Jutisi: Jurnal Ilmiah Teknik Informatika dan Sistem Informasi
https://ojs.stmik-banjarbaru.ac.id/index.php/jutisi/index
Jl. Ahmad Yani, K.M. 33,5 - Kampus STMIK Banjarbaru
Loktabat – Banjarbaru (Tlp. 0511 4782881), e-mail: puslit.stmikbjb@gmail.com
e-ISSN: 2685-0893

Implementasi Enkripsi Hybrid AES-RSA pada Layanan *Cloud Storage* AWS S3

DOI: http://dx.doi.org/10.35889/jutisi.v14i2.2724

Mohamad Ricky Firdaus[1*], Shofa Shofiah Hilabi[2], Elfina Novalia[3], April Lia Hananto[4]
Sistem Informasi, Universitas Buana Perjuangan Karawang, Karawang, Indonesia
*e-mail *Corresponding Author*: si21.mohamadfirdaus@mhs.ubpkarawang.ac.id

Abstract

Data security has become a critical aspect in the digital era, especially with the increasing use of cloud storage services for data storage and management. This study examines the implementation of hybrid encryption combining the Advanced Encryption Standard (AES) and Rivest-Shamir-Adleman (RSA) algorithms on the Amazon Web Services Simple Storage Service (AWS S3) cloud platform. The hybrid method leverages the speed of AES for encrypting large data volumes and the security of RSA for key management. The encryption and decryption processes are performed entirely on the client side before uploading data to AWS S3, ensuring that the stored data remains securely encrypted. The developed application provides features for encryption, decryption, and uploading encrypted data with a user-friendly interface. Testing results demonstrate that this method achieves an optimal balance between performance and data security. This research contributes to the development of efficient and reliable cloud-based data security solutions.

Keywords: *Advanced Encryption Standard; Rivest–Shamir–Adleman; Amazon Web Services Simple Storage Service; Data Security; Cloud Storage*

Abstrak

Keamanan data menjadi aspek penting dalam era digital, terutama dengan meningkatnya penggunaan layanan cloud storage untuk penyimpanan dan pengelolaan data. Penelitian ini mengkaji implementasi enkripsi hybrid yang menggabungkan algoritma *Advanced Encryption Standard* (AES) dan *Rivest-Shamir-Adleman* (RSA) pada layanan cloud storage *Amazon Web Services Simple Storage Service* (AWS S3). Metode hybrid ini memanfaatkan kecepatan AES dalam mengenkripsi data berukuran besar dan keamanan RSA dalam pengelolaan kunci enkripsi. Proses enkripsi dan dekripsi dilakukan sepenuhnya di sisi klien sebelum data diunggah ke AWS S3, sehingga memastikan data tersimpan dalam bentuk terenkripsi yang aman. Aplikasi yang dikembangkan menyediakan fitur enkripsi, dekripsi, dan upload data terenkripsi dengan antarmuka yang mudah digunakan. Hasil pengujian menunjukkan bahwa metode ini memberikan keseimbangan optimal antara performa dan keamanan data. Penelitian ini memberikan kontribusi dalam pengembangan solusi keamanan data berbasis cloud yang efisien dan dapat diandalkan.

Kata kunci: *Advanced Encryption Standard; Rivest–Shamir–Adleman; Amazon Web Services Simple Storage Service; Keamanan Data; Cloud Storage*

1. Pendahuluan

Di era digital saat ini, penyimpanan data secara *online* melalui layanan *cloud* menjadi kebutuhan penting bagi banyak individu dan organisasi. *Amazon Web Services Simple Storage Service* (AWS S3) adalah salah satu layanan *cloud storage* yang banyak digunakan karena kemudahan akses dan kapasitas penyimpanan yang besar. Namun, penggunaan *cloud storage* juga menimbulkan kekhawatiran terkait keamanan data, terutama risiko pencurian, manipulasi, atau akses tanpa izin yang dapat merugikan pemilik data. Hal ini menegaskan bahwa meskipun *cloud* menawarkan kemudahan, ancaman terhadap keamanan data tetap menjadi perhatian utama yang harus diwaspadai oleh pengguna dan penyedia layanan [1].

Untuk mengatasi masalah tersebut, enkripsi data menjadi solusi utama yang dapat menjaga kerahasiaan dan integritas informasi. Metode enkripsi *hybrid* yang menggabungkan algoritma *Advanced Encryption Standard (AES)* dan *Rivest-Shamir-Adleman (RSA)* menawarkan keunggulan dengan memanfaatkan kecepatan *AES* dalam mengenkripsi data besar dan keamanan *RSA* dalam pengelolaan kunci enkripsi. Pendekatan ini telah banyak diteliti, namun sebagian besar implementasi masih berfokus pada enkripsi di sisi server atau *cloud*, sehingga data masih berisiko saat dikirim dari pengguna ke *cloud* [2].

Enkripsi *hybrid*, seperti pendekatan gabungan AES-RSA, umumnya menjadi pilihan utama untuk berbagi data yang aman dalam lingkungan cloud, karena memungkinkan anda memanfaatkan keunggulan masing-masing metode dalam hal kinerja dan keamanan secara bersamaan. Penelitan Aditya pada 2021 memberikan hasil yang cukup baik dimana kedua algoritma AES dan RSA ini hanya membutuhkan waktu 495,56 ms untuk enkripsi dan 7,428 ms untuk dekripsi [3], algoritma ini sudah dipakai didalam aplikasi komersil seperti keuangan dan telekomunikasi. Terlebih lagi tidak memakan sumber daya yang besar terhadap sistem [4]

