ISSN (Online) 2747-0563
Volume 6 Tahun 2026

Analisis Kerentanan Keamanan pada Website E-Commerce

Arif Setyo Wibowo[1*], Henni Endah Wahanani[2], Andreas Nugroho Sihananto[3]

[1,2,3]Informatika, Universitas Pembangunan Nasional "Veteran" Jawa Timur
[1]henniendah.if@upnjatim.ac.id
[2]andreas.nugroho.jarkom@upnjatim.ac.id
*Corresponding author email: 21081010057@student.upnjatim.ac.id

Abstrak*— Perkembangan pesat e-commerce telah meningkatkan efisiensi transaksi digital, namun sekaligus memperbesar risiko ancaman keamanan siber. Penelitian ini bertujuan menganalisis kerentanan keamanan pada *website e-commerce* XYZ menggunakan metode penetration testing dengan acuan OWASP Top 10 2021. Tahapan penelitian meliputi *planning, information gathering, vulnerability analysis, exploitation*, dan *reporting*. *Information gathering* dilakukan menggunakan WHOIS, Nmap, dan Wappalyzer, sedangkan analisis kerentanan dilakukan secara otomatis menggunakan OWASP ZAP. Hasil penelitian menunjukkan bahwa ditemukan 10 kerentanan dengan tingkat risiko *High* dan *Medium*, di mana kategori Security *Misconfiguration* (A05) mendominasi temuan. Tahap *exploitation* membuktikan bahwa website target rentan terhadap serangan *Clickjacking*, tereksposnya kode sumber internal melalui halaman error debug Laravel, serta tidak menerapkan header Content-Security-Policy sehingga membuka peluang serangan XSS, serta memiliki konfigurasi CORS yang mengizinkan akses dari semua origin tanpa pembatasan. Seluruh 10 temuan dikonfirmasi valid, di mana 4 kerentanan dibuktikan melalui exploitation dan 6 temuan lainnya dikonfirmasi berdasarkan hasil pemindaian OWASP ZAP. Penelitian ini memberikan rekomendasi perbaikan guna meningkatkan keamanan sistem dan menjaga kepercayaan pengguna terhadap platform e-commerce.

Kata Kunci— **Penetration Testing, Keamanan Aplikasi Web, ECommerce, OWASP Top 10, Analisis Kerentanan**

I. PENDAHULUAN

Pemanfaatan aplikasi web dalam aktivitas perdagangan digital terus berkembang seiring meningkatnya penggunaan internet dalam sektor ekonomi. *e-commerce* membuka peluang bagi pelaku bisnis untuk menjalankan transaksi tanpa harus bertatap muka secara langsung. [1]. Namun, platform *e-commerce* juga menjadi sistem yang mengelola berbagai data penting, mulai dari identitas pengguna hingga informasi transaksi keuangan, sehingga platform *e-commerce* diharuskan memilik tingkat keamanan yang tinggi.

Aplikasi web *e-commerce* masih sering menjadi target serangan peretas. Web *e-commerce* yang rentan terhadap eksploitasi peretas dapat menimbulkan ancaman serius bagi pemilik maupun pengguna platform *e-commerce*, mulai dari kebocoran data pribadi, pencurian identitas, hingga kerugian finansial yang berdampak pada berbagai pihak. [2]. Selain menimbulkan gangguan teknis, serangan peretas juga dapat menyebabkan kerugian finansial serta menurunkan tingkat kepercayaan pengguna terhadap situs web *e-commerce* yang terdampak.

Banyak perancangan aplikasi *e-commerce* yang hanya berfokus pada pengembangan fitur-fitur dan *user experience* sehingga sering kali proses evaluasi kemanan dilakukan sangat minim. Minimnya pengujian keamanan dapat berpotensi meninggalkan celah pada sistem. Oleh karena itu, pengujian keamanan secara menyeluruh diperlukan untuk mengidentifikasi dan dapat mengurangi potensi kerentanan sebelum dieksploitasi oleh pihak yang tidak bertanggung jawab [3].

Metode yang umum digunakan pada industri digital untuk menguji tingkat keamanan aplikasi web adalah *penetration testing*. Metode ini bekerja dengan mensimulasikan serangan seperti *hacker* untuk mengetahui seberapa kuat aplikasi dalam menghadapi upaya eksploitasi dari luar[4]. Dalam penelitian ini, klasifikasi kerentanan mengacu pada OWASP Top 10, yaitu daftar sepuluh jenis kerentanan paling kritis yang sering ditemukan pada aplikasi web berdasarkan tingkat risiko dan frekuensi kemunculannya [5].

