Analisis Keamanan dan Performa OwnCloud *Server* dengan OWASP ZAP *Penetration Test*

Mohammad Bima Jalaluddin Akbar[1], Agus Prihanto[2]

[1,2] Universitas Negeri Surabaya Fakultas Teknik Program Studi S1 Teknik Informatika
[1]mohammadbima.20089@mhs.unesa.ac.id
[2]agusprihanto@unesa.ac.id

***Abstrak*—Seiring dengan meningkatnya kebutuhan akan layanan penyimpanan data yang aman, efisien, dan fleksibel, solusi *cloud storage* mandiri seperti OwnCloud menjadi alternatif yang menjanjikan dibandingkan layanan cloud publik. Penelitian ini bertujuan untuk menganalisis tingkat keamanan dan performa OwnCloud Server dengan pendekatan penetration testing menggunakan OWASP ZAP serta uji performa menggunakan Apache JMeter. Metode penelitian yang digunakan adalah PPDIOO, yang terdiri dari tahapan *Prepare, Plan, Design, Implement, Operate*, dan *Optimize*, guna memastikan proses berjalan secara sistematis dan terstruktur. Hasil pengujian keamanan menunjukkan terdapat 23 jenis kerentanan, termasuk satu risiko tinggi berupa *SQL Injection*, dengan skor risiko total sebesar 19,3 yang termasuk dalam kategori risiko sedang. OwnCloud juga berhasil menerapkan fitur keamanan seperti autentikasi dua faktor (2FA) dan kebijakan kata sandi *(password policy),* yang terbukti mampu mencegah akses tidak sah. Sementara itu, pengujian performa menunjukkan bahwa OwnCloud mampu menangani beban hingga 100 pengguna simultan dengan tingkat kesalahan dan waktu respons dalam batas toleransi. Berdasarkan hasil tersebut, OwnCloud dinilai layak digunakan sebagai alternatif *cloud on-premise*, khususnya bagi instansi yang memprioritaskan kendali terhadap data dan aspek keamanan informasi. Meskipun demikian, disarankan untuk melakukan mitigasi terhadap kerentanan yang ditemukan sebelum sistem diterapkan di Lingkungan produksi secara penuh.**

Kata Kunci—**OwnCloud, penetration testing, OWASP ZAP, Apache JMeter, keamanan informasi, *cloud on-premise***

I. PENDAHULUAN

Perkembangan teknologi digital mendorong meningkatnya kebutuhan terhadap layanan penyimpanan data yang aman, fleksibel, dan mudah diakses. Banyak organisasi, *startup*, maupun institusi kini memanfaatkan *cloud computing* untuk mengelola aset digital mereka. Berdasarkan laporan *We Are Social* dan Data Reportal (2025), tingkat penetrasi internet di Indonesia telah mencapai 79,5% dari total populasi, dengan lebih dari 224 juta pengguna aktif yang berpotensi menghasilkan dan mengelola data setiap hari. Seiring meningkatnya aktivitas digital tersebut, kebutuhan terhadap layanan cloud yang aman dan andal menjadi semakin penting [1].

Namun penggunaan layanan *cloud* publik menimbulkan tantangan baru, terutama dalam hal keamanan dan kendali atas data. Oleh karena itu, masyarakat kini mulai beralih ke *cloud on-premise* (pribadi) yang dapat dikelola secara mandiri untuk memenuhi tuntutan data *sovereignty*, privasi, dan kepatuhan terhadap regulasi lokal [2]. Dengan pendekatan ini, organisasi dapat memiliki kontrol penuh terhadap konfigurasi keamanan, enkripsi, serta kebijakan akses data. Salah satu platform penyimpanan *cloud open-source* yang memungkinkan hal tersebut adalah OwnCloud, yang dapat diinstal pada *server* pribadi tanpa ketergantungan pada pihak ketiga [3].

OwnCloud menawarkan sejumlah fitur seperti sinkronisasi lintas perangkat, manajemen pengguna, autentikasi dua faktor (2FA), dan enkripsi data, menjadikannya alternatif menarik bagi organisasi yang menuntut privasi tinggi [4]. Namun, meskipun OwnCloud dikenal memiliki mekanisme keamanan yang cukup baik, masih terdapat potensi kerentanan akibat konfigurasi sistem yang tidak optimal, celah pada komponen web, atau praktik keamanan yang kurang disiplin. Penelitian [5] menunjukkan bahwa platform berbasis *web open-source* rentan terhadap serangan seperti *SQL Injection, Cross-Site Scripting (XSS),* dan *Brute Force Login* jika tidak dikonfigurasi sesuai standar keamanan.

