EXPERT JURNAL MANAJEMEN SISTEM INFORMASI DAN TEKNOLOGI

Analisis Kerentanan Aplikasi Web E-commerce Berdasarkan Standar OWASP Top 10: Studi Kasus pada Situs Kopi Lampung Nusantara

Agiska Ria Supriyatna [1*], **Imam Asrowardi** [2], **Septafiansyah Dwi Putra** [3], **Eko Subyantoro** [4]
[1234] Jurusan Teknologi Informasi Program Studi Teknologi Rekayasa Internet,
Politeknik Negeri Lampung, Bandar Lampung, Indonesia
[1*] agiskaria@polinela.ac.id, [2] imam@polinela.ac.id, [3] septa@polinela.ac.id, [4] eko@polinela.ac.id

ABSTRACT – This study aims to analyze security vulnerabilities within the Kopi Lampung Nusantara e-commerce web application by employing the Open Web Application Security Project (OWASP) standards as the primary reference framework. The OWASP Top 10 standards were applied to identify the most common types of vulnerabilities that pose significant risks to web applications. The research method involved penetration testing to uncover potential security gaps that could compromise user data security and system integrity. Testing results revealed several critical vulnerabilities, including Personally Identifiable Information (PII) Disclosure, which poses a risk to user privacy; the absence of Anti-CSRF Tokens, heightening the risk of Cross-Site Request Forgery (CSRF) attacks; and insufficient security headers such as Content Security Policy (CSP) and X-Content-Type-Options. These findings underscore the importance of implementing OWASP security standards in the development and maintenance of web applications, particularly in the e-commerce sector, which is highly susceptible to cyber-attacks.

Keywords: security vulnerabilities; OWASP; e-commerce; penetration testing; web application.

ABSTRAK –Penelitian ini bertujuan untuk menganalisis kerentanan keamanan pada aplikasi web e-commerce Kopi Lampung Nusantara menggunakan standar *Open Web Application Security Project* (OWASP) sebagai acuan utama. Standar OWASP Top 10 diterapkan untuk mengidentifikasi jenis-jenis kerentanan paling umum yang memiliki risiko tinggi terhadap aplikasi web. Metode penelitian melibatkan pengujian penetrasi yang bertujuan untuk mengungkap potensi celah keamanan yang dapat mengancam keamanan data pengguna serta integritas sistem. Hasil pengujian menunjukkan adanya beberapa kerentanan kritis, termasuk PII *Disclosure* yang berisiko terhadap privasi pengguna, absennya Anti-CSRF Tokens yang meningkatkan risiko serangan *Cross-Site Request Forgery* (CSRF), serta kurangnya header keamanan seperti *Content Security Policy* (CSP) dan *X-Content-Type-Options.* Temuan ini menggarisbawahi pentingnya penerapan standar keamanan OWASP dalam pengembangan dan pemeliharaan aplikasi web, terutama pada sektor e-commerce yang rentan terhadap serangan siber.

Kata Kunci: Kerentanan keamanan; OWASP; e-commerce; pengujian penetrasi; aplikasi web.

1. PENDAHULUAN

Dalam era digital, keamanan aplikasi web menjadi aspek yang semakin penting, terutama pada sektor e-commerce yang rentan terhadap berbagai serangan siber. Aplikasi web yang tidak aman dapat menyebabkan kebocoran data pengguna, pencurian informasi pribadi, hingga kerugian finansial yang signifikan bagi organisasi maupun konsumen [1], [2]. Sebagai salah satu upaya untuk mengatasi masalah ini, *Open Web Application Security Project* (OWASP) menyusun standar keamanan OWASP Top 10, yang berfungsi sebagai pedoman bagi pengembang dalam membangun aplikasi yang aman dari kerentanan umum seperti *SQL Injection, Cross-Site Scripting (XSS),* dan *Sensitive Data Exposure* [3], [4]. Studi sebelumnya menunjukkan bahwa implementasi standar OWASP tidak hanya membantu melindungi data pengguna, tetapi juga meningkatkan kepercayaan pelanggan dan reputasi perusahaan [5].

Standar OWASP Top 10 telah diakui secara luas sebagai referensi utama dalam keamanan aplikasi web, memberikan peta jalan untuk mengurangi risiko dengan mendeteksi dan menangani kerentanan secara efektif [6]. Beberapa penelitian telah membuktikan bahwa kepatuhan terhadap pedoman OWASP meningkatkan efektivitas pengujian keamanan dan meminimalkan potensi eksploitasi kerentanan [7], [8]. Dalam konteks *e-commerce*, keamanan yang optimal menjadi semakin krusial karena tingginya lalu lintas data dan transaksi yang melibatkan informasi pribadi konsumen [9]. Selain itu, penerapan OWASP dalam pengembangan aplikasi web memberikan kerangka kerja yang mudah diikuti oleh pengembang, sehingga membantu mengintegrasikan keamanan dari tahap awal pengembangan [10].

Penelitian ini berfokus pada analisis kerentanan keamanan aplikasi web *e-commerce* Kopi Lampung Nusantara dengan mengacu pada standar OWASP. Melalui pengujian penetrasi, berbagai celah keamanan dapat diidentifikasi, termasuk masalah serius seperti PII *Disclosure* dan kurangnya mekanisme *Anti-CSRF Tokens*, yang dapat mengarah pada serangan serius seperti *Cross-Site Request Forgery* [11], [12].

