap implementasi Blowfish dan Rijndael (AES), termasuk evaluasi kecepatan enkripsi-dekripsi, perubahan ukuran file, serta analisis kerentanan seperti serangan Sweet32 pada Blowfish.

Sebaliknya, AES terus menjadi pilihan utama karena ukuran bloknya yang lebih besar, ketahanannya terhadap kriptanalisis yang diketahui, dan adopsi luas dalam standar industri serta regulasi global. Selain itu, kinerja AES pada sistem modern seringkali jauh melampaui Blowfish berkat optimasi perangkat keras. Oleh karena itu, implementasi dan perbandingan kinerja serta keamanan kedua algoritma ini dalam pengamanan file teks menjadi sangat relevan. Penelitian ini bertujuan untuk memberikan wawasan praktis mengenai efektivitas Blowfish dan Rijndael dalam melindungi kerahasiaan dan integritas file teks, serta mengidentifikasi praktik terbaik untuk penerapannya di tengah lanskap ancaman siber yang terus berkembang (Riza et al., 2018). Hasil penelitian ini diharapkan dapat memberikan rekomendasi praktis dalam pemilihan algoritma enkripsi berdasarkan kebutuhan sistem dan karakteristik data, sehingga berkontribusi pada peningkatan keamanan data digital di berbagai aplikasi.

METODE PENELITIAN

Mengenai Metode Penelitian yang dilakukan peneliti ada beberapa metode yang dilakukan
Yaitu:

Pendekatan penelitian

Penelitian ini menggunakan metode pengembangan sistem berbasis eksperimen, dengan pendekatan yang sistematis dalam merancang, membangun, dan menguji kinerja algoritma enkripsi. Langkah pertama yang dilakukan adalah **mengkaji pustaka**, yaitu menelaah berbagai referensi yang membahas algoritma Blowfish dan Rijndael (AES), kriptografi simetris, serta teknik pengamanan file teks. Kegiatan ini bertujuan untuk memperoleh pemahaman teoritis yang kuat sebagai dasar pelaksanaan penelitian.Tahap selanjutnya adalah **merancang sistem enkripsi dan dekripsi**, yang mencakup perencanaan alur kerja sistem, perumusan struktur program, serta penentuan parameter pengujian, seperti waktu proses enkripsi-dekripsi, perubahan ukuran file setelah dienkripsi, dan keakuratan hasil dekripsi. Setelah desain sistem selesai, proses dilanjutkan ke tahap **implementasi**, yaitu mengembangkan dua aplikasi enkripsi-dekripsi menggunakan bahasa pemrograman Python. Setiap algoritma—Blowfish dan Rijndael—diimplementasikan dalam modul terpisah untuk memudahkan pengujian dan

perbandingan. Langkah berikutnya adalah **melakukan pengujian dan pengumpulan data**, di mana file teks dengan ukuran dan isi yang bervariasi dienkripsi dan didekripsi menggunakan kedua algoritma. Selama proses ini, waktu pemrosesan, ukuran file hasil enkripsi, dan kesesuaian hasil dekripsi dengan file aslinya dicatat sebagai data penelitian.Tahap akhir berupa **analisis dan penarikan kesimpulan** dari hasil yang diperoleh. Data yang terkumpul dianalisis untuk mengevaluasi kinerja masing-masing algoritma, baik dari segi kecepatan, efisiensi, maupun keakuratan dalam menjaga integritas data. Berdasarkan temuan tersebut, disusun kesimpulan mengenai efektivitas algoritma dalam pengamanan file teks, serta rekomendasi penggunaannya sesuai dengan karakteristik kebutuhan sistem

Desain Penelitian

Penelitian ini menggunakan pendekatan eksperimen kuantitatif, di mana peneliti mengimplementasikan dua algoritma enkripsi—Blowfish dan Rijndael (AES)—dalam bentuk program komputer, kemudian membandingkan performa keduanya melalui serangkaian pengujian terhadap file teks. Tujuan dari penelitian ini adalah untuk menilai seberapa efektif dan efisien kedua algoritma tersebut dalam melindungi data teks.