Di sisi lain, skala ancaman siber di Indonesia juga terus meningkat. Laporan Badan Siber dan Sandi Negara (BSSN) mencatat terdapat 370 juta anomali trafik serangan siber sepanjang tahun 2024, meningkat 25% dibanding tahun sebelumnya (BSSN, 2025). Selain itu, [6] menempatkan Indonesia di peringkat 26 dunia dalam jumlah kebocoran data, dengan lebih dari 2,1 juta akun terdampak hanya dalam kuartal pertama 2024. Data tersebut menunjukkan bahwa risiko kebocoran informasi tidak hanya terjadi pada sistem publik, melainkan juga dapat muncul akibat celah keamanan yang belum ditangani dengan tepat, baik pada sistem internal maupun mandiri.

Fakta tersebut menegaskan pentingnya evaluasi keamanan sistem *cloud* seperti OwnCloud agar mampu melindungi data dari potensi eksploitasi. Untuk mengukur sejauh mana sistem cloud seperti OwnCloud mampu menjaga keamanan data, diperlukan pendekatan yang sistematis melalui pengujian penetrasi. Metode ini mensimulasikan serangan siber untuk menemukan dan mengevaluasi potensi celah keamanan sebelum dimanfaatkan oleh pihak yang tidak berwenang [7]. Salah satu alat yang umum digunakan adalah OWASP Zed *Attack Proxy* (ZAP), sebuah alat *open-source* yang dikembangkan oleh Open Web Application Security Project (OWASP) dan mengacu pada standar OWASP *Web Security Testing Guide* (WSTG).

OWASP juga merilis daftar OWASP Top 10 (2021), yaitu sepuluh kategori risiko keamanan aplikasi web paling kritis di dunia, meliputi : *Broken Access Control, Cryptographic Failures, Injection, Insecure Design, Security Misconfiguration, Vulnerable and Outdated Components,*

Identification and Authentication Failures, Software and Data Integrity Failures, Security Logging and Monitoring Failures, serta Server-Side Request Forgery (SSRF) (OWASP, 2023).

Beberapa penelitian terdahulu mendukung relevansi penggunaan OWASP Top 10 sebagai acuan analisis keamanan. Misalnya, studi [8] menemukan bahwa aplikasi berbasis web di Indonesia masih rentan terhadap serangan *SQL Injection* dan *XSS* karena lemahnya kontrol autentikasi dan konfigurasi *server*. Penelitian [9] juga menunjukkan bahwa penggunaan OWASP ZAP mampu mengidentifikasi lebih dari 80% kerentanan utama OWASP Top 10 pada sistem *e-learning* berbasis *cloud*. Di sisi lain, riset Omar et al. (2023) membuktikan bahwa kombinasi antara *penetration testing* OWASP ZAP dan *load testing* Apache JMeter dapat memberikan hasil komprehensif dalam menilai keamanan dan performa *web server*, khususnya pada aspek *availability* yang sering diabaikan.

Pengujian performa menjadi aspek penting karena sistem yang aman tetapi tidak responsif tetap dianggap gagal dari sisi keandalan. Dalam hal ini, Apache JMeter digunakan untuk mengukur *availability* dan stabilitas sistem melalui simulasi pengguna secara bersamaan [10]

Penelitian ini bertujuan untuk mengisi kesenjangan penelitian terkait evaluasi keamanan dan performa OwnCloud dengan menerapkan metode pengujian penetrasi *(penetration testing)* menggunakan OWASP ZAP dan pengujian beban (load/stress testing) menggunakan Apache JMeter. Melalui pendekatan berbasis prinsip CIA Triad. Hasil penelitian diharapkan dapat memberikan gambaran komprehensif mengenai tingkat ketahanan OwnCloud Server sebagai solusi *cloud on-premise*, sekaligus sebagai rekomendasi penguatan konfigurasi keamanan dan optimasi performa yang relevan bagi organisasi yang mengutamakan keamanan, privasi, dan keandalan sistem dalam pengelolaan data digital.

