Pasinformatik, Volume 4, Issue 2, July 2025
E-ISSN : 2986-5360

Analisis Keamanan Fitur *Login* Aplikasi: Studi Kasus Sistem Manajemen Mutu Sekolah OWASP Top 10 dengan OWASP ZAP

Miftahul Fadli Muttaqin*, Doddy Ferdiansyah, Sali Alas Majapahit***, Rita Rijayanti*****

Jurusan Teknik Informatika, Fakultas Teknik, Universitas Pasundan
Jln. Dr. Setiabudhi no. 193 Bandung, Jawa Barat
*fadli@unpas.ac.id, **doddy@unpas.ac.id, *** sali@unpas.ac.id , ****rita.rijayanti@unpas.ac.id

Abstrak : Keamanan aplikasi *web* menjadi aspek krusial dalam pengembangan sistem informasi, terlebih pada sistem yang digunakan di lingkungan pendidikan seperti Sistem Manajemen Mutu Sekolah. Fitur *login* merupakan salah satu komponen paling vital karena berfungsi sebagai gerbang utama autentikasi pengguna. Penelitian ini bertujuan untuk mengidentifikasi potensi kerentanan pada fitur *login* menggunakan alat bantu pengujian keamanan OWASP ZAP, dengan pendekatan *passive scan* dan *active scan*. Seluruh temuan kemudian dianalisis dan diklasifikasikan berdasarkan standar OWASP Top 10:2021. Hasil pengujian menunjukkan adanya sejumlah kelemahan mendasar, seperti tidak diterapkannya atribut keamanan pada *cookie* (*Secure*, *HttpOnly*), absennya *token anti*-CSRF, penggunaan pustaka *JavaScript* yang rentan, serta tidak adanya *header* keamanan standar. Selain itu, ditemukan indikasi potensi *open redirect*, *parameter pollution*, dan pengungkapan informasi teknis melalui *timestamp* dan komentar HTML. Temuan-temuan tersebut masuk ke dalam kategori *Identification and Authentication Failures* (A07), *Security Misconfiguration* (A05), dan *Vulnerable Components* (A06). Penelitian ini menyimpulkan bahwa fitur *login* belum memenuhi standar keamanan modern dan berisiko menjadi titik masuk serangan. Hasil ini dapat menjadi dasar bagi pengembang dan pengelola sistem untuk melakukan perbaikan berkelanjutan dalam upaya peningkatan postur keamanan aplikasi *web* pendidikan.

Kata Kunci : Keamanan Aplikasi *Web*, Fitur *Login*, OWASP ZAP, OWASP Top 10, Pemindaian Kerentanan

I. PENDAHULUAN

Perkembangan teknologi informasi yang pesat telah mendorong digitalisasi di berbagai sektor, termasuk sektor pendidikan. Salah satu implementasi digitalisasi di lingkungan sekolah adalah penggunaan Sistem Manajemen Mutu Sekolah, sebuah aplikasi berbasis *web* yang dirancang untuk mempermudah pelaksanaan proses manajemen mutu, mulai dari pengisian dokumen mutu, pelaporan, hingga monitoring dan evaluasi kegiatan akademik dan non-akademik [1]. Sebagai sistem yang memuat data penting dan sensitif, seperti data guru, siswa, serta dokumen mutu sekolah, keamanan sistem menjadi aspek krusial dalam menjaga integritas, kerahasiaan, dan ketersediaan informasi [2]. Salah satu komponen vital dari sistem ini adalah fitur *login*, yang berfungsi sebagai gerbang utama untuk mengakses seluruh layanan dalam aplikasi[3]. Jika fitur *login* tidak dilindungi dengan baik, sistem akan rentan terhadap berbagai serangan siber, seperti *brute force*, credential stuffing, atau *injection* [4]. Berbagai laporan dan penelitian menunjukkan bahwa sebagian besar insiden keamanan pada aplikasi *web* bermula dari kelemahan pada lapisan autentikasi dan otorisasi [5], [6], [7]. Untuk itu, diperlukan metode pengujian keamanan yang dapat mengidentifikasi potensi kerentanan secara sistematis. Salah satu alat yang banyak digunakan oleh profesional keamanan adalah OWASP ZAP (*Zed Attack Proxy*), sebuah *tools open-source* yang mampu melakukan pengujian keamanan otomatis terhadap aplikasi *web* [8], [9]. Penelitian ini bertujuan untuk menguji keamanan fitur *login* pada Sistem Manajemen Mutu Sekolah menggunakan OWASP ZAP, dengan merujuk pada kerangka kerja OWASP Top 10. OWASP Top 10 merupakan daftar sepuluh jenis kerentanan paling umum dan berisiko tinggi dalam aplikasi *web*, yang dijadikan standar acuan internasional dalam pengembangan dan evaluasi keamanan perangkat lunak [10], [11]. Melalui penelitian ini, diharapkan dapat diperoleh gambaran kerentanan yang mungkin dimiliki oleh sistem, serta rekomendasi mitigasi yang dapat diterapkan untuk meningkatkan *post*ur keamanan aplikasi.

II. METODE PENELITIAN

Penelitian ini menggunakan pendekatan eksperimental studi kasus yang berfokus pada pengujian keamanan fitur *login* pada aplikasi Sistem Manajemen Mutu Sekolah. Metode yang digunakan bersifat kualitatif-deskriptif dengan pendekatan teknis melalui pengujian otomatis menggunakan alat bantu OWASP ZAP (*Zed Attack Proxy*). Hasil dari pengujian ini kemudian dibandingkan dan diklasifikasikan berdasarkan kerangka kerja OWASP Top 10 versi 2021 untuk mengidentifikasi kategori kerentanannya.