2. DASAR TEORI

Keamanan aplikasi web menjadi perhatian utama dalam industri digital, khususnya pada sektor e-*commerce* yang semakin rentan terhadap serangan siber. Peningkatan aktivitas siber berisiko tinggi, seperti *SQL Injection, Cross-Site Scripting (XSS)*, dan *Sensitive Data Exposure*, menuntut adanya standar keamanan yang komprehensif. Untuk mengatasi tantangan ini, *Open Web Application Security Project* (OWASP) memperkenalkan standar OWASP Top 10, yang menyusun daftar sepuluh ancaman keamanan web paling umum yang dapat menimbulkan risiko serius bagi data dan integritas aplikasi [5]. Pedoman OWASP Top 10 diterima secara luas dan memberikan panduan bagi pengembang aplikasi web dalam mengidentifikasi, mengurangi, dan mencegah ancaman-ancaman tersebut melalui pendekatan berbasis praktik terbaik dan pemantauan keamanan yang berkelanjutan [6].

OWASP Top 10 terdiri dari sepuluh kategori kerentanan yang sering muncul dalam aplikasi web. Setiap kategori mencakup jenis-jenis serangan dan rekomendasi mitigasi untuk mengurangi risiko. Berikut adalah daftar OWASP Top 10 beserta deskripsinya:

1. *Broken Access Control*
   Kerentanan yang memungkinkan pengguna mengakses data atau fungsi yang seharusnya dibatasi, misalnya dengan mengakses data pengguna lain atau mengubah hak akses [8].
2. *Cryptographic Failures*
   Kegagalan dalam enkripsi data yang menyebabkan data sensitif rentan diakses atau dicuri. Kerentanan ini termasuk dalam risiko besar, terutama pada aplikasi yang menangani informasi pribadi dan transaksi [9].
3. I*njection*
   Meliputi *SQL Injection* dan *Command Injection*, yang memungkinkan penyerang mengeksekusi kode berbahaya pada sistem dengan cara memanfaatkan input pengguna yang tidak divalidasi dengan baik [10].
4. *Insecure Design*
   Kelemahan yang muncul karena desain aplikasi yang kurang memperhatikan aspek keamanan sejak awal, mengakibatkan aplikasi mudah dieksploitasi oleh pihak ketiga [11].
5. *Security Misconfiguration*
   Konfigurasi keamanan yang tidak tepat, seperti tidak adanya security headers, dapat meningkatkan risiko serangan seperti *clickjacking* atau MIME *sniffing* [12].
6. *Vulnerable and Outdated Components*
   Penggunaan komponen perangkat lunak yang usang atau rentan, seperti pustaka *JavaScript* yang sudah tidak aman, yang meningkatkan risiko terhadap serangan yang diketahui [13].
7. *Identification and Authentication Failures*
   Kelemahan dalam otentikasi pengguna yang memungkinkan penyerang melakukan akses tidak sah, misalnya dengan menggunakan kredensial yang tidak aman atau tidak melakukan otentikasi ganda [14].
8. *Software and Data Integrity Failures*
   Kerentanan yang terkait dengan integritas perangkat lunak dan data, terutama pada pembaruan perangkat lunak yang tidak diverifikasi, yang bisa dimanfaatkan oleh penyerang untuk menyusupkan kode berbahaya [15].
9. *Security Logging and Monitoring Failures*
   Kurangnya mekanisme log dan pemantauan yang dapat menghambat upaya mendeteksi dan merespon serangan tepat waktu, meningkatkan risiko eksploitasi berulang [16].
10. *Server-Side Request Forgery (SSRF)*
    Kerentanan yang memungkinkan penyerang memanipulasi server untuk membuat permintaan tak sah ke sumber daya internal, yang dapat digunakan untuk mengakses data sensitif atau mengontrol layanan internal lainnya [17].

Implementasi OWASP Top 10 ini telah terbukti dapat mengurangi risiko keamanan secara signifikan. Alat uji keamanan OWASP ZAP, sebagai bagian dari pendekatan OWASP, memungkinkan deteksi otomatis terhadap berbagai jenis ancaman yang termasuk dalam daftar OWASP Top 10 ini. Dengan pemindaian aktif dan pasif, alat ini dapat mengidentifikasi kelemahan pada aplikasi web sejak awal proses pengembangan [18].

Beberapa penelitian menunjukkan bahwa penerapan OWASP Top 10 dalam siklus pengembangan perangkat lunak dapat memperkuat keamanan aplikasi, khususnya di sektor *e-commerce* yang berisiko tinggi, dan meningkatkan ketahanan terhadap serangan siber yang berpotensi membahayakan data pengguna dan integritas sistem [19]. Dengan demikian, OWASP Top 10 menjadi referensi penting dalam pengembangan aplikasi web yang aman dan berkelanjutan [20].