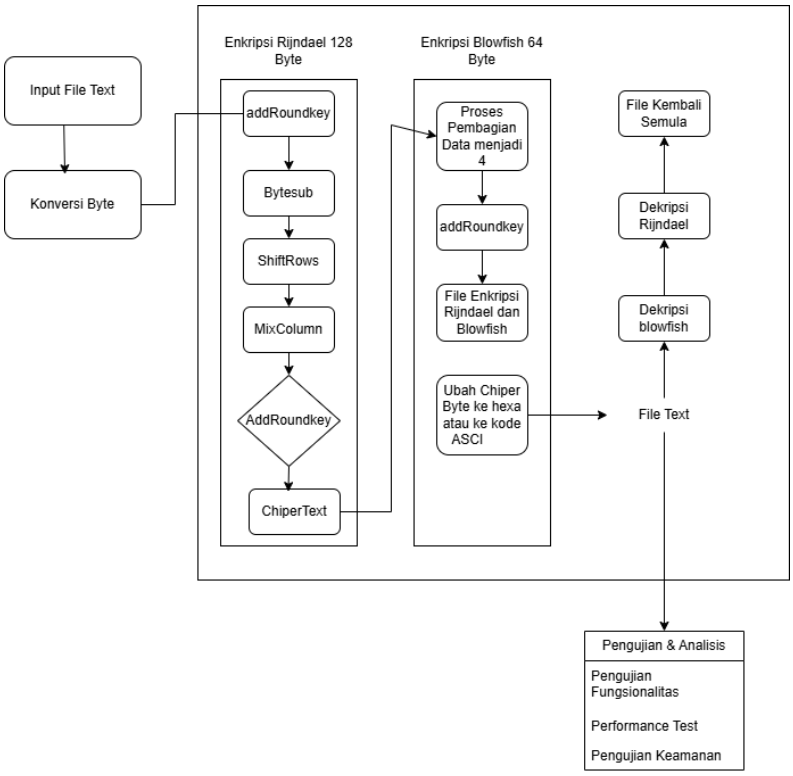
Langkah-langkah penelitian dilakukan secara sistematis. Dimulai dari identifikasi permasalahan dan pengumpulan teori-teori dasar, peneliti mempelajari secara mendalam konsep kriptografi simetris, mekanisme kerja algoritma Blowfish dan Rijndael, serta hasil-hasil penelitian sebelumnya yang berkaitan. Selanjutnya, dilakukan proses perancangan sistem, di mana masing-masing algoritma dimasukkan ke dalam aplikasi yang dapat melakukan proses enkripsi dan dekripsi terhadap file teks.

Bagian utama dari desain penelitian ini adalah implementasi program menggunakan bahasa Python. Dua modul aplikasi dikembangkan secara terpisah, satu menerapkan algoritma Blowfish dan yang lainnya menggunakan Rijndael. Keduanya kemudian diuji terhadap file teks dengan berbagai ukuran.

Desain ini juga melibatkan proses pengujian dan pengumpulan data kuantitatif, antara lain:

- Waktu yang dibutuhkan untuk proses enkripsi dan dekripsi
- Perbandingan ukuran file sebelum dan sesudah dienkripsi
- Keakuratan hasil dekripsi dibandingkan dengan file asli

Data yang diperoleh dari pengujian kemudian dianalisis untuk mengevaluasi keunggulan dan kelemahan dari masing-masing algoritma berdasarkan kinerja dan efektivitasnya. Dengan pendekatan eksperimental ini, peneliti memperoleh hasil yang objektif dan terukur dalam menilai penggunaan algoritma Blowfish dan Rijndael untuk pengamanan file teks.



Gambar 1. Analisis Kinerja Algoritma Blowfish dan Rijndael (AES) dalam Pengamanan File Teks
Sumber: Dokumen Penelitian, 2024

1. **Prosedur Pengumpulan Data**

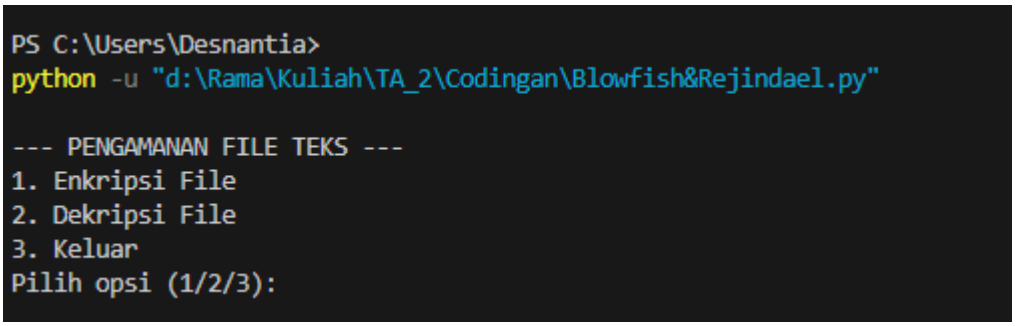
   Prosedur pengumpulan data dalam penelitian ini dilakukan dengan memanfaatkan Sumber internet publik dan private yang tersedia di platform Google. Data yang digunakan Berupa File Doc,Txt dan File Pdf.