Penelitian ini bertujuan mengembangkan aplikasi desktop yang menerapkan enkripsi *hybrid AES-RSA* secara langsung di perangkat pengguna sebelum data dikirim ke *AWS S3*. Dengan cara ini, data yang tersimpan di *cloud* sudah dalam kondisi terenkripsi, sehingga meningkatkan tingkat keamanan secara signifikan. Selain itu, penelitian ini juga mengevaluasi performa dan efektivitas metode enkripsi *hybrid* dalam konteks penyimpanan *cloud*, serta memberikan solusi praktis yang dapat digunakan oleh pengguna umum untuk melindungi data mereka. Kontribusi utama dari penelitian ini adalah pengembangan aplikasi yang mengintegrasikan enkripsi *hybrid* dengan layanan *cloud AWS S3*, serta analisis mendalam mengenai kelebihan dan keterbatasan pendekatan ini dibandingkan dengan metode enkripsi konvensional. Dengan demikian, penelitian ini tidak hanya memberikan solusi teknis, tetapi juga memperkuat pemahaman tentang pentingnya keamanan data di era *cloud computing*. Selain itu, sebagaimana teknologi informasi dapat digunakan untuk melestarikan budaya dan meningkatkan minat generasi muda terhadap nilai-nilai tradisional melalui media interaktif, pendekatan enkripsi hybrid ini juga diharapkan dapat menjadi sarana efektif dalam melindungi data digital secara aman dan efisien.

2. Tinjauan Pustaka

Penelitian terdahulu telah mengilustrasikan penerapan algoritma enkripsi *AES* dan *RSA* dalam proses pengamanan pengiriman pesan teks. Metode ini memanfaatkan kekuatan *AES* sebagai enkripsi simetris yang efisien untuk mengenkripsi isi pesan, sementara *RSA* digunakan untuk mengamankan distribusi kunci enkripsi secara asimetris [1][3].

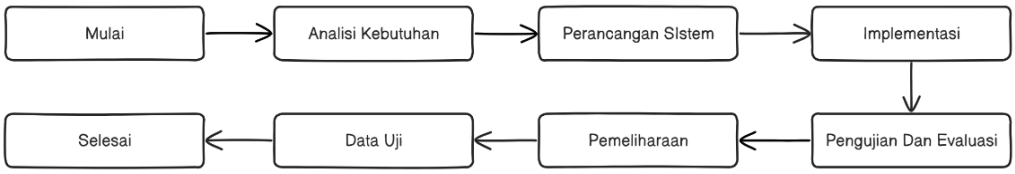
Penelitian yang dilakukan oleh Rinmar Siringoringo mengungkapkan penerapan algoritma enkripsi *AES* dan *RSA* secara bersamaan untuk mengamankan file digital. Dalam studinya, metode enkripsi *hybrid* ini digunakan untuk melindungi data file dengan menggabungkan kecepatan dan efisiensi *AES* dalam mengenkripsi isi file serta kekuatan *RSA* dalam mengamankan distribusi kunci enkripsi [5].

Pada penelitian yang dilakukan oleh Galih Wening Werdi Mukti menunjukkan penerapan algoritma enkripsi *AES* dan *RSA* dalam konteks pengamanan data pencatatan kesehatan. Dalam studinya, metode enkripsi hybrid ini digunakan untuk melindungi informasi sensitif pasien dengan menggabungkan kecepatan dan efisiensi *AES* dalam mengenkripsi data rekam medis, serta kekuatan *RSA* dalam mengamankan distribusi kunci enkripsi [6].

Berdasarkan berbagai penelitian terdahulu, penerapan enerapan sistem keamanan data berbasis enkripsi *hybrid AES-RSA* yang sepenuhnya terdesentralisasi dan terintegrasi dengan *cloud storage modern* [5]. Kebaruan penelitian (*novelty)* terletak pada model implementasi *end-to-end* yang mengutamakan *client-side encryption* dan integrasi langsung dengan *AWS S3*, yang belum banyak dieksplorasi dalam penelitian-penelitian sebelumnya.[7].

3. Metodologi

Penelitian ini menggunakan model pengembangan perangkat lunak *Waterfall* yang sistematis dan berurutan, dengan pengujian menggunakan metode *Black box* untuk memastikan aplikasi enkripsi *hybrid AES-RSA* yang terintegrasi dengan layanan *cloud AWS S3* berjalan sesuai harapan [8][9][10]. Tahapan pengembangan yang dilakukan dapat dilihat pada gambar alur penelitian.



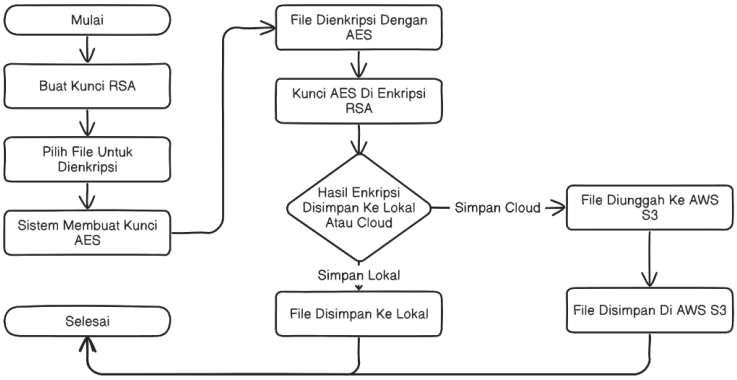
Gambar 1 Alur Penelitian

3.1 Analisis Kebutuhan

Pada tahap awal, dilakukan identifikasi kebutuhan sistem, seperti fitur enkripsi dan dekripsi [11][12], jenis file yang didukung (.*docx, .jpg, .mp4*), serta integrasi dengan *AWS S3* sebagai media penyimpanan file terenkripsi. Analisis ini menjadi dasar perancangan aplikasi.

