BETRIK
BESEMAH TEKNOLOGI INFORMASI & KOMPUTER
Editor Office : Pusat Penelitian & Pengabdian Pada Masyarakat (PPPM) ITPA
Phone : 0857-9716-9578
email : betrikitpa@itpa.ac.id
ISSN : 2339 - 1871

Analisis Komparatif Performa Algoritma AES Dan RSA Dalam Keamanan Data Digital Berbasis *Python*

Habib Nurfaizal[1*], Abdul Choliq[2], Fahrul Roji [3], Ayu Permatasari[4]
Fakultas Ilmu Komputer, Sistem Informasi, Universitas Pamulang, Tangerang Selatan,Indonesia[1,2,3,4]
Sur-el :* dosen02807@unpam.ac.id[1], dosen03161@unpam.ac.id[2], dosen03151@unpam.ac.id[3], ayupermatasari871@gmail.com[4]

Penulis Korespondensi: Habib Nurfaizal, dosen02807@unpam.ac.id

Abstrak*:* Keamanan data merupakan aspek krusial dalam pertukaran informasi digital. Pengembang sistem sering dihadapkan pada pilihan antara algoritma simetris Advanced Encryption Standard (AES) yang cepat namun memiliki kendala dalam distribusi kunci, dan algoritma asimetris Rivest Shamir Adleman (RSA) yang lebih aman dalam distribusi kunci tetapi memiliki beban komputasi lebih tinggi. Selain itu, kajian mengenai efisiensi performa kedua algoritma pada variasi ukuran dan jenis file masih terbatas. Penelitian ini bertujuan untuk menganalisis perbandingan performa AES dan RSA berdasarkan waktu enkripsi dan dekripsi serta efisiensi komputasi. Metode yang digunakan adalah eksperimen melalui simulasi menggunakan bahasa pemrograman Python dengan antarmuka grafis (GUI). Pengujian dilakukan pada berbagai jenis dan ukuran file, yaitu .txt (1 KB, 2 KB), .doc (48 KB), .pdf (97 KB), .png (122 KB), dan .jpg (359 KB), dengan parameter pengukuran meliputi waktu enkripsi, waktu dekripsi, penggunaan memori, serta kompleksitas komputasi yang dihitung berdasarkan waktu eksekusi menggunakan fungsi time pada Python. Hasil eksperimen menunjukkan bahwa AES cenderung lebih efisien dan cepat pada file berukuran menengah hingga besar, sedangkan RSA memiliki performa lebih stabil pada ukuran file kecil namun kurang efisien pada ukuran besar karena kompleksitas komputasi. Dengan demikian, AES lebih direkomendasikan untuk enkripsi data utama, sementara RSA efektif digunakan dalam mekanisme distribusi kunci secara aman.

Kata kunci *:* AES, Dekripsi, Enkripsi, Kriptografi, RSA

Abstract. *Data security is a crucial aspect of digital information exchange. System developers are often faced with the choice between the symmetric algorithm Advanced Encryption Standard (AES), which is fast but has limitations in key distribution, and the asymmetric algorithm Rivest–Shamir–Adleman (RSA), which provides more secure key distribution but incurs higher computational overhead. In addition, studies examining the performance efficiency of both algorithms across varying file sizes and types remain limited. This study aims to analyze the comparative performance of AES and RSA based on encryption and decryption time as well as computational efficiency. The method employed is an experimental approach through simulation using the Python programming language with a graphical user interface (GUI). Testing was conducted on various file types and sizes, namely .txt (1 KB, 2 KB), .doc (48 KB), .pdf (97 KB), .png (122 KB), and .jpg (359 KB), with measurement parameters including encryption time, decryption time, memory usage, and computational complexity, which were evaluated based on execution time using Python's time function. The experimental results indicate that AES tends to be more efficient and faster for medium*

Received: 01-04-2026 | Accepted: 12-04-2026 | Published Online: 30-04-2026

to large-sized files, whereas RSA demonstrates more stable performance for small file sizes but becomes less efficient for larger files due to its computational complexity. Therefore, AES is more recommended for primary data encryption, while RSA is effectively utilized in secure key distribution mechanisms.

