ISSN (Online) 2747-0563
Volume 6 Tahun 2026

Analisis Kerentanan Sistem ERP XYZ Berdasarkan OWASP ZAP dengan Pendekatan PTES

Ken Narendra Ekamartha[1], Henni Endah Wahanani[2*], Achmad Junaidi[3]

[1,2,3] Informatika, Universitas Pembangunan Nasional "Veteran" Jawa Timur
[1]22081010083@student.upnjatim.ac.id
[3]achmadjunaidi.if@upnjatim.ac.id

[2] Informatika Universitas Pembangunan Nasional "Veteran" Jawa Timur
**Corresponding author email*: henniendah@upnjatim.ac.id

***Abstrak*— Perkembangan teknologi informasi mendorong perusahaan untuk mengadopsi sistem Enterprise Resource Planning (ERP) sebagai solusi terintegrasi dalam pengelolaan proses bisnis. Namun, implementasi ERP berbasis web juga meningkatkan potensi risiko keamanan siber yang dapat mengancam kerahasiaan, integritas, dan ketersediaan data perusahaan. Oleh karena itu, identifikasi dan analisis kerentanan pada sistem ERP menjadi langkah penting untuk memastikan sistem dapat beroperasi secara aman dan andal. Penelitian ini bertujuan untuk menganalisis kerentanan keamanan pada sistem ERP XYZ yang digunakan oleh PT XYZ berdasarkan temuan kerentanan OWASP ZAP. Proses pengujian dilakukan menggunakan metodologi Penetration Testing Execution Standard (PTES) yang menyediakan tahapan pengujian keamanan secara sistematis, mulai dari pengumpulan informasi, identifikasi kerentanan, hingga analisis dan pelaporan hasil pengujian. Kerentanan yang ditemukan kemudian dianalisis dan dipetakan untuk mengetahui jenis dan tingkat risiko kerentanan yang terdapat pada sistem. Hasil analisis menunjukkan bahwa sebagian kerentanan memiliki tingkat risiko yang signifikan dan berpotensi dieksploitasi oleh pihak tidak bertanggung jawab. Oleh karena itu, diperlukan penerapan langkah mitigasi yang tepat, seperti validasi dan sanitasi input, penguatan mekanisme autentikasi, perbaikan manajemen sesi, serta konfigurasi keamanan server sesuai praktik terbaik**

Kata Kunci— **ERP, Kerentanan Sistem, OWASP ZAP, PTES, Penetration Testing.**

I. PENDAHULUAN

Perkembangan teknologi informasi telah membawa perubahan signifikan terhadap sistem bisnis modern, termasuk dalam pemanfaatan sistem Enterprise Resource Planning (ERP). ERP merupakan sistem informasi terintegrasi yang dirancang untuk mengelola dan menyelaraskan berbagai proses bisnis utama dalam sebuah organisasi, seperti keuangan, persediaan, produksi, hingga sumber daya manusia[1]. Dengan memusatkan proses bisnis dan data operasional ke dalam satu platform, ERP diharapkan mampu meningkatkan efisiensi operasional, konsistensi data, serta mendukung pengambilan keputusan yang lebih efektif dalam organisasi.

Seiring dengan meningkatnya implementasi ERP berbasis web, risiko terhadap ancaman keamanan siber juga semakin meningkat. Sistem berbasis web memiliki potensi kerentanan yang dapat dimanfaatkan oleh pihak tidak bertanggung jawab untuk memperoleh akses tidak sah terhadap sistem maupun data perusahaan. Berdasarkan laporan dari Fortinet (2024), serangan siber di Indonesia mengalami peningkatan sebesar 43% dibandingkan tahun sebelumnya[2]. Temuan ini mendukung laporan Badan Siber dan Sandi Negara (2025), yang mengungkap bahwa ancaman siber BSSN tahun 2025 semakin serius, khususnya bagi sektor finansial Indonesia. Serangan digital kini tidak lagi sekadar mencuri data, melainkan langsung menargetkan kerugian finansial dan reputasi bisnis. Mulai dari ransomware yang melumpuhkan operasional layanan publik, hingga social engineering berbasis AI yang menipu karyawan dengan modus *deepfake voice*, tren ini menegaskan rapuhnya ekosistem keuangan digital nasional[3]. Kondisi ini menunjukkan bahwa keamanan aplikasi berbasis web, termasuk sistem ERP, menjadi aspek yang sangat penting untuk diperhatikan.

Berbagai penelitian menunjukkan bahwa serangan terhadap aplikasi web umumnya disebabkan oleh adanya kerentanan pada sistem, seperti kelemahan autentikasi, kesalahan konfigurasi server, serta kurangnya validasi terhadap input pengguna. Kerentanan tersebut dapat dimanfaatkan oleh penyerang untuk melakukan berbagai bentuk eksploitasi, seperti pencurian data, manipulasi informasi, maupun gangguan terhadap operasional sistem.

Meskipun penelitian mengenai keamanan aplikasi web telah banyak dilakukan, hingga saat ini belum terdapat penelitian yang secara khusus menganalisis kerentanan pada sistem ERP XYZ yang digunakan oleh PT XYZ. Kondisi ini menunjukkan adanya kesenjangan penelitian (research gap) yang perlu dikaji lebih lanjut untuk mengetahui kondisi keamanan aktual sistem tersebut.

Berdasarkan permasalahan tersebut, penelitian ini dilakukan untuk menganalisis kerentanan keamanan pada sistem ERP XYZ dengan mengacu pada temuan kerentanan dalam OWASP ZAP. Proses pengujian dilakukan menggunakan metodologi Penetration Testing Execution Standard (PTES) yang menyediakan tahapan pengujian keamanan secara sistematis, mulai dari pengumpulan informasi, identifikasi kerentanan, hingga analisis dan pelaporan hasil pengujian[4]. Melalui pendekatan tersebut, penelitian ini diharapkan mampu mengidentifikasi potensi kerentanan pada sistem ERP XYZ serta memberikan rekomendasi mitigasi yang dapat diterapkan untuk meningkatkan keamanan sistem. Melalui pendekatan tersebut, penelitian ini diharapkan mampu mengidentifikasi potensi kerentanan pada sistem ERP XYZ serta memberikan

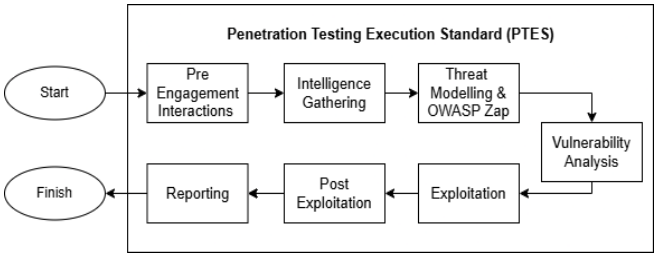
ISSN (Online) 2747-0563
Volume 6 Tahun 2026

rekomendasi mitigasi yang dapat diterapkan untuk meningkatkan keamanan sistem, sekaligus meminimalkan risiko serangan siber di masa mendatang.

II. METODOLOGI PENELITIAN

Metodologi penelitian yang dilakukan sebagaimana ditunjukkan pada Gbr. 1 diawali dengan metodologi Penetration Testing Execution Standard (PTES). Setelah itu dilakukan tahap pre-engagement interaction untuk menentukan ruang lingkup pengujian serta objek penelitian yaitu sistem ERP XYZ. Tahap berikutnya adalah intelligence gathering, yaitu proses pengumpulan informasi terkait sistem yang akan diuji. Informasi yang diperoleh kemudian digunakan pada tahap threat modeling untuk mengidentifikasi potensi ancaman yang mungkin terjadi pada sistem.

Selanjutnya dilakukan vulnerability analysis untuk mengidentifikasi kerentanan pada sistem ERP XYZ melalui proses pengujian keamanan. Kerentanan yang ditemukan kemudian dianalisis dan dipetakan untuk mengetahui jenis kerentanan yang terdapat pada sistem. Tahap berikutnya adalah exploitation, yaitu pengujian lebih lanjut terhadap kerentanan yang ditemukan untuk mengetahui potensi eksploitasi yang dapat dilakukan oleh penyerang. Setelah itu dilakukan tahap post-exploitation untuk menganalisis dampak dari kerentanan yang berhasil dieksploitasi terhadap keamanan sistem. Tahap terakhir adalah reporting, yaitu penyusunan laporan hasil pengujian yang berisi temuan kerentanan serta rekomendasi mitigasi guna meningkatkan keamanan sistem ERP XYZ.

