Edu Elektrika Journal Vol 13 No 1
Januari – Juni 2025

E-ISSN 2723-5602
P-ISSN 2252-7095

https://journal.unnes.ac.id/journals/eduel/index

Analisis Pengamanan File Menggunakan Enkripsi dan Dekripsi dengan Algoritma *AES-GCM-SIV*

Asgap Putra Zulian[1], Kartarina[2], dan I Made Yadi Dharma[3]

[1, 2,3]*Universitas Bumigora*
Jl. Ismail Marzuki No.22, Cilinaya, Kec. Cakranegara, Kota Mataram, Nusa Tenggara Barat, 83127
asgafputra9@gmail.com[1], kartarina@universitasbumigora.ac.ids[2], yadi_dharma@universitasbumigora.ac.id[3]

Abstrak— Keamanan file pada aplikasi web merupakan aspek krusial dalam menjaga integritas, kerahasiaan, dan ketersediaan data pengguna. Seiring meningkatnya ancaman keamanan digital, diperlukan pendekatan kriptografi yang mampu mengatasi kelemahan algoritma konvensional, terutama terhadap serangan akibat reuse nonce. Penelitian ini membahas implementasi algoritma AES-GCM-SIV sebagai metode enkripsi dan dekripsi file untuk meningkatkan perlindungan terhadap ancaman tersebut. Sistem dikembangkan menggunakan model SDLC Agile dan diimplementasikan pada framework Laravel untuk sisi back-end, serta React untuk front-end. Proses kriptografi dijalankan melalui binary eksternal berbasis bahasa pemrograman Rust guna memperoleh performa dan efisiensi optimal. Evaluasi dilakukan dengan membandingkan performa dan tingkat keamanan antara algoritma AES-GCM-SIV dan AES-GCM. Hasil pengujian menunjukkan bahwa meskipun AES-GCM memiliki waktu eksekusi lebih cepat, AES-GCM-SIV memberikan keunggulan signifikan dari sisi keamanan, terutama dalam skenario penggunaan ulang nonce, di mana kerahasiaan data tetap terjaga. Dengan demikian, AES-GCM-SIV layak dipertimbangkan sebagai solusi kriptografi yang lebih aman untuk pengamanan file pada aplikasi web yang membutuhkan jaminan keamanan tingkat tinggi.

Kata kunci— Kriptografi, AES-GCM-SIV, AES-GCM, Laravel, Rust, keamanan aplikasi web.

Abstract— File security in web applications is a crucial aspect in maintaining the integrity, confidentiality, and availability of user data. As digital security threats increase, a cryptographic approach is needed that can overcome the weaknesses of conventional algorithms, especially against attacks due to nonce reuse. This study discusses the implementation of the AES-GCM-SIV algorithm as a file encryption and decryption method to improve protection against these threats. The system is developed using the Agile SDLC model and implemented on the Laravel framework for the back-end, and React for the front-end. The cryptographic process is run through an external binary based on the Rust programming language to obtain optimal performance and efficiency. The evaluation was carried out by comparing the performance and level of security between the AES-GCM-SIV and AES-GCM algorithms. The test results show that although AES-GCM has a faster execution time, AES-GCM-SIV provides significant advantages in terms of security, especially in nonce reuse scenarios, where data confidentiality is maintained. Thus, AES-GCM-SIV is worth considering as a more secure cryptographic solution for securing files in web applications that require high-level security guarantees.

Keywords— Kriptografi, AES-GCM-SIV, AES-GCM, Laravel, Rust, Web Security.

I. PENDAHULUAN

Di era digital saat ini, keamanan data menjadi sala satu aspek yang sangat krusiak dalam dunia teknologi informasi. Meningkatnya penggunaan teknologi berbasis internet dan penyimpanan cloud telah memberikan kemudahan dalam pengelolaan serta akses data. Namun disisi lain, ancaman terhadap keamanan data juga semakin meningkat, salah satunya adalah kebocoran data (data breach) (Kusuma & Rahmani, 2022.).

Dalam beberapa tahun terakhir, indonesia mengalami berbagai insiden kebocoran data yang berdampak besar. Salah satu kasus terbesar adalah kebocoran data pengguna e-commerce tokopedia pada tahun 2020, dimana lebih dari 92 juta data pengguna dilaporkan bocor dan di perjual belikan di dark web (Destriani Firmansyah Putri et al., 2021). Selain itu kebocoran data BPJS kesehatan pada tahun 2021 juga mengejutkan publik dengan dugaan 279 juta data penduduk indonesia terekspos.

Salah satu satu solusi dalam mengatasi ancaman ini adalah melalui penerapan kriptografi. Kriptografi merupakan ilmu sekaligus seni untuk menjaga kerahasiaan pesan dengan cara menyamarkannya menjadi bentuk tersandi yang mempunyai makna (Sari et al., 2022). Terdapat dua konsep utama dalam Kriptografi, yaitu Enkripsi dan Dekripsi (Febrianto & Waluyo, 2023). Enkripsi adalah suatu proses Dimana informasi/data yang akan dikirim diubah menggunakan algoritma tertentu menjadi bentuk yang hamper tidak dapat dikenali sebagai data aslinya. Deskripsi adalah kebalikan dari enkripsi, yaitu konversi bentuk yang