3. METODOLOGI

Keamanan aplikasi web menjadi prioritas utama dalam sektor e-commerce, terutama mengingat risiko yang timbul dari meningkatnya serangan siber yang berpotensi merugikan pengguna dan perusahaan. Untuk menghadapi tantangan ini, standar OWASP Top 10 digunakan sebagai acuan utama dalam penelitian ini,

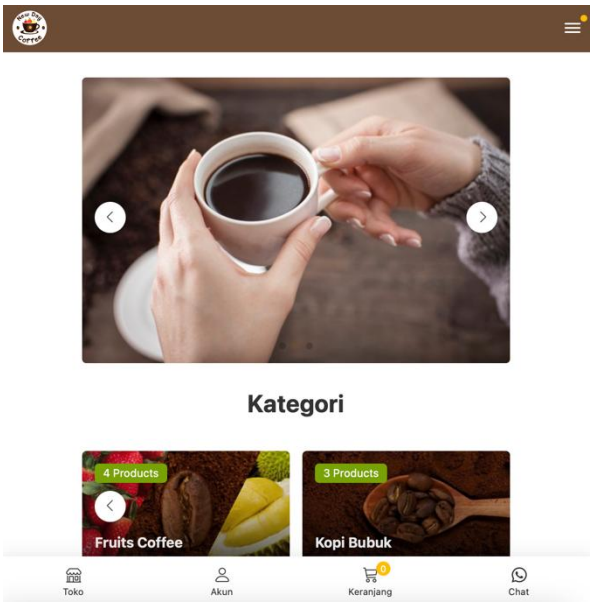
mengingat relevansinya sebagai panduan keamanan yang menyeluruh dan diakui secara internasional untuk mengidentifikasi serta mengelola kerentanan pada aplikasi web. Alat OWASP ZAP dipilih untuk pengujian penetrasi karena kemampuannya dalam mendeteksi kerentanan keamanan secara otomatis dengan fitur pemindaian pasif dan aktif yang efisien. Dengan mengikuti panduan OWASP, penelitian ini menyusun metode pengujian keamanan dalam beberapa tahapan sebagai berikut:

1. Tahap Persiapan dan Pengumpulan Data. Pada tahap awal ini, dilakukan persiapan dengan mengumpulkan informasi dasar tentang aplikasi web Kopi Lampung Nusantara. Pengumpulan data mencakup struktur aplikasi, fitur utama, serta alur data pengguna, mengingat situs ini melibatkan transaksi data sensitif dalam konteks *e-commerce*. Pemilihan OWASP ZAP sebagai alat utama pengujian juga dilakukan karena kemampuannya yang kompatibel dengan standar OWASP, yang memungkinkan pemantauan lalu lintas web dan deteksi celah keamanan secara menyeluruh [13].
2. Pengujian Penetrasi (*Penetration Testi*ng). Tahap pengujian penetrasi dilakukan dengan OWASP ZAP, di mana simulasi serangan yang menargetkan kelemahan yang telah teridentifikasi sebelumnya dilakukan. Tujuan dari tahap ini adalah untuk menguji efektivitas setiap kerentanan yang ditemukan serta mengevaluasi tingkat risiko yang ditimbulkan terhadap aplikasi. Proses ini melibatkan manipulasi parameter masukan, pengujian akses ke data sensitif tanpa otorisasi, serta pengujian terhadap ketahanan aplikasi dalam menghadapi serangan injeksi [14].
3. Identifikasi Kerentanan Berdasarkan Standar OWASP Top 10.
   Standar OWASP Top 10 digunakan untuk mendeteksi kerentanan keamanan yang paling umum dan kritis pada aplikasi web. Setiap komponen dari aplikasi diuji secara sistematis untuk mengidentifikasi kerentanan seperti *SQL Injection, Cross-Site Scripting (XSS), Sensitive Data Exposure, dan Security Misconfiguration*. Panduan OWASP digunakan untuk memastikan bahwa semua aspek keamanan pada lapisan *front-end, middleware,* dan *back-end* dapat diidentifikasi dan dianalisis secara tepat [15], [16].
4. Analisis Risiko dan Dampak Kerentanan. Setelah mengidentifikasi kerentanan, tahap ini melakukan analisis risiko dengan menilai dampak dari setiap kerentanan terhadap keamanan data dan integritas sistem. Evaluasi risiko mengacu pada tingkat keparahan serta kemungkinan eksploitasi kerentanan yang ditemukan. Analisis ini memberikan prioritas untuk tindakan korektif yang perlu diambil, terutama untuk kerentanan dengan tingkat risiko tinggi yang mengancam keamanan data pengguna [17], [18].
5. Penyusunan Rekomendasi Perbaikan Berdasarkan hasil analisis kerentanan dan dampaknya.
   Pada tahapan ini disusun rekomendasi perbaikan dengan mengacu pada panduan OWASP. Rekomendasi yang diberikan meliputi penerapan *security hea*ders, penggunaan *Anti-CSRF Tokens*, serta pengaturan konfigurasi keamanan yang lebih ketat untuk mencegah serangan serupa di masa mendatang. Langkah-langkah ini bertujuan untuk memperkuat sistem keamanan aplikasi secara menyeluruh [19].
6. Evaluasi dan Validasi.
   Tahap akhir melibatkan evaluasi terhadap aplikasi yang telah diperbaiki melalui pengujian ulang untuk memastikan bahwa semua kerentanan telah ditangani secara efektif. Evaluasi ini dilakukan untuk memvalidasi bahwa rekomendasi yang diterapkan dapat meningkatkan keamanan aplikasi web dan memenuhi standar keamanan yang diharapkan.