A. Objek Penelitian

Objek yang menjadi fokus dalam penelitian ini adalah fitur *login* dari aplikasi *web* Sistem Manajemen Mutu Sekolah, yaitu sebuah sistem yang dikembangkan untuk mendukung pelaksanaan dan pelaporan kegiatan manajemen mutu di lingkungan sekolah. Aplikasi ini digunakan oleh berbagai pemangku kepentingan, antara lain kepala sekolah, guru, staf administrasi, dan tim penjaminan mutu internal, sehingga akses ke dalam sistem bersifat otentik dan terbatas [12]. Fitur *login* berperan sebagai pintu gerbang utama yang mengatur proses autentikasi dan akses pengguna ke berbagai modul dalam sistem, seperti pengisian dokumen mutu, pemantauan kinerja, dan verifikasi kegiatan program sekolah [13]. Dari sisi keamanan informasi, fitur *login* menjadi titik kritis karena kesalahan dalam perancangannya dapat dimanfaatkan oleh pihak yang tidak berwenang untuk:

a. Mengakses informasi sensitif, seperti dokumen evaluasi diri sekolah (EDS), instrumen akreditasi, dan laporan program kerja.
b. Mencuri kredensial pengguna melalui serangan *brute force* atau phishing.
c. Melakukan injeksi atau eksploitasi parameter input jika validasi tidak dilakukan dengan baik[7].

Secara teknis, fitur *login* terdiri dari komponen-komponen berikut:

a. Form autentikasi yang terdiri dari input username dan password.
b. Proses verifikasi terhadap basis data pengguna menggunakan metode *POST*.
c. Penggunaan *cookie*s dan session untuk menyimpan status *login*.
d. Redireksi ke *dashboard* pengguna setelah autentikasi berhasil. [14]

Sistem ini dikembangkan dengan teknologi *web* terkini, menggunakan *framework* PHP Laravel untuk backend, MySQL sebagai sistem manajemen basis data, serta JavaScript dan Bootstrap untuk antarmuka pengguna [15]. Berdasarkan observasi awal terhadap struktur HTML dan *log*ika permintaan HTTP, diketahui bahwa endpoint *login* tidak dilengkapi dengan proteksi eksplisit seperti:

a. Validasi sisi klien dan server yang kuat [10].
b. Pengelolaan session dan token berbasis standar keamanan modern (misalnya JWT, HttpOnly *cookie*, atau token CSRF) [16].

Kondisi ini menjadikan fitur *login* sebagai kandidat yang layak untuk diuji menggunakan OWASP ZAP, dengan tujuan mengidentifikasi kerentanan potensial yang mungkin tidak terdeteksi secara kasat mata oleh pengembang sistem. Selain itu, fitur *login* dipilih karena memiliki interaksi langsung dengan pengguna dari berbagai latar belakang teknis, sehingga risiko kesalahan input, reuse password, atau pola penggunaan yang tidak aman menjadi sangat tinggi[4]. Pengujian terhadap fitur ini diharapkan dapat memberikan kontribusi nyata dalam meningkatkan keamanan aplikasi *web* pendidikan, yang umumnya dibangun tanpa dukungan tim keamanan siber profesional.

B. Alat dan Bahan

Dalam konteks ini, penggunaan OWASP ZAP menjadi krusial karena kemampuannya dalam melakukan pemindaian dinamis terhadap elemen-elemen aplikasi *web*, termasuk form *login*, untuk mengidentifikasi celah kerentanan secara otomatis [17]. Penelitian ini memanfaatkan seperangkat perangkat lunak dan konfigurasi lingkungan uji untuk memungkinkan pelaksanaan pengujian keamanan secara sistematis terhadap fitur *login* pada aplikasi Sistem Manajemen Mutu Sekolah. *Tools* dan bahan yang digunakan dipilih berdasarkan kesesuaian dengan pendekatan *black-box testing*, kemampuan otomatisasi, serta kompatibilitas dengan arsitektur *web* aplikasi.

1. OWASP ZAP (*Zed Attack Proxy*)

OWASP ZAP merupakan *tools* utama yang digunakan dalam penelitian ini. ZAP adalah aplikasi *open-source* yang dikembangkan oleh komunitas OWASP, berfungsi sebagai intercepting *proxy* yang memungkinkan analisis lalu lintas HTTP/HTTPS antara *browser* dan aplikasi target [17]. OWASP ZAP dipilih karena:

a. Mendukung passive dan *active scan* untuk mendeteksi berbagai jenis kerentanan keamanan.
b. Menyediakan fitur automated *attack* mode dan manual fuzzing terhadap form *login* dan parameter input.
c. Kompatibel dengan berbagai *browser* melalui pengaturan *proxy*.
d. Memiliki basis referensi kerentanan yang diperbarui secara berkala dan mengacu pada OWASP Top 10.

2. *Browser* (Mozilla Firefox / Google Chrome)

Browser digunakan sebagai antarmuka untuk mengakses aplikasi *web* selama proses intersepsi lalu lintas. *Browser* dikonfigurasi agar lalu lintas HTTP/HTTPS-nya dialihkan melalui OWASP ZAP sebagai *proxy*. Penggunaan Firefox atau Chrome dipilih karena kompatibilitas tinggi dengan *add-on* debugging dan kemudahan konfigurasi *proxy* secara manual [17], [18].