Diterima: 5 Juni 2025 | Disetujui: 14 Juli 2025 |Dipublikasi: 15 Juli 2025

E-ISSN 2723-5602
P-ISSN 2252-7095
https://journal.unnes.ac.id/journals/eduel/index

disamarkan menjadi data asli (Harun Alfirdaus et al., 2023). Salah satu algoritma kriptografi yang banyak digunakan adalah Advance Encryption Standard (AES) (Pinuyut et al., 2024.). AES merupakan algoritma simetris yang menggunakan kunci yang sama untuk proses enkripsi dan dekripsi data (Azhari et al., 2022). Sejak diadopsi sebagai standar oleh National Institute of Standards and Technology (NIST) pada tahun (2001), AES telah menjadi pilihan utama dalam berbagai aplikasi keamanan data karena keamanannya yang tinggi dan efiesiensi dalam pengolohan data (Ahmad Sitorus et al., 2020). Salah satu mode popular dari AES adalah AES-GCM (Galouis/Counter Mode), yang menggabungkan kerahasiaan data dengan autentikasi untuk mencegah manipulasi (Albertini et al., 2022.).

Mode AES-GCM merupakan salah satu mode operasi popular dalam enkripsi yang digunakan untuk memastikan keamanan data. GCM menawarkan kombinasi antara kerahasiaan (confidentiality) dan keaslian data (authenticity) (Kim et al., 2020). Namun salah satu kelemahan GCM adalah ketergantungannya pada nonce yang unik untuk setiap operasi enkripsi (Karnaukhov et al., 2024).

Untuk mengatasi masalah tersebut, dikembangkan varian AES-GCM-SIV. AES-GCM-SIV adalah varian dari GCM yang dirancang untuk memberikan keamanan yang lebih kuat terhadap penggunaan ulang nonce (nonce misuse-resistance) (Chung et al., 2025). Dalam AES-GCM-SIV, pendekatan yang digunakan adalah mengganti ketergantungan pada nonce unik dengan cara menggabungkan elemen nonce dengan data autentikasi untuk menghasilkan nilai inisialisasi sintetis (Synthetic initialization Vector) (Choi et al., 2021.). Hal ini memungkinkan AES-GCM-SIV tetap aman meskipun terjadi kesalahan dalam pengelolaan nonce. Selain itu SIV juga mempertahankan efisiensi dari GCM, baik dalam aspek kecepatan enkripsi maupun autentikasi (Pirzada et al., 2020).

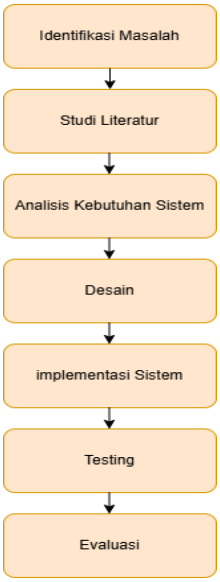
Algoritma AES-GCM-SIV dirancang untuk memberikan keamanan data yang kuat dengan memanfaatkan nonce dan penghitung untuk menghasilkan kunci dinamis yang unik pada setiap operasi, sehingga mampu mencegah kelemahan pada mode enkripsi tradisional, seperti kebocoran informasi akibat penggunaan ulang nonce atau plaintext yang serupa (Beyne et al., n.d.). Hal ini menjadikannya solusi ideal untuk implementasi enkripsi yang andal dan efisien dalam melindungi data sensitif pada aplikasi web[].

Tantangan utama dalam implementasi enkripsi adalah manajamen kunci yang aman (Suranta et al., 2022). Implementasi enkripsi sering kali melibatkan kerja sama antara front-end dan back-end. Dalam konteks aplikasi web, data yang diterima atau dikirimkan dari front-end harus di enkripsi dan didekripsi secara aman tanpa mengekspos key nya. Penggunaan algoritma AES-GCM-SIV dalam pengamanan file pada aplikasi web menawarkan solusi yang kuat untuk melindungi data sensitive.

Penelitian ini bertujuan untuk menganalisis dan mengimplementasikan pengamanan file di aplikasi web dengan fitur unggah dan unduh segala jenis file. File ini sering kali berisi data sensitive, seperti tugas akademik, materi pembelajaran, atau laporan penelitian sehingga penulis menggunakan algoritma AES-GCM-SIV yang memiliki mekanisme pengamanan yang kuat. Dengan pendekatan yang tepat, diharapkan dapat dihasilkan solusi keamanan data yang tidak hanya kuat, tetapi juga mendukung pengalaman pengguna yang optimal

II. METODOLOGI

Dalam penelitian ini, penulis menggunakan prosedur penelitian yang menggambarkan secara jelas tahapan-tahapan yang dilakukan. Meetodologi penelitian berperan penting dalam setiap studi, karena membantu mempermudah pelaksanaan dan penyusunan penelitian secara sistematis, berikut tahapan dari prosedur penelitian:



Gambar 1. Prosedur penelitian

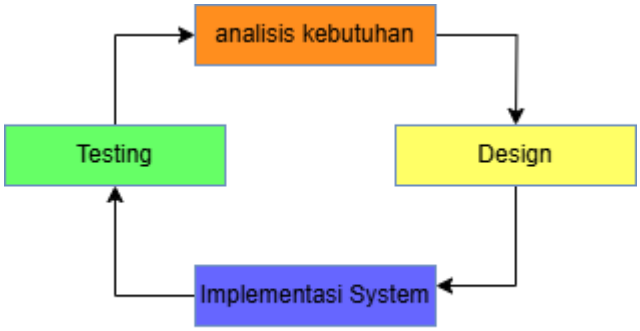
2.1 Metode Penelitan

Studi literatur adalah metode penelitian yang dilakukan dengan mengumpulkan , menelaah, dan menganalisis berbagai refrensi atau sumber ilmiah yang relevan dengan topik yang sedang diteliti, studi ini bertujuan untuk memahami konsep, teori, serta penelitian sebelumnya yang berkaitan dengan topik tersebut.