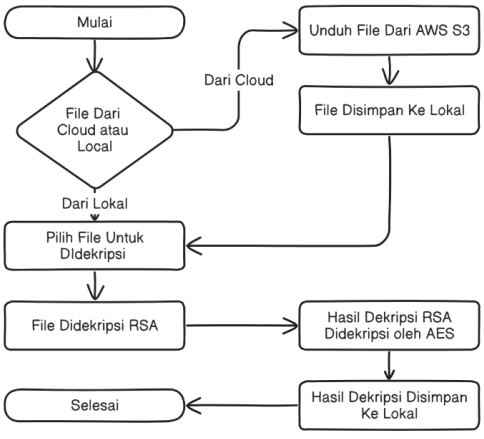
3.2 Perancangan Sistem

Merancang arsitektur aplikasi yang mencakup desain antarmuka pengguna berbasis *Python* dan *Tkinter*, alur kerja enkripsi *hybrid AES-RSA*, serta mekanisme pengelolaan kunci enkripsi. Desain juga mencakup integrasi dengan layanan *AWS S3* melalui *API*.



Gambar 2 *Flowchart* Enkripsi

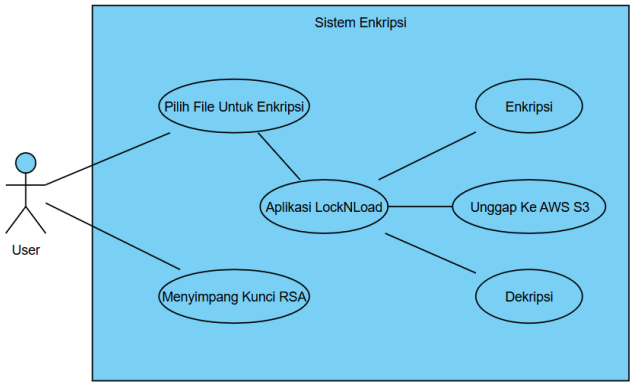
Pada proses enkripsi *user* diharus membuat terlebih dahulu kunci untuk *RSA* untuk melakukan enkripsi. Setelah itu memilih file yang akan di enkripsi dan memasukan kunci publik *RSA* terakhir memilih akan menyimpan file hasil enkripsi di lokal atau *cloud*. Pada tahap ini user bebas menentukan lokasi penyimpanan dan melakukan enkripsi.



Gambar 3 *Flowchart* Dekripsi

Saat proses dekripsi user harus menyiapkan file yang telah dienkripsi. Jika file ada di *AWS S3* maka user harus mengunduh terlebih dahulu filenya, setelah itu pilih filenya dilokal dan

pilih kunci privat *RSA* untuk melakukan dekripsi dan hasil file dekripsi akan disimpan dilokasi file sebelumnya.



Gambar 4 Diagram *Use Case*

Diagram ini memperlihatkan proses kerja utama dari aplikasi *LockNLoad* yang berfungsi untuk melindungi data menggunakan metode enkripsi *hybrid* serta terhubung dengan layanan *AWS S3*. Pengguna dapat dengan mudah memilih file yang ingin diamankan, melakukan enkripsi, mengunggah file terenkripsi ke *cloud*, dan melakukan dekripsi saat diperlukan. Sistem ini juga mengelola kunci *RSA* dengan aman, sehingga hanya pengguna yang memiliki kunci yang sah yang dapat mengakses data tersebut. Dengan demikian, aplikasi ini menjamin keamanan data selama penyimpanan dan pengiriman di lingkungan *cloud*.

3.3 Implementasi

Tahap ini meliputi pengkodean aplikasi sesuai rancangan menggunakan bahasa *Python*. Algoritma *AES* dan *RSA* diimplementasikan untuk proses enkripsi dan dekripsi, serta integrasi dengan *AWS S3* dilakukan untuk penyimpanan dan pengambilan file [13][14][15].

3.4 Pengujian dan Evaluasi

Pengujian dilakukan dengan metode *Black box*, yang berfokus pada pengujian fungsi aplikasi dari sisi pengguna tanpa melihat kode sumber. Pengujian meliputi:

1) Pengujian fungsional untuk memastikan proses enkripsi, dekripsi, *upload*, dan *download* file berjalan dengan baik.
2) Pengujian validasi input untuk memastikan aplikasi dapat menangani file yang tidak valid atau rusak dengan benar.
3) Pengujian kinerja untuk mengukur waktu proses enkripsi dan dekripsi serta respon aplikasi saat berinteraksi dengan *AWS S3*.

Hasil pengujian dianalisis untuk menilai keandalan, keamanan, dan performa aplikasi dalam melindungi data pengguna

3.5 Pemeliharaan

Setelah aplikasi dinyatakan memenuhi standar kelayakan dan siap digunakan, tahap pemeliharaan menjadi langkah penting berikutnya. Pada fase ini tidak diimplementasikan pada penelitian kali ini, namun pengembang secara rutin melakukan identifikasi dan perbaikan terhadap kesalahan atau bug yang mungkin muncul selama penggunaan aplikasi [16]. Selain itu, pemeliharaan juga mencakup pembaruan sistem untuk meningkatkan performa, menambah fitur baru, atau menyesuaikan dengan perubahan kebutuhan pengguna dan teknologi terkini. Proses ini bertujuan agar aplikasi tetap berjalan dengan optimal, aman, dan relevan dalam jangka panjang, sehingga memberikan pengalaman terbaik bagi penggunanya [17].

3.6 Data Uji

Data yang dijadikan bahan dalam penelitian ini meliputi berbagai jenis file, yaitu dokumen berformat *.docx*, gambar dengan ekstensi *.jpg*, serta video berformat *.mp4*. File-file tersebut dipilih dengan ukuran yang bervariasi, mulai dari yang sangat kecil sekitar 100 *KB* hingga yang cukup besar mencapai 200 *MB*. Data pengujian berjumlan 9 file dimana 3 dokumen *.docx*, 3 gambar *.jpg* dan 3 video *.mp4,* Masing-masing file dilkakukan enkripsi dan

dekripsi sebanyak 3 kali. Pemilihan rentang ukuran file yang luas ini bertujuan untuk menguji dan memperoleh gambaran menyeluruh mengenai kinerja aplikasi ketika menangani data dalam berbagai skala. Dengan demikian, evaluasi performa yang dilakukan dapat mencerminkan kemampuan aplikasi dalam mengelola file dengan ukuran dan tipe yang berbeda-beda secara efektif dan efisien [18].