4. HASIL DAN PEMBAHASAN

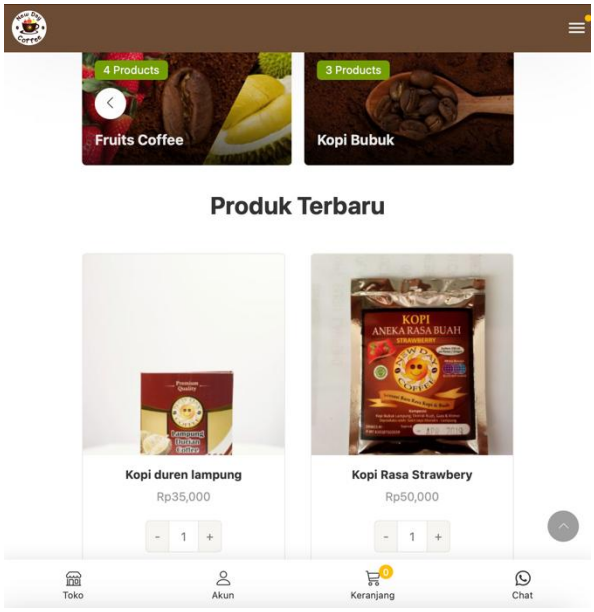
A. Profil Website Kopi Lampung Nusantara

Website **Kopi Lampung Nusantara** adalah sebuah platform e-commerce yang dirancang untuk memudahkan pelanggan dalam membeli berbagai produk kopi khas Lampung. Dengan tampilan sederhana dan ramah pengguna, situs ini menghadirkan pengalaman belanja online yang nyaman dan informatif. Tampilan utama website, seperti yang terlihat pada Gambar 1, menampilkan desain yang hangat dengan dominasi warna cokelat, memberikan kesan alami dan premium yang sesuai dengan karakteristik kopi khas Lampung.



Gambar 1. Tampilan utama website Kopi Lampung Nusantara.

Produk-produk yang ditawarkan dikategorikan dengan jelas, seperti **Fruits Coffee**, yang menghadirkan kopi dengan rasa buah unik, serta **Kopi Bubuk**, yang menampilkan cita rasa kopi tradisional Lampung. Bagian **Produk Terbaru**, sebagaimana ditampilkan dalam Gambar 2, menampilkan item unggulan seperti **Kopi Duren Lampung**, yang merupakan perpaduan kopi robusta khas Lampung dan rasa durian, serta **Kopi Rasa Strawberry**, yang memberikan sensasi rasa buah stroberi dalam setiap tegukan.



Gambar 2. Tampilan bagian produk terbaru pada website Kopi Lampung Nusantara.

Website ini juga dilengkapi dengan fitur navigasi yang memudahkan pelanggan untuk berpindah antar halaman, termasuk fitur **Keranjang** untuk mengatur pesanan dan **Chat** untuk berkomunikasi langsung dengan penjual. Dengan desain visual yang ditampilkan secara menarik pada gambar, website ini mencerminkan identitas lokal yang kuat sekaligus memberikan kenyamanan bagi pelanggan. Tujuan utama dari website ini adalah untuk memperkenalkan produk kopi lokal Lampung kepada pasar yang lebih luas, sekaligus memberikan kemudahan bagi pelanggan untuk menikmati kopi berkualitas langsung dari produsen. Dengan fitur dan desain yang intuitif, website ini diharapkan dapat meningkatkan daya tarik kopi Lampung, baik di tingkat nasional maupun internasional.

B. Pengujian Penetrasi (*Penetration Testi*ng).

Berdasarkan hasil pemindaian menggunakan aplikasi ZAP untuk aplikasi web Kopi Lampung Nusantara (ditemukan beberapa kerentanan keamanan yang diklasifikasin menjadi beberapa tingkat risiko dan keyakinan (*confidence)*. Kerentanan ini mencakup mulai mulai dari tingkat risiko tinggi hingga informasional dimana setiap tingkat risiko akan memberikan gambaran tingkat keparahan dan potensi dampak terhadap aplikasi. Kerentanan-kerentanan yang ditemukan tersebut telah dianalisis lebih lanjut, dengan hasil yang diklasifikasikan sesuai tingkat risiko dan tingkat keyakinan serta dipetakan berdasarkan standar OWASP sebagaimana diuraikan dalam poin berikut :

1. Ringkasan Tingkat Risiko dan Keyakinan.
   Berdasarkan laporan ZAP, temuan kerentanan diklasifikasikan ke dalam empat kategori risiko: tinggi, sedang, rendah, dan informasional. Penjelasan lebih lanjut tentang distribusi risiko tersebut dapat dilihat pada **Tabel 1.**

Tabel 1. Ringkasan Klasifikasi Kerentanan

No	Risiko	Tinggi	Sedang	Rendah	Total
1	Tinggi	4.8%	0.0%	0.0%	4.8%
2	Sedang	4.8%	9.5%	4.8%	19.0%
3	Rendah	4.8%	19.0%	4.8%	28.6%
4	Informasional	4.8%	14.3%	28.6%	47.6%

   Berdasara pada Tabel 1, dar total 21 kerentanan yang ditemukan, kategori informasional mendominasi dengan persentase tertinggi sebesar 47,6%, diikuti oleh kategori rendah dan sedang yang masing-masing menyumbang 28,6% dan 19,0%. Sementara itu, hanya satu kerentanan (4,8%) yang masuk dalam kategori risiko tinggi, yaitu *PII Disclosure*, yang memiliki potensi dampak serius terhadap privasi pengguna karena terkait dengan kebocoran data pribadi.
2. Distribusi Kerentanan Berdasarkan Situs
   Kerentanan yang ditemukan tersebar di dua domain aplikasi, yaitu:
   a. https://kopilampungnusantara.com, dan
   b. http://kopilampungnusantara.com.
   Sebagian besar kerentanan dengan risiko rendah dan informasional ditemukan pada domain tanpa protokol **HTTPS**, hal ini menunjukkan pentingnya penerapan protokol yang lebih aman di seluruh situs untuk meningkatkan integritas data.
3. Analisis Jenis Kerentanan Berdasarkan laporan ZAP.
   Berdasarkan analisis lebih mendalam, terdapat delapan jenis kerentanan utama yang ditemukan. Penjelasan detail disajikan pada **Tabel 2** berikut.