Keywords: *AES, Cryptography, Decryption, Encryption, RSA*

1. PENDAHULUAN

Dalam era digital, hampir seluruh aktivitas manusia bergantung pada sistem informasi berbasis teknologi. Data digital menjadi aset yang sangat bernilai dalam berbagai bidang, seperti bisnis, pemerintahan, pendidikan, dan komunikasi pribadi. Seiring dengan meningkatnya volume dan sensitivitas data, risiko terhadap keamanan informasi juga semakin tinggi. Ancaman seperti pencurian data, manipulasi, dan akses tidak sah menuntut adanya sistem keamanan yang dirancang secara komprehensif untuk meminimalkan risiko kerusakan maupun kebocoran data [1].

Keamanan informasi pada dasarnya mengacu pada prinsip CIA Triad, yaitu kerahasiaan (confidentiality), integritas (integrity), dan ketersediaan (availability), yang menjadi landasan dalam menjaga keandalan sistem informasi. Selain itu, otentikasi juga berperan sebagai mekanisme awal dalam memverifikasi identitas pengguna guna mencegah akses tidak sah terhadap sistem [2].

Salah satu pendekatan utama dalam menjaga keamanan data adalah kriptografi, yaitu teknik untuk melindungi informasi dengan mengubahnya ke dalam bentuk yang tidak dapat dipahami tanpa kunci tertentu. Kriptografi modern memiliki tujuan utama meliputi kerahasiaan, integritas, autentikasi, dan non-repudiation, serta secara umum dibagi menjadi algoritma simetris dan asimetris [3]. Algoritma simetris menggunakan satu kunci yang sama untuk proses enkripsi dan dekripsi sehingga memiliki keunggulan dalam kecepatan, namun bergantung pada keamanan distribusi kunci [4]. Salah satu algoritma yang dapat digunakan adalah Advanced Encryption Standard (AES), yang dikenal memiliki efisiensi tinggi dan tingkat keamanan yang kuat dalam pengamanan data digital [5][6].

Di sisi lain, algoritma asimetris menggunakan pasangan kunci publik dan privat, sehingga lebih unggul dalam mekanisme distribusi kunci dan mendukung autentikasi melalui tanda tangan digital [7]. Salah satu algoritma yang paling banyak digunakan adalah Rivest Shamir Adleman (RSA), yang keamanannya didasarkan pada kompleksitas pemfaktoran bilangan besar.

Meskipun demikian, kedua algoritma memiliki keterbatasan. AES cenderung mengalami kendala pada aspek distribusi kunci, sedangkan RSA memiliki kompleksitas komputasi yang tinggi sehingga kurang efisien untuk data berukuran besar. Selain itu, kajian empiris yang membandingkan performa kedua algoritma pada variasi ukuran dan jenis file masih relatif terbatas, khususnya dalam konteks implementasi langsung.

Penelitian-penelitian sebelumnya telah mengkaji berbagai aspek terkait performa dan implementasi algoritma kriptografi. Studi oleh Fajrin (2024) membahas keamanan transaksi berbasis web dan menyatakan bahwa performa AES dan RSA relatif sebanding dengan penyesuaian panjang kunci, namun belum melibatkan pengujian berbasis GUI maupun variasi ukuran file [8]. Sementara itu, Nawawi (2023) hanya

Received: 01-04-2026 | Accepted: 12-04-2026 | Published Online: 30-04-2026

berfokus pada waktu enkripsi tanpa integrasi antarmuka dan visualisasi, serta menekankan keunggulan kecepatan AES [9]. Penelitian Risna (2022) mengimplementasikan kriptografi berbasis GUI dengan fokus pada kemudahan penggunaan, tetapi belum disertai analisis performa yang mendalam [10]. Di sisi lain, Prasetyo (2021) mengkaji optimasi RSA menggunakan komputasi terdistribusi dengan penekanan pada efisiensi teknis tanpa mempertimbangkan aspek antarmuka pengguna [11].

Dari beberapa penelitian terdahulu yang sudah dijelaskan, penelitian ini menawarkan kebaruan melalui analisis performa AES dan RSA yang terintegrasi dalam GUI berbasis Python, dengan pengujian pada berbagai ukuran dan jenis file serta penyajian hasil yang lebih interaktif dan aplikatif.