3. Aplikasi Target: Sistem Manajemen Mutu Sekolah

Aplikasi target diuji melalui *domain* lokal atau *staging* (pengembangan), bukan server produksi, untuk menghindari gangguan layanan yang aktif[19]. Aplikasi ini berjalan di atas arsitektur client-server, menggunakan protokol HTTP/HTTPS, serta menyediakan endpoint *login* melalui form berbasis HTML.

4. Lingkungan Uji (Test Environment)

a. Sistem operasi: Windows
b. OWASP ZAP dijalankan secara lokal.
c. Aplikasi diakses melalui IP lokal atau *domain* pengujian internal sekolah (jika tersedia).
d. *Browser* telah disesuaikan dengan konfigurasi *proxy*: *localhost*:8080 sebagai endpoint ZAP *default*.

5. Data Uji dan Skenario Pengujian

Data uji yang digunakan mencakup kredensial uji standar (bukan kredensial riil) yang telah disiapkan untuk keperluan eksperimen. Skenario pengujian *login* melibatkan berbagai variasi input, seperti:

a. *Login* dengan kredensial valid dan tidak valid.
b. Input karakter spesial (XSS *payload*, SQL *injection payload*).
c. Pengiriman permintaan *login* berulang (*brute-force* simulation).
d. Pengamatan terhadap session ID dan *header* keamanan (misalnya Set-*Cookie*, *Secure*, HttpOnly, SameSite).

6. Dokumentasi dan *Log*

Seluruh proses pengujian terdokumentasi dalam bentuk:

a. *Log* serangan dari OWASP ZAP.
b. *Screenshot* dari hasil pemindaian.
c. Laporan otomatis (HTML/PDF) dari OWASP ZAP.
d. Catatan manual untuk pengamatan anomali atau kerentanan yang perlu dianalisis lebih lanjut.

Penggunaan *tools* dan bahan ini dirancang untuk mendukung proses pengujian yang berulang, dapat direproduksi, dan sesuai dengan standar pengujian keamanan aplikasi *web*.

C. Kerangka Evaluasi: OWASP Top 10

Sebagai dasar evaluasi hasil pengujian kerentanan, penelitian ini menggunakan kerangka OWASP Top 10 versi tahun 2021 yang dikembangkan oleh *Open Web* Application Security Project (OWASP) [20]. OWASP Top 10 merupakan daftar sepuluh kategori ancaman keamanan aplikasi *web* paling umum dan berisiko tinggi, yang dikompilasi berdasarkan data global dari berbagai sumber industri keamanan siber. OWASP Top 10 tidak hanya berfungsi sebagai panduan pengembangan aplikasi aman (*secure coding*), tetapi juga sebagai kerangka referensi klasifikasi risiko keamanan untuk audit, asesmen keamanan, dan penilaian pasca-implementasi aplikasi *web* [21]. Dengan demikian, kerangka ini sangat relevan untuk digunakan sebagai acuan analisis hasil uji OWASP ZAP terhadap fitur *login* aplikasi Sistem Manajemen Mutu Sekolah. Berikut ini adalah daftar dan deskripsi singkat dari kategori OWASP Top 10:2021 yang dijadikan acuan dalam penelitian [8], [10], [17], [18]:

Kode	Kategori Kerentanan	Deskripsi Singkat
A01:2021	*Broken Access Control*	Akses tidak sah terhadap sumber daya atau fitur sistem karena pembatasan akses yang lemah.
A02:2021	*Cryptographic Failures*	Kegagalan dalam melindungi data sensitif, termasuk penggunaan enkripsi yang tidak aman atau konfigurasi TLS yang salah.
A03:2021	*Injection*	Serangan yang memungkinkan penyerang menyisipkan perintah berbahaya melalui input pengguna (contoh: SQL *injection*).
A04:2021	*Insecure Design*	Desain sistem yang tidak mempertimbangkan prinsip keamanan, seperti tidak adanya validasi input atau kontrol sesi.
A05:2021	*Security Misconfiguration*	Pengaturan keamanan yang salah atau *default*, seperti penggunaan kredensial standar, error message detail, atau *header* keamanan yang tidak lengkap.
A06:2021	*Vulnerable and Outdated Components*	Penggunaan *library*, *framework*, atau modul pihak ketiga yang sudah tidak diperbarui dan memiliki kerentanan diketahui.
A07:2021	*Identification and Authentication Failures*	Kegagalan dalam proses otentikasi, seperti *brute force* yang tidak dibatasi, password yang lemah, atau token session yang mudah ditebak.
A08:2021	*Software and Data Integrity Failures*	Tidak adanya verifikasi integritas terhadap software atau data yang dikonsumsi sistem, seperti update dari sumber yang tidak terpercaya.
A09:2021	*Security Logging and Monitoring Failures*	Kurangnya mekanisme *log* dan pemantauan terhadap aktivitas mencurigakan dalam sistem.
A10:2021	*Server-Side Request Forgery (SSRF)*	Sistem backend yang melakukan permintaan HTTP berdasarkan input pengguna tanpa validasi, sehingga dapat disalahgunakan untuk mengakses layanan internal.