2. **Teknik Pengumpulan Data**

   Teknik pengumpulan data dalam penelitian ini dilakukan melalui Pengumpulan File Text Yang berada Di Internet Dan File-File Text lainnya, dengan memanfaatkan data yang tersedia secara publik melalui platform Google dan Sumber Website lainnya. Penggunaan dokumen sebagai teknik pengumpulan data.

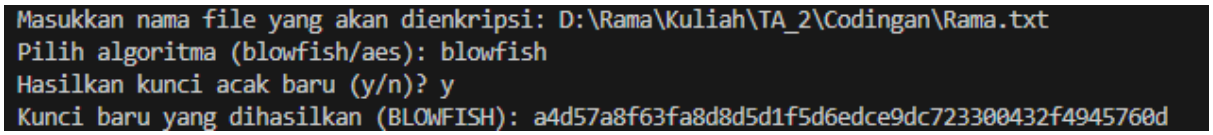
HASIL DAN PEMBAHASAN

Pengguna dapat memulai program melalui IDE atau Command Prompt dengan mengetikkan nama proyek yang ingin dijalankan. Setelah program berhasil dijalankan, antarmuka akan tampil seperti pada gambar berikut. Selanjutnya, pengguna akan diberi pilihan untuk melakukan proses enkripsi atau dekripsi sesuai kebutuhan.



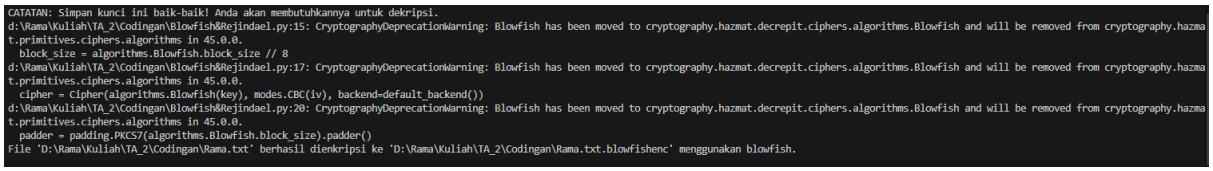
Gambar 2. Antarmuka Program Enkripsi/Deskripsi File Teks
Sumber: Dokumen Penelitian, 2024

Setelah pengguna memilih opsi enkripsi, program akan meminta pengguna untuk menentukan file yang ingin dienkripsi. Pengguna kemudian memasukkan nama file tersebut. Selanjutnya, program akan menampilkan pilihan algoritma enkripsi yang akan digunakan, yaitu Blowfish atau Rijndael. Setelah pengguna memilih algoritma, program akan memberikan opsi tambahan, apakah ingin menggunakan kunci acak yang dihasilkan secara otomatis atau memasukkan kunci buatan sendiri.



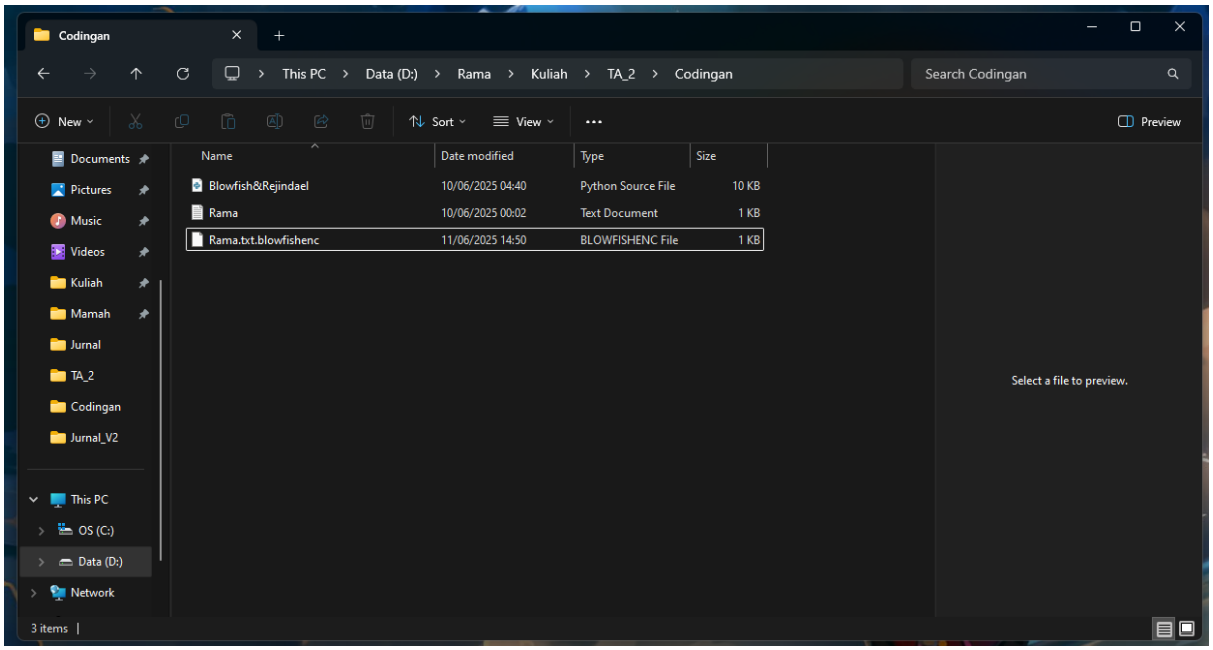
Gambar 3. Proses Pemilihan File dan Algoritma Enkripsi
Sumber: Dokumen Penelitian, 2024

