BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 3, April 2026 | Hal 877-885
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i3.1044

Evaluasi Keamanan Website Direktori Akademik Menggunakan NIST SP 800-115

Fito Nardian, Rahmad Abdillah*, Benny Sukma Negara, Reski Mai Candra

Fakultas Sains dan Teknologi, Prodi Teknik Informatika, Universitas Islam Negeri Sultan Syarif Kasim Riau, Pekanbaru, Indonesia
Email: [1]12050113994@students.uin-suska.ac.id., [2,*]rahmad.abdillah@uin-suska.ac.id, [3]bsnegara@uin-suska.ac.id, [4]reski.candra@uin-suska.ac.id
Email Penulis Korespondensi: rahmad.abdillah@uin-suska.ac.id

Abstrak−Evaluasi keamanan sistem informasi akademik berbasis web menjadi krusial seiring meningkatnya ancaman siber di lingkungan perguruan tinggi. Adanya rekam jejak insiden keamanan pada sistem informasi di lingkungan UIN Sultan Syarif Kasim Riau memicu urgensi tindakan preventif; oleh karena itu, website https://seminar-fst.uin-suska.ac.id sebagai layanan akademik yang aktif dan menyimpan data sensitif perlu dievaluasi secara proaktif. Pengujian menggunakan pendekatan black-box testing melalui empat fase, yaitu planning, discovery, attack, dan reporting. Hasil menunjukkan kerentanan kritis berupa SQL Injection pada parameter URL yang memungkinkan enumerasi basis data (MariaDB) tanpa otorisasi, sehingga mengancam kerahasiaan dan integritas data. Selain itu, ditemukan kerentanan tingkat menengah, seperti penggunaan pustaka JavaScript usang (Moment.js 2.8.1) dan miskonfigurasi header keamanan HTTP, termasuk tidak adanya Content Security Policy (CSP) dan mekanisme Anti-CSRF. Rekomendasi meliputi prepared statements, validasi input ketat, pembaruan dependensi, dan penguatan konfigurasi keamanan.

Kata Kunci: Keamanan Sistem Informasi; Pengujian Keamanan Web; NIST SP 800-115; Black Box; SQL Injection

Abstract−Evaluating the security of web-based academic information systems has become crucial as cyber threats in higher education environments increase. The track record of security incidents in information systems at UIN Sultan Syarif Kasim Riau has prompted an urgent need for preventative action; therefore, the website https://seminar-fst.uin-suska.ac.id, as an active academic service that stores sensitive data, requires a proactive evaluation. Testing used a black-box testing approach through four phases: planning, discovery, attack, and reporting. The results revealed a critical vulnerability in the form of SQL injection in URL parameters, which allows unauthorized database enumeration (MariaDB), thus threatening data confidentiality and integrity. Additionally, medium-level vulnerabilities were discovered, such as the use of an outdated JavaScript library (Moment.js 2.8.1) and misconfiguration of HTTP security headers, including the absence of a Content Security Policy (CSP) and an Anti-CSRF mechanism. Recommendations include prepared statements, strict input validation, updating dependencies, and strengthening security configurations.

Keywords: Information System Security; Web Security Testing; NIST SP 800-115; Black Box; SQL Injection

1. PENDAHULUAN

Website https://seminar-fst.uin-suska.ac.id merupakan salah satu layanan akademik berbasis web yang digunakan untuk mendukung pengelolaan kegiatan seminar di lingkungan Fakultas Sains dan Teknologi UIN Sultan Syarif Kasim Riau. Sebagai sistem yang mengelola data akademik, aspek keamanan menjadi krusial untuk menjamin kerahasiaan (*confidentiality*), integritas (*integrity*), dan ketersediaan (*availability*) informasi [1].

Penggunaan sistem informasi berbasis web dalam institusi pendidikan tinggi meningkatkan eksposur terhadap ancaman keamanan siber yang semakin kompleks dan beragam [2]. Dalam konteks ini, kelemahan pada sistem dapat dimanfaatkan oleh pihak tidak bertanggung jawab untuk mengakses atau memanipulasi data secara tidak sah [3].

Di lingkungan UIN Sultan Syarif Kasim Riau, pengelolaan infrastruktur teknologi informasi berada di bawah Pusat Teknologi Informasi dan Pangkalan Data (PTIPD) yang bertanggung jawab terhadap keamanan sistem [4]. Namun, insiden keamanan pernah terjadi pada salah satu subdomain layanan akademik, yaitu https://sireg.uin-suska.ac.id, yang mengalami *defacement* dengan penyisipan konten ilegal. Mengingat layanan akademik beroperasi di bawah infrastruktur jaringan institusi yang sama, insiden pada satu subdomain tersebut menjadi indikator kuat bahwa subdomain lainnya juga memiliki risiko paparan kerentanan yang tinggi terhadap eksploitasi.

Berdasarkan kondisi tersebut, diperlukan evaluasi keamanan secara preventif terhadap sistem akademik lain yang aktif digunakan, termasuk *website* https://seminar-fst.uin-suska.ac.id, guna mengidentifikasi potensi kerentanan sebelum dieksploitasi oleh penyerang serta meningkatkan ketahanan sistem terhadap ancaman siber [5]. Salah satu metode yang umum digunakan untuk mengidentifikasi kerentanan adalah *penetration testing*, yaitu teknik pengujian keamanan dengan mensimulasikan serangan terhadap sistem secara terkontrol [6]. Dalam penelitian ini, pengujian dilakukan mengacu pada standar NIST SP 800-115 yang menyediakan panduan sistematis dalam pelaksanaan pengujian keamanan, meliputi tahapan *planning*, discovery, attack, dan reporting [7].

Sejumlah penelitian sebelumnya menunjukkan bahwa metode *penetration testing* berbasis NIST SP 800-115 efektif dalam mengidentifikasi kerentanan pada aplikasi web. Penelitian oleh Mambo, Yuniarto, dan Setiadi membuktikan bahwa pendekatan ini mampu mengungkap berbagai celah keamanan, seperti kelemahan autentikasi dan konfigurasi sistem, melalui tahapan yang sistematis mulai dari *planning*, *discovery*, hingga *attack*, sehingga evaluasi keamanan dapat dilakukan secara menyeluruh[8]. Hal serupa juga ditemukan dalam penelitian Maherza, yang menunjukkan bahwa penerapan NIST SP 800-115 pada website sekolah mampu mengidentifikasi kerentanan seperti *SQL Injection*, *Cross-Site Scripting (XSS)*, serta kelemahan manajemen sesi, sehingga penting untuk meningkatkan keamanan sistem informasi berbasis web[9]. Selain itu, penelitian Imtias et al. melalui analisis komparatif dengan framework lain seperti OWASP dan PTES menyimpulkan bahwa NIST SP 800-115 memiliki keunggulan dalam struktur metodologi dan dokumentasi

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 3, April 2026 | Hal 877-885
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i3.1044

yang lebih komprehensif, sehingga memudahkan proses identifikasi, analisis, dan mitigasi kerentanan[10]. Dengan demikian, metode NIST SP 800-115 dapat dianggap sebagai pendekatan yang efektif, sistematis, dan komprehensif dalam melakukan evaluasi keamanan website.