Penelitian ini menggunakan website *e-commerce* milik perusahaan XYZ sebagai studi kasus. XYZ merupakan perusahaan kasur yang sedang mencoba mengembangkan penjualan secara digital. Namun, hingga saat ini platform web *e-commerce* tersebut belum dilakukan pengujian keamanan secara menyeluruh. Kondisi ini dapat berpotensi menimbulkan risiko keamanan yang memengaruhi stabilitas layanan serta perlindungan data pengguna.

Berdasarkan permasalahan tersebut, tujuan penelitian ini dibuat untuk menganalisis kerentanan keamanan pada aplikasi web *e-commerce* XYZ dengan menggunakan pendekatan *penetration testing* serta klasifikasi kerentanan yang mengacu pada OWASP Top 10 2021. Hasil penelitian ini diharapkan dapat memberikan gambaran mengenai tingkat keamanan sistem sekaligus menjadi dasar dalam penyusunan rekomendasi perbaikan untuk meningkatkan keamanan aplikasi *e-commerce*.

II. TINJAUAN PUSTAKA

Bagian ini berisi kajian teoritis yang menjadi landasan dalam pelaksanaan penelitian, meliputi konsep website e-commerce, keamanan aplikasi web, penetration testing, OWASP Top 10, serta tools pengujian yang digunakan.

A. Website E - Commerce

Website *e-commerce* adalah platform online yang dimiliki oleh penjual untuk menjual produk secara digital. Sistem ini memberikan keuntungan dengan mempercepat proses transaksi dan meminimalisir anggaran operasional karena penjual tidak

ISSN (Online) 2747-0563
Volume 6 Tahun 2026

harus memiliki toko fisik. Aktivitas seperti penjualan, pemasaran, dan pembelian dilakukan melalui internet, sehingga memungkinkan konsep penjualan jarak jauh secara online [6].

B. Penetration Testing

Penetration testing merupakan metode pengujian keamanan sistem komputer atau sebuah sistem aplikasi dengan mensimulasikan serangan siber, yang dilakukan oleh ahli di bidang keamanan, atau pentester [7].

C. Owasp TOP 10

Dokumen standar ini bertujuan meningkatkan kesadaran bagi para pengembang dan praktisi keamanan aplikasi web. Dokumen ini memuat daftar risiko keamanan paling kritis yang sering dihadapi oleh aplikasi web [8]. OWASP Top 10 banyak digunakan sebagai standar acuan dalam proses evaluasi dan pengujian keamanan aplikasi web. Versi yang dirilis pada tahun 2021 mencakup 10 kategori kerentanan yang setiap kategorinya menunjukkan kelompok kerentanan yang berpotensi pada aplikasi web.

D. Owasp Zap

OWASP ZAP merupakan tool open source yang digunakan untuk pengujian keamanan pada website dan dikembangkan oleh *Open Web Application Security Project* (OWASP). ZAP juga dikenal sebagai "*manipulator-in-the-middle proxy*", yang bekerja dengan menempatkan dirinya di antara peramban pentester dan aplikasi web sehingga dapat mencegat serta memeriksa pesan yang dikirimkan di antara keduanya [9].

E. Wappalyzer

Wappalyzer adalah ekstensi browser lintas platform yang berfungsi untuk mengenali teknologi yang digunakan pada sebuah website [10]. Teknologinya seperti framework, web server, bahasa pemrograman, database yang digunakan, hingga library. Pada *penetration testing, tools* ini berperan pada tahap information gathering untuk memperoleh gambaran awal mengenai teknologi yang digunakan pada sebuuah system website.

F. Burp Suite

Burp Suite merupakan *tools penetration testing* populer yang digunakan untuk menguji keamanan aplikasi web dan memantau lalu lintas jaringan [11]. Fitur-fitur utama Burp Suite seperti *intercepting proxy, scanner, repeater,* dan *intruder.*

G. Hacking

Hacking adalah kegiatan yang sering kali memiliki citra negatif di masyarakat yang sebernarnya memiliki tujuan bagus untuk mencari celah pada suatu system dan memperkuatnya[12]. Di era digital, keahlian *hacking* sangat dibutuhkan oleh perusahaan-perusahaan besar, lembaga pemerintahan, dan organisasi untuk melindungi aset digital mereka dari ancaman serangan *hacker*.

H. Hacker

Hacker adalah seseorang yang ahli di bidang komputer, mampu membuat dan membaca program tertentu, serta sangat tertarik dengan keamanan sistem digital hingga rela menghabiskan waktu untuk memahami dan menguji batas-batasnya[13]. Dalam dunia *hacking*, terdapat dua jenis *hacker* yang memiliki tujuan berbeda dan saling betolak belakang. *White hat hacker* akan melaporkan setiap celah keamanan yang ditemukan kepada pihak terkait, sedangkan *black hat hacker* justru memanfaatkan celah tersebut untuk kepentingan pribadi mereka tanpa peduli pada dampak yang ditimbulkan.