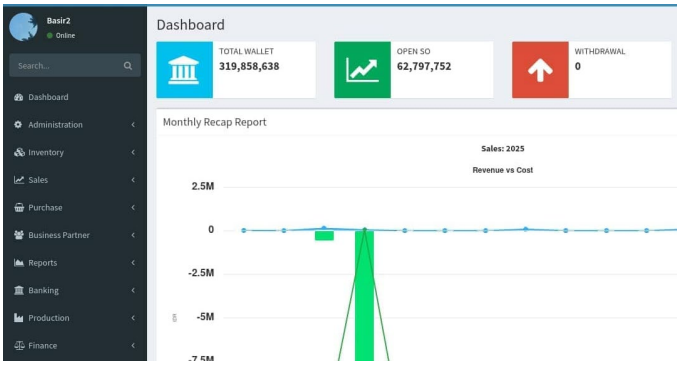


Gbr. 1 Metode Penelitian

A. *Pre-Engagement Interaction*

Pada tahap ini ditentukan objek penelitian serta ruang lingkup pengujian keamanan yang akan dilakukan[5]. Objek penelitian adalah sistem Enterprise Resource Planning (ERP) yang digunakan oleh PT XYZ, yaitu sebuah sistem berbasis web yang berfungsi untuk mengintegrasikan berbagai proses bisnis perusahaan dalam satu platform terpusat.

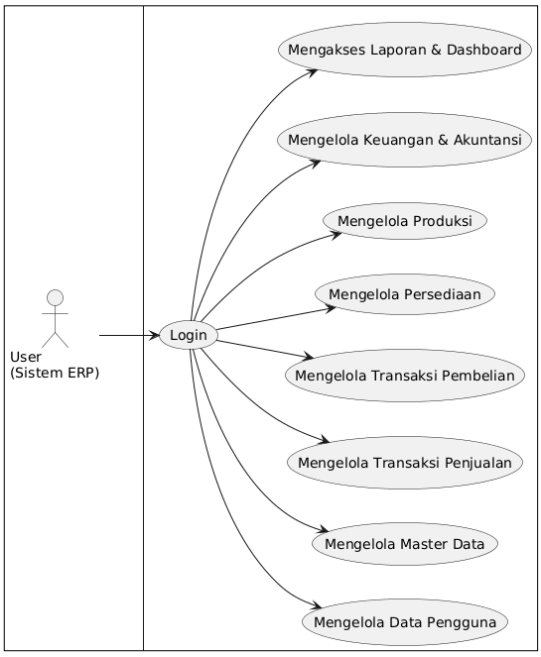
Pengujian difokuskan pada aspek keamanan aplikasi web yang dapat diakses melalui antarmuka pengguna, seperti mekanisme autentikasi, pengelolaan hak akses, serta proses pengolahan input dan output data. Penelitian ini tidak mencakup pengujian terhadap perangkat keras, infrastruktur jaringan internal perusahaan secara menyeluruh, maupun analisis terhadap kode sumber sistem. Penentuan objek dan ruang lingkup tersebut dilakukan untuk memastikan proses pengujian keamanan dapat dilakukan secara terarah dan sesuai dengan tujuan penelitian.



Gbr. 2 Dashboard ERP

Berdasarkan hasil observasi terhadap sistem ERP yang digunakan oleh PT XYZ menggunakan akun pengguna yang disediakan oleh pihak perusahaan, sistem menyediakan berbagai fungsi operasional yang dapat diakses setelah proses autentikasi berhasil dilakukan. Identifikasi fungsionalitas dalam penelitian ini dibatasi pada fitur yang dapat diamati secara langsung sesuai dengan pendekatan *grey box testing*.

Secara umum, sistem mengharuskan pengguna melakukan proses login sebelum dapat mengakses modul utama aplikasi. Setelah proses autentikasi berhasil, pengguna dapat mengakses berbagai fungsi yang mendukung operasional perusahaan. Fungsionalitas sistem yang teridentifikasi selanjutnya digambarkan dalam diagram *use case*.



Gbr. 3 Use Case Sistem ERP

ISSN (Online) 2747-0563
Volume 6 Tahun 2026

Berdasarkan hasil observasi, sistem ERP yang digunakan oleh PT XYZ menyediakan berbagai fungsi operasional yang dapat diakses setelah proses autentikasi pengguna berhasil dilakukan. Fungsionalitas utama yang teridentifikasi meliputi mekanisme login sebagai proses autentikasi pengguna, pengelolaan data pengguna, serta pengelolaan master data yang mencakup data item, business partner, lokasi penyimpanan, akun akuntansi, dan data pendukung lainnya.

Selain itu, sistem juga menyediakan modul operasional utama yang meliputi transaksi penjualan, transaksi pembelian, manajemen persediaan, produksi, serta keuangan dan akuntansi. Sistem juga dilengkapi dengan fitur dashboard dan laporan operasional yang digunakan untuk memantau aktivitas bisnis dan kinerja perusahaan.

Secara keseluruhan, sistem ERP yang diuji memiliki fungsionalitas yang terintegrasi dan mencakup berbagai proses bisnis perusahaan. Seluruh fungsi tersebut hanya dapat diakses setelah proses autentikasi berhasil dilakukan, sehingga mekanisme login menjadi komponen penting dalam pengendalian akses sistem.

B. Intelligence Gathering

Tahap intelligence gathering merupakan fase awal dalam proses pengujian keamanan berdasarkan metodologi Penetration Testing Execution Standard. Pada tahap ini dilakukan proses pengumpulan informasi terhadap sistem yang menjadi objek penelitian guna memahami struktur aplikasi, teknologi yang digunakan, serta potensi attack surface yang tersedia[6].

Dalam penelitian ini, proses pengumpulan informasi dilakukan menggunakan pendekatan grey box testing, di mana peneliti memperoleh satu akun pengguna resmi tanpa akses terhadap dokumentasi sistem, struktur basis data, maupun kode sumber aplikasi. Oleh karena itu, aktivitas intelligence gathering difokuskan pada observasi terhadap fitur aplikasi, identifikasi teknologi yang digunakan, serta pemetaan endpoint dan konfigurasi sistem berdasarkan akses yang tersedia.

Tahap ini menjadi dasar penting sebelum dilakukan analisis kerentanan, karena informasi yang diperoleh akan menentukan efektivitas proses identifikasi celah keamanan pada tahap berikutnya. Untuk mendukung proses tersebut digunakan beberapa alat bantu yang berfungsi untuk mengidentifikasi teknologi, layanan jaringan, serta struktur direktori pada sistem ERP milik PT XYZ.

TABEL 1
PERENCANAN ALAT BANTU INTELLIGENCE GATHERING

Alat Bantu	Keterangan	Hasil yang Diharapkan
Wappalyzer	Digunakan melalui ekstensi peramban untuk mengidentifikasi teknologi yang digunakan pada aplikasi web, seperti framework, library JavaScript, web server, dan layanan pihak ketiga[7].	Menampilkan informasi teknologi dan komponen perangkat lunak yang digunakan oleh sistem, termasuk indikasi penggunaan versi yang berpotensi memiliki kerentanan.
Nmap	Digunakan untuk melakukan pemindaian jaringan guna mengidentifikasi port terbuka, layanan yang berjalan, serta versi layanan pada server target[8].	Menghasilkan informasi mengenai port terbuka, layanan jaringan, sistem operasi yang digunakan, serta kemungkinan indikasi kerentanan berdasarkan skrip pemindaian.
Gobuster	Digunakan untuk melakukan enumerasi direktori dan file tersembunyi pada server web dengan memanfaatkan *wordlist*[9].	Menghasilkan daftar direktori atau file yang dapat diakses pada server, termasuk halaman tersembunyi yang tidak terindeks.
ParamSpider	Digunakan untuk mengekstraksi parameter URL yang terdapat pada domain target dari berbagai sumber arsip web[10].	Menghasilkan daftar endpoint dan parameter aplikasi yang dapat digunakan sebagai titik masuk pengujian kerentanan seperti injeksi.
Nuclei	Digunakan untuk melakukan pemindaian berbasis template terhadap target guna mendeteksi potensi kerentanan yang telah diketahui[11].	Menghasilkan daftar potensi kerentanan yang dikategorikan berdasarkan tingkat keparahan serta referensi kerentanan yang relevan.

Berdasarkan perencanaan tersebut, proses pengumpulan informasi dilakukan dengan mengombinasikan pendekatan *passive reconnaissance* dan *active scanning*. Pendekatan ini bertujuan untuk memperoleh gambaran yang lebih komprehensif terhadap struktur sistem dan potensi permukaan serangan pada aplikasi ERP milik PT XYZ.

