Jurnal Penelitian Teknologi Komputer
Vol. 10 No. 2, Februari 2026
ISSN: 21102135

EVALUASI KEAMANAN WEBSITE MENGGUNAKAN OWASP PADA MTSN KOTA PALOPO

Ihsa Amalia[1], Alim Surya Saruman[2]
Universitas Cokroaminoto Palopo
E-mail: ihsaamalia8@gmail.com[1], alim.suryasr@gmail.com[2]

Abstrak

Penelitian ini bertujuan untuk mengevaluasi keamanan website tersebut menggunakan standar OWASP dengan pendekatan kualitatif. Pengumpulan data dilakukan melalui observasi, wawancara, dan pengujian menggunakan OWASP ZAP. Hasil evaluasi menunjukkan adanya beberapa kerentanan, seperti SQL injection, hash disclosure, path traversal, serta konfigurasi keamanan yang lemah, seperti header Keamanan yang tidak diterapkan, Timestamp Disclosure, dan CSP tidak lengkap. Temuan ini menunjukkan bahwa website masih rentan terhadap serangan dan memerlukan perbaikan sistem keamanan.

Kata Kunci — Keamanan Website, OWASP, Kerentanan, MTsN Kota Palopo.

Abstract

This study aims to evaluate the security of the website using the OWASP standard using a qualitative approach. Data collection was conducted through observation, interviews, and testing using OWASP ZAP. The evaluation results revealed several vulnerabilities, such as SQL injection, hash disclosure, and path traversal, as well as weak security configurations, such as unimplemented security headers, timestamp disclosure, and incomplete CSP. These findings indicate that the website remains vulnerable to attacks and requires security system improvements.

Keywords *— Website Security, OWASP, Vulnerability, MTsN Kota Palopo.*

1. PENDAHULUAN

Perkembangan teknologi informasi yang pesat telah membawa banyak manfaat, terutama dalam dunia pendidikan. Institusi pendidikan seperti sekolah, termasuk MTsN Kota Palopo, kini menggunakan website sebagai salah satu media utama untuk memberikan informasi kepada guru, dan staf sekolah. Website sekolah menjadi sarana penting dalam menyampaikan informasi mengenai kegiatan sekolah, jadwal pelajaran, informasi akademik, hingga layanan administrasi. Namun, dengan semakin luasnya penggunaan website ini, ancaman keamanan cyber juga turut meningkat. Keamanan website merupakan salah satu aspek yang sangat penting dalam dunia teknologi informasi saat ini. Dengan meningkatnya ketergantungan terhadap system digital, ancaman keamanan terhadap website, seperti serangan cyber, pencurian data, dan peretasan, menjadi semakin kompleks dan sering terjadi. Hal ini juga berlaku pada institusi pendidikan, termasuk Madrasah Tsanawiyah Negeri (MTsN) Kota Palopo, yang kini banyak menggunakan website sebagai sarana untuk mendukung kegiatan administrasi, informasi, dan komunikasi. Website resmi MTsN Kota Palopo, yang dapat diakses di www.mtsnkotapalopo.sch.id, berfungsi sebagai sarana informasi utama bagi siswa, orang tua, serta masyarakat umum. Situs ini memuat berbagai informasi penting seperti pengumuman, jadwal kegiatan, dan layanan edukasi yang dapat diakses kapan saja.

Website yang tidak dilindungi dengan baik rentan terhadap berbagai ancaman, seperti peretasan, pencurian data, dan serangan lainnya. Dalam konteks pendidikan, serangan terhadap website sekolah tidak hanya mengganggu akses informasi, tetapi juga bisa berdampak pada kepercayaan system terhadap keamanan data siswa dan guru. Oleh karena itu, evaluasi keamanan website menjadi langkah penting dalam menjaga integritas dan kepercayaan masyarakat terhadap institusi pendidikan.

Metode open web application security project (OWASP) adalah salah satu standar yang

banyak digunakan dalam mengevaluasi keamanan aplikasi web. OWASP menyediakan serangkaian panduan dan alat untuk mengidentifikasi, menganalisis, serta memperbaiki kerentanan pada aplikasi web. Menggunakan metode ini, diharapkan dapat dilakukan identifikasi terhadap celah-celah keamanan pada website MTsN Kota Palopo sehingga pihak sekolah dapat mengambil tindakan pencegahan atau perbaikan yang diperlukan. Open web application security project (OWASP) merupakan organisasi non profit berfokus pada peningkatan keamanan perangkat lunak.

