Jurnal Teknologi Informasi dan Multimedia

p-ISSN : 2715-2529

e-ISSN : 2684-9151

https://journal.sekawan-org.id/index.php/jtim

Analisis Keamanan *Website* Sekolah MAN 1 Banyumas Menggunakan *Penetration Testing Execution Standard* (PTES)

Dewi Mardiana [1], Sigit Sugiyanto [1]*, Harjono [1], Ermadi Satria Wijaya [1]

[1] Program Studi Teknik Informatika, Universitas Muhammadiyah Purwokerto, Indonesia

* Korespondensi: : sigitsugiyanto@ump.ac.id

Sitasi: D. Mardiana, S. Sugiyanto, H. Harjono, and E. S. Wijaya, "Analisis Keamanan Website Sekolah MAN 1 Banyumas Menggunakan Penetration Testing Execution Standard (PTES)", *Jurnal Teknologi Informasi Dan Multimedia*, vol. 8, no. 3, pp. 660-680, 2026. https://doi.org/10.35746/jtim.v8i3.1102

Diterima: 01-07-2026
Direvisi: 23-07-2026
Disetujui: 31-07-2026

Abstract. Digital transformation is driving educational institutions to adopt web portals, yet cybersecurity evaluation is still often neglected. This study aims to identify and validate security vulnerabilities on the official website of MAN 1 Banyumas using the Penetration Testing Execution Standard (PTES) framework. Testing was conducted using a black-box penetration testing approach by combining Open Worldwide Application Security Project (OWASP ZAP) scanning and manual verification to reduce false positives. The results showed the discovery of 4 true positive vulnerabilities, 1 false positive, and medium-risk FTP and SSH ports. Validated vulnerabilities include Cross-Site Request Forgery (CSRF), Clickjacking, session management weaknesses, and Vulnerable JS Library, while Cross-Site Scripting (XSS) vulnerabilities were confirmed as false positives. Simulation of vulnerability chaining showed that the combination of these vulnerabilities has the potential to lead to account takeover. Therefore, server hardening and JavaScript library updates are recommended to improve system security.

Keywords: *Penetration Testing*, PTES, OWASP ZAP.

Abstrak. Transformasi digital mendorong lembaga pendidikan mengadopsi portal web, namun evaluasi keamanan siber masih sering diabaikan. Penelitian ini bertujuan mengidentifikasi dan memvalidasi kerentanan keamanan pada situs resmi MAN 1 Banyumas menggunakan kerangka kerja Penetration Testing Execution Standard (PTES). Pengujian dilakukan melalui pendekatan black-box penetration testing dengan mengombinasikan pemindaian Open Worldwide Application Security Project (OWASP ZAP) dan verifikasi manual untuk mengurangi false positive. Hasil penelitian menunjukkan ditemukannya 4 kerentanan true positive, 1 false positive, serta port FTP dan SSH berisiko menengah. Kerentanan yang tervalidasi meliputi Cross-Site Request Forgery (CSRF), Clickjacking, kelemahan manajemen sesi, dan Vulnerable JS Library, sedangkan kerentanan Cross-Site Scripting (XSS) dikonfirmasi sebagai false positive. Simulasi vulnerability chaining menunjukkan bahwa kombinasi kerentanan tersebut berpotensi menyebabkan account takeover. Oleh karena itu, direkomendasikan penerapan server hardening dan pembaruan pustaka JavaScript untuk meningkatkan keamanan sistem.

Kata kunci: *Penetration Testing*, PTES, OWASP ZAP.

1. Pendahuluan

Transformasi digital di sektor pendidikan mendorong lembaga sekolah untuk memanfaatkan situs web berbasis Content Management System (CMS) sebagai media utama penyampaian informasi akademik dan pelayanan administrasi[1]. Namun, kondisi

eksisting saat ini menunjukkan bahwa banyak sistem informasi sekolah di Indonesia masih belum memiliki mekanisme pengujian keamanan yang terstruktur dan berkesinambungan[2]. Permasalahan utama yang dihadapi institusi pendidikan adalah tingginya kerentanan aplikasi web akibat penyimpanan data sensitif yang tidak diimbangi dengan pengelolaan keamanan dan konfigurasi server yang optimal[3]. Urgensi penelitian ini semakin meningkat seiring dengan meningkatnya ancaman terhadap sistem berbasis web di Indonesia. Berdasarkan Laporan Tahunan Keamanan Siber Kominfo-CSIRT Tahun 2024, sepanjang tahun 2024 tercatat 27 insiden siber, dengan jenis insiden yang paling dominan berupa *Injected Malicious* File sebanyak 17 insiden. Selain itu, terdapat 214 permintaan *Vulnerability Assessment/Penetration Testing* (VA/PT) yang menunjukkan tingginya kebutuhan evaluasi keamanan pada sistem informasi. Kondisi tersebut menegaskan bahwa pengujian keamanan secara berkala diperlukan untuk mengidentifikasi dan memitigasi kerentanan sebelum dimanfaatkan oleh pihak yang tidak bertanggung jawab[4].

Sebagai solusi atas permasalahan tersebut, pengujian keamanan situs web dilakukan menggunakan kerangka kerja Penetration Testing Execution Standard (PTES) melalui pendekatan *Black Box Testing* [5]. Penelitian oleh Kurniawan et al [6]. menunjukkan bahwa penerapan PTES mampu mengidentifikasi berbagai kerentanan pada website melalui tahapan pengujian yang sistematis. Sedangkan penelitian Imtias at al [7]. memanfaatkan vulnerability scanner untuk mengidentifikasi kerentanan pada aplikasi web, namun belum melakukan validasi manual terhadap seluruh temuan. Berdasarkan penelitian-penelitian tersebut, masih terdapat kesenjangan penelitian berupa terbatasnya kajian keamanan pada portal akademik sekolah yang mengombinasikan pemindaian otomatis dengan validasi manual untuk membedakan temuan *true positive dan false positive*. Oleh karena itu, penelitian ini menerapkan PTES melalui pendekatan *Black Box Testing* yang dilengkapi dengan validasi manual dan *analisis vulnerability chaining* untuk memperoleh hasil identifikasi kerentanan yang lebih akurat.

Kontribusi utama dari penelitian ini terletak pada kedalaman tahap validasi eksploitasi celah keamanan serta penyusunan rekomendasi mitigasi konkret berbasis server hardening yang disesuaikan dengan arsitektur sistem sekolah. Adapun tujuan dari penelitian ini adalah untuk menganalisis tingkat kerentanan situs web resmi MAN 1 Banyumas secara proaktif dan menyajikan referensi teknis yang objektif guna memperkuat pertahanan infrastruktur digital sekolah.

Kajian ini difokuskan pada analisis tingkat keamanan situs web resmi MAN 1 Banyumas menggunakan kerangka kerja PTES melalui pendekatan *Black Box Testing*. Penelitian ini dilakukan sebagai langkah proaktif atas inisiatif peneliti guna mengevaluasi infrastruktur digital sekolah yang mengelola layanan akademik penting namun belum pernah melalui audit keamanan formal secara berkala. Berbeda dengan penelitian terdahulu yang mayoritas hanya berfokus pada pemindaian celah (*vulnerability scanning*), kajian ini terletak pada kedalaman tahap *eksploitasi* (*exploitation*) serta penyusunan rekomendasi mitigasi konkret berbasis *server hardening* yang disesuaikan dengan kondisi riil arsitektur sistem sekolah. Melalui identifikasi kerentanan, validasi *eksploitasi*, dan analisis dampak risiko yang disajikan, pengelola situs web diharapkan memperoleh referensi teknis yang objektif untuk memperkuat pertahanan sistem dan meminimalkan risiko serangan siber terhadap layanan akademik berbasis web secara efektif.