Secara umum, proses *intelligence gathering* difokuskan pada dua aktivitas utama. Pertama, pemetaan struktur aplikasi melalui proses enumerasi direktori, file, serta parameter yang digunakan oleh sistem. Aktivitas ini bertujuan untuk mengidentifikasi halaman tersembunyi, endpoint API, maupun file konfigurasi yang berpotensi menjadi titik masuk serangan. Kedua, identifikasi teknologi dan layanan jaringan yang digunakan oleh sistem, sehingga penguji dapat mengetahui komponen perangkat lunak yang berjalan pada server serta potensi kerentanan yang berkaitan dengan teknologi tersebut.

Seluruh informasi yang diperoleh pada tahap ini menjadi dasar bagi proses pengujian keamanan pada tahap berikutnya, yaitu pemindaian kerentanan menggunakan OWASP ZAP. Data berupa daftar URL, parameter, serta layanan yang teridentifikasi akan digunakan sebagai input dalam proses pemindaian lanjutan untuk memverifikasi secara lebih mendalam potensi kerentanan yang terdapat pada sistem.

C. Threat Modelling

Tahap *threat modeling* merupakan proses identifikasi dan pemetaan potensi ancaman terhadap sistem yang menjadi objek penelitian setelah tahap *intelligence gathering* selesai dilakukan[12]. Pada tahap ini, fokus penelitian tidak lagi hanya pada pengumpulan informasi teknis, tetapi pada analisis terhadap kemungkinan eksploitasi kerentanan serta dampaknya terhadap aset sistem. Dalam metodologi Penetration Testing

ISSN (Online) 2747-0563
Volume 6 Tahun 2026

Execution Standard, tahap ini berfungsi sebagai jembatan antara proses pengumpulan informasi dan pengujian teknis untuk memahami potensi risiko keamanan secara lebih terstruktur.

Pada penelitian ini, proses *threat modeling* didukung oleh pemindaian keamanan menggunakan OWASP ZAP, yaitu alat pengujian keamanan aplikasi web berbasis pendekatan *Dynamic Application Security Testing* (DAST). Pendekatan ini bekerja dengan menguji aplikasi yang sedang berjalan melalui pengiriman permintaan HTTP yang dimodifikasi, kemudian menganalisis respons sistem untuk mendeteksi indikasi kerentanan.

Proses pemindaian dilakukan dalam dua tahap utama. Tahap pertama adalah *spider* atau *crawling* otomatis yang bertujuan untuk memetakan struktur aplikasi, termasuk tautan, formulir, parameter, dan endpoint yang tersedia. Hasil dari proses ini berupa daftar URL dan parameter yang menggambarkan *attack surface* dari sistem ERP milik PT XYZ. Tahap kedua adalah *active scanning*, yaitu proses pengujian aktif terhadap endpoint yang telah teridentifikasi dengan berbagai payload untuk mendeteksi potensi kerentanan seperti *SQL injection*, *cross-site scripting* (XSS), dan kelemahan konfigurasi keamanan.

Hasil dari tahap ini digunakan untuk memetakan potensi ancaman serta memberikan gambaran awal mengenai risiko keamanan pada sistem. Dengan mengombinasikan hasil pemindaian teknis dan analisis konseptual, proses *threat modeling* menghasilkan pemahaman yang lebih komprehensif terhadap potensi kerentanan serta menjadi dasar dalam penyusunan strategi mitigasi pada tahap analisis selanjutnya. Gbr. 4 memperlihatkan contoh keluaran pemindaian dari OWASP ZAP yang menghasilkan 22 temuan kerentanan.

Gbr. 4 Contoh Hasil OWASP Zap

Seluruh temuan tersebut dikelompokkan berdasarkan tingkat keparahan (severity) dan tingkat kepercayaan (confidence) untuk memudahkan proses analisis dan penentuan prioritas penanganan. Ringkasan dan klasifikasi temuan yang ditampilkan pada Gbr. 4.

D. Vulnerability Analysis

Tahap *vulnerability analysis* merupakan fase dalam metodologi Penetration Testing Execution Standard yang bertujuan untuk mengidentifikasi dan menganalisis kerentanan keamanan pada sistem ERP milik PT XYZ[13]. Proses ini dilakukan berdasarkan hasil tahap *intelligence gathering* dan *threat modeling*, sehingga pengujian dapat difokuskan pada bagian sistem yang memiliki potensi risiko lebih tinggi.

Pengujian dilakukan dengan pendekatan *grey-box testing*, di mana penguji memiliki akses terbatas sebagai pengguna sistem tanpa dokumentasi internal maupun kode sumber aplikasi. Analisis difokuskan pada aspek autentikasi, validasi input, manajemen sesi, serta konfigurasi keamanan aplikasi web. Proses identifikasi kerentanan dilakukan melalui pemindaian menggunakan OWASP ZAP, kemudian setiap temuan dianalisis dan diklasifikasikan berdasarkan standar Common Weakness Enumeration (CWE). Klasifikasi ini digunakan untuk mengelompokkan jenis kelemahan keamanan secara sistematis serta mengevaluasi potensi dampaknya terhadap aspek *confidentiality*, *integrity*, dan *availability* (CIA) pada sistem yang diuji[14].

E. Exploitation

Tahap *exploitation* merupakan fase dalam metodologi Penetration Testing Execution Standard yang bertujuan untuk memverifikasi apakah kerentanan yang telah diidentifikasi pada tahap *vulnerability analysis* benar-benar dapat dieksploitasi[15]. Tahap ini dilakukan untuk memvalidasi tingkat keparahan kerentanan serta memahami potensi dampaknya terhadap keamanan sistem ERP milik PT XYZ.

Berdasarkan hasil pemindaian menggunakan OWASP ZAP, setiap temuan kerentanan yang telah diklasifikasikan menggunakan Common Weakness Enumeration (CWE) dianalisis lebih lanjut untuk mengidentifikasi kemungkinan skenario serangan yang dapat terjadi. Proses eksploitasi dilakukan secara terbatas dalam bentuk *proof of concept* untuk menunjukkan potensi pemanfaatan celah keamanan tanpa melakukan perubahan data, kerusakan sistem, maupun gangguan terhadap ketersediaan layanan aplikasi ERP yang diuji.

F. Post Exploitation

Tahap post-exploitation merupakan fase evaluasi lanjutan dalam metodologi Penetration Testing Execution Standard yang bertujuan untuk menganalisis secara komprehensif kerentanan yang telah teridentifikasi dan tervalidasi pada sistem ERP milik PT XYZ[16]. Evaluasi dilakukan terhadap temuan yang diperoleh melalui pemindaian menggunakan OWASP ZAP serta pengujian eksploitasi terbatas.

Setiap kerentanan kemudian diklasifikasikan berdasarkan Common Weakness Enumeration (CWE) untuk memastikan pengelompokan kelemahan mengikuti standar internasional. Pendekatan ini memungkinkan analisis kerentanan dilakukan secara sistematis dengan mempertimbangkan potensi dampaknya terhadap keamanan dan operasional sistem ERP.

G. Reporting

Tahap reporting merupakan fase akhir dalam metodologi Penetration Testing Execution Standard yang berfokus pada

ISSN (Online) 2747-0563
Volume 6 Tahun 2026

penyusunan laporan hasil pengujian keamanan secara sistematis[17]. Laporan ini bertujuan menyampaikan temuan kerentanan pada sistem ERP milik PT XYZ secara jelas kepada pihak teknis maupun manajemen.
Pada tahap ini dilakukan pengumpulan dan validasi seluruh bukti pengujian, termasuk hasil pemindaian menggunakan OWASP ZAP serta bukti eksploitasi yang diperoleh selama proses pengujian. Setiap temuan diverifikasi untuk memastikan kerentanan yang dilaporkan merupakan temuan yang valid dan bukan false positive.
Laporan yang disusun memuat executive summary yang memberikan gambaran umum kondisi keamanan sistem, termasuk rekapitulasi tingkat kerentanan seperti critical, high, medium, dan low. Selain itu, disajikan pula penjelasan teknis setiap temuan beserta proof of concept (PoC) yang menunjukkan mekanisme eksploitasi kerentanan secara terstruktur.
Untuk menilai dampak keamanan, setiap kerentanan dianalisis berdasarkan konsep CIA Triad yang mencakup aspek confidentiality, integrity, dan availability.
Sebagai tahap akhir, laporan dilengkapi dengan rekomendasi mitigasi berbasis prioritas risiko yang dapat diterapkan oleh tim teknis PT XYZ untuk meningkatkan keamanan sistem ERP. Selain itu, dilakukan pengujian ulang terbatas terhadap salah satu kerentanan yang telah diperbaiki guna memverifikasi efektivitas mitigasi yang diterapkan.

III. HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil pengujian keamanan yang dilakukan terhadap sistem ERP milik PT XYZ berdasarkan metodologi Penetration Testing Execution Standard. Proses pengujian dilakukan secara bertahap mulai dari pre-engagement interaction, intelligence gathering, threat modeling, vulnerability analysis, exploitation hingga tahap post-exploitation dan reporting.
Hasil pengujian kemudian dianalisis dengan mengacu pada temuan OWASP ZAP untuk mengidentifikasi kategori risiko keamanan aplikasi web yang ditemukan pada sistem ERP yang diuji. Setiap temuan kerentanan selanjutnya diklasifikasikan menggunakan Common Weakness Enumeration.
Melalui tahapan tersebut, penelitian ini tidak hanya mengidentifikasi kerentanan yang terdapat pada sistem, tetapi juga menganalisis potensi dampak serta memberikan rekomendasi mitigasi yang dapat diterapkan untuk meningkatkan keamanan sistem ERP PT XYZ.

A. *Pre-Engagement Interaction*

Seluruh Tahap Pre-Engagement Interaction merupakan fase awal dalam metodologi Penetration Testing Execution Standard (PTES) yang bertujuan untuk menetapkan kesepakatan antara tim penguji keamanan dengan organisasi yang menjadi objek penelitian. Pada tahap ini dilakukan proses komunikasi awal guna menentukan ruang lingkup pengujian, batasan pengujian, serta aturan pelaksanaan penetration testing agar aktivitas pengujian tidak menimbulkan gangguan terhadap operasional sistem yang sedang berjalan.

Dalam penelitian ini, proses pre-engagement dilakukan melalui koordinasi dengan tim pengelola sistem di PT XYZ untuk memperoleh izin resmi dalam melakukan pengujian keamanan terhadap sistem ERP yang digunakan oleh perusahaan. Diskusi awal dilakukan untuk menentukan aset sistem yang akan diuji, jenis pengujian yang diperbolehkan, serta jadwal pelaksanaan pengujian. Selain itu, tahap ini juga mencakup penyusunan rules of engagement yang berfungsi sebagai pedoman selama proses penetration testing berlangsung.
Hasil dari tahap ini berupa dokumen kesepakatan yang mencakup informasi target pengujian, metode pengujian yang digunakan, serta batasan aktivitas yang diperbolehkan selama pengujian keamanan dilakukan. Penentuan ruang lingkup pengujian sangat penting untuk memastikan bahwa seluruh aktivitas pengujian dilakukan secara etis, terkontrol, dan tidak melanggar kebijakan keamanan organisasi.
Ruang lingkup pengujian yang telah disepakati pada penelitian ini ditunjukkan pada Tabel berikut.

TABEL 2
INFORMASI TARGET PENGUJIAN

Informasi	Keterangan
Sistem yang diuji	Sistem ERP XYZ
Domain aplikasi	app.wesk****.id
IP Address Server	66.42.5*.**
Lokasi Server	Singapura
Jenis pengujian	Web Application Penetration Testing
Metodologi pengujian	Penetration Testing Execution Standard (PTES)
Tim penguji	Peneliti
Tanggal pengujian	18 Februari 2026

Selain penentuan target sistem, tahap ini juga menetapkan batasan pengujian yang bertujuan untuk mencegah aktivitas yang berpotensi mengganggu layanan sistem perusahaan. Batasan tersebut disepakati bersama antara peneliti dan pihak organisasi sebelum proses penetration testing dilakukan.

TABEL 3
RULES OF ENGAGEMENT PENGUJIAN

Aturan Pengujian	Deskripsi
Ruang lingkup pengujian	Pengujian hanya dilakukan pada alamat IP ERP yang telah disepakati
Larangan Pengambilan Hak Akses	Pengujian tidak diperbolehkan melakukan pengambilan hak akses yang menyebabkan perubahan konfigurasi dan memengaruhi kinerja sistem ERP
Waktu pengujian	Pengujian dilakukan di luar jam operasional utama

ISSN (Online) 2747-0563
Volume 6 Tahun 2026

Akses sistem	Pengujian dilakukan sebagai pengguna eksternal tanpa hak akses administratif
Hasil Pengujian	Peneliti memberikan hasil laporan temuan pengujian serta rekomendasi terkait pelaksanaan pentest

Berdasarkan hasil kesepakatan pada tahap pre-engagement ini, proses penetration testing selanjutnya dapat dilakukan secara terstruktur sesuai dengan metodologi PTES yang meliputi tahapan pengumpulan informasi, analisis kerentanan, eksploitasi, hingga penyusunan laporan hasil pengujian.

B. *Intelligence Gathering*

Pada penelitian ini, proses intelligence gathering dilakukan terhadap sistem ERP milik PT XYZ dengan pendekatan pengumpulan informasi secara pasif dan aktif. Pengumpulan informasi pasif dilakukan tanpa berinteraksi langsung secara signifikan dengan sistem target, seperti identifikasi domain, alamat IP, serta teknologi yang digunakan oleh aplikasi web. Sementara itu, pengumpulan informasi aktif dilakukan melalui proses pemindaian terhadap sistem target untuk mengidentifikasi layanan yang berjalan, struktur direktori aplikasi, serta endpoint yang dapat diakses.
Beberapa alat bantu digunakan untuk mendukung proses pengumpulan informasi ini, di antaranya adalah Nmap untuk melakukan pemindaian jaringan dan identifikasi layanan yang berjalan pada server, Wappalyzer untuk mengidentifikasi teknologi web yang digunakan oleh aplikasi, serta Gobuster untuk menemukan direktori atau endpoint tersembunyi yang dapat menjadi bagian dari permukaan serangan (attack surface).

1. Identifikasi Teknologi Aplikasi menggunakan Wappalyzer

TABEL 4
IDENTIFIKASI TEKNOLOGI WEBSITE

Kategori Teknologi	**Nama Teknologi**
Web Server	Apache 2.4.6
Bahasa Pemrograman	PHP 5.4.16
Tech	HTML 5
Database	MariaDB 5.5.64
JavaScript Library	jQuery 3.1.1
Framework Frontend	Bootstrap

2. Pemindaian Infrastruktur Jaringan menggunakan Nmap

TABEL 5
DAFTAR PORT WEBSITE

Port	***Status***	***Service***	**Keterangan**
80	*Open*	HTTP	Web server aktif
443	*Open*	HTTPS	Akses aplikasi ERP
21	*Closed*	FTP	Jalur transfer file antara client dan server
22	*Open*	SSH	Akses administrasi server
3306	*Open*	MySQL	Port database tidak terbuka publik
465	*Close*	SMTPS	Pengiriman email menggunakan enkripsi SSL/TLS.

3. Enumerasi Parameter Aplikasi menggunakan ParamSpider

TABEL 6
HASIL ENUMERASI PARAMSPIDER

Endpoint	**Parameter**	**Metode**
/login	username	POST
/login	password	POST
/report	id	GET
/user	user_id	GET
/search	keyword	GET

4. Enumerasi Direktori menggunakan Gobuster

TABEL 7
HASIL ENUMERASI GOBUSTER

Direktori	**Status HTTP**
/application	301
/asset	301
/cgi-bin	403
/data	301
/index.php	200
/pos	301
/system	301
/.hta	403
/.htaccess	403
/.htpasswd	403

5. Identifikasi Kerentanan Awal menggunakan Nuclei

TABEL 8
HASIL IDENTIFIKASI NUCLEI

Jenis Temuan	***Severity***	**Deskripsi**
Missing Security Header	*Medium*	Header keamanan HTTP tidak lengkap
Server Version Disclosure	*Low*	Informasi versi server terekspos
Cookie Without Secure Flag	*Medium*	Cookie tidak menggunakan atribut Secure

ISSN (Online) 2747-0563
Volume 6 Tahun 2026

Directory Listing Enabled	*Low*	Direktori dapat diakses publik

Berdasarkan hasil proses intelligence gathering, diperoleh berbagai informasi teknis mengenai infrastruktur sistem, teknologi aplikasi, serta endpoint yang tersedia pada sistem ERP. Informasi tersebut memberikan gambaran mengenai permukaan serangan (attack surface) aplikasi yang kemudian digunakan sebagai dasar dalam proses threat modeling untuk menganalisis potensi ancaman serta menentukan strategi pengujian keamanan yang lebih terarah.

C. *Threat Modelling*

Pada penelitian ini, proses threat modeling dilakukan dengan memanfaatkan alat pengujian keamanan aplikasi web yaitu OWASP ZAP. Alat ini menggunakan pendekatan Dynamic Application Security Testing (DAST) untuk mengidentifikasi potensi kerentanan pada aplikasi web yang sedang berjalan melalui analisis terhadap permintaan dan respons HTTP.