I. Nmap

Nmap adalah *tool opensource* yang digunakan untuk audit keamanan jaringan yang mampu mendeteksi *host, service*, sistem operasi, hingga firewall yang digunakan[14]. Selain itu nmap juga merupakan *tool* yang sering dipakai oleh para pentester.

J. Whois

Whois merupakan *tool* yang digunakan untuk mengumpulkan informasi awal sebuah website melalui domainnya, mencakup data seperti kepemilikan domain, masa berlaku, hingga nama server yang digunakan[15].

III. METODELOGI PENELITIAN

Penelitian ini menggunakan metode *penetration testing* untuk mengidentifikasi dan menganalisis kerentanan keamanan pada aplikasi web *e-commerce* XYZ.
Tahapan penelitian meliputi empat fase seperti yang ditunjukkan pada Gbr. 1., yaitu *planning, information gathering, vulnerability analysis, exploitation* dan *reporting*.

Gbr. 1 Tahapan Penelitian.

A. Planning

Pada tahap *planning*, ditetapkan ruang lingkup pengujian, target sistem, serta batasan pengujian agar proses pengujian tidak mengganggu operasional aplikasi yang sedang berjalan. Tahap ini juga mencakup penetapan tujuan pengujian dan persetujuan dari pihak perusahaan XYZ selaku pemilik sistem.

B. Information gathering

Tahap *information gathering* dilakukan untuk mengumpulkan informasi awal mengenai aplikasi target. Proses ini mencakup identifikasi struktur aplikasi, teknologi yang digunakan, serta endpoint yang tersedia melalui pendekatan pasif menggunakan

ISSN (Online) 2747-0563
Volume 6 Tahun 2026

Wappalyzer dan pendekatan aktif menggunakan OWASP ZAP. Informasi yang diperoleh menjadi dasar dalam proses analisis keamanan pada tahap selanjutnya.

C. Vulnerability analysis

Tahap *vulnerability analysis* bertujuan untuk mengidentifikasi potensi celah keamanan pada aplikasi web. Pengujian dilakukan secara otomatis menggunakan OWASP ZAP dengan fokus pada aspek autentikasi, pengelolaan sesi, input pengguna, dan konfigurasi aplikasi. Kerentanan yang ditemukan kemudian diklasifikasikan berdasarkan kategori OWASP Top 10 untuk memudahkan analisis tingkat risiko yang ditimbulkan.

D. Exploitation

Tahap *exploitation* dilakukan setelah proses *vulnerability analysis* selesai dilaksanakan, guna membuktikan dampak dari kerentanan yang berhasil diidentifikasi pada tahap sebelumnya. Proses eksploitasi dilakukan dengan memperhatikan batasan-batasan dari tahap *planning, exploitation* dilakukan menggunakan Burp Suite dengan menerapkan teknik serangan yang relevan sesuai kategori kerentanan. Hasil dari tahap ini berupa bukti eksploitasi (*proof of concept*) yang mendokumentasikan sejauh mana celah keamanan tersebut dapat dimanfaatkan oleh penyerang untuk mendapatkan akses tidak sah maupun mencuri data sensitif..

E. Reporting

Tahap *reporting* merupakan tahap akhir di mana seluruh hasil pengujian disusun dalam bentuk laporan yang memuat jenis kerentanan yang ditemukan, tingkat risiko, dampak potensial, serta rekomendasi perbaikan sebagai upaya peningkatan keamanan aplikasi web *e-commerce* XYZ.

IV. HASIL DAN PEMBAHASAN

Pengujian keamanan web *e-commerce* XYZ dilakukan melalui 5 tahapan mulai dari *planning, information gathering, vulnerability analysis, exploitation* dan *reporting*.

A. Tahap Planning

Pada tahap *planning*, ruang lingkup pengujian ditetapkan secara formal dengan target mencakup seluruh bagian aplikasi yang dapat diakses secara publik. Batasan pengujian yang disepakati, meliputi larangan eksploitasi penuh terhadap celah yang ditemukan, pembatasan waktu pengujian, serta kewajiban menjaga kerahasiaan data. Sebelum pengujian dimulai, persetujuan tertulis dalam bentuk *Rules of Engagement* diperoleh, mencakup izin pengujian, metode yang diperbolehkan. Hasil tahap planning disajikan pada Tabel 1.