II. METODE PENELITIAN

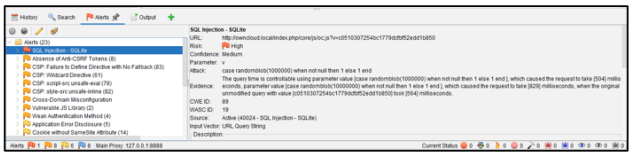
Penelitian ini menggunakan metode *PPDIOO (Prepare, Plan, Design, Implement, Operate, Optimize)* yang dikembangkan oleh Cisco Systems sebagai kerangka kerja pengelolaan sistem jaringan dan teknologi informasi secara menyeluruh. Metode ini dipilih karena penelitian mencakup instalasi, konfigurasi, pengujian keamanan menggunakan *OWASP ZAP*, serta pengujian performa menggunakan *Apache JMeter* pada *OwnCloud Server* sehingga membutuhkan alur kerja yang terstruktur dan mudah dievaluasi.

Tahap *Prepare* dilakukan dengan analisis permasalahan, penentuan tujuan, dan studi literatur terkait *cloud computing*, *OwnCloud*, *penetration testing*, serta parameter performa server. Tahap *Plan* menyusun skenario pengujian keamanan dan performa, menentukan parameter seperti *response time*, *throughput*, *error request*, dan *average bytes*, serta menyiapkan infrastruktur perangkat keras dan perangkat lunak. Selanjutnya tahap *Design* merancang topologi jaringan, alur *penetration testing* dengan *OWASP ZAP*, dan skenario uji beban menggunakan *Apache JMeter*.

Tahap *Implementation* meliputi instalasi dan konfigurasi *OwnCloud*, *OWASP ZAP*, dan *JMeter* serta pelaksanaan simulasi serangan dan pengujian performa. Tahap *Operate* menganalisis hasil pengujian keamanan berdasarkan *OWASP WSTG* dan membandingkan performa sebelum dan sesudah serangan. Tahap terakhir *Optimize* memberikan rekomendasi peningkatan keamanan dan performa sistem. Dengan penerapan *PPDIOO*, penelitian dapat berjalan sistematis, terukur, dan mendukung evaluasi efektivitas *OwnCloud* sebagai alternatif *cloud on-premise*.

III. HASIL DAN PEMBAHASAN

A. Uji Fungsionalitas OwnCloud Server



Gambar 1. Hasil dari *automated scan*

Berdasarkan hasil yang terdapat pada gambar 1, dapat terlihat bahwa terdapat 23 celah keamanan yang dimiliki oleh OwnCloud. Hal ini akan dijabarkan dalam tabel berikut :

Tabel 1. Tabel hasil dari *automated scan*

Tingkat Risiko	Jenis Kerentanan	Jumlah
High	*SQL Injection – SQLite*	1
Medium	*Absence of Anti-CSRF Tokens*	8
	CSP: Failure to Define Directive with No Fallback	83
	CSP: Wildcard Directive	61
	CSP: script-src unsafe-eval	78
	CSP: style-src unsafe-inline	82
	Cross-Domain Misconfiguration	1
	Vulnerable JS Library	2
	Weak Authentication Method	4
Low	*Application Error Disclosure*	5
	Cookie without SameSite Attribute	14
	Private IP Disclosure	5
	Server Leaks Version Information via "Server" header	234
	Timestamp Disclosure – Unix	20
	X-Content-Type-Options Header Missing	92
Informational	*Authentication Request Identified*	4
	Charset Mismatch	6
	Information Disclosure – Sensitive Information in URL	5
	Information Disclosure – Suspicious Comments	51
	Modern Web Application	65
	Session Management Response Identified	100
	User Agent Fuzzer	1548
	User Controllable HTML Element (Potential XSS)	3

Berdasarkan Tabel 1 ditemukan 23 jenis kerentanan dengan total kemunculan sebanyak 2.474 kali. Pada klasifikasi OWASP Top 10, kategori Broken *Access Control*

menunjukkan ketiadaan anti-CSRF token, pengungkapan IP privat, dan informasi sensitif pada URL yang berdampak pada kerahasiaan serta integritas data. Sementara itu *Cryptographic Failures* tidak ditemukan. Kerentanan paling serius terdapat pada Injection berupa SQL Injection pada parameter *v* di file *oc.js* dengan risiko tinggi karena memungkinkan akses dan manipulasi database.