2. METODE PENELITIAN

Metodologi penelitian yang digunakan dalam penelitian ini adalah *open web application security project* (*OWASP*) yang bertujuan untuk menguji keamanan sistem dengan melakukan penetration testing dengan menggunakan model Black- Box Testing pada *website* MTsN Kota Palopo.

Evaluasi celah keamanan dengan metode *OWASP* merupakan metode penelitian kualitatif karena lebih fokus pada pemahaman mendalam tentang celah keamanan, eksplorasi potensi risiko, dan deskripsi detail dari hasil penelitian. *Penetration testing* dilakukan dengan aplikasi otomatisasi *OWASPZap*. Kemudian dari hasil pengujian akan dilakukan identifikasi agar dapat merumuskan evaluasi keamanan pada *website* yang mengacu pada standarisasi *OWASP* TOP 10.

Gambar 1. Diagram Alur Metodologi *OWASP*

Menentukan Tujuan dan *Scope*

Langkah awal dalam metodologi penelitian ini adalah menentukan tujuan serta ruang lingkup (*scope*) penelitian. Tujuan penelitian berfungsi sebagai arah utama dalam pelaksanaan penelitian, sedangkan *scope* menetapkan batasan agar penelitian tetap fokus dan terarah.

Pemindaian Awal

Tahap pemindaian awal merupakan langkah penting dalam proses evaluasi keamanan *website*. Pada tahap ini dilakukan proses identifikasi terhadap *host*, layanan, serta *port* yang terbuka untuk mengetahui potensi celah yang dapat dimanfaatkan oleh pihak yang tidak berwenang. Pemindaian ini bertujuan untuk memperoleh gambaran menyeluruh mengenai kondisi awal sistem sebelum dilakukan pengujian lebih lanjut.

Eksekusi Proses Pengujian

Pengujian yang akan dilakukan berfungsi sebagai masukan untuk menilai risiko celah keamanan. Setiap pengujian yang menghasilkan kerentanan akan diidentifikasi, dianalisis, dan ditentukan rekomendasi evaluasi risiko celah keamanannya.

Identifikasi Celah Keamanan

Setiap kerentanan yang ditemukan di identifikasi berdasarkan deskripsi celah keamanan, risiko celah keamanan, dan dampaknya terhadap aplikasi. Segala sesuatu yang ditimbulkan oleh kerentanan yang ditemukan akan dijabarkan menjadi bahan pertimbangan ketika menganalisis celah keamanan.

Evaluasi dan Rekomendasi Celah Keamanan

1. Langkah terakhir adalah menentukan evaluasi dan rekomendasi risiko kerentanan apa harus dilakukan. Rekomendasi tindakan diberikan untuk meminimalkan celah keamanan yang dihasilkan. Nilai *severity* (keparahan) yang dihasilkan untuk setiap kerentanan kemudian diurutkan untuk menemukan prioritas yang harus diperbaiki terlebih.