TABEL I
HASIL *PLANNING*

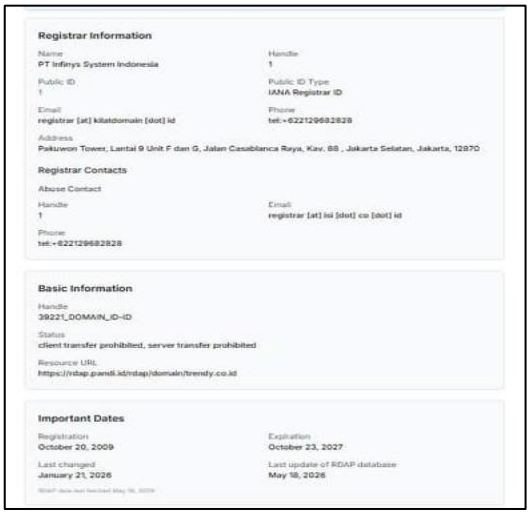
No	Aspek	Keterangan
1.	Ruang Lingkup Pengujian	Website *e-commerce* XYZ
2.	Batasan Pengujian	Batasan pengujian ditetapkan dalam dokumen RoE
3.	Urgensi Pengujian	Website belum pernah dilakukan pengujian keamanan sebelumnya
4.	Metode yang Digunakan	Pengujian mengacu pada kerangka OWASP Top 10 2021

B. Tahap Information gathering

Tahap information gathering dilakukan menggunakan WHOIS untuk identifikasi domain, Nmap untuk pemindaian port, dan Wappalyzer untuk mendeteksi teknologi yang digunakan aplikasi web target.

1) Whois

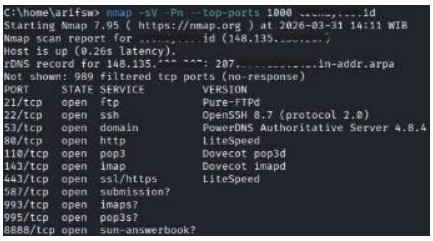
Dari hasil whois ditemukan bahwa domain terdaftar pada PT Infinys System Indonesia sejak 20 Oktober 2009 dengan masa berlaku hingga 23 Oktober 2027. Pengelolaan domain dilakukan melalui layanan Kilatdomain seperti yang terlihat pada Gbr. 2.,



Gbr. 2 Hasil Whois

2) Nmap

Nmap dilakukan untuk scanning port dan service yang digunakan oleh website target dan didapat beberapa port dan service seperti pada Gbr. 3.,



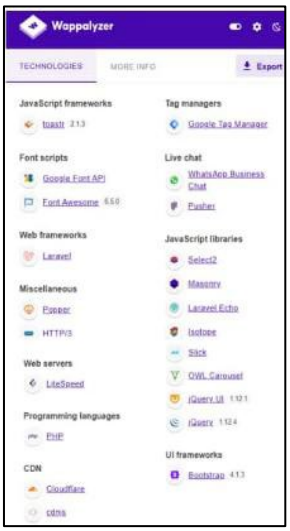
Gbr. 3 Hasil Nmap

Dari 1000 port yang dipindai, 989 port difilter dan hanya 11 port yang terbuka. Port 21 (FTP) perlu mendapat perhatian khusus karena tidak terenkripsi sehingga data berpotensi disadap.

ISSN (Online) 2747-0563
Volume 6 Tahun 2026

3) Wappalyzer

Teknologi yang digunakan meliputi framework Laravel, bahasa pemrograman PHP, web server LiteSpeed, antarmuka Bootstrap 4.1.3, serta penggunaan layanan CDN Cloudflare. Hasil seluruhnya dapat dilihat pada Gbr. 4.,

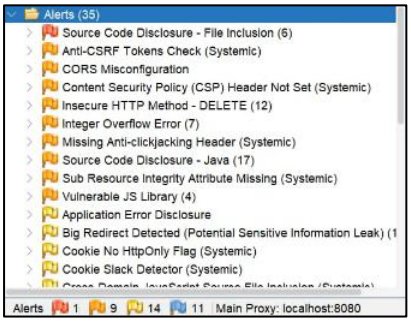


Gbr. 4. Hasil dari pemindaian Wappalyzer.

Hasil tahap *information gathering* menunjukkan bahwa aplikasi dibangun menggunakan *framework* Laravel dan dijalankan pada *web server* LiteSpeed dengan dukungan CDN Cloudflare.

C. Tahap Vulnerability Analysis

Vulnerability analysis dilaksanakan menggunakan pendekatan otomatis dengan menggunakan tools OWASP ZAP. Dari proses pemindaian yang dilakukan, OWASP ZAP berhasil mengidentifikasi 35 alerts yang terdiri dari 1 *alert High*, 9 *Medium*, 14 *Low*, dan 11 *Informational*. Pengelompokan kerentanan berdasarkan kategori OWASP Top 10 ditunjukkan pada Gbr. 5.



Gbr. 5. Hasil dari pemindaian OWASP ZAP.

