HIJM
HORIZON: Indonesian Journal of Multidisciplinary

p-ISSN: 2987-7776
e-ISSN: 2987-7180
Vol. 4, No. 4, 2026

ANALISIS KERENTANAN KEAMANAN SISTEM ENTERPRISE RESOURCE PLANNING MENGGUNAKAN PTES DAN OWASP ZAP

Ken Narendra Ekamartha[1], Henni Endah Wahanani[2], Achmad Junaidi[3]
[1, 2, 3]Universitas Pembangunan Nasional "Veteran" Jawa Timur, Jl. Rungkut Madya, Jawa Timur, Indonesia
Email: 2081010083@student.upnjatim.ac.id

Article History

Received: 20-07-2026

Revision: 05-08-2026

Accepted: 10-08-2026

Published: 13-08-2026

Abstract. This study aims to identify and analyze security vulnerabilities in the XYZ ERP system using the Penetration Testing Execution Standard (PTES) approach. The study was conducted through pre-engagement interactions, intelligence gathering, threat modeling, vulnerability analysis, exploitation, and reporting, utilizing OWASP ZAP as the primary testing tool. Vulnerabilities were classified based on the Common Weakness Enumeration (CWE), while their severity was assessed using the Common Vulnerability Scoring System (CVSS) version 3.1. The study identified four main vulnerabilities: SQL Injection (CVSS 8.3; High), Brute Force Login Page (CVSS 8.2; High), Cross-Site Scripting (CVSS 5.4; Medium), and the risk of active session abuse (CVSS 4.2; Medium). These vulnerabilities have the potential to threaten data confidentiality, integrity, and availability through unauthorized access, data manipulation, account takeover, and user session abuse. This research provides technical recommendations for improving ERP system security and serves as a reference for organizations in systematically evaluating and mitigating web application vulnerabilities.

Keywords: Security Vulnerability Analysis, Enterprise Resource Planning, Penetration Testing Execution Standard, OWASP ZAP.

Abstrak. Penelitian ini bertujuan mengidentifikasi dan menganalisis kerentanan keamanan pada sistem ERP XYZ menggunakan pendekatan *Penetration Testing Execution Standard* (PTES). Penelitian dilakukan melalui tahapan *pre-engagement interactions*, *intelligence gathering*, *threat modeling*, *vulnerability analysis*, *exploitation*, dan *reporting* dengan memanfaatkan OWASP ZAP sebagai alat utama pengujian. Kerentanan diklasifikasikan berdasarkan *Common Weakness Enumeration* (CWE), sedangkan tingkat keparahannya dinilai menggunakan *Common Vulnerability Scoring System* (CVSS) versi 3.1. Hasil penelitian mengidentifikasi empat kerentanan utama, yaitu *SQL Injection* (CVSS 8,3; High), *Brute Force Login Page* (CVSS 8,2; High), *Cross-Site Scripting* (CVSS 5,4; Medium), dan risiko penyalahgunaan *session* aktif (CVSS 4,2; Medium). Kerentanan tersebut berpotensi mengancam kerahasiaan, integritas, dan ketersediaan data melalui akses tidak sah, manipulasi data, pengambilalihan akun, serta penyalahgunaan sesi pengguna. Penelitian ini memberikan rekomendasi teknis untuk meningkatkan keamanan sistem ERP serta menjadi acuan bagi organisasi dalam melakukan evaluasi dan mitigasi kerentanan aplikasi web secara sistematis.

Kata Kunci: Analisis Kerentanan Keamanan, *Enterprise Resource Planning*, *Penetration Testing Execution Standard*, OWASP ZAP.

How to Cite: Ekamartha, K. N., Wahanani, H. E., & Junaidi, A. (2026). Analisis Kerentanan Keamanan Sistem *Enterprise Resource Planning* Menggunakan PTES dan Owasp Zap. *HORIZON: Indonesian Journal of Multidisciplinary*, 4 (4), 5425-5438. http://doi.org/10.54373/hijm.v4i4.7179

PENDAHULUAN

Enterprise Resource Planning (*ERP*) merupakan sistem informasi terintegrasi yang digunakan untuk mengelola berbagai proses bisnis organisasi, seperti keuangan, produksi, distribusi, sumber daya manusia, dan hubungan pelanggan (Tambunan, 2025). Implementasi *ERP* berbasis web memberikan kemudahan akses, meningkatkan efisiensi operasional, serta mendukung pengambilan keputusan secara *real-time*. Di sisi lain, keterhubungan sistem melalui jaringan juga memperluas *attack surface* sehingga meningkatkan risiko eksploitasi terhadap kerentanan keamanan aplikasi web. Apabila kerentanan tersebut tidak diidentifikasi dan ditangani dengan baik, kerahasiaan (*confidentiality*), integritas (*integrity*), dan ketersediaan (*availability*) data dapat terganggu (Kar Yee & Zolkipli, 2021). Oleh karena itu, evaluasi keamanan secara berkala menjadi bagian penting dalam pengelolaan sistem *ERP*.

