# Perancangan Sistem Berbagi File Aman Menggunakan Enkripsi dan Kontrol Akses pada Arsitektur Microservices Berbasis Web

Samuel Rinaldy*[1], Shofwan Adinaufal [2], Bintang Fajar Mustika Aji [3], Muhammad Syafiq Shidiq [4], Maulana Usamah Rochsyad [5], Agustina Srirahayu [6]

[1]Teknik Informatika/Komputer
Universitas Duta Bangsa Surakarta
Samuel Rinaldy
[1]*240103202*@mhs.udb.ac.id

[2]Teknik Informatika/Komputer
Universitas Duta Bangsa Surakarta
Shofwan Adinaufal
[2]240103203@mhs.udb.ac.id

[3]Teknik Informatika/Komputer
Universitas Duta Bangsa Surakarta
Bintang Fajar Mustika Aji
[3]240103186@mhs.udb.ac.id

[4]Teknik Informatika/Komputer
Universitas Duta Bangsa Surakarta
Muhammad Syafiq Shidiq
[4]240103197@mhs.udb.ac.id

[5]Teknik Informatika/Komputer
Universitas Duta Bangsa Surakarta
Maulana Usamah Rochsyad
[5]240103195@mhs.udb.ac.id

[5]Teknik Informatika/Komputer
Universitas Duta Bangsa Surakarta
Agustina Srirahayu
[6]agustina@udb.ac.id

***Abstrak*— Perkembangan teknologi informasi meningkatkan kebutuhan sistem berbagi file berbasis web yang cepat, mudah diakses, dan aman. Namun, proses pertukaran file digital masih memiliki risiko seperti akses tidak sah, kebocoran data, pencurian informasi, dan penyalahgunaan file. Penelitian berjudul "Perancangan Sistem Berbagi File Aman Menggunakan Enkripsi dan Kontrol Akses pada Arsitektur Microservices Berbasis Web" ini bertujuan merancang sistem berbagi file yang mampu menjaga keamanan data melalui penerapan enkripsi, autentikasi pengguna, serta kontrol akses berbasis peran. Sistem dirancang menggunakan arsitektur microservices agar layanan seperti autentikasi, manajemen file, enkripsi, kontrol akses, dan pencatatan aktivitas dapat berjalan secara terpisah namun tetap terintegrasi. Hasil perancangan ini diharapkan dapat menjadi solusi berbagi file yang aman, terstruktur, fleksibel, dan mudah dikembangkan untuk lingkungan organisasi, pendidikan, maupun perusahaan.**
***Kata kunci***— berbagi file, enkripsi, kontrol akses, microservices, keamanan data, web.

***Abstract*— The development of information technology increases the need for a fast, accessible, and secure web-based file sharing system. However, the process of exchanging digital files still carries risks such as unauthorized access, data leakage, information theft, and file misuse. This research, entitled "Designing a Secure File Sharing System Using Encryption and Access Control on a Web-Based Microservices Architecture," aims to design a file sharing system capable of maintaining data security through the implementation of encryption, user authentication, and role-based access control. The system is designed using a microservices architecture so that services such as authentication, file management, encryption, access control, and activity logging can run separately but remain integrated. The results of this design are expected to be a secure, structured, flexible, and easily developed file sharing solution for organizational, educational, and corporate environments.**
***Keywords***— file sharing, encryption, access control, microservices, data security, web.

## I. PENDAHULUAN

Sistem berbagi file berbasis web semakin banyak digunakan dalam lingkungan pendidikan, pemerintahan, organisasi, dan perusahaan. Sistem ini memudahkan pengguna untuk mengunggah, menyimpan, mengelola, dan membagikan dokumen melalui jaringan internet. Kemudahan tersebut mendukung kolaborasi digital, tetapi juga menimbulkan risiko keamanan, seperti akses tidak sah, pencurian informasi, perubahan data, dan kebocoran file sensitif [8], [9], [10].

Risiko keamanan pada sistem berbagi file menjadi semakin penting karena file digital sering berisi data pribadi, dokumen organisasi, informasi akademik, dan data perusahaan. Apabila sistem penyimpanan atau jalur distribusi file mengalami gangguan keamanan, file yang tidak dilindungi dapat diakses dan disalahgunakan oleh pihak yang tidak berwenang. Oleh karena itu, sistem berbagi file perlu dirancang dengan mekanisme keamanan yang mampu menjaga kerahasiaan data, membatasi hak akses pengguna, dan mencatat aktivitas sistem secara terstruktur [8], [10].