Dari 35 *alert* tersebut, sebanyak 11 di antaranya bersifat *informational* dan tidak merepresentasikan celah yang dapat dieksploitasi secara langsung, sehingga tidak dibahas lebih lanjut dalam penelitian ini. Oleh karena itu, Tabel 2 hanya menyajikan 10 temuan yang memiliki tingkat risiko *High* dan *Medium* dan dianggap relevan untuk dianalisis lebih mendalam.

TABEL 2
HASIL IDENTIFIKASI KERENTANAN OWASP ZAP 2.17.0

No	Nama Kerentanan	Risk	Kategori OWASP 2021
1.	*Source Code Disclosure - File Inclusion*	*High*	*A05 - Security Misconfiguration*
2.	*Anti-CSRF Tokens Check*	*Medium*	*A01 - Broken Access Control*
3.	*CORS Misconfiguration*	*Medium*	*A01 - Broken Access Control*
4.	*Content Security Policy (CSP) Header Not Set*	*Medium*	*A05 - Security Misconfiguration*
5.	*Insecure HTTP Method - DELETE*	*Medium*	*A05 - Security Misconfiguration*
6.	*Integer Overflow Error*	*Medium*	*A03 - Injection*
7.	*Missing Anti-clickjacking Header*	*Medium*	*A05 - Security Misconfiguration*
8.	*Source Code Disclosure - Java*	*Medium*	*A05 - Security Misconfiguration*
9.	*Sub Resource Integrity Attribute Missing*	*Medium*	*A05 - Security Misconfiguration*
10.	*Vulnerable JS Library*	*Medium*	*A06 - Vulnerable and Outdated Components*

Dari 10 temuan tersebut, 1 kerentanan berkategori *High* yaitu *Source Code Disclosure - File Inclusion* yang termasuk dalam kategori A05 - *Security Misconfiguration*, menunjukkan bahwa terdapat potensi kebocoran kode sumber aplikasi yang dapat dimanfaatkan penyerang untuk memahami struktur sistem secara mendalam.

D. Exploitation

Tahap *exploitation* dilakukan sebagai lanjutan dari tahap *vulnerability analysis* untuk membuktikan dampak dari setiap kerentanan yang ditemukan.

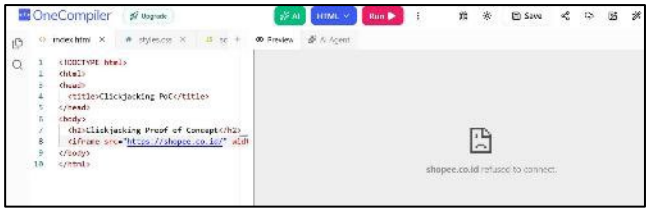
1) Clickjacking

Pengujian Clickjacking dilakukan dengan cara menyematkan URL target ke dalam elemen iframe pada halaman HTML yang dibuat secara khusus. Apabila halaman target berhasil dimuat di dalam iframe, maka dapat disimpulkan bahwa aplikasi tidak menerapkan proteksi terhadap serangan Clickjacking. yang dapat dilihat pada Gbr. 6., bahwa url target dapat ditempelkan di iframe html.

Gbr. 6. Hasil *clickjacking* website target.

ISSN (Online) 2747-0563
Volume 6 Tahun 2026

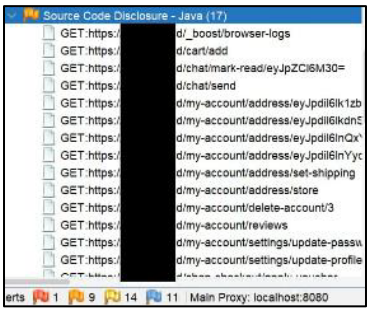
sebagai perbedaan jika menempelkan url *website marketplace* shopee pada iframe, maka yang dihasilkan adalah halaman *blank* dengan *text* shopee.co.id *refused to connect* tanpa ada isinya seperti pada Gbr. 7., yang menandakan bahwa website tersebut telah menerapkan proteksi terhadap serangan *Clickjacking*, sehingga *browser* menolak untuk memuat halaman di dalam iframe.



Gbr. 7. Hasil *clickjacking* halaman shopee.