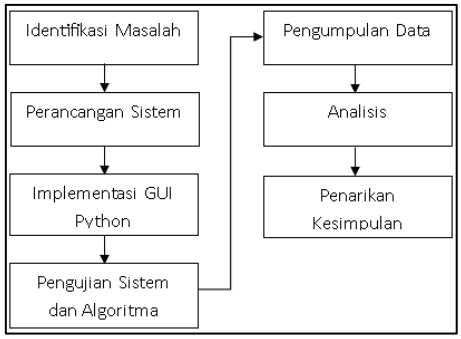
Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk melakukan analisis komparatif performa algoritma AES dan RSA melalui metode eksperimental. Pengujian dilakukan menggunakan aplikasi berbasis Python dengan antarmuka grafis (GUI) terhadap berbagai jenis dan ukuran file, untuk mengukur waktu enkripsi, dekripsi, serta efisiensi komputasi.

Melalui penelitian ini, diharapkan diperoleh pemahaman yang lebih komprehensif mengenai karakteristik performa kedua algoritma dalam penggunaan nyata. Hasil penelitian ini diharapkan dapat menjadi referensi dalam pemilihan algoritma kriptografi yang tepat sesuai kebutuhan sistem, baik dari segi efisiensi maupun keamanan.

2. METODOLOGI PENELITIAN

Penelitian ini menggunakan metode eksperimental dengan pendekatan kuantitatif. Metode eksperimental dipilih karena memungkinkan peneliti untuk menguji dan membandingkan secara langsung performa dua algoritma kriptografi, yaitu *Advanced Encryption Standard* (AES) dan *Rivest Shamir Adleman* (RSA), dalam kondisi yang terkontrol. Pendekatan kuantitatif digunakan untuk menghasilkan data numerik yang objektif, sehingga analisis dapat dilakukan secara terukur dan sistematis.

Eksperimen dilakukan dengan cara mengimplementasikan kedua algoritma tersebut pada lingkungan komputasi yang sama, kemudian mengukur kinerja masing-masing berdasarkan parameter yang telah ditentukan. Parameter utama yang digunakan dalam penelitian ini meliputi waktu proses (processing time) dan efisiensi komputasi. Waktu proses diukur untuk mengetahui kecepatan enkripsi dan dekripsi data, sedangkan efisiensi komputasi dianalisis berdasarkan penggunaan sumber daya sistem selama proses berlangsung.



Gambar 1. Alur Penelitian

Received: 01-04-2026 | Accepted: 12-04-2026 | Published Online: 30-04-2026

Pada gambar 1 di atas dapat diketahui proses penelitian sebagai berikut :

1. Identifikasi Masalah

   Tahap awal dilakukan dengan mengkaji permasalahan terkait keamanan data digital, khususnya dalam pemilihan algoritma kriptografi yang optimal antara AES dan RSA. Fokus utama adalah perbedaan performa kedua algoritma dalam proses enkripsi dan dekripsi pada berbagai jenis serta ukuran file.

2. Perancangan Sistem

   Pada tahap ini dilakukan perancangan sistem aplikasi yang akan digunakan sebagai media eksperimen, meliputi desain alur proses enkripsi–dekripsi, struktur input/output, serta rancangan antarmuka pengguna (GUI) untuk memudahkan interaksi dan pengujian.

3. Implementasi GUI Python

   Sistem yang telah dirancang kemudian diimplementasikan menggunakan bahasa pemrograman Python dengan antarmuka grafis (GUI). Pada tahap ini juga dilakukan integrasi algoritma AES dan RSA ke dalam aplikasi sehingga dapat digunakan untuk proses enkripsi dan dekripsi.

4. Pengujian Sistem dan Algoritma

   Pengujian dilakukan dengan menjalankan sistem secara fungsional dan menjalankan kedua algoritma pada berbagai jenis file (.txt, .doc, .pdf, .png, .jpg) dengan ukuran yang berbeda. Setiap proses enkripsi dan dekripsi diuji secara sistematis untuk memperoleh data performa.

5. Pengumpulan Data

   Data yang dikumpulkan berupa waktu enkripsi dan dekripsi dari masing-masing algoritma pada setiap jenis dan ukuran file. Data diperoleh secara langsung dari hasil eksekusi sistem.

6. Analisis Data

   Data yang telah dikumpulkan dianalisis untuk membandingkan performa AES dan RSA. Analisis difokuskan pada kecepatan proses, efisiensi komputasi, serta kestabilan performa pada berbagai kondisi pengujian.