2.2. Metode Pengembangan Sistem

Pada métodо pengembangan system penulis disini menggunakan SDLC (*Software Development Life Cycle*) model *Agile*, dengan alur yang digunakan sebagai berikut:

E-ISSN 2723-5602
P-ISSN 2252-7095
https://journal.unnes.ac.id/journals/eduel/index



Gambar Error! No text of specified style in document.**2. Alur yang digunakan**

2.2.1. Análisis Kebutuhan Sistem

Análisis kebutuhan merupakan tahap awak dalam pengembangan sistema yang bertujuan untuk mengidentifikasi dan memahami kebutuhan fungsional dan non fungsional dari system yang akan dibangun.

2.2.1.1. Kebutuhan fungsional

Kebutuhan fungsional mencakup fitur-fitur yang harus ada dalam system untuk memenuhi tujuan penelitian. Beberapa kebutuhan fungsional yang di identifikasi adalah

- Unggah file
- Enkripsi file
- Dekripsi file
- Download file
- Manajemen kunci
- Autentikasi pengguna

2.2.1.2. Kebutuhan Non fungsional

Kebutuhan non fungsional mencakup aspek-aspek yang mendukung kinerja system keamanan, dan pengalaman pengguna.

2.2.1.3. Kebutuhan perangkat lunak

Perangkat lunak yang akan digunakan untuk membangun sistema ini mencakup:

- React JS
- Laravel 12
- Php
- Pemrograman Rust
- Laragon
- Windows 11
- Vscode

2.2.1.4. Kebutuhan Perangkat Keras

Kebutuhan perangkat keras yang digunakan pada penelitian ini untuk membangun sistem dan menguji topik ini adalah sebuah laptop dengan spesifikasi sebagai berikut:

- Processor: 12thGenIntel(R)Core(TM)i712700H 2.30 GHz
- Ram: 8 GB 3200 MT/s
- Penyimpanan: 512 SSD

2.2.1.5. Kebutuhan sumber daya manusia

Gambar 3. Class diagram

Perangkat lunak dan perankat keras yang disebutkan diatas memerlukan sumber daya manusia yang mampu mengoperasikan sistem tersebut dengan baik dan benar.

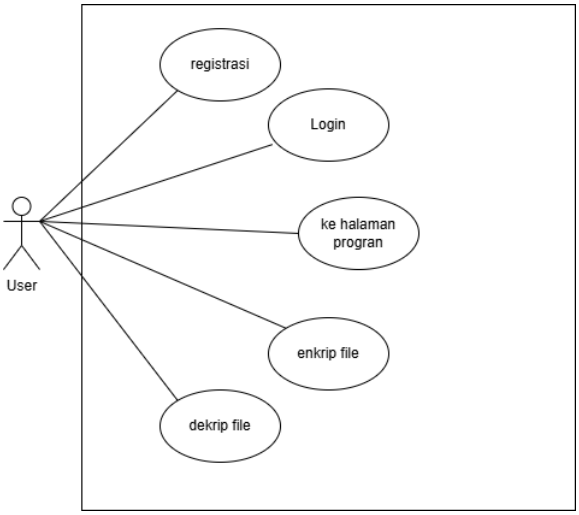
2.2.2. Desain

Pada desain disini, akan dibahas proses perancangan sistem menyeluruh yang mencakup Unified Modeling Language (UML) sebagai alat bantu visualisasi struktur dan alur sistem, serta desain review yang digunakan untuk mengevaluasi kesesuaian rancangan.

2.2.2.1. Unified Modeling Language

Dalam penelitian ini, uml digunakan untuk menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari sistema yang akan dibangun.

I. Use Case Diagram



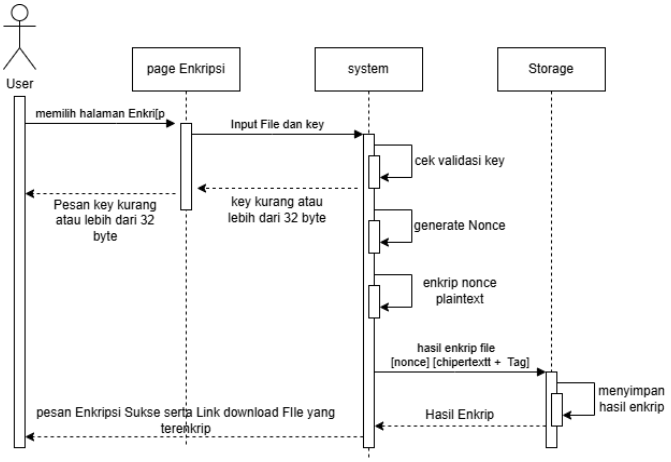
Gambar Error! No text of specified style in document.**.4. Use case diagram**

Pada gambar use case diagram diatas menjelaskan tentang alur dari client yang dimulai dari registrasi untuk membuat akun, setelah melakukan registrasi user bisa login untuk masuk ke dalam sistem yang dimana terdapat sebuah halaman enkripsi dan dekripsi.