Tabel 2. Daftar Kerentanan Berdasarkan Tingkat Resiko

Jenis Kerentanan	Risiko	Temuan
PII Disclosure	Tinggi	1

DOI http://dx.doi.org/10.36448/expert.v14i2.4034
e-ISSN 2745-7265 p-ISSN 2088-5555 EXPERT Vol. 14 No. 2 December 31, 2024 – Hal.98

Jenis Kerentanan	Risiko	Temuan
Absence of Anti-CSRF Tokens	Sedang	426
Content Security Policy (CSP) Header Not Set	Sedang	1244
Missing Anti-clickjacking Header	Sedang	1192
Vulnerable JS Library	Sedang	1
Cookie No HttpOnly Flag	Rendah	1808
Cookie Without Secure Flag	Rendah	192
Cookie without SameSite Attribute	Rendah	1816
Strict-Transport-Security Header Not Set	Rendah	4

Berdasarkan Tabel 2, terdapat delapan jenis kerentanan utama ditemukan pada aplikasi *Kopi Lampung Nusantara*, dengan klasifikasi tingkat risiko tinggi, sedang, dan rendah. Kerentanan yang paling kritis adalah *PII Disclosure* (Personal Identifiable Information Disclosure), yang masuk dalam kategori risiko tinggi. Kerentanan ini ditemukan pada satu endpoint dan berpotensi membuka data pribadi pengguna, yang dapat dimanfaatkan dalam serangan siber. Untuk memitigasi risiko ini, diperlukan penerapan enkripsi data dan kebijakan penyimpanan data yang lebih ketat guna melindungi privasi pengguna.

Pada tingkat risiko sedang, ditemukan lima jenis kerentanan dengan jumlah kasus yang signifikan. Salah satu yang menonjol adalah *Absence of Anti-CSRF Tokens*, yang teridentifikasi sebanyak 426 kasus. Ketiadaan token CSRF pada aplikasi meningkatkan risiko serangan **Cross-Site Request Forgery (CSRF)**, di mana penyerang dapat mengirim permintaan tidak sah atas nama pengguna. Selain itu, kerentanan *Content Security Policy (CSP) Header Not Set*merupakan temuan terbanyak dalam kategori ini, dengan 1.244 kasus. Tidak adanya header CSP menunjukkan bahwa aplikasi belum menerapkan kebijakan yang melindungi dari serangan berbasis **Cross-Site Scripting (XSS)** atau injeksi skrip jahat.

Kerentanan *Missing Anti-clickjacking Header* ditemukan sebanyak 1.192 kasus, yang membuat aplikasi rentan terhadap serangan **clickjacking**, di mana penyerang dapat membajak antarmuka aplikasi dengan memanipulasi tampilan dalam sebuah frame. Solusi yang direkomendasikan adalah penerapan header keamanan seperti X-Frame-Options untuk mencegah ancaman tersebut.

Pada tingkat risiko rendah, kerentanan utama adalah *Cookie No HttpOnly Flag*, dengan jumlah kasus tertinggi sebanyak 1.808. Kerentanan ini menunjukkan bahwa aplikasi belum sepenuhnya melindungi cookie dari akses JavaScript, sehingga rentan terhadap serangan seperti pencurian sesi pengguna. Selain itu, *Cookie Without Secure Flag* ditemukan pada 192 kasus, yang menunjukkan bahwa cookie tidak dilindungi oleh protokol aman HTTPS. Temuan ini juga melibatkan *X-Content-Type-Options Header Missing* (4 kasus), yang dapat memungkinkan serangan berbasis MIME-type sniffing karena absennya header keamanan yang sesuai.

Secara keseluruhan, analisis menunjukkan bahwa meskipun kerentanan risiko tinggi hanya satu, yaitu *PII Disclosure*, dampaknya sangat serius dan memerlukan prioritas penanganan. Kerentanan risiko sedang seperti *Absence of Anti-CSRF Tokens* dan *Content Security Policy (CSP) Header Not Set* mengindikasikan adanya kelemahan dalam kebijakan keamanan aplikasi yang harus segera diperbaiki untuk mengurangi potensi serangan seperti **CSRF** dan **XSS**. Selain itu, tingginya jumlah kasus pada kategori risiko rendah seperti kelemahan pada pengaturan cookie menekankan pentingnya penguatan implementasi HTTPS dan konfigurasi keamanan header.

C. Analisis Kerentanan Berdasarkan Standar OWASP Top 10.

Hasil pemindaian menggunakan alat ZAP pada aplikasi web *Kopi Lampung Nusantara* mengidentifikasi delapan jenis kerentanan keamanan yang signifikan. Setiap kerentanan dikelompokkan berdasarkan tingkat risikonya dan dikaitkan dengan panduan dari **OWASP Top 10**, yang mencakup *Sensitive Data Exposure*, *Cross-Site Scripting (XSS)*, *Cross-Site Request Forgery (CSRF)*, *Security Misconfiguration*, serta *Using Components with Known Vulnerabilities*. Analisis ini bertujuan untuk memahami dampak kerentanan terhadap keamanan aplikasi dan memberikan rekomendasi mitigasi yang relevan. **Tabel 3** menyajikan temuan kerentanan berdasarkan jenis, tingkat risiko, deskripsi singkat, dan relevansi dengan kategori OWASP Top 10..