Urgensi pengujian keamanan semakin meningkat seiring meningkatnya ancaman siber terhadap infrastruktur digital. Serangan terhadap Pusat Data Nasional Indonesia pada tahun 2024 menunjukkan bahwa kelemahan keamanan dapat mengganggu keberlangsungan layanan publik dan meningkatkan risiko kebocoran data (The Straits Times, 2024). Risiko serupa juga dapat terjadi pada sistem *ERP* karena sistem ini menyimpan berbagai informasi penting, seperti data pelanggan, transaksi, inventaris, laporan keuangan, dan data operasional perusahaan. Dengan karakteristik tersebut, sistem *ERP* memerlukan pengujian keamanan yang tidak hanya mampu mengidentifikasi kerentanan, tetapi juga memberikan dasar dalam menentukan prioritas mitigasi berdasarkan tingkat risiko.

Berbagai penelitian telah menerapkan *penetration testing* untuk mengevaluasi keamanan aplikasi web. Bagaskara et al. (2025) menggunakan pendekatan *OWASP Top 10* untuk mengidentifikasi kerentanan pada situs web instansi pemerintah dan menunjukkan bahwa pendekatan tersebut efektif dalam menemukan berbagai kelemahan keamanan aplikasi. Sementara itu, Astrida et al. (2022) menerapkan *Penetration Testing Execution Standard* (*PTES*) pada pengujian keamanan jaringan nirkabel dan menunjukkan bahwa metodologi tersebut menyediakan tahapan pengujian yang lebih sistematis, mulai dari *pre-engagement* hingga *reporting*. Kedua penelitian tersebut menunjukkan bahwa *penetration testing* efektif untuk menemukan kerentanan, tetapi memiliki fokus yang berbeda, yaitu pada aplikasi web dan jaringan.

Sebagian besar penelitian masih berfokus pada proses identifikasi kerentanan tanpa mengintegrasikan klasifikasi kelemahan perangkat lunak menggunakan *Common Weakness Enumeration* (*CWE*) maupun penilaian tingkat keparahan menggunakan *Common Vulnerability Scoring System* (*CVSS*). Padahal, *CWE* menyediakan klasifikasi standar terhadap

akar penyebab kelemahan perangkat lunak sehingga memudahkan proses analisis dan mitigasi (Common Weakness Enumeration Content Team, 2024). Di sisi lain, *CVSS* memberikan mekanisme penilaian tingkat keparahan kerentanan yang dapat digunakan untuk menentukan prioritas perbaikan secara objektif (FIRST, 2019). Integrasi kedua standar tersebut memungkinkan hasil *penetration testing* tidak hanya menghasilkan daftar kerentanan, tetapi juga memberikan prioritas mitigasi berdasarkan tingkat risiko (Security Journey, 2021).

Berdasarkan kondisi tersebut, penelitian ini menerapkan *Penetration Testing Execution Standard* (*PTES*) dengan memanfaatkan *OWASP ZAP* sebagai alat utama pengujian keamanan pada sistem *ERP* XYZ. Kerentanan yang ditemukan kemudian diklasifikasikan menggunakan *Common Weakness Enumeration* (*CWE*) dan dinilai tingkat keparahannya menggunakan *Common Vulnerability Scoring System* (*CVSS*) versi 3.1 sehingga menghasilkan prioritas mitigasi berdasarkan tingkat risiko. Pendekatan ini diharapkan memberikan analisis keamanan yang lebih komprehensif dibandingkan penelitian sebelumnya yang hanya berfokus pada identifikasi kerentanan. Penelitian ini bertujuan untuk menganalisis kerentanan keamanan pada sistem *ERP* XYZ menggunakan pendekatan *Penetration Testing Execution Standard* (*PTES*) serta menyusun rekomendasi mitigasi berdasarkan klasifikasi *Common Weakness Enumeration* (*CWE*) dan penilaian risiko *Common Vulnerability Scoring System* (*CVSS*) versi 3.1. Hasil penelitian diharapkan dapat menjadi referensi bagi organisasi dalam melaksanakan evaluasi keamanan aplikasi *ERP* berbasis web secara sistematis dan berbasis risiko.

METODE

Desain dan Ruang Lingkup Pengujian

Penelitian ini menggunakan metode eksperimen dengan pendekatan *Grey-Box Penetration Testing* berdasarkan *Penetration Testing Execution Standard* (PTES) (Penetration Testing Execution Standard, 2014). Pendekatan *grey-box* dipilih karena penguji memperoleh akses terbatas berupa akun pengguna (*user account*) tanpa memiliki akses terhadap kode sumber aplikasi, konfigurasi server, maupun struktur basis data. Pendekatan ini dipilih karena lebih merepresentasikan kondisi nyata dalam proses audit keamanan aplikasi web, yaitu ketika penguji berperan sebagai pengguna yang telah terautentikasi sehingga dapat mengevaluasi kerentanan yang hanya muncul setelah proses *login*. Berbeda dengan pendekatan *black-box* yang tidak memiliki informasi awal mengenai sistem, maupun *white-box* yang memberikan akses penuh terhadap kode sumber dan konfigurasi internal, pendekatan *grey-box* memungkinkan proses pengujian berlangsung lebih efisien sekaligus tetap menggambarkan

skenario serangan yang realistis terhadap aplikasi *ERP* (PTES, 2014; OWASP Foundation, 2024).

