JURNAL RISET SISTEM INFORMASI
Halaman Jurnal: https://journal.smartpublisher.id/index.php/jissi
Halaman UTAMA Jurnal : https://journal.smartpublisher.id

DOI: https://doi.org/10.69714/k6pr0s45

IMPLEMENTASI ALGORITMA KRIPTOGRAFI AES DALAM PENGAMANAN FILE TEKS

Pazrian Nurul Latip [a*]
[a] Teknik Informatika, nurulfazri54@gmail.com, STMIK Indonesia Mandiri, kota dan provinsi perguruan tinggi
[*] Korespondensi

ABSTRACT

In today's digital world, data security is crucial for communication and document storage, especially for text files containing sensitive information. The focus of this research is the use of modern cryptographic algorithms Advanced Encryption Standard (AES) in text file encryption. The research process includes the creation of a prototype application that uses (AES). After developing the prototype, we tested its performance both in terms of processing speed and security level. From there it is known: AES is much more efficient than Vigenère. Vigenère may have strong encryption, but unfortunately it is easily broken by cryptographic analysis.

Keywords: *Cryptography, AES, Text File Security, Encryption, Decryption, Information Security.*

Abstrak

Di dunia digital sekarang, keamanan data itu krusial untuk komunikasi dan penyimpanan dokumen, terutama berlaku buat file teks yang isinya informasi sensitif. Fokus dari penelitian ini adalah penggunaan algoritma kriptografi modern Advanced Encryption Standard (AES) dalam pengekripsi teks file. Proses penelitian mencakup pembuatan prototipe aplikasi yang menggunakan (AES). Setelah mengembangkan prototipe, kami menguji performanya baik kecepatan proses maupun tingkat keamanannya. Dari situ diketahui: AES jauh lebih efisien daripada Vigenère. Vigenère mungkin punya enkripsi kuat, tapi sayangnya mudah dipecahkan dengan analisis kriptograf.

Kata Kunci: Kriptografi, AES, Pengamanan File Teks, Enkripsi, Dekripsi, Keamanan Informasi.

1. PENDAHULUAN

Di zaman serba digital, pertukaran dan simpan data itu penting baik buat urusan pribadi, bisnis, maupun pemerintah. Tapi karena akses data sekarang mudah, muncul kekhawatiran serius: kebocoran data atau malah kehilangan data. Apalagi file teks yang isinya rahasia seperti (laporan keuangan atau data pribadi), sering jadi buruan pihak yang tidak bertanggung jawab. Maka butuh sistem proteksi data yang ampuh. kriptografi adalah jawaban daari kekhawatiran itu, ilmu kriptografi menjadi dasar untuk keamanan data..

Dalam era digital saat ini, perlindungan terhadap data menjadi aspek yang sangat penting, khususnya pada proses penyimpanan dan pertukaran informasi dalam bentuk file teks. Kriptografi merupakan salah satu solusi utama dalam menjaga kerahasiaan data, dan algoritma Advanced Encryption Standard (AES) dikenal sebagai salah satu metode enkripsi simetris yang paling kuat dan efisien. Penelitian ini bertujuan untuk mengimplementasikan algoritma AES dalam proses enkripsi dan dekripsi file teks guna menjamin keamanan informasi. Selain itu, penelitian ini juga berfokus pada analisis waktu yang dibutuhkan dalam proses enkripsi dan dekripsi sebagai indikator efisiensi algoritma. Aspek keamanan dari implementasi ini turut dianalisis untuk menilai sejauh mana algoritma mampu melindungi data dari potensi ancaman. Terakhir, efektivitas keseluruhan dari penerapan algoritma AES dalam konteks ini juga dievaluasi guna melihat kecocokannya dalam aplikasi praktis pengamanan file.

Received April 9, 2025; Revised April 15, 2025; Accepted Juni 2, 2025; Online Available Juni 17, 2025

Diharapkan bahwa hasil penelitian ini akan memberikan gambaran yang jelas tentang kekuatan dan kelemahan AES dan berfungsi sebagai panduan untuk memilih algoritma kriptografi yang sesuai untuk aplikasi analisis file teks.

2. METODOLOGI PENELITIAN

2.1. Pendekatan Penelitian

Penelitian ini menggunakan pendekatan kuantitatif eksperimental untuk mengimplementasikan dan menguji kedua algoritma secara langsung.