7. Penarikan Kesimpulan

   Tahap akhir adalah menyimpulkan hasil penelitian berdasarkan analisis yang telah dilakukan, serta memberikan rekomendasi penggunaan algoritma yang lebih sesuai dalam mendukung keamanan data digital.

2.1 Implementasi

Bahasa pemrograman yang digunakan dalam penelitian ini adalah Python dengan dukungan berbagai library kriptografi seperti *PyCryptodome* dan *Cryptography*, serta *library* GUI seperti Tkinter untuk membangun antarmuka interaktif. Pemilihan Python didasarkan pada kemampuannya dalam mendukung implementasi algoritma AES dan RSA secara efisien serta memudahkan proses eksperimen dan visualisasi. Adapun struktur program aplikasi yang digunakan dalam penelitian ini ditunjukkan pada Gambar 2.

Received: 01-04-2026 | Accepted: 12-04-2026 | Published Online: 30-04-2026

```
crypto_dashboard/
│
├── main.py
├── login.py
├── dashboard.py
├── crypto_aes.py
├── crypto_rsa.py
│
├── crypto_dashboard/ (sub-package)
│   ├── __init__.py
│   ├── performance_graph.png
│   ├── performance_report.csv
│   ├── secret.key
│   └── users.db      # database pengguna
│
└── files/
    ├── encrypted/
    └── decrypted/
```

Gambar 2. *Project Application Structure*

2.2 Pengujian

Pengujian sistem dilakukan menggunakan metode *black box testing* yang berfokus pada pengujian fungsionalitas aplikasi berdasarkan kesesuaian antara *input* dan *output* tanpa memperhatikan implementasi internal sistem, serta dilakukan pengukuran kinerja algoritma AES dan RSA melalui waktu eksekusi proses enkripsi dan dekripsi untuk menganalisis efisiensi dan performa komputasi kedua algoritma secara kuantitatif.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Implementasi Halaman Login

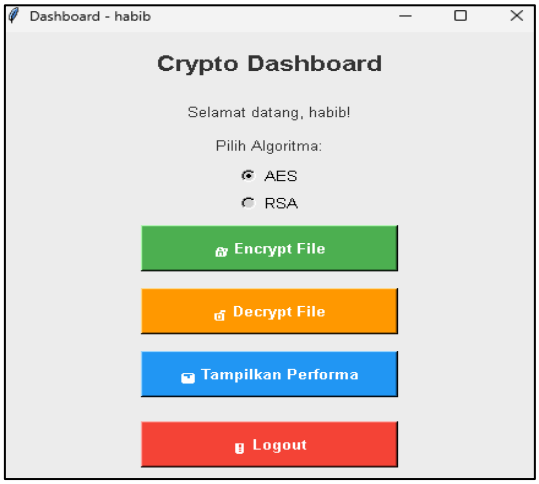
Halaman *login* merupakan antarmuka awal pada sistem kriptografi yang berfungsi sebagai tahap autentikasi pengguna sebelum mengakses fitur utama sistem. Halaman ini berjudul "Login Sistem Kriptografi" dan dirancang dengan tampilan sederhana serta fungsional. Terdapat dua input utama, yaitu *username* dan *password*, yang disusun secara terpusat untuk memudahkan pengguna dalam melakukan proses autentikasi. Selain itu, tersedia dua tombol aksi, yaitu Login untuk verifikasi akun dan Register untuk pembuatan akun baru, yang menunjukkan adanya pemisahan fungsi akses pengguna. Perbedaan warna tombol (hijau untuk *Login* dan biru untuk *Register*) digunakan untuk memperjelas fungsi masing-masing tombol sehingga mengurangi potensi kesalahan pengguna. Secara keseluruhan, desain antarmuka yang minimalis dan terstruktur ini bertujuan untuk meningkatkan kemudahan penggunaan serta efisiensi interaksi pengguna dengan sistem.