E-ISSN 2723-5602
P-ISSN 2252-7095
https://journal.unnes.ac.id/journals/eduel/index

II. Class diagram

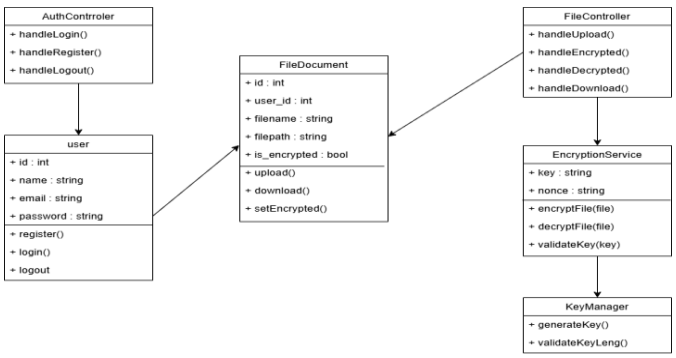
pada gambar class diagram diatas menggambarkan enam class utama dalam sistema, yaitu user, fileDocument, Encryption, KeyManager, AuthController, dan FileController. Diagram ini memperlihatkan interaksi antar class dalam mendukung proses autentikasi, manajemen file, serta enkripsi dan dekripsi menggunakan algoritma AES-GCM-SIV.

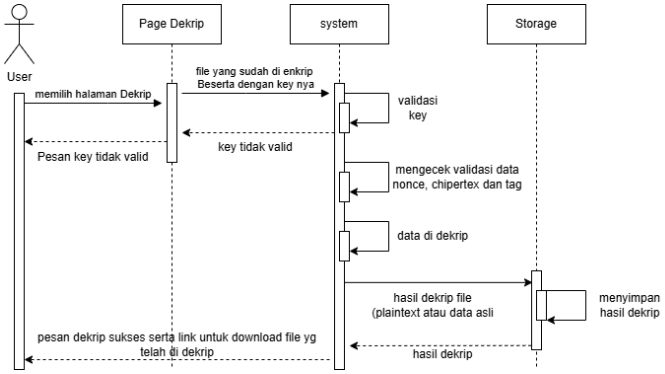


III. Sequence Diagram

Gambar 5. Sequence diagram Enkrip

Gambar di atas menunjukkan *sequence diagram* proses enkripsi. User mengunggah file dan memasukkan key enkripsi. Sistem memvalidasi panjang key agar sesuai 32 byte (256 bit). Jika tidak valid, sistem mengirimkan pesan kesalahan. Jika valid, sistem menghasilkan nonce acak 12 byte menggunakan CSPRNG (*OsRng*), lalu menginisialisasi objek *Aes256GcmSiv* untuk melakukan enkripsi. Hasil enkripsi berupa *ciphertext* dan *authentication tag* disimpan bersama nonce dalam satu file. File terenkripsi disimpan di storage, dan sistem memberi notifikasi sukses disertai tautan unduhan ke user.





Gambar 6. Sequence diagram Dekrip

Gambar di atas menggambarkan *sequence diagram* proses dekripsi. user mengunggah file terenkripsi beserta key ke sistem. Sistem terlebih dahulu memvalidasi key yang diberikan. Jika tidak valid, sistem mengirimkan pesan kesalahan. Jika valid, sistem memverifikasi struktur file, termasuk nonce, *ciphertext*, dan tag autentikasi. Setelah validasi berhasil, proses dekripsi menggunakan *Aes256GcmSiv* dilakukan untuk memperoleh kembali data asli (*plaintext*). Hasil dekripsi disimpan ke storage, dan sistem mengirimkan notifikasi sukses beserta tautan unduhan ke pengguna.

2.2.3. Impelementasi sistem

Pada tahap ini penulis akan menjelaskan proses implementasi sistem aplikasi web yang telah dibuat penulis berfungsi untuk melakukan enkripsi dan dekripsi file menggunakan algoritma *AES-GCM-SIV*. Implementasi dilakukan dengan menggabungkan berbagai teknologi, yatu back end berbasis Laravel 12 dan front end menggunakan inertia React, serta pemanfaatan binary dari bahasa pemrograman Rust yang di integrasikan ke dalam system untuk menjalankan proses enkripsi dan dekripsi.

III. Pembahasan

3.1. Testing

Pada tahap ini penulis melakukan pengujian terhadap system yang telah di implementasikan untuk memastikan bahwa seluruh fugsionalitas berjalan sesuai dengan apa yang dirancang. Pengujian dilakukan terhadap fitur enkripsi dan dekripsi file menggunakan algoritma AES-GCM-SIV yang telah diintegrasikan dalam aplikasi berbasis web. Pengujian ini bertujuan untuk mengetahui apakah sistem dapat mengenkripsi file dengan benar menggunakan kunci yang valid, serta memastikan bahwa file dapat di dekripsi kembali secara utuh menggunakan kunci yang sama.

Metode pengujian yang digunakan adalah white box testing dan black box testing.

3.1.1. White box testing

White box testing merupakan metode pengujian perangkat lunak yang dilakukan dengan mengetahui dan memahami struktur internal dari program. Pengujian ini bertujuan untuk menguji jalur logika dalam sistem secara

E-ISSN 2723-5602
P-ISSN 2252-7095
https://journal.unnes.ac.id/journals/eduel/index

menyeluruh, termasuk validasi, proses pengambilan keputusan, percabangan, dan hasil dari setiap proses.

Dalam penelitian ini, white-box testing diterapkan pada pengujian proses enkripsi dan dekripsi dari program ini.

