Analisis Keamanan Website Yayasan Pondok XYZ Menggunakan *OWASP Web Security Testing Guide*

1st M. Ato'ulloh
Fakultas Informatika
Universitas Telkom
Purwokerto, Indonesia
atoeks@student.telkomuniversity.ac.id

2nd Trihastuti Yuniati
Fakultas Informatika
Universitas Telkom
Purwokerto, Indonesia
trihastutiy@telkomuniversity.ac.id

Abstrak - Perkembangan teknologi digital membuat *website* menjadi *platform* utama untuk pengelolaan informasi, termasuk bagi lembaga pendidikan seperti Yayasan Pondok XYZ. Meskipun memberikan kemudahan, penggunaan website juga membawa ancaman keamanan siber. Keamanan website sangat penting, karena menyimpan data sensitif seperti informasi pribadi, pendidikan, dan keuangan santri. Ancaman seperti *Cross-Site Scripting (XSS), SQL Injection, Brute Force, Clickjacking,* dan DDoS dapat membahayakan data tersebut. Penelitian ini bertujuan untuk menganalisis keamanan website Yayasan Pondok XYZ melalui metode *Penetration Testing* yang mengikuti panduan OWASP *Web Security Testing Guide*. Pengujian dilakukan dengan alat seperti OWASP ZAP, SQLMap, WPScan, Burp Suite, dan LOIC untuk menguji beberapa serangan. Hasil penelitian menunjukkan bahwa website rentan terhadap serangan *Clickjacking* dan *DDoS,* namun tidak berhasil dieksploitasi oleh *XSS, SQL Injection*, atau *Brute Force*. Kerentanannya ditemukan pada konfigurasi server yang tidak optimal, seperti kurangnya pengaturan *X-Frame-Options* dan ketidakmampuan untuk menangani *DDoS*. Rekomendasi perbaikan termasuk menambahkan *X-Frame-Options*, menerapkan *Content Security Policy (CSP),* menggunakan *Web Application Firewall (WAF)* untuk DDoS, serta pembaruan rutin tema dan plugin WordPress. Mitra Yayasan Pondok XYZ menerima rekomendasi ini dan berkomitmen untuk memperbaiki pengaturan keamanan dan pemeliharaan sistem secara berkala.

I. PENDAHULUAN

Seiring dengan pesatnya perkembangan teknologi digital, website menjadi platform utama dalam pengelolaan informasi, termasuk di lembaga pendidikan seperti Yayasan Pondok XYZ. Meskipun membawa kemudahan, penggunaan teknologi ini juga meningkatkan risiko kejahatan siber, seperti penipuan dan pencurian data. Kejahatan siber memanfaatkan komputer dan internet untuk menyerang pihak lain. Website pesantren menjadi sasaran potensial bagi penjahat siber yang berusaha mencuri atau merusak data sensitif seperti informasi santri, keuangan, dan nilai pendidikan. Pada 24 Maret 2021, Pondok Pesantren X mengalami peretasan website berupa *defacement* yang mengganti konten situs dengan pesan yang merugikan, mengganggu proses pembelajaran jarak jauh [1]. Keamanan data sangat penting, mengingat website pesantren menyimpan data sensitif yang perlu dilindungi dari ancaman. Kesadaran akan pentingnya keamanan siber tidak hanya menjadi tanggung jawab administrator IT, tetapi juga santri dan pengajar [2]. Website Yayasan Pondok XYZ berfungsi untuk menyediakan informasi penting terkait pendaftaran, nilai, rapor, dan data santri. Oleh karena itu, pengujian terhadap keamanan website ini sangat penting untuk melindungi data dari ancaman. Penelitian ini menggunakan metode OWASP *Web Security Testing Guide (WSTG),* yang dikenal luas dalam pengujian keamanan siber. Metode ini mengidentifikasi kerentanannya melalui pengujian konfigurasi server, otentikasi, otorisasi, dan pengelolaan sesi [3]. Masalah utama adalah kurangnya pengujian menyeluruh pada website pesantren, yang membuatnya rentan terhadap ancaman. Penelitian ini bertujuan untuk mengidentifikasi kerentanannya dan memberikan rekomendasi perbaikan untuk meningkatkan keamanan. Hasil penelitian ini akan membantu Yayasan Pondok XYZ dalam melindungi data santri dan meningkatkan kesadaran akan pentingnya keamanan siber di kalangan pengajar dan santri.

II. KAJIAN TEORI

A. *Website*

Website adalah sekumpulan halaman yang menyajikan informasi dalam bentuk teks, gambar, animasi, suara, atau gabungan semuanya, baik statis maupun dinamis, yang saling terhubung. Situs website terdiri dari beberapa halaman dengan topik terkait, sering dilengkapi dengan gambar, video, atau file lainnya[4].

B. *Penetration Testing*

Penetration Testing adalah simulasi yang dilakukan untuk mendeteksi kerentanan dalam aplikasi, jaringan, dan sistem operasi. Proses ini mencakup analisis menyeluruh terhadap potensi kelemahan dalam sistem, seperti konfigurasi yang tidak aman dan kerentanannya pada perangkat lunak atau perangkat keras[5].

C. *Open Web Security Application Project (OWASP)*

OWASP adalah organisasi internasional *non-profit* yang didirikan pada 21 April 2004, berfokus pada peningkatan keamanan perangkat lunak dan membantu organisasi dalam mengembangkan dan memelihara aplikasi yang aman[4].

D. *Web Security Testing Guide (WSTG)*

WSTG adalah panduan untuk menguji keamanan situs web dengan mengevaluasi dan menganalisis aplikasi untuk mengidentifikasi kelemahan dan kerentanannya, termasuk pengujian seperti *Information Gathering, Authentication Testing, Input Validation Testing*, dan lainnya[3].

E. *OWASP Zed Attack Proxy (ZAP)*

ZAP adalah alat yang digunakan untuk melakukan *penetration testing* pada aplikasi web, memindai dan menganalisis kerentanannya dengan fitur pemindaian otomatis dan manual. ZAP juga dapat berfungsi sebagai server proxy untuk memonitor dan mengubah lalu lintas jaringan[6].

F. *DDoS (Distributed Denial of Service)*

DDoS adalah serangan siber yang bertujuan untuk membanjiri sistem, komputer, atau server dengan lalu lintas yang sangat tinggi, menguras sumber daya, dan menyebabkan layanan tidak dapat diakses oleh pengguna yang sah[7].

G. *SQL Injection*

SQL Injection adalah teknik serangan di mana penyerang menyisipkan kode SQL berbahaya ke dalam input aplikasi web untuk mengeksploitasi kerentanan dalam pengolahan input, memungkinkan akses atau kerusakan pada data di database[8].