3. HASIL DAN PEMBAHASAN

1. Menentukan Tujuan dan *Scope*

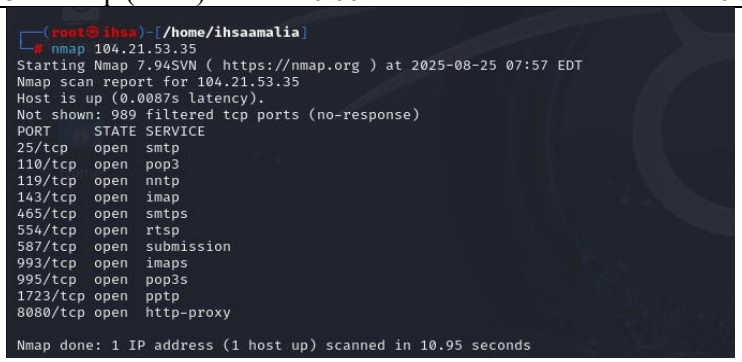
Tujuan dari penelitian ini adalah untuk mengetahui sejauh mana tingkat keamanan *website* MTsN Kota Palopo serta mengidentifikasi potensi kerentanan yang mungkin ada dengan menggunakan standar *OWASP* sebagai acuan. Adapun ruang lingkup pengujian difokuskan pada *website* resmi MTsN Kota Palopo yang dapat diakses secara publik. Pengujian meliputi halaman utama, form input, serta beberapa fitur interaktif yang digunakan oleh pengguna. Batasan ini ditetapkan agar proses evaluasi berjalan terarah, tidak mengganggu layanan lain di luar objek penelitian, serta tetap sesuai dengan etika pengujian keamanan. Dengan adanya penentuan tujuan dan *scope* ini, penelitian memiliki acuan yang jelas dalam pelaksanaan tahap selanjutnya, yaitu pemindaian awal dan pengujian kerentanan pada *website* yang menjadi objek penelitian.

2. Pemindaian Awal

Gambar 2. Hasil Pemindaian *Host*

Tabel 1. Hasil Pemindaian Awal dengan *Nmap -sn*

Target IP	Status *Host*	Latency (ms)	Jumlah *Host* Aktif	Waktu Pemindaian
104.21.53.35	Up (Aktif)	0.0012	1 dari 1	0.30 detik



Gambar 3. Hasil Pemindaian *Port* Umum

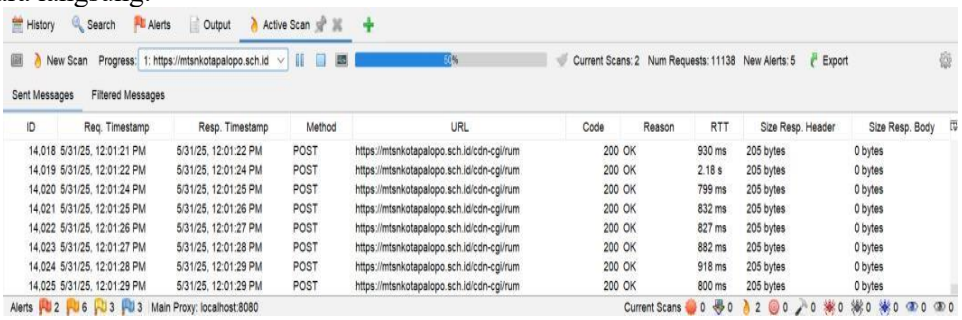
Tabel 2. Hasil Pemindaian *Port* dengan *Nmap*

Port	Status	Layanan	Keterangan
25/*tcp*	*Open*	*SMTP*	Layanan pengiriman email
110/*tcp*	*Open*	*POP3*	Layanan penerimaan email
119/*tcp*	*Open*	*NNTP*	*Network News Transfer Protocol*
143/*tcp*	*Open*	*IMAP*	Layanan email berbasis *IMAP*
465/*tcp*	*Open*	*SMTPS*	*SMTP* dengan *enkripsi SSL/TLS*
554/*tcp*	*Open*	*RTSP*	*Real Time Streaming Protocol*
587/*tcp*	*Open*	*Submission*	*SMTP Submission* untuk *client email*
993/*tcp*	*Open*	*IMAPS*	*IMAP* dengan *enkripsi* SSL/TLS
995/*tcp*	*Open*	*POP3S*	*POP3* dengan *enkripsi* SSL/TLS
1723/*tcp*	*Open*	*PPTP*	VPN berbasis *Point-to-Point Tunneling*
8080/*tcp*	*Open*	*HTTP Proxy*	Layanan *web/proxy* alternatif