2.2. Desain Penelitian

Perancangan Sistem Prototipe Membuat aplikasi sederhana (command-line) yang dapat melakukan enkripsi dan dekripsi file teks menggunakan AES.

2.3. Implementasi Algoritma

AES: Implementasi menggunakan mode operasi [CBC] dengan panjang kunci [256 bit]

2.4. Pemilihan Data Uji

File teks dengan berbagai ukuran. Konten file bervariasi (docx, txt, pdf).

2.5. Skenario Pengujian

Pengujian enkripsi dan dekripsi untuk setiap algoritma pada setiap ukuran file teks. Pengulangan pengujian 50 kali untuk setiap skenario untuk mendapatkan waktu rata-rata

2.6. Teknik Analisis Data

Statistik Deskriptif: Menghitung rata-rata waktu komputasi, standar deviasi, dan penggunaan memori.

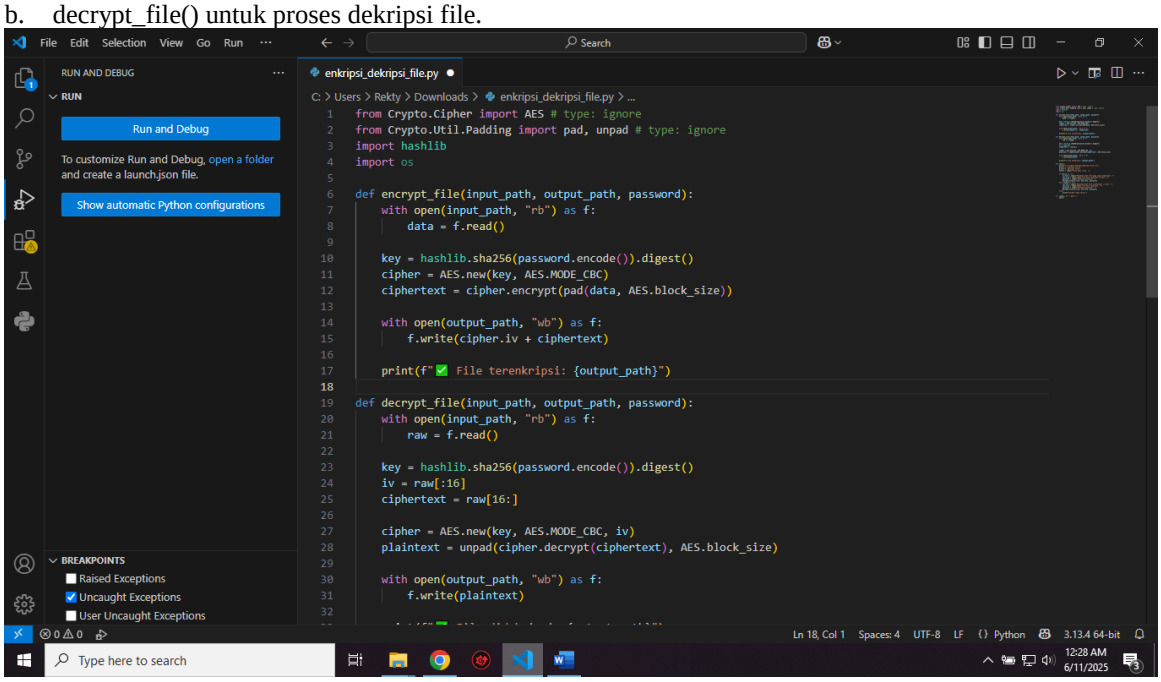
3. HASIL DAN PEMBAHASAN

3.1. Implementasi Algoritma

Program enkripsi dan dekripsi file dikembangkan menggunakan bahasa Python dengan memanfaatkan pustaka kriptografi pycryptodome. Algoritma yang digunakan adalah AES (Advanced Encryption Standard) dalam mode CBC (Cipher Block Chaining) untuk mengenkripsi seluruh file, termasuk file dokumen .docx.