Tabel 1. OWASP Top 10

Pemilihan OWASP Top 10 sebagai kerangka evaluasi memberikan nilai tambah dalam hal standarisasi keamanan, relevansi praktis di industri, serta kredibilitas akademik, karena banyak digunakan dalam sertifikasi, pengujian penetrasi, dan kebijakan keamanan aplikasi modern.

D. Langkah Penelitian

Langkah-langkah dalam penelitian ini dirancang untuk mengidentifikasi ada atau tidaknya kerentanan keamanan pada fitur *login* aplikasi Sistem Manajemen Mutu Sekolah dengan menggunakan alat bantu OWASP ZAP, tanpa melakukan penilaian lebih lanjut terhadap tingkat risiko atau dampak dari kerentanan tersebut. Proses penelitian terdiri atas enam tahapan utama sebagai berikut:

1. Persiapan Lingkungan Uji

Pada tahap ini, dilakukan pengaturan lingkungan pengujian dengan memastikan bahwa seluruh komponen teknis siap digunakan. Aktivitas pada tahap ini meliputi:

a. Instalasi OWASP ZAP di sistem operasi lokal.
b. Konfigurasi *browser* (Firefox/Chrome) agar menggunakan *proxy localhost*:8080 (*default* OWASP ZAP).
c. Verifikasi konektivitas antara *browser*, ZAP, dan aplikasi target.

2. *Passive scan*ning

OWASP ZAP secara otomatis akan melakukan pemindaian pasif terhadap lalu lintas HTTP/HTTPS yang terekam. Pemindaian pasif ini bertujuan untuk mendeteksi kerentanan umum tanpa mengubah atau menyerang sistem. Beberapa kerentanan yang dapat terdeteksi pada tahap ini meliputi:

a. Ketidakhadiran *header* keamanan (misalnya X-Content-Type-Options, Strict-Transport-Security, HttpOnly).
b. Penyimpanan session ID dalam *cookie* yang tidak dienkripsi.
c. Input field tanpa validasi atau proteksi dasar.

3. *Active scan*ning

Tahap ini melibatkan pemindaian aktif terhadap endpoint *login*. OWASP ZAP akan menyisipkan *payload* otomatis untuk menguji kemungkinan kerentanan seperti:

a. *Injection* (SQL, command, atau script).
b. *Brute force* dan credential stuffing.
c. *Response manipulation* dan *redirection* tanpa kontrol.

Fokus dalam tahap ini adalah mendeteksi indikasi kerentanan, bukan melakukan eksploitasi mendalam atau pengujian lanjutan terhadap hasil *payload*.

4. Analisis Temuan Kerentanan

Setelah pemindaian selesai, hasil yang ditampilkan oleh OWASP ZAP akan ditinjau untuk memastikan validitasnya. Peneliti kemudian:

a. Mengidentifikasi nama kerentanan yang terdeteksi.
b. Mencatat jenis kerentanan berdasarkan deskripsi otomatis dari ZAP.
c. Mengelompokkan temuan ke dalam kategori OWASP Top 10 tanpa menilai tingkat risiko atau skala keparahannya.

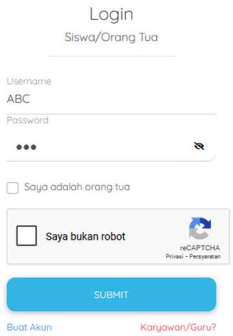
Penelitian ini tidak melanjutkan ke tahap penilaian risiko (*risk assessment*) seperti *severity rating*, *likelihood*, atau dampak kerentanan. Fokus penelitian hanya pada deteksi keberadaan kerentanan, bukan pada evaluasi dampaknya terhadap sistem secara keseluruhan.

III. HASIL DAN PEMBAHASAN

Pengujian terhadap fitur *login* pada aplikasi Sistem Manajemen Mutu Sekolah dilakukan menggunakan OWASP ZAP dalam dua tahap: *passive scan* dan *active scan*. Hasil dari kedua tahap tersebut menunjukkan adanya beberapa potensi kerentanan yang relevan, terutama yang terkait dengan pengelolaan autentikasi dan sesi pengguna. Temuan-temuan ini kemudian dianalisis dan diklasifikasikan berdasarkan kerangka kerja OWASP Top 10:2021.

A. Hasil *Passive scan*

Hasil dari pemindaian pasif menunjukkan bahwa terdapat beberapa pengaturan dasar keamanan yang belum diterapkan secara optimal pada fitur *login*.



Gambar 1. Form *Login*

Berdasarkan hasil *passive scan* terhadap fitur *login* pada aplikasi Sistem Manajemen Mutu Sekolah, OWASP ZAP menghasilkan 24 alert, yang mencakup berbagai kelemahan konfigurasi keamanan dan praktik pengembangan yang tidak aman. Temuan tersebut dikelompokkan sebagai berikut:

No	Kategori Temuan	Deskripsi	Klasifikasi OWASP
1	Vulnerable JS *Library*	Penggunaan pustaka JavaScript versi lama yang memiliki celah keamanan.	*A06: Vulnerable and Outdated Components*
2	Content Security Policy (CSP) Errors	CSP tidak diterapkan atau menggunakan konfigurasi yang lemah (unsafe-inline, unsafe-eval).	*A05: Security Misconfiguration*
3	Missing Security *Header*s	*Header* penting seperti Strict-Transport-Security, X-Frame-Options, dan X-Content-Type-Options tidak ditemukan.	*A05: Security Misconfiguration*
4	*Cookie* Flags Tidak Lengkap	*Cookie login* tidak menggunakan atribut HttpOnly, *Secure*, atau SameSite.	*A07: Identification and Authentication Failures*
5	Cross-*Domain* Misconfiguration	File JavaScript atau kebijakan CORS yang memungkinkan penyisipan kode dari *domain* lain.	*A01: Broken Access Control / A05*
6	Timestamp Disclosure	Server mengembalikan informasi waktu sistem Unix dalam respons, berpotensi untuk fingerprinting.	*A06: Information Disclosure* (umum)