Program akan mengeksekusi input yang diberikan oleh pengguna, kemudian melakukan proses enkripsi terhadap file yang telah ditentukan. Setelah proses enkripsi selesai, program akan menampilkan pesan seperti yang terlihat pada gambar di bawah. Perlu diperhatikan, kunci acak yang dihasilkan selama proses enkripsi sangat penting dan tidak boleh hilang, karena kunci tersebut dibutuhkan untuk mengembalikan file ke bentuk semula agar dapat dibuka kembali.

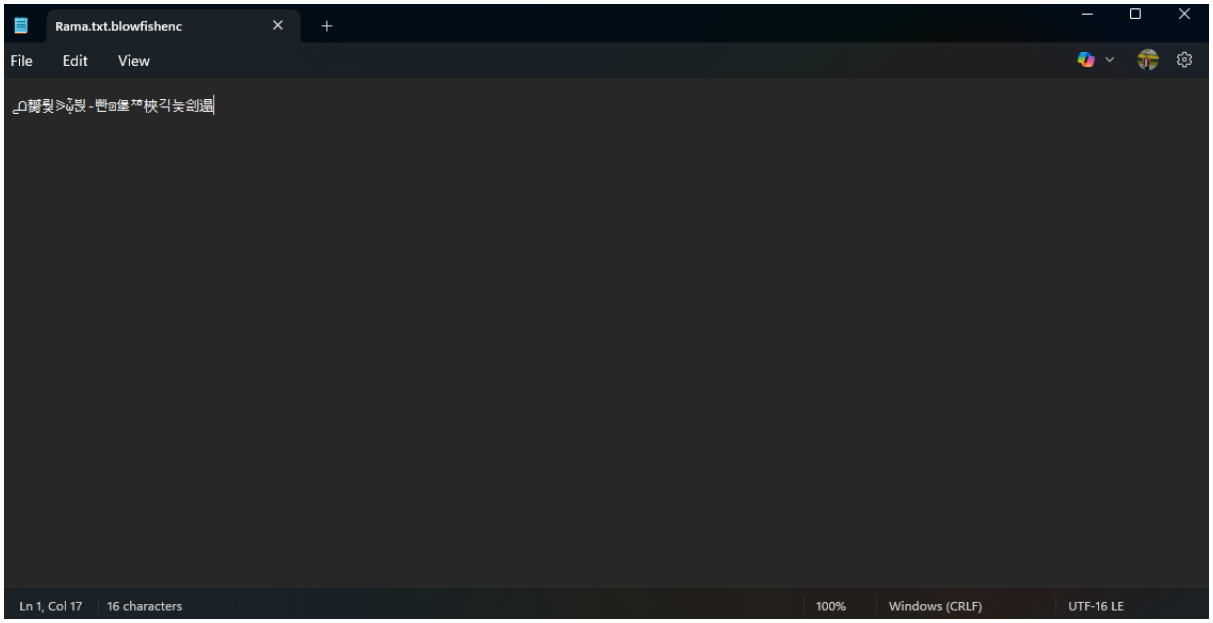


Gambar 4. Konfirmasi Proses Enkripsi Berhasil
Sumber: Dokumen Penelitian, 2024

Berikut ini adalah tampilan file setelah berhasil dienkripsi. Awalnya, file tersebut berisi teks ”Desnantia Eka Ramadhani”, namun setelah proses enkripsi dilakukan, isi file berubah menjadi sekumpulan karakter acak seperti yang ditampilkan, sebagai hasil dari perlindungan data melalui algoritma enkripsi.