1. Proses Enkripsi

Pengujian dimulai saat pengguna mengunggah file an memasukkan kunci enkripsi 32 karakter. Sistem Laravel memvalidasi panjang kunci, lalu menyimpan file sementara di storage/app/tmp. Selanjutnya, Laravel mengeksekusi binary Rust dengan mode *encrypt* menggunakan Process::run().

Binary Rust membaca file sebagai byte array, menghasilkan nonce acak 12 byte dengan OsRng, dan memverifikasi panjang kunci. Jika valid, proses enkripsi dijalankan menggunakan pustaka aes-gcm-siv, menghasilkan file terenkripsi berupa gabungan nonce dan ciphertext (nonce || ciphertext).

Hasil enkripsi disimpan di storage/app/encrypted, dengan waktu proses di bawah 2 detik untuk file <10MB. Sistem juga mencatat lama eksekusi dan hasilnya.

White box testing memastikan bahwa:

- Jalur validasi keu di laravel berjalan dengan benar
- Perintah command line Rust tersusun dan dijalankan dengan benar
- File terenkripsi disimpan dan berisi struktur: nonce+chipertext.

2. Proses Dekripsi

Dekripsi dilakukan saat pengguna mengunggah file .enc dan memasukkan kembali kunci enkripsi. Sistem Laravel mengekstrak informasi nama file dan menyusun nama file hasil dekripsi. Selanjutnya, Laravel menjalankan binary Rust dengan mode *decrypt*.

Program Rust membaca file, memisahkan 12 byte pertama sebagai nonce dan sisanya sebagai ciphertext. Nonce digunakan untuk mendekripsi ciphertext menggunakan pustaka aes-gcm-siv. Jika kunci salah atau data rusak, proses dihentikan dengan pesan *authentication failed.*

Laravel menangkap status proses dari output binary. Jika berhasil, file plaintext disimpan dan pengguna menerima tautan unduhan. Jika gagal, sistem menampilkan pesan kesalahan dan mencatat log untuk debugging.

White box testing memastikan bahwa:

- Format nama file terenkripsi diproses dengan benar
- Jalur validasi dan pembentukan nama file hasil dekripsi berjalan
- Proses dekripsi bekerja hanya jika nonce dan key cocok

3.1.2. Black box testing

Metode blackbox testing merupakan metode pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan Kumpulan kondisi input dan melakukan pengujian pada fungsional program. Proses blackbox testing dengan cara mencoba program yang telah dibuat dengan mencoba memasukkan data pada halaman enkrip dan dekrip nya. Pengujian ini diperkukan untuk mengetahui program tersebut berjalan sesuai dengan yang dibutuhkan penelitian ini. Berikut daftar table uji dan format dokumentasi sebagai berikut:

Tabel 1. uji fitur enkripsi

Kode	Nama Pengujian	Skenario	Hasil yang di harapkan	Hasil Pengujian
A01	Unggah dan Enkripsi File	User mengunggah file dengan key 32 byte	File berhasil terenkripsi	Sesuai
A02	Key tidak valid	User mengunggah file dengan key kurang atau lebih dari 32 byte	Muncul alert : Key harus 32 byte	Sesuai
A03	File tidak dipilih	User menekan tombol enrkrip tanpa memilih file	Muncul alert: pilih file:	Sesuai

Terlihat pada tabel diatas beberapa skenario yang dilkaukan untuk menguji keberhasilan system utuk merespon.

Tabel 2. uji fitur dekripsi

Kode	Nama Pengujian	Skenario	Hasil yang diharapkan	Hasil Pengujian
B01	Dekripsi file	User memilih file yang sudah di enkrip dan key yang valid	File berhasil di dekrip	Sesuai
B02	Key salah	User memasukkan key yang tidak valid dengan file	Muncul alert: key tidak valid	Sesuai
B03	Format file salah	User memasukkan file yang salah	Muncul alert: file tidak sesuai	Sesuai

Terlihat pada tabel diatas beberapa skenario yang dilakukan untuk menguji keberhasilan system untuk merespon.

3.1.3. Perbandingan algoritma

Pada tahapan ini penulis akan melakukan perbandingan algoritma yang digunakan yaitu AES-GCM-SIV dan AES-GCM. Perbandingan ini mencakup aspek performa enkripsi dan dekripsi dan keamanan terhadap serangan nonce reuse. Dengan

E-ISSN 2723-5602
P-ISSN 2252-7095
https://journal.unnes.ac.id/journals/eduel/index

membandingkan kedua algoritma tersebut, diharapkan dapat diperoleh pemahaman yang lebih mendalam mengenai kelebihan dan kekurangan masing-masing algoritma dalam konteks implementasi pada yang dikembangkan.