Objek penelitian adalah sistem *Enterprise Resource Planning* (*ERP*) XYZ berbasis web yang digunakan untuk mendukung proses bisnis perusahaan pada modul autentikasi pengguna, pengelolaan pengguna (*user management*), transaksi data, pelaporan, serta manajemen sesi (*session management*). Sistem dikembangkan menggunakan arsitektur *client-server* dan diakses melalui peramban (*web browser*) menggunakan protokol HTTP/HTTPS. Pengujian dilakukan menggunakan akun pengguna dengan hak akses terbatas (*low-privilege user*) sehingga seluruh aktivitas pengujian dilakukan pada fitur yang dapat diakses oleh pengguna tersebut. Pengujian dilaksanakan secara resmi berdasarkan persetujuan pemilik sistem sehingga seluruh aktivitas berada dalam batas etika dan ruang lingkup yang telah disepakati.

Ruang lingkup pengujian meliputi mekanisme autentikasi, validasi masukan (*input validation*), manajemen sesi, konfigurasi keamanan aplikasi web, serta layanan yang dapat diakses melalui antarmuka aplikasi. Pengujian tidak mencakup serangan yang berpotensi mengganggu operasional sistem, seperti *denial of service*, perubahan data produksi, penghapusan data, maupun pengujian terhadap infrastruktur jaringan internal perusahaan. Metodologi PTES digunakan sebagai pedoman pelaksanaan pengujian karena menyediakan tahapan yang sistematis mulai dari *pre-engagement*, *intelligence gathering*, *threat modeling*, *vulnerability analysis*, *exploitation*, hingga *reporting* (Penetration Testing Execution Standard, 2014). Identifikasi kerentanan dilakukan menggunakan *OWASP ZAP* sebagai alat utama *vulnerability assessment*. Selanjutnya, setiap kerentanan diklasifikasikan berdasarkan *Common Weakness Enumeration* (*CWE*) (MITRE, 2024), sedangkan tingkat keparahannya dinilai menggunakan *Common Vulnerability Scoring System* (*CVSS*) versi 3.1 (FIRST, 2019) untuk menentukan prioritas mitigasi.

Kerangka PTES, OWASP Zap dan CVSS

Penelitian ini menerapkan PTES sebagai metodologi utama karena menyediakan tahapan pengujian keamanan yang terstruktur mulai dari perencanaan hingga pelaporan hasil. Identifikasi kerentanan dilakukan menggunakan *OWASP ZAP* melalui kombinasi *passive scanning* dan *active scanning*. Informasi pendukung pada tahap *intelligence gathering* diperoleh menggunakan *Wappalyzer* dan *Nmap* untuk mengidentifikasi teknologi, layanan jaringan, serta *endpoint* yang tersedia.

Kerentanan yang ditemukan kemudian diklasifikasikan berdasarkan *Common Weakness Enumeration* (*CWE*) sehingga setiap temuan dapat dipetakan ke jenis kelemahan perangkat lunak yang sesuai. Selanjutnya, tingkat keparahan setiap kerentanan dihitung menggunakan *Common Vulnerability Scoring System* (*CVSS*) versi 3.1 sehingga diperoleh kategori risiko sebagai dasar penyusunan prioritas mitigasi.

Tabel 1. Ringkasan alur pengujian dan *output* penelitian

Tahap	**Aktivitas Utama**	**Alat/Standar**	**Output**
Tahap PTES	Aktivitas Utama	Alat/Standar	Output
Pre-engagement	Menentukan ruang lingkup, tujuan, dan izin pengujian	PTES	Ruang lingkup pengujian
Intelligence Gathering	Mengidentifikasi teknologi, layanan, endpoint, dan informasi target	OWASP ZAP, Nmap, Wappalyzer	Informasi permukaan serangan
Threat Modeling	Menganalisis potensi ancaman berdasarkan informasi yang diperoleh	PTES	Skenario ancaman
Vulnerability Analysis	Mengidentifikasi dan menganalisis kerentanan keamanan aplikasi	OWASP ZAP	Daftar kerentanan
Exploitation	Memvalidasi kerentanan yang ditemukan secara terbatas	Pengujian manual	Kerentanan tervalidasi

Teknik Analisis Data

Data penelitian diperoleh dari hasil pemindaian menggunakan *OWASP ZAP*, observasi manual, analisis respons HTTP, serta validasi terhadap setiap kerentanan yang teridentifikasi. Analisis data mengacu pada tahapan analisis dalam PTES (Penetration Testing Execution Standard, 2014) dan rekomendasi klasifikasi kerentanan berdasarkan CWE (MITRE, 2024). Tahap pertama adalah konsolidasi hasil pemindaian dengan mengelompokkan temuan yang memiliki karakteristik serupa sehingga mengurangi kemungkinan duplikasi maupun *false positive*. Tahap berikutnya adalah klasifikasi kerentanan berdasarkan *Common Weakness Enumeration* (*CWE*) untuk mengidentifikasi jenis kelemahan perangkat lunak yang mendasari setiap temuan. Selanjutnya dilakukan analisis dampak kerentanan terhadap aspek *confidentiality*, *integrity*, dan *availability* (Kar Yee & Zolkipli, 2021).