Hasil penelitian ini diharapkan tidak hanya memberikan rekomendasi mitigasi teknis yang relevan, tetapi juga berkontribusi dalam mengungkap kerentanan kritis yang tidak terdeteksi oleh alat pemindaian otomatis melalui pendekatan eksploitasi manual, serta menjadi acuan dalam meningkatkan keamanan layanan akademik berbasis web di lingkungan UIN Sultan Syarif Kasim Riau.

2. METODOLOGI PENELITIAN

2.1 Model Kerangka Kerja

Metodologi dalam penelitian ini mengadopsi kerangka kerja NIST SP 800-115 sebagai standar dalam pelaksanaan *penetration testing* [11]. Standar ini dipilih karena menyediakan pendekatan sistematis dan terstruktur dalam melakukan evaluasi keamanan sistem informasi, khususnya untuk pengujian berbasis simulasi serangan pada aplikasi web [12]. Penggunaan NIST SP 800-115 dalam penelitian ini juga bertujuan untuk memastikan bahwa proses pengujian dilakukan secara terkontrol, terukur, dan sesuai dengan praktik terbaik (best practices) dalam keamanan siber.

Secara umum, metodologi ini terdiri dari empat fase utama, yaitu *Planning, Discovery, Attack,* dan *Reporting* [13]. Keempat fase tersebut digunakan sebagai kerangka kerja dalam mengidentifikasi, memvalidasi, dan mendokumentasikan kerentanan keamanan pada sistem yang diuji. Pendekatan yang digunakan adalah *black-box testing*, di mana pengujian dilakukan tanpa akses terhadap informasi internal sistem, seperti arsitektur jaringan maupun kode sumber aplikasi, sehingga mensimulasikan kondisi serangan dari pihak eksternal [14].

Untuk menjaga validitas pengujian, seluruh proses dilakukan dengan memperhatikan batasan etis (*Rules of Engagement*), termasuk pembatasan eksploitasi hanya pada tahap verifikasi kerentanan tanpa melakukan modifikasi atau perusakan data [15]. Dengan demikian, pendekatan ini tidak hanya berfungsi untuk mengidentifikasi celah keamanan, tetapi juga untuk mengevaluasi ketahanan sistem dalam kondisi serangan nyata secara aman dan terkendali.

Tahapan metode NIST SP 800-115 yang digunakan dalam penelitian ini ditunjukkan pada Gambar 1.

Gambar 1. Metode NIST SP 800-115

Metode NIST SP 800-115 berdasarkan Gambar 1 memiliki beberapa tahapan di antaranya sebagai berikut:

a. Fase *Planning*: Tahap ini mencakup penentuan tujuan, ruang lingkup, target sistem, serta metode pengujian yang akan digunakan. Selain itu, ditetapkan pula *Rules of Engagement* untuk memastikan bahwa proses pengujian tidak mengganggu operasional sistem yang sedang berjalan [16].
b. Fase *Discovery*: Pada tahap ini dilakukan proses pengumpulan informasi (*information gathering*) dan pemindaian sistem, baik secara pasif maupun aktif. Aktivitas ini mencakup identifikasi layanan, teknologi yang digunakan, serta pemindaian kerentanan menggunakan tools seperti Nmap, Wappalyzer, dan OWASP ZAP untuk memperoleh gambaran permukaan serangan (*attack surface*) [17].
c. Fase *Attack*: Tahap ini bertujuan untuk memvalidasi kerentanan yang telah ditemukan pada fase sebelumnya melalui eksploitasi terkontrol. Pengujian dilakukan secara hati-hati untuk memastikan bahwa kerentanan tersebut benar-benar dapat dimanfaatkan tanpa menyebabkan gangguan pada sistem. Dalam penelitian ini, eksploitasi dilakukan menggunakan teknik manual dan tools otomatis seperti SQLMap untuk menguji kerentanan SQL Injection [18].
d. Fase *Reporting*: Tahap akhir berupa penyusunan laporan hasil pengujian yang mencakup identifikasi kerentanan, tingkat risiko, bukti eksploitasi, serta rekomendasi mitigasi teknis. Laporan ini disusun secara sistematis untuk memberikan gambaran menyeluruh mengenai kondisi keamanan sistem serta langkah perbaikan yang diperlukan [19]

3. HASIL DAN PEMBAHASAN

Pengujian keamanan (*penetration testing*) dalam studi ini mengacu pada standar NIST SP 800-115 dengan menggunakan metode *Black Box*. Lingkungan pengujian melibatkan sistem operasi Kali Linux sebagai *workstation* penyerang (*attacker machine)* dan Windows sebagai platform analisis. Fase pengujian dimulai dari *Planning* untuk menentukan ruang lingkup (*scope*) dan aturan main (*Rules of Engagement*). Selanjutnya, dilakukan fase *Discovery* guna mengumpulkan informasi aset dan kerentanan target. Informasi tersebut kemudian divalidasi pada tahap *Attack* melalui simulasi eksploitasi celah

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 3, April 2026 | Hal 877-885
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i3.1044

keamanan, dan diakhiri dengan tahap *Reporting* yang merangkum hasil temuan untuk rekomendasi perbaikan secara keseluruhan.