3.1.3.1. Kecepatan proses

Pada tahapan ini penulis melakukan perbandingan dengan cara membandingkan kecepatan proses enkripsi dan dekripsi dari dua algoritma yaitu AES-GCM-SIV dan AES-GCM, serta file yang digunakan untuk perbandingan adalah document dari dinas pendidikan dan kebudayaan Lombok barat. Berikut hasil dari perbandingannya:

Tabel 3. Perbandingan kecepatan proses enkrip

No	Nama File	Hasil enkrip AES-GCM-SIV	Hasil enkrip AES-GCM
1	KAK DAK 2024 BIDANG SMP.docs (17 hal)	0.309	0.212
2	KAK DAK 2024 BIDANG SD.docs (19 hal)	0.249	0.075
3	Tabel kegiatan.docs (3hal)	0.256	0.050
4	Buku_Profil_Pendidikan_Tahun_2022_Dikbud_Lobar_GAB.xlsx	0.270	0.083
5	RAPOR-KAB-LOMBOK-BARAT-2024.xlsx	0.262	0.134
6	Data Sekolah Prov. Nusa Tenggara Barat – Dapodikdasmen.xlsx	0.276	0.011
7	Gambar dan RAB Lab Komputer SMPN 1 Lingsar.pdf	2.058	0.067
8	Identifikasi-Kerusakan-SMPN 1 Lingsar_Rehab Toilet.pdf	0.268	0.031
9	Laporan_Monev_Pengaduan_2022.pdf	0.329	0.129

Tabel 4. Perbandingan kecepetan proses dekrip

No	Nama File	Hasil dekrip AES-GCM-SIV	Hasil dekrip AES-GCM
1	KAK DAK 2024 BIDANG SMP.docs (17 hal)	0.309	0.101
2	KAK DAK 2024 BIDANG SD.docs (19 hal)	0.249	0.062
3	Tabel kegiatan.docs (3hal)	0.256	0.017
4	Buku_Profil_Pendidikan_Tahun_2022_Dikbud_Lobar_GAB.xlsx	0.270	0.048
5	RAPOR-KAB-LOMBOK-BARAT-2024.xlsx	0.262	0.069
6	Data Sekolah Prov. Nusa Tenggara Barat – Dapodikdasmen.xlsx	0.276	0.011
7	Gambar dan RAB Lab Komputer SMPN 1 Lingsar.pdf	2.058	0.067
8	Identifikasi-Kerusakan-SMPN 1 Lingsar_Rehab Toilet.pdf	0.268	0.023
9	Laporan_Monev_Pengaduan_2022.pdf	0.329	0.071

Berdasarkan hasil pengujian terhadap beberapa file dengan ukuran dan jenis yang berbeda, diperoleh data waktu proses enkripsi dan dekripsi menggunakan algoritma AES-GCM-SIV dan AES-GCM. Dari hasil tersebut, terlihat bahwa algoritma AES-GCM secara konsisten memiliki waktu proses yang lebih cepat dibandingkan AES-GCM-SIV, baik pada proses enkripsi maupun dekripsi. Sebagai contoh, pada file *"KAK DAK 2024 BIDANG SMP.docs (17 hal)"*, waktu enkripsi menggunakan AES-GCM-SIV adalah 0,309 detik, sedangkan AES-GCM hanya membutuhkan 0,212 detik. Hal serupa juga terjadi pada proses dekripsi, di mana AES-GCM-SIV membutuhkan waktu 0,309 detik sementara AES-GCM hanya 0,101 detik. Membuktikan bahwa algoritma AES-GCM-SIV proses nya lebih lama daripada algoritma AES-GCM.

3.1.3.2. Nonce Reuse

Pada tahap ini, penulis membandingkan ketahana kedua algoritma terhadap nonce reuse (penggunaan nonce berulang-ulang). Nonce reuse yang dimaksud adalah kondisi dimana nilai nonce yakni sebuah kombinasi acak dari huruf dan angka yang digunakan dalam proses enkripsi dan dipakai berulang kali. Pada tahapan ini penulis menggunakan library cryprography dari bahasa pyton untuk membandingkan ketahanan dari *nonce reuse* dari dua algoritma tersebut. Berikut source code perbandingan nya:

E-ISSN 2723-5602
P-ISSN 2252-7095
https://journal.unnes.ac.id/journals/eduel/index

```
aesgcm = AESGCM(key)
ciphertext1 = aesgcm.encrypt(nonce, plaintext1, None)
ciphertext2 = aesgcm.encrypt(nonce, plaintext2, None)
# XOR dua ciphertext → hasil = p1 ⊕ p2 (sepanjang plaintext)
min_len = min(len(plaintext1), len(plaintext2))
xor_c1_c2 = xor_bytes(ciphertext1[:min_len],
ciphertext2[:min_len])
# Recovery: Dapatkan bagian dari p2 dari p1
recovered_plaintext2 = xor_bytes(plaintext1[:min_len],
xor_c1_c2)
# ----------------------------------
# AES-GCM-SIV:
# ----------------------------------
aesgcmsiv = AESGCMSIV(key)
ciphertext1_siv = aesgcmsiv.encrypt(nonce, plaintext1, None)
ciphertext2_siv = aesgcmsiv.encrypt(nonce, plaintext2, None)
xor_siv = xor_bytes(ciphertext1_siv, ciphertext2_siv)
```

Gambar 7. source code perbandingan

Parameter yang dipakai ialah:

Key : ‘12345678901234567890qwertyuiopas’

Nonce : ‘123456789012’

Plaintext : P1 = ‘password: admin’

P2 = ‘password:root’

```
===== AES-GCM  =====
Ciphertext1: 2087161e066580d24b69c247a7567e7036612a92c30ac21bdcfb12badc8a
Ciphertext2: 2087161e066580d24b69d14ca54beda546b9968176c99efa971a361dc953
c1 XOR c2 : 0000000000000000000000130b021d
pembuktian untuk tau isi dari p2: Password: root
```