Tabel 1 Data Uji

No.	Jenis File	Ukuran File
1.	*.docx*	100 *KB*
2.	*.docx*	1 *MB*
3.	*.docx*	5 *MB*
4.	*.jpg*	1 *MB*
5.	*.jpg*	5 *MB*
6.	*.jpg*	10 *MB*
7.	*.mp4*	50 *MB*
8.	*.mp4*	100 *MB*
9.	*.mp4*	200 *MB*

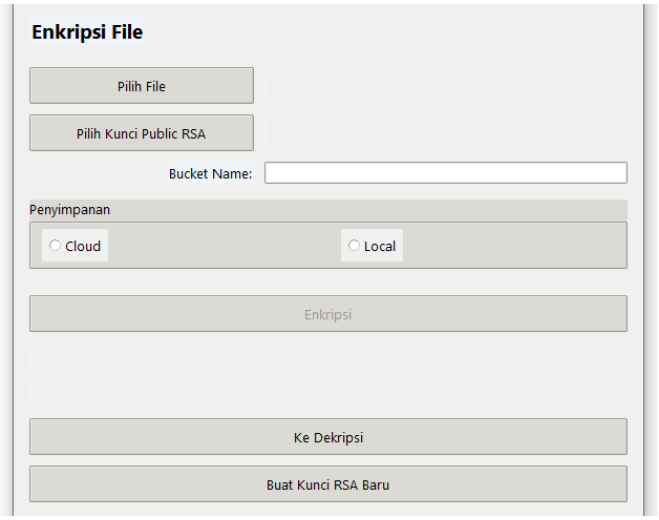
4. Hasil dan Pembahasan

Pada tahap implementasi, sistem berhasil dikembangkan sesuai dengan rancangan yang telah dirumuskan sebelumnya. Aplikasi ini mampu menjalankan fungsi utama yaitu enkripsi dan dekripsi file dengan menggunakan metode *hybrid AES-RSA* secara efektif. Selain itu, aplikasi juga menyediakan opsi penyimpanan hasil enkripsi baik secara lokal maupun melalui layanan *cloud AWS S3*, sehingga memberikan fleksibilitas bagi pengguna dalam mengelola data terenkripsi [19].

4.1 Hasil Implementasi Sistem

1) Halaman Enkripsi

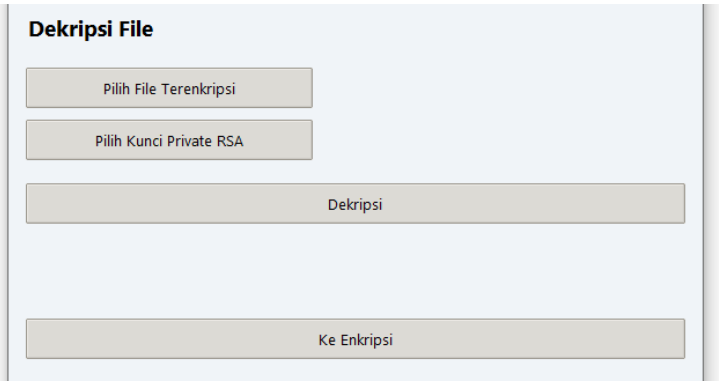
Halaman Enkripsi merupakan bagian utama dari aplikasi desktop yang mengintegrasikan dua algoritma kriptografi, yaitu *Advanced Encryption Standard (AES)* dan *Rivest-Shamir-Adleman (RSA)*, untuk melindungi data pengguna. Pada tampilan awal halaman ini, pengguna akan disajikan menu yang memungkinkan mereka melakukan proses enkripsi file dengan mudah. Selain itu, terdapat opsi navigasi yang memungkinkan pengguna beralih ke halaman dekripsi jika diperlukan. Fungsi utama dari halaman ini adalah menjalankan proses enkripsi data menggunakan kombinasi algoritma *AES* dan *RSA* secara efisien dan aman, sehingga data yang diolah dapat terlindungi dengan baik sebelum disimpan atau dikirimkan. Dengan antarmuka yang sederhana dan intuitif, halaman ini dirancang untuk memberikan pengalaman pengguna yang nyaman dalam mengamankan informasi digital mereka.



Gambar 5 Halaman Enkripsi

2) Halaman Dekripsi

Halaman Dekripsi adalah bagian dari aplikasi yang berfungsi khusus untuk mengembalikan file yang sebelumnya telah dienkripsi ke bentuk aslinya. Aplikasi ini dirancang dengan fokus utama memberikan kemudahan bagi pengguna dalam menjalankan proses dekripsi menggunakan kunci privat *RSA* yang sesuai. Pada halaman ini, pengguna dapat dengan mudah memilih file terenkripsi dan memasukkan kunci privat yang diperlukan untuk membuka data tersebut. Selain itu, halaman ini juga menyediakan akses cepat dan intuitif untuk beralih kembali ke fitur enkripsi, sehingga pengguna dapat berpindah antar fungsi dengan lancar. Desain antarmuka yang sederhana, dilengkapi dengan tombol-tombol yang jelas dan mudah dipahami, memastikan bahwa pengguna dari berbagai tingkat keahlian dapat menggunakan aplikasi ini tanpa kesulitan dalam memilih file dan kunci yang tepat untuk proses dekripsi. Dengan demikian, halaman ini mendukung pengalaman pengguna yang efisien dan aman dalam mengelola data terenkripsi mereka.



Gambar 6 Halaman Dekripsi

4.2 Hasil Pengujian Sistem

Pengujian sistem meliputi pengujian *black box* untuk memastikan fungsi aplikasi berjalan sesuai spesifikasi, pengujian pembuatan kunci untuk memvalidasi proses generasi kunci kriptografi, serta pengujian data guna mengevaluasi kemampuan sistem dalam menangani berbagai jenis dan ukuran data secara efektif.