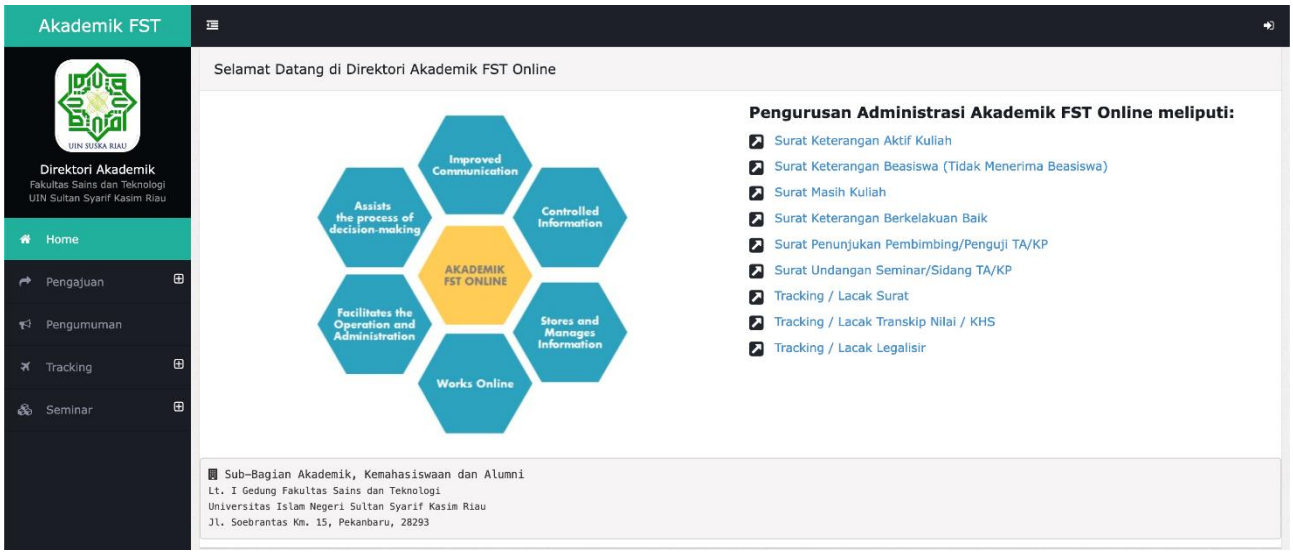
3.1 *Planning*

Tahap *planning* merupakan tahap perencanaan yang mencakup penentuan target, ruang lingkup, metode, serta tujuan pengujian yang akan dilakukan [20]. Pada penelitian ini, target pengujian adalah *website* https://seminar-fst.uin-suska.ac.id/ yang diuji di luar jam kampus untuk menghindari gangguan terhadap sistem yang sedang beroperasi. Identifikasi target pengujian beser ta alamat IP yang digunakan dalam penelitian ini ditampilkan pada Tabel 1.

Tabel 1. Identifikasi Target Pengujian

Target	IP
https://seminar-fst.uin-suska.ac.id	103.193.xx.xx

Berdasarkan Tabel 1, *website* seminar-fst.uin-suska.ac.id memiliki alamat IP 103.193.xx.xx yang menjadi target utama dalam proses pengujian keamanan pada penelitian ini. Tampilan awal dari sistem informasi yang menjadi objek pengujian ditunjukkan pada Gambar 2.



Gambar 2. Tampilan Dashboard

Berdasarkan Gambar 2, halaman *dashboard* menampilkan beberapa fitur layanan akademik seperti pengajuan seminar, pengumuman, dan menu navigasi sistem. Halaman ini berfungis titik awal dalam proses identifikasi struktur sistem pada tahap pengujian keamanan.

Selain target teknis, fae ini juga menetapkan *Rules of Engagement* (RoE) sebagai standar etika dan batasan keamanan, dengan mempertimbangkan bahwa objek pengujian adalah sistem akademik yang sedang beroperasi (live server). Mengacu pada izin riset dan kerangka kerja NIST SP 800-115, batasan pengujian ditetapkan dengan poin-poin sebagai berikut:

a. Waktu Pengujian (*Timing*): Eksekusi pemindaian dan injeksi aktif hanya dilakukan di luar jam operasional akademik (pukul 22.00 – 04.00 WIB) untuk mencegah kelebihan beban server.
b. Batas Eksploitasi *(Exploitation Limit*): Pengujian SQL Injection dihentikan hanya pada tahap enumerasi struktur basis data. Penulis dilarang keras melakukan ekstraksi data massal (*dumping*), modifikasi, atau penghapusan data.
c. Larangan Serangan Destruktif: Penggunaan metode yang merusak ketersediaan layanan (*Availability*), seperti pengujian Denial of Service (DoS), sama sekali tidak diperkenankan.
d. Prosedur Darurat (Emergency Stop): Semua aktivitas pengujian wajib segera dihentikan jika sistem target menunjukan penurunan performa drastis atau gagal muat (crash).

3.2 Implementasi dan Pengujian

Bagian ini berisi hasil eksekusi dari metodologi NIST SP 800-115 yang diterapkan pada domain seminar-fst.uin-suska.ac.id, meliputi tahap penemuan (*discovery*) hingga penyerangan (*attack*).

3.2.1 Fase Penemuan (*Discovery Phase*)

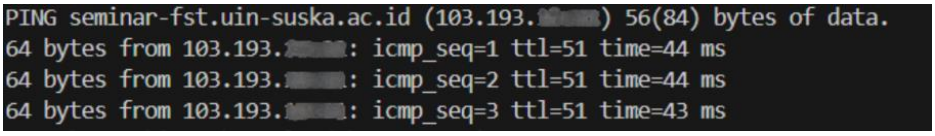
Fase penemuan (*discovery phase*) ini adalah tahap awal dalam proses *penetration testing* yang bertujuan untuk mengumpulkan berbagai informasi terkait sistem target untuk menentukan strategi pengujian untuk tahap selanjutnya [21]. Dalam penelitian ini, proses *discovery* dilakukan melalui tahap *Information Gathering* dan *Vulnerability Identification.*

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 3, April 2026 | Hal 877-885
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i3.1044

a. Information Gathering

Fase ini difokuskan pada pengumpulan data profil target secara komprehensif melalui berbagai sumber terbuka di internet. Langkah-langkah analitis yang dilakukan mencakup pelacakan alamat IP publik serta pendeteksian jenis *Content Management System* (CMS) yang diterapkan pada situs web tersebut.

1. *Verifikasi* Ketersediaan *Host*: Dalam tahap *information gathering*, dilakukan verifikasi ketersediaan server untuk memastikan bahwa target dapat diakses sebelum dilakukan pengujian lebih lanjut. Pada penelitian ini, penulis mengimplementasikan sebuah skrip berbasis Python Socket Programming yang berfungsi sebagai *ICMP availability monitor*. Skrip ini digunakan untuk mengirimkan permintaan ICMP secara berulang guna mengevaluasi respons server secara real-time. Secara fungsional, mekanisme yang digunakan sejalan dengan utilitas standar seperti *ping*. Namun, skrip yang dikembangkan memungkinkan penyesuaian format keluaran sehingga menampilkan parameter jaringan seperti ICMP sequence, time-to-live (TTL), dan waktu respons secara terstruktur (*Linux-style output*), yang memudahkan proses observasi dan dokumentasi hasil pengujian. Hasil pengujian menunjukkan bahwa server seminar-fst.uin-suska.ac.id berada dalam kondisi aktif (*up*) dan mampu merespons permintaan ICMP secara konsisten. Selain itu, alamat IP publik server berhasil diidentifikasi, yaitu 103.193.xx.xx, dengan latensi rata-rata yang relatif stabil. Kondisi ini mengindikasikan bahwa server dapat dijangkau dengan baik dan siap untuk dilakukan tahapan pengujian lanjutan.



Gambar 3. Hasil Eksekusi Network Scanner untuk Identifikasi IP Target

Hasil eksekusi proses verifikasi tersebut ditampilkan pada Gambar 3, yang menunjukkan respons ICMP dari server target beserta informasi parameter jaringan seperti ukuran paket dan nilai *time-to-live* (TTL). Berdasarkan hasil tersebut, dapat disimpulkan bahwa server dalam kondisi aktif dan siap untuk dilakukan pengujian lanjutan.