2. Bahan dan Metode

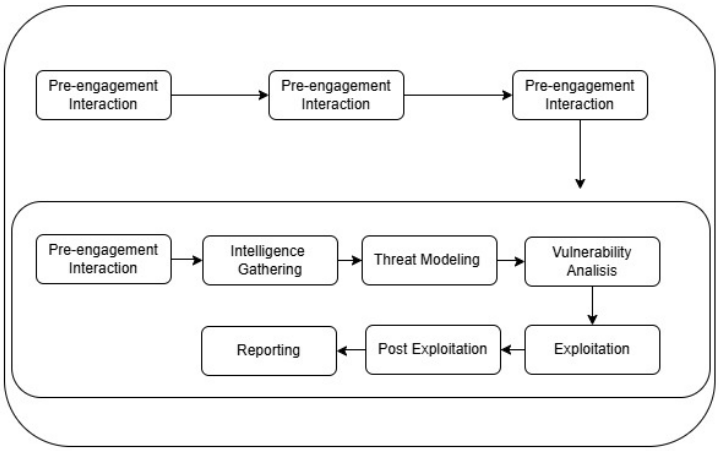
Penelitian ini menggunakan metode kualitatif deskriptif dengan pendekatan studi kasus pada website resmi MAN 1 Banyumas. Pengujian dilakukan menggunakan pendekatan Black Box Testing, yaitu pengujian tanpa akses terhadap kode sumber maupun konfigurasi internal sistem. Kerangka kerja yang digunakan adalah Penetration Testing Execution Standard (PTES) karena menyediakan tahapan pengujian keamanan

yang sistematis, mulai dari perencanaan hingga penyusunan laporan hasil pengujian. Pengujian difokuskan pada identifikasi kerentanan keamanan aplikasi web serta penyusunan rekomendasi mitigasi berdasarkan hasil pengujian.

Kerangka kerja yang digunakan dalam penelitian ini adalah PTES, yang terdiri atas tujuh tahapan terorganisir secara sistematis, meliputi *pre-engagement interaction*, *intelligence gathering*, *threat modeling*, *vulnerability analysis*, *exploitation*, *post exploitation*, dan *reporting*[8]. Setiap tahapan dalam kerangka kerja ini saling terhubung dan membentuk alur pengujian yang teratur, sehingga proses analisis keamanan tidak hanya terbatas pada penemuan kelemahan sistem, tetapi juga mencakup evaluasi terhadap potensi eksploitasi serta konsekuensi yang dapat ditimbulkan terhadap sistem yang menjadi target pengujian. Tahapan PTES Bisa dilihat pada gambar 1.

Gambar 1. Tahapan Metode PTES[9]

Guna memastikan penelitian berjalan secara sistematis dan terukur, tahapan pelaksanaan penelitian direpresentasikan ke dalam sebuah alur kerja (*workflow*) yang mengadopsi standar kerangka kerja PTES. Alur penelitian tersebut menggambarkan rangkaian transisi secara menyeluruh, mulai dari fase perencanaan, pelaksanaan teknis, hingga tahap pelaporan hasil pengujian keamanan sistem. Alur kerja bisa dilihat pada gambar 2.



Gambar 2. Alur Penelitian

Tahapan penelitian diawali dengan penentuan objek penelitian dan studi literatur mengenai PTES, OWASP, serta keamanan aplikasi web. Selanjutnya dilakukan pengumpulan data yang terdiri atas data primer dan data sekunder. Data primer diperoleh melalui pengujian langsung terhadap website menggunakan OWASP ZAP untuk pemindaian kerentanan, Nmap untuk identifikasi port dan layanan yang berjalan pada server,

serta *Developer Tools* pada peramban untuk *validasi* konfigurasi keamanan. Data sekunder diperoleh melalui teknik *Open Source Intelligence* (OSINT) menggunakan layanan *Whois* untuk memperoleh informasi *registrasi domain* dan infrastruktur server.

Pelaksanaan audit keamanan mengikuti tujuh tahapan Penetration Testing Execution Standard (PTES). Tahap *Pre-engagement Interactions* dilakukan dengan menetapkan ruang lingkup pengujian menggunakan pendekatan *Black Box Testing* dan prinsip *non-destructive testing*[10]. Pada tahap Intelligence Gathering, peneliti melakukan pengintaian pasif menggunakan Whois serta pengintaian aktif menggunakan Nmap dan fitur Spider pada OWASP ZAP untuk memperoleh informasi domain, port yang terbuka, layanan yang aktif, serta struktur URL dan endpoint website. Informasi tersebut kemudian digunakan pada tahap Threat Modeling untuk mengidentifikasi aset yang berpotensi menjadi target serangan, jenis ancaman yang mungkin terjadi, dan dampaknya terhadap sistem. [10].

Tahap *Vulnerability Analysis* dilakukan menggunakan fitur *Active Scan* pada OWASP ZAP untuk mengidentifikasi kerentanan keamanan pada website.[11]. Selanjutnya, seluruh temuan diverifikasi pada tahap Exploitation melalui pengujian manual menggunakan payload, inspeksi elemen HTML, pemeriksaan HTTP response header, dan analisis atribut cookie guna membedakan temuan true positive dan false positive [12]. Tahap Post-Exploitation dilakukan dengan menganalisis keterkaitan antar kerentanan (vulnerability chaining) tanpa melakukan perubahan terhadap sistem target. Tahap terakhir yaitu Reporting, dilakukan dengan menyusun hasil pengujian, menganalisis tingkat risiko setiap kerentanan, serta memberikan rekomendasi mitigasi berupa server hardening sesuai dengan hasil pengujian. [13].

3. Hasil

Pelaksanaan penelitian ini berjalan secara sistematis mengikuti alur kerja yang terstruktur, dimulai dari tahapan persiapan hingga penemuan data di lapangan. Alur pelaksanaan tersebut menjadi fondasi penting guna memastikan arah pengujian tetap konsisten dan sesuai dengan kaidah ilmiah. Kronologi alur pelaksanaan penelitian ini dijabarkan sebagai berikut

3.1. Penuntuan Topik

Tahapan awal pelaksanaan penelitian ini berhasil menetapkan fokus bahasan dan objek penelitian secara definitif, yaitu audit keamanan terhadap portal informasi akademik MAN 1 Banyumas dengan alamat domain utama https://man1banyumas.sch.id. Pemilihan topik dan target penelitian tersebut didasarkan pada signifikansi peran portal dalam mengelola data akademik sivitas sekolah, sehingga identifikasi celah keamanan secara dini dapat menjadi langkah pencegahan yang esensial guna melindungi sistem dari ancaman eksploitasi nyata di dunia siber.

3.2. Study Literatur

Pada tahap penerapan landasan teori, peneliti mengimplementasikan parameter dari dokumentasi OWASP Top 10 edisi terbaru serta kerangka kerja PTES yang telah dikaji sebelumnya. Hasil kajian pustaka tersebut digunakan secara langsung di lapangan untuk menyusun skenario interaksi *non-destruktif* (*Rules of Engagement*) serta merancang komponen kode injeksi *payload* yang valid guna menguji kerentanan sistem target selama fase eksploitasi berlangsung.