Kategori *Insecure Design* memperlihatkan *error disclosure* dan komentar HTML yang membocorkan informasi sistem dengan risiko rendah. Pada *Security Misconfiguration* ditemukan kesalahan konfigurasi CSP, *missing security headers*, *cross-domain misconfiguration*, serta *server version disclosure* dengan risiko rendah hingga sedang yang membuka peluang *information disclosure* dan XSS. Selain itu terdapat penggunaan komponen lama pada *Vulnerable and Outdated Components* serta autentikasi lemah dan paparan token sesi pada *Identification and Authentication Failures* yang mengancam keamanan akun pengguna.

Pada *Software and Data Integrity Failures* ditemukan *library* eksternal tanpa verifikasi integritas dengan risiko sedang, sedangkan *Security Logging and Monitoring Failures* dan *SSRF* tidak ditemukan. Secara umum sebagian besar temuan berada pada tingkat rendah hingga sedang, namun tetap perlu diperbaiki agar informasi tidak dimanfaatkan penyerang dan keamanan aplikasi lebih kuat secara keseluruhan.

B. Perhitungan Skor

OWASP ZAP memberikan klasifikasi terhadap setiap temuan berdasarkan dua parameter utama, yaitu: Risk Level (Skor Risiko) merupakan gambaran tingkat keparahan dampak jika kerentanan dieksploitasi. Kategori risiko dibagi menjadi: High, Medium, Low, dan Informational, sesuai dengan standar dari OWASP Top 10 (OWASP, 2021). Confidence Level (Skor Validitas) merupakan gambaran tingkat keyakinan terhadap validitas temuan tersebut (High, Medium, Low), yang mengacu pada standar dokumentasi ZAP (ZAP Scanning Report, 2025).

Berdasarkan kedua hal tersebut, berikut ini tabel 2 merupakan rekapitulasi jumlah temuan berdasarkan kombinasi skor risiko dan skor validitas dari hasil pemindaian pada sistem OwnCloud.

Tabel 2. Rekapitulasi Temuan Berdasarkan Risiko dan Validitas

Risk Level	Confidence: High	Medium	Low	Jumlah Temuan
High	0	1	0	1
Medium	4	3	1	8
Low	1	4	1	6
Informational	1	4	3	8
Total	6	12	5	23

Meskipun OWASP Risk Rating Methodology tidak memberikan nilai numerik spesifik, dalam praktik pengujian penetrasi modern dan pelaporan keamanan kuantitatif. Pendekatan bobot risiko dan validitas ini merupakan adaptasi dari praktik industri seperti yang digunakan oleh IBM X-Force (2020) dan Tenable (2022), di mana sistem skoring numerik digunakan untuk mengklasifikasikan risiko secara kuantitatif dalam laporan keamanan. Berikut ini tabel 3 digunakan sebagai pendekatan untuk menyederhanakan analisis :

Tabel 3. Adaptasi Penggunaan Bobot Risiko dan Validitas

Kategori	Tingkat	Skor
Risiko	High	3
	Medium	2
	Low	1
	Informational	0
Validitas	High	1.0
	Medium	0.7
	Low	0.4

Perhitungan menggunakan formula:

$$\textbf{\textit{Skor Risiko Total}}$$

$$\sum(\textbf{\textit{Jumlah Temuan}} \times \textbf{\textit{Severity Skor}} \times \textbf{\textit{Confidence Skor}})$$