Gambar 3. Halaman Login

Received: 01-04-2026 | Accepted: 12-04-2026 | Published Online: 30-04-2026

3.2 Hasil Implementasi Menu Utama

Halaman menu utama merupakan dashboard utama aplikasi yang muncul setelah proses autentikasi berhasil dilakukan. Halaman ini berfungsi sebagai pusat kendali sistem yang menyediakan akses ke seluruh fitur utama kriptografi dengan tata letak yang terstruktur dan mudah dipahami oleh pengguna. Pada halaman ini, pengguna dapat memilih algoritma AES atau RSA melalui radio button yang bersifat eksklusif, sehingga hanya satu algoritma dapat digunakan dalam satu proses. Selain itu, tersedia empat tombol aksi utama, yaitu Encrypt File, Decrypt File, Tampilkan Performa, dan Logout, yang dibedakan berdasarkan warna untuk memudahkan identifikasi fungsi serta meningkatkan kemudahan penggunaan. Dashboard juga menampilkan pesan personalisasi sesuai dengan pengguna yang sedang aktif, sehingga mendukung pengalaman pengguna yang lebih interaktif sekaligus memperkuat aspek keamanan berbasis akses individual.

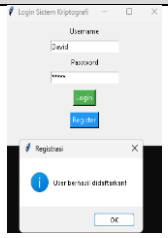
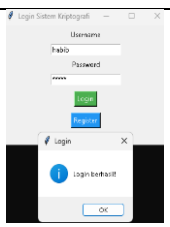


Gambar 4. Halaman Menu Utama

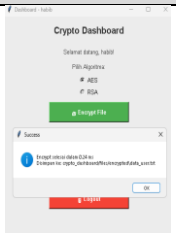
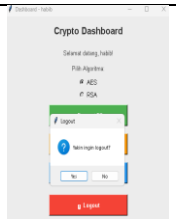
3.3 Hasil Pengujian Sistem

Pengujian *black box* secara fungsional merupakan metode pengujian perangkat lunak yang berfokus pada pemeriksaan fungsi dan keluaran sistem tanpa memperhatikan struktur internal atau implementasi kode program. Berikut merupakan hasil dari pengujian *black box* :

Tabel 1. Hasil Pengujian Black Box

No	Pengujian Yang Diharapkan	Hasil	Gambar
1	Melakukan registrasi dan sistem melakukan penyimpanan data pengguna untuk dapat login	[√] Berhasil [ ] Gagal	
2	Melakukan login, kemudian memberikan keterangan sebelum masuk ke sistem	[√] Berhasil [ ] Gagal	

Received: 01-04-2026 | Accepted: 12-04-2026 | Published Online: 30-04-2026

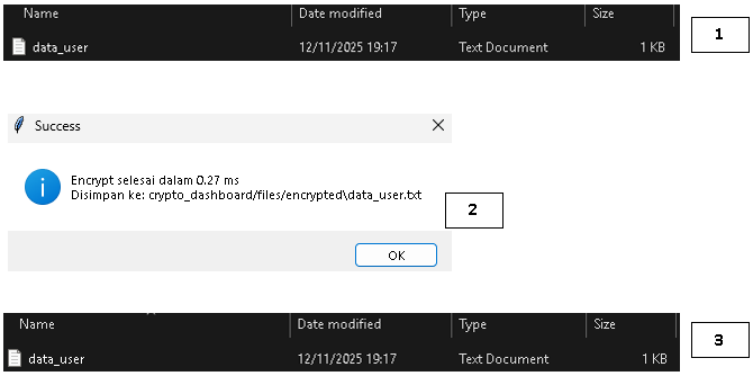
No	Pengujian Yang Diharapkan	Hasil	Gambar
3	Melakukan enkripsi file AES/ RSA, kemudian masuk ke direktori, pilih file dan memproses mengenkripsi file	[√] Berhasil [ ] Gagal	
4	Melakukan dekripsi file AES/ RSA, sistem masuk ke direktori, pilih file yang sudah terenkripsi dan memproses dekripsi file	[√] Berhasil [ ] Gagal	
5	Memilih menu tampil performa, kemudian sistem akan menampilkan performa proses encrypt dan decrypt file	[√] Berhasil [ ] Gagal	
6	Melakukan pengecekan hasil dari performa pada file excel, sistem menyimpan hasil performa tersebut dengan format berekstensi excel	[√] Berhasil [ ] Gagal	
7	Memilih logout sistem, kemudian aplikasi memberikan popup pesan dialog	[√] Berhasil [ ] Gagal	

3.4 Hasil Pengujian Enkripsi File

Tahapan enkripsi file menggunakan algoritma AES atau RSA dimulai dari pengguna memilih algoritma RSA atau dan menekan menu "*Encrypt File*", kemudian sistem mengarahkan ke direktori penyimpanan untuk memilih file. Setelah file dipilih, sistem akan melakukan proses enkripsi dan menyimpan hasilnya ke dalam direktori "*encrypted*".