3.3. Pengumpulan data

Proses pengumpulan data di lapangan diselesaikan dengan menggabungkan dua pendekatan, yaitu data primer dan data sekunder. Pengumpulan data primer dilakukan melalui interaksi teknis secara langsung dengan peladen web MAN 1 Banyumas,

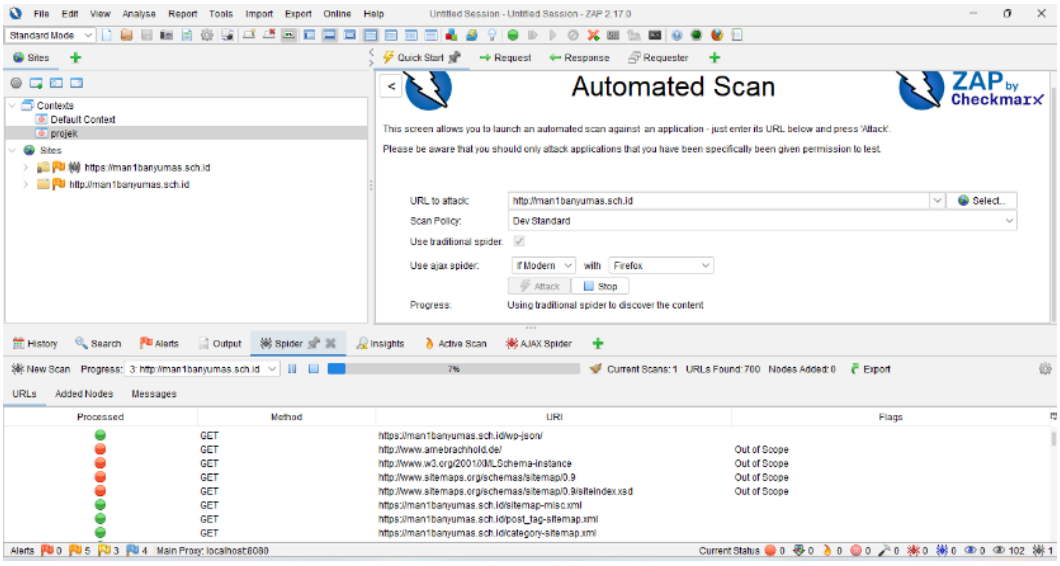
mencakup perekaman catatan lalu lintas jaringan (*HTTP Response Headers*) melalui peramban web, pemetaan alamat IP publik asli (103.31.251.68) beserta keterbukaan porta manajemen kritis seperti FTP (21) dan SSH (22) menggunakan utilitas Nmap, ekstraksi laporan otomatis daftar potensi kerentanan dari instrumen OWASP ZAP, serta dokumentasi skrip *payload Hypertext Markup Language* (HTML) yang dimasukan secara manual sebagai bukti konsep (*Proof of Concept*) untuk memvalidasi serangan *Clickjacking* dan CSRF. Sementara itu, pengumpulan data sekunder berhasil diperoleh tanpa memicu sistem pertahanan target melalui teknik *Open-Source Intelligence* (OSINT) menggunakan utilitas *Whois*, yang menghasilkan data arsitektur *Domain Name System* (DNS) berupa penggunaan lapisan perlindungan proksi pihak ketiga dari layanan *Name Server Cloudflare* (*dara.ns.cloudflare.com dan gordon.ns.cloudflare.com*).

3.4. *Pre-engagement interactions*

Pada tahap (*pre-engagement*), ditetapkan bahwa target spesifik dari evaluasi ini adalah aplikasi web portal MAN 1 Banyumas. Ruang lingkup pengujian difokuskan secara eksklusif pada lapisan aplikasi (antarmuka web) guna mengidentifikasi celah keamanan struktural seperti XSS, CSRF, dan kesalahan konfigurasi keamanan server. Sesuai dengan kesepakatan ketika uji penetrasi *non-destruktif*, pengujian ini secara tegas mengecualikan teknik rekayasa sosial (*social engineering*) terhadap staf sekolah, pengujian keamanan fisik, maupun peluncuran serangan *Denial of Service* (DoS). Hal tersebut diterapkan guna mencegah terjadinya gangguan ketersediaan layanan (*downtime*) pada sistem informasi akademik yang sedang beroperasi

3.5. *Intelligence Gathering*

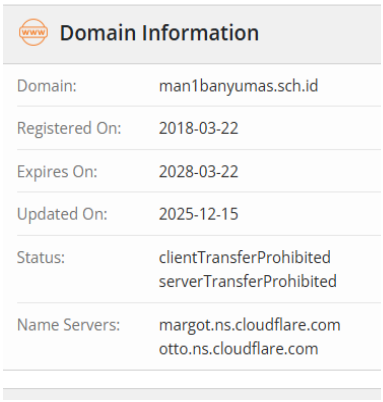
Proses pengumpulan informasi diawali dengan pemetaan struktur direktori dan tautan aplikasi web menggunakan fitur *Spidering* pada OWASP ZAP. Pemetaan pasif dan aktif ini bertujuan untuk mengekstrak URL (*Uniform Resource Locator*) beserta titik akhir (*endpoint*), yakni titik akses spesifik tempat pengguna berinteraksi dengan sistem tanpa memerlukan kredensial autentikasi. Hasil pemetaan tersebut kemudian direpresentasikan dalam bentuk pohon situs (*site tree*) yang memvisualisasikan keterkaitan antara setiap direktori secara hierarkis dari tingkat teratas hingga ke tingkat terdalam. Secara keseluruhan, himpunan URL dan *endpoint* yang terpetakan tersebut membentuk permukaan serangan (*attack surface*) yang merepresentasikan keseluruhan area pemaparan sistem, dan selanjutnya menjadi target utama pada tahap pemindaian berikutnya. Bisa dilihat pada gambar 3.



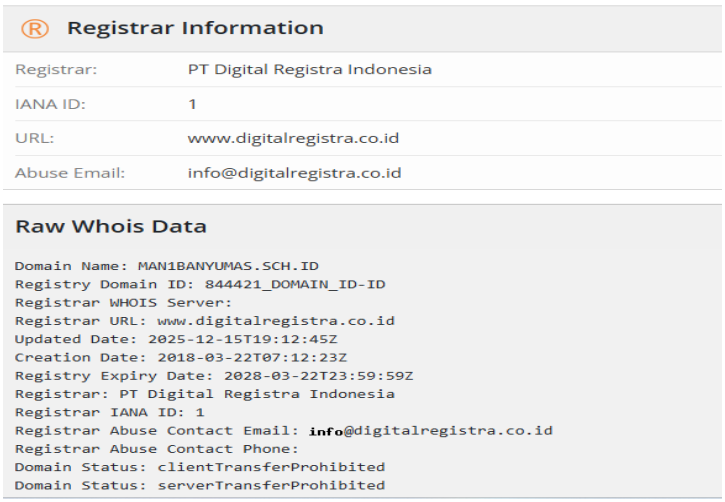
Gambar 3. Proses SpiderScan tools OWASPZap

3.5.1. Pengintaian Pasif (*Analisis Whois*)

Langkah pertama dilakukan secara pasif menggunakan utilitas pemindaian *Whois* terhadap domain man1banyumas.sch.id. Pemindaian ini bertujuan untuk mengekstraksi informasi publik terkait registrasi domain, *Name Server* (NS), dan alamat IP server, tanpa melakukan kontak langsung yang berpotensi memicu sistem pendeteksi intrusi (*Intrusion Detection System*/IDS) pada target. Hasil pengintaian bisa dilihat pada gambar 4 dan 5.