Kerentanan yang telah teridentifikasi kemudian divalidasi melalui pengujian manual secara terbatas sesuai ruang lingkup penelitian untuk memastikan bahwa temuan dapat direproduksi (*reproducible*) tanpa mengganggu operasional sistem. Tahapan validasi ini sejalan dengan rekomendasi PTES yang menekankan pentingnya verifikasi terhadap hasil *vulnerability scanning* sebelum penyusunan laporan akhir (Penetration Testing Execution

Standard, 2014). Tingkat keparahan masing-masing kerentanan kemudian dinilai menggunakan *Common Vulnerability Scoring System* (*CVSS*) versi 3.1 (FIRST, 2019). Suatu kerentanan dinyatakan sebagai temuan utama apabila memenuhi tiga kriteria, yaitu dapat direproduksi selama pengujian, memiliki dampak terhadap keamanan sistem, dan dapat dipetakan ke kategori *Common Weakness Enumeration* (*CWE*). Temuan yang hanya terdeteksi sebagai peringatan otomatis tanpa bukti eksploitasi yang memadai tidak dimasukkan sebagai temuan utama, tetapi tetap dicatat sebagai potensi risiko untuk evaluasi lanjutan.

HASIL DAN DISKUSI

Hasil *Intelligence Gathering*

Tahap intelligence gathering menghasilkan gambaran awal mengenai komponen teknologi, layanan jaringan, endpoint, parameter, dan struktur direktori yang dapat diamati dari sisi penguji. Hasil Wappalyzer menunjukkan penggunaan Apache 2.4.6, PHP 5.4.16, MariaDB 5.5.64, jQuery 3.1.1, HTML5, dan Bootstrap. Beberapa komponen tersebut perlu diperhatikan karena versi yang teridentifikasi tergolong lama dan dapat meningkatkan risiko apabila tidak mendapat pembaruan keamanan yang memadai. Pemindaian Nmap menunjukkan port 80, 443, 22, dan 3306 dalam kondisi terbuka. Port 80 dan 443 berkaitan dengan akses aplikasi web, sedangkan port 22 berkaitan dengan administrasi server. Temuan yang perlu diprioritaskan adalah port 3306 karena layanan basis data sebaiknya tidak terekspos langsung ke jaringan publik. Enumerasi parameter juga menunjukkan bahwa beberapa endpoint menerima input yang langsung berkaitan dengan proses autentikasi dan pencarian data. Parameter username dan password berkaitan dengan risiko brute force, sedangkan parameter id, user_id, dan keyword berkaitan dengan validasi input, manipulasi identifier, dan kemungkinan injeksi. Informasi ini digunakan untuk menentukan area uji yang memiliki peluang risiko paling relevan terhadap aplikasi ERP.

Tabel 2. Ringkasan Permukaan Serangan Awal

Kategori	Temuan Penting	Implikasi Keamanan
Teknologi aplikasi	Apache 2.4.6, PHP 5.4.16, MariaDB 5.5.64, jQuery 3.1.1, Bootstrap	Versi lama perlu diperiksa patch dan dukungan keamanannya
Layanan jaringan	HTTP, HTTPS, SSH, dan MySQL terdeteksi terbuka	SSH dan MySQL perlu dibatasi dengan firewall atau allowlist IP
Parameter input	*username, password, id, user_id, keyword*	Parameter autentikasi dan pencarian menjadi titik uji validasi input
Direktori	*/application, /asset, /data, /pos, /system, /index.php*	Struktur internal aplikasi dapat membantu enumerasi lanjutan
Temuan awal	*Missing Security Header, Server Version Disclosure, Cookie Without Secure Flag*	Indikasi awal security misconfiguration dan kelemahan manajemen cookie

Hasil *Threat Modeling dan Vulnerability Analysis*

Pemindaian OWASP ZAP menghasilkan beberapa kategori risiko yang kemudian dikonsolidasikan agar tidak terjadi penghitungan ganda. Temuan utama meliputi *Content Security Policy Header Not Set, Absence of Anti-CSRF Tokens, Server Version Disclosure, Cookie No HttpOnly Flag, X-Content-Type-Options Header Missing, Retrieved from Cache, Suspicious Comments,* dan *Sensitive Information in URL*. Temuan ini menunjukkan bahwa sebagian besar risiko awal berkaitan dengan konfigurasi header, manajemen cookie, validasi request, serta pengungkapan informasi.

Tabel 3. Ringkasan Hasil Active Scanning OWASP ZAP

No	Jenis Kerentanan	Risk	Confidence	Jumlah
1	Content Security Policy Header Not Set	Medium	High	1
2	Absence of Anti-CSRF Tokens	Medium	Low	1
3	Server Leaks Version Information	Low	High	1
4	Cookie No HttpOnly Flag	Low	Medium	1
5	X-Content-Type-Options Header Missing	Low	Medium	5
6	Retrieved from Cache	Informational	Medium	2
7	Suspicious Comments	Informational	Low	1
8	Sensitive Information in URL	Informational	Medium	2

Temuan pada Tabel 3 menunjukkan bahwa tidak seluruh risiko yang teridentifikasi merupakan kerentanan yang dapat dieksploitasi secara langsung. Sebagian besar temuan berupa *security misconfiguration* yang secara individual memiliki tingkat risiko relatif rendah, tetapi dapat meningkatkan dampak apabila dikombinasikan dengan kerentanan lain. Misalnya, *cookie* tanpa atribut *HttpOnly* memungkinkan *session cookie* diakses melalui *JavaScript* apabila terjadi *Cross-Site Scripting* (XSS), sedangkan tidak diterapkannya *Content Security Policy* (CSP) memperluas peluang eksekusi skrip berbahaya pada sisi klien. Kondisi tersebut menunjukkan bahwa kelemahan konfigurasi keamanan pada ERP XYZ belum menerapkan mekanisme *defense in depth*, sehingga beberapa lapisan perlindungan yang seharusnya saling melengkapi belum tersedia.