Proses pemindaian dilakukan melalui dua tahapan utama, yaitu spider (crawling) dan active scanning. Tahap spider bertujuan untuk memetakan struktur aplikasi dengan mengidentifikasi halaman web, tautan, parameter, serta formulir yang tersedia pada sistem ERP.

Setelah proses pemetaan struktur aplikasi selesai dilakukan, tahap selanjutnya adalah active scanning, yaitu proses pengujian aktif terhadap endpoint yang telah ditemukan dengan menggunakan berbagai payload pengujian. Proses ini bertujuan untuk mendeteksi indikasi kerentanan seperti SQL Injection, Cross-Site Scripting (XSS), serta kelemahan konfigurasi keamanan pada server maupun aplikasi.

TABEL 9
HASIL ACTIVE SCANNING ZAP

Jenis Kerentanan	***Risk Level***	***Confidence***	**Jumlah**
Content Security Policy (CSP) Header Not Set	*Medium*	*High*	1
Absence of Anti-CSRF Tokens	*Medium*	*Low*	1
Server Leaks Version Information via "Server" HTTP Response Header	*Low*	*High*	1
Cookie No HttpOnly Flag	*Low*	*Medium*	1
X-Content-Type-Options Header Missing	*Low*	*Medium*	5
Retrieved from Cache	*Informational*	*Medium*	2
Information Disclosure – Suspicious Comments	*Informational*	*Low*	1
Information Disclosure – Sensitive Information in URL	*Informational*	*Medium*	2

selanjutnya adalah memetakan setiap kerentanan terhadap potensi ancaman yang dapat ditimbulkannya. Proses ini bertujuan untuk mengubah temuan teknis menjadi gambaran risiko yang lebih kontekstual terhadap sistem. Melalui pemetaan tersebut, setiap kerentanan dianalisis berdasarkan kemungkinan skenario serangan serta dampaknya terhadap kerahasiaan, integritas, dan ketersediaan data. Oleh karena itu, tabel Pemetaan Kerentanan Hasil dari OWASP ZAP disusun untuk menghubungkan hasil identifikasi awal dengan tahap analisis risiko yang lebih komprehensif.

TABEL 10
PEMETAAN KERENTANAN

Komponen Sistem	**Temuan pada Report**	**Risk**	**Potensi Ancaman**
Mekanisme Header Keamanan	*CSP Header Not Set*	*Medium*	Peningkatan risiko eksekusi skrip tidak sah
Mekanisme Form & Request	*Absence of Anti-CSRF Tokens*	*Medium*	Permintaan tidak sah atas nama pengguna
Web Server	*Server Header Leak*	*Low*	Pengungkapan informasi versi server
Manajemen Cookie	*Cookie No HttpOnly Flag*	*Low*	Akses cookie melalui skrip sisi klien
HTTP Response Security	*X-Content-Type-Options Header Missing*	*Low*	Potensi kesalahan interpretasi MIME type

Penentuan tujuh kategori pada Tabel tersebut dilakukan melalui proses klasifikasi seluruh alert yang dihasilkan dari pemindaian menggunakan OWASP ZAP. Laporan pemindaian menunjukkan total 21 alert, namun beberapa di antaranya merupakan kemunculan berulang dari jenis kerentanan yang sama pada endpoint yang berbeda. Oleh karena itu, dilakukan pengelompokan berdasarkan nama kerentanan (alert name) untuk menghindari redundansi, sehingga diperoleh tujuh kategori kerentanan unik yang selanjutnya dianalisis dalam konteks potensi ancaman.

D. *Vulnerability Analysis*

Pada tahap ini, pengujian difokuskan pada identifikasi kelemahan autentikasi, validasi input, manajemen sesi, dan konfigurasi keamanan aplikasi web. Proses vulnerability analysis dilakukan dengan pendekatan grey-box, di mana pengujian dilakukan melalui antarmuka aplikasi dengan informasi terbatas seperti yang dapat diakses pengguna umum. Temuan dari pemindaian menggunakan OWASP ZAP kemudian dianalisis dan diklasifikasikan berdasarkan Common Weakness Enumeration (CWE) untuk mengelompokkan jenis kelemahan, mengurangi duplikasi temuan, serta mengidentifikasi pola kerentanan yang dominan. Setiap CWE selanjutnya dianalisis berdasarkan konteks kemunculan dan

ISSN (Online) 2747-0563
Volume 6 Tahun 2026

potensi dampaknya terhadap aspek confidentiality, integrity, dan availability (CIA).

TABEL 11
PEMETAAN HASIL ZAP TERHADAP CWE

CWE	**Nama CWE**	**Risk (ZAP)**	**Jumlah Temuan**	**Deskripsi Singkat**
CWE-352	*Cross-Site Request Forgery*	*Medium*	1	Tidak adanya validasi token pada permintaan sensitif.
CWE-693	*Protection Mechanism Failure*	*Medium*	2	Mekanisme perlindungan tidak diterapkan atau tidak efektif.
CWE-264	*Permissions, Privileges, and Access Controls*	*Medium*	1	Pengaturan hak akses tidak memadai.
CWE-1021	*Improper Restriction of Rendered UI Layers*	*Low*	1	UI dapat dimanipulasi atau dilapisi konten lain.
CWE-345	*Insufficient Verification of Data Authenticity*	*Medium*	2	Data yang diterima tidak diverifikasi keasliannya.
CWE-1395	*Dependency on Vulnerable Third-Party Component*	*Low*	1	Ketergantungan pada library/komponen rentan.
CWE-1004	*Sensitive Cookie Without HttpOnly Flag*	*Low*	1	Cookie tidak diberi atribut HttpOnly.
CWE-614	*Sensitive Cookie in HTTPS Session Without Secure Flag*	*Low*	1	Cookie tidak diberi atribut Secure.
CWE-1275	*Sensitive Cookie With Improper SameSite Attribute*	*Low*	1	Cookie tidak dikonfigurasi dengan SameSite yang tepat.
CWE-829	*Inclusion of Functionality from Untrusted Control Sphere*	*Medium*	1	Memuat script dari sumber tidak terpercaya.
CWE-497	*Exposure of System Data*	*Medium*	2	Informasi sistem terekspos dalam response.
CWE-319	*Cleartext Transmission of Sensitive Information*	*Medium*	1	Data dikirim tanpa enkripsi.
CWE-598	*Information Exposure Through Query String*	*Medium*	1	Data sensitif dikirim via URL parameter.
CWE-615	*Information Exposure Through Comments*	*Low*	1	Informasi sensitif terdapat pada komentar HTML/JS.
CWE-525	*Information Exposure Through Browser Caching*	*Low*	1	Konten sensitif dapat tersimpan di cache browser.

Kerentanan yang ditemukan pada tahap ini selanjutnya akan diuji pada tahap exploitation untuk membuktikan apakah kelemahan tersebut benar-benar dapat dimanfaatkan. Dengan demikian, tahap vulnerability analysis berperan penting dalam memastikan bahwa pengujian keamanan dilakukan secara sistematis, terarah, dan sesuai dengan tujuan penelitian.

E. *Exploitation*

Setelah dilakukan identifikasi dan klasifikasi kelemahan berdasarkan daftar Common Weakness Enumeration (CWE) yang diperoleh dari hasil pemindaian OWASP ZAP, tahap selanjutnya adalah menganalisis relevansi setiap kelemahan terhadap pola dan strategi serangan yang umum digunakan oleh adversary.

Seluruh proses eksploitasi dilakukan secara terkontrol untuk memastikan bahwa pengujian tidak mengganggu operasional sistem ERP yang menjadi objek penelitian. Hasil pengujian kemudian dirangkum dalam satu tabel berdasarkan jenis kerentanan yang telah diidentifikasi sebelumnya, sehingga tidak terjadi duplikasi temuan.