3. Eksekusi Proses Pengujian

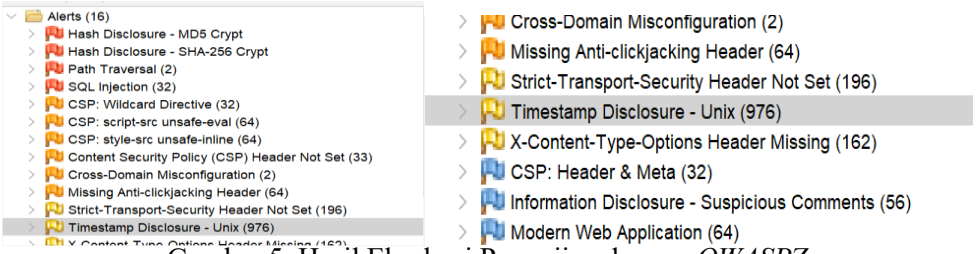
a. Manual *Explore*

Melalui pendekatan *manual explore*, pengujian dapat dilakukan dengan berinteraksi langsung pada aplikasi *web* yang sedang diuji. Pengguna dapat mengirim permintaan langsung, mengeksplorisasi beragam fitur dan *fungsionalitas*, serta menganalisis respon yang diterima dari server secara langsung.



Gambar 4. *Progres Massage Active Scan*

4. Identifikasi Celah Keamanan



Gambar 5. Hasil Eksekusi Pengujian dengan *OWASPZap*

Terlihat pada gambar 5. Bahwa hasil pemindaian menunjukkan adanya 16 jenis peringatan keamanan (*alerts*) yang ditemukan pada aplikasi atau sistem yang diuji. Beberapa di antaranya memiliki tingkat risiko tinggi, seperti:

a. *Hash Disclosure – MD5 Crypt* dan *SHA-256 Crypt*

Menunjukkan bahwa *hash* dari *password* atau data sensitif terungkap, yang bisa dimanfaatkan oleh penyerang untuk mencoba membobol sistem (2 kasus).

b. *Path Traversal* (2)

Sebanyak 2 potensi celah *path traversal* ditemukan. Ini merupakan kerentanan yang memungkinkan penyerang mengakses file atau direktori di luar *root web* melalui manipulasi parameter *URL*.

c. *SQL Injection* (32)

Ditemukan 32 titik yang rentan terhadap serangan *SQL Injection*. Ini adalah kerentanan serius yang memungkinkan penyerang mengeksekusi perintah *SQL* berbahaya untuk mencuri, mengubah, atau menghapus data dari database.

d. *CSP: Wildcard Directive* (32)

Ada 32 kasus penggunaan *wildcard* (*) dalam *direktif Content Security Policy (CSP), seperti script-src* *, yang terlalu longgar dan dapat memungkinkan pemuatan skrip dari sumber tidak tepercaya.

e. *CSP: script-src unsafe-eval* (64)

Sebanyak 64 temuan penggunaan *unsafe-eval* dalam *script-src*, yang memungkinkan eksekusi kode dinamis dan berpotensi disalahgunakan oleh penyerang untuk menjalankan skrip berbahaya.

f. *CSP: style-src unsafe-inline* (64)

Ditemukan 64 penggunaan unsafe-inline pada gaya *CSS*, yang memungkinkan penyisipan gaya langsung dalam *HTML*. Ini dapat dimanfaatkan untuk serangan injeksi jika tidak dikontrol.

g. *Content Security Policy (CSP) Header Not Set* (33)

Ada 33 permintaan *HTTP* yang tidak menyertakan *header CSP*. Tanpa *CSP*, tidak ada pembatasan terhadap sumber daya *eksternal* (skrip, gambar, dll), sehingga meningkatkan risiko *XSS*.

h. *Cross-Domain Misconfiguration* (2)

Terdapat 2 konfigurasi lintas domain yang salah, misalnya penggunaan *header Access-Control-Allow-Origin*: (*) yang terlalu longgar, sehingga memungkinkan situs lain mengakses sumber daya secara bebas.

i. *Missing Anti-clickjacking Header* (64)

Sebanyak 64 halaman tidak memiliki *header* anti-*clickjacking seperti X- Frame-Options*. Tanpa *header* ini, situs bisa dimasukkan ke dalam *iframe* oleh penyerang untuk melakukan *clickjacking*.

j. *Strict-Transport-Security Header Not Set* (196)

Ada 196 permintaan tanpa *header Strict-Transport-Security (HSTS). Header* ini mencegah *downgrade* dari *HTTPS* ke *HTTP* dan penting untuk menjaga koneksi tetap aman.

k. *Timestamp Disclosure - Unix* (976)