Hasil pemetaan terhadap *Common Weakness Enumeration* (CWE) menunjukkan bahwa kerentanan yang ditemukan meliputi *CWE-352, CWE-693, CWE-264, CWE-1021, CWE-345, CWE-1395, CWE-1004, CWE-614, CWE-1275, CWE-829, CWE-497, CWE-319, CWE-598, CWE-615,* dan *CWE-525*. Namun, hanya sebagian temuan yang divalidasi melalui tahap eksploitasi karena penelitian mengikuti *rules of engagement* untuk menghindari gangguan terhadap operasional sistem. Selain itu, hasil pemindaian otomatis dikonsolidasikan agar *alert* yang muncul pada beberapa *endpoint* tidak dihitung sebagai kerentanan yang berbeda.

Pendekatan ini menghasilkan daftar temuan yang lebih representatif terhadap kondisi keamanan ERP XYZ.

Empat kerentanan yang berhasil divalidasi, yaitu *SQL Injection*, *Cross-Site Scripting* (XSS), *Brute Force Login Page*, dan penyalahgunaan *session* aktif, menunjukkan bahwa kelemahan keamanan pada ERP XYZ tidak hanya berasal dari kesalahan konfigurasi, tetapi juga dari mekanisme validasi masukan dan pengelolaan autentikasi. Kerentanan *SQL Injection* mengindikasikan bahwa sebagian fungsi aplikasi belum menerapkan *parameterized query* atau *prepared statement* secara konsisten sehingga masukan pengguna masih dapat diproses sebagai bagian dari perintah SQL. Sementara itu, kerentanan XSS menunjukkan bahwa proses validasi dan *output encoding* terhadap data yang ditampilkan kembali ke peramban belum diterapkan secara menyeluruh. Kedua kondisi tersebut mengindikasikan bahwa mekanisme validasi input pada sisi aplikasi masih belum optimal.

Kerentanan pada mekanisme autentikasi juga menunjukkan adanya kelemahan pada implementasi kontrol keamanan. Risiko *Brute Force Login Page* muncul karena aplikasi belum menerapkan pembatasan jumlah percobaan login, *account lockout*, maupun *rate limiting*, sehingga penyerang memiliki peluang melakukan percobaan kredensial secara berulang. Di sisi lain, risiko penyalahgunaan *session* aktif menunjukkan bahwa pengelolaan sesi pengguna belum sepenuhnya mengikuti praktik terbaik, misalnya belum diterapkannya pengaturan waktu kedaluwarsa sesi yang memadai, regenerasi *session ID* setelah autentikasi, atau konfigurasi atribut *cookie* seperti *Secure*, *HttpOnly*, dan *SameSite*. Kombinasi kelemahan tersebut dapat meningkatkan peluang pengambilalihan akun apabila kredensial atau *session cookie* berhasil diperoleh oleh pihak yang tidak berwenang. Temuan ini menunjukkan bahwa sebagian besar kerentanan pada ERP XYZ berasal dari implementasi kontrol keamanan aplikasi yang belum memenuhi prinsip *secure coding* dan *secure configuration*. Oleh karena itu, perbaikan tidak cukup dilakukan pada kerentanan individual, tetapi juga perlu mencakup penerapan praktik pengembangan perangkat lunak yang aman, penguatan konfigurasi keamanan aplikasi, serta pengujian keamanan secara berkala agar kerentanan serupa dapat dideteksi sebelum sistem digunakan pada lingkungan produksi.

Validasi Eksploitasi

Tahap *exploitation* pada *Penetration Testing Execution Standard* (PTES) dilakukan untuk memvalidasi kerentanan yang telah diidentifikasi pada tahap *vulnerability analysis*. Validasi dilakukan secara terbatas terhadap empat temuan utama, yaitu *SQL Injection*, *Cross-Site Scripting* (XSS), *Brute Force Login Page*, dan risiko penyalahgunaan *session* aktif. Proses

validasi bertujuan memastikan bahwa setiap kerentanan benar-benar dapat direproduksi sesuai ruang lingkup pengujian tanpa mengganggu operasional sistem serta membedakan temuan yang benar-benar dapat dieksploitasi dari *false positive*.

Kerentanan *SQL Injection* ditemukan pada parameter pencarian yang berinteraksi langsung dengan basis data. Hasil validasi menunjukkan bahwa masukan pengguna belum diproses menggunakan mekanisme *parameterized query* atau *prepared statement* secara konsisten, sehingga sebagian input masih diperlakukan sebagai bagian dari perintah SQL. Kondisi ini mengindikasikan bahwa validasi masukan dan pemisahan antara data dengan perintah (*query separation*) belum diterapkan secara optimal. Apabila tidak segera diperbaiki, kerentanan tersebut berpotensi dimanfaatkan untuk membaca, memodifikasi, atau menghapus data penting pada sistem ERP. Kerentanan *Cross-Site Scripting* (XSS) menunjukkan bahwa beberapa kolom masukan belum menerapkan validasi dan *output encoding* secara menyeluruh sebelum data ditampilkan kembali ke peramban pengguna. Selain itu, tidak ditemukannya penerapan *Content Security Policy* (CSP) memperbesar peluang eksekusi skrip berbahaya apabila terdapat masukan yang lolos dari proses penyaringan. Kondisi tersebut mengindikasikan bahwa mekanisme perlindungan pada sisi aplikasi (*server-side*) maupun sisi klien (*client-side*) belum diterapkan secara berlapis. Akibatnya, penyerang berpotensi mencuri *session cookie*, memanipulasi tampilan aplikasi, atau menjalankan aksi atas nama pengguna yang telah melakukan autentikasi.