TABEL 12
EKSPLOITASI KERENTANAN

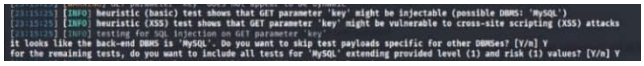
Jenis Kerentanan	**Endpoint Terdampak**	**Metode Pengujian**	**Payload / Teknik**
SQL Injection	app.wesk***.id/index.php/welc	*SQLmap*	sqlmap -u 'https://app.wesk***.id/index.php/wel

ISSN (Online) 2747-0563
Volume 6 Tahun 2026

	ome/search/?key=test&search='		come/search/?key=test&search='\--cookie="PHPSESSID=5ohd50monh152p9njpktebdju1" --batch --dbs
Cross-Site Scripting (XSS)	app.wesk***.id/index.php/bp/detail/26 (kolom input)	*Input injection*	<script>alert('XSS terdeteksi');</script>
Brute Force Login Page	app.weskonek.id/ (form login)	*Burpsuite Intruder Attack*	Intruder Attack & Wordlist.txt
Insufficient Session Expiration	https://app.wesk***.id/	*Inspect Element*	Inspect Element (PHPSESSID)

1. *SQL Injection*

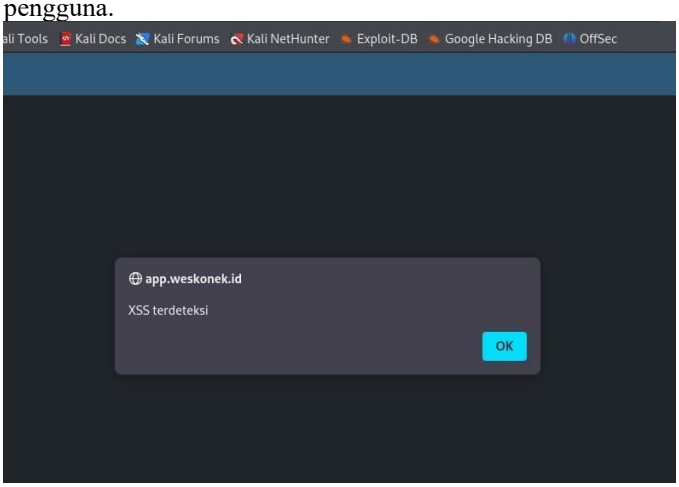
Kerentanan SQL Injection pada sistem ditemukan melalui pengujian terhadap salah satu kolom input pada fitur pencarian. Pengujian diawali dengan memasukkan data uji ke dalam kolom input tersebut, kemudian URL hasil request yang dihasilkan diambil untuk dianalisis lebih lanjut. URL tersebut selanjutnya digunakan sebagai target pada tools sqlmap untuk mengidentifikasi adanya celah injeksi pada query database. Pengujian dilakukan dengan menyertakan parameter URL yang mengandung indikasi manipulasi input, serta menambahkan cookie sesi yang valid agar permintaan dianggap sebagai pengguna yang sah oleh sistem. Dengan menggunakan perintah pengujian otomatis dari sqlmap, sistem terbukti rentan terhadap SQL Injection, yang memungkinkan penyerang untuk mengekstrak informasi dari database, seperti daftar basis data yang tersedia, tanpa melalui mekanisme autentikasi yang semestinya.



Gbr. 5 bukti kerentanan *SQL Injection*

2. *Cross-Site Scripting (XSS)*

Kerentanan Cross-Site Scripting (XSS) ditemukan melalui pengujian pada kolom input yang tersedia di dalam sistem. Pengujian dilakukan dengan memasukkan payload berupa skrip sederhana, yaitu <script>alert('XSS terdeteksi');</script>, ke dalam field input tersebut. Setelah data dikirim dan diproses oleh aplikasi, sistem tidak melakukan validasi maupun sanitasi input dengan baik, sehingga skrip yang dimasukkan dieksekusi langsung oleh browser. Hal ini ditandai dengan munculnya pop-up alert pada halaman aplikasi, yang menunjukkan bahwa kode JavaScript berhasil dijalankan. Temuan ini mengindikasikan bahwa aplikasi rentan terhadap serangan XSS, yang dapat dimanfaatkan oleh penyerang untuk menyisipkan skrip berbahaya, mencuri data sensitif pengguna, atau melakukan manipulasi tampilan halaman tanpa sepengetahuan pengguna.



Gbr. 6 bukti kerentanan XSS

3. *Brute Force Login Page*

Kerentanan Brute Force Login ditemukan melalui pengujian terhadap mekanisme autentikasi pada halaman login aplikasi. Pengujian dilakukan dengan memanfaatkan fitur Intruder Attack pada tools Burp Suite, di mana penyerang dapat mengotomatisasi percobaan login menggunakan berbagai kombinasi username dan password. Dalam proses ini, digunakan sebuah wordlist (misalnya Wordlist.txt) yang berisi daftar kemungkinan kredensial untuk diuji secara berulang. Hasil pengujian menunjukkan bahwa sistem tidak memiliki mekanisme pembatasan percobaan login, seperti rate limiting, account lockout, atau CAPTCHA, sehingga memungkinkan percobaan login dilakukan secara terus-menerus tanpa hambatan. Kondisi ini meningkatkan risiko keberhasilan serangan brute force, di mana penyerang dapat memperoleh akses tidak sah ke akun pengguna dengan menebak kredensial yang benar.

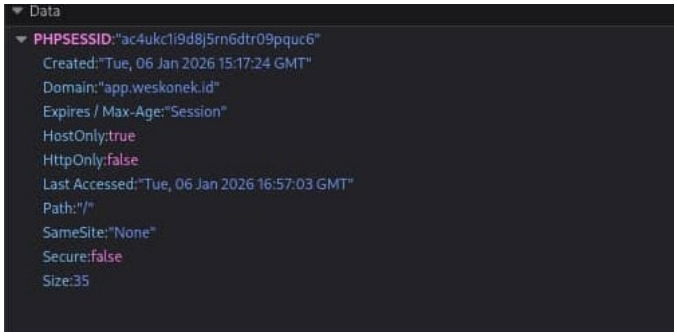
Gbr. 7 Bukti kerentanan Brute Force

4. *Insufficient Session Expiration*

Kerentanan Insufficient Session Expiration ditemukan pada mekanisme pengelolaan sesi pengguna setelah proses autentikasi. Pengujian dilakukan dengan mengamati perilaku sesi login menggunakan tools Burp Suite, khususnya pada pengelolaan cookie seperti PHPSESSID. Hasil pengujian menunjukkan bahwa sesi pengguna tetap aktif dalam jangka waktu yang lama meskipun tidak ada aktivitas (idle), dan tidak terdapat mekanisme automatic session timeout yang memadai.

ISSN (Online) 2747-0563
Volume 6 Tahun 2026

Selain itu, sesi juga tetap valid meskipun pengguna tidak melakukan logout secara eksplisit.



Gbr. 8 bukti kerentanan *Insufficient Session Expiration*

F. *Post Exploitation*

Berdasarkan hasil tahap *exploitation*, beberapa kerentanan yang berhasil divalidasi menunjukkan potensi dampak yang signifikan terhadap aspek keamanan sistem, khususnya pada confidentiality, integrity, dan availability (CIA).

TABEL 13
PEMETAAN HASIL EKSPLOITASI

Jenis Kerentanan	Dampak (CIA)	Potensi Serangan Lanjutan	Risiko
SQL Injection	*Confidentiality, Integrity*	*Data exfiltration, data manipulation, privilege escalation*	*High*
Cross-Site Scripting (XSS)	*Confidentiality, Integrity*	*Session hijacking, user impersonation, phishing berbasis web*	*Medium*
Brute Force Login	*Confidentiality, Integrity*	*Account takeover, privilege escalation*	*High*
Insufficient Session Expiration	*Confidentiality, Integrity, Availability*	*Session hijacking, authentication bypass*	*High*

1. SQL Injection

Masalah SQL Injection ini terjadi karena input dari pengguna langsung dimasukkan ke query database tanpa dicek atau diamankan dulu. disarankan menggunakan prepared statements atau parameterized queries agar input pengguna tidak langsung dieksekusi dalam query SQL. Selain itu, input pengguna harus divalidasi dan disanitasi dengan benar, serta hindari penggunaan query dinamis. Kerentanan ini tergolong kritis karena memungkinkan penyerang membaca atau memodifikasi data dalam database, serta berpotensi mengekspos informasi sensitif pengguna. Oleh karena itu, perbaikan harus dilakukan sesegera mungkin untuk menjaga integritas dan keamanan sistem.

Temuan ini sejalan dengan penelitian Maulana dkk. (2025) yang melakukan penetration testing menggunakan OWASP ZAP dan menemukan bahwa SQL Injection masih menjadi salah satu kerentanan dengan dampak paling besar terhadap keamanan aplikasi web karena memungkinkan akses langsung ke basis data[18]. Penelitian Neupane (2025) juga menunjukkan bahwa penerapan prepared statements, parameterized queries, serta validasi input merupakan mekanisme mitigasi yang paling efektif untuk mencegah eksploitasi SQL Injection. Berdasarkan penelitian Neupane (2025), hasil kerentanan SQL Injection berhasil divalidasi menggunakan SQLMap sehingga ditempatkan sebagai prioritas mitigasi tertinggi karena berpotensi mengganggu aspek confidentiality dan integrity data perusahaan[19].