Salah satu mekanisme utama untuk menjaga kerahasiaan file adalah enkripsi. Enkripsi bekerja dengan mengubah file asli atau plaintext menjadi bentuk tersandi atau ciphertext sehingga isi file tidak dapat dibaca tanpa proses dekripsi dan kunci yang sesuai. Advanced Encryption Standard atau

AES merupakan salah satu algoritma enkripsi simetris yang banyak digunakan karena memiliki tingkat keamanan yang baik dan efisiensi komputasi yang sesuai untuk pemrosesan data digital. Dalam sistem berbagi file, AES dapat diterapkan untuk melindungi file sebelum disimpan atau dibagikan kepada pengguna lain [1], [5], [8].

Selain enkripsi, sistem berbagi file juga memerlukan autentikasi dan kontrol akses. Autentikasi digunakan untuk memastikan identitas pengguna yang masuk ke dalam sistem. Kontrol akses digunakan untuk membatasi tindakan pengguna terhadap file tertentu, seperti mengunggah, mengunduh, menghapus, atau mengakses menu administrasi. Role-Based Access Control atau RBAC dapat digunakan untuk mengatur hak akses berdasarkan peran pengguna. JSON Web Token (JWT) dapat digunakan sebagai token autentikasi yang membawa informasi identitas dan hak akses pengguna sehingga setiap permintaan ke API Gateway dan layanan sistem dapat diverifikasi secara aman [6].

Arsitektur sistem juga memiliki peran penting dalam mendukung pengelolaan layanan. Sistem berbagi file yang dibangun secara monolitik memiliki keterbatasan karena seluruh fungsi berjalan dalam satu kesatuan aplikasi. Apabila salah satu fungsi mengalami gangguan, bagian lain dari sistem dapat ikut terdampak. Untuk mengurangi keterbatasan tersebut, penelitian ini menggunakan arsitektur microservices. Arsitektur ini memisahkan fungsi utama sistem ke dalam beberapa layanan, seperti layanan autentikasi, layanan file, layanan enkripsi, layanan kontrol akses, dan layanan pencatatan aktivitas [6], [7].

Penggunaan microservices dalam penelitian ini ditujukan untuk membuat sistem lebih terstruktur, fleksibel, dan mudah dikembangkan. Setiap layanan menjalankan fungsi tertentu secara mandiri dan berkomunikasi melalui API Gateway. Dengan pemisahan tersebut, proses autentikasi, pengelolaan file, enkripsi, dan pencatatan aktivitas dapat dikelola secara lebih jelas [6], [7]. Namun, penelitian ini tidak mengklaim peningkatan high availability secara empiris karena pengujian failover, load testing, throughput, dan simulasi kegagalan layanan belum menjadi fokus utama penelitian.

Penelitian sebelumnya telah membahas penggunaan AES untuk pengamanan dokumen dan pengiriman file digital [1], [5], penerapan autentikasi menggunakan AES maupun JWT [2], [6], penggunaan RBAC untuk pengaturan hak akses [4], dan penerapan microservices pada aplikasi berbasis web [6], [7]. Akan tetapi, sebagian penelitian masih membahas komponen tersebut secara terpisah. Oleh karena itu, penelitian ini berfokus pada perancangan dan implementasi sistem berbagi file aman yang mengintegrasikan enkripsi AES-256, autentikasi JWT, kontrol akses RBAC, API Gateway, pembatasan akses berbagi, activity log, dan arsitektur microservices dalam satu sistem berbasis web.

Kontribusi penelitian ini bukan pada pengembangan algoritma keamanan baru, melainkan pada integrasi mekanisme keamanan yang terdiri atas enkripsi AES-256, autentikasi JWT, kontrol akses RBAC, API Gateway, activity log, dan arsitektur microservices ke dalam satu sistem berbagi file berbasis web yang terstruktur. Integrasi tersebut memungkinkan proses penyimpanan, pengelolaan, dan distribusi file dilakukan dengan mekanisme keamanan yang saling terhubung sehingga meningkatkan perlindungan terhadap akses tidak sah.

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem berbagi file aman berbasis web dengan menggunakan enkripsi AES-256, autentikasi JWT, kontrol akses RBAC, API Gateway, dan arsitektur microservices. Sistem ini diharapkan dapat meningkatkan perlindungan terhadap file digital, mencegah akses tidak sah, serta mendukung pengelolaan file yang lebih aman, fleksibel, dan terstruktur pada lingkungan organisasi, pendidikan, maupun Perusahaan.