H. *Burp Suite*

Burp Suite adalah alat keamanan web yang memungkinkan pemantauan dan analisis lalu lintas HTTP/HTTPS antara browser dan server, serta mendeteksi kerentanannya seperti *SQL Injection, XSS,* dan *CSRF*[9].

I. *Nmap (Network Mapper)*

Nmap adalah alat untuk memindai port pada sistem untuk mengidentifikasi layanan yang berjalan, sangat berguna dalam tahap pengintaian untuk mendeteksi port terbuka dan informasi tambahan terkait server[10].

J. *OWASP DirBuster*

DirBuster adalah alat untuk brute force direktori dan nama file yang tersembunyi di server web untuk menemukan path yang tidak terlindungi, yang mungkin mengandung informasi sensitif atau kerentanannya[11].

K. *Nikto*

Nikto adalah alat open-source yang digunakan untuk mendeteksi kerentanan pada aplikasi web, termasuk konfigurasi server yang tidak tepat dan file yang tidak terlindungi, serta mendeteksi kerentanannya seperti *SQL Injection* dan *XSS*[12].

L. *Slowloris*

Slowloris adalah serangan *DDoS* yang bekerja pada lapisan aplikasi dengan mengirimkan permintaan HTTP parsial ke server untuk mempertahankan banyak koneksi aktif dan menguras sumber daya server, membuatnya tidak dapat melayani permintaan dari pengguna sah[13].

M. *Low Orbit Ion Cannon*

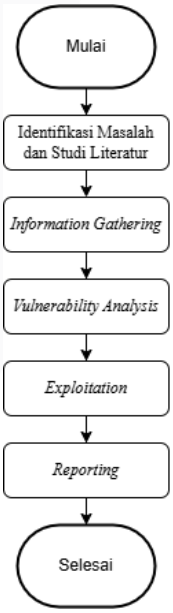
LOIC adalah alat yang mengirimkan permintaan TCP, UDP, atau HTTP dalam jumlah besar untuk membanjiri sistem, mengakibatkan layanan tidak dapat diakses. Meskipun mudah digunakan, LOIC tidak menyembunyikan identitas penyerangnya, yang membuatnya rentan untuk penyalahgunaan[14].

III. METODE

Penelitian yang digunakan ada pada situs website Institusi Perguruan Tinggi Yayasan Pondok XYZ, yang akan dianalisis untuk mengidentifikasi potensi kerentanannya dan mengevaluasi tingkat keamanannya. Pengujian dilakukan menggunakan metode *penetration testing* yang berlandaskan pada panduan *OWASP Web Security Testing Guide.* Metode ini dipilih karena OWASP merupakan salah satu standar yang paling umum digunakan dalam menguji keamanan aplikasi web. Panduan ini memberikan langkah-langkah yang jelas dan terstruktur untuk mengidentifikasi berbagai kerentanan yang ada, seperti masalah di pengaturan login, pengelolaan data, dan validasi input. Dengan menggunakan *OWASP*, pengujian dapat dilakukan secara menyeluruh untuk menemukan potensi ancaman yang mungkin tidak terlihat dengan pengujian biasa, sehingga membantu memastikan keamanan situs web dengan lebih baik.

A. Diagram Alir Penelitian

Dalam tahapan proses yang akan dilakukan pada penelitian ini, alur penelitian dapat digambarkan pada diagram alur yang terdapat pada Gambar 1 berikut:



GAMBAR 1
DIAGRAM ALIR PENELITIAN

Berdasarkan Gambar 1 diatas tahapan penelitian ini dapat dijelaskan sebagai berikut:

1. Identifikasi Masalah dan Studi Literatur

Pada identifikasi terhadap masalah keamanan yang mungkin terdapat di website. Proses ini melibatkan analisis risiko untuk mengidentifikasi potensi kerentanan dan

dampaknya terhadap website. Studi literatur yang mencakup penelitian terdahulu, standar keamanan, dan praktik terbaik yang relevan juga dilakukan. Penelitian ini memanfaatkan langkah langkah pada *OWASP Web Security Testing Guide* sebagai panduan utama untuk metode pengujian.

2. Information Gathering

Pada tahap ini, pengujian dimulai dengan mengumpulkan informasi terkait objek penelitian, yaitu website Yayasan XYZ, untuk mendeteksi potensi kerentanannya. Proses *Information Gathering* terbagi menjadi sepuluh tahapan sebagai berikut:

- *Conduct Search Engine Discovery and Recommencement for Information Leakage (WSTG-INFO-01):* Menggunakan mesin pencari dan *Google Dorking* untuk menemukan informasi sensitif yang terekspos..
- *Fingerprint Web Server (WSTG-INFO-02):* Mengidentifikasi jenis server menggunakan tools seperti Nmap dan Shodan.
- *Review Webserver Metafiles for Information Leakage (WSTG-INFO-03):* Memeriksa metadata dan konfigurasi publik dengan Wget dan cURL.
- *Enumerate Applications on Webserver (WSTG-INFO-04*): Mengidentifikasi aplikasi yang berjalan di server menggunakan Nmap dan nslookup.
- *Review Webpage Comments and Metadata for Information Leakage (WSTG-INFO-05):* Mengumpulkan informasi dari komentar dan source code dengan Wget dan cURL.
- *Identify Application Entry Points (WSTG-INFO-06):* Mengidentifikasi titik masuk aplikasi menggunakan *Burp Suite*.
- *Map Execution Paths through Application (WSTG-INFO-07):* Memetakan jalur eksekusi aplikasi dengan *Disrearch* dan *Spidering.*
- *Fingerprint Web Application Framework dan Web Application (WSTG-INFO-08 & WSTG-INFO-09):* Mengidentifikasi framework dan CMS menggunakan *Wappalyzer* dan *WhatWeb*.
- *Map Application Architecture (WSTG-INFO-10):* Pemetaan arsitektur aplikasi untuk perencanaan pengujian lebih lanjut.

3. Vulnerability Analysis

Pada tahap ini, dilakukan analisis kerentanan terhadap website Yayasan Pondok XYZ. Karena website dibangun menggunakan platform *WordPress*, pengujian dilakukan menggunakan dua alat, yaitu OWASP ZAP dan WPScan. OWASP ZAP digunakan untuk pemindaian umum terhadap kerentanannya, sementara WPScan digunakan untuk mendeteksi kerentanan yang berkaitan dengan *WordPress,* seperti tema, plugin, dan versi CMS. Hasil analisis ini akan memberikan informasi mengenai kerentanannya serta tingkat risikonya.