Bukti tampilan hasil eksekusi program yang menunjukkan data waktu enkripsi dapat dilihat pada Gambar 5.

Received: 01-04-2026 | Accepted: 12-04-2026 | Published Online: 30-04-2026



Gambar 5. Hasil Enkripsi File

Berdasarkan keluaran program tersebut, hasil pengujian enkripsi selanjutnya disajikan secara lebih sistematis dalam Tabel 2 berikut :

Tabel 2. Hasil Pengujian Enkripsi File

No	Jenis File	Ukuran File	AES	RSA
1	.txt	1 kb	0.16 ms	0.27 ms
2	.txt	2 kb	0.18 ms	Gagal
3	.doc	48 kb	1.47 ms	Gagal
4	.pdf	97 kb	1.18 ms	Gagal
5	.png	122 kb	3.28 ms	Gagal
6	.jpg	359 kb	9.66 ms	Gagal

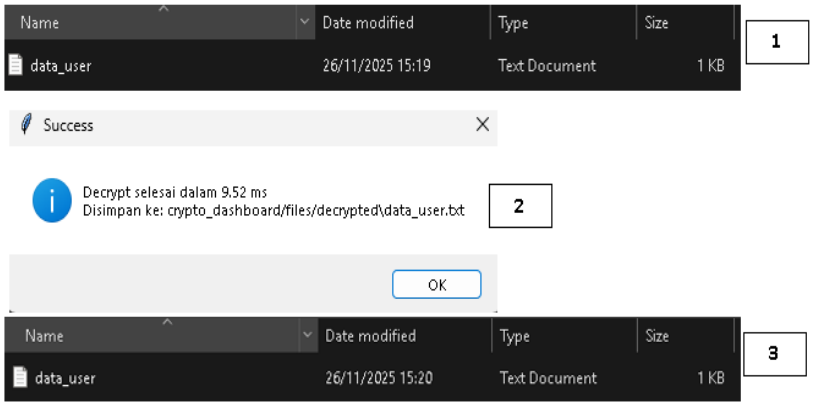
Dari tabel 2, dapat dilihat bahwa algoritma AES berhasil melakukan proses enkripsi pada seluruh jenis file dengan waktu eksekusi yang relatif stabil, meskipun terjadi peningkatan waktu seiring dengan bertambahnya ukuran file. Hal ini menunjukkan bahwa AES mampu menangani berbagai ukuran file secara efisien karena menggunakan metode *block cipher* yang dirancang untuk memproses data dalam jumlah besar. Sebaliknya, algoritma RSA menunjukkan banyak kegagalan pada proses enkripsi, terutama pada file dengan ukuran lebih besar dari 2 KB. Kegagalan ini disebabkan oleh keterbatasan RSA dalam menangani data berukuran besar secara langsung. Secara konseptual, RSA tidak dirancang untuk mengenkripsi file secara utuh, melainkan lebih optimal digunakan untuk mengenkripsi data berukuran kecil seperti kunci sesi (*session key*). Hal ini berkaitan dengan sifat RSA yang memiliki batasan panjang input berdasarkan ukuran kunci serta kompleksitas komputasi yang tinggi.

3.5 Hasil Pengujian Dekripsi File

Tahapan dekripsi file menggunakan algoritma AES atau RSA dimulai dengan pengguna memilih algoritma AES atau RSA dan menekan menu "*Decrypt File*", kemudian sistem mengarahkan ke direktori penyimpanan untuk memilih file yang akan didekripsi. Setelah file dipilih, sistem akan menjalankan proses dekripsi dan menyimpan hasilnya ke dalam direktori "*decrypted*".

Bukti tampilan hasil dekripsi file yang menunjukkan data waktu dekripsi dapat dilihat pada Gambar 6.