2. Pemetaan Jaringan (*Network Mapping*): Proses pemetaan jaringan (*network mapping*) dilakukan untuk mengidentifikasi layanan yang tersedia pada server target serta memetakan potensi titik masuk (*entry points*) yang dapat dimanfaatkan dalam proses pengujian. Tahap ini diawali dengan pemindaian terhadap alamat IP target, yang kemudian dilanjutkan dengan identifikasi layanan (*services*) dan sistem yang berjalan pada server. Pemindaian port dilakukan sebagai langkah awal untuk mengetahui port terbuka yang dapat diakses secara publik. Informasi ini sangat penting karena setiap port terbuka merepresentasikan layanan aktif yang berpotensi menjadi vektor serangan apabila tidak dikonfigurasi dengan baik. Pengujian dilakukan menggunakan perangkat lunak Nmap berbasis *Command Line Interface (CLI)* dengan menjalankan perintah nmap -sV 103.193.xx.xx. Parameter -sV (*Service Version Detection*) digunakan untuk memungkinkan identifikasi tidak hanya terhadap status port (open/closed), tetapi juga terhadap nama dan versi perangkat lunak yang berjalan pada masing-masing port. Dengan demikian, hasil pemindaian tidak hanya memberikan informasi mengenai keberadaan layanan, tetapi juga memberikan gambaran awal terkait potensi kerentanan berdasarkan versi perangkat lunak yang digunakan. Detail output hasil pemindaian tersebut disajikan pada Gambar 4, yang menampilkan daftar port terbuka beserta layanan yang teridentifikasi pada server target.

Gambar 4. Pemindaian Port Menggunakan Nmap

Hasil proses *scanning* port menggunakan *tools* nmap pada Gambar 4 diperoleh hasil ada beberapa port yang terbuka pada *scanning* port yang dilakukan menggunakan *tools* nmap pada gambar diatas di peroleh beberapa port yang terbuka pada *website* seminar-fst.uin-suska.ac.id yang di tunjukan pada Tabel 3.

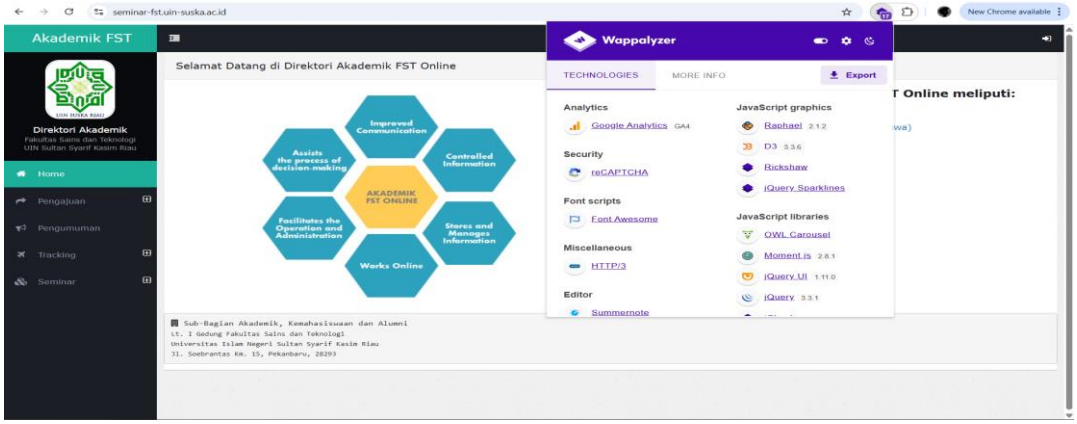
Tabel 3. Hasil Pemindaian *Port* Menggunakan Nmap

Port	Protokol	Status	Layanan (*Service*)	Versi / Keterangan
80	TCP	Open	HTTP	LiteSpeed (*Web Server*)
443	TCP	Open	HTTPS (SSL)	LiteSpeed (*Web Server*)
465	TCP	Open	SMTP (SSL)	Postfix smtpd (*Mail Server*)
587	TCP	Open	SMTP	Postfix smtpd (*Mail Server*)

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 3, April 2026 | Hal 877-885
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i3.1044

Berdasarkan hasil pemindaian yang ditampilkan pada Tabel 3, server target memiliki beberapa port terbuka, yaitu port 80 dan 443 yang digunakan untuk layanan *web server* berbasis LiteSpeed, serta port 465 dan 587 yang digunakan untuk layanan SMTP pada *mail server* berbasis Postfix. Keberadaan layanan-layanan ini menunjukkan bahwa server tidak hanya berfungsi sebagai web server tetapi juga menjalankan layanan pengiriman email.

3. Identifikasi Teknologi dan CMS: Setelah memastikan bahwa layanan web server berjalan dengan baik, proses *information gathering* dilanjutkan dengan tahap identifikasi teknologi dan komponen sistem yang digunakan oleh aplikasi web. Tahap ini bertujuan untuk memperoleh gambaran arsitektur teknologi yang mendasari sistem, khususnya komponen perangkat lunak pihak ketiga (*third-party components*) yang berpotensi memperluas permukaan serangan (*attack surface*). Identifikasi dilakukan menggunakan ekstensi Wappalyzer, yang mampu mendeteksi berbagai teknologi berbasis *client-side* maupun *server-side*, termasuk *content management system* (CMS), pustaka JavaScript, serta framework yang digunakan oleh aplikasi web. Informasi yang diperoleh dari proses ini memiliki peran penting dalam konteks keamanan, karena setiap teknologi yang digunakan berpotensi memiliki kerentanan yang telah terdokumentasi, terutama apabila menggunakan versi yang usang atau tidak diperbarui. Dengan demikian, hasil identifikasi ini tidak hanya memberikan gambaran struktur teknologi sistem, tetapi juga menjadi dasar dalam proses analisis kerentanan pada tahap selanjutnya. Detail hasil identifikasi teknologi yang digunakan oleh sistem target disajikan pada Gambar 5, yang menunjukkan komponen-komponen utama yang terdeteksi selama proses pemindaian.