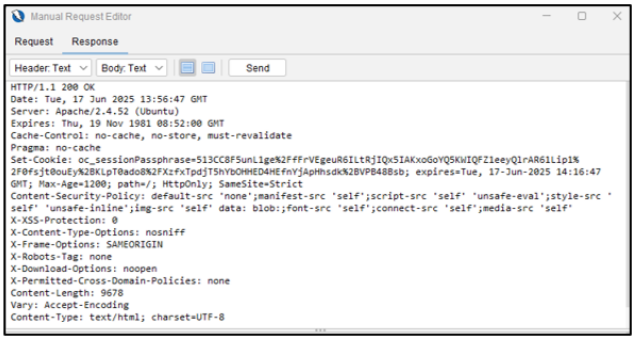
Tabel 4. Perhitungan Skor Risiko Total

Risk Level	Confidence Level	Jumlah Temuan	Perhitungan	Skor
High	Medium	1	1 × 3 × 0.7	2.1
Medium	High	4	4 × 2 × 1.0	8.0
Medium	Medium	3	3 × 2 × 0.7	4.2
Medium	Low	1	1 × 2 × 0.4	0.8
Low	High	1	1 × 1 × 1.0	1.0
Low	Medium	4	4 × 1 × 0.7	2.8
Low	Low	1	1 × 1 × 0.4	0.4
Informational	Semua Level	8	8 × 0 × 0.0	0.0
Total		23		19.3

Berdasarkan hasil perhitungan skor risiko total yang terdapat pada tabel 4, diperoleh nilai akumulatif sebesar 19,3 yang merupakan hasil analisis dari 23 temuan kerentanan menggunakan OWASP ZAP. Nilai ini merupakan hasil penggabungan antara tingkat *risiko (high, medium, low, informational)* dengan tingkat validitas *(confidence level)* setiap temuan, menggunakan pendekatan bobot kuantitatif yang umum digunakan dalam praktik industri keamanan informasi. Meskipun pendekatan ini bukan bagian eksplisit dari OWASP Risk Rating Methodology, sistem skoring berbobot yang digunakan merupakan adaptasi dari kerangka kerja penilaian risiko yang diterapkan oleh berbagai platform keamanan profesional seperti IBM X-Force dan Tenable. Berdasarkan skala klasifikasi risiko total yang telah ditetapkan, skor 19,3 termasuk ke dalam kategori risiko sedang (medium risk), yang mengindikasikan bahwa sistem OwnCloud masih memiliki potensi kerentanan yang dapat dimanfaatkan oleh pihak tidak bertanggung jawab, terutama pada temuan risiko tinggi seperti SQL Injection. Oleh karena itu, sistem ini belum sepenuhnya aman untuk digunakan dalam lingkungan produksi yang sensitif tanpa terlebih dahulu dilakukan mitigasi terhadap celah-celah keamanan yang ditemukan. Kesimpulan ini mempertegas pentingnya tindakan korektif dan penerapan kontrol keamanan tambahan untuk menurunkan tingkat risiko keseluruhan dari sistem.

C. Hasil *Manual Explore* OWASP ZAP

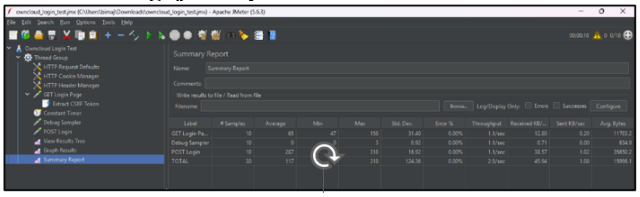
Gambar 2. POST menuju /login/challenge/totp dengan otp palsu



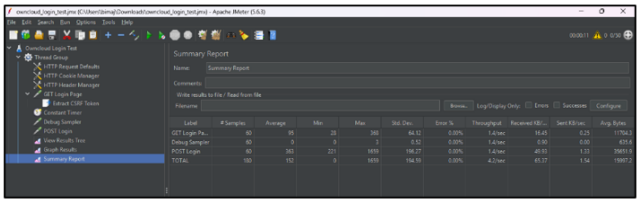
Gambar 3. Response dari pengiriman otp palsu

Hasil pengujian menunjukkan bahwa OwnCloud menolak percobaan autentikasi jika kode OTP yang diberikan tidak valid. Pada Gambar 3 memperlihatkan permintaan POST ke endpoint /login/challenge/totp dengan challenge=000000 . Permintaan ini mengandung kode OTP yang salah. Server OwnCloud membalas dengan 200 OK, akan tetapi meskipun server membalas dengan 200 OK tidak terjadi perubahan terhadap UI OwnCloud, yang mana OwnCloud tidak masuk ke dalam dashboard. Dengan demikian tidak ditemukan akses masuk ke sistem ketika kode kedua tidak sesuai. Ini menegaskan bahwa sistem 2FA di OwnCloud berfungsi, akan tetapi masih berpotensi terjadinya pelanggaran akses dikarenakan tidak ditemukan adanya rate limiting yang berfungsi sebagai pembatas antara permintaan dan respon, sehingga aktivitas berlebihan yang mencurigakan dapat dicegah tanpa mengganggu aktivitas pengguna valid (Haniyah et al., 2024).