Pada mekanisme autentikasi, temuan *Brute Force Login Page* menunjukkan bahwa sistem belum menerapkan kontrol keamanan seperti *rate limiting*, pembatasan jumlah percobaan login (*login attempt limitation*), *account lockout*, maupun *progressive delay*. Ketiadaan mekanisme tersebut memungkinkan percobaan kredensial dilakukan secara berulang hingga ditemukan kombinasi nama pengguna dan kata sandi yang valid. Kerentanan ini menunjukkan bahwa implementasi autentikasi pada ERP XYZ masih berfokus pada proses verifikasi kredensial, tetapi belum dilengkapi mekanisme perlindungan terhadap serangan otomatis (*automated authentication attacks*).

Risiko penyalahgunaan *session* aktif berkaitan dengan pengelolaan sesi pengguna setelah autentikasi berhasil dilakukan. Hasil validasi menunjukkan bahwa konfigurasi *session management* masih dapat ditingkatkan, terutama terkait masa berlaku sesi, regenerasi *session ID* setelah proses login, serta penerapan atribut keamanan *cookie* seperti *HttpOnly*, *Secure*, dan *SameSite*. Kondisi tersebut meningkatkan peluang penyalahgunaan sesi apabila *session token* berhasil diperoleh melalui serangan lain, misalnya XSS atau akses terhadap perangkat pengguna. Temuan ini menunjukkan bahwa kelemahan *session management* pada ERP XYZ

tidak berdiri sendiri, tetapi dapat memperbesar dampak kerentanan lain apabila dieksploitasi secara bersamaan. Hasil validasi menunjukkan bahwa kerentanan pada ERP XYZ terutama disebabkan oleh belum optimalnya penerapan praktik *secure coding* dan *secure configuration*. Sebagian besar kelemahan berasal dari mekanisme validasi masukan, pengelolaan autentikasi, dan *session management* yang belum sepenuhnya mengikuti praktik keamanan aplikasi web. Mitigasi tidak hanya difokuskan pada penutupan setiap kerentanan, tetapi juga perlu diarahkan pada perbaikan proses pengembangan perangkat lunak yang aman agar kerentanan serupa dapat dicegah pada pengembangan sistem berikutnya.

Tabel 4. Temuan tervalidasi dan pemetaan standar keamanan

Kerentanan	CWE	Status Validasi	Dampak Utama
SQL Injection	CWE-89	Tervalidasi	Akses tidak sah terhadap basis data dan manipulasi data
Cross-Site Scripting (XSS)	CWE-79	Tervalidasi	Pencurian sesi, manipulasi halaman, dan eksekusi skrip pada browser
Brute Force Login Page	CWE-307	Tervalidasi	Pengambilalihan akun melalui percobaan autentikasi berulang
Risiko Penyalahgunaan Session Aktif	CWE-613	Tervalidasi	Penyalahgunaan sesi pengguna yang masih aktif

Selain empat temuan utama, beberapa skenario dari matriks pengujian tidak dimasukkan sebagai temuan akhir karena tidak ditemukan indikator awal atau belum dapat divalidasi secara aman dalam ruang lingkup penelitian. Status ini penting dicantumkan agar pembaca dapat membedakan antara kerentanan yang terbukti, potensi risiko yang perlu diuji lanjutan, dan skenario yang tidak ditemukan pada endpoint yang diuji.

Tabel 5. Status validasi skenario pengujian eksploitasi

Skenario	Status	Keterangan
SQL Injection	Tervalidasi	Parameter pencarian menunjukkan respons yang konsisten dengan kelemahan injeksi
Cross-Site Scripting	Tervalidasi	Input tertentu dapat memicu eksekusi skrip pada sisi browser
Brute Force Login	Tervalidasi	Form login belum menunjukkan pembatasan percobaan yang cukup kuat
Session aktif	Tervalidasi	Sesi valid masih dapat disalahgunakan jika token atau cookie bocor
CSRF	Perlu validasi lanjutan	Token anti-CSRF tidak terdeteksi, tetapi aksi lintas situs belum dibuktikan secara penuh
Authorization bypass dan IDOR	Tidak tervalidasi	Belum ditemukan akses data di luar hak pengguna dari akun yang tersedia
Path traversal, command injection, SSRF	Tidak ditemukan	Tidak ditemukan parameter yang relevan untuk membuktikan skenario tersebut
Resource exhaustion	Tidak diuji mendalam	Pengujian beban agresif dibatasi untuk menjaga ketersediaan layanan