1) Pengujian *Black Box*

Proses yang menjadi fokus dalam pengujian *black box* ini mencakup berbagai tahapan yang telah dirinci secara lengkap dalam Tabel 1. Pengujian tersebut bertujuan untuk memastikan bahwa setiap fungsi sistem berjalan sesuai dengan spesifikasi tanpa melihat struktur internal aplikasi. Dengan demikian, evaluasi dilakukan berdasarkan hasil keluaran yang diharapkan dari setiap proses yang diuji.

Tabel 2 Pengujian *Black Box*

Skenario	Hasil Diharapkan	Hasil
Pengguna menekan tombol Buat Kunci *RSA* Baru	Menghasilkan *public key* dan *private key*	Berhasil
Pengguna menekan tombol Pilih File kemudian menekan tombol Pilih Kunci *RSA* Publik kemudian memasukan *Bucket* name kemudian memilih penyimpanan *Cloud* dan Menekan Tombol Enkripsi	Menghasillan file terenkripsi	Berhasil
Pengguna tidak menekan tombol Pilih File kemudian menekan tombol Pilih Kunci *RSA* Publik kemudian memasukan *Bucket* name kemudian memilih penyimpanan *Cloud* dan Menekan Tombol Enkripsi	*Error* tidak memilih file	Berhasil

Skenario	Hasil Diharapkan	Hasil
Pengguna menekan tombol Pilih File kemudian menekan tombol Pilih Kunci *RSA* Publik yang salah kemudian memasukan *Bucket* name kemudian memilih penyimpanan *Cloud* dan Menekan Tombol Enkripsi	*Error* salah memasukan kunci	Berhasil
Pengguna menekan tombol Pilih File kemudian menekan tombol Pilih Kunci *RSA* Publik kemudian memasukan *Bucket* name yang salah kemudian memilih penyimpanan *Cloud* dan Menekan Tombol Enkripsi	Error nama *bucket* salah	Berhasil
Pengguna menekan tombol Pilih File kemudian menekan tombol Pilih Kunci *RSA* Publik kemudian memilih penyimpanan Lokal dan Menekan Tombol Enkripsi	Menghasillan file terenkripsi	Berhasil
Pengguna tidak menekan tombol Pilih File kemudian menekan tombol Pilih Kunci *RSA* Publik kemudian memasukan *Bucket* name kemudian memilih penyimpanan Lokal dan Menekan Tombol Enkripsi	*Error* tidak memilih file	Berhasil
Pengguna menekan tombol Pilih File kemudian menekan tombol Pilih Kunci *RSA* Publik yang salah kemudian memasukan *Bucket* name kemudian memilih penyimpanan Lokal dan Menekan Tombol Enkripsi	*Error* salah memasukan kunci	Berhasil
Pengguna menekan tombol Pilih File kemudian menekan tombol Pilih Kunci *RSA* Publik kemudian memasukan *Bucket* name yang salah kemudian memilih penyimpanan Lokal dan Menekan Tombol Enkripsi	*Error* nama bucket salah	Berhasil
Pengguna menekan tombol Pilih File kemudian menekan tombol Pilih Kunci *RSA* Publik kemudian memasukan *Bucket* name kemudian tidak memilih penyimpanan dan Menekan Tombol Enkripsi	Tombol enkripsi tidak bisa ditekan	Berhasil
Pengguna menekan tombol Pilih File kemudian menekan tombol Pilih Kunci *RSA Private* dan Menekan Tombol Dekripsi	Menghasilkan file terdekripsi	Berhasil
Pengguna tidak menekan tombol Pilih File kemudian menekan tombol Pilih Kunci *RSA* Private dan Menekan Tombol Dekripsi	*Error* tidak memilih file	Berhasil
Pengguna menekan tombol Pilih File kemudian menekan tombol Pilih Kunci *RSA Private* yang salah dan Menekan Tombol Dekripsi	*Error* salah memasukan kunci	Berhasil

Berdasarkan hasil pengujian *black box*, dapat disimpulkan bahwa sistem yang dikembangkan mampu membedakan antara proses yang berjalan dengan benar dan yang mengalami kesalahan. Dengan demikian, secara fungsional sistem ini berhasil menghasilkan *output* yang sesuai dengan harapan dan tujuan penelitian. Hal ini menunjukkan bahwa aplikasi telah memenuhi persyaratan fungsional yang ditetapkan dan dapat diandalkan dalam penggunaannya.

2) Pengujian Pembuatan Kunci *RSA*

Sebelum melaksanakan pengujian data, tahap awal yang dilakukan adalah pengujian terhadap proses pembuatan kunci privat dan kunci publik. Kunci yang dihasilkan menggunakan panjang kunci sebesar 2048 bit, yang merupakan standar keamanan yang cukup kuat untuk memastikan integritas dan kerahasiaan data dalam sistem. Pengujian ini bertujuan untuk memastikan bahwa kunci yang dibuat valid dan dapat digunakan secara efektif dalam proses enkripsi dan dekripsi.

3) Pengujian Data

Setelah proses pengujian pembangkitan kunci selesai dilakukan, peneliti melanjutkan dengan pengujian terhadap data yang akan digunakan dalam sistem. Hasil dari pengujian ini dapat dilihat secara rinci pada Tabel 2 yang menampilkan hasil pengujian enkripsi secara lokal, Tabel 3 yang memperlihatkan hasil enkripsi data yang diunggah ke *cloud*, serta Tabel 4 yang memuat hasil pengujian proses dekripsi. Data-data tersebut memberikan gambaran lengkap mengenai performa dan keandalan sistem dalam mengelola enkripsi dan dekripsi pada berbagai kondisi penggunaan.