Tabel 3. Jenis Kerentanan dan Tingkat Risiko Berdasarkan Standar OWASP

No	Jenis Kerentanan	Risiko	Keterangan Singkat	Standar OWASP yang Relevan
1	*PII Disclosure*	Tinggi	Informasi pribadi terpapar pada permintaan GET	OWASP Top 10: *Sensitive Data Exposure*
2	*Absence of Anti-CSRF Tokens*	Sedang	Tidak adanya token CSRF meningkatkan risiko serangan CSRF	OWASP Top 10: *Cross-SiteRequest Forgery (CSRF)*

No	Jenis Kerentanan	Risiko	Keterangan Singkat	Standar OWASP yang Relevan
3	*Content Security Policy (CSP) Header Not Set*	Sedang	*CSP header* tidak disetel, membuat situs rentan terhadap berbagai serangan	OWASP Top 10: *Security Misconfiguration*
4	*Missing Anti-clickjacking Header*	Sedang	*Header anti-clickjacking* tidak ada, sehingga situs dapat dibajak melalui teknik *clickjacking*	OWASP Top 10: *Security Misconfiguration*
5	*Vulnerable JS Library*	Sedang	Penggunaan pustaka *JavaScript* yang rentan terhadap eksploitasi	OWASP Top 10: U*sing Components with KnownVulnerabilities*
6	*Cookie No HttpOnly Flag*	Rendah	*Cookie* tidak dilindungi dari akses *JavaScript*, rentan terhadap pencurian *cookie*	OWASP Top 10: *Security Misconfiguration*
7	*Cookie Without Secure Flag*	Rendah	Cookie tidak memiliki *Secure Flag*, rentan jika situs diakses melalui HTTP	OWASP Top 10: *Security Misconfiguration*
8	*X-Content-Type-Options Header Missing*	Rendah	*Header X-Content-Type-Options* tidak disetel, membuat situs rentan terhadap *MIME-type sniffing*	OWASP Top 10: *Security Misconfiguration*

Tabel 3 menggambarkan delapan jenis kerentanan keamanan yang ditemukan pada aplikasi *Kopi Lampung Nusantara*, yang dikategorikan berdasarkan tingkat risiko dan relevansinya dengan standar **OWASP Top 10**. Pada tingkat risiko **tinggi**, ditemukan kerentanan *PII Disclosure*, yang mengacu pada paparan informasi pribadi pengguna melalui permintaan GET tanpa perlindungan yang memadai. Berdasarkan OWASP, kerentanan ini termasuk dalam kategori *Sensitive Data Exposure* dan dianggap sangat kritis karena data sensitif seperti nama, alamat, atau kredensial pengguna dapat diakses oleh pihak tidak berwenang, berpotensi menyebabkan pencurian identitas atau eksploitasi privasi.

Selanjutnya, pada tingkat risiko **sedang**, terdapat lima kerentanan yang memengaruhi keamanan aplikasi. *Absence of Anti-CSRF Tokens* adalah salah satu kelemahan signifikan, di mana ketiadaan token *Cross-Site Request Forgery (CSRF)*memungkinkan penyerang memalsukan permintaan yang sah atas nama pengguna, meningkatkan risiko manipulasi data atau transaksi tidak sah. Selain itu, ketiadaan *Content Security Policy (CSP) Header* mengakibatkan aplikasi rentan terhadap serangan injeksi skrip seperti *Cross-Site Scripting (XSS)*, karena tidak adanya pembatasan pada sumber daya yang dapat dimuat oleh browser. Kerentanan lain yang ditemukan adalah *Missing Anti-clickjacking Header*, yang membuat aplikasi rentan terhadap serangan *clickjacking*, di mana penyerang dapat memanipulasi interaksi pengguna melalui elemen frame tersembunyi. Selain itu, penggunaan *Vulnerable JS Library* menunjukkan adanya pustaka JavaScript yang usang atau rentan terhadap eksploitasi, yang dapat menjadi pintu masuk bagi serangan berbahaya.

Pada tingkat risiko **rendah**, ditemukan tiga kerentanan yang berfokus pada pengaturan cookie dan header keamanan. *Cookie No HttpOnly Flag* memungkinkan cookie diakses oleh *JavaScript*, membuatnya rentan terhadap pencurian sesi pengguna. Selain itu, *Cookie Without Secure Flag* menunjukkan bahwa cookie dapat dikirim melalui koneksi HTTP yang tidak aman, meningkatkan risiko serangan *Man-in-the-Middle (MitM)*. Ketiadaan header *X-Content-Type-Options* juga ditemukan, yang memungkinkan browser melakukan *MIME-type sniffing*, membuka celah bagi file berbahaya untuk dimuat secara tidak sah. Meskipun dampak kerentanan risiko rendah relatif kecil, perbaikannya tetap diperlukan untuk memperkuat keamanan aplikasi secara menyeluruh.