Received: 01-04-2026 | Accepted: 12-04-2026 | Published Online: 30-04-2026



Gambar 6. Hasil Dekripsi File

Berdasarkan *output* program gambar 6, hasil pengujian dekripsi berikutnya disajikan secara lebih sistematis dalam Tabel 3 berikut :

Tabel 3. Hasil Pengujian Dekripsi File

No	Jenis File	Ukuran File	AES	RSA
1	.txt	1 kb	0.16 ms	9.52 ms
2	.txt	3 kb	0.23 ms	Gagal
3	.doc	63 kb	0.52 ms	Gagal
4	.pdf	129 kb	1.0 ms	Gagal
5	.png	162 kb	1.32 ms	Gagal
6	.jpg	479 kb	5.25 ms	Gagal

Dari tabel 3, algoritma AES berhasil melakukan proses dekripsi pada seluruh jenis file dengan waktu eksekusi yang relatif cepat dan konsisten. Hal ini menunjukkan bahwa AES memiliki kinerja yang stabil dalam mengembalikan *ciphertext* menjadi *plaintext*, bahkan pada file dengan ukuran yang lebih besar. Sebaliknya, algoritma RSA mengalami kegagalan pada hampir seluruh proses dekripsi, kecuali pada file berukuran sangat kecil (1 KB). Kegagalan ini berkaitan langsung dengan hasil proses enkripsi sebelumnya yang juga mengalami kegagalan. Dalam sistem kriptografi, proses dekripsi hanya dapat dilakukan jika *ciphertext* berhasil dihasilkan pada tahap enkripsi. Oleh karena itu, ketika proses enkripsi RSA gagal, maka tidak tersedia data terenkripsi yang *valid* untuk didekripsi, sehingga proses dekripsi juga tidak dapat dilakukan.

3.6 Pembahasan

Hasil pengujian menunjukkan bahwa algoritma AES memiliki kinerja yang lebih unggul dibandingkan RSA dalam proses enkripsi dan dekripsi file. Keunggulan ini disebabkan oleh karakteristik AES sebagai algoritma kriptografi simetris yang dirancang untuk memproses data dalam jumlah besar secara efisien menggunakan operasi block cipher. Waktu eksekusi yang relatif stabil pada berbagai ukuran file menunjukkan bahwa kompleksitas komputasi AES bersifat lebih terkontrol dan skalabel terhadap peningkatan ukuran data.

Sebaliknya, keterbatasan kinerja RSA terlihat dari kegagalannya dalam memproses file berukuran menengah hingga besar. Hal ini disebabkan oleh sifat RSA sebagai algoritma kriptografi asimetris yang tidak dirancang untuk mengenkripsi data dalam jumlah besar secara langsung. RSA memiliki batasan

Received: 01-04-2026 | Accepted: 12-04-2026 | Published Online: 30-04-2026

panjang input yang bergantung pada ukuran kunci, serta melibatkan operasi matematika kompleks seperti perpangkatan modular dengan bilangan besar, yang menyebabkan beban komputasi meningkat secara signifikan. Kegagalan pada proses dekripsi RSA juga merupakan konsekuensi langsung dari kegagalan pada tahap enkripsi, karena tidak tersedianya *ciphertext* yang valid untuk diproses lebih lanjut. Kondisi ini menunjukkan bahwa penggunaan RSA secara langsung untuk enkripsi file tidak efisien dan kurang sesuai dalam konteks pengolahan data berukuran besar.

Berdasarkan hasil tersebut, pendekatan yang lebih tepat adalah penggunaan skema kriptografi hybrid, di mana RSA digunakan untuk mengamankan distribusi kunci, sedangkan AES digunakan untuk proses enkripsi data utama. Pendekatan ini mampu menggabungkan keunggulan masing-masing algoritma, yaitu efisiensi AES dan keamanan distribusi kunci pada RSA.

4. KESIMPULAN

Berdasarkan hasil penelitian dan pengujian yang telah dilakukan, maka dapat disimpulkan bahwa:

1. Algoritma AES menunjukkan kinerja yang sangat baik dalam proses enkripsi dan dekripsi pada berbagai ukuran file, dengan tingkat keberhasilan mencapai 100% serta waktu eksekusi yang relatif cepat dan stabil, meskipun meningkat seiring bertambahnya ukuran data.
2. Algoritma RSA hanya menunjukkan kinerja optimal pada file berukuran kecil, namun mengalami keterbatasan pada file berukuran lebih besar akibat batasan panjang *input* dan tingginya kompleksitas komputasi, sehingga kurang efisien untuk enkripsi data secara langsung.
3. Kegagalan proses enkripsi dan dekripsi pada RSA dalam pengujian ini menunjukkan bahwa algoritma tersebut tidak dirancang untuk menangani data berukuran besar, melainkan lebih sesuai digunakan dalam proses distribusi kunci dan mekanisme autentikasi.
4. Penelitian ini memberikan kontribusi dalam bentuk analisis komparatif performa algoritma AES dan RSA yang terintegrasi dalam aplikasi berbasis GUI, dengan pengujian pada berbagai jenis dan ukuran file, sehingga menghasilkan pendekatan yang lebih aplikatif dan interaktif dalam evaluasi kinerja algoritma kriptografi.