4. Exploitation

Pada tahap eksploitasi, peneliti melakukan pengujian terhadap celah keamanan website Yayasan XYZ berdasarkan temuan kerentanan pada tahap *Vulnerability Analysis*. Pengujian yang dilakukan meliputi:

- *Testing for Reflected Cross-Site Scripting (WSTG-INPV-01):* Menguji apakah aplikasi rentan terhadap *Reflected XSS* dengan menyisipkan skrip berbahaya melalui URL atau form input.
- *SQL Injection Testing (WSTG-CLNT-09):* Mengidentifikasi apakah input website dapat disalahgunakan untuk menyisipkan perintah SQL berbahaya, menggunakan tools seperti *SQLmap*.
- *Testing for Clickjacking (WSTG-INPV-05):* Menguji apakah website rentan terhadap *Clickjacking*, dengan mencoba memuat halaman target dalam elemen iframe transparan.
- *Testing for Weak Lockout Mechanism (WSTG-ATHN-03):* Menguji apakah sistem login rentan terhadap serangan *brute force*, yakni percobaan login yang tidak dibatasi.
- *Distributed Denial of Service (DDoS):* Menguji ketahanan website terhadap serangan *DDoS*, yang dapat menguras sumber daya server dan menghambat ketersediaan layanan.

5. Reporting

Pada tahap terakhir, peneliti menyusun laporan yang mencakup jenis serangan yang diuji, tools yang digunakan, status keberhasilan serangan, serta rekomendasi perbaikan dan solusi untuk mengatasi kerentanannya yang ditemukan.

IV. HASIL DAN PEMBAHASAN

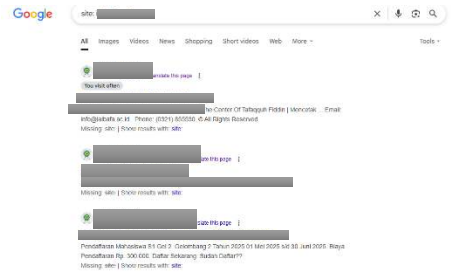
Pada bagian ini, akan dipaparkan hasil pengujian keamanan yang telah dilakukan terhadap website Yayasan XYZ. Pengujian menggunakan metode *penetration testing* yang mengacu pada panduan *OWASP Web Security Testing Guide*.

A. *Information Gathering*

Information Gathering adalah tahap pertama dalam *penetration testing* menurut *OWASP Web Security Testing Guide*, yang bertujuan untuk mengumpulkan data penting tentang target, seperti teknologi yang digunakan, konfigurasi server, dan titik masuk aplikasi.

1. Conduct Search Engine Discovery Reconnaissance for Information Leakage (WSTG-INFO-01)

Pada tahap ini, informasi dikumpulkan secara pasif dengan memanfaatkan mesin pencari seperti Google untuk menemukan data atau file yang tidak sengaja terekspos di website target. Teknik ini menggunakan *query* pencarian *khusus (Google Dorking)* untuk mengidentifikasi potensi kebocoran informasi, seperti direktori *plugin* yang terbuka, file konfigurasi, dokumen publik, serta *URL login* penting yang terindeks.

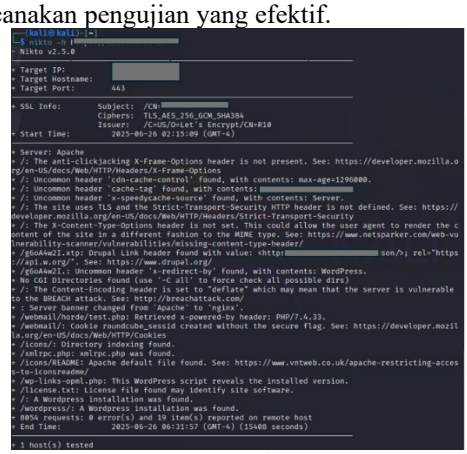


GAMBAR 2
HASIL PENELUSURAN MELALUI *SEARCH ENGINE GOOGLE*

Dengan menggunakan parameter *site* dan *filetype* pada mesin pencari *Google*, hasil pencarian disaring untuk menampilkan hanya file dengan format tertentu yang ada di website target. Dari hasil pencarian tersebut, ditemukan berbagai dokumen seperti surat edaran dan beberapa jurnal yang dapat diakses secara publik.

2. *Fingerprint Web Server (WSTG-INFO-02)*

Sidik jari server web (*fingerprint web server*) adalah proses untuk mengidentifikasi jenis dan versi perangkat lunak server yang digunakan oleh target, yang berguna untuk menentukan potensi kerentanannya. Proses ini sering dilakukan menggunakan alat otomatis, namun pemahaman dasar tentang cara kerjanya sangat penting untuk merencanakan pengujian yang efektif.



GAMBAR 3
HASIL SCANNING *NIKTO*

Pemindaian dengan Nikto mengonfirmasi penggunaan *Apache* dan menemukan kerentanannya, seperti absennya header *anti-clickjacking (X-Frame-Options)* dan pengaturan *Strict-Transport-Security* yang tidak ditemukan. Selain itu, header seperti *x-speedy-cache-source* dan *cache-tag* yang terdeteksi memberikan informasi lebih lanjut tentang konfigurasi server.

3. *Review Webserver Metafiles for Information Leakage (WSTG-INFO-03)*

Pada tahap ini, dilakukan pemeriksaan terhadap metadata server dan file konfigurasi yang dapat diakses publik untuk mengidentifikasi potensi kebocoran informasi.

GAMBAR 3
HASIL SCANNING CURL

Pemeriksaan header HTTP menggunakan perintah *curl -I* pada domain target mengonfirmasi bahwa server menggunakan *Apache* dan mengungkapkan adanya *endpoint REST API WordPress* (/wp-json/) yang dapat diakses publik.

4. *Enumerate Applications on Webserver (WSTG-INFO-04)*

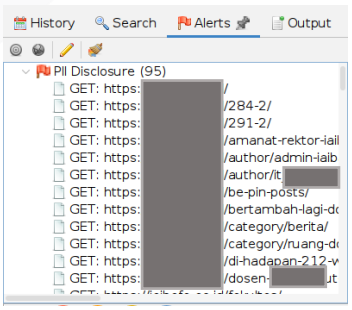
Enumerasi aplikasi pada *webserver* dilakukan untuk mengidentifikasi layanan yang berjalan di balik server target. Informasi ini sangat penting sebagai dasar untuk mengarahkan pengujian keamanan lebih lanjut, khususnya terhadap layanan yang berpotensi memiliki celah keamanan. Pengujian awal pada Gambar 4.6 dilakukan menggunakan alat Nmap dengan perintah *nmap -sV*

GAMBAR 4
HASIL *SCANNING NMAP*

Layanan-layanan yang terdeteksi menunjukkan bahwa server menjalankan berbagai fungsi, termasuk web server, *email server*, dan FTP. Beberapa layanan ini juga mencantumkan versi yang dapat digunakan untuk referensi identifikasi kerentanannya.