Gambar 5. Hasil Enkripsi File Teks
Sumber: Dokumen Penelitian, 2024



Gambar 6. Tampilan File Terenkripsi
Sumber: Dokumen Penelitian, 2024

Untuk melakukan dekripsi file, langkah yang dilakukan pengguna serupa dengan proses enkripsi. Pengguna terlebih dahulu memilih opsi dekripsi pada program. Setelah itu, program akan meminta pengguna untuk memasukkan nama file yang ingin didekripsi. Program kemudian mengeksekusi perintah tersebut dan meminta kunci acak atau kode enkripsi yang digunakan sebelumnya. Jika pengguna tidak memiliki kunci tersebut, maka proses dekripsi tidak dapat dilakukan, dan isi file tidak akan bisa dibuka atau dibaca.

```
--- PENGAMANAN FILE TEKS ---
1. Enkripsi File
2. Dekripsi File
3. Keluar
Pilih opsi (1/2/3): 2
Masukkan nama file yang akan didekripsi: D:\Rama\Kuliah\TA_2\Codingan\Rama.txt.blowfishenc
Pilih algoritma yang digunakan untuk enkripsi (blowfish/aes): blowfish
Masukkan kunci blowfish (dalam heksadesimal, misal: '0123456789abcdef...'): dfcb62f13dcbab4bc0b166d2ae6999548ffe927ebed0ba60
```

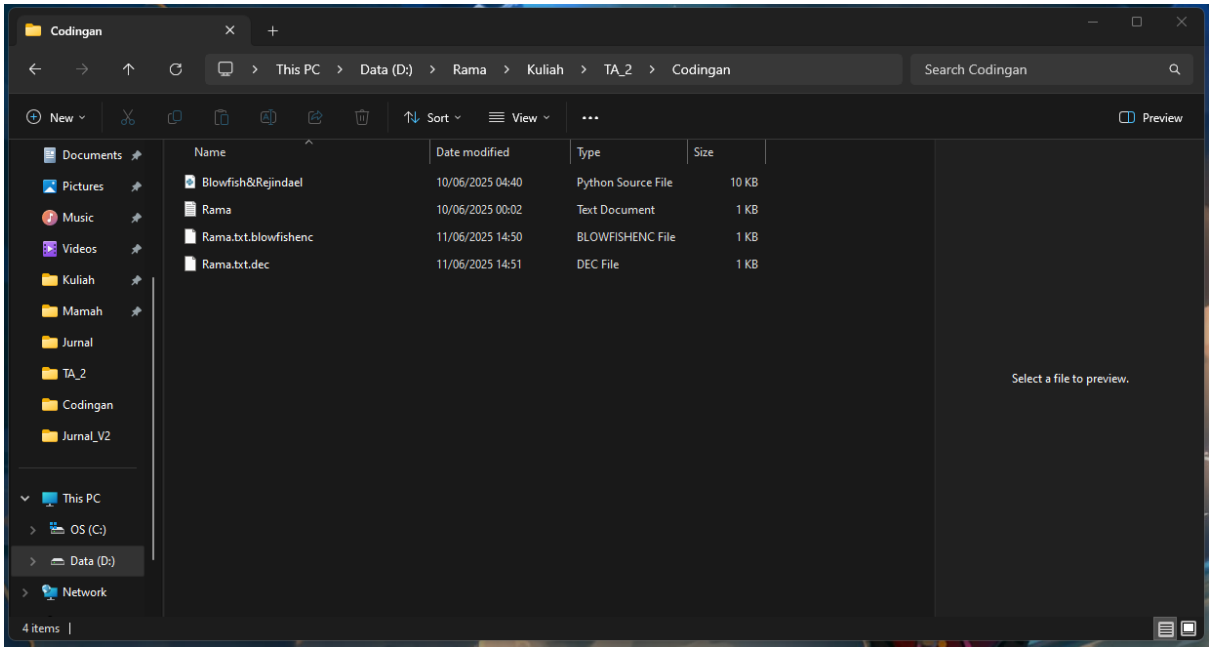
Gambar 7. Proses Dekripsi File
Sumber: Dokumen Penelitian, 2024

Setelah pengguna memasukkan kunci dekripsi, program akan memproses perintah tersebut dan menjalankan proses dekripsi pada file yang sebelumnya telah dienkripsi. Jika kunci yang dimasukkan benar, maka file akan berhasil dikembalikan ke bentuk semula dan dapat dibuka serta dibaca kembali.