DAFTAR RUJUKAN

[1] D. S. Wiratomo, B. Hananto, and I. W. widi Pradnyana, "Implementasi Keamanan File Pada Aplikasi Penyimpanan Berbasis Cloud Computing Dengan Algoritma Advanced Encryption Standard (AES) dan Kompresi Lempel Ziv Welch (LZW)," *SENAMIKA*, vol. 3, no. 2, pp. 521–530, 2022.

[2] I. G. Arianto, W. Witanti, and H. Ashaury, "Sistem Keamanan Otentikasi Pengguna pada Modul Single Sign On Menggunakan OAuth 2 . 0 dan One Time Password," *J. Ilmu Komput. dan Teknol.*, vol. 6, no. 1, pp. 25–31, 2025.

[3] M. H. Prasetyo, H. Mulyo, and T. Tamrin, "Analisis Algoritma RSA Dan AES Dalam Enkripsi Gambar Digital," *J. Tek. Inform.*, vol. 4, no. 1, pp. 197–207, 2025.

[4] A. H. Nasrullah, A. S. Amaliah, and G. R. Jannah, "Hybrid Kriptografi Menggunakan RSA dan AES untuk Keamanan Pengiriman File Digital," *J. Comput. Informatics, Vocat. Educ.*, vol. 2, no. 2, pp. 11–17, 2025.

[5] R. A. Juandana, A. P. Harahap, Z. P. B. Hutabarat, A. D. M. Matondang, U. Maulida, and M. A. Nasywa, "Implementasi Algoritma AES Sebagai Kriptografi Simetris Untuk Proses Enkripsi Dan

Received: 01-04-2026 | Accepted: 12-04-2026 | Published Online: 30-04-2026

Dekripsi Data," *J. Ilmu Komput.*, vol. 1, no. 2, pp. 45–48, 2025.
[6] F. Yusri *et al.*, "Implementasi Algoritma Advanced Encryption Standard (AES) Secara Manual Menggunakan Python," *J. Ilmu Komput.*, vol. 1, no. 1, pp. 12–16, 2025.
[7] A. F. Pranata, A. F. Saragih, R. Fikri, K. Rahmadani, and N. P. Aulia, "Analisis Perbandingan Enkripsi dan Dekripsi Menggunakan Algoritma Simetris AES dan Hybrid (AES + RSA) Pada File Video," *J. Ilmu Komput.*, vol. 2, no. 1, pp. 102–107, 2026.
[8] A. M. Fajrin, C. Kelvin, B. Owen, and B. Aji, "Perbandingan Performa dari Algoritma AES dan RSA dalam Keamanan Transaksi," vol. 5, no. 2, pp. 696–705, 2024.
[9] T. Nawawi, D. B., Huda, M. M. ., & Prabowo, "G-Tech : Jurnal Teknologi Terapan," *J. Teknol. Terap.*, vol. 8, no. 1, pp. 186–195, 2024.
[10] Risna, Y. Amaliah, and S. Yunita, "Implementasi Kriptografi Pada Pengamanan Data Pembayaran Piutang Pelanggan Menggunakan Vigenere Cipher," vol. 26, no. 2, pp. 525–534, 2022, doi: 10.46984/sebatik.v26i2.2061.
[11] A. Prasetyo, S. N. Arief, and R. Wakhidah, "Optimasi Pemrosesan Enkripsi Dan Dekripsi RSA Pada Single Board Computer (SBC) Dengan Pembagian Beban Komputasi Dalam Sistem Terdistribusi," *J. Inform. Polinema*, vol. 7, no. 4, pp. 7–12, 2021.

Received: 01-04-2026 | Accepted: 12-04-2026 | Published Online: 30-04-2026