5. *Review Web Page Content for Information Leakage (WSTG-INFO-05)*

Review Web Page Content for Information Leakage adalah proses pengujian keamanan untuk menemukan informasi sensitif atau data internal yang secara tidak sengaja terekspos dalam konten halaman website, seperti komentar HTML, metadata, skrip, atau file konfigurasi yang dapat memberikan petunjuk atau celah bagi penyerang.



GAMBAR 5
HASIL SCANNING *NIKTO*

Pemindaian otomatis menggunakan OWASP ZAP dilakukan untuk mendeteksi kebocoran informasi sensitif yang mungkin tersembunyi dalam halaman website. Hasil pemindaian menunjukkan beberapa alert dengan kategori *PII Disclosure*, yang menunjukkan adanya potensi data

pribadi yang terekspos. Setelah analisis lebih lanjut, sebagian besar informasi yang ditemukan, seperti nomor *WhatsApp* dan kontak lainnya, ternyata sengaja dipublikasikan untuk keperluan umum dan komunikasi publik

6. *Identify Application Entry Points (WSTG-INFO-06)*

Identifikasi Titik Masuk Aplikasi adalah tahap penting dalam pengujian keamanan untuk mengetahui jalur interaksi yang tersedia bagi pengguna dan sistem eksternal. Titik masuk ini meliputi URL, endpoint API, form, dan parameter yang memungkinkan pengguna untuk memberikan input atau mengakses data. Identifikasi titik masuk ini penting untuk menentukan area yang akan diuji dalam pengujian keamanan aplikasi.

TABEL 1
HALAMAN NAVIGASI DAN KONTEN PUBLIK

No	*URL Path*	*Keterangan*
1	/	Halaman utama
2	/new_akademik	Halaman akademik baru
3	/284-2	Halaman 284-2
4	/291-2	Halaman 291-2
5	/amanat-rektor-xyz-pada-pelepasan...	Halaman informasi rektor
6	/author	Halaman penulis
7	/be-pin-posts	Halaman berisi postingan
8	/category	Halaman kategori
9	/comments	Halaman komentar

TABEL 2
REST API Endpoint (WordPress)

No	*URL Endpoint*	*Fungsi*
1	/wp-json/	API utama untuk integrasi
2	/wp-json/wp/v2/pages/	Mengakses halaman berbasis API
3	/wp-json/wp/v2/posts/	Akses konten/artikel via API

Tampilan struktur *entry point* tersebut dapat dilihat pada Gambar 4.12 yang menunjukkan struktur pohon dari panel *Sites* di OWASP ZAP. Visualisasi ini mendukung proses identifikasi dan validasi bahwa URL dan API tersebut benar-benar aktif dan merespons permintaan HTTP.

7. *Map Execution Paths Through Application (WSTG-INFO-07)*

Pemetaan jalur eksekusi aplikasi penting untuk mengidentifikasi interaksi pengguna atau sistem dengan aplikasi website. Proses ini memungkinkan pengujian terhadap alur logika aplikasi dan potensi kerentanannya.

Dalam penelitian ini, *OWASP ZAP* digunakan untuk memetakan jalur eksekusi aplikasi website target. Dengan fitur Spider dan hasil pada panel Sites, jalur interaksi pengguna dan aplikasi berhasil dilacak.

Berikut adalah daftar jalur eksekusi yang dipetakan:

TABEL 3
REST API Endpoint (WordPress)

No.	Halaman / Endpoint	Deskripsi
1	/	Halaman utama yang menampilkan informasi umum situs.
2	/new_akademik	Halaman yang berisi informasi mengenai akademik baru.
3	/284-2	Halaman informasi terkait dengan subkategori 284-2.
4	/291-2	Halaman informasi terkait dengan subkategori 291-2.
5	/amanat-rektor-xyz-pada-pelepasan...	Halaman yang berisi amanat rektor terkait kegiatan tertentu.
6	/author	Halaman yang berisi informasi penulis.
7	/be-pin-posts	Halaman yang berisi postingan terkait dengan informasi penting.
8	/category	Halaman kategori untuk konten yang terstruktur.
9	/comments	Halaman komentar atau feedback.

8. *Fingerprint Web Application Framework dan Web Application (WSTG-INFO-08 & WSTG-INFO-09)*

Fingerprinting pada aplikasi web adalah proses identifikasi teknologi yang mendasari aplikasi, termasuk *framework*, CMS, pustaka *JavaScript*, dan server web, yang bertujuan mengumpulkan informasi teknis untuk mengarahkan pengujian keamanan terhadap kerentanannya yang spesifik.

GAMBAR 6
HASIL *FINGERPRINTING* DENGAN WHATWEB

Hasil pemindaian menggunakan WhatWeb mengonfirmasi bahwa website target menggunakan server *Apache* dan *CMS WordPress* versi 6.8.1. Selain itu, terdeteksi juga penggunaan *framework* dan plugin utama seperti Elementor versi 3.28.3 dan pustaka *jQuery* versi 3.7.1. Informasi meta generator yang ditemukan juga mendukung hasil ini, memberikan gambaran lebih jelas tentang teknologi yang digunakan pada website target.

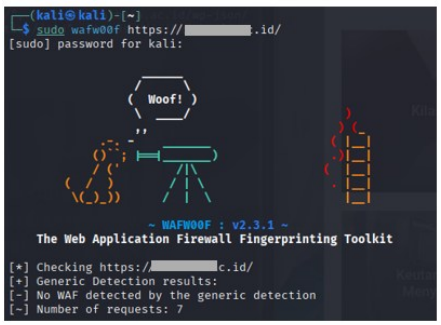
9. *Map Application Architecture (WSTG-INFO-10)*

Pada tahap ini, arsitektur aplikasi diidentifikasi berdasarkan hasil pemindaian dan *fingerprinting* yang dilakukan. Informasi ini penting untuk memahami teknologi yang digunakan dan merancang pengujian keamanan selanjutnya.

Berikut adalah komponen utama arsitektur aplikasi berdasarkan temuan:

- Web server: Apache HTTP Server
- Programming language: PHP versi 7.4.33
- Content Management System (CMS): WordPress versi 6.8.1

Gambar 6 menunjukkan hasil pemindaian dengan WhatWeb yang mengonfirmasi komponen tersebut. Namun, pemindaian lebih lanjut dengan wafw00f (Gambar 4.14) menunjukkan bahwa tidak ada Web Application Firewall (WAF) yang aktif, yang membuat server lebih rentan terhadap serangan seperti SQL Injection dan XSS.