Penilaian *CVSS* dan Prioritas Mitigasi

Penilaian menggunakan *Common Vulnerability Scoring System* (CVSS) v3.1 menunjukkan bahwa *SQL Injection* dan *Brute Force Login Page* berada pada kategori *High*, sehingga menjadi prioritas utama dalam mitigasi. Tingginya tingkat risiko *SQL Injection* menunjukkan bahwa mekanisme validasi masukan dan penggunaan *parameterized query* pada ERP XYZ belum diterapkan secara konsisten, sehingga membuka peluang terjadinya manipulasi maupun akses tidak sah terhadap basis data. Sementara itu, tingginya skor *Brute Force Login Page* mengindikasikan bahwa mekanisme autentikasi belum dilengkapi kontrol keamanan seperti *rate limiting*, pembatasan percobaan login, atau *account lockout*. Kondisi tersebut menunjukkan bahwa kelemahan utama pada ERP XYZ tidak hanya berasal dari kesalahan konfigurasi, tetapi juga dari implementasi kontrol keamanan aplikasi yang belum optimal.

Di sisi lain, *Cross-Site Scripting* (XSS) dan risiko penyalahgunaan *session* aktif memperoleh kategori *Medium*, namun tetap memerlukan perhatian karena dapat memperbesar dampak kerentanan lain. XSS mengindikasikan bahwa proses validasi masukan dan *output encoding* belum diterapkan secara menyeluruh, sedangkan kelemahan *session management* menunjukkan bahwa konfigurasi *cookie* dan pengelolaan sesi masih dapat ditingkatkan. Kombinasi kedua kerentanan tersebut berpotensi dimanfaatkan untuk melakukan pengambilalihan sesi pengguna apabila *session cookie* berhasil diperoleh. Secara keseluruhan, hasil penilaian CVSS menunjukkan bahwa risiko keamanan pada ERP XYZ lebih banyak mengancam aspek *confidentiality* dan *integrity*, sehingga prioritas mitigasi perlu difokuskan pada penguatan validasi input, autentikasi, dan pengelolaan sesi sesuai praktik *secure coding* dan *secure configuration*.

Tabel 6. Skor CVSS dan Prioritas Mitigasi

Temuan	Skor	Severity	Prioritas	Rekomendasi Utama
SQL Injection	8,3	High	1	Gunakan parameterized query, validasi input server-side, least privilege pada akun database, dan audit query berisiko
Brute Force Login Page	8,2	High	2	Terapkan rate limiting, lockout adaptif, CAPTCHA berbasis risiko, MFA, dan monitoring percobaan login
Risiko Session Aktif	4,2	Medium	2	Gunakan HttpOnly, Secure, SameSite, session timeout, rotasi session ID, dan invalidasi saat logout
Cross-Site Scripting	5,4	Medium	3	Terapkan output encoding, validasi input, Content Security Policy, dan sanitasi HTML pada field masukan

Agar rekomendasi dapat diterapkan secara operasional, mitigasi dikelompokkan berdasarkan lapisan sistem. Pengelompokan ini membantu tim pengembang, administrator server, dan manajemen menentukan pekerjaan yang harus dilakukan tanpa mencampur kontrol aplikasi, infrastruktur, dan kebijakan.

Tabel 7. Mitigasi Teknis Berdasarkan Lapisan Sistem

Lapisan	Risiko yang Ditangani	Kontrol yang Disarankan
Aplikasi	*SQL Injection* dan XSS	Parameterized query, validasi server-side, output encoding, sanitasi input, dan Content Security Policy
Autentikasi	*Brute force* dan *account takeover*	Rate limiting, lockout adaptif, MFA, alert login gagal, dan audit log autentikasi
Session management	Penyalahgunaan sesi aktif	HttpOnly, Secure, SameSite, rotasi session ID, idle timeout, absolute timeout, dan invalidasi logout
Infrastruktur	Eksposur SSH dan database	Firewall, allowlist IP, penutupan port publik yang tidak diperlukan, dan pembatasan akses administrasi
Tata kelola	Risiko berulang setelah perbaikan	Patch management, retesting berkala, dokumentasi temuan, dan review keamanan sebelum rilis

Penerapan *Penetration Testing Execution Standard* (PTES) memberikan alur pengujian yang sistematis sehingga proses identifikasi, validasi, dan pelaporan kerentanan dapat dilakukan secara terstruktur. Temuan penelitian menunjukkan bahwa sebagian besar kerentanan pada ERP XYZ bukan disebabkan oleh kesalahan pada satu komponen tertentu, tetapi berasal dari belum optimalnya penerapan prinsip *secure coding* dan *secure configuration*. Hal ini terlihat dari masih ditemukannya kelemahan pada validasi masukan, mekanisme autentikasi, pengelolaan *session*, serta konfigurasi keamanan aplikasi. Kondisi tersebut mengindikasikan bahwa aspek keamanan belum sepenuhnya diintegrasikan ke dalam *secure software development lifecycle* (SSDLC), sehingga beberapa kontrol keamanan masih diterapkan setelah aplikasi dikembangkan. Penggunaan OWASP ZAP mempermudah proses identifikasi kerentanan pada aplikasi web, sedangkan klasifikasi *Common Weakness Enumeration* (CWE) dan penilaian *Common Vulnerability Scoring System* (CVSS) membantu mengidentifikasi akar kelemahan perangkat lunak sekaligus menentukan prioritas mitigasi berdasarkan tingkat risiko.