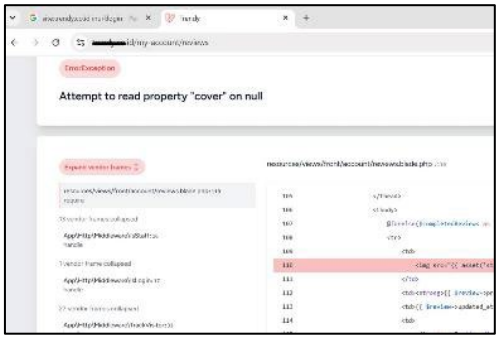
2) Source Code Disclosure

Pengujian *Source Code Disclosure - Java* bertujuan membuktikan bahwa aplikasi mengekspos kode sumber internal melalui halaman error yang tidak seharusnya tampil di lingkungan production. Pengujian dilakukan dengan memeriksa detail alert pada OWASP ZAP untuk memperoleh URL yang terdampak beserta bukti kode sumber yang terekspos, sebagaimana ditunjukkan pada Gbr. 8.



Gbr. 8. Detail *Alert Source Code Disclosure* - Java

Berdasarkan Gbr. 8, OWASP ZAP mendeteksi kerentanan *Source Code Disclosure* pada 17 URL dengan tingkat risiko *Medium*. Verifikasi dilakukan dengan mengakses salah satu URL yang terdampak yaitu */my-account/reviews* melalui browser, sebagaimana ditunjukkan pada Gbr. 9.

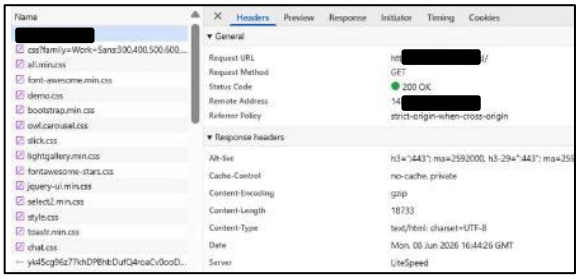


Gbr. 9. Halaman *Error* Laravel pada /my-account/reviews

Berdasarkan Gbr. 9, url */my-account/reviews* pada website target menampilkan halaman *error debug* Laravel dengan pesan "*Attempt to read property 'cover' on null*" yang tidak seharusnya terekspos di lingkungan production. Versi PHP 8.3.21 dan Laravel 11.46.1 terekspos pada halaman tersebut. Selain itu, struktur file internal, middleware seperti *App\Http\Middleware\IsStaff, IsLogin,* dan *TrackVisitor*, serta sebagian kode sumber dari halaman view terekspos, memungkinkan penyerang mempelajari logika internal aplikasi secara langsung.

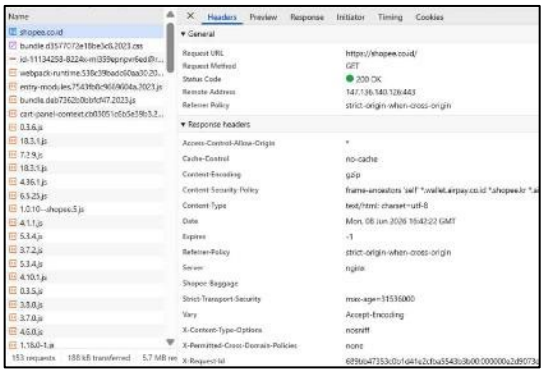
3) Content Security Policy Not Set

Pengujian *Content Security Policy* (CSP) *Header Not Set* dilakukan untuk membuktikan bahwa aplikasi target tidak menerapkan kebijakan keamanan konten yang dapat melindungi pengguna dari serangan injeksi skrip berbahaya. Pembuktian dilakukan dengan menganalisis response header server menggunakan DevTools browser, sebagaimana ditunjukkan pada Gbr. 10.



Gbr. 10. *Response Header* xyz.com

Berdasarkan Gbr. 10, *response header website* target tidak mengandung header *Content-Security-Policy*, sehingga tidak terdapat pembatasan terhadap sumber konten yang dapat ditampilkan browser. Sebagai perbandingan, pada Gbr. 11 menunjukkan response header Shopee yang telah menerapkan *Content-Security-Policy* dengan direktif lengkap, *Strict-Transport-Security, X-Content-Type-Options,* serta *Referrer-Policy.*



Gbr. 11. *Response Header* shopee.co,id

Ketiadaan CSP pada *website* target membuka celah bagi *hacker* untuk menyisipkan skrip berbahaya dari sumber eksternal, serta memungkinkan serangan XSS berjalan tanpa pembatasan

ISSN (Online) 2747-0563
Volume 6 Tahun 2026

karena tidak ada mekanisme yang mencegah eksekusi skrip yang tidak sah.

4) CORS Misconfiguration

Pengujian CORS *Misconfiguration* dilakukan berdasarkan temuan kategori A01 - Broken Access Control pada OWASP Top 10 2021. Verifikasi dilakukan menggunakan Burp Suite Repeater dengan mengirimkan request ke endpoint /api/get-harga menggunakan header Origin: https://evil.com untuk mensimulasikan request dari domain berbahaya, sebagaimana ditunjukkan pada Gbr. 12.