D. Penyusunan Rekomendasi Perbaikan Berdasarkan hasil analisis kerentanan dan dampaknya.

Berdasarkan hasil analisis kerentanan dan dampaknya, langkah perbaikan perlu difokuskan pada masing-masing jenis kerentanan dengan mempertimbangkan tingkat risiko dan relevansi terhadap standar OWASP Top 10. Rekomendasi ini bertujuan untuk memitigasi dampak kerentanan, memperkuat keamanan aplikasi, dan melindungi data pengguna dari potensi eksploitasi. Berikut ini adalah penjelasan rinci mengenai setiap kerentanan yang ditemukan, dampaknya terhadap keamanan aplikasi, dan saran mitigasi yang sesuai.

Berikut ini adalah penjelasan rinci mengenai masing-masing kerentanan, dampaknya, dan saran mitigasi sesuai dengan standar OWASP Top 10.

1. PII *Disclosure*.
   PII (*Personally Identifiable Information) Disclosure* ditemukan sebagai risiko tinggi. Menurut *OWASP, Sensitive Data Exposure* adalah salah satu

kerentanan kritis yang perlu segera diperbaiki karena informasi pribadi dapat diakses melalui metode HTTP yang tidak aman. Kerentanan ini harus ditangani dengan menerapkan enkripsi data saat transit, seperti penggunaan HTTPS untuk semua halaman yang berisikan data pengguna.

2. *Absence of Anti-CSRF Tokens*
   Tidak adanya token *CSRF (Cross-Site Request Forgery)* adalah kelemahan signifikan yang membuat situs rentan terhadap serangan CSRF. OWASP merekomendasikan penggunaan token yang unik dan acak untuk setiap sesi pengguna, guna mencegah penyerang mengakses data pengguna melalui permintaan yang tampak sah namun berbahaya.
3. *Content Security Policy (CSP) Header Not Set*.
   Laporan juga mengidentifikasi ketiadaan *header CSP* yang penting untuk membatasi sumber daya yang dapat dimuat oleh *browser*. CSP berfungsi sebagai mekanisme pencegahan terhadap serangan injeksi skrip, seperti *Cross-Site Scripting* (XSS). Dengan mengaktifkan CSP, situs web dapat mengontrol sumber daya yang diizinkan, sehingga mengurangi risiko injeksi kode eksternal.
4. *Missing Anti-clickjacking Header.*
   Ketiadaan *header X-Frame-Options* membuat situs rentan terhadap serangan *clickjacking*, di mana penyerang bisa menggunakan situs tersebut sebagai media untuk memanipulasi interaksi pengguna tanpa sepengetahuan mereka. Mengikuti panduan OWASP, header ini harus disetel ke "DENY" atau "SAMEORIGIN" untuk mencegah elemen-elemen situs ditampilkan dalam frame yang mencurigakan.
5. *Vulnerable JS Library*.
   Penggunaan pustaka *JavaScript* yang rentan menjadi masalah serius karena pustaka yang usang atau tidak aman bisa menjadi pintu masuk bagi penyerang. OWASP mengidentifikasi penggunaan pustaka yang sudah diketahui memiliki kerentanan sebagai bagian dari *Using Components with Known Vulnerabilities*. Situs disarankan untuk memeriksa dan memperbarui pustaka secara berkala.
6. *Cookie No HttpOnly Flag.*
   Cookie yang tidak dilengkapi dengan *flag HttpOnly* rentan dicuri melalui *JavaScript,* meningkatkan risiko *Session Hijacking.* Penambahan *HttpOnly flag* memastikan bahwa *cookie* hanya dapat diakses melalui protokol HTTP dan tidak dapat diakses oleh *JavaScript.*
7. *Cookie Without Secure Flag.*
   Cookie tanpa *Secure Flag* juga dapat menimbulkan risiko jika situs diakses melalui koneksi HTTP yang tidak terenkripsi. OWASP merekomendasikan penggunaan *Secure Flag* untuk memastikan bahwa cookie hanya dikirim melalui koneksi HTTPS, guna meningkatkan keamanan sesi pengguna.
8. *X-Content-Type-Options Header Missing.*
   Ketiadaan *header X-Content-Type* Options membuat situs rentan terhadap MIME-*type sniffing*, di mana browser menebak jenis file berdasarkan konten. *Header* ini sebaiknya diaktifkan untuk menghindari risiko pemuatan konten yang berbahaya.

5. KESIMPULAN

Penelitian ini berhasil mengidentifikasi berbagai kerentanan keamanan kritis pada aplikasi web *e-commerce* Kopi Lampung Nusantara dengan menggunakan *standar Open Web Application Security Project* (OWASP) Top 10 sebagai kerangka acuan utama. Hasil pengujian penetrasi menunjukkan adanya beberapa celah keamanan penting, termasuk risiko tinggi pada PII *Disclosure*, ketidakhadiran *Anti-CSRF Tokens* yang meningkatkan ancaman serangan *Cross-Site Request Forgery* (CSRF), serta kurangnya header keamanan seperti *Content Security Policy* (CSP) dan *X-Content-Type-Options.* Kerentanan ini, jika tidak ditangani, dapat menyebabkan eksposur data pribadi pengguna dan menurunkan integritas sistem, yang berpotensi mengakibatkan kerugian finansial dan reputasi bagi perusahaan.