Tabel 2. Hasil *Passive scan*

Fitur *login* aplikasi Sistem Manajemen Mutu Sekolah menunjukkan beberapa kelemahan penting dalam praktik konfigurasi keamanan dasar. Meskipun belum ditemukan eksploitasi langsung, hasil ini menunjukkan bahwa sistem perlu:

a. Mengaktifkan atribut keamanan pada *cookie*.
b. Menambahkan *header* HTTP standar untuk perlindungan dasar.
c. Memperbarui pustaka JavaScript ke versi terbaru.

B. *Active scan*

Setelah melakukan *passive scan*, tahap selanjutnya dalam penelitian ini adalah *active scan* terhadap fitur *login* aplikasi Sistem Manajemen Mutu Sekolah. Pengujian dilakukan menggunakan fitur *Active scan* dari OWASP ZAP, yang dirancang untuk mengirimkan berbagai *payload* dan permintaan berbahaya ke endpoint aplikasi guna mendeteksi respons yang mencurigakan atau tidak aman. Berbeda dengan *passive scan* yang hanya menganalisis lalu lintas tanpa mengubahnya, *active scan* bersifat lebih intrusif karena secara aktif mencoba menyisipkan pola serangan ke dalam parameter input aplikasi. Oleh karena itu, pengujian ini dilakukan pada lingkungan pengujian (*staging*), bukan pada server produksi. Hasil dari *active scan* menunjukkan adanya beberapa indikasi kerentanan yang berkaitan langsung dengan fitur *login*. Berikut adalah ringkasan temuan yang tercatat:

No	Temuan	Deskripsi	Kategori OWASP
1	*Vulnerable JS Library*	*Library* JavaScript yang digunakan memiliki versi yang diketahui rentan terhadap eksploitasi.	*A06: Vulnerable and Outdated Components*
2	*Absence of Anti-CSRF Tokens*	Form *login* tidak memiliki token anti-CSRF, berisiko terhadap serangan permintaan palsu.	*A04: Insecure Design*
3	*Content Security Policy (CSP) Header Not Set*	Tidak ada *header* CSP yang mencegah injeksi konten dan serangan berbasis *browser*.	*A05: Security Misconfiguration*
4	*Cross-Domain Misconfiguration*	Pengaturan keamanan antar *domain* (CORS) tidak dikonfigurasi dengan benar.	*A01: Broken Access Control*
5	*Big Redirect Detected (Potential Sensitive Info Leak)*	Parameter redirect berpotensi diarahkan ke URL lain yang berbahaya.	*A01: Broken Access Control*
6	*Cookie No HttpOnly Flag*	*Cookie* session tidak dilindungi dari akses JavaScript (tidak menggunakan *flag* HttpOnly).	*A07: Identification and Authentication Failures*
7	*Cookie Without Secure Flag*	*Cookie* tidak diatur agar hanya dikirim melalui koneksi HTTPS.	*A07: Identification and Authentication Failures*
8	*Cross-Domain JS Source File Inclusion*	File JavaScript dari *domain* lain disertakan tanpa pembatasan keamanan.	*A05: Security Misconfiguration*
9	*Strict-Transport-Security Header Not Set*	Tidak ditemukan *header* Strict-Transport-Security (HSTS) dalam respons.	*A05: Security Misconfiguration*
10	*Timestamp Disclosure - Unix*	Server merespons dengan informasi *timestamp* Unix, yang dapat digunakan untuk fingerprinting.	*A06: Information Disclosure*
11	*X-Content-Type-Options Header Missing*	*Header* untuk mencegah MIME-sniffing tidak ditemukan.	*A05: Security Misconfiguration*
12	*Authentication Request Identified*	Permintaan autentikasi ditemukan, memungkinkan enumerasi kredensial jika tidak dilindungi.	*A07: Identification and Authentication Failures*
13	*Information Disclosure - Suspicious Comments*	Terdapat komentar HTML dalam kode sumber yang mengandung informasi sensitif atau debug.	*A06: Information Disclosure*

No	Temuan	Deskripsi	Kategori OWASP
14	*Modern Web Application*	Informasi ini menunjukkan aplikasi menggunakan teknologi modern; tidak berbahaya (informasi).	*Tidak dikategorikan*
15	*Re-examine Cache-control Directives*	Respon tidak menetapkan kebijakan cache secara tegas, memungkinkan data sensitif tersimpan.	*A06: Information Disclosure*
16	*Retrieved from Cache*	Konten yang disajikan dapat berasal dari cache, mengindikasikan potensi kebocoran data.	*A06: Information Disclosure*
17	*Session Management Response Identified*	Pola respons *login* mengindikasikan sesi, perlu diawasi agar tidak bocor atau disalahgunakan.	*A07: Identification and Authentication Failures*

Tabel 3. Hasil *Active scan*

Hasil *active scan* terhadap fitur *login* aplikasi Sistem Manajemen Mutu Sekolah menunjukkan adanya sejumlah kerentanan penting yang berkaitan dengan autentikasi, konfigurasi keamanan, serta pengungkapan informasi teknis.