Gambar 5. Identifikasi Teknologi Menggunakan Wappalyzer

Berdasarkan hasil pemindaian pada Gambar 5, berhasil diidentifikasi beberapa komponen teknologi utama yang menyusun arsitektur aplikasi *web* target. Pemetaan ini difokuskan pada teknologi yang relevan dengan memperluasan permukaan serangan (*attack surface*) aplikasi. Detail teknologi tersebut disajikan pada Tabel 4 berikut:

Tabel 4. Hasil Identifikasi Teknologi Utama Menggunakan Wappalyzer

No.	Kategori	Nama Teknologi	Versi Terdeteksi	Keterangan
1	*Security*	reCAPTCHA	-	Mekanisme anti-bot / *spam*
2	*Text Editor*	Summernote		Titik input pengguna (potensi XSS)
3	*JavaScript Library*	Moment.js	2.8.1	Pustaka terindikasi rentan (Usang)
4	*JavaScript Library*	jQuery	3.3.1	Pustaka interaksi *front-end*
5	*JavaScript Library*	jQuery UI	1.11.0	Pustaka antarmuka *front-end*

b. *Vulnerability Identification*

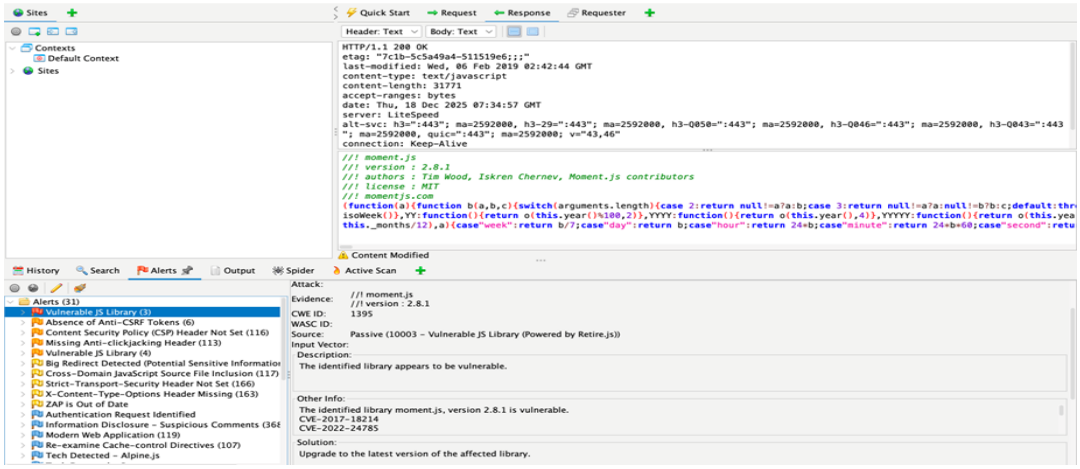
Pada tahap ini, proses identifikasi kerentanan keamanan dilakukan secara sistematis menggunakan *automated scanner tools* OWASP ZAP (Zed Attack Proxy) pada lingkungan sistem operasi Windows. Pemindaian difokuskan pada URL target seminar-fst.uin-suska.ac.id untuk menganalisis respons aplikasi web terhadap berbagai skenario serangan umum (*common attack vectors*), serta untuk mengidentifikasi potensi kesalahan konfigurasi keamanan (*security misconfiguration*). Penggunaan OWASP ZAP pada tahap ini memungkinkan deteksi awal terhadap berbagai jenis kerentanan, baik yang berkaitan dengan konfigurasi server, mekanisme proteksi aplikasi, maupun penggunaan komponen perangkat lunak yang tidak aman. Proses pemindaian dilakukan secara otomatis dengan mengirimkan berbagai payload uji untuk mengevaluasi bagaimana sistem merespons permintaan yang berpotensi berbahaya. Hasil pemindaian menunjukkan bahwa sistem memiliki sejumlah kerentanan dengan tingkat risiko yang bervariasi, yaitu dari kategori Low hingga Medium. Berdasarkan analisis otomatis yang dihasilkan oleh OWASP ZAP, terdeteksi beberapa *security alerts* yang mengindikasikan adanya potensi kelemahan pada aplikasi web yang diuji. Untuk memberikan gambaran yang lebih terstruktur dan komprehensif, seluruh temuan kerentanan tersebut kemudian dirangkum dan disajikan secara sistematis dalam Tabel 5, yang memuat jenis kerentanan, tingkat risiko, jumlah temuan, serta deskripsi singkat dari masing-masing kerentanan yang teridentifikasi.

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 3, April 2026 | Hal 877-885
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i3.1044

Tabel 5. Rekapitulasi Temuan Kerentanan Menggunakan OWASP ZAP

No	Jenis Kerentanan (*Vulnerability*)	Tingkat Risiko (*Risk Level*)	Jumlah Temuan	Deskripsi Singkat
1	*Vulnerable JS Library*	*Medium*	3	Terdeteksi penggunaan pustaka JavaScript versi lama yang memiliki celah keamanan publik.
2	*Content Security Policy (CSP) Header Not Set*	*Medium*	116	Server tidak mengirimkan header CSP, meningkatkan risiko serangan XSS.
3	*Missing Anti-clickjacking Header*	*Medium*	113	Tidak adanya proteksi X-Frame-Options yang memungkinkan situs dimuat dalam *iframe* berbahaya.
4	*Absence of Anti-CSRF Tokens*	*Medium*	6	Formulir HTML tidak memiliki token unik untuk mencegah serangan pemalsuan permintaan (CSRF).
5	*Strict-Transport-Security Header Not Set*	*Low*	166	*Header* HSTS tidak aktif, memungkinkan koneksi di-downgrade dari HTTPS ke HTTP.
6	*X-Content-Type-Options Header Missing*	*Low*	163	Header keamanan MIME-sniffing tidak dikonfigurasi pada respon server.

Dari hasil rekapitulasi pada Tabel 5, Dari seluruh temuan tersebut, kerentanan yang memiliki dampak keamanan paling signifikan adalah penggunaan pustaka JavaScript versi lama (Vulnerable JS Library), karena komponen ini secara langsung berkaitan dengan kode yang berjalan pada aplikasi web dan telah diketahui memiliki celah keamanan publik. Untuk memberikan gambaran lebih jelas mengenai hasil pemindaian dan peringatan kerentanan yang terdeteksi oleh alat pemindai, tampilan panel alerts dashboard dari OWASP ZAP ditunjukkan pada Gambar 6.