Gbr. 12. *Hasil Pengujian CORS Misconfiguration via Burp Suite Repeater*

Berdasarkan Gbr. 12, server mengembalikan response HTTP/2 200 dengan header *Access-Control-Allow-Origin*: *, yang membuktikan bahwa server mengizinkan akses dari semua origin tanpa pembatasan. Kondisi ini memungkinkan penyerang mengambil data dari endpoint API target menggunakan sesi pengguna yang sedang aktif.

E. Reporting

Berdasarkan keseluruhan proses pengujian, ditemukan 10 kerentanan dengan tingkat risiko High dan Medium yang relevan untuk ditindaklanjuti. Seluruh 10 temuan dikonfirmasi valid berdasarkan hasil pemindaian OWASP ZAP, di mana 4 kerentanan dibuktikan lebih lanjut melalui tahap exploitation sebagai proof of concept, sementara 6 temuan lainnya tidak dieksploitasi karena berada di luar ruang lingkup yang ditetapkan pada tahap planning. Hasil validasi seluruh temuan disajikan pada Tabel 3.

TABEL 3
HASIL VALIDASI KERENTANAN

No	Temuan	*Risk*	Hasil Validasi
1.	*Source Code Disclosure - File Inclusion*	High	Terkonfirmasi (pemindaian ZAP)
2.	*Anti-CSRF Tokens Check*	*Medium*	Terkonfirmasi (pemindaian ZAP)
3.	*CORS Misconfiguration*	*Medium*	Terkonfirmasi (*exploitation*)
4.	*Content Security Policy (CSP) Header Not Set*	*Medium*	Terkonfirmasi (*exploitation*)
5.	*Insecure HTTP Method - DELETE*	*Medium*	Terkonfirmasi (pemindaian ZAP)
6.	*Integer Overflow Error*	*Medium*	Terkonfirmasi (pemindaian ZAP)
7.	*Missing Anti-clickjacking Header*	*Medium*	Terkonfirmasi (*exploitation*)
8.	*Source Code Disclosure - Java*	*Medium*	Terkonfirmasi (*exploitation*)
9.	*Sub Resource Integrity Attribute Missing*	*Medium*	Terkonfirmasi (pemindaian ZAP)
10.	*Vulnerable JS Library*	*Medium*	Terkonfirmasi (pemindaian ZAP)

Rekomendasi perbaikan difokuskan pada kerentanan yang telah dieksploitasi dan memerlukan penanganan prioritas, sebagaimana disajikan pada Tabel 4.

TABEL 4
REKOMENDASI PERBAIKAN HASIL PENETRATION TESTING

No	Temuan	*Risk*	Rekomendasi Perbaikan
1.	*Source Code Disclosure*	High	Menonaktifkan mode debug Laravel di lingkungan production dengan mengatur APP_DEBUG menjadi false pada file .env
2.	*Clickjacking*	*Medium*	Menambahkan header X-Frame-Options dengan nilai DENY atau SAMEORIGIN
3.	*CSP Header Not Set*	*Medium*	Menambahkan header *Content-Security-Policy* dengan dokumentasi Laravel 11 untuk membatasi sumber konten yang dapat dimuat browser
4.	*CORS Misconfiguration*	*Medium*	Membatasi nilai header *Access-Control-Allow-Origin* dengan menentukan domain yang diizinkan secara spesifik, serta menghindari penggunaan wildcard (*) pada endpoint yang mengembalikan data sensitif

V. KESIMPULAN

Berdasarkan pengujian penetrasi yang telah dilakukan terhadap website e-commerce XYZ, ditemukan 10 kerentanan dengan tingkat risiko *High* dan *Medium* dari hasil pemindaian OWASP ZAP. Kategori *Security Misconfiguration* (A05) mendominasi temuan dengan 5 alert, diikuti kategori *Broken Access Control* (A01), *Injection* (A03), dan *Vulnerable and Outdated Components* (A06). Tahap *exploitation* membuktikan bahwa website target rentan terhadap serangan *Clickjacking*, mengekspos kode sumber internal melalui halaman error debug Laravel, serta tidak menerapkan header *Content-Security-Policy* sehingga membuka peluang serangan XSS, serta mengizinkan akses dari semua origin akibat konfigurasi CORS yang tidak tepat. Seluruh 10 temuan dikonfirmasi valid, di mana 4 kerentanan dibuktikan melalui exploitation sebagai proof of concept dan 6 temuan lainnya dikonfirmasi berdasarkan hasil pemindaian OWASP ZAP. Meskipun sebagian besar temuan tergolong risiko *Medium*, kombinasi kerentanan tersebut tetap berpotensi membahayakan keamanan

ISSN (Online) 2747-0563
Volume 6 Tahun 2026

sistem dan data pengguna apabila dimanfaatkan secara bersamaan. Oleh karena itu, diperlukan perbaikan segera terhadap seluruh temuan yang teridentifikasi serta evaluasi keamanan secara berkala agar celah keamanan dapat ditemukan dan ditangani sebelum dimanfaatkan oleh pihak yang tidak bertanggung jawab.