C. Analisis Temuan Kerentanan

Pengujian keamanan terhadap fitur *login* pada aplikasi Sistem Manajemen Mutu Sekolah dilakukan melalui dua pendekatan, yaitu *passive scan* dan *active scan* menggunakan *tools* OWASP ZAP. Kedua pendekatan ini memberikan hasil yang saling melengkapi dalam mengidentifikasi kelemahan konfigurasi, desain sistem, dan potensi risiko terhadap serangan berbasis *web*. Berdasarkan hasil pengujian tersebut, dilakukan analisis terpadu untuk mengevaluasi pola kerentanan yang berulang, kategorisasi berdasarkan OWASP Top 10, serta kekritisan area yang perlu segera diperbaiki.

1. Kategori Kerentanan yang Konsisten pada Kedua Scan

Baik *passive scan* maupun *active scan* menunjukkan bahwa kelemahan paling menonjol berada pada aspek identifikasi dan autentikasi pengguna (A07). Ini terlihat dari temuan berulang seperti:

a. *Cookie* tidak memiliki atribut *Secure*, *HttpOnly*, dan *SameSite*.
b. Tidak adanya mekanisme *anti-CSRF token*.
c. Respon *login* yang mengungkap pola autentikasi (*session management identified*).
d. Perbedaan pesan error yang memungkinkan *username enumeration*.

Temuan-temuan tersebut menunjukkan bahwa sistem belum dilengkapi dengan lapisan kontrol yang memadai dalam menjaga sesi pengguna dan mencegah penyalahgunaan kredensial.

2. Kelemahan Konfigurasi Keamanan Server (A05)

Kedua metode juga konsisten mengidentifikasi kekurangan dalam konfigurasi keamanan, seperti:

a. Tidak adanya *header Content-Security-Policy, Strict-Transport-Security*, dan *X-Content-Type-Options*.
b. Kesalahan konfigurasi CORS dan *cache control*.
c. Penggunaan script eksternal dari *domain* pihak ketiga tanpa validasi ketat.

Kondisi ini dapat memperbesar risiko terjadinya serangan *client-side*, seperti *XSS*, *clickjacking*, atau manipulasi konten melalui *domain* tidak terpercaya.

3. Pengungkapan Informasi Sistem (A06)

Baik melalui *passive* maupun *active scan*, ditemukan berbagai bentuk informasi teknis yang seharusnya tidak ditampilkan kepada pengguna akhir, seperti:

a. Timestamp Unix dalam respons server.
b. Komentar mencurigakan dalam kode HTML.
c. Petunjuk struktur *cache* atau konfigurasi *browser*.

Informasi-informasi ini mungkin tidak berbahaya secara langsung, tetapi dapat memberikan peta permukaan serangan (*attack* surface) yang lebih jelas bagi penyerang dalam tahap rekayasa awal (*reconnaissance*).

4. Potensi Kerentanan Desain dan Akses Kontrol (A01, A04)

Active scan mengungkap beberapa kerentanan tambahan yang tidak muncul di *passive scan*, seperti:

a. *Open redirect* melalui parameter redirect.
b. Parameter *pollution* (input duplikat) yang membingungkan logika sistem.

Ini menunjukkan bahwa uji aktif dapat mengungkap kelemahan yang tidak terlihat hanya dari analisis trafik, dan menyoroti pentingnya pengujian intrusif dalam proses evaluasi keamanan.