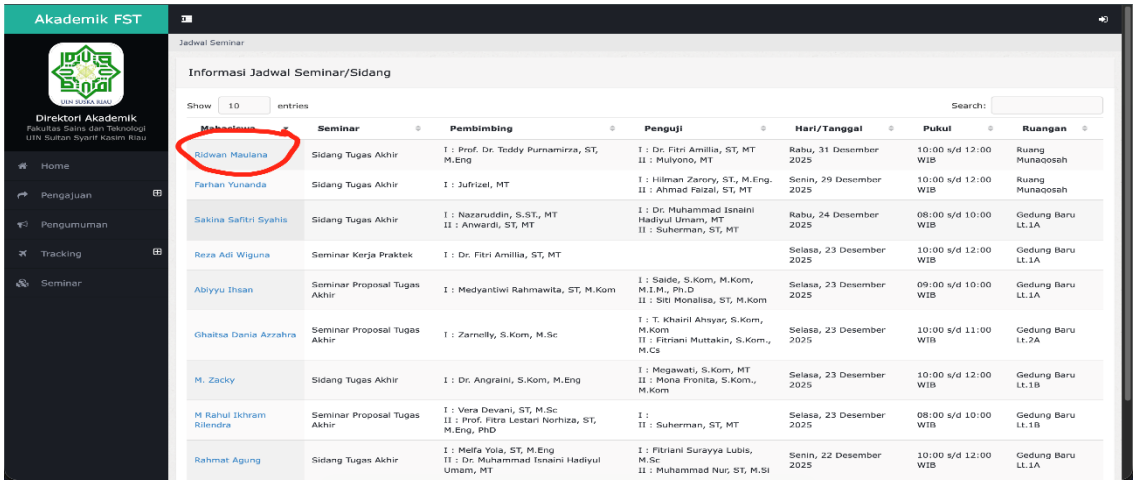
Gambar 6. Panel Dashboard Alerts Hasil Pemindaian OWASP ZAP

Berdasarkan hasil pemindaian yang ditunjukan pada Gambar 6, perangkat lunak OWASP ZAP mengeluarkan peringatan (*alert*) dengan tingkat risiko Menengah (*Medium*) pada komponen *Vulnerable JS Library*. Temuan ini mengonfirmasi hasil identifikasi awal pada Tabel 5 (Wappalyzer) bahwa pustaka Moment.js versi 2.8.1 yang digunakan oleh server target memiliki kerentanan keamanan yang tercantum dalam basis data kerentanan global (CVE-2017-18214 dan CVE-2022-24785). Peringatan ini menunjukkan bahwa sistem memiliki kerentanan yang dapat dieksploitasi melalui manipulasi pustaka sisi klien (*client-side library*) atau melalui serangan Path Traversal. Meskipun demikian, penting untuk ditekankan bahwa pemindaian otomatis ini memiliki batasan *false-negative*. Kerentanan yang sesungguhnya berisiko paling tinggi dan bersifat kritis (*Critical*), yaitu *SQL injection*, justru luput dari deteksi OWASP ZAP. Celah mematikan ini baru berhasil diidentifikasi dan divalidasi melalui eksploitasi manual pada fase selanjutnya (*Attack*). Hal ini membuktikan kelemahan alat otomatis jika tidak diimbangi dengan pengujian manual secara komprehensif.

3.2.2 Fase Penyerangan (*Attack Phase*)

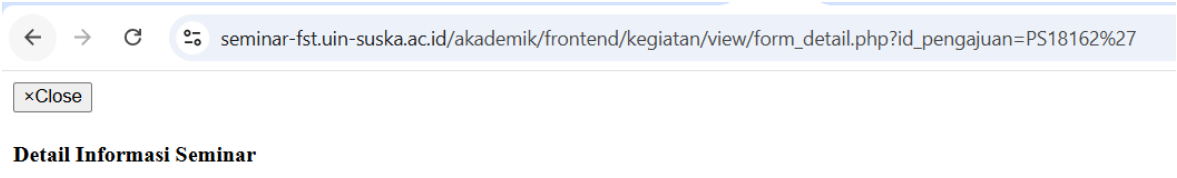
Fase ini bertujuan untuk memvalidasi kerentanan sistem secara lebih mendalam, khususnya dalam interaksi aplikasi dengan basis data. Berdasarkan hasil pemetaan pada fase sebelumnya, adanya parameter input yang berinteraksi langsung dengan basis data, sehingga pengujian dilanjutkan dengan simulasi serangan SQL Injection secara manual. Penelusuran difokuskan pada parameter URL yang menggunakan metode HTTP GET. Berdasarkan pemetaan fitur pada modul "Jadwal Seminar", diidentifikasi sebuah tautan aktif pada nama mahasiswa yang mengarah ke endpoint spesifik sistem (Gambar 7). Tautan ini dijadikan sebagai titik masukan (entry point) untuk menyisipkan payload SQL Injection guna menguji ketahanan validasi input pada basis data target. Tampilan antarmuka dari tautan tersebut dapat dilihat pada Gambar 7.

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 3, April 2026 | Hal 877-885
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i3.1044



Gambar 7. Titik Masuk (Entry Point) Pengujian SQL Injection Manual

Setelah mengidentifikasi titik masuk pada Gambar 7, pengujian dilanjutkan dengan memodifikasi parameter URL menggunakan *payload* karakter khusus untuk memicu respons dari basis data. Hasil uji coba dapat dilihat pada Gambar 8.



Gambar 8. Hasil Uji Coba Injeksi SQL pada Parameter URL

Berdasarkan Gambar 8, pengujian dilakukan dengan memasukkan karakter *single quote* (') yang terenkode sebagai nilai %27 di akhir parameter id_pengajuan di URL. Hasil eksekusi menunjukkan anomali dalam respons server target. Aplikasi tidak dapat memproses permintaan secara normal dan mengalami kegagalan fatal dalam menampilkan dalam menampilak antarmuka grafis secara keseluruhan (menghasilkan halaman putih atau blank page tanpa struktur tata letak). Kegagalan sistem dalam menangani galat (*improper error handling*) dan tidak adanya sanitasi input (*input validation*) pada parameter tersebut membuktikan secara empiris bahwa titik masuk ini sangat rentan terhadap eksploitasi *SQL Injection*. Untuk memvalidasi sejauh mana dampak tingkat keparahan (*severity*) dari celah *SQL Injection* tersebut, pengujian dilanjutkan ke tahap eksploitasi penuh (*full exploitation*). Proses ini dilakukan secara otomatis menggunakan perangkat lunak SQLMap. Parameter URL yang sebelumnya teridentifikasi rentan dieksekusi untuk melakukan enumerasi terhadap struktur basis data (*database enumeration*) di belakang sistem. Eksekusi dilakukan dengan perintah dasar sqlmap -u "https://[DOMAIN_TARGET]/.../view/form_detail.php?id_pengajuan=abc" --random-agent --dbs.

Gambar 9. Hasil Enumerasi Basis Data Menggunakan SQLMap

Berdasarkan hasil eksekusi perintah pada gambar 9, SQLMap berhasil mengonfirmasi kerentanan secara pasti melalui tiga teknik injeksi berbeda, yaitu *boolean-based blind*, *time-based blind*, dan *UNION query*. Lebih lanjut, proses enumerasi berhasil mengekstrak informasi tentang sistem manajemen basis data (DBMS) yang beroperasi di balik layar, yaitu MySQL (MariaDB). Eksploitasi ini juga berhasil mengungkap keberadaan dua basis data utama pada server target,

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 3, April 2026 | Hal 877-885
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i3.1044

yaitu information_schema dan semi_fst_uinsuska. Temuan kebocoran informasi (*information Disclosure*) ini membuktikan secara empiris bahwa penyerang jarak jauh dapat memperoleh akses tidak sah terhadap struktur data sensitif pada aplikasi akademik tersebut.