Tabel 3 Hasil Enkripsi ke Lokal

Data Ke-	Tipe	Ukuran	Durasi Enkripsi 1	Durasi Enkripsi 2	Durasi Enkripsi 3	Durasi Enkripsi Rata-rata
Data 1	*.docx*	100 *KB*	0.03 Detik	0.03 Detik	0.02 Detik	0.02 Detik
Data 2	*.docx*	1 *MB*	0.03 Detik	0.02 Detik	0.02 Detik	0.02 Detik
Data 3	*.docx*	5 *MB*	0.05 Detik	0.03 Detik	0.05 Detik	0.04 Detik
Data 4	*.jpg*	1 *MB*	0.05 Detik	0.05 Detik	0.09 Detik	0.06 Detik
Data 5	*.jpg*	5 *MB*	0.05 Detik	0.05 Detik	0.03 Detik	0.04 Detik
Data 6	*.jpg*	10 *MB*	0.08 Detik	0.06 Detik	0.05 Detik	0.06 Detik
Data 7	*.mp4*	50 *MB*	0.94 Detik	0.30 Detik	0.28 Detik	0.50 Detik
Data 8	*.mp4*	100 *MB*	0.98 Detik	0.64 Detik	0.66 Detik	0.76 Detik
Data 9	*.mp4*	200 *MB*	2.86 Detik	2.75 Detik	2.42 Detik	2.67 Detik

Durasi enkripsi meningkat secara signifikan seiring dengan bertambahnya ukuran file, terutama untuk file berukuran besar seperti video *(.mp4*). Untuk file kecil dan sedang, durasi enkripsi relatif sangat cepat dan hampir konstan. Dengan durasi rata-rata 0.46 detik.

Tabel 4 Hasil Enkripsi ke *Cloud*

Data Ke-	Tipe	Ukuran	Durasi Enkripsi 1	Durasi Enkripsi 2	Durasi Enkripsi 3	Durasi Enkripsi Rata-rata
Data 1	*.docx*	100 *KB*	2.83 Detik	1.63 Detik	1.43 Detik	1.96 Detik
Data 2	*.docx*	1 *MB*	2.21 Detik	1.59 Detik	1.72 Detik	1.84 Detik
Data 3	*.docx*	5 *MB*	2.28 Detik	2.34 Detik	2.27 Detik	2.29 Detik
Data 4	*.jpg*	1 *MB*	1.71 Detik	2.68 Detik	1.83 Detik	2.07 Detik
Data 5	*.jpg*	5 *MB*	2.36 Detik	2.38 Detik	2.37 Detik	2.37 Detik
Data 6	*.jpg*	10 *MB*	3.66 Detik	8.65 Detik	3.51 Detik	5.27 Detik
Data 7	*.mp4*	50 *MB*	12.54 Detik	13.20 Detik	35.64 Detik	20.46 Detik
Data 8	*.mp4*	100 *MB*	37.18 Detik	25.63 Detik	21.62 Detik	28.14 Detik
Data 9	*.mp4*	200 *MB*	50.80 Detik	59.12 Detik	64.76 Detik	58.22 Detik

Kecepatan enkripsi dan upload ke *cloud* sangat dipengaruhi oleh ukuran file. Untuk file kecil dan sedang, proses relatif cepat dengan kecepatan internet 40 *Mbps*. Namun, untuk file *video* besar, waktu enkripsi/*upload* bisa mencapai puluhan detik bahkan lebih, dengan variasi waktu yang cukup besar antar percobaan. Dengan waktu rata-rata 13.64 detik.

Tabel 5 Hasil Dekripsi

Data Ke-	Tipe	Ukuran	Durasi Enkripsi 1	Durasi Enkripsi 2	Durasi Enkripsi 3	Durasi Enkripsi Rata-rata
Data 1	*.docx*	100 *KB*	0.30 Detik	0.08 Detik	0.09 Detik	0.15 Detik
Data 2	*.docx*	1 *MB*	0.11 Detik	0.09 Detik	0.11 Detik	0.10 Detik
Data 3	*.docx*	5 *MB*	0.14 Detik	0.09 Detik	0.11 Detik	0.11 Detik
Data 4	*.jpg*	1 *MB*	0.11 Detik	0.08 Detik	0.09 Detik	0.09 Detik
Data 5	*.jpg*	5 *MB*	0.12 Detik	0.11 Detik	0.11 Detik	0.11 Detik
Data 6	*.jpg*	10 *MB*	0.17 Detik	0.11 Detik	0.14 Detik	0.14 Detik
Data 7	*.mp4*	50 *MB*	0.82 Detik	0.42 Detik	0.39 Detik	0.54 Detik
Data 8	*.mp4*	100 *MB*	1.00 Detik	0.73 Detik	0.74 Detik	0.82 Detik
Data 9	*.mp4*	200 *MB*	2.51 Detik	2.63 Detik	1.76 Detik	2.30 Detik

Secara keseluruhan, proses dekripsi menunjukkan kinerja yang sangat baik dan efisien, bahkan ketika menangani file dengan ukuran besar sekalipun. Waktu rata-rata yang dibutuhkan untuk menyelesaikan proses ini tercatat sekitar 0,48 detik, menandakan bahwa sistem mampu melakukan dekripsi dengan cepat tanpa mengorbankan performa, sehingga sangat cocok untuk aplikasi yang memerlukan responsivitas tinggi dalam pengolahan data berukuran besar.

4) Pengujian Modifikasi file hasil enkripsi

Untuk memastikan file yang telah terenkripsi aman dan tidak dapat dimodifikasi pada tahapan ini dilakukan beberapa pengujian yaitu merubah bit pada bagian kunci AES dan chipertext.