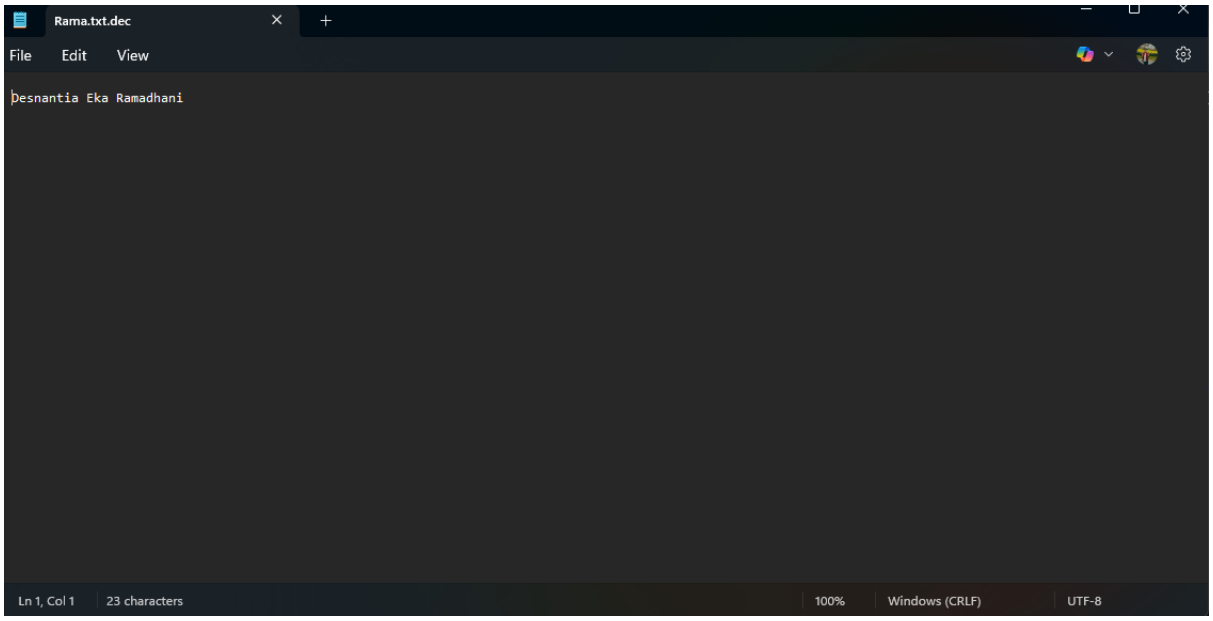
```
d:\Rama\Kuliah\TA_2\Codingan\Blowfish&Rejindael.py:33: CryptographyDeprecationWarning: Blowfish has been moved to cryptography.hazmat.decrepit.ciphers.algorithms.Blowfish and will be removed from cryptography.hazmat.primitives.ciphers.algorithms in 45.0.0.
  block_size = algorithms.Blowfish.block_size // 8
d:\Rama\Kuliah\TA_2\Codingan\Blowfish&Rejindael.py:37: CryptographyDeprecationWarning: Blowfish has been moved to cryptography.hazmat.decrepit.ciphers.algorithms.Blowfish and will be removed from cryptography.hazmat.primitives.ciphers.algorithms in 45.0.0.
  cipher = Cipher(algorithms.Blowfish(key), modes.CBC(iv), backend=default_backend())
d:\Rama\Kuliah\TA_2\Codingan\Blowfish&Rejindael.py:42: CryptographyDeprecationWarning: Blowfish has been moved to cryptography.hazmat.decrepit.ciphers.algorithms.Blowfish and will be removed from cryptography.hazmat.primitives.ciphers.algorithms in 45.0.0.
  unpadder = padding.PKCS7(algorithms.Blowfish.block_size).unpadder()
Error dekripsi: Invalid padding bytes.
```

Gambar 8. Konfirmasi Dekripsi Berhasil
Sumber: Dokumen Penelitian, 2024

Berikut ini merupakan hasil dari file yang telah berhasil didekripsi. Nama file yang telah didekripsi akan secara otomatis ditambahkan ekstensi ".txt.dec" di belakangnya, sehingga pengguna dapat dengan mudah membedakan antara file yang sudah didekripsi dan file yang masih dalam bentuk terenkripsi.