Temuan penelitian ini sejalan dengan Astrida et al. (2022) yang menunjukkan bahwa PTES mampu menghasilkan proses *penetration testing* yang lebih sistematis karena setiap tahap pengujian memiliki keluaran yang jelas. Hasil penelitian ini juga konsisten dengan Rahman et al. (2023), yang melaporkan bahwa penggunaan OWASP ZAP efektif mengidentifikasi kerentanan pada aplikasi web, terutama yang berkaitan dengan validasi masukan dan konfigurasi keamanan. Sementara itu, penelitian Bagaskara et al. (2025) berfokus pada

identifikasi kerentanan berdasarkan *OWASP Top 10*, sedangkan penelitian ini melengkapinya dengan validasi eksploitasi, klasifikasi berdasarkan CWE, dan penilaian tingkat keparahan menggunakan CVSS v3.1 sehingga menghasilkan prioritas mitigasi yang lebih terukur. Berbeda pula dengan penelitian Pratama dan Nugroho (2024), yang hanya melaporkan hasil pemindaian otomatis tanpa validasi eksploitasi, penelitian ini memastikan bahwa setiap kerentanan utama dapat direproduksi dalam ruang lingkup pengujian sehingga mengurangi kemungkinan *false positive*. Dengan demikian, kombinasi PTES, OWASP ZAP, CWE, dan CVSS memberikan evaluasi keamanan yang lebih komprehensif serta dapat menjadi dasar penyusunan strategi mitigasi yang lebih tepat sasaran.

KESIMPULAN

Penelitian ini menunjukkan bahwa penerapan *Penetration Testing Execution Standard* (PTES) yang didukung OWASP ZAP mampu mengidentifikasi dan memvalidasi kerentanan keamanan pada sistem *Enterprise Resource Planning* (ERP) XYZ secara sistematis. Empat kerentanan utama berhasil diidentifikasi, dengan *SQL Injection* dan *Brute Force Login Page* menjadi prioritas mitigasi karena memiliki tingkat risiko tertinggi berdasarkan penilaian *Common Vulnerability Scoring System* (CVSS) v3.1. Temuan tersebut mengindikasikan bahwa kelemahan keamanan pada ERP XYZ terutama berkaitan dengan validasi masukan, mekanisme autentikasi, dan pengelolaan *session* yang belum menerapkan praktik *secure coding* dan *secure configuration* secara optimal.

Penelitian ini memberikan kontribusi berupa penerapan pendekatan PTES yang dipadukan dengan OWASP ZAP, klasifikasi *Common Weakness Enumeration* (CWE), dan penilaian CVSS untuk menghasilkan evaluasi keamanan yang tidak hanya mengidentifikasi kerentanan, tetapi juga membantu menentukan prioritas mitigasi berbasis risiko. Hasil penelitian diharapkan dapat menjadi acuan bagi pengembang dan administrator sistem dalam meningkatkan keamanan aplikasi ERP melalui penguatan mekanisme validasi input, autentikasi, *session management*, serta pelaksanaan pengujian keamanan secara berkala. Penelitian selanjutnya disarankan melibatkan audit kode sumber, pengujian API, berbagai tingkat hak akses pengguna, dan *retesting* setelah mitigasi diterapkan untuk mengevaluasi efektivitas perbaikan keamanan.

REFERENSI

Astrida, D. N., Saputra, A. R., & Assaufi, A. I. (2022). Analysis and evaluation of wireless network security with the Penetration Testing Execution Standard (PTES). Sinkron, 7(1), 147–154. https://doi.org/10.33395/sinkron.v7i1.11249

Bagaskara, B. A., Idhom, M., & Wahanani, H. E. (2025). Pengujian website Dinas Sosial Surabaya menggunakan metode penetration testing dan OWASP Top 10. Jurnal Inform, 8(1).

Common Weakness Enumeration (CWE) Content Team. (2024). CWE Top 25 most dangerous software weaknesses. https://cwe.mitre.org/top25/

FIRST. (2019). Common Vulnerability Scoring System (CVSS) v3.1 specification document. https://www.first.org/cvss/v3.1/specification-document

Kar Yee, C., & Zolkipli, M. F. (2021). Review on confidentiality, integrity and availability in information security. Journal of ICT in Education, 8(2), 34–42. https://doi.org/10.37134/jictie.vol8.2.4.2021

Penetration Testing Execution Standard. (2014). The Penetration Testing Execution Standard. http://www.pentest-standard.org/

Pratama, R., & Nugroho, A. (2024). Implementasi penetration testing pada aplikasi berbasis web menggunakan OWASP ZAP. Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi), 8(2), 245–254.

Rahman, M. A., Hidayat, T., & Nugraha, A. (2023). Analisis kerentanan keamanan aplikasi web menggunakan OWASP ZAP. Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), 7(5), 1023–1032.

Security Journey. (2021). Why vulnerability list methodologies matter (and why we trust CWE & OWASP). https://www.securityjourney.com/post/why-vulnerability-list-methodologies-matter-and-why-we-trust-cwe-owasp :contentReference[oaicite:1]{index=1}

Tambunan, E. (2025). Enterprise resource planning (ERP) dalam transformasi digital: Konsep, evolusi, implementasi, dan tantangannya (Vol. 2).

The Straits Times. (2024). More than 40 Indonesian agencies hit by cyberattack on data centres.