Tabel 6 Perubahan bit pada kunci AES

No.	Tipe File	Bit 150 Sebelum	Bit 150 Sesudah	Hasil Keluaran
Data 1	*.docx*	0D	1D	*Incorrect decryption*
Data 2	*.docx*	33	34	*Incorrect decryption*
Data 3	*.docx*	CA	CB	*Incorrect decryption*
Data 4	*.jpg*	A0	A1	*Incorrect decryption*
Data 5	*.jpg*	Ea	Eb	*Incorrect decryption*
Data 6	*.jpg*	05	15	*Incorrect decryption*
Data 7	*.mp4*	3d	4d	*Incorrect decryption*
Data 8	*.mp4*	33	34	*Incorrect decryption*
Data 9	*.mp4*	89	90	*Incorrect decryption*

Tabel 7 Perubahan bit pada chipertext

No.	Tipe File	Bit 150 Sebelum	Bit 150 Sesudah	Hasil Keluaran
Data 1	*.docx*	23	24	*Mac check failed*
Data 2	*.docx*	F5	F6	*Mac check failed*
Data 3	*.docx*	C4	C5	*Mac check failed*
Data 4	*.jpg*	ed	ee	*Mac check failed*
Data 5	*.jpg*	A1	A2	*Mac check failed*
Data 6	*.jpg*	F8	F9	*Mac check failed*
Data 7	*.mp4*	37	38	*Mac check failed*
Data 8	*.mp4*	B8	B9	*Mac check failed*
Data 9	*.mp4*	66	67	*Mac check failed*

Tabel 8 Penggunaan privat key RSA

No.	Tipe *File*	Kunci Asli	Kunci Palsu	Hasil Keluaran
Data 1	*.docx*	*Private.key.pem*	*Private.key.pem*	*Incorrect decryption*
Data 2	*.docx*	*Private.key.pem*	*Private.key.pem*	*Incorrect decryption*
Data 3	*.docx*	*Private.key.pem*	*Private.key.pem*	*Incorrect decryption*
Data 4	*.jpg*	*Private.key.pem*	*Private.key.pem*	*Incorrect decryption*
Data 5	*.jpg*	*Private.key.pem*	*Private.key.pem*	*Incorrect decryption*
Data 6	*.jpg*	*Private.key.pem*	*Private.key.pem*	*Incorrect decryption*
Data 7	*.mp4*	*Private.key.pem*	*Private.key.pem*	*Incorrect decryption*
Data 8	*.mp4*	*Private.key.pem*	*Private.key.pem*	*Incorrect decryption*
Data 9	*.mp4*	*Private.key.pem*	*Private.key.pem*	*Incorrect decryption*

4.3 Pembahasan

Dari hasil pengujian modifikasi bit pada file hasil enkripsi didapatkan hasil file tidak bisa didekripsi. Tabel 6 melakukan perubahan bit ke 150 pada kunci AES ini akan membuat kunci AES jika digunakan kembali maka proses dekripsi akan gagal karena AES tidak mentoleransi perbedaan walaupun 1 bit dan itu akan merubah ciphertext sepenuhnya dan hasil dekripsi akan menjadi text acak bahkan gagal.

Pada tabel 7 bit ciphertext dilakukan perubahan salah satu bit maka hasil dekripsi akan rusak karena *ciphertext* tidak bisa dikenali oleh AES dan RSA kerusakan *ciphertext* tergantung pada bit keberapa yang diubah. Ini salah satu fitur keamanan yang diberikan agar tidak ada

perbedaan satu bit sebelum dan sesusah dekripsi dilakukan untuk menjadi keaslian file *ciphertext* tersebut [20].

Terakhir pada tabel 8 bit pada kunci private RSA diubah 1 saja maka seluruh proses dekripsi dan penandatanganan digital akan gagal total karena RSA bergantung pada struktur matematika yang sangat presisi. *Library kriptografi PyCryptodome* bisa langsung menolak kunci yang berbeda.

5. Simpulan

Penelitian mengenai penerapan enkripsi data dengan metode *hybrid* yang menggabungkan algoritma *AES* dan *RSA*, yang masing-masing menggunakan tipe enkripsi simetris dan asimetris, menunjukkan bahwa pendekatan ini efektif untuk mengamankan proses enkripsi dan dekripsi data. Keamanan sistem ini didukung oleh kebutuhan akan dua tahap dekripsi serta penggunaan kunci privat untuk mengakses informasi asli. Dalam implementasinya, kunci yang direkomendasikan adalah 256 *bit* untuk *AES* dan 2048 *bit* untuk *RSA* guna memastikan tingkat keamanan yang optimal. Perlu dicatat bahwa proses dekripsi cenderung memakan waktu sedikit lebih lama dibandingkan dengan proses enkripsi. Berdasarkan hasil pengujian, rata-rata waktu yang dibutuhkan untuk enkripsi data secara lokal adalah sekitar 0,46 detik, sementara untuk proses penyimpanan data ke *cloud* memerlukan waktu rata-rata 13,64 detik. Sedangkan proses dekripsi rata-rata memakan waktu sekitar 0,48 detik. Data ini menggambarkan bahwa meskipun proses enkripsi dan dekripsi berjalan dengan efisien, pengiriman data ke *cloud* menjadi faktor yang paling mempengaruhi durasi keseluruhan proses keamanan data.

Penelitian ini masih menyisakan beberapa keterbatasan yang perlu diatasi melalui studi lanjutan untuk mencapai hasil yang lebih optimal. Beberapa aspek yang sebaiknya menjadi fokus dalam penelitian berikutnya antara lain: 1. Penelitian selanjutnya dapat diarahkan pada peningkatan keamanan dengan mengadopsi teknik enkripsi yang lebih maju dan kompleks, yang mengoptimalkan integrasi algoritma *AES* dan *RSA* untuk memperkuat perlindungan data secara menyeluruh. 2. Selain itu, penelitian selanjutnya dapat memperluas fungsi aplikasi dengan menambahkan fitur yang memungkinkan penyimpanan data terenkripsi tidak hanya pada *AWS S3*, tetapi juga pada berbagai layanan *cloud storage* lainnya, sehingga meningkatkan fleksibilitas dan kompatibilitas aplikasi dalam berbagai lingkungan penyimpanan awan

Daftar Referensi

[1] A. Hermawan and H. I. E. Ujianto, “Implementasi Enkripsi Data Menggunakan Kombinasi AES dan RSA,” *J. Nas. Inform. dan Teknol.*, vol. 5, no. 2, pp. 325–330, 2021.