## II. METODOLOGI PENELITIAN

### 1. Identifikasi Masalah

Karena ancaman penyusupan pihak ketiga di dalam jaringan, infrastruktur penyimpanan awan konvensional dan saluran komunikasi data saat ini menghadapi tantangan besar. Penyusupan pihak ketiga ini dapat menyebabkan kebocoran data berharga, yang dapat terjadi selama proses berbagi file [8], [10]. Sebaliknya, karena belum ada mekanisme penyandian dokumen yang lengkap untuk melindungi berkas digital yang diunggah, privasi media penyimpanan bersama yang digunakan oleh banyak orang sering kali tidak memadai [5], [8]. Karena ancaman penyusupan pihak ketiga di dalam jaringan, infrastruktur penyimpanan awan konvensional dan saluran komunikasi data saat ini menghadapi tantangan besar. Penyusupan pihak ketiga ini dapat menyebabkan kebocoran data berharga, yang dapat terjadi selama proses berbagi file [8], [10]. Sebaliknya, karena belum ada mekanisme penyandian dokumen yang lengkap untuk melindungi berkas digital yang diunggah, privasi media penyimpanan bersama yang digunakan oleh banyak orang sering kali tidak memadai [5], [8].

## 2. Studi Literatur

Studi keamanan infrastruktur awan oleh Purba et al. meyakinkan bahwa penerapan perlindungan berbasis End-to-End Encryption (E2EE) sangat penting untuk mencegah kemungkinan kebocoran data karena memastikan bahwa berkas tetap aman dari perubahan luar selama proses pengiriman [10]. Selain itu, penelitian Patel dan Trivedi tentang sistem berbagi file menunjukkan bahwa algoritma Advanced Encryption Standard (AES) dengan kontrol delegasi hak akses yang ketat dapat memaksimalkan perlindungan dokumen dengan berbagai ukuran [8]. Selain itu, analisis arsitektur terdistribusi yang dilakukan oleh Irawan dan Fauzi menunjukkan bahwa sistem berbasis microservices web yang memungkinkan komunikasi antar-layanan independen menggunakan kombinasi metode pengendalian akses berbasis peran (RBAC) dan token web JSON (JWT) pada komponen API Gateway berhasil [7]. Namun, panduan teoretis Munir memberikan dasar matematis dan prosedur teknis yang tepat untuk mengubah teks biasa (plaintext) menjadi kode acak (ciphertext). Ini dilakukan untuk menjaga aspek kerahasiaan dan integritas dokumen pada aplikasi digital [3]. Selain itu, penelitian sistem keamanan oleh Nasrullah et al. mengusulkan solusi praktis dengan menggabungkan algoritma simetris AES dan algoritma asimetris RSA, yang keduanya dapat digunakan untuk enkripsi isi berkas karena kecepatan pemrosesan datanya [5].

## 3. 3.Analisis Kebutuhan Sistem

Analisis kebutuhan sistem menunjukkan bahwa dalam merancang sistem berbagi file yang aman, platform harus menyediakan layanan unggah dan unduh file. Dengan demikian, dokumen yang diterima akan langsung dialihkan ke modul enkripsi, yang berdasarkan aturan matematika kriptografi kontemporer akan mengubah kode tersandi sebelum disimpan ke media penyimpanan [3]. Sebelum sistem memungkinkan permintaan pengunduhan berkas, proses validasi identitas dan pembatasan hak akses pengguna wajib diawasi secara ketat melalui metode RBAC yang dilekatkan pada token digital JWT di pintu gerbang utama, juga dikenal sebagai API Gateway [7]. Selain itu, infrastruktur platform web harus sangat responsif terhadap peningkatan beban kerja karena sistem dibagi menjadi bagian-bagian layanan mikro, atau microservices, yang mandiri dan terisolasi. Pemrosesan perlindungan file harus berjalan secara responsif melalui integrasi mekanisme hibrida untuk meningkatkan efisiensi komputasi. Ini dilakukan agar pengamanan berkas berukuran besar tidak mengganggu kinerja server web saat melakukannya [5]. Secara keseluruhan, seluruh arsitektur sistem ini harus menerapkan standar perlindungan tingkat tinggi untuk penyimpanan data untuk meminimalkan kerentanan penyusupan jaringan dan memastikan bahwa data yang disimpan dan disimpan aman. yang ditransmisikan [8], [10].

## 4. Perancangan Sistem

Pembagian sistem menjadi klaster layanan indep enden seperti layanan autentikasi, layanan file, dan l ayanan kriptografi mewujudkan desain arsitektur. A PI Gateway berbasis web mengontrol lalu lintas per mintaan secara keseluruhan [7]. Struktur otorisasi arsitektur ini dibangun dengan me nggunakan hierarki hak akses pengguna berbasis pe ran (RBAC). Hierarki ini diintegrasikan ke dalam to ken keamanan JWT nirkondisi (stateless) sebagai sy arat utama untuk mengakses layanan penyimpanan file [7]. Sementara itu, perancangan layanan keamanan (Cryptographic Service) berkonsentrasi pada pembuatan mekanisme yang dapat mengubah data teks biasa menjadi teks tersandi dengan tepat, menggunakan prinsip algoritma penyandian kontemporer [3]. Subsistem ini menggunakan pendekatan hibrida untuk pemrosesan berkas; algoritma simetris AES mengunci konten dokumen secara instan, dan proses pembungkusan kunci, atau pembungkusan kunci, menjamin kunci publik RSA penerima file [5].