GAMBAR 7
HASIL DARI WAF

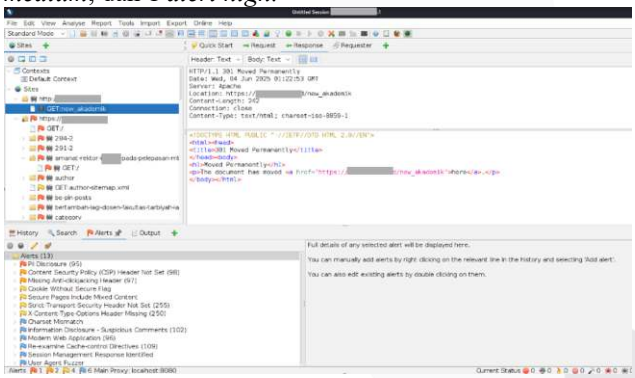
Aplikasi menggunakan teknologi umum WordPress dengan framework Elementor, namun tanpa perlindungan WAF, sehingga rentan terhadap beberapa jenis serangan.

B. *Vulnerability Analysis*

Pada tahap ini, kerentanan pada website dianalisis dengan melakukan pemindaian menggunakan berbagai tools keamanan. Dua tools yang digunakan untuk pemindaian website adalah ZAP dan WPScan.

1. Analisis Temuan dari *OWASP ZAP*

OWASP ZAP digunakan untuk melakukan pemindaian kerentanan pada website. Gambar 8 menunjukkan hasil pemindaian dengan ZAP yang mengidentifikasi sejumlah *alert*, dengan rincian 12 *alert* yang terdiri dari 5 *alert* tingkat keparahan informational, 4 *alert low*, 2 *alert medium*, dan 1 *alert high.*



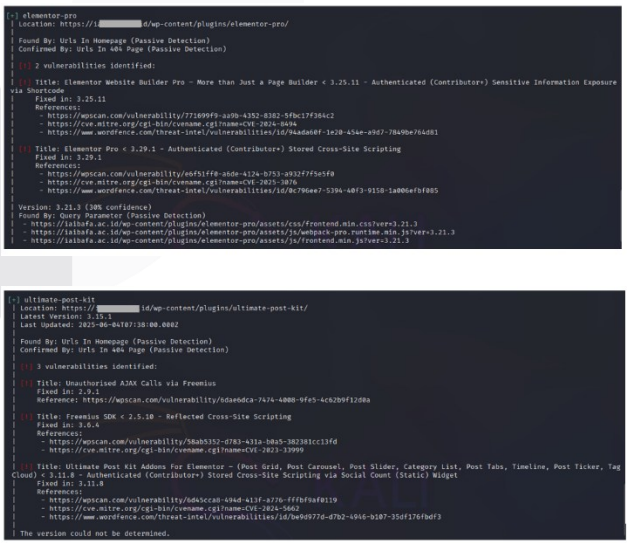
GAMBAR 8
HASIL DARI ZAP

Berdasarkan analisis dengan OWASP ZAP, berikut adalah deskripsi kerentanannya:

a. *PII Disclosure:* Kebocoran informasi pribadi yang dapat diidentifikasi (PII), seperti nomor kartu kredit atau data sensitif lainnya, yang bisa dimanfaatkan oleh pihak yang tidak bertanggung jawab.
b. *Content Security Policy (CSP) Header Not Set*: Tidak adanya header CSP yang meningkatkan risiko serangan seperti XSS dan injeksi, yang dapat memuat skrip berbahaya.
c. *Missing Anti-clickjacking Header:* Tidak adanya header untuk mencegah serangan *Clickjacking*, yang dapat memanipulasi klik pengguna menggunakan elemen tersembunyi.
d. *Cookie Without Secure Flag: Cookie* yang tidak mengatur flag Secure, memungkinkan pengiriman melalui HTTP yang tidak terenkripsi, meningkatkan risiko serangan MITM.
e. *Secure Pages Include Mixed Content:* Halaman HTTPS memuat elemen-elemen melalui HTTP, yang rentan terhadap serangan MITM.
f. *Strict-Transport-Security Header Not Set*: Tanpa header HSTS, situs rentan terhadap serangan downgrade yang memaksa komunikasi melalui HTTP yang tidak aman.
g. *Timestamp Disclosure – Unix*: Kebocoran informasi timestamp yang dapat digunakan untuk memperkirakan waktu serangan atau mengeksploitasi kelemahan.
h. *X-Content-Type-Options Header Missing*: Tanpa *header X-Content-Type-Options,* browser lama bisa melakukan MIME sniffing, berisiko menampilkan konten dengan jenis MIME yang salah.
i. *Charset Mismatch:* Ketidakcocokan charset antara header dan konten yang dimuat, yang dapat mempengaruhi pemrosesan karakter dan menambah kerentanannya.
j. *Information Disclosure – Suspicious Comments:* Pengungkapan komentar atau informasi yang tidak seharusnya terlihat oleh pengguna yang dapat memberikan petunjuk kepada penyerang.
k. *Modern Web Application:* Situs ini diidentifikasi sebagai aplikasi web modern, yang perlu pengawasan lebih ketat terhadap potensi kerentanannya.
l. *Re-examine Cache-control Directives:* Pengaturan cache-control perlu diperiksa untuk mencegah penyimpanan data sensitif di cache yang dapat diakses oleh pihak ketiga.
m. *Session Management Response Identified:* Pengelolaan sesi yang perlu diperiksa untuk memastikan sesi pengguna aman dari potensi session hijacking.

2. Analisis Temuan dari *WPScan*

Berdasarkan hasil *Information Gathering*, website target menggunakan CMS WordPress. Oleh karena itu, WPScan digunakan untuk menganalisis situs ini.



GAMBAR 9
HASIL DARI WPSCAN

Pemindaian dengan WPScan mengidentifikasi tema *Elementor* dan *Ultimate Post Kit* yang memiliki kerentanannya. Berikut adalah temuan kerentanannya:

TABEL 4
KERENTANANNYA PADA TEMA

Title	*Vulnerability*	*Fixed in Version*
Elementor-Pro	*Stored Cross-Site Scripting (XSS)*	3.29.1
Ultimate Post Kit	*Reflected XSS*	3.6.4
Ultimate Post Kit	*Stored Cross-Site Scripting via Widget*	3.11.8

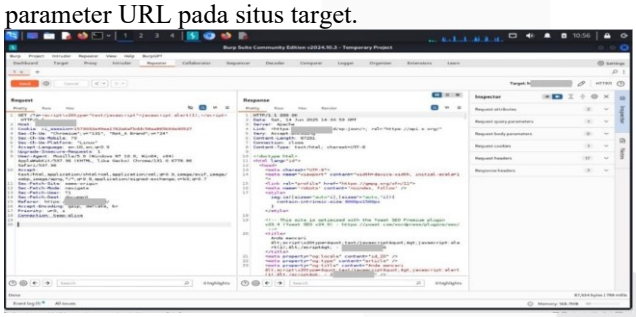
Elementor-Pro rentan terhadap *Stored XSS*, yang memungkinkan penyisipan skrip berbahaya. *Ultimate Post Kit* rentan terhadap *Reflected XSS* dan *Stored XSS* pada *widget* sosial. Pembaruan ke versi terbaru sangat disarankan untuk memperbaiki kerentanannya.