2. Cross-Site Scripting (XSS)

Kerentanan Cross-Site Scripting (XSS) terjadi akibat tidak adanya validasi dan sanitasi input pengguna secara memadai, sehingga memungkinkan penyerang menyisipkan skrip berbahaya ke dalam aplikasi web. Skrip tersebut dapat dieksekusi pada browser pengguna lain dan berpotensi menyebabkan pencurian data sensitif seperti cookie dan session, pengambilalihan akun, serta manipulasi tampilan halaman web. Untuk mengatasi kerentanan ini, pengembang perlu menerapkan validasi input yang ketat, melakukan encoding atau escaping terhadap output sebelum ditampilkan ke halaman web, serta memanfaatkan mekanisme keamanan tambahan seperti Content Security Policy (CSP) dan fitur proteksi bawaan dari framework yang digunakan guna mencegah eksekusi skrip berbahaya.

Hasil tersebut konsisten dengan penelitian Hannousse dkk. (2022) yang menjelaskan bahwa Cross-Site Scripting masih menjadi salah satu kerentanan aplikasi web yang paling banyak ditemukan meskipun berbagai mekanisme mitigasi telah tersedia[20]. Penelitian Nelmiawati dan Dealova (2025) juga menunjukkan bahwa XSS masih termasuk kerentanan yang dominan pada aplikasi web modern sehingga diperlukan penerapan input validation, output encoding, dan Content Security Policy (CSP) secara bersamaan untuk mengurangi risiko eksploitasi[21]. Berdasarkan tingkat risiko yang diperoleh pada penelitian ini, kerentanan XSS ditempatkan sebagai prioritas mitigasi menengah karena memerlukan interaksi pengguna sebelum eksploitasi berhasil dilakukan.

3. Brute Force Login Page

Kerentanan brute force pada form login terjadi karena tidak adanya pembatasan terhadap jumlah percobaan login, sehingga penyerang dapat mencoba berbagai kombinasi username dan password secara berulang hingga menemukan kredensial yang valid. Kerentanan ini berpotensi mengakibatkan pengambilalihan akun pengguna atau administrator serta membuka peluang terjadinya kebocoran data dan serangan lanjutan. Upaya mitigasi yang dapat diterapkan meliputi pembatasan jumlah percobaan login, penerapan CAPTCHA setelah beberapa kali kegagalan autentikasi, penggunaan mekanisme rate limiting, serta penerapan kebijakan kata sandi yang kuat dan metode penyimpanan password yang aman menggunakan algoritma hashing yang sesuai.

Temuan ini memiliki kesesuaian dengan penelitian Alamsyah dkk. (2025) yang menunjukkan bahwa kelemahan mekanisme autentikasi dan pengelolaan hak akses masih menjadi salah satu

ISSN (Online) 2747-0563
Volume 6 Tahun 2026

sumber utama kerentanan aplikasi web. Penelitian tersebut merekomendasikan penerapan account lockout, rate limiting, serta mekanisme autentikasi yang lebih kuat sebagai langkah mitigasi[22]. Berdasarkan hasil pengujian pada penelitian ini, kerentanan brute force dikategorikan sebagai prioritas mitigasi tinggi karena berpotensi menyebabkan account takeover dan menjadi pintu masuk bagi serangan lanjutan.

4. Insufficient Session Expiration

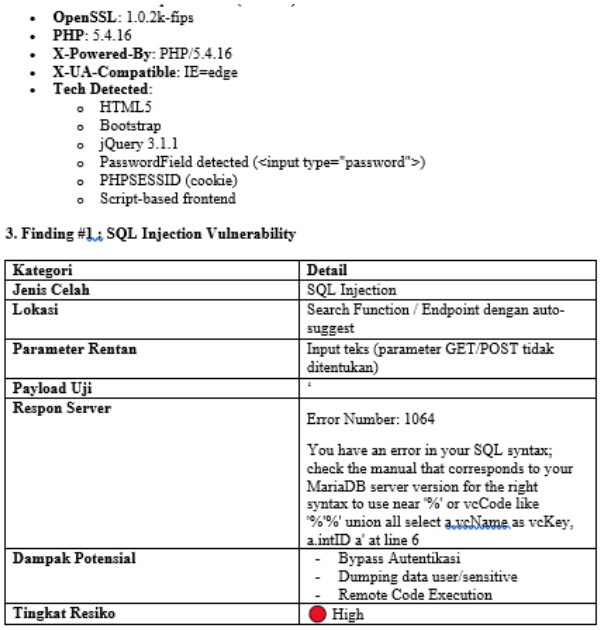
Solusi terhadap kelemahan manajemen sesi tersebut adalah dengan menerapkan regenerasi session identifier setiap kali terjadi proses login serta melakukan invalidasi session secara menyeluruh di sisi server saat logout, sehingga session lama tidak dapat digunakan kembali. Selain itu, session harus diikat secara ketat dengan identitas pengguna dan konteks autentikasi untuk mencegah penggunaan lintas akun. Kesimpulannya, meskipun aplikasi telah membatasi akses ke halaman terproteksi setelah logout, penggunaan session identifier yang tetap sama menunjukkan kelemahan pada desain manajemen sesi yang tidak sesuai dengan praktik keamanan yang baik dan berpotensi menimbulkan risiko keamanan apabila dikombinasikan dengan kerentanan lain.

Hasil penelitian ini juga sejalan dengan penelitian Alamsyah dkk. (2025) yang menempatkan session management sebagai salah satu aspek penting dalam pengujian keamanan aplikasi web menggunakan OWASP ZAP[22]. Selain itu, penelitian Reza dan Sutanto (2025) yang menerapkan metodologi PTES menjelaskan bahwa kelemahan pada manajemen sesi dapat meningkatkan peluang terjadinya session hijacking apabila dikombinasikan dengan kerentanan lain seperti XSS maupun pencurian cookie[23]. Oleh karena itu, kerentanan ini dikategorikan sebagai prioritas mitigasi tinggi dengan rekomendasi utama berupa regenerasi session identifier, session timeout otomatis, serta invalidasi sesi di sisi server.

G. *Reporting*

Tahap reporting merupakan fase akhir dalam penetration testing yang bertujuan untuk menyusun hasil pengujian secara sistematis dan terstruktur. Laporan disusun berdasarkan seluruh tahapan sebelumnya, mulai dari intelligence gathering hingga post-exploitation.

Proses ini diawali dengan pengumpulan dan verifikasi bukti pengujian, termasuk hasil pemindaian OWASP ZAP dan proof of concept (PoC), untuk memastikan setiap temuan valid dan bukan false positive. Selanjutnya, kerentanan diklasifikasikan berdasarkan Common Weakness Enumeration (CWE) guna memberikan pemahaman yang terstruktur terhadap risiko keamanan.



- **OpenSSL**: 1.0.2k-fips
- **PHP**: 5.4.16
- **X-Powered-By**: PHP/5.4.16
- **X-UA-Compatible**: IE=edge
- **Tech Detected**:
  - HTML5
  - Bootstrap
  - jQuery 3.1.1
  - PasswordField detected (<input type="password">)
  - PHPSESSID (cookie)
  - Script-based frontend

3. Finding #1 : SQL Injection Vulnerability

Kategori	Detail
Jenis Celah	SQL Injection
Lokasi	Search Function / Endpoint dengan auto-suggest
Parameter Rentan	Input teks (parameter GET/POST tidak ditentukan)
Payload Uji	'
Respon Server	Error Number: 1064 You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '%' or vcCode like '%'%' union all select a.vcName as vcKey, a.intID a' at line 6
Dampak Potensial	- Bypass Autentikasi - Dumping data user/sensitive - Remote Code Execution
Tingkat Resiko	High

Gbr. 9 contoh dokumen *Reporting*

Setiap temuan disajikan dalam bentuk deskripsi kerentanan, endpoint terdampak, metode pengujian, serta bukti eksploitasi. Selain itu, dilakukan analisis dampak berdasarkan aspek Confidentiality, Integrity, dan Availability (CIA) untuk menentukan tingkat risiko.

Hasil pengujian menunjukkan bahwa sistem ERP memiliki beberapa kerentanan dengan tingkat risiko tinggi hingga menengah, terutama pada validasi input, autentikasi, dan manajemen sesi. Oleh karena itu, diberikan rekomendasi mitigasi seperti penerapan validasi input, penggunaan parameterized query, pembatasan percobaan login, serta perbaikan manajemen sesi dan konfigurasi keamanan. Prioritas mitigasi yang diusulkan pada penelitian ini disajikan pada Tabel 14.