## 5. Implementasi Sistem

Penerjemahan arsitektur cetak biru dilakukan dengan menggabungkan basis data dan repositori kode ke dalam layanan mikro berbasis web yang saling terhubung menggunakan protokol komunikasi aman melalui API Gateway [7]. Untuk menyelesaikan pemrograman komponen otorisasi, fungsi verifikasi token JWT dipasang pada server untuk memeriksa status peran (RBAC) pengguna sebelum memberikan izin akses terhadap file yang diminta. [7]. Pada tahap berikutnya, proses pengamanan dokumen diwujudkan secara nyata pada layanan kriptografi dengan menggunakan fungsi matematika penyandian untuk mengaburkan informasi asli menjadi bentuk berkas tersandi sebelum masuk ke direktori penyimpanan. [3]. Penggunaan komputasi algoritma hibrida menyelesaikan logika pemrograman yang diperlukan untuk mengamanan dokumen skala besar tersebut. Algoritma hibrida menggunakan kunci AES untuk mengenkripsi isi file dengan cepat dan enkripsi asimetris RSA untuk melindungi sirkulasi kunci [5].

## 6. 6. Pengujian Sistem

Untuk menguji kinerja arsitektur, skenario uji coba manipulasi token—juga dikenal sebagai uji coba bypass otorisasi—digunakan. Ini mengonfirmasi ketangguhan validasi token JWT dan kemanjuran hak akses RBAC dalam menghalangi pengguna ilegal dari arsitektur terdistribusi [7]. Pengujian modul keamanan juga berfokus pada memastikan bahwa hasil translasi file menjadi ciphertext yang tidak dapat dibaca atau diubah tanpa menggunakan sepasang kunci dekripsi yang sah [3]. Untuk mengevaluasi efisiensi sistem pada jaringan web, tahap pengujian diakhiri dengan menganalisis catatan waktu respons operasi enkripsi dan dekripsi hibrida pada berbagai sampel ukuran dokumen [5].

## 7. Evaluasi dan Analisis Hasil

Hasil evaluasi menunjukkan bahwa penerapan kontrol akses berbasis RBAC dan otentikasi JWT pada arsitektur microservices berhasil membatasi ruang gerak pengguna dan mencegah akses tidak sah ke gerbang utama secara keseluruhan [7]. Dengan menggunakan pengujian ketahanan informasi, analisis berkas hasil enkripsi memastikan bahwa data yang disimpan telah dilindungi sepenuhnya. Ini menurunkan risiko kebocoran dokumen, atau kebocoran data, yang disebabkan oleh intersepsi jaringan [8], [10]. Terakhir, evaluasi kinerja sistem menunjukkan bahwa metode kriptografi hibrida menggabungkan kecepatan algoritma AES dengan keandalan distribusi kunci algoritma RSA, yang menghasilkan waktu komputasi yang sangat baik untuk pemrosesan file web berskala besar [5].

## III. HASIL DAN PEMBAHASAN

Penelitian ini telah menciptakan dan menerapkan sistem berbagi file berbasis web yang mengutamakan keamanan data dengan menggunakan arsitektur microservices. Bagian ini menguraikan temuan dari tahap identifikasi masalah, studi literatur, analisis kebutuhan, perancangan sistem, proses implementasi, hasil pengujian keamanan, dan evaluasi kinerja sistem secara keseluruhan, sesuai dengan tahapan metodologi yang telah ditetapkan [1], [6].

### A. Hasil Identifikasi Masalah

Hasil dari tahapan identifikasi masalah menunjukkan bahwa arsitektur monolitik sistem berbagi file konvensional adalah masalah utama. Kekurangan perlindungan data saat berada di media penyimpanan, juga dikenal sebagai "data at rest," adalah penyebabnya. Kegagalan satu fungsi, seperti kegagalan modul autentikasi, dapat menyebabkan seluruh sistem berbagi file lumpuh dalam sistem monolitik. Hasil ini menunjukkan bahwa penyelesaian masalah tersebut memerlukan arsitektur microservices yang membagi beban kerja dan kriptografi end-to-end pada sisi server untuk melindungi kerahasiaan file [1], [6].