D. Pembahasan

Pengujian terhadap fitur *login* aplikasi Sistem Manajemen Mutu Sekolah menggunakan OWASP ZAP berhasil mengidentifikasi sejumlah kerentanan yang tersebar dalam berbagai kategori OWASP Top 10. Dari hasil pengujian, terlihat bahwa fitur *login* sebagai komponen krusial autentikasi pengguna masih menyimpan berbagai kelemahan yang berpotensi dieksploitasi, baik dari sisi konfigurasi, desain, maupun pengelolaan sesi. Hal ini mengindikasikan bahwa implementasi keamanan aplikasi belum sepenuhnya mengikuti prinsip *secure-by-design* yang seharusnya menjadi standar dalam pengembangan sistem informasi berbasis *web*, terutama yang digunakan di sektor pendidikan. Temuan utama dari *passive scan* dan *active scan* menunjukkan pola yang konsisten, yaitu dominasi kerentanan pada aspek Identification and Authentication Failures (A07). Atribut keamanan penting pada *cookie login* seperti *Secure*, *HttpOnly*, dan *SameSite* belum diterapkan, padahal komponen ini sangat penting untuk mencegah *session hijacking* dan *cross-site request forgery* (CSRF). Ditambah dengan ketiadaan token CSRF, fitur *login* menjadi rentan terhadap eksploitasi berbasis *browser*, terutama ketika pengguna mengakses aplikasi dari jaringan publik atau perangkat bersama. Kelemahan lainnya muncul dalam bentuk Security Misconfiguration (A05), yang terlihat dari tidak adanya *header* keamanan standar seperti Content-Security-Policy (CSP), Strict-Transport-Security (HSTS), dan X-Content-Type-Options. Ketiadaan *header* ini membuat aplikasi rentan terhadap XSS, sniffing, dan clickjacking, yang berbahaya jika dilakukan terhadap akun pengguna dengan hak akses tinggi. Di sisi lain, keberadaan *library* JavaScript versi lama dan komentar HTML yang mengandung informasi teknis memperluas *attack* surface dan mengindikasikan lemahnya proses hardening aplikasi dan audit kode sumber. Dalam konteks desain sistem, ditemukan pula indikasi *Insecure* Design (A04) seperti tidak adanya pembatasan pada redirect URL (*open* redirect) dan kemungkinan parameter pollution. Kelemahan ini mencerminkan bahwa sistem belum dilengkapi dengan mekanisme validasi dan sanitasi input yang ketat, serta belum mengimplementasikan pengujian keamanan dalam siklus pengembangan sistem (*Secure Software Development Lifecycle* - SSDLC) [22]. Temuan-temuan ini perlu mendapat perhatian serius karena aplikasi yang diuji merupakan sistem internal sekolah yang memuat data sensitif, termasuk data tenaga pendidik, laporan mutu, dan hasil evaluasi kegiatan sekolah. Ancaman terhadap integritas, kerahasiaan, dan ketersediaan informasi dalam sistem pendidikan tidak hanya berimplikasi teknis, tetapi juga berdampak pada kepercayaan publik terhadap institusi pendidikan tersebut. Namun demikian, perlu dicatat bahwa ZAP memiliki keterbatasan dalam mendeteksi kerentanan logika bisnis dan tidak menggantikan perlunya validasi manual atau uji penetrasi lanjutan oleh profesional. Secara keseluruhan, pembahasan ini menunjukkan bahwa fitur *login* dari Sistem Manajemen Mutu Sekolah perlu diperkuat dari berbagai sisi, baik pada level konfigurasi server, desain input dan sesi, hingga pembaruan komponen eksternal. Implikasi praktis dari hasil ini dapat dijadikan dasar bagi pengelola sistem untuk menyusun roadmap peningkatan keamanan, misalnya melalui penerapan *security header policy*, *enforced* HTTPS, *token-based session management*, dan pembatasan *login* berbasis *rate-limiting* atau CAPTCHA. Selain itu, penting pula bagi pengembang untuk mengadopsi prinsip *defense-in-depth* agar setiap lapisan sistem memiliki mekanisme perlindungan berlapis terhadap berbagai jenis serangan.

IV. KESIMPULAN

Penelitian ini bertujuan untuk mengidentifikasi keberadaan kerentanan keamanan pada fitur *login* aplikasi Sistem Manajemen Mutu Sekolah menggunakan metode pengujian berbasis *tools* OWASP ZAP, dengan pendekatan *passive scan* dan *active scan*. Seluruh temuan kemudian dianalisis dan diklasifikasikan berdasarkan kategori OWASP Top 10:2021, yang menjadi acuan global dalam evaluasi keamanan aplikasi *web*. Berdasarkan hasil pengujian dan analisis, dapat disimpulkan bahwa fitur *login* pada aplikasi yang diuji belum sepenuhnya aman, dan masih mengandung berbagai kelemahan mendasar yang berpotensi menjadi titik serangan. Keseluruhan temuan tersebut menunjukkan bahwa fitur *login* belum dilengkapi dengan praktik keamanan modern yang selaras dengan prinsip *secure*-by-design. Meskipun belum dilakukan uji eksploitasi lanjutan maupun analisis risiko secara kuantitatif, identifikasi kerentanan ini cukup untuk menjadi dasar dalam perbaikan desain, penguatan konfigurasi sistem, dan pembaruan teknologi yang digunakan. Sebagai kesimpulan akhir, penelitian ini menegaskan bahwa OWASP ZAP merupakan alat yang efektif dan efisien untuk melakukan deteksi dini kerentanan aplikasi *web*, terutama pada tahap pengujian awal sistem. Fitur *login* sebagai titik akses utama memerlukan perhatian khusus dalam hal keamanan, mengingat fungsinya yang sangat vital dalam pengelolaan akses pengguna terhadap sistem informasi. Hasil penelitian ini diharapkan dapat menjadi referensi bagi pengembang aplikasi, tim TI sekolah, dan pihak manajemen dalam menyusun langkah strategis untuk peningkatan keamanan aplikasi berbasis *web* di lingkungan pendidikan.

REFERENSI

[1] Hidayati, D., Komariah, A., dan Mirfani, A. M. School Based Management of Information Technology for Quality Improvement of Junior Secondary Academic Service in Bandung. Jan. 2019, doi: 10.2991/icream-18.2019.38.

[2] Anas, A. S., Utami, I. G. A. S. D. G., Maulachela, A. B., dan Juliansyah, A. KAMI index as an evaluation of academic information system security at XYZ university. Matrix Jurnal Manajemen Teknologi dan Informatika, vol. 11, no. 2, p. 55, Jul. 2021, doi: 10.31940/matrix.v11i2.2447.

[3] Sugiyanti, S. D., Widayanti, R., Ulum, M. B., Firmansyah, G., and Azizah, A. H. Design Dashboard Monitoring Teacher Performance Assessment at Cinta Kasih Tzu Chi High School. IAIC Transactions on Sustainable Digital Innovation (ITSDI), vol. 4, no. 1, p. 46, Sep. 2022, doi: 10.34306/itsdi.v4i1.569.