Gambar 8. Hasil output AES-GCM

Pada gambar diatas terlihat hasil output menggunakan algoritma AES-GCM, dimana terlihat hasil chipertext pada byte awal terlihat sama dikarenakan AES-GCM memiliki sensitivitas terhadap nonce reuse. Pada gambar diatas nonce reuse menyebabkan chipertext yang dihasilkan memilikan bagian yang sama pada byte awal nya, dan dengan melakukan operasi XOR terhadap chipertext yang di enkripsi menggunakan nonce yang sama, maka informasi dari plaintext dapat bocor.

```
===== AES-GCM-SIV  =====
Ciphertext1: 160244c17cabea06c756487b8a21f903596add7bdda2630efb3b67e40c1a
Ciphertext2: 4acc98dd9bdd02bf80423beb9660c8625e0f479a3df2fc9fd5a72f6a271b
Ciphertext1 == Ciphertext2? False
c1 XOR c2: 5ccedc1ce776e8b9471473901c41316107659ae1e0509f912e9c488e2b01
```

Gambar 9. Hasil output AES-GCM-SIV

Pada gambar 4.2 terlihat hasil output dari AES-GCM_SIV, algoritma ini dirancang khusus untuk mengatasi reuse nonce. Meskipun pada perbandingan ini digunakan nonce yang sama, hasil ciphertext tetap berbeda dan tidak menunjukkan pola apapun. XOR antara dua ciphertext AES-GCM-SIV pun tidak memberikan informasi yang dapat dieksploitasi, karena algoritma ini menggunakan konstruksi Synthetic IV (SIV) yang menggabungkan proses autentikasi sebelum enkripsi. Makan dari itu bahwa AES-GCM-SIV lebih terbukti bisa mengatasi nonce reuse daripada AES-GCM.

3.1.3.3. Hasil dari perbandingan

Tabel 5. Hasil perbandingan dari dua algoritma

Algoritma	Kecepatan proses	Nonce Reuse
AES-GCM	Lebih cepat	Rentan terhadap nonce reuse
AES-GCM-SIV	Lebih lambat sedikit	Terbukti bisa mengatasi nonce reuse

Terlihat pada tabel tersebut hasil perbandingan dari algoritma AES-GCM dengan algoritma AES-GCM-SIV, bahwa algoritma AES-GCM-SIV lebih lambat untuk proses enkripsi dan dekripsi, akan tetapi lebih aman digunakan daripada algoritma AES-GCM.

IV. PENUTUP

Berdasarkan hasil penelitian yang telah dilakukan mengenai pengamanan file pada aplikasi web menggunakan algoritma AES-GCM-SIV, dapat disimpulkan bahwa algoritma ini mampu memberikan tingkat keamanan data yang tinggi, khususnya dalam proses enkripsi dan dekripsi file dokumen. Salah satu keunggulan utama AES-GCM-SIV adalah kemampuannya mencegah kebocoran data meskipun terjadi reuse nonce, menjadikannya pilihan yang andal dalam skenario penggunaan ulang nonce yang tidak dapat dihindari. Implementasi algoritma ini menggunakan bahasa pemrograman Rust terbukti memberikan performa yang optimal karena efisiensi tinggi dan manajemen memori yang aman yang dimiliki Rust.

Selain itu, integrasi algoritma ke dalam aplikasi web berbasis Laravel 12 untuk sisi back-end dan ReactJS untuk sisi front-end berhasil dilakukan dengan baik. Hal ini memungkinkan pengguna untuk mengunggah, mengenkripsi, mendekripsi, dan mengunduh file dengan mudah melalui antarmuka yang user-friendly. Hasil pengujian sistem menunjukkan bahwa algoritma AES-GCM-SIV tidak hanya mampu menjaga integritas dan kerahasiaan data, tetapi juga menawarkan kecepatan yang lebih baik dalam proses enkripsi dan dekripsi dibandingkan dengan algoritma lain yang tidak memiliki fitur nonce misuse resistance. Aspek manajemen

E-ISSN 2723-5602
P-ISSN 2252-7095
https://journal.unnes.ac.id/journals/eduel/index

kunci juga menjadi perhatian penting dalam penelitian ini, di mana pendekatan validasi panjang kunci (32 byte) diterapkan guna menjamin keamanan proses enkripsi dan dekripsi secara keseluruhan.

Adapun beberapa saran yang dapat dipertimbangkan untuk pengembangan lebih lanjut dari penelitian mengenai algoritma AES-GCM-SIV dalam pengamanan file pada aplikasi web, antara lain sebagai berikut.

Pertama, pengembangan sistem ke lingkungan produksi sebaiknya dilakukan dengan mempertimbangkan aspek skalabilitas, penerapan autentikasi multi-level, serta integrasi dengan layanan cloud yang aman seperti AWS S3 atau Google Cloud Storage untuk mendukung penyimpanan data terenkripsi secara andal.

Kedua, demi meningkatkan kenyamanan dan keamanan pengguna, disarankan untuk menambahkan fitur manajemen kunci secara otomatis (key vault), sehingga pengguna tidak perlu mengelola kunci enkripsi secara manual.

Ketiga, penelitian lanjutan dapat mengkaji dan membandingkan algoritma kriptografi lain yang juga mendukung AEAD (Authenticated Encryption with Associated Data), seperti ChaCha20-Poly1305, untuk memperoleh wawasan lebih mendalam mengenai perbedaan performa dan tingkat keamanannya dibandingkan dengan AES-GCM-SIV.

Terakhir, penting juga untuk meningkatkan pengamanan pada sisi transportasi data, misalnya dengan menerapkan protokol HTTPS dan WebSocket Secure, agar komunikasi antara antarmuka front-end dan back-end tetap terenkripsi dan aman selama proses pengiriman file.