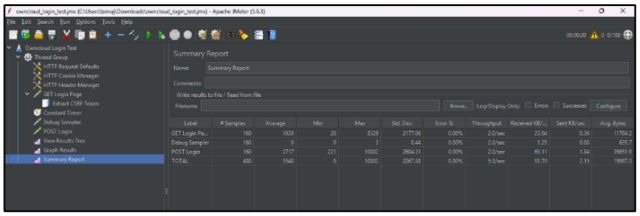
D. Hasil Pengujian Apache JMeter



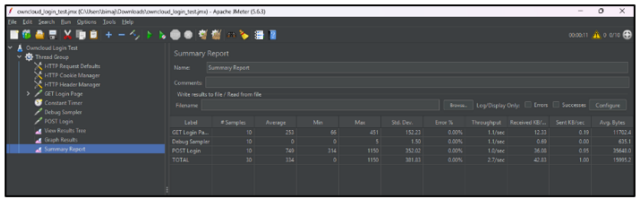
Gambar 4. Tanpa serangan dengan 10 user



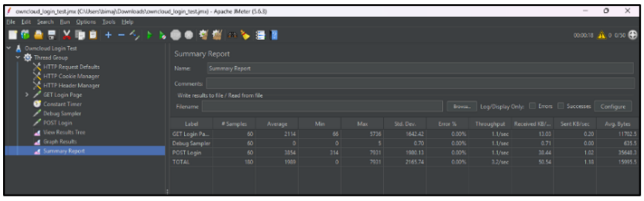
Gambar 5. Tanpa serangan dengan 50 user



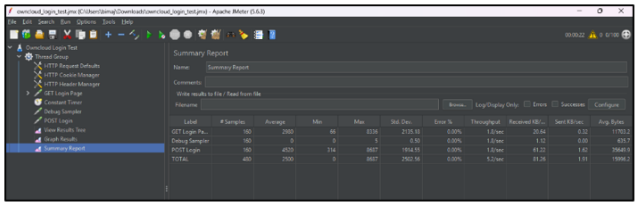
Gambar 6. Tanpa serangan dengan 100 user



Gambar 7. Serangan dengan 10 user



Gambar 8. Serangan dengan 50 user



Gambar 9. Serangan dengan 100 user

Tabel 5. Tabel Hasil Pengujian JMeter tanpa serangan OWASP ZAP

Parameter	Uji 1 (User : 100)	Uji 2 (User : 50)	Uji 3 (User : 10)
Jumlah Sampel	480	180	30
Average (ms)	1548	152	117
Min (ms)	0	0	0
Max (ms)	10802	1659	318
Std. Dev. (ms)	2267.88	194.59	124.36
Error %	0.00%	0.00%	0.00%
Throughput (req/sec)	5.9/sec	4.2/sec	2.9/sec
Received KB/sec	91.70	65.37	45.94
Sent KB/sec	2.15	1.54	1.08
Avg. Bytes	15,997.3	15,997.2	15,996.1

Tabel 6. Tabel Hasil Pengujian JMeter dengan serangan OWASP ZAP

Parameter	Uji 1 (User : 100)	Uji 2 (User : 50)	Uji 3 (User : 10)
Jumlah Sampel	480	180	30
Average (ms)	2500	1989	334
Min (ms)	0	0	0
Max (ms)	8687	7931	1150
Std. Dev. (ms)	2502.56	2165.74	381.83
Error %	0.00%	0.00%	0.00%
Throughput (req/sec)	5.2/sec	3.2/sec	2.7/sec

Parameter	Uji 1 (User : 100)	Uji 2 (User : 50)	Uji 3 (User : 10)
Received KB/sec	81.26	50.54	42.83
Sent KB/sec	1.91	1.18	1.00
Avg. Bytes	15,996.2	15,995.5	15,995.2