[4] Wiefling, S., Jørgensen, Thunem, S., dan Iacono, L. L. Pump Up Password Security! Evaluating and Enhancing Risk-Based Authentication on a Real-World Large-Scale Online Service. ACM Transactions on Privacy and Security, vol. 26, no. 1, p. 1, Jun. 2022, doi: 10.1145/3546069.
[5] Amuthadevi, C., Srivastava, S., Khatoria, R., dan Sangwan V. A Study on *Web* Application Vulnerabilities to find an optimal Security Architecture. arXiv (Cornell University), Jan. 2022, doi: 10.48550/arxiv.2204.07107.
[6] Zhong L. A Survey of Prevent and Detect Access Control Vulnerabilities. arXiv (Cornell University), Jan. 2023, doi: 10.48550/arxiv.2304.10600.
[7] Saputra, L. A., Akbar, F. M., Cahyaningtias, F., Ningrum, M. P., dan Fauzi, A. Ancaman Keamanan Pada Sistem Informasi Manajemen Perusahaan. Jurnal Pendidikan Siber Nusantara, vol. 1, no. 2, p. 58, Aug. 2023, doi: 10.38035/jpsn.v1i2.48.
[8] Putra, F. P. E., Ubaidi, U., Ḥamzah, A., Pramadi, W. A., dan Nuraini, A. Systematic Literature Review: Security Gap Detection On Websites Using Owasp Zap. Brilliance Research of Artificial Intelligence, vol. 4, no. 1, p. 348, Jul. 2024, doi: 10.47709/brilliance.v4i1.4227.
[9] Neupane, S. Detecting and Mitigating SQL Injection Vulnerabilities in Web Applications. 2025, doi: 10.48550/ARXIV.2506.17245.
[10] Mu'min, M. A., Fadlil, A., and Riadi, I. Analisis Keamanan Sistem Informasi Akademik Menggunakan Open Web Application Security Project Framework. JURNAL MEDIA INFORMATIKA BUDIDARMA, vol. 6, no. 3, p. 1468, Jul. 2022, doi: 10.30865/mib.v6i3.4099.
[11] Stephen, J., dan Young, B. Toward Secure Web Application Design: Comparative Analysis of Major Languages and Framework Choices. International Journal of Advanced Computer Science and Applications, vol. 7, no. 2, Jan. 2016, doi: 10.14569/ijacsa.2016.070206.
[12] Luthfia, D., dan Sumarto, S. Implementation of Total Quality Management in SMK Negeri 10 Bandung. Jan. 2020, doi: 10.2991/assehr.k.200130.156.
[13] Sudana, O., Kusuma, A. T. A. P., Raharja, I. M. S., dan Wirdiani, N. K. A. Penerapan Model Single Sign-On dengan Sistem Front Office dan Layanan pada Sistem E-Bengkel Terintegrasi. Jurnal Edukasi dan Penelitian Informatika (JEPIN), vol. 7, no. 3, p. 379, Dec. 2021, doi: 10.26418/jp.v7i3.48667.
[14] Mangca, D. C. Web-Based Sea Transport Booking System: Design and Development using Laravel Framework. International Journal of Advanced Research in Science Communication and Technology, p. 223, Jul. 2023, doi: 10.48175/ijarsct-12131.
[15] Aryani, R., Suratno, T., Mauladi, M., dan Utomo, P. E. P. Implementasi Sistem Informasi Manajemen Arsip Di Fakultas Sains dan Teknologi Universitas Jambi. Jurnal Ilmiah Media Sisfo, vol. 13, no. 2, p. 146, Oct. 2019, doi: 10.33998/mediasisfo.2019.13.2.713.
[16] Acker, S. V., Hausknecht, D., dan Sabelfeld, A. Measuring login webpage security," Apr. 2017, doi: 10.1145/3019612.3019798.
[17] Yuswandi, Y. Analisis Kerentanan Keamanan Pada Management Information System USAID SEA-PROJECT Menggunakan Metode OWASP. Jurnal Ilmiah Komputasi, vol. 19, no. 4, Dec. 2020, doi: 10.32409/jikstik.19.4.355.
[18] Nisa, F., Nurfebruary, N. S., dan Ikhwani, M. Analisis Keamanan Sistem Informasi Website Portal Akademik Universitas Malikussaleh Menggunakan OWASP ZAP. Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI), vol. 7, no. 6, Dec. 2024, doi: 10.32672/jnkti.v7i6.8345.
[19] Varghese, N., dan Sinha, R. Can Commercial Testing Automation Tools Work for IoT? A Case Study of Selenium and Node-Red. in IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society, Oct. 2020, p. 4519. doi: 10.1109/iecon43393.2020.9254910.
[20] Top *Web* Application Security. Accessed: Jul. 21, 2025. [Online]. Available: https://net.safe.security/assets/img/research-paper/pdf/*web*-application-security-threats.pdf
[21] Ezenwoye, O., dan Liu, Y. Web Application Weakness Ontology Based on Vulnerability Data. arXiv (Cornell University), Jan. 2022, doi: 10.48550/arxiv.2209.08067.
[22] Ferdiansyah, D., Isnanto, R. R., dan Suseno, J. E. Organizational indicators on startup software for implementing secure software development lifecycle (SSDL): A systematic literature review. THE 6TH INTERNATIONAL CONFERENCE ON ENERGY, ENVIRONMENT, EPIDEMIOLOGY AND INFORMATION SYSTEM (ICENIS) 2021: Topic of Energy, Environment, Epidemiology, and Information System, p. 50010, Jan. 2023, [Online]. Available: https://www.semanticscholar.org/paper/cb0adf6bce53ddcba782ba5f5a82517d73899a84