Gambar 9. Hasil Dekripsi File Teks
Sumber: Dokumen Penelitian, 2024



Gambar 10. Perbandingan File Sebelum/Sesudah Enkripsi
Sumber: Dokumen Penelitian, 2024

Metode penelitian yang Anda adopsi, yang secara sistematis melalui tahapan Studi Literatur, Analisis Kebutuhan, Pengumpulan Data, Perancangan Sistem, Pengembangan Prototipe Sistem, Pengujian dan Analisis, serta Pembuatan Laporan Akhir dan Publikasi, adalah kerangka kerja yang sangat menyeluruh dan terstruktur untuk mengimplementasikan dan mengevaluasi algoritma enkripsi Blowfish dan Rijndael dalam pengamanan file teks. Pendekatan ini secara efektif memulai dengan membangun fondasi teoretis yang kokoh melalui studi literatur yang mendalam tentang kriptografi, algoritma spesifik, dan mode operasi, memastikan pemahaman komprehensif sebelum beralih ke aplikasi praktis. Selanjutnya, analisis kebutuhan yang teliti memungkinkan identifikasi persyaratan fungsional dan non-fungsional yang jelas, seperti keamanan, kinerja, dan kegunaan, yang kemudian memandu seluruh proses pengembangan. Dengan perancangan sistem yang modular dan aman, termasuk pemilihan mode operasi yang tepat (misalnya, memprioritaskan mode terautentikasi seperti GCM/CCM) dan praktik manajemen kunci yang kuat, solusi yang dikembangkan akan memiliki dasar yang solid. Tahap pengembangan prototipe sistem kemudian mewujudkan desain ini menjadi implementasi yang berfungsi, dengan penekanan pada penggunaan pustaka yang aman dan praktik pengkodean yang baik untuk mencegah kerentanan. Puncak dari metodologi ini adalah pengujian dan analisis yang ketat, yang tidak hanya memverifikasi fungsionalitas tetapi juga secara objektif mengukur kinerja (kecepatan, throughput, penggunaan sumber daya) dan mengevaluasi keamanan melalui berbagai teknik seperti tinjauan kode dan pengujian penetrasi, termasuk analisis kerentanan spesifik seperti serangan Sweet32 pada Blowfish. Akhirnya, pembuatan laporan akhir dan publikasi memastikan bahwa semua temuan, analisis, dan rekomendasi disajikan secara jelas dan didukung oleh data, memberikan wawasan berharga tentang kesesuaian relatif Blowfish dan Rijndael untuk pengamanan file teks di lingkungan modern.. Pendekatan ini secara efektif memulai dengan membangun fondasi teoretis yang kokoh melalui studi literatur yang mendalam tentang

kriptografi, algoritma spesifik, dan mode operasi, memastikan pemahaman komprehensif sebelum beralih ke aplikasi praktis. Selanjutnya, analisis kebutuhan yang teliti memungkinkan identifikasi persyaratan fungsional dan non-fungsional yang jelas, seperti keamanan, kinerja, dan kegunaan, yang kemudian memandu seluruh proses pengembangan. Dengan perancangan sistem yang modular dan aman, termasuk pemilihan mode operasi yang tepat (misalnya, memprioritaskan mode terautentikasi seperti GCM/CCM) dan praktik manajemen kunci yang kuat, solusi yang dikembangkan akan memiliki dasar yang solid. Tahap pengembangan prototipe sistem kemudian mewujudkan desain ini menjadi implementasi yang berfungsi, dengan penekanan pada penggunaan pustaka yang aman dan praktik pengkodean yang baik untuk mencegah kerentanan. Puncak dari metodologi ini adalah pengujian dan analisis yang ketat, yang tidak hanya memverifikasi fungsionalitas tetapi juga secara objektif mengukur kinerja (kecepatan, throughput, penggunaan sumber daya) dan mengevaluasi keamanan melalui berbagai teknik seperti tinjauan kode dan pengujian penetrasi, termasuk analisis kerentanan spesifik seperti serangan Sweet32 pada Blowfish. Akhirnya, pembuatan laporan akhir dan publikasi memastikan bahwa semua temuan, analisis, dan rekomendasi disajikan secara jelas dan didukung oleh data, memberikan wawasan berharga tentang kesesuaian relatif Blowfish dan Rijndael untuk pengamanan file teks di lingkungan modern.

KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa baik algoritma Blowfish maupun Rijndael (AES) memiliki keunggulan masing-masing dalam pengamanan file teks. Blowfish terbukti lebih efisien dalam menangani file berukuran kecil dengan waktu enkripsi yang lebih cepat, sementara AES menunjukkan performa yang lebih konsisten dan aman untuk file berukuran besar. Dari segi keamanan, AES memiliki keunggulan dengan ukuran blok yang lebih besar dan ketahanan terhadap serangan modern. Kedua algoritma menyebabkan peningkatan ukuran file setelah proses enkripsi, namun tetap mempertahankan integritas data asli. Temuan ini memberikan panduan berharga bagi pengembang sistem dalam memilih algoritma enkripsi yang sesuai dengan kebutuhan spesifik, baik untuk aplikasi yang memprioritaskan kecepatan maupun keamanan tinggi. Penelitian ini juga membuka peluang untuk pengembangan lebih lanjut, termasuk eksplorasi implementasi hybrid system yang memadukan keunggulan kedua algoritma serta pengujian pada lingkungan komputasi modern. Hasil penelitian ini diharapkan dapat menjadi referensi bagi praktisi IT dan pengembang sistem dalam merancang solusi enkripsi yang optimal untuk berbagai skenario pengamanan data.

DAFTAR PUSTAKA

Adewale, T. (2025). *Performance benchmarking of DES, AES and RSA in modern computing environments*. ResearchGate. https://www.researchgate.net/

Anand Kumar, M., & Karthikeyan, S. (2012). Investigating the efficiency of Blowfish and Rejindael (AES) algorithms. *International Journal of Computer Network and Information Security, 4*(2), 22–28. https://doi.org/10.5815/ijcnis.2012.02.04

Balli, S. (2020). Multi criteria usability evaluation of symmetric data encryption algorithms. *SN Applied Sciences*. SpringerLink. https://link.springer.com/

Chic, Sebastian Areen, & Bilqisthi, Muhammad Fardian. (2024). Tantangan dan Peluang Blockchain

di Era Digital dalam Bidang Keamanan Data dan Transaksi Digital. *Journal of Comprehensive Science (JCS)*, *3*(11).

Gamaliel, F., Yudi, P., & Arliyanto, D. (2024). Implementasi algoritma Blowfish untuk pengamanan file PDF. *Jurnal Ilmu Rekayasa dan Elektronika, 2*(1). http://e-journal.stmiklombok.ac.id/index.php/jireISSN.2620-6900

Haryono, W. (2020). Comparison encryption of how to work Caesar cipher, Hill cipher, Blowfish and Twofish. *Data Science: Journal of Computing and Applied Informatics, 4*(2), 100–110. https://doi.org/10.32734/jocai.v4.i2-4004

Harto, Budi, Rukmana, Arief Yanto, Subekti, Rino, Tahir, Rusdin, Waty, Ervina, Situru, Agatha Christy, & Sepriano, Sepriano. (2023). *Transformasi bisnis di era digital: Teknologi informasi dalam mendukung transformasi bisnis di era digital*. PT. Sonpedia Publishing Indonesia.

Istifan, S., & Makovac, M. (2022). *Performance benchmarking of data at rest encryption in relational databases with AES variations* (Bachelor thesis). DIVA Portal. https://www.diva-portal.org/

Joesyiana, Kiki, Basriani, Agustin, & Susanti, Desi. (2024). Implementasi Manajemen Sumber Daya Manusia (MSDM) Berbasis Kreatifitas Individu Di Era Digital. *Journal of Tax and Business*, *5*(2), 376–384.

Nova, Vivian, Hamzah, Hamdan, & Unsong, Imelda F. (2024). Merancang strategi cerdas bisnis inovatif di tengah revolusi digital yang terus berkembang. *Meraja journal*, *7*(3), 26–40.

Rana, S. (2023). *A comprehensive survey of cryptography key management systems*. ScienceDirect. https://www.sciencedirect.com/

Riza, F., Sridewi, N., Husein, A. M., & Harahap, M. K. (2018). Analisa frekuensi hasil enkripsi algoritma Blowfish terhadap keamanan informasi. *[Artikel belum diterbitkan atau informasi tidak lengkap]*

Saputra, A., & Widyanto, A. (2023). Enkripsi dan dekripsi file dengan algoritma Blowfish. *[Artikel belum diterbitkan atau informasi tidak lengkap]*

ScienceDirect Topics. (2025). *Symmetric cryptography – an overview*. ScienceDirect. https://www.sciencedirect.com/topics/

Yang, M. K., et al. (2025). Optimized hybrid CPU–GPU workflow for efficient AES encryption in cloud environments. *Applied Sciences, 15*(7), 3863. https://www.mdpi.com/2076-3417/15/7/3863