[2] D. Satrinia, S. N. Yutia, and I. M. M. Matin, “Analisis Keamanan dan Kenyamanan pada Cloud Computing,” *J. Informatics Commun. Technol.*, vol. 4, no. 1, pp. 85–91, 2022, doi: 10.52661/j_ict.v4i1.111.

[3] R. Siringoringo *et al.*, “Pengujian Kinerja Web Server Elastic Cloud Compute (Ec2) Free Tier Pada Amazon Web Service (Aws) Menggunakan Jmeter,” *J. Teknol. Dan Ilmu Komput. Prima*, vol. 4, no. 1, pp. 82–94, 2020, doi: 10.1088/1757-899X/852/1/012148.

[4] T. Limbong, “Pengujian kriptografi klasik caesar chipper menggunakan matlab,” *Semin. Nas. Inov. dan Teknol. Inf. 2015*, no. February, pp. 77–80, 2017, [Online]. Available: https://www.researchgate.net/publication/313791310

[5] R. Siringoringo, “Analisis dan Implementasi Algoritma Rijndael (AES) dan Kriptografi RSA pada Pengamanan File,” *KAKIFIKOM (Kumpulan Artik. Karya Ilm. Fak. Ilmu Komputer)*, vol. 02, no. 01, pp. 31–42, 2020, doi: 10.54367/kakifikom.v2i1.666.

[6] G. W. W. Mukti and H. Setiawan, “Designing and building secure electronic medical record application by applying AES-256 and RSA digital signature,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 852, no. 1, pp. 1–7, 2020, doi: 10.1088/1757-899X/852/1/012148.

[7] R.R. Permata, A.M. Ramli, S.D. Rosadi, & T.S. Ramli, "Pelanggaran Kekayaan Intelektual dalam Cloud Computing". ACTA DIURNAL Jurnal Ilmu Hukum Kenotariatan, vol. 4, no. 2, pp. 226-243, 2021.

[8] T. Tukino and B. Huda, “Penerapan Algoritma K-Means Untuk Mendukung Keputusan Dalam Pemilihan Tema Tugas Akhir Pada Prodi Sistem Informasi Universitas Buana Perjuangan Karawang.,” *Techno Xplore J. Ilmu Komput. dan Teknol. Inf.*, vol. 4, no. 1, pp. 1–10, 2019, doi: 10.36805/technoxplore.v4i1.542.

[9] B. Priyatna, S. S. Hilabi, N. Heryana, and A. Solehudin, "293325-Aplikasi-Pengenalan-Tarian-Dan-Lagu-Trad-80a5226F," vol. 1, no. 2, pp. 89–98, 2019.
[10] F. Alkhadzik, S. S. Hilabi, N. Kastiawan, and ..., "Implementasi Sistem Manajemen Keuangan Berbasis Website Pada 'UMKM Teh Mvit,'" *J. Multimed. dan …*, pp. 82–94, 2023, [Online]. Available: http://journal.cattleyadf.org/index.php/jatilima/article/view/416%0Ahttp://journal.cattleyadf.org/index.php/jatilima/article/download/416/316
[11] F. Nurapriani, "Meningkatkan Kemampuan Komunikasi Dan Koneksi Matematik Siswa Smp Melalui Strategi Think Talk Write," *Buana Ilmu*, vol. 1, no. 1, pp. 45–55, 2016, doi: 10.36805/bi.v1i1.97.
[12] S. A. Alkadrie, "Keamanan Cloud Computing di Era Industri 4 . 0 : Systematic Literature Review," vol. 4, no. 2, pp. 1–15, 2024.
[13] A. L. Hananto, A. Hananto, B. Huda, A. Y. Rahman, E. Novalia, and B. Priyatna, "Determination of Training Participants in Community Work Training Centers Using the Naïve Bayes Classifier Algorithm," *Int. J. Informatics Vis.*, vol. 8, no. 3, pp. 1162–1167, 2024, doi: 10.62527/joiv.8.3.1995.
[14] S. S. Hilabi, "Rancang Bangun Sistem Inventory Usaha (UMKM) 'Karpet' Desa Kamurang Berbasis Web," *Pros. Konf. Nas. Penelit. dan Pengabdi.*, no. 2, pp. 1147–1155, 2022.
[15] S. Sinha, M. Shankar, Y. Pande, K. Kumar, S. Sharma, and K. Virdi, "Secure Cloud Storage using End-to-End Encryption," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 11, no. 12, pp. 567–574, 2023, doi: 10.22214/ijraset.2023.57414.
[16] A. Mulyanto and Abimanyu, "Penerapan Cover Generation Steganografi Dan Kriptografi Rsa Pada Enkripsi Kunci Simetris Aes Rijndael Untuk Keamanan Data Dalam Jaringan Lan Pt. Hero Supermarket Tbk.," *J. Inform. SIMANTIK*, vol. 4, no. 1, pp. 1–8, 2019.
[17] G. V Liander, "Penggunaan Algoritma Elliptic Curve Diffie Hellman dan AES 256 pada Implementasi End-to-End Encryption WhatsApp," *Sumber*, no. 18219075, 2022.
[18] K. Fadhlil Khaliq, "Pengamanan Data Akta Dengan Metode Aes Berbasis Cloud Computing," *J. Teknol. Dan Ilmu Komput. Prima*, vol. 4, no. 1, pp. 509–512, 2021, doi: 10.34012/jutikomp.v4i1.1555.
[19] M. R. Alfazry, F. Fadilah, A. P. Putra, and A. Setiawan, "Perlindungan Keamanan Website NextCloud: Mengatasi Serangan DoS dengan Konfigurasi Firewall pada Ubuntu," *J. Internet Softw. Eng.*, vol. 1, no. 3, p. 11, 2024, doi: 10.47134/pjise.v1i3.2639.
[20] M. Prerna, A. Sachdeva, and P. Mahajan, "A Study of Encryption Algorithms AES, DES and RSA for Security," *Type Double Blind Peer Rev. Int. Res. J. Publ. Glob. Journals Inc*, vol. 13, no. 15, pp. 1–9, 2013.