Gambar 4. Hasil Whois

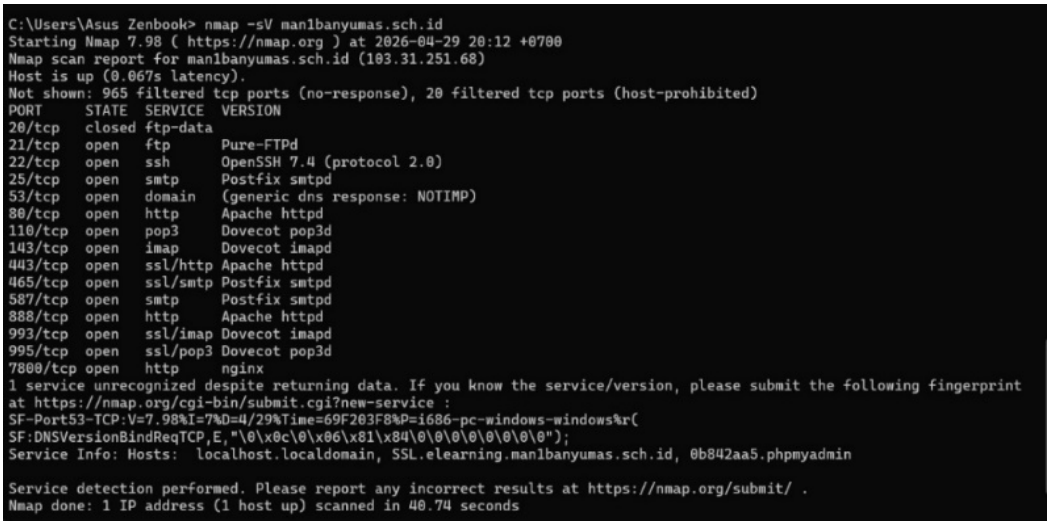


Gambar 5. Hasil Whois

Utilitas Whois digunakan untuk mengekstraksi informasi registrasi domain dan infrastruktur dasar server portal akademik MAN 1 Banyumas. Berdasarkan rekaman data yang diperoleh, sistem target menggunakan perlindungan jaringan dan layanan *Name Server* (NS) yang dialihkan menuju infrastruktur pihak ketiga, yaitu *cloudflare.com*. Penggunaan *Cloudflare* ini berfungsi sebagai proksi pembalik (reverse proxy) yang menyembunyikan alamat IP publik asli dari server web sekolah. Strategi penyamaran ini memberikan lapisan pertahanan awal yang efektif guna meminimalkan risiko pemindaian langsung dan serangan *Distributed Denial of Service* (DDoS) dari pihak luar.

3.5.2. Pengintaian Aktif (Pemindaian Port dengan Nmap)

Setelah memperoleh gambaran awal dari pengintaian pasif, langkah selanjutnya adalah melakukan pengintaian aktif menggunakan instrumen *Network Mapper* (Nmap). Pemindaian ini dieksekusi menggunakan perintah *nmap -sV* (*Service Versioning*) guna mencacah (*enumerasi*) porta jaringan yang terbuka (*open ports*) sekaligus mendeteksi versi perangkat lunak yang berjalan pada layanan tersebut. Bisa dilihat pada gambar 6.



Gambar 6. Hasil Nmap

Berdasarkan hasil pemindaian gambar 6, teridentifikasi bahwa server target dengan alamat IP 103.31.251.68 mengekspos beberapa port krusial ke publik. Selain port standar layanan web, yaitu port 80 (HTTP) dan 443 (HTTPS) yang dioperasikan oleh *server* Apache httpd, ditemukan pula keterbukaan layanan manajemen *server* seperti port 21 (FTP dengan Pure-FTPd) dan port 22 (SSH dengan OpenSSH 7.4). Keterbukaan port manajemen tersebut memperluas permukaan serangan (*attack surface*), sehingga mengharuskan administrator *server* untuk menerapkan kebijakan kata sandi yang kuat serta pembatasan akses berbasis IP (*IP Whitelisting*) guna mencegah potensi serangan *Brute Force*.

3.6. Threat Modeling

Berdasarkan topologi arsitektur yang berhasil dipetakan, tahapan selanjutnya adalah menyusun pemodelan ancaman guna mengidentifikasi aset-aset kritis yang berpotensi dieksploitasi oleh pihak eksternal. Pendekatan yang diterapkan adalah pemodelan berbasis aset (*Asset-Based Threat Modeling*), yang memetakan elemen fungsional aplikasi terhadap potensi vektor serangan serta probabilitas dampaknya terhadap sistem. Hasil pemodelan tersebut dirangkum dalam tabel 1.

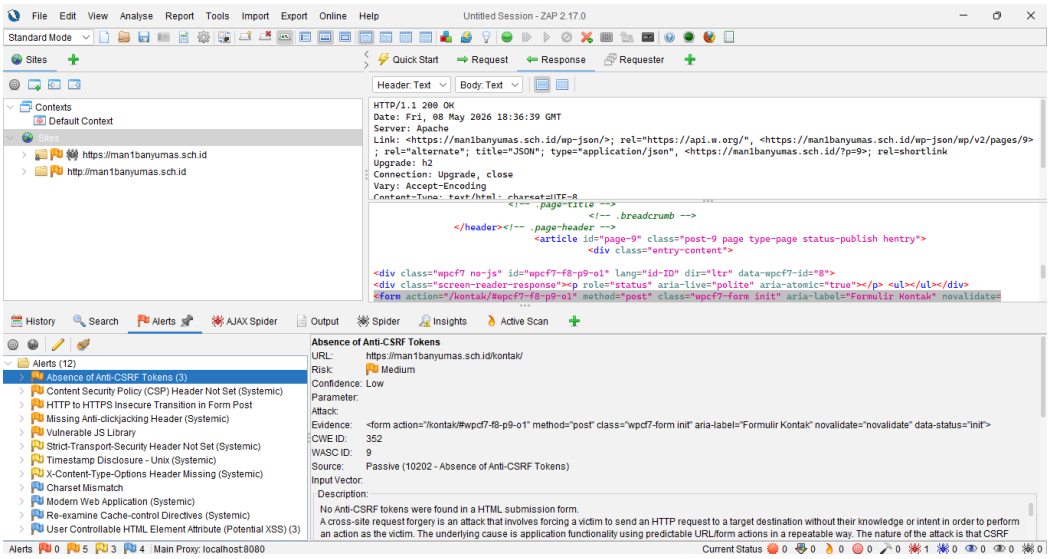
Tabel 1. Pemodelan ancaman

No	Aset/Komponen Target	Potensi Ancaman/Serangan	Dampak Potensial
1.	Formulir Input & Transmisi Data (misal: halaman /kontak atau *login*)	*Cross-Site Request Forgery* (CSRF)	Memaksa peramban pengguna sah (seperti admin atau guru) untuk mengirimkan data atau melakukan tindakan ilegal di dalam sistem tanpa disadari oleh korban.
2.	Antarmuka Keseluruhan Halaman Web	*Clickjacking* / *UI Redressing* (Ketiadaan *Anti-clickjacking Header*)	Pengguna tertipu untuk mengeklik tombol atau tautan tersembunyi, karena halaman web sekolah berhasil disembunyikan di balik *iframe* situs palsu milik penyerang.
3.	Manajemen Sesi & Autentikasi Pengguna	*Session Hijacking* (Ketiadaan *Flag HttpOnly* dan *Secure* pada *Cookie*)	Tereksposnya token *cookie* di jaringan yang memungkinkan peretas mencegat dan mengambil alih sesi *login* pengguna secara paksa.
4.	Modul Konten Web & Rendering JavaScript	*Cross-Site Scripting* (XSS) / Injeksi Data (Ketiadaan *Content Security Policy* / *CSP*)	Penyerang dapat menyisipkan skrip berbahaya ke dalam halaman web untuk mencuri data pribadi pengunjung, mengarahkan lalu lintas ke situs *phishing*, atau mengubah tampilan (*defacement*).