3.2.3 Fase Pelaporan (*Reporting Phase)*

Tahap pelaporan (*reporting phase*) merupakan fase akhir dalam metodologi keamanan NIST SP 800-115 yang bertujuan untuk menyusun dan menyajikan seluruh hasil pengujian secara sistematis dan komprehensif [22]. Pada tahap ini, seluruh temuan kerentanan yang diperoleh dari fase penemuan (*discovery*) dan fase penyerangan (*attack*) dikonsolidasikan untuk dilakukan evaluasi tingkat risiko berdasarkan dampak dan potensi eksploitasinya terhadap sistem.

Proses analisis pada tahap ini tidak hanya berfokus pada identifikasi jenis kerentanan, tetapi juga mencakup penilaian terhadap tingkat keparahan (*severity level*), kemungkinan eksploitasi (*likelihood*), serta dampaknya terhadap aspek keamanan informasi, khususnya kerahasiaan (*confidentiality*), integritas (*integrity*), dan ketersediaan (*availability*) sistem.

Berdasarkan hasil pengujian terhadap aplikasi akademik yang menjadi objek penelitian, diidentifikasi tiga kategori kerentanan utama yang memerlukan perhatian dan penanganan segera. Kerentanan tersebut berpotensi menimbulkan gangguan terhadap integritas dan keamanan data institusi apabila tidak dilakukan mitigasi secara tepat.
Rincian temuan kerentanan beserta rekomendasi mitigasi teknis disajikan secara sistematis pada Tabel 6, yang mencakup klasifikasi jenis kerentanan, tingkat risiko, dampak yang ditimbulkan, serta langkah-langkah perbaikan yang direkomendasikan untuk meningkatkan postur keamanan sistem.

Tabel 6. Rekapitulasi Temuan dan Rekomendasi Mitigasi Kerentanan

No.	Jenis Kerentanan	Tingkat Risiko	Rekomendasi Mitigasi (Perbaikan)
1	*SQL Injection*	*(Parameter URL id_pengajuan)* Tinggi (*Critical/High*)	1. Implementasikan *Prepared Statements* (*Parameterized Queries*) pada kode sumber PHP untuk memisahkan logika kueri dari *input* pengguna. 2. Lakukan sanitasi dan validasi *input* secara ketat (misalnya menggunakan fungsi mysqli_real_escape_string).
2	*Vulnerable JS Library*	*(Moment.js versi 2.8.1)* Menengah (*Medium*)	1. Lakukan pembaruan (*upgrade*) pustaka Moment.js ke versi terbaru yang stabil (minimal versi 2.29.4 ke atas) untuk menambal celah CVE-2022-24785.
3	*Misconfigured Security Headers*	*(Absence of Anti-CSRF, Missing CSP, dll)* Menengah - Rendah (*Medium - Low*)	1. Terapkan *Content Security Policy* (CSP) pada *header* respons HTTP. 2. Implementasikan token *Anti-CSRF* pada setiap formulir *input* HTML.

Berdasarkan Tabel 6, kerentanan dengan tingkat risiko tertinggi yang ditemukan pada sistem adalah *SQL Injection* pada parameter URL id_pengajuan. Kerentanan ini berpotensi memungkinkan penyerang mengambil kueri basis data sehingga dapat mengakses data secara tidak sah. Selain itu, kerentanan risiko menengah juga teridentifikasi yang dimana penggunaan pustaka JavaScript versi lama (*Vulnerable JS Library*) serta kesalahan konfigurasi pada beberapa security headers, seperti tidak diterapkannya *Content Security Policy (CSP)* dan mekanisme *Anti-CSRF*. Oleh karena itu, penerapan rekomendasi mitigasi sangat penting untuk meningkatkan keamanan sistem dan meminimalkan potensi eksploitasi.

4. KESIMPULAN

Evaluasi tingkat keamanan sistem informasi pada subdomain akademik Fakultas Sains dan Teknologi UIN Sultan Syarif Kasim Riau dilakukan menggunakan metodologi NIST SP 800-115 melalui fase discovery dan attack. Hasil pengujian mengidentifikasi kerentanan berisiko kritis (critical) berupa SQL Injection pada parameter URL id_pengajuan dengan metode HTTP GET, yang tervalidasi melalui eksploitasi dan memungkinkan ekstraksi struktur basis data (MariaDB) tanpa otorisasi, sehingga secara langsung mengancam aspek kerahasiaan dan integritas data. Selain itu, ditemukan pula kerentanan berisiko menengah (medium), seperti penggunaan pustaka JavaScript usang (Moment.js 2.8.1) serta kelemahan konfigurasi header keamanan HTTP, termasuk tidak diterapkannya Content Security Policy dan proteksi Anti-CSRF, yang berkontribusi terhadap perluasan attack surface. Sebagai langkah mitigasi, direkomendasikan implementasi prepared statements untuk mencegah SQL Injection, pembaruan pustaka sistem, serta penguatan konfigurasi keamanan server. Secara keseluruhan, hasil evaluasi ini menegaskan pentingnya penguatan mekanisme validasi input dan konfigurasi keamanan sebagai upaya meningkatkan ketahanan sistem informasi berbasis web di lingkungan perguruan tinggi. Kontribusi utama penelitian ini terletak pada keberhasilan mengidentifikasi kerentanan kritis yang tidak terdeteksi oleh alat pemindaian otomatis melalui pendekatan eksploitasi manual, serta penyusunan rekomendasi mitigasi berbasis standar NIST SP 800-115 yang dapat dijadikan acuan praktis dalam peningkatan keamanan sistem informasi akademik.

BULLETIN OF COMPUTER SCIENCE RESEARCH
ISSN 2774-3659 (Media Online)
Vol 6, No 3, April 2026 | Hal 877-885
https://hostjournals.com/bulletincsr
DOI: 10.47065/bulletincsr.v6i3.1044

UCAPAN TERIMAKASIH

Penelitian ini sepenuhnya didukung oleh Universitas Islam Negeri Sultan Syarif Kasim Riau, khususnya melalui bantuan dan sumber daya yang tak ternilai dari Complexity Research Hub. Para penulis mengucapkan terima kasih yang tulus atas dukungan institusional, bimbingan, dan fasilitas yang telah memungkinakan penelitian ini.

REFERENCES

[1] I. M. A. S. Permana, I. G. P. K. Juliharta, and I. G. J. E. Putra, "Analisis Keamanan Sistem Informasi Menggunakan Metode Vulnerability Assesment pada Aplikasi Web Karangasem. go. id," *REMIK Ris. dan E-Jurnal Manaj. Inform. Komput.*, vol. 9, no. 2, pp. 466–473, 2025, doi: http://doi.org/10.33395/remik.v9i2.14561.