Terungkap 976 informasi *timestamp* dalam format *unix* (jumlah detik sejak 1 Januari 1970). Informasi ini bisa digunakan oleh penyerang untuk mengidentifikasi versi sistem, waktu *build*, atau aktivitas sistem.

l. *X-Content-Type-Options Header Missing* (162)

Ditemukan 162 permintaan tanpa *header X-Content-Type-Options: nosniff.* Tanpa *header* ini, *browser* dapat menebak jenis konten, sehingga meningkatkan risiko serangan *MIME sniffing*.

m. *CSP: Header & Meta* (32)

Sebanyak 32 kasus yang terkait dengan *CSP* (*Content Security Policy*) melalui tag *<meta>* atau *header HTTP*, yang mungkin tidak disetel dengan benar atau terlalu permisif.

n. *Information Disclosure - Suspicious Comments* (56)

Ada 56 komentar mencurigakan dalam kode *HTML* atau *JavaScript* yang mungkin mengandung informasi sensitif seperti petunjuk pengembangan, *URL* admin, atau konfigurasi sistem yang tidak seharusnya dipublikasikan.

o. *Modern Web Application* (64)

Terdapat 64 elemen atau fitur yang diklasifikasikan sebagai karakteristik aplikasi *web* modern. Ini bukan kerentanan, melainkan penanda untuk informasi, biasanya mencakup penggunaan *framework* modern (seperti *React, Angular*, dll).

Setelah proses *scanning* selesai, *OWASPZap* akan menghasilkan laporan yang berisi daftar kerentanan yang ditemukan dan tingkat keparahannya. Tingkat 39 keparahan ditentukan berdasarkan standar *OWASP Top* 10, yang merupakan daftar sepuluh celah keamanan *web* paling umum dan berbahaya. Untuk menentukan tingkat keparahan, *OWASP Zap* mempertimbangkan kompleksitas kerentanan, dampak potensial, kemungkinan *eksploitasi*, dan tingkat akses yang diperlukan untuk mengeksploitasi kerentanan tersebut. Setiap kerentanan akan diberi tingkat keparahan yang sesuai, yaitu rendah, sedang, atau tinggi, berdasarkan hasil evaluasi tersebut.

Tingkat Resiko	Jumlah Kerentanan
Tinggi	4
Sedang	6
Lemah	3
Informasi	3

Gambar 6. *Summary Of Alert* dari Hasil Eksekusi Pengujian

Angka-angka tersebut diperoleh dari hasil *scanning* otomatis dengan tool *OWASP ZAP* terhadap *website* target (*website* MTsN Kota Palopo). *OWASP ZAP* akan melakukan eksplorasi dan pengujian terhadap halaman-halaman di *website*, kemudian mengelompokkan hasil temuannya berdasarkan tingkat risikonya.

5. Evaluasi dan Rekomendasi Celah Keamanan

Langkah selanjutnya adalah melakukan evaluasi dan rekomendasi celah keamanan. Pada tahap ini, dilakukan evaluasi mendalam terhadap celah keamanan yang telah diidentifikasi sebelumnya. Evaluasi mencakup pengklasifikasian tipe kerentanan yang ditemukan, seperti kerentanan pada aplikasi *web*. Setiap kerentanan dinilai berdasarkan level risikonya, yang biasanya dikategorikan menjadi rendah, sedang, dan tinggi, dengan mempertimbangkan kemungkinan eksploitasi dan dampak terhadap sistem.

Vulnerability Type	Level	*Rekomendation*
Hash – Disclosure	*Medium*	Mengganti algoritma hashing menjadi algoritma