Temuan ini menggarisbawahi pentingnya penerapan dan pemeliharaan standar OWASP secara berkala dalam pengembangan aplikasi web, terutama pada sektor *e-commerce* yang rentan terhadap berbagai ancaman siber. Dengan mengimplementasikan rekomendasi yang diberikan, perusahaan dapat memperkuat pertahanan keamanan aplikasi mereka, melindungi data pengguna, serta meningkatkan kepercayaan pelanggan. Secara keseluruhan, penelitian ini memberikan kontribusi penting dalam literatur keamanan aplikasi web di Indonesia, menunjukkan bahwa penerapan standar keamanan yang ketat sangat dibutuhkan di sektor *e-commerce* untuk mengantisipasi potensi risiko di masa depan.

DAFTAR PUSTAKA

[1] S. A. Kumar and Y. U. Rani, "Implementation and analysis of web application security measures using OWASP Guidelines," 2022 Int. Conf. Recent Trends Microelectronics, Autom. Comput. Commun. Syst., pp. 182-187, 2022. DOI: 10.1109/ICMACC54824.2022.10093657.

[2] M. Srivastava, A. Raghuvanshi, and D. Khandelwal, "Security and scalability of e-commerce website by OWASP threats," 2023 6th Int. Conf. Inf. Syst. Comput. Networks (ISCON), pp. 1-8, 2023. DOI: 10.1109/ISCON57294.2023.10111955.

[3] T. Petranović and N. Zaric, "Effectiveness of using OWASP TOP 10 as AppSec standard," 2023 27th Int. Conf. Inf. Technol. (IT), pp. 1-4, 2023. DOI: 10.1109/IT57431.2023.10078626.

[4] Y. Wijaya, "Web-based dashboard for monitoring penetration testing activities based on OWASP standards," J. Teknol. Inf. dan Komunikasi, vol. 6,

no. 1, pp. 36-41, 2020. DOI: 10.26555/jiteki.v16i1.17019.

[5] O. B. Fredj, O. Cheikhrouhou, M. Krichen, H. Hamam, and A. Derhab, "An OWASP Top Ten driven survey on web application protection methods," TechRxiv, pp. 235-252, 2020. DOI: 10.36227/techrxiv.13265180.

[6] K. D. D. Ayunda, A. Widjajarto, and A. Budiono, "Implementation and analysis ModSecurity on web-based application with OWASP standards," J. Teknol. Inf. dan Komunikasi, vol. 8, no. 3, pp. 1638-1650, 2021. DOI: 10.35957/JATISI.V8I3.1223.

[7] J. Li, "Vulnerabilities mapping based on OWASP-SANS: A survey for static application security testing (SAST)," ArXiv, vol. abs/2004.03216, 2020. DOI: 10.33166/AETiC.2020.03.001.

[8] T. D. Sobola, P. Zavarsky, and S. Butakov, "Experimental study of ModSecurity web application firewalls," 2020 IEEE Int. Conf. Big Data Secur. Cloud (BigDataSecurity), IEEE Int. Conf. High Perform. Smart Comput. (HPSC), IEEE Int. Conf. Intell. Data Secur. (IDS), pp. 209-213, 2020. DOI: 10.1109/BigDataSecurity-HPSC-IDS49724.2020.00045.

[9] V. Tan, C. Cheh, and B. Chen, "From Application Security Verification Standard (ASVS) to regulation compliance: A case study in financial services sector," 2021 IEEE Int. Symp. Softw. Reliab. Eng. Work. (ISSREW), pp. 69-76, 2021. DOI: 10.1109/ISSREW53611.2021.00046.

[10] L. G. Petkova, "HTTP security headers," Knowledge - Int. J., 2019. DOI: 10.35120/kij3003701p.

[11] J. R. B. Higuera, J. B. Higuera, J. A. M. Sicilia, J. C. Villalba, and J. P. Nombela, "Benchmarking approach to compare web applications static analysis tools detecting OWASP top ten security vulnerabilities," Computers, Mater. & Continua, 2020. DOI: 10.32604/cmc.2020.010885.

[12] A. Sołtysik-Piorunkiewicz and M. Krysiak, "The cyber threats analysis for web applications security in Industry 4.0," Springer, pp. 127-141, 2020. DOI: 10.1007/978-3-030-40417-8_8.

[13] M. Srivastava et al., "Security and scalability of e-commerce website by OWASP threats," 2023 6th Int. Conf. Inf. Syst. Comput. Networks (ISCON), pp. 1-8, 2023.

[14] S. A. Kumar and Y. U. Rani, "Implementation and analysis of web application security measures using OWASP Guidelines," 2022 Int. Conf. Recent Trends Microelectronics, Autom. Comput. Commun. Syst., pp. 182-187, 2022.

[15] T. Petranović and N. Zaric, "Effectiveness of using OWASP TOP 10 as AppSec standard," 2023 27th Int. Conf. Inf. Technol. (IT), pp. 1-4, 2023.

[16] O. B. Fredj et al., "An OWASP Top Ten driven survey on web application protection methods," TechRxiv, pp. 235-252, 2020.

[17] K. D. D. Ayunda et al., "Implementation and analysis ModSecurity on web-based application with OWASP standards," J. Teknol. Inf. dan Komunikasi, vol. 8, no. 3, pp. 1638-1650, 2021.

[18] J. Li, "Vulnerabilities mapping based on OWASP-SANS: A survey for static application security testing (SAST)," ArXiv, vol. abs/2004.03216, 2020.

[19] L. G. Petkova, "HTTP security headers," Knowledge - Int. J., 2019.