E. Pembahasan

Hasil pengujian performa menunjukkan bahwa nilai throughput mengalami penurunan pada seluruh skenario ketika OWASP ZAP dijalankan secara bersamaan dengan proses akses pengguna. Kondisi ini mengindikasikan bahwa server harus membagi sumber daya antara melayani permintaan pengguna dan menangani aktivitas pemindaian keamanan. Akibatnya kemampuan server untuk memproses permintaan per detik menjadi lebih rendah dibandingkan kondisi normal. Penurunan throughput ini memperlihatkan bahwa proses penetration testing memberikan beban tambahan yang cukup signifikan terhadap kapasitas layanan server, terutama pada layanan login yang bersifat sensitif terhadap waktu respons.

Selaras dengan penurunan throughput, rata-rata waktu respons juga mengalami peningkatan yang cukup tajam. Bahkan pada skenario beban ringan sebanyak 10 pengguna, waktu respons meningkat hampir tiga kali lipat. Hal tersebut menunjukkan bahwa aktivitas scanning tidak hanya menambah beban jaringan, tetapi juga memengaruhi proses komputasi di sisi server seperti validasi autentikasi, pengelolaan sesi, dan pemrosesan permintaan HTTP. Dampaknya, pengguna akan merasakan keterlambatan saat melakukan login maupun saat mengakses fitur awal aplikasi. Jika kondisi ini terjadi pada lingkungan produksi dengan pengguna aktif yang banyak, maka kualitas pengalaman pengguna *(user experience)* dapat menurun karena aplikasi terasa lambat dan kurang responsif.

Berbeda dengan dua parameter sebelumnya, nilai rata-rata byte per request tidak mengalami perubahan signifikan antara kondisi normal dan saat diserang. Hal ini menunjukkan bahwa isi komunikasi data antara klien dan server tetap konsisten, sehingga OWASP ZAP tidak mengubah ataupun merusak konten pertukaran data. Selain itu tidak ditemukan *error request* pada seluruh skenario pengujian. Artinya server OwnCloud masih mampu memproses semua permintaan secara benar dan mengembalikan respons yang valid meskipun terjadi penurunan performa. Kondisi ini menandakan sistem tetap stabil secara fungsional dan mekanisme penanganan permintaan masih berjalan dengan baik.

Secara keseluruhan dapat disimpulkan bahwa serangan OWASP ZAP lebih berdampak pada aspek performa dibandingkan kestabilan sistem. Throughput yang menurun menunjukkan berkurangnya kapasitas layanan, sedangkan peningkatan waktu respons menandakan penurunan efisiensi pemrosesan yang berpotensi mengganggu kenyamanan pengguna. Namun demikian, tidak adanya error maupun perubahan data membuktikan bahwa server tetap bekerja dengan benar dan tidak mengalami kerusakan fungsi. Dengan demikian, OwnCloud masih tergolong stabil secara operasional, tetapi membutuhkan optimalisasi konfigurasi atau peningkatan sumber daya agar mampu mempertahankan performa ketika menghadapi aktivitas pengujian keamanan maupun beban akses yang tinggi.

IV. KESIMPULAN DAN SARAN

Berdasarkan hasil pengujian fungsionalitas, keamanan, dan performa terhadap sistem OwnCloud Server, dapat disimpulkan bahwa OwnCloud berpotensi menjadi alternatif *cloud on-premise* yang fleksibel dan dapat dikendalikan secara mandiri oleh institusi. OwnCloud mampu menyediakan layanan penyimpanan, pengelolaan, dan distribusi data dengan kontrol akses pengguna serta fitur administratif yang mendukung privasi dan kepemilikan data.

Dari sisi keamanan, pengujian menggunakan OWASP ZAP mengidentifikasi 23 temuan kerentanan dengan skor risiko total 19,3, yang dikategorikan sebagai risiko sedang. Temuan paling kritis adalah kerentanan SQL Injection yang berpotensi membahayakan integritas dan kerahasiaan data. Celah lain seperti absennya *header CSP*, konfigurasi *cookie* yang tidak aman, serta potensi manipulasi autentikasi menunjukkan perlunya penguatan proteksi sistem secara menyeluruh.