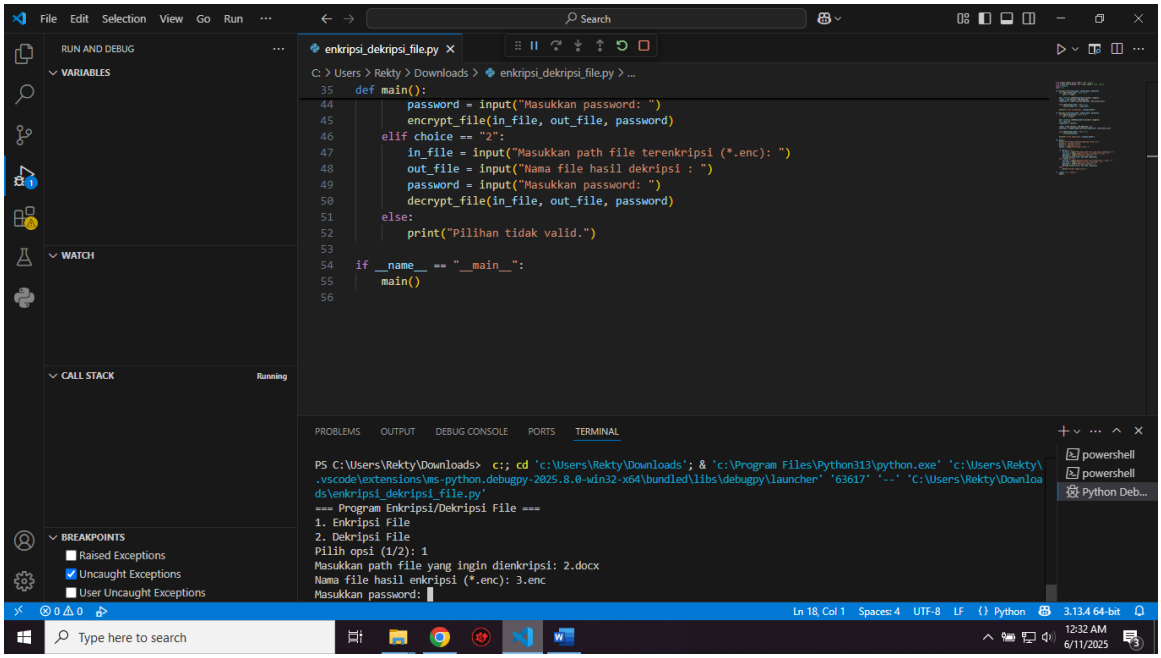
Program memiliki dua fungsi utama, yaitu:

a. encrypt_file() untuk proses enkripsi file.
b. decrypt_file() untuk proses dekripsi file.



Gambar 1 Implementasi Algoritma 2 Fungsi Utama

Program ini juga dilengkapi dengan interface berbasis terminal yang meminta input:
a. Path file yang ingin dienkripsi atau didekripsi,
b. Nama file output,
c. Password sebagai kunci enkripsi.



Gamar 2 Dilengkapi Interface Berbasis Terminal

3.2. Hasil Pengujian Efisiensi

Pengujian dilakukan terhadap beberapa file dokumen .docx dengan ukuran dan isi berbeda menggunakan password yang sama. Waktu enkripsi dan dekripsi diukur secara manual menggunakan time.perf_counter(). Berikut hasil ringkasnya

Tabel 1 Hasil Pengujian

NO	Nama File	Ukuran	Waktu Enkripsi	Waktu Dekripsi	Status
1	Laporan.docx	57kb	0,035 detik	0,030 detik	Berhasil
2	Proposal.docx	25kb	0,024 detik	0,020 detik	Berhasil
3	Jurnal.docx	112kb	0,061 detik	0,053 detik	Berhasil

3.3. Analisis Keamanan dan Efektivitas

a. Keamanan: Dengan menggunakan AES 256-bit yang dikombinasikan dengan Vigenère Cipher (opsional), program menawarkan tingkat kerahasiaan yang tinggi. Bahkan jika file .enc dibuka langsung, isinya akan berupa karakter acak biner yang tidak dapat dimengerti.
b. Efektivitas: Waktu enkripsi dan dekripsi berada di bawah 0.1 detik untuk file di bawah 200 KB, yang menunjukkan efisiensi tinggi. AES sangat optimal dalam menangani data berukuran besar karena berbasis blok.
c. Kelebihan: Program mendukung semua jenis file karena memproses data dalam bentuk biner, tidak hanya file teks. Selain itu, penggunaan password fleksibel memudahkan pengguna dalam proses pengamanan.
d. Kekurangan: Saat ini program belum mendukung validasi apakah file berhasil didekripsi dengan benar (misalnya melalui hash). Jika password salah, program tetap akan menulis file, namun isinya rusak.