No	Aset/Komponen Target	Potensi Ancaman/Serangan	Dampak Potensial
5.	Pustaka JavaScript Pihak Ketiga (Client-side Scripts)	Eksploitasi Komponen Rentan (Vulnerable JS Library / jQuery Migrate 3.4.1)	Pemanfaatan fungsi-fungsi usang atau rentan (*legacy functions*) yang dibawa oleh pustaka pembantu eksternal. Jika terintegrasi dengan kode utama yang tidak disanitasi, penyerang dapat mengeksploitasi celah turunan seperti *Client-side Cross-Site Scripting* (DOM-XSS) atau gangguan fungsi

3.7. *Vulnerability Analysis*

Berdasarkan parameter ancaman yang telah dimodelkan, fase selanjutnya adalah melakukan pemindaian aktif (*active scanning*) menggunakan instrumen OWASP ZAP. Pemindaian ini dilakukan dengan menyisipkan berbagai variasi *payload* standar ke setiap titik akhir (*endpoint*) yang telah dipetakan, guna mengidentifikasi celah keamanan pada konfigurasi *server* maupun logika aplikasi web target. Hasil pemindaian *Alert* bisa dilihat pada gambar 7.



Gambar 7. Proses AlertsScan tools OWASPZap

Hasil pemindaian otomatis mendeteksi adanya sejumlah peringatan (*alerts*) yang mengindikasikan ketiadaan mekanisme keamanan fundamental pada arsitektur web target. Rincian hasil identifikasi kerentanan tersebut direpresentasikan pada tabel 2 :

Tabel 2. Alert

Jenis Kerentanan (Alert Type)	Tingkat Risiko (Risk)	Jumlah Temuan (Count)
Absence of Anti-CSRF Tokens	Medium (Oranye)	3
Content Security Policy (CSP) Header Not Set	Medium (Oranye)	Systemic (Menyeluruh)
HTTP to HTTPS Insecure Transition in Form Post	Medium (Oranye)	1
Missing Anti-clickjacking Header	Medium (Oranye)	Systemic (Menyeluruh)
Vulnerable JS Library	Medium (Oranye)	1
Strict-Transport-Security Header Not Set	Low (Kuning)	Systemic (Menyeluruh)
Timestamp Disclosure - Unix	Low (Kuning)	Systemic (Menyeluruh)
X-Content-Type-Options Header Missing	Low (Kuning)	Systemic (Menyeluruh)
Charset Mismatch	Informational (Biru	1

Jenis Kerentanan (Alert Type)	Tingkat Risiko (Risk)	Jumlah Temuan (Count)
Modern Web Application	Informational (Biru	Systemic (Menyeluruh)
Re-examineCache-control Directives	Informational (Biru)	Systemic (Menyeluruh)
User Controllable HTML Element Attribute (Potential XSS)	Informational (Biru)	3

3.8. *Exploitation*

Pemindai kerentanan otomatis (*automated scanner*) memiliki kelemahan inheren berupa potensi pelaporan positif semu (*false positive*). Oleh karena itu, fase eksploitasi dalam penelitian ini diimplementasikan sebagai langkah validasi manual (*Proof of Concept*) guna membuktikan apakah peringatan kerentanan yang terdeteksi pada fase sebelumnya benar-benar dapat dieksploitasi dalam skenario serangan nyata. Validasi dilakukan terhadap empat temuan kerentanan paling kritis berikut:

3.8.1. Validasi Kerentanan *Clickjaking* (*IU Redresing*)

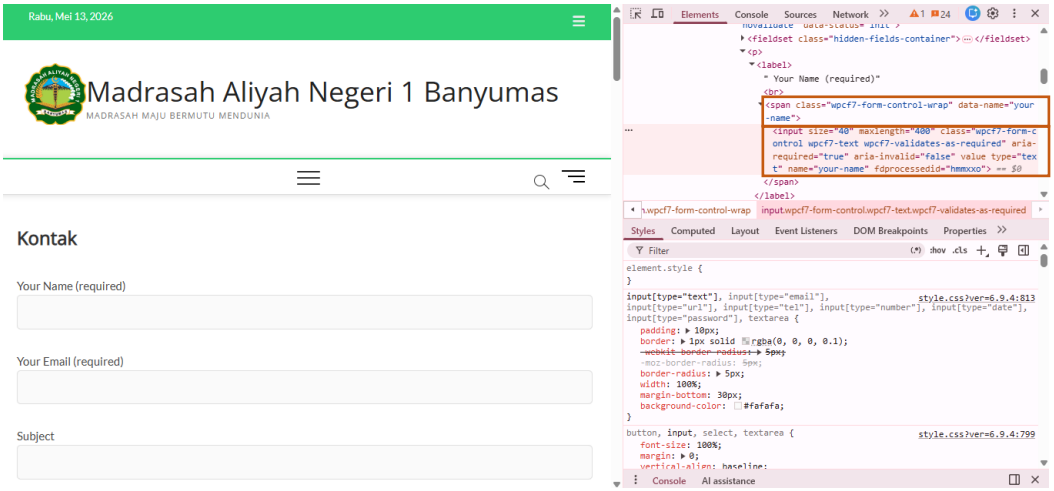
Hasil pemindaian mengindikasikan ketiadaan tajuk respons (*response header) X-Frame-Options* pada server target. Guna memvalidasi kerentanan tersebut, peneliti merancang sebuah skrip jebakan berbasis HTML yang memanfaatkan elemen *iframe* untuk memuat antarmuka web portal MAN 1 Banyumas di dalam domain milik penyerang. Hasil validasi bisa dilihat pada gambar 8.

Gambar 8. Hasil Clickjacking

Hasil pengujian manual menunjukkan bahwa peramban web mengeksekusi *iframe* tersebut tanpa penolakan. Hal ini mengonfirmasi status *true positive*, di mana sistem secara faktual rentan terhadap teknik *Clickjacking*. Celah keamanan ini berpotensi dimanfaatkan oleh penyerang untuk mengelabui pengguna sah, seperti administrator, agar mengeklik elemen antarmuka tersembunyi yang telah dimanipulasi sebelumnya.

3.8.2. Validasi Kerentanan *Croses-Site Request Forgery* (CSRF)

Sistem deteksi melaporkan ketiadaan token Anti-CSRF pada formulir aplikasi web target. Pembuktian dilakukan melalui inspeksi elemen (*Inspect Element*) terhadap struktur kode sumber antarmuka formulir kontak publik guna memverifikasi ada atau tidaknya mekanisme perlindungan CSRF pada formulir tersebut. Hasil validasi bisa dilihat pada gambar 9.

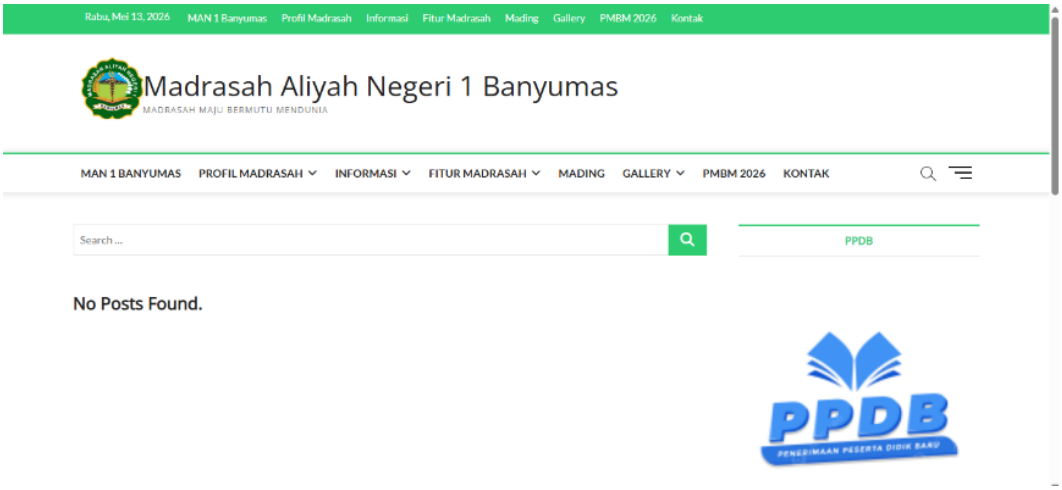


Gambar 9. Hasil CSRF

Berdasarkan inspeksi manual, terbukti bahwa hierarki tag <form> dan <input> tidak menyertakan atribut *type="hidden"* yang memuat token autentikasi acak (*nonce*). Ketiadaan parameter tersebut memvalidasi kerentanan ini sebagai *true positive*, yang membuka vektor serangan di mana peramban pengguna dapat dipaksa untuk mengeksekusi tindakan tidak sah tanpa disadari pada saat sesi masuk (*login*) yang sah sedang berlangsung.