[2] H. H. Solihin *et al.*, *Konsep Sistem Informasi di Era Digital*. Kaizen Media Publishing, 2024.

[3] E. Z. Darojat, E. Sediyono, and I. Sembiring, "Vulnerability Assessment Website E-Government dengan NIST SP 800-115 dan OWASP Menggunakan Web Vulnerability Scanner," *J. Sist. Inf. Bisnis*, vol. 12, no. 1, pp. 36–44, 2022, doi: https://doi.org/10.21456/vol12iss1pp36-44.

[4] A. Afrizal and A. Angraini, "Perancangan Cetak Biru Teknologi Informasi Dengan Zachman Framework (Studi kasus: PTIPD UIN Suska Riau)," *J. Ilm. Rekayasa dan Manaj. Sist. Inf.*, vol. 2, no. 1, pp. 15–18, 2024, doi: http://dx.doi.org/10.24014/rmsi.v2i1.1687.

[5] I. Maita and M. R. Muttaqin, "Layanan Konsultasi Penasehat Akademik Berbasis Android di Fakultas Sains dan Teknologi UIN Suska Riau," *J. Sains, Teknol. dan Ind.*, 2023, doi: http://dx.doi.org/10.24014/sitekin.v19i2.16618.

[6] Z. A. Khan, "Penetration Testing Information System Security Assessment Framework (ISSAF)," *Penetration Test. Inf. Syst. Secur. Assess. Framew.*, vol. 4, no. 3, pp. 1593–1601, 2023, doi: https://doi.org/10.30865/klik.v4i3.1507.

[7] A. Agustinus and I. Sembiring, "Website Vulnerability Testing Using The Penetration Testing Method Referring To NIST SP 800–155 (Case Study (Astonprinter. com Domain))," *J. Tek. Inform.*, vol. 5, no. 6, pp. 1651–1662, 2024, doi: https://doi.org/10.52436/1.jutif.2024.5.6.3859.

[8] F. Mambo, D. Yuniarto, and D. Setiadi, "Evaluasi Keamanan Website dengan Menggunakan Metode NIST SP 800-115," *Pop. J. Penelit. Mhs.*, vol. 3, no. 4, pp. 255–264, 2024, doi: https://doi.org/10.58192/populer.v3i4.2805.

[9] S. A. Maherza, "Penetration testing terhadap website sekolah menengah atas ABC dengan metode NIST SP 800-115," Universitas Pembangunan Nasional Veteran Jakarta, 2022. [Online]. Available: http://repository.upnvj.ac.id/id/eprint/20860

[10] 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, doi: https://doi.org/10.30871/jaic.v9i6.9846.

[11] K. Scarfone, M. Souppaya, A. Cody, and A. Orebaugh, "Technical guide to information security testing and assessment," *NIST Spec. Publ.*, vol. 800, no. 115, pp. 2–25, 2023, doi: https://doi.org/10.6028/NIST.SP.800-115.

[12] M. Mifthahuddin, H. J. Setyadi, and M. R. Ibrahim, "Penetration Testing Website E-Journals Metode NIST SP 800-115 dan OWASP," *METIK J. (AKREDITASI SINTA 3)*, vol. 9, no. 1, pp. 72–81, 2025, doi: https://doi.org/10.47002/metik.v9i1.1030.

[13] R. A. Wibowo and S. Widyarto, "Kajian Pustaka: Penetration Testing dengan NIST SP 800-115 dan OSSTMM," in *Proceedings of the Informatics Conference*, 2020, pp. 96–111. [Online]. Available: https://ojs.journals.unisel.edu.my/index.php/icf/article/view/96

[14] S. Handaya and R. Islamadina, "Imlementasi Penetration Testing Pada Aplikasi Web Sistem Evaluasi Data Bidang Tik Polda Aceh Menggunakan Metode Owasp Dan Nist Sp 800-115," *Cybersp. J. Pendidik. Teknol. Inf.*, vol. 9, no. 1, pp. 27–41, 2025, doi: https://doi.org/10.22373/cj.v9i1.27978.

[15] A. Muhammad, A. I. Hadiana, and R. Ilyas, "Eksploitasi Broken Access Control Untuk Eskalasi Hak Akses Pada LMS Universitas XYZ," *J. Algoritm.*, vol. 22, no. 2, pp. 1–11, 2025, doi: https://doi.org/10.33364/algoritma/v.1-1.2287.

[16] M. Syani, R. Nurhakim, F. R. Pratama, H. Maulana, A. Nurdin, and B. Pamungkas, "Uji Keamanan Aplikasi Website XYZ Menggunakan Burp Suite Berdasarkan Kerangka NIST SP 800-115," *J. Sist. Inf. Galuh*, vol. 3, no. 2, pp. 54–60, 2025, doi: https://doi.org/10.25157/jsig.v3i2.4965.

[17] I. M. Raazi, M. Malahayati, B. Basrul, R. Malia, and M. Fadhli, "Analysis server security assessment of staffing management information system using the NIST SP 800-115 method at UIN Ar-Raniry Banda Aceh," *Circuit J. Ilm. Pendidik. Tek. Elektro*, vol. 8, no. 1, pp. 46–58, 2024, doi: https://doi.org/10.22373/crc.v8i1.20808.

[18] R. S. Wiandani, M. Tahir, I. A. Dyransyha, and R. Ummah, "Identifikasi Serangan SQL Injection Berbantuan Aplikasi Pengujian Keamanan Web DVWA (Damn Vulnerable Web Application)," *Digit. Transform. Technol.*, vol. 5, no. 1, pp. 375–382, 2025, doi: https://doi.org/10.47709/digitech.v5i1.5922.

[19] M. Syani, T. F. Mustafa, H. M. Falah, T. Rohayati, and U. A. Rosid, "Vulnerability Assessment pada Situs XYZ Menggunakan Web Vulnerability Scanner Burp Suite," *J. Sist. Inf. Galuh*, vol. 3, no. 2, pp. 47–53, 2025, doi: https://doi.org/10.25157/jsig.v3i2.4961.

[20] M. Arifudin, F. Z. Sholeha, and L. F. Umami, "Planning (Perencanaan) Dalam Manajemen Pendidikan Islam," *MA'ALIM J. Pendidik. Islam*, vol. 2, no. 02, pp. 162–183, 2021, doi: https://doi.org/10.21154/maalim.v2i2.3720.

[21] M. A. Rojabi, *Penetration Testing Profesional: Cara Menguasai Skill Hacking Legal*. Afdan Rojabi Publisher, 2025.

[22] G. T. Wandinil and R. Islamadina, "Penerapan Penetration Testing Pada Website Laporan Harian Polda Aceh Menggunakan Metode Nist," *J. Transform. Pendidik.*, vol. 6, no. 3, 2025, [Online]. Available: https://ejurnals.com/ojs/index.php/jtp/article/view/2738.