– MD5 Crypt		yang lebih aman seperti *bcrypt*, *Argon2*, atau setidaknya *SHA-256* dengan pengamanan tambahan seperti *salt* dan *pepper*. Selain itu, sistem harus memastikan bahwa *hash* tidak pernah ditampilkan secara publik baik melalui pesan *error*, *debug log*, atau *endpoint API*
Hash – Disclosure – SHA-256 Crypt	*Medium*	Memastikan semua hash dilindungi dengan *salt* unik per pengguna, serta menambahkan iterasi (misalnya menggunakan PBKDF2 atau *bcrypt* dengan *cost factor* tinggi). Selain itu, seperti pada kasus *MD5*, *hash* tidak boleh diungkap ke publik dalam bentuk apapun
Path Traversal	*Medium*	Melakukan validasi dan normalisasi input *path* dari pengguna untuk memastikan mereka tidak mengandung elemen *traversal direktori*. Sebaiknya gunakan *whitelist* file atau *direktori* yang memang boleh diakses, serta batasi akses file ke *direktori* tertentu saja. Selain itu, sistem operasi juga harus dikonfigurasi agar hak akses file dibatasi sesuai kebutuhan aplikasi (prinsip *least privilege*).
SQL Injection – SQLite	*Medium*	Salah satu rekomendasi utama adalah menggunakan *prepared statements* atau *parameterized queries* dalam setiap *query* yang melibatkan input dari pengguna. Dengan cara ini, *SQLite* akan memisahkan antara perintah *SQL* dan data, sehingga mencegah manipulasi struktur *query* oleh penyerang. Selain itu, disarankan untuk melakukan validasi dan sanitasi terhadap semua input pengguna, baik dari *form*, *URL*, maupun parameter lainnya, agar hanya data yang sesuai dengan format yang diharapkan.
CSP: Wildcard Directive	*High*	Hindari penggunaan *wildcard* (*) pada direktif penting seperti *sricpt-src* dan *style- src*. Sebaiknya tentukan sumber konten secara spesifik dan gunakan *self* untuk membatasi hanya dari domain sendiri. Hindari juga penggunaan '*unsafe-inline*' dan '*unsafe-eval*' karena sangat rentan terhadap *XSS*. Dengan kebijakan CSP yang ketat dan terkontrol, *website* menjadi lebih aman dari penyisipan konten berbahaya.
CSP: style-src unsafe-inline	*High*	Sebaiknya hindari penggunaannya karena memungkinkan eksekusi *JavaScript* dinamis yang rentan terhadap serangan *XSS*. Sebagai gantinya, gunakan *skrip* yang aman dan statis, serta pastikan *library* atau *framework* yang digunakan tidak memerlukan *'unsafe-eval'*. Dengan penghapus *'unsafe-eval'*, keamanan situs terhadap penyisipan skrip berbahaya akan meningkatkan secara signifikan.
CSP: style-src unsafe-inlane	*High*	Hindari penggunaan gaya *inline* karena memungkinkan penyisipan *CSS* berbahaya. Sebaiknya pindahkan semua *CSS* ke file

		eksternal dan gunakan *nonce* atau *hash* untuk mengizinkan gaya tertentu jika diperlukan. Dengan menghapus 'unsafe-inline', situs menjadi lebih aman dari serangan injeksi gaya berbahaya.
Content Security Policy (CSP) Header Not Set	*High*	Pastikan server selalu mengirim header c*ontent-security-policy* yang jelas. Minimal tetapkan *default-src self* lalu spesifikkan *script-src, style-src, img-src,* dan direktif lain hanya ke domain tepercaya (mis. *CDN* tertentu) serta hindari *'unsafe-inline'* dan *'unsafe-eval'*. Uji kebijakan di staging lebih dulu, pantau laporan pelanggaran (mode *Content- Security-Policy-Report-Only*)untuk melihat apa saja yang masih perlu diizinkan, lalu terapkan kebijakan final di produksi. Dengan begitu, *browser* mem*blokir* konten dari sumber berbahaya dan situs jauh lebih terlindungi meski sebelumnya *CSP header.*
Cross-Domain Misconfiguration	*Medium*	Batasi domain yang diizinkan dalam *header* seperti *access-control-allow-origin* hanya ke domain tepercaya, jangan gunakan *wildcard* * untuk data sensitif. Hindari mengizinkan *kredensial* (*credentials: true*) kecuali benar-benar diperlukan dan hanya untuk domain tertentu. Selalu validasi dan kontrol asal permintaan lintas domain agar tidak dieksploitasi untuk mencuri data atau melakukan serangan lintas situs.
Missing Anti-clickjacking Header	*Medium*	Tambahkan header "*x-frame-options*" dengan nilai "*DENY"* atau *"SAMEORIGIN"*, atau gunakan *CSP* dengan direktif *frame-ancestors self*. Ini akan mencegah halaman dimuat dalam *iframe* oleh situs lain, sehingga melindungi dari serangan *clickjacking*.
Strict-Transport-SecurityHeader Not Set	*High*	Tambahkan *header* "*strict-transport-security*" pada server dengan nilai seperti: *"Strict-Transport-Security:max- age=31536000 includeSubDomains; preload"*
Timestamp Disclosure – Unix	*Low*	Hindari menampilkan informasi waktu sistem (seperti *Unix timestamp*) kepada pengguna secara langsung, terutama di *URL, header,* atau *respons error*. Jika *timestamp* dibutuhkan, ubah tampilannya ke format yang tidak mengungkap sistem internal, atau *enkripsi* nilainya. Ini mencegah penyerang memanfaatkan waktu untuk analisis sistem atau pola serangan
CSP: Header & Meta	*High*	Pastikan *server* mengirimkan *header* "*Content-Security-Policy"* yang ketat dan sesuai kebutuhan, dan jika perlu, tambahkan tag *<meta> CSP* di *HTML* sebagai cadangan. Kebijakan ini membatasi sumber konten yang boleh dimuat sehingga mencegah serangan seperti *XSS* dan penyisipan konten berbahaya.