3.8.3. Validasi False Positive pada Cross-site Scripting (XSS)

Pemindaian arsitektur melaporkan ketiadaan *Content Security Policy* (CSP) pada sistem target, yang secara teoretis membuka peluang bagi eksekusi skrip lintas situs (*Cross-Site Scripting*/XSS). Namun demikian, validasi manual yang dilakukan dengan memasukan *payload* <script>alert('Celah XSS Dian')</script> pada parameter fungsi pencarian (*/search*) menghasilkan temuan yang berbeda dari perkiraan awal. Hasil validasi ditunjukan pada gambar 10.



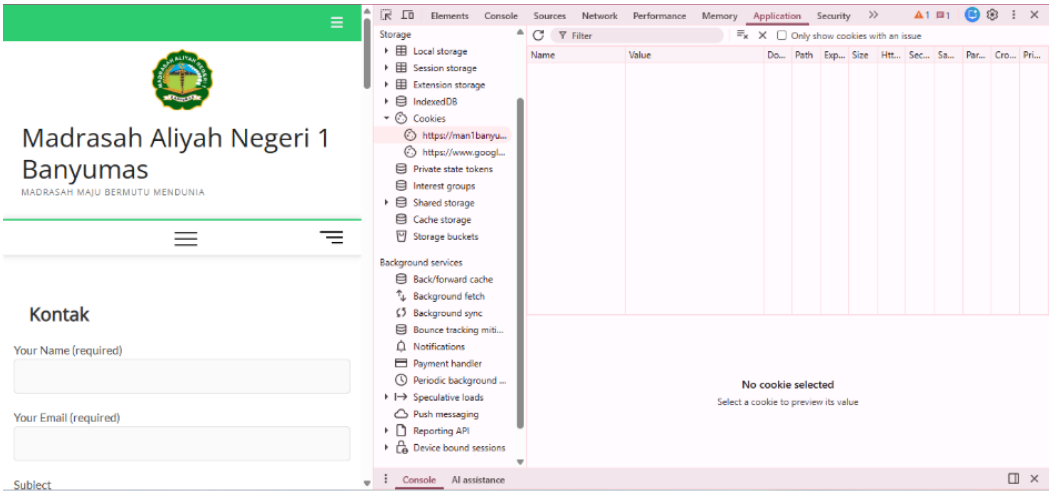
Gambar 10. Hasil XSS

Pengujian manual tidak berhasil memicu kotak peringatan (*pop-up*) pada peramban web. Kode *payload* yang disuntikkan justru diubah menjadi bentuk entitas teks aman (%3Cscript%3E...), yang membuktikan bahwa meskipun tajuk (*header*) *Content Security Policy* (CSP) tidak dikonfigurasi pada tingkat server, arsitektur aplikasi *Content Management System* (CMS) yang digunakan telah memiliki lapisan pertahanan tambahan berupa mekanisme sanitasi masukan (*input*) dan validasi karakter khusus. Dengan demikian,

temuan otomatis terkait risiko XSS melalui fitur pencarian ini dapat dikategorikan sebagai *false positive*.

3.8.4. Validasi Ketiadaan atribut Proteksi

Berdasarkan inspeksi visual pada baris data *cookie* yang aktif, ditemukan bahwa kolom parameter kontrol untuk atribut *HttpOnly* dan *Secure* berada dalam kondisi kosong tanpa tanda aktif, yang mengindikasikan bahwa kedua atribut keamanan tersebut tidak dikonfigurasi pada sistem target. Hasil validasi ditunjukan pada gambar 11.

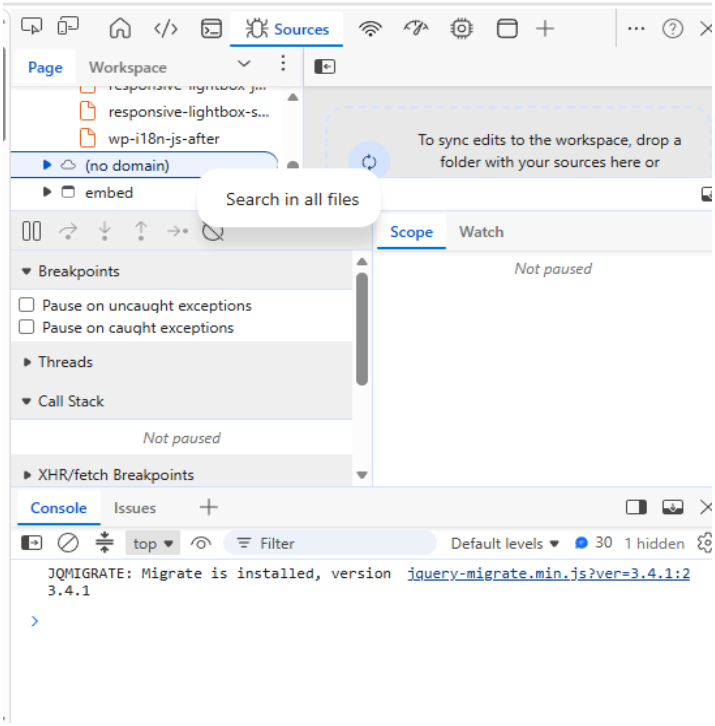


Gambar 11. Hasil Hijacking

Ketiadaan kedua tanda proteksi (*flag*) tersebut secara mutlak memvalidasi temuan ini sebagai *true positive* dengan tingkat risiko yang signifikan. Absennya *flag HttpOnly* memungkinkan skrip berbahaya pihak ketiga untuk membaca dan mencuri data *cookie* sesi menggunakan perintah JavaScript sederhana (*document.cookie*), apabila terjadi serangan XSS pada bagian lain sistem. Sementara itu, ketiadaan *flag Secure* menyebabkan peramban mengizinkan pengiriman data *cookie* sensitif melalui jalur HTTP biasa yang tidak terenkripsi. Kombinasi ketiadaan kedua konfigurasi tersebut membuka vektor serangan pembajakan sesi (*session hijacking*), di mana penyerang dapat menyadap atau mencuri pengidentifikasi sesi guna menduplikasi hak akses korban dan mengambil alih akun administrator (*account takeover*) tanpa perlu mengetahui kata sandi yang sebenarnya.

3.8.5. Analisis Kerentanan Vulnerable JS Library

Berdasarkan hasil pemindaian otomatis instrumen OWASP ZAP yang dikombinasikan dengan validasi manual melalui konsol peramban (*Developer Tools*), ditemukan bahwa situs target menggunakan pustaka JavaScript pihak ketiga yaitu *jQuery Migrate* versi 3.4.1. Hasil ditunjukan pada gambar 12.



Gambar 12. Hasil Vulnerable JS Library

Keberadaan pustaka ini berfungsi sebagai pembantu kompatibilitas untuk skrip-skrip usang. Meskipun pustaka pembantu ini berada pada versi yang relatif baru (v3.4.1), peringatan *Medium Risk* tetap muncul karena sistem mendeteksi adanya ketergantungan (dependency) terhadap fungsi-fungsi *manipulasi Document Object Model* (DOM) bawaan dari framework core yang berpotensi dieksploitasi jika pustaka utama tidak disanitasi dengan ketat. Namun, karena fungsi input utama seperti pencarian telah memiliki mekanisme output encoding, maka temuan eksekusi langsung pada library ini dapat dikategorikan sebagai low-to-medium risk yang memerlukan perhatian berupa pemeliharaan (*maintenance*) berkala.