C. *Exploitation*

Exploitation dalam pengujian keamanan bertujuan untuk mengeksploitasi kerentanannya yang ditemukan selama tahap *Information Gathering* untuk memahami sejauh mana kerentanannya dapat dimanfaatkan. Pada pengujian situs target, beberapa kerentanan teridentifikasi, termasuk pada plugin *Elementor* yang rentan terhadap *Cross-Site Scripting (XSS)* dan beberapa misconfigurations di server. Berikut adalah langkah-langkah eksploitasi yang diterapkan:

1. *Testing for Reflected Cross Site Scripting (WSTG-INPV-01)*

Pada pengujian *Reflected Cross Site Scripting (XSS),* diuji apakah aplikasi rentan terhadap eksekusi skrip berbahaya yang dipantulkan oleh aplikasi setelah dimasukkan oleh pengguna. Pengujian dilakukan dengan mengirimkan payload XSS melalui kolom pencarian dan parameter URL pada situs target.



GAMBAR 10
PENGUJIAN DARI BURP SUITE

Pada gambar ini, terlihat bagaimana *Burp Suite* menangkap permintaan HTTP yang berisi payload XSS, serta respons yang dipantulkan oleh server. Respons ini menunjukkan bahwa aplikasi gagal melakukan sanitasi pada input, sehingga skrip berbahaya tetap dipantulkan kembali oleh server.

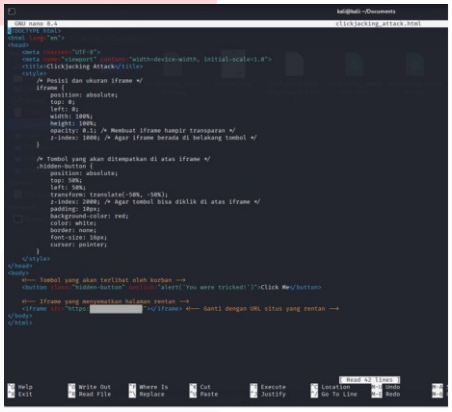
Hasil pengujian menunjukkan bahwa aplikasi rentan terhadap *Reflected XSS*. Meskipun alert box tidak muncul, skrip yang dimasukkan dipantulkan kembali sebagai teks biasa di halaman pencarian, yang menandakan bahwa aplikasi tidak melakukan penyaringan terhadap input pengguna.

GAMBAR 11
HASIL DARI XSS

Pada gambar ini, terlihat bahwa payload XSS yang dikirimkan muncul sebagai teks biasa di halaman pencarian, menunjukkan bahwa aplikasi tidak menyaring input dengan benar sebelum memantulkannya kembali.

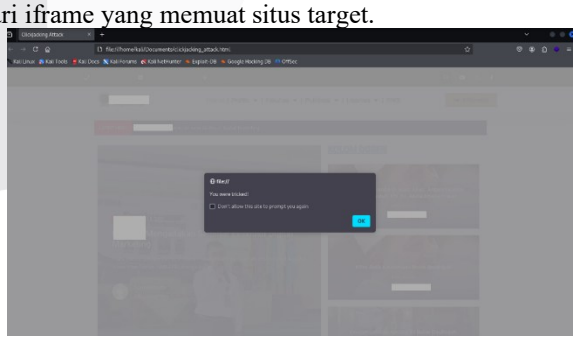
2. *Testing for Clickjacking (WSTG-CLNT-09)*

Pada pengujian ini, eksploitasi dilakukan terhadap kerentanan *clickjacking* yang ditemukan pada website target. *Clickjacking* terjadi jika situs tidak mengatur *X-Frame-Options*, yang memungkinkan halaman web rentan disematkan dalam iframe di situs lain tanpa sepengetahuan pengguna.



GAMBAR 12
Skrip Pengujian *Clickjacking*

Gambat 12 menunjukkan percobaan serangan *clickjacking* yang menggunakan skrip untuk menyematkan halaman website dalam iframe transparan pada halaman yang dikendalikan oleh penyerang. Pengguna yang mengunjungi halaman tersebut melihat tampilan biasa, namun mereka sebenarnya mengklik elemen tersembunyi dari iframe yang memuat situs target.



GAMBAR 13
HASIL PERCOBAAN PENGUJIAN *CLICKJACKING*

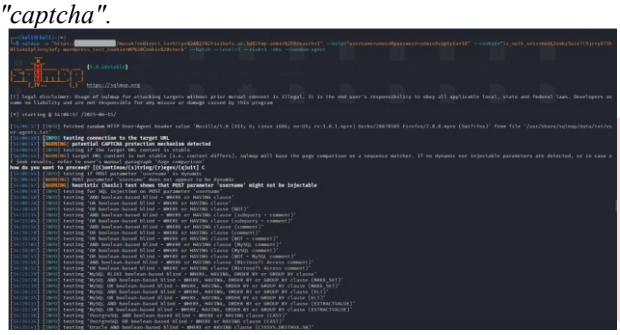
Gambar 13 memperlihatkan hasil pengujian, yang menunjukkan bahwa tanpa pengaturan *X-Frame-Options*, situs dapat dengan mudah disematkan pada situs lain dan dieksploitasi untuk *clickjacking*. Dalam percobaan ini, penyerang berhasil menampilkan pesan "*You were tricked!*" kepada pengguna yang tanpa sadar mengklik elemen tersembunyi. Hasil ini mengonfirmasi bahwa situs

target rentan terhadap serangan *clickjacking* jika X-Frame-Options tidak diatur.

Untuk mencegah serangan ini, sangat penting untuk segera mengonfigurasi *X-Frame-Options* pada situs untuk melindungi pengguna dari potensi eksploitasi.

3. *Testing for SQL Injection (WSTG-INPV-05)*

Pada tahap eksploitasi kerentanan, pengujian dilakukan untuk mengeksploitasi potensi kerentanan *SQL Injection* pada website yang diuji. Pengujian ini menggunakan alat SQLMap untuk mencoba memanipulasi parameter pada URL, seperti *"username", "password",* dan *"captcha"*.