### B. Hasil Studi Literatur

Parameter teknis yang paling cocok untuk sistem ini ditetapkan berdasarkan analisis literatur tentang keamanan informasi. Karena kemampuannya yang luar biasa dalam proses enkripsi dan dekripsi dokumen digital berbasis web, algoritma Advanced Encryption Standard (AES) dengan kunci 256-bit dipilih untuk memastikan data aman tanpa mengganggu kinerja server secara signifikan [1]. Selain itu, AES dapat digunakan untuk mendukung keamanan sistem informasi, seperti menjaga data autentikasi agar tidak mudah dibaca oleh orang lain [2]. Namun, karena JSON Web Token (JWT) memenuhi kebutuhan pengamanan API dan komunikasi antarlayanan sistem microservices, mereka dipilih sebagai standar untuk pertukaran informasi autentikasi untuk arsitektur microservices yang tidak memiliki batas [6].

### *C.* Hasil Analisis Kebutuhan Sistem

Spesifikasi kebutuhan yang harus dipenuhi oleh sistem dibuat pada tahap analisis. Hasil dari pengenalan persyaratan ini dibagi menjadi dua kategori utama, yaitu:

Kebutuhan Fungsional: Sistem harus memiliki endpoint API untuk registrasi pengguna, autentikasi atau login untuk mendapatkan JWT, pengunggahan file yang memicu proses enkripsi AES otomatis, pengunduhan file melalui proses dekripsi otomatis, dan pengaturan kontrol akses berbasis peran (RBAC) di mana administrator memiliki kontrol penuh dan pengguna hanya dapat mengakses file mereka sendiri [1], [4], [6].

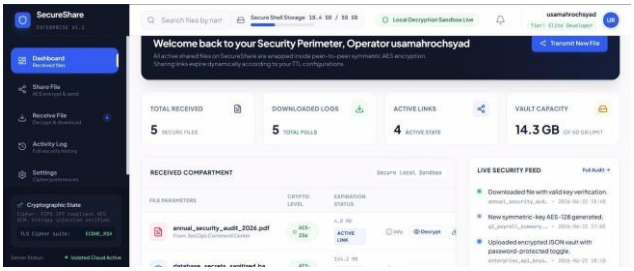
Kebutuhan Non-Fungsional: Sistem harus dapat diperluas atau ditingkatkan kapasitasnya dengan mudah; memiliki tingkat ketersediaan tinggi (High Availability) karena pemisahan microservice container; dan memiliki waktu respons enkripsi yang rendah sehingga proses pengunggahan dokumen tidak tertunda. Kebutuhan ini sesuai dengan fitur microservices, yang membantu pemisahan layanan, fleksibilitas pengembangan, dan ketahanan sistem yang lebih baik [6].

### D. Implementasi Arsitektur Microservices

Untuk membangun sistem, fungsi monolitik dibagi menjadi beberapa layanan yang berbeda. API Gateway, Pengesahan, File, dan Enkripsi adalah beberapa dari layanan ini. Layanan ini dibangun dalam lingkungan container unik dan berinteraksi melalui protokol HTTP berbasis REST API. Penyebaran layanan ini didasarkan pada konsep arsitektur microservices, yang berarti bahwa aplikasi dibagi menjadi layanan kecil yang dapat dikembangkan secara mandiri dan berkomunikasi satu sama lain melalui API [6].

Pemisahan ini telah menunjukkan bahwa itu lebih mudah untuk mengelola sistem. Pengguna dapat mengirimkan permintaan ke layanan yang sesuai dengan API Gateway sebagai satu pintu masuk. Jika

terjadi lonjakan proses unggah file, file service dapat meningkatkan kapasitas (scaling) sendiri tanpa mengganggu kinerja Verifikasi service. Selain itu, autentikasi berbasis JWT dan API Gateway menjamin akses yang aman ke sistem berbasis microservices [6].




Gambar 1. Halaman Dashboard Utama

## E. Implementasi dan Pengujian Enkripsi AES

Untuk melindungi kerahasiaan file yang diunggah atau data di tempat tidur, sistem menggunakan algoritma enkripsi Advanced Encryption Standard (AES) dengan panjang kunci 256-bit. Layanan enkripsi melakukan proses enkripsi secara otomatis sebelum file disimpan ke dalam database atau media penyimpanan cloud [1]. Penerapan enkripsi dan dekripsi file dokumen berbasis web juga dapat menjadi perlindungan tambahan sebelum dokumen disimpan atau dibagikan melalui media penyimpanan cloud [9].