3.9. Post exploitation

Sesuai dengan batasan *non-destructive testing*, fase pascaeksploitasi tidak melibatkan tindakan penanaman *backdoor* maupun pengambilalihan server secara fisik. Sebagai gantinya, fase ini difokuskan pada analisis skenario tingkat lanjut menggunakan pendekatan penggabungan kerentanan (*vulnerability chaining*). Meskipun temuan individual seperti ketiadaan *X-Frame-Options* (*Clickjacking*), ketiadaan token autentikasi *Cross-Site Request Forgery* (CSRF), serta penggunaan komponen pihak ketiga yang usang (*Vulnerable JS Library berupa jQuery Migrate* 3.4.1) masing-masing diklasifikasikan sebagai kerentanan tingkat menengah (*medium risk*), kombinasi ketiganya berpotensi menciptakan eskalasi serangan yang fatal. Dalam skenario serangan nyata yang lebih kompleks, penyerang dapat memanfaatkan kelemahan fungsi manipulasi DOM pada pustaka JavaScript lama untuk memasukan skrip berbahaya (*Client-side XSS*) terlebih dahulu. Selanjutnya, penyerang dapat menyembunyikan formulir perubahan kata sandi, formulir modifikasi data akademik, atau skrip jahat tersebut di dalam iframe tak terlihat dengan memanfaatkan celah *Clickjacking*. Apabila administrator atau pengajar yang sedang berada dalam sesi masuk (login) yang sah tanpa sengaja berinteraksi dengan halaman jebakan tersebut, peramban akan mengeksekusi permintaan paksa dengan memanfaatkan celah CSRF dan menjalankan skrip JavaScript usang tanpa sepengetahuan korban. Kombinasi bertingkat ini pada akhirnya dapat berujung pada pengambilalihan akun secara penuh (*account takeover*) serta kompromi integritas data di dalam *Content Management System* (CMS) sekolah.

3.10. Reporting

Sesuai dengan Proses analisis keamanan portal akademik MAN 1 Banyumas dilaksanakan secara menyeluruh dengan penetapan batasan masalah yang ketat guna menjaga kestabilan sistem operasional sekolah. Berdasarkan batasan pengujian *non-destructive*, fase pascaeksploitasi tidak melibatkan tindakan yang bersifat merusak seperti penanaman *backdoor* maupun pengambilalihan server secara fisik, melainkan difokuskan pada analisis skenario tingkat lanjut melalui pendekatan penggabungan kerentanan (*vulnerability chaining*). Celah Keamanan Infrastruktur Jaringan server (Port FTP & SSH Usang)

a. Pemindaian Nmap pada IP 103.31.251.68 mengungkapkan bahwa Port 21 (FTP Pure-FTPd) dan Port 22 (SSH OpenSSH 7.4 usang) berada dalam kondisi terbuka ke publik tanpa pengaman, sehingga rentan terhadap serangan *brute force* dan pembajakan akses. Sebagai langkah mitigasi, administrator server perlu segera memperbarui versi *OpenSSH* ke rilisan terbaru serta menerapkan kebijakan pembatasan akses (*IP Whitelisting*) agar port manajemen tersebut hanya dapat diakses oleh alamat IP khusus administrator sekolah.

b. Celah Lapisan Aplikasi Web (*Clickjacking* & Ketiadaan *Header* Pengaman). Halaman utama situs web teridentifikasi rentan terhadap serangan *Clickjacking* karena server tidak mengonfigurasi HTTP *security headers* sebagai pelindung, yang divalidasi melalui pengujian iframe palsu. Sebagai langkah mitigasi, administrator server perlu melakukan pengerasan server (*server hardening*) dengan memasukan perintah *konfigurasi Header always set X-Frame-Options* "SAMEORIGIN" serta mengaktifkan *Content Security Policy* (CSP) pada berkas *.htaccess* server *Apache*.

c. Penggunaan Komponen Pihak Ketiga yang Usang (*Vulnerable JS Library*) Inspeksi manual melalui konsol peramban (*Developer Tools*) membuktikan bahwa aplikasi web memuat pustaka client-side eksternal yang usang, yaitu *jQuery Migrate* versi 3.4.1. Keberadaan pustaka usang ini memperluas permukaan serangan (*attack surface*) dan memicu risiko penyerangan turunan seperti *Client-side Cross-Site Scripting* (DOM-XSS) jika berinteraksi dengan fungsi manipulasi DOM inti yang tidak disanitasi. Sebagai langkah mitigasi, pengembang web atau administrator sistem wajib melakukan pembaruan (*upgrade*) komponen pustaka JavaScript tersebut ke versi stabil terbaru atau mengeliminasi ketergantungan skrip lama yang sudah tidak digunakan lagi oleh CMS.

d. Kerentanan Formulir Kontak (*Cross-Site Request Forgery* / CSRF). Inspeksi manual membuktikan bahwa elemen tag *form* pada modul halaman kontak sekolah tidak menyertakan atribut token pengaman acak tersembunyi (anti-CSRF token), sehingga peretas berpotensi memanipulasi tindakan pengguna secara ilegal. Sebagai langkah mitigasi, pengembang aplikasi web perlu memodifikasi struktur kode dengan mengimplementasikan fungsi pembangkit token acak bawaan *WordPress*, yakni fungsi wp_*nonce_field*() untuk membuat token dan wp_*verify_nonce*() untuk memvalidasinya sebelum data diproses.

e. Kelemahan Manajemen Sesi *Cookie* (*Session Hijacking* & Risiko *Account Takeover*) Parameter kontrol *Application Cookies* utama *WordPress* teridentifikasi dalam kondisi tidak aktif untuk *flag HttpOnly* dan Secure, yang apabila digabungkan dengan celah lain (*vulnerability chaining*) berpotensi memicu pengambilalihan penuh akun administrator (*account takeover*). Sebagai langkah mitigasi, pengembang aplikasi web perlu melakukan penguatan sesi (*session hardening*) dengan mengubah konfigurasi berkas peladen seperti php.ini atau wp-config.php guna memastikan *flag session.cookie_httponly = On dan session.cookie_secure = On* aktif secara otomatis pada setiap *cookie autentikasi* yang diterbitkan."

4. Pembahasan

Pengujian keamanan situs web MAN 1 Banyumas yang dilaksanakan berdasarkan kerangka kerja PTES (Penetration Testing Execution Standard) menghasilkan sejumlah temuan teknis signifikan. Pada tahap intelligence gathering, fitur Spidering OWASP ZAP berhasil memetakan 700 tautan aktif serta aset pustaka skrip client-side. Pemindaian Nmap pada IP peladen 103.31.251.68 mengungkapkan terbukanya porta kritis seperti FTP (21) dan SSH OpenSSH 7.4 (22) yang rawan brute force, serta porta control panel dan layanan surel tanpa enkripsi standar. Selanjutnya, tahap eksploitasi dan validasi manual membuktikan empat kerentanan true positive, yaitu Clickjacking (ketiadaan X-Frame-Options), CSRF pada formulir kontak (tanpa token acak), kelemahan manajemen sesi (cookie tanpa atribut HttpOnly dan Secure), serta Vulnerable JS Library (jQuery Migrate 3.4.1). Sebaliknya, indikasi celah XSS dikonfirmasi sebagai false positive karena adanya mekanisme sanitasi input otomatis. Pada tahap pascaeksploitasi, analisis vulnerability chaining membuktikan bahwa kombinasi celah Clickjacking, CSRF, dan manipulasi DOM berpotensi memicu account takeover administratif. Sebagai solusi, peneliti merekomendasikan server hardening melalui konfigurasi security headers pada berkas .htaccess serta pembaruan pustaka JavaScript.