Vol. 10 No. 2, Februari 2026
ISSN: 21102135

4. KESIMPULAN

Berdasarkan hasil evaluasi keamanan website MTsN Kota Palopo dengan menggunakan metode OWASP, khususnya melalui tools OWASP ZAP, ditemukan sejumlah kerentanan yang bervariasi tingkat risikonya, mulai dari informational hingga high risk. Proses evaluasi dilakukan dengan tahapan identifikasi target, scanning otomatis dan manual, analisis hasil temuan, serta kategorisasi kerentanan sesuai dengan standar OWASP Top 10. Beberapa temuan signifikan mencakup potensi SQL injection, konfigurasi header HTTP yang lemah, serta kurangnya pengamanan pada input pengguna. Temuan-temuan ini menunjukkan bahwa website belum sepenuhnya memenuhi standar keamanan OWASP dan masih memiliki celah yang dapat dimanfaatkan oleh pihak tidak bertanggung jawab. Oleh karena itu, diperlukan langkah-langkah perbaikan dan penguatan sistem keamanan untuk meminimalisir risiko serangan siber di masa mendatang.

DAFTAR PUSTAKA

Arifin, A., & Pratama, R. D. (2021). Analisis Keamanan Website pada Sistem Informasi Akademik Menggunakan Metode Penetration Testing. Jurnal Teknologi dan Sistem Komputer, 9(2), 145–152.

Darul Fata (2023). Evaluasi resiko celah keamanan menggunakan metodelogi open web application security project (OWASP) pada aplikasi web sistem informasi akademik (SIAKAD) UIN AR-RANIRY

Darul Fata, (2023) Evaluasi Risiko Celah Keamanan Menggunakan Metodologi Open Web Application Security Project (OWASP) Pada Aplikasi Web Sistem Informasi Akademik (Siakad) UIN Ar- Raniry. Other thesis, Universitas Islam Negeri Ar-Raniry

Dede Ending Narhudin, Bambang Irawan, Agus Bahtiar (2024), Evaluasi Keamanan Website Menggunakan Metode OWASP: Penilaian Terhadap Serangan Injeksi SQL dan Cross-Site Scripting. JATI (Jurnal Mahasiswa Teknik Informatika).

Sutabri, T., Wijaya, A., Herdiansyah, M. I., & Negara, E. S. (2024). Evaluasi Risiko Celah Keamanan Aplikasi E-Office menggunakan Metode OWASP. Edumatic: Jurnal Pendidikan Informatika, 8(1), 113–122.

Y. Yudiana, A. Elanda, and R. L. Buana, Analisis Kualitas Keamanan Sistem Informasi E-Office Berbasis Website Pada STMIK Rosma Dengan Menggunakan OWASP Top 10," CESS (Journal Comput. 2021)

Zheng, Z., Zhang, Y., Wang, Y., & Liu, J. (2023). A deep learning-based hybrid intrusion detection system for industrial networks. Computers & Security.