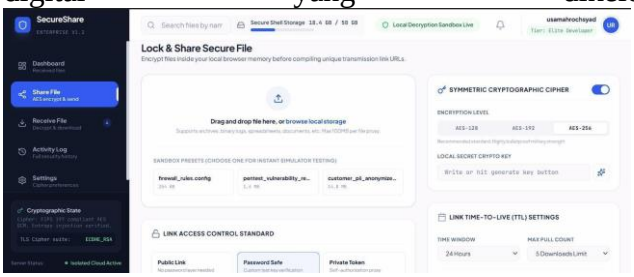
Pengujian kinerja enkripsi dilakukan untuk mengetahui jumlah waktu ekstra yang diperlukan untuk mengamankan file. Pengujian ini menggunakan berbagai sampel dokumen. Tabel I menunjukkan hasil tes [1].

Tabel I. Waktu Komputasi Enkripsi dan Dekripsi File

| Kapasitas file | Waktu Enkripsi | Waktu Dekripsi |
|---|---|---|
| 2 mb | 45 | 42 |
| 10 mb | 150 | 142 |
| 50 mb | 720 | 690 |
| 100 mb | 1450 | 1380 |

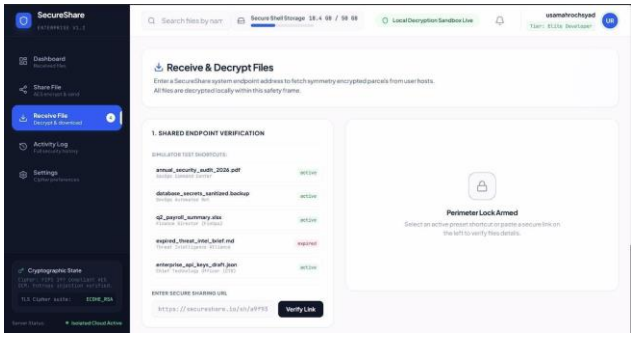
Sebagai hasil yang ditunjukkan dalam Tabel I, algoritma AES-256 memiliki tingkat efisiensi yang sangat tinggi. Waktu yang dibutuhkan untuk enkripsi file 100 MB hanya memakan waktu sekitar 1,45 detik, sebanding dengan ukuran file. Hal ini menunjukkan bahwa penggunaan enkripsi tidak menyebabkan bottleneck atau beban komputasi yang signifikan, sehingga kemudahan pengguna saat mengunggah dan mengunduh file tetap terjaga.

Halaman dashboard menampilkan jumlah file yang diterima, jumlah link aktif, dan kapasitas penyimpanan yang terpakai secara real-time, seperti yang ditunjukkan pada Gambar 1. Pada bagian ini juga ditampilkan feed keamanan langsung yang mencatat setiap peristiwa penting, seperti proses unduh, pembuatan kunci simetris, dan unggah file. Setiap file yang dikirim memiliki informasi tentang tingkat enkripsi (seperti AES-256), ukuran, dan status kedaluwarsa, sehingga pengguna dapat melihat secara langsung bagaimana keamanan aset digital yang dikelola.



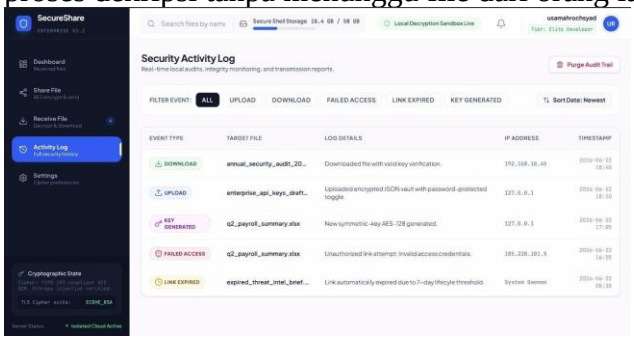

Gambar 2. Halaman Lock & Share (Enkripsi File)

pusat mana file dienkripsi sebelum dibagikan kepada pihak lain Untuk mengunggah file, pengguna dapat menggunakan metode drag-and-drop atau penelusuran lokal. Kemudian, mereka dapat memilih tingkat enkripsi AES yang mereka butuhkan: 128-bit, 192-bit, atau 256-bit. Pengguna juga dapat mengatur parameter keamanan lainnya, seperti link Time-To-Live (TTL), batas jumlah unduhan, dan metode kontrol akses, seperti token privat, kata sandi, atau publik. Proses enkripsi dilakukan melalui layanan kriptografi sebelum file disimpan ke media penyimpanan. File yang diunggah akan diproses menjadi ciphertext menggunakan algoritma AES-256, sehingga file yang tersimpan tidak berada dalam bentuk plaintext dan hanya dapat dibaca kembali melalui proses dekripsi yang sah.