DAFTAR PUSTAKA

Ahmad Sitorus, F., Budi Nugroho, N., Fatimah Sari Sitorus Pane, U., Studi Mahasiswa, P., Triguna Dharma, S., & Studi Dosen Pembimbing, P. (2020). Implementasi Algoritma Advanced Encryption Standard (AES) 128 Bit Untuk Keamanan Data Transaksi Penjualan Pada PT. MITSUBISHI ELECTRIC INDONESIA. In *Jurnal CyberTech: Vol. x. No.x*. https://ojs.trigunadharma.ac.id/

Albertini, A., Duong, T., Research, G., Gueron, S., Kölbl, S., Luykx, A., & Schmieg, S. (n.d.). *How to Abuse and Fix Authenticated Encryption Without Key Commitment How to Abuse and Fix Authenticated Encryption Without Key Commitment **. https://www.usenix.org/conference/usenixsecurity22/presentation/albertini

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

Beyne, T., Chen, Y. L., & Verbauwhede, M. (n.d.). *A Robust Variant of ChaCha20-Poly1305*.

Choi, W., Lee, B., Lee, J., & Lee, Y. (n.d.). *Toward a Fully Secure Authenticated Encryption Scheme From a Pseudorandom Permutation (Full Version)*.

Chung, W., Hwang, S., Kim, S., Lee, B., & Lee, J. (2025). *Making GCM Great Again: Toward Full Security and Longer Nonces* (pp. 33–61). https://doi.org/10.1007/978-3-031-91107-1_2

Destriani Firmansyah Putri Universitas Pembangunan Nasional Veteran Jakarta Jalan Fatmawati, D. R., Labu, P., Selatan, J., Jakarta, D., & Helmi Fahrozi Universitas Pembangunan Nasional Veteran Jakarta Jalan Fatmawati, M. R. (2021). UPAYA PENCEGAHAN KEBOCORAN DATA KONSUMEN MELALUI PENGESAHAN RUU PERLINDUNGAN DATA PRIBADI (STUDI KASUS E-COMMERCE BHINNEKA.COM). In *Borneo Law Review* (Vol. 5, Issue 1). https://tirto.id/jumlah-pelanggan-e-commerce-tercatat-meningkat-383-selama-

Febrianto, R., & Waluyo, S. (2023). *IMPLEMENTASI ALGORITME KRIPTOGRAFI ADVANCED ENCRYPTION STANDARD (AES-256) UNTUK MENGAMANKAN DATABASE PENILAIAN KARYAWAN PADA KJPP NDR* (Vol. 20, Issue 1).

Harun Alfirdaus, M., Tahir, M., Enno Dewanti, N., Ardianto, R., Nur Azurah, N., Firman Cahyono, N., & Informatika, P. (2023). Perancangan Aplikasi Enkripsi Deskripsi Mengunakan Metode Caesar Chiper Berbasis Web. In *JTMEI)* (Vol. 2, Issue 2).

Karnaukhov, A., Tymoshchuk, V., & Orlovska, A. (2024, June 14). *USE OF AUTHENTICATED AES-GCM ENCRYPTION IN VPN*. https://doi.org/10.62731/mcnd-14.06.2024.004

Kim, K., Choi, S., Kwon, H., Kim, H., Liu, Z., & Seo, H. (2020). PAGE-Practical AES-GCM encryption for low-end microcontrollers. *Applied Sciences (Switzerland)*, *10*(9). https://doi.org/10.3390/app10093131

Kusuma, A. C., & Rahmani, A. D. (n.d.). *Analisis Yuridis Kebocoran Data Pada Sistem Perbankan Di Indonesia (Studi Kasus Kebocoran Data Pada Bank Indonesia)*. www.bi.go.id.

Pinuyut, C. A., Utami, E., & Muhammad, A. H. (n.d.). ANALISIS KINERJA ALGORITMA ADVANCED ENCRYPTION STANDARD (AES) TERMODIFIKASI DALAM ENRKIPSI DAN DEKRIPSI DATA (PERFORMANCE ANALYSIS OF MODIFIED ADVANCED ENCRYPTION STANDART (AES) ALGORITHM FOR ENCRYPTING AND DECRYPTING DATA). In *Desember* (Vol. 5, Issue 2).

Pirzada, S. J. H., Murtaza, A., Xu, T., & Jianwei, L. (2020). Architectural optimization of parallel authenticated encryption algorithm for satellite application. *IEEE Access*, *8*, 48543–48556. https://doi.org/10.1109/ACCESS.2020.2978665

Sari, M., Kriptografi Keamanan, A., Dwi Purnomo, H., Sembiring, I., Sistem Informasi, M., Teknologi Informasi, F., Kristen Satya Wacana, U., & Notohamidjojo, J. O. (2022). Review : Algoritma Kriptografi Sistem Keamanan SMS di Android.

E-ISSN 2723-5602
P-ISSN 2252-7095
https://journal.unnes.ac.id/journals/eduel/index

JOURNAL OF INFORMATION TECHNOLOGY, *2*(1), 3.

Suranta, A. I., Virgian, D., & Sakti, S. Y. (2022). Penerapan Algoritma AES (Advance Encryption Standart) 128 untuk Enkripsi Dokumen di PT. Gunung Geulis Elok Abadi. *SKANIKA: Sistem Komputer Dan Teknik Informatika*, *5*(1), 1–10.