REFERENSI

[1] V. L. D. Pasaribu, ECOMMERCE: Menciptakan Daya Saing Melalui Informasi Teknologi, 1st ed. Cilacap, Indonesia: PT Media Pustaka Indo, 2025.

[2] A. R. Supriyatna, I. Asrowardi, S. D. Putra, and E. Subyantoro, "Analisis Kerentanan Aplikasi Web E-commerce Berdasarkan Standar OWASP Top 10: Studi Kasus pada Situs Kopi Lampung Nusantara," EXPERT: Jurnal Manajemen Sistem Informasi dan Teknologi, vol. 14, no. 2, hal. 95–101, 2024.

[3] R. al Ridwan Bintang Firdaus and T. Ismardiko Widyawan, "Pengujian Kerentanan Website Menggunakan Metode Penetration Testing Dengan OWASP: Studi Kasus Pemerintah Kabupaten Semarang," vol. 8, no. 2, hal. 106–114, 2025.

[4] M. M. Nur Hariyanto and C. Umam, "Analisis Keamanan Sistem Kepegawaian dan Pengembangan Sumber Daya Manusia di Sektor Pemerintahan Dengan Metode OWASP," Jurnal Algoritma, vol. 22, no. 2, hal. 1661–1668, Nov. 2025, doi: 10.33364/algoritma/v.22-2.2631.

[5] M. Fadli Muttaqin, D. Ferdiansyah, S. Alas Majapahit, and R. Rijayanti, "Analisis Keamanan Fitur Login Aplikasi: Studi Kasus Sistem Manajemen Mutu Sekolah OWASP Top 10 dengan OWASP ZAP," Pasinformatik, vol. 4, no. 2, Jul. 2025.

[6] G. Melisa and I. Anastasia Sitanggang, "Perancangan Website E-Commerce INEED.ID," Jurnal Teknik Informatika, vol. 14, no. 1, hal. 19–23, Jan. 2022.

[7] Linknet, "Apa itu OWASP? Apa itu OWASP Top 10?," 2023. [Online]. Available: https://www.linknet.id/article/penetration-testing/, tanggal akses: 14 Maret 2026.

[8] OWASP, "OWASP Top 10:2021," 2021. [Online]. Available: https://owasp.org/Top10/2021/, tanggal akses: 14 Maret 2026..

[9] ZAP, "Getting Started with ZAP," 2021. [Online]. Available: https://www.zaproxy.org/getting-started/, tanggal akses: 14 Maret 2026.

[10] Y. Noviko Rahman, R. Maulana Hadi, M. Nabilah, M. Hanif Waskito, and N. Aini Rakhmawati, "Analisis Penggunaan Framework Website JDIH Khusus Peraturan Kementerian Republik Indonesia," Jurnal Teknologi dan Open Source, vol. 3, no. 1, hal. 78–89, Jun. 2020.

[11] E. Novitasari, T. Ariyadi, and D. Pertiwiningsih, "Penerapan Burpsuite untuk Mengidentifikasi Serangan pada Jaringan Wireless," Jurnal Tera, vol. 5, no. 2, hal. 49–58, Sep. 2025.

[12] I. Sari, "Mengenal Hacking Sebagai Salah Satu Kejahatan di Dunia Maya," JSI (Jurnal Sistem Informasi) Universitas Suryadarma, vol. 10, no. 2, hal. 169–186, 2023, doi: 10.35968/jsi.v10i2.1086.

[13] A. Subagyo, "Sinergi Dalam Menghadapi Ancaman Cyber Warfare," Jurnal Pertahanan & Bela Negara, vol. 5, no. 1, hal. 89–108, 2018, doi: 10.33172/jpbh.v5i1.350.

[14] Nmap, "Panduan Referensi Nmap (Man Page, Bahasa Indonesia)," 2026. [Online]. Available: https://nmap.org/man/id/index.html#man-description, tanggal akses: 15 Maret 2026.

[15] R. Erwin Gunadhi Rahayu, D. Ramdhan, and D. Kurniadi, "Penetration Testing Untuk Menguji Tingkat Keamanan Sistem Pada Academic Information System (AISnet) Institut Teknologi Garut Dengan Metode Zero Entry Hacking (ZEH)," Jurnal Algoritma, vol. 22, no. 2, Nov. 2025, doi: 10.33364/algoritma/v.22-2.2149.