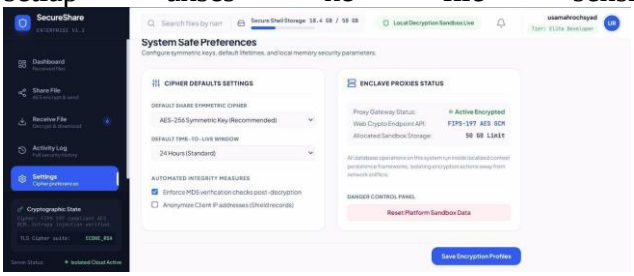

Gambar 3. Halaman Receive & Decrypt (Dekripsi File)

Halaman ini berfungsi sebagai antarmuka untuk mengambil file dan mendekripsinya melalui link berbagi. Untuk mendapatkan paket terenkripsi dari pengirim, pengguna memasukkan alamat endpoint sistem SecureShare. Untuk menjaga kerahasiaan data, proses dekripsi dilakukan secara lokal di sandbox keamanan yang telah disediakan. Selain itu, halaman ini menawarkan shortcut file uji yang memungkinkan pengguna mencoba melakukan proses dekripsi tanpa menunggu file dari orang lain.




Gambar 4. Halaman Activity Log (Log Aktivitas)

Halaman Activity Log menampilkan rekam jejak keamanan sistem yang lengkap. Tabel yang mengandung jenis peristiwa, nama file target, detail log, alamat IP, dan timestamp dibuat untuk mencatat setiap peristiwa, termasuk unggahan, unduhan, pembangkitan kunci, upaya akses gagal, dan kedaluwarsa link. Fungsi ini sangat penting untuk mendukung fungsi pemantauan integritas dan audit, serta memungkinkan administrator untuk melacak setiap akses ke file sensitif.




Gambar 5. Halaman System Safe Preferences (Pengaturan Keamanan Sistem)

Gambar 5 di halaman ini menunjukkan bahwa antarmuka memungkinkan administrator untuk mengkonfigurasi parameter keamanan global sistem. Untuk menyesuaikan pengaturan, Anda dapat memilih cipher simetris yang disarankan (AES-256), jendela waktu kadaluarsa standar (Time-To-Live), dan mengaktifkan fitur otomatis seperti verifikasi checksum MD5 pasca-dekripsi dan anonimisasi alamat IP klien. Halaman tersebut juga menampilkan status gateway proxy, konfigurasi API kriptografi (FIPS-197 AES-GCM), dan alokasi penyimpanan sandbox yang membatasi ruang isolasi untuk operasi enkripsi.

### A. *Hasil Penerapan Kontrol Akses dan RBAC*

Token web JSON (JWT) dan kontrol akses berbasis peran (RBAC) melindungi sistem dari akses yang tidak sah. Token JWT dengan masa berlaku tertentu dan informasi peran (Admin atau User) akan diberikan kepada setiap pengguna yang masuk.

Kontrol akses dievaluasi menggunakan metode Black Box Testing melalui skenario percobaan akses endpoint API. Tabel II menunjukkan hasil evaluasi.

Tabel II. Skenario Pengujian Akses Sistem

| Skenario Pengujian | Hasil yang Diharapkan | Status Kesimpulan |
|---|---|---|
| Akses tanpa token JWT | Sistem menolak dengan HTTP 401 (Unauthorized) | Valid |
| Akses token kedaluwarsa | Sistem menolak dengan HTTP 401 (Unauthorized) | Valid |
| *User* mengakses menu *Admin* | Sistem menolak dengan HTTP 403 (Forbidden) | Valid |
| *User* mengunduh file | Akses diizinkan | Valid |

| | | |
|---|---|---|
| miliknya | (HTTP 200 *OK*), file didekripsi | |
| User mengunduh file orang lain tanpa hak | Sistem menolak dengan HTTP 403 (*Forbidden*) | Valid |

Hasil Tabel II menunjukkan bahwa mekanisme kontrol akses berfungsi dengan persentase keberhasilan seratus persen. Aturan RBAC memastikan bahwa data antar-pengguna terpisah, mencegah kebocoran data internal karena peningkatan hak akses. Di sisi lain, JWT memastikan identitas pengguna pada tingkat API Gateway.

### B. Diskusi dan Evaluasi Keseluruhan

Hasil pengujian menunjukkan bahwa sistem berbagi file yang menggunakan arsitektur microservices, enkripsi AES-256, dan kontrol akses JWT/RBAC dapat menutupi kerentanan sistem berbagi file biasa. Meskipun media penyimpanan diretas, data plaintext dapat diamankan menjadi ciphertext yang tidak dapat dibaca. Pada saat yang sama, pemisahan layanan menjamin bahwa layanan lain (seperti autentikasi) tetap dapat diakses jika salah satu node (misalnya layanan File) mengalami gangguan, sehingga meningkatkan ketersediaan.