TABEL 14
PRIORITAS MITIGASI

Kerentanan	**Risk Level**	**Dampak Utama**	**Rekomendasi Mitigasi**
SQL Injection	*High*	Kebocoran dan manipulasi database	Prepared Statement, Parameterized Query, Validasi Input
Brute Force Login	*High*	Pengambilalihan akun	Rate Limiting, CAPTCHA, Account Lockout, MFA
Insufficient Session Expiration	*High*	Session Hijacking	Session Timeout, Regenerate Session ID, Logout Server-Side
Cross-Site Scripting (XSS)	*Medium*	Pencurian Cookie, Defacement	Input Validation, Output Encoding, CSP

ISSN (Online) 2747-0563
Volume 6 Tahun 2026

Missing Security Headers dan Cookie Configuration	*Low-Medium*	Mempermudah eksploitasi lain	Tambahkan CSP, HSTS, X-Content-Type-Options, HttpOnly, Secure, SameSite

Dengan demikian, tahap reporting tidak hanya menyajikan hasil pengujian, tetapi juga menjadi dasar dalam peningkatan keamanan sistem ERP secara berkelanjutan.

IV. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, sistem ERP XYZ milik PT XYZ masih memiliki sejumlah kerentanan keamanan yang berpotensi mengancam aspek kerahasiaan, integritas, dan ketersediaan data. Melalui penerapan metodologi Penetration Testing Execution Standard (PTES), berbagai celah keamanan berhasil diidentifikasi secara sistematis, mulai dari tahap pengumpulan informasi hingga eksploitasi terbatas. Beberapa kerentanan utama yang ditemukan meliputi SQL Injection, Cross-Site Scripting (XSS), brute force pada mekanisme login, serta kelemahan dalam manajemen sesi. Selain itu, terdapat pula kelemahan konfigurasi seperti tidak adanya header keamanan, pengelolaan cookie yang kurang tepat, serta paparan informasi sensitif melalui URL dan komentar sistem.

Hasil analisis menunjukkan bahwa sebagian kerentanan memiliki tingkat risiko yang signifikan dan berpotensi dieksploitasi oleh pihak tidak bertanggung jawab. Oleh karena itu, diperlukan penerapan langkah mitigasi yang tepat, seperti validasi dan sanitasi input, penguatan mekanisme autentikasi, perbaikan manajemen sesi, serta konfigurasi keamanan server sesuai praktik terbaik. Secara keseluruhan, penelitian ini memberikan gambaran kondisi keamanan sistem ERP XYZ sekaligus rekomendasi yang dapat dijadikan acuan untuk meningkatkan keamanan sistem, serta menjadi referensi bagi penelitian selanjutnya dalam evaluasi keamanan aplikasi ERP berbasis web.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada PT XYZ yang telah memberikan izin, kepercayaan, dan dukungan dalam pelaksanaan penelitian ini, khususnya dalam proses pengujian keamanan sistem ERP yang digunakan. Ucapan terima kasih juga disampaikan kepada pihak Universitas Pembangunan Nasional "Veteran" Jawa Timur atas dukungan akademik, fasilitas, serta lingkungan penelitian yang kondusif selama proses penelitian berlangsung. Penulis juga menyampaikan apresiasi kepada rekan-rekan yang telah memberikan masukan, diskusi ilmiah, serta dukungan selama proses pengumpulan data, analisis, dan penyusunan naskah penelitian. Selain itu, penulis juga mengapresiasi semua pihak yang telah membantu secara langsung maupun tidak langsung dalam penyusunan penelitian ini, sehingga penelitian dapat diselesaikan dengan baik.

REFERENSI

[1] Master in Management Information Systems, College of Business, Lamar University, Texas, USA. *et al.*, "A SYSTEMATIC REVIEW OF ERP IMPLEMENTATION STRATEGIES IN THE RETAIL INDUSTRY: INTEGRATION CHALLENGES, SUCCESS FACTORS, AND DIGITAL MATURITY MODELS," *AJSRI*, vol. 2, no. 2, pp. 135–165, Sep. 2023, doi: 10.63125/pfdm9g02.
[2] D. Howard Kass, "Hacker Attacks Target ERP, SAP and Oracle Business Applications," *Hacker Attacks Target ERP, SAP and Oracle Business Applications*, https://www.msspalert.com/, Oct. 09, 2019. [Online]. Available: https://www.msspalert.com/news/hackers-target-erp-applications
[3] The Straits Times, "More than 40 Indonesian agencies hit by cyberattack on data centres," *The Straits Times*, Jun. 2024, [Online]. Available: https://www.straitstimes.com/asia/se-asia/more-than-40-indonesian-agencies-hit-by-cyberattack-on-data-centres
[4] "Penetration Testing Execution Standard - the FAQ," *http://www.pentest-standard.org/*, http://www.pentest-standard.org/, Jan. 14, 2017. [Online]. Available: http://www.pentest-standard.org/index.php/FAQ
[5] "Pre-engagement," Penetration Testing Execution Standard. [Online]. Available: http://www.pentest-standard.org/index.php/Pre-engagement
[6] "Intelligence Gathering," Penetration Testing Execution Standard. [Online]. Available: http://www.pentest-standard.org/index.php/Intelligence_Gathering
[7] Wappalyzer, "Wappalyzer: Technology Profiler for Websites." 2026. [Online]. Available: https://www.wappalyzer.com/
[8] Nmap Project, "Nmap – Free Security Scanner and Network Exploration Tool." 2026. [Online]. Available: https://nmap.org/
[9] OJ, "gobuster." 2026. [Online]. Available: https://github.com/OJ/gobuster
[10] devanshbatham, "ParamSpider." 2026. [Online]. Available: https://github.com/devanshbatham/ParamSpider
[11] ProjectDiscovery, "Nuclei – Fast & Flexible Vulnerability Scanner by ProjectDiscovery." 2026. [Online]. Available: https://projectdiscovery.io/nuclei
[12] "Threat Modeling," Penetration Testing Execution Standard. [Online]. Available: http://www.pentest-standard.org/index.php/Threat_Modeling
[13] "Vulnerability Analysis," Penetration Testing Execution Standard. [Online]. Available: http://www.pentest-standard.org/index.php/Vulnerability_Analysis
[14] C. Kar Yee and M. F. Zolkipli, "Review on Confidentiality, Integrity and Availability in Information Security," *JICTIE*, vol. 8, no. 2, pp. 34–42, Jul. 2021, doi: 10.37134/jictie.vol8.2.4.2021.
[15] "Exploitation," Penetration Testing Execution Standard. [Online]. Available: http://www.pentest-standard.org/index.php/Exploitation
[16] "Post Exploitation," Penetration Testing Execution Standard. [Online]. Available: http://www.pentest-standard.org/index.php/Post_Exploitation
[17] "Reporting," Penetration Testing Execution Standard. [Online]. Available: http://www.pentest-standard.org/index.php/Reporting
[18] E. Susanto, Lady Antira, K. Kevin, E. Stanzah, and A. A. Majid, "MANAJEMEN KEAMANAN CYBER DI ERA DIGITAL," *JoBE*, vol. 11, no. 1, p. 23, Jun. 2023, doi: 10.46273/jobe.v11i1.365.
[19] S. Neupane, "Detecting and Mitigating SQL Injection Vulnerabilities in Web Applications," Jun. 07, 2025, *arXiv*: arXiv:2506.17245. doi: 10.48550/arXiv.2506.17245.
[20] A. Hannousse, S. Yahiouche, and M. C. Nait-Hamoud, "Twenty-two years since revealing cross-site scripting attacks: a systematic mapping and a comprehensive survey," *Computer Science Review*, vol. 52, p. 100634, May 2024, doi: 10.1016/j.cosrev.2024.100634.
[21] N. Nelmiawati and K. Dealova, "Analysis of Polyglot Obfuscation Techniques against ModSecurity in Preventing Cross-Site Scripting (XSS) and SQL Injection Attacks with Experimental Method," *J. Tek. Inform. (JUTIF)*, vol. 6, no. 4, pp. 2540–2549, Sep. 2025, doi: 10.52436/1.jutif.2025.6.4.5000.
[22] H. Alamsyah, T. Roynaldi, and T. U. Kalsum, "Analisa Sistem Keamanan Web Menggunakan OWASP Zed Attack Proxy (ZAP)," *j.inf.syst.int.*, vol. 5, no. 1, May 2025, doi: 10.53514/jco.v5i1.613.
[23] I. Sutanto and M. F. Reza, "Penerapan Pentesting pada EasyCart untuk Menghadapi Ancaman Keamanan Siber".