GAMBAR 14
HASIL PERCOBAAN PENGUJIAN *SQLMAP*

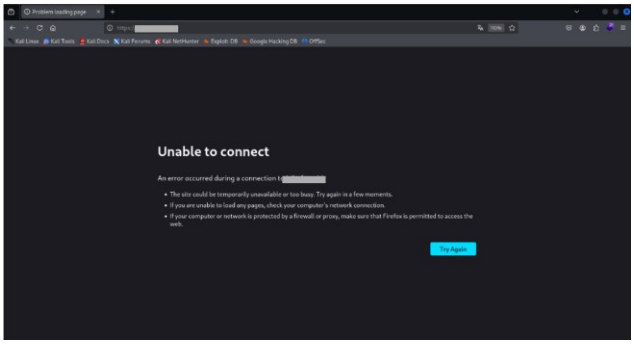
Gambar 14 menunjukkan hasil pengujian menggunakan *SQLMap*, yang menunjukkan bahwa meskipun percobaan injeksi SQL dilakukan pada parameter *username*, tidak ada kerentanannya yang dapat dieksploitasi. *SQLMap* memberikan peringatan bahwa parameter ini lebih statis dan tidak dinamis. Pengujian juga tidak menemukan parameter *POST* yang bisa dieksploitasi, mengindikasikan perlindungan yang baik atau pengamanan server yang diterapkan.

4. *Testing for Weak Lockout Mechanism (WSTG-ATHN-03)*

Pada pengujian ini, digunakan dua alat untuk menguji ketahanan website terhadap serangan *Distributed Denial of Service (DDoS)*, yaitu *LOIC (Low Orbit Ion Cannon)* dan *Slowloris*. Kedua alat ini digunakan untuk mengirimkan sejumlah besar paket data ke website target dengan tujuan untuk membanjiri server dan menyebabkan gangguan atau penghentian layanan.

GAMBAR 15
PERCOBAAN PENGUJIAN *LOIC & SLOWLORIS*

Pada Gambar 15, terlihat penggunaan LOIC, di mana peneliti mengatur URL dan IP website target serta mengirimkan 1000 thread melalui protokol HTTP pada port 80. Dengan pengaturan ini, LOIC melakukan serangan DDoS dengan membanjiri server website dengan sejumlah besar permintaan, yang dapat memperlambat atau menghentikan akses ke website.



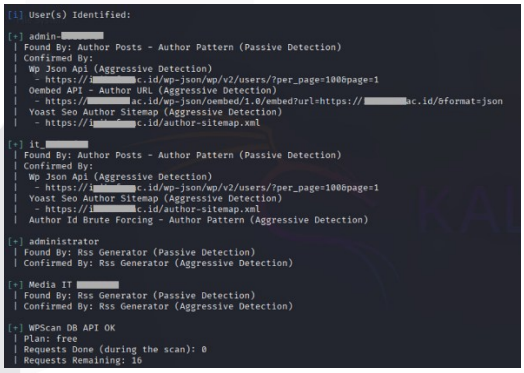
GAMBAR 16
Hasil Percobaan Pengujian *DDos*

Setelah serangan dilakukan, Gambar 16 menunjukkan bahwa website target tidak dapat diakses dan menampilkan pesan *"Unable to connect"*, yang mengindikasikan bahwa website mengalami gangguan akibat tidak dapat memproses semua permintaan yang datang dari serangan DDoS. Hal ini menunjukkan bahwa website target tidak memiliki perlindungan yang memadai terhadap serangan DDoS, yang menyebabkan server tidak dapat merespons permintaan pengguna.

5. *Testing for Weak Lockout Mechanism (WSTG-ATHN-03)*

Pada pengujian ini, WPScan dan DirBuster digunakan untuk mengidentifikasi kerentanan terhadap serangan brute force di situs target.

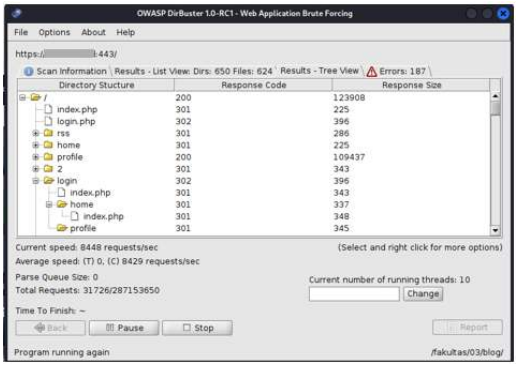
WPScan mendeteksi beberapa nama pengguna yang rentan, seperti *"admin-target"* dan *"administrator",* yang bisa menjadi target jika kata sandinya lemah. Meskipun percobaan menebak kata sandi gagal, pengujian menunjukkan bahwa tidak ada mekanisme penguncian akun setelah percobaan login gagal, memberi peluang bagi penyerang untuk mencoba berbagai kombinasi tanpa batasan.



GAMBAR 17
HASIL PENCARIAN *USERNAME*

GAMBAR 18
Hasil Pencarian *Password*

DirBuster menemukan beberapa direktori terbuka seperti */profile/* dan */fakultas/* yang dapat diakses tanpa batasan. Beberapa direktori penting lainnya mengembalikan respons *403 (Forbidden),* menunjukkan pengaturan akses yang tidak merata, yang meningkatkan potensi kerentanannya.



GAMBAR 19
Hasil Pengujian *DirBuster*

D. *Reporting*

Berdasarkan tahapan pengujian yang telah dilakukan, hasil pengujian dengan menggunakan metode *OWASP Testing Guide* pada situs Instansi Yayasan Pondok XYZ disajikan dalam laporan berikut yang mencakup hasil dan status pengujian.

TABEL 5
Hasil Pengujian

Jenis Serangan	*Tools*	Status
Cross Site Scripting (XSS)	*BurpSuite & Manual*	Gagal
Clickjacking	*Mousepad, Nano & Mozilla*	Berhasil
SQL Injection	*SQL Map*	Gagal
Brute Force	*Wpscan dan Dirbuster*	Gagal
Distributed Denial of Service (DDos)	*Low Orbit Ion Common (LOIC) dan Slowroris*	Berhasil

Dari lima serangan yang diuji pada tahap *exploitation*, hanya serangan *Clickjacking* dan *DDos* yang berhasil dieksploitasi. *Cross-Site Scripting (XSS)* dan *Brute Force* gagal karena dilindungi oleh *firewall Wordfence*, sementara *SQL Injection* tidak berhasil dieksekusi karena tidak adanya parameter *GET* atau *POST* pada halaman login admin. Serangan *DDoS* juga tidak berhasil karena sistem keamanan berhasil memblokir upaya akses dari penyerang.

TABEL 5
Solusi dan Saran Perbaikan

Vulnerability	Solusi
Clickjacking	- Menambahkan header X-Frame-Options untuk mencegah pemuatan dalam iframe.
	- Menerapkan Content Security Policy (CSP) dengan pengaturan frame-ancestors.
	- Mengonfigurasi cookie dengan atribut SameSite (Strict/Lax).
Distributed Denial of Service (DDoS)	- Menggunakan Web Application Firewall (WAF) untuk menyaring lalu lintas berbahaya.
Tema WordPress yang Usang	- Melakukan pembaruan berkala pada semua perangkat lunak yang digunakan.