## IV. KESIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem berbagi file aman berbasis web yang telah dilakukan, dapat ditarik beberapa kesimpulan sebagai berikut:

**Pertama**, Terbukti bahwa menggunakan arsitektur microservices dapat meningkatkan skalabilitas, fleksibilitas, dan keandalan sistem secara keseluruhan. Dengan memisahkan fungsi-fungsi utama menjadi layanan yang independen, seperti layanan autentikasi, layanan file, dan enkripsi, sistem dapat menghindari Single Point of Failure (SPOF) dan menangani beban kerja secara lebih terdistribusi.

**Kedua**, Dengan menggunakan algoritma enkripsi Advanced Encryption Standard (AES) 256-bit, Anda dapat memberikan jaminan kerahasiaan tingkat tinggi terhadap file yang diunggah, juga dikenal sebagai data di tengah jalan. Menurut pengujian performa, tidak ada hambatan (bottleneck) yang mengganggu pengalaman pengguna karena proses kriptografi berjalan dengan lancar dan waktu komputasi berbanding lurus dengan ukuran file.

**Ketiga**, Terbukti bahwa integrasi JSON Web Token (JWT) dan Pengendalian Akses Berbasis Rol (RBAC) berhasil mencegah akses yang tidak sah ke sistem. Pengujian aksesibilitas memastikan bahwa sistem secara akurat mencegah pengguna melakukan tindakan di luar hak akses mereka. Sistem berbagi file ini sangat aman dan tangguh terhadap berbagai risiko kebocoran data karena kombinasi keamanan jaringan API Gateway dan enkripsi data di sisi penyimpanan.

## UCAPAN TERIMA KASIH

Bagian ini memberikan apresiasi kepada perorangan maupun organisasi yang memberikan bantuan kepada penulis dan atau kepada pihak sponsor apabila ada.

## REFERENSI

[1] M. A. R. Akbar Ridho, "*Sistem Pengamanan Dokumen Menggunakan Algoritma Kriptografi*," 2024.
[2] F. Fadlullah, M. Tahir, B. P. Bintari, M. L. Dewi, dan M. F. Ilmy, "*Implementasi Algoritma AES pada Autentikasi Login Sistem Informasi*," Jurnal Bintang Pendidikan Indonesia (JUBPI), vol. 1, no. 2, 2023.
[3] R. Fatwa Ikhwani, "*Makalah IF4020 Kriptografi – Teknik Informatika ITB – Semester I Tahun*," 2025.
[4] D. Fitrianto dan W. Y. Sulistyo, "*Penerapan Role Based Access Control untuk Keamanan Data Multi Tenant SIMPEL Lazismu*," INSECT, vol. 12, no. 01, 2026.
[5] A. Husnah Nasrullah, A. S. Amaliah, dan G. R. Jannah, "*Hybrid Kriptografi Menggunakan RSA dan AES untuk Keamanan Pengiriman File Digital*," vol. 2, no. 2, 2025. [Online].
Available: https://journal.unm.ac.id/index.php/CIVE/
[6] G. R. Purba, R. T. Adek, dan Y. Afrillia, *"Keamanan Endpoint API Menggunakan OAuth2 pada Unit Layanan Terpadu Universitas Malikussaleh,"* RABIT: Jurnal Teknologi dan Sistem Informasi Univrab, vol. 10, no. 2, pp. 1424–1434, 2025.
[7] D. Irawan dan Fauzi, "*Implementasi Arsitektur Microservices pada Pengembangan Aplikasi Absensi Web Terdistribusi*," Bit-Tech, vol. 7, no. 3, pp. 1118–1127, 2025. doi:10.32877/bt.v7i3.2401.
[8] J. Patel dan A. Trivedi, "*Cloud Based Secure File Sharing Using Access Control*," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, pp. 337–348, 2022. doi:10.32628/cseit228458.
[9] I. Kosat, J. Anastasia, A. Tanesib, M. O. Bona, F. Sallu, dan M. K. Sifa, "*Perancangan Sistem Enkripsi dan Dekripsi File Dokumen Berbasis Web pada Google Drive Menggunakan Algoritma Advanced Encryption Standard (AES)*," Blantika: Multidisciplinary Jornal, vol. 3, 2025. [Online].
Available: https://blantika.publikasiku.id/

[10] T. P. Lia, S. P. Sitorus, D. P. Sartika, A. Pratiwi, dan Z. Hasibuan, "*Evaluasi Efektivitas Mekanisme Enkripsi End-to-End dalam Mengurangi Resiko Kebocoran Data pada Layanan Komunikasi Berbasis Cloud*," Jurnal Minfo Polgan, vol. 14, no. 2, pp. 3344–3348, 2025. doi:10.33395/jmp.v14i2.15744.