Pengujian performa melalui Apache JMeter menunjukkan bahwa OwnCloud mampu menangani 10 hingga 100 pengguna bersamaan dengan respons dan *throughput* yang stabil tanpa kegagalan signifikan, yang mengindikasikan keandalan dari sisi ketersediaan layanan.

Jika dianalisis berdasarkan kerangka CIA Triad, *Confidentiality* telah didukung melalui autentikasi dua faktor (2FA), kebijakan *password* minimum, dan kontrol akses berbasis peran. *Integrity* menjadi aspek paling rentan akibat ditemukannya SQL Injection, yang menunjukkan belum memadainya mekanisme validasi dan *filtering input*. *Availability* terbukti memadai berdasarkan hasil pengujian performa yang stabil di bawah beban tinggi.

Secara keseluruhan, OwnCloud layak dijadikan solusi cloud on-premise pengganti layanan pihak ketiga, dengan syarat dilakukan mitigasi menyeluruh terhadap celah keamanan yang ditemukan. Rekomendasi utama meliputi penerapan validasi input dan prepared statement untuk mengatasi SQL Injection, implementasi Web Application Firewall (WAF), penguatan header HTTP (CSP, X-Frame-Options, Strict-Transport-Security), serta konfigurasi cookie yang aman (Secure, HttpOnly, SameSite). Pengujian lanjutan pada skala lebih besar dengan pemantauan sumber daya server juga tetap diperlukan.

Untuk penelitian selanjutnya, disarankan memperluas cakupan pengujian menggunakan alat tambahan seperti Burp Suite, Nikto, atau Nessus, serta mengkaji aspek enkripsi data, redundansi, dan integrasi protokol keamanan lanjutan. Implementasi di sektor spesifik seperti pendidikan atau pemerintahan juga menarik untuk diteliti lebih lanjut. Dengan evaluasi berkala dan penguatan keamanan yang berkelanjutan, OwnCloud berpotensi menjadi solusi cloud storage mandiri yang handal berdasarkan prinsip CIA Triad.

REFERENSI

[1] Haris, A., Sari, P., & Wijaya, R. (2023). *Implementasi Cloud Server pada Layanan Data Perusahaan.* Jurnal Teknologi dan Sistem Informasi, 8(1), 23–30.
[2] Mayendra, I., Saputra, H., & Hasanah, U. (2021). *Rancang Bangun Local Cloud Server dengan NextCloud pada CentOS 7*. JUTSI: Jurnal Teknologi dan Sistem Informasi, 1(1), 39–44.
[3] Saputra, A., Nugraha, B., & Pramono, C. (2024). *Analisis Penerapan OwnCloud untuk Penyimpanan Data Mandiri.* Jurnal Teknologi dan Komputer, 9(1), 45–52.
[4] Syamsuddin, I., Prabuwono, A. S., Basori, A. H., & Yunianta, A. (2021). *Review on OwnCloud Features for Private Cloud Data Center. TEM Journal*, 10(2), 857–862.
[5] Hermanto, A., & Haeruddin, A. (2022). *Implementasi OWASP ZAP dalam Pengujian Keamanan Aplikasi Web.* Jurnal Ilmu Komputer, 9(1), 77–84.
[6] Kelrey, N., & Muzaki, A. (2019). *Analisis Keamanan Menggunakan Penetration Testing.* Jurnal Sistem Informasi, 5(2), 99–105.
[7] Kushardianto, A., Pramono, S., & Anjani, R. (2024). Metode Penetration *Testing untuk Identifikasi Kerentanan Web.* Jurnal Teknologi Informasi, 11(1), 32–39.
[8] Haryani, N., Putri, S., & Prasetyo, H. (2023). *Penerapan Penetration Testing untuk Menjamin Keamanan Cloud Storage.* Jurnal Informatika, 7(2), 89–96.
[9] Ahmadi. (2024). Analisis Pengujian Penetrasi Sistem Informasi Menggunakan Metode Black Box Testing. Jurnal Teknologi Informasi dan Komputer, 10(1), 15–22.
[10] Suryaningrat, A., Ramayanti, D., & Sakti, A. D. (2024). *Bottleneck Identification in Web Application using Apache JMeter and Elastic Stack.* Jurnal Sistem Informasi, 10(1), 12–21.