E. Evaluasi dan Tindak Lanjut oleh Mitra

Setelah hasil analisis kerentanan dan rekomendasi perbaikan disampaikan, pihak mitra dari Yayasan Pondok XYZ memberikan respons terhadap saran yang diberikan. Mereka mengakui kerentanan *Clickjacking* sebagai prioritas dan siap menambahkan *header X-Frame-Options* serta mempertimbangkan penerapan *Content Security Policy (CSP)*. Untuk solusi lebih teknis, seperti pengaturan *cookie SameSite*, mitra membutuhkan pendampingan teknis lebih lanjut.

Terkait dengan ancaman DDoS, mitra menyadari pentingnya penerapan *Web Application Firewall (WAF)* namun terbentur masalah anggaran dan waktu, mengingat ujian akhir semester. Sebagai solusi sementara, mereka telah menerapkan *traffic filtering* pada router untuk memblokir lalu lintas berbahaya dan memastikan akses yang sah tetap berjalan. Hasilnya, jumlah permintaan yang berhasil diproses meningkat dari 6.447 menjadi 558, sementara permintaan yang gagal turun signifikan dari 7.598 menjadi 595.

Mitra juga berkomitmen untuk memperbarui tema dan plugin *WordPress* setelah masa ujian dan menyusun jadwal pemeliharaan untuk mencegah kerentanannya di masa depan.

Secara keseluruhan, meskipun implementasi solusi penuh akan dilakukan secara bertahap, pihak mitra menunjukkan komitmen untuk memperbaiki kerentanannya dengan langkah-langkah yang telah diidentifikasi.

V. KESIMPULAN

Berdasarkan pengujian dengan *OWASP Web Security Testing Guide*, situs Yayasan Pondok XYZ menunjukkan kerentanannya terhadap *Clickjacking* karena tidak mengatur *X-Frame-Options*, serta tidak mampu menangani serangan *DDoS*. Pengujian *Reflected XSS* berhasil dipantulkan, namun skrip tidak dieksekusi dengan aman, sementara *SQL Injection* dan *Brute Force* berhasil dicegah, menunjukkan perlindungan yang baik.

Mitra akan menambahkan *X-Frame-Options* untuk melindungi dari *Clickjacking*, serta menerapkan *Web Application Firewall (WAF)* untuk mengatasi *DDoS*. Sebagai langkah sementara, *traffic filtering* juga diterapkan pada router untuk mengurangi dampak serangan *DDoS*. Mitra juga berencana memperbarui tema dan plugin *WordPress* secara berkala, serta memperkuat pengelolaan kata sandi dan pengaturan akses untuk mengurangi potensi ancaman lainnya.

VI. DAFTAR PUSTAKA

[1] M. Fathurrahman, Zulhelman, and A. Aziz, "Vulnerability Assessment dan Penetration Test Pada Website MA/MTS Husnul Khatimah Kuningan," *ISAS Publ.*, vol. 8, no. 3, pp. 138–145, 2022.

[2] M. R. Maulana *et al.*, "Pengenalan dan Pemahaman Tentang Cyber Security di Pondok Pesantren Daarul Rahman III," *APPA J. Pengabdi. Kpd. Masy.*, vol. 1, no. 4, pp. 265–270, 2023.

[3] A. I. Rafeli, H. B. Seta, and I. W. Widi, "Pengujian Celah Keamanan Menggunakan Metode OWASP Web Security Testing Guide (WSTG) pada Website XYZ," *Inform. J. Ilmu Komput.*, vol. 18, no. 2, p. 97, 2022, doi: 10.52958/iftk.v18i2.4632.

[4] A. W. Kuncoro and F. Rahma, "Analisis Metode Open Web Application Security Project (OWASP) pada Pengujian Keamanan Website: Literature Review," *Automata*, vol. 3, no. 1, pp. 1–5, 2021.

[5] S. Utoro, B. A. Nugroho, M. Meinawati, and S. R.

Widianto, "Analisis Keamanan Website E-Learning SMKN 1 Cibatu Menggunakan Metode Penetration Testing Execution Standard," *Multinetics*, vol. 6, no. 2, pp. 169–178, 2020, doi: 10.32722/multinetics.v6i2.3432.

[6] G. Kusuma, "Implementasi Owasp Zap Untuk Pengujian Keamanan Sistem Informasi Akademik," *J. Teknol. Inf. J. Keilmuan dan Apl. Bid. Tek. Inform.*, vol. 16, no. 2, pp. 178–186, 2022, doi: 10.47111/jti.v16i2.3995.

[7] M. Ilman Aqilaa, D. Firdaus, and N. Naofal, "Identifikasi Serangan Lowrate Distributed Denial Of Services Dalam Jaringan Dengan Menggunakan Algoritma Adaboost," *Simpatik J. Sist. Inf. dan Inform.*, vol. 3, no. 1, pp. 34–41, 2023, doi: 10.31294/simpatik.v3i1.1829.

[8] D. P. Putranto, J. Jayanta, and B. Hananto, "Analisis Keamanan Website Leads UPNVJ Terhadap Serangan SQL Injection & Sniffing Attack," *Inform. J. Ilmu Komput.*, vol. 18, no. 3, p. 230, 2022, doi: 10.52958/iftk.v18i3.4690.

[9] M. A. Al Hilmi and R. K. Yunan, "Pengujian Keamanan Fitur Upload File pada Sistem Aplikasi Web," *J. Inform. J. Pengemb. IT*, vol. 7, no. 1, pp. 37–42, 2022, doi: 10.30591/jpit.v7i1.3336.

[10] M. Anif, S. Hws, and M. D. Huri, "Penerapan Intrusion Detection System (IDS) dengan metode Deteksi Port Scanning pada Jaringan Komputer di Politeknik Negeri Semarang," *J. TELE, Vol. 13 Nomor 1*, vol. 13, no. 1, pp. 25–30, 2015.

[11] A. R. Mukti, S. Rizal, F. S. Teknologi, U. B. Darma, and W. Analyzer, "PENINGKATAN KEAMANAN JARINGAN WIRELESS DI FAKULTAS," pp. 51–60.

[12] A. Aryapranata, Y. Al Rasyid, Y. P. Agsena, and S. Hermanto, "Keamanan Siber dalam Era Digital : Tantangan dan Solusi," no. October, 2024, doi: 10.55886/infokom.v8i2.932.

[13] M. Zidane, "Klasifikasi Serangan Distributed Denial-of-Service (DDoS) menggunakan Metode Data Mining Naïve Bayes," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 1, pp. 172–180, 2022, [Online]. Available: http://j-ptiik.ub.ac.id

[14] S. Surya, T. M. Diansyah, and R. Liza, "Paper Analisis Pemanfaatan Teknik Serangan DDOS pada Mikrotik Cloud dan Melakukan upaya penangannya".