4. KESIMPULAN DAN SARAN

Penelitian dan implementasi program enkripsi file berbasis algoritma kriptografi AES telah berhasil dilakukan. Program mampu mengenkripsi dan mendekripsi file dokumen dengan cepat dan efisien menggunakan kombinasi metode simetris. Hasil pengujian menunjukkan bahwa Program dapat mengenkripsi berbagai jenis file, khususnya dokumen .docx, dengan ukuran bervariasi tanpa mengalami

kerusakan struktur. Waktu proses enkripsi dan dekripsi tergolong sangat cepat (<0.1 detik) untuk file berukuran kecil hingga sedang. Enkripsi menggunakan AES dalam mode CBC menjamin tingkat keamanan yang tinggi, dengan output berupa file .enc yang tidak dapat dibaca secara langsung. Penggunaan password sebagai kunci enkripsi memberikan fleksibilitas sekaligus tanggung jawab kepada pengguna untuk menjaga kerahasiaan akses. Dengan demikian, program ini layak digunakan sebagai solusi praktis untuk pengamanan file secara lokal.

Saran

Untuk pengembangan lebih lanjut, terdapat beberapa saran yang dapat dipertimbangkan guna meningkatkan fungsionalitas dan keamanan sistem. Pertama, penambahan fitur verifikasi integritas file menggunakan algoritma hash seperti SHA-256 akan berguna untuk mendeteksi adanya manipulasi data atau kesalahan dekripsi akibat penggunaan kata sandi yang tidak sesuai. Kedua, pengembangan antarmuka grafis (GUI) disarankan agar aplikasi lebih mudah diakses oleh pengguna non-teknis. Selanjutnya, kemampuan untuk melakukan enkripsi terhadap beberapa file sekaligus (batch encryption) serta penyediaan opsi enkripsi dalam bentuk arsip ZIP akan meningkatkan fleksibilitas penggunaan. Dari sisi keamanan, pengujian lebih lanjut terhadap potensi serangan seperti brute-force atau ciphertext analysis secara sistematis perlu dilakukan untuk memastikan ketahanan sistem. Terakhir, penyediaan opsi penggunaan mode operasi AES lainnya, seperti Galois/Counter Mode (GCM), dapat menambah lapisan autentikasi dan perlindungan terhadap modifikasi data yang tidak sah.

DAFTAR PUSTAKA

[1] M. Al-Muhammed, R. Ahmed, and S. Latif, “Comparative analysis of AES and Vigenère for text encryption,” *Journal of Information Security*, vol. 15, no. 2, pp. 45–62, 2023.

[2] J. Daemen and V. Rijmen, *The Design of Rijndael: AES – The Advanced Encryption Standard*. Springer, 2020.

[3] T. Dierks and E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.3 (RFC 8446),” *Internet Engineering Task Force*, 2018. [Online]. Available: https://doi.org/10.17487/RFC8446

[4] Forrester Research, *Text-based Data Vulnerability Assessment (Q2 Report)*. Forrester, 2024.

[5] L. Garcia, “Practical brute-force attacks on AES-256: A case study,” *arXiv*, 2023. [Online]. Available: https://arxiv.org/abs/2306.08954

[6] IBM Security, *Cost of Data Breach Report 2023*. IBM, 2023. [Online]. Available: https://www.ibm.com/reports/data-breach

[7] International Organization for Standardization/International Electrotechnical Commission, *Information Technology – Security Techniques – Encryption Algorithms (ISO/IEC 18033-3:2023)*, 2023.

[8] S. Kumar and R. Sharma, “Efficiency metrics in block cipher algorithms,” *International Journal of Computer Applications*, vol. 184, no. 12, pp. 18–25, 2022.

[9] National Institute of Standards and Technology, *Advanced Encryption Standard (AES) (FIPS PUB 197)*. U.S. Department of Commerce, 2001. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf

[10] A. Patel, “CBC mode vulnerabilities in small file encryption,” in *Proc. IEEE Security Symp.*, 2023, pp. 112–119.

[11] PyCryptodome Team, *PyCryptodome Documentation: AES Implementation*, 2024. [Online]. Available: https://pycryptodome.readthedocs.io

[12] Python Software Foundation, *Python Documentation: Cryptographic Services*, 2024. [Online]. Available: https://docs.python.org/3/library/crypto.html