Apabila dibandingkan dengan penelitian Kurniawan et al**.** [5] dan Nurrizki et al. [6], penelitian ini tidak hanya mengandalkan hasil pemindaian otomatis menggunakan OWASP ZAP, tetapi juga melakukan validasi manual melalui pengujian payload, inspeksi elemen HTML, dan pemeriksaan konfigurasi *cookie.* Tahapan tersebut menghasilkan empat kerentanan *true positive* dan satu *false positive*, sehingga mengurangi potensi kesalahan interpretasi hasil pemindaian otomatis. Selain itu, berbeda dengan penelitian Tahir dan Risky [11], yang berfokus pada website instansi pemerintah, penelitian ini mengkaji portal akademik sekolah serta melengkapi analisis dengan pendekatan *vulnerability chaining* untuk menunjukkan potensi eskalasi serangan hingga *account takeover*.

5. Kesimpulan

Secara keseluruhan, audit postur keamanan siber pada portal akademik MAN 1 Banyumas menggunakan kerangka kerja *Penetration Testing Execution Standard* (PTES) berhasil mengidentifikasi kondisi keamanan sistem secara menyeluruh. Hasil pengujian menunjukkan bahwa proses *spidering* berhasil memetakan sekitar 700 *URL*, sedangkan pemindaian menggunakan *Nmap* mengidentifikasi 4 port terbuka, yaitu FTP (21), SSH (22), HTTP (80), dan HTTPS (443). Selanjutnya, OWASP ZAP menghasilkan 11 jenis *alert*, yang terdiri atas 5 temuan kategori *Medium Risk*, 3 *Low Risk*, dan 3 *Informational*. Setelah dilakukan validasi manual, diperoleh 4 kerentanan *true positive*, yaitu *Clickjacking, Cross-Site Request Forgery* (CSRF), *Session Hijacking* akibat ketiadaan atribut *HttpOnly* dan *Secure*, serta *Vulnerable JS Library*, sedangkan 1 temuan *Cross-Site Scripting* (XSS) dikonfirmasi sebagai *false positive*. *Analisis vulnerability chaining* menunjukkan bahwa kombinasi beberapa kerentanan tersebut berpotensi meningkatkan risiko hingga *account takeover*. Berdasarkan hasil tersebut, direkomendasikan penerapan *server hardening*, pembaruan layanan dan pustaka yang usang, penambahan *security headers*, serta penguatan manajemen sesi untuk meningkatkan keamanan website.

Sebagai pengembangan penelitian selanjutnya, disarankan untuk menerapkan pendekatan *White Box Testing* guna menganalisis celah keamanan langsung dari sisi kode sumber (*source code review*). Selain itu, penelitian masa depan juga dapat diarahkan pada perancangan sistem deteksi intrusi otomatis (*Intrusion Detection System*) serta integrasi konsep *DevSecOps* guna memastikan siklus pengembangan dan pemeliharaan situs web sekolah dapat berjalan secara aman dan berkelanjutan.

Referensi

[1] A. M. Nurjanah, "Analisis Keamanan Website Sekolah SMAN 1 TEMPULING dengan Menggunakan Open Web Application Securiti Project (OWASP)," *J. PERANGKAT LUNAK*, vol. 6, no. 2, pp. 351–361, 2024, https://doi.org/10.32520/jupel.v6i2.3442.

[2] M. A. Zein, U. Yunan, K. Septo, and A. Almaarif, "Security Hardening Sistem Operasi Virtual Private Server Pada Instansi Pendidikan XYZ Berdasarkan NIST SP 800-123," *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 8, no. 1, pp. 230–241, 2023, https://doi.org/10.29100/jipi.v8i1.3438.

[3] J. Khatib and S. Dalam, "Indonesian Journal of Computer Science," *Indones. J. Comput. Sci.*, vol. 13, no. 1, pp. 3315–3327, 2024, https://doi.org/10.33022/ijcs.v13i2.3736.

[4] D. K. S. dan S. (Kominfo-CSIRT), "Laporan Tahunan Keamanan Siber 2024," 2024. [Online]. Available: https://csirt.komdigi.go.id/storage/uploads/document_centers/attachments/Laporan_Tahunan_Kamsiber_2024.pdf

[5] B. Kurniawan, I. Ruslianto, and S. Bahri, "Implementasi Penetration Testing Pada Website Menggunakan Metode Penetration Testing Execution Standard (PTES) Implementation of Penetration Testing on the Website Using the Penetration Testing Execution Standard (PTES) Method," *CESS (Journal Comput. Eng. Syst. Sci.*, vol. 8, no. July, pp. 518–528, 2023, https://doi.org/10.24114/cess.v8i2.47096.

[6] M. A. Nurrizki, E. Iman, and H. Ujianto, "Analisis Keamanan Website Desa Budaya DIY Dengan Metode Penetration Testing (Pentest) dan OWASP ZAP," *J. Apl. Teknol. Inf. dan Manaj.*, vol. 5, no. 1, pp. 22–27, 2024, https://doi.org/10.31102/jatim.v5i1.2620.

[7] M. B. Imtias, K. Umam, H. Mustofa, and M. H. Subowo, "Comparative Analysis of Penetration Testing Frameworks : OWASP , PTES , and NIST SP 800-115 for Detecting Web Application Vulnerabilities," *J. Appl. Informatics Comput.*, vol. 9, no. 6, pp. 3689–3696, 2025, https://doi.org/10.30871/jaic.v9i6.9846.

[8] D. A. Utama and R. Supardi, "Analisis Keamanan Website Menggunakan PTES (Penetration Testing Execution And Standart)," *J. media Inform.*, vol. 20, no. 1, pp. 106–112, 2024, https://doi.org/10.37676/jmi.v20i1.5367.

[9] G. zaidan muflih abdur Rahman Fadilah, "Analisis Keamanan Website SMK VIP Al-Huda Kebumen Menggunakan Metode Penetration Testing Execution Standard (PTES)," *Chain J. Comput. Technol. Comput. Eng. Informatics*, vol. 4, no. 3, pp. 218–228, 2026, https://ejournal.techcart-press.com/index.php/chain/article/view/290.

[10] M. N. Fikri, B. P. Zen, R. Adhitama, and E. A. Firdaus, "Analisis Keamanan Sistem Informasi Website SMA Negeri 1 Sokaraja Menggunakan Metode Penetration Testing Execution Standard (PTES)," *J. Inform.*, vol. 2, no. September, pp. 19–27, 2023, https://doi.org/10.57094/ji.v2i2.1046.

[11] M. Tahir and M. Risky, "Analisis Keamanan Website Dinas Pemerintahan Yogyakarta Dengan Metode PTES (Penetration Testing Execution Standard)," *J. Tek. Inform. UNIKA St. Thomas*, vol. 09, no. 1, pp. 118–125, 2024, https://ejournal.ust.ac.id/index.php/JTIUST/article/view/3334.

[12] L. O. A. M. W. Selvi Dwi Hartiyani, Agung Prayogo, Ardy Wicaksono, "Pengujian Keamanan Jaringan Menggunakan Penetration Testing Execution Standart (PTES) di SMK Bina Harapan Yogyakarta," *Media Inform.*, vol. 24, no. 2, pp. 139–153, 2025, https://doi.org/10.37595/mediainfo.v24i2.355.

[13] L. S. Sign-on, S. Ulfa, and D. Surya, "Implementasi Penetration Testing Execution Standard Untuk Uji Penetrasi Pada," *Edu Komputika J.*, vol. 8, no. 1, pp. 48–56, 2021, https://doi.org/10.15294/edukomputika.v8i1.47179.